Mobile Agent Control in Intelligent Space using Reinforcement Learning

László Jeni, Zoltán Istenes

Faculty of Informatics, Eötvös Loránd University
email: {jedi,istenes}@inf.elte.hu

Péter Korondi

Dept. of Automation and Applied Informatics, Budapest University of Technology and Economics
email: korondi@elektro.get.bme.hu

Hideki Hashimoto

Institute of Industrial Science, University of Tokyo
email: hashimoto@iis.u-tokyo.ac.jp

Abstract: Finding the safest shortest path in an unknown environment is a fundamental task in mobile robotics. To emulate the human adaptibility in this field, we can use the Intelligent Space concept. The Intelligent Space is a distributed sensory system, which is the background infrastructure to observe human walking in a limited area. The observation of human beings is applied to create a walkable area map of the environment and this map is applied to a learning framework to find the safest path through the environment. The proposed learning framework applies Temporal Difference learning. The main contribution of this paper is that it integrates the Reinforcement Learning and the Intelligent Space concept.

Keywords: Intelligent Space, Mobile Agents, Reinforcement Learning

1 Introduction

The essential task in robotics is to create an intelligent machine which achieve a desired task through interaction with the environment. The behaviour of mobile robot is reasoning and acting based on knowledge and sensed information.

Flexible behaviour requires the ability to acquire new knowledge automatically to learn from experience and to 'forget' obsolete knowledge.

Finding the safest shortest path in a completely unknown environment is still a hard problem, because mobile robots need topological maps in order to operate in the environment. Advanced mobile systems can explore their environment and build such maps by themselves, but these methods have the problem that most sensors will not detect all possible types of obstacles (for example yellow lines on the floor or signs saying ‘don't enter’) without large amounts of contextual knowledge.

The Intelligent Space can recognize and track the path of moving object (human beings). In indoor environments people and robots consider similar things as obstacles (the only common exceptions here are steps and stairs). As our environment is build to be safe for people the robot can usually rely upon them to make few mistakes.

There are two primary threads of research that have led to what is now called reinforcement learning. One thread concerns the problem of optimal control and its solution using value functions and dynamic programming The other thread concerns learning by trial and error and started in the psychology of animal learning. Recently, several modifications and applications were published, a good overview can be found in [10]. It can be formalized as a Markov Decision Process as well [7].

We propose the first version of a learning framework which uses the capability of the Intelligent Space and reinforcement learning in order to learn the safest path through an environment. The Intelligent Space will serve us as a test bed for the existing reinforcement learning algorithms and the development of new ones.

This paper is organized as follows. The next section introduces the Intelligent Space concept. Section 3 introduces reinforcement learning and Temporal Difference learning. Section 4 introduces the learning framework integrated into the Intelligent Space. Section 5 shows experiment for learning the safest shortest path in the environment.

2 Intelligent Space

2.1 History and Concept of the Intelligent Space

Hashimoto Lab. in University of Tokyo has proposed 'Intelligent Space' since 1996 [1]. At the beginning it consisted of two sets of vision cameras and computers with a home made 3D tracking software, this was written in C and

tcl/tk under Linux. Later, a large-sized video projector (100 inches) was added to the Intelligent Space as an actuator. Mobile robots were located in the Intelligent Space for supporting people as well as for being supported. Vision cameras and computers sets were arranged around an entire room and it changed into the Intelligent Space. Conventionally, there is a trend to increase the intelligence of a robot operating in a limited area. The Intelligent Space concept is the opposite of this trend. The surrounding space has sensors and intelligence instead of the robot. A robot without any sensor or own intelligence can operate in an Intelligent Space. In the conventional solution the robot measures, calculates and decides. The heart of the iSpace concept is that the robots must not measure, calculate or make decision. They just carry out, execute commands getting information from the distributed devices called Ubiquitous Sensory Intelligence which is realized by Distributed Intelligent Networked Devices (DIND).

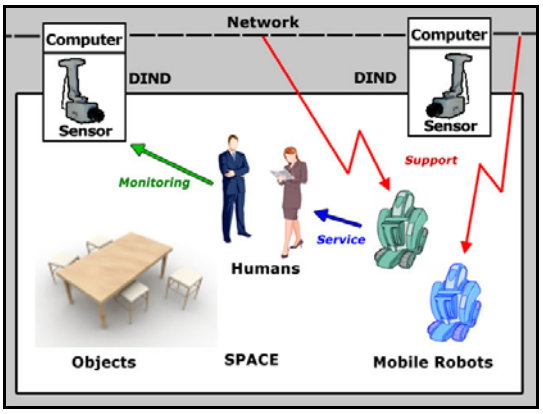

Figure 1
Intelligent Space Concept

The Intelligent Space consists of humans not only sensors cameras or robots. In the Intelligent Space DINDs monitor the space, achieve data and share them trough the network. Since robots in the iSpace are equipped with wireless network devices, DINDs and robots together organize a network.

The basic concept of Intelligent Space has extended with its development. The iSpace is a system for supporting people in it. Events, which happen in it, are understood. However, to support people physically, the intelligent space needs robots to handle real objects. Mobile robots become physical agents of the Intelligent Space and they execute tasks in the physical domain to support people in the space. Task includes movement of objects, providing help to aged or disabled persons etc. Thus, the Intelligent Space is an environmental system, which supports people in it electrically and physically. Another interesting application here is that the room can serve as a high level, context sensitive

interface to robots. The Intelligent Space is a platform to which desultory technologies are installed.

The ongoing research activities about Intelligent Space achieved several results and solutions in the field of recognition and tracking the path of moving objects [2], feature extraction [3] and motion control [4]. These algorithms mainly use classical mathematical and soft-computing methods. Although these algorithms perform well in Intelligent Space, in some aspects they face their limits. Recent research focuses on image recognition and on solutions that are developed on the analogy of the human vision processing [5].

2.2 Walking Area Identification in the Intelligent Space

To identify the walking areas in the environment the Intelligent Space tracks the movement of humans [6]. Recognizing the human is done in two steps. First the area of a human is separated from the background. Second features of the human as head, hands, feet, eyes, etc. are located. Taking the images of several cameras we can then calculate the 3D position of the human. See Figure 2.

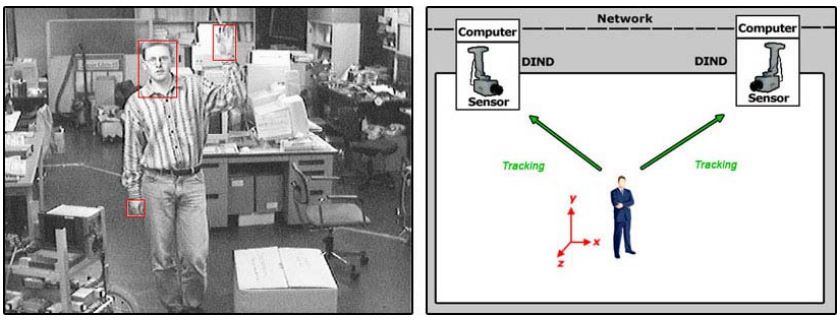

Figure 2
Human recognition and 3D position calculation

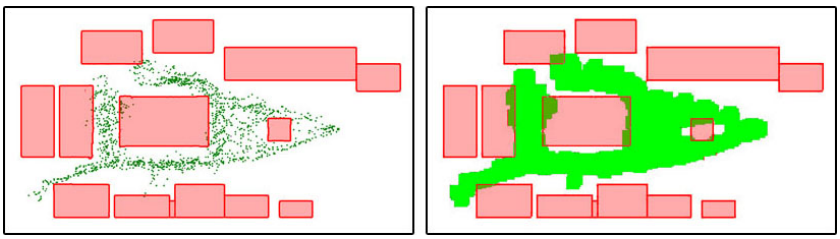

Figure 3
Ground plane of the Intelligent Space (rectanges are tables and other objects). Filtered positions marked with dots (left), and the dilated walkable area marked (right).

To determine which areas people walk in the Intelligent Space recognizes what people are doing when they were seen. Positions where people sitting on chairs, learning over tables or having their hands on tables while working have to be filtered out. The Intelligent Space filters out these positions by height tresholding: if a person is stainding upright its head is above its walking area. Then the positions are dilated with a morphological operator to obtain a connected walking area.

3 Reinforcement Learning

3.1 Markov Decision Processes

The concept of a reinforcement learning problem is easiest to describe by considering an agent situated in some environment, as shown in Figure 4. The agent can sense information about the state of the environment. The agent can also affect the environment by taking one of a set of actions available to it. After each action is taken, the agent receives a feedback signal from the environment called the reward, which determines how well the agent is performing the target task in the environment. The goal in a reinforcement learning problem is to learn which action to take in each state to maximise some optimality criterion based on the rewards received over time. Some examples of optimality criteria are average reward per time step, total return over a finite horizon and total discounted return.

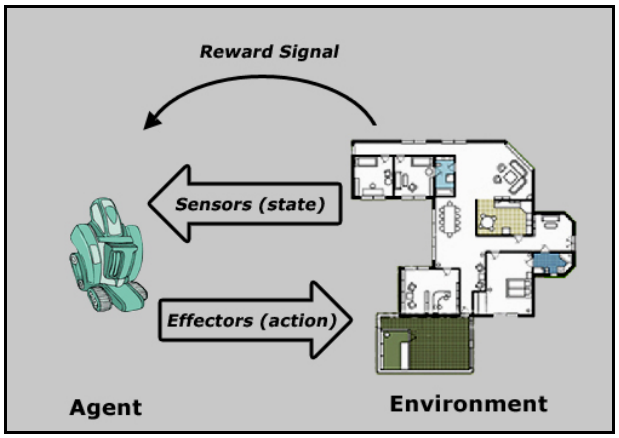

Figure 4
Basic components of a reinforcement learning problem

A reinforcement learning problem can be formalized as a Markov Decision Process [7] or MDP. An MDP is described by a quadruple <S,A,T,R> where:

- S is the set of possible states.
- A is the set of available actions.
- T(s,a,s') $\rightarrow$ [0,1] is the transition function defining the probability that taking action a in state s will result in a transition to state s'.
- R(s,a,s') $\rightarrow$ **R** is the reward function defining the reward received when a transition is made.

A particular strategy for choosing actions in an MDP is known as a policy, and is specified formally as a function $\pi(s,a) \rightarrow [0,1]$, which defines the probability of selecting each action in a given state.

For some policy π and a discount factor $\gamma \in [0,1)$, the value function $V^{\pi}(s)$ can be defined as the expected total discounted return when starting in state s and using policy π to choose actions:

$$V^{\pi}(s) = \sum_{s'} T(s, \pi(s), s') \left[R(s, \pi(s), s') + \gamma V^{\pi}(s') \right]. \quad (1)$$

Intuitively, the value function $V^{\pi}(s)$ represents how good it is for an agent to be in a particular state of the MDP, given that subsequent actions are to be chosen using policy π. The discount factor is used to determine the relative worth of future rewards in comparison to rewards available immediately in the current state. The value of γ is chosen to be less than 1 to give V^{π} a finite value for each state. The optimal policy π^* is the policy which, according to the optimality criterion, performs better in the environment than any other policy π. The formal definition of π^* is:

$$V^{\pi^*}(s) = \max_{\pi} V^{\pi}(s), \quad \forall s \in S. \quad (2)$$

While our goal is to find π^*, MDP solution methods are often based on a calculation of the value function for the optimal policy V^{π^*}, also denoted by V^*. Once V^* has been calculated, the parameters of the MDP can be used to calculate π^* as well:

$$\pi^*(s) = \arg\max_{a} \sum_{s'} T(s, a, s') \left[R(s, a, s') + \gamma V^*(s') \right]. \quad (3)$$

3.2 Temporal Difference Learning

In the learning framework we used Temporal Difference (TD) Learning [8] to learn the value function V^{π} for the policy π being followed by the learning agent.

The Temporal Difference method is a learning-method driven by the difference between two successive state values to adjust former state values which decrease the difference between all two successive state values. This method guaranteed converges to the optimal policy within a finite amount of evaluation.

In the framework TD is implemented using an eligibility trace. The eligibility trace for a state s is a value e_s which determines the extent to which s should be updated using the value of the current state s_t. At every time step each of the e_s values is updated as follows:

$$e_s \leftarrow \begin{cases} \gamma\lambda e_s & if\ s \neq s_t \\ \gamma\lambda e_s + 1 & if\ s = s_t \end{cases} \tag{4}$$

Once the eligibility trace values have been updated, the current estimate of each state value can also be updated:

$$\begin{aligned} \delta_t &\leftarrow r_{t+1} + \gamma V(s_{t+1}) - V(s_t) \\ V(s) &\leftarrow V(s) + \alpha e_s \delta_t \end{aligned} \tag{5}$$

Furthermore we used ε-greedy [9] strategy (with ε = 0.01) to balancing exploration and exploitation in the learning process. This is a simple but effective mechanism for trading off the exploration of the random policy against the exploitation of the greedy policy. There is a small probability ε at each time step of picking an action at random, otherwise the greedy policy is followed. With a good choice of the value for ε, the policy will quickly converge to one which selects the optimal action with probability (1-ε).

4 Learning Process

Figure 5 shows the scheme of the whole system. The distributed sensors (DIND) of the Intelligent Space observe the position and speed of inhabitants (a), then the system identifies walking areas from sensed situations (b). The walking area map is given to the learning system, where the safest shortest path is learned (c). The learned safest path is used to control mobile agents in the Intelligent Space (d).

Before we give the walking area map to the learning system, the system discretizes the map into squares, to reduce computational complexity of the learning (Figure 6).

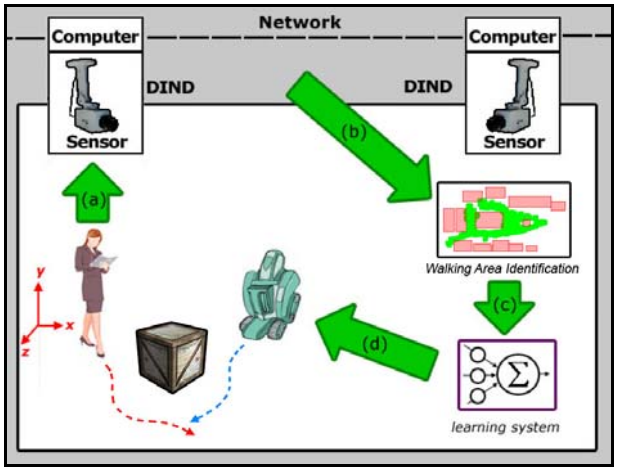

Figure 5
Scheme of the whole system: (a) observing, (b) walking area identification, (c) learning system, (d) control mobil agents.

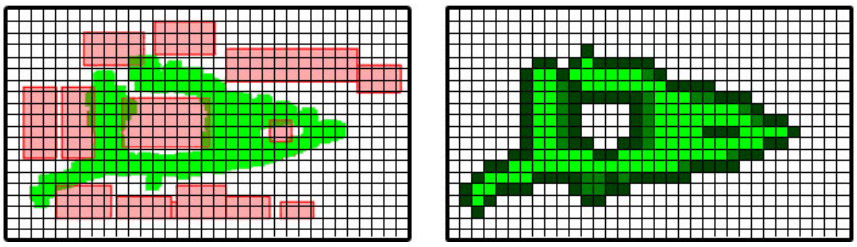

Figure 6
Discretized map and safe areas.

5 Simulations and Results

In this section we present some experimental results. Our experiments were performed using the walkable area map of the Intelligens Space and we defined two navigation tasks in this domain. The agent starts from the point ‘A’, and gets a reward of +10 for reaching the point ‘B’, as shown in Figure 7. The agent can move in the four cardinal directions on the discretized map, with a reward of -1 on every step that does not end at the goal state. Furthermore the agent gets a reward of -3 for each bounce (walk into the unsafe area).

It is possible to learn the task with the mentioned algorithm in very few learning episodes. Figure 8 displays the learning curves of the navigation tasks obtained by

averaging over 100 runs. The curve of the first task starts from –0.99 and stabilizes around a performance of –0.52 after 12 episodes. In the second task the curve starts from –0.93 and stabilizes around a performance of –0.64 after 18 episodes.

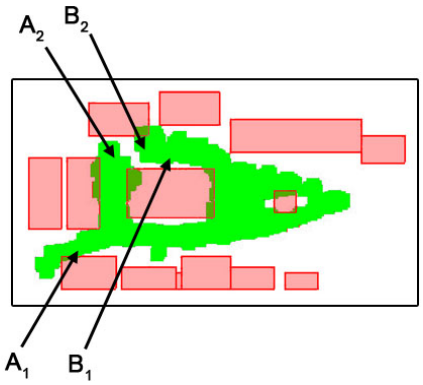

Figure 7
Navigation task (from A_1 to B_1 and from A_2 to B_2).

Figure 8
Learning curves for the navigation tasks.

The results presented in this paper depend on the parameters of the used algorithm. In this example we set the trace-decay parameter λ to 0.95 and the discount factor γ to 0.9. Furthermore we used $\varepsilon = 0.01$ for the ε-greedy strategy.

Conclusions

We have described a learning framework integrated into the Intelligent Space, which is able to learn the shortest safest path in the environment. The results of the

simulations illustrate that the system can learn the optimal path in very few learning episodes and therefore can be used to control mobile robots in the Intelligent Space.

The final goal is to get the learning framework on real mobile robots of the Intelligent Space. To achieve this the next step is to expand the learning framework for a more complex model, which describes better the real robots. In future work we will use hierarchical reinforcement learning methods, which have been proved to be useful for learning in large domains.

Acknowledgement

The authors wish to thank the National Science Research Fund (OTKA K62836), Control Research Group and János Bolyai Research Scholarship of Hungarian Academy of Science for their financial support and the support stemming from the Intergovernmental S & T Cooperation Program.

References

[1] Péter Korondi, Hideki Hashimoto: Intelligent Space, as an Integrated Intelligent System, Keynote paper of International Conference on Electrical Drives and Power Electronics, Proceedings, 2003, pp. 24-31

[2] Barna Reskó, Andor Gaudia, Péter Baranyi, Trygve Thomessen: Ubiquitous Sensory Intelligence in Industrial Robot Programming, 5th International Symposium of Hungarian Researchers on Computational Intelligence, 2004, pp. 347-358

[3] Kazuyuki Morioka, Hideki Hashimoto: Color Appearance-based Object Identification in Intelligent Space, The 8th IEEE International Workshop on Advanced Motion Control, 2004, pp. 505-510

[4] Péter T. Szemes: Human Observation-based Motion Control Strategies, PhD Thesis, Tokyo, 2005

[5] Zoltán Petres, Barna Reskó, Péter Baranyi, Hideki Hashimoto: Biology Inspired Intelligent Contouring Vision Device in Intelligent Space, Proceedings of the 6th International Symposium on Advanced Intelligent Systems (ISIS 2005), 2005, pp. 865-870

[6] Guido Appenzeller, Joo-Ho Lee, Hideki Hashimoto: Building Topological Maps by Looking at People: An Example of Cooperation between Intelligent Spaces and Robots. Proceedings International Conference on Intelligent Robots and Systems, Grenoble, France, 1997

[7] R. E. Bellman: Dynamic Programming. Princeton University Press, Princeton, NJ, 1957

[8] R. S. Sutton: Learning to Predict by the Methods of Temporal Differences. Machine Learning, 1988, 3:9–44

[9] C. J. C. H. Watkins: Learning from Delayed Rewards. PhD thesis, Cambridge University, U.K., 1989

[10] R. S. Sutton, A. G. Barto: Reinforcement Learning: An Introduction. MIT Press, Cambridge, MA, 1998 A Bradford Book