Planning Dextrous Robot Hand Grasps from Range Data, using Preshapes and Digit Trajectories

David Owen Wren

Ph.D.
University of Edinburgh
1997

Planning Dextrous Robot Hand Grasps from Range Data, using Preshapes and Digit Trajectories

David Owen Wren

Ph.D.
University of Edinburgh
1997

Abstract

Dextrous robot hands have many degrees of freedom. This enables the manipulation of objects between the digits of the dextrous hand but makes grasp planning substantially more complex than for parallel jaw grippers. Much of the work that addresses grasp planning for dextrous hands concentrates on the selection of contact sites to optimise stability criteria and ignores the kinematics of the hand. In more complete systems, the paradigm of preshaping has emerged as dominant. However, the criteria for the formation and placement of the preshapes have not been adequately examined, and the usefulness of the systems is therefore limited to grasping simple objects for which preshapes can be formed using coarse heuristics.

In this thesis a grasp metric based on stability and kinematic feasibility is introduced. The preshaping paradigm is extended to include consideration of the trajectories that the digits take during closure from preshape to final grasp. The resulting *grasp family* is dependent upon task requirements and is designed for a set of "ideal" object-hand configurations. The grasp family couples the degrees of freedom of the dextrous hand in an anthropomorphic manner; the resulting reduction in freedom makes the grasp planning less complex. Grasp families are fitted to real objects by optimisation of the grasp metric; this corresponds to fitting the real object-hand configuration as close to the ideal as possible. First, the *preshape aperture*, which defines the positions of the fingertips in the preshape, is found by optimisation of an approximation to the grasp metric (which makes simplifying assumptions about the digit trajectories and hand kinematics). Second, the full preshape kinematics and digit closure trajectories are calculated to optimise the full grasp metric.

Grasps are planned on object models built from laser striper range data from two viewpoints. A surface description of the object is used to prune the space of possible contact sites and to allow the accurate estimation of normals, which is required by the grasp metric to estimate the amount of friction required. A voxel description, built by ray-casting, is used to check for collisions between the object and the robot hand using an approximation to the Euclidean distance transform.

Results are shown in simulation for a 3-digit hand model, designed to be like a simplified human hand in terms of its size and functionality. There are clear extensions of the method to any dextrous hand with a single thumb opposing multiple fingers and several different hand models that could be used are described. Grasps are planned on a wide variety of curved and polyhedral objects.

Acknowledgements

Thank you to the EPSRC for funding my work through studentship no. 9231476X. Many people assisted me along the way. First of all, I would like to give a big thank you to my supervisor, Bob Fisher. He was always very approachable and offered sound advice. Thanks to all members, past and present, of the Machine Vision Unit at the University of Edinburgh: I couldn't have asked for a friendlier environment in which to work. Special thanks must go to my fellow student and friend Edvaldo Bispo, and to Andrew Fitzgibbon, who more than made up for his lack of code documentation with his willingness to help! Thanks to my officemate Xueyue Huang, who always provided some light relief down the stairs from the vision lab. Thanks also to all the folks at Amerinex Applied Imaging for the use of their facilities, not to mention their understanding, in the closing few weeks of my thesis work. Thanks go to my Dad for his advice about all matters academic, my Mum for her constant encouragement and support, my brother Andrew for keeping me down-to-earth and finally Kathy, who makes it all worthwhile.

Declaration

I hereby declare that I composed this thesis entirely myself and that it describes my own research.

David Wren
Edinburgh
June 1, 1997

Contents

List of Figures

Chapter 1

Robot Grasping with Dextrous Hands

1.1 Grasping, Manipulation and Dextrous Hands

Before the definition of the research problem tackled in this thesis, informal definitions of grasping, manipulation and dextrous hands are presented.

In the context of robotics, an object is **grasped** if it is held motionless in the presence of gravity by contact forces applied by a robot hand (which may be a dextrous hand, a parallel jaw gripper or a more specialised tool).

In a **precision** grasp, the object is contacted by only the final segment of each digit. In a **power** grasp, the hand wraps around the object such that more than one segment of each digit — and very possibly the palm — contact the object. The precision grasp allows manipulation, because the final segment of each digit can roll along the object surface. The power grasp gives greater stability, because the object is strongly constrained by multiple contacts.

Objects can be manipulated in two ways by a robot end-effector. In **simple manipulation**, the object remains fixed relative to the robot hand reference frame and any motion of the object is due only to movement of the robot arm. In **dextrous manipulation**, the object is moved relative to the hand reference frame, by motion of the digits.

A formal definition of a dextrous hand is given in Chapter 2. For now, the following

simple definition will suffice:

Definition 1 *A* **dextrous hand** *can perform dextrous manipulation,* i.e. *it can manipulate an object between its digits whilst keeping its wrist fixed.*

1.2 Why Use Dextrous Hands for Grasping?

Before proceeding further it is appropriate to consider why dextrous hands should be used for grasping objects. Planning grasps for a parallel jaw gripper is much more straightforward than for a dextrous hand because it involves fewer degrees of freedom (DOF): a typical dextrous hand/arm configuration has upwards of 15 DOF, whereas a parallel jaw gripper/arm combination typically has 6 or 7 DOF (1 DOF for the gripper plus 5 or 6 DOF for the robot arm). For example, [Roth & O'Hara 87] plan grasps with a parallel jaw gripper by superimposing a model of the gripper directly onto raw 2.5D data (acquired using a laser striper).

A parallel jaw gripper can be used to pick up a wide variety of objects, but cannot manipulate the object relative to the gripper reference frame, as a dextrous hand can. Even when the aim is just to acquire the object in a stable grasp, however, dextrous hands have big advantages over parallel jaw grippers:

- Due to the increased number of hand-object contacts, dextrous hand grasps can be more stable than parallel jaw gripper grasps. Put another way, the issue of stability becomes less crucial as the number (and area, in the case of soft-fingered hands) of hand-object contacts increases.
- A dextrous hand can grasp an object from a much wider range of wrist positions and approach directions than can a parallel jaw gripper. This is useful in a cluttered environment and facilitates easier integration of arm motion planning and grasp planning.

Both these also mean that we can use grasp planners which are sub-optimal, but fast and robust.

1.3 Issues in Robot Grasping

A grasp must be:

- Stable, subject to small arbitrary perturbations of object position and small arbitrary external forces.
- Kinematically feasible, *i.e.* the hand should be able to reach the contacts without collision with itself, the object or the environment.

Dextrous hands are, by their very nature, redundant. When mounted on a 6 DOF arm, the two most widely used dextrous hands, the Salisbury hand [Salisbury & Craig 82] and the Utah/MIT hand [Jacobsen *et al.* 86], have 15 and 22 DOF respectively. This means that, in general, there will be many possible sets of contact positions and forces that give a stable grasp, and many possible hand configurations that provide those contact positions. The grasp must then be optimised according to metrics such as:

- Friction required to execute a grasp.
- Contact angle between digit and object surface.
- Distance from joint limits.

Much of the literature on robot grasping has been concerned with the generation of stable grasps without reference to the kinematic feasibility of the resulting grasps, *e.g.* Nguyen used static mechanics to synthesize force closure grasps [Nguyen 87a] by maximising the leeway in contact placement and went on to make these grasps stable [Nguyen 87b] by modeling the contacts as virtual springs. This type of approach runs the risk of synthesising contact sets which require hand configurations that are out of the workspace of the hand or close to singularities.

In recent years the preshaping paradigm has been widely recognized as a useful way to resolve the kinematic redundancy of dextrous hands. A hand preshape is the digit posture adopted as the wrist moves towards the object. The grasp is then executed by placing the wrist into a position that encompasses the object, and then closing the

digits until they make contact with the object. For example, [Stansfield 91] preshaped a Salisbury hand using a knowledge-based system, and grasped polyhedral and simple curved objects. This type of approach relegates the stability of the grasp to a secondary issue, limiting its usefulness to either simple objects or whole-hand grasps (where the large contact area means that stability is not such a crucial issue). A key problem with the preshaping paradigm is that, by consideration of the preshape alone, it is not possible to estimate accurately the stability of some grasps: the trajectories of the digits during closure must be considered.

In this thesis, the concept of the **grasp family** is used to associate digit closure trajectories with hand preshape, thus enabling the stability of the grasp to be analysed at the same time as the kinematic quality. A **grasp family** consists of a preshape and a set of digit closure trajectories, from which a grasp can be formed without movement of the robot wrist. The preshape is a prescribed hand configuration and the digit closure trajectories are the motions of the fingertips after the preshape is formed and the wrist position has been fixed.

Grasp families constrain the range of possible digit movements whilst still allowing a sufficient number of DOF to be able to cope with a wide range of objects: the digit trajectories are specified, but each digit can be stopped at any point along its trajectory by contact with the object. This approach reduces the complexity of the problem whilst preserving the flexibility of a dextrous hand. Rather than viewing a dextrous hand as a completely general device, it is viewed as a set of flexible tools, where each grasp family is a different "tool".

1.4 Thesis Topic

The research problem tackled in this thesis is the planning of dextrous hand *precision* grasps on objects of *arbitrary shape*, such that they are stable, kinematically feasible and satisfy task requirements. Real input data is used, in the form of dense laser striper range images from two known viewpoints. Grasps are planned for a 3-digit hand model known as the simplified anthropomorphic model (SAM).

The following basic assumptions are made:

- The object to be grasped is localised and lies on a planar surface.
- The two views of the object are accurately registered (see Chapter 4).
- The object is rigid.
- The effect of gravity is negligible (see Chapter 3).
- The dextrous hand can be positioned accurately with respect to the object.
- Grasp forces can be accurately reproduced by the dextrous hand, or a control mechanism can compensate for errors in force application.
- Contacts between the hand and object are frictional.
- Equilibrium grasps can be made stable through use of soft digits and a suitable control mechanism (see Chapter 3).

It is shown that using task-specific grasp families, fingertip grasps of reasonable quality can be planned for a range of polyhedral and curved objects. The optimality of a grasp planned in this way depends on how close the actual object geometry is to the "ideal" contact geometry of the grasp family that is being used: this tends to favour polyhedral objects and symmetrical curved objects, but good grasps are planned on irregular curved objects because grasp families provide a way of satisfying task requirements whilst allowing the grasp to deform to different object shapes. The algorithm is readily adaptable to any hand consisting of a thumb opposing an arbitrary number of fingers.

A surface model of the object provides high quality geometric information about the object to be grasped. It is also used to prune the search space for feasible grasps, but this becomes less useful as the complexity of the object increases. A distance-transformed voxel model proves useful for collision checking.

The principal contributions of the thesis are:

- A surface and volumetric object model tailored to robot grasping.
- The notion of a grasp family, which associates digit trajectory with a preshape.

- Formulation and use of a grasp metric, based on both kinematic and mechanical considerations, for the formation and placement of the preshape.
- A grasp planning algorithm that uses grasp families to embody task requirements and reduce the DOF of the problem.

The grasp planning algorithm is:

- Capable of planning grasps on a wide variety of object shapes, both curved and polyhedral, regular and irregular.
- Readily adaptable to different hand designs — no "hand-specific" inverse kinematics are used.[1]
- Fast — with un-optimised code, on a Sparc 10 it takes 45 seconds to plan each grasp shown in Figure 1.2.

Secondary contributions of the thesis include a constrained Euclidean distance transform for collision and contact checking and an analysis of the inverse kinematics and singularities of the grasp families.

1.5 Overview of thesis

Figure 1.1 shows the general approach taken in this thesis. Figure 1.2 shows an example grasp on an egg, lying on its side. Figures 1.2(a) and (b) show the laser striper range images of the egg taken from $\pm 45°$ to the horizontal. Figure 1.2(c) shows the surface model of the egg and Figure 1.2(d) shows the voxel model of the egg. Figure 1.2(e) shows a tip grasp, in which contact is made with the end of the fingertips (as would be used if precise manipulation between the digits is required). Figure 1.2(f) shows a lateral grasp, in which contact is made with the inside of the last segment of each digit (as would be used if more security is required).

[1] i.e. No inverse kinematic algorithm that relies on properties such as the symmetry of the hand is used — see [Koehler & Donath 88] for an example of an inverse kinematics algorithm that is very specific to the hand for which it is used.

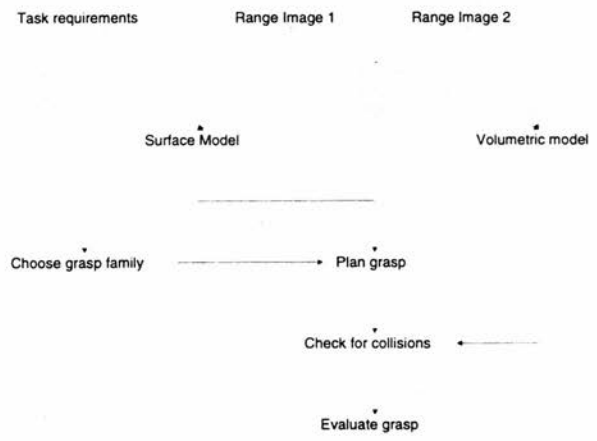


Figure 1.1: The grasp planning system.

1.5.1 Chapter 1: Robot Grasping with Dextrous Hands

1.5.2 Chapter 2: Literature Review

Basic grasping terminology is introduced and a short overview of grasp mechanics is given. A literature review considers previous work in grasp planning, specifically mechanics, kinematics, object representations and complete systems. Noted are a lack of formal criteria for the formation and placement of preshapes, little consideration of digit closure once the preshape has been formed, and the small number of task-specific object models formed from real data.

1.5.3 Chapter 3: Grasp Families

The hand modelling system is described and models of four hands are given (in order of complexity): a parallel jaw gripper, the Salisbury hand, the Simplified Anthropomorphic Model (SAM) and the human hand. The SAM is used throughout the rest of the thesis. Grasp families (see Section 1.3) are introduced and explained and three grasp families, for tip, lateral and manipulation grasps, are defined. The inverse kinematics of the grasp families are analysed for singularities.

1.5.4 Chapter 4: Acquisition of the Object Model

Much research in robot grasping assumes complete geometric knowledge of the object to be grasped and few task-oriented object representations have been used (notable exceptions are [Stansfield 91] and [Bard *et al.* 95]). In this chapter, surface and volumetric object models for grasp planning are acquired from two views of laser striper range data (see Figures 1.2(a) and (b)). The surface model (see Figure 1.2(c)) provides the contact geometry information necessary to plan fingertip grasps, and the volumetric model (see Figure 1.2(d)) provides the spatial occupancy information for collision checking. Two views of the object are taken. The surface model consists of planes and quadrics, fitted using a surface growing algorithm described by [Trucco & Fisher 95] and derived from [Besl 88]. The volumetric model is a spatial occupancy array of voxels constructed by ray-tracing, called a voxmap. An approximation to the Euclidean distance transform is used for very fast collision and contact checking.

1.5.5 Chapter 5: A Grasp Planning Algorithm

A grasp quality metric is introduced. This metric embodies both the mechanical and kinematic requirements of a grasp for a given task. The complexity of the grasp planning problem is discussed, and two possible approaches — the forward kinematic and inverse kinematic approaches — are compared. A forward kinematic grasp planning algorithm for the SAM is described. The algorithm takes as input the object model described in Chapter 4 and uses grasp families that embody task requirements and constrain the DOF of the problem. The grasp is then checked for collisions using the volumetric model described in Chapter 5. A complex polyhedral object is used to illustrate each phase of the algorithm.

1.5.6 Chapter 6: Results

The behaviour of the grasp planning algorithm is analysed. The setting of parameters is examined empirically. Grasps are planned on the polyhedron from Chapter 5, a cuboid, an egg, a tube-like curved object and a toy rabbit. Problems with the grasp planning algorithm are highlighted and discussed. The grasp quality is compared

between objects. Figure 1.2(e) shows the "tip-distal" grasp planned on the egg and Figure 1.2(f) the "lateral-proximal" grasp planned on the same object.

1.5.7 Chapter 7: Conclusions

The weaknesses of the algorithm that were identified in Chapter 6 are summarised and possible remedies are suggested. The usefulness of the object models and grasp families are assessed and avenues for future research are outlined.

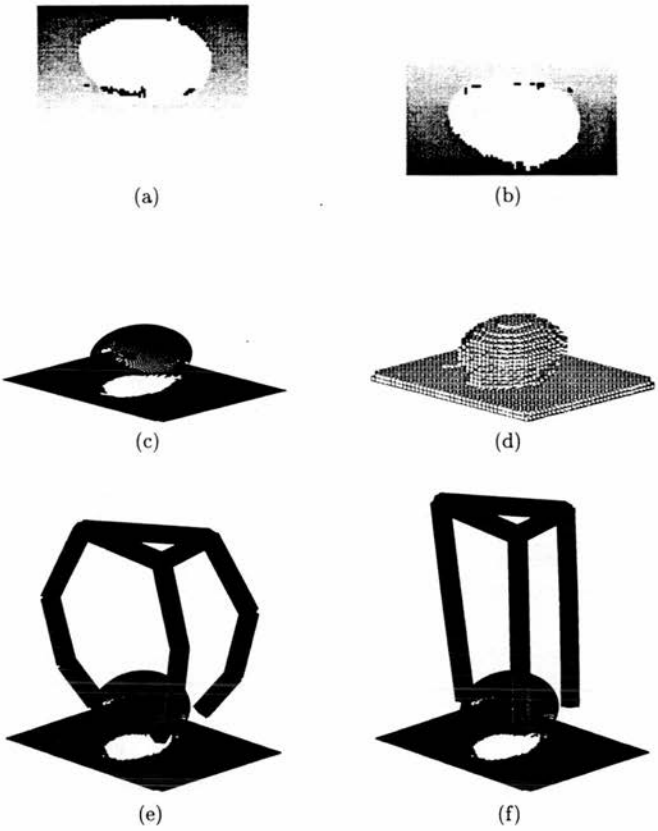


Figure 1.2: Example grasp on an egg with the simplified anthropomorphic model: (a) range image at $+45°$ to the horizontal, (b) range image at $-45°$ to the horizontal, (c) surface model, (d) voxel model, (e) tip grasp, (f) lateral grasp.

Chapter 2

Literature Review

In this chapter the grasping literature is reviewed, and the concepts essential to the understanding of the rest of the thesis are introduced.

2.1 Basic Terminology

In the grasping literature, it is common to refer to a "finger" when meaning either thumb or finger. Often it is useful to differentiate between finger and thumb, because of their different designs and functions, so in this thesis **digit** is used to mean fingers or thumb, and **finger** and **thumb** are used in their stricter sense. However, **fingertip** is used in preference to the rather clumsy "digit-tip". Fingertip refers to the very end of the digit.

Each segment (phalange, in biological parlance) of a digit is referred to as a **link**. A 3-link digit consists of **proximal, middle**, and **distal links** respectively. A 2-link digit consists of **proximal** and **distal links**.

Curling motion of a digit is called **flexion**. Strictly speaking, motion towards the palm is flexion, and motion away is **extension**, but in this thesis flexion is usually used to refer to motion in both directions. Motion of a digit in a plane perpendicular to this is referred to as **abduction**. Strictly speaking, motion of the fingers away from one another is abduction, motion of the fingers towards one another is **adduction**, but abduction is used to refer to motion in both directions.

Joints within a digit are labelled in a similar manner to the links. In a human digit,

the joints are labelled proximal, middle and distal. The digit can both abduct and flex about the proximal joint (the knuckle). Such ball joints are not always possible in robotics, due to tendon-routing complications. The Utah/MIT hand, for instance, has four joints: three for flexion, referred to as the **proximal, middle** and **distal joints of flexion**, and one for abduction, referred to simply as the **joint of abduction**.

2.2 Types of grasp

[Napier 56] introduces the concept of *precision* and *power* grasps. In a precision grasp, the object is contacted by only the distal link of each digit. In a power grasp, the hand wraps around the object such that more than one link of each digit — and possibly the palm — contact the object. The precision grasp allows maximum manipulation, as the distal link of each digit can roll along the object surface. The power grasp allows maximum stability, as the object is strongly constrained by multiple contacts and/or contacts of large extent (such as the palm of the hand).

[Lyons 85] defines three types of grasp, in order of increasing power and decreasing precision: *precision, lateral, encompass.* In a precision grasp, the object is held only at the fingertips. In a lateral grasp, the object is held by the inside of the distal link of each digit. In an encompass grasp, the inside of every link of each digit, plus the palm, are touching the object (often creating the form closure grasp described in Section 2.4.1). Lyons' precision and lateral grasps are equivalent to Napier's precision grasp; Lyons' encompass grasp is equivalent to Napier's power grasp. The taxonomy is somewhat flawed, in that it mixes task-level description (precision grasp) and contact description (lateral and encompass grasps) at the same level.

[Stansfield 91] uses a similar taxonomy to that of Lyons, with application to the 3-fingered Salisbury Hand, referring to them as *pinch*, *grip* and *wrap* instead of precision, lateral and encompass respectively. However, the *pinch* grasp uses only two digits.

In this thesis the following terminology is used (in order of increasing power and decreasing precision):

1. **tip** — contact with the fingertips

e.g. holding a dart ready to throw.

2. **lateral** — contact with the length of the distal links
   e.g. grasping a computer mouse by the sides.

3. **encompass** — contact with links other than distal links, and/or the palm
   e.g. holding a baseball bat, securely grasping a key to open a Yale-lock [1]

This is similar to Lyons' terminology; the difference is that the most precise grasp is referred to as a *tip* grasp. This means that the taxonomy terminology is compatible with that of [Napier 56]: *tip* and *lateral* grasps are referred to as *precision* grasps and *encompass* grasps are also referred to as *power* grasps. Napier's classification can be refined somewhat: *precision* grasps have contacts at distal links only and emphasise manipulability, whereas *power* grasps have contacts at links other than distal links, possibly including the palm, and emphasise security. Figure 2.1 shows the taxonomy, with examples of each type of grasp for a two-fingered hand.

In this thesis, tip and lateral grasps are planned, along with a third type of precision grasp known as the manipulation grasp (see Chapter 3).

2.3 Basics of Grasping

In this section the basics of grasping are reviewed: the mechanics of grasps, the design of dextrous hands and the inverse kinematics of dextrous hands.

2.3.1 Mechanics

Consider N frictional contacts on an object surface, as shown in Figure 2.2. The object geometry at contact i is described by a position vector $\mathbf{r_i}$ with respect to an arbitrary origin, and a unit normal $\hat{\mathbf{n}}_\mathbf{i}$. Contact i can exert a force $\mathbf{f_i}$ ($| \mathbf{f_i} | < f_{max}$) lying within the friction cone at that contact, *i.e.* :

[1] There are only two regions of contact — the inside of the distal link of the thumb and the side of the middle link of the first finger — but since a non-distal link is used and the emphasis is on security rather than stability, this is classed as an encompass grasp.

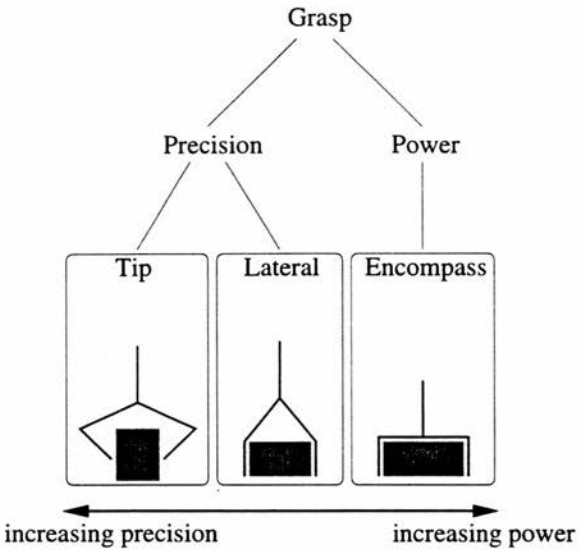


Figure 2.1: Grasp taxonomy: tip, lateral and encompass grasps

$$\cos^{-1}\left(\frac{\mathbf{f_i}\cdot\hat{\mathbf{n}}_\mathbf{i}}{|\,\mathbf{f_i}\,|}\right) < \tan^{-1}\mu \tag{2.1}$$

See Figure 2.5(a) for an example of such a friction cone. The grasp contact mechanics are described in terms of wrenches. A **wrench** is a vector of force and moment, and has 6 elements in 3D (3 of force and 3 of moment) and 3 elements in 2D (2 of force and 1 of moment). Contact i exerts a wrench $\mathbf{w_i}$, where:

$$\mathbf{w_i} = \begin{pmatrix} \mathbf{f_i} \\ \mathbf{r_i} \times \mathbf{f_i} \end{pmatrix} \tag{2.2}$$

The total wrench on the object due to the grasp is

$$\mathbf{W_g} = \sum_{i=1}^{N} \mathbf{w_i} \tag{2.3}$$

and the wrench on the object due to external forces (such as gravity) is $\mathbf{W_e}$. For

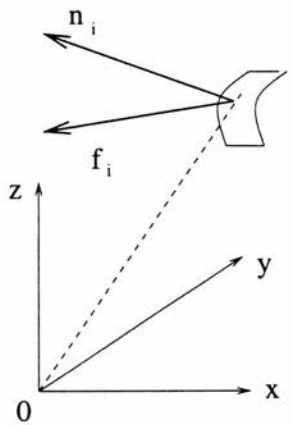


Figure 2.2: N frictional contacts on an object surface. The object geometry at contact i is described by a position vector $\mathbf{r_i}$ with respect to an arbitrary origin, and a unit normal $\hat{\mathbf{n}}_\mathbf{i}$. Contact i can exert a force $\mathbf{f_i}$ ($\mid \mathbf{f_i} \mid < f_{max}$) lying within a friction cone at that contact.

equilibrium,

$$\mathbf{W_g} + \mathbf{W_e} = \mathbf{0} \qquad \textbf{and} \qquad \exists \mathbf{w_i} \neq \mathbf{0} \tag{2.4}$$

Often gravity is neglected, to give $\mathbf{W_e} = \mathbf{0}$, in which case the constraint that at least one wrench must be non-zero prevents trivial solutions. Figure 2.3 shows an example of a 3-contact grasp in equilibrium with zero gravity. For a 3-contact grasp to be in equilibrium, all forces must be coplanar, and coincident at a common point.

Commonly, as in [Mason & Salisbury 85], each force is in a fixed direction. This is possible if there are frictionless contacts, or if the frictional contacts have been decomposed into sets of constituent unit wrenches. Figure 2.4 shows a friction cone in 2D and the equivalent pair of unit wrenches, $\hat{\mathbf{a}}_{\mathbf{i1}}$ and $\hat{\mathbf{a}}_{\mathbf{i2}}$. The convex hull of these two wrenches defines the range of possible contact forces. The contact force is given by:

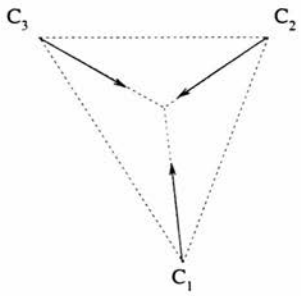


Figure 2.3: 3-contact grasp in equilibrium with zero gravity: forces coplanar and coincident at a common point

$$\mathbf{f_i} = \mid \mathbf{f_i} \mid \begin{pmatrix} \sin\theta \\ \cos\theta \end{pmatrix} = f_{i1}\hat{\mathbf{a}}_{\mathbf{i1}} + f_{i2}\hat{\mathbf{a}}_{\mathbf{i2}} \qquad f_{i1} \geq 0, f_{i2} \geq 0 \tag{2.5}$$

In 3D, a set of such wrenches can only approximate a friction cone. Figure 2.5 shows a friction cone in 3D and a set of four fixed-direction wrenches, the convex hull of which approximates the circular-base friction cone by a square-base friction cone. Increasing the number of constituent wrenches increases the accuracy of the approximation. If the i^{th} contact force is assumed to composed of M_i constituent forces:

$$\mathbf{f_i} = \mid \mathbf{f_i} \mid \begin{pmatrix} \sin\theta\cos\psi \\ \sin\theta\sin\psi \\ \cos\theta \end{pmatrix} \approx \sum_{j=1}^{M_i} f_{ij}\hat{\mathbf{a}}_{\mathbf{ij}} \qquad f_{ij} \geq 0 \tag{2.6}$$

where N is the number of constituent wrenches.

After such a decomposition, each component can be treated as a contact force in its own right:

$$\mathbf{f_{ij}} = f_{ij}\hat{\mathbf{a}}_{\mathbf{ij}} \qquad f_{ij} \geq 0 \tag{2.7}$$

The notation can be simplified by dropping one layer of indices:

$$\mathbf{f_i} = f_i\hat{\mathbf{a}}_{\mathbf{i}} \qquad f_i \geq 0, \quad i = 1 \text{ to } P, \quad P = \sum_{k=1}^{N} M_k \tag{2.8}$$

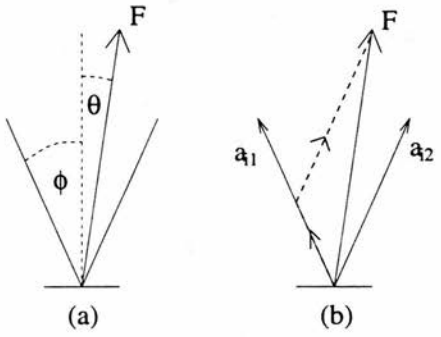


Figure 2.4: Planar friction cone, with friction angle $\phi = \tan^{-1}\mu$: (a) actual cone, (b) the exactly equivalent convex hull of two wrench vectors

where $\hat{\mathbf{a}}_i$, the wrench direction, is a constant and f_i is the contact force magnitude. If the grasp is frictionless, $\hat{\mathbf{a}}_i = \hat{\mathbf{n}}_i$ This formulation leads to an easier mathematical analysis of the equilibrium conditions and, as shall be seen in Section 2.4.1, is convenient for the calculation of some stability metrics. The wrench at contact i becomes:

$$\mathbf{w}_i = \begin{pmatrix} f_i\hat{\mathbf{a}}_i \\ \mathbf{r}_i \times f_i\hat{\mathbf{a}}_i \end{pmatrix} = f_i \begin{pmatrix} \hat{\mathbf{a}}_i \\ \mathbf{r}_i \times \hat{\mathbf{a}}_i \end{pmatrix} = f_i\hat{\mathbf{w}}_i \qquad f_i \geq 0 \qquad (2.9)$$

where $\hat{\mathbf{a}}_i$ and $\hat{\mathbf{w}}_i$ are constants.

The total wrench due to contacts is then given by:

$$\mathbf{W_g} = [\hat{\mathbf{w}}_0\hat{\mathbf{w}}_1 \ldots \hat{\mathbf{w}}_P] \begin{bmatrix} f_0 \\ f_1 \\ \cdot \\ \cdot \\ f_P \end{bmatrix} = \mathbf{W}\,\mathbf{f} \qquad f_i \geq 0 \qquad (2.10)$$

where $\mathbf{W}$ is the *grasp matrix* and $\mathbf{f}(i) = f_i$.

The solution, if one exists, is then given by

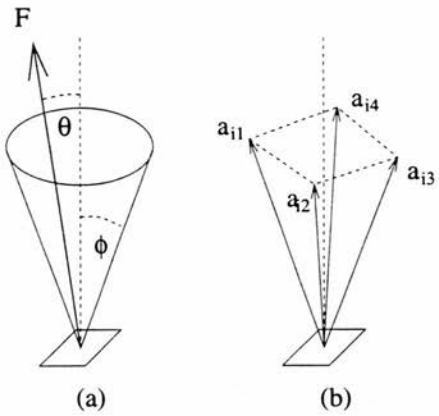


Figure 2.5: 3D Friction cone, with friction angle $\phi = \tan^{-1}\mu$: (a) actual cone, (b) the *approximation* by the convex hull of wrench vectors

$$\mathbf{f} = \mathbf{f^p} + \left[\mathbf{f_0^h} \mathbf{f_1^h} \ldots \mathbf{f_R^h}\right] \begin{bmatrix} v_0 \\ v_1 \\ \cdot \\ \cdot \\ v_R \end{bmatrix} \qquad 0 \leq R \leq P, \quad \mathbf{f}(i) \geq 0 \tag{2.11}$$

where $\mathbf{f^p}$ is the particular solution and $\{\mathbf{f_i^h}\}$ is the set of homogeneous solutions. $\mathbf{f_i^h}$ lie in the null space of the grasp matrix $\mathbf{W}$ and are basis vectors for the range of *internal forces*. An internal force is a contact force that can be applied without disturbing the equilibrium of the grasp [Mason & Salisbury 85]. Internal forces exist if the solution to the equilibrium equation is *underconstrained*.

The internal forces are controlled by the variables v_i. The *unisense* constraint, $\mathbf{f}(i) \geq 0$, is a most important part of Equation (2.11): v_i must be chosen such that this is true.

From Equations (2.4) and (2.10):

$$\mathbf{f} = -\mathbf{W}^{\#}\mathbf{W_e} + \left(\mathbf{I} - \mathbf{W}^{\#}\mathbf{W}\right)\mathbf{u} \qquad \mathbf{f(i)} \geq \mathbf{0} \tag{2.12}$$

where $\mathbf{W}^{\#}$ is the *pseudo-inverse* [2] of $\mathbf{W}$, $\mathbf{I}$ is the identity matrix, $\text{rank}(\mathbf{I} - \mathbf{W}\mathbf{W}^{\#}) = R$

[2] See Section 2.3.3 for a discussion of how best to calculate the pseudo-inverse of a matrix, this time applied to local inverse kinematics.

and $\mathbf{W_e}$ is the external wrench. The first term (obtained when $\mathbf{u} = \mathbf{0}$) gives the least squares solution of minimum norm (*i.e.* minimum $|\mathbf{f}|$). For an equilibrium grasp, a least squares solution will be the exact solution, and corresponds to the particular solution $\mathbf{f_p}$ in Equation (2.11). The second term is the null space of the grasp matrix $\mathbf{W}$, and corresponds to the range of possible internal forces (the homogeneous solutions $\mathbf{f_i^h}$ in Equation (2.11). It is tempting to think that the least-squares solution is a good solution to pick for the equilibrium forces because it minimises $\mathbf{f}$. This is wrong on two counts. First of all, the least-squares solution may not satisfy the constraint that $\mathbf{f}(i) \geq 0$. Secondly, there may be stronger requirements than just minimising $\mathbf{f}$, such as minimising the friction required for the grasp. Equation (2.11) is therefore best regarded as a description of the space of possible equilibrium forces.

As pointed out in [Yoshikawa & Nagai 90] the formulation of Equation (2.12) is of limited use because the internal force term is a rank-deficient matrix, *i.e.* the vector basis specifying the internal forces is redundant. For a 3-contact grasp, there are 6 null space basis vectors in the formulation of Equation (2.12), yet the rank of the internal force matrix is only 3 (*i.e.* the space of possible internal forces is 3 dimensional). [Yoshikawa & Nagai 90] therefore proposes using a different set of basis vectors: the vectors lying along the lines joining the contact points. See Figure 2.6 for an illustration of this. The basis vectors are $\{ \mathbf{e_{12}}, \mathbf{e_{23}}, \mathbf{e_{31}} \}$, with $\mathbf{e_{ij}} = -\mathbf{e_{ji}}$. This is the same set of basis vectors as [Mason & Salisbury 85] use. There are other ways of representing the null space of a 3-contact grasp. One uses the point where the contact forces meet, called the **focus of internal force**. This point lies in the plane of the contacts, and by variation of its position within the plane plus a force scaling factor, the null space can be spanned. The second method finds the basis vectors of the null space of the grasp matrix by singular value decomposition, which is also useful in the analysis of the local inverse kinematics — see Section 2.3.3.

When grasping, it is the ability of a dextrous hand to provide a range of internal forces, and hence optimise some grasp metric, that distinguishes it from a parallel jaw gripper. A parallel jaw gripper can only control the magnitude of internal forces, not the direction (and hence the focus). An ability to select the focus of internal forces is herein referred to as **dextrous grasping**.

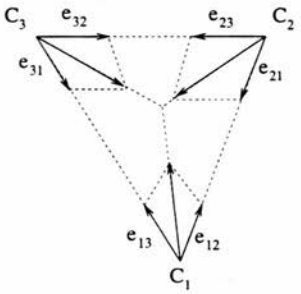


Figure 2.6: 3-contact grasp in equilibrium: all internal forces can be constructed from the set of basis vectors $\{ \mathbf{e_{12}}, \mathbf{e_{23}}, \mathbf{e_{31}} \}$ $(\mathbf{e_{ij}} = -\mathbf{e_{ji}})$

Typically, a dextrous hand should be able to control the internal forces by actuation of the digits only (while keeping the wrist fixed). A sufficient (though not necessary) condition for this is that each contact can exert an arbitrary force. In this case, each digit should have three joints. Figure 2.7 shows the kinematics of one such digit — the design used in the Salisbury hand (the two fingers and thumb are identical). It can abduct about the first joint, and flex about the second and third joints.

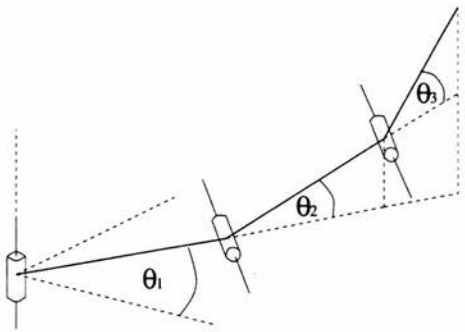


Figure 2.7: Salisbury hand digit: *abducts* about first joint, *flexes* about second and third joints.

Such a digit can exert force in an arbitrary direction, except at the edges of its workspace, such as when it is fully flexed, and certain points interior to the workspace, such as when the links are folded over one another. At such points, the Jacobian matrix becomes singular and hence non-invertible; these points are referred to as *singularities*. Digits that have fewer than three degrees of freedom will not be capable of exerting an arbitrary force from a fixed wrist position, and therefore the range of possible internal forces is limited, and the usefulness for dextrous grasping (and manipulation) is likewise limited. However, such hands may still be used to obtain stability above and beyond that offered by parallel jaw grippers.

2.3.2 Dextrous/multifingered Hands

A large number of dextrous/multifingered hands are now in existence. All the hands described in this section, with the exception of the Belgrade/USC hand, are actuated by tendons. This means that the large array of motors can be placed some distance from the hand, but adds friction at the pulleys, making it difficult to accurately model the behaviour of a hand. The Salisbury hand [Salisbury & Craig 82] (formerly known as the Stanford/JPL hand) was one of the first truly "dextrous" hands. This hand has three digits: one "thumb" opposing two "fingers". Each digit is identical, consisting of 3 joints. Each digit can *abduct* at one joint and *flex* at two joints. The design and placement of the digits were optimised, according to criteria based on possible restraint of a sample object and workspace. See Chapter 3 for a model of this hand.

Soon after the Salisbury hand, the Utah/MIT hand [Jacobsen *et al.* 86] was produced. This hand has four digits — three fingers opposing a thumb — and is more anthropomorphic in its design than the Salisbury hand. Each digit has four joints — one of abduction and three of flexion. The design of both the Salisbury and Utah/MIT hands differ from the human hand in that the joint of abduction is at a different part of the digit to the proximal joint of flexion, due to tendon-routing complications.

There are now a wide variety of dextrous hands in existence.

The Anthrobot-2 hand [Ali *et al.* 93], as the name suggests, is an attempt to produce as anthropomorphic a hand as possible. This has five digits, each with four joints,

arranged very close the the arrangement of the human hand. Unlike the Salisbury and Utah/MIT hands, the abduction joints are coincident with the proximal joints of flexion, as with the human hand. The middle and distal joints of each finger are coupled, so the hand is controlled by sixteen motors.

"Cog", the humanoid construction project at MIT, has a multi-fingered hand [Matsuoka 95]. This hand has four digits and is roughly anthropomorphic in appearance, with three fingers opposing a thumb. Each finger has two joints, which are coupled together, and the thumb has one joint. The hand is therefore actuated by four motors in total. The motors are placed in the palm of the hand, making this one of the more self-contained dextrous hands. The fingers are not capable of abduction.

The USC/Belgrade hand [Bekey *et al.* 90] grew out of the Belgrade prosthetic hand of the 1960s. It consists of four fingers and a single thumb. Each finger consists of three coupled joints. One motor drives two adjacent fingers; they are connected via a rocker arm which allows for some shape adaptation during a grasp whilst keeping the number of actuators low. The thumb consists of two joints, each of which is controlled by a separate motor. The four motors are mounted in the palm of the hand. As with the Cog hand, the fingers are not capable of abduction. Note that the fingers are not actuated by tendons, but rather by a system of mechanical linkages. This increases reliability of the hand, since tendons are liable to break after prolonged use.

The Beijing/Simon Fraser University hand [Yang *et al.* 95] has two fingers and an opposing thumb. It is available in two configurations. In one, the thumb directly opposes the two fingers; in the other the thumb is somewhat to the side, in a more anthropomorphic configuration. Each digit consists of four joints, three for flexion and one for abduction.

Table 2.1 summarises the different hand designs. It shows the number of fingers, the number of joints of flexion (with the number of independent DOF in brackets), the number of joints of abduction, and the total DOF. Each finger in the USC/Belgrade hand is listed as having $\frac{1}{2}$ a motor because one motor actuates two fingers, though a rocker arm provides some decoupling of the fingers on contact. Note also that in the Anthrobot-2 hand, the proximal joint of flexion coincides with the joint of abduction

(they are listed separately).

	Fingers			Thumb		
	Number	Flexion	Abduction	Flexion	Abduction	Total DOF
Cog	3	2(1)	0	1(1)	0	4
USC/Belgrade	4	3($\frac{1}{2}$)	0	1(1)	1	4
Salisbury	2	2(2)	1	2(2)	1	9
Beijing/SFU	2	3(3)	1	3(3)	1	12
Anthrobot-2	4	3(2)	1	3(3)	1	16
Utah/MIT	3	3(3)	1	3(3)	1	16

Table 2.1: DOF for different dextrous hands, showing number of fingers, number of joints of flexion (with the number of independent DOF in brackets), number of joints of abduction, and the total DOF. Each finger in the USC/Belgrade hand is listed as having $\frac{1}{2}$ a motor because one motor actuates two fingers, though a rocker arm provides some decoupling of the fingers on contact. Note also that in the Anthrobot-2 hand, the proximal joint of flexion coincides with the joint of abduction (they are listed separately).

The Salisbury hand and Utah/MIT hand have been used widely within the research community. The Utah/MIT hand has been demonstrated to be capable of dextrous manipulation, as in [Michelman & Allen 94], where it was used to take the top off a childproof medicine bottle. The Salisbury hand has been used principally for grasping objects, both with precision and power grasps (*e.g.* [Stansfield 91], [Bard *et al.* 95]), though [Fearing 86a] used it to twirl a block. The use of the other hands has been limited mainly to within the research groups that developed them. The Beijing/SFU hand and the Anthrobot-2 hand should be capable of dextrous manipulation, though little work has been done on this to date.

The Cog and USC/Belgrade hands are not capable of dextrous manipulation because they cannot exert arbitrary force at the fingertip, due to the absence of an actuated joint of abduction. The usefulness of these hands lies in their ability to grasp objects more stably than a parallel jaw gripper. Indeed, the Belgrade/USC hand recognises two important points. Firstly, passive compliance can be used to make the digits adapt to object shape. Secondly, only the thumb needs more than one actuated DOF, which can be used to select which finger(s) to oppose. Hands such as the Utah/MIT, Salisbury and Beijing/SFU hand are more complex than is necessary for simple grasping. If we are performing grasping operations with these hands, certain DOF can be coupled

together, to make the grasp planning more tractable.

2.3.3 Inverse kinematics of dextrous hands

Dextrous hands can exert a range of internal forces on the object, due to the large number of degrees of freedom (DOF) they possess. However, this means that the inverse kinematics problem is correspondingly complex. Inverse kinematics problems can be divided into the local and the global. Local inverse kinematics is the study of what joint motions should be used to produce a given small motion of the fingertips. Included is the study of singularities of the workspace — points at which an arbitrary motion is not allowed. Global inverse kinematics is the study of how to synthesise a hand configuration with a desired set of fingertip positions and/or orientations, subject to some metric, such as optimisation of joint values and collision avoidance.

Global inverse kinematics

The **target configuration** is a set of desired positions and orientations for one or more specified links of the hand. The global inverse kinematics problem can be posed thus:

Given a target configuration calculate the (set of) possible hand configuration that minimises the distance (according to some metric) to the target configuration.

When planning precision grasps, the target configuration is specified for the distal link of each digit, whereas a power grasp might specify a target configuration that included, for instance, the palm.

It is not always possible to access the target configuration. Fixing the DOF of the wrist, a hand with 3 DOF per digit will be able to reach an arbitrary set of distal link positions within some limited workspace. This workspace is limited by the length of the digits and the constraint that digits must not collide. However, to reach an arbitrary set of distal link positions *and* orientations, 6 DOF per digit are required. If the distal links are cylindrical, this may reasonably be reduced to 5 DOF per digit by neglecting the "roll" of the distal link - 3 DOF for position and only 2 DOF for orientation. No such dextrous hand exists. The wrist DOF must therefore be utilised to produce

the desired distal link configurations, and this introduces interdependence between the digits. In order to reach an arbitrary target configuration (within a restricted workspace), a dextrous hand with N digits requires a minimum of $5N$ DOF in the hand, arm and wrist, with no more than 5 DOF in any one digit.

The Salisbury hand has three digits, each with 3 DOF. This gives 15 DOF when mounted on a 6 DOF wrist, which yields the minimum number of DOF required to access an arbitrary target configuration. The Utah/MIT hand has 4 digits with 4 DOF each. This gives 22 DOF when mounted on a 6 DOF wrist, which yields two more than the minimum number of DOF required to access an arbitrary target configuration. This redundancy increases the size of the workspace and enables avoidance of singularities.

In both these cases, however, it is often not possible to reach an arbitrary target configuration because it lies outside of the workspace. Also, the analysis takes no account of obstacles — in the context of this thesis the obstacles are the object, the ground plane and the hand itself. Generally, therefore, the inverse kinematics problem is one of minimising a function depending on some or all of the following:

- Distance from desired distal link positions and orientations.
- Distance of joints from joint limits or from desired joint angles.
- Distance of digit links from one other.
- Distance of digit links from the object and ground plane.

The function combining these factors depends on the task.

[Koehler & Donath 88] studied the inverse kinematics of the Minnesota hand, the kinematics of which were essentially those of a Salibury hand. It was shown that if the position (3 DOF) and partial orientation (2 DOF) of the distal link of each digit is fixed, the hand configuration is either uniquely determined, kinematically inaccessible or - in a small number of pathological cases - underdetermined due to hand symmetries. A method to determine the inverse kinematics is given, in which the hand is modelled as two connected chains of links, one for each finger, both with the thumb in common. This approach is not always practical because the hand kinematics are completely con-

strained by the target configuration. If the hand cannot reach the target configuration, as will be the case for many contact configurations, the system fails.

Roberts [Roberts 90] presents a more workable solution. He solves the inverse kinematics of the Utah/MIT hand, by minimisation of a cost function. The cost function is based on distance from desired position (3 DOF) and partial orientation (2 DOF) of the distal link, and on the distance of the joint angles from their limits. The cost function is minimised using Powell's method [Press *et al.* 89] in the space of wrist position/orientation (a six parameter space) : given a particular wrist position/orientation, the digit kinematics are calculated such that the distal links are in the desired position/orientation; if the desired digit tip location is inaccessible, a nearby location is substituted. Quaternions, with some constraints on their signs, are used to describe wrist rotation, because there is a homeomorphism between the Euclidian space and the surface of the quaternion sphere. This means that two points close to one another on the surface of the quaternion sphere are close to one another in real space.

See Section 2.4.2 for a review of obstacle avoidance in dextrous grasping.

Local inverse kinematics

The local inverse kinematics problem can be posed thus:

Given a perturbation of the fingertips, calculate the joint motion that produces that perturbation, optimised according to some metric.

The Jacobian of a manipulator is defined by the following equation:

$$\dot{\boldsymbol{x}} = \boldsymbol{J}\dot{\boldsymbol{\theta}} \tag{2.13}$$

where $\boldsymbol{x}$ is the vector of fingertip positions and $\boldsymbol{\theta}$ is the vector of joint positions. For small motions of the joints and fingertips:

$$\delta\boldsymbol{x} = \boldsymbol{J}\delta\boldsymbol{\theta} \tag{2.14}$$

Assuming that the Jacobian, $\boldsymbol{J}$, is square, the perturbation in joint positions to give a

desired perturbation of the fingertips is therefore given by:

$$\delta\theta = J^{-1}\delta x \tag{2.15}$$

When J is rank-deficient, its determinant is zero and the inverse, J^{-1}, does not exist. This occurs when the manipulator is at a singularity. In the region of a singularity, very large values of $\delta\theta$ are required to produce small perturbations in joint position. This is undesirable from a control point of view, and also violates the assumption that perturbations in fingertip *and* joint positions are small, making the above equality invalid. The pseudo-inverse can be used to overcome these problems, as well as providing a solution for redundant manipulators with non-square Jacobians. Using the pseudo-inverse, the required joint perturbation can be written as:

$$\delta\theta = J^{\#}\delta x + (I - JJ^{\#})\phi \tag{2.16}$$

where $J^{\#}$ is the pseudo-inverse of the Jacobian and ϕ is an arbitrary vector. The first term corresponds to the solution of $\delta\theta$ which, out of the set of $\delta\theta$ that minimise $\mid \delta x - J\delta\theta \mid$, minimises $\mid \delta\theta \mid$. The second term corresponds to the null space of the Jacobian, $\mathcal{N}(J)$, which is the space of all the joint perturbations that give $\delta x = 0$. The position within the null space is controlled by the vector ϕ. For a square, full rank Jacobian, $J^{\#} = J^{-1}$ and $(I - JJ^{\#}) = 0$. For a square, but rank deficient Jacobian or a Jacobian with more columns than rows, as for a redundant manipulator, $(I - JJ^{\#}) \neq 0$. Note that unless the Jacobian has rank 0, the number of elements in ϕ does not correspond to the dimension of the null space, *i.e.* $(I - JJ^{\#})$ is rank deficient.

The pseudo-inverse is sometimes given by:

$$J^{\#} = (J^T J)^{-1} J^T \tag{2.17}$$

but sometimes $J^T J$ is singular when J is singular. It is always better to calculate the pseudo-inverse from the singular value decomposition (SVD) of the matrix. The singular value decomposition of the Jacobian is given by $J = USV^T$, where U and V

are rotation matrices in the fingertip and joint spaces respectively, and $\boldsymbol{S}$ is a diagonal matrix of singular values ordered from largest to smallest. The pseudo-inverse is then given by:

$$\boldsymbol{J^{\#} = VTU^{T}} \tag{2.18}$$

where

$$\begin{aligned} T(i,j) &= 0 \quad \text{if} \quad i \neq j \\ T(i,j) &= 1/S(i,j) \quad \text{if} \quad i = j \quad \text{and} \quad S(i,j) > \tau_{svd} \\ T(i,j) &= 0 \quad \text{if} \quad i = j \quad \text{and} \quad S(i,j) \leq \tau_{svd} \end{aligned} \tag{2.19}$$

and $\boldsymbol{\tau_{svd}}$ is the smallest singular value allowed. It is this threshold that prevents the joint values from becoming large near singularities. The null space may still be defined by $(\boldsymbol{I - JJ^{\#}})$ as in Equation (2.16), but is better extracted from the SVD: the null space is described by the set of basis vectors that are the columns of $\boldsymbol{V}$ corresponding to the diagonal elements of $\boldsymbol{S}$ lying below $\boldsymbol{\tau_{svd}}$

[Nakamura 91] states that while the pseudo-inverse is useful at singularities, it takes the same solution as the inverse close to singularities, so suffers from some of the same problems as the inverse. This is not true if the SVD formulation of the pseudo-inverse is used, because it enables small singular values to be removed from consideration, and it is these small singular values that cause the large joint perturbations for small fingertip perturbations. [Nakamura 91] goes on to state that the problems with the pseudo-inverse arise because it always enforces the condition that the error $\mid \boldsymbol{\delta x - J\delta\theta} \mid$ is minimised. He therefore proposes the **singularity-robust inverse**. This gives a weighted least squares solution, which allows a weighting to be assigned to the relative weighting of $\mid \boldsymbol{\delta\theta} \mid$ and $\mid \boldsymbol{\delta x - J\delta\theta} \mid$. The singularity-robust inverse is given by:

$$\mathbf{J^{*} = J^{T}(JJ^{T} + kI)^{-1}} \tag{2.20}$$

where k is a scalar weighting. It aims to do much the same as the SVD computation

of the pseudo-inverse, which prevents large joint torques by the removal of components with small singular values. $k = 0$ corresponds to the pseudo-inverse. $k = \infty$ corresponds to no motion of the manipulator.

Local inverse kinematics is important because it provides a scheme to perturb the hand to give desired fingertips motions. This can be used to follow prescribed digit trajectories, or to perturb a sub-optimal grasp to give a better solution, either by movement of the fingertip positions or by keeping the fingertip positions fixed and moving in the null space of the Jacobian (to improve the kinematics of the grasp). Local inverse kinematics can also be used to analyse hand kinematics for singularities, as is done for grasp families in Chapter 3.

2.4 Grasp Planning

Parallel jaw grippers are widely used in both industry and robotics research. The gripper itself has only one degree of freedom, so is inherently much simpler to analyse than a multifingered hand. [Roth *et al.* 89] use a system with a laser rangefinder mounted on the robot wrist and directly model placement of a parallel jaw gripper onto the range data in order to determine the best grasp position. HANDEY [Lozano-Perez *et al.* 87] uses simple geometrical constructions to plan a grasping position for an object. [Lozano-Perez 87] present a path planning method in configuration space (C-space) for a 6 DOF robot arm, which is used by [Pertin-Troccaz 87].

[Rutishauer & Ade 95] plan grasps for a parallel jaw gripper from dense range data. The range data is triangulated, opposing patches are then identified and grasps are determined by searching for a contact position on each patch, by movement over the triangular meshes of each surface. The search metric does not explicitly include the inverse kinematics of the hand, which are very simple (as it is a parallel jaw gripper). It is constructed, however, in such a way that it is likely that the two contact points will be accessible to a parallel jaw gripper. This method cannot be readily scaled up to work with a dextrous hand, due to the increased number of contacts, and because the inverse kinematics must be included explicitly to avoid planning an inaccessible grasp.

The most sophisticated parallel jaw gripper grasp planner (also capable of planning

two-fingered grasps) was developed by [Blake 92, Blake *et al.* 93], who gives a method for generating all force closure grasps on a smooth, closed 2D contour for two fingers. It finds "minimal" grasps, which are those where the coefficient of friction required for force closure is at a minimum. It therefore makes no assumption about the coefficient of friction. The grasps are found by locating the intersections of symmetry, anti-symmetry and "critical" sets, which are functions of the contour geometry. Grasps are classified as "type I" (stable for $\mu > 0$) and "type II" (stable for $\mu > \alpha$, where $\alpha > 0$). He suggests that the human aptitude at grasping might be linked to the human skill at detecting symmetries. The system is implemented using a contour tracker to provide the object data. [Taylor & Blake 94] extends the work to planning grasps on 3D objects by tracking the contour around the object. A full extension to 3D has not been developed.

Unfortunately, none of the parallel jaw gripper methods scale up to planning grasps for dextrous hands, due to the increased complexity of the configuration space. The principal problem faced when planning grasps for a dextrous hand is the conflicting interests of two requirements when choosing contact positions: they must be kinematically feasible (henceforth referred to as **accessible**) and they must be **stable**.

2.4.1 Stable grasps

There is some confusion in the grasp literature with respect to the terminology used to describe the restraint of an object in a grasp. The definitions used in this thesis are now described.

Definition 2 *A grasp is* **force closure** *if it can exert forces of arbitrary direction on the object.*

Definition 3 *A grasp is* **torque closure** *if it can exert torques of arbitrary direction on the object.*

Definition 4 *A grasp is* **force/torque closure** *if it can exert wrenches of arbitrary direction on the object.*

That a grasp is force/torque closure means that it is capable of resisting an arbitrary force and torque on the object, to within the bounds allowed by the fingertip forces. This means that the set of possible contact wrenches spans the entire wrench space and encloses the origin. Geometrically, this means that the convex hull of the contact wrenches encloses the origin. Mathematically, in an n-dimensional wrench space, $n+1$ wrenches, $\mathbf{w_i}$, can give a force/torque closed grasp if [Xiong *et al.* 93b]:

$$\begin{vmatrix} \mathbf{w_1} & \cdots & \mathbf{w_{n+1}} \\ 1 & \cdots & 1 \end{vmatrix} \neq 0 \tag{2.21}$$

and

$$\sum_{i=1}^{n+1} \lambda_i \mathbf{w_i} = 0 \qquad \lambda_i \geq 0, \quad \exists \lambda_i > 0 \tag{2.22}$$

[Nguyen 88] synthesised force/torque closure grasps for polyhedra, choosing the force/torque closure grasp that gave the maximum leeway in contact placement. [Xiong *et al.* 93b] synthesised force/torque closure grasps for frictionless contacts and [Xiong *et al.* 93a] synthesises force/torque closure grasps for frictional contacts. Often force/torque closure is referred to as simply force closure (as in [Nguyen 88]). Force/torque closure with frictionless contacts is sometimes referred to as **form closure** (as in [Nguyen 88] and [Xiong *et al.* 93b]).

Force/torque closure is essentially a geometric property of a grasp. It depends solely on the wrenches that can be exerted on the object, assuming rigid digits. It is therefore determined solely by the static frictional properties of the object and the bounds on the wrenches that the digits can exert. An assessment of grasp stability is rather more complex, and just how complex depends on the model of the contacts and the exact definition of stability.

The grasp can be defined to be stable if, when the object is perturbed by a small arbitrary wrench, this can be opposed by the grasp. This leads to the simplest way of addressing stability, which is to assume that there is some control scheme for the robot hand that can sense and oppose arbitrary small wrenches and is therefore, like force/torque closure, a purely geometric notion. The condition for stability then becomes simply that the convex hull of the contact wrenches must strictly contain the complement of the external wrench on the object (*i.e.* the complement of the external

wrench cannot lie on the edge of the convex hull of the contact wrenches). Sometimes, *e.g.* [Park & Starr 92], equilibrium grasps are synthesised for frictionless contacts, and it is then assumed that the effect of friction will be to make the grasp stable.

Definition 5 *A grasp possesses* **static stability** *if it can exert arbitrary small wrenches about the equilibrium grasp.*

If there is no external wrench on the object, then a static stable grasp is also a force/torque closure grasp. Conversely, a force/torque closure grasp is always static stable.

The grasp can also be said to be stable if, when the object is given arbitrary small perturbations of position and orientation, it returns to its original position. The most common way of addressing this sort of stability is to model each contact as one or more springs. A grasp that is stable corresponds to a minimum of the total spring energy. [Baker *et al.* 85] consider 3-digit grasps, with each contact modelled as a single linear spring. [Nguyen 87b] models each contact as three linear springs, and therefore having a stiffness $\mathbf{K_i}$, where $\mathbf{K_i}$ is a 3×3 stiffness matrix. He shows that any **non-marginal** force/torque closure grasp may be made stable, assuming that the stiffness of each contact can be selected. A force/torque closure grasp is **non-marginal** if none of the contact forces are at the edge of their friction cones. He presents a method for selection of the centre of compliance (the point about which a rotation of the object can be opposed by pure torque) by variation of the contact stiffnesses. The centre of compliance of a grasp is important when doing tasks such as the peg-in-hole task.

Definition 6 *A grasp possesses* **dynamic stability** *if, when the object is given an arbitrary small perturbation of position and orientation, it returns to its original position.*

Generally force/torque closure is too strong a constraint to expect of most grasps because it requires too much friction and/or too many contacts. Furthermore, if a force/torque closure grasp can be synthesised, the contact forces required to give a force/torque closure grasp may be very unbalanced, *i.e.* there may be large differences

in magnitude between contact forces, which is undesirable in terms of control and because fingertip force limits may be exceeded. Furthermore, force/torque closure ignores the help that gravity can be in establishing a good grasp. Static or dynamic stability will be sufficient to pick up objects and are more reasonable constraints to impose. Static stability is much easier to establish, since it is a purely geometric property; dynamic stability can only be established with knowledge of the hand control scheme.

Stability metrics

Having outlined how to assess whether a grasp is stable or force/torque closed, the next question to face is the assessment of the grasp quality. [Blake 92] identifies grasps that require the minimum coefficient of friction. If the coefficient of friction is unknown, this is a very sensible approach. [Ji & Roth 88], given three contact positions, find the force directions that give an equilibrium grasp with minimum friction.

[Nguyen 88] synthesises force/torque closed grasps such that the range of possible contact positions is maximised. This allows for uncertainty in contact position and for integration with an obstacle avoidance module.

[Pollard 94] optimises grasps using a stability metric. The unit contact wrenches are plotted in wrench space, and the convex hull is found. The maximum radius of sphere, centred at the origin, that can fit within the convex hull is taken as a measure of stability. The bigger the sphere, the more evenly the contacts are placed around wrench space, and the better they will be able to oppose arbitrary wrenches without becoming unbalanced (*i.e.* without one contact force becoming much larger than another). If the grasp is not force closed, the sphere radius becomes negative. Such a stability metric is shown to be useful. However, it is not a true metric because the dimensions of wrench space are not uniform: three dimensions have the units of position and three have the units of moment. The moment axes can therefore be scaled arbitrarily with respect to the position axes and the metric can change. Furthermore, the grasp metric is not independent of the origin with respect to which the torques are measured (in [Pollard 94] the centre of mass of the object is chosen as the origin).

Ultimately, the stability metric should depend on the control scheme of the hand. See Chapter 5 for a more in-depth discussion of stability metrics, as applied to the system described herein.

2.4.2 Accessibility

The problem of accessibility is the problem of solving the inverse kinematics of the hand in the presence of obstacles. These obstacles may be parts of the hand itself, the object to be grasped, the ground plane or other objects in the environment and must be taken into account when grasp planning. As outlined in Section 2.3.3, the inverse kinematic solutions for dextrous hands are a function of various factors. The hand must access the target configuration of distal links whilst avoiding collisions with obstacles. Task requirements, such as preferences for certain joint values, may also influence the solution.

As described in Section 2.3.3, [Koehler & Donath 88] and [Roberts 90] present inverse kinematic solutions without regard to obstacles in the environment. Little work has been done on obstacle avoidance for dextrous hands. [Lozano-Perez 87] gives a method for motion planning of a robot arm in configuration space, which successively breaks down an n dimensional configuration space into an $n-1$ dimensional space, until the obstacles in a 1D space are found. The free space is then described by rectangular regions which are represented in a graph and used to plan the path. The method is shown to be reasonably fast for a 6 DOF arm if the search space is appropriately subdivided. The method is not, however, readily applicable to the dextrous hand problem because of the large increase in number of DOF.

[Khatib 85] introduced the potential field for real-time obstacle avoidance of a robot arm. The potential field is used to artificially generate torques on the robot joints which keep the robot arm away from obstacles, and to keep the joints away from singularities. [Pollard 93, Pollard 90] use a similar technique to plan grasps. A first grasp is planned using heuristics, without taking into account possible collisions with the object. If this initial configuration collides with the object, it is deformed away from the object. This is done by simulating motion of the hand. The collision sites generate forces on the links of the hand, and a fingertip force attempts to keep the fingertips at the target

configuration. These forces create joint torques which move the hand away from the collision sites, resulting in a collision-free grasp. Additional torques keep joints away from their limits. This is a useful method to be used in any grasp planner where there is a possibility that the planned grasp may collide with the object. A big improvement would be to only allow joint movements that map to the null space of the Jacobian, in which case the method would not require a force to keep the fingertips at the target configuration.

[Lyons 86a] uses potential fields to compactly specify hand configurations (though it can be used for obstacle avoidance). Instead of a dynamic model of the manipulator, as used by [Khatib 85] and [Pollard 93], multiple attractive and repulsive potential fields are used to define desired link positions, and the manipulator is then moved to those joints by calculating the inverse kinematics. No method is given for resolving manipulator kinematic redundancy, however, whereas [Khatib 85] and [Pollard 93] resolve this locally by use of the dynamic model. Furthermore, for the specification of hand configurations, the potential fields must be chosen such that the desired hand configuration corresponds to a global minimum, and no local minima exist. The principal motivation for the work is not, however, collision avoidance, but rather to *preshape* hands (see Section 2.5). Potential fields to preshape hands for *tip*, *lateral* and *encompass* grasps are given.

The most widely used method of collision avoidance in the research literature is to side-step the issue altogether, by using a model of human grasping — preshaping.

2.5 Hand Preshaping

The paradigm of hand preshaping in robotics research arises from studies in how humans grasp objects. Before reviewing preshaping work in robotics, therefore, it is advisable to review the research into human grasping.

2.5.1 Human grasping

A human grasping action can be decomposed into four components:

1. Transport: movement of wrist to some pose and position.
2. Preshape: digit posture adopted as the wrist is transported.
3. Closure: movement of digits from preshape to contact with object, with the wrist fixed.
4. Execution: contacts exert forces on the object.

Preshape and transport components occur simultaneously. Closure may overlap with the transport phase. Furthermore, closure and execution may overlap, if some digits exert contact forces while others have not yet made contact.

Grasp classification

The simplest grasp classification schemes are those proposed by [Napier 56] and [Lyons 85], which are described in Section 2.2. [Iberall 87] proposes describing grasps in terms of pad, palm and side opposition and introduces the concept of **virtual fingers** whereby fingers that share the same function are grouped together and treated as one for the purpose of grasp classification and planning. [Iberall *et al.* 88] describes work on using a knowledge base to control dextrous hands using these ideas.

[Kang & Ikeuchi 91] and [Kang & Ikeuchi 93] study human grasping specifically for the purpose of recognising grasps so that robot manipulators can learn by observation. [Kang & Ikeuchi 91] describes a comprehensive grasp taxonomy and a basis on which to *recognise* grasps. Grasps are classed as *volar* (*i.e.* involving a palm contact) or *non-volar*. Further classification of non-volar grasps is based on the *contact web*, which is the relative positions of the fingertips in the grasp. Non-volar grasps can be classed as *fingertip* grasps, in which the contact web is approximately planar, and *composite non-volar* grasps, in which the contact web is non-planar. The bottom level of the classification hierarchy classes grasps according to the spatial arrangement of the contact web. Volar grasps are classed directly according to the spatial arrangement of the contact web: planar (non-prehensile support), prismatic/cylindrical and spherical. All volar grasps, with the exception of the planar non-prehensile grasp, are *encompass* grasps. Kang's fingertip grasp maps directly to the *tip* grasp, and Kang's composite non-volar grasp maps directly to the *lateral* grasp.

Digits in a grasp are grouped into virtual fingers (as in [Iberall 87]), by consideration of the opposition of contact forces. The grasps are classed according to the number of virtual fingers, the geometry of contact points and the *cohesive index* of the virtual fingers, which measures how well the functions of the constituent fingers map onto the function of the virtual finger. Human grasps are extracted from range and intensity data. The grasp extraction process starts with a hand model initialisation: intensity and range images are taken of an outstretched hand. From the intensity images, digits are segmented into their constituent phalanges (note that black lines were drawn on the subject's interphalangeal joints). Cylinders are then fitted to each segment from the range image data. The digits are then tracked over three subsequent frames and the positions of the digits in the final grasp are obtained. The grasp is then classified.

Kang's grasp taxonomy has been constructed for the recognition of grasps. As will become apparent in Chapter 3, the lowest level of the taxonomy — where the classification depends on the spatial arrangement of the contact web (*i.e.* the relative positions of the contacts) — is, for purposes of grasp planning, unnecessary. This division is solely a property of object shape; for grasp planning only functional classification of grasps is required.

[Cutkosky & Wright 86] and [Cutkosky & Howe 90] give a large, though incomplete, taxonomy of manufacturing grasps. These are classed according to precision and power of the grasp at the highest level, and then according to the relative contact arrangements at lower levels. Again, for grasp planning, the lower level of classification, which is based on contact position, is unnecessary. The taxonomies of both [Kang & Ikeuchi 91] and [Cutkosky & Wright 86] are therefore over-complex. Most robot grasping systems use the simplest ideas of grasp classification, based on the classifications of [Napier 56] and [Lyons 85]. The main lesson of human grasp classification is that grasps should be classified according to function, not object shape. The grasp planner must incorporate a metric that assesses how well that function is implemented, and it is the optimisation of the function that produces the grasp for a given object shape. In this thesis, a grasp metric is developed which optimises for mechanical and kinematic function.

Human preshaping

There have been studies of human grasping in the field of psychological research. Much of the work is concerned with the temporal aspects of the grasping action, such as when the preshape is formed, when the digits begin to close, and the overlap of these actions with the transport phase. This is of limited use in planning robotic grasps, because generally it is desirable to completely separate the transport, preshape and closure components for ease of planning. The work that deals with the actual shape that the hand adopts during preshaping is unfortunately rather too qualitative to be of direct applicability to robot grasping. [Jeannerod 81] measures the size of the preshape with respect to the size of object to be grasped. The preshape **aperture** is measured. The aperture of the hand is the distance between first finger and thumb in the preshape. The preshape is assumed to be the shape of the hand when the aperture is largest. In this thesis, the concept of preshape aperture is expanded to include the relative distance between all digits of the hand. Similar to this concept of preshape aperture is the *contact web* of [Kang & Ikeuchi 91], which was defined by the relative positions of the fingertips in the final grasp, as opposed to preshape (and was used to classify grasps rather than plan them).

[Wing *et al.* 86] showed that if uncertainty exists in the position of the object to be grasped, the preshape aperture is widened. This was shown by getting subjects to grasp an object, measuring the preshape aperture, blindfolding them, getting them to grasp an object again and again measuring the preshape aperture. Blindfolded, the preshape aperture is bigger.

[Kang & Ikeuchi 93] presents a method for the temporal segmentation of human grasping actions. The work is intended as a precursor to further characterisation of the individual phases. The grasping action is segmented into the pregrasp, manipulation and depart phases. The fingertips form the *fingertip polygon* and the grasping action is segmented by examination of the profile of *volume sweep rate* which is the volume swept out by the fingertip polygon in a time-step. The minima in the volume sweep rate profile correspond to transitions between different phases.

Even though there has been a significant (though not very useful) amount of work on

grasp classification, there has been no comprehensive study of preshape classification. The main lessons from the studies are that the width of preshape aperture is proportional to object size and that the width of preshape aperture increases with uncertainty in object location.

2.5.2 Robotic grasping using preshapes

Preshapes have been used extensively in dextrous hand research. The main reason for this, as stated in Section 2.4.2 is the complexity of a global solution to grasp accessibility.

[Lyons 85] plans preshaping of a dextrous hand, taking into account the type of task (precision/power), the size of object to be grasped and the shape of the object (flat/rounded). Grasps are chosen to be either tip, lateral or encompass. A potential field is used in Lyons [Lyons 86b] to preshape a dextrous hand; however, the preshaping parameters are not derived from real data. [Stephanou & Erkmen] describe use of a knowledge base to preshape dextrous hands according to task and object shape. In [Nguyen & Stephanou 90], task-driven heuristics are used to partially or fully define the hand preshape, and then the hand configuration is determined using topological techniques. [Stansfield 91] preshaped a Salisbury hand using a knowledge-based system, which is used to grasp polyhedral and simple curved objects.

[Bard *et al.* 95] use preshaping to plan grasps on objects with a Salisbury hand. The objects are modelled using *elliptical cylinders*, which are especially suitable for planning power grasps. The preshapes are planned using a set of heuristics based on the properties of the elliptical cylinders. The digit closure is then planned using a dynamic model of the hand. This system is important for two reasons. Firstly, it is the only system to explicitly consider digit closure. Unfortunately, the dynamic model used is very complex and it is not clear that the model parameters can be set realistically. Secondly, it is one of the few systems which provides a vision system tailored to grasp planning. The elliptical cylinder representation is particularly suited to planning power grasps, though the digit closure planning means that precision grasps can also be planned.

[Tomovic *et al.* 87] sketches out a plan for a grasping system which uses a dextrous

hand with relatively few degrees of freedom (such as the Belgrade/USC hand). Objects are modelled by a small number of geometric primitives - cylinders, cones, cuboids. Preshape and target approach are planned using an expert system. Unfortunately the heuristics used to form a grasp are over-simplified and no complete working system (in simulation or reality) is shown.

None of the above work defines what a preshape is, or exactly how it helps solve the grasping problem. Instead, heuristics are relied upon to form and place the preshape. Furthermore, no consideration of digit closure is made when forming the preshape. This is worth investigating because it is the digit closure that determines the final grasping configuration, so the preshape cannot be truly assessed without consideration of the digit closure. Note that [Bard *et al.* 95] do consider digit closure when planning the transition from preshape to final grasp, though this is only after the preshape has been placed and formed, which restricts its usefulness for planning precision grasps. See Chapter 5 for a more in-depth discussion of the significance of preshaping and digit closure.

2.6 Task-oriented representations

As noted in Section 2.5.1, very little consideration has been made of the type of object model required for robot grasping.

[Blake 92] plans grasps for a parallel jaw gripper on smooth continuous contours, extracted from a contour tracking system. [Rutishauer & Ade 95] plan grasps for a parallel jaw gripper from dense range data, acquired using structured light. The range data is triangulated and different views are merged and retriangulated. It is desirable, though not necessary, to merge overlapping data from different views because the bulk of the algorithm is a search over opposing surfaces: overlapping surfaces should therefore simply slow the algorithm down.

[Gatrell 89] plans grasps on CAD models of polyhedra, using the Extended Gaussian Image to find opposing planes. [Rao *et al.* 88] describes an object modelling system for grasp planning (with the Belgrade/USC hand) which uses generalised cylinders, which are not suitable when planning precision grasps (though may be suitable when

planning power grasps).

[Stansfield 88] extracts object models from various views. The views are not fused; rather they are treated separately as corresponding to different approach directions of the robot hand. The object model (and subsequent grasp planner) uses contours, planes and curved surfaces. Data is derived from two CCD cameras which trigger haptic modules that further explore the object. Examples are shown for kitchen-ware. The modelling system is too simple to model irregular, arbitrary curved objects, but is interesting in that the task requirements mean that it is not necessary to fuse views because of the assumptions that that planner makes about contours (it assumes that the hand can grasp "behind" the contours).

[Bard *et al.* 95] offers a more complete modelling system. Two object models are constructed from stereo images, taken from several viewpoints. A voxel model is built. This model is quite coarse, and is constructed by the fusion of occupancy probabilities from different views. Objects with the voxel model are separated and approximated by convex hulls, which are then used for motion planning — this is a reasonable approximation to make, since it is generally undesirable for a manipulator to enter concavities until the digit closure phase of the grasp. The second model is based on a primitive called the *elliptical cylinder*, which is the convex hull of a set of similar ellipses. This is extracted from the voxel model. The objects are modelled as composites of elliptical cylinders, arranged hierarchically. The parameters of elliptical cylinders map directly to preshape parameters. Note the similarity between the elliptical cylinders and the generalised cylinders of [Rao *et al.* 88].

In [Wren 92] [Wren & Fisher 93] range images are segmented for robot grasp planning. The segmentation consists of planes and biquadratics. Erosion and dilation are used to discard surface patches that were too small to provide a surface contact and the patches were classified according to size and shape. The shape descriptions are intended as input to a grasp planner, and so concentrate on describing the direction and degree of curvature of the patches. The use of such a description is restricted largely to rule-based grasp planners which, as described in Section 2.7, are limited with respect to the range of object shapes that they can grasp.

It can be seen that little consideration has been made of the type of representations required for grasping. [Wren 92] and [Wren & Fisher 93] show that a combination of planes and biquadratic patches can give a good description of contact geometry, but a complete model of the object is not built. [Bard *et al.* 95] uses a shape descriptor that maps directly to a preshape. However, it does not provide detailed surface information, which may be necessary for planning precison grasps. [Stansfield 88] uses surfaces in the model, but the data is too sparse to provide quantitative curvature information. [Rutishauer & Ade 95] uses dense range data to plan parallel jaw gripper grasps. [Bard *et al.* 95] shows that more than one type of model may be necessary to plan grasps.

2.7 Complete systems

Very few complete dextrous grasping systems are in existence.

[Stansfield 91] uses a task-oriented representation of the object and then uses a knowledge base to plan tip, lateral or encompass preshapes with a Salisbury hand. Digits are then closed in an arbitrary manner to acquire the object. Unfortunately, the types of objects that can be modelled are limited, there is no explicit collision avoidance and the set of heuristics used will not work with arbitrary objects.

[Bard *et al.* 95] have developed the most comprehensive system for grasping. They build a task-oriented representation of the object and use rules to plan preshapes for the hand. The digit closure is then planned using a dynamic simulation, to give a stable grasp. This is the only system to consider digit closure, but it does so only after the preshape has been formed and placed. The nature of the object representation and the digit closure simulation mean that it is rather more suited to planning power grasps than precision grasps. In my thesis, digit closure plays a more crucial role and is considered at two stages of the grasp planning. In the formation of the preshape, the digit closure is approximated and used to estimate the quality of the final grasp given a preshape. Once the preshape is formed, the digit closure is chosen from a set of prescribed trajectories to give the best final grasp (see Chapter 5).

[Pollard 90] uses a simple rule-base to plan grasps on polyhedra with the Salisbury

hand. If the grasp is inaccessible due to collisions between the hand and the object, the hand is deformed using what amounts to a potential field method. However, only a small set of results are shown and no assessment is made of the limitations of the deformation approach. The object model is not derived from real data.

[Pollard 94] plans power grasps using *grasp prototypes* (see Chapter 5). A grasp prototype is an optimal grasp for some given object model. Given an arbitrary object, the grasp is planned starting from the "closest" grasp prototype. The system searches hand configurations around the grasp prototype and identifies the best grasp, according to a stability metric. This provides an excellent method of combining accessibility and stability analysis, but unfortunately the hardest part of the problem — identification of which grasp prototype to start with — is left unsolved. Again, the object model is not derived from real data.

[Tomovic *et al.* 87] describes a human-inspired grasping system; though this is not implemented as a fully working system, it recognises the importance of constraining the degrees of freedom of dextrous hands when simply grasping objects.

There remains much to be done in building a complete grasping system. No system has used a detailed surface model of the object, and no system has demonstrated an ability to plan tip, lateral and encompass grasps on a wide variety of objects.

2.8 Conclusions

The major problems in previous research are as follows:

- There are very few complete dextrous hand grasping systems, perhaps because the question of how to integrate stability and kinematic constraints has not been resolved.
- Few task oriented object representations have been used, with much grasping planning work assuming perfect geometric descriptions of the object to be grasp.
- No grasp metric integrates mechanical and kinematic requirements.
- Formation and placement of preshapes is based on heuristics, which limits the

usefulness of the algorithms to simple object shapes.

- There has been little consideration of the digit closure, and how this affects the placement of the preshape: only [Bard *et al.* 95] consider digit closure and this is for a system more suited to planning power rather than precision grasps.
- Many planners cannot cope with arbitrary curved objects.

In this thesis, precision grasps are planned (in simulation) on unknown objects of arbitrary shape (curved or polyhedral). A surface model of the object is used to provide information about contact geometry and a voxel model is used for collision checking. A grasp metric that integrates mechanical and kinematic requirements is introduced. The function of preshapes is examined and a new method of preshape specification, based on task-specific ideal grasps, is introduced. Preshapes are associated with digit trajectories to form **grasp families**. This enables the formation and placement of preshapes according to the grasp metric rather than heuristics. Example grasps are shown for several different objects, both curved and polyhedral.

Chapter 3

Grasp Families

This chapter describes the grasp families used to plan precision grasps. First of all, the hand modelling system and the contact model are described. Four examples of hand models are given: a parallel jaw gripper, the Salisbury hand, a 3-digit anthropomorphic hand model (referred to herein as *SAM*, which stands for *simplified anthropomorphic model*) and a human hand. The use of preshapes to constrain the search for a grasp is discussed and the idea of a grasp family is introduced: a grasp family consists of a preshape and a prescribed digit closure trajectory. Related to each grasp family is a set of task requirements and a particular object shape: essentially it is a mode of joint actuation which enables high quality grasps to be planned whilst coupling certain degrees of freedom in the hand. Grasp families are then given for the SAM and extensions to the other 3 hand models are outlined. Finally, an alternative approach to the derivation of grasp families is described, illustrated by a simple 2D example.

The main contributions of the work covered in this chapter are:

- A formal definition of a preshape.
- The grasp family, which associates digit trajectories with preshapes.
- Formation of three different task-specific grasp families for the SAM.

3.1 SAM: Simplified Anthropomorphic Model

In this section, the Simplified Anthropomorphic Model of the hand is introduced. It is designed to be used throughout the rest of the thesis to illustrate the proposed method of grasp planning. It is designed to have the same basic characteristics as a human hand, but be somewhat simpler. It also bears similarities to existing robotic dextrous hands, such as the Salisbury hand and Utah/MIT hand. Figure 3.1 shows a picture of the SAM. The palm of the SAM is an isosceles triangle formed by placing a finger at each endpoint of the base and the thumb at the apex. The height of the triangle is 60mm and the length of the base is 45mm. Each side of the triangle is a cylinder of radius 5mm. Each digit has the same dimensions: the proximal, middle and distal links have lengths 45mm, 35mm and 25mm respectively, and each link is a cylinder of radius 5mm. Each fingertip is a hemisphere of radius 5mm. The unit on each axis is the millimeter.

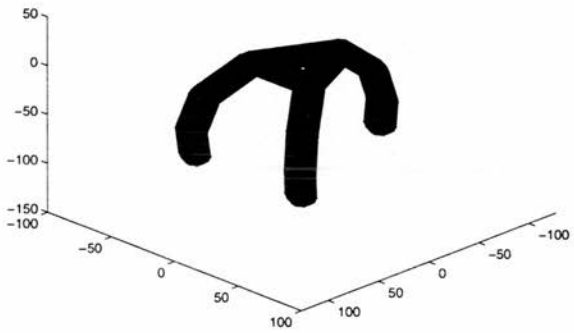


Figure 3.1: The SAM. The palm of the SAM is an isosceles triangle formed by placing a finger at each endpoint of the base and the thumb at the apex. The height of the triangle is 60mm and the length of the base is 45mm. Each side of the triangle is a cylinder of radius 5mm. Each digit has the same dimensions: the proximal, middle and distal links have lengths 45mm, 35mm and 25mm respectively, and each link is a cylinder of radius 5mm. Each fingertip is a hemisphere of radius 5mm. The unit on each axis is the millimeter.

The SAM is designed to retain the basic properties and degrees of freedom of individual digits, while reducing the total number of digits for ease of analysis. The thumb has three revolute joints which enable it to *flex* and *extend* (*i.e.* open and close in a plane perpendicular to the palm). The two fingers add an extra revolute joint at the knuckle, to form a ball joint (two revolute joints with orthogonal axes of rotation and coincident origins). This gives them the ability to *abduct* and *adduct* (*i.e.* move from side to side), as well as to flex and extend.

There is some coupling between joints in the SAM. The middle and distal links of each digit are coupled, such that the distal joint angle is always $\frac{2}{3}$ of the middle joint angle. This approximates the behaviour of the human finger (when moving freely). The angle of abduction for each finger is identical (though oriented in an opposite sense due to the difference in orientations of the base frame of each finger). This is based on the observation that when forming grasps, humans tend to abduct all their fingers as one.

The SAM has some similarities to the Anthrobot hand, in that the middle and distal joints of each digit are coupled. The choice of which digits can be abducted should be contrasted with the Belgrade/USC hand: in the SAM the fingers abduct and the thumb does not, but in the Belgrade/USC hand the thumb abducts and the fingers do not. The Belgrade/USC configuration is good in that it enables the thumb to oppose any finger, which is necessary for two-contact grasps. The SAM was designed primarily for three-contact grasps and so the thumb remains fixed and the abduction of the fingers can be chosen to produce an equilibrium grasp.

The effect of this coupling is to prevent the hand from performing dextrous manipulation, because it cannot exert arbitrary torque and force on an object due to the coupling of finger abductions and the lack of thumb abduction. However, equilibrium grasps can be constructed with this hand (the hand is designed with this in mind). Soft fingers, achieved through rubber-coated fingertips for example, mean that these equilibrium grasps will be stable and the natural compliance of the soft fingers will expand the range of manipulations that the 9 DOF hand is capable of.

3.2 Hand Model

In this section the requirements placed on the hand model are given, the model itself is described and example hand models are given for a parallel jaw gripper, the Salisbury hand [Salisbury & Craig 82], SAM the 3-fingered anthropomorphic hand and a human hand.

3.2.1 Function

The hand model is to be used for the planning of grasps. It will therefore be used for the following functions:

- Collision checking.
- Calculation of forward and inverse kinematics.
- Modelling of hand-object contacts.
- Display of results.

3.2.2 Requirements

The hand model must satisfy the following requirements:

- It must be able to model hands with the following type of joints:
  - Rotational: the type of joint used in almost every dextrous hand.
  - Prismatic: used in robotic arms rather than dextrous hands, but it is included to enable us to model parallel jaw grippers.
  - Ball: not commonly used in robotics, due to tendon-routing complications, but necessary in order to model human hands (the knuckle is essentially a ball joint).
- It must be capable of modelling arbitrary arrangements of multiple digits, with arbitrary kinematics within each digit.
- The palm of the hand must be represented, in order to:

- Enable collision checking between palm and object.
- Model grasps made with palm contacts if power grasps are to be planned (in this thesis, however, power grasps are not planned).

A secondary requirement might be that the representation of the hand should be unique, with no degeneracy in the orientation of joint axes, so that researchers can readily compare hand models. However, doing this complicates the production of hand models so much that it is better to allow the hand modeller to resolve any degeneracy in the most intuitive way.

3.2.3 Model

The hand model consists of a kinematic model — essentially a "stick model" of the hand — and a volumetric description which is overlaid on the kinematic model.

Kinematic model

The homogeneous transformation from the global reference frame to the hand reference frame is given by the matrix H. For a hand with N digits, a digit model, f $(0 \leq f < N)$, has an associated 4×4 matrix A_0^f which describes the homogeneous transformation from the hand reference frame to the digit base reference frame.

A digit is modelled as a serial link manipulator. A digit f consists of M_f joints and M_f links. Joint i $(1 \leq i \leq M_f)$ lies between links i and $i-1$, where link 0 is the palm of the hand. The reference frame for each link is placed at the end of the link that is closest to the digit tip. The reference frame for link M_f is therefore placed at the digit tip; the reference frame for link 0 is the digit base reference frame. See Figure 3.2.

There are three types of links, depending on the type of joint that they move around (or not, as may be the case): revolute, prismatic or fixed. The base link of a digit is always a fixed link. A ball joint is modelled as two revolute joints with coincident origin (*i.e.* joined by a link of zero length). For a digit f $(0 \leq f < N)$, the relative positions of the links are described as follows:

- A **revolute link** i $(1 \leq i \leq M_f)$ contains the following basic information:

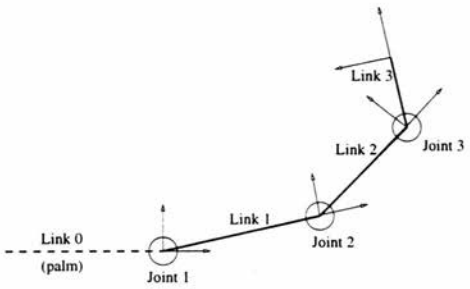


Figure 3.2: Finger model, with joints and links labelled. The reference frame for each link is placed at the end of the link.

- Joint variable θ_i^f, the angle of rotation of link i about joint i.
- Joint limits, θ_{i0}^f and θ_{i1}^f $(\theta_{i0}^f \leq \theta_i^f \leq \theta_{i1}^f)$.
- L_i^f, the 4×4 transformation matrix from link $i-1$ to link i when $\theta_i^f = 0$.

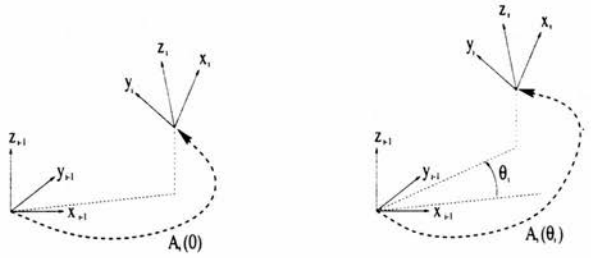
Rotations of joint i are anticlockwise about the positive z axis of link $i-1$, so the transformation from link $i-1$ to link i is given by $A_i^f(\theta_i^f) = R_i^f(\theta_i^f)L_i^f$ where

$$R_i^f(\theta_i^f) = \begin{bmatrix} \cos\theta_i^f & -\sin\theta_i^f & 0 & 0 \\ \sin\theta_i^f & \cos\theta_i^f & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.1}$$

Figure 3.3 shows the relationship between the reference frames of links i and $i-1$ for a revolute joint: (a) shows the reference frames in their base position, (b) shows the reference frames with joint angle θ_i^f.

- A **prismatic link** i $(1 \leq i \leq M_f)$ contains the following basic information:
  - Joint variable d_i^f, the distance of translation of link i along joint i.
  - Joint limits, d_{i0}^f and d_{i1}^f $(d_{i0}^f \leq d_i^f \leq d_{i1}^f)$.
  - L_i^f, the 4×4 transformation matrix from link $i-1$ to link i when $d_i^f = 0$.

Translations of joint i are along the positive z of link $i-1$, so the transformation from link $i-1$ to link i is therefore given by $A_i^f(d_i^f) = T_i^f(d_i^f)L_i^f$ where

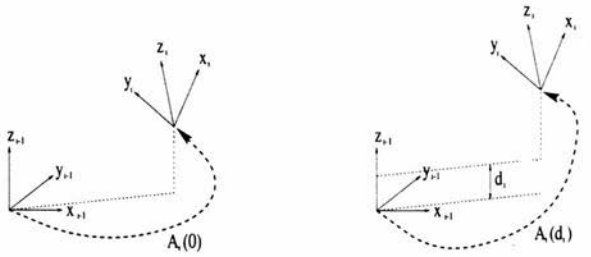


(a) (b)

Figure 3.3: Relationship between reference frames surrounding revolute joint i. The dotted lines show the transformation from the $(i-1)^{th}$ frame to the i^{th} frame.

$$T_i^f(\theta_i^f) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i^f \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.2}$$

Figure 3.4 shows the relationship between the reference frames of links i and $i-1$ for a prismatic joint i: (a) shows the reference frames in their base position, (b) shows the reference frames with joint distance d_i^f.



(a) (b)

Figure 3.4: Relationship between reference frames surrounding prismatic joint i. The dotted lines show the transformation from the $(i-1)^{th}$ frame to the i^{th} frame.

- A **fixed link** i $(1 \leq i \leq M_f)$ contains the following basic information:

– L_i^f, the 4×4 transformation matrix from link $i-1$ to link i.

The transformation from link $i-1$ to link i is given by $A_i^f = L_i^f$.

Given a set of A matrices, the position of the origin of the reference frame of the digit tip of digit f, with respect to the global frame, is given by

$$F^f = HA_0^f A_1^f \ldots A_{M_f}^f \tag{3.3}$$

This method does not define a unique set of transformation matrices L_i^f. The translational components of the transformation are uniquely determined, but there is degeneracy in the rotational component. At a joint, the z axis is fixed to lie along the joint axes, but its sense is not fixed; the x and y axes are under-constrained, in that they may swivel around the z axis. At the digit tip, the rotational component of the local reference frame is completely unconstrained. In order to remove this degeneracy, a convention such as the following could be adopted:

- Fixing the joint limits, such that $\theta_{i0}^f = 0$ and $\theta_{i1}^f > 0$ for a revolute joint and $d_{i0}^f = 0$ and $d_{i1}^f > 0$ for a prismatic joint. However, this would mean that models of the same hand with different joint limits would be different, and prevent individual algorithms from altering the joint limits easily.

- The Denavit-Hartenberg representation [Paul 81], which gives a method of calculating L_i^f from link length and link twist, which uniquely determines the local reference frame for each joint. However, this method does not fix the link reference frames to be at the end of each link — the geometry of each link with respect to its local reference frame has to be given.

It was decided therefore that no convention should be forced on the hand modeller, leaving it up to them to define the hand in the most pleasingly intuitive way. (In many circumstances it transpires that this coincides with the first convention, of making all joint angles positive).

Volumetric model

The i^{th} link of digit f is modelled as a cylinder of radius r_i^f. The axis of the cylinder lies along the line joining the the reference frames of link $i-1$ and link i. A sphere of radius r_i^f is placed at the end of the link (*i.e.* at the origin of the reference frame of link i).

If $N > 2$ (where N is the number of digits), the palm of the hand is modelled by a set of N cylinders joining the digit bases. This describes the edges of the palm and should be sufficient for collision checking against most objects. If palm contacts were to be used, then the model would have to include a palm surface, such as a plane. However, for collision checking it is more straightforward to use cylinders as the primitive for hand modelling (see Section 4.4.5). The axis of the palm cylinder f $(1 \leq f \leq (N-1))$ lies along the line joining the bases of digits f and $f+1$; the axis of the palm cylinder N lies along the line joining the bases of digit N and digit 1. The radius of each palm cylinder is the same: r_{palm}. A sphere of radius r_{palm} is placed at each of the digit bases.

If $N = 2$, the palm of the hand is a single cylinder of radius r_{palm} joining the two digit base reference frames, with a sphere of radius r_{palm} at each digit base.

3.2.4 Examples

In this section four hand models are presented. In each model, the digit with index 0 is the thumb [1] (this makes no difference to the hand model, but there is a difference in how thumb and fingers are used in the grasp planning). Remember that superscripts refer to a digit index, and subscripts refer to a link index. Link i rotates about joint i.

[1] The parallel jaw gripper, which has no "thumb", has no digit with index 0.

Parallel Jaw Gripper

hand, H

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

base, L_0^1

$$\begin{bmatrix} 0 & 0 & 1 & -50 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

base, L_0^2

$$\begin{bmatrix} 0 & 0 & -1 & 50 \\ 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

prismatic, L_1^1

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

prismatic, L_1^2

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

fixed, L_2^1

$$\begin{bmatrix} 1 & 0 & 0 & 50 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

fixed, L_2^2

$$\begin{bmatrix} 1 & 0 & 0 & 50 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$0 \leq d_1^i \leq (50 - r)$$

$$r_{palm} = r_2^1 = r_2^2 = r$$
$$r_1^1 = r_1^2 = 0$$

See Figure 3.5(a) for a picture of the parallel jaw gripper model. The end of each link is labelled for one finger of the hand: the other finger is identical. Note that even though it appears that each digit consists of only one link, it must be modelled as two: a prismatic link and a fixed link. The prismatic link is a link of variable length which acts to position the fixed link which constitutes the digit proper. The reason that it is not modelled as one single link is that it would lead to a more complicated link representation.

The prismatic link lies along the palm of the hand, and is therefore modelled with zero radius. The link axes are oriented such that both joint distances are positive.

Salisbury Hand

This is a model of the three-fingered Salisbury hand [Mason & Salisbury 85].

$$\text{hand, } H \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{base, } L_0^0 \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.7071 & 0.7071 & 0 \\ 0 & -0.7071 & 0.7071 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{base, } L_0^1 \begin{bmatrix} 1 & 0 & 0 & -23.5 \\ 0 & 1 & 0 & 31.0 \\ 0 & 0 & 1 & 75.0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{base, } L_0^2 \begin{bmatrix} 1 & 0 & 0 & 23.5 \\ 0 & 1 & 0 & 31.0 \\ 0 & 0 & 1 & 75.0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{revolute,} L_0^1 \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 35.0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{revolute,} L_1^1 \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 35.0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{revolute,} L_2^1 \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 35.0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{revolute,} L_0^2 \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 50.7 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{revolute,} L_1^2 \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 50.7 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{revolute,} L_2^2 \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 50.7 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{revolute,} L_0^3 \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 40.7 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{revolute,} L_1^3 \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 40.7 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{revolute,} L_2^3 \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 40.7 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$r_{palm} = r_i^f = r$$

Note that the 3 digits all have an identical set of transformation matrices A_i^f $(i > 0)$, making it explicit that all 3 digits are identical. See Figure 3.5(b) for a picture of the Salisbury hand model. The end of each link is labelled for the thumb and one finger: both fingers have identical configurations.

SAM

hand, H

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

base, L_0^0

$$\begin{bmatrix} -1 & 0 & 0 & -40 \\ 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

base, L_0^1

$$\begin{bmatrix} 1 & 0 & 0 & 20 \\ 0 & -1 & 0 & -22.5 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

base, L_0^2

$$\begin{bmatrix} 1 & 0 & 0 & 20 \\ 0 & 1 & 0 & 22.5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

revolute, L_1^1

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

revolute, L_1^2

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

revolute, L_1^0

$$\begin{bmatrix} 1 & 0 & 0 & 45 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

revolute, L_2^1

$$\begin{bmatrix} 1 & 0 & 0 & 45 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

revolute, L_2^2

$$\begin{bmatrix} 1 & 0 & 0 & 45 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

revolute, L_2^0

$$\begin{bmatrix} 1 & 0 & 0 & 35 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

revolute, L_3^1

$$\begin{bmatrix} 1 & 0 & 0 & 35 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

revolute, L_3^2

$$\begin{bmatrix} 1 & 0 & 0 & 35 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

revolute, L_3^0

$$\begin{bmatrix} 1 & 0 & 0 & 25 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

revolute, L_4^1

$$\begin{bmatrix} 1 & 0 & 0 & 25 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

revolute, L_4^2

$$\begin{bmatrix} 1 & 0 & 0 & 25 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\theta_i^f \geq 0$$

$$\theta_1^0 \leq \pi/2$$
$$\theta_2^0 \leq \pi/2$$
$$\theta_3^0 \leq \pi/3$$

$$\theta_3^0 = \tfrac{2}{3}\theta_2^0$$

$$\theta_1^1 \leq \pi/4$$
$$\theta_2^1 \leq \pi/2$$
$$\theta_3^1 \leq \pi/2$$
$$\theta_4^1 \leq \pi/3$$

$$\theta_4^1 = \tfrac{2}{3}\theta_3^1$$

$$\theta_1^2 \leq \pi/4$$
$$\theta_2^2 \leq \pi/2$$
$$\theta_3^2 \leq \pi/2$$
$$\theta_4^2 \leq \pi/3$$

$$\theta_4^2 = \tfrac{2}{3}\theta_3^2$$

$$r_{palm} = r_i^f = r$$

This model is of the SAM, described in Section 3.1. To recap, the thumb has three revolute joints which enable it to *flex* and *extend* (*i.e.* open and close in a plane perpendicular to the palm). The fingers add an extra revolute joint at the knuckle, to

form a ball joint (two revolute joints with orthogonal axes of rotation and coincident origins). The middle and distal links of each digit are coupled, such that the distal joint angle is always $\frac{2}{3}$ of the middle joint angle. The angle of abduction for each finger is identical (though oriented in an opposite sense due to the difference in orientations of the base frame of each finger).

The link axes are oriented such that all joint angles are positive. See Figure 3.5(c) for a picture of the SAM. The end of each link is labelled for the thumb and one finger: both fingers have identical configurations.

Human Hand

$$\text{hand, } H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

base, L_0^0	base, L_0^1	base, L_0^2	base, L_0^3	base, L_0^4
$\begin{bmatrix} .71 & 0 & -.71 & -80 \\ .71 & 0 & .71 & 10 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & -4 \\ 0 & 1 & 0 & 23 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & -2 \\ 0 & -1 & 0 & -25 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & -6 \\ 0 & -1 & 0 & -47 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
revolute, L_1^0	revolute, L_1^1	revolute, L_1^2	revolute, L_1^3	revolute, L_1^4
$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
revolute, L_2^0	revolute, L_2^1	revolute, L_2^2	revolute, L_2^3	revolute, L_2^4
$\begin{bmatrix} 1 & 0 & 0 & 60 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 55 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 58 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 55 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 43 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
revolute, L_3^0	revolute, L_3^1	revolute, L_3^2	revolute, L_3^3	revolute, L_3^4
$\begin{bmatrix} 1 & 0 & 0 & 35 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 30 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 35 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 35 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 25 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
revolute, L_4^0	revolute, L_4^1	revolute, L_4^2	revolute, L_4^3	revolute, L_4^4
$\begin{bmatrix} 1 & 0 & 0 & 28 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 25 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 28 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 28 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 24 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

$$\theta_i^f \geq 0$$

$\theta_1^0 \leq \pi/3$	$\theta_1^1 \leq \pi/5$	$\theta_1^2 \leq \pi/6$	$\theta_1^3 \leq \pi/6$	$\theta_1^4 \leq \pi/4$
$\theta_2^0 \leq \pi/4$	$\theta_2^1 \leq \pi/2$	$\theta_2^2 \leq \pi/2$	$\theta_2^2 \leq \pi/2$	$\theta_2^2 \leq \pi/2$
$\theta_3^0 \leq \pi/2$	$\theta_3^1 \leq 2\pi/3$	$\theta_3^2 \leq 2\pi/3$	$\theta_3^2 \leq 2\pi/3$	$\theta_3^2 \leq 2\pi/3$
$\theta_4^0 \leq \pi/2$	$\theta_4^1 \leq 4\pi/9$	$\theta_4^2 \leq 4\pi/9$	$\theta_4^2 \leq 4\pi/9$	$\theta_4^2 \leq 4\pi/9$

$$r_{palm} = r_i^f = r$$

The human hand model consists of 5 digits (digit 0 is the thumb). Each digit has three joints: proximal (at the base), middle and distal. The proximal joint is a ball joint,

and is therefore modelled as two coincident revolute joints with orthogonal z axes (L_1^f and L_2^f). The middle and distal joints are simple revolute joints.

The link axes are oriented such that all joint angles are positive.

See Figure 3.5(d) for a picture of the human hand model. The end of each link is labelled for the thumb and one finger: all fingers have similar configurations.

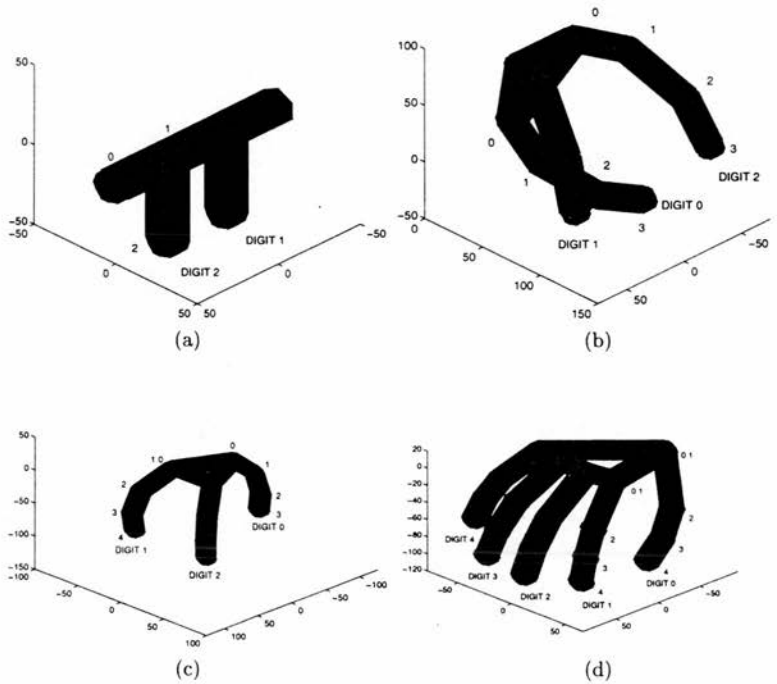


Figure 3.5: Hand models. All digits are labelled and the links of selected digits are labelled (each label is located at the far end of the corresponding link). (a) parallel jaw gripper, (b) Salisbury hand, (c) SAM (note that within each finger, link no. 1 is coincident with link no. 0), (d) human hand.

3.3 Contact Model

The type of contact model used should depend on the level of realism required and on the type of task that is being analysed. For instance, analysis of static grasp mechanics generally requires much simpler contact models than does the analysis of quasistatic manipulation.

There are various ways of modelling digit tip-object contacts. The three simplest models were introduced by Salisbury (see [Mason & Salisbury 85]) and used to synthesise force-closure grasps by [Nguyen 88]. In the **frictionless point contact** model, the contact can only exert forces perpendicular to the contact surface, as shown in Figure 3.6(a). In the **point contact with friction** model, the contact can exert forces which are directed within a friction cone with an axis coincident with the surface normal, of half-angle $\phi = tan^{-1}\mu$, where μ is the coefficient of friction: see Figure 3.6(b). The **soft finger** model adds the ability to exert a torque about the surface normal to the point contact with friction, as shown in Figure 3.6(c). This can be illustrated by placing a sheet of paper on a smooth surface. If the reader presses down on this paper with a fingertip, and then rotates the fingertip, the paper rotates. This is due to the torque about the contact surface normal.

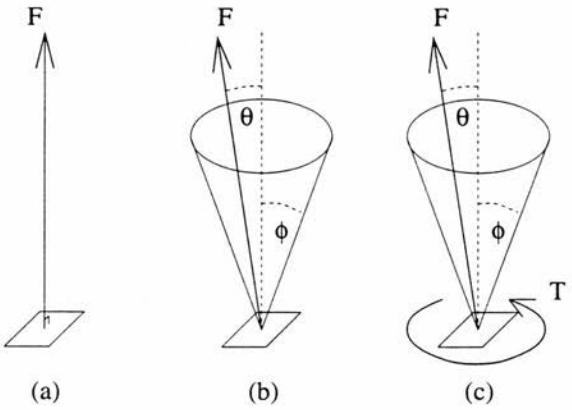


Figure 3.6: Contact types: (a) frictionless point contact, (b) point contact with friction, (c) soft finger contact.

Salisbury characterised these contact types by the constraints they place on the object motion (by consideration of the twist and wrench systems). [Nguyen 88] provides a convenient summary of the minimum number of contacts of each type that is required for force closure grasp in 3D: two soft finger contacts, three point contacts with friction or seven point contacts without friction. It should be noted that ultimately the object geometry determines whether this minimum is achievable. For example, no number of frictionless contacts can grasp a sphere because they cannot exert the tangential forces required to oppose rotations of the sphere about its centre.

[Cutkosky 84] describes three more complex contact models. A *curved finger* model acts much like the point contact with friction but allows rolling of contact with respect to object surface. A *very soft finger* model is deformable, adhesive, but does not roll: instead the fingertip material contacting the object is elastic. He notes that while a fingertip, such as those on the human hand, is deformable it does exhibit some plasticity and, more importantly, allows some rolling of contacts. The contact model corresponding to the human finger is called the *soft, curved finger*, and has properties lying between the extremes of the hard curved fingertip and the very soft fingertip. However, a full model of the soft, curved finger is not developed.

The relationship between the coefficient of static friction and the amount of torque that can be exerted by a single contact is non-trivial to analyse. [Howe *et al.* 88] conducted experiments on the force/torque required to cause a fingertip to slip for different contact types. The effect of friction is to place a limit on the possible wrenches, and this frictional limit can be pictured as a surface in the wrench space formed by the tangential contact forces and the wrench about the contact normal [Kao & Cutkosky 92]. This *friction limit surface* is a property of contact geometry, contact compliance, friction and the applied force. If the contact wrench lies outside this surface, the contact slips. [Xiong *et al.* 93a] model the limit surface as an half-ellipsoid, but do not derive the actual radii of the ellipsoid. [Goyal *et al.* 89] describe the derivation of the friction limit surface from contact properties.

It is interesting to note that there is a connection between the contact model and the number of contacts being considered. For instance, [Pollard 94] plans power grasps on cylindrical objects with seven contacts (two for each of the Salisbury hand's three

digits, and one for the palm [2]), and is therefore able to use the frictionless contact model to plan force closure grasps. In order to plan 3-contact force closure grasps, a frictional model must be used. In order to plan 2-contact force closure grasps, a soft finger model must be used.

In the absence of detailed knowledge of contact properties, the simplest model that will still be able to yield a force closed grasp is used. This recognises the fact that when grasping unknown objects, contact properties are too unpredictable to be modelled with any degree of accuracy. When assessing the static properties of 2-contact grasps, the soft finger contact model due to [Xiong *et al.* 93a] should be used. For grasps with 3 or more contacts, the point contact with friction should be used. If there are 7 or more contacts, as in a power grasp, it would be possible to establish whether or not some grasps would be force closed using a frictionless point contact model (as in [Pollard 94]). However, for many grasps this would not give force closure (*e.g.* grasping a brick lying on the ground) so it is advisable to use the point contact with friction model, even with 7 or more contacts, in order to avoid being overly conservative when assessing the wrench space properties of a grasp.

In order to assess the resulting complexity static equilibrium grasp synthesis for each case, the degrees of freedom of internal force (*i.e.* the dimensionality of the null space of the equilibrium grasp) can be assessed. The internal degrees of freedom for two contact types (Xiong's soft finger model and the point contact with friction) are:

	2	3	4	5
Soft finger	2	6	10	14
Point contact with friction	1	3	6	9

Note that two point contacts with friction are not sufficient to offset an arbitrary gravitational force (*i.e.* the grasp is not force closed). There are more internal degrees of freedom for the soft finger than for the point contact with friction because the soft finger can exert torque about the contact normal.

When considering manipulation of the object, more complex contact models may be required. Given a set of contacts on an object, an arbitrary motion of the object can be imparted (or countered) through:

[2] Pollard's work is one of the few to formulate a stable grasp that also satisfies the hand kinematics.

- Digit tip movement: [Nguyen 87b] models the contact stiffness [3] and allows motion of the digit tips in order to accommodate the object's motion.
- Sliding of object along contacts: [Fearing 86b] allows sliding contacts and therefore avoids the need to consider contact stiffness.
- Rolling of object about contacts.

or a combination of all three: *e.g.* [Kao & Cutkosky 92] allow digit tip movement, sliding and rolling in their quasistatic analysis, which uses a complex contact model.

3.4 Grasp Families

A grasping task can be decomposed into four components:

1. Transport: movement of wrist to some pose and position.
2. Preshape: digit posture adopted as the wrist is transported.
3. Closure: movement of digits from preshape to contact with object, with the wrist fixed.
4. Execution: contacts exert forces on the object.

The preshape and transport components can occur simultaneously. It is assumed here that the closure component occurs once the preshape is formed and the wrist is fixed.[4] Closure and execution may overlap, if some digits exert contact forces while others have not yet made contact.

3.4.1 Coupling degrees of freedom

Dextrous robot hands have a large number of degrees of freedom. In order to make the grasp planning problem tractable, these DOF are reduced by coupling the joints together. In a typical coupling, the digits will flex and extend "in parallel" and the

[3] [Mason & Salisbury 85] derives the stiffness of the digit tips from the hand's Jacobian matrix and the individual joint stiffnesses, but the contact stiffness is a property of object and hand.

[4] In human grasping, closure can start whilst the wrist is still being transported.

abduction/adduction of the digits is coupled. There are various approaches to using this coupling, each of which places different constraints on the grasp:

- Purely as a kinematic planner. This assumes that each digit stops on contact with the object, and that it can then exert a range of possible forces at each contact (this would be possible with a hand such as the Utah/MIT hand or the Salisbury hand).
- Primarily as a kinematic planner, but with some constraints placed on the force control during grasp execution, *e.g.* by requiring that the joint of abduction is not actuated during grasp execution (this is the approach taken in this thesis).
- As a kinematic and force control planner. In this approach, the same joint actuation scheme is used for the digit closure and the grasp execution. This can be divided into two sub-schemes:
  - The digits stop once they have contacted the object. When all digits have made contact the digits recommence actuation. This separates the digit closure phase from the grasp execution phase, but by keeping the same digit trajectory for each, couples the kinematic and force planning.
  - The digits are actuated continuously until the grasp is executed. This may result in pushing of the object before it is acquired in a stable grasp.

The act of stopping digits when contact is made with the object is referred to as **digit stoppage**. A contact with the object may be detected through either force sensing at the actuators (see [Mason & Salisbury 85]) or by tactile sensors at the digit tips.

3.4.2 Preshapes and digit trajectories

Preshapes have been used extensively in grasping research. Their basis in human grasping and use in robot grasping is described in Chapter 2. Two approaches to preshaping are now discussed in more detail.

[Stansfield 91] used a rulebase to derive preshapes for a Salisbury hand from a set of object measurements. Precision, lateral or encompass grasps (referred to as pinch,

grip, wrap grasps respectively) could be planned. An arbitrary digit closure was then used to execute the grasp. Bard *et al* [Bard *et al.* 93] derived preshapes for tip, lateral and encompass grasps (referred to as tip, pad and palm grasps) with a Salisbury hand given a volumetric object model (which they call an *elliptical cylinder*). Digit closure is planned so as to give a stable grasp.

In each of these cases, appeal is made to human grasping to justify use of preshaping. However, in neither of the cases is it explicitly considered what preshaping actually achieves, or how it should be derived from object shape. What the preshape achieves in Bard's and Stansfield's work will now be considered, and then a formal definition of a preshape will be formulated.

Both Bard and Stansfield use a Salisbury hand. Each digit has three degrees of freedom (two of flexion and one of abduction), to give a total of nine degrees of freedom. When mounted on a 6 DOF arm/wrist this gives us a 15 DOF configuration space in which to search for the "optimal" solution with respect to some task metric, which will depend on the hand configuration and the contact geometry.

Koehler [Koehler & Donath 88] studied the inverse kinematics of the Minnesota hand, the kinematics of which were essentially the same as those of the Salibury hand. It was shown that if the position and orientation of the distal link of each digit is fixed, the hand configuration is either kinematically inaccessible, uniquely determined, or - in a small number of pathological cases - underdetermined due to hand symmetries. This is hardly a surprising result, since 15 parameters have been fixed - 3 position and 2 orientation for each digit - in order to find a solution in a 15 DOF space. A method to determine the inverse kinematics is given. This approach is not always practical because for the hand in question, the hand kinematics are completely constrained. If the hand cannot reach the desired distal link positions/orientations, as will be the case for many contact configurations, the system fails.

Roberts [Roberts 90] solves the inverse kinematics of the Utah/MIT hand by minimisation of a cost function. The cost function is based on distance from desired position and orientation of the distal link, and on the distance of the joint angles from their limits. It is minimised using Powell's method [Press *et al.* 89] in the space of wrist posi-

tion/orientation (a six parameter space) : given a particular wrist position/orientation, the digit kinematics are calculated such that the distal links are in the desired position/orientation; if the desired digit tip location is inaccessible, a nearby location is substituted. Quaternions, with some constraints on their signs, are used to describe wrist rotation, because there is a homeomorphism between the Euclidian space and the surface of the quaternion sphere. This means that two points close to one another on the surface of the quaternion sphere are close to one another in real space.

If a single set of target grasp points is given, this approach works well. However, in reality we must optimise the grasp over a range of possible contact positions. We therefore have a much higher dimension space to search in and require a cost function based on relative contact position as well as hand kinematics. A set of well-placed contact points might be outside of the hand workspace, or close to a singularity. Conversely, a set of kinematically feasible points is not necessarily going to give a stable grasp. Search of the entire space is therefore a difficult problem.

The grasp search can be pictured as a search over the possible hand configurations subject to the constraint that the digits contact the surface of the object *i.e.* a search over the configuration space hyper-surface defined by the object (hereafter referred to as the *C-space object surface.* Each point on the C-space object surface can be given a quality metric, which embodies the task requirements, based on:

- hand kinematics (such as distance from joint limits)
- contact geometry (such as the angle between surface and distal links)
- contact mechanics (such as the minimum friction required for the grasp).

Preshapes provide an initialisation to this search. They generally have a high quality in terms of the kinematics metric and are close to, but not necessarily on, the C-space object surface. This is achieved by suitable specification of the preshape kinematics and by suitable placement of this with respect to the object features.

[Stansfield 91] and [Bard *et al.* 93] implement this using a set of rules; given the object description a heuristic is used to preshape the hand and determine the placement of the preshape. Stansfield then performs no further search; the digits are simply closed

until contact is made with the object. This is equivalent to moving along a prescribed trajectory in C-space; if this intersects with the C-space object surface the grasp is successful; if there is no intersection, then the grasp fails. With sensible heuristics, and placement of the preshape close to the object, a grasp should be successfully executed most of the time.

[Bard *et al.* 93], however, plan digit closure such that a stable grasp is achieved. In their work, the preshape is used to give a kinematically feasible initialisation for the grasp search, and then a search is conducted in the local neighbourhood of that configuration space for a point on the C-space object surface with a stability metric of sufficiently high quality.

In both these pieces of work, the joint angles of the fingers in the preshape are coupled together, to form one "virtual finger" (introduced by [Iberall 87]). This has the effect of reducing the degrees of freedom of the preshape, making it easier to fit to the object. A rulebase is then used to decide on the preshape. Stansfield specifies the preshape by two parameters — the *aperture*, which is the length of the perpendicular from thumb tip to the line joining the finger tips and the *span*, which is the distance between the two fingertips. A set of rules map from object description to preshape and hand position. Bard *et al* model the object as an "elliptical cylinder", from which the preshape and hand position are derived. A heuristic measure of stability is used to position the preshape.

Digit closure is addressed explicitly by Bard *et al*; they plan a digit closure that produces a stable grasp in simulation. Stansfield, however, ignores the issue of digit closure, making the implicit assumption that the preshape is close enough to the object that any flexion of the digits will bring them into contact with the object.

Preshaping can be compared to the approach taken by Pollard [Pollard 94], in which **grasp prototypes** are used. A **grasp prototype** specifies an optimal grasp for a given object model. An object model close to the real object is chosen and the hand is initialized at the corresponding grasp prototype. This initializes the search close to the real object C-space hypersurface. Grasp prototypes and preshapes are therefore very similar in how they constrain the grasp search. Since grasp prototypes correspond

to the final grasp, rather than a preshape, no planning of digit closure is necessary, though it does necessitate path planning to reach the desired grasping configuration.

There are therefore three approaches to using preshapes:

1. Plan the preshape, choose an arbitrary digit closure and hope that a stable grasp is achieved (Stansfield's approach).
2. Plan the preshape, then plan the digit closure such that a stable grasp is achieved (Bard's approach).
3. Plan the preshape and digit closure together (the approach in this thesis).

Specification of the preshape and digit closure implicitly specifies the path that the hand takes to achieve the final grasp. Grasps should be planned such that they are executable over the range of possible hand/object positioning errors. How a hand positioning error maps to a change in the contact positions depends on the path taken to achieve the final grasp because where the digits collide with the object depend on the path they take. It therefore follows that the planning of the final grasp should take into account the path taken by the hand to reach that grasp; if a preshape is placed such that it does not collide with the object over the range of possible hand positioning errors, then it is sufficient just to consider the path from preshape to final grasp — the digit closure.

In this thesis, methods for grasping known as **grasp families** are used. These define the digit closure trajectories and a unique mapping from idealised object geometries to preshapes, such that the task requirements are optimised.

3.4.3 Definitions

A **contact configuration** is a set of digit-object contacts, defined by their position and orientation relative to the reference frame of the hand.

A **grasp** consists of:

- a contact configuration

- a hand position
- a hand preshape
- a closure trajectory for each digit

such that the preshape does not collide with the object/environment and the digit closure trajectories do not collide with any part of the object/environment before contact.

This definition of a grasp has not placed very strict requirements on the set of possible preshapes or digit closures. These come from the task requirements and are embodied in the grasp family. A **grasp family** consists of:

- a set of contact configurations.
- a set of preshapes
- a closure trajectory for each digit.

The contact configurations of the grasp family are referred to as **ideal contact configurations**. Generally these are contact configurations for which a high quality grasp can be synthesised. A preshape is associated with each ideal contact configuration, and is the optimal grasp for that contact configuration. The preshapes therefore embody task requirements and given an ideal contact configuration, a preshape can be selected to form an optimal grasp. The digit closure trajectories allow deformation from the preshape to fit non-ideal contact configurations. The smaller the deviation from the preshape, the more optimal the grasp formed.

Grasp families constrain the range of possible digit movements whilst still allowing a sufficient number of degrees of freedom to be able to cope with a wide range of objects.

3.4.4 An example grasp family

In this section an example grasp family for the execution of lateral grasps with the SAM is presented. It is assumed that the digits can exert arbitrary force at each contact,

and therefore the grasp family is just concerned with kinematics. The grasp family is synthesised by forming a grasp for the ideal contact configuration. This corresponds to the grasp family preshape. Digit closure trajectories are then chosen to allow the grasp family to be used with non-ideal contact configurations, and to let the preshape to be expanded in order to allow for errors in preshape placement and finite digit widths.

One ideal contact configuration is shown in Figure 3.7(a). The asterisks show the contact positions, and the dotted lines the contact normals (pointing into the object). The task requirements are given in Section 3.5. The distinctive feature of a lateral grasp is that the distal link of each digit should be perpendicular to the contact normal.

Figure 3.7(b) shows the preshape. This is uniquely determined by the task requirements and the ideal contact set.

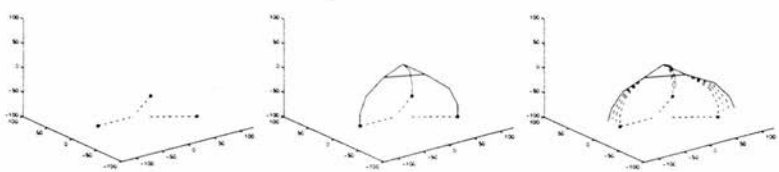


Figure 3.7: Grasp family: (a) ideal contact set, (b) preshape, (c) digit closure trajectories.

Figure 3.7(c) shows the digit closure trajectory. Here, each digit was actuated at the proximal joint. This is generally a good choice for a lateral grasp because often the distal joint can be close to its upper limit, preventing expansion from the task-oriented preshape by actuation of that joint.

The grasp is synthesised for a set of ideal contact configurations. A question therefore arises: what happens if the object to be grasped does not contain an ideal contact set? A grasp family could be synthesised for each set of possible contacts. However, the inverse kinematics of the hand has to be calculated for each set, and this may be non-trivial (or the contact set may lie outside of the hand's workspace). Furthermore, this has to be done separately for each contact set on the object. An alternative is to choose the grasp contact set that fits most closely the ideal contact set, which is

the approach taken in this thesis. In Chapter 5 a metric is defined which enables the distance "away" from the ideal contact set to be measured. To use a grasp family, the following steps are taken:

1. Find the ideal contact configuration that best "fits" the object.
2. Fit a preshape to the ideal contact configuration.
3. Choose a digit trajectory.
4. Find intersection of digit trajectory with object to give the final grasp.

3.5 Grasp Families

Six three-digit grasping families are given for the SAM. These are derived from three types of preshapes and two types of digit closure trajectories. It is assumed that the joint actuation during grasp execution can be different to that during digit closure.

It is assumed that all joints can be position controlled, and that the joints of flexion can be stiffness controlled. In stiffness control, a perturbation of fingertip position, $\delta \boldsymbol{x}$, produces a force

$$\delta \boldsymbol{f} = \boldsymbol{K_x} \delta \boldsymbol{x} \tag{3.4}$$

where

$$\boldsymbol{K_x} = (\boldsymbol{J} \boldsymbol{K_\theta^{-1}} \boldsymbol{J}^T)^{-1} \tag{3.5}$$

$\boldsymbol{K_x}$ is the cartesian stiffness matrix, $\boldsymbol{K_\theta}$ is the stiffness matrix of the joints and $\boldsymbol{J}$ is the Jacobian.

All the 3-contact grasp families for the SAM have two task requirements in common:

- *Minimum friction condition*: the grasp should be in equilibrium for frictionless contacts.

- *Force control condition*: the contact forces should be exerted only by flexion of the digit and not by abduction, *i.e.* the contact forces must lie in the plane of the digit. This simplifies control of the grasp execution and enables the abduction of the fingers to be just position controlled.

These are enough to constrain the preshape to within 1 DOF, which determines the flexion of the digits. Each grasp family has a third task requirement that specifies this DOF.

3.5.1 Derivation of the grasp families

In this section the six grasp families are derived. This is a five step process:

1. A generic preshape is introduced, which defines a coupling used in all preshapes of the SAM hand.
2. Sets of ideal contact configurations are generated to fit the generic preshape.
3. Task-specific preshapes are generated for tip, lateral and manipulation grasps.
4. Two types of digit closure trajectories are introduced.
5. Bounds on the sets of ideal contact configurations are derived from the task-specific preshapes.

The generic preshape

In all the preshapes used, certain joints of the SAM are coupled (in the absence of external forces). Each digit has equal angles of flexion (α, β, γ at the proximal, middle and distal joints respectively). The fingers abduct by equal amounts (in opposite sense). Within each digit, the distal and middle joints are coupled such that $\gamma = \frac{2}{3}\beta$, as is approximately true for unopposed motion of the human hand. See Figure 3.8 This coupling of joints could be hard-wired into the hand design, in which case they would have a strong influence on how the object could be manipulated, or just used as modes of movement to facilitate the planning of the grasps.

The joint limits of the SAM are as follows:

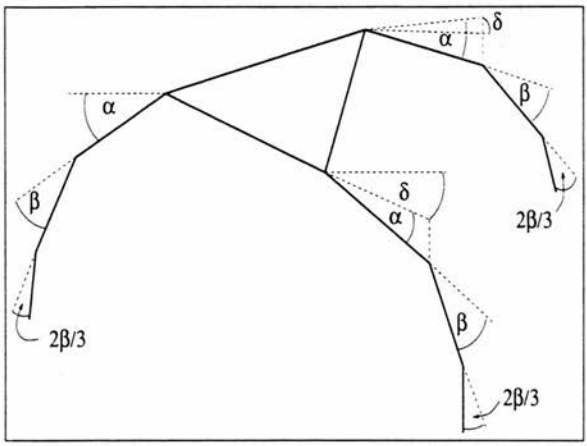


Figure 3.8: The generic preshape for the SAM, showing the coupled DOF.

- $0 \leq \alpha \leq 1.91$
- $0 \leq \beta \leq \pi/2$
- $0 \leq \delta \leq \pi/4$

Ideal Contact Configurations

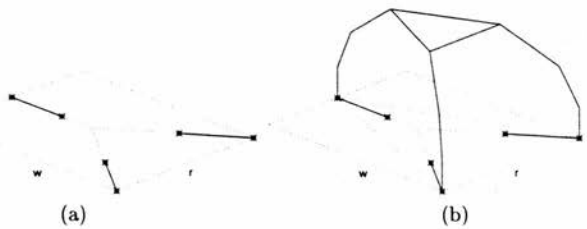


Figure 3.9: An ideal contact configuration. Contact positions and normals are shown. In (a) the meeting point of these normals are shown — this means that a frictionless equilibrium grasp is possible. In (b) one preshape that fits the ideal contact configuration is shown.

The ideal contact configurations are generated from the generic preshape shown in

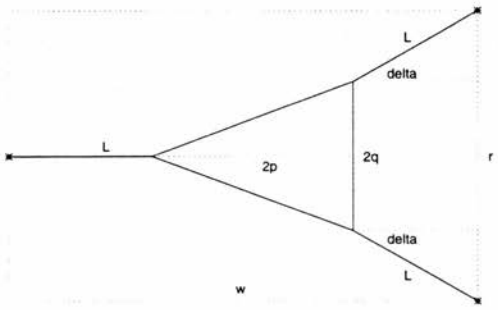


Figure 3.10: An ideal contact configuration viewed perpendicular to the plane of the wrist.

Figure 3.8. Each of the contact configurations has contacts lying at the corners of the isosceles triangle formed by the positions of the fingertips in the preshape. Figure 3.9(a) shows the contact positions and normals. The contact positions are fixed by the two distances shown as r and w. The contact normals meet at a common point in the plane of the contacts (in order to satisfy the minimum friction condition) and each contact normal should lie in the plane of the corresponding digit (in order to satisfy the force control condition). This latter condition means that the orientation of the contact normal depends on the hand configuration in the preshape. This will now be derived.

Given that each digit has identical angles of flexion and the two fingers have identical angles of abduction, the preshape that fits this contact configuration will look something like that shown in Figure 3.9(b). The abduction of the preshape is fixed by the relative contact positions. The flexion of the preshape may vary. The flexion of the preshape does not affect the contact normals, because it is assumed that each digit can exert any force in the plane of the digit. However, the abduction of the preshape does affect the contact normals since the contact normal must lie in the plane of the corresponding digit. In fact, the angle between the thumb contact normal and the finger contact normal is *equal* to the angle of abduction of the preshape. In order to derive this angle, the angle of abduction of the preshape is calculated. This is done by considering an overhead view of a preshape that fits the ideal contact configuration.

This is shown in Figure 3.10. From this overhead viewpoint, all preshapes, whatever their flexion, appear identical.

There are two unknowns in this diagram: L and δ. By elimination of L, the angle of abduction δ is given by:

$$w = 2p + \frac{1}{2\sin\delta}(r - 2q)(1 + \cos\delta) \tag{3.6}$$

unless $r = 2q$, in which case $\delta = 0$.

For a given set of contact positions (w, r), the angle between each finger normal and the thumb normal is equal to the δ in Equation 3.6.

Task-specific preshapes

In the previous section, the derivation of the set of ideal contact configurations for a generic preshape was shown. Given one such ideal contact configuration, it has been shown that the abduction of the preshape is fixed. One DOF remains to be set: the flexion of the preshape, that depends on the particular task requirements.

Figure 3.11 shows the three types of preshape. They are all for precision grasps. The TIP and LATERAL preshapes are for planning tip and lateral grasps respectively (see Section 2.2). The MANIPULATION preshape specifies the hand kinematics rather than the contact type, and is for when subsequent manipulation of the object is required with no strong preference for tip or lateral contact types.

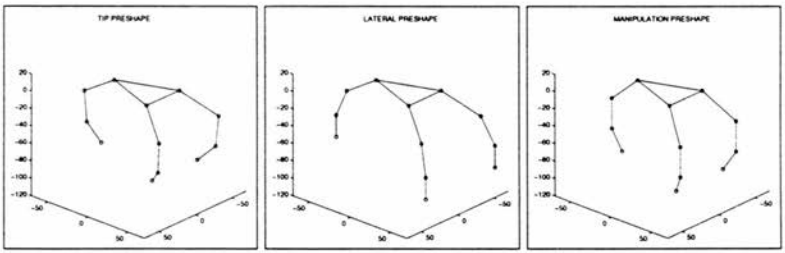


Figure 3.11: TIP, LATERAL and MANIPULATION preshapes

The preshapes are determined by the following task-specific requirements:

- The TIP preshape keeps the angle between the contact surface and the distal link of each digit at $\frac{\pi}{4}$, *i.e.* $\alpha + \beta + \gamma = \frac{3\pi}{4}$. This gives a contact surface at the tip of each digit.

- The LATERAL preshape keeps the distal link of each digit parallel to the contact surface, *i.e.* $\alpha + \beta + \gamma = \frac{\pi}{2}$. This gives a contact surface along the length of the distal link of each digit.

- The MANIPULATION preshape minimizes the sum of the squares of the deviations of the joints from their central values, *i.e.* it minimizes $(\alpha - \alpha_c)^2 + (\beta - \beta_c)^2 + (\gamma - \gamma_c)^2$. This gives a preshape which has maximum leeway for object manipulation between the digits. The angle of the distal link is constrained so that it cannot point away from the "centre" of the preshape, *i.e.* $\alpha + \beta + \gamma \geq \frac{\pi}{2}$. This is done to make it less likely that collisions occur between the object surface and the hand. It is useful if there is no preference for a tip or lateral grasp.

Each preshape is therefore defined by just two variables — one controlling the flexion of the digits, another controlling the abduction of the fingers. Such preshape classifications were first introduced in [Wren & Fisher 95b] and refined in [Wren & Fisher 95a]. [5] In that paper, however, the tip and lateral preshapes were allowed to deform from the above configurations, enabling the preshapes to have a wider range of fingertip positions. This is not done here, and the digit closure is relied upon to extend the range of the preshapes. The reason for this is that the metric for fitting the preshape uses the difference between final grasp and preshape as an indicator of how task-specific the grasp is, so the preshape should always conform to the optimal hand configuration.

The choice between TIP, LATERAL and MANIPULATION preshapes depends solely on the task-requirements. However, the grasps will have to deform from the preshape in order to fit to a given object. Generally, the closer the grasp to the preshape, the better the grasp. Choice of preshape therefore specifies a *preference*, not a firm constraint, and ultimately the type of grasp is always strongly influenced by object

[5] In these papers, however, the TIP preshape was called a *precision* preshape.

size and shape; *e.g.* it is not possible to execute a lateral grasp on an object which is very small compared to the hand, nor is it possible to execute a tip grasp on an object which is large compared to the hand.

For an arbitrary object, *the closer that the final contact configuration configuration is to the ideal contact configuration, the better the task-specificity of the grasp.*

Digit Trajectories

The digit closure trajectory is a function of the hand's joint rates. The following constraints are placed on the digit closure:

- The digits may only flex during closure; there should be no abduction.
- In the reference frame of the digit, each digit trajectory is identical.
- The joint rates are constant during the closure.

Figure 3.12 shows the two categories of digit trajectory. Figure 3.12(a) shows the PROXIMAL digit trajectory, in which the digits are flexed at the proximal joints, and Figure 3.12(b) shows the DISTAL digit trajectory, in which the digits are flexed at the middle and distal joints. Each digit trajectory is defined by just one variable, since the middle and distal joints are coupled. The trajectories give a digit closure motion similar to that of the human hand and are easy to control because they only involve actuation of 1 DOF per digit.

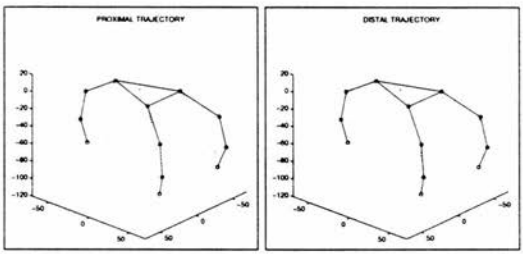


Figure 3.12: PROXIMAL and DISTAL digit trajectories.

The TIP, LATERAL and MANIPULATION preshapes can be combined with either the DISTAL or PROXIMAL digit trajectories to give 6 different grasp families. The type of digit trajectory can be predetermined in advance, or chosen online in order to fit the preshape as closely as possible to the object to be grasped.

Grasp families are given double-barreled names. The first name refers to the preshape category used, the second to the digit trajectory used. This is followed by a number denoting the number of digits involved in the grasp. *e.g.* Lateral-Proximal-3 is a 3 digit grasp family generated from a LATERAL preshape and PROXIMAL digit trajectory. In this thesis, however, all grasps planned are 3-digit grasps, so the final number is dropped.

Bounds on the sets of ideal contact configurations

Naturally there are bounds on the set of ideal contact configurations for each grasp family. Given a set of contact positions as specified by (r, w), it has already been shown that there is only one possible set of contact normals, for which the angle between finger normals and thumb normal must be the same as the abduction of the hand, δ, as defined by Equation 3.6. This means that the abduction joint limits map directly to a limit on the contact normal orientation.

The bounds on (r, w) depend on the bounds on L (see Figure 3.10), which depends on the type of preshape used. L is the length of the digit in the plane of the contacts, and is given by:

$$L = a\cos\alpha + b\cos(\alpha + \beta) + c\cos(\alpha + \frac{5}{3}\beta) \tag{3.7}$$

where a, b, c are the lengths of the proximal, middle and distal links respectively. The bounds on L for different types of preshape are shown below:

	L_{min}	L_{max}
Tip	-34	33
Lateral	0	66
Manipulation	-105	66

Given contact normal orientation δ, the lengths r_{min} and w_{min} are related to L_{min} by

the following equations:

$$\begin{aligned} r_{min} &= 2q + 2L_{min}\sin\delta \\ w_{min} &= 2p + L_{min}(1+\cos\delta) \end{aligned} \tag{3.8}$$

Similarly,

$$\begin{aligned} r_{max} &= 2q + 2L_{max}\sin\delta \\ w_{max} &= 2p + L_{max}(1+\cos\delta) \end{aligned} \tag{3.9}$$

3.5.2 Inverse kinematics

In the grasp families the kinematics of the robot hand are coupled in order to make grasp planning more tractable. This affects the workspace of the hand. The task specific preshapes have 2 DOF (one of flexion and one of abduction), the wrist position 6 DOF, and each of the three digits has a single DOF during digit closure. This gives 11 DOF for the entire hand.

It is sometimes convenient to constrain the hand further, by restricting the wrist to lie in a plane. As discussed in Chapter 1 one of the advantages of a dextrous hand is that it can stably grasp objects from a wide variety of approaches. This is therefore not too much of a restriction to impose. The wrist is therefore restricted to lie parallel to the xy plane, which gives the wrist DOF in (x, y, z, ϕ_z) and the whole hand 9 DOF. In Chapter 5 a reasonably fast algorithm to calculate these 9 DOF is presented. 9 DOF may also be convenient to work with because the positions of the digit tips define 9 DOF (3 DOF per digit tip position) which means that the Jacobian relating joint movement to tip position is square and therefore invertible away from singularities. However, it will be seen that singularities occur quite often for the grasp families.

Proximal digit closure gives a final grasp with the following DOF (see Figure 3.13). The joint notation of Section 3.2 is used.

- $\theta_1^1 = \theta_1^2 = \delta$

- $\theta_1^0 = \alpha_0, \theta_2^1 = \alpha_1, \theta_2^2 = \alpha_2$
- $\theta_2^0 = \theta_3^1 = \theta_3^2 = \beta$
- $\theta_3^0 = \theta_4^1 = \theta_4^2 = 2\beta/3$
- Wrist at $(x, y, z, 0, 0, \phi_z)$

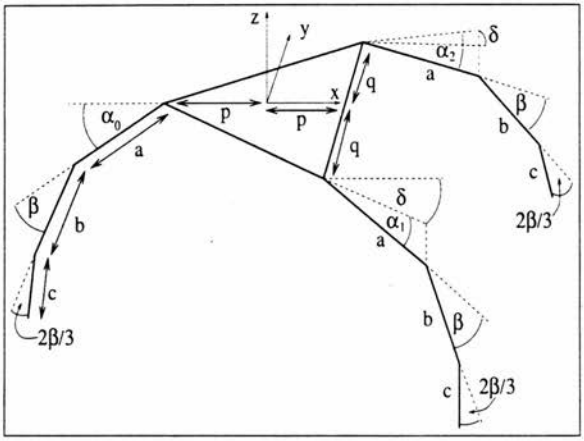


Figure 3.13: SAM showing coupled DOF for proximal grasp family.

The digit tip positions for the proximal grasp family, $\boldsymbol{x} = [x_0, y_0, z_0, x_1, y_1, z_1, x_2, y_2, z_2]^T$, are as follows:

$$\mathbf{x} = \begin{bmatrix} x - (p + a\cos\alpha_0 + b\cos(\alpha_0 + \beta) + c\cos(\alpha_0 + \frac{5}{3}\beta))\cos\phi_z \\ y - (p + a\cos\alpha_0 + b\cos(\alpha_0 + \beta) + c\cos(\alpha_0 + \frac{5}{3}\beta))\sin\phi_z \\ z - (a\sin\alpha_0 + b\sin(\alpha_0 + \beta) + c\sin(\alpha_0 + \frac{5}{3}\beta) \\ x + p\cos\phi_z + q\sin\phi_z + (a\cos\alpha_1 + b\cos(\alpha_1 + \beta) + c\cos(\alpha_1 + \frac{5}{3}\beta))\cos(\phi_z - \delta) \\ y + p\sin\phi_z - q\cos\phi_z + (a\cos\alpha_1 + b\cos(\alpha_1 + \beta) + c\cos(\alpha_1 + \frac{5}{3}\beta))\sin(\phi_z - \delta) \\ z - (a\sin\alpha_1 + b\sin(\alpha_1 + \beta) + c\sin(\alpha_1 + \frac{5}{3}\beta)) \\ x + p\cos\phi_z - q\sin\phi_z + (a\cos\alpha_2 + b\cos(\alpha_2 + \beta) + c\cos(\alpha_2 + \frac{5}{3}\beta))\cos(\phi_z + \delta) \\ y + p\sin\phi_z - q\cos\phi_z + (a\cos\alpha_2 + b\cos(\alpha_2 + \beta) + c\cos(\alpha_2 + \frac{5}{3}\beta))\sin(\phi_z + \delta) \\ z - (a\sin\alpha_2 + b\sin(\alpha_2 + \beta) + c\sin(\alpha_2 + \frac{5}{3}\beta)) \end{bmatrix}$$

If the joint vector $\boldsymbol{\psi} = [x, y, z, \phi, \beta, \delta, \alpha_0, \alpha_1, \alpha_2]^T$ (setting $\phi_z = \phi$), the Jacobian is given by:

$$
\mathbf{J} = \begin{bmatrix}
1 & 0 & 0 & (p+F_0)\sin\phi & G_0\cos\phi & 0 & E_0\cos\phi & 0 & 0 \\
0 & 1 & 0 & -(p+F_0)\cos\phi & G_0\sin\phi & 0 & E_0\sin\phi & 0 & 0 \\
0 & 0 & 1 & 0 & -H_0 & 0 & -F_0 & 0 & 0 \\
1 & 0 & 0 & -p\sin\phi + q\cos\phi \\ -F_1\sin(\phi-\delta) & -G_1\cos(\phi-\delta) & F_1\sin(\phi-\delta) & 0 & -E_1\cos(\phi-\delta) & 0 \\
0 & 1 & 0 & p\cos\phi + q\sin\phi \\ +F_1\cos(\phi-\delta) & -G_1\sin(\phi-\delta) & F_1\sin(\phi-\delta) & 0 & -E_1\sin(\phi-\delta) & 0 \\
0 & 1 & 0 & -H_1 & 0 & 0 & -F_1 & 0 \\
1 & 0 & 0 & -p\sin\phi + q\cos\phi \\ -F_2\sin(\phi+\delta) & -G_2\cos(\phi+\delta) & F_2\sin(\phi+\delta) & 0 & 0 & -E_2\cos(\phi+\delta) \\
0 & 1 & 0 & p\cos\phi - q\sin\phi \\ +F_2\cos(\phi+\delta) & -G_2\sin(\phi+\delta) & F_2\cos(\phi+\delta) & 0 & 0 & -E_2\sin(\phi+\delta) \\
0 & 0 & 1 & 0 & -H_2 & 0 & 0 & 0 & -F_2
\end{bmatrix}
$$

where

$$
\begin{aligned}
E_i &= a\sin\alpha_i + b\sin(\alpha_i+\beta) + c\sin(\alpha_i+\tfrac{5}{3}\beta) \\
F_i &= a\cos\alpha_i + b\cos(\alpha_i+\beta) + c\cos(\alpha_i+\tfrac{5}{3}\beta) \\
G_i &= b\sin(\alpha_i+\beta) + \tfrac{5}{3}c\sin(\alpha_i+\tfrac{5}{3}\beta) \\
H_i &= b\cos(\alpha_i+\beta) + \tfrac{5}{3}c\cos(\alpha_i+\tfrac{5}{3}\beta)
\end{aligned}
$$

This becomes singular (and therefore non-invertible) at $\beta_1 = \beta_2$, which happens to be the preferred positions of the grasp family, because it corresponds to a symmetrical configuration of fingers. Figure 3.14 shows one such singularity: the configuration of the hand is $\psi = [0, 0, 0, 0, \pi/4, 0, \pi/6, \pi/6, \pi/6]^T$. In this configuration the rank of the Jacobian is 8, and the digit tip velocity, $\dot{\boldsymbol{x}}$ can only be orthogonal to some direction $\dot{\boldsymbol{x}}_s$ that is a function of the joint angles. $\dot{\boldsymbol{x}}_s$ can be calculated from the **SVD** of the Jacobian, and is shown in Figure 3.14 as a line at each digit tip (bear in mind that in fact this only holds for small deviations from the current configuration). This is also going to cause problems because for joint values near the singularity, some desired changes in digit tip position will lead to large changes in joint values. This can be overcome using the Singularity Robust Inverse (see Chapter 2 for a definition of this), by increasing the DOF of the grasp family or by simply avoiding the issue altogether by using forward kinematics for planning the grasp kinematics, which is the approach taken in this thesis (see Chapter 5). It is *not* a problem for the execution of grasps in a preshape because the coupling of the DOF is for the kinematic planning only: each digit can still exert an arbitrary force anywhere in the plane of the digit, which enables equilibrium grasps (see below).

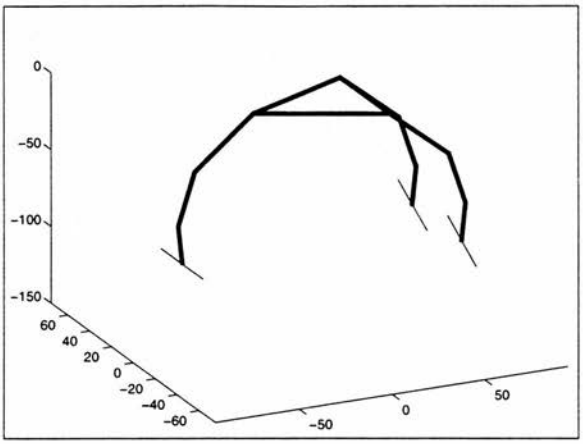


Figure 3.14: SAM proximal grasp family at a singularity of its Jacobian. The hand configuration is $\theta = [0, 0, 0, 0, \pi/4, 0, \pi/6, \pi/6, \pi/6]^T$. The rank of the Jacobian is 8, and the change in digit tip position vector, $\delta \boldsymbol{x}$, can only be orthogonal to $\delta \boldsymbol{x}_s$. $\delta \boldsymbol{x}_s$ can be calculated from the **SVD** of the Jacobian, and is shown as a line at each digit tip (bear in mind that in fact this only holds for small deviations from the current configuration).

Distal digit closure yields a final grasp with the following DOF (see Figure 3.15):

- $\theta_1^1 = \theta_1^2 = \delta$
- $\theta_1^0 = \theta_2^1 = \theta_2^2 = \alpha$
- $\theta_2^0 = \beta_0, \theta_3^1 = \beta_1, \theta_3^2 = \beta_2$
- $\theta_3^0 = 2\beta_0/3, \theta_4^1 = 2\beta_1/3, \theta_4^2 = 2\beta_2/3$
- Wrist at $(x, y, z, 0, 0, \phi_z)$

The digit tip positions for the distal grasp family, $\boldsymbol{x} = [x_0, y_0, z_0, x_1, y_1, z_1, x_2, y_2, z_2]^T$, are as follows:

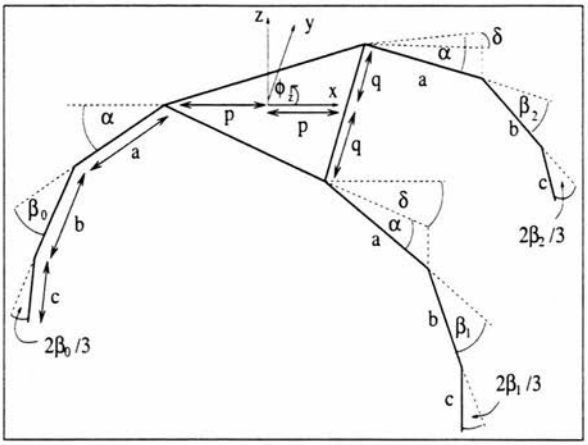


Figure 3.15: SAM showing coupled DOF for distal grasp family.

$$\mathbf{x} = \begin{bmatrix} x - (p + a\cos\alpha + b\cos(\alpha+\beta_0) + c\cos(\alpha+\frac{5}{3}\beta_0))\cos\phi_z \\ y - (p + a\cos\alpha + b\cos(\alpha+\beta_0) + c\cos(\alpha+\frac{5}{3}\beta_0))\sin\phi_z \\ z - (a\sin\alpha + b\sin(\alpha_0+\beta_0) + c\sin(\alpha+\frac{5}{3}\beta_0) \\ x + p\cos\phi + q\sin\phi + (a\cos\alpha + b\cos(\alpha+\beta_1) + c\cos(\alpha+\frac{5}{3}\beta_1))\cos(\phi-\delta) \\ y + p\sin\phi - q\cos\phi + (a\cos\alpha + b\cos(\alpha+\beta_1) + c\cos(\alpha+\frac{5}{3}\beta_1))\sin(\phi-\delta) \\ z - (a\sin\alpha + b\sin(\alpha+\beta_1) + c\sin(\alpha+\frac{5}{3}\beta_1)) \\ x + p\cos\phi + q\sin\phi + (a\cos\alpha + b\cos(\alpha+\beta_2) + c\cos(\alpha+\frac{5}{3}\beta_2))\cos(\phi+\delta) \\ y + p\sin\phi - q\cos\phi + (a\cos\alpha + b\cos(\alpha+\beta_2) + c\cos(\alpha+\frac{5}{3}\beta_2))\sin(\phi+\delta) \\ z - (a\sin\alpha + b\sin(\alpha+\beta_2) + c\sin(\alpha+\frac{5}{3}\beta_2)) \end{bmatrix}$$

If the joint vector $\psi = [x, y, z, \phi, \alpha, \delta, \beta_0, \beta_1, \beta_2]^T$ (setting $\phi_z = \phi$), the Jacobian $\mathbf{J}$,

such that $\delta \boldsymbol{x} = \boldsymbol{J}\delta\boldsymbol{\psi}$, is given by:

$$
\mathbf{J} = \begin{bmatrix}
1 & 0 & 0 & (p+B_0)\sin\phi & A_0\cos\phi & 0 & C_0\cos\phi & 0 & 0 \\
0 & 1 & 0 & -(p+B_0)\cos\phi & A_0\sin\phi & 0 & C_0\sin\phi & 0 & 0 \\
0 & 0 & 1 & 0 & -B_0 & 0 & -D_0 & 0 & 0 \\
1 & 0 & 0 & -p\sin\phi + q\cos\phi - B_1\sin(\phi-\delta) & -A_1\cos(\phi-\delta) & B_1\sin(\phi-\delta) & 0 & -C_1\cos(\phi-\delta) & 0 \\
0 & 1 & 0 & p\cos\phi + q\sin\phi + B_1\cos(\phi-\delta) & -A_1\sin(\phi-\delta) & -B_1\cos(\phi-\delta) & 0 & -C_1\sin(\phi-\delta) & 0 \\
0 & 0 & 1 & 0 & -B_1 & 0 & 0 & -D_1 & 0 \\
1 & 0 & 0 & -p\sin\phi + q\cos\phi - B_2\sin(\phi+\delta) & -A_2\cos(\phi+\delta) & B_2\sin(\phi+\delta) & 0 & 0 & -C_2\cos(\phi+\delta) \\
0 & 1 & 0 & p\cos\phi - q\sin\phi + B_2\cos(\phi+\delta) & -A_2\sin(\phi+\delta) & B_2\cos(\phi+\delta) & 0 & 0 & -C_2\sin(\phi+\delta) \\
0 & 0 & 1 & 0 & -B_2 & 0 & 0 & 0 & -D_2
\end{bmatrix}
$$

where

$$
\begin{aligned}
A_i &= a\sin\alpha + b\sin(\alpha+\beta_i) + c\sin(\alpha+\tfrac{5}{3}\beta_i) \\
B_i &= a\cos\alpha + b\cos(\alpha+\beta_i) + c\cos(\alpha+\tfrac{5}{3}\beta_i) \\
C_i &= b\sin(\alpha+\beta_i) + \tfrac{5}{3}c\sin(\alpha+\tfrac{5}{3}\beta_i) \\
D_i &= b\cos(\alpha+\beta_i) + \tfrac{5}{3}c\cos(\alpha+\tfrac{5}{3}\beta_i)
\end{aligned}
$$

As before, the Jacobian is singular when $\beta_1 = \beta_2$. One such singular configuration is shown in Figure 3.16.

3.5.3 Stability

It was seen above that there are singularities in the grasp families which complicate the kinematic planning. However, it is assumed that the coupling of the grasp family is relaxed to some extent for the grasp execution: the digits cannot exert forces by abduction, but they can exert forces by flexion of the proximal and/or middle joints (the distal joint is coupled to the middle joint) which enables them to exert an arbitrary force in the plane of the digit, except at singularities.

If a digit i has proximal joint angle α_i and middle joint angle β_i, its Jacobian in the plane is:

$$
\mathbf{J_i} = \begin{bmatrix} A_i & C_i \\ -B_i & -D_i \end{bmatrix} \tag{3.10}
$$

where

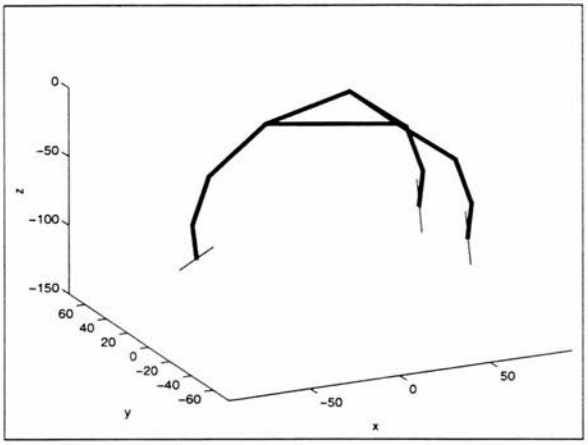


Figure 3.16: SAM distal grasp family at a singularity of its Jacobian. The hand configuration is $\theta = [0, 0, 0, 0, \pi/4, 0, \pi/6, \pi/6, \pi/6]^T$. The rank of the Jacobian is 8, and the change in digit tip position vector, δx, can only be orthogonal to δx_s. δx_s can be calculated from the **SVD** of the Jacobian, and is shown as a line at each digit tip (bear in mind that in fact this only holds for small deviations from the current configuration).

$$
\begin{aligned}
A_i &= a\sin\alpha_i + b\sin(\alpha_i + \beta_i) + c\sin(\alpha_i + \tfrac{5}{3}\beta_i) \\
B_i &= a\cos\alpha_i + b\cos(\alpha_i + \beta_i) + c\cos(\alpha_i + \tfrac{5}{3}\beta_i) \\
C_i &= b\sin(\alpha_i + \beta_i) + \tfrac{5}{3}c\sin(\alpha_i + \tfrac{5}{3}\beta_i) \\
D_i &= b\cos(\alpha_i + \beta_i) + \tfrac{5}{3}c\cos(\alpha_i + \tfrac{5}{3}\beta_i)
\end{aligned}
$$

and the joint torques for a required force $\mathbf{f}$ are given by

$$\boldsymbol{\tau} = \boldsymbol{J}^{\boldsymbol{T}}\boldsymbol{f} \tag{3.11}$$

unless $\mathbf{J}$ is singular, which only occurs at $\beta_i = 0$.

In order to execute an equilibrium grasp, the digit must just be able to exert some force in the plane of the digit: because the abduction of the fingers is always equal in a grasp, it is not necessary to exert forces out of the plane of the digit. Away from singularities, arbitrary forces can be exerted in the plane of the digit. At the singularity, the digit can still exert planar forces perpendicular to the distal link. This is very important

because such a singular configuration will arise for the lateral grasp on an object the same width as the palm of the hand.

3.5.4 Other hand models

In this section it is explained how to produce grasp families for the Salisbury and human hands, taking the SAM as a starting point. In Chapter 5 it will be seen that the complexity of the grasp planning algorithm is dependent on the number of DOF in the *preshape* rather than the grasp family itself, because the DOF of the digit trajectories are resolved by intersection with the object surface. In the preshape, typically, all the degrees of flexion of the digits are coupled to one variable and the degrees of abduction of the fingers (but not the thumb) are coupled to one variable. This gives a preshape with 8 DOF, including the 6 DOF for the wrist position.

Salisbury Hand

The grasping families of the SAM map naturally to the Salisbury hand to give a preshape with the same degree of complexity. In the SAM, after the distal and middle joints of each digit have been coupled, each digit has 2 DOF, just as in the Salisbury hand. The abduction of the Salisbury thumb can be set to zero, and the abduction of the fingers coupled and the Salisbury hand now has the same number of DOF of flexion and abduction as the SAM. Some functionality is lost by fixing the thumb abduction to zero, but this is primarily of use in 2-digit grasps (indeed, for 2-digit grasps, it is very important to have this degree of abduction in the thumb).

Human Hand

The human hand differs from the SAM in two important ways. Firstly, the thumb has an extra degree of freedom, corresponding to abduction about the proximal joint. Secondly, there are four fingers as opposed to two in the SAM. The human hand preshape has the same number of DOF as the SAM preshape if the abduction of the thumb is fixed, the abduction of the fingers is coupled and the flexion of all digits is coupled.

However, due to the placement of the human thumb relative to the fingers, the ability of the thumb to abduct to give radically different preshapes is important. For this reason, the thumb should be allowed to abduct, which gives the human preshape one more DOF than the SAM preshape. If the DOF of the preshape need to be kept low, then it is better to fix the abduction of the fingers rather than the abduction of the thumb. Indeed, this is what is done in the Belgrade/USC hand. This is not unreasonable because there are four fingers, each finger contact is less critical than in the SAM hand.

3.6 Coupling of Digit Closure and Grasp Execution

In Section 3.5 the planning of grasp families was only considered from a kinematic point of view: the digit closure and grasp execution were decoupled and the grasp families produced grasps which were optimal under the assumption that the digits could exert suitable forces during grasp execution. In this section is presented a method of grasp family derivation which assumes that the digit closure and grasp execution are coupled. For clarity, a 2D hand model is used, with only 2 links per finger.

To devise a grasp family it is necessary to make some assumptions about the geometry of the object. The following grasp properties are required:

- Precision grasp.
- Amount of friction required for an equilibrium grasp is minimised.

and the following are assumed:

- Rectangular object.
- Zero object weight.
- Perfect object model.
- Perfect positioning of hand.

Consider the 2D hand in Figure 3.17.

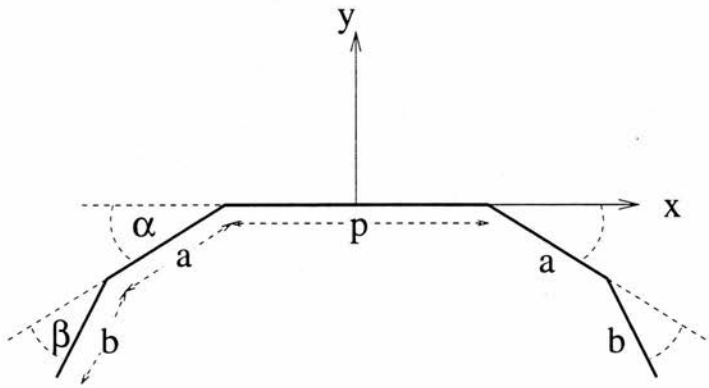


Figure 3.17: 2D Hand Model

Once the wrist is fixed, each digit only has two degrees of freedom. The position of the distal link of the lefthand digit is given by:

$$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} -(\frac{p}{2} + a\cos\alpha + b\cos(\alpha+\beta)) \\ -(a\sin\alpha + b\sin(\alpha+\beta)) \end{bmatrix} \tag{3.12}$$

The Jacobian for this digit is therefore given by:

$$\mathbf{J} = \begin{bmatrix} (a\sin\alpha + b\sin(\alpha+\beta)) & b\sin(\alpha+\beta) \\ -(a\cos\alpha + b\cos(\alpha+\beta)) & -b\cos(\alpha+\beta) \end{bmatrix} \tag{3.13}$$

To exert a force $\mathbf{f}$ at the digit tip, the required torque is:

$$\tau = J^T\mathbf{f} = \begin{bmatrix} (a\sin\alpha + b\sin(\alpha+\beta)) & -(a\cos\alpha + b\cos(\alpha+\beta)) \\ b\sin(\alpha+\beta) & -b\cos(\alpha+\beta) \end{bmatrix} \begin{bmatrix} f_x \\ f_y \end{bmatrix} \tag{3.14}$$

Given a grasp with a particular hand configuration $\theta_g = (\alpha_g, \beta_g)$, the desired force at the contacts and the stiffness of each joint, it is possible to derive the joint displacement input that is required to exert this force. The joint torque for a given joint displacement input is:

$$\tau = K\delta\theta_g \tag{3.15}$$

This can be written as:

$$J_g^T f = K\delta\theta_g \tag{3.16}$$

where J_g is the Jacobian of the grasp configuration, and assuming constant stiffness. The desired force can therefore be achieved by actuation of the joints to give the following displacement:

$$\delta\theta_g = K^{-1} J_g^T f \tag{3.17}$$

If the digit actuation during unopposed digit closure can be different to that during grasp execution, this places no constraints on digit closure, and therefore no constraints on preshape formation. If the joint rates during digit closure are the same as those during grasp execution, the required joint rate during digit closure must be proportional to $\delta\theta_g$. The relative joint rate is therefore fixed by J_g (and hence the hand configuration in the grasp) and f, which must be directed along the x axis to give an equilibrium grasp. The following *relative* joint rates are therefore required:

$$\begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} (a\sin\alpha_g + b\sin(\alpha_g + \beta_g))/k_\alpha \\ b\sin(\alpha_g + \beta_g)/k_\beta \end{bmatrix} \tag{3.18}$$

For a given grasp, it has been shown there is a single relative joint rate that will give an equilibrium grasp. Contrast this with the case where digit closure and grasp execution are *not* coupled, where the joint rate can be chosen.

The final grasp configuration and the joint rate during digit closure have been determined. The preshape should be chosen that over the range of possible errors in wrist position, the hand does not collide with the object. The joint angles in the preshape are given by:

$$\begin{bmatrix} \alpha_0 \\ \beta_0 \end{bmatrix} = \begin{bmatrix} \alpha_g \\ \beta_g \end{bmatrix} - \lambda \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \end{bmatrix} \tag{3.19}$$

How big λ must be depends on the uncertainty in object model and in wrist position.

3.7 Summary

In this chapter grasp families have been introduced. The problem of grasp planning was discussed and preshapes were presented as a sensible way of localising the search for a good grasp. Consideration of digit trajectories was then advocated as a way of binding a given preshape to an actual grasp. The grasp family is a prescribed set of preshapes and digit trajectories with a set of ideal contact configurations, for which the grasp family synthesises an optimal grasp. Three grasp families were presented for the SAM, and their inverse kinematics were analysed. It was shown that singularities occur at preferred grasping configurations, which can complicate the kinematic planning and mean that a forward kinematic planner may be preferable to an inverse kinematic planner. The coupling of the grasp family is relaxed for grasp execution, however, so the singularities do not prevent an equilibrium grasp from being formed.

The chapter also included a description of a dextrous hand kinematic modeler which can readily be used to describe a range of different hand designs. An alternative approach to the use of grasp families was presented, in which the force control and kinematic planning phases of the grasp were linked.

Chapter 4

Acquisition of the object model

As noted in Chapter 2, there has been little consideration of the type of object model that is required for grasp planning. In this chapter this question is addressed and the construction of a suitable model from laser striper range data is described. It is assumed that the object has already been localized and lies on a horizontal ground plane. The object model consists of a surface model and a volumetric model. The surface model is built using a segmentation system constructed for the Imagine II object recognition system [Trucco & Fisher 95]. The volumetric model is a voxel model built from ray-casting.

By ensuring that the voxel size is sufficiently large, a high quality voxmap without gaps is built up. A simple method for calculation of voxel size is presented. Morphological processing removes noise outliers and a distance transform that conservatively estimates the Euclidian distance is introduced.

4.1 Requirements placed on the object model

The requirements placed on the object model depend on the type of grasp to be planned. In this thesis precision grasps, which rely on accurate digit positioning and modelling of grasp of stability, are being planned. Power grasps, due to their increased area/number of hand-object contacts and reliance on hand compliance, place less strict requirements on the model — a simpler model will suffice, such as that used by [Bard *et al.* 95].

In precision grasp planning, an object model needs to supply information for the following tasks:

- Assessment of grasp stability, requiring:
  - estimation of centre of mass and moments of inertia of object.
  - estimation of force and moment due to digit contacts on the object surface.
- Planning of hand kinematics, requiring:
  - estimation of digit/surface contact angles.
  - collision checking between hand and object.

Each of these will now be examined in more detail.

4.1.1 Estimation of centre of mass

To estimate the centre of mass and moments of inertia of an arbitrarily shaped object, a volumetric model is required. No specific requirements are placed on the particular type of volumetric model that should be used, though the accuracy of the model will affect the accuracy of the centre of mass estimate.

4.1.2 Estimation of force and moment

The exact requirements for this task depend on the contact model used. In this thesis, fingers are modelled as point contacts with friction and therefore require position and normal information for surfaces. If fingers are modelled as soft finger contacts, second order curvature derivatives would be required in order to describe the shape of the object local to the contact.

4.1.3 Estimation of digit/object contact angles

This is required because different tasks require different digit/object contact properties. For example, fingertip precision grasps require contacts at only the tips of the digits, whereas lateral grasps require that the inside of the distal segment contacts the surface. This requires position and first and second order derivatives.

4.1.4 Collision checking

This simply requires spatial occupancy information, such as would be provided by any volumetric model. However, the collision checking should be fast enough to be implemented in real time. The choice of volumetric model is therefore related to the method of collision checking used.

4.1.5 Other requirements

The model should describe enough of the object to enable a successful grasp to be planned: it is not sufficient to build the model from one view only. Two or more views are required.

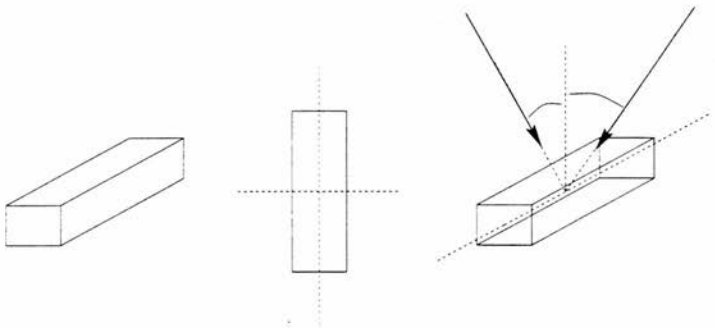
The model need not be as accurate as those required for the industrial inspection of parts. With soft fingers, and a suitable hand control algorithm, errors in fingertip placement can be allowed for.

4.2 Visual input

Since first and perhaps even second order curvature derivatives are required, dense range data is used as the input. A laser striper is used to image the object. A laser striper works by shining a stripe of laser light on the object and imaging the stripe using a CCD camera. The relative geometry of the laser stripe and the camera is known, making it posisble to calculate the 3D location of the object surface as it is illuminated by the stripe.

In a working system, the laser range finder would be mounted on the robot arm (such as the system used in [Whaite & Ferrie 93]). An intensity-based vision system could localise the object and the arm would then be directed so that two or more views of the object could be taken. The view directions should be chosen so that the object surfaces revealed are likely to lead to the identification of accessible and stable grasps. They could be determined from either the intensity image or an initial range image. Heuristics would be sufficient to determine two reasonable views from which to take further range images. For example, given a silhouette of the object derived from a 2D

intensity image, the principal axes of the moment of inertia can be found. The two views could then be generated by rotations of $\pm 45°$ about the axis of lesser moment. See Figure 4.1 for an illustration of this. Constraints are also placed on this process by the environment and the kinematics of the robot arm.



(a) (b) (c)

Figure 4.1: (a) the object (b) the silhouette of the object, looking from above. with the principal axes of moment of inertia shown (c) two viewpoints planned by rotation about the axis of lesser moment.

The accuracy of the resulting images and depends on the accuracy of the ranging system itself and on the accuracy of the robot arm placement. In a controlled environment, these could be highly accurate. However, if the arm was attached to a mobile robot in unknown lighting conditions, the errors in imaging and between different views of the object could be relatively high. Errors due to noise in one particular range image can be discarded by smoothing and surface fitting as long as there are not enough errors to remove all the possible grasping points, or extruding parts of the object which could prevent a grasp from being realised. More likely to be a problem are errors due to incorrect registration between two views of the object. These can be alleviated by:

- automatic registration between views using, for example, the ICP algorithm or a variant thereof [Eggert *et al.* 96].

- digit placement which minimises the error in wrench space resulting from the expected error in contact position.

One problem that need not be addressed is the merging of surfaces split between views. This is equivalent to the problem of oversegmentation, discussed in Section 4.3.4, and does not prevent grasps being planned. This is an example of how the type of model required for grasp planning is easier to build than the type of model required for industrial inspection or the construction of CAD models.

The laser striper used in the experiments is in fact fixed, so movement of the striper is simulated by rotation of the ground plane. The ground plane is translated along the x axis by a microstepper, and can rotate about the x axis. In order to register two views, the parameters required are the (y,z) position of the axis of rotation and the angle of rotation. The angle of rotation can be measured from the rotation apparatus to within 1 degree. Figure 4.2 shows a schematic diagram of the rotator. Two cameras are positioned on either side of the light stripe plane, looking down at the striper platform. Using two cameras instead of one makes it possible to image more of the object: with just one camera, some light stripes will be occluded by other parts of the object.

The position of the axis can be estimated from one or more range images of the rotator apparatus.

Alternatively, the axis of rotation can be estimated by fixing 3 spheres to the ground plane. The centres of the 3 spheres define a plane, and since the centres of the spheres can be accurately estimated by sphere fitting, the equation of the defined plane can be accurately estimated. If 3 images of the plane are taken at 3 different rotations, the position of the axis of rotation can be accurately estimated.

The x and y resolution of the laser striper is set by the user. The z axis is autoscaled to give 256 z values. In the experiments in this thesis, the x and y resolution is typically set to 1mm, and a typical z resolution is 0.5mm. However, the resolution of a surface description depends on the orientation of the surface with respect to the laser striper.

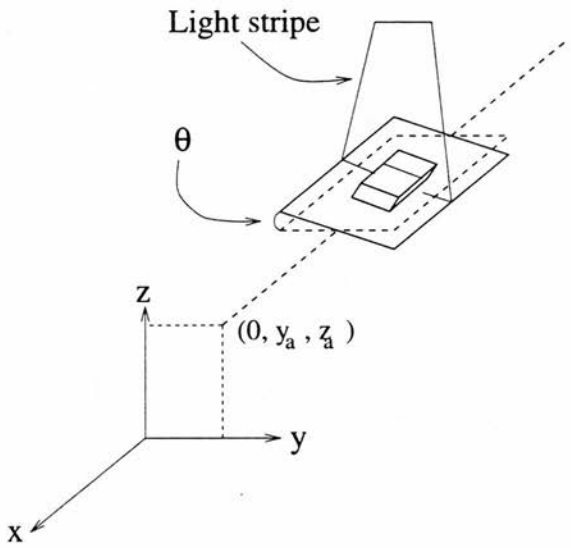


Figure 4.2: Schematic diagram of the rotator aparatus and laser striper. Two cameras are positioned on either side of the light stripe plane, looking down at the striper platform.

4.3 The Surface Model

A surface model of the object to be grasped is required for:

- Estimation of force and moment due to digit contacts on the object surface.
- Estimation of digit/surface contact angles.

The surface model is made up of a set of **graspable features**.

4.3.1 Graspable features

Graspable features are object features that provide grasping points. In reality, every part of the object constitutes a graspable feature. The exact type of graspable feature is important, however. This determines its contact geometry and consequently its wrench space characteristics. There are three types: vertices, edges and planes.

Vertex and edge contacts are only viable with soft fingers, when in fact they exhibit a highly desirable contact. Because the finger deforms around the edge or vertex, it can exert forces in a wide range of directions. The classification of a feature depends on the scale of the robot digit tip. A feature may be a surface for a thin digit, and an edge for a thick digit. Similarly, a rough surface could be classed as a set of edge features for a thin digit or one surface feature for a thick digit. See Figure 4.3. In (a) and (b) the feature is an edge with respect to the digit tip, wherease in (c) and (d) it has the contact properties of a surface. A method for scale-based feature extraction may therefore be required to model objects with rough surfaces, such as tree bark.

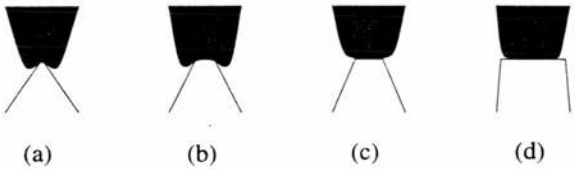


Figure 4.3: Relative scale of digit tip and contact feature affects feature classifications: (a) and (b) are edge features; (c) and (d) are surface features.

The graspable features should have common wrench space properties, which will mean that small errors in positioning within a graspable feature will lead to small deviations in wrench space (*i.e.* small errors in stability analysis).

In this thesis, planes and quadrics are used as graspable features. These are enough to describe a wide range of object shapes. Edges and vertices are not extracted. Edges and vertices are very good grasping features but small errors in positioning can lead to large errors in grasp stability. On a plane or quadric, however, small errors in positioning do lead to small errors in grasp stability: i.e. they map to small deviations in wrench space. This is particularly true for planes: an error in positioning for a plane leads only to an error in the torque applied to the object by that contact, while the force direction remains the same. Generally, the robustness of a particular grasp point with respect to positioning error depends on the curvature of the surface at the grasp point, and on the positions of the other grasp points.

While it is not clear that planes and quadrics provide the best primitives for describing

graspable surfaces, they certainly do provide a good estimation of first and second derivatives of position, which is the basic requirement for grasp planning.

4.3.2 Segmentation of a range image

In [Wren 92] [Wren & Fisher 93], each range image was segmented into planes and biquadratics. In this thesis, the range images are segmented into planes and quadrics (as opposed to biquadratics of [Wren 92] [Wren & Fisher 93]) using *Rangeseg* [Hoover *et al.* 96],[Trucco & Fisher 95]. [1] The segmentation works as follows:

1. Identify depth and orientation discontinuities.
2. Smooth surfaces while preserving discontinuities, using diffusion smoothing [Cai 88].
3. Segment according to HK curvatures [Besl & Jain 85].
4. Erode/dilate to remove small surfaces.
5. Grow segmented region into planes or quadrics [Fitzgibbon 92].

There are two important parameters when building a model for robot grasping:

- Morphology schedule. This described the schedule of erosions and dilations before the region growing. Strictly speaking, it should be set to zero in order to avoid removing the sole seed for a particular graspable feature, but this results in too many seeds. It is therefore set to be two erosions followed by two dilations. This tends to remove surplus seeds while leaving enough to yield all the graspable features.
- Threshold for region growing. This is the threshold which determines whether a point lies on an existing surface or not. It is typically set to 1mm.

Rangeseg generally works well. There are some problems in the region growing: each seed is grown in turn, and the resulting segmentation therefore depends on the order

[1] The size and shape classifications of [Wren 92] [Wren & Fisher 93] are no longer used, as they are much more suited to a rule-based grasp planner, whereas the work in this thesis is interested in finding grasps through optimisation of a grasp metric.

in which the seeds are considered. Also, it fails to segment, or oversegments, rough surfaces.

4.3.3 Post-processing

The output of RangeSeg is a set of planes and quadrics. This is then processed to find the boundaries of the surfaces, which means that the surfaces can just be described by their analytical equation, a set of boundaries and an included point. These surfaces are taken to be graspable features. Finally, these descriptions are rotated into a global coordinate frame and combined with features from other views to form the surface model.

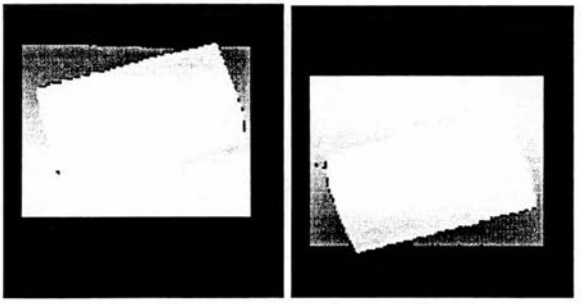
4.3.4 Oversegmentation

Oversegmentation is when graspable features are split into two or more parts. It does not affect the nuts and bolts of the algorithm. However, it will affect the choice of which wrist positions are considered, because the graspable features are used to cut down the search space. This will mean that the final grasping result will depend on the segmentation, and the speed it is reached in will depend on the segmentation. The more graspable features into which the object is split, the slower the algorithm will run.

4.3.5 Example Surface Models

Figure 4.4 shows range images of a cuboid taken from two different views. The cuboid is resting on a planar ground plane tilted at $\pm 35°$ about the x axis.

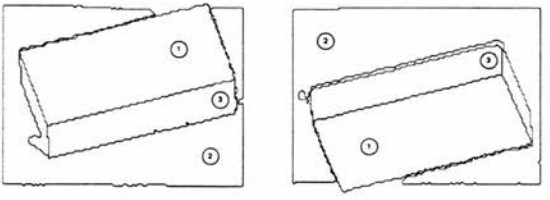
Figure 4.5 shows the segmentation of each viewpoint. Separate surface patches are numbered. In this scene, all patches are planar. Figure 4.6 shows the raw range data in global coordinates. The axis of rotation was measured as described in Section 4.2 and the angles of rotation about that axis are read off the rotation apparatus. The resulting global positions of surface points are only accurate to within about 1mm, but this suffices for the purposes of grasp planning. Note that some portions of the scene (in the ground plane and the top plane of the cuboid) are covered in both views.



(a) (b)

Figure 4.4: Range images of cuboid taken at (a) $-35°$, (b)$+35°$ about the x axis.

Figure 4.7 shows the surface model formed by registration of the two segmentations of Figure 4.5. The top plane of the cuboid is formed by two overlapping planes — one from each view. The surfaces are shown as meshes, sampled by a factor of 2.



(a) (b)

Figure 4.5: Segmentation of cuboid at (a) $-35°$, (b)$+35°$ about the x axis.

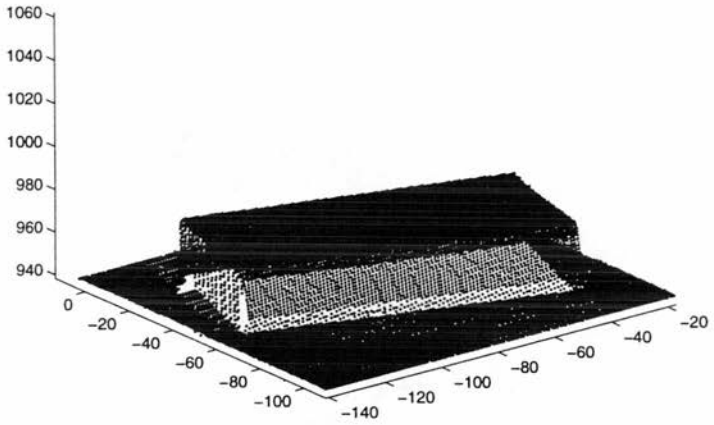


Figure 4.6: Registered raw range data from cuboid.

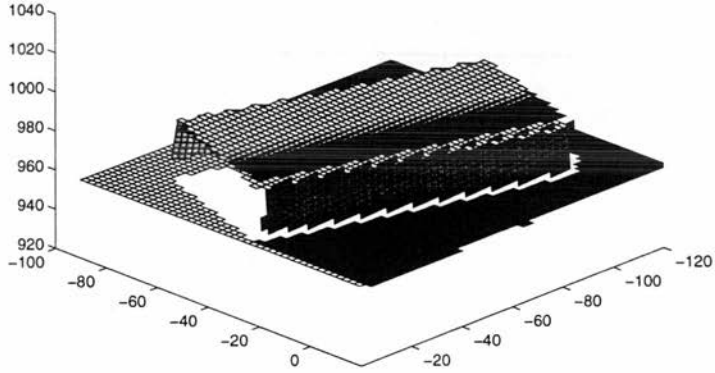
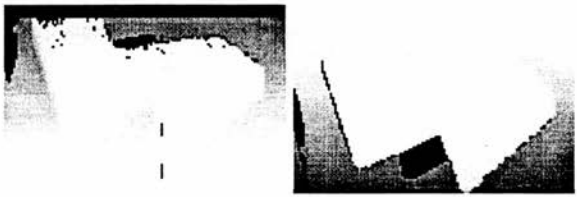


Figure 4.7: Surface model of cuboid

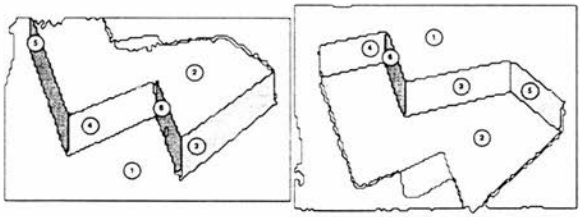
Figure 4.8 shows range images of a polyhedron taken from two different views. The polyhedron is resting on a planar ground plane tilted at $\pm 35°$ about the x axis.



(a) (b)

Figure 4.8: Range images of polyhedron taken at (a) $-35°$, (b)$+35°$ about the x axis.

Figure 4.9 shows the segmentation of each viewpoint. Separate surface patches are numbered. In this scene, all patches are planar. Figure 4.10 shows the raw range data in global coordinates. Note that again some portions of the scene (in the ground plane and the top plane of the polyhedron) are covered in both views. Figure 4.11 shows the surface model formed by registration of the two segmentations of Figure 4.9. The top plane of the polyhedron is formed by two overlapping planes — one from each view. The surfaces are shown as meshes, again sampled by a factor of 2.



(a) (b)

Figure 4.9: Segmentation of polyhedron at (a) $-35°$, (b)$+35°$ about the x axis.

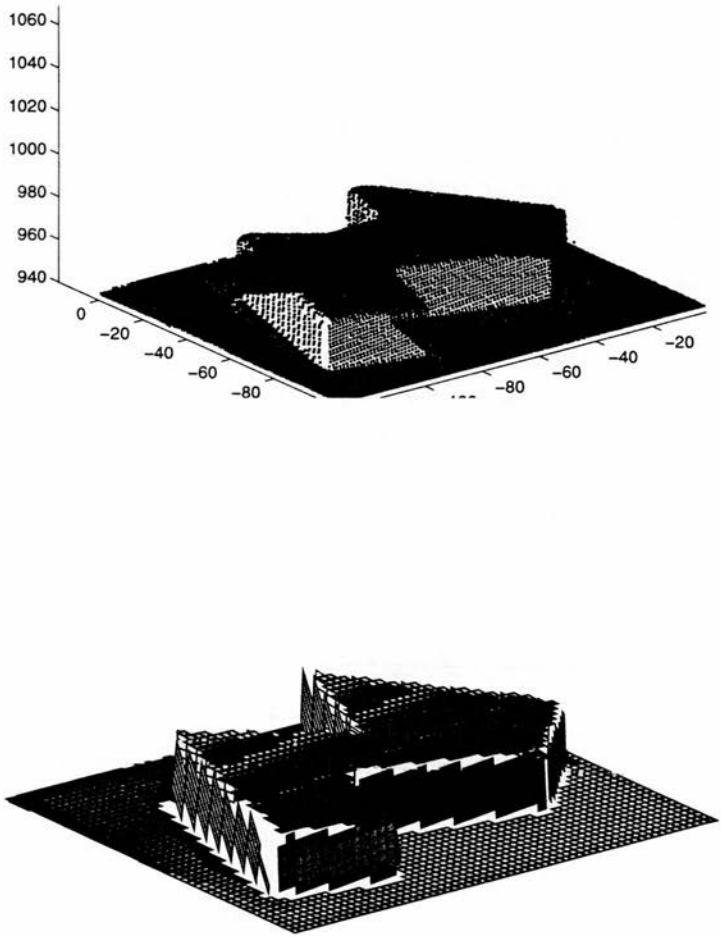
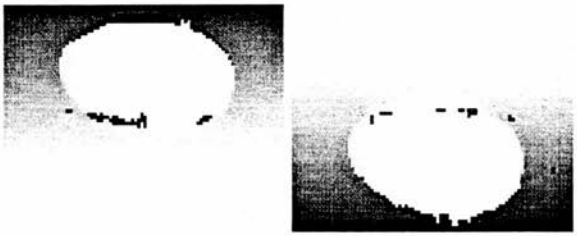


Figure 4.11: Surface model of polyhedron

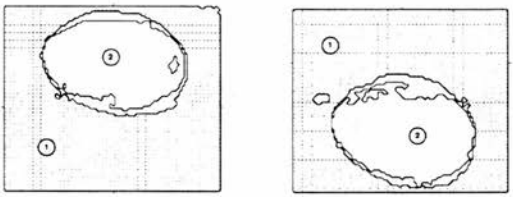
Figure 4.12 shows range images of a egg taken from two different views. The egg is resting on a planar ground plane tilted at $\pm 45°$ about the x axis.



(a) (b)

Figure 4.12: Range images of egg taken at (a) $-45°$, (b)$+45°$ about the x axis.

Figure 4.13 shows the segmentation of each viewpoint. Separate surface patches are numbered. In this scene, the two ground plane patches are planar, and the egg itself is comprised of two ellipsoids. Figure 4.14 shows the raw range data in global coordinates. Figure 4.15 shows the surface model formed by registration of the two segmentations of Figure 4.13.



(a) (b)

Figure 4.13: Segmentation of egg at (a) $-45°$, (b)$+45°$ about the x axis.

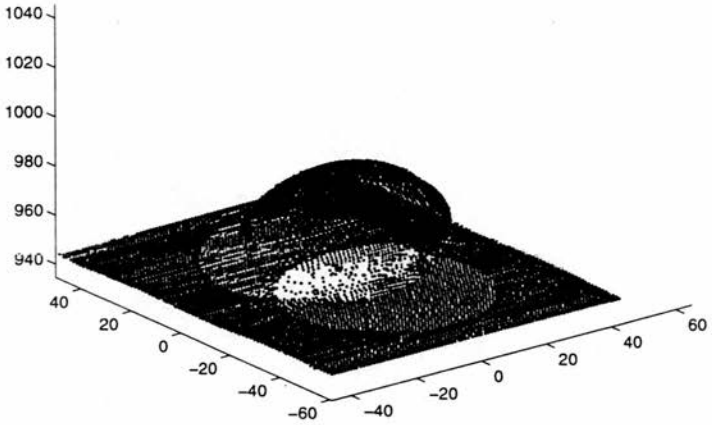


Figure 4.14: Registered raw range data from egg.

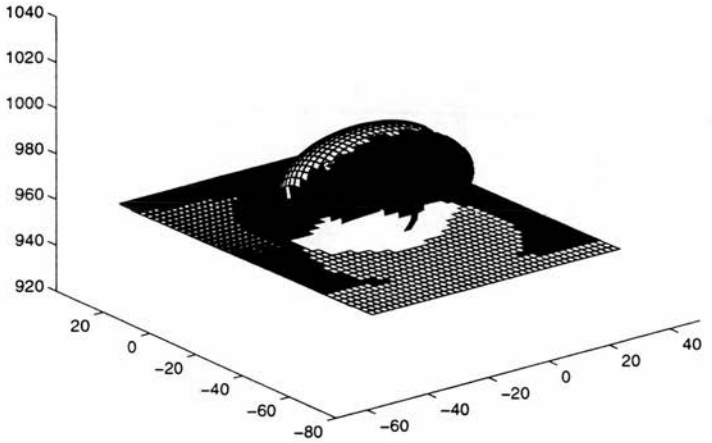


Figure 4.15: Surface model of egg

4.4 Volumetric Model

A volumetric model of the object to be grasped is required for

- Hand/object collision avoidance.
- Estimation of object centre of mass (and moments of inertia).

Efficient checking for hand/object collisions is essential when planning the kinematics of a grasp. When checking for such collisions, either a closed surface model of the object (*e.g.* the convex hull) or a volumetric representation is required [2]. The collision checking must be conservative, *i.e.* it must ensure that a collision is definitely detected. This means that if it is unknown whether some region of space is occupied or not, it should be assumed to be occupied.

An estimate of the centre of mass (which would assume, in the absence of other information, that the mass distribution of the object is uniform) is required for the static and dynamic stability analysis of a grasp. The requirements that this places on the volumetric model are less clear, in that there is no "safe" assumption to make about regions of space with unknown occupancy. In building the volumetric model, therefore, just the requirements imposed by the collision checking are considered.

The input to the volumetric model building is a set of laser striper range images, taken from one or more viewpoints. Typically, two sensibly chosen viewpoints are enough to get a reasonable volumetric (and surface) description of the object. The range images come in the form of a 2D xy grid, where each point on the grid may have an associated z value. The images are *dense* in the sense that almost every xy point will have an associated z value (occasionally points are rejected due to inconsistencies arising from sensor noise). Figure 4.4 shows an example of two range images of the same object from different views. The lighter the pixel, the closer the surface point. The object is a cuboid, taken from angles of rotation of $\pm 35°$ about the x axis. The black pixels correspond to background or unscanned points, and have no associated z value.

Each point in the range image data gives us two pieces of occupancy information:

[2] These are equivalent: a volumetric representation can be extracted from any closed surface model, and a piecewise closed surface can be extracted from any volumetric model.

- A point on the surface of the object, which marks a point in space which is occupied.
- A straight line, in the direction of the collimated laser light, leading from the laser source to that range image point, which marks a line of space which is empty.

See Figure 4.16 for an illustration of this. It shows the laser light ray that passes through a range point, which is on the surface of the object. The plane of the laser striper is parallel to the page; the dotted line represents the ray of collimated laser light that passes through a particular range point.

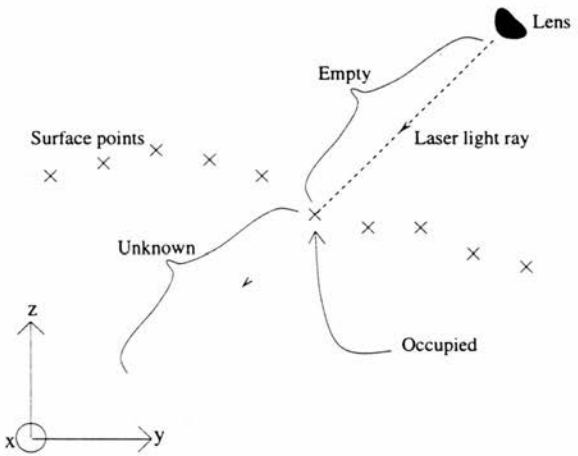


Figure 4.16: Occupancy information from a single laser striper range point. The plane of the laser striper is parallel to the page; the dotted line represents a ray of collimated laser light that passes through a particular range point.

This information is not volumetric — it deals with points and lines. In order to build the volumetric model a basic assumption has to be made about the region of space that can be said to be described by a single point. This must be small enough to give a sufficiently detailed model of the object, and large enough to avoid gaps in the occupancy data.

4.4.1 Voxel model

A voxel model of the object is formed. This is a 3D grid of identical cubes, each of which is referred to as a *voxel*. Each voxel can be either *empty* or *full*.

This model has the following advantages:

- It automatically associates a region of space with a given point in space. If any part of a voxel has its occupancy specified, the whole voxel shares that occupancy. If a voxel contains points of conflicting occupancies, the voxel is marked as occupied.
- It is very simple to implement collision checking on a voxel model.
- It allows for a direct interpretation of the occupancy information given above.

The voxel model is called the *voxmap*. It is possible to build a voxmap where each voxel contains a value between 0 and 1 corresponding to the probability of space being occupied (see [Bard *et al.* 93], who build a voxmap of occupancy probabilities from stereo depth data). This means that a threshold on the probabilities must be introduced, above which a voxel is marked as occupied. The laser striper range data used in this thesis is accurate and dense enough that such a method is unnecessary.

Occupancy is established in the voxmap by a combination of ray-casting and point-plotting.

4.4.2 Building the voxmap

The voxmap is built from one or more range images of the object and ground plane. Each range image v is taken from a different viewpoint; the striper is fixed, so the effect of moving the striper around the object is simulated by rotation of the ground plane on which the object is placed. The ground plane is rotated by a measured amount θ^v (accurate to within one degree) clockwise about an axis parallel to the world x axis, through the point $(0, y_a, z_a)$ (measured from the rotator apparatus). This means that an image point in view v (X_i^v, Y_i^v, Z_i^v) maps to the world point (x_i^v, y_i^v, z_i^v) where:

$$\begin{pmatrix} x_i^v \\ y_i^v \\ z_i^v \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta^v & -\sin\theta^v \\ 0 & \sin\theta^v & \cos\theta^v \end{pmatrix} \begin{pmatrix} X_i^v \\ Y_i^v - y_a \\ Z_i^v - z_a \end{pmatrix} + \begin{pmatrix} 0 \\ y_a \\ z_a \end{pmatrix} \tag{4.1}$$

The pixel separations within an image v, Δ_x^v and Δ_y^v, are set to be equal and the z separation, Δ_z^v, is auto-scaled. Typical values are Δ_x^v = 1mm, Δ_y^v = 1mm, Δ_z^v = 0.6mm.

The side of each voxel in the voxmap must be equal in order for standard distance transforms to be used. The voxel size, Δ_V, must be high enough to prevent any gaps in the original range images leading to gaps in the voxmap. Such gaps arise due to specular reflections or inconsistencies between the two cameras viewing the stripe. If G_{max} is the largest expected gap (in pixels) in a range image, Δ_V is determined by resolving the Δ_x^v,Δ_y^v and Δ_z^v of each range image v along the x, y and z axes of the voxmap respectively, choosing the maximum and multiplying this by $(1 + G_{max})$:

$$\Delta_V = (1 + G_{max}) \times \max_v \left\{ \max \left\{ \Delta_x^v, \Delta_y^v / \cos\theta^v, \Delta_z^v \cos\theta^v \right\} \right\} \tag{4.2}$$

For the voxmap to be sufficiently detailed for collision checking, it is requred that $\Delta_V < 2R$, where R is the fingertip radius. G_{max} is typically chosen to be 1. For two range images of xyz precision 1 × 1 × 1mm, taken at angles of ±45°, this gives $\Delta_V \simeq$ 3mm.

This aims to give maximum detail whilst avoiding, any gaps in the voxmap. To work, it relies on a correct estimate of G_{max}. In the case of G_{max} being too low, however, morphological processing provides a way of cleaning up the voxmap. If G_{max} is too high, the voxmap will be correct but of a coarser resolution than is necessary.

The voxmap can be constructed in two ways. The first method assumes strictly that any unknown voxel is *full*, and then marks the voxels which are unoccluded — *i.e.* through which a laser ray passes.

1. **Initialise voxmap:** Mark all voxels as *full* (*i.e.* unknown).

2. **Mark unoccluded voxels:** For each range point (x_i^v, y_i^v, z_i^v), cast a ray in

the direction $[-\sin\theta^v, -\cos\theta^v]^T$ up to the range point (x_i^v, y_i^v, z_i^v), marking each voxel intersected by the ray as *empty*.

3. **Mark full voxels:** Mark each voxel containing a range point (x_i^v, y_i^v, z_i^v) as *full*.

Full voxels are marked after *empty* voxels because, in the case of one voxel being marked as both *full* and *empty*, it should be marked as *full*, to ensure the validity of collision checking.

When using this method, errors can occur for two reasons:

- If the estimate of G_{max} is too small, it produces an overly conservative voxmap that consists of "spikes" coming out from the object or whole regions of erroneously occupied voxels. This will prevent collision checking from working properly.
- It relies on the striper to scan the *whole* of the space covered by the voxmap; failure to do so will result in free space being marked as *full*.

A different approach is therefore used, based on the assumption that if a voxel is not spanned by a ray, then it is *empty*. The ramifications of this are as follows:

- If the estimate of G_{max} is too small, this leads to gaps in the object. Whilst giving an inaccurate voxmap, this does not have a terminal affect on the collision checking.
- If the striper does not scan a region of space, then this assumes that it is *empty*.

Occluded regions of space are intersected to form the set of *full* voxels. [3] The algorithm is as follows:

1. **Initialise voxmap:** Mark all voxels as *empty*.

[3] This approach is also being used at Columbia University, though they work with planes rather than voxels.

2. **Mark occluded voxels:** For each range image j, cast a ray in the direction $[-\sin\theta^v, -\cos\theta^v]$ beyond each range point in the image, marking each voxel along the length of the ray as *full*. *i.e.* Mark as *full* each voxel intersected by the line:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x_i^v \\ y_i^v \\ z_i^v \end{pmatrix} + \lambda \begin{pmatrix} 0 \\ -\sin\theta^v \\ -\cos\theta^v \end{pmatrix}, \lambda > 0 \tag{4.3}$$

3. **Mark unoccluded voxels:** For each range image j, cast a ray in the direction $[-\sin\theta^v, -\cos\theta^v]$ up to each range point, marking each voxel along the length of the ray as *empty*. *i.e.* Mark as *empty* each voxel intersected by the line:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x_i^v \\ y_i^v \\ z_i^v \end{pmatrix} + \lambda \begin{pmatrix} 0 \\ -\sin\theta^v \\ -\cos\theta^v \end{pmatrix}, \lambda < 0 \tag{4.4}$$

4. **Mark full voxels:** Mark each voxel containing a range point (x_i, y_i, z_i) as *full*.

Figure 4.17 shows the process for the cuboid shown in Figure 4.4. A single slice, parallel to the plane of laser light, is displayed. Figure 4.17(a) shows the empty voxmap, (b) shows the occluded space for the $-35°$ image, (c) adds the occluded space for the $+35°$ image, (d) marks the unoccluded space for $-35°$ image, (e) marks the unoccluded space for the $+35°$ image, (f) adds the surface points from both views again, in case some voxels contain a surface point and an unoccluded point.

In Figure 4.17(f), note the gap in the top left corner of the cuboid. This arises due to an error in the registration of the two views. This is not serious from the point of view of collision checking or centre of mass determination. Much more serious is the stray point out to the left of the cuboid. This is a voxel containing a surface point which has arisen due to sensor noise, and will cause serious problems when collision checking.

Figure 4.18 shows a slice (from $x = -77.1301$ to $x = -74.6886$) of the voxmap for the cuboid at two different values of G_{max}. In Figure 4.18(a) $G_{max} = 0$, in (b) $G_{max} = 1$. Figure 4.19 shows a slice (from $x = -78.3509$ to $x = -74.6886$) for (a) $G_{max} = 0$, (b) $G_{max} = 2$.

4.4.3 Sources of error

There are three sources of error in the voxmap as it now stands:

- **Sensor noise.** This can produce rogue outlier surface points which incorrectly mark some voxels as *full*.
- **Gaps in the range images greater than G_{max} pixels.** Since a given pixel in a range image marks some voxels as *empty* and others as *full* this can lead to voxels being incorrectly marked as either *empty* or *full*.
- **Incorrect registration.** Not only does an error in registration lead to a corresponding error in the voxmap, it can lead to holes in the voxmap which should, given the correct registration, appear as *full*.

Voxels incorrectly marked as *empty* are not a problem under the assumption that the size of the gaps are less than the fingertip radius. In general, for the dense range images used here, this is true. In this case they contribute slight inaccuracies to the determination of centre of mass, and have no effect on the collision checking other than to distort the distance transform (see Subsection 4.4.5).

Voxels incorrectly marked as *full* are a much bigger problem. Again, they only contribute slight inaccuracies to the determination of the centre of mass but they can have a critical effect on the collision checking phase. Morphological operations are therefore required to remove such outliers.

4.4.4 Morphological processing

The principal aim of this is to remove all voxels incorrectly marked as *full*.

There are three stages to the morphological processing:

1. **Closing:** N cycles of dilation, followed by N cycles of erosion. This aims to fill in regions of the voxmap incorrectly marked as *empty* due to an under-estimate of G_{max}, and is necessary to ensure that step 2 does not erode parts of the object.

2. **Opening:** N cycles of erosion, followed by N cycles of dilation. This aims to remove voxels marked as *full* due to an under-estimate of G_{max}.

3. **Floodfill:** remove any voxel marked as *full* that is not connected to the bottom of the voxmap. This aims to remove noise, manifested as surface outliers in the range images, and is based on the assumption that all objects in the environment are connected to the ground plane — either by resting directly on top of it, or by resting on top of other objects.

N should be chosen such that $(N+1) \times \Delta_V < 2R$, where R is the fingertip radius, in order to preserve validity of subsequent collision checking. Typically, the estimate of G_{max} is accurate enough for $N = 1$ to suffice. In fact, this usually renders the floodfilling unnecessary; however, floodfilling is included for safety's sake; also, it is attractive due to being a parameter-free process based on sound physical assumptions.

When eroding, dilating and floodfilling in 3D, there is a choice of whether to use 6, 18 or 26-neighbour operators. Either the 18-neighbour or the 26-neighbour operator is used for erosion and dilation, and the 26-neighbour operator for floodfilling.

Figure 4.20 shows the morphological processing on a slice of the cuboid in Figure 4.4. Figure 4.20(a) shows the result of the ray-casting. In (b), a single 26-neighbour closing fills in the gap produced by incorrect registration. In (c) a single 26-neighbour opening removes the outlier range point from the left of the object. As a result, the floodfill stage (d) has no effect on the voxmap.

Figure 4.21 shows the resulting voxmap for the entire cuboid.

4.4.5 The Distance transform

"A distance transform [DT] converts a binary picture, consisting of feature and non-feature elements, to a picture where each element has a value that approximates the distance to the nearest feature element" [Borgefors 84]. A *constrained* distance transform works in the presence of obstacles — *i.e.* non-feature elements around which the distance must be measured.

A constrained distance transform inserts into every *empty* voxel a value approximately

corresponding to the distance to the nearest point on the object (*i.e.* the nearest *full* voxel). This is used for collision checking, so the estimate of distance should never be greater than the actual distance.

The 3D Euclidian distance transform ensures that the estimate of distance is exact, but cannot be constrained to work in the presence of obstacles. However, the simplest form of distance transform, the *n-neighbour* distance transform, can be used as a constrained distance transform if implemented as a series of dilations: the M^{th} dilation fills empty elements with the distance M. (The fast implementation of the n-neighbour DT uses raster scanning; however, it is not constrained by obstacles).

There is a choice of which dilation mask to use: 6-neighbour (*i.e.* dilation along coordinate axes), 18-neighbour or 26 neighbour (*i.e.* dilation in all directions). The 18-neighbour and 26-neighbour dilations will yield results less than or equal to the Euclidian distance, and the 6-neighbour dilation will yield results greater than or equal to the Euclidian distance. In each case, the DT produced is very far from the Euclidian DT. A dynamically determined sequence of 6, 18 and 26-neighbour dilations is therefore used. This keeps the distance as close to the Euclidean as possible, whilst ensuring that *the distance is never overestimated.*

The appropriate dilation operator is determined as follows. If N_6, N_{18}, N_{26} are the number of 6, 18 and 26 neighbour dilations respectively then:

$$\text{if} \quad \left\{ \begin{array}{l} \sqrt{3}(N_{26} + N_{18}/2 + (N_6+1)/3) \geq (N_6 + N_{18} + N_{26} - 1) \\ \text{and} \quad \sqrt{2}(N_{26} + N_{18} + (N_6+1)/2) \geq (N_6 + N_{18} + N_{26} - 1) \end{array} \right\} \quad \text{6-dilate}$$

$$\text{else if} \quad \left\{ \sqrt{3}(N_{26} + (N_{18}+1)/2 + N_6/3) \geq (N_6 + N_{18} + N_{26} - 1) \right\} \quad \text{18-dilate}$$

$$\text{else} \quad \text{26-dilate}$$

In order to see how this performs, a $101 \times 101 \times 101$ voxmap was formed, the central voxel was filled and the distance transform was calculated. For each voxel, the difference between the D_n, the distance from the dynamic n-neighbour DT, and D_e, the Euclidian distance was then calculated. For conservative collision checking, the DT distance must always be less than or equal to the Euclidian distance, *i.e.* $D_n - D_e \leq 0$. The distribution of $D_n - D_e$ is plotted as a histogram in Figure 4.22(a). Note that all values of $D_n - D_e$ are negative, demonstrating that the dynamic DT is suitable for collision

checking.

Figure 4.23 shows the distribution of $D_n - D_e$ for the dynamic, 6, 18 and 26-neighbour DTs respectively. It is clear that the distance errors for the dynamic DT are less than for the 6, 18 or 26-neighbour DTs (the spread along the horizontal axis is smaller).

Figure 4.24 shows the dynamic DT for the egg scene at various slices along the x axis. The brighter the pixel, the closer it is to a full voxel.

It is worth noting that the distance transform was implemented as a *constrained* distance transform, *i.e.* one that would "flow" around obstacle, so that it could also be used to find the distance away from specified contact points (single voxels) or surfaces on the object (connected groups of voxels). The specified region is tagged as a feature, and the rest of the object is classed as an obstacle. The constrained distance transform then fills out the voxmap with the distance away from the tagged voxel/region. The intention was that this can be used as a metric to measure how far a particular hand configuration is from the tagged region and be used to control the inverse kinematics of the hand. However, the grasp planning algorithm that was eventually used does not require this information.

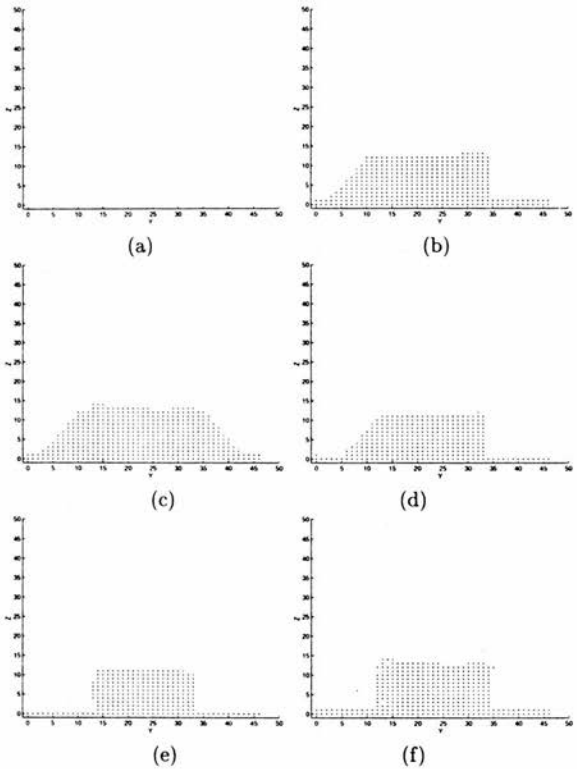
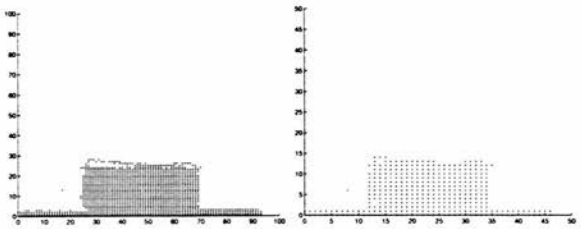
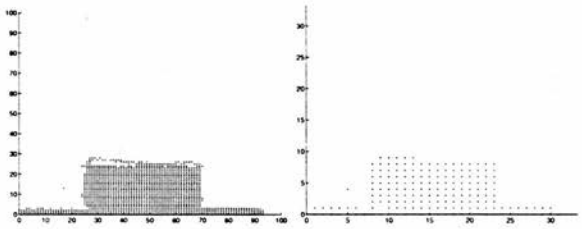


Figure 4.17: Example of ray-casting for the cuboid. The axis unit is a voxel, and a slice through the voxmap perpendicular to the x axis is shown. (a) shows the empty voxmap, (b) shows the occluded space for the $-35°$ image, (c) adds the occluded space for the $+35°$ image, (d) removes the unoccluded space for $-35°$ image, (e) removes the unoccluded space for the $+35°$ image, (f) adds the surface points from both views again in case some voxels contain a surface point and an unoccluded point.



(a) (b)

Figure 4.18: Slices of cuboid voxmap for (a) $G_{max} = 0$, (b) $G_{max} = 1$. The axis unit is a voxel, and a slice through the voxmap perpendicular to the x axis is shown.



(a) (b)

Figure 4.19: Slices of cuboid voxmap for (a) $G_{max} = 0$, (b) $G_{max} = 2$. The axis unit is a voxel, and a slice through the voxmap perpendicular to the x axis is shown.

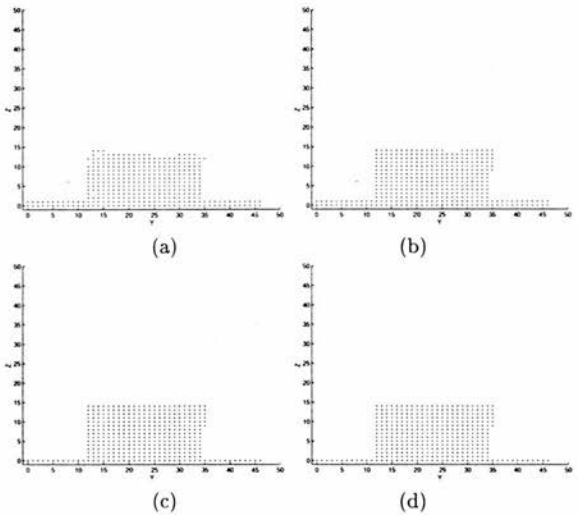


Figure 4.20: Example of morphology for the cuboid. The axis unit is a voxel, and a slice through the voxmap perpendicular to the x axis is shown. (a) shows the end result of the ray-casting, (b) shows the result of 1 cycle of 26-neighbour closing, followed by (c) 1 cycle of 26-neighbour opening followed by (d) a floodfill from the ground plane (no change).

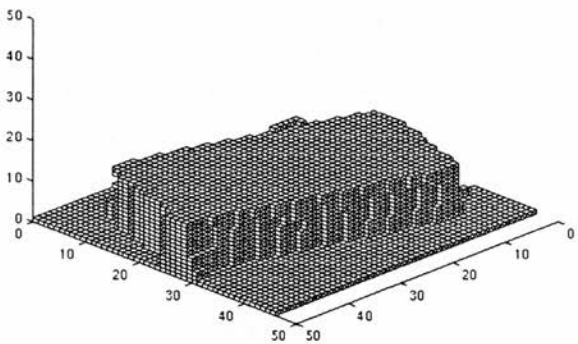


Figure 4.21: Final Result. The axis unit is a voxel.

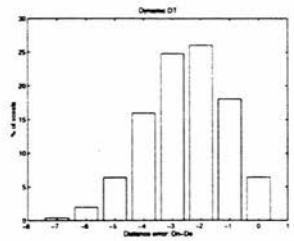


Figure 4.22: Distribution of errors in distance transform: $D_n - D_e$. Note that all errors are negative, *i.e.* the DT never over-estimates the distance and is therefore suitable for collision checking.

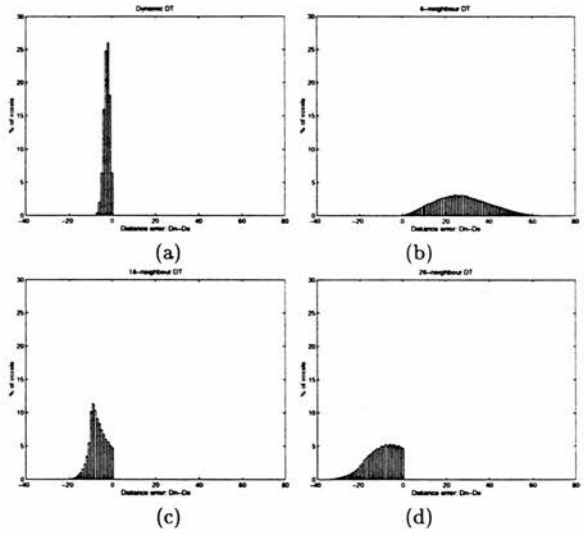


Figure 4.23: Distribution of errors in distance transform: $D_e - D_n$ for (a) dynamic algorithm, (b) 6-neighbour algorithm, (c) 18-neighbour algorithm, (d) 26-neighbour algorithm

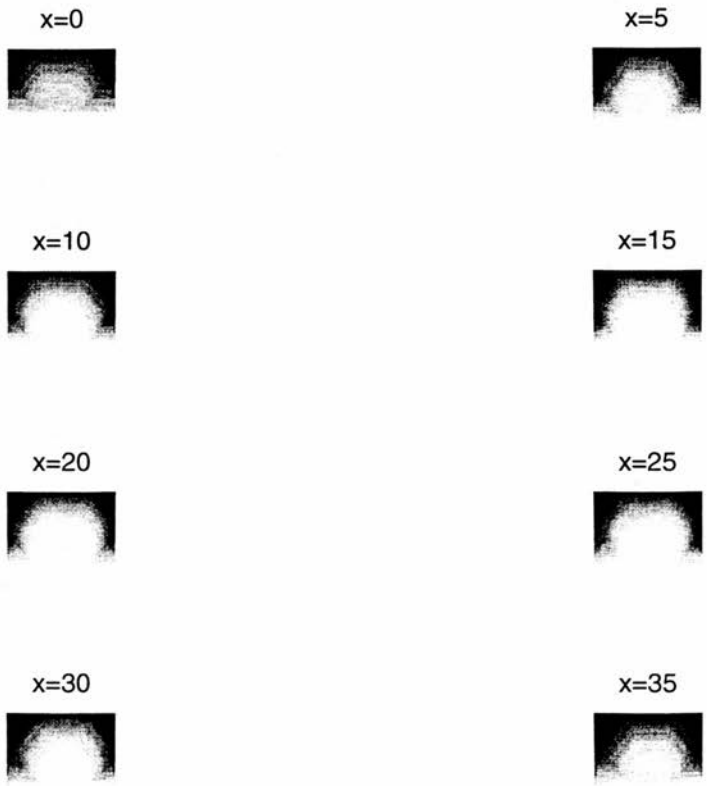


Figure 4.24: Dynamic DT for the egg scene at various slices along the x axis. The brighter the pixel, the closer it is to a full voxel.

4.4.6 Hierarchical decomposition

The facility to hierarchically decompose the voxmap into voxmaps of lower resolution (*i.e.* larger voxel size) is provided. As the voxmap moves to a lower resolution, each voxel takes on the minimum distance value from the set of voxels contained within it at the higher resolution, thus preserving the validity of the DT for collision checking. This is similar to an oct-tree representation, though without the recursive data structure.

In some instances this can speed up collision checking by describing large regions of free space by a single voxel. However, the object usually occupies a significant enough proportion of the voxmap to render this method no faster, or even slower, than collision checking without hierarchical decomposition. Generally, therefore, it is not used.

4.4.7 Speed

The algorithm has three main components:

1. **Ray-casting.** The speed of ray-casting depends on the number of points in the range images, the angle of rays cast and the size of the voxmap. The process is speeded up because all the ray casting is done in 2D: all rotations of the viewpoint are about the x axis, which means that all the rays lie in the yz plane. It is reasonable to confine all viewpoint rotations to be along either the x, y or z axis in order to speed up the ray-casting process.

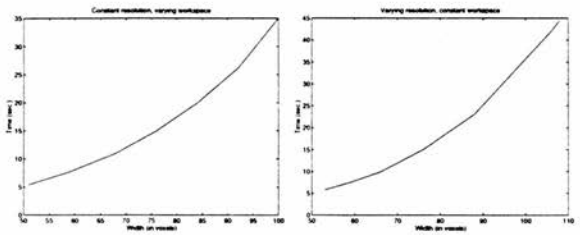
2. **Morphological processing.** If G_{max} is chosen conservatively, there need be only two cycles of erosion and two of dilation, followed by the floodfilling. The speed depends on the size of voxmap, and on the geometry of the object.

3. **The distance transform.** The time this takes depends on the size of voxmap and the geometry of the object.

If the algorithm runs too slowly, then the resolution should be lowered, *i.e.* Δ_V should be increased, provided that $(N+1) \times \Delta_V < 2R$, where R is the fingertip radius.

The algorithm is implemented in C++. To build a $51 \times 51 \times 51$ voxmap of the cuboid, from 22535 surface points, it takes between 5 and 6 seconds on an unloaded Sparc10.

Figure 4.25 shows the times taken to construct different sizes of voxmaps for the cuboid. In Figure 4.25(a) the resolution is kept constant but the size of workspace modelled is varied (the object is always kept at the centre of the bottom of the voxmap). In Figure 4.25(b) the same size of workspace is modelled at varying resolution. Figure 4.26 shows timings for different voxmap constructions of the cuboid in which the number of voxels is kept constant, but the resolution is varied (*i.e.* the size of object is varied with respect to the workspace). These results suggest that the primary determiner of the speed of voxmap construction is simply the number of voxels. The size of the object with respect to the workspace modelled is not very important.



(a) (b)

Figure 4.25: Speed of voxmap construction: (a) constant resolution, (b) constant size workspace.

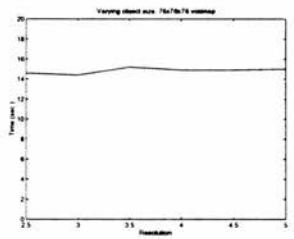


Figure 4.26: Speed of voxmap construction for 76 by 76 by 76 voxmap containing an object of varying size

4.4.8 Determination of viewpoints

The best set of viewpoints to use is determined partly by the formation of the voxmap and partly by the construction of the surface model. For the voxmap, the set of viewpoints should maximise the region of free space spanned by a laser stripe, such that the voxel size, as determined in Section 4.4.2 is sufficiently small for accurate collision checking. For the surface model, a sufficiently large number of (opposing) graspable surfaces should be viewed from an angle such that the range points are dense enough to give an accurate surface fit.

The constraints placed on viewpoint by the surface model are, in general, more severe than the requirements of the volumetric model. Indeed, for objects with vertical sides, a single overhead viewpoint gives a good voxmap description. Generally, however, this does not yield useful surface information for grasping, so views are typically taken at angles of rotation of $\pm 35°$ or $\pm 45°$ about the x axis. Again, for objects with vertical sides, given a perfect range image, this yields a perfect voxmap. For objects such as spheres and cylinders of horizontal axis, this yields a reasonably realistic voxmap. Generally, if a region of the voxmap is incorrectly filled, the subsequent collision checking will probably be unaffected. This is because a region of significant volume (as opposed to stray points or thin slices of the voxmap [4]) will be incorrectly filled if, and only if, no view was taken that covered that region of space. This means that no grasps will be planned in that region of space because there is no surface recorded that the fingertips can touch.

4.4.9 Collision checking

Collision checking between hand and object is done in physical space. With 6 or fewer degrees of freedom, configuration space collision checking is possible; the obstacles are mapped into the configuration space of the robot and the robot configuration is represented by a single point in configuration space (*e.g.* [Lozano-Perez 87]). However, with the high number of degrees of freedom of a dextrous hand (typically ≥ 15), such an approach is not feasible; collisions between robot and obstacle must be checked for

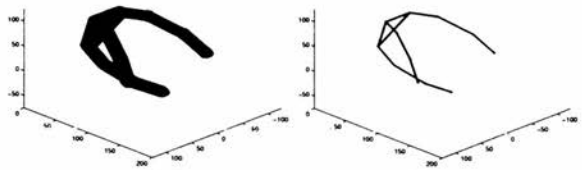
[4] Stray points may be incorrectly filled due to noise and thin slices of the voxmap may be incorrectly filled due to registration errors.

in physical space.

It is assumed that the robot links are cylindrical (or can be modelled by bounding cylinders). The fingertips are assumed to be spherical (or can be modelled by bounding spheres) and of the same radius as the cylindrical link. The palm of the robot hand is usually modelled as a set of cylindrical links between the bases of adjacent digits. Given these assumptions, the distance transform can be used for fast and simple collision checking.

Given a cylindrical link of the robot hand, the voxels intersected by the axis of that link are inspected. If any of the voxels contain a distance value of less than the cylinder radius then there is a collision. It is not therefore necessary to calculate the distance transform for voxels further away than the maximum cylinder radius of the hand. These can just be marked as "safe".

The distance transform has enabled us to use a simple stick model of the robot hand, and removed the need for calculation of surface intersections. Figure 4.27(a) shows a model of the Salisbury hand, with its links modelled as cylinders and its fingertips modelled as spheres. Figure 4.27(b) shows the corresponding stick model: each line corresponds to an axis of a cylinder.



(a) (b)

Figure 4.27: Salisbury hand: (a) cylinder/sphere model, (b) stick model.

Note that if all the cylindrical links of the hand are the same size, or can be modelled as being so, then there is no need to use the distance transform: the object is simply dilated by the radius of the cylinders (using the same sequence of 6,18 and 26-neighbour operators as in the distance transform) and check for collisions of the stick model

against the binary voxmap. This may be advisable for very large voxmaps. However, for accurate models of many hands, such as the Salisbury hand, the Utah/MIT hand and the human hand, links of variable radius may need to be modelled. Whether this justifies the need for the distance transform depends very much on the accuracy of the voxmap.

As noted above, hierarchical decomposition of the voxmap does not tend to speed up the collision checking. Indeed, it usually slows it down. A far better way to increase the speed of collision checking is to establish x, y and z bounds on the portion of the voxmap which contains voxels marked as *full*. Any link which lies a distance R outside of those bounds, where R is the link radius, does not need to be checked against the voxmap: it is definitely in free space.

4.4.10 Determination of centre of mass

In order to determine the centre of mass of the object, it must be segmented away from the ground plane. It is assumed that the ground is planar and horizontal. The ground plane can be determined by one of the following two methods:

- It can be assumed that the ground plane is much wider than the object. Considering horizontal slices starting from the bottom of the voxmap, the ground/object boundary is where the fraction of full voxels in a slice drops below a certain threshold (typically set to be 0.7).
- Find the lowest horizontal planar surface patch in the surface model. This is part of the ground plane. Take all horizontal planar surface patches lying within some threshold of this patch (typically 3mm) to be part of the ground plane also. Take the height of the ground plane to be the maximum height of its component horizontal planes.

The second method is used, since this enables the ground planes to be identified without building the voxmap.

4.4.11 Example voxmaps

Examples are shown for two more objects. For each one, the maximum expected gap in a range image is 1. Morphological processing consists of 1 cycle of dilation, followed by 2 of erosion, followed by 1 of dilation (equivalent to 1 cycle of closing followed by one cycle of opening). The voxmap is then floodfilled to remove any remaining outliers.

Figure 4.28 shows the voxmap for the egg. The voxmap tends to "fill out" the underside of the egg close to the ground plane because the laser striper did not image this part of it. Figure 4.29 shows the voxmap for the polyhedron. Note the gaps in the range images of the polyhedron in Figure 4.8 (due to portions of the ground plane being out of the view of the striper cameras): this does not affect the final voxmap, because each obscured region is seen in the other range image.

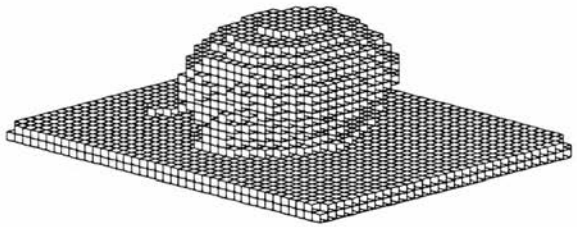


Figure 4.28: Voxmap of the egg.

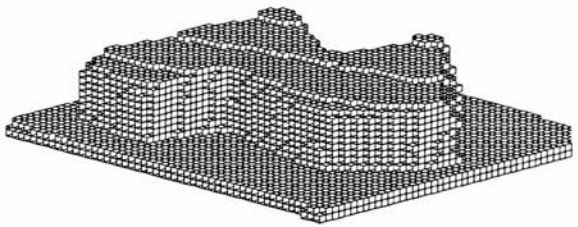


Figure 4.29: Voxmap of the polyhedron.

4.5 Summary

In this chapter a complete object model for *precision* grasp planning has been described. A surface model provides the contact information geometry necessary to analyse the mechanics of the grasp and to provide the contact angle information required to characterise tip and lateral grasps. A voxel model, called the voxmap, provides estimation of the centre of mass. A distance transformed voxmap provides very fast collision checking for a hand consisting of cylindrical segments of different radii. An approximate constrained Euclidean distance transform is formulated and used. As the distance transform is constrained, it is also useful for finding the distance from the robot hand to specified graspable features.

Chapter 5

The Grasp Planning Algorithm

In this chapter the grasp planner is described. The planner uses the object model from Chapter 4 and the grasp families of Chapter 3. The aim is to produce algorithms that can plan, on arbitrarily shaped objects, grasps that embody simple task requirements, such as whether a tip or lateral grasp is desired. A grasp metric is formulated and used by the grasp planner. Results are shown for a polyhedral object. The polyhedron is very nearly a generalised cylinder of polygonal cross-section. However, it is short along the axis of the cylinder, so cylinder preshaping algorithms would fail with this sort of object, and precision grasps are certainly required. This makes it ideal to illustrate the grasp planning algorithm. Further examples are shown in Chapter 6, for different types of objects (including curved objects) and task.

5.1 Two approaches to grasp planning

Using the grasp families described in Chapter 3, a hand with N digits has the following degrees of freedom (DOF) in determining fingertip position:

- 2 DOF for preshape (1 DOF for abduction, 1 DOF for flexion)
- N DOF for digit closures
- 3 DOF for wrist position
- 3 DOF for wrist orientation

As was shown in Chapter 3 the SAM, Salisbury and human hands can all be constrained to operate with these DOF whilst retaining the ability to grasp a wide range of object types. For a hand with three digits, such as the SAM or Salisbury hand, the space of possible grasps has 11 dimensions, of which only a subspace corresponds to kinematically feasible grasps (namely the subspace corresponding to fingertip-object contacts). Within this search space, the desired solution optimises the grasp metric, the formulation of which depends on the task. The grasp metric includes the basic constraint that the digit closure trajectories must intersect with graspable features to ensure that the grasp configuration lies in the subspace of kinematically feasible grasps.

Search methods in this space may be characterised as belonging to one of two categories. In the **forward kinematic** method the search takes place in the joint space of the hand (or rather in the configuration space of the grasp family). In the **inverse kinematic** method the search takes place by sliding the fingertips across their respective graspable surfaces.

5.1.1 Forward kinematic search

The forward kinematic search method is a direct descendant of those methods which use heuristics to form and fit preshapes to objects, such as [Bard *et al.* 95], [Pollard 90], [Stansfield 91]. These use heuristics to form and fit a preshape to an object and, as a result, [Pollard 90] and [Stansfield 91] can only cope with simple objects because it is too difficult to define heuristics that work for arbitrary curved objects. [Bard *et al.* 95] can cope with complex objects, due to a compact object representation, but is more suited to planning power grasps due to the lack of surface orientation information in the object model: contact (and therefore surface) orientation is necessary to plan fingertip grasps.

Given a grasp family, consisting of preshape and digit trajectory, the forward kinematic search method developed fits the preshape by minimisation of the grasp metric. The search takes place directly in configuration space of the preshape. This is an 8D space: there are 3 DOF for preshape position, 3 DOF for preshape orientation, 1 DOF for the abduction of the fingers in the preshape and 1 DOF for the flexion of the preshape (remembering that the flexion of all the digits in a preshape is described by

the same variable). The space of possible grasps is (8+N) dimensional: the N DOF that are not determined by the preshape correspond to the distance travelled along the digit trajectories, and are determined by intersection of the digit trajectories with the graspable surfaces. The search space is bounded by the requirement that the digit trajectories should each intersect a graspable feature.

This method has the advantage that there is no need for computation of the inverse kinematics at any point in the search. However, it is difficult to bound the search space sufficiently other than through the use of heuristics. However, it will be shown that it is possible to plan grasps on arbitrarily shaped objects.

5.1.2 Inverse kinematic search

In the inverse kinematic search method, the fingertips are constrained to lie on their respective graspable features. The DOF of the search is the DOF of the fingertip positions plus any remaining DOF due to the redundancy of the hand. There are 3N DOF for the fingertip positions, which are specified by 2N DOF due to the constraint that the fingertip positions lie on a surface. The redundancy of the hand accounts for the remaining (8-2N) DOF. An initial (poor quality) solution is assumed to be given — in practice this is calculated from heuristics and very coarse sampling of the hand configuration space. The subsequent search then proceeds by successive perturbation of the fingertip positions such that contact with the graspable surfaces is maintained. For a given move in fingertip position, the corresponding perturbation in joint configuration is calculated using local inverse kinematics (see Section 2.3.3).

This method has the advantage that every hand configuration considered satisifies the basic constraint that each fingertip contacts a graspable feature, and therefore avoids searching unnecessary parts of the configuration space. However, as will be seen, it is relatively simple to use heuristics to sensibly bound the search space in the forward kinematic search method. The local inverse kinematic search as presented has the principal disadvantages that it must still be initialised by some forward kinematic search method (albeit a coarse-resolution one). One method around this is to initialise the hand close to the object using heuristics, and then drag the fingertips to the contacts. The distance from fingertip to contact region can be calculated by the constrained

distance transform, as described in Chapter 4.

5.2 Grasp metric

In this section the metric according to which grasps are assessed is formulated. In order to produce the best possible grasp, the metric should be *minimised*. The metric consists of two parts: the **mechanical metric** and the **kinematic metric**. The mechanical metric assumes that all contact forces can be exerted with the same ease, and therefore just depends on the contact geometry and the relative contact positions. The kinematic metric takes into account the kinematics of the hand configuration, and is based on the ability of the robot to exert a required force at each contact, and on the task requirements, which may include desired specifications of joint angles or angles between contact surface and distal link. Each term of the metric can take values between 0 and 1, or ∞ (in which case the grasp is not acceptable). It is important to normalise each term in this way, so that the coefficients can be set more meaningfully.

5.2.1 Mechanical metric

It is assumed that any equilibrium grasp will be stable, given suitable fingertip geometry (*e.g.* soft fingertips, the advantages of which are outlined in [Brockett 85] and examples of which are given in [Shimoga & Goldenberg 92]) and a suitable control strategy (*e.g.* stiffness control, as described in [Mason & Salisbury 85]).

The mechanical metric has three terms:

- $\mathcal{M}_1^M$, the equilibrium metric.
- $\mathcal{M}_2^M$, the required friction metric.
- $\mathcal{M}_3^M$, the task wrench space metric.

Consider a set of N contacts. Contact i has position $\mathbf{r_i}$ (relative to an arbitrary origin, which may be the centre of mass) and inward normal $\hat{\mathbf{n}}_\mathbf{i}$. The force at contact i is given by $\mathbf{f_i} = f_i\hat{\mathbf{a}}_\mathbf{i}$ ($\hat{\mathbf{a}}_\mathbf{i}$ may or may not be equal to $\hat{\mathbf{n}}_\mathbf{i}$). The first mechanical metric

depends on whether the grasp is in equilibrium. Assuming that gravity is negligible, this reduces to whether the sum of all contact wrenches is zero:

$$\begin{aligned} \mathcal{M}_1^M &= 0 \\ &\text{if} \quad \sum_{i=1}^{N} \begin{pmatrix} f_i \hat{\mathbf{a}}_i \\ \mathbf{r}_i \times f_i \hat{\mathbf{a}}_i \end{pmatrix} = 0 \\ \mathcal{M}_1^M &= \infty \\ &\text{otherwise} \end{aligned} \tag{5.1}$$

Given three points that do not lie on the same line, *without any constraint on the angle between contact force and surface normal*, it is always possible to construct a set of equilibrium grasps using contact forces at those points. The above formulation assumes that the equilibrium forces can be calculated analytically in such a way that is either independent of, or minimises, the remaining terms of the grasp metric. The alternative would be to use a continuous equilibrium metric weighted to be the most important term of the grasp metric. The grasp metric would then have not only the joint positions as variables, but also have joint velocities (*i.e.* torques) as variables, which would make the search for a good grasp intractable. In the case of the SAM hand, the fingers are not allowed to exert forces by abduction, so for a particular hand configuration, the equilibrium forces are uniquely specified to within a scale factor. Appendix A describes how to construct such a grasp for the SAM hand.

The second mechanical metric constrains the angle between surface normal and contact force by minimising the amount of friction required for the grasp. This is a sensible approach to take when planning grasps for which no coefficient of friction is known (as is almost always the case), and this is the metric that [Blake 92] uses. The second mechanical metric is therefore:

$$\begin{aligned} \psi_i &= \cos^{-1}(\hat{\mathbf{n}}_i \cdot \hat{\mathbf{a}}_i) \\ \mathcal{M}_2^M &= \max_i(\psi_i) \\ &\text{if} \quad \max_i(\psi_i) < \frac{\pi}{4} \end{aligned} \tag{5.2}$$

$$\mathcal{M}_2^M = \infty$$

otherwise

The angle of friction must be less than $\pi/4$ because anything greater than that would be an extremely bad grasp.

The third mechanical metric is a contact wrench distribution metric, which determines how closely the wrenches that can be applied to the object match the **task wrench space**, which is typically a subspace of 6D wrench space. 6D wrench space has three axes of force and three axes of moment, and the relative scaling between the force and moment axes is task-dependent, depending on the relative importance of being able to exert force and moment. The task wrench space is typically a 6D hyperellipsoid centred at the origin.

In the absence of a task wrench space, as in this thesis, $\mathcal{M}_3^M = 0$.

5.2.2 Kinematic metric

The kinematic metric has four terms:

- $\mathcal{M}_1^K$, the joint-torque metric: the reciprocal of the norm of joint torque due to unit fingertip force.
- $\mathcal{M}_2^K$, the joint limit metric: the distance from desired joint configuration.
- $\mathcal{M}_3^K$, the contact angle metric: the distance from desired angle between contact surface and distal links.
- $\mathcal{M}_4^K$, the collision metric.

$\mathcal{M}_1^K$ is a measure of the ability of the hand to exert a given fingertip force and, if the force cannot be exerted, the error in the actual force applied. It is in fact a measure of the sensitivity of joint torques with respect to forces applied at the fingertips. Conventional measures of how close a manipulator is to a singularity, such as $\mid \boldsymbol{J}^T\boldsymbol{J} \mid$, disregard the direction that forces are to be exerted in. For example, an outstretched digit is in a singular configuration in that it cannot exert forces parallel to the length

of the digit. It can, however, exert forces perpendicular to the length of the digit which may be sufficient for the task in hand. The joint-torque metric takes the direction of fingertip force into account.

Assuming that the wrist of the robot hand is fixed, if a force $\boldsymbol{f}$ is applied at the fingertip of a digit, the resulting joint torques are given by:

$$\tau = J^T f \tag{5.3}$$

where J is the Jacobian for a single digit.

If the digit is not at a singularity, this equation also gives the values of the joint torques required to exert a particular fingertip force. The bigger τ, the more sensitive the digit with respect to application of the force, and the better the grasp. At a singularity, there will be some directions in which a force cannot be exerted. If the required fingertip force lies along one of these directions, then $\tau = 0$ and the force cannot be applied. If the required fingertip force is orthogonal to one of these directions, then τ is large and finite, and the force can be applied. Between these extremes the force can be applied with some degree of directional error, and τ is small but still finite. The smaller τ, the bigger the error in the exerted force. The joint torque per unit force for a single digit i is given by:

$$\tau_i = \mid \mathbf{J}_i^{\mathbf{T}} \hat{\mathbf{f}} \mid \tag{5.4}$$

Each digit is actuated independently during grasp execution and the total metric is therefore given by:

$$\begin{aligned} \mathcal{M}_1^K &= \frac{\tau_{JT}}{\min_i \tau_i} \\ &\quad \text{if} \quad \min_i(\tau_i) > \tau_{JT} \\ \mathcal{M}_1^K &= \infty \\ &\quad \text{otherwise} \end{aligned} \tag{5.5}$$

where τ_{JT} is a user-defined constant that determines the allowable error in fingertip force.

The maximum value is chosen in preference to the mean, because it only takes one digit to be near a singularity to cause problems. $\mathcal{M}_1^K$ is infinite if the error in applied force is too large. If there is no error in applied force then it is an indication of the sensitivity of the applied forces to the controlling joint torques.

$\mathcal{M}_2^K$ is a measure of how far from joint limits the grasp is. There are M joints in total. The angle of joint i is θ_i, and can range from θ_i^{min} to θ_i^{max} with central value θ_i^{mid}. The joints should be as close to their central values as possible:

$$\mathcal{M}_2^K = \frac{1}{\sum_{i=1}^{M} k_i} \sum_{i=1}^{M} k_i \frac{(\theta_i - \theta_i^{mid})^2}{((\theta_i^{max} - \theta_i^{min})/2)^2} \tag{5.6}$$

where $k_i \geq 0$. k_i is the weighting that determines the relative importance of keeping the i^{th} joint away from its limits. If it is known that a subsequent manipulation relies more on some joints than others, then k can be set accordingly. Otherwise, k_i can be set to 1 for all i.

$\mathcal{M}_3^K$ is a measure of how well the contact between surface and distal link suits the task requirements for TIP or LATERAL grasps. For a tip grasp, this means that the angle between distal link and surface is kept as close to $\frac{\pi}{4}$ as possible. For a lateral grasp, the distal link is kept as close as parallel to the surface as possible. The metric also prevents the distal link from crossing the tangent plane of the contact (which would mean that it had penetrated the object).

Mathematically, the metric is expressed in the following terms. For digit i, the contact surface normal is $\hat{\mathbf{n}}_\mathbf{i}$, the direction of the axis of the distal link is $\hat{\mathbf{l}}_\mathbf{i}$ (going from distal joint to fingertip), and there are N digits in total. The metric is then given by:

$$\begin{aligned}
\mathcal{M}_3^K(\text{tip}) &= (\frac{4}{N\pi})^2 \sum_{i=1}^{N} (\pi/4 - \cos^{-1}(\hat{\mathbf{n}}_\mathbf{i} \cdot \hat{\mathbf{l}}_\mathbf{i}))^2 \\
\mathcal{M}_3^K(\text{lateral}) &= (\frac{2}{N\pi})^2 \sum_{i=1}^{N} (\pi/2 - \cos^{-1}(\hat{\mathbf{n}}_\mathbf{i} \cdot \hat{\mathbf{l}}_\mathbf{i}))^2 \\
\mathcal{M}_3^K(\text{manipulation}) &= 0
\end{aligned} \tag{5.7}$$

$$\begin{aligned} & \text{if} \quad (\hat{\mathbf{n}}_i \cdot \hat{\mathbf{l}}_i) \geq \mathbf{0} \\ \mathcal{M}_3^K &= \infty \\ & \text{otherwise} \end{aligned}$$

Finally, $\mathcal{M}_4^K$ is a collision metric:

$$\begin{aligned} \mathcal{M}_4^K &= 0 \\ & \text{if there is no hand/object collision} \\ \mathcal{M}_4^K &= \infty \\ & \text{otherwise} \end{aligned} \tag{5.8}$$

5.2.3 Complete metric

The complete metric is given by:

$$\mathcal{M} = \frac{1}{\sum_{i=1}^{3} \lambda_i^M} \sum_{i=1}^{3} \lambda_i^M \mathcal{M}_i^M + \frac{1}{\sum_{i=1}^{4} \lambda_i^K} \sum_{i=1}^{4} \lambda_i^K \mathcal{M}_i^K \tag{5.9}$$

where $\lambda_i \geq 0$. $\mathcal{M}$ should be *minimised*. In this thesis, values of λ_i are not set, as this is very dependent upon task requirements. As with all previous metrics, the final grasp metric is normalised to lie between 0 and 1.

To summarise, the grasp metric depends principally on the following factors:

- Kinematic configuration of the hand.
- Positions of the fingertips.
- Directions of applied forces.
- Collisions with the object.

5.3 Relationship of grasp families to the grasp metric

Now that the grasp metric has been designed, the grasp families designed in Section 3.5 can be re-evaluated in terms of the grasp metric. As an example, the lateral grasp family for the SAM will be considered.

5.3.1 Construction of an ideal lateral grasp for the SAM

The order of importance of the various terms of the metric depends on the task.The collision metric and equilibrium metrics must always be satisfied, however.

For the execution of a lateral grasp with the SAM, each term of the metric will be optimised in the following order of priority:

1. $\mathcal{M}_1^M$, the equilibrium metric.
2. $\mathcal{M}_2^M$, the required friction metric.
3. $\mathcal{M}_3^K$, the contact angle metric.
4. $\mathcal{M}_1^K$, the joint torque metric.
5. $\mathcal{M}_2^K$, the joint configuration metric.

$\mathcal{M}_3^M$, the task wrench space metric, is discarded in the absence of any more specific task information, and it is assumed that there are no collisions, *i.e.* $\mathcal{M}_4^K = 0$.

This ordering is equivalent to setting the coefficients of λ_i as follows. The coefficients of the metric terms which can be either zero or infinite, namely the equilibrium and collision metrics, are set to 1:

$$\lambda_1^M = \lambda_4^K = 1$$

The coefficient of the task wrench space term is zero:

$$\lambda_3^M = 0$$

All other coefficients are non-zero:

$$\lambda_2^M > 0, \lambda_1^K > 0, \lambda_2^K > 0, \lambda_3^K > 0$$

and ordered thus:

$$\lambda_2^M \gg \lambda_3^K \gg \lambda_1^K \gg \lambda_2^K$$

(required friction) (contact angle) (joint torque) (joint limit)

"$\gg$" is used rather than ">" to indicate that in the following derivation, the metric can be optimised with respect to one metric at a time, in an order according to the size of λ_i.

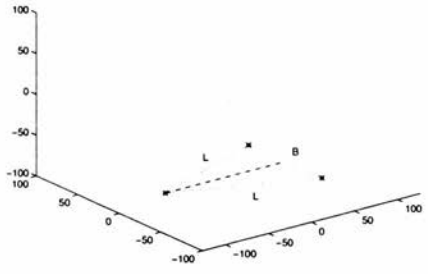


Figure 5.1: Contact configuration.

For the SAM hand, a grasp will be synthesised for one particular type of contact configuration: the contact positions lie in an isosceles triangle, with each finger contact at the base of the triangle. Figure 5.1 shows one such contact configuration. The contact normals are not specified at this stage; they will be fixed during the derivation of the optimal hand configuration. No unique hand configuration fits the contact set shown in Figure 5.1 and so the optimal hand configuration must be found. One assumption is made:

- The hand configuration is symmetric about the line joining the thumb knuckle with the midpoint between the knuckles of the fingers (the dashed line in Figure 5.1).

The equilibrium metric, the contact angle metric and the required friction metric combine to give the following constraints:

- **Equilibrium metric** The contact forces should meet at a common point in the plane of the contacts.
- **Required friction metric:** The contact normals are in the same direction as the contact forces.
- **Contact angle metric:** The distal link of each digit should be perpendicular to its contact normal (because lateral grasps are being planned).

This means that the distal link of each finger is perpendicular to the contact force.

Figure 5.2 shows two sets of contact normals/forces that satisfy these criteria; the distal link of the digit at each contact should be perpendicular to the contact normal/force. The criteria define a subspace of hand configurations, each of which optimises the equilibrium, required friction and contact angle metrics.[1] The next step is to optimise the joint torque metric within this subspace: this selects a unique hand configuration *and* a unique set of force directions.

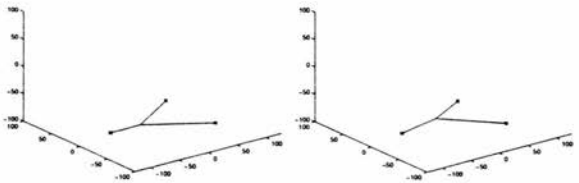


Figure 5.2: Two possible sets of contact normals/forces that satisfy the equilibrium criteria and required friction metrics. The distal link of each digit should be perpendicular to each contact normal/force.

Before considering the optimisation of the joint torque metric for the whole hand, the joint torque metric for a single digit is analysed, because the joint torque metric of the

[1] The order of priority among these three terms of the metric is therefore arbitrary.

hand is equal to the joint torque metric of the worst digit. A single digit is shown in Figure 5.3: the joint angles are given by $\alpha, \beta, \gamma, \delta$ with $\gamma = \frac{2}{3}\beta$.

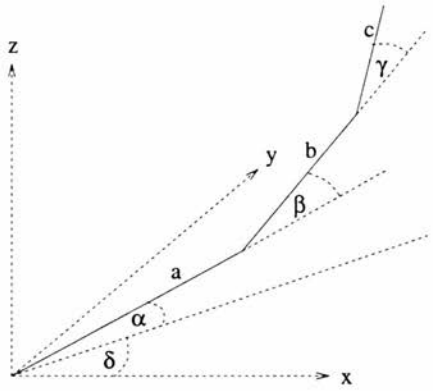


Figure 5.3: Single SAM digit.

If the Jacobian for the digit is $\mathbf{J}$, the relationship between joint torques $\boldsymbol{\tau}$ and fingertip force $\boldsymbol{f}$ is given by:

$$\begin{pmatrix} \tau_\alpha \\ \tau_\beta \\ \tau_\delta \end{pmatrix} = \mathbf{J^T} \begin{pmatrix} f_x \\ f_y \\ f_z \end{pmatrix} \tag{5.10}$$

and

$$\mathbf{J} = \begin{pmatrix} -(S_1 + S_2 + S_3)\cos\delta & -(S_2 + \frac{5}{3}S_3)\cos\delta & -(C_1 + C_2 + C_3)\sin\delta \\ -(S_1 + S_2 + S_3)\sin\delta & -(S_2 + \frac{5}{3}S_3)\sin\delta & (C_1 + C_2 + C_3)\cos\delta \\ (C_1 + C_2 + C_3) & (C_2 + \frac{5}{3}C_3) & 0 \end{pmatrix} \tag{5.11}$$

where

$$S_1 = \sin\alpha, S_2 = \sin(\alpha + \beta), S_3 = \sin(\alpha + \frac{5}{3}\beta)$$

$$C_1 = \cos\alpha, C_2 = \cos(\alpha + \beta), C_3 = \cos(\alpha + \frac{5}{3}\beta)$$

As noted above, since the grasp is a lateral grasp, the fingertip force must be perpendicular to the distal link. The local configuration of the digit can be varied by variation

of β. Without loss of generality, the distal link is fixed to be parallel to the z axis and the fingertip force therefore acts in the xy plane, at an angle ψ to the plane of the digit.

The space of all digit configurations was sampled in β at 2 ° intervals in the range $0 \leq \beta \leq 54°$. The range of possible fingertip forces are then applied, by variation of ψ in the range $-90° \leq \psi \leq 90°$ at 2° intervals, and the force direction ψ that minimises $\mathcal{M}_1^K$ is chosen. It turns out that if β is fixed, the force direction that minimises $\mathcal{M}_1^K$ for a single digit is given by $\psi = 0$, *i.e.* the force that lies in the plane of the digit. This means that, for a given digit configuration, it is always best to apply lateral grasp fingertip forces by means of flexion, not abduction. Furthermore, if the force direction ψ is kept fixed at 0, the joint torque metric decreases (*i.e.* improves) as β decreases, *i.e.* as the whole digit straightens out, the quality increases. Figure 5.4 shows the reciprocal of the joint torque metric plotted against β and ψ. The reciprocal is plotted because the metric can become very large, and means that the higher the value shown, the better the digit/force configuration. A ridge can be seen at $\psi = 0$, where the force is in the plane of the digit, and the height of this ridge increases as β decreases.

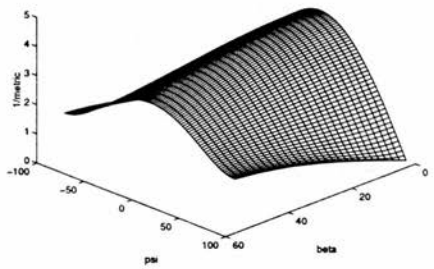
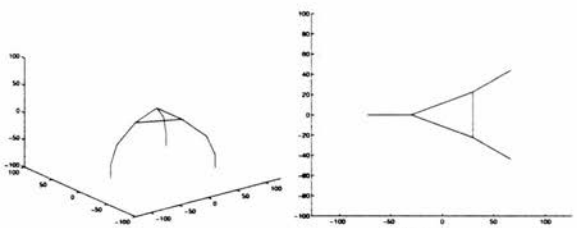


Figure 5.4: The reciprocal of the joint torque metric plotted with against β and ψ. The reciprocal is plotted because the metric can become very large,and means that the higher the value shown, the better the digit/force configuration. A ridge can be seen at $\theta = 0$, where the force is in the plane of the digit, and the height of this ridge increases as β decreases.

For the grasp under consideration, the force direction is therefore chosen to lie in the

plane of the digit. This is also desirable from a control point of view, as it means that no abduction of the digits is required to exert grasp forces (indeed, for the SAM hand, the thumb is not capable of abduction). It has been assumed that the hand configuration is symmetric, so all three fingertip forces will therefore intersect and by correct choice of force magnitude (which is independent of the grasp metric), the grasp will be in equilibrium. The task is to therefore minimise the largest value of β for any digit.



(a) (b)

Figure 5.5: The hand configuration being proposed as optimal, shown in two views.

Figure 5.5(a) shows one particular hand configuration that is being proposed as optimal, with $\beta_0 = \beta_1 = \beta_2$ (where β_0 corresponds to the thumb and β_1, β_2 to the fingers). Figure 5.5(b) shows a top view of that hand. Consider one digit in isolation: if β_i is decreased in order to decrease (*i.e.* improve) the joint torque metric for that digit, the projected length in the view shown is reduced because the digit is "straightening" out. If β_i is increased, the joint torque metric increases (*i.e.* worsens), and the projected length increases.

Consider if, in order to decrease the joint-torque metric, β_0 is decreased. This means that the projected length of the thumb in the top view is decreased. The projected length and width of the palm in the top view is already at a maximum, and therefore it is clear that the projected length of the fingers in the top view must be increased to compensate for the decrease in the projected length of the thumb. Reducing the joint torque metric for the thumb therefore increases the joint torque metric for the fingers. This means that there will be a *decrease* in the quality of the joint torque metric because it is only as good as the worst digit. Similiarly, decreasing β_1 or β_2 leads to

an increase in β_0, *i.e.* decreasing the joint torque metric for the fingers increases it for the thumb. This means that the hand configuration shown is the locally optimal (symmetric) hand configuration.

The joint torque metric has been optimised with respect to β. Each force direction is chosen to lie in the plane of its respective digit in order to optimise the joint torque metric with respect to ψ. Because the hand configuration is symmetric, these force directions satisfy the equilibrium requirement. The required friction metric is therefore optimised by selecting contact normals which lie in the same direction as these forces, as shown in Figure 5.6.

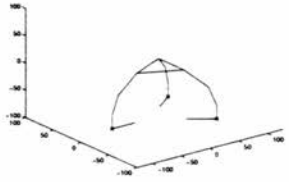


Figure 5.6: The optimal (symmetric) hand configuration for the contact configuration shown in Figure 5.1. Two different views are shown. Contact positions are marked with an asterisk, and directions of contact normals are shown. The fingertip forces are exerted in the same direction as the contact normals.

In the SAM example, the hand configuration is now fixed and the grasp cannot be optimised further: the joint configuration metric can therefore not be optimised for the ideal grasp. However, it is retained because in real applications it will be required as a "tie-breaker" between grasps with all other metrics equal (or nearly equal).

5.4 Forward Kinematic Grasp Planning

In an uncluttered environment, using simple heuristics, grasps may be planned easily on certain object types, such as cuboids, cylinders or spheres (as demonstrated by [Pollard 90], [Stansfield 91]). In a cluttered environment which places constraints on the wrist position, or with objects that do not fit into any of the aforementioned classes, such methods are not sufficient to capture the range of possible contact combinations.

In this section a method is presented that can plan grasps on arbitrary objects, in accordance with the grasp metric, $\mathcal{M}$.

Figure 5.7 shows a flowchart of the grasp planner. It has all been implemented with the exception of the optimisation by local deformations of the hand kinematics. The centre of mass is not used in the grasp metric described in Section 5.2 but might be required for certain tasks, and is extracted as described in Chapter 4.

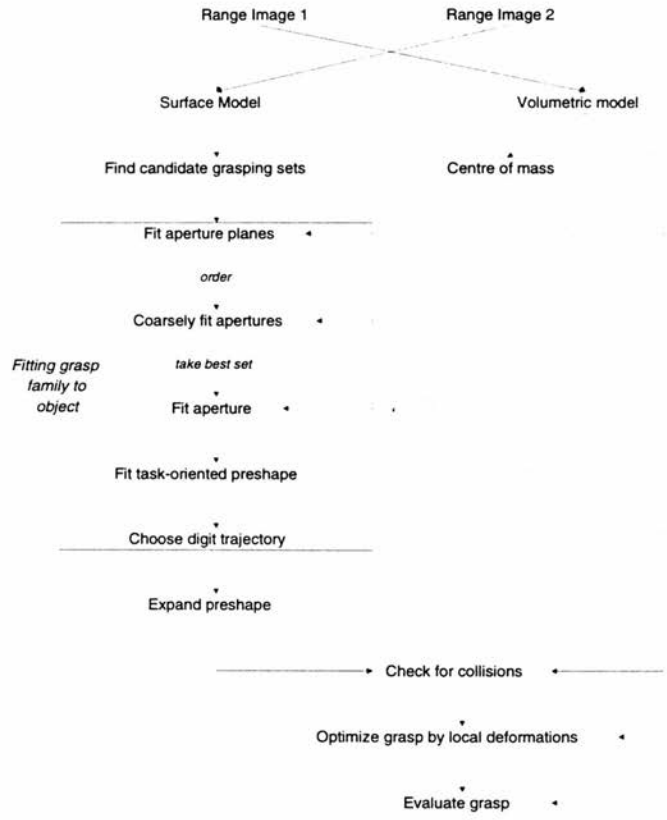


Figure 5.7: Heuristic Grasp Planner

(a) (b)

Figure 5.8: Object model of polyhedron: (a) surface model, with patches shown by their boundaries (all patches are planar), (b) volumetric voxel model.

The grasp planner is now described using the polyhedron object as an example. Figure 4.8 shows the input to the algorithm: two range images taken from $\pm 35°$ to the horizontal.

5.4.1 Build object model

The surface and volumetric models are constructed as described in Chapter 4. The surface model is used to drive the grasp planner; the volumetric model may influence the whole process of grasp planning, if the grasp metric involves the centre of mass of the object (in this thesis, it does not), or may just be used at the end of the process to prevent collisions of the grasp with the object. The nature of the grasp family kinematics used here is such that there will not be collisions between hand and object for most grasps of most objects. Leaving the collision adjustment to the end of the process is therefore appropriate and simplifies the process considerably. Figure 5.8 shows the object model for the polyhedral object (see Chapter 4 for more details).

5.4.2 Find Candidate grasping sets

A **candidate grasp set** is a set of surface patches which are possible contacts for the robot hand. Within each candidate grasp set, one surface patch will be nominated as containing the thumb contact. The remaining patches will contain the finger contacts — allocation of specific fingers to specific patches at this stage is not necessary. Each possible set of surface patches is considered in turn; it becomes a candidate grasping set if the thumb patch **opposes** all the finger patches. Two patches are said to oppose

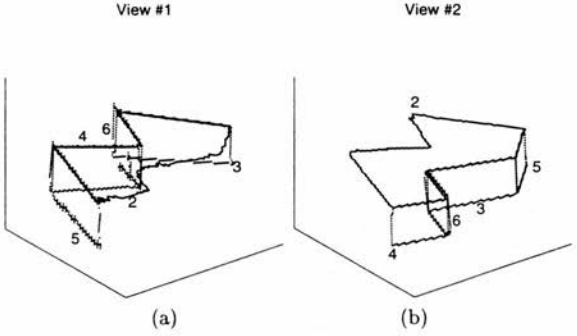


Figure 5.9: The surfaces in each view of the polyhedron in global coordinates: (a) shows the first view and (b) the second view. The patches within each view are labelled by a number. Patch #1 is missing in each view: this is the ground plane and has been excluded for ease of presentation.

one another if the angle between any normal on one patch and any normal on the other patch is greater than the **angle of opposition**, θ_{opp}. This is a good heuristic measure that throws out grasping sets that are very unlikely to be good. Note that it can still admit very poor candidate grasping sets because it does not measure the relative position of the patches.

This condition favours the thumb by requiring opposition between the thumb and each finger, but not between the fingers themselves. Because of the geometry of the hands considered in this thesis, in order to apply opposing *finger* forces (as distinct from opposing *finger/thumb* forces), a large degree of actuation at the joint of abduction must be applied. As seen in Section 5.2.2, during the derivation of the kinematic metric, this generally lowers the quality of the joint torque metric $\mathcal{M}_1^K$, and is also undesirable from the point of view of control during grasp execution.

In all the experiments shown in this thesis, the angle of opposition is set to be 120°.

Figure 5.9 shows the surfaces in each view of the polyhedron example in global coordinates: (a) shows the first view and (b) the second view. The patches within each view are labelled by a number. Patch #1 is missing in each view: this is the ground plane and has been excluded for ease of presentation. Figures 5.10– 5.13 show the candidate

grasping sets for the polyhedral object selected according to the angle of opposition rule. There are 30 candidate grasping sets for this object: 12 three-patch sets and 18 two-patch sets. Above the picture of each candidate grasping set are the identification numbers of the patches that comprise the candidate grasping set. Each bracketed pair of numbers is a patch identifier: the first for the thumb patch, followed by the finger patch(es). In each patch identifier, the first number is the viewpoint (1 or 2), and the second number is the patch number in that view, as shown in Figure 5.9.

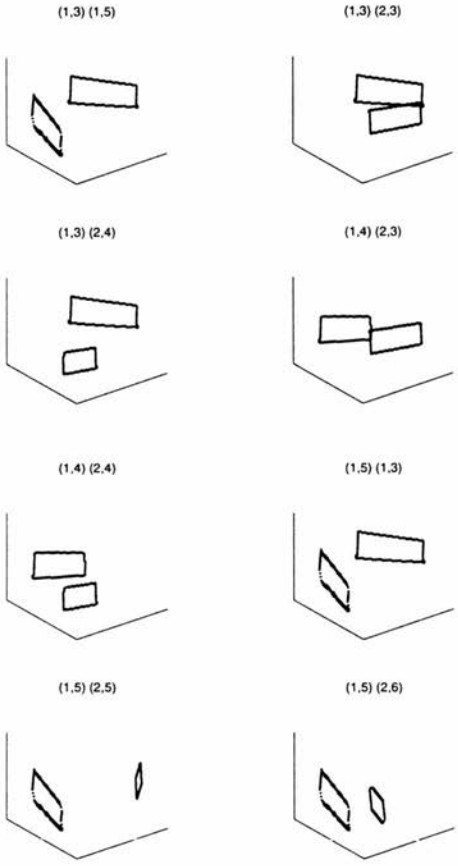


Figure 5.10: Candidate grasping sets for the polyhedron (1 of 4). Above the picture of each candidate grasping set are the identification numbers of the patches that comprise the candidate grasping set. Each bracketed pair of numbers is a patch identifier: the first for the thumb patch, followed by the finger patch(es). In each patch identifier, the first number is the viewpoint (1 or 2), and the second number is the patch number in that view, as shown in Figure 5.9.

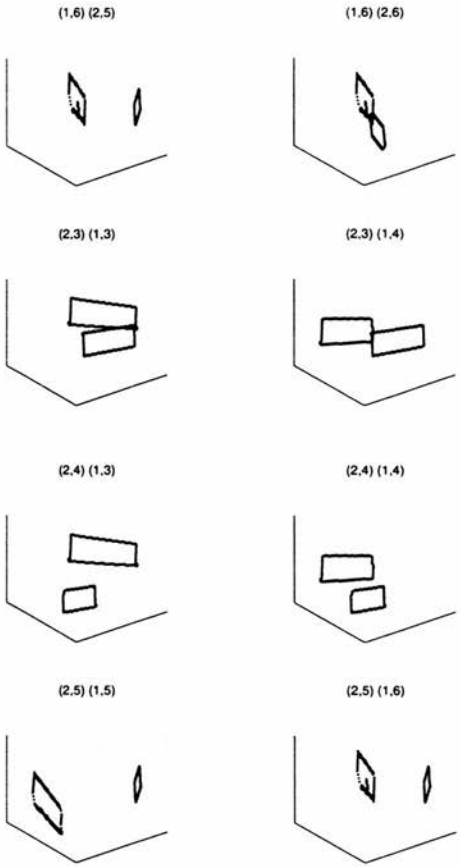


Figure 5.11: Candidate grasping sets for the polyhedron (2 of 4).

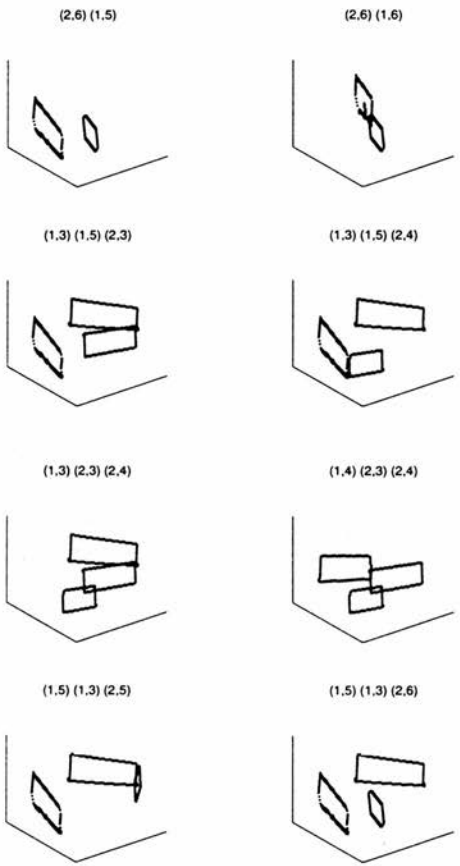


Figure 5.12: Candidate grasping sets for the polyhedron (3 of 4).

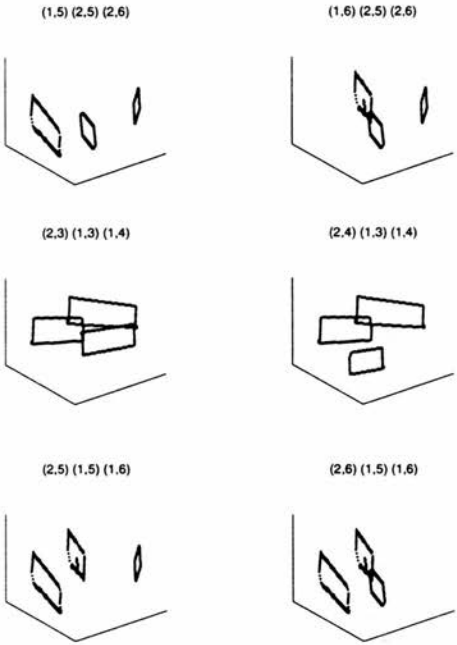


Figure 5.13: Candidate grasping sets for the polyhedron (4 of 4).

5.4.3 Choose aperture plane for each candidate grasping set

Before proceeding further, an important concept must be introduced.

Definition 7 *The* **preshape aperture** *is the set of relative fingertip positions of the hand in the preshape.*

This is an extension of the usage of "aperture" in the psychology literature, where it is used to refer to the distance between the first finger and thumb (indeed, the preshape is usually defined to be the shape of the hand where the aperture is a maximum, as in [Jeannerod 81]).

The following assumption is made:

Assumption 1 *The preshape aperture lies in a plane.*

For a two or three digit hand, this is automatically true. For a hand with four or more digits, this may or may not be true depending on the specification of the preshapes. [2] The **aperture plane** of a preshape is the plane in which the fingertips of the preshape will lie. An additional assumption is made:

Assumption 2 *For a given grasp family, all aperture planes of a grasp family are parallel to one another.*

This means that the range of possible preshape apertures can be drawn in a single plane.

For the SAM hand, in all the preshapes defined in Chapter 3, each digit configuration is identical and so this is true. Strictly speaking, the maximum aperture width occurs when each digit is fully extended, *i.e.* $\alpha = \beta = \gamma = 0$. However, this configuration is clearly incapable of grasping an object. The range of the aperture positions is determined by the type of preshape that will be formed. For example, in the manipulation grasp, the maximum extension of a digit is assumed to occur at $\alpha = 0, \beta = \frac{3\pi}{10}, \gamma = \frac{\pi}{5}$,

[2] For the human hand, the preshape aperture is approximately planar.

which gives the digit configuration shown in Figure 5.14. The minimum bound on the manipulation aperture width occurs when the tip of the thumb crosses the line joining the tip of each finger, as shown in Figure 5.15.

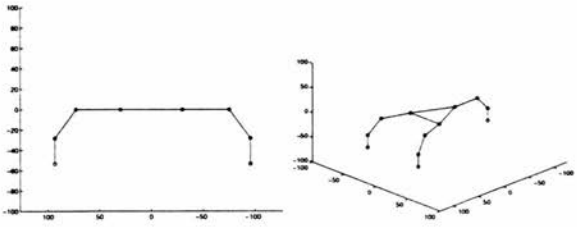


Figure 5.14: Hand configuration giving maximum manipulation aperture, shown from two views.

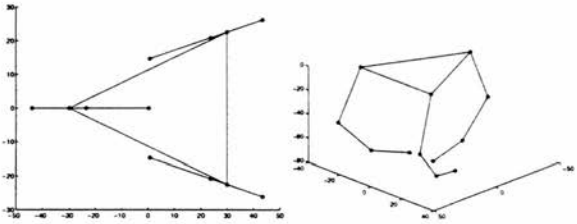


Figure 5.15: One (of many) hand configurations giving the minimum manipulation aperture, shown from two views.

This gives the range of manipulation apertures shown in Figure 5.16. The maximum aperture is marked by an asterisk and the minimum aperture by a circle. The dotted lines show the projection of the palm into the aperture plane. The tip and lateral grasps have different bounds on the aperture range, so as to ensure that given any aperture in the range, a tip or lateral preshape can be formed.

If the position of the i^{th} fingertip in the minimum aperture is given by $\mathbf{r}_i^{min}$ and its position in the maximum aperture is given by $\mathbf{r}_i^{max}$, all other apertures have fingertip positions given by:

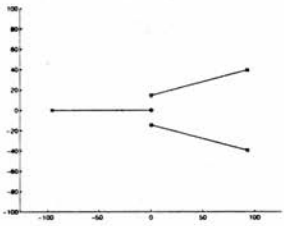


Figure 5.16: Range of apertures for a manipulation grasp. The maximum aperture is marked by an asterisk and the minimum aperture by a circle. The dotted lines show the projection of the palm into the aperture plane.

$$\mathbf{r_i}^{ap} = \mathbf{r_i}^{\max} + \nu(\mathbf{r_i}^{\min} - \mathbf{r_i}^{\max}) \qquad 0 \leq \nu \leq 1 \tag{5.12}$$

Note that ν is the same for each fingertip.

To simplify the planning of the aperture position, it is assumed that during digit closure each fingertip moves in a straight line in the aperture plane.

Assumption 3 *The digit trajectories are straight lines in the aperture plane.*

These straight lines are in fact the line joining the minimum and maximum aperture positions, though in the final grasp each fingertip may lie at any position along its respective trajectory, *i.e.* the position of the i^{th} fingertip is given by

$$\mathbf{r_i}^{grasp} = \mathbf{r_i}^{ap} + \nu_i(\mathbf{r_i}^{\min} - \mathbf{r_i}^{ap}) \qquad 0 \leq \nu_i \leq 1 \tag{5.13}$$

Note that ν is now *different* for each fingertip. This reduces aperture planning to a planar problem. If the aperture position is not known, then $\mathbf{r_i^{ap}} = \mathbf{r_i^{max}}$.

These digit trajectories are referred to as the **aperture closure trajectories**. It may be possible to make Assumption 3 completely true by a suitable choice of digit trajectories later on in the algorithm. With the Salisbury hand, for instance, all the joints of each digit can be actuated simultaneously in order to achieve the required trajectory. However, even with a simpler digit closure, with actuation of just one joint

per digit (as in a typical digit closure motion with the human hand), as long as the preshape is reasonably close to the final grasping positions and the hand configuration is reasonably close to the centre of its workspace, this is a good approximation to the real digit trajectories.

The aperture plane is chosen to be horizontal. This minimises the chance of the robot hand colliding with the ground plane during any part of the grasping action. As stated in Chapter 1, one of the advantages of dextrous hands is that they are able to grasp objects in a wide variety of configurations, so this should still enable a high quality grasp to be found. Furthermore, the final stage of the grasp planner could be to optimise the hand configuration using local inverse kinematics and at this stage the aperture plane is allowed to move from the horizontal.

Assumption 4 *The aperture plane is at, or close to* [3]*, horizontal.*

Given that the aperture plane is initially set to be horizontal, lower and upper bounds on the height of the aperture plane for a particular candidate grasping set are found by finding the horizontal planes of minimum and maximum height that intersect all patches of the grasping set. The patches are then divided into horizontal slices of fixed width w: typically $w = 2mm$. The height of a slice i is given by z_i, and it consists of the set of points from the candidate grasping set that have z coordinates between $z_i - w$ and $z_i + w$. The aperture plane will lie at the centre of one of these slices, which is chosen by consideration of the following (possibly conflicting) criteria:

1. There should be maximum allowable error in wrist position.
2. The grasp should require the minimum amount of friction.

The first criterion means that slices close to the centre of the range of possible aperture slices are preferred. The second criterion means that it is desirable that the contact normals lie as close to the aperture plane as possible (at this stage the desired orientation of the contact normals *within* the aperture plane is not considered). Slice i is

[3] If the final grasp is adjusted.

therefore given a score $\mathcal{A}_i$, referred to as the **aperture plane metric**, according to the following formula:

$$n_z^t = \frac{1}{N_t}\left(\sum_{j=1}^{N_t} \mid n_z^{tj} \mid\right) \tag{5.14}$$

$$n_z^f = \frac{1}{N_f}\left(\sum_{j=1}^{N_f} \mid n_z^{fj} \mid\right)$$

$$\mathcal{A}_i = \frac{1}{\mid 1+\alpha_{\mathcal{A}} \mid}\left(\frac{n_z^t + n_z^f}{2} + \alpha_{\mathcal{A}} \mid 2\frac{z_i - z_c}{z_u - z_l} \mid\right)$$

N_t is the number of thumb points lying within the slice (which has a typical finite width of 2mm), N_f the number of finger points in the slice and z_i is the height of the slice. n_z^{tj} is the z component of the normal of the j^{th} thumb point in the slice, n_z^{fj} is the z component of the normal of the j^{th} finger point in the slice. z_u and z_l are the heights of the highest and lowest slices respectively and $z_c = \frac{(z_l+z_u)}{2}$. The weighting between the two terms of the metric is given by $\alpha_{\mathcal{A}} \geq 0$: this selects between Criteria 1 and 2 listed above. The term post-multiplying $\alpha_{\mathcal{A}}$ varies between 0 and 1; $\alpha_{\mathcal{A}}$ is typically chosen to be 0.1, and the slice that minimises $\mathcal{A}_i$ is chosen. For planar thumb and finger patches, n_z^t and n_z^f are constant over all the slices so the central slice is always chosen.

The aperture plane is then placed at the same height as the chosen slice.

In Figures 5.17– 5.20 the candidate grasping sets for the polyhedron example are shown, with the intersection of the aperture plane with each patch shown as a line drawn on the patch.

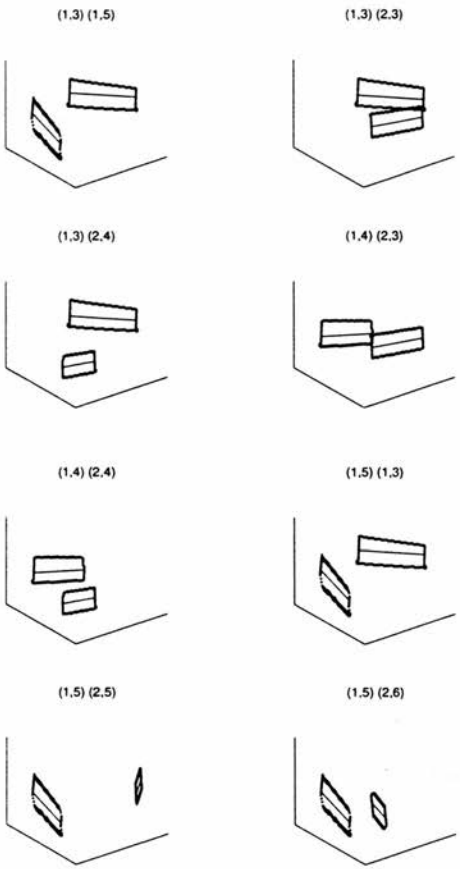


Figure 5.17: Aperture plane found for each candidate grasping set. For each candidate grasping set, the intersection of its aperture plane with each patch is drawn (1 of 4).

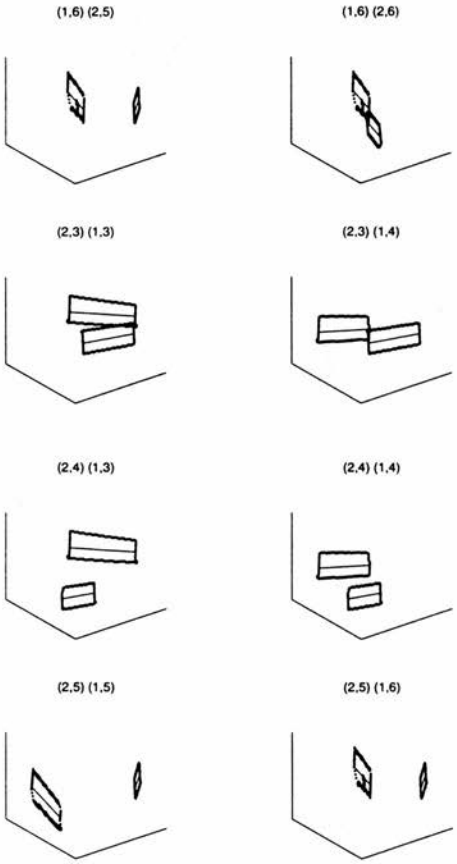


Figure 5.18: Aperture plane found for each candidate grasping set. For each candidate grasping set, the intersection of its aperture plane with each patch is drawn (2 of 4).

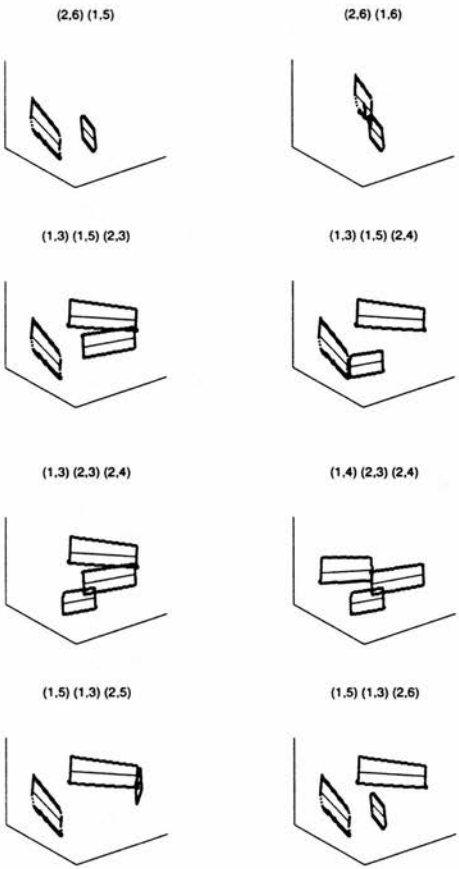


Figure 5.19: Aperture plane found for each candidate grasping set. For each candidate grasping set, the intersection of its aperture plane with each patch is drawn (3 of 4).

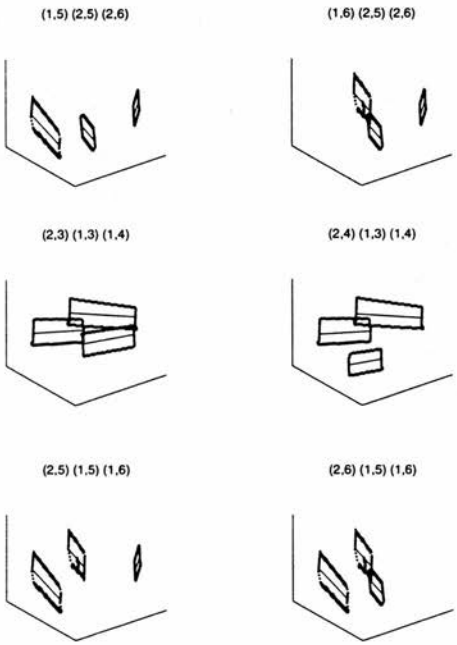


Figure 5.20: Aperture plane found for each candidate grasping set. For each candidate grasping set, the intersection of its aperture plane with each patch is drawn (4 of 4).

5.4.4 Order candidate grasping sets

The amount of processing up to this point has been relatively light, but finding the aperture for each candidate grasping set is more computationally expensive. At this point, therefore, the candidate grasping sets should be ordered according to an estimate of their quality. This is difficult because it is very hard to establish from which patch the best grasp will come without planning a grasp on each and every patch.

The aperture plane metric $\mathcal{A}_i$ of each grasping set is a good indicator of quality. An initial ranking of the candidate grasping sets is therefore based on the aperture plane metric for each one. However for objects consisting of several parallel planar faces (such as the polyhedron example), this will not discriminate adequately between candidate grasping sets because there will only be very small differences in the metric (there will always be some differences because of image noise and sampling). However, the aperture planning algorithm (see Section 5.4.5) is fast enough to be run at a *coarser resolution* for each candidate grasping set. The estimated quality of the grasping set is given by:

$$\mathcal{Q}_i = \frac{\mathcal{A}_i + \alpha_{\mathcal{Q}} \mathcal{F}_i}{1 + \alpha_{\mathcal{Q}}} \tag{5.15}$$

where $\mathcal{A}_i$ is the quality of the aperture plane and $\mathcal{F}_i$ the quality of the *coarse* aperture fit of the i^{th} candidate grasping set (see Section 5.4.5 for the formulation of $\mathcal{Q}_i$). This is an estimate of the quality of the grasp that will be found for that candidate grasping set. The aperture plane quality term evaluates the quality with respect to the z (vertical) direction, while the aperture fit quality term evaluates the quality within the xy (horizontal) plane. $\alpha_{\mathcal{Q}}$ is typically chosen to be 0.1. For the polyhedron example, Table 5.1 shows the candidate grasping sets ordered according to this formula, with sampling parameters $(\pi/16, \pi/16, 8, 4)$. Section 5.4.5 explains what these parameters mean and how they are chosen. The grasp type is lateral — as shall be seen, the type of grasp affects the aperture fitting stage, and hence the candidate grasp set ordering.

It is much more expensive to estimate $\mathcal{F}_i$ than $\mathcal{A}_i$ but this need not necessarily be done for all candidate grasping sets if the following procedure is followed:

Grasping Set	$\mathcal{A}$	$\mathcal{F}$	$\mathcal{Q}$
(1,3) (1,5) (2,4)	0.0287	0.2831	0.0518
(1,5) (1,3) (2,5)	0.0349	0.2566	0.0551
(2,5) (1,5)	0.0337	0.2708	0.0553
(2,4) (1,3) (1,4)	0.0291	0.3448	0.0578
(2,5) (1,5) (1,6)	0.0309	0.3746	0.0621
(1,4) (2,3) (2,4)	0.0336	0.3834	0.0654
(1,3) (1,5) (2,3)	0.0410	0.3224	0.0666
(1,3) (2,3) (2,4)	0.0339	0.4907	0.0754
(2,6) (1,5)	0.0293	0.5558	0.0772
(2,3) (1,3) (1,4)	0.0442	0.4308	0.0793
(2,3) (1,3)	0.0443	0.4365	0.0800
(1,5) (2,5) (2,6)	0.0347	0.5364	0.0803
(1,5) (1,3) (2,6)	0.0282	0.7171	0.0909

Table 5.1: Ordered candidate grasping sets: $\mathcal{A}$ is aperture plane metric (see Section 5.4.3), $\mathcal{F}$ is coarse aperture fit metric (see Section 5.4.5), $\mathcal{Q}$ is grasping set quality metric.

1. Calculate $\mathcal{A}_i$ for each candidate grasping set.
2. Find the candidate grasping set with the minimum value of $\mathcal{A}$, $\mathcal{A}_{\text{min}}$.
3. Discard candidate grasping set with $\mathcal{A}_i > \mathcal{A}_{\text{min}} + \alpha_{\mathcal{Q}}$ (because $0 \leq \mathcal{F}_i \leq 1$).
4. Calculate $\mathcal{F}_i$ and hence $\mathcal{Q}_i$ for each remaining candidate grasping set.
5. Select the candidate grasping set with the minimum value of $\mathcal{Q}_i$.
6. Plan grasp for this candidate grasping set.
7. If the grasp planned is satisfactory, stop. Otherwise, discard the selected candidate grasping set from all further consideration and go to 2.

This procedure was not followed in generating the results shown in Table 5.1, because it is instructive to display results for all the candidate grasping sets.

5.4.5 Fit aperture

The DOF of the problem have now been heavily constrained by the assumption that the digit trajectories are straight lines in a horizontal aperture plane, and by fixing the height of the plane using the aperture plane metric. The position and orientation of the digit trajectories are therefore determined by just 4 DOF: 2 DOF for the position in the aperture plane, 1 DOF for the orientation in the aperture plane and 1 DOF for the abduction of the digits (assuming that this is fixed during the digit closure, as it is in all the grasp families used in this thesis). The aperture is fitted by intersecting the aperture closure trajectories with the graspable surfaces (or rather curves, as the problem has been reduced to a planar one).

The aperture is fitted so as to minimise the **aperture metric**. This is an adaptation of the actual grasp metric and aims to produce an aperture fit that will lead to a high quality grasp metric. It is essentially of the same form as the grasp metric, but with some adaptations to allow for the fact that the kinematics of the hand are not specified completely by the aperture and that the aperture fit is an essentially planar process. It is assumed that

- The digit trajectories are straight lines in the aperture plane.
- Each digit has one DOF, specifying the position along its trajectory.
- Forces are exerted in the direction of the digit trajectory.
- There is not a complete specification of the joint configuration of the hand.

Figure 5.21 shows an aperture fit to an example set of graspable features. The aperture closure trajectories lie in one plane and so the properties of the aperture fit are essentially 2D. The solid lines are the intersection of the graspable features with the aperture plane, and the dotted lines outline the object. The aperture is marked by circles, which represents where the fingertips of the preshape will lie. The aperture closure trajectories are solid thin lines. The length along each closure trajectory is called the **distance to contact** and for digit i is denoted by d_i. The angle of friction required (under 2D analysis) is ϕ_i.

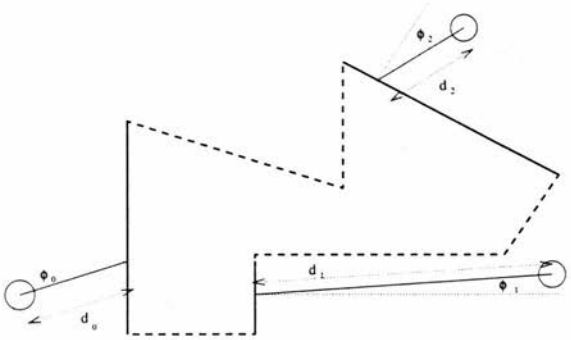


Figure 5.21: An example aperture fit. The aperture closure trajectories lie in one plane and so the properties of the aperture fit are essentially 2D. The solid lines are the intersection of the graspable features with the aperture plane, and the dotted lines outline the object. The aperture is marked by circles, which represents where the fingertips of the preshape will lie. The aperture closure trajectories are solid thin lines. The length along each closure trajectory is called the **distance to contact** and for digit i is denoted by d_i. The angle of friction required by each digit (under 2D analysis) is ϕ_i.

The aperture metric is based on the intersection of the aperture closure trajectories (see Section 5.4.3) with the slices of the graspable features within the aperture plane. The components of the grasp metric are then adapted as follows:

- $\mathcal{M}_1^M$, the equilibrium metric. There is no specification of joint configuration at the aperture fitting stage, so the equilibrium metric will always be zero for a fitted aperture, since all three digit trajectories (and hence contact forces) intersect at one point, and the finger trajectories oppose the thumb trajectory.
- $\mathcal{M}_2^M$, the required friction metric — retained.
- $\mathcal{M}_3^M$, a task wrench space metric — as in the rest of this thesis, the task wrench space metric is neglected.
- $\mathcal{M}_1^K$, the reciprocal of the norm of joint torque due to unit fingertip force — neglected, since there is no joint information at this stage.
- $\mathcal{M}_2^K$, the distance from desired joint configuration — neglected, since there is no

joint information at this stage.

- $\mathcal{M}_3^K$, the distance from desired angle between contact surface and distal links — this reduces to the sum of the squares of the distance to contact.
- $\mathcal{M}_4^K$, a binary collision metric — retained.

These reduce to the following three metrics (which should be minimised):

- m_1, the mean distance to contact for each digit (from $\mathcal{M}_3^K$).
- m_2, the maximum angle between a digit trajectory and a contact normal (from $\mathcal{M}_2^M$).
- m_4, the binary collision metric (from $\mathcal{M}_4^K$).

m_4 is a direct interpretation of the collision metric for the straight line digit trajectories that are assumed in the aperture fitting process. If any digit trajectory intersects with part of the object before it reaches its designated graspable feature, it is assumed that there is a collision.

m_1 is derived from $\mathcal{M}_3^K$, the contact angle metric. m_2 is directly related to $\mathcal{M}_2^M$, the required friction metric. If the final contacts are in the same plane as the aperture (*i.e.* Assumption 3 is true) and all the object normals lie in the aperture plane, $m_2 = \mathcal{M}_2^M$. In reality, this will rarely be true with much degree of accuracy, in which case the relationship between m_2 and $\mathcal{M}_2^M$ depends how far from this ideal the final grasp configuration differs.

As explained in Chapter 3, preshapes can be associated with certain contact geometry to give examples of an "ideal grasp". The above criteria can be viewed as fitting the hand/object configuration as close as possible to an ideal grasp. In order to ensure robustness of grasps against positioning errors, whilst resolving ambiguities in aperture fit, the following criterion is added:

- m_3, the mean distance from contact point to graspable feature centre.

If d_i is the distance to contact for digit i, $d_{\max}$ denotes the maximum length of digit closure trajectory (which is the same for each digit) and τ_1 is a threshold on the maximum feasible distance to contact, m_1 is given by:

$$\begin{aligned} \psi_1 &= \frac{1}{N}\sum_{i=1}^{N}\frac{d_i}{d_{\max}} \\ m_1 &= \psi_1/\tau_1 \\ &\quad \text{if} \quad \psi_1 < \tau_1 \\ m_1 &= \infty \\ &\quad \text{otherwise} \end{aligned} \tag{5.16}$$

If ϕ_i denotes the angle between the contact normal and digit trajectory and τ_2 is the maximum angle allowed, m_2 is given by:

$$\begin{aligned} \psi_2 &= \max(\phi_i) \\ m_2 &= \psi_2/\tau_2 \\ &\quad \text{if} \quad \psi_2 < \tau_2 \\ m_2 &= \infty \\ &\quad \text{otherwise} \end{aligned} \tag{5.17}$$

If r_i is the distance between the contact point and the centre of the contact feature divided by the length of the contact feature ($0 < r_i < \frac{1}{2}$).

$$m_3 = \frac{2}{N}\sum_{i=1}^{N} r_i \tag{5.18}$$

The collision metric, m_4 is binary:

$$\begin{aligned} m_4 &= 0 \\ &\quad \text{if there is a collision} \end{aligned} \tag{5.19}$$

$$m_4 = \infty$$

otherwise

All these metrics vary between 0 and 1, or are infinite. The overall aperture metric is given by:

$$\mathcal{M}^{ap} = \frac{\mu_1 m_1 + \mu_2 m_2 + \mu_3 m_3 + m_4}{\mu_1 + \mu_2 + \mu_3} \tag{5.20}$$

The choice of μ_1 and μ_2 is based on the coefficients of the grasp metric — see Chapter 6 for an examination of the relationship between the aperture metric and the grasp metric. The coefficient μ_3 has no direct bearing on the grasp metric; it is there to resolve ambiguities and preserve robustness with respect to positioning error. It is therefore set to an order of magnitude less than the smaller of μ_1 and μ_2, *i.e.*

$$\mu_3 = \frac{1}{10}\min(\mu_1, \mu_2) \tag{5.21}$$

The setting of μ_1 and μ_2 cannot be resolved satisfactorily by theory. In Chapter 6 the effect of aperture metric on grasp metric is investigated. For now, the following parameters are used:

- $\mu_1 = 20$
- $\mu_2 = 10$
- $\mu_3 = 1$

These values reflect the importance of μ_1 in satisfying the assumption that the final contacts lie in the aperture plane. If coplanar digit trajectories are possible, then it may be wise to increase μ_2. The setting of exact coefficients should ultimately be determined by extensive experimentation with a real system, and is therefore not investigated to great extent, except for an examination of what each individual term achieves in Chapter 6.

Figures 5.22– 5.23 show the best aperture fit to each candidate grasping set. The aperture fitting score, multiplied by 100, is shown above each one. They are shown in

the order established in the previous stage, which is not necessarily reflected by the aperture fitting scores, due to a coarser sampling in the ordering. When viewing the results, the reader should bear in mind that of the terms in the aperture metric, the distance to contact term has the biggest weight, and the next biggest is the required friction metric. The highest ranked grasp has a very small distance to contact and fairly small angles between graspable features and digit trajectories (and it is assumed that the contact forces lie in the same direction as the digit trajectories). The third highest-ranked set has a distance to contact that is very close to zero, but the angle between the thumb trajectory and its graspable feature pulls down its ranking.

The aperture is fitted by sampling of the 4 dimensional solution space, using the thumb feature as the basis for the parameterisation. The four parameters are:

- The intersection of the thumb trajectory along its graspable feature (a line or curve in the aperture plane).
- The position along the thumb trajectory at the intersection with the thumb surface.
- The orientation of the thumb trajectory in the aperture plane.
- The abduction of the hand.

By using this set of parameters, it can be ensured that at least the thumb trajectory intersects the thumb graspable feature at each aperture position and this bounds the solution space sufficiently to make the exhaustive search tractable.

The parameters are sampled in the following order:

- Abduction, δ $(0 < \delta < 30°)$
- Distance along thumb patch, s $(0 < s < s_{\max})$
- Orientation, θ $(-45° < \theta < 45°)$
- Distance along thumb trajectory, d $(0 < d < d_{max})$

The abduction parameter is varied in the outer loop because it varies the actual shape of the aperture: the others just vary the position of the aperture. It was found empirically that for a given δ, s, θ, there was a single minimum of d, so this is found by a simple gradient descent in one dimension.

The determination of appropriate sampling rates is a complex issue. Ideally, the sampling resolution of each parameter should be equivalent in some canonical frame of reference, such as the fingertip positions. However, the change in fingertip position depends on the shape of the graspable features as well as the change in parameters. This means that the absolute sampling resolution would change at each point in the search. [4] The sampling resolution is therefore set empirically.

The sampling rates in the δ, θ, d parameters are constant, whilst the number of samples in s is constant. This ensures that the number of samples is constant for a given task, even though the size of the thumb graspable feature varies. The following sampling rates are used:

- $\Delta\delta = \pi/32$
- $\Delta s = \frac{s_{max}}{16}$
- $\Delta\theta = \pi/32$
- $\Delta d = 2$ mm

[4] It would be calculated by working out the intersection of the aperture with the graspable features at each point. Each intersection point is expressed as a function of the sampling parameters and then the coordinates are differentiated with respect to the parameter whose sample resolution is to be determined. This gives the small change in fingertip position with respect to a small change of the parameter, and the sampling rate can then be chosen to give a prescribed change in fingertip position.

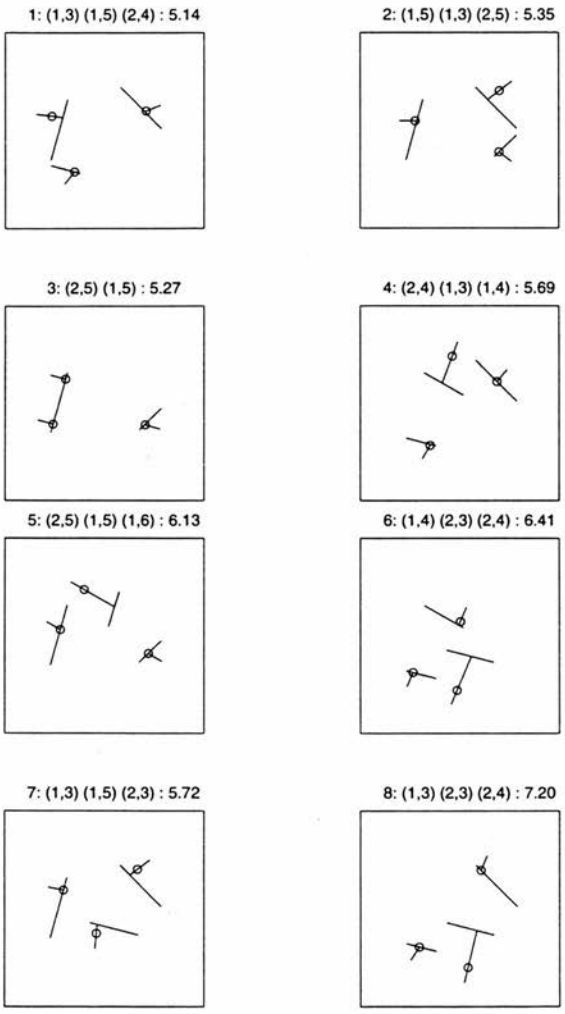


Figure 5.22: Aperture fits to each grasping set, ordered left to right, top to bottom.

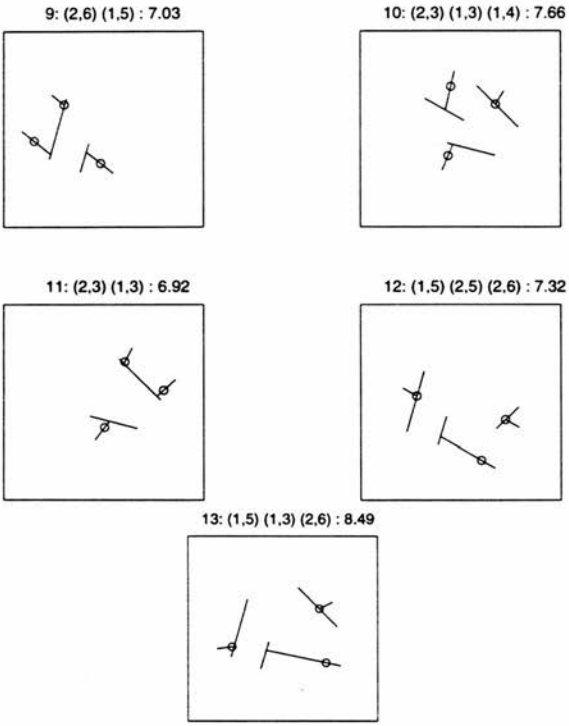


Figure 5.23: Aperture fits to each grasping set (cont.), ordered left to right, top to bottom.

5.4.6 Fit preshape

Once the aperture has been found, the preshape can be determined. The aperture determines the positions of the fingertips and the abduction of the hand in the preshape. The preshape is further specified by 2 joint angles — the angles of flexion at the proximal and middle joints respectively (α and β as shown in Figure 3.8). The aperture forms an isosceles triangle (see Figure 3.10): the length of the line joining the tips of the fingers is r and the length of the perpendicular from thumb to the line joining the fingers is w. The hand kinematics are constrained by the requirements

$$2p + (1 + \cos\delta)L = r \tag{5.22}$$

$$2q + (2\sin\delta)L = r \tag{5.23}$$

where L is the length of a projection of a digit in the aperture plane. The above two equations can be solved for δ and L. α and β, the angles of flexion, can be calculated as follows once L is known:

$$L = a\cos\alpha + b\cos(\alpha+\beta) + c\cos(\alpha + \frac{5}{3}\beta) \tag{5.24}$$

There are therefore two unknowns and one equation. This means that there is one DOF in the preshape which is assumed to be α. This is determined by the task requirements.

To recap from Chapter 3, the possible preshapes are TIP, LATERAL, and MANIPULATION. They are defined in the following way:

- The TIP preshape keeps the angle between the contact surface and the distal link of each digit at $\frac{\pi}{4}$, *i.e.* $\alpha+\beta+\gamma = \frac{3\pi}{4}$. This gives a contact surface at the tips of each digit and determines α uniquely.
- The LATERAL preshape keeps the distal link of each digit parallel to the contact surface, *i.e.* $\alpha+\beta+\gamma = \frac{\pi}{2}$. This gives a contact surface on the insides of the distal link of each digit and determines α uniquely.

- The MANIPULATION preshape minimises the sum of the squares of the deviations of the joints from their central values, *i.e.* it minimizes $(\alpha - \alpha_c)^2 + (\beta - \beta_c)^2 + (\gamma - \gamma_c)^2$. This gives a preshape which has maximum leeway for object manipulation between the digits. The angle of the distal link is constrained so that it cannot point away from the "center" of the preshape, *i.e.* $\alpha + \beta + \gamma \geq \frac{\pi}{2}$. This is done to make it less likely that collisions occur between the object surface and the hand. It is useful if there is no preference for a tip or lateral grasp. α is therefore chosen to minimise $(\alpha - \alpha_c)^2 + (\beta - \beta_c)^2 + (\gamma - \gamma_c)^2$.

α is varied within its range at 1° intervals. For each value of α, β is found such that Equation 5.24 is satisfied, to give one of the possible set of preshapes. The preshape is then chosen using the task criteria listed above.

Figures 5.24– 5.25 show the preshape fitted to each of the apertures in Figure 5.22– Figure 5.23. Normally, the preshape would only be fitted to the best aperture. However, to illustrate the algorithm working on a range of different candidate grasping sets, the preshapes are shown for each aperture. Note that many of the preshapes consist of fully extended digits, which is an indication that the object to be grasped is too thin to be grasped by a true lateral grasp: the digit trajectories will deform the final grasp away from the true lateral configuration that the preshape represents.

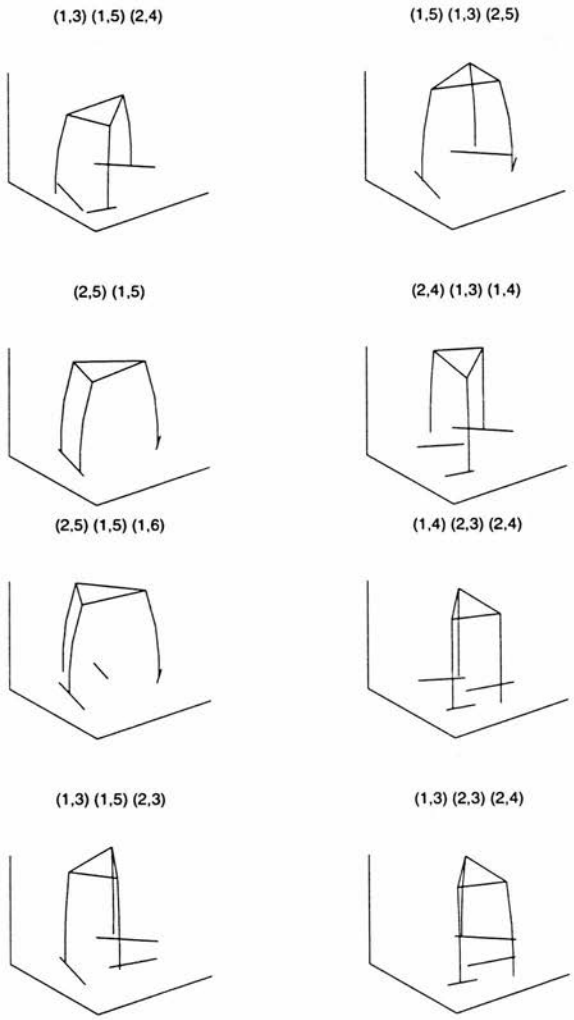


Figure 5.24: Preshape fits to each grasping set.

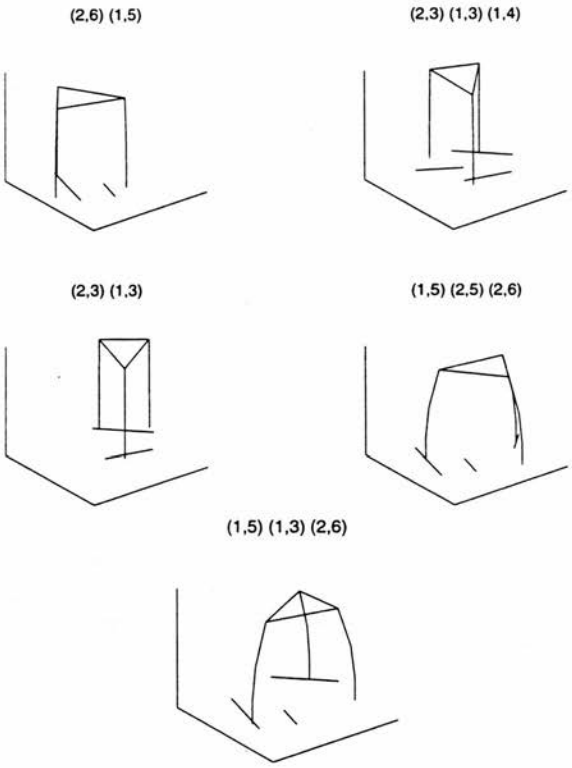


Figure 5.25: Preshape fits to each grasping set.

5.4.7 Choose digit trajectory

The digit trajectory is chosen from one of two possibilities: the *proximal* digit trajectory and the *distal* distal trajectory (see Chapter 3). In [Wren & Fisher 95a] and [Wren & Fisher 95b] the digit trajectory was chosen so as to minimise the deviation of the final grasp from the preshape (*i.e.* the "ideal" grasp). This was done in the absence of a grasp metric. Since a grasp metric has been formulated in this thesis, the digit trajectory is chosen such that the grasp metric is minimised. In Chapter 6 an alternative method of digit trajectory generation is discussed, which constrains the digit trajectory to lie on the straight lines assumed in the aperture fitting.

The proximal digit trajectory is computed by a simulation of proximal joint actuation for each digit by increasing the angle of the proximal joint in small steps. At each stage a test determines if a digit has crossed the surface equation of its graspable feature. If it has, a test determines if the intersection point is within the boundary of the graspable surface. If it is, digit has intersected its graspable feature and it stops moving. If all the digits intersect with graspable features, the grasp has been found. A similar method is used to compute the distal trajectory, by movement of the distal joint instead of the proximal joint.

Figures 5.26– 5.29 show each of the grasps formed by proximal trajectories from the side and the top. Each surface involved in the grasp is outlined, and the contact points are shown as asterisks. The contact normals are drawn as dotted lines. Table 5.2 shows the separate components of the grasp metric for each of the lateral-proximal grasps shown. The coefficients for each term are displayed in the top row, and the final column shows the total grasp metric. Note that the first and third ranked grasping sets actually yield relatively poor grasps. This is because the distance to contact term of the metric is being weighted too heavily with respect to the contact angle term. In Chapter 6, however, grasps will be planned on curved objects. For such objects, the distance to contact metric is much more important because of a greater difference in the estimated grasp quality using the straight line digit trajectories and the actual grasp quality with proximal or distal digit trajectories.

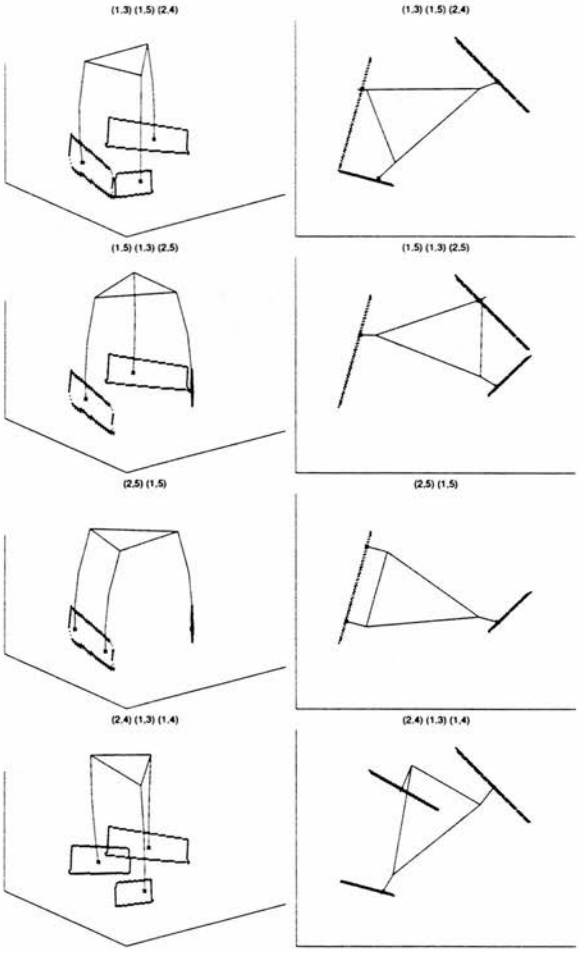


Figure 5.26: Lateral-proximal grasps (1 of 4).

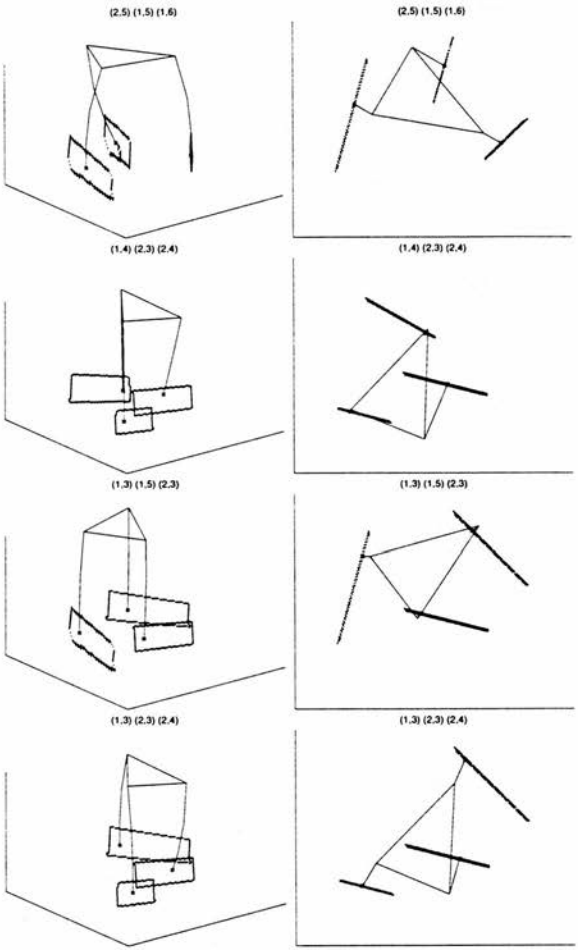


Figure 5.27: Lateral-proximal grasps (2 of 4).

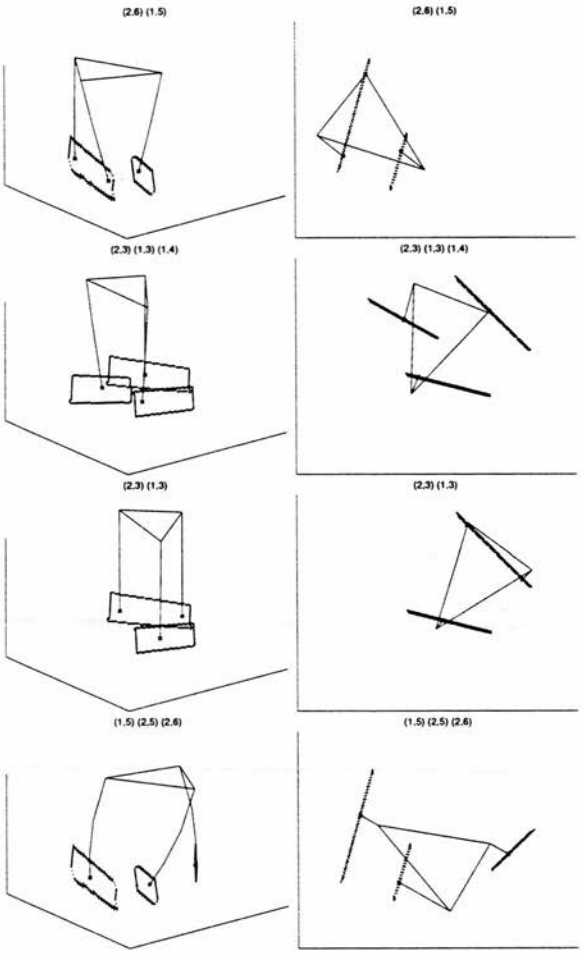


Figure 5.28: Lateral-proximal grasps (3 of 4).

	Required friction $\lambda_2^M = 20$	Joint Torque $\lambda_1^K = 2$	Joint Limit $\lambda_2^K = 1$	Contact Angle $\lambda_3^K = 10$	Total
(1,3) (1,5) (2,4)	0.5983	0.0088	0.4508	0.0101	0.3799
(1,5) (1,3) (2,5)	0.3359	0.0097	0.4978	0.0142	0.2236
(2,5) (1,5)	0.6677	0.0077	0.4458	0.0018	0.4192
(2,4) (1,3) (1,4)	0.3364	0.0079	0.4869	0.0507	0.2345
(2,5) (1,5) (1,6)	0.3376	0.0080	0.5172	0.0567	0.2380
(1,4) (2,3) (2,4)	0.2134	0.0084	0.7784	0.0872	0.1798
(1,3) (1,5) (2,3)	0.2269	0.0114	0.6358	0.0169	0.1626
(1,3) (2,3) (2,4)	0.5029	0.0080	0.4533	0.0890	0.2359
(2,6) (1,5)	0.5236	0.0078	0.7727	0.0447	0.3548
(2,3) (1,3) (1,4)	0.3954	0.0079	0.7563	0.0485	0.2777
(2,3) (1,3)	0.5032	0.0078	0.7203	0.0166	0.3323
(1,5) (2,5) (2,6)	0.3441	0.0083	0.5512	0.1157	0.2608
(1,5) (1,3) (2,6)	0.5492	0.0134	0.5308	0.2200	0.4164

Table 5.2: The terms of the grasp metric for lateral-proximal grasps.

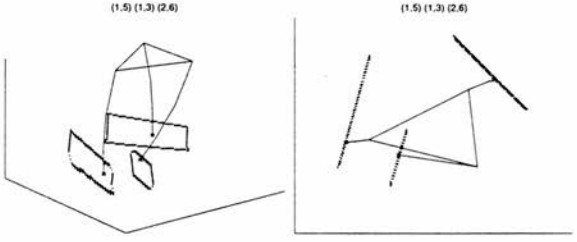


Figure 5.29: Lateral-proximal grasps (4 of 4).

5.4.8 Extend hand model

So far, the grasp has been planned for a "stick" hand, *i.e.* a hand with links of zero width. In fact, this is equivalent to modelling the distal link as consisting of a very thin hard cylinder surrounded by very soft material, which will yield on contact with the object. A more complex alternative would be to model the distal link as being a rigid cylinder of finite width, rounded off by a hemisphere at the fingertip. It is envisaged that in reality the softness of the distal link will be some way between these extremes, and to model it would be computationally expensive and of questionable value since it is very difficult to accurately model non-rigid contacts. For this reason, it is reasonable to pick the simplest model and model the links as having zero width. This is also the rationale behind picking the simplest contact models possible.

At this stage, therefore, the final grasp is adjusted to allow for the finite width of the links. This is chiefly for the purposes of display because in a practical implementation the exact position that the finger joints stop at will be determined by the control scheme rather than the grasp planner.

Each digit is moved backwards along its chosen trajectory until its fingertip is a distance R away from the object, where R is the radius of the distal link. The distance R is measured by inspection of the distance transformed voxmap at each fingertip (see Chapter 4). This scheme may produce a grasp that contacts the object at slightly different points than planned, depending on the shape of the object and the softness of the contacts.

Figures 5.30– 5.31 show the lateral-proximal grasps for a hand of finite width — each link has a radius of 5mm.

5.4.9 Lateral-distal grasp

Figures 5.32– 5.33 show the lateral-distal grasps for the same hand. Table 5.3 shows the grasp metric for each distal grasp. Five grasps have been discarded: this is because the distal trajectories deviate to a much larger degree from the straight line trajectories assumed by the aperture fit. Grasps can be discarded for several reasons:

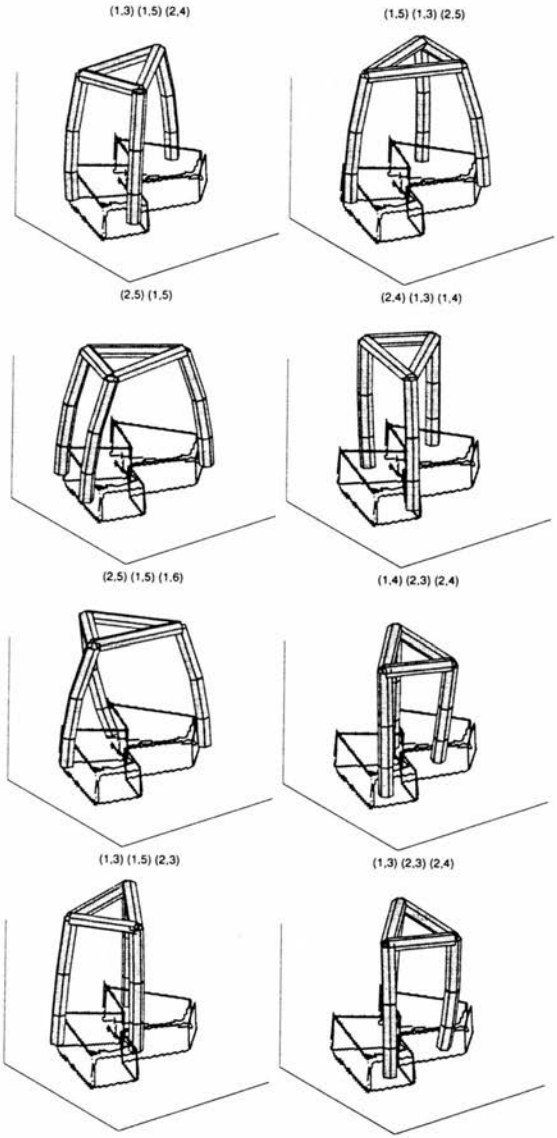


Figure 5.30: Lateral-proximal grasps, with finite link width (2 of 2).

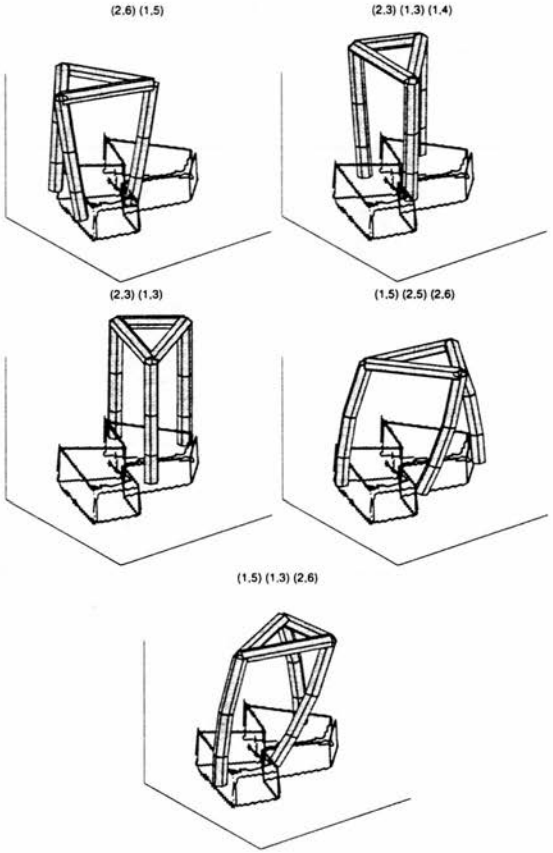


Figure 5.31: Lateral-proximal grasps, with finite link width (2 of 2).

- The quality of one of the metrics, such as the required friction metric, steps up to infinity, *e.g.* (2,5)(1,5).

- There is an unwanted collision with the object, caused by protruding parts of the object, or a failure of the hand to open wide enough to accommodate the width of the digits (which is due to a combination of the effects of joint limits and the fact that the preshape fitting does not take into account the fact that the digits of the hand have a finite width), *e.g.* (2,6)(1,5)

- The digit trajectory misses the desired grasping feature (due to the object shape and distance from preshape to contact), *e.g.* (2,5)(1,5)(1,6) and (1,5)(2,5)(2,6).

- The digit trajectory fails to reach the desired grasping feature (due to joint limits), *e.g.* (1,5)(1,3)(2,6).

Table 5.4 compares the scores from both trajectory types and chooses the best one for each grasp: it can be seen that the proximal trajectory is always better than the distal — for this object. This is because proximal trajectories tend to deviate less from the assumed straight line trajectories and to produce grasp points that are closer to the aperture plane. Since the graspable features are all near-vertical planes, there is no curvature of the contact surface to compensate for this. The difference between proximal and distal trajectories is smallest for grasping sets such as (1,3),(1,5),(2,3), where the distance to contact is very small, which means that the choice of digit trajectories is not crucial. Note that this is not the case for (2,5)(1,5) because this is discarded from the set of lateral-distal grasps due to the fact that the extended hand model results in the distal links of the digits crossing the contact planes: this is only a problem, however, given hard — as opposed to soft — digits.

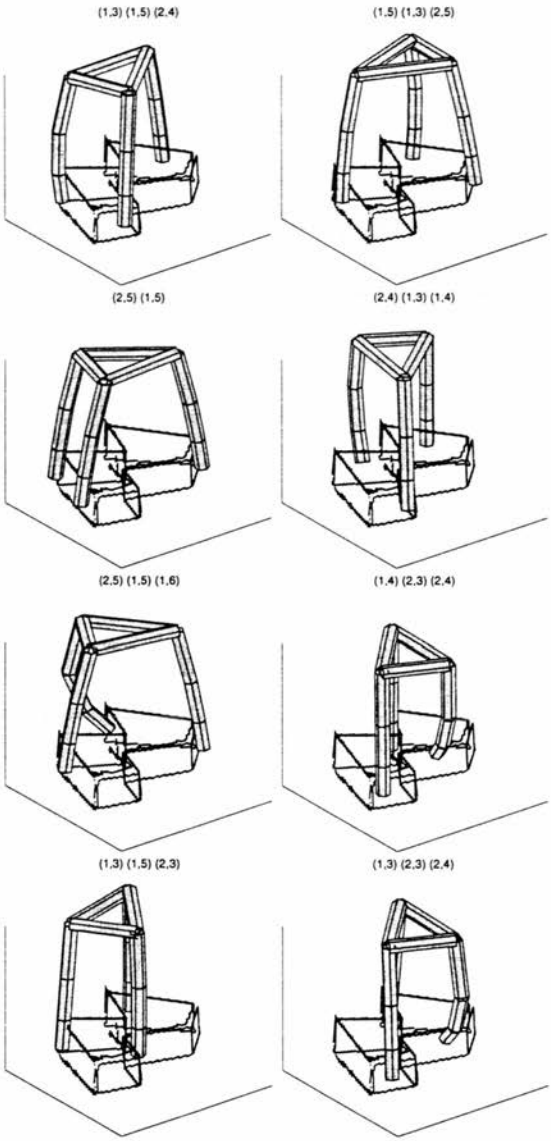


Figure 5.32: Lateral-distal grasps, with finite link width (1 of 2).

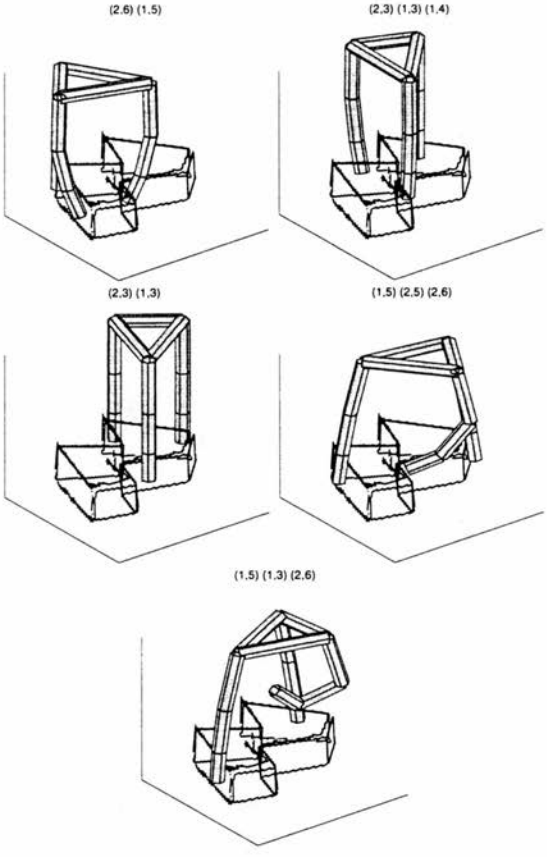


Figure 5.33: Lateral-distal grasps, with finite link width (2 of 2).

	Required friction $\lambda_2^M = 20$	Joint Torque $\lambda_1^K = 2$	Joint Limit $\lambda_2^K = 1$	Contact Angle $\lambda_3^K = 10$	Total
(1,3) (1,5) (2,4)	0.5988	0.0089	0.3828	0.0344	0.3855
(1,5) (1,3) (2,5)	0.3374	0.0097	0.4115	0.0517	0.2332
(2,5) (1,5)	-	-	-	-	-
(2,4) (1,3) (1,4)	0.3489	0.0087	0.3326	0.2401	0.2948
(2,5) (1,5) (1,6)	-	-	-	-	-
(1,4) (2,3) (2,4)	0.2944	0.0101	0.5098	0.4422	0.3285
(1,3) (1,5) (2,3)	0.2266	0.0114	0.5207	0.0555	0.1706
(1,3) (2,3) (2,4)	0.5258	0.0099	0.2789	0.4599	0.4671
(2,6) (1,5)	-	-	-	-	-
(2,3) (1,3) (1,4)	0.3967	0.0084	0.5096	0.2084	0.3196
(2,3) (1,3)	0.5029	0.0079	0.5843	0.0526	0.3389
(1,5) (2,5) (2,6)	-	-	-	-	-
(1,5) (1,3) (2,6)	-	-	-	-	-

Table 5.3: The terms of the grasp metric for lateral-distal grasps.

	Lateral-Proximal	Lateral-Distal	Chosen trajectory (P=proximal,D=distal)
(1,3) (1,5) (2,4)	0.3799	0.3855	P
(1,5) (1,3) (2,5)	0.2236	0.2332	P
(2,5) (1,5)	0.4192	-	P
(2,4) (1,3) (1,4)	0.2345	0.2948	P
(2,5) (1,5) (1,6)	0.2380	-	P
(1,4) (2,3) (2,4)	0.1798	0.3285	P
(1,3) (1,5) (2,3)	0.1626	0.1706	P
(1,3) (2,3) (2,4)	0.2359	0.4671	P
(2,6) (1,5)	0.3548	-	P
(2,3) (1,3) (1,4)	0.2777	0.3196	P
(2,3) (1,3)	0.3323	0.3389	P
(1,5) (2,5) (2,6)	0.2608	-	P
(1,5) (1,3) (2,6)	0.4164	-	P

Table 5.4: Comparison of the grasp scores for lateral-proximal and lateral-distal grasps.

5.4.10 Check for collisions

Once the hand configuration has been modified to allow for a finite link width, the final grasping configuration is checked for collisions by intersecting the hand model with the distance-transformed voxmap, as described in Chapter 4. If there is a collision between the object and any part of the hand, the grasp is discarded and a grasp generated for the next best grasping set.

It may also be desirable to check for collisions at various points during closure. The closure could be sampled finely enough for a good accuracy because the collision checking is so fast. This step may not be necessary, however, for many object-grasp combinations.

5.4.11 Adjust grasp

The goal of this stage is to improve the quality of a collision-free grasp. Similar methods may be used to attempt to free a hand configuration of collisions. At the time of writing, however, this has not been implemented. The algorithm to improve the grasp proceeds as follows:

1. Find the pseudo-inverse and null-space of the hand Jacobian.
2. By variation of the hand configuration in the null-space of the Jacobian, attempt to refine the grasp.
3. Move to the best neighbouring fingertip configuration using the pseudo-inverse of the Jacobian.
4. If no change in grasp - stop, otherwise - go to step 2

The ideal system would, at step 3 perform a null-space search for each fingertip position under consideration. However, it is reasonable to make the assumption that the nullspace solutions will be close for neighbouring fingertip positions, so for the purposes of determining the next move in fingertip position the variation in nullspace can be ignored.

5.5 Summary

Grasps are characterised by the grasp metric, which is developed according to the mechanical and kinematic requirements of the task. Grasp families are used to constrain the degrees of freedom of the hand and forward kinematics are used to plan grasps. The solution space is pruned by use of heuristics based around the preshaping model, but this is coupled with a consideration of the digit trajectories (or an approximation to them) for better fitting of the preshape to the object. The remainder of the solution space has 4 DOF, constrained by the requirement that the thumb contacts its designated graspable feature, and can be explored by exhaustive sampling. Several parameters need to be set for the grasp planner: these are related to the grasp metric or control the sampling of the solution space.

The method was seen to work well for the polyhedral object. Good grasps, according to the task requirements as defined by the grasp metric, were synthesised. It is clear, however, from Tables 5.2 and 5.3, that the ranking of candidate grasping sets is only a very approximate way of finding the optimal grasp: the order of final grasp metrics is quite different to the ordering of the candidate grasping sets.

Chapter 6

Results

In this chapter the behaviour of the algorithm described in Chapter 5 is examined in more detail, with special attention paid to the choice of parameter settings. Lateral, tip and manipulation grasps are planned on a variety of objects. These objects are chosen to be difficult, either in that the geometry of their surface is complex (for example, the tube in Section 6.2.2) or in the number of surface patches (the toy rabbit in Section 6.2.3 or the polyhedron example in Chapter 5).

6.1 Grasp Parameters

The grasp planning process uses several parameters which have to be set by the user. Some parameters, namely the sampling rates, determine the accuracy and optimality of the final result. Others, namely the coefficients of the aperture and grasp metrics, should be set according to the task. In order to perform a given task, the coefficients of the aperture and grasp metrics should be determined by practical experimentation, possibly combined with theoretical analysis.

In this section, the relationship between the coefficients of the aperture metric and the coefficients of the grasp metric is illustrated. First, suggested values for the coefficients of the grasp metric are presented for lateral, tip and manipulation grasps.

6.1.1 The grasp metric coefficients

The formulation of each term of the grasp metric is described in detail in Chapter 5. The grasp metric consists of the mechanical metric and the kinematic metric. The terms of the mechanical metric are as follows:

- $\mathcal{M}_1^M$, the **equilibrium** metric.
- $\mathcal{M}_2^M$, the **required friction** metric.
- $\mathcal{M}_3^M$, the **task wrench** space metric.

Each term varies between 0 and 1 and is weighted by a coefficient λ_i^M $(i = 1 - 3)$. The terms of the kinematic metric are as follows:

- $\mathcal{M}_1^K$, the **joint torque** metric.
- $\mathcal{M}_2^K$, the **joint limit** metric.
- $\mathcal{M}_3^K$, the **contact angle** metric (based on the desired angle between distal link and contact surface).
- $\mathcal{M}_4^K$, the **collision** metric

Again, each term varies between 0 and 1. The weighting of each term is determined by the coefficient λ_i^K $(i = 1 - 4)$. In this section, suitable settings of the grasp metric coefficients are suggested for each task. In practice, such settings should be refined by practical experimentation, possibly using an automated learning process. This is beyond the scope of this thesis.

Two of the terms of the grasp metric, namely the **equilibrium metric** and the **collision metric**, take on values of zero or infinity. For this reason, they may have arbitrary non-zero coefficients. Therefore,

$$\lambda_1^M = \lambda_4^K = 1$$

The **task wrench space metric** is ignored in this thesis, therefore:

$$\lambda_3^M = 0$$

This leaves four remaining terms of interest: the **required friction metric**, the **joint torque metric**, the **joint limit metric** and the **contact angle metric**.

The joint torque metric is an indicator of how easily the hand can exert the fingertip forces. This is a critical term, in that if it takes on a high value, this indicates that the hand cannot exert the required forces, then the grasp is very unsuitable. For this reason, if it lies above a certain threshold, the joint torque term becomes infinite. Below this threshold, the joint torque metric is finite, indicating that the required fingertip forces may be exerted with little or no error.

Assuming that the grasp metric is finite, for the lateral, tip and manipulation grasps, the required friction metric is the most important, as this strongly influences the stability of the final grasp. The order of the other terms depends on the task.

Lateral Grasp

A lateral grasp is usually used when a firm grasp is desired on an object. By keeping the distal links parallel to the contact surface, the contact surface is maximised. This is reflected in the contact angle metric. The joint torque metric is more important than the joint limit metric because lateral grasps are not usually used in conjunction with dextrous manipulation. For this reason, the digit configuration is unlikely to change and so the distance from the joint limits in the original grasp is not very important. As with all the grasps, the required friction term is the most important: this is crucial to any grasp, regardless of the exact task requirements.

1. Required friction
2. Contact angle
3. Joint torque
4. Joint limit

Table 6.1 shows the values of each grasp metric coefficient for the lateral grasp.

Tip Grasp

A tip grasp is usually used when there is to be dextrous manipulation of the object subsequent to grasping. For this reason, the joint limit metric becomes more important relative to the lateral grasp. The importance of the joint torque metric also increases, because the lower the joint torque metric is in the grasp, the lower it is likely to be during the subsequent manipulation. The joint torque and joint limit metrics are therefore given equal weighting and this weighting is increased with respect to the other metrics. The required friction metric is still given most weighting, since it is of fundamental importance that the amount of friction required be as small as possible, and the contact angle metric remains as the second most important metric, because it is this that characterises the tip grasp over the lateral grasp. Table 6.1 shows the values of each grasp metric coefficient for the tip grasp.

Manipulation Grasp

A manipulation grasp has no contact angle metric. The metric that characterises the manipulation grasp is, in fact, the joint limit metric. The ordering of the terms is therefore:

1. Required friction
2. Joint limit
3. Joint torque

Table 6.1 shows the values of each grasp metric coefficient for the manipulation grasp.

6.1.2 The aperture metric

This section is concerned with how to set the aperture metric coefficients, given a set of grasp metric coefficients. There are three aperture metric terms:

	Lateral	Tip	Manipulation
λ_1^M Equilibrium	1	1	1
λ_2^M Required friction	20	20	20
λ_3^M Task wrench	0	0	0
λ_1^K Joint torque	2	5	5
λ_2^K Joint limit	1	5	10
λ_3^K Contact angle	10	10	0
λ_4^K Collision	1	1	1

Table 6.1: Grasp metric coefficients for each type of grasp.

- The distance to contact metric, m_1.
- The required friction metric, m_2.
- The distance to centre metric. m_3.

The coefficients of these three terms are μ_1, μ_2, μ_3 respectively (see Section 5.4.5). These were introduced in [Wren & Fisher 95b] and used in [Wren & Fisher 95a]. In both these papers, little justification was given for use of these parameters and there was no grasp metric from which to derive them. They were used on the grounds that they tend to pull the grasp towards "ideal" grasps, as defined by the grasp family. This is true, but the question of exactly how each term in the aperture metric relates to each term in the grasp metric must be addressed. This is complicated because while the aperture metric is an estimate of the final grasp metric, it also tries to pull the grasp into configurations for which the assumptions made during the planning are true. Furthermore, the relationship between the final grasp metric and the aperture grasp metric depends fundamentally on the object shape. It can be analysed "experimentally" in one of two ways. This is done here by taking a complex object, the polyhedron used in Chapter 5, and for each candidate grasping set (14 in all), comparing the terms of the aperture metric with the final grasp metric. As described in Chapter 5, the aperture

metric is supposed to provide an estimate of the final grasp metric, subject to certain assumptions about the geometry of the object and digit trajectories. The polyhedral object is a good choice because not only is it a "complex" object with many candidate grasping sets, but it consists of just (almost) vertical planes. This means that it satisfies closely the assumptions made about object shape in the aperture fit: assuming that the digit trajectories do not deviate too far from the aperture plane, the required friction and distance to contact terms of the aperture metric give a good estimate of the required friction and contact angle terms of the grasp metric.

Aperture and grasp metrics for the ideal grasp

In Chapter 5 a lateral grasp is synthesised in order to minimise the grasp metric for a particular ideal contact configuration. In this section, the aperture metric for that grasp is derived and the results compared. In Chapter 5, the parameters of the grasp metric were not set to specific values, rather they were ordered according to their magnitudes thus:

$$\lambda_2^M \gg \lambda_3^K \gg \lambda_1^K \gg \lambda_2^K > 0$$

(required friction) (contact angle) (joint torque) (joint limits)

The resulting grasp had $\mathcal{M}_2^M = \mathcal{M}_3^K = 0$, $\mathcal{M}_1^K > 0$ and $\mathcal{M}_2^K > 0$. The preshape for the ideal grasp is the grasp itself: because of the symmetry of the contacts and hand, no digit closure is required. The aperture corresponding to the ideal grasp is therefore the position of the fingertips in the ideal grasp. So, for the ideal grasp described above, the terms of the aperture metric are given by $m_1 = m_2 = 0$ and $m_3 > 0$ (since no information about patch extent is given). Assuming that that $\mu_1 >> \mu_3$ and $\mu_2 >> \mu_3$, it can be seen that the optimal aperture metric does indeed correspond to the optimal grasp metric.

The Required Friction Metric

To assess the effect of the required friction term of the aperture metric, the coefficients of the aperture metric are set as follows:

$$\mu_1 = 1, \mu_2 = 1000, \mu_3 = 1$$

This weights strongly the required friction term , but gives the other terms in the metric very small positive values in order to avoid degeneracy of aperture fits. Lateral-proximal grasps are planned on the polyhedron object. Figures 6.1– 6.2 show the ordered set of aperture fits for this set of coefficients. If the results are compared to those using the standard parameterisatin (see Figures 5.22– 5.23), the effect of the required friction weighting can be seen clearly, in that digit trajectories which are perpendicular to their graspable features are favoured strongly. For instance, the candidate grasping set (2,5),(1,5) in Figure 6.2 would have a much higher aperture metric with the standard parameterisation because, while the required friction is large due to the thumb contact, the fact that the distance to contact is very small means that it still has a good aperture metric and rightly so, since the smaller the distance to contact, the closer the assumptions made in the aperture fitting process are satisfied.

Figure 6.3 shows a graph of the required friction *grasp* metric against the required friction *aperture* metric for the lateral-proximal grasp. The aperture metric varies approximately linearly with the grasp metric, with one outlier. This outlier corresponds to the fifth grasping set (1,5)(1,3)(2,6), shown in Figure 6.1. This has an exceptionally long distance to contact, which means that the plane of the final contacts is skewed from the horizontal a great deal. The grasp forces lie in the plane of the final contacts and are therefore similarly skewed from the horizontal. Since the contact normals all lie (approximately) in the horizontal plane, this means that the required friction increases. The difference therefore arises because this grasp strongly violates the assumption made by the aperture fitting that the final contacts lie in the aperture plane.

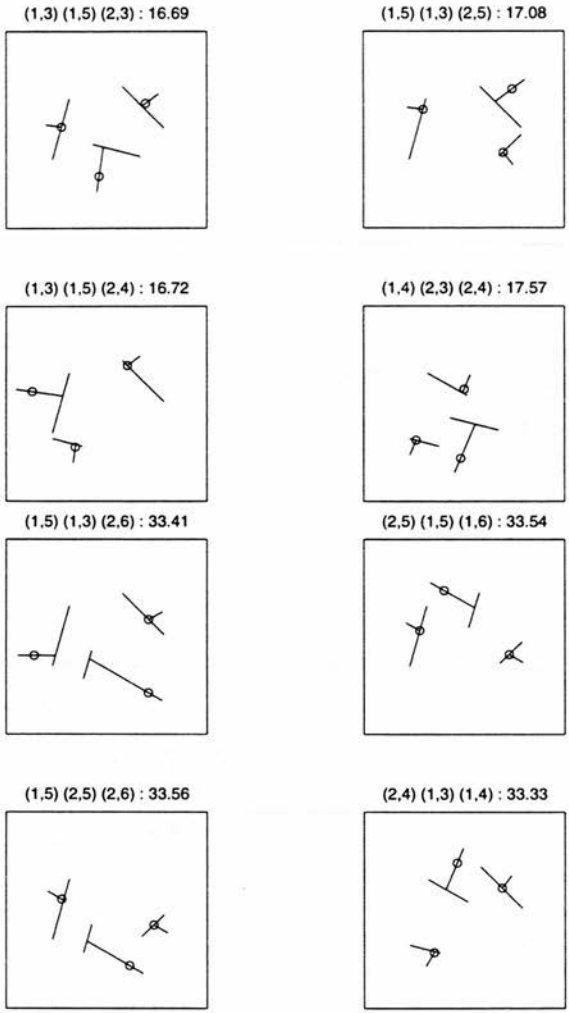


Figure 6.1: Required friction weighted aperture fits on the polyhedron for a lateral grasp.

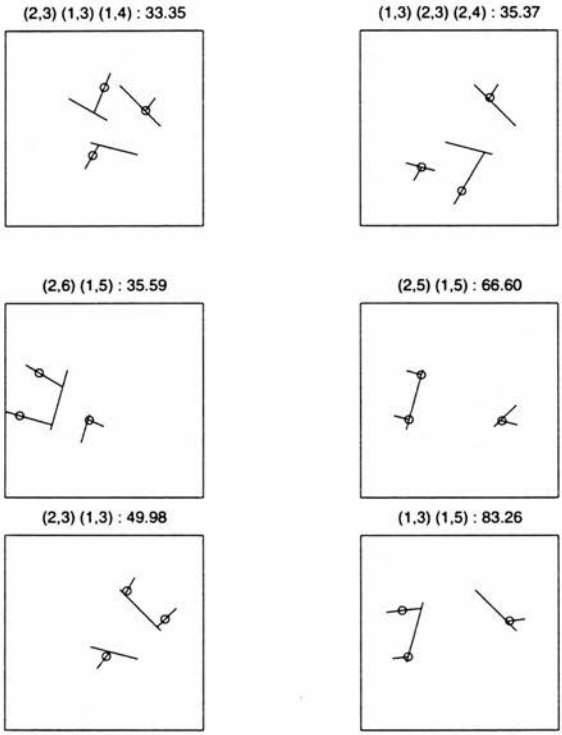


Figure 6.2: Required friction weighted aperture fits on the polyhedron for a lateral grasp (cont.)

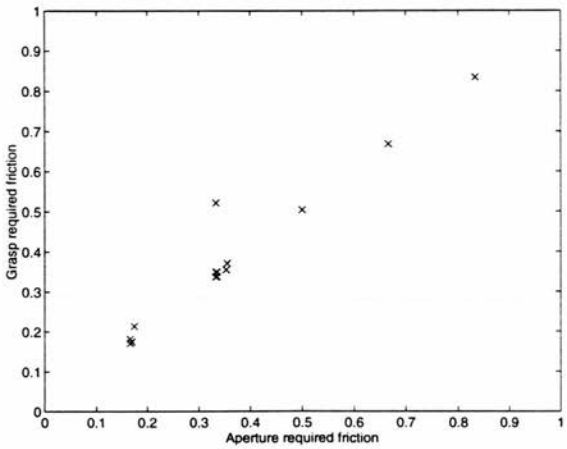


Figure 6.3: Required friction *grasp* metric vs. required friction *aperture* metric for a lateral-proximal grasp.

The effect of the distance to contact aperture metric on the required friction grasp metric can be seen more clearly for the lateral-distal grasp of the same object (which has the same set of aperture fits as the lateral-proximal grasp). Distal trajectories tend to deviate further from the aperture plane than do proximal trajectories and therefore show to a greater extent the effect of the distance to contact aperture metric on the required friction grasp metric. Figure 6.4 shows a graph of the required friction aperture metric against the required friction grasp metric for the lateral-distal grasp. Against each point is marked the distance to contact. It can be seen that many of the grasps deviate far enough from the aperture plane to give a very pronounced dependency of the required friction grasp metric on the distance to contact as well as the required friction aperture metric. This arises because the greater the distance to contact, the more the plane of the final contacts deviates from the aperture plane (which is horizontal). Since the contact forces lie in the plane of the contact points, this means that the contact forces also deviate from the horizontal. Since the polyhedron has vertical sides, this means that more friction is required as the distance to contact increases.

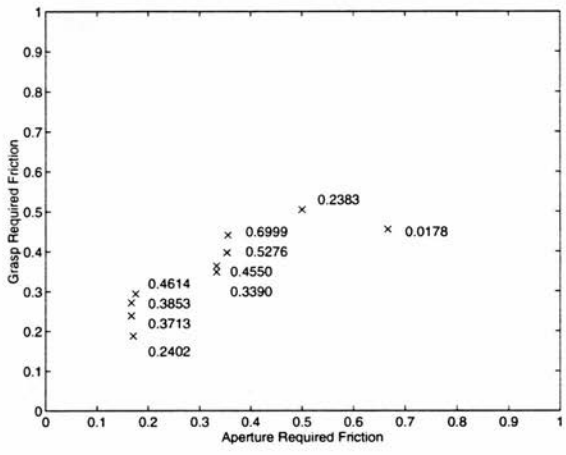


Figure 6.4: Required friction grasp metric vs. required friction aperture metric for a lateral-distal grasp. The distance to contact for each point is marked. Note that the axes are scaled slightly differently to emphasise the spread of points.

The distance to contact metric

To assess the effect of the distance to contact term of the aperture metric, the aperture metric coefficients are set as follows:

$$\mu_1 = 1000, \mu_2 = 1, \mu_3 = 1$$

This weights greatly the distance to contact term, but gives the other terms in the metric very small positive values in order to avoid degeneracy of aperture fits. Once again, Lateral-proximal grasps are planned on the polyhedron object. Figures 6.5– 6.6 show the ordered set of aperture fits for this set of coefficients. These should be compared to the aperture fits in Figures 6.1– 6.2. Figure 6.7 shows a graph of the distance to contact aperture metric against the contact angle grasp metric for the lateral-proximal grasp. It shows that the contact angle metric increases (very approximately linearly) the with the distance to contact aperture metric. Figure 6.8 shows the distance to contact aperture metric against the contact angle grasp metric for the lateral-distal. Here the variation is less linear, but the contact angle grasp

metric does tend to increase as the distance to contact increases. In each case, the outliers are due to the fact that the relationship between the distance to contact and the angle of the distal link depends on the range of digit configurations over the digit closure.

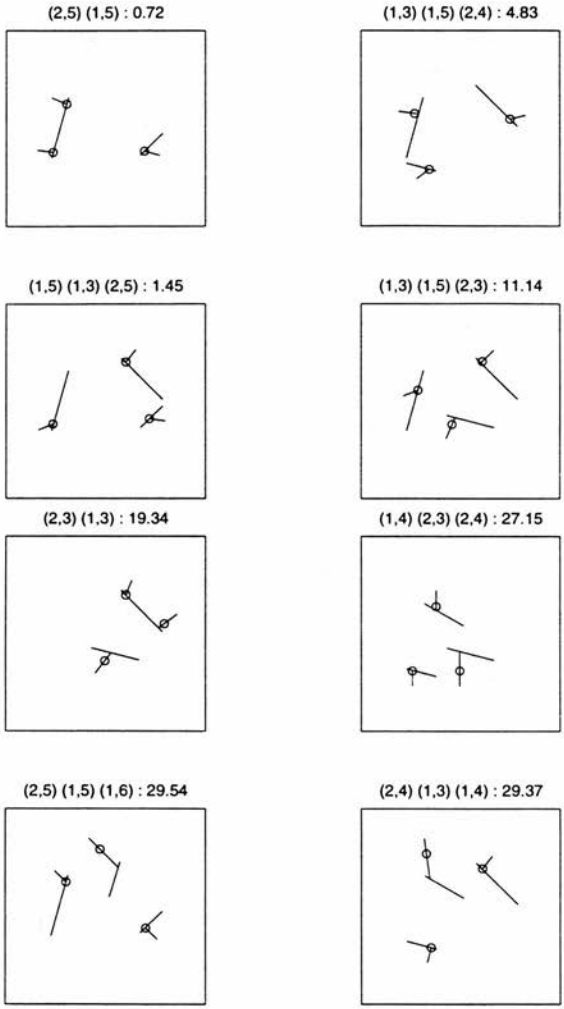


Figure 6.5: Distance to contact weighted aperture fits on the polyhedron for a lateral grasp.

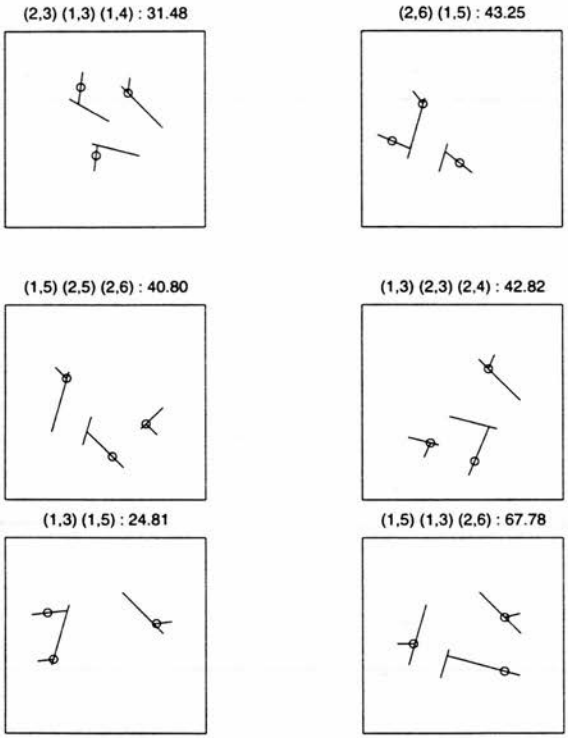


Figure 6.6: Distance to contact weighted aperture fits on the polyhedron for a lateral grasp (cont.)

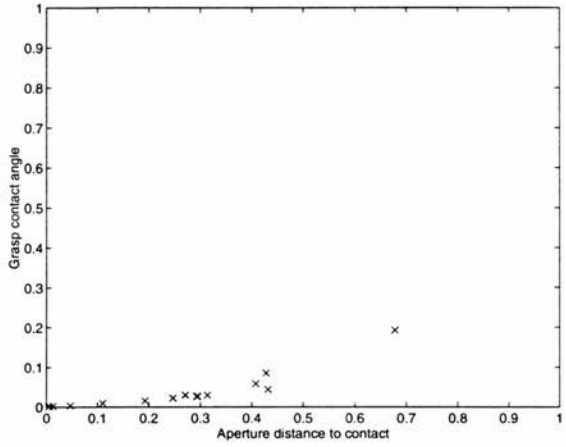


Figure 6.7: Contact angle grasp metric vs. distance to contact aperture metric for a lateral-proximal grasp

For these grasp parameters, there is no apparent correlation between the distance to contact metric and the required friction grasp metric, joint torque grasp or joint limit grasp metric. Figure 6.9 shows the variation of each with respect to the distance to contact metric. The required friction grasp metric is shown as crosses, the joint torque grasp metric is shown as circles and the joint limit grasp metric as asterisks.

Conclusions

It is not possible to draw more than general conclusions about the relationship between aperture and grasp metrics, due to the crucial influence of object shape. For most object shapes, an increase in the required friction aperture metric causes an increase in the required friction grasp metric. An increase in the distance to contact aperture metric, however, causes an increase in both the distance to contact grasp metric and the required friction grasp metric. The strong influence of the distance to contact aperture metric is due to the fact that it determines the validity of the assumptions made in the aperture fitting as well as correlating directly with the deviation of the final grasp from the preshape (the "ideal" grasp for the given task). The distance

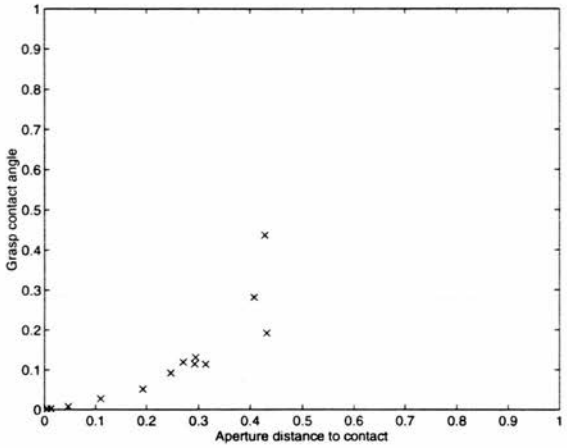


Figure 6.8: Contact angle grasp metric vs. distance to contact aperture metric for a lateral-distal grasp.

to contact aperture metric is given greater weight than the required friction aperture metric.

6.2 Further Results

In Chapter 5 a lateral grasp was shown planned on a polyhedron. The best grasp found was a lateral-proximal grasp which used three different planar patches on the surface of the polyhedron. Figure 6.10 shows two other grasps: a tip-distal grasp and a manipulation-proximal grasp, along with the lateral-proximal grasp planned in Chapter 5. Each one uses the same set of contact surfaces, but with differing hand configurations and contact points. In each case, the digit trajectory that gave better results (out of proximal or distal) was chosen. In order to make the figures more clear, the stick model of the hand is used (this also has the merit that the grasp metric calculation uses the stick model of the hand). The boundaries of the three contact planes are drawn and the contact normals are shown as dotted lines. The lateral grasp, with its extended digits, is very distinctive. The precision grasp can be distinguished from the manipulation grasp by the angle that the distal links make with the object

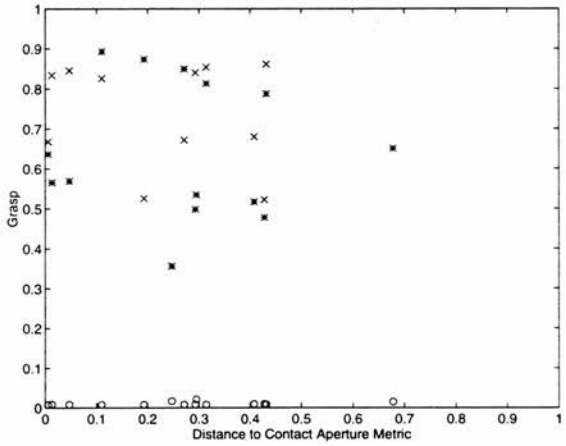


Figure 6.9: Required friction grasp metric ('x'), joint torque grasp metric ('o'), joint limit grasp metric ('*') vs. distance to contact aperture metric.

surface. Table 6.2 shows the terms of the grasp metric for each grasp. It can be seen that the best grasp is the manipulation grasp. This may be because it places least constraint on what the hand should be, since the contact angle need not be specified. The worst grasp is the lateral grasp, probably because of the limited range of lateral preshapes that can be adopted: this often leads to final grasps that deviate from the preshape significantly.

In order to provide an example of an object on which a very high quality grasp can be planned, Figure 6.11 shows lateral-proximal, tip-proximal and manipulation-proximal grasps on the cuboid for which a model was built in Chapter 4. There is little difference between the tip and manipulation grasps because of the shape and width of the object (compare this with Figure 6.10 where there is a pronounced difference between tip and manipulation grasps). Table 6.3 shows the terms of the grasp metric for each grasp. In each case, the proximal digit trajectory gave better results than the distal digit trajectory.

While the polyhedron is complex in terms of the number and location of (planar) surface patches, the success of the grasp planning algorithm will largely be determined

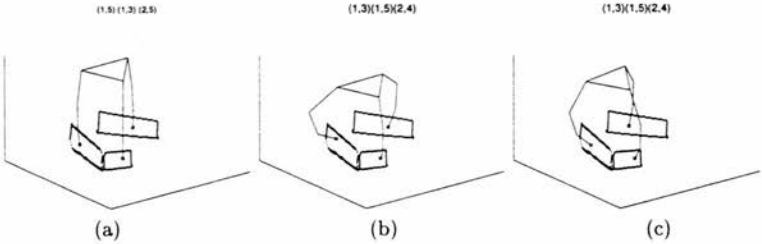


Figure 6.10: Planned grasps on the polyhedron: (a) lateral-proximal, (b) tip-distal, (c) manipulation-proximal. The boundaries of the three contact planes are drawn and the contact normals are shown as dotted lines.

	Lateral-Proximal	Tip-Distal	Manipulation-Proximal
Equilibrium	0	0	0
Required friction	0.5983	0.5159	0.3352
Task wrench	0	0	0
Joint torque	0.0088	0.0135	0.0113
Joint limit	0.4508	0.1075	0.0060
Contact angle	0.0101	0.0605	0
Collision	0	0	0
TOTAL	0.3799	0.2882	0.1949

Table 6.2: The terms of the grasp metric for lateral, tip and manipulation grasps of the polyhedron.

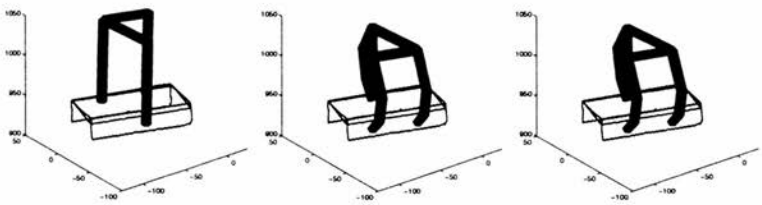


Figure 6.11: Planned grasps on the cuboid: (a) lateral-proximal, (b) tip-proximal, (c) manipulation-proximal. The boundaries of the three contact planes are drawn.

	Lateral-Proximal	Tip-Proximal	Manipulation-Proximal
Equilibrium	0	0	0
Required friction	0.1293	0.1446	0.1441
Task wrench	0	0	0
Joint torque	0.0077	0.0091	0.0092
Joint limit	0.7094	0.0054	0.0041
Contact angle	0.0295	0.0160	0
Collision	0	0	0
TOTAL	0.1093	0.0781	0.085

Table 6.3: The terms of the grasp metric for lateral, tip and manipulation grasps of the cuboid.

by whether it can plan good grasps on complex curved objects as it is here that the existing preshaping techniques are inapplicable. In this section, results are presented and analysed for three curved objects. Grasps are planned on an egg, which should yield a high quality grasp. They are then planned on a tube, which has a complex surface that is very far from the ellipsoidal shape of the egg, and a toy rabbit which is a good test of the algorithm because it has many quadric surface patches. To segment the images, the *Rangeseg* parameters shown in Table 6.4 were used: see Chapter 4 for an explanation of what these mean. It can be seen that the parameters were changed from object to object. This was done in order to tune the segmentation to give the "correct" set of surface patches: generally the set that a human would describe the object with. The process is sensitive to the segmentation, because over-segmentation produces too many candidate grasping sets and under-segmentation produces poor contact geometry descriptions. The sensitivity of the segmentation process is a key problem in the algorithm (see Section 7.2 for a discussion of this).

The same grasp parameters were used for all the objects, however: these are shown in Table 6.5. Some will necessarily be the same, since they describe task requirements. In the case of other parameters, such as sampling rates, the same parameters can be kept for all objects,due largely to the fairly constant imaging characteristics of the laser striper. This demonstrates the robustness of the grasp planning portion of the system (and one advantage of using laser striper range data). Table 6.1 shows the coefficients of the grasp metric that were used.

	Depth dis.	Orient. dis.	Num of smooth.	Morph. Schedule	H low thresh.	H high thresh.	Max Residual
Polyhedron	10	25	4	+-++	0.01	10	1
Egg	5	25	1	-++	0.01	0.03	1
Tube	5	25	1	-++	0.01	0.05	2
Toy Rabbit	5	25	4	+-	0.05	0.1	1.3

Table 6.4: Segmentation parameters for each object.

Miscellaneous	Opposition Angle	60°
	Ground Plane Threshold	70%
	Hand radius	5mm
Aperture slice	Aperture slice width	2mm
	Aperture slice alpha	0.1
Aperture fit	Aperture fit alpha	0.1
	Distance to Contact coefficient	20
	Required Friction metric	10
	Distance to Centre metric	1
Coarse aperture fit	Thumb contact position	8 per feature
	Rotation sampling	15°
	Abduction sampling	15°
	Thumb dist. to contact sampling	4mm
Final aperture fit	Thumb contact position	16 per feature
	Rotation sampling	7.5°
	Abduction sampling	7.5°
	Thumb dist. to contact sampling	2mm

Table 6.5: Parameters used in the experiments through this chapter, unless specified otherwise

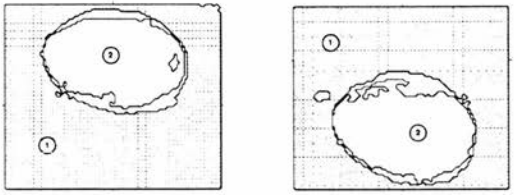
6.2.1 Egg

Grasps were planned on the egg object first encountered in Chapter 4. It is approximately 6cm long and 4cm wide. Two views are taken of the egg, and are shown in Figure 6.12. The views are taken at $\pm 45°$ to the horizontal, rotated about x axis. Figure 6.13 shows the segmentation: in each view, the ground plane is segmented as a plane and the egg itself as a quadric surface patch.

Figure 6.14 shows the resulting surface model and voxmap. The egg itself consists of two quadric surface patches which overlap slightly at the top, where a slight error in registration can be seen. The ground planes were successfully segmented away automatically by imposing the condition that if 70% of a horizontal slice of the voxmap

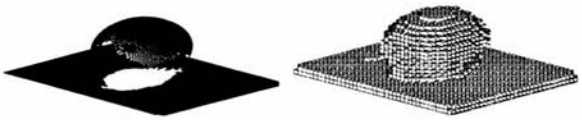
(a) (b)

Figure 6.12: Range images of the egg object at (a) 135°, (b) 225°.



(a) (b)

Figure 6.13: Segmentation of the egg object at (a) 135° (b) 225°



(a) (b)

Figure 6.14: Acquired model of the egg: (a) surface model, (b) voxmap

is filled, it corresponds to the ground, as described in Chapter 4. The ground plane consists of 8535 points and the egg itself of 2970 points. The voxmap has dimensions $47 \times 54 \times 32$ with 20481 filled voxels out of 81216. Figure 6.15 shows the four surfaces fitted to the egg scene. Each patch is labeled (i, j): i is the view, j is the patch number within that view.

There are two candidate grasping sets for the egg. Each one consists of the two quadric surfaces, with the thumb and finger contacts swapping around to form the two sets. Figure 6.16 shows the selected slice for each candidate grasping set (it is the same for each one). The aperture fitting and the remainder of the algorithm is task-specific.

Tip Grasp

Figure 6.17 shows the ranking score of each candidate grasping set (which can range from 0 to 1) multiplied by 100. The coarsely fitted aperture used to estimate this is displayed. There is very little difference in scores, due to the symmetry of the egg. Both scores are very low, because the shape of the egg is such that it is possible to fit an aperture (and hence a preshape) very closely to the object.

Figure 6.18 then shows the final aperture fitted to the higher ranked of the two grasping sets. The refined ranking score is displayed above it. As always, the coarse aperture fitting produces a higher ranking score: 9.85 compared with 9.82. The individual terms of this ranking metric are displayed in Table 6.6.

Figure 6.19(a) shows the grasp planned for the proximal digit trajectory and Figure 6.19(b) shows the grasp planned for the distal digit trajectory. The contact normals are shown as dotted lines. The individual terms of the grasp metric are displayed

```
Plane (1,1), Nx+D=0
4361 points
N = [ -0.00113283 0.0116292 0.999932 ]
D = [ -944.282 ]

Quadric (1,2), x'Ax+Bx+C=0
1443 points
A = [ 0.00278803  -0.00103925  -0.000652765
     -0.00103925   0.00544794   0.000495653
     -0.000652765  0.000495653  0.00671282 ]
B = [ 1.26074      -0.953699     -12.9662 ]
C = [ 6259.37 ]

Plane (2,1), Nx+D=0
4174 points
N = [ 0.00381511 0.0188272 0.999815 ]
D = [ -943.837 ]

Quadric (2,2), x'Ax+Bx+C=0
1527 points
A = [ 0.00259327  -0.000521809  -0.000204482
     -0.000521809  0.00500602   -0.000172866
     -0.000204482  -0.000172866  0.00615557 ]
B = [ 0.384064 0.403621 -11.8665 ]
C = [ 5717.1 ]
```

Figure 6.15: Surfaces fitted to the egg scene. Each patch is labeled (i, j): i is the view, j is the patch number within that view.

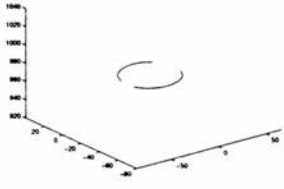


Figure 6.16: The chosen slice through each candidate grasping set

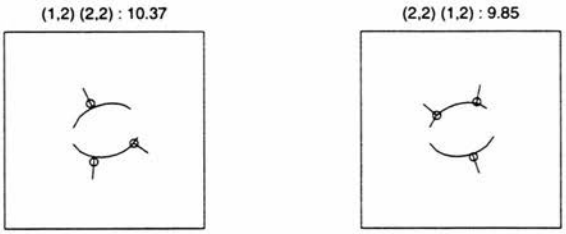


Figure 6.17: A tip aperture coarsely fitted to each candidate grasping set, with the ranking score (multiplied by 100) displayed above each one.

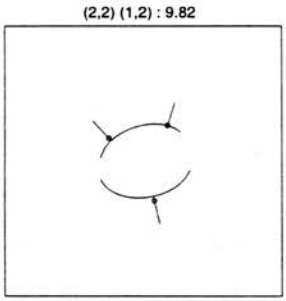


Figure 6.18: Tip aperture fitted to top-ranked candidate grasping set of the egg, with the refined ranking score displayed above it.

	Tip	Lateral
Aperture Plane	0.0992566	0.0992566
Distance to contact	0.0815746	0.331882
Required friction	0.0549904	0.0322334
Distance to centre	0.527553	0.599525

Table 6.6: The terms of the aperture metric for tip and lateral grasps on the egg.

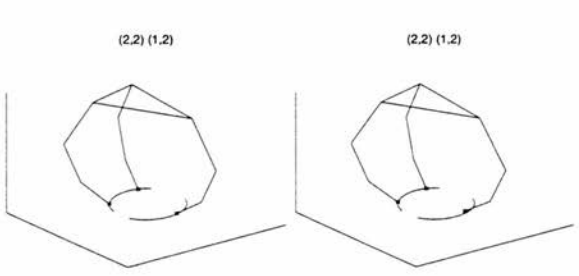


Figure 6.19: Grasps planned for (a) proximal trajectory, (b) distal trajectory. The contact normals are shown as dotted lines.

in Table 6.7. Due to the fact that the aperture fits very closely to the object, there is a very small amount of digit closure and therefore very little difference between the two grasps. It can be seen that the distal trajectory is better for all terms except for the contact angle term. Proximal trajectories often yield better contact angle metrics than distal trajectories, because the distal link rotates less during a proximal digit closure, and it is a combination of distal link rotation from the preshape and object shape that determines the contact angle metric. In both cases, the grasp metric is very low, indicating a high quality grasp. The worst aspect of these grasps is the required friction metric: this is because some of the points in the aperture plane slice have far-from-horizontal normals. The slice itself had a high score, due to the effect of these normals being averaged with many near-horizontal normals, but since only the projection of the contact normal in the horizontal plane is considered when fitting the aperture, the aperture may be fitted such that the grasp contacts have those far from horizontal normals. It is clear that this has happened when the tip-proximal grasp is looked at from the side — see Figure 6.20. This type of problem also arises when the distance of the final grasp contacts from the aperture plane are large enough that the contact normals are significantly different from those in the aperture plane. In the case of the egg, however, the distance to contact is very small and the problem arises solely due to far-from-horizontal normals in the aperture slice.

	Tip-Proximal	Tip-Distal
Equilibrium	0	0
Required friction	0.2519	0.1939
Task wrench	0	0
Joint torque	0.0104	0.0103
Joint limit	0.1055	0.0927
Contact angle	0.0235	0.0554
Collision	0	0
TOTAL	0.1463	0.1237

Table 6.7: The terms of the tip grasp metric for proximal and distal digit trajectories

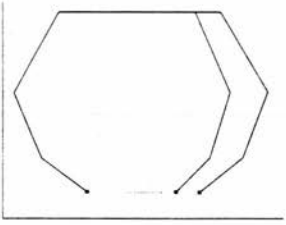


Figure 6.20: Grasp planned for proximal trajectory, viewed side-on. It can be seen that one of the contact normals lies out of the plane of contacts: this means that the required friction metric is high to compensate for this.

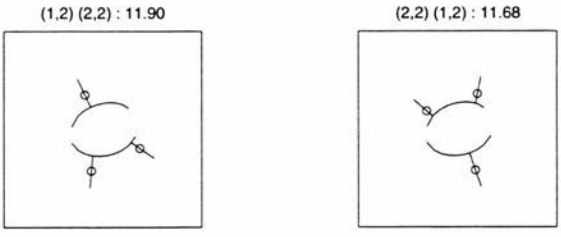


Figure 6.21: A lateral aperture coarsely fitted to each candidate grasping set of the egg, with the ranking score (multiplied by 100) displayed above each one.

Lateral Grasp

Figure 6.21 shows the ranking score of each candidate grasping set (which can range from 0 to 1) multiplied by 100. The coarsely fitted aperture used to estimate this is displayed. The apertures for a lateral grasp are slightly different to those for a precision grasp, due to different bounds on the digit configurations. Again, there is very little difference in scores, due to the symmetry of the egg. Figure 6.22 shows the aperture fitted to the higher ranked of the two grasping sets. The refined ranking score is displayed above it. The individual terms of this ranking metric are displayed in Table 6.6. It is clear from the distance to contact metric and the aperture diagram that the aperture has not been fitted flush to the object. This is because the object width is less than the minimum width for a lateral preshape (the minimum width of a lateral preshape is the width of the palm — 60cm). This also leads to a worsening of the required friction metric because the grasp points are further away from the aperture plane (and their normals are therefore less horizontal). This illustrates a problem with the distance to contact metric. For the egg, the distances to contact for each finger are roughly coupled. The fingers are therefore pulled much closer to the object surface than the thumb, which produces an unbalanced aperture fit.

Figure 6.23(a) shows the grasp planned for the proximal digit trajectory and Figure 6.23(b) shows the grasp planned for the distal digit trajectory. Note that the thumb contact lies above the aperture plane of the egg shown in the figure but still contacts the egg surface (the egg surface being omitted so that the grasp configuration

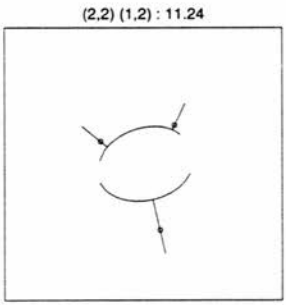


Figure 6.22: Lateral aperture fitted to top-ranked candidate grasping set of the egg, with the refined ranking score displayed above it.

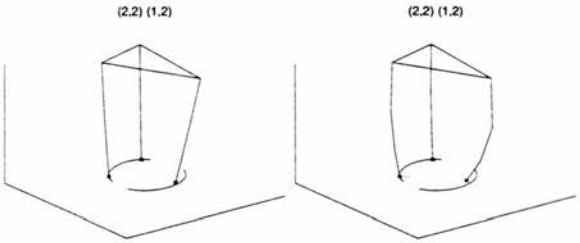


Figure 6.23: Lateral grasps planned on the egg for (a) proximal trajectory, (b) distal trajectory. Note that the thumb contact lies above the aperture plane of the egg shown in the figure but still contacts the egg surface (the egg surface is omitted so that the grasp configuration is clearly visible).

is clearly visible). The individual terms of the grasp metric are displayed in Table 6.8. It can be seen that again the distal trajectory is better for all terms except in the case of the contact angle term. Here, however, there is a bigger relative difference in the contact angle metrics so the distal trajectory grasp has a lower score than the proximal trajectory grasp.

6.2.2 Tube

The tube is a tube of hand-wash liquid, similar to a toothpaste tube, and is shown in Figure 6.24. It is interesting because of its shape: it is a curved object, tapered at

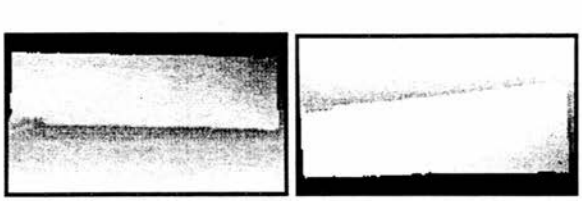
	Lateral-Proximal	Lateral-Distal
Equilibrium	0	0
Required friction	0.2713	0.2142
Task wrench	0	0
Joint torque	0.0097	0.0097
Joint limit	1.0876	0.8222
Contact angle	0.0714	0.2575
Collision	0	0
TOTAL	0.2196	0.2334

Table 6.8: The terms of the grasp metric for lateral-proximal and lateral-distal grasps on the egg.

Figure 6.24: The tube object.

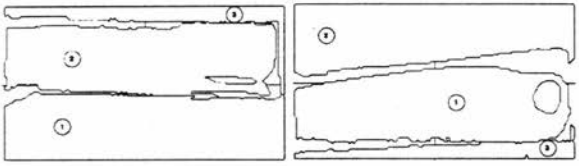
one end. It is therefore not so suited to the planar/quadric segmentation that fits the egg and polyhedron so well. The range images of the object are shown in Figure 6.25. As for the egg, they are taken at $\pm 45°$ to the horizontal, rotated about the x axis. Figure 6.26 shows the segmentation: in each view, the ground is segmented into a plane and a quadric, and the tube itself as a quadric.

Figure 6.27(a) shows part of the resulting surface model: for clarity the four ground patches have been omitted. Figure 6.27(b) shows the voxmap as a cloud of points, with the lower part of the voxmap (corresponding to ground) removed for clarity. The ground patches consists of 58009 points and the tube itself of 54402 points. The voxmap has dimensions $66 \times 57 \times 32$. Figure 6.28 shows the six surfaces fitted to the tube scene. Each patch is labeled (i, j): i is the view, j is the patch number within that view. Note



(a) (b)

Figure 6.25: Range images of the tube object at (a) 135°, (b) 225°.

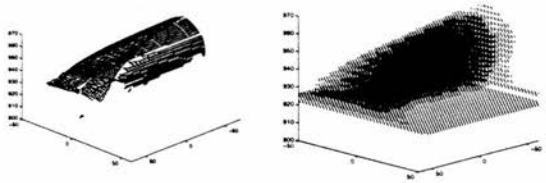


(a) (b)

Figure 6.26: Segmentation of the tube object at (a) 135° (b) 225°

that two of the ground patches are planes and two of the ground patches are quadrics, due to slight errors in the segmentation where some of the object was joined to the ground patches, thus distorting their apparent shape from planar to quadric.

There are four candidate grasping sets for the tube. There should only be two. The error arises because there is no explicit labelling of patches as being ground or object. *Voxels* are labelled as ground or object (this is necessary to find the centre of mass of



(a) (b)

Figure 6.27: Acquired model of the tube: (a) surface model, (b) voxmap

```
Plane (1,1), Nx+D=0
4856 points
N = [ -0.00607397 0.00461543 0.999971 ]
D = [-925.04 ]

Quadric (1,2), x'Ax+Bx+C=0
4728 points
A = [ 8.38899e-05  -0.0014241   -9.07394e-05
     -0.0014241     0.000905156 -0.00442143
     -9.07394e-05  -0.00442143   0.0059126  ]
B = [ 0.238224 8.48664 -10.978 ]
C = [ 5090.3 ]

Plane (1,3), Nx+D=0
1112 points
N = [ -0.011471 0.0123836 0.999858 ]
D = [ -924.097 ]

Quadric (2,1), x'Ax+Bx+C=0
4466 points
A = [ 0.000304499  0.000683178  0.00121434
      0.000683178  0.000620674  0.00123205
      0.00121434   0.00123205   0.0124928    ]
B = [ -2.27207 -2.40805 -23.5878]
C = [ 11131.8 ]

Plane (2,2), Nx+D=0
4339 points
N = [ -0.00169766 -0.0125923 0.999919 ]
D = [ -924.96 ]

Quadric (2,3), x'Ax+Bx+C=0
965 points
A = [ 1.17552e-05  0.000191008  0.0107987
      0.000191008  0.00954831   0.140547
      0.0107987    0.140547    -0.195615     ]
B = [ -19.9892 -260.9 347.395 ]
C = [ -153946 ]
```

Figure 6.28: Surfaces fitted to the tube scene. Each patch is labeled (i, j): i is the view, j is the patch number within that view.

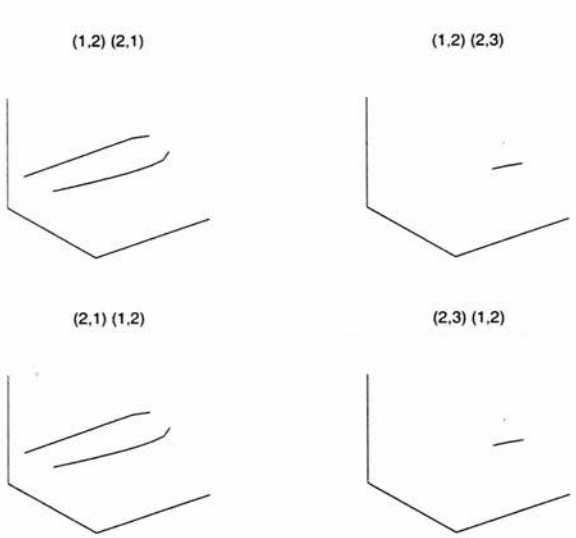


Figure 6.29: The chosen slice through each candidate grasping set

the object) but the system relies on the fact that because ground patches will tend to be nearly horizontal and nearly planar, they will never be grouped into a candidate grasping set. However, as stated above, there is an error in the segmentation that means that two ground patches include a small part of the object as well. This is why they are quadric when the ground itself is planar. It also means that they can form candidate grasping sets with patches from the tube itself and one of the ground patches does indeed do this.

Figure 6.29 shows the selected slices for each of the candidate grasping sets. There are four candidate grasping sets. Two of them are correct, in that they consist of slices through the object. Two are incorrect, having arisen from part of the ground plane opposing part of the object, as described above. Fortunately the overlap between ground plane and object is so small that these slices are very short. This will mean that in subsequent processing, it will be impossible to fit an aperture to them and the bogus candidate grasping sets will be discarded.

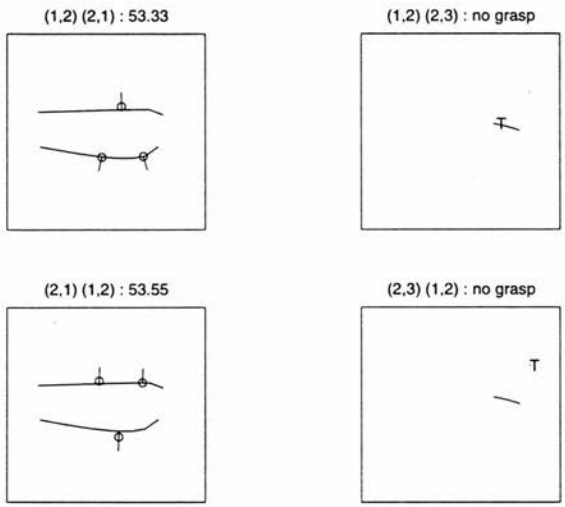


Figure 6.30: A tip aperture coarsely fitted to each candidate grasping set, with the ranking score (multiplied by 100) displayed above each one.

Tip Grasp

Figure 6.30 shows the ranking score of each candidate grasping set (which can range from 0 to 1) multiplied by 100. The coarsely fitted aperture used to estimate this is displayed.

Figure 6.31 then shows the aperture fitted to the higher ranked of the two grasping sets. The refined ranking score is displayed above it. The individual terms of this ranking metric are displayed in Table 6.9.

	Tip	Lateral
Aperture Plane	0.0575564	0.575564
Distance to contact	0.0260141	0.159167
Required friction	0.0561299	0.0468348
Distance to centre	0.335067	0.387052

Table 6.9: The terms of the aperture metric for tip and lateral grasps on the tube.

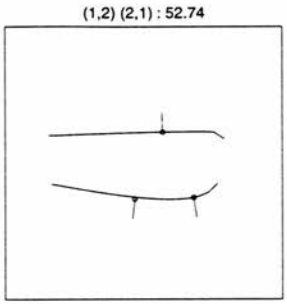


Figure 6.31: Tip aperture fitted to top-ranked candidate grasping set, with the refined ranking score displayed above it.

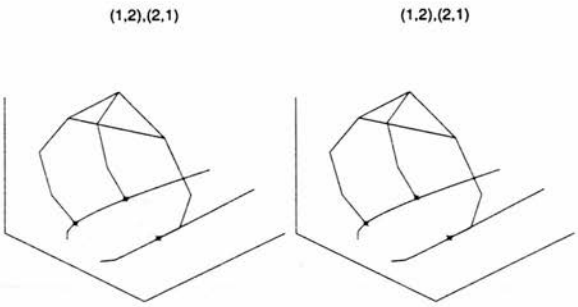


Figure 6.32: Tip grasps planned for (a) proximal trajectory, (b) distal trajectory. The contact normals are shown as dotted lines.

Figure 6.32(a) shows the grasp planned for the proximal digit trajectory and Figure 6.32(b) shows the grasp planned for the distal digit trajectory. The contact normals are shown as dotted lines. The individual terms of the grasp metric are displayed in Table 6.10.

Lateral Grasp

Figure 6.33 shows the ranking score of each candidate grasping set (which can range from 0 to 1) multiplied by 100. The coarsely fitted aperture used to estimate this is displayed.

	Tip-Proximal	Tip-Distal
Equilibrium	0	0
Required friction	0.8686	0.8653
Task wrench	0	0
Joint torque	0.0092	0.0092
Joint limit	0.0656	0.0627
Contact angle	0.3561	0.3091
Collision	0	0
TOTAL	0.5327	0.5189

Table 6.10: The terms of the grasp metric for tip-proximal and tip-distal grasps on the tube.

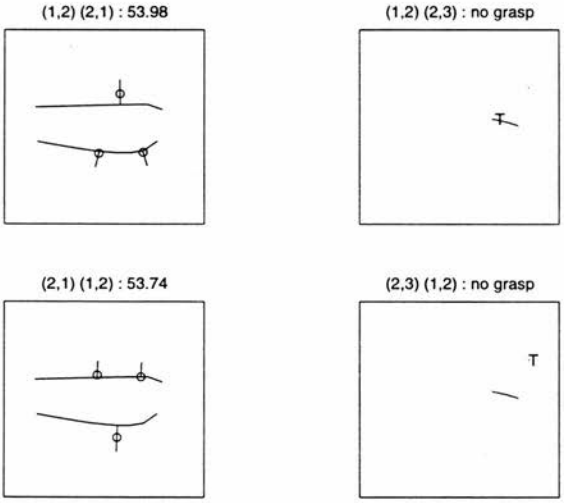


Figure 6.33: A lateral aperture coarsely fitted to each candidate grasping set, with the ranking score (multiplied by 100) displayed above each one.

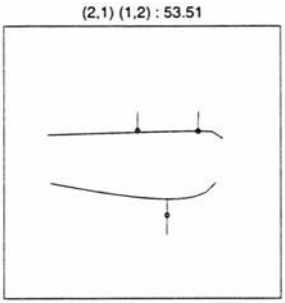


Figure 6.34: Lateral aperture fitted to top-ranked candidate grasping set, with the refined ranking score displayed above it.

Figure 6.34 then shows the aperture fitted to the higher ranked of the two grasping sets. As with the egg in Section 6.2.1, the lateral aperture cannot fit snug to the object surface because of the limited range of possible lateral apertures. The refined ranking score is displayed above it. The individual terms of this ranking metric are displayed in Table 6.9.

Figure 6.35(a) shows the grasp planned for the proximal digit trajectory and Figure 6.35(b) shows the grasp planned for the distal digit trajectory. The contact normals are shown as dotted lines. The individual terms of the grasp metric are displayed in Table 6.11. Note that the joint limit metric is very poor, especially for the proximal grasp. Fortunately this is not a very important metric for the lateral grasp because, as described in Section 6.1.1, the lateral grasp is rarely used for manipulation between the digits and can therefore tolerate joint values close to their limits. Note also that the lateral-proximal grasp has a much better contact angle metric because the proximal trajectory alters the orientation of the distal link less than the distal trajectory.

6.2.3 Toy rabbit

The "Toy Rabbit" is a children's toy rabbit, lying on its back. This is perhaps the most challenging object for the algorithm consisting as it does of 8 quadric surfaces at various orientations. It highlights some problems with the grasp planning algorithm,

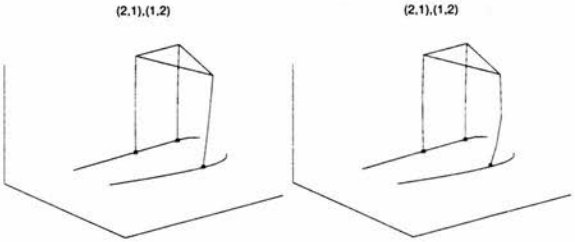


Figure 6.35: Grasps planned for (a) proximal trajectory, (b) distal trajectory. The contact normals are shown as dotted lines.

	Lateral-Proximal	Lateral-Distal
Equilibrium	0	0
Required friction	0.7223	0.7026
Task wrench	0	0
Joint torque	0.0077	0.0079
Joint limit	0.9997	0.8730
Contact angle	0.2350	0.4311
Collision	0	0
TOTAL	0.5397	0.5834

Table 6.11: The terms of the grasp metric for lateral-proximal and lateral-distal grasps on the tube.

(View, Patch)	Surface type
(1,1)	plane
(1,2)	quadric
(1,3)	quadric
(1,4)	quadric
(1,5)	quadric
(1,6)	quadric
(2,1)	plane
(2,2)	quadric
(2,3)	quadric
(2,4)	quadric
(2,5)	quadric
(2,6)	quadric

Table 6.12: Type of fit for each patch in each view.

but a reasonable grasp is still achieved. Figure 6.36(a) shows a picture of the toy rabbit. (b) and (c) are the range images, taken from $\pm 35°$ to the horizontal about the x axis. (d) and (e) show the segmentation of each range image. (f) shows the surface model and (g) the voxmap (which has dimensions $67 \times 63 \times 45$).

Table 6.12 shows the type of surface that is fitted to each patch. Note the large number of quadrics. This produces over 400 candidate grasping sets, because the procedure that identifies candidate grasping sets always accepts pairs or triples of quadrics, regardless of their relative shape and orientation. All but 7 of these are rejected by the coarse aperture fitting, but the grasp planning is slowed substantially.

Figure 6.37 shows the result for the top ranked candidate grasping set, which is a lateral-proximal grasp. (a) shows the preshape aperture. (b) and (c) show the grasp from two different viewpoints: the dotted lines are the contact normals and the dashed lines the directions of the grasp forces. The object surface is omitted for clarity. (d) shows the grasp from another viewpoint, with finite width fingers and a surface mesh of the object. The ground is omitted for clarity.

Table 6.13 shows the terms of the aperture metric and Table 6.14 the resulting grasp metric. The proximal trajectory gives the best final grasp metric.

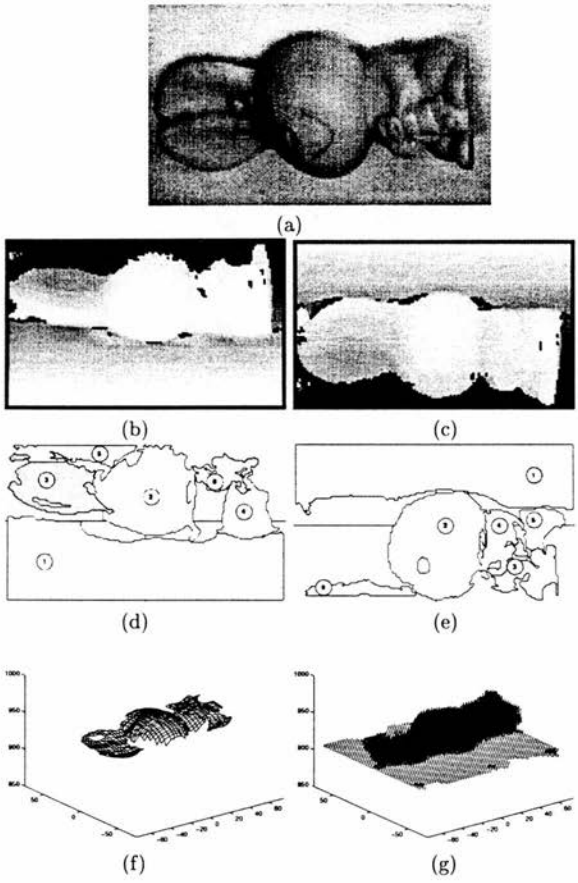


Figure 6.36: (a) picture of the toy rabbit, (b) range images at $+35°$, (c) range image at $-35°$, (d) segmentation of view #1, (e) segmentation of view #2, (f) surface model, (g) voxmap.

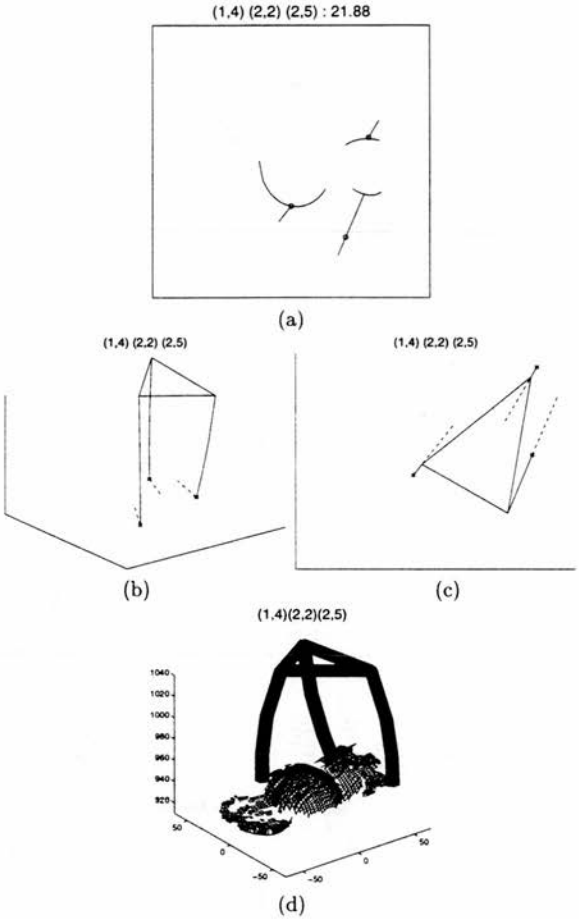


Figure 6.37: Lateral-proximal grasp on the toy rabbit. (a) Preshape aperture, (b) Grasp: the dotted lines are the contact normals and the dashed lines the directions of the grasp forces. The object surface is omitted for clarity. (c) Grasp shown from above. (d) Grasp shown from a third viewpoint, with finite width fingers and a surface mesh of the object. The ground is omitted for clarity.

	Lateral
Distance to contact	0.429868
Required friction	0.666677
Distance to centre	0.172994
Aperture Plane	0.190922
Total	0.218836

Table 6.13: The terms of the aperture metric for a lateral grasp on the toy rabbit.

	Lateral-Proximal
Equilibrium	0
Required friction	0.7070
Task wrench	0
Joint torque	0.0082
Joint limit	0.9437
Contact angle	0.2892
Collision	0
TOTAL	0.5452

Table 6.14: The terms of the lateral-proximal grasp metric for the toy rabbit.

6.3 Summary and Conclusions

Tip, lateral and manipulation grasps have been planned on a variety of objects. Table 6.15 shows the grasp score for each grasp planned. In each case, the digit trajectory that gives the best grasp metric is chosen.

It can be seen that in the above results, tip grasps tend to work better with distal trajectories and lateral grasps tend to work better with proximal trajectories (not enough results have been presented to determine whether the lateral grasp has a trend towards proximal or distal trajectories). The exception to this was the cuboid, which planned a tip-proximal grasp. However in this case, there was a very small difference between the results gained with proximal and distal trajectories. Generally, there is little difference between grasps that use the proximal and distal digit trajectories. This is because the aperture metric tries to minimise the distance to contact, and the smaller the distance to contact, the smaller the effect of the digit trajectory on the final result. For this reason, it would be reasonable to always group lateral grasps with proximal

Object	Grasp	Digit trajectory	Score
Cuboid	lateral	proximal	0.1093
Cuboid	tip	proximal	0.0781
Cuboid	manipulation	proximal	0.085
Polyhedron	tip	distal	0.2822
Polyhedron	lateral	proximal	0.3799
Polyhedron	manipulation	proximal	0.1949
Egg	tip	distal	0.1237
Egg	lateral	proximal	0.2196
Tube	tip	distal	0.5189
Tube	lateral	proximal	0.5397
Toy rabbit	lateral	proximal	0.5452

Table 6.15: Grasp metrics for each grasp planned.

digit trajectories, and tip grasps with distal digit trajectories.

The polyhedron and egg have better quality grasps than the tube and the toy rabbit. This because the tube and toy rabbit are complex curved objects, while the polyhedron has no curved surfaces and the egg has just two ellipsoidal surfaces and strong symmetry. The reason for the difference in performance is simply that the egg and polyhedron match the ideal contact configurations of the grasp families more closely (and also match the assumptions made by the grasp planner itself more closely)

Chapter 7

Conclusions

In this chapter a summary of the work done is presented. The success of the approach is then evaluated, followed by a discussion of improvements and extensions to the work.

7.1 Summary

As was explained in Chapter 1, dextrous hands are useful for planning grasps. More stable grasps can be synthesised than with a parallel jaw gripper because of the increased number of contacts. The redundancy of the hand means that it is possible to integrate arm and hand motion planning more easily than with a parallel jaw gripper where wrist position is crucial to the success of the grasp. While the dimension of the solution space is much bigger than for a parallel jaw gripper, the quality of possible grasps is such that sub-optimal but fast grasp planners may be built.

In Chapter 2 the research literature of dextrous hand grasping was reviewed. It was seen that few complete grasp planners have been constructed, and the key problem of synthesising a grasp that is stable *and* kinematically desirable was largely unresolved. The paradigm of preshaping was noted as a good way of bounding the search for a good grasp, but a lack of methods of preshape specification had prevented preshapes from being used on arbitrarily shaped objects and from taking into account all but the simplest task requirements. Also noted was a lack of object models specifically constructed for robot grasp planning, particularly precision grasp planning.

Hand models were constructed for four different hands: a parallel jaw gripper, the

Salisbury hand, a simplified anthropomorphic model with 3 digits (known as the SAM) and a human hand. Each link on the hand was modelled as a cylinder and the fingertips were modelled as hemispheres in order to make for fast collision checking. The concept of a grasp family was introduced in Chapter 3. A grasp family is constructed according to the grasping task, such as whether a tip, lateral or manipulation grasp is desired. It associates a set of preshapes with prescribed digit trajectories and a set of ideal contact configurations. The preshape embodies the task requirements, and each preshape is an optimal grasp for one of the ideal contact configurations. For a non-ideal contact configuration, *i.e.* most real objects, the grasp family is fitted to the object so as to minimise the deviation of the final grasp from the preshape. Grasp families were constructed for the SAM hand model, for tip, lateral and manipulation grasps, with a choice of two digit trajectories.

In Chapter 4 was described how to construct an object model specifically for planning precision grasps. Precision grasps required accurate geometric information, for which a surface model consisting of planes and quadrics was constructed. The laser striper provides the dense range data necessary for building a sufficiently accurate model. A voxmap model is constructed to provide centre of mass information and fast collision checking against the cylindrical-link models of the hand described in Chapter 3.

Chapter 5 described a grasp planning algorithm that uses grasp families to constrain sensibly the number of degrees of freedom (DOF) of the grasp planning problem and embody the task requirements. In order to drive the grasp planning algorithm a grasp metric is formulated. This grasp metric integrates both mechanical and kinematic constraints, and it is shown that the grasp families derived in Chapter 3 optimise this metric. The grasp planning algorithm fits an approximation to the preshape, known as the preshape aperture, using an approximation to the digit trajectories which enable a planar analysis of the problem. The full grasp family is then fitted to further optimise the solution.

The algorithm was demonstrated to work for a lateral grasp on a polyhedral object in Chapter 5. In Chapter 6, tip and lateral grasps were planned on three different curved objects ranging from simple — the egg object — to much more complex in terms of surface geometry (in the case of the tube) and in terms of the number of surface patches

(in the case of the toy rabbit). The relationship between the quality of the aperture fit and the final grasp was examined in detail.

7.2 Conclusions

The algorithm plans grasps of a reasonably high quality on all the objects shown. Unfortunately, there is no ground truth against which to measure the success of the algorithm, other than by the judgement of the reader on viewing the results. The task requirements are simple enough, however, that the effectiveness of the method should be clear from the examples shown.

The grasp planning algorithm uses heuristics extensively. Most striking is the fact that the palm of the hand is fixed to be parallel to the ground plane. The justification for this was that in this configuration, collision of the hand with the ground plane is least likely. The height of the palm is set using knowledge of the object geometry but without reference to the kinematics of the hand. That reasonable grasps can be planned even when using such heuristics shows that the redundancy of the dextrous hand is indeed an asset in planning grasps. It enables some DOF of the hand to be fixed without detailed consideration of the kinematic and mechanical ramifications, and a good grasp can still be synthesised. In a complete hand-arm system, the constraint on the palm orientation and/or position might be set by the kinematics of the arm. For example, when a human is reaching up to a high shelf, there is little choice as to the angle at which the preshape is placed.

An equivalent use of the redundant DOF of the dextrous hand is seen in determining the grasp forces to be used. Much of the grasping literature has been concerned with synthesising equilibrium grasps for "disembodied" contacts. This thesis shows that such an approach is not necessary. In this thesis the grasp forces are fixed by making the assumption that the fingers do not abduct during grasp execution (this is generally true of human grasps). For a three-digit grasp this uniquely specifies the grasping forces to within a scaling factor. The position of the hand is chosen to minimise the friction that this type of grasp requires: this is how the mechanical and kinematic requirements of the grasp are resolved. It essentially uses the redundant DOF of the

hand to position the hand such that a prescribed set of internal forces, known to give a good grasp, can be exerted, rather than use the redundant DOF of the hand to try to find a good set of internal forces given a prescribed set of contact positions.

The usefulness of the grasp metric is difficult to quantify without a real system performing real-world tasks. Four terms of the metric are essential, namely the equilibrium, required friction, collision and joint limit terms. The contact angle metric is a reflection of certain task requirements and, for those tasks, is essential. Task wrench space metrics were not examined in this thesis. The joint torque metric is the least useful metric, as it was consistently very low and did not vary much between grasps (or correlate strongly with a human's perception of "good" and "bad" grasps). It is included to keep the hand away from singularities, while taking into account the forces to be exerted. This is probably too ambitious, and it would be better to rely on the joint limit term to achieve this.

The grasp family is the key to planning the grasp. For a given task, it provides a set of ideal grasps, in the form of the preshapes and ideal contact configurations, and the algorithm then tries to plan a grasp as close to this ideal as possible. The digit trajectories provide a way of deforming the hand from the ideal grasp to the real object contacts. This is an easy way of embodying task requirements, because one simply has to construct an ideal grasp for a a ideal contact configuration. The more ideal grasps one constructs, the closer an arbitrary point in the solution space is to one of the ideal grasps and the more optimal the grasp solution is likely to be.

An object model was constructed specially for the grasp planner. A surface model was used to drive the grasp planner. Essentially it is used to prune the solution space and to divide it into separate parts to which further pruning can be applied in a sensible fashion, *e.g.* it finds opposing planes which can then have an aperture plane constructed through their centre. However, as was seen with the toy rabbit example in Chapter 6, if there are several quadric patches on the object surface, the number of candidate grasping sets is huge (over 400 for the toy rabbit). Many of these are discarded erroneously because the ranking of the candidate grasping sets relies on a very coarse fitting of the preshape aperture, that runs the risk of failing to find valid grasps on small patches (*i.e.* there may be many false negatives). There may be a

large number of quadric patches on an object surface for one of two reasons. Firstly, as in the case of the toy rabbit, it may be a complex object. Secondly, the segmentation algorithm may over-segment a simple object object (see [Wren 92]).

Furthermore, as was shown in the polyhedron example of Chapter 5(Table 5.4), the ordering of the candidate grasping sets does not reflect particularly well the ordering of the final grasps; this leads to a sub-optimal grasp being planned on the polyhedron. If a surface model is used to prune the search space in this way, a better method of choosing candidate grasping sets is required.

For properly segmented simple objects, such as the cuboid shown in Chapter 4 or the egg example in Chapter 6, the surface model is useful in that it provides a compact description which aids the process of grasp planning, *e.g.* given two opposing planes, the grasp planner can automatically place the aperture plane at the centre of their overlapping z values.

The surface model is very useful for providing accurate estimation of the contact geometry. By fitting quadrics and planes to large numbers of range points, the effect of noise is much less than if normals were estimated using, for instance, 3×3 windows in the range images.

The volumetric model works well. It is fast to construct: the ray-casting is made faster by choosing the viewpoints to all be rotations about one of the axes of the voxmap. The distance transformed voxmap can be used for very fast collision checking for a hand of variable link width. The collision checking is deferred until after the grasp is planned. This did not produce any unwanted collisions in any of the grasps planned. This is mainly due to the nature of the grasp families, and the fact that the aperture fitting does take collision avoidance into account, albeit to a planar approximation. While there are pathological shapes of object, *e.g.* an object with protruding spikes, that would cause problems, the majority of object shapes and grasp families should produce collision-free grasps.

7.3 Further Work

Grasp families have been shown to be useful and the approach that the grasp planner takes — synthesising a real grasp as close to the ideal grasp specified by the grasp family — works well. Further work is needed to apply grasp families to more specialised tasks, such as gripping a tool such as a hammer or picking up an object with an exact manipulation in mind. Work also needs to examine the range of objects that can be acquired in a good grasp by the grasp families described in this thesis, and what type of object shapes cause them problems.

In Section 7.2 it was concluded that the surface model is more useful for surface *description* rather than surface *segmentation*. This means that it may be desirable to fit surfaces to the range images and use these to estimate the positions and normals at each point, but to conduct the search for a good grasp over the whole of the object without reference to surface boundaries. A natural way to do this would be to constrain the thumb contact to lie in one view and the finger contacts to lie in another. The problem then would be to bound the search sufficiently. The present grasp planner takes a candidate grasping set and constructs an aperture plane through it; without constraining the contacts to lie on a particular candidate grasping set, it may not be possible to select just one suitable aperture plane. However, by use of gradient descent seeded at various thumb contact points, it should be possible to find a good grasp.

The grasp planner needs to be refined to work better with curved objects. In Chapter 6 it was seen that there are problems due to the planarity assumption when fitting the aperture: only the components of the normals in the aperture plane are considered. Because the grasp families assume that the contact normals lie in the aperture plane, this can cause poor grasps that require more friction than necessary. A simple way to alleviate this is to consider the 3D normal when fitting the aperture, rather than just its projection in the 2D aperture plane. This should not slow the algorithm down significantly.

An alternative approach to grasp planning that should be investigated is the use of local inverse kinematics. The SVD description of the Jacobian provides a good way of describing its null and range space, which can be used to move the fingertips in a

prescribed direction or change the joint configuration without moving the fingertips (see Chapter 2). This scheme can be used to drag the fingertips to potential contact points (using the constrained distance transform as a metric), to slide the fingertips over the surface of the object or to change the hand configuration for a given set of contact points. These methods may also be used to refine the grasps planned in this thesis and to deform the hand away from collisions with the object or environment.

The true test of a grasp planner is that it is successful when applied to a full, working system. This thesis has provided a fast method of planning grasps for dextrous hands; the next stage is to apply it to a real system, consisting of an arm, hand and laser range-finder.

Bibliography

[Ali *et al.* 93] M.S. Ali, K.J. Kyriakopoulos, and H.E. Stephanou. The kinematics of the anthrobot-2 dextrous hand. In *IEEE International Conference on Robotics and Automation*, 1993.

[Baker *et al.* 85] B.S. Baker, S. Fortune, and E. Grosse. Stable prehension with a multi-fingered hand. In *IEEE International Conference on Robotics and Automation*, pages 570–575, 1985.

[Bard *et al.* 93] C. Bard, C. Bellier, C. Laugier, J. Troccaz, G. Vercelli, and B. Triggs. Achieving dextrous grasping by integrating planning and vision based sensing. Technical report, LIFIA-IRIMAG, 1993.

[Bard *et al.* 95] C. Bard, C. Laugier, C. Milesi-Bellier, J. Troccaz, B. Triggs, and G. Vercelli. Achieving dextrous grasping by integrating planning and vision-based sensing. *International Journal of Robotics Research*, 14(5):445–464, October 1995.

[Bekey *et al.* 90] G.A. Bekey, R. Tomovic, and I. Zeljkovic. *Control Architecture for the Belgrade/USC Hand*, chapter 7, pages 136–149. Springer-Verlag, 1990.

[Besl & Jain 85] P. Besl and R. Jain. Intrinsic and extrinsic surface characteristics. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1985.

[Besl 88] P.J. Besl. *Surfaces in Range Image Understanding.* Springer-Verlag, 1988.

[Blake 92] A. Blake. A theory of planar grasp. OUEL Report 1958/92, Oxford University Engineering Department, October 1992.

[Blake *et al.* 93] A. Blake, M. Taylor, and A. Cox. Grasping visual symmetry. In *Fourth International Conference on Computer Vision*, pages 724–733, 1993.

[Borgefors 84] G. Borgefors. Distance transformations in arbitrary dimensions. *Computer Vision, Graphics and Image Processing.*, (27):321–345, 1984.

[Brockett 85] R.W. Brockett. Robotic hands with rheological surfaces. In *IEEE Intl. Conference on Robotics and Automation*, pages 942–946, 1985.

[Cai 88] Li-Dong Cai. Diffusion smoothing: an approach to sculptured surfaces. D.A.I. research paper no. 406, Dept Artificial Intelligence, University of Edinburgh, September 1988.

[Cutkosky & Howe 90] M.R. Cutkosky and R.D. Howe. *Human Grasp Choice and Robotic Grasp Analysis*, chapter 1, pages 5–31. Springer-Verlag, 1990.

[Cutkosky & Wright 86] M.R. Cutkosky and P.K. Wright. Modeling manufacturing grips and correlations with the design of robotic hands. In *IEEE International Conference on Robotics and Automation*, volume 3, pages 1533–1539, 1986.

[Cutkosky 84] M. R. Cutkosky. Mechanical properties for the grasp of a robotic hand. Technical Report CMU-RI-TR-84-24, Carnegie-Mellon University, September 1984.

[Eggert *et al.* 96] D. Eggert, A.W. Fitzgibbon, and R.B. Fisher. Simultaneous registration of multiple range views for use in reverse engineering. In *Proceedings of the International Conference of Pattern Recognition*, pages 243–247, August 1996.

[Fearing 86a] R. Fearing. Implementing a force strategy for object reorientation. In *IEEE International Conference on Robotics and Automation*, pages 96–102, 1986.

[Fearing 86b] R.S. Fearing. Simplified grasping and manipulation with dextrous robot hands. *International Journal of Robotics and Automation*, (4):188–195, 1986.

[Fitzgibbon 92] Andrew W. Fitzgibbon. User guide for the bisque range data segmentation program. Imagine project software paper no.30, Dept Artificial Intelligence, University of Edinburgh, 1992.

[Gatrell 89] L.B. Gatrell. CAD-based grasp synthesis utilizing polygons, edges and vertices. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 184–189, 1989.

[Goyal *et al.* 89] S. Goyal, A. Ruina, and J. Papadopoulos. Limit surface and moment function description of planar sliding. In *IEEE International Conference on Robotics and Automation*, pages 794–799, 1989.

[Hoover *et al.* 96] A. Hoover, G. Jean-Baptiste, X. Jiang, P.J. Flynn, H. Bunke, D. Goldgof, K. Bowyer, D. Eggert, A. Fitzgibbon, and R. Fisher. An experimental comparison of range segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(7):673–689, July 1996.

[Howe *et al.* 88] R.D. Howe, I. Kao, and M.R. Cutkosky. The sliding of robot fingers under combined torsion and shear loading. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 103–105, 1988.

[Iberall 87] T. Iberall. The nature of human prehension: Three dextrous hands in one. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 396–401, 1987.

[Iberall *et al.* 88] T. Iberall, J. Jackson, L. Labbe, and R. Zampano. Knowledge-based prehension: Capturing human dexterity. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 82–87, 1988.

[Jacobsen *et al.* 86] S.C. Jacobsen, E.K. Iverson, D.F. Knutti, R.T. Johnson, and K.B. Biggers. Design of the utah/m.i.t. dextrous hand. In *IEEE International Conference on Robotics and Automation*, volume 3, pages 1520–1532, 1986.

[Jeannerod 81] M. Jeannerod. Intersegmental coordination during reaching at natural visual objects. In *Attention and Performance IX*. Erlbaum, Hillsdale, NJ, 1981.

[Ji & Roth 88] Z. Ji and B. Roth. Direct computation of grasp force for three-finger tip-prehension grasps. *Journal of Mechanisms, Transmissions and Automation in design.*, 110:405–413, December 1988.

[Kang & Ikeuchi 91] S.B. Kang and K. Ikeuchi. A framework for recognising grasps. Technical Report CMU-RI-TR-91-24, Carnegie Mellon University, November 1991.

[Kang & Ikeuchi 93] S.B. Kang and K. Ikeuchi. Temporal segmentation of tasks from human hand motion. Technical Report CMU-CS-93-150, Carnegie Mellon University, April 1993.

[Kao & Cutkosky 92] I. Kao and M.R. Cutkosky. Quasistatic manipulation with compliance and sliding. *International Journal of Robotics Research*, 11(1):20–40, February 1992.

[Khatib 85] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. In *IEEE International Conference on Robotics and Automation*, pages 500–505, 1985.

[Koehler & Donath 88] T. Koehler and M. Donath. Inverse kinematics for a multifingered hand. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 234–239, 1988.

[Lozano-Perez 87] T. Lozano-Perez. A simple motion-planning algorithm for general robot manipulators. *IEEE Transactions of Robotics and Automation*, 3(3):224–238, 1987.

[Lozano-Perez *et al.* 87] T. Lozano-Perez, J.L. Jones, E. Mazer, P.A. O'Donnell, and E.L. Grimson. Handey: A task-level robot system. In R.C. Bolles and B. Roth, editors, *The Fourth International Symposium on Robotics Research*, pages 29–36. MIT Press, 1987.

[Lyons 85] D.M. Lyons. A simple set of grasps for a dextrous hand. In *IEEE International Conference on Robotics and Automation*, pages 588–593, 1985.

[Lyons 86a] D. Lyons. Tagged potential fields: An approach to specification of complex manipulator configurations. In *IEEE International Conference on Robotics and Automation*, volume 3, pages 1749–1754, 1986.

[Lyons 86b] D. Lyons. Tagged potential fields: An approach to specification of complex manipulator configurations. In *IEEE International Conference on Robotics and Automation*, volume 3, pages 1749–1754, 1986.

[Mason & Salisbury 85] M.T. Mason and J.K. Salisbury. *Robot Hands and the Mechanics of Manipulation.* MIT Press, 1985.

[Matsuoka 95] Y. Matsuoka. Embodiment and manipulation learning process for a humanoid hand. Unpublished M.Sc. thesis, MIT, 1995.

[Michelman & Allen 94] P. Michelman and P. Allen. Forming complex dextrous manipulations from task primitives. In *IEEE International Conference on Robotics and Automation*, 1994.

[Nakamura 91] Y. Nakamura. *Advanced Robotics: Redundancy and Optimization.* Addison Wesley, 1991.

[Napier 56] J.R. Napier. The prehensile movement of the human hand. *J. Bone Joint Surgery*, 38B:902–913, 1956.

[Nguyen & Stephanou 90] T.N. Nguyen and H.E. Stephanou. A topological algorithm for continuous grasp planning. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 670–675, 1990.

[Nguyen 87a] V.-D. Nguyen. Constructing force-closure grasps in 3D. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 240–245, 1987.

[Nguyen 87b] V.-D. Nguyen. Constructing stable grasps in 3D. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 234–239, 1987.

[Nguyen 88] V.-D. Nguyen. Constructing force-closure grasps. *International Journal of Robotics Research*, 7(3):3–16, 1988.

[Park & Starr 92] Y.C. Park and G.P. Starr. Grasp synthesis of polygonal objects using a three-fingered robot hand. *International Journal of Robotics Research*, 11(3):163–184, 1992.

[Paul 81] R.P. Paul. *Robot Manipulators: Mathematics, Programming, and Control.* MIT Press, 1981.

[Pertin-Troccaz 87] J. Pertin-Troccaz. On-line automatic robot programming: A case study in grasping. In *IEEE International Conference on Robotics and Automation*, volume 3, pages 1292–1297, 1987.

[Pollard 90] N.S. Pollard. The grasping problem: Toward task-level programming for an articulated hand. Technical Report 1214, MIT Artificial Intelligence Lab, 1990.

[Pollard 93] N.S. Pollard. Planning grasps for a robot hand in the presence of obstacles. In *IEEE International Conference on Robotics and Automation*, 1993.

[Pollard 94] N.S. Pollard. Parallel methods for synthesizing whole-hand grasps from generalized prototypes. Technical Report AI-TR-1464, MIT Artificial Intelligence Laboratory, 1994.

[Press *et al.* 89] Press, Flannery, Teukolsky, and Vettering. *Numerical Recipes in C.* Cambridge, 1989.

[Rao *et al.* 88] K. Rao, G. Medioni, H. Liu, and G.A. Bekey. Robot hand-eye coordination: Shape description and grasping. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 407–411, 1988.

[Roberts 90] K.S. Roberts. Coordinating a robot arm and multi-finger hand using the quaternion representation. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1252–1257, 1990.

[Roth & O'Hara 87] G. Roth and D. O'Hara. A holdsite method for parts acquisition using a laser rangefinder mounted on a robot wrist. In *IEEE Conference on Robotics and Automation*, pages 1517–1523, 1987.

[Roth *et al.* 89] G. Roth, D. O'Hara, and M.D. Levine. A holdsite method for parts acquisition using a laser rangefinder mounted on

a robot wrist. *Journal of Robotic Systems*, 6(5):573–600, 1989.

[Rutishauer & Ade 95] M. Rutishauer and F. Ade. Detecting grasping opportunities in range data. In E.P. Baltsavias, editor, *ISPRS Intercommission Workshop: From Pixels to Sequences*, volume 30, pages 237–244, 1995.

[Salisbury & Craig 82] K. Salisbury and S. Craig. Articulated hands: Force control and kinematic issues. *International Journal of Robotics Research*, 1(1):4–17, 1982.

[Shimoga & Goldenberg 92] K.B. Shimoga and A.A. Goldenberg. Soft materials for robotic fingers. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1300–1305, 1992.

[Stansfield 88] S.A. Stansfield. Representing generic objects for exploration and recognition. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1090–1095, 1988.

[Stansfield 91] S.A. Stansfield. Robotic grasping of unknown objects: A knowledge-based approach. *Int. J. of Robotics Research*, 10(4), August 1991.

[Stephanou & Erkmen] H.E. Stephanou and A.M. Erkmen. Evidential classification of dextrous grasps for the integration of perception and action. *Journal of Robotic Systems*, 5(4):309–336.

[Taylor & Blake 94] M.J. Taylor and A. Blake. Grasping the apparent contour. In *Computer Vision - ECCV '94*, volume 1, pages 25–34. Springer-Verlag, 1994.

[Tomovic *et al.* 87] R. Tomovic, G.A. Bekey, and W.J. Karplus. A strategy for grasp synthesis with multifingered robot hands. In *IEEE International Conference on Robotics and Automation*, pages 83–89, 1987.

[Trucco & Fisher 95] E. Trucco and R. Fisher. Experiments in curvature-based segmentation of range data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2):177–182, February 1995.

[Whaite & Ferrie 93] P. Whaite and F.P. Ferrie. Active exploration: Knowing where we're wrong. In *Fourth International Conference on Computer Vision*, pages 41–48, 1993.

[Wing *et al.* 86] A.M. Wing, A. Turton, and C. Fraser. Grasp size and accuracy of approach in reaching. *Journal of Motor Behaviour*, 18(3), 1986.

[Wren & Fisher 93] D. Wren and R.B. Fisher. Identifying robot finger grasping points from range data. In *Sixth Intl. Conf. on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, pages 522–530. Gordon and Breach, 1993.

[Wren & Fisher 95a] D. Wren and R.B. Fisher. Dextrous hand grasping strategies using preshapes and digit trajectories. In *IEEE International Conference on Systems, Man and Cybernetrics*, pages 910–915, 1995.

[Wren & Fisher 95b] D. Wren and R.B. Fisher. Planning dextrous hand precision grasps from range data, using preshaping and finger trajectories. In *Third International Symposium on Intelligent Robotic Systems*, pages 213–220, 1995.

[Wren 92] D. Wren. Identifying robot finger grasping points from range data. Unpublished M.Sc. thesis, Dept. of A.I., Edinburgh University, 1992.

[Xiong *et al.* 93a] Y.L. Xiong, D.J. Sanger, and D.R. Kerr. Geometric modelling of bounded and frictional grasps. *Robotica*, 11:185–192, 1993.

[Xiong *et al.* 93b] Y.L. Xiong, D.J. Sanger, and D.R. Kerr. Geometric modelling of boundless grasps. *Robotica*, 11:19–26, 1993.

[Yang *et al.* 95] Y. Yang, Y. Zhang, and Q. Zhang. A performance evaluation of bh-2 dexterous robotic hand. In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, volume 1, pages 922–927, October 1995.

[Yoshikawa & Nagai 90] T. Yoshikawa and K. Nagai. *Analysis of Multi-fingered Grasping and Manipulation*, chapter 9, pages 187–208. Springer-Verlag, 1990.

Appendix A

The synthesis of an equilibrium grasp for the SAM

In this appendix, the synthesis of an equilibrium grasp (if one exists) with the SAM hand is described. This is used in Chapter 5 to find the forces for a set of planned contacts and a given hand configuration. These are then used by the grasp metric, in order to assess the quality of the grasp and choose between possible digit trajectories.

The procedure used follows the assumptions outlined in Chapter 3, namely:

- No abduction is used in the grasp execution, *i.e.* each contact force lies within the plane of its digit.
- The digits can exert any force within their plane.

The first assumption is reasonable to make because, as discussed in Section 5.2.2, the digit can exert forces better by flexion than abduction. The second assumption is true,unless the angle of the middle and distal joints is zero (see Section 5.2.2). This tends to occur for certain lateral grasps of objects that are of a width equal to the width of the palm of the hand. Even in this case, the digits can still exert forces perpendicular the distal link, however.

As explained in Chapter 2, an equilibrium grasp for 3 digits has the following three properties:

- All contact forces are coplanar (coplanarity constraint).
- All contact forces meet at a common point, known as the focus of internal force (coincidence constraint).
- The sum of the contact forces must be zero.

This fixes the set of contact forces to lie within the plane of the contacts, within which they have 3 DOF which can be characterised by the (x, y) position of the focus

of internal force and a scaling factor which multiplies the magnitude of each contact force. The focus of the internal force is determined by the hand kinematics and the scaling is set by the user. First, the directions of the contact forces are determined:

1. Find the plane that passes through the contact points: the contact forces must lie in this plane (coplanarity constraint)
2. Find the line of intersection of all the digit planes (which may be at infinity): they all intersect at a common line because, in the grasp strategies, the abduction of the fingers is set to be equal (and the abduction of the thumb is fixed at zero). The digits can only exert forces in their respective planes, so the focus of internal force must lie somewhere along this line (coincidence constraint).
3. The intersection of the line and the contact plane gives the position of the focus of internal force.

This reduces the problem to planar, as shown in Figure A.1. The i^{th} contact force is given by $\mathbf{f_i} = f_i\mathbf{a_i}$ ($\mathbf{a_i}$ has already been fixed by the coplanarity and coincidence constraints). The magnitudes of the contact forces can now be determined. The The i^{th} contact force is therefore given by:

$$\mathbf{f_i} = f_i \begin{pmatrix} a_i^x \\ a_i^y \end{pmatrix}$$

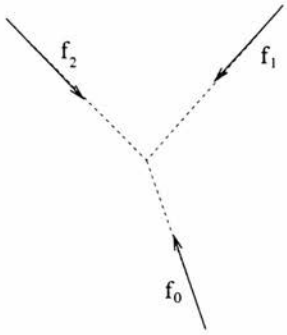


Figure A.1: The i^{th} contact force is given by $\mathbf{f_i} = f_i\mathbf{a_i}$ ($\mathbf{a_i}$ has already been fixed by the coplanarity and coincidence constraints).

For equilibrium, the forces sum to zero:

$$\begin{pmatrix} a_0^x & a_1^x & a_2^x \\ a_0^y & a_1^y & a_2^y \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \\ f_2 \end{pmatrix} = \mathbf{0}$$

The magnitude of the thumb contact force can be set arbitrarily in order to fix the scaling:

$$f_0 = 1$$

so the finger contact forces are then given by:

$$\mathbf{F}\begin{pmatrix} f_1 \\ f_2 \end{pmatrix} = \begin{pmatrix} -a_0^x \\ -a_0^y \end{pmatrix}$$

where

$$\mathbf{F} = \begin{pmatrix} a_1^x & a_2^x \\ a_1^y & a_2^y \end{pmatrix}$$

If $\det(\mathbf{F}) = 0$, then the two finger contact forces are parallel and the solution is simply:

$$f_1 = f_2 = \frac{1}{2}$$

Otherwise:

$$\begin{pmatrix} f_1 \\ f_2 \end{pmatrix} = \mathbf{F}^{-1}\begin{pmatrix} -a_0^x \\ -a_0^y \end{pmatrix}$$

If $f_1 > 0$ and $f_2 > 0$, the solution is valid; otherwise there is no solution, which indicates that the hand configuration and/or the contact points must be changed in order to give an equilibrium grasp. However, due to the geometry of the grasp strategies, such a situation rarely arises (indeed, it arises in none of the grasps shown in this thesis).

Appendix B

Glossary of terms

Abduction. Motion of a digit in a plane perpendicular to that of flexion.

Aperture closure trajectories. The projection of the digit closure trajectories in the aperture plane.

Aperture metric. An adaptation of the actual grasp metric which aims to produce an aperture fit that will lead to a high quality grasp metric.

Aperture plane. The plane in which the preshape aperture lies.

Dextrous hand. A hand capable of dextrous manipulation, *i.e.* it can manipulate an object between its digits whilst keeping its wrist fixed.

Dextrous manipulation. A manipulation where the object is moved relative to the hand reference frame, by motion of the digits.

Digit closure trajectories. The motions of the fingertips after the preshape is formed and the wrist position has been fixed.

Digit stoppage. The act of stopping the digits when contact is made with the object. A contact with the object may be detected through either force sensing at the actuators or by tactile sensors at the digit tips.

Distal link. The link of a digit furthest from the hand.

Dynamic stability. A grasp possesses dynamic stability if, when the object is given an arbitrary small perturbation of position and orientation, the object and hand return to their original positions.

Encompass grasp. A grasp with contact by links other than distal links, and/or the palm, *e.g.* holding a baseball bat, securely grasping a key to open a Yale-lock. It is a type of power grasp.

Force closure grasp. A grasp that can exert forces of arbitrary direction on the object.

Force/torque closure grasp. A grasp that can exert wrenches of arbitrary direction on the object.

Form closure grasp. A force/torque closure grasp with frictionless contacts.

Frictionless point contact model. A contact model where the contact can only exert forces perpendicular to the contact surface.

Flexion. Curling motion of a digit.

Focus of internal force. The point where the contact forces meet in a 3-contact grasp without gravity. This point lies in the plane of the contacts, and by variation of its position within the plane plus a force scaling factor, the null space of the grasp can be spanned.

Global inverse kinematics. The study of how to synthesise a hand configuration with a desired set of fingertip positions and/or orientations, subject to some metric, such as optimisation of joint values and collision avoidance.

Grasp. In the context of robotics, an object is grasped if it is held motionless in the presence of gravity by contact forces applied by a robot hand (which may be a dextrous hand, a parallel jaw gripper or a more specialised tool). A grasp consists of a contact configuration, a hand position, a preshape and a digit closure trajectory.

Grasp Family. A set of ideal contact configurations with associated preshapes and digit closure trajectories.

Ideal Contact Configuration. A set of contacts for which a high quality grasp can be synthesised.

Kinematically feasible. A grasp is kinematically feasible if the hand is able to reach the contacts without collision with itself, the object or the environment.

Lateral grasp. A grasp with contacts along the length of the distal links, *e.g.* grasping a computer mouse by the sides. It is a type of precision grasp.

Local inverse kinematics. The study of what joint motions should be used to produce a given small motion of the fingertips. Included is the study of singularities of the workspace.

Manipulation grasp. A grasp that, for a given set of fingertip positions, tries to keep the joints as close as possible to their central values. It is a type of precision grasp.

Precision grasp. The object is contacted by only the final segment of each digit. The precision grasp allows dextrous manipulation of the object, because the final segment of each digit can roll along the object surface, but is generally less stable than the power grasp.

Power grasp. The hand wraps around the object such that more than one segment of each digit — and very possibly the palm — contact the object. The power grasp gives greater stability than the precision grasp, because the object is strongly constrained by multiple contacts, but does not lie dextrous manipulation.

Preshape. A prescribed hand configuration which gives an optimal grasp on an ideal contact configuration.

Proximal link. The Link of a digit closest to the hand.

Preshape aperture. The relative positions of the fingertips in the preshape.

Point contact with friction model. The contact can exert forces which are directed within a friction cone with an axis coincident with the surface normal, of half-angle $\phi = tan^{-1}\mu$, where μ is the coefficient of friction.

Simple manipulation. The object remains fixed relative to the robot hand reference frame and any motion of the object is due only to movement of the robot arm.

Static stability. A grasp possesses static stability if it can exert arbitrary small wrenches about the equilibrium grasp.

Soft finger model. Adds the ability to exert a torque about the surface normal to the point contact with friction model. This can be illustrated by placing a sheet of paper on a smooth surface. If the reader presses down on this paper with a fingertip, and then rotates the fingertip, the paper rotates. This is due to the torque about the contact surface normal.

Tip grasp. A grasp made with fingertip contacts, *e.g.* holding a dart ready to throw. It is a type of precision grasp.

Torque closure grasp. A grasp that can exert torques of arbitrary direction on the object.

Appendix C

List of Symbols

$\mathbf{x}$	fingertip position vector
θ	joint vector
$\mathbf{f}$	fingertip force vector
τ	joint torque vector
$\mathbf{w_i}$	wrench exerted by the i^{th} contact
$\mathbf{n_i}$	inward surface normal at the i^{th} contact
$\mathbf{r_i}$	position vector of the i^{th} contact
$\mathbf{f_i}$	force exerted by the i^{th} contact
$\mathbf{a_i}$	direction of force exerted by the i^{th} contact
f_i	magnitude of force exerted by the i^{th} contact
μ	coefficient of friction
$\mathbf{J}$	Jacobian matrix
$\mathbf{K}$	stiffness matrix
L_i^f	transformation from link $i-1$ to link i in digit f when $\theta_i^f = 0$ (or $d_i^f = 0$)
θ_i^f	angle of rotation of link i about joint i in digit f
d_i^f	distance of translation of link i along joint i in digit f
a	length of the proximal link of the SAM
b	length of the middle link of the SAM
c	length of the distal link of the SAM
p	half the thumb-finger distance inthe palm of the SAM
q	half the finger-finger distance in the palm of the SAM
L	length of digit of SAM projected into the aperture plane

α	proximal angle of flexion of SAM
β	middle angle of flexion of SAM
γ	distal angle of flexion of SAM
δ	angle of abduction of SAM
$\mathcal{M}_i^M$	mechanical metric i
λ_i^M	coefficient of mechanical metric i
$\mathcal{M}_1^M$	equilibrium metric
$\mathcal{M}_2^M$	required friction metric
$\mathcal{M}_3^M$	task wrench space metric
$\mathcal{M}_i^K$	kinematic metric i
λ_i^K	coefficient of kinematic metric i
$\mathcal{M}_1^K$	joint torque metric
$\mathcal{M}_2^K$	joint limit metric
$\mathcal{M}_3^K$	contact angle metric
$\mathcal{M}_4^K$	collision metric
m_i	aperture metric i
m_1	distance to contact aperture metric
m_2	required friction aperture metric
m_3	distance to centre aperture metric
m_4	collision aperture metric
$\mathcal{A}_i$	aperture plane metric of i^{th} grasping set
$\mathcal{Q}_i$	grasping set quality metric of i^{th} grasping set
$\mathcal{F}_i$	coarse aperture fit metric of i^{th} grasping set