
추상 그래프를 활용한 경로 탐색 알고리즘의 구현 및 성능 평가

김지수* · 이지완* · 조대수**

Implementation and Evaluation of Path-Finding Algorithm using Abstract Graphs

Ji-Soo Kim* · Ji-wan Lee* · Dae-Soo Cho**

요 약

최근 단말기 기반의 경로 탐색에서도 동적인 정보를 반영하기 위한 연구가 진행되고 있다. 제시하는 대부분의 알고리즘은 A*알고리즘을 기반으로 한다. 휴리스틱을 이용한 알고리즘에서는 탐색 비용이 증가하는 문제가 발생할 수 있다. 추상 그래프는 실제 도로 네트워크를 단순화한 그래프로, 휴리스틱의 의존성과 탐색 비용을 줄이기 위해 제안된 방법이다.

이 논문에서는 생성 방법이 다른 두 가지의 추상 그래프를 이용하여 성능 평가 하였다. 추상 그래프는 생성 방법에 따라 동일 특성 노드 합병을 통한 추상 그래프(AG^H)와 연결 노드 합병을 통한 추상 그래프(AG^C)로 구별된다. 성능 실험 결과 생성 비용과 노드 접근 비용 측면에서 AG^C 가 좋은 성능을 보였지만, 탐색된 경로의 이동시간 측면에서는 AG^H 가 좋은 성능을 보였다.

ABSTRACT

Recently, Many studies have been progressing to path-finding adapted dynamic information on the Terminal Based Navigation System(TBNS). The algorithms proposed are based on A* algorithm. Path-finding algorithms which use heuristic function may occur a problem of the increase of exploring cost. Path-finding with an abstract graph which expresses real road network as a simple graph is proposed for reducing dependency of heuristic and exploring cost.

In this paper, two abstract graph that are different method of construction, Homogeneous Node merging(AG^H) and Connected Node Merging(AG^C), are implemented. In result of evaluation of performance, AG^C has better performance than AG^H at construction cost and the number of node access but AG^C has worse performance than AG^H at exploring cost.

키워드

동적 휴리스틱, 탐색 영역 가지치기, A* 알고리즘, 추상 그래프

Key word

Dynamic Heuristic, Pruning Search Space, A* Algorithm, Abstract Graph

* 동서대학교 컴퓨터정보공학부

접수일자 : 2009. 10. 30

** 동서대학교 컴퓨터정보공학부 조교수(교신저자)

I. 서 론

일반적으로 단말기 기반의 시스템에서는 하드웨어 성능을 고려하여 A* 또는 변형 A* 알고리즘[1,2,3,4]을 사용한다. 최근 TPEG[5,6]와 같은 방송 채널을 통해 도로의 차량 속도, 실시간 유고 정보 등의 동적인 정보를 단말기 기반의 시스템에서도 활용할 수 있게 되었다.

현재까지 도로의 상황을 고려한 경로 탐색 알고리즘으로는 [2,3,4]가 있다. 그러나 제시하는 대부분의 알고리즘은 A*알고리즘을 변형한 형태이다. 일반적으로 휴리스틱을 이용한 알고리즘에서는 탐색 비용이 증가하는 문제가 발생할 수 있다. 휴리스틱에 의해 결정된 추정 경로에 실제 경로가 존재하지 않을 경우, 휴리스틱 가중치 값이 비슷한 2가지 이상의 경로가 존재할 경우 탐색 비용이 증가하게 된다.

휴리스틱을 이용한 경로 탐색에서의 문제점을 해결하기 위해 추상 그래프를 이용한 경로 탐색 방법을 제안하였다[7,8]. 추상 그래프는 실제 도로 네트워크를 단순화한 그래프로, 경로 탐색에 활용하여 휴리스틱 의존성과 탐색 비용을 줄이는 방법이다.

이 논문에서는 생성 방법이 다른 두 추상 그래프를 이용하여 성능 평가를 수행하였다. 추상 그래프 생성 방법에 따라 동일 특성 노드 병합을 통한 추상 그래프(AG^H)[7]와 연결 노드 병합을 통한 추상 그래프(AG^C)[8]로 분류되어진다. 성능 실험 결과 추상 그래프 생성 비용과 노드 접근 횟수 측면에서 AG^C가 AG^H에 비해 좋은 성능을 보였으나, 탐색된 경로의 이동 시간 측면에서는 AG^H가 AG^C에 비해 좋은 성능을 보였다.

II. 관련 연구

A* 알고리즘[1]은 가장 범용적으로 사용되고 있으며, 특히 AI 분야에서 많이 쓰이는 방법이다. A* 알고리즘은 노드 평가 함수 $F=G+H$ 를 사용하여 다음 탐색될 노드를 선택한다.

가변적인 그리드 속력을 이용한 경로탐색 기법[2]은 실시간 교통정보를 활용할 경우를 대비한 경로탐색 기법이다. 이 경로탐색 기법은 경로 탐색 시 우선권을 가지는 노드를 탐색할 때 고정된 데이터인 거리만으로 평가하지 않고 가변적인 데이터인 각 도로의 속도정보를 고

려한 탐색을 한다. 즉, 시간(T)= $\text{거리}(S)/\text{속력}(V)$ 를 이용하여 이동 시간(T)을 노드 평가 시 가중치로 사용하였다. 최저속력 동적 휴리스틱[3]은 그리드 셀의 최저속력을 동적 휴리스틱으로 사용한다. 경로 탐색 시 최저속력이 낮은 구역을 탐색범위에서 제외시킴으로 속력이 낮은 구간을 탐색할 경우의 수를 줄임으로서 경로의 길이를 크게 향상시켰다. 그러나 경로 탐색에서 제외된 그리드 내에 주요 교차로나 강 또는 하천 등을 가로지르는 다리가 있는 경우 크게 돌아가는 경로를 탐색해야 하므로 탐색 비용이 증가하는 단점이 있다.

최고 속도 동적 휴리스틱[4]을 이용한 경로 탐색 알고리즘은 A*알고리즘을 기반으로 하지만, 열린 목록에서 노드 추출 방법을 2단계 노드 추출로 변형하였다. 1차 노드 추출은 고정된 데이터인 거리 정보를 이용하여 우선순위가 정해진 노드 목록을 추출하고, 2차 노드 추출은 1차로 추출된 노드목록에서 우선순위가 높은 N개의 노드들을 대상으로 해당 노드를 포함하고 있는 그리드의 최고 속도 정보를 이용하여 최우선 순위 노드를 추출하게 된다. 동적 휴리스틱을 이용함으로써, 경로의 길이는 높일 수 있었으나, 2단계 노드 추출을 이용하기 때문에 탐색 비용이 증가하는 단점이 있다.

III. 추상 그래프를 이용한 경로 탐색

추상 그래프를 생성하기 위해서는 실제로 도로 네트워크와 그리드 셀을 기반으로 경계 그래프를 생성하여야 한다. 경계 그래프를 구성하는 경계 노드는 도로와 그리드 셀의 경계의 교차 점을 의미한다. 경계 간선은 경계 노드간의 연결을 의미하며, 경계 노드 간에 존재하는 경로가 오직 하나의 그리드 셀 내에 존재할 경우 생성된다. 그림 1.(b)는 그림 1.(a)의 예제 그래프를 기반으로 생성된 경계 그래프를 보여준다.

추상 그래프는 동일 특성 노드 병합을 통한 추상 그래프(AG^H)와 연결 노드 병합을 통한 추상 그래프(AG^C)로 분류된다. 동일 특성 노드 병합과 연결 노드 병합은 경계 그래프를 구성하는 경계 노드와 경계 간선의 수를 줄이기 위한 기법이다.

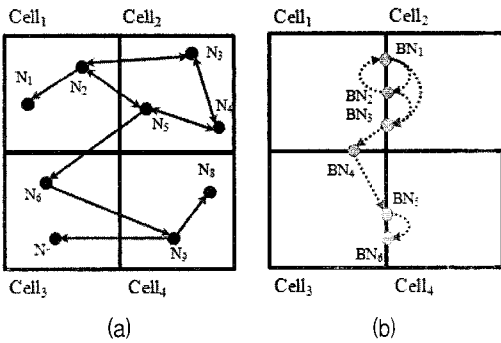


그림 1. 경계 그래프 생성
 (a) 예제 그래프 (b) 경계 그래프
 Fig 1. Construction of Boundary Graph
 (a) Example of Graph (b) Boundary Graph

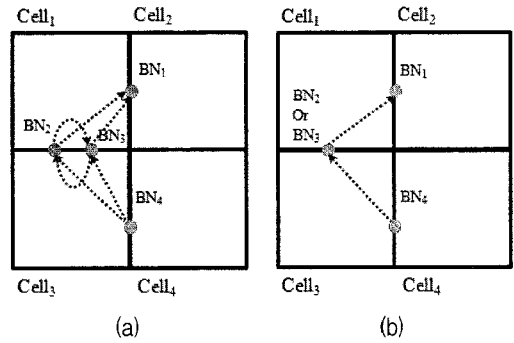


그림 2 동일 특성 노드 병합 예제
 (a) 예제 경계 그래프 (b) 동일 노드 병합 후 경계 그래프(추상 그래프)
 Fig2. Example of Merging Homogeneous Nodes
 (a) Example of Boundary Graph (b) Boundary Graph after Merging Homogeneous Nodes

3.1 동일 특성 노드 병합을 통한 추상 그래프 생성[7]

경계 그래프는 인접 셀 간의 이동 경로의 유무를 표현한 그래프이다. 하나의 경계 노드는 두 개의 인접 셀 Cell_x, Cell_y를 가지며, Cell_x와 Cell_y간에 경로가 있음을 의미한다. 예를 들어 그림 1.(b)에서 경계 노드 BN₁은 두 인접 셀 Cell₁과 Cell₂를 가지므로, Cell₁과 Cell₂간에 경로가 있음을 의미한다. 두 경계 노드 간에 경계 간선이 존재한다는 것은 두 경계 노드가 가지는 인접 셀 간에 경로가 존재한다는 것이다. 예를 들어 경계 노드 BN₃과 BN₄간에 경계 간선이 존재한다는 것은 Cell₂에서 Cell₁을 거쳐 Cell₃로 가는 경로가 존재함을 의미한다. 그런데 인접 셀 간의 경로의 유무를 표현하고 있는 경계 노드와 경계 간선들 중에는 중복된 의미를 가지는 경계 노드와 경계 간선이 존재한다.

동일 특성 노드란 동일한 인접 셀을 가진 경계 노드 중에 자신들을 제외한 들어오는 노드와 나가는 노드 정보가 동일한 노드를 의미한다. 예를 들어 그림 2.(a)에서 BN₂와 BN₃은 인접 셀이 Cell₁과 Cell₃으로 동일하며 자신들을 제외한 들어오는 노드는 BN₄로 동일하고 나가는 노드는 BN₁으로 동일하다. 따라서 두 경계 노드가 가지는 연결 정보의 의미는 동일하기 때문에 그림 2.(b)와 같이 하나의 노드로 표현해도 경계 그래프의 의미는 동일하게 표현할 수 있다. 그림 3.(b)는 그림 3.(a)의 경계 그래프를 동일 특성 노드 병합으로 생성된 추상 그래프(AG^H)를 보여주고 있다.

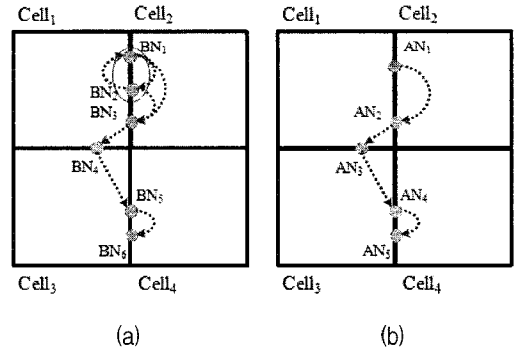


그림 3. 동일 노드 병합을 통한 추상 그래프 생성(AGH)
 (a) 예제 경계 그래프 (b) 노드 병합 후 경계 그래프(추상 그래프)
 Fig3. Abstract Graph Created by Merging Homogeneous Nodes(AGH)
 (a) Example of Boundary Graph (b) Boundary Graph after Merging Nodes

3.2 연결 노드 병합을 통한 추상 그래프 생성[8]

연결 노드 병합 기법은 동일한 인접 셀을 가지는 경계 노드들 중에 경계 간선으로 연결되어 있을 경우 병합하는 기법이다. 경계 노드는 두 인접 셀 간에 경로가 존재함을 의미하며, 경계 간선은 인접 셀 간의 이동 경로 유무를 표현한 것이다. 그런데 동일한 인접 셀을 가지는 경계 노드들 간에 경계 간선으로 연결되어 있을 경우 경계

노드가 가지는 의미와 경계 간선이 가지는 의미가 동일하다. 예를 들어 그림 4.(a)의 예제 경계 그래프에서 BN_1, BN_2, BN_3 이 가지는 의미는 $Cell_1$ 과 $Cell_2$ 간에 경로가 존재한다는 것이며, 경계 간선으로 연결되었을 경우 $Cell_1$ 과 $Cell_2$ 간에 경로가 존재한다는 의미를 지니게 된다. 따라서 경계 노드가 가지는 의미와 경계 간선이 가지는 의미가 동일하다는 것이다. 연결 노드 병합 기법은 동일한 인접 셀을 가지는 경계 노드 중에 경계 간선으로 연결되어 있는 경계 노드들을 하나의 노드로 표현하여 노드와 간선의 수를 줄이는 것이다. 그림 4.(b)는 그림 4.(a)의 경계 그래프를 연결 노드 병합으로 생성된 추상 그래프(AG^C)를 보여주고 있다.

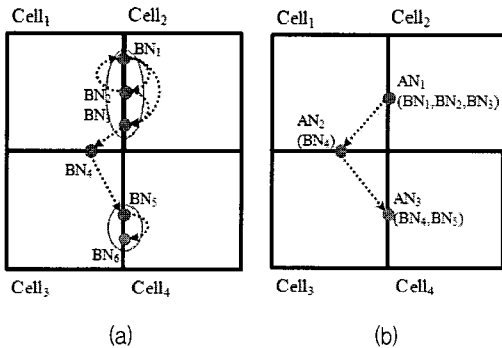


그림 4. 연결 노드 병합을 통한 추상 그래프 생성(AGC)
(a) 예제 경계 그래프 (b)노드 병합 후 경계 그래프(추상 그래프)

Fig 4. Abstract Graph Created by Connection Nodes(AGC) (a) Example of Boundary Graph (b)Boundary Graph after Merging Nodes

3.3 추상 그래프를 활용한 경로 탐색

추상 그래프를 활용한 탐색 방법은 추상 그래프 경로 탐색과 실제 경로 탐색 2단계로 이루어진다. 1단계 탐색은 실제 네트워크에 비해 노드의 수와 간선의 수가 적은 추상 그래프에서 A^* 알고리즘 탐색을 수행하며, 2단계 탐색은 1단계 탐색의 결과를 기반으로 실제 네트워크에서 A^* 알고리즘 탐색으로 수행된다. 노드 평가 방법은 시간(T)= $\text{거리}(S)/\text{속력}(V)$ 를 이용하여 이동 시간(T)을 노드 가중치로 사용하며, 해당 노드를 포함하고 있는 그리드의 속력 정보를 적용한다. A^* 알고리즘의 휴리스틱 설정은 [9]에서 제안한 고정 그리드 기반의 추정치 감소화

기법을 이용한다.

추상 그래프 탐색을 통해 경로의 존재여부, 탐색 범위를 추출할 수 있다. 실제 경로 탐색은 추상 탐색의 결과인 추상 경로를 포함하는 유효 셀 범위 내에서 수행된다. 추상 경로를 포함하는 셀들을 유효 셀이라 한다.

IV. 성능 실험 및 비교 분석

이 장에서는 생성 방법이 다른 두 추상 그래프에 대해 성능 실험을 하였다. 그리드 셀의 크기에 따른 추상 그래프 생성 비용을 비교하고 생성된 두 가지의 추상 그래프를 활용하여 탐색 성능을 비교한다.

실험은 실제 부산시내 도로 데이터로서, 42,357개의 노드와 106,254개의 간선으로 구성되어 있다. 그리드 생성을 위해 전체 지도 크기는 부산시내 데이터를 구성하는 노드의 최소 좌표와 최고 좌표를 고려하여 가로 45,487m, 세로 39,996m로 설정하였다.

그리드는 추상 그래프 생성과 탐색에 핵심적인 역할을 담당하기 때문에 가로, 세로 길이가 각 500m, 1,000m, 1,500m, 2,000m인 4개의 그리드 셀을 준비하여 성능 실험에 사용하였다. 탐색 성능 실험은 출발지와 목적지까지의 직선거리를 10km미만, 10~15km미만, 15~20km미만, 20~30km미만, 30~40km미만으로 5등급으로 나누어, 각각 200개의 출발지와 목적지를 추출하여 탐색을 수행하였다. 또한 크기가 다른 4개의 그리드 셀별로 탐색 성능을 비교하였다.

4.1 추상 그래프 생성 비용 비교

4개의 그리드 셀을 활용하여 추상 그래프 생성 비용을 비교해 보았다. 표 1은 그리드 셀을 기반으로 추상 노드 수, 추상 간선 수, 추상 그래프 생성 시간을 보여주고 있다. 그리드 셀의 크기가 커질수록 생성되는 추상 노드 수와 추상 간선 수가 적게 생성됨을 알 수 있다. 각 추상 그래프별 생성된 추상 노드 수와 추상 간선 수를 비교하였을 때 추상 노드 수는 약 2배, 추상 간선 수는 약 4배 AG^H 가 AG^C 에 비해 많았다.

표 1. 그리드 셀 크기별 추상 그래프 생성 비용 비교
Table 1. Comparison of Construction cost between AGH and AGC

	AG ^H				AG ^C			
	500m	1,000m	1,500m	2,000m	500m	1,000m	1,500m	2,000m
추상 노드 수	4,224	1,649	633	280	2,791	958	496	311
추상 간선 수	24,861	10,408	4,453	2,340	5,372	1,826	964	603
추상 그래프 생성 시간	663.91	958.01	1652.73	2,039.4	424.3	119.2	68.7	53.4

4.2 경로 탐색 성능 비교

각 추상 그래프를 활용한 경로 탐색의 성능을 비교해 보았다. 비교 내용은 추상 그래프 탐색의 결과인 유효 셀의 수, 실제 경로 탐색의 결과인 노드 접근 횟수와 탐색된 경로 비용을 비교하였다. 탐색된 경로의 비용 평가는 산출된 경로의 [각 간선의 길이]/[각 간선의 속력]의 공식을 이용하여 이동시간을 평가하였다.

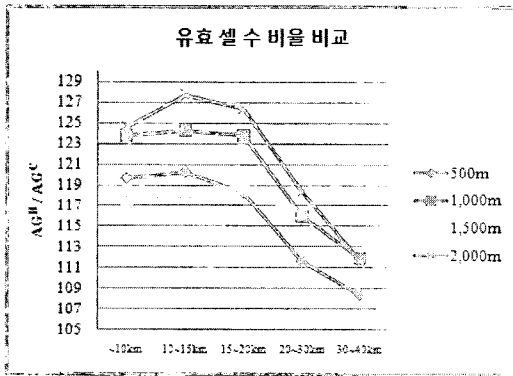


그림 5. 유효 셀 수 비교
Fig 5. Comparison of The Number of Valid-Cell

그림 5, 그림 6, 그림 7은 AG^C에 대한 AG^H의 탐색된 유효 셀 수, 노드 접근 횟수와 탐색된 경로의 이동시간의 비율을 보이고 있다. 탐색된 유효 셀의 수가 많을수록 실제 탐색에서 접근하게 될 노드의 수가 많을 수가 있다. 그림 5에서는 보여주듯이 AG^H가 AG^C에 비해 약 9~28% 많은 유효 셀을 추출하였으며, 이에 따라 노드 접근 횟수도 많음을 그림 6를 통해 알 수 있다.

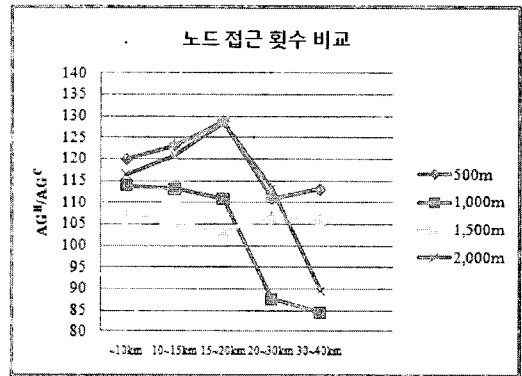


그림 6. 노드 접근 횟수 비교
Fig 6. Comparison of The Number of Node Access

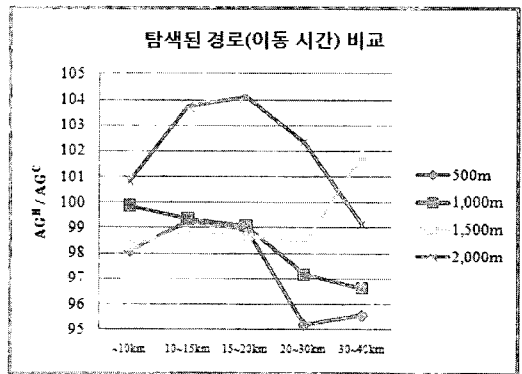


그림 7. 탐색된 경로(이동 시간) 비교
Fig 7. Comparison of Path Quality

그림 7은 탐색된 경로에 대한 결과를 보이고 있다. 실험 결과 전반적으로 AG^H가 AG^C에 비해 좋은 경로를 탐색함을 알 수 있다. 그러나 그리드 셀의 크기가 가장 큰 2,000m에서는 유효 셀 수와 노드 접근 횟수뿐만 아니라 탐색된 경로 또한 좋지 못함을 알 수 있다.

V. 결론 및 향후 과제

휴리스틱을 이용한 알고리즘에서는 탐색 비용이 증가하는 문제가 발생할 수 있다. 추상 그래프는 실제 그래프를 단순화한 그래프로, 추상 그래프에서 우선 탐색을 통해 경로 존재 여부를 판별하고 탐색 영역을 추출하여 탐색 비용을 줄이기 위해 제안된 방법이다. 추상 그래프 생성 방법에 따라 동일 특성 노드 병합을 통한 추상 그래프(AG^H)와 연결 노드 병합을 통한 추상 그래프(AG^C)로 구분된다.

이 논문에서는 생성 방법이 다른 두 추상 그래프를 이용하여 성능 실험을 하였다. 생성 비용과 노드 접근 측면에서는 AG^C 가 AG^H 에 비해 좋은 성능을 보였지만, 탐색된 경로 비용 측면에서는 AG^H 가 AG^C 에 비해 좋은 성능을 보였다. 향후, 기존에 존재하는 알고리즘과 탐색 성능 평가를 수행하고, 최적의 추상 그래프를 생성하기 위한 연구가 진행되어야 할 것이다.

참고문헌

[1] Peter E. Hart, Nils J. Nilsson and Bertram Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths in Graphs," IEEE Trans. on Systems Science and Cybernetics, Vol. SSC-4, No. 2, pp 100-107, 1968

[2] 이현섭, 김진덕, "고정 그리드 기반 가변 휴리스틱을 이용한 최적경로 탐색," 한국해양정보통신학회 2005 추계 종합학술대회, Vol.9, No.2, pp.137-141, 2005

[3] 문대진, 조대수, "최저 속력 동적 휴리스틱을 이용한 경로 탐색", 한국공간정보시스템학회 2008 Vol.19, No2, 2008

[4] 문대진, 조대수, "실시간 도로 정보를 이용한 최고속력 동적 휴리스틱의 설계", 한국해양정보통신학회 2008 추계 종합학술대회, Vol.12, No.1, pp.827-830, 2008

[5] <http://en.wikipedia.org/wiki/TPEG>

[6] EBU B/TPEG, "Transport Protocol Experts Group (TPEG) TPEG specifications - Part 1: Introduction, Numbering and Versions", TPEG-INV/002, draft, October 2002.

[7] 김지수, 이지완, 조대수 "동적 교통 정보를 적용하기 위한 도로망 추상화 기법의 설계", 2009 추계 종합학술대회, Vol.13, No.1, pp.199-202, 2009

[8] 이지완, 조대수 "동적정보를 이용하는 환경에서 탐색영역 최소화를 위한 이웃노드 병합 추상그래프 설계", 한국공간정보시스템학회 2009 추계학술대회, pp.220-226, 2009

[9] 김진덕, "가변 휴리스틱 기반 추정치 간소화를 통한 경로 탐색 기법의 설계 및 성능평가", 한국해양정보통신학회논문지 Vol 10, No 11, pp2002-2007, 2006

저자소개

김지수(Ji-soo Kim)

한국해양정보통신학회 논문지
제13권 제8호 참조

이지완(Ji-wan Lee)

한국해양정보통신학회 논문지
제13권 제8호 참조

조대수(Dae-Soo Cho)

한국해양정보통신학회 논문지
제13권 제8호 참조