Matei T. Ciocarlie
Peter K. Allen

Department of Computer Science,
Columbia University,
New York, NY 10027,
USA
{cmatei, allen}@cs.columbia.edu

Hand Posture Subspaces for Dexterous Robotic Grasping

Abstract

In this paper we focus on the concept of low-dimensional posture subspaces for artificial hands. We begin by discussing the applicability of a hand configuration subspace to the problem of automated grasp synthesis; our results show that low-dimensional optimization can be instrumental in deriving effective pre-grasp shapes for a number of complex robotic hands. We then show that the computational advantages of using a reduced dimensionality framework enable it to serve as an interface between the human and automated components of an interactive grasping system. We present an on-line grasp planner that allows a human operator to perform dexterous grasping tasks using an artificial hand. In order to achieve the computational rates required for effective user interaction, grasp planning is performed in a hand posture subspace of highly reduced dimensionality. The system also uses real-time input provided by the operator, further simplifying the search for stable grasps to the point where solutions can be found at interactive rates. We demonstrate our approach on a number of different hand models and target objects, in both real and virtual environments.

KEY WORDS—interactive grasping, dexterous robotic hands, hand prosthetics.

1. Introduction

The vision of ubiquitous robotic assistants, whether in the home, the factory or in space, will not be realized without the ability to grasp typical objects in human environments. The human hand, the most versatile end-effector known, is capable of a wide range of configurations and subtle adjustments.

Figures 1, 3–8, 10–13 appear in color online: http://ijr.sagepub.com

In an attempt to match its abilities, a number of anthropomorphic robotic designs have been proposed in the literature, e.g. by Vande Weghe et al. (2004); such models often include human-like kinematics and simplified tendon networks. However, the increase in versatility has come at the cost of similarly increased complexity. As the number of degrees of freedom (DOFs) of robotic hands starts to approach the case of the human hand, effective autonomous algorithms that can handle high-dimensional configuration spaces are required in order to take advantage of the new designs.

If we wish to reproduce human-like grasping it would seem natural to draw inspiration not only from the hardware of the human hand, but also from the software; that is, the way the hand is controlled by the brain. This may initially sound like an overly lofty goal: a large part of the human cortex is dedicated to grasping and manipulation, and it would seem reasonable to assume that all of this cognitive machinery is dedicated to finely controlling individual joints and generating highly flexible hand postures. However, results in both robotics and neuroscience research that we review in this paper point to the contrary, suggesting that a majority of the human hand control during common grasping tasks lacks individuation in finger movements.

1.1. Low-dimensional Posture Subspaces

In this paper, we use low-dimensional hand posture subspaces to express coordination patterns between multiple DOFs for robotic hands. In particular, we consider linear subspaces defined by a number of basis vectors that we refer to as *eigengrasps*. Each eigengrasp is a vector in the high-dimensional hand posture space; we use linear combinations of a relatively small number of these vectors to obtain a wide range of hand postures for grasping tasks.

A key aspect when using this approach is the trade-off between its computational advantages and the implied reduction in the range of directly accessible hand postures. An eigengrasp subspace is only useful in as much as it contains enough

851

variance in hand posture to allow for successful completion of the grasping task. In this paper we start from the results of Santello et al. (1998), who applied dimensionality reduction methods on a large set of human grasping postures obtained from user studies. Their results show that a two-dimensional subspace contains more than 80% of the variance in hand posture. The analysis of human digit coordination patterns during grasping is in general a very active area of research; in Section 2 we provide an overview of current results and discuss their implications for our approach to robotic grasping.

Our main interest in this study is the application of low-dimensional posture subspaces for robot hands, taking a constructive, rather than exploratory approach. Instead of attempting to derive optimal posture subspaces (either analytically or through user studies), we focus on the applicability of this concept: given a particular set of eigengrasps, we aim to construct algorithms that take advantage of operating in a low-dimensional domain. In Section 3 we present an eigengrasp planning algorithm that can be used to obtain form-closure grasps using dexterous hands that have traditionally been very difficult to plan for. The core of this algorithm is an optimization procedure that operates along two eigengrasp directions; even when using such a reduced dimensionality space, we show that the planner is successful in deriving stable multi-fingered grasps for a large variety of target objects.

1.2. Interactive Grasp Planning

One of the key features of the low-dimensional grasp planning algorithm we introduce is the ability to simplify the search for a form-closure grasp posture when using a dexterous hand. However, a grasp is completely defined only by a combination of intrinsic DOFs (finger joint angles) and extrinsic DOFs (wrist position and orientation relative to target object). The application of eigengrasp subspaces only addresses the finger posture component; a significant amount of computational effort still has to be spent in order to determine the position and orientation of the wrist. In order to take full advantage of the eigengrasp dimensionality reduction, we need to also reduce the complexity of the extrinsic components of a grasp.

In Section 4 we show how the eigengrasp planning framework can be applied in the case where the approach direction for a grasping task is partially provided by a human operator in real time. This application stems from the area of hand neuroprosthetics, where a human user must interact with an artificial limb using limited communication channels. In particular, we assume that the operator has no direct control over finger posture, which has to be set by the automated grasp planner. Computational efficiency thus becomes a critical requirement, as the system's response should be fast enough to allow for interactive operation.

We define an interactive grasp planner as a system that can accept input from a human operator during the execution of the grasp and adapt to on-line input changes. We show that by combining reduced dimensionality grasp planning with wrist position input provided by a human operator, we can meet these constraints: the system that we present generally requires approximately 2 seconds to find a form-closure grasp for a user-specified wrist position, and between 10 and 15 seconds for the complete execution of a grasping task. This framework allows the human user to not only set initial guidelines for the grasp planner, but also to react to its behavior and successfully complete the grasping task even if they have no direct control over finger posture.

1.3. Related Work

Attempts to formalize the human tendency to simplify the space of possible grasps can be traced back to Napier's pioneering grasp taxonomy (Napier 1956), updated later by Cutkosky (1989). Iberall (1997) later reviewed a large field of work on grasp taxonomies, from areas such as anthropology, biomechanics, rehabilitation and robotics. These studies suggest that, while the configuration space of dexterous hands is high-dimensional and very difficult to search directly, most useful grasps can be found in the vicinity of a small number of discrete points. This approach has generated significant interest in both human and robotic grasping research; in this section we review a number of results on autonomous grasp planning for robotic hands. In the following section we also discuss in more detail the kinesiologic aspects that are closely related to the approach presented in this paper.

Miller et al. (2003) used Cutkosky's grasp taxonomy concept to define a number of starting positions, or pre-grasps, when searching for good grasps of a given object using a robotic hand. Cipriani et al. (2006) applied this concept for prosthetic hands, assuming that the human operator can only select from a small set of pre-grasp shapes, relying on the passive mechanical adaptability of the CyberHand design (Carrozza et al. 2006) to complete the grasp. Aleotti and Caselli (2006) used a Cyberglove to record human grasp trajectories and postures and replicated them on the same target objects using NURBS. Li et al. (2007) used a shape-matching approach, sampling an object into a dense cloud of oriented points and matching against a small database of known human hand poses.

We note that choosing a good grasp can also be formulated as a problem in the contact space of the object to be grasped rather than the configuration space of the hand, as shown for example by Roa and Suarez (2007). However, such approaches usually require inverse kinematics in order to guarantee that the contacts are physically satisfiable by a real robotic hand. An alternative to the use of inverse kinematics is presented by Platt et al. (2002, 2004), starting with the hand in contact with an object and combining multiple control laws for performing incremental contact adjustments.

When a robot is operating in an unknown environment, the amount of sensory information can be insufficient for constructing a complete three-dimensional model of the target object. Saxena et al. (2008) present a learning approach where logistic regression is used to infer good grasping points for a simple gripper based directly on two-dimensional images, without building an explicit object model. Their method focuses on the ability to learn the appearance subspace of graspable objects or features; in the eigengrasp framework presented here we explore the use of subspaces for hand postures. These approaches can thus be considered as complementary. Other methods for operating in unstructured environments include explicitly modeling the uncertainty associated with inaccurate range sensors, as shown by Hsiao et al. (2007), and using tactile sensing to compensate for other sensing errors, as demonstrated by Edsinger and Kemp (2006). Finally, for comprehensive overviews regarding fully autonomous grasp synthesis for robotic hands we also refer the reader to the reviews of Shimoga (1996) and Bicchi and Kumar (2000).

2. Eigengrasps

An important difficulty in understanding and reproducing human grasping ability is the large number of DOFs involved, which gives rise to an enormous set of possible configurations. One possible explanation for human efficiency in selecting appropriate grasps assumes that humans unconsciously simplify the large search space through learning and experience. Santello et al. (1998) investigated this hypotheses by collecting a large set of data containing grasping poses from subjects that were asked to shape their hands in order to mime grasps for a large set ($n = 57$) of familiar objects. Principal component analysis of this data revealed that *the first two principal components account for more than 80% of the variance*, suggesting that a very good characterization of the recorded data can be obtained using a much lower-dimensionality approximation of the DOF space. In our work, we refer to the principal components of the dataset of hand configurations described above as *eigengrasps*.

While numerical analysis of human hand postures can reveal the "synergies" in the data, it tells us very little about the *causes* of this intrinsic low-dimensional nature. Two explanations seem natural: the first assumes that inter-digit coordination is caused by mechanical constraints in the anatomy of the hand. This direction suggests building robotic hands with highly interconnected finger actuation mechanisms. An example is the prototype developed by Brown and Asada (2007), using a low-dimensional control system along directions similar to those presented by Santello et al. (1998). The second explanation assumes that motor control synergies take place at a higher level in the central nervous system, as discussed for example by Mason et al. (2001) and Cheung et al. (2005).

This approach implies the use of low-dimensional control algorithms for dexterous robotic hands, such as that presented in this paper. However, the nature of human control synergies is still an open question and an active area of research, and combinations of the two approaches discussed above also seem very likely.

Another important aspect concerns the relationship between eigengrasps and the task being performed. Todorov and Ghahramani (2004) have shown that the execution of different manipulation tasks (such as flipping pages or crumpling paper) is characterized by different sets of principal components. Interestingly, Thakur et al. (2008) have identified a posture subspace even in the less-constrained setting of haptic exploration tasks. Mason et al. (2001) and Santello et al. (2002) have also shown that hand posture during the reach phase of a complete reach-to-grasp action is described by a different (and lower-dimensional) principal component spectrum than the grasp phase. These results show that, when using a low-dimensional control space for robotic hands, the choice of the subspace has to be correlated with the proposed task.

Finally, all of the studies discussed so far have used principal component analysis, and thus have addressed only linear subspaces that can be extracted from hand posture data. Linear decomposition has been successfully used in the past on different types of biometric data, ranging from face appearance (Turk and Pentland 1991) to the dynamics of arm motion (Fod et al. 2002). However, non-linear dimensionality reduction methods can potentially reveal different manifold structures of the same data. Tsoli and Jenkins (2007) compared a number of such methods, including isomap and locally linear embedding, for extracting two-dimensional non-linear manifolds from human hand motion data. Their results show that, while low-dimensional manifolds can be obtained using a number of different methods, non-linear approaches can provide better separation between the low-dimensional projections of different task domains and thus simplify the task of low-dimensional teleoperation.

2.1. Application for Robotic Hand Models

One common thread that can be observed in the body of work discussed above is that the usefulness of a hand posture subspace has traditionally been quantified by how well it approximates a given set of input data. This exploratory approach is natural in the context of studying the human hand. In this paper we propose a modified approach oriented towards application for artificial hands: given a hand posture subspace, we use it to synthesize new hand postures to accomplish a particular task. We see this effort as complementary to current attempts of understanding and extracting relevant low-dimensional data: if eigengrasp-based algorithms can be proven effective, they would only benefit from further optimization of the operation subspace.

The task that we focus on throughout this paper is achieving stable grasps of a target object. We are thus interested in dexterous grasps that can resist a wide range of disturbances, rather than object-specific manipulation scenarios. To achieve this goal, we propose an algorithm that actively searches a low-dimensional subspace for appropriate hand postures. We base our approach on published results obtained from human grasping data, which can also be applied to robotic models using an empirical mapping as described below. While we have found our choices to produce good results for achieving stable grasps of a large variety of objects common in human environments, the optimal choice of eigengrasps for non-human hands, as well as the choice of which eigengrasps to use for different or more specialized tasks, are open questions and interesting directions for future research.

In this study, we apply the eigengrasp concept to a total of four dexterous hand models: the Barrett hand (Barrett Technologies, Cambridge, MA), the DLR hand (Butterfass et al. 1998), the Robonaut hand (Lovchik and Diftler 1998) and finally a 20-DOF model of a human hand. For the human hand we have directly used the eigengrasp subspace obtained by Santello et al. (1998), taking advantage of the fact that it has been derived through rigorous study over a large number of recorded samples. Since such data is not available for robotic hand models, we have derived eigengrasp directions attempting to define grasp subspaces similar to those obtained using human hand eigengrasps. In most cases, such decisions could be based directly on the similarities with the human hand: for example, the human metacarpophalangeal (MCP) and interphalangeal (IP) joints can be mapped to the proximal and distal joints of robotic fingers. In the case of the Barrett hand, changes in the spread angle DOF were mapped to human finger abduction. All of our hand models, as well as the two dominant eigengrasps used in each case, are presented in Table 1.

The eigengrasp concept allows us to design flexible control algorithms that operate identically across all of the presented hand models. The key to this approach is that the eigengrasps encapsulate the kinematic characteristics of each hand design. Control algorithms that operate on eigengrasp directions do not need to be customized for low-level operations, such as setting individual DOFs, and can concentrate on the high-level task. All of the results presented in this paper *were obtained by treating all hand models identically*, without the need for any hand-specific tuning or change in parameters.

2.2. Effective DOFs

In the applied example of grasp planning, we need to consider whether the eigengrasp subspace contains the hand postures needed for stable grasps of the target objects. A corollary question is whether results obtained using a small set of eigengrasps would imply that the other DOFs of the hand are useless. As mentioned above, in the case of the human hand, the two dominant eigengrasps have been shown to encapsulate most of the variance in posture over a large set of grasps. However, eigengrasps 3 through 6 (in decreasing order of importance), while accounting for less than 15% of the posture variance, do not represent noise and have been shown by Santello et al. (1998) to be related to the object to be grasped. Furthermore, the study was performed in the absence of the real object, as subjects were asked to reproduce grasps from memory. This suggests that, even if we choose to perform the grasp planning stage in a low-dimensional space, during the final stages of the grasp the shape of the object will force the hand to deviate from eigengrasp space in order to conform exactly to its surface.

We therefore use a two-stage approach to the task of automated grasp synthesis: first, hand posture is optimized in a low-dimensional eigengrasp space. The dimensionality reduction makes this process computationally tractable even for complex dexterous hand models. In the second stage, starting from the best hand posture found in eigengrasp space, the hand is closed by flexing all of the finger joints at equal rates until contact with the target object stops all motion. This step does not require the control algorithm to perform any more pose refinement at a computational level, but only to issue a binary "close all fingers" command after which the final pose is determined implicitly through contact with the object. However, it takes advantage of the versatility of complex kinematic chains, where multiple DOFs allow the hand to better match the surface of the object.

3. Grasp Synthesis through Low-dimensional Posture Optimization

In general, automatic grasp synthesis can be approached as an optimization problem, seeking to maximize the value of a high-dimensional quality function Q that characterizes a given combination of hand posture and position:

$$Q = f(\boldsymbol{p}, \boldsymbol{w}), \quad \boldsymbol{p} \in \mathcal{R}^d, \; \boldsymbol{w} \in \mathcal{R}^6, \quad (1)$$

where d is the number of intrinsic hand DOFs, $\boldsymbol{p}$ represents the hand posture and $\boldsymbol{w}$ contains the position and orientation of the wrist.

We first present our implementation of the quality function, then discuss the optimization algorithm that is applied to maximize it over a space of possible hand postures. In general, most quality function formulations are highly non-linear, with complex constraints as well as gradients that are difficult, or even impossible, to compute analytically. These problems are compounded by the high dimensionality of the optimization domain. Consider, for example, the case of the human hand model, where $d = 20$: this results in a 26-dimensional optimization domain, rendering most optimization algorithms intractable. However, we can choose a basis comprising b eigengrasps, with $b \ll d$, and a hand posture placed in the subspace

Table 1. Eigengrasps defined for the robotic hand models used in this paper.

Model	DOFs	Eigengrasp 1			Eigengrasp 2		
		Description	min	max	Description	min	max
Barrett	4	Spread angle opening			Finger flexion		
DLR	12	Prox. joints flexion Finger abduction Thumb flexion			Dist. joints flexion Prox. joints extension Thumb flexion		
Robonaut	14	Thumb flexion MCP flexion Index abduction			Thumb flexion MCP extension PIP flexion		
Human	20	Thumb rotation Thumb flexion MCP flexion Index abduction			Thumb flexion MCP extension PIP flexion		

defined by this basis can be expressed as a function of the amplitudes a_i along each eigengrasp direction:

$$p = p_m + \sum_{i=1}^{b} a_i e_i, \qquad (2)$$

where p_m is a "mean" posture that describes the origin of the eigengrasp subspace. Each eigengrasp e_i is a d-dimensional vector and can also be thought of as a direction of motion in joint configuration space. Motion along one eigengrasp direction will usually imply motion along all (or most) degrees of freedom of the hand:

$$e_i = [e_{i,1} \quad e_{i,2} \quad \ldots \quad e_{i,d}]. \qquad (3)$$

Once this subspace is defined, a hand posture can be completely determined by the amplitude vector $a = [a_1 \quad \ldots \quad a_b] \in \mathcal{R}^b$. Therefore, when hand posture optimization is performed in eigengrasp space, the grasp quality function over this subspace takes the form

$$Q = f(a, w), \quad a \in \mathcal{R}^b, \, w \in \mathcal{R}^6, \qquad (4)$$

where a is the vector of eigengrasp amplitudes. When operating in a two-dimensional subspace, we therefore have a total of eight variables to optimize, including two eigengrasp amplitudes and six variables for the wrist position and orientation, independent of the particular hand model that is being used for the grasping task.

3.1. Quality Function Formulation

Most grasp quality metrics that have been proposed in the literature are based on the locations of the contacts between the hand and the target object. Our context is somewhat different: we need a quality metric that can also assess the quality of a *pre-grasp*, where the hand is very close, but not in contact with the target. For each hand model, we pre-define a number of *expected* contact locations by sampling each link of the fingers as well as the palm, as shown in Figure 1(a). The value of the quality function is maximized for those hand postures that bring each expected contact location as close as possible to the target object. We are therefore searching for postures where the hand is wrapped around the object, generating a large contact area using all of the fingers as well as the palm. As shown in Figure 1(b), for each desired contact location on the hand, identified by the index i, we define the local surface normal $\hat{n}_i$ as well as the distance o_i between the desired contact location and the target object. We then compute a measure δ_i of the distance (both linear and angular) between the desired contact and the surface of the object:

$$\delta_i = \frac{|o_i|}{\alpha} + \left(1 - \frac{\hat{n}_i \cdot o_i}{|o_i|} \right), \qquad (5)$$

where α is a scaling parameter required to bring the range of useful linear distances (measured in millimeters) in the same range as the normalized dot product between $\hat{n}_i$ and o_i (in our study we use a value of $\alpha = 50$). For a given hand posture, the total value of the quality function is then computed as:
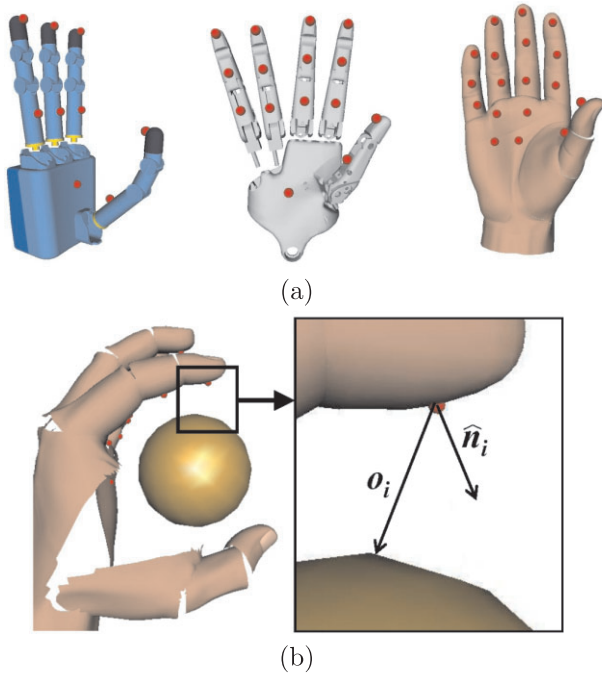
(a)

(b)

Fig. 1. Examples of desired contact locations for posture optimization. (a) Complete set of pre-defined desired contact locations for the DLR, Robonaut and human hands. (b) For a desired contact with index i, we define the surface normal $\hat{n}_i$ and the current distance to the target object o_i.

$$Q = \sum_{\substack{\text{all desired}\\ \text{contacts}}} (1 - \delta_i). \qquad (6)$$

In most cases, the hand postures that maximize the value of Q create an enveloping grasp of the object, especially for complex models grasping objects similar in size to the hand. The optimized value of this function can be seen as a measure of how well the hand shape can be set in order to match a given object while operating in a low-dimensional subspace. In Section 4, we also present an alternative quality function formulation that includes a built-in notion of grasp wrench space analysis.

3.2. Optimization Algorithm

After choosing the formulation of the quality function Q, the optimization is performed using the simulated annealing algorithm with the fast cooling schedule and neighbor-generation function presented by Ingber (1989). The stochastic nature of this algorithm makes it a particularly good choice for our task: since new states are generated as random neighbors of the current state, computation of the quality function gradient is not necessary, and the algorithm works well on non-linear functions. Furthermore, the possibility of a "downhill move" to a

state of lower quality allows it to escape local optima which can trap greedier methods such as gradient ascent.

Algorithm 1 Simulated annealing over grasp quality function.

```
for all variables of CurrentState do
    CurrentState.variable = RandomValue()
end for
QCurrent = Quality(CurrentState)
Iterations = 0
QSaved = 0

while  Iterations ≠ MaxIterations  do
    {Generate a new state as a neighbor of current state}
    repeat
        for all variables of NewState do
            {Sim. Annealing neighbor generation function}
            NewState.variable = Ngbr(CurrentState.variable)
        end for
        Apply ForwardKinematics(NewState)
        if collisions detected or joint limits exceeded
            legalState = false
        else legalState = true
    until legalState == true

    QNew = Quality(NewState)
    if QNew > QSaved then
        Insert NewState in SavedStatesList
        QSaved = lowest quality value in SavedStateList
    end if

    {Sim. Annealing probability of "jumping" to new state.}
    PJump = Probability(QCurrent, QNew)
    if PJump > 0.5 then
        CurrentState = NewState
        QCurrent = QNew
    end if
    Iterations = Iterations + 1
end while
```

The complete optimization procedure is presented in Algorithm 1, which uses the following conventions. The variables that make up a given state (such as CurrentState or NewState) are the entries of the eigengrasp amplitude vector a and the wrist position and orientation vector w. These variables are the target of the optimization. The Quality function for a given hand state is computed as in (6). We have implemented this algorithm using our publicly available *GraspIt!* simulation engine (Miller and Allen 2004), which performs the ForwardKinematics computation and contact and collision detection. Finally, the functions Nbr, for computing a "neighbor" of a variable, and Probability, for deciding whether a "jump" to a new state is performed, are implemented
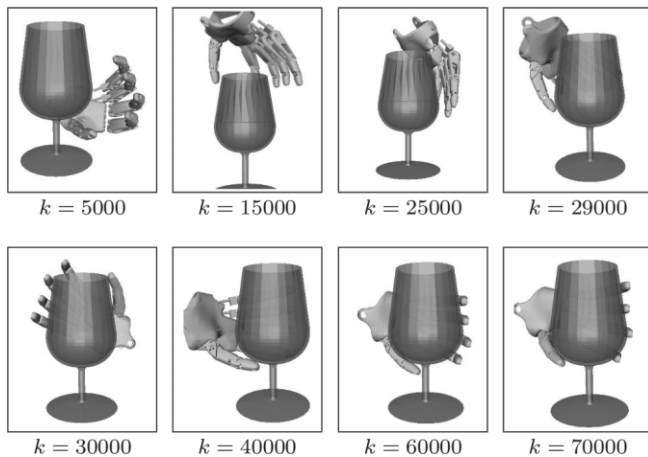
Fig. 2. Simulated annealing example over 70,000 iterations. Each image shows the best state found until iteration k.

Fig. 3. Evolution of quality function (top) and eigengrasp amplitudes (bottom) of the current search state during simulated annealing

as described by Ingber (1989). Briefly, the simulated annealing algorithm implements the following guidelines: (a) during early iterations, it allows large changes in the search variables and often jumps to worse states in order to sample the entire domain of the optimized function; (b) as the algorithm progresses, it predominantly samples increasingly smaller neighborhoods of the current solution and only allows jumps that improve its quality measure.

A detailed example of the execution of this algorithm, involving the Robonaut hand grasping a glass, is presented in Figure 2. The figure shows the temporary solution (best state found so far) at various points during the optimization. Figure 3 also shows how the current search state evolves over the full iteration range. We can observe what is considered typical behavior for a simulated annealing implementation. First, the search goes through random states, accepting bad positions as well as good positions. As the annealing schedule progresses, the search space is sampled more often in the vicinity of the good states, while bad states are no longer accepted.

Owing to the stochastic nature of simulated annealing, different executions of the optimization algorithm can result in slightly different hand postures. However, the same stochastic nature enables it to "jump out" of unpredictable local optima (such as the intermediate "peaks" in Figure 3) and, with a high probability, converge to the same regions of the optimization space, leading to consistency between different executions. Finally, in the later stages, the search is confined in a small neighborhood around the best state, which is progressively refined. The total time required for the optimization presented here was 143 seconds, or 2.0 ms per iteration, using a commodity desktop computer. The most significant amount of computation was spent checking the feasibility of each generated state (i.e. checking for collisions and inter-penetrations). We also note that increasing the number of iterations beyond 70,000 yields
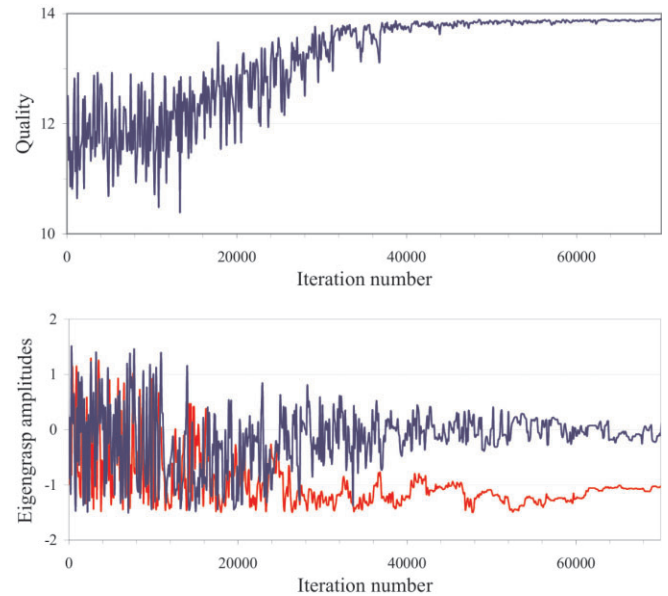
highly diminished returns; all of the optimizations reported in the rest of this section where performed over an identical range of 70,000 iterations, or approximately 150 seconds of computation.

3.3. Optimization Results and Discussion

In this section we present quantitative testing results of the optimization method presented above. We discuss the best dimensionality of the optimization subspace, the nature of the hand postures that can be found through our optimization algorithm in this subspace, and its overall applicability for the task of dexterous grasp planning.

In order to study the impact of the dimensionality of the search space on the results of the optimization, we compared the results obtained using the human hand in an eigengrasp subspace of dimensionality ranging from one to six. All of the tests were performed on a set of seven objects with diverse geometry, such as a flask, shoe, hammer, etc. To reduce the influence of the stochastic component of simulated annealing, the optimization for each combination of object and number of eigengrasps was repeated five times and the results were averaged. The complete results, showing how the value of the quality function varies with the dimensionality of the space at various points in the optimization, are presented in Figure 4.

The results show that, in our optimization range, a two-dimensional subspace provides the best results. A more detailed analysis also reveals that, in the early stages of the optimization, a one-dimensional space is qualitatively similar,
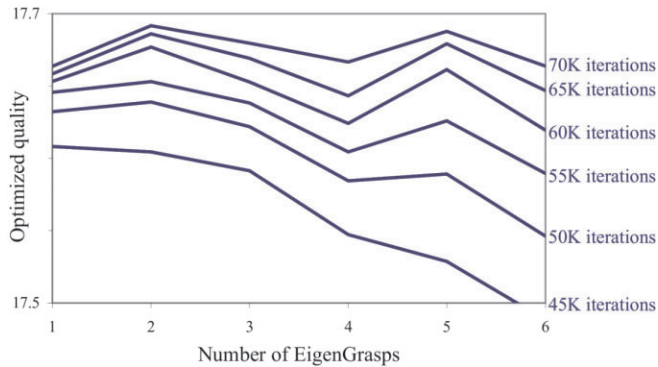
Fig. 4. The best value of the quality function at different moments during the optimization algorithm, depending on the dimensionality of the optimization subspace. All of the tests were performed using the human hand and averaged over a set of five executions for each of seven test objects.

while in the latter stages a higher-dimensional space can provide a viable alternative. This is an expected trend, as an increase in the dimensionality of the search space intuitively requires additional computational power to provide benefits. However, we must note that these results could also be indicative of our specific optimization algorithm, or of the set of chosen objects, rather than the intrinsic nature of the eigengrasp subspace. In particular, it is difficult to explain exceptions to the overall trend, such as the relative benefit yielded by a five-dimensional space compared with either four or six dimensions. This is compounded by the fact that it is very difficult to find intuitive explanations for human eigengrasps ranked below the first two. Furthermore, to the best of our knowledge, no set of objects has been accepted as a definitive benchmark of robotic grasping performance. Based on our current results, we have chosen a two-dimensional subspace as offering the best compromise between computational effort and optimization results; all of the experiments presented in the rest of this paper, including the interactive grasp planning application that is the focus of Section 4, were performed using two eigengrasps.

In order to test the effectiveness of our framework for the task of dexterous robotic grasp planning, we have applied the two-dimensional eigengrasp optimization using all four previously discussed robotic hand models on a set of six objects. Figure 5 shows the result of the annealing search for each hand–object combination. Our focus in this section is to evaluate the best hand postures that can be found in eigengrasp space. Therefore, Figure 5 presents the best hand posture found by the optimization algorithm without any additional refinements, allowing a direct assessment of the optimization method through visual inspection of its output.

These results show that, when the search is confined to a low-dimensional eigengrasp space, it does not reach a global optimum of the quality function where all of the desired contact locations touch the target object. However, the local optimum found in eigengrasp space can be used as a *pre-grasp* by performing the additional adjustment where the hand leaves the planning subspace in order to conform to the surface of the object: execution of the binary "close all fingers" command, allowing all fingers to close until motion is stopped by contact with the object (Figure 6). We use form-closure as the analysis criterion for the resulting grasps, as our goal is the synthesis of stable grasps with no weak points.

In order to perform a quantitative analysis of the pre-grasps obtained through posture optimization, we can apply this adjustment to the 20 distinct solutions with the highest quality values found by one execution of the optimization algorithm. We consider two solutions as being distinct if either the distance between the hand positions they define exceeds 20% of the object size, or the difference between wrist orientations exceeds 20°. After closing the fingers, we count the number of distinct optimized pre-grasps that result in form-closure. In order to account for the stochastic element, we repeated the test for each hand–object combination five times. The average number of form-closed grasps (as well as the standard deviation) for all cases are presented in Table 2. Each optimization was performed over 70,000 iterations, with an average running time of 158 seconds. In the case of the human hand, Figure 6 also shows all of the final grasps obtained when using as pre-grasps the corresponding postures from Figure 5.

These findings confirm our expectations of eigengrasp space as a *pre-grasp* or *grasp planning space*: in general, closing the fingers of a dexterous hand starting from a random configuration is not enough to obtain a stable grasp. Our results show that if the starting position is the result of the eigengrasp optimization algorithm we can obtain multiple solutions: on average, 20 optimized pre-grasps result in seven form-closed grasps for a given hand and object. Interestingly, our algorithm performs at its best for the more dexterous designs, with kinematic structures approaching that of the human hand. This result can be explained by the fact that all of the eigengrasps subspaces that we use originate from a study of human grasping; as the mapping to robotic hands becomes less intuitive, the effectiveness of the planning method is also decreased. Future methods for subspace mapping between hands should also take into account their relative size (for example, the palm and finger span of the DLR hand are approximately twice as large as those of its human counterpart). Overall, the results confirm our starting hypothesis: a low-dimensional algorithm can take advantage of highly dexterous hand designs for synthesizing stable grasps in a computationally efficient way.

4. On-line Interactive Dexterous Grasping

In the previous section we have presented an optimization algorithm that uses a low-dimensional subspace when searching
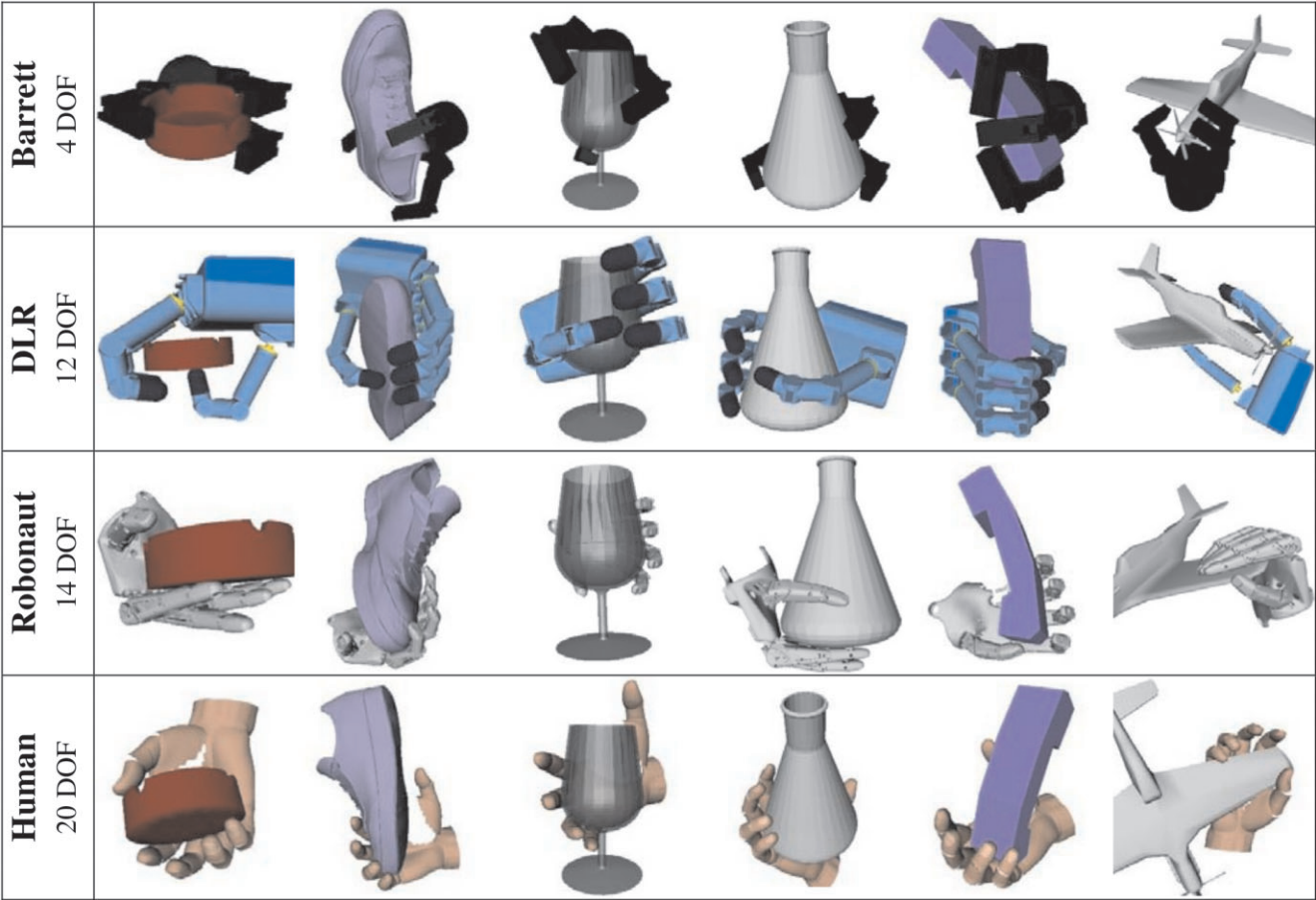
Fig. 5. Best hand postures found in a two-dimensional eigengrasp space using simulated annealing optimization.
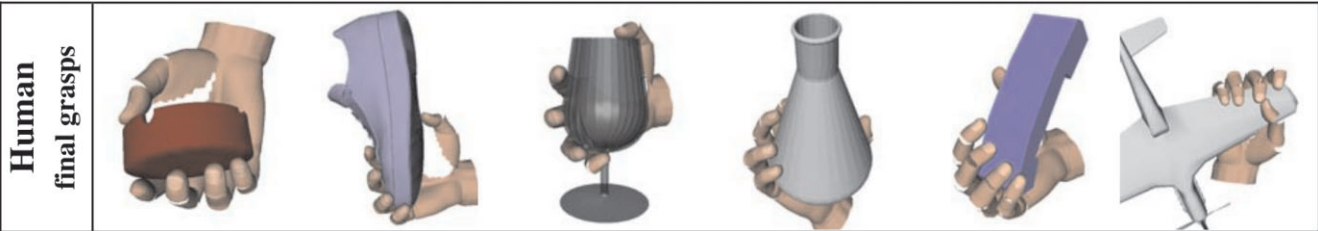
Fig. 6. Examples of final grasps obtained from optimized postures by closing each finger until motion is stopped by contact with the object.

for hand postures that match the shape of a grasped object. However, a significant amount of computational effort was dedicated to optimizing extrinsic DOFs (wrist position and orientation, six variables) versus intrinsic DOFs (eigengrasp amplitudes, two variables). As the focus has been on dimensionality reduction for the intrinsic DOF domain, no attempt has been made to simplify the extrinsic DOF search domain. For fully autonomous grasp synthesis this is a necessary compo-

nent: a correct finger posture is only relevant when combined with an appropriate wrist position relative to the target object.

An important category of grasp planning applications that does not require complete autonomy stems from the field of neural hand prosthetics. Such devices combine a degree of human control with artificial hardware and algorithms. Formalizing this concept using our framework means that an external operator can specify desired values for some, but not all, of

Table 2. Number of form-closed grasps obtained from 20 pre-grasps found in a two-dimensional eigengrasp space (average and standard deviation over five executions for each hand and object).

	Ashtray		Shoe		Glass		Flask		Phone		Plane	
	Average	SD	Average	SD	Average	SD	Average	SD	Average	SD	Average	SD
Barrett	2.8	2.2	1.0	1.0	5.8	5.3	3.4	1.7	1.4	2.1	4.0	3.1
DLR	11.0	3.6	6.0	3.4	0.8	0.8	3.2	2.7	2.8	3.6	1.8	0.8
Robonaut	7.0	2.3	9.0	1.7	14.4	3.7	14.4	3.7	10.0	2.8	3.6	2.3
Human	14.6	2.3	11.0	2.5	11.2	1.6	13.4	3.5	8.4	2.3	1.8	1.3

the variables that define a grasp. For example, Taylor et al. (2002) have enabled a primate to directly control the linear velocity of the endpoint of a robot arm through three DOFs in real time. This control was achieved by measuring the activity of individual cortical neurons that correspond to individual preferred directions of each neuron in space. The vector sum of preferred directions of a population of neurons, each scaled by their individual unit activity, provided the velocity of robotic end-effector movement.

In contrast, controlling finger posture has proven to be significantly more difficult. A number of possible approaches are described in the literature, including electromyography (Zecca et al. 2002) and cortical implants (Taylor et al. 2003). These studies have shown success in decoding a limited number of information channels, therefore controlling a highly dexterous hand for interactive grasping remains an open and challenging problem. In this study we propose a grasp planning method that combines the eigengrasp framework for reducing the dimensionality of the hand configuration space with real-time operator input simplifying the spatial components of the search. Our goal is to enable an operator to complete dexterous grasping tasks with limited direct control over finger posture.

4.1. System Overview

In our current implementation, the user provides on-line information on the position and orientation of the wrist. This data is currently provided using a six-DOF magnetic tracker. While we have not yet integrated this component in a real prosthetic system, it is our directional goal; we envision that hand position information will by extracted from cortical activity in a similar fashion to the primate study described above. In contrast, finger posture is entirely controlled by the automatic component, which selects an appropriate hand shape by combining information about the geometry and pose of the target object with the position input provided by the operator. The only additional information needed from the user is a binary "click" command for completing a grasp, which we describe below.

It is important to note that our approach must compensate for the lack of complete user grasp data by using knowledge of the target object geometry, as well as its initial position relative to the hand. In previous work (Kragic et al. 2001), we have shown that it is possible to perform grasp planning using a vision-based system for object recognition and localization. Compared with the optimization method presented in the previous section, this system also has to satisfy two important criteria: first, the output has to be in the form of explicit form-closure grasps rather than optimized pre-grasps; second, solution grasps must be found at a fast enough rate to enable on-line interaction with the operator and usage of real-time input. The execution of a grasping task must therefore be performed at a speed that approaches natural human behavior, of the order of seconds as opposed to minutes.

A high-level overview of the complete system and the interaction with the operator is provided in Figure 7. The planning algorithm runs on the *GraspIt!* simulator platform. Even though the grasp planner runs in a simulated environment, the results can be applied to a real robotic hand, allowing the user to interact with the hand directly and use it to pick up surrounding objects. The simulator receives user input and sends it to the grasp planner which processes it and outputs potential grasps, which are in turn used to generate commands for the robotic hand. The operator can hold the hand and approach the target object; the position of the hand relative to the target is tracked using a Flock of Birds (Ascension Corp., VA) magnetic tracker. We have also applied our method on a range of more complex hand designs (including the DLR hand, the Robonaut hand as well as the human hand model) using the virtual environment in *GraspIt!*; the operator can change the position of the virtual wrist by directly manipulating the magnetic tracker. In both cases, wrist position is supplied as the input to the grasp planner, but the operator has no direct control over finger posture.

4.2. User Interaction with Grasp Planner

In general, in order to uniquely identify a grasp, three variables are needed to specify the position of the wrist and three more for its orientation. In the context of our application, we expect the user to specify a desired approach direction to the target; however, the presence of such external input does not fully

Fig. 8. Search directions defined for the Barrett and human hand models. The direction of the vector *d* is pre-defined relative to the palm. Its magnitude, as well as the values of the angles θ and ϕ are variables defining a conical search area.
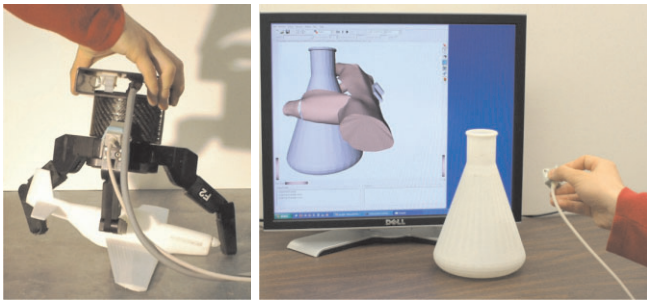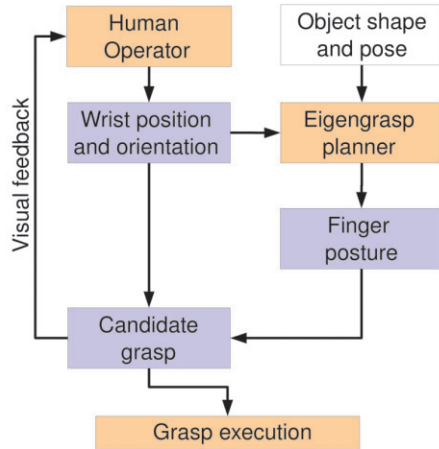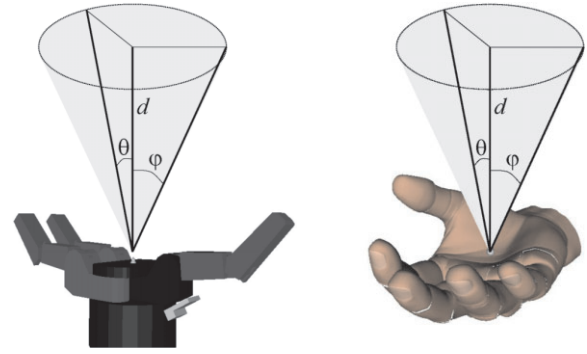
Fig. 7. Interactive grasp planning using wrist position input from a human operator. Top: system overview; Bottom: applied examples using a real Barrett hand (left) and a dexterous hand in a simulated environment (right).

eliminate the spatial component of the grasp planning search. First, it is not practical to wait until the user has brought the wrist into a final grasping position before starting the search for an appropriate finger posture, as this behavior would decrease the interactivity of the system. Rather, it is preferable to start the search early, and attempt to predict where the user intends to place the wrist. Second, this prediction allows the system to offer feedback to the user: as soon as an anticipated grasp is found, the grasp planner can shape the fingers accordingly. The user can then decide whether the grasp is satisfactory and either continue towards the target or choose another approach direction if the system is unable to find an acceptable solution.

This behavior can be implemented efficiently by re-parameterizing the spatial component of the grasp planner as shown in Figure 8. For each hand model, we define a preferred search direction *d* based on the kinematics of the hand, usually normal to the palm. Then, starting from a hand position specified by the operator, we search for good grasps in a conical region around the search direction using three variables: the distance |*d*| along the approach direction, as well as two angular variables, θ and ϕ. The operator is instructed to ap-

proach the object along a direction that is generally similar to the search cone; however, the search directions are defined in order to make this a natural choice. In the examples in Figure 8 this means that the user is asked to keep the palm approximately facing the target, as opposed to other possibilities such as a sideways or backwards approach.

The role of this parameterization is to reduce the number of extrinsic DOFs that are used for grasp planning, focusing on areas where good grasps are most likely to be found. Using this heuristic, the search will automatically ignore states where, for example, the hand is facing away from the target object. However, the user is not expected to specify an exact wrist position for a good grasp; by searching along the approach direction *d* the planner attempts to anticipate the intended final grasp. The angular variables θ and ϕ allow the planner to compensate for noisy measurements in the intended wrist position, and allow for more flexibility in the search for solution grasps. By adding these three variables to the eigengrasp amplitudes describing hand posture, we obtain a low-dimensional domain that can be searched fast enough to respond to on-line changes in the wrist position input provided by the human operator.

4.3. Quality Function Formulation using Scaled Contact Wrench Spaces

When the posture optimization algorithm is used for on-line grasping tasks, we use a formulation of the quality function Q that is better adapted for interactive operation. Recall that, in the form presented in Section 3, our formulation rewards hand postures that bring all of the fingers, as well as the palm, as close to the surface of the object as possible. For the application presented here, it is necessary to also reward hand postures that create stable, but not necessarily enveloping grasps (consider as an example the case of a fingertip pinch grasp applied on a thin object). We therefore propose an alternative quality
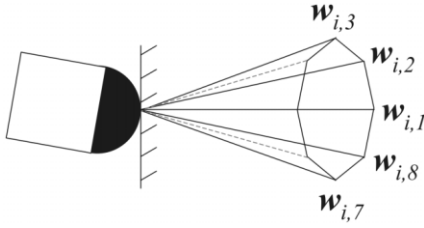
Fig. 9. Contact wrench space example using a Coulomb friction cone.

Fig. 10. Multiple contact wrench spaces, each scaled based on the contact distance metric δ_i.

function which is fast to compute and can assess the *potential* quality of a pre-grasp posture using a modified version of the grasp wrench space (GWS) ϵ metric introduced by Ferrari and Canny (1992). A detailed description of this metric in its original form is beyond the scope of this paper; we provide a brief overview below, and for further details we also refer the reader to the study by Miller and Allen (1999).

For each contact i, we assume that the space of forces and torques that can be transmitted is bounded by the convex hull of a finite set of six-dimensional wrenches $w_{i,j}$ where $1 \leq j \leq k$. The convex hull of these wrenches forms the contact wrench space. We note that this approach imposes a linear form for what are normally quadratic friction constraints; for example, in the case of Coulomb friction, the force components of $w_{i,j}$ sample the contact friction cone (Figure 9), and the respective torque components are null. In order to define the GWS, the contact wrenches from all contacts are first expressed relative to a common coordinate system. This coordinate system is usually anchored at the center of mass of the object and the choice of axes directions is arbitrary. We denote the matrix that transforms a wrench from the local coordinate system of contact i to the global object coordinate system by $R_i \in \mathcal{R}^{6 \times 6}$.

In our implementation, we are usually assessing the quality of a pre-grasp shape where the fingers are not in contact with the target. Therefore, we assume that the hand can apply *potential* contact wrenches using the desired contact locations shown in Figure 4. When computing the GWS, we scale the potential wrenches at each desired contact proportional to the inverse of the distance metric δ_i computed as in (5):

$$\text{GWS} = \text{ConvexHull} \left\{ \bigcup_{\substack{\text{all desired} \\ \text{contacts}}} (1 - \delta_i) R_i \bigcup_{j=1}^{k} w_{i,j} \right\}. \quad (7)$$

Thus, if the value of δ_i is small, the contact will have a significant contribution to the GWS, and states that bring it closer to the object surface will be rewarded with a higher quality value. If, in contrast, the desired contact is far from the object, it will not significantly affect the grasp quality measurement. If the contact is far enough from the object so that its correspond-
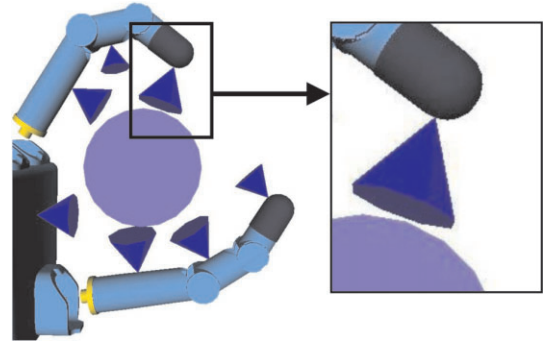
ing weight of $1 - \delta_i$ is negative, it is completely excluded from the computation.

After building the GWS, we compute the ϵ quality measure as described by Ferrari and Canny (1992) and Miller and Allen (1999). The quality of the grasp is considered equal to the radius ϵ of the largest six-dimensional ball, centered at the wrench space origin, that can be enclosed within the GWS. If $\epsilon = 0$, then the origin itself is not contained in the hull, and the grasp does not have form-closure. For $\epsilon > 0$, the grasp can resist any disturbance, and the maximum magnitude of the contact forces needed to resist a disturbance is inversely proportional to ϵ.

The process is illustrated in Figure 10 for the DLR hand grasping a disk. In this example, each contact is modeled by a friction cone, approximating Coulomb friction for rigid bodies, but other local contact models can also be used. For example, the ability to create stable, encompassing grasps with subsets of fingers is increased by using soft fingertips that deform during contact. In addition to tangential friction, such contacts can also apply frictional torque. The friction cone is thus replaced by a four-dimensional "friction ellipsoid" which constrains the relationship between tangential force and frictional torque (Howe and Cutkosky 1996). This effect can be captured by linearizing the friction ellipsoid and using the appropriate contact wrenches $w_{i,j}$, as shown by Ciocarlie et al. (2007). This method enables the use of rubber-coated fingertips for our robotic hands, without compromising the accuracy of the grasp quality computations.

4.4. Computation of Form-closure Grasps

The automated grasp planner searches for solution grasps in two stages. The first stage is the posture optimization algorithm presented in Section 3, using the quality function formulation described above. For interactive tasks, each run of the simulated annealing algorithm is performed over 2,000
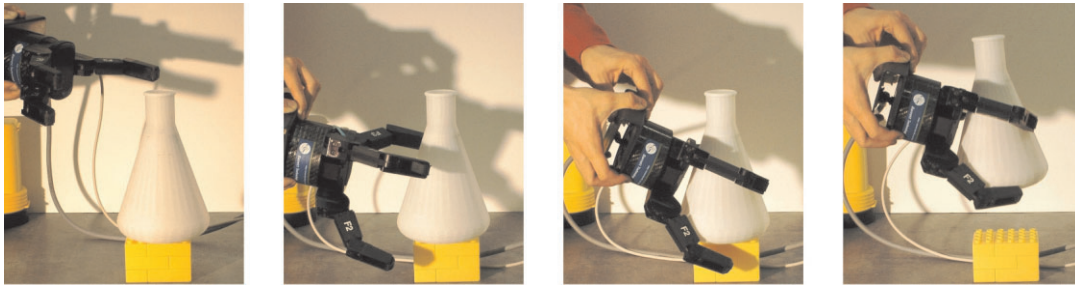
Fig. 11. Example of a complete grasping task: initial approach, finger pre-shaping using grasp planning result, continued approach and final grasp execution.

iterations, taking advantage of the fact that the search domain is five-dimensional (two eigengrasp amplitudes and three wrist position/orientation variables), as opposed to the eight-dimensional domain used for fully autonomous searches. After reaching this number of iterations, the search is restarted by resetting the annealing temperature. As a result, the planner does not get stuck if one particular search fails; rather, the search is restarted and takes advantage of any changes in the approach direction provided by the operator.

The user-specified reference wrist position is updated continuously during the search. The results of the optimization are therefore always relative to the current position of the wrist. However, we recall that the low-dimensional optimization procedure can still only produce *pre-grasp* shapes; in order for the system to allow successful completion of the task, *final* grasping postures satisfying the form-closure requirement are necessary. In order to achieve interactive rates, this expensive computation is only performed using the best pre-grasps found during each run of the annealing optimization, which are queued and sent to the second stage of the planning process.

For each candidate pre-grasp resulting from the first stage, we use the contact detection engine within *GraspIt!* to compute the final grasp that results by closing the fingers on the object. Once the contacts between the hand and the object have been determined, we compute the exact quality value of the final grasp using the GWS ϵ quality in its original form presented by Ferrari and Canny (1992). If the grasp is found to have form-closure, it is saved, along with its associated quality value, as a potential solution, and used by the next component of the system, which is responsible for interaction with the human user.

When computing the final grasping posture resulting from a candidate pre-grasp, we take into account specific mechanical properties of the hand, such as passive mechanical adaptation to the shape of the target. A number of robotic hands, such as the Barrett hand, the SDM hand (Dollar and Howe 2007) and the CyberHand (Carrozza et al. 2006) rely on passive mechanical adaptation, as it significantly increases grasp stability without increasing the complexity of the control mechanisms. All

of the results involving the Barrett hand presented in this paper take into account its adaptive actuation mechanism which allows distal joints to close even when proximal joints controlled by the same motor have been stopped due to contact.

In our implementation, the two planning phases described in this section (simulated annealing search for pre-grasps and final grasp testing for form-closure) run in separate threads. As soon as a candidate pre-grasp is found, it is queued for testing, but the search for new candidates continues independently of the testing phase. Also, candidate pre-grasps are independent of each other, and can be tested simultaneously. This parallelism allows us to take advantage of the current evolution in multi-core architectures, largely available on standard desktop computers.

We can now provide a complete step-by-step walkthrough of a grasping task that combines user input and automated grasp planning. Figure 11 shows the execution of a grasp proceeding through the following stages.

- As the user approaches the target object, the grasp planner searches for a good grasp in a cone-shaped area around the given approach direction. When a solution is found, it is used to set the hand posture, allowing the user to react. If multiple solutions are found, that which is closest to the current user approach direction is chosen for presentation (i.e. the solution with the lowest values for the angular variables θ and ϕ).

- The planner continuously attempts to improve the current result, by finding new grasps that are closer to the current position established by the user.

- If the planner is unable to find a grasp in the current search area, or if the user is not satisfied with the resulting hand posture, the user can reposition the hand and attempt to grasp a different part of the target object.

- If the user is satisfied with the hand posture, they continue along the current approach direction. As the real hand position approaches the target grasp, the fingers are
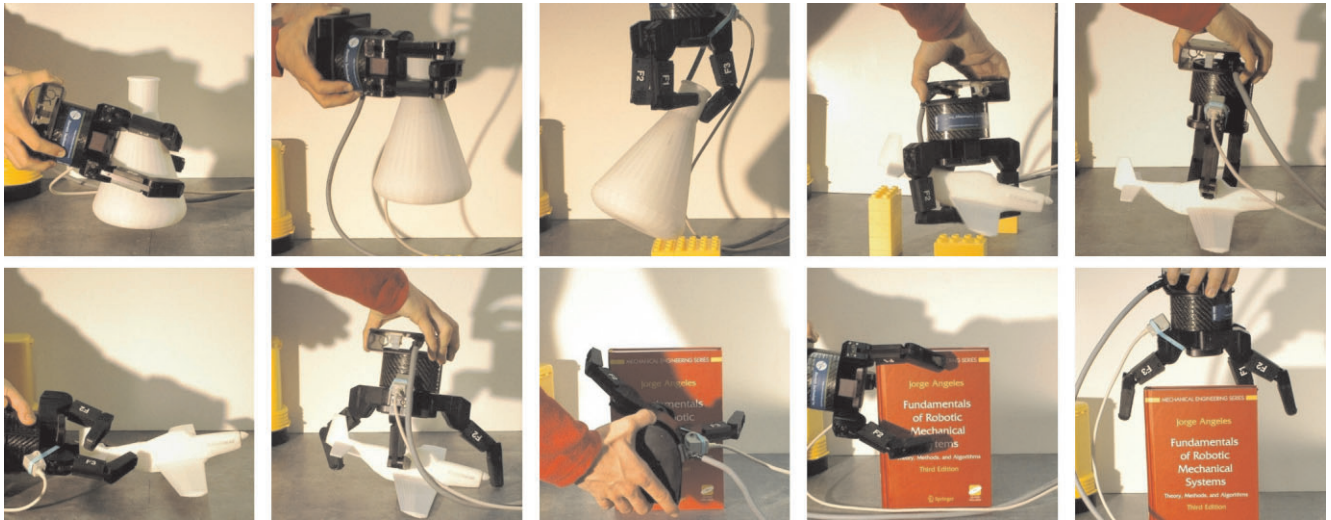
Fig. 12. Examples of interactive grasping tasks; each image shows the grasp found for a different approach direction or target object. In all cases the object was successfully grasped and lifted off the table.

Fig. 13. Examples of interactive grasping tasks executed in simulated environments. Bottom row images also show the user providing the approach direction via a magnetic tracker. All of the presented grasps have form-closure.

gradually closed around the object. The user can therefore predict where the object will be touched and finally issue a "close all fingers" command which completes the grasping task.

4.5. Results

Figure 12 presents the application of our method using the Barrett hand in a real environment, while Figure 13 shows interactive grasps performed in a simulated environment using the

DLR hand, the Robonaut hand and the human hand model. In most cases, the images show only the final grasp applied by the user; owing to space constraints we are unable to include images showing the evolution of the grasping task from approach direction, pre-grasp and final grasp. In order to better evaluate the interactive nature of our application, a video clip showing a number of complete examples is provided in Extension 1.

For any given grasping task, the exact computational effort required to find a stable grasp depends on the complexity of the hand and target object, as well as the approach direction

chosen by the user. On average, the first stage of the grasp planning algorithm processes approximately 1,000 hand postures per second, while the second testing phase, running in parallel, can evaluate approximately 20 candidate pre-grasps per second. In most cases, solution grasps are found at interactive rates: in the example presented in Figure 11, the grasp planner found eight stable grasps in 13.6 seconds of computation. These are representative numbers for the behavior of the system, which generally requires less than 2 seconds to find a solution grasp for a new approach direction. All of our tests were performed using a commodity desktop computer equipped with a 2.13 GHz Intel Core2 CPU.

The ability of the system to allow for successful task completion in a short time is more difficult to quantify, as it also depends on how well the user reacts to the behavior of the automated components. All of the results presented in Figures 12 and 13, as well as in the video presented in Extension 1, were obtained at interactive rates, usually requiring between 5 and 15 seconds from first approach to final grasp execution. For the more difficult tasks, taking up to 30 seconds to complete, we found two main reasons that led to the increased execution time: either the planner repeatedly failed to find form-closure grasps for selected approach directions, or the human user could not interpret some of the finger postures selected by the planner and had to attempt different grasps. These cases represent a small minority of our tests and examples; however, the tests were performed by well-trained users familiar with the inner workings of the planning algorithm.

As a next development step, we intend to test our system in user studies with untrained subjects. This study will allow us to quantify more precisely how the interaction paradigm that we have chosen affects user experience. Informal responses from our current experiments showed that the attempts of the planner to anticipate where the operator intended to place the hand, and thus shape the fingers accordingly, were occasionally unsuccessful. In such cases, even if the planner succeeded in finding a stable grasp, the user did not execute it and rather attempted a different approach direction. This communication channel can potentially be improved in both directions, by providing the operator with more clues about the results of the planner as well as more means to influence its behavior. We believe that such features must complement improvements to the core planning algorithm itself, as we progress towards a prosthetic system that can be deployed in the real world.

5. Conclusions

In this paper we have proposed the use of a low-dimensional subspace of the hand DOF space for finding hand postures appropriate for a given task. Using quantitative data derived from human studies we have defined such a control subspace for grasping common objects. We have introduced the concept of *eigengrasps* as the defining dimensions of this subspace and have extended this framework for a number of robotic hand designs, with the complete set of hand models used in this study ranging from 4 to 20 intrinsic DOFs. As long as the eigengrasp space provides a good approximation of the hand motion required for a given task, algorithms can be designed to operate in this space and take advantage of the dimensionality reduction. We have presented a low-dimensional hand posture optimization method applied for stable grasp synthesis; the results show that, while not containing exact grasping postures, a two-dimensional eigengrasp space can serve as an effective pre-grasp or planning space even for highly dexterous hand models.

The eigengrasp space also acts as an interface between the kinematic structure of the hand and higher-level task planning. Therefore, for a given task, it is possible to use a unified treatment for a number of robotic hand models, even though the kinematic specifications may be significantly different. Owing to the dimensionality reduction of the configuration space, it allows also algorithms for complex dexterous hands to be used in applications that require fast computational rates. One such example, arising in the field of hand neuroprosthetics, is an interactive grasping task where a human operator controls the position and orientation of the wrist, but has no direct control over finger posture. We have presented a grasp planner that interfaces between the operator and the artificial hand by selecting appropriate finger postures fast enough for on-line interaction.

In order to achieve interactive rates, this search is separated into two processes: the first finds a small number of optimized pre-grasps in a low-dimensional eigengrasp subspace, while the second one processes these results and computes the quality of the final grasping positions. Our interaction method allows the user to provide wrist position and orientation input to the grasp planner, effectively guiding the search in a small region around a desired approach direction. We have tested this system in both real and simulated environments and have presented results involving a number of different hand models and objects.

In the work presented here we have used control subspaces derived from user studies on human grasping, mapped empirically to robot hand kinematics. Instead, we would like to be able to compute the *optimal eigengrasp subspace* for a given robotic hand model and task. One possible approach is to perform a dense sampling of the high-dimensional control space of the robotic hand, then find the low-dimensional decomposition that contains most of the desirable hand postures. The sampling process for the hand configuration space can be performed off-line, therefore computational restrictions can be somewhat relaxed. However, in the case of very complex hands with 20 or more intrinsic DOFs, this task is intractable even with an off-line assumption. Such cases will require a novel approach and will be the subject of future research.

Acknowledgement

Appendix: Index to Multimedia Extensions

The multimedia extension page is found at http://www.ijrr.org

Table of Multimedia Extensions

Extension	Type	Description
1	Video	Examples of grasping tasks executed using our interactive planner

References

Aleotti, J. and Caselli, S. (2006). Grasp recognition in virtual reality for robot pregrasp planning by demonstration. *IEEE International Conference on Robotics and Automation*, Orlando, FL, pp. 2801–2806.

Bicchi, A. and Kumar, V. (2000). Robotic grasping and contact: a review. *IEEE International Conference on Robotics and Automation*, pp. 348–353.

Brown, C. and Asada, H. (2007). Inter-finger coordination and postural synergies in robot hands via mechanical implementation of principal components analysis. *IEEE-RAS International Conference on Intelligent Robots and Systems*, pp. 2877–2882.

Butterfass, J., Hirzinger, G., Knoch, S. and Liu, H. (1998). DLR's articulated hand, part I: hard- and software architecture. *IEEE International Conference on Robotics and Automation*, pp. 2081–2086.

Carrozza, M. C., Cappiello, G., Micera, S., Edin, B. B., Beccai, L. and Cipriani, C. (2006). Design of a cybernetic hand for perception and action. *Biological Cybernetics*, **95**(6): 629–644.

Cheung, V. C. K., d'Avella, A., Tresch, M. C. and Bizzi, E. (2005). Central and sensory contributions to the activation and organization of muscle synergies during natural motor behaviors. *Journal of Neuroscience*, **25**(27): 6419–6434.

Ciocarlie, M., Lackner, C. and Allen, P. (2007). Soft finger model with adaptive contact geometry for grasping and manipulation tasks. *Joint Eurohaptics Conference and IEEE Symposium on Haptic Interfaces*, pp. 219–224.

Cipriani, C., Zaccone, F., Stellin, G., Beccai, L., Cappiello, G., Carrozza, M. and Dario, P. (2006). Closed-loop controller for a bio-inspired multi-fingered underactuated prosthesis. *IEEE International Conference on Robotics and Automation*, pp. 2111–2116.

Cutkosky, M. R. (1989). On grasp choice, grasp models, and the design of hands for manufacturing tasks. *IEEE Transactions on Robotics and Automation*, **5**: 269–279.

Dollar, A. and Howe, R. (2007). Simple, robust autonomous grasping in unstructured environments. *IEEE International Conference on Robotics and Automation*, pp. 4693–4700.

Edsinger, A. and Kemp, C. (2006). Manipulation in human environments. *IEEE-RSJ International Conference on Humanoid Robotics*, pp. 102–109.

Ferrari, C. and Canny, J. (1992). Planning optimal grasps. *IEEE International Conference on Robotics and Automation*, pp. 2290–2295.

Fod, A., Mataric, M. and Jenkins, O. (2002). Automated derivation of primitives for movement classification. *Autonomous Robots*, **12**: 39–54.

Howe, R. and Cutkosky, M. (1996). Practical force–motion models for sliding manipulation. *The International Journal of Robotics Research*, **15**(6): 557–572.

Hsiao, K., Kaelbling, L. and Lozano-Perez, T. (2007). Grasping POMDPs. *IEEE International Conference on Robotics and Automation*, pp. 4685–4692.

Iberall, T. (1997). Human prehension and dexterous robot hands. *The International Journal of Robotics Research*, **16**: 285–299.

Ingber, L. (1989). Very fast simulated re-annealing. *Journal of Mathematical and Computer Modelling*, **12**(8): 967–973.

Kragic, D., Miller, A. and Allen, P. (2001). Real-time tracking meets online planning. *IEEE International Conference on Robotics and Automation*, Seoul, pp. 2460–2465.

Li, Y., Fu, J. L. and Pollard, N. S. (2007). Data-driven grasp synthesis using shape matching and task-based pruning. *IEEE Transactions on Visualization and Computer Graphics*, **13**(4): 732–747.

Lovchik, C. S. and Diftler, M. A. (1998). The Robonaut hand: a dextrous robot hand for space. *IEEE International Conference on Robotics and Automation*, 907–912.

Mason, C. R., Gomez, J. E. and Ebner, T. J. (2001). Hand synergies during reach-to-grasp. *Journal of Neurophysiology*, **86**: 2896–2910.

Miller, A. and Allen, P. (1999). Examples of 3-D grasp quality computations. *IEEE International Conference on Robotics and Automation*, Detroit, MI, pp. 1240–1246.

Miller, A. and Allen, P. K. (2004). *GraspIt!*: a versatile simulator for robotic grasping. *IEEE Robotics and Automation Magazine*, **11**(4): 110–122.

Miller, A. T., Knoop, S., Christensen, H. I. and Allen, P. K. (2003). Automatic grasp planning using shape primitives. *IEEE International Conference on Robotics and Automation*, pp. 1824–1829.

Napier, J. R. (1956). The prehensile movements of the human hand. *Journal of Bone and Joint Surgery*, **38**: 902–913.

Platt, R., Fagg, A. H. and Grupen, R. (2002). Nullspace composition of control laws for grasping. *IEEE International Conference on Robotics and Automation*, Washington, DC, pp. 1717–1723.

Platt, R., Fagg, A. H. and Grupen, R. (2004). Manipulation gaits: sequences of grasp control tasks. *IEEE International*

Conference on Robotics and Automation, New Orleans, LA, pp. 801–806.

Roa, M. and Suarez, R. (2007). Geometrical approach for grasp synthesis on discretized 3D objects. *IEEE-RSJ International Conference on Intelligent Robots and Systems*, pp. 3283–3288.

Santello, M., Flanders, M. and Soechting, J. F. (1998). Postural hand synergies for tool use. *Journal of Neuroscience*, **18**(23): 10105–10115.

Santello, M., Flanders, M. and Soechting, J. F. (2002). Patterns of hand motion during grasping and the influence of sensory guidance. *Journal of Neuroscience*, **22**: 1426–1435.

Saxena, A., Driemeyer, J. and Ng, A. (2008). Robotic grasping of novel objects using vision. *The International Journal of Robotics Research*, **27**(2): 157–173.

Shimoga, K. B. (1996). Robot grasp synthesis algorithms: a survey. *The International Journal of Robotics Research*, **15**: 230–266.

Taylor, D. M., Tillery, S. H., and Schwartz, A. B. (2002). Direct cortical control of 3D neuroprosthetic devices. *Science*, **296**(5574): 1829–1832.

Taylor, D., Tillery, S. H. and Schwartz, A. (2003). Information conveyed through brain control: cursor versus robot. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, **1**(2): 195–199.

Thakur, P. H., Bastian, A. J. and Hsiao, S. (2008). Multidigit movement synergies of the human hand in an unconstrained haptic exploration task. *Journal of Neuroscience*, **28**(6): 1271–1281.

Todorov, E. and Ghahramani, Z. (2004). Analysis of the synergies underlying complex hand manipulation. *26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 4637–4640.

Tsoli, A. and Jenkins, O. C. (2007). 2D subspaces for user-driven robot grasping. *Robotics, Science and Systems Conference: Workshop on Robot Manipulation*.

Turk, M. and Pentland, A. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, **3**(1): 71–86.

Vande Weghe, M., Rogers, M., Weissert, M. and Matsuoka, Y. (2004). The ACT hand: design of the skeletal structure. *IEEE International Conference on Robotics and Automation*, pp. 3375–3379.

Zecca, M., Micera, S., Carrozza, M. C. and Dario, P. (2002). Control of multifunctional prosthetic hands by processing the electromyographic signal. *Critical Reviews in Biomedical Engineering*, **30**: 459–485.