Supplementary Material: Fast Swept Volume Estimation with Deep Learning

Hao-Tien Lewis Chiang[1,2], Aleksandra Faust[2], Satomi Sugaya [1], Lydia Tapia[1]

[1] Department of Computer Science, University of New Mexico, MSC01 11301
University of New Mexico, Albuquerque, NM 87131, USA. Email: lewispro@unm.edu and tapia@cs.unm.edu.
[2] Google Brain, Mountain View, CA 94043, USA. E-mail: faust@google.com and lewispro@google.com.

1 Training Data Generation

The approximation is an octree-based SV algorithm, where the robot trajectory is represented by N_{lerp} intermediate C-space configurations. The j^{th} intermediate configuration is

$$\boldsymbol{c}_{i,j} = (1 - \frac{j}{N_{\text{lerp}}})\boldsymbol{c_{i,1}} + \frac{j}{N_{\text{lerp}}}\boldsymbol{c_{i,2}}, \quad j = 1, \cdots, N. \tag{1}$$

Next, the forward kinematics of the robot maps each $\boldsymbol{c}_j$ to the workspace occupancy of the robot

$$\mathcal{G}_{i,j}(x,y,z) = \begin{cases} 1, & \text{robot overlaps with point } (x,y,z) \in \mathcal{V}(\boldsymbol{c}_j) \\ 0, & \text{otherwise.} \end{cases}, \tag{2}$$

where x, y, z are the positional coodinates in the 3D workspace. The SV can be approximated by taking the union of all $\mathcal{G}_i$,

$$\boldsymbol{y_i} = \tilde{\mathcal{SV}}(\boldsymbol{c_{i,1}}, \boldsymbol{c_{i,2}}) = \sum_{x,y,z} \mathcal{V}(x,y,z) \cup_{j=1,\cdots,N} \mathcal{G}_{i,j}(x,y,z), \quad i = 1, \cdots, n,$$

where $\mathcal{V}(x,y,z)$ is volume of the (x,y,z) cell. The occupancy and the union operation are approximated by an octree decomposition of workspace up to a resolution, Δ, in order to speed up computation compared to an uniformly distributed voxel grid.

2 Proof of Proposition 1

Consider the C-space of a non-deformable robot $\mathcal{R}$ with d_f degrees of freedom, operating in 3-dimensional workspace. The robot consists of n-joints connecting total of $n+1$ rigid body linkages. Each joint $i = 1, \ldots, n$ rotates the

$i+1, \cdots, n+1$ linkages along the three rotational degrees of freedom yaw, pitch, roll $(\rho_i, \theta_i, \psi_i)$ relative to the i^{th} rigid body. Under these assumptions a configuration point in the C-space is $3(n+1)-$dimensional vector, $c = (x, y, z, \rho_1, \theta_1, \psi_1, \cdots, \rho_n, \theta_n, \psi_n)$, where x, y, z are coordinates of the fist linkage in the global coordinate frame. This C-space is general and can fully describe free-floating rigid and articulated bodies.

Proposition 1. *$\mathcal{SV}(\boldsymbol{c}_1, \boldsymbol{c}_2)$ along a continuous trajectory between two configuration points $\boldsymbol{c}_1, \boldsymbol{c}_2$ in the C-space above is Lipschitz continuous, i.e.*

$$\|\mathcal{SV}(\boldsymbol{c}_1, \boldsymbol{c}_2) - \mathcal{SV}(\boldsymbol{c}_1 + \Delta\boldsymbol{c}_1, \boldsymbol{c}_2 + \Delta\boldsymbol{c}_2)\| \le K\|\Delta\boldsymbol{c}_1 + \Delta\boldsymbol{c}_2\|, \tag{3}$$

where K is a positive constant.

Proof. To show that the Lipschitz continuity extends to the cases where the robot changes shape due to linkage rotations, we use mathematical induction by the linkage.

(Base Case $i = 1$) $\mathcal{SV}(\boldsymbol{c}_1, \boldsymbol{c}_2)$ of a rigid body with respect to translation and rotation is a Lipschitz continuous per [1].

(IH Case $i = n-1$) Assume that the size of swept volume for the first $n-1$ joints is Lipschitz continuous w.r.t. the rotation of all the joints.

(Step Case $i = n$) We need to show that (3) holds, for $\boldsymbol{c}_1, \boldsymbol{c}_2, \Delta\boldsymbol{c}_1, \Delta\boldsymbol{c}_2 \in \mathbb{R}^{3(n+1)}$. Let us denote four configurations:

$$\boldsymbol{c}_j^n = (x^{(j)}, y^{(j)}, z^{(j)}, \rho_1^{(j)}, \theta_1^{(j)}, \psi_1^{(j)}, \cdots, \rho_n^{(j)}, \theta_n^{(j)}, \psi_n^{(j)}),\ j = 1, \cdots, 4, \tag{4}$$

where $\boldsymbol{c}_j^n$ is the j^{th} configuration of the entire robot.

$$\boldsymbol{c}_j^{n-1} = (x^{(j)}, y^{(j)}, z^{(j)}, \rho_1^{(j)}, \theta_1^{(j)}, \psi_1^{(j)}, \cdots, \rho_{n-1}^{(j)}, \theta_{n-1}^{(j)}, \psi_{n-1}^{(j)}),\ j = 1, \cdots, 4, \tag{5}$$

are start and goal configurations for the first n linkages, and

$$\boldsymbol{c}_i^0 = (x_i, y_i, z_i, \rho_i, \theta_i, \psi_i), \qquad i = 1, \cdots, 4 \tag{6}$$

$$\boldsymbol{c}_{j+2}^0 = \boldsymbol{c}_j^0 + \Delta\boldsymbol{c}_j^0,\ \Delta\boldsymbol{c}_j^0 = (\Delta x_j, \Delta y_j, \Delta z_j, \Delta\rho_j, \Delta_j\theta_j, \Delta\psi + j), \qquad j = 1, 2 \tag{7}$$

$\boldsymbol{c}_1^0, \boldsymbol{c}_2^0$ are start and end configuration of the $n+1^{th}$ linkage in the global coordinate frame, i.e., start and end positions of the end of the n^{th} linkage, and corresponding joint angles. Then, we can decompose the $\mathcal{SV}$ into

$$\|\mathcal{SV}(\boldsymbol{c}_1^n, \boldsymbol{c}_2^n) - \mathcal{SV}(\boldsymbol{c}_2^n, \boldsymbol{c}_3^n)\| \tag{8}$$

$$\le \|\mathcal{SV}(\boldsymbol{c}_1^{n-1}, \boldsymbol{c}_2^{n-1}) - \mathcal{SV}(\boldsymbol{c}_3^{n-1}, \boldsymbol{c}_4^{n-1})\| + \|\mathcal{SV}(\boldsymbol{c}_1^0, \boldsymbol{c}_2^0) - \mathcal{SV}(\boldsymbol{c}_3^0, \boldsymbol{c}_4^0)\|, \tag{9}$$

$$\le K_{n-1}\|\Delta\boldsymbol{c}_1^{n-1} + \Delta\boldsymbol{c}_2^{n-1}\| + K_n\|\Delta\boldsymbol{c}_1^0 + \Delta\boldsymbol{c}_2^0\| \tag{10}$$

$$\le K\|\Delta\boldsymbol{c}_1^{n-1} + \Delta\boldsymbol{c}_2^{n-1} + \Delta\boldsymbol{c}_1^0 + \Delta\boldsymbol{c}_2^0\|,\quad K = max(K_{n-1}, K_n) \tag{11}$$

$$\le K\|\Delta\boldsymbol{c}_1^n + \Delta\boldsymbol{c}_2^n\|, \tag{12}$$

because the first term satisfies (IH), the second satisfies conditions of [1], and the translation factor in the second term is result of the rotational motion of the n^{th} linkage. □

3 Implementation Details

The three DNNs used to learn $\widetilde{\mathcal{SV}}$ share the same hyper-parameters. These include: the number of hidden layers $k = 3$, the number of neurons in the hidden layers = [1024, 512, 256], learning rate = 0.1, training batch size = 100 and the number of training epochs = 500 (the number of times the network utilizes the entire training dataset during training). A gradient descent-based optimizer is used by both the single layer networks and the DNNs. One hundred intermediate configurations are generated to compute $\mathcal{SV}$. The octree resolution was $\Delta =$ 0.025m. The DNNs are trained with Tensorflow on an Intel i7-6820HQ at 2.7GHz with 16GB of RAM. The training data generation is implemented within the open-source V-REP robot simulator platform. The performance of the network was evaluated by an evaluation dataset with ten thousand samples. This dataset was generated in the same fashion as the training data, but it was previously unseen by the network.

Parameters of RRT other than ones mentioned in main paper are set to the default values in OMPL. This means an extend step size of 0.2 times the maximum Euclidean distance of any pair of points in C-space and a goal bias of 0.05. The V-REP platform is used to simulate the robot and collision detection. All planning was repeated 20 times on the same computer mentioned above.

4 Robot Details

The 15 DOF planar manipulator has a fixed round base and 15 rigid cuboid bodies connected by 15 joints. The length of the bodies are [0.8, 0.2, 0.1, 0.3, 0.4, 0.5, 0.1, 0.3, 0.4, 0.5, 0.1, 0.1, 0.4, 0.1, 0.1]m long and 0.1m wide while the corridors are 1.5m wide. The square obstacle in the corridor increases the planning difficulty and is 0.1m in width. The 15 joint angles describe a configuration of the robot. Training sample configurations are uniform-randomly sampled from [-π, π] for the base joint and [-$\pi/2$, $\pi/2$] for all other joints.

The L-shaped free-floating rigid body is sized at 0.4m, 0.6m, 0.1m (width, height, depth). There are also 100 randomly placed rectangular obstacles of size 0.4m, 1.1m, 0.1m in an environment of size 6m, 2.5m, 2.5m. Training sample configurations are uniform randomly sampled from [-1.5, 1.5]m for the position axes. The planning environment is 11 times bigger than the learning environment. To ensure uniform sampling for rotation, we sample from [-π, π] for yaw, pitch and row and then convert them to quaternions.

The Kuka LBR IIWA 14 R820 fixed-based manipulator is a commonly used industrial robot. The manipulator's 7 joint angles form a configuration. The primary difference of this robot from the 15 DOF planar manipulator is that the Kuka manipulator is 3D, which gives rise to much more complex SVs.

References

1. Schymura, D.: An upper bound on the volume of the symmetric difference of a body and a congruent copy. Advances in Geometry **14**(2) (2014) 287–298