View-based Programming with Reinforcement Learning for Robotic Manipulation

Yusuke MAEDA*, Takumi WATANABE**, and Yuki MORIYAMA*
*Yokohama National University
**Seiko Epson Corp.
Background

- Conventional Teaching/Playback
  - still widely used
  - versatile
  - for constant task conditions
    - e.g.) initial pose of object does not change
When the initial object pose is not constant...

- Object localization with cameras
  - Model-based image processing
    - Feature extraction: edge, vertex, ...
    - Pattern matching
  - Object-specific: versatility is limited
Motivation

- To develop a **versatile** robot programming method that can cope with change of task conditions

“View-based teaching/playback”: robot programming with **view-based** image processing
Model-based vs. View-based

- Model-based approach
  - with object-specific models
  - accurate

- View-based (Appearance-based) approach
  - without object-specific models
  - versatile
  - no need for camera calibration
  - not so sensitive to lighting
Overview of view-based teaching/playback (1/3)

1. Human demonstration: Operator commands a robot to perform a manipulation task
Overview of view-based teaching/playback (2/3)

2. Obtain a mapping from image to motion
Overview of view-based teaching/playback (3/3)

3. Playback: Robot motion generation according to the mapping
Mapping from image to motion (1/2)

- Neural Network

```
Input Layer

Hidden Layer

Output Layer

Raw pixel data?

Movement(t)

Robot Motion

```

9
Mapping from image to motion (2/2)

- PCA (Principal Component Analysis)

Factor scores of current image

\[ FS(t) \]

Movement \((t - \Delta t)\)

\[
\begin{align*}
\text{Input Layer} & \rightarrow \text{Hidden Layer} & \rightarrow \text{Output Layer} \\
\text{Movement} (t) & \rightarrow \text{Robot Motion} \\
\end{align*}
\]
View-based teaching/playback

- View-based image processing using PCA
  - not object-specific
  - no need for camera calibration
- Adaptability to change of initial object pose using the generalization ability of neural networks
  - generalization from multiple demonstrations
Implementation of view-based teaching/playback

Pushing/Pick-and-Place by a virtual robot hand
[Maeda 2010 ICAM]

Pushing by an actual industrial manipulator
[Maeda 2011 ICRA]
Objective

- To deal with wider change of task conditions *without* additional human demonstrations

Integration of Reinforcement Learning

- tested on virtual manipulation environment
Virtual Hand

- PD-controlled in ODE (Open Dynamics Engine) according to keyboard input
- 12 DOF (at most)
  - 6 DOF for palm
  - 3 DOF for thumb
  - 3 DOF for index finger
Target Manipulation

- Manipulation by grasping (pick-and-place)
- Graspless manipulation (pushing)
Overview of view-based teaching/playback with RL (1/3)

1. Human demonstration: Operator commands a robot to perform a manipulation task
Overview of view-based teaching/playback with RL (2/3)

2. Obtain an initial mapping from image to motion
Overview of view-based teaching/playback with RL (3/3)

3. Repeated playback and reinforcement learning:
   Robot motion generation according to the mapping and its iterative update
Neural Network for actor-critic-based Reinforcement Learning

- **Input Layer**
  - \( \alpha_{t-1} \)
  - Factor Scores

- **Hidden Layer**
  - State \( s_t \)
  - \( \text{Output Layer} \)

- **Actor** (for motion generation)
  - Robot motion \( \alpha_t \)

- **Critic** (for state value estimation)
  - State Value \( V(s_t) \)
Actor-critic-based Reinforcement Learning

- State: $s_t$
- Action: $a_t$
- Value: $V(s_t)$
- Reward: $r_t$

Random output perturbation for trial and error
Action execution
Reward acquisition
Determination of teaching signals based on reward
NN update by backpropagation
Details of reinforcement learning (based on [Shibata 03])

1. Random perturbation to actor output for trial and error

\[ a'_{i}(s_t) = a_{i}(s_t) + R_t + R_e \]

2. Calculation of TD error based on reward

\[ \text{TD}_{\text{error}} = r_{t+1} + \gamma V(s_{t+1}) - V(s_t) \]

3. Setting teaching signals according to TD error

\[ T_c(s_t) = V(s_t) + \beta (\text{TD}_{\text{error}}) \] (for critic)

\[ T_a(s_t) = a(s_t) + \rho (\text{TD}_{\text{error}})(a'(s_t) - a(s_t)) \] (for actor)
Reward function for RL

- Reward is necessary for reinforcement learning
  - Typical reward for manipulation: (negative of) distance between current object position and goal position

Not available because of view-based approach
View-based reward

- Distance between current image and goal image

\[
D_{I}(I, I_G) = \sum_{j=1}^{N_{\text{pixel}}} \frac{|I_j - I_{Gj}|}{N_{\text{pixel}}}
\]
Target task 1: Pushing

- Push the object to the goal position

**Hand**
- Reduced to 3 DOF
  - Horizontal translation + rotation
    
    \( (x, y, \theta) \)

**Object**
- Cube
Playback before Reinforcement Learning

- Successful playback from the initial position of the demonstration

![Demonstration](image1)

![Playback](image2)
Reinforcement Learning

- Repeated manipulation from an initial position from which playback is not successful before reinforcement learning

initial position of human demonstration  shifted initial position
Learning result

before RL

after RL

Computation time: 6554 [s] (CPU: core i7 870, 1000 episodes)

Successful manipulation from the shifted initial position from which it was not possible before RL
Range of initial positions for successful pushing

before RL

after RL

- Teaching point
- Learning point
- Success
- Failure

Range of initial positions for successful pushing before RL after RL
Range of initial positions for successful pushing

Wider success region after RL
Target task 2: Pick-and-place

- Pick the object up and place it at the goal

Hand
- Reduced to 3 DOF
  Translation in sagittal plane + finger bending

Object
- cube

$\alpha, (y, z, \alpha)$
Successful manipulation from an initial position from which it was not possible before RL

Computation time: 6244 [s] (CPU: core i7 870, 1000 episodes)
Range of initial positions for successful pick-and-place

Wider success region after RL
Conclusion

- Reinforcement learning was integrated with our view-based teaching/playback
- Autonomous adaptation to wider task conditions was achieved on a virtual environment
Future Work

- Computation reduction (current: \( \sim 7000 \) [s])
- Improvement of learning success rate (current: \( \sim 30\% \))
- Application to various tasks that require higher DOF
- Application to actual robot systems