Efficient Models for Grasp Planning With A Multi-fingered Hand

Jean-Philippe Saut, Daniel Sidobre

LAAS-CNRS, FRANCE

IROS 2010
Workshop on Grasp Planning and Task Learning by Imitation
Taipei, October, 18th 2010
Introduction

- Goal: computing a dense re-usable grasp set for a given multi-fingered hand and object.
- Applications:
  - Manipulation task planning (pick-and-place, dual-hand manipulation)
  - Interactive grasping (HRI)
Computing a set of grasps vs computing a unique (“optimal”) grasp

Pros

- The choice of a grasp is very context-dependent and this context is susceptible to changes:
  - During pick-and-place tasks, the objects are moved by the robot.
  - In HRI tasks, the human may move the objects.
- Re-usable grasp sets allow back-tracking during planning of complex manipulation tasks.

Cons

- Can be computationally expensive.
Definitions

We define a grasp by:

- A transform between the object and the hand palm *Grasp Frame*.
- A set of joint parameters for each finger $i$: $\theta^i_1, \theta^i_2, \ldots, \theta^i_n$.
- A set of contact points ($p_1, p_2, \ldots$), on the fingertips, that can be deduced from the two previous items.
Grasp List Computation

Method overview:

1. **Object Model**
   - Object Partitioning

2. **Hand Model**
   - Finger Workspace Approximation

3. **Grasp Frames Sampling**
   - Grasp Frames Set

4. **Grasp Computation**
   - Collision + Stability Filter

5. **Quality Score Computation**
   - Quality-ordered Grasp Set
Grasp Frame (Hand Pose)

Sampling

Uniform sampling of frames (position + orientation):

- Center the frame roughly where the contact may occur.
- Input $\leftarrow$ number of positions $n_p$, number of orientations $n_o$
- Positions: uniformly sampled in the object’s axis-aligned bounding box (with a step computed to fit $n_p$).
- The $n_o$ first elements of an incremental grid ([Yershova 2004])
Grasp Computation From Grasp Frame

Finding the contact for a given hand pose.

Two approaches:

- **Forward kinematics**: close the fingers until contact occur. Usually requires many collision tests. Can not find contact in loops (e.g. mug handle).

- **Inverse kinematics**: compute the points on object surface that are reachable by the fingers. Exact solutions is computationally very expensive.
Grasp Computation From Grasp Frame

→ introduce models to quickly find a conservative approximation of the accessible part of the object’s surface

→ find intersection of a surface (object) and a volume (finger workspace)

- Model of object’s surface.
- Model of finger workspace.
Object Surface Model

The object surface is approximated with a point set:

- The points are obtained by uniform sampling of the object’s 3D model (triangle mesh).
- The sampling step magnitude is chosen from the fingertip radius.
- Local information about the surface is stored with each point (surface normal and curvature).

A space-partitioning tree is built upon the point set in order to have a hierarchical space partition of the points.
Object Surface Points Kd-tree

Recursive subdivision of point set bounding-boxes. Each bounding-box is splitted in two along its larger dimension until each node contain only one point.
Finger Workspace Model

Volumetric approximation based on a sphere hierarchy.

**Figure:** Schunk Anthropomorphic Hand (4 joints/3 DOFs per finger).
Finger Workspace Model

Construction

From forward kinematics, build two point sets:

- $W =$ Points strictly inside the workspace $\leftarrow$ sampling each joint parameter over $[\theta_{\min}; \theta_{\max}]$.
- $E =$ Points on the workspace boundary $\leftarrow$ set a joint angle to $\theta_{\min}$ or $\theta_{\max}$ and sample the other ones over $[\theta_{\min}; \theta_{\max}]$. 
Finger Workspace Model
Construction

**input**: $W$= a set of points strictly inside the finger workspace ;
$E$= a set of points on the envelope of the finger workspace ;
$k_{\text{max}}$= the desired maximal size of the sphere decomposition ;
$r_{\text{min}}$= the desired minimal sphere radius ;

**output**: $S$= a set of spheres $S_k$ ordered from the biggest to the smallest ;
$S = \emptyset$ ; $k = 1$ ;

while $k < k_{\text{max}}$ do

theforeach $p \in W$ do

\[ d(p) = \min_{p_i \in (E \cup S)} (\|p - p_i\|) ; \]

$p_{\text{best}} = \{p \in W : d(p) = \max_{p_i \in W}(d(p_i))\}$

$S_k = \text{sphere}(center = p_{\text{best}} , radius = d(p_{\text{best}}))$ ;

$S = S \cup S_k$ ;

$W = W - \{p \in W : p \subset S_k\}$ ;

$k = k + 1$ ;

if $d(p_{\text{best}}) < r_{\text{min}}$ then

break ;

return $S$ ;
Intersection between object surface and finger workspace

The two hierarchies are tested from their respective roots. This requires two elementary operations:

- box-sphere intersection
- point-sphere inclusion
Intersection between object surface and finger workspace

Points in the intersection are reachable but can lead to collisions.
Intersection between object surface and finger workspace

\begin{verbatim}
foreach finger_i ∈ [1; nbFingers] do
  foreach S_j ∈ [1; nbSpheres] do
    point_set = intersect(S_j, object_tree);
    foreach p ∈ point_set do
      set_finger_config_from_IK(p);
      collision_test(finger_i, object);
      collision_test(finger_i, palm);
      collision_test(finger_i, finger_1,...,i-1);
      if no collision then
        → next finger;
\end{verbatim}
Grasp Filtering and Ranking

- Grasps that do not verify force-closure are discarded.
- A stability score is computed (based on [Bounab 2008]).

The above stability criterion does not guarantee the robustness of the grasp with respect to localization error. → favor contact on areas where the surface normal is less varying (low curvature).

Finding a trade-off between different scores.
Some Results

Computation time: 5 minutes on a standard PC for a list with more than 100 grasps.
Selecting a Grasp Interactively

- The grasp set can be browsed to select a grasp adapted to the context.
- Arm and/or base inverse kinematics are tested for each grasp of the list until a solution is found.
Conclusion

- Introduction simple models to reduce the number of IK tests.
- The use of point cloud to represent the object makes no assumption on object shape or mesh quality.
- Outlook
  - Better stability criterion: too “geometric” → use of a dynamics simulator
  - Better robustness with respect to object localization error.
For Further Reading I

- A. Yershova and S. LaValle.
  Deterministic Sampling Methods for Spheres and SO(3).
  *IEEE International Conference on Robotics and Automation, 2004.*

- B. Bounab, D. Sidobre and A. Zaatri.
  Central axis approach for computing n-finger force-closure grasps.
  *IEEE International Conference on Robotics and Automation, 2008.*