Perception 2

Tuesday, 16 March 2021 20:02

⑤ Motion 2 Vec : Semi-supervised representation learning:

Learn a motion-centric representation of surgical video demonstrations by grouping them into action segments / subgoals / options in a semi-supervised manner.

Siamese networks: learns a similarity function across images in an embedding space.

-: assume access to a supervisor that assign segment labels to small set of the demonstrations.

+: learn to align the unlabeled demonstrations with the labeled ones using the sequencing model (embedding parameters)

embedding $\leftarrow z_t = f(I_t \mid \hat{\theta}_D, \hat{\theta}_s)$ (sequence model parameters)

pseudo-labels $\leftarrow \hat{z}_{t-\ell:t} = h(z_{t-\ell:t} \mid \hat{\theta}_D, \hat{\theta}_s)$

mini-batch of length ℓ

Embedding model : Triplet loss, sequence model : Cross-entropy loss

Embedding model : TCN, sequence model: 1-layer bi-directional LSTM

⑥ keypoints into the Future (kP_DON):

Model-based prediction with self-supervised visual correspondence learning

+: trained with a small amount of interaction data (10 minutes) and a single demonstration for the goal specification.

+: Category-level generalization.

-: DON needs 3D reconstruction.

The forward model is trained to minimize the dynamic prediction error (a.k.a. simulation error) over a horizon H: $\mathcal{L}_{dyn} = \sum^{H} \| \hat{z}_{t+h} - z_{t+h} \|_2^2$

Uses Model-predictive path integral (MPPI) as a planner.

-: The camera has to be calibrated to the robot's coordinate frame.

-: Commanding the end-effector velocity in the xy plane.

⑦ kPAM 2.0: Feedback Control for Category-level Robotic Manipulation

+: Works for contact-reach tasks.

kPAM: keyPoint Affordance-based Manipulation

Uses 3D keypoints as the object representations instead of 6-DOF pose

+: Automatic generalization to new object instances, camera positions, object initial configurations and robot grasp poses.

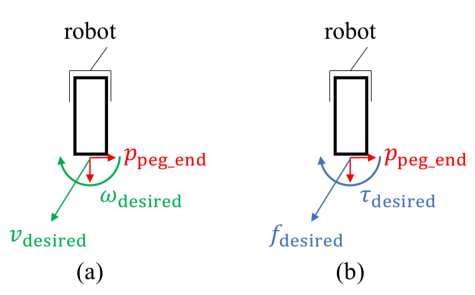

Fig. 4: Overview of the object-centric action representation. With the oriented keypoint in Fig. 3 (e) as the object representation, the action can be represented as: (a) the desired linear/angular velocity of an oriented keypoint; or (b) the desired force/torque of an oriented keypoint. Note that these two action representations are agnostic to the robot grasp pose and unrelated geometric details (see Sec. III-A).

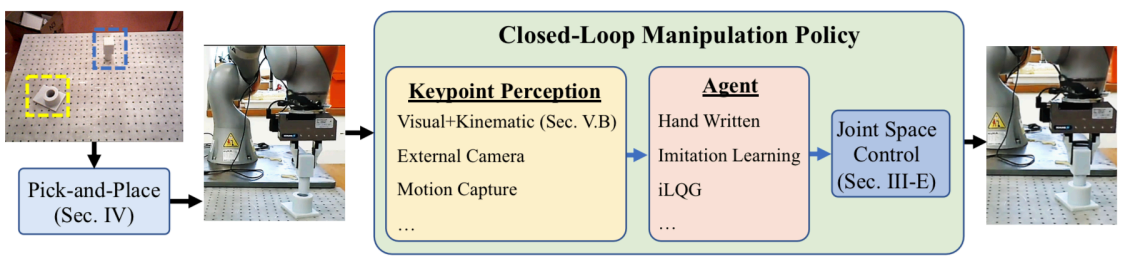

Fig. 5: Overview of the manipulation framework. The closed-loop policy consists of 1) a perception module that produces oriented keypoints in real time; 2) an agent with the state and action space shown in Fig. 3 and Fig. 4, respectively; 3) a joint-space controller that maps agent outputs to joint-space commands. Note that many different implementations of the perception module and agent can be used within our framework and the resulting pipeline automatically generalize to new objects and task setups. For many applications the objects are randomly placed initially. In this scenario, we perform a kinematic pick-and-place to move the object to some desired initial condition (for instance moving the peg right above the hole), from where the closed-loop policy starts operating (see Sec. IV for details).

-: kPAM has to detect a set of semantic 3D keypoints.

⑧ Intermittent Visual Servoing (IVS):

Intermittently switches to a learned visual servo policy for high precision segments of repitive surgical tasks while relying on a coarse open-loop policy for the segments where precision is not necessary.

+: Uses RGBD sensing to construct open-loop trajectories but for visual servoing use RGB sensing as it can capture images at a much higher frequency.

10.0 corrective updates per second (1.6 for RGBD)

+: The policy consist of an ensemble of 4 CNNs to make the policy more robust.

The weights in the convolutional layers are shared ; while the weights in the dense layers are independent.