An Object-Based, Hand-Independent Algorithm for Creating Teleoperation Mappings

Cassie Meeker[1], Maximilian Haas-Heger[1], and Matei Ciocarlie[1]

As human-robot collaboration becomes more prevelant, and as the physical barriers between humans and robots begin to disappear, it becomes easier for humans to provide guidance to robots while simultaneously collaborating with them. This is useful because human-in-the-loop teleoperation can often deal with corner cases faster than automated programs in unstructured environments. There exist a variety of teleoperation controls which harvest the motions of the user's hand to provide intuitive, efficient, and safe control interfaces for users. However, the field is lacking in intuitive, novice-friendly ways to create these teleoperation controls.

To create a teleoperation control, one must create a teleoperation mapping, which tells the robot hand how to move in response to movements of the human hand. We would ideally like to avoid using hand specific intuition from the user when creating new teleoperation mappings for new robots. When the human and robot hands are very different, creating mappings which compensate for kinematic differences often requires a great deal of work, computation and human intuition to create and test. This is a significant barrier for novices and non-roboticists who wish to work with robotic hands.

To address this gap, we create an algorithm that generates teleoperation mappings between human and robot hands without requiring intuition at a hand specific level. We have previously shown that there is a subspace which is relevant to teleoperation [1]. We hypothesize that this subspace can be described by the objects for which it enables grasping. In this work, we present a set of objects which describe the teleoperation subspace. We create teleoperation mappings by fitting a subspace to hands grasping these object sets. Creating mappings in this manner does not require any hand specific intuition.

Teleoperation subspace defined by an object set: We define a set of objects consisting of 17 shape primitives. We hypothesize that these objects define a three-dimensional teleoperation subspace. While a hand of any kinematic configuration is grasping an object in the set, it lies at the same point in the teleoperation subspace as the object (Figure 1).

The fact that a hand grasping an object in the set lies at the same point in the subspace as the object itself is key to our method. It implies that we can find a model of the subspace defined by the hand joint angles by fitting a three dimensional space to a grasp set where the hand is holding each of the objects. The model of the subspace, in turn, allows us to project between hand joint space and the teleoperation subspace.

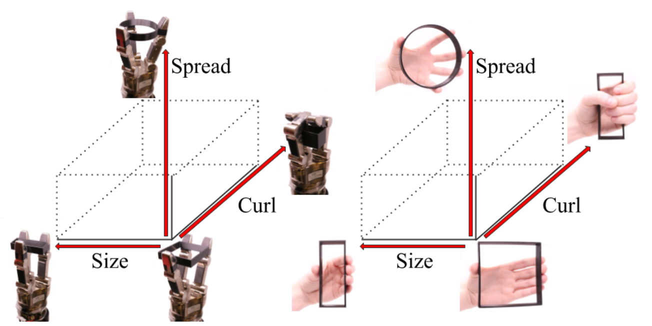


Fig. 1. The Schunk SDH and human hand grasping sample objects, which places them at specific locations in the subspace.

Our object set was designed based on a previous teleoperation subspace we have shown to be relevant for teleoperation [1]. The basis vectors of this subspace are defined by vectors along certain hand motions - hand open/close, finger spread, and finger curl. Our object set is designed to elicit similar motions when grasped. We note that we specify the type of grasp (power or precision) with which each object must be grasped when designing the set. Design of the object set requires human intuition, but once it is designed, creating a teleoperation mapping for each new hand does not require any additional human intuition.

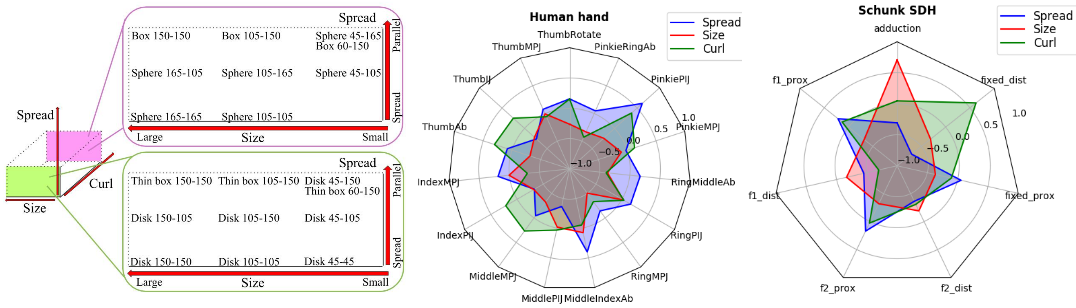
Our object set's shape primitives include discs, thin boxes, boxes and spheres. Figure 2a shows the object names and where each object is placed in the subspace. Our naming convention lists the shape primitive, the width in millimeters, then the length of the object. The object height is 30mm for discs and thin boxes, and the minimum value between the length and width for boxes and spheres. The approach direction of the hand is along the height dimension, and we orient the objects in the same way relative to the hand.

Grasp Generation: For each hand, we generate a set of grasps that illustrates how the hand stably holds each object.

For robotic hands, we use a grasp planner provided along with the *GraspIt!*simulator. Given a hand and an object, the planner returns up to 1000 stable grasp configurations for that object, ranked by the epsilon quality metric. We parse the dataset by removing duplicate grasps which are closer than 0.1 in Euclidean distance to a higher ranking grasp. The threshold is increased in intervals of 0.1 and the dataset reparsed until each object has fewer than 20 grasps remaining.

* This work was supported in part by the ONR Young Investigator Program award N00014-16-1-2026.

[1]Department of Mechanical Engineering, Columbia University, New York, NY 10027, USA.
{c.meeker, m.haas, matei.ciocarlie}@columbia.edu



(a) (b)

Object	Average Time (Seconds)
Box	20.2 $\pm$ 3.0
Ball	14.1 $\pm$ 1.6
Legos	14.49 $\pm$ 2.4
Apple	12.6 $\pm$ 3.0
Tape	15.0 $\pm$ 2.7
Drill	26.7 $\pm$ 18.8

(c)

Fig. 2. (Left) objects placed in the teleoperation subspace, (middle) teleoperation mapping which our algorithm created for the human hand and the Schunk SDH and (right) average time and standard deviation to pick and place each object during the experiments in seconds.

We note that this means all the objects will not have the same number of grasps in the final dataset.

For the human hand, we generate the dataset using test subjects. Subjects were asked to don a instrumented data-glove (a Cyberglove III) and grasp scaled versions of the object set. Our object set was developed for hands like the Schunk SDH, which has fingers approximately 1.5 times the size of the human hand. We therefore divided the dimensions of the objects by 1.5 in all directions to create the human object set. After the subjects grasp a given object stably, their joint angles are collected from the Cyberglove. We collected grasps from 10 subjects.

Creating a Teleoperation Mapping from Object Grasps: Once we generate the dataset, we fit a subspace to the grasps in order to find the teleoperation mapping between the hand and the subspace. We fit the grasps with RANSAC. RANSAC typically consists of 3 steps - selecting a random hypothesis (in this case, points which form the three vectors of the subspace), comparing the hypothesis model to the best model thus far, and replacing the best model with the hypothesis model if the hypothesis model does better. We perform RANSAC on both hands simultaneously, because the nature of the teleoperation mapping requires that the subspace vectors for both hands point in the same directions so that the hands move in the same way during teleoperation.

To select the random sample set, first, we select an origin grasp randomly. We then pick the next three grasps from objects which lie at the same location in the subspace as the origin object except for along one of the basis vectors. The basis vectors of our subspace model are formed by subtracting the origin grasp from the three other grasps. The origin and the basis vectors define our hypothesis subspace. We choose the samples for the master hand randomly, but constrain the samples for the second hand to be chosen from the same objects as the first hand. To compare the hypothesis model to the best model we compare, in order of importance, the number of inliers for each object, the number of objects which have the minimum number of inliers, the total number of inliers for all objects and the sum of the distances between all the grasps and the subspace. The hypothesis model only replaces the best model if scores for both of the hands improve or scores for one of the hands improves and the score for the other hand remains the same.

RANSAC outputs two subspace models, one for each of the hands. Our teleoperation mapping requires a projection matrix, a scaling factor and an origin [1]. The basis vectors of the subspace model give us the projection matrix and the subspace origin is the mapping origin as well. The scaling factor can be iteratively determined based on joint maxima and minima. In this way, the subspace models allow us to generate teleoperation mappings which project between the joint spaces of the two hands and the teleoperation subspace.

Experiments: We used our algorithm to create teleoperation mappings for the human hand and the Schunk SDH hand. We teleoperate by projecting from the joint space of the human hand to the teleoperation subspace, and then projecting from the teleoperation subspace to the joint space of the Schunk SDH. Figure 2b shows the mapping generated by the algorithm for both the human hand and the robot hand.

We have previously shown that a teleoperation subspace defined by human intuition is relevant to teleoperation. We wished to show that the teleoperation subspace created by our algorithm is similarly relevant to teleoperation. We asked an expert user to use the mapping generated by our algorithm to pick and place six objects - a ball, a box, a roll of tape, a stack of legos, an apple and a drill. The user picked up and moved each object five times using visual feedback.

To teleoperate, we mounted a Schunk SDH on a Sawyer arm. The arm is teleoperated by a Cartesian controller guided by a magnetic tracker on the back of the user's hand. Arm control is completely separate from the hand control.

We report the average amount of time it took to pick and place each object and the standard deviation in Figure 2c. The pick and place experiments show that the subspace defined by the object set is relevant to teleoperation because our expert user was able to pick and place objects.

Our algorithm requires no hand specific intuition and still creates a subspace relevant to teleoperation which can be used for pick and place tasks. This helps novices without any prior robotic knowledge to create intuitive teleoperation mappings for new hands with which they wish to work.

References

[1] C. Meeker, T. Rasmussen, and M. Ciocarlie, "Intuitive hand teleoperation by novice operators using a continuous teleoperation subspace," in *Robotics and Automation, 2018. Proceedings. ICRA'18. 2018 IEEE International Conference on*, 2018.