A Robotic Auto-Focus System based on Deep Reinforcement Learning

Xiaofan Yu, Runze Yu, Jingsong Yang, Xiaohui Duan*

School of EECS, Peking University
Beijing, China

Speaker: Xiaofan Yu
Outline

Background
- Passive Auto-Focus
- How to deal with auto-focus using vision input

Method
- System model
- Reward Function Design
- Deep Q Network Design

Experiments
- Hardware Setup
- Training in Virtual Environment
- Training in Real Environment

Conclusion
I. Background
Background

- Passive Auto-Focus
  - First and foremost step in cell detection
  - Two phases in passive auto-focus techniques:
    - focus measure functions
    - search algorithms

Figure 1: Mechanisms of passive auto-focus techniques.
How to deal with auto-focus using vision input?

- Vision-based model-free decision-making task
- Deep Reinforcement Learning (DRL) is the solution!
  - Deep Q Network (DQN) can deal with high dimensional input
Our Contribution

- Apply DRL to auto-focus problems, which does not utilize human knowledge
- Demonstrate a general approach to vision-based control problems
  - Discrete state and action spaces
  - Reward function with an active terminal mechanism
II. Method
Method

System model

- State ($s_t$): three successive images ($x_t$) and their corresponding actions ($a_t$)
  - $s_t = \{x_t, a_t, x_{t-1}, a_{t-1}, x_{t-2}, a_{t-2}\}$
- Action ($a_t$): one in the action set
  - Action set = {coarse positive, fine positive, terminal, fine negative, coarse negative}
- Reward ($r_t$)
- DQN

Figure 4: System model.
Method

Reward Function Design

Reward Function

- reward = c \cdot (cur\_focus - max\_focus) + t
- c: coefficient
- cur\_focus and max\_focus: current and max focus value
- t: termination bonus, \[ t = \begin{cases} 100, & \text{success} \\ -100, & \text{failure} \end{cases} \]
Method

DQN Design

Figure 5: The architecture of our DQN.
III. Experiment
Experiment

- Hardware Setup
- Training in Virtual Environment
- Training in Real Environment

Figure 6: Auto-focus system implementation
Experiment

Training in Virtual Environment

■ Save time in real training phase
■ Before training, perform equal-spacing sampling to construct a simulator

<table>
<thead>
<tr>
<th>No.</th>
<th>Goal</th>
<th>Focus Range (rad)</th>
<th>Train &amp; Test Data Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Basic experiment to assess the feasibility</td>
<td>30.0-69.0</td>
<td>Same view</td>
</tr>
<tr>
<td>2</td>
<td>Comparison experiment to assess the adaptability to broader focus range</td>
<td>10.2-89.7</td>
<td>Same view</td>
</tr>
<tr>
<td>3</td>
<td>Comparison experiment to assess the adaptability to different views</td>
<td>30.0-69.0</td>
<td>Three different views, one for training and the rest two for testing</td>
</tr>
</tbody>
</table>

Figure 7: Result of virtual training phase.
Experiment

- Training in Real Environment
  - Deploy the virtual-trained model to real scenarios
  - Apply real training phase and obtain a new model
  - Compare those two models by performing tests in real world

Figure 8: Real world testing scene.

Figure 9: The histogram of focus positions.
Experiment

Summary

- In virtual training phase, our model shows great viability on larger range but need improvements on generalization capacity.
- In real training phase, our method is feasible to learn accurate policies (100% success rate) in real world but is susceptible to environmental factors.
IV. Conclusion
Conclusion

- In this paper, we
  - use DQN to achieve end-to-end auto-focus
  - demonstrate that discretization in state and action spaces and active termination mechanism could be a general approach in vision-based control problems

- Next Step
  - Improve generalization capacity by training with larger dataset
  - Improve robustness towards environmental factors
  - Reduce training time
  - ......
THANK YOU

Q & A
A Robotic Auto-Focus System based on Deep Reinforcement Learning

Reference


Reference


