The Columbia Grasp Database

Corey Goldfeder, Matei Ciocarlie, Hao Dang and Peter K. Allen

Abstract— **Collecting grasp data for learning and benchmarking purposes is very expensive. It would be helpful to have a standard database of graspable objects, along with a set of stable grasps for each object, but no such database exists. In this work we show how to automate the construction of a database consisting of several hands, thousands of objects, and hundreds of thousands of grasps. Using this database, we demonstrate a novel grasp planning algorithm that exploits geometric similarity between a 3D model and the objects in the database to synthesize form closure grasps. Our contributions are this algorithm, and the database itself, which we are releasing to the community as a tool for both grasp planning and benchmarking.**

I. INTRODUCTION

Dexterous robotic grasping has been an active research subject for decades, and many grasp planning algorithms have been proposed. However, very few grasp planning algorithms are data driven, as collecting training data for robotic grasping is difficult. In this paper we introduce the Columbia Grasp Database, a freely available collection of hundreds of thousands of form closure grasps for thousands of 3D models. Our primary interest is in using an object's 3D geometry as an index into the database. Given a new 3D object, we can find geometric neighbors in the database, and the accompanying stable grasps for these similar objects. If the number of objects to be grasped in the database is very large and comprehensive then robotic grasping becomes a pre-computed database lookup. While we have not yet achieved this level of performance it is our directional goal.

The most direct way to construct a grasp database is to collect grasping data from human volunteers. We could gather a large set of example objects, outfit an army of graduate students with grasp-capture devices such as datagloves, and record the results. Unfortunately, this approach is prohibitively time consuming for large scale data acquisition. More importantly, data collection from humans can *only* produce grasps with the human hand. Since many popular robotic hands cannot be easily mapped to the human hand, a useful database should include grasps with multiple hands.

The Columbia Grasp Database was created using *GraspIt!*, a publicly available grasp planning and analysis tool developed by our group [12]. The database is intended to be used in conjunction with *GraspIt!* or a similar simulation tool; as we have shown in previous work [10], [3], planning results obtained in simulation can be successfully applied to real robotic hands performing grasping tasks.

Even for grasp planning algorithms that do not rely on simulation, an environment such as *GraspIt!* is an important tool for evaluation, as grasp quality measures are generally impossible to compute in physical experiments. Part of our motivation in producing a grasp database was to provide a benchmark for robotic grasping tasks. Using a common benchmark will make it possible to directly compare grasp planning algorithms, which is currently difficult to do.

Another contribution of this paper is a database-backed grasp planning algorithm based on the data we have collected. Using this algorithm, we illustrate the usefulness of a database for grasping, and highlight some of the lessons learned during its construction. We also provide execution results over the entire set of objects in the database at their primary scale. We believe this to be one of the most comprehensive tests found in the grasp planning literature, demonstrating the use of the database as a benchmarking tool.

In sum, our grasp database provides a number of new tools for the general robotics community, including:

- baseline grasps for a very large number of objects.
- a set of scaled 3D models for grasp benchmarking.
- a large labeled dataset for machine learning of robotic grasping.

We are not aware of any previous attempt to construct a large scale grasp database, or of any commonly used benchmarks for evaluating robotic grasping. However, researchers have investigated grasp planning approaches that assume such a database already exists. Bowers and Lumia [2] collected grasps for a small number of planar objects and used fuzzy logic to extrapolate grasping rules. Morales *et al.* [14] used *GraspIt!* to compute offline grasps for a small database of graspable objects, and successfully identified and executed those grasps on real objects in a complex environment. Unlike the planner we present in this paper, their approach requires an exact model of every possible graspable object.

Other researchers have experimented with different forms of precomputed grasp knowledge. Li and Pollard [11] collected a database of 17 hand poses, and used shape matching to match poses to graspable objects. Their work highlighted the difficulty of automatically generating grasps that are both stable and plausibly humanlike. Aleotti and Caselli demonstrated grasp synthesis using examples acquired with a dataglove [1]. Saxena *et al.* [18] generated 2D renderings of a large set of example objects, and learned a model-free mapping from images to graspable features. Their work uses a two fingered gripper and does not have an obvious extension to dexterous hands.

All authors are with the Dept. of Computer Science, Columbia University, NY, USA. Email: {coreyg, cmatei, allen}@cs.columbia.edu, hd2181@columbia.edu

This work was funded in part by NIH BRP grant 1RO1 NS 050256-01A2, and by a Google research grant.

II. Building a Grasp Database

Although building a grasp database by direct data collection is infeasible, the basic strategy of grasping many objects and recording the grasp poses is still valid. However, it requires replacing human input with an automated procedure that does not need user attention. In our own previous work [13], [8] we have demonstrated the use of a grasp planner running in a simulation engine. Grasp planning can be considered an optimization task that can be performed in simulation without user supervision. However, even when simulations are performed on a powerful computer, the space of possible grasps is too large to sample directly. This is particularly true in the case of dexterous hands with many intrinsic degrees of freedom (DOFs).

A. Eigengrasp planner

Recently, we have made substantial progress in reducing the number of DOFs in grasping tasks by defining *eigengrasps*, a low-dimensional grasping subspace that has shown promise in allowing online computation and sampling of the high dimensional grasping parameter space. Eigengrasps can reduce the number of DOFs from over twenty in the case of multi-fingered anthropomorphic hands to as few as two. It thus becomes feasible to stochastically sample this low-dimensional grasping parameter space using simulation techniques, and determine a wide range of stable, form closure grasps on complex objects even for dexterous hand designs.

In previous work we presented a grasp planning algorithm that optimizes hand posture in a low dimensional eigengrasp space to find effective pre-grasp postures. These pre-grasps are then heuristically developed into stable, form closure grasps. The core of this algorithm is a simulated annealing optimization, and since this is a stochastic method, we can use multiple runs to find different form closure grasps for the same model. For a complete description of this planner as well as discussion of the eigengrasp concept, we refer the reader to [4].

Our goal in this paper is to apply this planner to a very large set of objects, and obtain multiple results for each object. We therefore use a slightly modified multithreaded version which takes advantage of multi-core architectures widely available today on commodity computers. A single parent thread searches the eigengrasp space for likely pre-grasps; for each pre-grasp position that crosses a quality threshold, a separate child thread is created to refine it. The child thread performs a quick local simulated annealing search with finer step values, attempting to develop the pre-grasp into form closure. If the resulting grasp indeed has form closure, the pre-grasp and grasp are both saved in the result list.

After creating a child thread for a pre-grasp state, the parent thread's state generation function rejects states close to the child thread's search area, forcing it to look elsewhere in the state space for new grasps. The planner can be restarted at any point by resetting the annealing temperature. The process continues until either the desired number of grasps are found, or a pre-set time limit is exceeded.

B. Constructing the Database

One of the main advantages of the grasp planner discussed above is its flexibility: it can be successfully applied to a wide range of both object models and robotic hands. As eigengrasp definitions encapsulate the kinematic characteristics of each hand design, the planner can operate on eigengrasp amplitudes and thus ignore low-level operations and concentrate on the high-level task. The reduced dimensionality framework also enables the use of anthropomorphic models with more than 20 intrinsic DOFs. The planner can thus operate identically across different hand models, without any change or parameter tuning. In addition, the planner makes no assumptions regarding the nature of the target object and can operate on a wide range of 3D models. These characteristics make it a strong candidate for large scale application: once the desired object set and hand models have been chosen, the batch execution of the planner over the chosen set requires little, if any, user attention. For each hand-object combination, the output is a list of stable grasps, each with an associated pre-grasp in eigengrasp space. For each grasp, the planner also computes two widely used quality metrics that characterize the Grasp Wrench Space (GWS): the epsilon and volume quality metrics introduced by Ferrari and Canny [6].

A crucial aspect of the database construction is the running time required for each hand-object combination. As the planner has no intrinsic time limit, one has to be imposed based on the size of the object set and the computing power available. When building the database described in the next section, each execution of the planner ran until we found 15 form closure grasps of the target object. In general, this required about 10 minutes of run time. When dealing with large datasets and different hand models, a subset of objects will inevitably prove difficult to grasp using our algorithm. To prevent these from dominating the computation time we also set an upper time limit of 20 minutes per model.

The total number of planner executions used to build the database was approximately 22,000, as described in following section. The total computation time was approximately 1 month on 6 multi-core computers.

III. The Columbia Grasp Database

In this section we describe the contents and structure of the Columbia Grasp Database. The database is freely available at grasping.cs.columbia.edu both as a PostgreSQL database and as flat text files.

A. 3D Models and Scale

The first requirement for a grasp database is a set of objects to grasp. Rather than distributing a new set of 3D models, we chose to reuse the models from the Princeton Shape Benchmark (PSB) [19], which is already in common use in the shape matching community. We hope that a shared dataset will encourage increased collaboration between shape researchers and roboticists.

The disadvantage of this choice is that the PSB models were not originally selected with an eye towards robotic grasping, and so some of the models are not obvious choices for grasping experiments. For example, the database contains both airplanes and insects, all of which are outside the normal grasping range of a human-sized hand. We chose to treat all such "ungraspable" objects as toys, and rescaled them accordingly. The rescaling factor for each model, as compared with the "original" scale in the standard PSB, is included in the database.

Even with all of the models at graspable size, the issue of scale required further attention. Grasping is inherently scale-dependent, but most of the models in the PSB might plausibly exist anywhere within a small range of scales. This was particularly true for the models rescaled to "toy" size. To soften the impact of scale, we cloned each object at four distinct scales, 0.75, 1.0, 1.25 and 1.5, where 1.0 represents the rescaled size from above. As the PSB contains 1,814 distinct models, this left us with a total of 7,256 models to grasp.

For each model we store the radius of a ball that approximately contains the model. We use an approximately enclosing ball rather than an absolutely enclosing ball to avoid outlier sensitivity. We assume that all of the points on a model's surface are normally distributed around some fixed radius from the center or mass. The fixed radius is simply the mean distance from the surface to the center of mass, and the "approximate radius" stored in the database is the mean distance to the surface plus two standard deviations of the distance to the center once the mean has been subtracted.

Along with shape and scale, the space of possible grasps is influenced by the frictional properties and deformability of both the hand and the object. Both in the construction the database and in our experiments we treated all models as being made of rigid plastic. The properties of the different hands are described below.

B. Robotic Hands

Grasping is strongly hand-dependent, and so we need to specify the hands used in our database. For the first version of the database, we chose to focus on two hands; a human hand model in order to emphasize the "humanlike" nature of the grasp selection, and the three-fingered Barrett hand, which is ubiquitous in robotics research due to its durability and relatively low cost. The human hand model has 20 degrees of freedom. The Barrett hand has 4 degrees of freedom, plus a disengaging clutch mechanism which allows conformance even when the proximal link of a finger is blocked. Both models are available for download with *GraspIt!*.

Frictional forces play an important role in grasping, and so we must specify the materials for each of our hands. There is no exact consensus on the friction coefficient of human skin and so we chose $\mu = 1.0$ as a plausible value for the friction between the human hand and plastic [20]. The ability to create stable, encompassing grasps with subsets of fingers is also increased by using soft fingertips that deform during contact and apply a larger space of frictional forces and moments than their rigid counterparts. In order to take into account such effects, we use a fast analytical model for soft finger contacts that we have introduced in previous work [5]. The Barrett hand is made of metal, but can be coated with a higher friction material. We created two versions of the Barrett hand, one uncoated and one with rubberized fingers, and computed grasps for then independently, in effect giving us three hand models. For the metal Barrett hand we used $\mu = 0.4$ and for the rubber coated version we used $\mu = 1.0$. As the kinematic models are identical, grasps computed for either Barrett model can be executed using the other, making it possible to evaluate the advantage afforded by using the higher friction material. We note that grasps from the regular hand can be assumed to be form closure for the rubberized hand as well, but that this guarantee does not hold in reverse.

In the future we intend to augment these hands with other models, such as the DLR hand and the Robonaut hand. We encourage researchers with novel hands to send us models, both for inclusion with *GraspIt!* and for use in future revisions of the database.

C. Grasps and Pre-Grasps

Our grasp database is intended to be used in conjunction with *GraspIt!* or a similar grasp simulation tool. As such, we provide the necessary data to recreate each grasp, in the form of joint angles and hand position, and the contact points between hand and object, which can be used as a check to ensure that the grasp was simulated correctly. We also provide the two measures of grasp quality mentioned in Section II-B.

Each grasp entry consists of a *pre-grasp* and the final grasp pose. A pre-grasp is a pose from the instance before the hand contacts the object; it represents "pure" grasp information untainted by conformance to an exact object. Each of our pre-grasps lies within the two dimensional Eigengrasp subspace, as described in [4]. In contrast, the grasp poses represent final positions for form closure. This first version of our database contains 238,737 distinct form closure grasps, comprising 25,585 grasps with the metal Barrett hand, 132,421 grasps with the rubber Barrett hand and 80,731 grasps with the human hand. Along with each grasp we record a pregrasp pose and the epsilon and volume quality measures of [6].

D. Caveats

Since the grasps in the database were found using an automated planner, not all of the grasps are truly humanlike or reliable. There can be cases where a grasp satisfies our quality metrics, but would require a degree of precision that cannot be obtained in real-life execution. Aside from the intrinsic limitations of grasp quality metrics, for which there is as of yet no firm consensus on which to use, our approach to grasp planning is purely geometric. This presents problems for objects that do not match our assumptions. For example, our assumption that all objects are rigid plastic results in geometrically correct but unrealistic grasps on objects such as flowers or leaves. Furthermore, the lack of domain-specific knowledge means that some of our grasps are semantically incorrect, such as a mug grasped by placing the fingers inside, although they are still form closed.

Finally, all of our grasps were obtained from pre-grasps that sample a low-dimensional subspace of the hand DOF space. This is for the moment a necessary simplification, without which the planning problem for dexterous hands is intractable at this scale. While our choice of subspace is theoretically justified and shown to be effective [4], we cannot reasonably claim that the database covers the entire space of possible grasps. The choice of optimal subspace is one of our directions for future research.

IV. Databased-Backed Grasp Planning

One of our primary motivations of building a grasp database was to collect enough grasping data to build new grasp planners based on learning. In this section we present a grasp planner that uses a k-Nearest-Neighbors approach to find candidate grasps for a model not in the database. In general, the relation between hand pose and grasp quality for a given object is both nonlinear and discontinuous, and more sophisticated learning methods such as SVMs have so far been shown to work only for simple objects [16], [2]. We hope that the data we have collected will facilitate further research in this direction.

A. Algorithm

Our grasp planning algorithm is based on the intuition that similar objects are likely to have similar grasps. Therefore, if we wish to grasp an object not in our database, it makes sense to look at the most similar objects that *are* in the database and to attempt to grasp the new object in the same way.

Given a model to grasp α, we use a shape matching algorithm to find $N = \{n_1 ... n_k\}$, the k models in the database most similar to α under some shape similarity metric. In this paper we used L^2 distances between Zernike descriptors [15], which we have previously shown to be scalable to very large libraries of 3D models [7]. Zernike matching is scale-normalized, but as detailed in Section III-A, each PSB model exists in our database at 4 distinct scales. For each n_i we consider up to 2 models, $n_i^<$, the largest neighbor smaller than α and $n_i^>$, the smallest neighbor larger than α, using the scaled approximate radius described in Section III-A. In the case of α smaller or larger than all 4 versions of the neighbor we only used one model for n_i.

We present our grasp planning algorithm here. For simplicity, we have ignored the issue of scale and treated each n_i as a single model.

Algorithm 1 Database-Backed Grasp Planning

```
Require: Model $\alpha$ to grasp with hand $H$, using $k$ neighbors
  and quality threshold $\tau$.
  $N \leftarrow$ NearestNeighbors$(\alpha, k)$
  $R \leftarrow \{\}$ {The resulting form closure grasps.}
  for all $n_i \in N$ do
      Align $\alpha$ and $n_i$ using PCA.
      Co-locate the centers of mass of $\alpha$ and $n_i$.
      $G \leftarrow$ the precomputed pre-grasps on $n_i$.
      for all $g \in G$ do
          Transform $g$ to $\alpha$'s coordinate system.
          Place $H$ in the position and configuration of $g$.
          repeat
              Move $H$ backwards.
          until NoCollisions(Palm$(H), \alpha)$
          for all $finger \in$ Fingers$(H)$ do
              repeat
                  Open $finger$.
              until NoCollisions$(finger, \alpha)$
          end for
          repeat
              Move $H$ forwards.
          until InContact$(H, \alpha)$
          CloseFingers$(H)$
          if EpsilonQuality$(H, \alpha) > \tau$ then
              $R \leftarrow R \cup g$
          end if
      end for
  end for
  return $R$
```

The entire process, from shape matching through final output, takes approximately 20 seconds. To illustrate the behavior of this algorithm, we provide a number of examples in Fig. 1.

B. Experiments

For our experiments, we removed each PSB model at scale 1.0 from the database one at a time and attempted to grasp it using only the known grasps from the remaining models. To isolate the effects of shape matching, we used three methods of choosing similar models. In each case we used $k = 5$ neighbors for every model.

Our first method used the ground-truth labels provided with the PSB. For each model, we chose neighbors within the same shape category, starting with the finest categorization and moving up to coarser categories if fewer than k neighbors were available. Within the same category the choice of neighbors was arbitrary. We designated the chosen models as the '*PSB classes*' neighbors. This method of indexing, while not usable for arbitrary unclassified models, approximates the performance of a theoretical ideal[1] shape matching algorithm that has perfect precision and recall over the PSB.

Our second method used L^2 distances between Zernike descriptors [15]. For each model, we designated the k models with the smallest L^2 distance in Zernike descriptor space as the '*zernike*' neighbors. These descriptors are computed on voxel grids and are quite robust, making them suitable for use in matching newly acquired objects into the database.

[1] Even with perfect precision and recall, this theoretical algorithm may not truly be 'ideal', as the categories in the PSB are semantic rather than purely geometric. Nevertheless, since shape matching algorithms are regularly evaluated using these categories as a ground truth, we adopt the same convention.

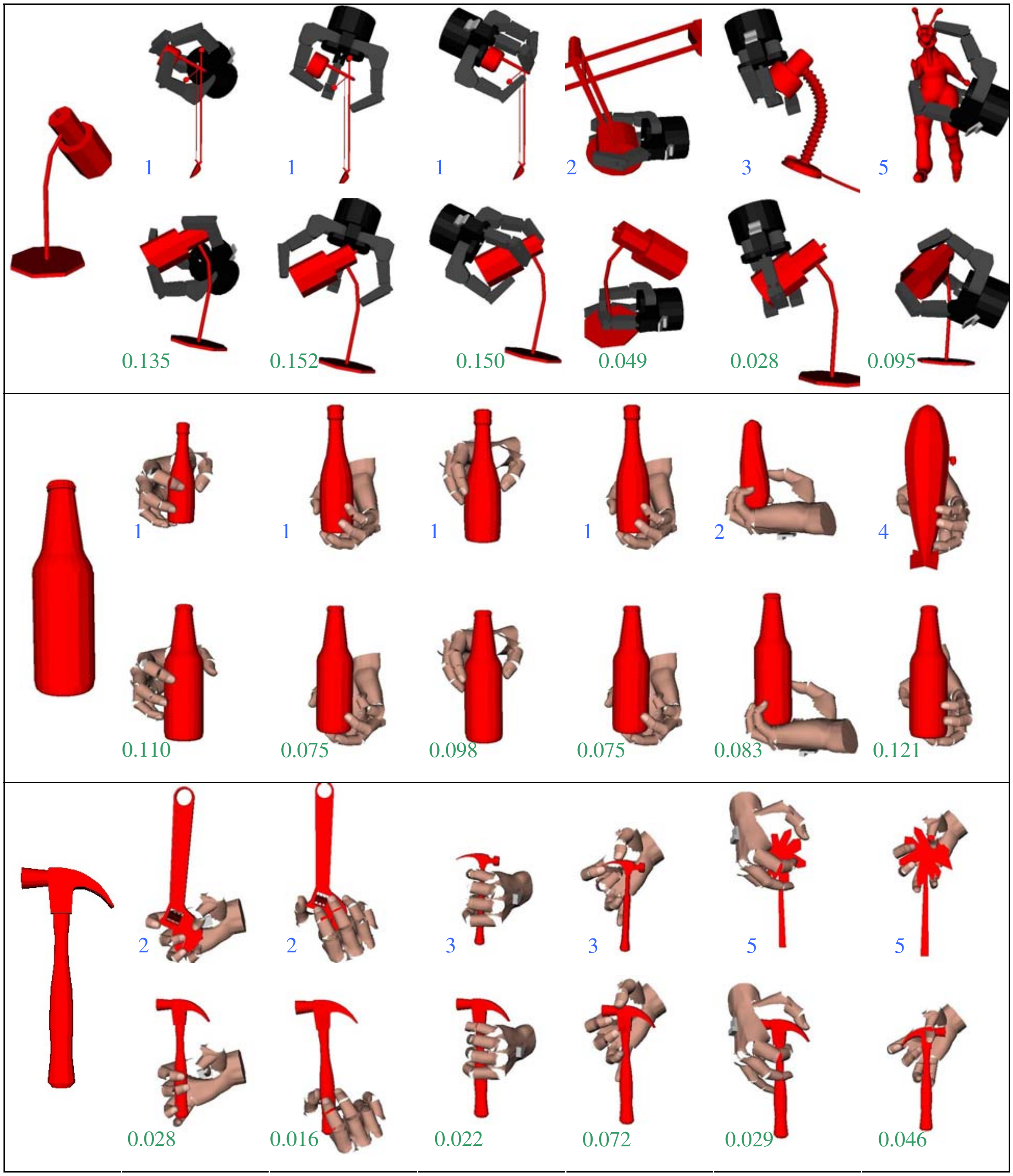
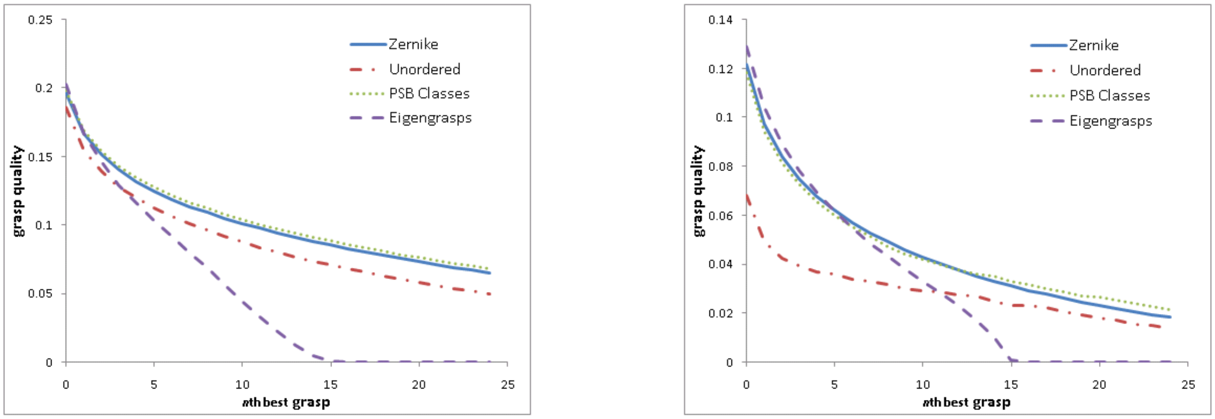


Fig. 1. Three example models and their grasps, using the database-backed planner with Zernike neighbors. For each model α (left), the top row of images shows a neighbor n_k from the database, the value of k, and a pre-computed grasp on that neighbor. Directly below each neighbor is the same grasp executed on α, along with its GWS epsilon quality measure.

For our third method, we randomly selected k models from the database and designated them as '*unordered*' neighbors. We emphasize that these are *not* random grasps in any sense; α has been translated and axis-aligned with some model of a similar scale. Furthermore, the pre-grasps applied to it are pointed in the right direction, with joint angles drawn from a high quality eigengrasp subspace and known to produce form closure on another model with aligned principal axes. We therefore expect that some of the pregrasps taken from unordered neighbors will result in form closure grasps. Our aim in using the unordered neighbors is to isolate the performance gains based on shape matching while holding constant the performance due to the overall high quality of all of the grasps in the database.

We ran the experiment separately for each type of neighbor selection and averaged the grasp quality of the *n*th best grasp on each model over all 1,814 models in the database at scale 1.0. As mentioned in the introduction, we believe



Barrett hand

Human hand

Fig. 2. The *n*th best grasp from database-backed grasping with 3 neighbor selection methods and from the eigengrasp planner, averaged over the 1,814 models in the database at scale 1.0. As can be seen, the information provided by shape matching is more important for the complex human hand than it is for the clawlike Barrett hand. This figure is best viewed in color.

this to be one of the most comprehensive grasp planning experiments in the literature, as it consists of thousands of runs on a highly varying set of objects. We can analyze these results from a number of different perspectives: the absolute performance of database-backed grasp planning, the relative behavior of different neighbor selection methods and finally the performance compared to running the eigengrasp planner of Section II-A directly on the target object. Fig. 2 shows these results for the human hand and the Barrett hand.

Although Zernike descriptors do not have perfect precision and recall over the PSB, their performance for our grasp planner is nearly identical to the ground truth PSB classification. This is likely because the PSB classification is partially semantic, whereas our grasp quality measures are purely geometric; a ‘poor’ neighbor for semantic shape matching may still be geometrically close enough to share high quality grasps.

The performance of the ‘unordered’ neighbors is as expected; good, due to the general quality of grasps in the database and the axis-alignment between α and the unordered neighbor, but not as good as the shape matching methods. Although the improvement due to shape matching is small for the Barrett hand, for the human hand the difference is quite significant. We attribute this difference to the many additional DOFs of the human hand, which creates a need for careful pre-grasping. The Barrett hand, with its 4 degrees of freedom, has a far simpler configuration space, and the importance of pre-grasping is correspondingly less.

Of special interest is the comparison between the database-backed methods and the eigengrasp planner. For the first few grasps, the performance of the shape matching methods is essentially identical to that of the eigengrasp planner. However, for subsequent grasps the quality quickly diverges, with the advantage going to the database-backed methods. This is even more impressive when we recall that the eigengrasp planner ran for approximately 10 minutes per model, whereas the database-backed planners ran for about 20 seconds. The database-backed approach can take advantage of pre-computed grasp data from multiple objects, essentially extracting the useful information obtained from several runs of the eigengrasp planner.

C. Planning for Real Objects

The ultimate goal of our database-backed planner is to grasp new objects that are not in our database using sensor data. While a comprehensive evaluation using such acquired data is left for future work, we present here our preliminary results. Using a commodity desktop 3D scanner, we acquired a range image of a plastic toy. Due to the intrinsic limitations of the acquisition method, the range image was both noisy and incomplete, with several occlusions. We computed the Zernike descriptor of the scan, found the nearest neighbors in the database, and ran the planner as before for both the Barrett and human hands. We found 88 form closure grasps using the human hand and 112 form closure grasps using the Barrett hand. Some of these results are shown in Fig. 3. In the future we intend to experiment with more scanned objects, and to validate the planner output by executing the grasps with a physical hand on the real object.

V. CONCLUSIONS AND FUTURE WORKS

We have created the Columbia Grasp Database, a large collection of scaled models ($n = 7,256$) and grasps ($n = 238,737$) for several hands which we are making available to the robotics community. Building the database was a non-trivial exercise, and at present, no other grasp database of comparable scale is available. We believe this to be an important first step in building a valuable tool for researchers developing grasp synthesis algorithms, particularly those pursuing machine learning approaches to grasping.

In addition to providing this data, we also demonstrated its application as both a back-end and benchmark tool for a novel, data-driven grasp planner. Our experiments showed this approach to be efficient at obtaining good form closure grasps. In the process, we gained some new insights into grasp planning. We found that a good choice of grasp subspace, combined with axis-alignment, produced grasps

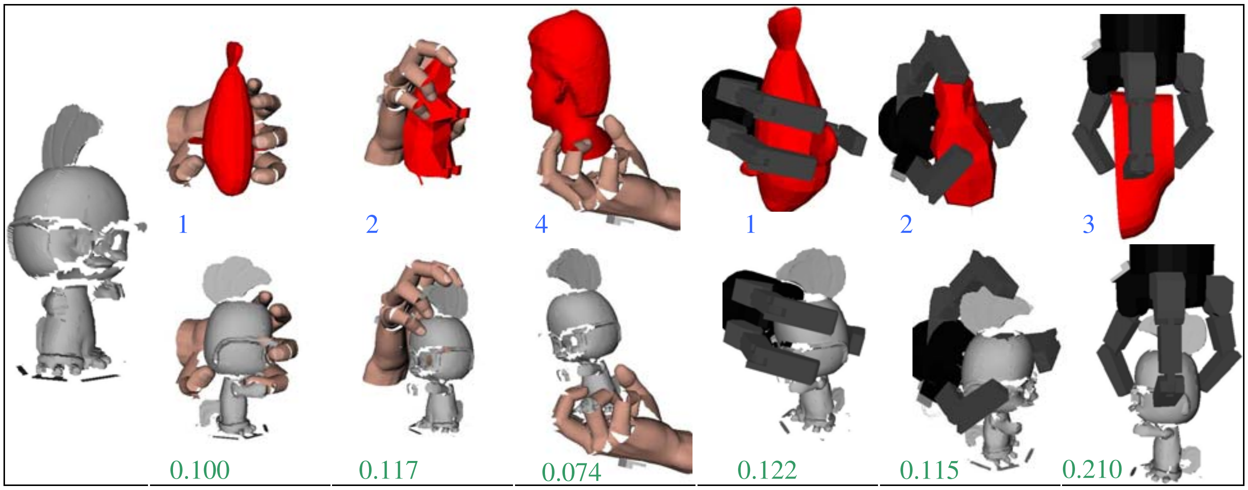


Fig. 3. Some of the grasps planned for an acquired object with holes and occlusions, using the database-backed planner with Zernike neighbors. In total 88 form closure grasps were found for the human hand and 112 for the Barrett hand.

that could often be transferred to new objects even in the absence of strong geometric correlation. As expected, however, using geometric neighbor selection produced better results, particularly in the case of the high DOF human hand. Furthermore, the database approach outperformed the single object planner of [4], showing that the database is effective at distilling planning results from multiple objects into a compact form.

In many cases the geometric similarity provided by the Zernike and PSB classes neighbors was not fully exploited due to poor alignment between a model and its neighbors. We believe that the shape matching approach would benefit from using an alignment method more powerful than global PCA, which is known to be suboptimal for aligning 3D models [9]. Better alignment might be obtained using a different global transform, such as the Principal Symmetry Axes transform [17] or by using a pairwise alignment method such as Iterative Closest Point.

Our database consists predominantly of complete 3D models. However, in real-life applications 3D sensor data such as laser scans or stereo reconstructions often exhibits holes or occlusions. Our preliminary results showed the database-backed approach to be applicable to acquired scan data, as an imperfect model can still be matched against the models in the database. In future work we intend to evaluate the performance of our planner more fully on sensed data and real robotic grasping applications with real hands. We also intend to augment the grasp database with additional hands and additional 3D models.

References

[1] J. Aleotti and S. Caselli. Robot grasp synthesis from virtual demonstration and topology-preserving environment reconstruction. In *Intl. Conf. on Intelligent Robots and Systems*, 2007.

[2] David L. Bowers and Ron Lumia. Manipulation of unmodeled objects using intelligent grasping schemes. *Transactions on Fuzzy Systems*, 11(3), 2003.

[3] Matei Ciocarlie and Peter K. Allen. On-line interactive dexterous grasping. In *Eurohaptics*, 2008.

[4] Matei Ciocarlie, Corey Goldfeder, and Peter K. Allen. Dimensionality reduction for hand-independent dexterous robotic grasping. In *International Conference on Intelligent Robots and Systems*, 2007.

[5] Matei Ciocarlie, Claire Lackner, and Peter K. Allen. Soft finger model with adaptive contact geometry for grasping and manipulation tasks. In *World Haptics*, 2007.

[6] C. Ferrari and J. Canny. Planning optimal grasps. In *Intl. Conf. on Robotics and Automation*, pages 2290–2295 vol.3, 1992.

[7] Corey Goldfeder and Peter K. Allen. Autotagging to improve text search for 3d models. In *Joint Conference on Digital Libraries*, 2008.

[8] Corey Goldfeder, Peter K. Allen, Claire Lackner, and Raphael Pelossof. Grasp planning via decomposition trees. In *Intl. Conf. on Robotics and Automation*, 2007.

[9] Michael Kazhdan, Bernard Chazelle, David P. Dobkin, Adam Finkelstein, and Thomas A. Funkhouser. A reflective symmetry descriptor. In *European Conf. on Computer Vision*, 2002.

[10] Danica Kragic, Andrew Miller, and Peter K. Allen. Real-time tracking meets online planning. In *Intl. Conf. on Robotics and Automation*, 2001.

[11] Ying Li and Nancy S. Pollard. A shape matching algorithm for synthesizing humanlike enveloping grasps. In *Humanoid Robots*, 2005.

[12] Andrew Miller, Peter K. Allen, V. Santos, and F. Valero-Cuevas. From robot hands to human hands: A visualization and simulation engine for grasping research. *Industrial Robot*, 32(1), 2005.

[13] Andrew T. Miller, S. Knoop, H. I. Christensen, and Peter K. Allen. Automatic grasp planning using shape primitives. In *Intl. Conf. on Robotics and Automation*, 2003.

[14] Antonio Morales, Tamim Asfour, Pedram Azad, Steffen Knoop, and Rudiger Dillmann. Integrated grasp planning and visual object localization for a humanoid robot with five-fingered hands. In *Intl. Conf. on Intelligent Robots and Systems*, 2006.

[15] Marcin Novotni and Reinhard Klein. 3D zernike descriptors for content based shape retrieval. In *Solid Modeling and Applications*, June 2003.

[16] Rafael Pelossof, Andrew Miller, Peter Allen, and Tony Jebara. An SVM learning approach to robotic grasping. In *Int. Conf. on Robotics and Automation*, 2004.

[17] Joshua Podolak, Philip Shilane, Aleksey Golovinskiy, Szymon Rusinkiewicz, and Thomas Funkhouser. A planar-reflective symmetry transform for 3d shapes. In *SIGGRAPH*, 2006.

[18] Ashutosh Saxena, Justin Driemeyer, and Andrew. Robotic grasping of novel objects using vision. *Intl. Journal of Robotics Research*, 27(2), 2008.

[19] Philip Shilane, Patrick Min, Michael Kazhdan, and Thomas Funkhouser. The princeton shape benchmark. In *Shape Modeling and Applications*, 2004.

[20] Raja K. Sivamani, Jack Goodman, Norm V. Gitis, and Howard I. Maibach. Coefficient of friction: tribological studies in man an overview. *Skin Research and Technology*, 9(3), 2003.