Manipulation Planning Among Movable Obstacles

I. Problem Statement

a robot with n DOF, $O_F = \{F_1, \dots, F_k\}$ fixed obstacles, $O_m = \{O_1, \dots, O_m\}$ movable obstacles

O_m : Geom, Center of Gravity, Motion constraints, grasps

Each O : q workspace position (x, θ, y)

Initial configuration : $W^0 = (0, r^0, q_1^0, q_2^0 \dots q_m^0)$, final config : $g_{G_1}^{goal}$

Operators: Navigate : $N(\tau)$, (only robot) Manipulate : $M(\tau, O_i)$, (with object) $\tau : [0,1] \to r$, path, $\tau(s)$ a config on the path

Maps $W^t \to W^{t+1}$ $(t, r^t, q_1^t, q_2^t, \dots, q_m^t)$, $q^{t+1} = q^t$ for unattached objects

Subject to : $N(\tau(r^t, r^{t+1}))$ for any $\tau(s)$, $s \in [r^t, r^{t+1}]$, doesn't collide with any O

Valid Manipulation : $\tau_{O_i}(s) = T_{q_i}^{K(r^t)} K(\tau(s))$ (→ end effector pos; relative transform to starting pose)

- $T_{q,t}^{K(r^t)}$ → valid grasp
- $\tau(s)$ → collision free
- $\tau_{O_i}(1)$ must be a statically stable placement.

Simplifying Assumptions : monotone (move once)

II. Challenges

Complexity: $O(m!(pe)^m)$ m objects, p placements, time to verify paths

Future uncertainty.

III. Algorithm

- Last step is always Manipulate O_{G_1}
- Initially, Manipulate (τ, O_G) will collide with O_{PAST}, → displace them. (but may also be blocked).
- O_{PAST} is expanded to include indirectly blocking objects.
- Let C_v be the volume of reserved space for future operations.
- All objects that collide with C_v are placed in O_{PAST}
- Initialize by setting O_G, and goal robot config r^{t+2}. $\phi = O_{MST} = O_{FUT} - C_v$

IV Motion Sampling

Paths are generated using the rapid RRT-Connect algorithm.

Placements are drawn from a uniform distribution.

A. Sampling Paths :

- PlanGrasp decides r^t, grasp of O_c
- PlanManipulation decides r^{t+1} : multi-goal connect $r \rightarrow r^{t+1}$
- PlanNavigation : final grasp of O_c : $T_m(1) = r^{t+1} \rightarrow r^{t+2}$ (grasp of the next object).

B. Sampling Placements

- Object $\rightarrow$ triangle T, upward facing normals, $n(T_j)$

$$|S| = \sum_{O_i \in \mathcal{A}} \sum_{T_j \in O_i} \text{area}(b(T_j)) \, \delta_{(n(T_j) = [0\ 0\ 1]^T)}$$

(condition / implies)

sum of bounding box areas for the triangles $b(T)$

$$P(p) = U(S) = \frac{\text{area}(b_T(p)) \cup T(p))}{|S|}$$

pdf for point p

C. Overall Algorithm

Start from the last action

- PlansGrasp $(O_c) \rightarrow (r^t, \text{grasp})$ return grasp to pick the object
- FindPlacement $(O_c, \text{Grasp}, O_{FUT}, C_V)$
  - ↳ Use it to PlanManipulation() return path from pick to place
    - Add O_c to O_{FUT} Update future object
- PlanNavigation ( ) return path from place to input config r^{t+2}
  - Add T_M and T_N to C_V Update future volume
- Add any blocking objects in T_m and T_n to O_{PAST}
- Recursively call the algorithm for every $O \in O_{PAST}$.

V. Constraints

Placement Constraints : Early placements are constrained by O_{FUT} and C_V (volume swept).

→ Collision checking can be optimized by approximation.

Motion Constraints : Ex, a door can only be moved around the Z-axis.

→ Resolution : First-Order Retraction (FR-RRT)