ISSN No:-2456-2165

Research of Visual Navigation System Based on Virtual Scene Simulation

Tao Wang
School of Computer Science and Technology
Shandong University of Technology
Zibo, China

Abstract:- Training autonomous navigation agent in the natural environment usually requires expensive investments in terms of the cost of the machine, the environment, and the manual time consumption caused by repeated experiments to obtain a large number of training data. In this study, a deep reinforcement learning method for training visual navigation intelligence agent in virtual scenarios is proposed. The virtual scenarios which simulate the natural environment, are constructed with Unity3D engine. The intelligence agent gradually learns the spatial position relationship through many iterations of training, which is finally used in every step of the action decision.

Keywords:- *Autonomous Navigation, Deep Reinforcement Learning, Unity3D.*

I. INTRODUCTION

Robotics play an essential strategic role in national economic construction and the development of various emerging fields. Many countries have included robotics in their national high technology development plans for the 21st century. Most existing service-oriented robots, such as food delivery robots, handling robots, and security robots, usually need autonomous navigation functions. Taking AVG (Automated Guided Vehicle) driving robots in factories as an example the typical products have navigation methods based on electromagnetic induction and laser induction. The above method requires additional markers to be added to the environment. In some application scenarios where it is challenging to lay markers in the environment in advance or where it is necessary to prevent the markers from being destroyed, it is hard to rely only on the marker information in the environment. In that case, it requires intelligent robots to rely on visual image information for autonomous navigation, i.e., to accept image pixel information from the screen as input to the training algorithm during motion. It is an essential complement to the original multi-sensor information fusion technology.

One of the main features of autonomous navigation intelligence is adapting and learning from unknown environments. Depending on the feedback mechanism, the commonly used learning techniques can be divided into three main categories: supervised learning, unsupervised learning, and reinforcement learning. Among them, reinforcement learning (RL) is different from supervised learning in that it tells the intelligence what behavior to adopt by positive and negative examples. Still, a trial-and-error mechanism interacts with the environment a lot and learns the optimal policy by maximizing the cumulative reward [1]. It has been widely used in robotics due to its strong decision-making and control capabilities [2]. However, in the design and development of intelligent agent that rely on visual image information for navigation, it is not enough to only rely on reinforcement learning mechanisms. There is also a need for a method that can efficiently capture the image feature information obtained from the camera during the navigation process of the intelligent agent to make it truly "visual."

Convolutional Neural Network (CNN), an efficient method for obtaining information about image features, was developed by Yann LeCun's team in 1990 in their seminal paper [3], establishing the modern convolutional neural network framework, and later improved in [4]. They developed LeNet-5 to classify handwritten digits. LeNet-5 has multiple layers and can be trained using the backpropagation algorithm [5]. CNN achieved good results on small-scale problems such as handwritten digits. But for a long time, it was limited by the lack of training datasets and computational power to train a large, high-volume convolutional neural network without overfitting. With the rapid growth of image annotation data size and the rapid increase of hardware computing power, many methods have been developed to overcome the difficulties encountered in the training of convolutional neural networks. Deep convolutional neural networks based on deep learning methods have gained rapid development. Deep learning methods learn by training sample data to obtain a deep network structure containing multiple layers. The features obtained from the learning are more useful for visualization or classification [6]. Deep learning can obtain good features using a generic learning training process when a large amount of data is available [7]. In particular, with devices such as GPU and FPGA being used to accelerate the learning process of deep networks, the training time of the networks has been drastically reduced. In turn, they have facilitated the rapid development of deep reinforcement learning algorithms [8] that combine reinforcement learning and deep learning methods.

This study proposes a deep reinforcement learning method for training visual navigation intelligence in virtual scenarios. It can be used in fields such as factory handling robots or home service robots. Deep reinforcement learning requires many training samples and experiments and relying on a large number of trial-and-error processes. It is obtainable in fields such as image and sound recognition. When using it in a natural environment, it requires expensive investments in terms of the cost of the machine, the environment, and the manual time consumption caused by repeated experiments to obtain a large number of training samples. This study simulates the natural working environment in the Unity3D engine, gets sufficient training data in the laboratory environment, and trains the navigating intelligence in the virtual scene.

II. VIRTUAL TRAINING ENVIRONMENT

Deep reinforcement learning allows intelligence to master the skill of completing a task through extensive vision-based learning training. However, since deep reinforcement learning often requires a large number of samples and experiments and relies on many trial-and-error processes, this is still accessible in areas such as image and sound recognition but requires a costly investment when used to train real robots. Compared to training in a natural environment, training an intelligent agent in a virtual scenario is a tremendous advantage in investment cost and training speed. By simulating a robot's working environment and task requirements in an actual physical space in a virtual engine, a highly simulated virtual scenario is built as a training ground for an intelligent agent. The agent in the virtual scene can carry out trial-and-error and learn for a long time without interruption. The engine generates many training samples through computation and interacts with the back-end deep reinforcement training program. Large organizations have provided related research toolsets one after another. For example, DeepMind has released DeepMind Lab, a 3D game-like platform that presents scenes with rich SCI-FI visuals customized for AI research based on intelligence. The agent in a virtual scene can observe the environment from its perspective through a virtual camera and can perform actions such as looking around and moving in all directions in 3D. The current tasks in DeepMind include collecting fruits, walking a maze, traversing a dangerous passage, avoiding a fall from a cliff, etc. These tasks test the AI intelligence's ability to navigate, remember, and see in 3D environment.

The existing tools cannot provide virtual scenes that highly simulate the natural environment in this case. For instance, it is difficult to consider ground friction, collision detection, and other factors in the training process and customize the training details in-depth. This study will use the Unity3D engine to customize the development of the virtual scenes for training. The time required to conduct virtual simulation training will become shorter as the computing power of computer hardware continues to improve. And it is only necessary to reset the initial experimental environment in the virtual scene to restart training, which also has a great time advantage compared to training an intelligent agent in a natural environment.

III. TRAINING OF VISUAL NAVIGATION AGENT

This study uses the Unity3D engine to customize virtual scenes. It provides an API programming interface to interact with the back-end deep reinforcement training program in real-time with all the movements and environmental feedback information needed to train the intelligent agent with deep reinforcement algorithms. By designing the network structure of the deep reinforcement learning method, the intelligent agent can navigate autonomously in the virtual scene based on vision after training. The overall block diagram of the system is shown in Fig. 1.

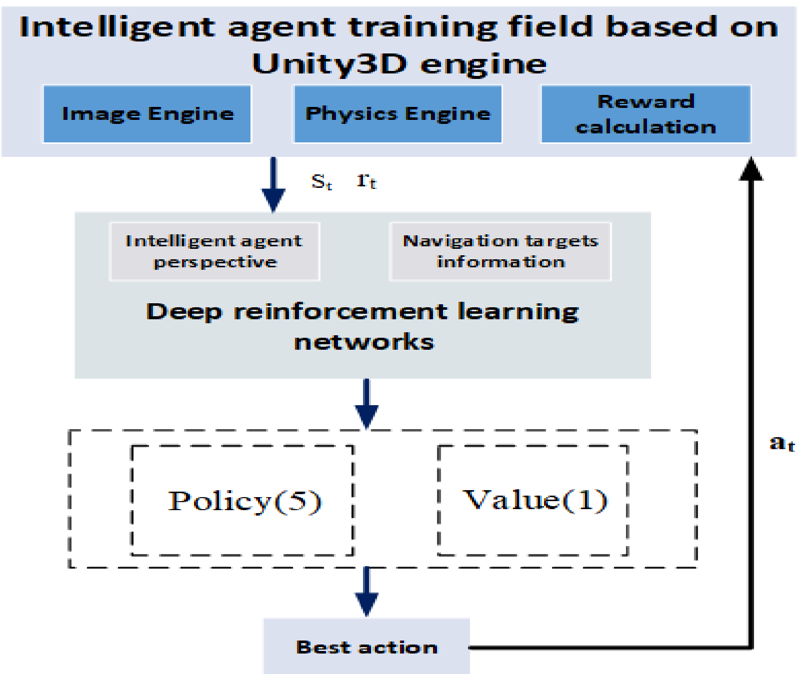

Fig. 1 Block diagram of the visual navigation system

The upper dashed box shows a virtual training scene developed using Unity3D engine. The lower dashed box shows a visual navigation intelligence based on a deep reinforcement learning approach. The two parts interact through a programming interface and a customized data format. After gridding the navigation scene, the initial position of the intelligent agent is a random location on the grid, and a point in the gridded virtual scene is selected as the navigation target. At each simulation moment, the intelligent agent receives information about the screen pixels that can be seen from its viewpoint (Fig. 2 a) as well as the navigation target image information (Fig. 2 b) and outputs the probability of the intelligent agent executing different action strategies such as forward and backward. The action selection strategy of the intelligent agent at discrete time point t is to select the action in the strategy space that has the highest probability of obtaining the optimal evaluation value at the moment t+1.

(a) Intelligent agent’s perspective (b) Navigation target
Fig. 2 Image information in virtual scene simulation

At each discrete simulation time point t, the intelligent agent selects an action execution from the set of actions. The number of action sets will directly affect the difficulty of solving the whole task. The intelligent agent can select a action from forward, backward, left turn, right turn, and brake to execute at the current position. Each action has a fixed amplitude of step length. During the training process, the intelligent agent is restricted to travel only in the road area of the virtual scene, and other regions are set as impassable areas by the collision component of Unity3D engine.

The optimal strategy is not known at the beginning of the training, so we start with a random strategy. After using a random strategy to perform a navigation task in a virtual scene, a set of states, actions, and reward values are obtained. There is often no way to traverse all states and all actions in single navigation, so only a limited sequence can be obtained. The simulation information obtained in every 1000 discrete steps is used as one exploration batch in this study. The sequences obtained from a single exploration of the intelligence are organized to form the experience pool of this exploration D.

Since the intelligent agent receives the screen pixel information from the virtual scene as input, there are too many possible screen pixel states, and it is not possible to store all the reward values in a table-like data structure. Therefore, a convolutional neural network is needed to reduce the dimensionality of the input high-dimensional image information and perform the reward value approximation. In order to shorten the training time, this study conducts image feature extraction by two ResNet-101 pre-trained models [9], respectively.

Before that, the scene pixel information (superposition of 4 consecutive images) perceived by the intelligent agent instantaneously, and the pixel information of the navigation destination are processed to be the format of $224 \times 224 \times 3$.

Also, during the training process, most parameters of the ResNet pre-training models are fixed, and only the parameters of the fully connected layer are trained. The extracted image features are fed into an asynchronous parallel multi-threaded framework using the A3C algorithm [10] to train the intelligent agent for navigation.

After the model is trained, the intelligent agent needs to decide the corresponding action at each step of the navigation process. The choice of actions, such as forward or backward, is based on the spatial position relationship between the currently obtained visual image information and the image information of the navigation target. This spatial position relationship is obtained through the simulation of a large number of training samples. The deep reinforcement learning network gradually learns the spatial position relationship through many iterations of training, which is finally used in every step of the action decision.

After the model has been trained, ten different navigation targets were reselected in the virtual scene to verify the model's generalization ability. The results showed that the intelligence agent was able to navigate to the target location autonomously in different test areas.

IV. SUMMARY

In this study, a deep reinforcement learning method for training visual navigation intelligence in virtual scenarios is proposed. Unity3D engine is used to provide virtual scenes that highly simulate the natural environment. Through the API programming interface to interact with the back-end training program in real-time, the intelligent agent collects all the movements and environmental feedback information needed for the training process. This method saves the cost of the machine, the environment, and the time consumption caused by repeated experiments to obtain a large number of training samples in a natural environment.

REFERENCES

[1]. Littman ML. Reinforcement learning improves behaviour from evaluative feedback. Nature, 2015, 521, 445 -451.
[2]. Fatemeh F, Vali D, Mehdi R. Supervised fuzzy reinforcement learning for robot navigation. Applied Soft Computing, 2016, 40, 33-41.
[3]. LeCun BB, Denker JS, Henderson D, Howard RE, Hubbard W, Jackel LD. Handwritten digit recognition with a back-propagation network. Advances in Neural Information Processing Systems, 1990, 396-404.
[4]. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 1998, 86, 2278-2324.
[5]. Hecht-Nielsen R. Theory of the backpropagation neural network. Neural networks for perception, 1992, 65-93.
[6]. Bengio Y. Learning deep architectures for AI. Foundations and Trends in Machine Learning, 2009, 2, 1-127.
[7]. George D, Dong Y, Li D, Alex A. Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition. IEEE Transactions on Audio, Speech, and Language Processing, 20, 30-43.
[8]. Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning. Nature, 2015, 518, 529-533.
[9]. He K , Zhang X , Ren S , et al. Deep Residual Learning for Image Recognition[J]. 2015.
[10]. Babaeizadeh M , Frosio I , Tyree S , et al. Reinforcement Learning through Asynchronous Advantage Actor-Critic on a GPU[J]. 2016.