

<https://doi.org/10.7236/IIBC.2019.19.2.243>

IIBC 2019-2-33

행성 표면탐사를 위한 블랙보드 구조를 가진 멀티에이전트 루버 시스템

Multi-Agent Rover System with Blackboard Architecture for Planetary Surface Soil Exploration

딜루샤*, 최석규**, 김희숙***

K. Dilusha Malintha De Silva*, SeokGyu Choi**, Heesook Kim***

요약 행성 탐사의 첫 번째 단계는 일반적으로 자율적 탐사선을 사용하여 수행된다. 이 탐사선은 자신의 길을 찾고 행성 표면에 대한 실험을 수행 할 수 있다. 이 논문에서는 각 에이전트의 지식과 노력을 공유하기 위해 블랙보드 시스템을 효과적으로 활용하는 다중 에이전트 시스템을 제안한다. 에이전트는 BDI (Belief Desire Intension) 모델의 조합으로 반응 모델을 사용하고 최단 거리를 계산하고 행성 표면에서 이동 경로를 계산하는 경로 찾기 알고리즘을 사용한다. 이 접근법은 짧은 시간 내에 주어진 지형에서 표면 탐사를 수행 할 수 있다. 블랙보드에 수집 된 정보는 상세한 표면 토양 변화 결과를 산출하는데 사용된다. 이 연구에서 제안된 다중 에이전트 시스템에 의한 탐사는 다양한 지형 크기별로 잘 수행되는 것으로 나타났다.

Abstract First steps of Planetary exploration are usually conducted with the use of autonomous rovers. These rovers are capable of finding its own path and perform experiments about the planet's surface. This paper makes a proposal for a multi-agent system which effectively take the advantage of a blackboard system for share knowledge and effort of each agent. Agents use Reactive Model with the combination of Belief Desire Intension (BDI) Model and also use a Path Finding Algorithm for calculate shortest distance and a path for travel on the planet's surface. This approach can perform a surface exploration on a given terrain within a short period of time. Information which are gathered on the blackboard are used to make an output with detailed surface soil variance results. The developed Multi-Agent system performed well with different terrain sizes.

Key Words : Multi-Agent systems, Autonomous Rovers, Belief Desire Intension (BDI) Model, Blackboard Architecture, Path Finding.

1. Introduction

Over the last few decades, human's curiosity for explore outer space has contributed many projects for

Extra Terrestrials Exploration and those projects helped to expand human's knowledge on solar system and other planets. However, modern technology is not sufficiently adequate for transport humans to planets

*정회원, 딜루샤, 전북대학교 컴퓨터공학부

**정회원, 최석규, 전북대학교 (교신저자)

***정회원, 김희숙, 한국폴리텍대학 인천캠퍼스 컴퓨터정보과
접수일자 2019년 3월 14일, 수정완료 2019년 4월 18일
게재확정일자 2019년 4월 19일

Received: 14 March, 2019 / Revised: 18 April, 2019 /

Accepted: 19 April, 2019

**Corresponding Author: csg119@jbnu.ac.kr
Chonbuk National University, Korea

which are far away from earth. Harsh environments of those planets reduce possibility for surface experiments and make the surroundings and atmosphere are dangerous for humans. Robots and rovers become a good option but in some systems, human intervention at the middle of a task is still inevitable. Due to the speed-of-light delay in communications the method of tele operating a rover is heavily downgraded. Autonomous rovers have overcome such problems and their autonomy provides faster task completion.

Multi-agent systems can be more efficient and they are cooperative. Cooperation is not very often explicitly defined and the few definitions tend to be very broad, some include communication, some progressive results such as increased performance^[1]. One type of cooperation is active cooperation and a communication link is needed where agents actively coordinate their actions and decision making. Agent environment can also be used for communication. In passive cooperation explicit communication is not needed. Planning and scheduling techniques can be used to cooperate multi-agents. Distributed planning and scheduling methods can be used for efficient, multi-agent coordination plans, re-planning and monitoring^[3]. Unlike single monolithic rover, multi-agent rovers can more effectively explore a site and return more data^[4]. These systems are capable of explore not only planet's surface but also under the surface^[7]. Some multi-agent rovers are modeled with simulators. BRAHMS^[2] is used for such rover simulation. However, this will bring the style of the simulator for all agents and actions.

This paper proposes a multi-agent planetary surface soil exploration system with blackboard architecture for ensure an efficient information sharing among agents and fast coordination between them. This approach effectively speeds up the surface exploration process for given terrain. In this system, the blackboard is not with the level of agents. It is so fundamental and entire multi-agent system is built and managed on it. This system uses several components from Belief Desire

Intention (BDI) model and Reactive model.

II. AGENT TYPES

Multi-Agent planetary surface soil exploration system contains two types of agents.

1. Rovers

Rover agents are autonomous motor vehicles. They use cameras and ultrasonic sensors to receive inputs from the environment. Ultrasonic sensors are used to detect obstacles on the path. Cameras are useful for detect obstacles and analysis of surroundings. System also includes a blackboard so an agent can also sense its environment by using the blackboard. For actuators, rovers have built in research equipment for dig in the planet's surface.

2. Landers

Landers are supportive crafts for rovers which descend from its space craft and touch down on a planet. Landers carry rovers to surface and let rovers to be fully initiated. Usually one lander brings one rover but multi-agent soil exploration method exploits more than one rover per lander. Hence lander should be capable for carry more than one rovers. In this system, landers are modeled as agents and there are several advantages. They are actively participating in soil exploration process. They become access points for one or more rovers.

Lander agents sit still on a surface of a planet. They do not need to move. They handle all communications and they maintain the blackboard. All rover agents have to communicate through a lander agent in order to access the blackboard.

III. COMMUNICATIONS

Communications network has been established

between landers and rovers by using radio communications. Lander agents act as routers in a network. Every rover agent connects with one lander agent. Each rover agent's knowledge about its environment and effort of surface exploration are stored in the blackboard.

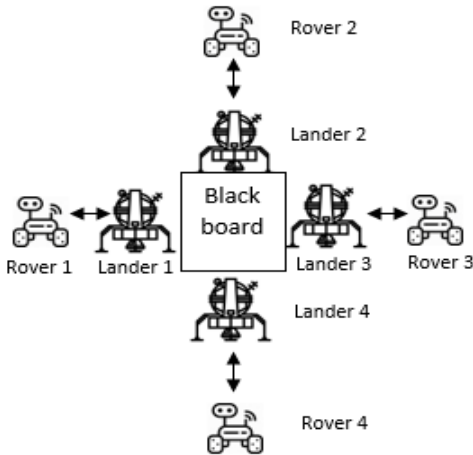


그림 1. 탐사선-착륙선 통신의 전형적인 네트워크
 Fig. 1. A typical network for rover-lander communication.

Figure 1 shows a typical arrangement for a rover-lander communication network. Every two lander agents must maintain a separate communication link for blackboard data exchange. This is shown in Figure 2.

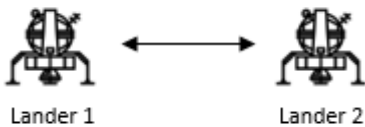


그림 2. 블랙보드 데이터 통신을 위하여 분리되어 연결된 두 개의 착륙선 에이전트
 Fig. 2. Every two lander agents have a separate link for blackboard data communication.

IV. BLACKBOARD MAINTENANCE

In Blackboard system represents a common space which is available for all agents for access. It is a

shared location for every agent to place their knowledge and information. The proposed multi-agent system's functionality is centralized with a blackboard and it is maintained in lander agents. Here are the main categories of information which is hold by the blackboard during the surface soil exploration process.

- Rover Agent location.
- Rover Agent states.
- Lander Agent location.
- Visited locations of the terrain.
- Visited frequencies for each location of the terrain.
- Soil status of a location.
- Obstacles on the terrain.

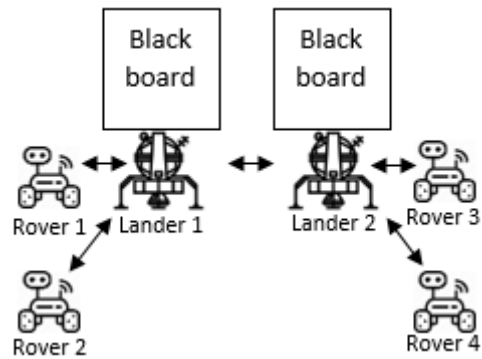


그림 3. 착륙선 에이전트들과 블랙보드 유지. 착륙선을 통해 블랙보드에 접근하는 탐사선 에이전트
 Fig. 3. Blackboard is maintained in lander agents. Rover agents access blackboard through landers.

For a given terrain there may be more than a one lander agent and each single lander agent holds an exact copy of same blackboard. Consistency and integrity features are managed by using a fast data transmission and work in synchronization. All rovers see same blackboard at any given time. Blackboard also has information about rover locations and current task of each rover. There is no direct communication needed between rovers. Figure 3 illustrates blackboard maintenance with two lander agents.

V. ROVER ARCHITECTURE

Rover agents are necessarily being a bit of sophisticated and they should be able to save power by reducing energy consumption on processing. But the performance is still considerable and a good performance can be achieved by using efficient algorithms. Autonomous and soil exploration features are the main characteristics of a rover so that some sort of reactivity is suitable for build a rover. According to the inputs from sensors, it can map to one action or most relevant action from a set of actions. For a set of percepts P and the set of actions A , the agent function F can be written as, Asterisk denotes a relation of zero more percepts. (A set of tuples of length n , $n \geq 0$. $n = 1$ for purely reactive agents.) Rovers can sense maximum distance r and circumference of the circle which is directly perceivable through its sensors is given by, where (a, b) represents the current location of the rover.

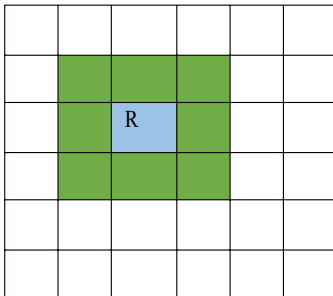


그림 4. 한 지형 위에 있는 탐사선 R. 녹색으로 표시된 상자들 구역은 외부 간섭 없이 탐사선이 직접 인지할 수 있는 영역을 나타냄

Fig. 4. Rover R is on a terrain. Green color boxes denote rover's directly perceivable area without outside intervention.

However, a rover can perceive more than a distance of radius r . The entire multi-agent system built with a blackboard and any rover agent can share their knowledge and information about the terrain. If a rover needs to know a location far away than its sense radius, it requests from the blackboard. Since all agents are connected, any other rover agent may put some

information on the blackboard. Suppose rover R1 needs information about location P1 but it is not in R1's directly perceivable area. Another rover R2 can perceive P1 through its sensors and puts information about P1 on the blackboard. After that R1 request P1 information from the blackboard.

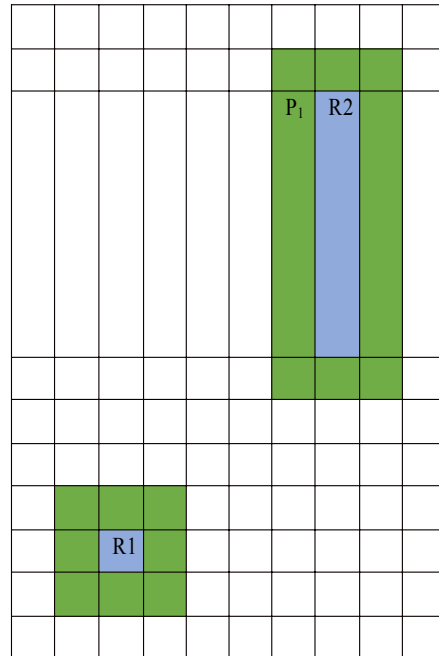


그림 5. 탐사선이 직접 인지할 수 있는 영역을 녹색으로 표시한 구역. 탐사선 R1은, R2가 이미 블랙보드에 놓여져 있는 블랙보드로부터 P1에 대한 정보를 요청함

Fig. 5. Green color boxes denote rover's directly perceivable area. Rover R1 request information about P1 from the blackboard which R2 has already put on the blackboard.

Implementation of a rover agent is accomplished with the combination of two major models. They are BDI (Beliefs Desire Intention) and Reactive architectures. Rover agent contains several major components from each architecture. Use of the BDI components helps rover agents to initiate by finding a good portion of the terrain and make rover agents to travel and investigate that portion. Figure 6 shows the architecture of the agent rover.

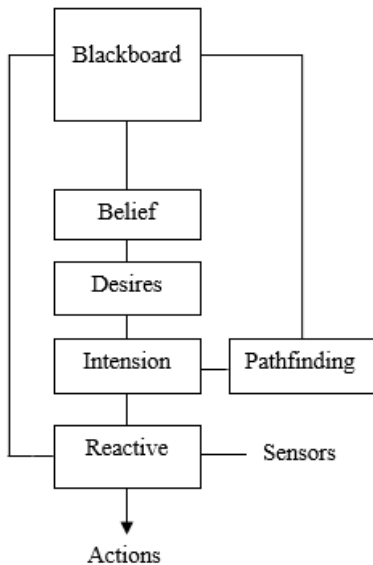


그림 6. 탐사선 에이전트 구조
 Fig. 6. Rover agent architecture.

1. Belief

Initially when a rover agent starts, it does not have any beliefs so it makes some beliefs with the information based on the blackboard. Purpose of a rover agent is to cover a wider area and explore surface soil and make available its knowledge on the blackboard. A rover agent can divide its terrain into equally sized sub terrains and visited that location one time. Rover agent calculates a sub terrain which contains locations with lowest frequencies. At this point a belief has been made for travel to a sub terrain with minimum visited frequency. Other sub terrains may already have crowded with rovers and a considerable traffic situations make a rover as a temporal obstacle for another rover.

A rover divides its terrain into N sub terrains and requests visited frequencies of every location in each sub terrain from the blackboard. If each sub terrain has M locations, where is the number of times any rover has visited location. Rover believes that a terrain with minimum is the least explored region of the terrain.

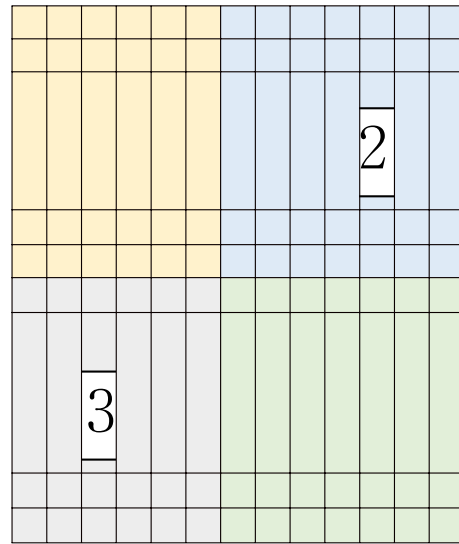


그림 7. 탐사선 에이전트에 의해 세분화된 지형. 그림에서 각 셀은 지역 내의 한 개 위치이며 하나의 탐사선이 한 개의 위치를 몇 번 방문했는지를 블랙보드가 기록함. 이는 어느 한 개의 탐사선 에이전트에 의해 보여진 하부 지형집합에 관한 하나의 예시에 지나지 않음. 여타 탐사선 에이전트들은 동일한 지역을 다른 각도로 볼 수 있음.

Fig. 7. Terrain is divided by a rover agent into equally sized sub terrains. Each cell in the figure is a location in the terrain and blackboard records how many times a rover has visited a location. This is only a one instance of a sub terrains set which is seen by a rover agent. Other rover agents may see the same terrain differently.

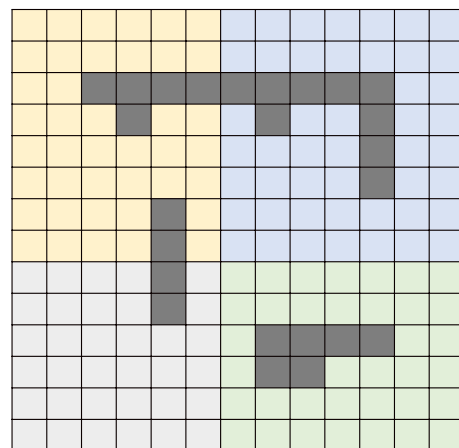


그림 8. 탐사선 에이전트가 신뢰도를 형성하는 동안 t 의 시점에서 관찰한 지형의 예시. 방문된 지역들은, 블랙보드로부터 수신한 '방문상태와 비방문상태 및 방문 빈도에 대한 자료'의 도움을 받아 확인될 수 있음. 어두운 색으로 표시된 셀이 확인된 방문지역임.

Fig. 8. This is an instance of terrain which is seen by a rover agent at time t_1 while it tries to make a belief. Visited locations can be identified with the help of data it received from the blackboard about visited/not visited status and visited frequencies. Dark cells show identified visited locations which are visited at least one time by any rover. Another agent (rover or lander) who tries to see the terrain at time t_2 ($t_2 > t_1$) will make a different instance of the terrain because every agent in the system can update the blackboard by time t_2 .

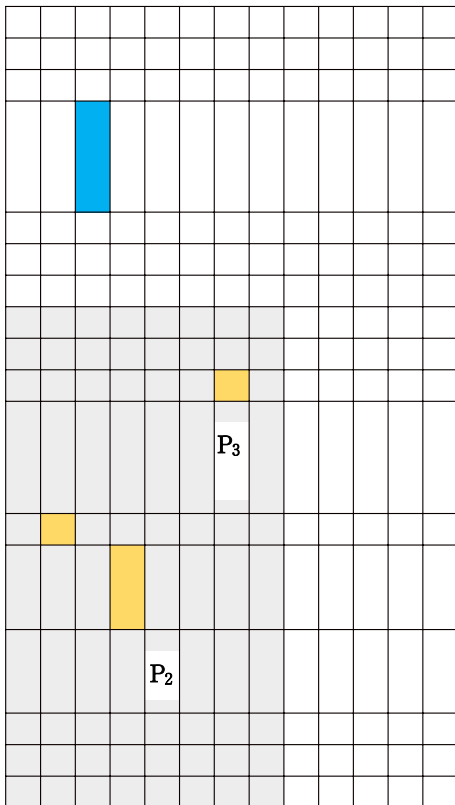


그림 9. 탐사선 에이전트 R1세 개의 위치(P1, P2, P3)를 발견, 포착함

Fig. 9. Rover agent R1 has found three locations P1, P2 and P3.

2. Desires

Rover agent now has a belief and it wants to move to a sub terrain which satisfies its belief. Rover finds out n locations ($n < M$) which are reachable in sub terrain N . These locations are organized into a list and finally one of them will be a destination for sub terrain N .

Rover desires of the proposed multi-agent system declares that locations should have two characteristics which are truly necessary for become a destination.

- 1) A location should not be an obstacle. Obstacles can be identified by using blackboard data. Sometimes blackboard is not updated at the time when a rover agent makes desires. Destination locations which contain obstacles are identified in later steps and removed from desires list.
- 2) A location should not be examined by any agent. Sometimes the destination location is already examined by another agent before a rover arrives. If the destination location already examined, control will be transferred to reactive section and rover will move to adjacent location.

3. Intention

Rover selects one location out of the set of n locations and wants to visit it. Selection process is conducted with the help of learning techniques. For the proposed multi-agent system, leaning data is in the blackboard. Preliminary requirement is the distance between agent's current location and the destination. This system uses Manhattan distance formula. If the current location of the rover is S and the destination is D , Manhattan distance d between S and D is given by, Selected location should have a maximum distance from obstacles and another agents. Targeted location should have guaranteed maximum possible area for exploration. Pathfinding algorithm finds a shortest path from agent's current location to the destination.

4. Pathfinding

For a given object which needs to be autonomous should have some sort of self-controlling mechanism to travel over its environment and arrive to its destination with the shortest possible way. The proposed multi-agent planetary surface soil exploration system has two types of agents. Lander agents do not move but they include actuators which are instruments for dig in and examine the soil in its own location. Rover

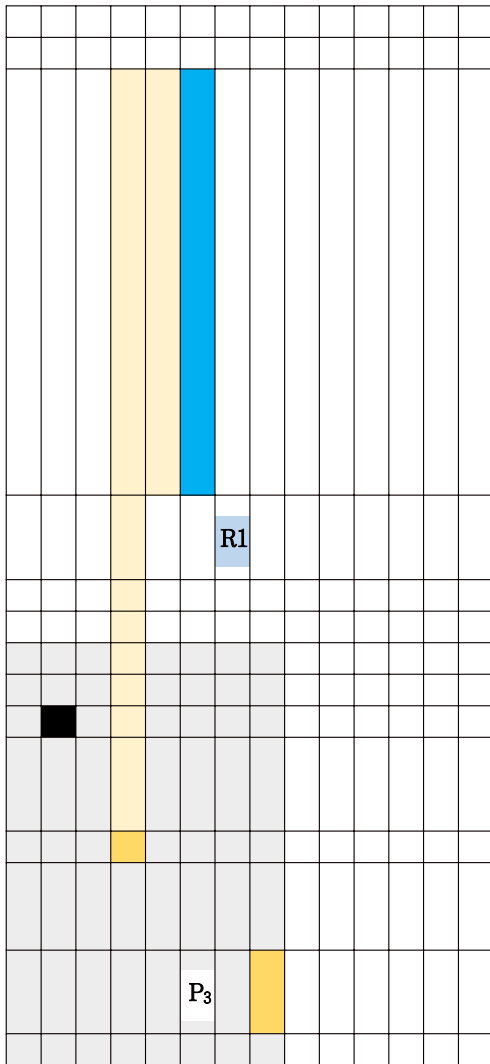


그림 10. 블랙보드에서 장애물로 보고된 P2를 위치집합에 포함한 요구사항들

Fig. 10. Desires include P2 in the locations set but blackboard reports in its lasted update that P2 is an obstacle. The path calculated for P2 is discarded immediately. Manhattan distance is one of the decision making factor in Intensions section and based on that a new path will be calculated for location P1. Location P3 is not considered for pathfinding because the distance it too high.

agents are travelers and a pathfinding section is included and connected with its intension part in order to find a path to fulfill rover’s intension and travel to its destination. The entire system is built with a

blackboard which is essentially needed to be a fundamental unit for sharing knowledge and information. Pathfinding unit has a connection to the blackboard in order to understand the terrain and calculate a path, if travel from a given source to a destination is possible.

Pathfinding algorithm finds the shortest path between two locations on the terrain. Rover agent uses Breadth First Search (BFS) Algorithm with blackboard data for find the shortest path to its intended destination. Blackboard may contain locations with obstacles but algorithm avoids them and finds a path. If an obstacle which is not seen by algorithm found on a path, the algorithm will immediately calculate a new path for rover agent so it can continue to travel to its destination.

Pathfinding algorithm which is used by rover agents is listed below.

Algorithm 1: Rover Pathfinding algorithm

Input: Terrain, Source location, Destination location.

Output: Shortest path between source and destination.

- 1: make a new queue called q
- 2: $x \leftarrow$ source
- 3: mark x as visited
- 4: push x onto q
- 5: while q is not empty:
- 6: pop q into x
- 7: for each y is adjacent to x
- 8: if y is not visited and y is not an obstacle:
- 9: mark y as visited
- 10: push y into q
- 11: $path[y] \leftarrow path[x] + y$
- 12: if y is equal to destination:
- 13: return path[y]

그림 11. 경로발견 알고리즘

Fig. 11. Pathfinding algorithm.

If an obstacle is found in the middle of a path, remaining part will be ignored. A new path will be calculated in order to fulfill rover agent’s intention^[9]. However, new path will be longer than the original path^[10]. Algorithm manages the new path and the distance to be closer to the old path and distance. That is, Distance old \approx Distance new.

5. Reactive

This is the reactive part of the rover agent. Inputs from sensors are listed and mapped into actions. Planetary surface analyzation is the main responsibility of this part. Surface soil status is categorized into three groups "ROCKY", "WATER/ICE", "SAND". Finally, this information is stored on the blackboard.

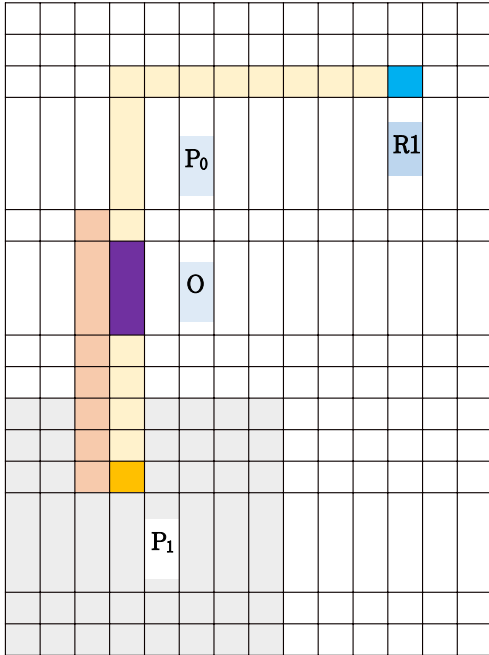


그림 12. P0에 도달했을 때 장애물 O를 감지하는 탐사선 에이전트 R1. 경로발견 알고리즘이 위치 P1에 접근하는 새로운 경로를 추정함.

Fig. 12. Rover agent R1 senses an obstacle O when it arrives at P0. Pathfinding algorithm calculates a new path for travel to the location P1. Agent's pathfinding algorithm manages the extra added distance to be very small. (Distance old \approx Distance new)

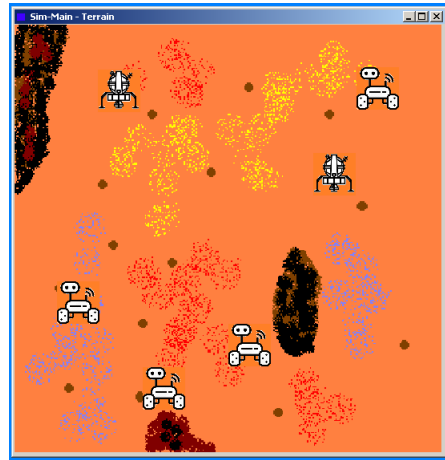


그림 13. 움직이는 에이전트들의 10x10 지형
Fig. 13. 10x10 terrain with agents in action.

Figure 14 shows a graphical representation of the blackboard during soil exploration process.

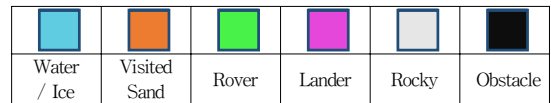
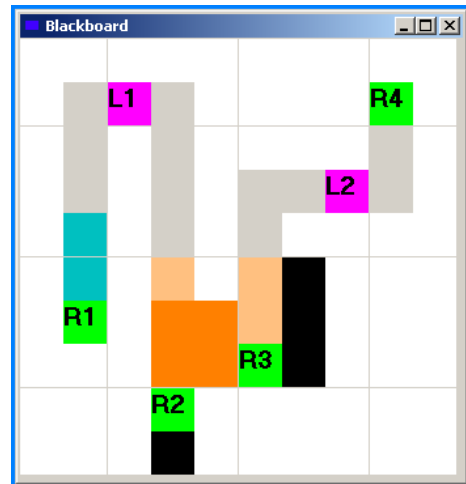


그림 14. 블랙보드 도해
Fig. 14. Graphical representation of the Blackboard.

VI. SIMULATION RESULTS

Multi-Agent Planetary Surface Soil Exploration System is implemented with a simulated terrain. Note that 1x1 is the area of one block for the simulator. Figure 13 shows two lander agents, four rover agents on a 10x10 terrain.

Simulation showed several kinds of agent behaviors and good observations those are worth for discuss. These are three important notes that will help to understand some internal facts about the proposed multi-agent system.

1. Initial Locations of the rover agents.

Rovers are usually having their start up position closer to landers. Random locations for rovers are not possible in the real world machines. Figure 15 shows initial locations of rovers and landers.

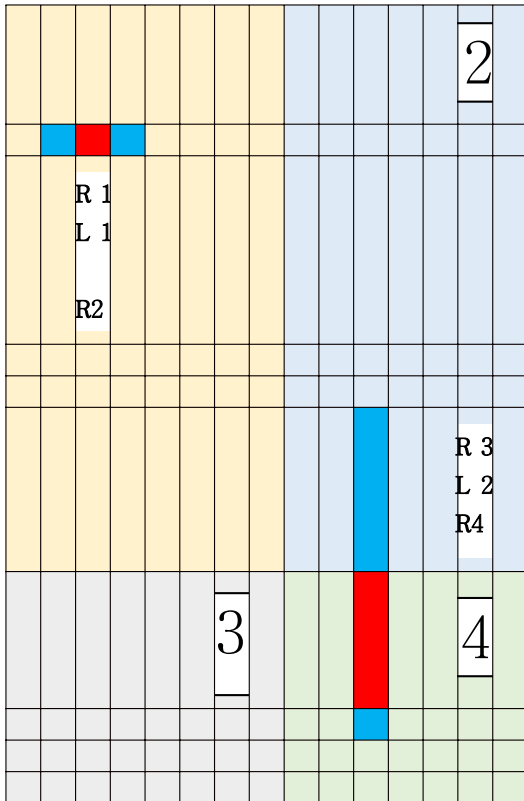


그림 15. 지형 위에 놓여진, 탐사선 에이전트 R1, R2, R3와 착륙선 에이전트 L1, L2
 Fig. 15. Rover agents R1, R2, R3, and R4 with lander agents L1 and L2 on the terrain.

Terrain has four sub terrains. Current location of any agent is marked as visited. So sub terrain 1 has three visited locations. Sub terrain 2 has one, sub terrain 3 has no visited locations and sub terrain 4 has two. Since sub terrain 3 does not have any visited locations, all rover agents make a belief that sub terrain 3 is the least explored region of the terrain and they all move to sub terrain 3. Multi-agent system assigns different start up time for each rover so that each of them make their beliefs at different times and they

would have different beliefs.

2. Rover agents per sub terrain and their distribution.

Simulation avoids locations for landers with many rovers to be much closer to each other because congestion of rovers will increase the exploration times because rovers tries to move but other rovers act as temporal obstacles.

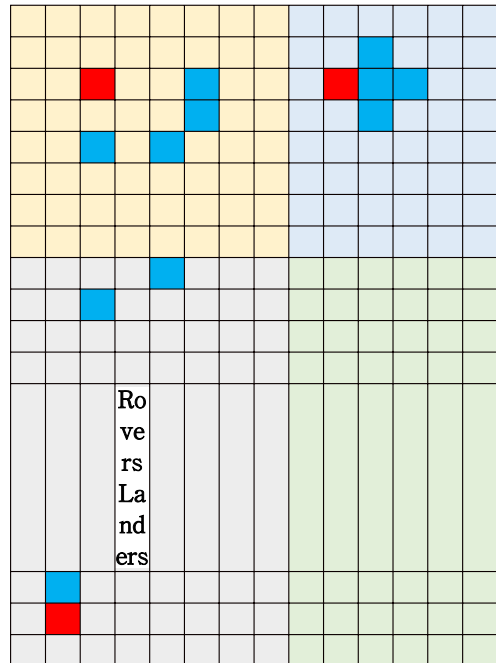


그림 16. 탐사선들에 의해 초래된 혼잡
 Fig. 16. Congestion caused by rovers.

3. Blackboard data synchronization between lander agents.

Figure 17(a) shows that blackboard data are not synchronized and no rover agent can request data from it. Usually lander-lander communication bandwidth is higher than rover-lander communication. System keeps the blackboard data synchronized at any given time. And no delays for rover - lander communication. This is shown in figure 17(b).

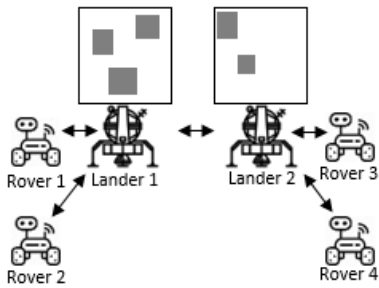


그림 17(a). 동시 통합되지 않은 블랙보드 데이터
Fig. 17(a). Blackboard data are not synchronized.

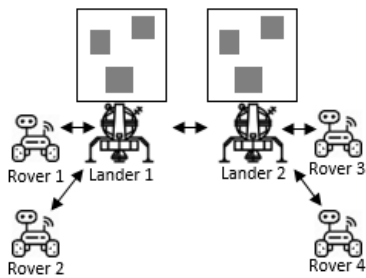


그림 17(b). 동시 통합된 블랙보드 데이터
Fig. 17(b). Blackboard data synchronization completed.

After completing several number of terrains, scientists can obtain a surface soil variations report. A sample report is shown in Figure 18.



그림 18. 표면 토양의 변동
Fig. 18. Variance of the surface soil.

Simulator has gone through several execution sessions with different number of agents and different terrain sizes. Table 1 summarizes the observations and

gives a decision per each session on speed of task completion with the help of relative measurement of elapsed time.

표 1. 다양한 지형 규모에 따라 얻은 결과

Table 1. Results obtained with different terrain sizes.

Terrain Size Rovers	5x5	10x10	15x15
1	Fair	Poor	Poor
4	Good	Good	Fair
8	Good	Good	Good

VII. CONCLUSION

Planetary exploration by using rovers is a time consuming process because until now rovers are sent out to space only one by one. This paper presents a multi-agent rover system with a blackboard for surface soil exploration on a planet. Soil exploration rovers use a blackboard for share their information and knowledge about its terrain. This work also denotes landers as agents which facilitate agent communications and maintain the blackboard.

Experimental results showed a good improvement with the multi-agent system over single rover. In this approach blackboard is not implemented in the level of agents, rather it is more fundamental and entire multi-agent system is constructed on it. This system can be examined more and many improvements are possible with the development of space transportation.

References

- [1] Jürgen Leitner. Multi-robot Cooperation in Space: A Survey. 2009 Advanced Technologies for Enhanced Quality of Life.
- [2] Chin Seah, Maarten Sierhuis, William J. Clancey. Multi-Agent Modeling and Simulation Approach for Design and Analysis of MER Mission Operations. Modeling and Simulation Human Computer Interaction; 23-27 Jan. 2003; New

Orleans, LA; United States.

- [3] Tara Estlin, Daniel Gaines, Forest Fisher and Rebecca Castano. Coordinating Multiple Rovers with Interdependent Science Objectives. Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems, 2005, Pages 879-886.
- [4] T. Estlin, H. Das, S. Hayati, A. Jain, J. Yen, R. Petras, G. Rabideau, R. Castano, S. Chien, E. Mjolsness, R. Steele, D. Mutz, A. Gray, T. Mann, D. Decoste, E. Tunstel. An Integrated Architecture for Cooperation Rovers. Symposium on AI Robotics Automation Space, Noordwijk, 1999.
- [5] Giuliano Punzo, Gordon Dobie, Derek J. Bennet, Jonathan Jamieson, and Malcolm Macdonald. Low-cost, multi-agent systems for planetary surface exploration, 63rd International Astronautical Congress, 2012.
- [6] P. Schenker, T. Huntsberger, P. Pirjanian, S. Dubowsky, K. Iagnemma, V. Sujan. Rovers for intelligent, agile traverse of challenging terrain. 11th International Conference on Advanced Robotics Coimbra, Portugal. 2003.
- [7] Wolfgang Fink, Victor R. Baker, Dirk Schulze-Makuch, Christopher W. Hamilton, Mark A. Tarbell. Autonomous exploration of planetary lava tubes using a multi-rover framework. 2015 IEEE Aerospace Conference.
- [8] P.S. Schenker. Advances in rover technology for space exploration. 2006 IEEE Aerospace Conference.
- [9] Eun-Ah Lee, Young-Bum Park, "The Technique of Automatic Generation Algorithm of Maze", The Journal of The Institute of Internet, Broadcasting and Communication(JIIBC), Vol. 3, NO. 1, pp.15-23, 2003.
- [10] Jongho Kim, "Single Image Haze Removal Algorithm using Dual DCP and Adaptive Brightness Correction", Journal of the Korea Academic-Industrial Cooperation Society, Vol. 19, No. 11, pp.31-37, 2018.

저자 소개

딜 루 사(정회원)



- 2017년~현재 : 전북대학교 대학원 컴퓨터공학과 석사과정
- 주관심분야 : 인공지능

최 석 규(정회원)



- 2000년~2006년 : 전북산업경영연구소 지역경제연구팀장(이사)
- 2006년~현재 : 전북대학교 (연구교수, 교수)
- 주관심분야 : 핀테크, 인공지능

김 희 숙(정회원)



- 2001년~현재 : 한국폴리텍대학 인천캠퍼스 컴퓨터정보과 교수
- 주관심분야 : 정보 보호, 유비쿼터스 컴퓨팅(AR), 디지털