
열화와 연기를 고려한 고층 건물 화재시의 대피 시뮬레이션

Evacuation Simulation of High-Rise Building Fires Considering Temperature and Smoke

이범중, Bum-Jong Lee^{*}, 박종승, Jong-Seung Park^{**}, 이동호, Dong-Ho Rie^{***}

요약 본 논문은 화재 발생 시 고층 건물에서 열화와 연기에 의해 거주자가 피해를 입는 것을 고려하여 초기 탈출을 계획할 수 있는 시뮬레이션 시스템의 설계 및 구현 방법을 제안한다. 화재 시에 사람들은 자신의 위치에서 가장 가까운 출구를 찾아 탈출하게 된다. 먼저 출구까지의 탈출 경로를 탐색하고 자신의 최고 속도로 이동을 시도한다. 연기 및 열화와 같은 주변의 요소에 따라서 이동 속도가 제한된다. 본 논문에서는 화재 시 사람의 속도를 저해하는 요소인 연기를 고려하여 사람의 탈출속도를 계산하고, 열화와 연기를 고려하여 사람의 피해량을 계산하여 사망 여부를 판단한다. 계산된 속도를 반영하여 고층 건물에서 화재 시의 탈출 상황을 3차원 시뮬레이션하는 시스템을 제안한다. 실제 환경에서 사람이 문을 빠져나가는 것과 같은 환경을 만들기 위해 문의 크기에 따라 시간당 사람이 빠져나갈 수 있는 수를 제한한다. 실험결과들을 통해서 본 논문에서 제안한 방법을 통하여 연기의 농도가 짙어질수록 사람의 탈출속도가 감소하는 것과 온도와 연기 농도에 의해 사람이 피해를 입고 사망하는 것을 확인할 수 있다.

Abstract This paper proposes a method of design and implementation a simulation system in which humans can plan their evacuation paths in a high-rise building considering damages from high temperature and smoke in case of fire. Humans in evacuation search the nearest exit and follow a path to the exit. When humans are evacuating toward an exit, they are going to move with their highest speeds. However, many environmental factors prohibit their fast movements. In this paper, we calculate the evacuation speed of each human considering temperature damage and smoke damage. We restrict the number of humans that can be evacuated per second according to the actual size of the exit door. Experimental results showed that the evacuation speed is affected by the temperature condition and the smoke density.

핵심어: *Fire evacuation simulation, Smoke, Temperature*

* 이범중 : 인천대학교 컴퓨터공학과 석사과정; e-mail: leeyanga@incheon.ac.kr

** 박종승 : 인천대학교 컴퓨터공학과 교수; e-mail: jong@incheon.ac.kr

*** 이동호 : 인천대학교 안전공학과 교수; e-mail: riedh@incheon.ac.kr

1. 서론

근래에는 다른 공학분야와 마찬가지로, 급속한 하드웨어 및 소프트웨어 기술의 발달에 따라 화재에 대한 컴퓨터 시뮬레이션에 대한 연구가 활발히 진행되었으며 개발된 소프트웨어가 국내외 실제 현장에서 적용되고 있다. 이러한 국내외의 소프트웨어들은 모두 2 차원을 바탕으로 하고 있기 때문에 해당 건축물의 전반적인 피난 성능을 판단할 수 있는 지표로 활용할 수는 있지만, 실제 환경은 3 차원으로 이루어져 있기 때문에 3 차원 공간에 대한 피난 효율의 지표로 활용할 수는 없다. 이는 기존의 피난 시뮬레이션 도구들이 재난 발생 시 이용자 및 거주자의 형태 재현에 초점을 맞추고, 그에 대한 공간 구조적 영향에 대해서는 따로 다루고 있기 때문이다.

본 논문에서는 이러한 문제점들을 해결하고, 사람들에게 가시적인 효과를 높여 주기 위해서 3 차원 렌더링 기술을 활용하여 화재 발생 시 사람들이 제한된 탈출구를 통해 효율적이고, 안전하게 탈출할 수 있는 경로를 찾고, 출구까지 탈출하는 동안에 생존 유무, 탈출하는 동안의 속도를 계산할 수 있는 시뮬레이션 시스템을 제안한다. 경로를 찾기 위한 알고리즘으로는 B* 알고리즘[1], Bellman-Ford 알고리즘[2], Best-First 탐색 알고리즘[3], Bidirectional 탐색 알고리즘[4][5], Breadth-First 탐색 알고리즘[3] Depth-First 탐색 알고리즘[6], Dijkstra 의 알고리즘[7], Floyd-Warshall 알고리즘[8][9][10], Hill Climbing 알고리즘[11], Johnson 의 알고리즘[12] 등이 있는데 본 논문에서는 사람들의 현재 위치에서 탈출구까지의 경로를 찾기 위해 Hart 등에 의해 처음으로 소개된 A* 알고리즘[13][14]을 사용하였다.

실제 환경과 같은 화재의 조건을 만들어 주기 위해서 문에 도달한 사람을 무조건 탈출한 것으로 간주하지 않고, 시간당 문을 통과할 수 있는 사람의 수를 문의 크기에 따라 다르게 한다. 또한 실제 화재 시에 사람들에게 피해를 줄 수 있는 요소인 온도와 연기를 적용한다. 온도에 의해 사람은 사망할 수도 있으며, 연기의 농도에 따라 사람은 사망할 수도 있고, 이동 속도가 느려질 수도 있다.

2. 본론

본 논문에서 제안하는 시뮬레이션 시스템은 먼저 각 사람의 경로를 탐색하게 된다. 그 다음 단계로는 탐색된 경로를 따라 탈출하게 되는데 탈출하는 동안의 속도를 계산하게 된다. 다음 단계로는 탐색된 경로를 통해 계산된 속도로 탈출하는 동안 연기 농도나 열에 의해 사람이 입는 피해를 계산하게 된다. 그 다음 단계는 계산된 피해에 따라 사람의 사망 여부를 판단하게 되고, 모든 사람이 사망하거나, 탈출하게 되면 시뮬레이션을 종료하는 순서로 진행된다.

본 시뮬레이션 시스템의 계산 모듈은 크게 두 가지로 나누어 볼 수 있다. 하나는 사람의 사망 여부를 판단하는 것이고, 나머지 하나는 사람의 속도를 계산하는 것이다.

사람의 탈출 속도에 영향을 미치는 요소는 연기 농도가 직접적으로 관련되고 사람의 피해에 영향을 미치는 요소는 연기 농도와 열화가 있다.

2.1 연기 농도에 따른 사람의 탈출 속도 계산

사람이 탈출할 때의 속도에 영향을 미치는 요소는 사람의 밀집도외에 연기 농도도 영향을 미칠 수 있다. 연기의 농도가 짙어지면 사람의 시야가 가려지기 때문에 사람의 탈출속도가 줄어들기 때문이다. 연기 농도에 따른 사람의 속도는 식 (1)에 의해 계산된다.

$$S_{sm} = 1.1 - 8.8889 * (7600 * (smoke * 10^{-6}) + (0.2)^2) / 1.1 \quad (1)$$

여기서 S_{sm} 은 연기 농도에 따른 사람의 속도이며, $smoke$ 는 연기 농도를 의미한다.

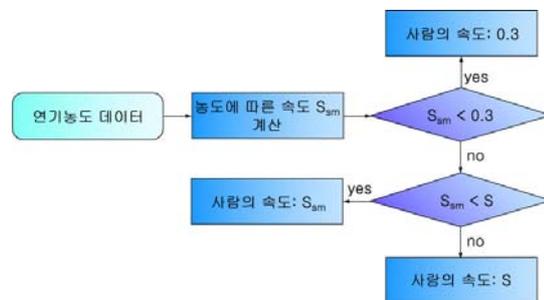


그림 1. 연기 농도에 따른 사람의 속도 계산의 전체 과정

연기 농도에 따른 사람의 속도를 계산하는 전체 과정은 그림 1 에 나타나 있다. 가장 먼저 식 (1)을 사용하여 연기 농도에 따른 속도 S_{sm} 을 계산한다. 그 다음으로 계산된 S_{sm} 이 0.3 보다 작으면 사람의 속도는 0.3 으로 설정하는데 밀집도에 따른 속도가 0.3 보다 작다면 사람의 탈출속도는 밀집도에 따라 계산된 속도를 적용한다. 반대로 S_{sm} 이 0.3 보다 작지 않다면 밀집도에 따라 계산된 사람의 속도와 비교한다. 이때 밀집도에 따른 사람의 속도인 S 보다 S_{sm} 이 작으면 사람의 속도는 S_{sm} 으로 설정한다. 반대로 밀집도에 따른 사람의 속도인 S 보다 S_{sm} 이 크다면 사람의 속도는 S 로 설정한다. 다시 말해서 밀집도에 의한 속도에 연기 농도에 의한 속도만큼을 감소시키는 것이 아니라 밀집도에 따라 계산된 속도와 연기 농도에 따라 계산된 속도 중에서 느린 속도를 사람의 탈출속도로 지정한다. 밀집도에 따른 사람의 속도와 연기 농도에 따른 사람의 속도는 매 프레임마다 계산하여 사람의 속도를 갱신하여 준다.

2.2 온도 및 연기 농도에 따른 사람의 피해도 계산

화재 시에 사람에게 피해를 줄 수 있는 요소는 연기와 온도가 있다. 연기의 농도가 높으면 사람이 질식할 수 있기 때문에 사망할 수 있고, 온도가 높게 되면 온도에 의해 사망할 수도 있기 때문에 사망할 수도 있기 때문에 온도

및 연기 농도에 따른 사람의 피해를 계산해야 한다. 연기에 의해 사람이 입는 피해는 식 (2)에 의해 계산된다.

$$F_{in} = \left(\frac{((8.2925 \times 10^{-4} \times mono^{1.036})) / 30 + 0.0045}{e^{0.1903 \times dio \times 100 + 2.0004 / 71} + e^{8.13 - 0.54 \times (20.721 - oxy \times 100)}} \right) \times (thick / 60) + F_{in-1} \quad (2)$$

F_{in} 은 연기에 의한 피해를 의미하며, $mono$ 는 일산화탄소의 농도, dio 는 이산화탄소의 농도, oxy 는 산소의 농도, $thick$ 는 렌더링 시간 간격, F_{in-1} 은 이전의 연기에 의한 피해를 의미한다. 이러한 모든 연기의 요소를 고려하여 연기 농도에 따른 사람의 피해를 계산한다.

온도에 따른 사람의 피해는 식 (3)에 의해 계산된다.

$$F_{Ith} = (thick / 60) / e^{5.185 - 0.0273 \times temp} + F_{Ith-1} \quad (3)$$

위의 식 (2)와 식 (3)을 이용하여 사람마다의 피해를 계산하고, 이 두 가지의 피해 중 하나라도 사람의 종별 사망 기준치에 도달하면 그 사람은 사망한 것으로 판단한다.

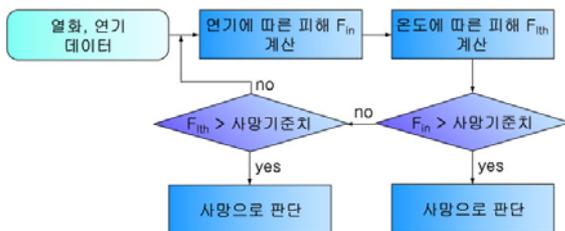


그림 2. 온도 및 연기 농도에 따른 사람의 피해 계산 과정

온도 및 연기 농도에 따른 사람의 피해량을 계산하는 전체 과정이 그림 2 에 나타나 있다. 가장 먼저 연기 농도에 따른 피해 F_{in} 을 식 (2)를 사용하여 계산한다. 다음으로 온도에 따른 피해 F_{Ith} 를 식 (3)을 사용하여 계산한다. 그 다음 과정으로는 연기 농도에 따른 피해량 F_{in} 이 사람의 사망 기준치를 초과하게 되면 사망한 것으로 판단하고, 그렇지 않다면 온도에 따른 피해량을 비교하게 된다. 온도에 따른 피해량이 사망기준치보다 크다면 그 사람은 사망한 것으로 판단하고 그렇지 않다면 다시 처음으로 돌아가서 그 사람의 연기 및 온도에 따른 피해량을 다시 계산한다. 식 (2)와 식 (3)을 보면 알 수 있듯이 피해량은 사람이 탈출하는 동안의 누적값으로 계산된다.

3. 실험 결과

제안된 열화와 연기를 고려한 고층 건물 화재시의 대피 시뮬레이션 시스템을 입증하기 위해 본 논문에서는 윈도우 플랫폼에서 C++와 DirectX 9.0 을 사용하여 시뮬레이션 시스템을 구현하였다. PC 는 2.4GHz 코어 2 프로세서로

1GB DDR RAM 의 메모리를 장착하였고, 512MB DRAM 의 GeForce 6800 GPU 의 그래픽 카드를 사용하였다.

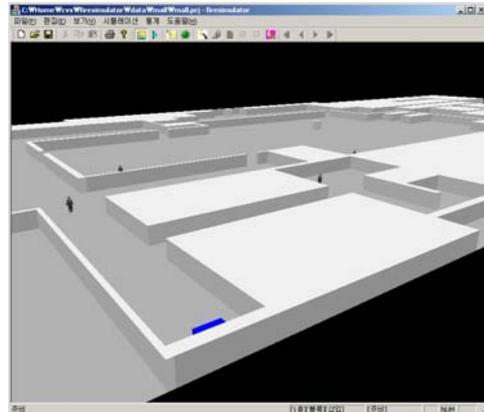


그림 3. 시뮬레이션 초기 단계의 모습

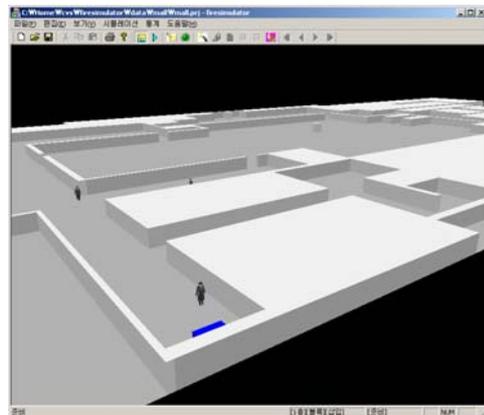


그림 4. 시뮬레이션 중간 단계의 모습

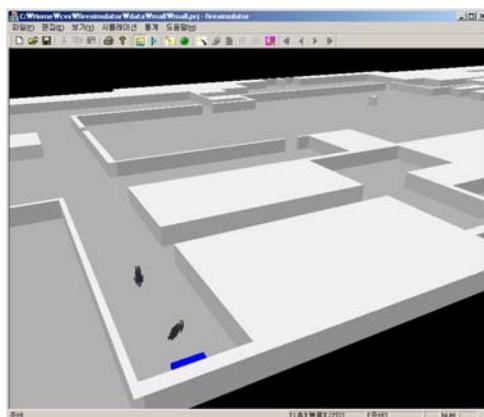


그림 5. 시뮬레이션 종료 직전의 모습

그림 3 은 사람이 탈출하기 위한 시뮬레이션의 초기 단계의 모습이고, 그림 4 는 한 사람이 탈출하고 난 후의 시뮬레이션이 반 정도 진행되었을 때의 모습이며, 그림 5 는 시뮬레이션이 종료되기 직전의 모습이다. 이 중 그림 5 에서 두 명의 사람이 누워 있는 모습을 볼 수 있는데,

이것은 사람이 입은 피해가 사망 한계치를 초과하여 사망한 것을 나타낸다.

그림 6 은 시간에 따른 온도 피해량을 나타내는 그래프로서, 사람 3 과 사람 4 의 그래프가 계속적으로 증가하는 것은 탈출하지 못하고 사망하였기 때문에 피해량이 계속 증가하는 것이고, 사람 1 과 사람 2 의 그래프가 증가하다가 평행해지는 이유는 탈출에 성공하여 더 이상 피해를 입지 않기 때문이다. 여기서 가장 많은 피해를 입은 사람 3 과 사람 4 의 최종 피해량이 약 4.5×10^{-3} 이기 때문에 사망 기준치를 초과하지 않아 온도 피해로는 사망하지 않는다. 그림 7 는 시간에 따른 연기 피해량을 나타내는 그래프로서, 사람 3 과 사람 4 의 그래프가 계속적으로 증가하는 것은 온도 피해와 마찬가지로 탈출하지 못하고 사망하였기 때문이고, 사람 1 과 사람 2 의 그래프가 증가하다가 평행해지는 이유는 탈출에 성공하여 더 이상 피해를 입지 않기 때문이다. 여기서 사람의 사망 기준치가 0.1 이기 때문에 사람 3 과 사람 4 가 시뮬레이션 종료 직전에 그림 5 과 같이 사망한 것을 확인할 수 있다.

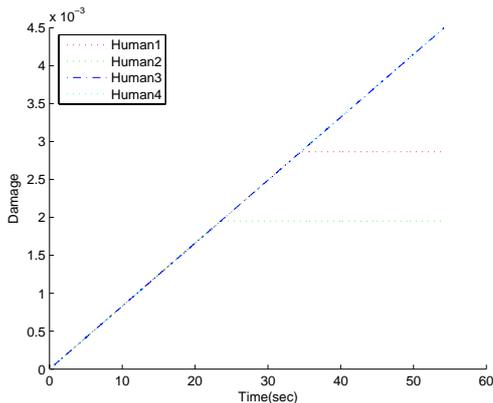


그림 6. 시간에 따른 온도 피해량

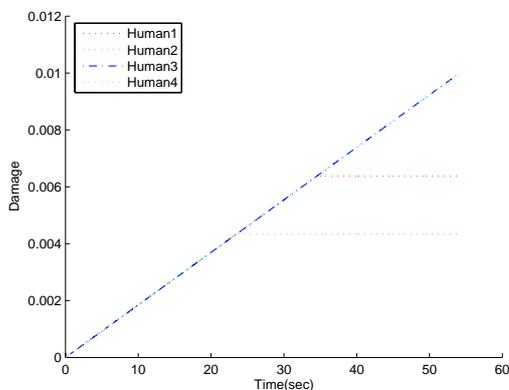


그림 7. 시간에 따른 연기 피해량

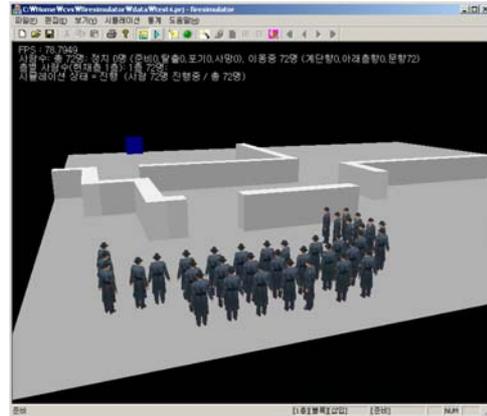


그림 8. 시뮬레이션의 초기 모습

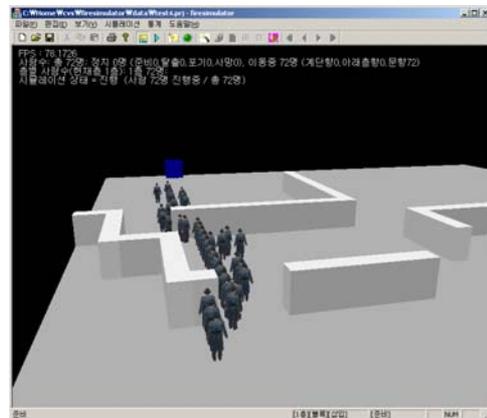


그림 9. 시뮬레이션의 진행 중 모습

그림 8, 그림 9, 그림 10, 그림 11 은 단층 건물에서 여러 사람이 하나의 출구를 향해 탈출하는 것을 시뮬레이션하는 모습을 보여준다. 그림 8 은 72 명의 사람이 파랑색으로 나타나는 하나의 출구를 향해 탈출하기 위해 준비하는 시뮬레이션 초기 단계의 모습이다. 그림 9 은 72 명의 사람들이 아직 한 명도 탈출하지 못했고, 출구를 향해 진행하는 모습을 나타낸다. 그림 8 에서는 사람들이 비슷한 위치에서 출발을 하지만 그림 9 에서는 사람들이 좁은 통로를 통해 서로 같은 방향으로 진행하려 하기 때문에 진행한 거리가 서로 차이가 많이 나는 것을 확인할 수 있다. 그림 10 는 이제 막 사람들이 출구를 통해 탈출을 시작하는 모습을 나타낸다. 그림 11 은 하나의 출구를 향해 72 명의 사람이 모두 탈출하고 난 후의 모습이다.

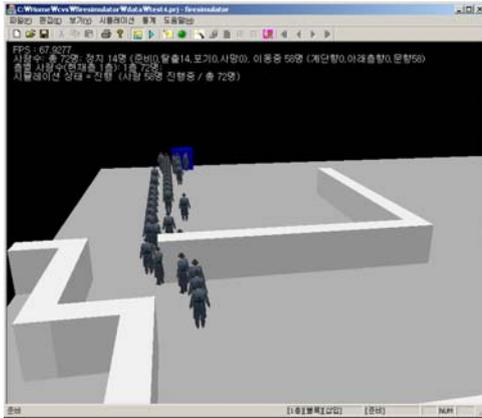


그림 10. 시뮬레이션의 진행 중 모습

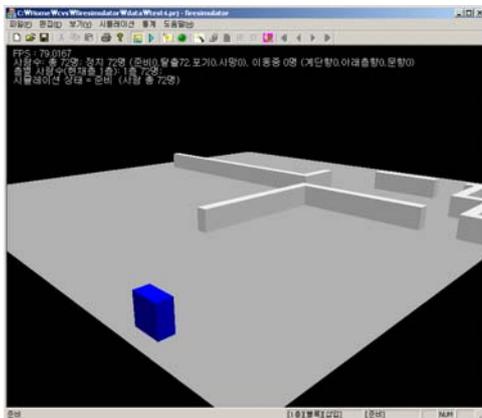


그림 11. 시뮬레이션의 종료 후 모습

4. 결론

본 논문에서는 화재 발생 시 사람이 탈출할 때에 실제 환경과 같은 조건을 만족시키기 위해 열화와 연기를 고려한 시뮬레이션 시스템을 제안하였다. 본 논문에서 제안한 방법에 따르면 사람이 출구까지 탈출하는 동안 연기 농도에 따라 사람의 탈출 속도를 계산하여 그 속도로 탈출하게 된다. 사람은 탈출하는 동안 열화와 연기에 의해 피해를 입게 되는데, 본 논문에서는 이러한 요소들을 고려하여 사람이 입는 피해를 계산하여 적용하였다. 계산된 피해가 사람의 중별 사망 기준치에 도달하게 되면 그 사람은 사망한 것으로 간주하게 된다.

본 논문에서 제안한 시뮬레이션 시스템을 이용하면 건물의 화재 발생시 사망자 및 피해 정도를 예측할 수 있는 화재 예방에 활용할 수 있다는 장점이 있다. 화재에는 수 많은 요인들이 있기 때문에 이러한 모든 요인들을 추가하여 좀더 완벽한 시뮬레이션 시스템을 만드는 것을 앞으로의 연구 목표로 하고 있다.

참고 문헌

- [1] B. Hans, "The B* Tree Search Algorithm. A Best-First Proof Procedure," *Journal of Artificial Intelligence*, Vol. 12, No. 1, pp. 23-40, 1979.
- [2] R. Bellman, "On a Routing Problem," *In Quarterly of Applied Mathematics*, Vol. 16, No. 1, pp. 87-90, 1958.
- [3] J. Pearl, "Heuristics: Intelligent Search Strategies for Computer Problem Solving," Addison-Wesley, p. 48, 1984.
- [4] D. Champeaux, L. Sint, "An Improved Bi-Directional Heuristic Search Algorithm," *Journal of ACM*, Vol. 24, No. 2, pp. 177-191, 1977.
- [5] D. Champeaux, "Bi-Directional Heuristic Search Again," *Journal of ACM*, Vol. 30, No. 1, pp. 22-32, 1983.
- [6] C. H. Thomas, L. E. Charles, R. L. Ronald, S. Clifford, "Introduction to Algorithm, Second Edition," MIT Press and McGraw-Hill, pp. 540-549, 2001.
- [7] C. H. Thomas, L. E. Charles, R. L. Ronald, S. Clifford, "Introduction to Algorithm, Second Edition," MIT Press and McGraw-Hill, pp. 595-601, 2001.
- [8] R. W. Floyd, "Algorithm 97: Shortest Path," *Communications of the ACM*, Vol. 5, No. 6, p. 345, 1962.
- [9] S. C. Kleene, "Representation of Events in Verve Nets and Finite Automata," *In C. E. Shannon and J. McCarthy: Automata Studies*, Princeton University Press, pp. 3-42, 1956.
- [10] S. Warshall, "A Theorem on Boolean Matrices," *Journal of ACM*, Vol. 9, No. 1, pp. 11-12, 1962.
- [11] S. J. Russell, P. Norvig, "Artificial Intelligence: A Modern Approach, Second Edition," Upper Saddle River, NJ: Prentice Hall, pp. 111-114, 2003.
- [12] D. B. Johnson, "Efficient Algorithms for Shortest Paths in Sparse Networks," *Journal of ACM*, Vol. 24, No. 1, pp. 1-13, 1977.
- [13] D. Rina, J. Pearl, "Generalized Best-First Search Strategies and The Optimality of A*," *Journal of the ACM*, Vol. 32, No. 3, pp. 505-536, 1985.
- [14] P. E. Hart, N. J. Nilsson, B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Transactions of System Science and Cybernetics SSC4*, No. 2, pp. 100-107, 1968.