

# 논문 글쓰기 방법

## (지능화 파일럿 프로젝트 3주차 특강)

# 목차

---

## 1. 학습 목표

## 2. 논문 작성 Tip

## 3. 논문의 구성

## 4. 논문 작성의 원칙

## 5. 논문 글쓰기 방법

- |           |           |
|-----------|-----------|
| (1) 제목    | (5) 본론    |
| (2) 초록    | (6) 실험 결과 |
| (3) 서론    | (7) 결론    |
| (4) 관련 연구 | (8) 참고문헌  |

## 6. 참고 자료

# 학습 목표

---

- 공학 석사 학위 논문 글쓰기 방법 제시  
➔ 산업인공지능학과 석사 학위 논문 (보고서)
- 논문 글쓰기 방법의 이해
- 논문 구성 요소의 이해
- 논문 구성 요소별 포함해야 할 내용 이해

# 논문 작성 Tip

---

- 논문을 작성하는 것은 공부가 아니라 일이다.
- 매일 조금씩 일하는 것                      vs.                      몰아서 일하는 것
- 논문 집필을 위해 충분한 시간을 확보해야 함

# 논문 작성 Tip

- 충북대 도서관 홈페이지에서 석사 학위 논문 검색해 볼 것 (가능하면 지도 교수님 담당 학생)  
: 학위 논문 양식 참고
- 주요 항목 제목 작성  
: 본론1, 본론2, 본론3... 에 해당하는 소제목 → 주요 방법론이 잘 드러나도록 선정
- 개요 작성 → 구성 항목별 작성
- 논문 쓰는 순서  
: 제목 (가칭) → 본론(본문) → 실험 결과 → 결론 → 서론 → 초록 → 제목

# 논문의 구성

---

1. **제목 (Title):** 목표형, 방법형, 결과형
2. **초록 (Abstract):** 핵심 결과, 동기, 나의 방법, 논문의 결과, 일반적 응용
3. **서론 (Introduction):** 연구의 목적, 연구의 필요성, 문제 제기, 본 논문의 기여점
4. **관련 연구 (Related Work):** 선행 연구 조사
5. **본론:** 연구 방법론 서술
6. **실험 결과 (Simulation / Result):** 연구 수행한 후의 도출된 결과
7. **결론 (Conclusion):** 결과 요약 및 의미 해석, 한계점, 후속 연구, 학문적/실무적 시사점, 응용 분야
8. **참고논문 (Reference)**

# 논문 작성의 원칙

1. 논문의 기여가 무엇인지를 분명하게 할 것
2. 전달하고자 하는 메시지를 추측하지 않도록 할 것 → Readers are lazy!
3. 자세히 설명할 것 → 독자가 논문을 열심히 읽어줄 것을 기대하지 말 것
4. 관련 연구를 충분히 조사할 것
5. 그림을 잘 그릴 것 → 어려운 개념을 쉽게 설명 가능
6. 다른 결과와의 비교
7. 연구의 한계점 기술
8. (결론) 후속 연구 및 연구의 강점 기술

# 논문 글쓰기 방법 (제목)

1. 논문의 핵심적인 내용, 목표 및 방법론이 포함되도록
2. 너무 뻘하거나, 당연해 보이는 내용이 포함되지 않도록 (sales 필요!)
3. 지도 교수님과 상의 必 → feedback 필요

(좋은 예시)

- 디지털트윈 기반의 스마트시티 3차원 태양광 발전 시뮬레이션 시스템 개발
- 딥 러닝 알고리즘을 활용한 뇌파 분석 기반 졸음운전 사고예방 시스템
- 딥러닝을 활용한 마스크 착용 얼굴 체온 측정 시스템
- 빅데이터를 이용한 인공지능 기반 시간별 전력수요 예측
- 인공지능 기반의 자세 추정에 따른 모바일 헬스케어 동작 연구
- 자연어 처리를 이용한 감정 스트레스 인지 및 관리 챗봇 개발
- 자율주행 자동차를 위한 적응형 브레이크 시스템 소프트웨어 플랫폼 개발
- 저온특성을 향상시키기 위한 리튬이온 배터리 분리막 설계
- 차량용 카메라 센서를 이용한 딥러닝 기반의 자율주행 주행가능영역 검출 개발

(잘못된 예시)

- 인간-로봇 협업 시스템  
→ 방법론 부재
- 주행 로봇의 충돌 회피 기술  
→ 방법론 부재
- 딥러닝 연구  
→ 주제가 지나치게 광범위함
- 장애물 감지를 위한 데이터셋 기반의 딥러닝 학습 기법  
→ 너무 뻘한 내용



# 논문 글쓰기 방법 (초록)

\* 목표: **내 글을 읽도록 만들자!**

핵심: 이 글을 읽을만한 결과가 있는가?

\* 4개의 주요 문장으로 구성됨

1. 문제 정의 (state the **problem**)
2. 이 논문이 왜 흥미로운가? (say **why** it is interesting)
3. 문제를 해결하기 위한 해결책/제안점은 무엇인가? (say **what your solution** achieves)
4. 논문의 결과가 어떤 의의가 있는가? (say **what follows** form your solution)

## Abstract

Autonomous mobile robots need very reliable navigation capabilities in order to operate unattended for long periods of time. This paper reports on first results of a research program that uses partially observable Markov models to robustly track a robot's location in office environments and to direct its goal-oriented actions. The approach explicitly maintains a probability distribution over the possible locations of the robot, taking into account various sources of uncertainty, including approximate knowledge of the environment, and actuator and sensor uncertainty. A novel feature of our approach is its integration of topological map information with approximate metric information. We demonstrate the robustness of this approach in controlling an actual indoor mobile robot navigating corridors.

→ 일반적인 문제에 대한 정의 및 "why" 설명

→ 이 논문에서 다루고자 하는 것이 무엇인지에 대한 설명

→ 문제 해결을 위해 이 논문에서 제시하는 방법론에 대한 설명

→ 무엇이 새로운지, 더 좋은 점인지 강조함

→ 제시하는 방법론의 장점에 대한 결과에 대해 언급함

(파파고 번역) 자율 이동형 로봇은 장시간 무인 작동하기 위해 매우 안정적인 항법 기능이 필요하다. 이 논문은 부분적으로 관찰 가능한 마르코프 모델을 사용하여 사무실 환경에서 로봇의 위치를 강력하게 추적하고 목표 지향적인 행동을 지시하는 연구 프로그램의 첫 번째 결과에 대해 보고한다. 이 접근법은 환경에 대한 대략적인 지식과 작동기 및 센서 불확실성을 포함하여 불확실성의 다양한 소스를 고려하여 로봇의 가능한 위치에 대한 확률 분포를 명시적으로 유지한다. 우리 접근 방식의 새로운 특징은 위상 지도 정보와 대략적인 메트릭 정보를 통합하는 것이다. 우리는 실제 실내 이동 로봇 복도를 제어할 때 이 접근 방식의 견고성을 입증한다.

# 논문 글쓰기 방법 (초록) – 잘못된 예시

판단 기준이 무엇인가? 실체가 없는 정의임

요 약

문제 동기를 설명.  
아무런 근거가 없고, 동의하기 어려운 내용

이 논문에서는 판단 기준(decision criteria)을 적용한 인간-로봇 협업 시스템을 제시한다. 로봇은 인간보다 환경 및 물체 인식 능력이 떨어지기 때문에 인식에 혼동이 발생하기 쉽다. 인간의 높은 인지 능력을 이용하는 것은 작업을 원활히 진행하는 데에 유용하다. 또한, 로봇에게 모든 판단 권한을 부여하기 위해 고비용을 들여서 많은 센서 장착 및 복잡한 시스템 설계를 하기보다 필요한 시점에 인간의 도움을 받는 것은 최소한의 시스템 구축 비용을 위해서 이득이다. 본 논문에서는 인간의 개입 정도를 정량화한 수치인 판단 기준을 제안하고, 이를 적용하여 인간-로봇 협업 시스템을 구성했다. 시스템을 검증하기 위해 e-puck 로봇 3대를 이용하여 모의 광물 채취 실험을 수행했다. 제시된 기술은 제한된 로봇의 인지 능력을 인간의 도움을 통해 개선할 수 있는 분야인 채집(foraging), 물체 인식(object recognition), 환경 인식(environment recognition) 등에 활용될 수 있다.

필요한 시점이 무엇인가?  
맥락이 없음

인간의 도움이 어떻게 활용  
분야에서 로봇의 인지 능력을 개선시키는가?

이론적 근거 또는 구체적인 방법론이 없음

- 풀고자 하는 문제 동기에 대한 근거 부족
- 구체적인 방법론 제시 안됨
- 다수의 모호한 표현 (예: 인식, 인지, 판단, 도움 등)

# 논문 글쓰기 방법 (서론)

\* 내 논문이 왜 필요한지를 분명하게 제시  
결과를 미리 보여줘서 관심을 계속 유도

\* 서론의 구성

1. 연구의 동기 (Start with a motivation)
2. 이 연구의 방법론은 무엇인지 (Tell what this paper is about)
3. 이 연구와 관련된 내용 서술 (Explain what makes this work relevant)
4. 이 논문의 기여점
5. (option) 논문 구성 소개

# 논문 글쓰기 방법 (서론) - 좋은 예시

Simmons, Reid, and Sven Koenig. "Probabilistic robot navigation in partially observable environments." IJCAI. Vol. 95. 1995.

(파파고 번역)

우리는 사무실 환경(복도, 화장실 및 방이 있는)에서 장기 자율 항법 작업에 관심이 있다. 자율 사무용 내비게이션의 최첨단 기술은 상당히 발달되어 있지만, 로봇이 길을 잃지 않고 장시간 복도를 횡단할 수 있는 것은 일반적으로 충분하지 않다. 이에 대한 증거는 최근 AAAI가 후원하는 로봇 대회에서 볼 수 있는데, 로봇들은 종종 그들이 어디에 있는지 혼동하고 그것이 발생하면 다시 현지화하는 데 어려움을 겪었다.

우리는 로봇이 공간 및 센서 불확실성을 명시적으로 표현함으로써 내비게이션의 신뢰성을 높일 수 있다고 주장한다.

→ 연구의 동기 설명

(파파고 번역)

이를 위해, 우리는 마르코프 모델을 사용하여 로봇의 위치를 강력하게 추적하고 방향을 지시하는 항법 기술을 개발했다. 부분적으로 관찰 가능한 마르코프 의사 결정 프로세스(POMDP) 모델은 환경의 연결성, 대략적인 거리 정보, 센서 및 작동기 특성에 대한 위상 정보로부터 구성된다. 마르코프 모델은 확률 분포의 형태로 로봇의 위치를 추정합니다. 로봇이 이동 또는 회전했다고 보고할 때와 벽 및 코리더 접합과 같은 기능을 관찰할 때 확률이 업데이트됩니다. 로봇의 동작을 지시하기 위해 설계자는 지시(예: 회전 또는 정지)를 모든 마르코프 상태와 연결합니다. 마르코프 모델의 확률 분포가 업데이트될 때마다 각 지시문에 대한 총 확률 질량이 계산되고 로봇은 확률 질량이 가장 큰 것을 실행합니다.

→ 논문의 방법론 설명

논문과 관련된 내용 설명

cision process (POMDP) model is constructed from topological information about the connectivity of the environment, approximate distance information, plus sensor and actuator characteristics. The Markov model estimates the position of the robot in the form of probability distributions. The probabilities are updated when the robot reports that it has moved or turned, and when it observes features such as walls and corridor junctions. To direct the robot's behavior, a planner associates a directive (e.g., turn or stop) with every Markov state. Whenever the probability distribution of the Markov model is updated, the total probability mass for each directive is calculated, and the robot executes the one with the largest probability mass.

Our approach has several features that make it well-suited for the office navigation task. It explicitly accounts for uncertainty in actuation, sensor data and their interpretation, and the robot's position. It can utilize all available sensor information to track position, and is particularly amenable to adding new sources of sensor information. It seamlessly combines topological and metric map information, enabling the robot to utilize as much, or as little, metric information as it has available. It is also very reactive – once the robot believes it has strayed from the nominal (optimal) path, it will automatically execute corrective actions. On the other hand, it is relatively immune to temporary uncertainty in position. For example, even if the robot does not know for certain which of two parallel corridors it is traversing, it does not stop and replan, as long as the control directives associated with both corridors are the same. In this way, it can continue making progress towards its desired goal, while at the same time collecting sensor readings to help disambiguate its true location.

An important aspect of this work is that it must run in real time on board an actual robot. Problems include not only how to model the navigation problem as a POMDP, but also how to deal with memory and time constraints. While still preliminary, our experimental results, both in simulation and on the actual robot, are encouraging. In particular, they indicate that the approach produces very robust navigation, even when using estimates of the actual sensor and action models. While, to date, we have concentrated more on implementation and validation aspects of the approach, our work opens up new application areas for more theoretical results

## 1 Introduction

We are interested in the task of long-term autonomous navigation in an office environment (with corridors, foyers, and rooms). While the state of the art in autonomous office navigation is fairly advanced, it is not generally good enough to permit robots to traverse corridors for long periods of time without getting lost. Evidence for this can be seen in recent AAAI-sponsored robot competitions [Konolige, 1994; Simmons, 1995], where the robots often got confused as to where they were, and had difficulty relocalizing once that occurred.

We contend that navigation can be made more reliable by having the robot explicitly represent spatial and sensor uncertainty. To this end, we have developed a navigation technique that uses Markov models to robustly track the robot's position and direct its course. A partially observable Markov de-

This research was supported in part by NASA under contract NAGW-1175 and by the Wright Laboratory and ARPA under grant number F33615-93-1-1330. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of NASA, the Wright Laboratory, or the United States government.



## 1. 서론

로봇은 인간보다 낮은 수준의 인지 능력을 가지고 있기 때문에 고도의 판단을 필요로 하는 작업 수행 시에 로봇에게 모든 판단 권한을 주는 것은 작업의 위험성을 증가시킨다. 예를 들어, 로봇이 용접 작업을 할 때 비전 및 레이저 센서를 이용하여 3-D 환경 정보를 알아낼 수 있지만 에러가 존재하기 때문에 잘못된 판단을 내려 원치 않는 결과가 발생할 수 있다[인식]. 또한 주변 환경 변화에 변화가 생겼을 때, 로봇의 인지 능력에는 한계가 존재한다[1].

따라서 인간-로봇 협업을 통해 로봇의 제한된 인지 능력에 대해서 인간의 도움을 받아 작업을 수행하는

것은 효율성(efficiency)과 적응성(adaptive) 측면에서 이점을 갖는다[그림 1]. 즉, 완전한 자율(fully-



[그림 1] 인간-로봇 협업(Iron Man, 2008)

→ 이런 그림은 안 넣는게 좋음

제안하는 방법론에 대한 구체적인 설명 부족

autonomous) 로봇을 위해 복잡한 프로세스를 설계하기보다 판단이 필요한 시점에 인간의 인지 및 사고 능력을 이용하여 작업을 수행하면, 최소한의 시스템 설계 비용을 들여 원하는 결과를 얻을 수 있다.

초창기의 인간과 로봇의 상호 작용(interaction)은 단순히 인간이 원하는 명령을 내리면 로봇이 명령을 수행하는 수준이었다. 이를테면, 로봇에게 조이스틱이나 리모컨으로 작동 명령을 내리면 로봇은 내부에 입력된 행동을 수행하는 방식이었다[2]. 로봇이 지능화 되면서 인간과 로봇은 서로에게 영향을 주는 관계를 갖게 되었고 이를 인간-로봇 상호작용(Human-Robot Interaction, HRI)이라고 부른다. 이러한 상호 작용을 통해 인간과 로봇이 팀을 이루어서 작업을 수행하는 일이 가능하게 되었다[3].

협업(collaboration)은 같은 목표를 위해 최소 두 개 이상의 개체가 그룹을 이루어 작업을 수행하는 과정을 의미한다. 인간과 로봇은 협업을 통해 작업 수행 능력을 향상시킬 수 있는데, 크게 인간 에뮬레이션(Human-Emulation, HE) 방법과 인간 보완(Human-Complementary, HC) 방법이 존재한다[4]. 인간 에뮬레이션은 로봇이 인간과 동일한 기능을 가질 수 있게 하여 로봇이 인간처럼 판단하고 동작할 수 있게 한다. 인간 보완 방법은 로봇이 스스로 판단을 내리기 보다는 인간의 도우미로서 인간의 행동을 도와주는 보조적인 역할만을 담당한다.

→ 연구 분야 및 배경에 대한 설명. 앞 단락과의 호응 관계 안 맞고, 흐름상 부자연스러움

문제의 동기를 설명. 하지만, 설득력이 부족함. 에러가 있을 때 잘못된 판단이 결과에 어떤 영향을 주는지?

- 연구 동기에 대한 설득력 부족
- 초록에 이어 여전히 구체적인 방법론에 대한 설명 없음
- 분야 소개 및 배경 부분으로 이어지는 흐름이 맞지 않음

# 논문 글쓰기 방법 (관련 연구)

다른 연구자들은 이전에 어떤 연구를 하였는가?  
(What is the work previously done by others?)

기존 연구들을 기술하고, 나의 논문과의 차별성 언급  
(Describe for every other paper, how your work differs)

내 논문이 기존 연구들에 비해서 더 나은 점이 무엇인지 간단하게 요약  
(Summarize in which way your paper goes beyond the state of the art)

in the area of planning with Markov models, including some of our own group's work [Chrisman, 1992; Goodwin, 1994; Koenig and Simmons, 1994].

## 2 Related Work

Most recent work in robotic office navigation has used a *landmark-based* approach that relies on topological maps whose nodes correspond to landmarks (locally distinctive places), such as corridor junctions, and whose edges indicate how the robot should navigate between nodes [Kortenkamp and Weymouth, 1994; Kuipers and Byun, 1988]. This approach is attractive because it does not depend on geometric accuracy and is *reactive* to sensed features of the environment (the landmarks). It suffers, however, from problems of sensors occasionally not detecting landmarks and of sensor *aliasing* (not being able to distinguish between similar landmarks). On the other hand, using purely metric maps is vulnerable to inaccuracies in both the map making and dead-reckoning abilities of the robot. While some researchers augment topological maps with approximate metric information, such information is primarily used to resolve topological ambiguities [Kuipers and Byun, 1988; Mataric, 1991; Simmons, 1994]. In contrast, our Markov model approach seamlessly integrates topological, landmark-based, information and approximate metric information.

Some navigation techniques represent uncertainty in position using models that presume a certain probability distribution, typically Gaussian [Kosake and Kak, 1992; Smith and Cheeseman, 1986]. While such models are efficient to encode and update, they are not ideally suited for office navigation. In particular, due to sensor aliasing, one often wants to encode the belief that the robot might be in one of a number of non-contiguous locations. This cannot be represented precisely using Gaussian distributions, but is quite easy for our Markov models. On the other hand, we need to tessellate space into discrete states, rather than representing position using real numbers. Thus, there is a tradeoff between the precision and expressiveness of the different models. We contend that for office navigation, however, that the added expressiveness of the Markov models outweighs the decrease in precision from discretization.

Like our own work, several researchers have investigated Bayesian approaches for probabilistic planning and execution monitoring in office navigation. [Nourbakhsh *et al.*, 1995] use a partially observable Markov model approach similar to ours, but do not utilize any metric information. The states of the robot are either a topological node, or somewhere in a connecting corridor. In contrast, our approach can use estimates of how far the robot has traveled and sensor reports that occur within a corridor to further constrain the robot's location. For example, knowing that two corridor junctions are approximately 5 meters apart enables the robot to estimate when it is in the vicinity of the second junction, even if it misses seeing the junction.

Most of the other Bayesian approaches rely on metric maps. [Kirman *et al.*, 1991] and [Nicholson and Brady, 1994]

use approaches based on temporal belief networks. With such methods the size of the models grows linearly with the amount of temporal lookahead, which limits their use to rather small lookaheads. [Dean *et al.*, 1993] use robot navigation as an example to describe a planning and monitoring algorithm that uses a totally observable Markov model, which assumes that the location of the robot is always known precisely. [Hu and Brady, 1994] use Bayesian techniques to detect unforeseen obstacles in an otherwise completely known environments.

## 3 Constructing the Markov Models

Before describing how we construct Markov models of an office environment, we introduce some terminology. A finite Markov model consists of a finite set of states  $S$ , a finite set of actions  $A$ , a set of actions  $A(s) \subseteq A$  for each state  $s \in S$ , that can be executed in that state, and transition probabilities  $p(s'|s, a)$  for all  $s, s' \in S$  and  $a \in A(s)$  (the probability that the new state is  $s'$  if action  $a$  is executed in state  $s$ ). We also define a set of sensors  $i \in I$ . The sensors are characterized by observation probabilities  $p_i(o|s)$  for all  $s \in S$  and  $o \in O(i)$  (the probability that sensor  $i$  reports feature  $o$  when the robot is in state  $s$ ). Note that Markov models assume that the transition and observation probabilities are determined only by the current state of the robot (the "Markov assumption").

In our case, the Markov model is partially observable because the robot may never know exactly which state it is in. Instead, it maintains a belief of its current state in form of a probability distribution  $p(s)$  over the states  $s \in S$ . The probability distribution is updated in two ways: When an action report  $a$  is received, indicating a move or turn, the new probabilities become:

$$p_{\text{posterior}}(s) = K \times \sum_{s' \in S | a \in A(s')} p(s|s', a) \times p_{\text{prior}}(s')$$

where  $K$  is a normalization factor to ensure that the probabilities all sum to one (this is necessary because not all actions are defined for all states). When a sensor report  $o$  is received from sensor  $i$ , indicating that a feature has been detected, the probabilities become:

$$p_{\text{posterior}}(s) = K \times p_i(o|s) \times p_{\text{prior}}(s)$$

The Markov model is constructed from three sources of information: the topology of the environment (which we presume can be easily obtained), general knowledge about office environments (such as that corridors are straight and perpendicular to each other), and approximate metric knowledge (obtained either from rough measurements or from general knowledge, such as the fact that, in our building, corridors are two meters wide and all doorways are between two and ten meters apart).

The map information is initially encoded as a graph of nodes and edges (Figure 1). A node represents a junction between corridors (and/or doorways or foyers). Nodes are connected by a pair of directed edges, which are augmented

# 논문 글쓰기 방법 (본문)

---

- 다른 사람이 재현이 가능하도록 설명함 (충분한 설명)
- 수식을 사용하여 설명하면 좋음
- 그림을 이용하여 쉽게 설명함
- 제안된 방법의 장점을 설명
- 핵심 방법을 소제목으로



3 Constructing the Markov Models

Before describing how we construct Markov models of an office environment, we introduce some terminology. A finite Markov model consists of a finite set of states  $S$ , a finite set of actions  $A$ , a set of actions  $A(s) \subseteq A$  for each state  $s \in S$ , that can be executed in that state, and transition probabilities  $p(s'|s, a)$  for all  $s, s' \in S$  and  $a \in A(s)$  (the probability that the new state is  $s'$  if action  $a$  is executed in state  $s$ ). We also define a set of sensors  $i \in I$ . The sensors are characterized by observation probabilities  $p_i(o|s)$  for all  $s \in S$  and  $o \in O(i)$  (the probability that sensor  $i$  reports feature  $o$  when the robot is in state  $s$ ). Note that Markov models assume that the transition and observation probabilities are determined only by the current state of the robot (the "Markov assumption").

In our case, the Markov model is partially observable because the robot may never know exactly which state it is in. Instead, it maintains a belief of its current state in form of a probability distribution  $p(s)$  over the states  $s \in S$ . The probability distribution is updated in two ways: When an action report  $a$  is received, indicating a move or turn, the new probabilities become:

$$p_{\text{posterior}}(s) = K \times \sum_{s' \in S | a \in A(s')} p(s|s', a) \times p_{\text{prior}}(s')$$

where  $K$  is a normalization factor to ensure that the probabilities all sum to one (this is necessary because not all actions are defined for all states). When a sensor report  $o$  is received from sensor  $i$ , indicating that a feature has been detected, the probabilities become:

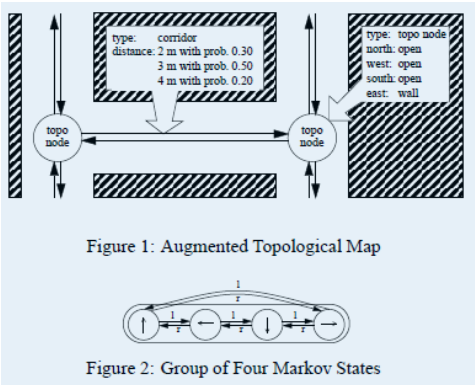
$$p_{\text{posterior}}(s) = K \times p_i(o|s) \times p_{\text{prior}}(s)$$

The Markov model is constructed from three sources of information: the topology of the environment (which we presume can be easily obtained), general knowledge about office environments (such as that corridors are straight and perpendicular to each other), and approximate metric knowledge (obtained either from rough measurements or from general knowledge, such as the fact that, in our building, corridors are two meters wide and all doorways are between two and ten meters apart).

The map information is initially encoded as a graph of nodes and edges (Figure 1). A node represents a junction between corridors (and/or doorways or foyers). Nodes are connected by a pair of directed edges, which are augmented

그림을 이용하여 제안하는 방법을 설명

Observable Markov 모델이 무엇인지 설명 (배경 이론)



with approximate length information in the form of a probability distribution over possible lengths. Rooms and foyers (not shown) are also represented in the map.

The rest of this section describes how the augmented topological map is compiled into a Markov model.

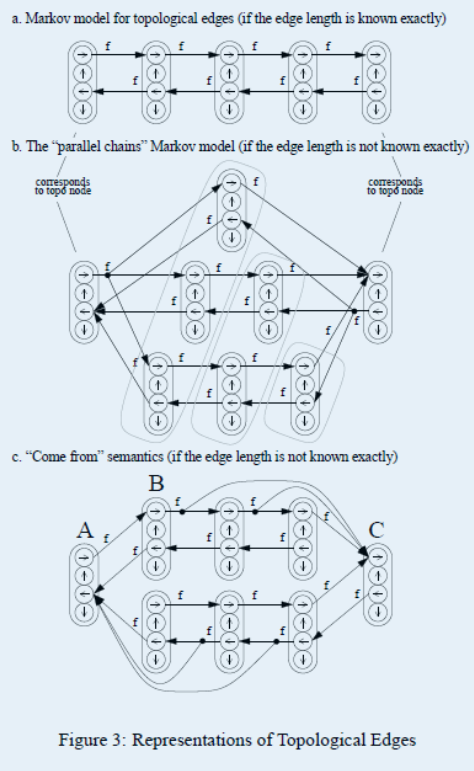
**Modeling Locations**

Each Markov state encodes both the orientation and location of the robot. To insulate the model from low-level control aspects (such as turning to avoid obstacles), we encode the commanded heading of the robot rather than its instantaneous orientation. Since our corridors are straight and perpendicular to each other, it is sufficient to discretize orientation into the four compass directions: North, South, East, West. The spatial locations of the robot are also discretized. While more fine-grained discretizations yield more precise models, they also result in more memory requirements and more time-consuming computations. We use a resolution of one meter, which we have found to be sufficient.

Since our Markov states encode both orientation and location, four states are needed to fully represent each spatial location. Three actions are modeled: turning right 90 degrees ( $r$ ), turning left 90 degrees ( $l$ ), and going forward one meter ( $f$ ). Right and left turn actions are defined for every state (Figure 2). Since they correspond to changes in commanded heading and not to changes in position, we have found it sufficient to model them deterministically. Some states also have "forward" actions defined for transitioning from location to location (note that forward actions are not defined for states that face walls). Dead-reckoning uncertainty is modeled by a self-transition, that is, the forward action transitions with some probability from a state into itself.

**Modeling Corridors**

Our representation of topological edges is a key to our approach. If the edge lengths are known exactly, it is simple to model the ability to traverse a corridor with a Markov chain that has forward actions between those states whose orientations are parallel to the corridor axis (Figure 3a). The model



becomes more complex when only approximate edge lengths are known. While one approach is to represent a corridor edge by a single Markov state [Nourbakhsh *et al.*, 1995], this loses the ability to utilize dead-reckoned information in doing position estimation.

Another approach is to model an edge as a set of parallel Markov chains, each corresponding to one of the possible lengths of the edge (Figure 3b). The transition probabilities into the first state of each chain are the same as the probability distribution over edge lengths associated with the topological map (see Figure 1). Each forward transition after that is deterministic (modulo dead-reckoning uncertainty — note that the identity transitions are not shown in these figures). While this representation best captures the actual structure of the environment, it is relatively inefficient: the number of states is quadratic in the maximum length of the edges.

As a compromise between fidelity and efficiency, our current implementation models edges by collapsing the parallel chains in a way that we call the "come from" semantics (Figure 3c). In this representation, the spatial location of a Markov state is known relative to the topological node from which the robot comes, but its location relative to the end of

# 논문 글쓰기 방법 (실험 결과)

- 실험 구성 조건 및 방법에 대해 자세히 설명할 것 (재현 가능할 수 있도록)
  - 시뮬레이션 뿐만 아니라 실제 실험이 있으면 좀 더 설득력 있음
  - 각 실험을 설계한 이유에 대한 설명 필요 (예: 정확도, 연산 시간 등)
  - 표 또는 그래프 등을 적극적으로 활용하여 결과를 기록할 것
  - 다른 방법론과의 비교 필요
    - 내 방법은 기존의 A 방법보다 OO % 더 향상된 결과를 보여준다
- 실험 결과를 단순히 기록하는 것이 아니라 의미를 해석할 수 있어야 함

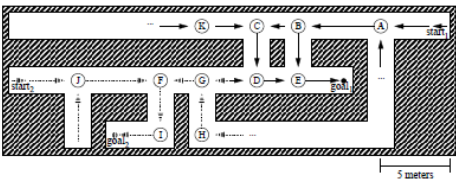


Figure 7: An Office Corridor Environment

path	best-action			best-above-threshold		
	freq.	time s	speed cm/s	freq.	time s	speed cm/s
ABE	12	68.2	25.7	5	63.6	27.5
ABCDE	3	79.7	29.5	8	81.0	29.0
ABCKDE	—	—	—	2	104.1	N/A

Table 1: Experiment 1

## 5.1 Experiments with the Simulator

Two navigation experiments were performed with the robot simulator in the corridor environment shown in Figure 7. The topological map has 17 nodes and 36 directed edges. We modeled the uncertainty of the length of a topological edge as a uniform distribution over the interval ranging from 80 to 150 percent of the real length of the edge. The resulting Markov model has 1184 Markov states. The initial positional uncertainty for both experiments is minimal: the initial probability for the robot's actual location is about 90 percent. The remaining probability mass is distributed in the vicinity of the actual location.

In the first experiment, the task was to navigate from  $start_1$  to  $goal_1$ . The preferred headings assigned by our planner are shown with solid arrows. Note that the preferred heading between B and C is towards C because, even though the goal distance is a bit longer, this way the robot does not have to turn around if it overshoots B. We ran a total of 15 trials for both the best-action and the best-above-threshold strategies, all of which were completed successfully (Table 1).

The robot has to travel a rather long distance from  $start_1$  before its first turn. Since this distance is uncertain and corridor openings are occasionally missed, the robot occasionally overshoots B, and then becomes uncertain whether it is really at C or B. However, since the same directive is assigned to both nodes, this ambiguity does not need to be resolved: the robot turns left in both cases and then goes straight. The same thing happens when it gets to D, since it thinks it may be at either D or E. The robot eventually corrects its beliefs when, after turning left and traveling forward, and it detects an opening to its left. At this point, the robot becomes fairly certain that it is at E. A purely landmark-based navigation technique can easily get confused in this situation, since it has no expectations about seeing this opening, and can only attribute it to sensor error (which, in this case, is incorrect).

In the second experiment, the robot had to navigate from

path	best-action			best-above-threshold		
	freq.	time s	speed cm/s	freq.	time s	speed cm/s
JFI	11	60.6	28.9	8	65.4	26.8
JFGFI	2	91.5	25.7	—	—	—
JFGHGFI	1	116.0	23.7	5	120.2	22.9
JFGGFI	1	133.0	22.2	—	—	—
JFGDGFI	—	—	—	2	176.5	N/A

Table 2: Experiment 2

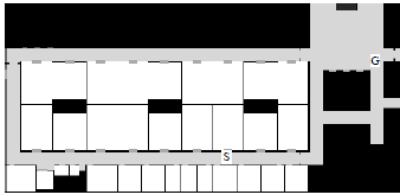


Figure 8: Wean Hall at CMU

$start_2$  to  $goal_2$ . The preferred headings for this task are shown with dashed arrows. Again, we ran 15 trials for both action selection strategies (Table 2).

For reasons that are similar to those in the first experiment, the robot can confuse G with F. If it is at G but thinks it is probably at F, it turns right and goes forward. However, when it detects the end of the corridor but does not detect a right corridor opening, it realizes that it must be at H rather than I. Since the probability mass has now shifted, it turns around and goes over G, F, and I to the goal. This shows that our navigation technique can gracefully recover from misjudgements based on wrong sensor reports – even if it takes some time to correct its beliefs. It is important to realize that this behavior is not triggered by any explicit exception mechanism, but results automatically from the way the position estimation and action selection interact.

## 5.2 Experiments with Xavier

Xavier, our indoor mobile robot, is built on an RWI B24 base and includes bump sensors, sonars, a laser range sensor, and a color camera on a pan-tilt head. Control, perception and planning are all carried out on two on-board, multi-processing 486-based machines.

As mentioned, the probabilistic navigation system uses a modified version of the planner and essentially the same robot controller and sensor interpretation components as our landmark-based navigation system. Thus, differences in performance can be directly attributed to the different navigation approaches. In addition, to facilitate comparisons we ran Xavier along the same routes as reported in [Simmons, 1994]. In particular, the robot traversed from point S to G and back again (Figure 8) in some trials (45 meters each way), and in some circumnavigated around the building (150 meters).

The topological map used to represent the corridors in Figure 8 has 95 nodes and 180 directed edges. As with the simulator trials, the edge lengths ranged uniformly from 80 to

실험 환경에 대한 자세한 설명 기술

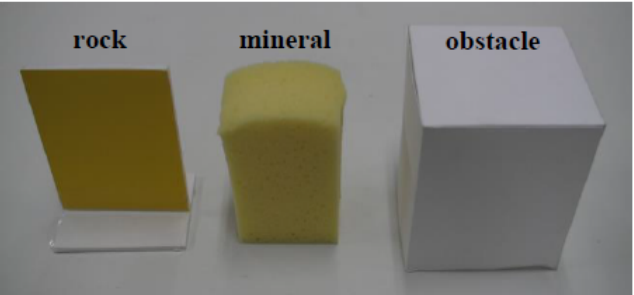
표 또는 그래프를 이용하여 적절하게 결과를 기술함

실제 실험 수행



## 5. 실험 결과

[그림 5]와 같이 장애물과 광물이 섞여 있는 환경을 구성했다. 장애물은 노랑색 계열의 암석(rock)과 흰색 계열의 상자로 구성되어 있으며, 광물은 노랑색 스펀지를 사용하였다[그림 광물]. 같은 색 계열의 암석과 광물을 사용한 것은 인간의 판단이 필요한 혼동 상황을 연출하기 위함이었다. 광물 채취를 위해 총 3 대의 e-puck 소형 로봇에 무선 카메라를 장착하여 광물 탐색 및 채취를 시작한다. 로봇에 장착되어 있는 적외선 센서를 이용하여 주변의 장애물을 피하면서 이동하고, 무선 카메라를 통해 광물에 해당하는 노랑색이 검출될 경우 판단 기준에 의하여 사용자에게 광물이 맞는지를 묻는다[그림 6]. 이 때, 기준 픽셀 수는 100 개로 정했다. 즉, 100 개의 픽셀보다 많이 노랑색이 검출될 경우 로봇은 인간의 판



[그림 광물] 로봇이 감지하는 물체들. 광물(mineral)만 캐내야 하므로, 암석과 장애물은 충돌 회피해야 한다. 이 중 암석은 광물과 유사한 색깔을 가지므로 인간의 판단을 통해 구별한다.

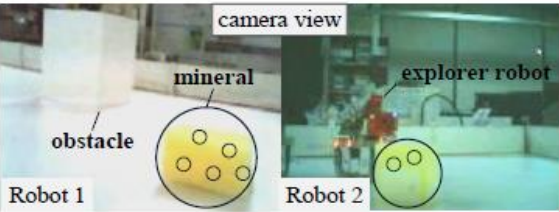
실험 환경에 대한 구체적인 설명 부족

- rock, mineral, obstacle의 크기는?
- 실험 환경의 전체 크기는?
- 장애물의 배치 위치는?
- 사용한 로봇의 specification 은?



[그림 5] 다개체 로봇의 광물 채취

그림에 표현된 O의 의미는? 매칭 결과?



[그림 6] 광물 채취 로봇이 획득한 영상

정량적인 결과 없음

- 성공률
- 매칭 정확도

단 없이 스스로 광물을 채취하였으며, 그 이하의 개수가 검출되었을 경우에는 시스템에 적용된 옵션에 따라 인간에게 판단을 요구하였다. 사용자의 판단을 기다리는 동안에는 다른 로봇들과의 간섭을 피하기 위하여 전체 시스템을 일단 정지 시켰다. 사용자는 로봇이 전송한 메시지 팝업 창에서 광물 여부의 판단을 내려주면 광물이라고 판단된 경우 광물 채취를 시작하며 아닌 경우에는 광물 탐색 모드로 복귀하였다. 그 결과, 각각의 로봇들은 혼동 없이 광물을 구분해낼 수 있었고, 광물 채취를 성공적으로 수행할 수 있었다.

# 논문 글쓰기 방법 (결론)

---

- 논문에서 제시한 방법을 다시 한번 서술함
- 제안한 방법의 장점과 새로운 점에 대해 기술함
- 실험을 통해 성공적으로 검증되었음을 기술함
- 제안한 방법의 한계 및 후속 연구 제시
  - : 지나치게 제안한 방법을 깎아내리지 말 것
  - 후속 연구에 대한 방향성을 제시하는 관점에서 설명

We intend to pursue planning and action selection algorithms that approximate optimal POMDP policies, and to compare their performance to the greedy heuristics described here. Finally, we intend to add new sources of sensor information, primarily vision-based feature detectors.

The implemented probabilistic navigation system has demonstrated its reliability, both in simulation and on Xavier, even in the face of significant uncertainty. We believe that such probabilistic navigation techniques hold great promise for getting robots reliable enough to operate unattended for long periods of time in complex, uncertain environments.

### Acknowledgements

Thanks to Lonnie Chrisman, Richard Goodwin and Joseph O'Sullivan for helping to implement parts of the navigation system and for many valuable discussions. Swantje Willms helped perform some of the experiments with the simulator.

### References

[Arkin, 1987] R.C. Arkin. Motor schema based navigation for a mobile robot: An approach to programming by behavior. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 264-271, 1987.

[Cassandra *et al.*, 1994] A.R. Cassandra, L.P. Kaelbling, and M.L. Littman. Acting optimally in partially observable stochastic domains. In *Proceedings of the AAAI*, pages 1023-1028, 1994.

[Chrisman, 1992] L. Chrisman. Reinforcement learning with perceptual aliasing: The perceptual distancions approach. In *Proceedings of the AAAI*, pages 183-188, 1992.

[Dean *et al.*, 1993] T. Dean, L.P. Kaelbling, J. Kirman, and A. Nicholson. Planning with deadlines in stochastic domains. In *Proceedings of the AAAI*, pages 574-579, 1993.

[Elfes, 1989] A. Elfes. Using occupancy grids for mobile robot perception and navigation. *IEEE Computer*, pages 46-57, 6 1989.

[Goodwin, 1994] R. Goodwin. Reasoning about what to plan. In *Proceedings of the AAAI*, page 1450, 1994.

[Hu and Brady, 1994] H. Hu and J.M. Brady. A Bayesian approach to real-time obstacle avoidance for a mobile robot. *Autonomous Robots*, 1(1):69-92, 1994.

[Kirman *et al.*, 1991] J. Kirman, K. Basye, and T. Dean. Sensor abstractions for control of navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2812-2817, 1991.

[Koenig and Simmons, 1994] S. Koenig and R. Simmons. How to make reactive planners risk-sensitive. In *Proceedings of the International Conference on Artificial Intelligence Planning Systems*, pages 293-298, 1994.

[Konolige, 1994] K. Konolige. Designing the 1993 robot competition. *AI Magazine*, 15(1):57-62, 1994.

[Kortenkamp and Weymouth, 1994] D. Kortenkamp and T. Weymouth. Topological mapping for mobile robots using a combination of sonar and vision sensing. In *Proceedings of the AAAI*, pages 979-984, 1994.

논문에서 제시한 바를 다시 설명함

### 6 Future Work and Conclusions

This paper has presented our first efforts at using partially observable Markov models (POMDPs) for autonomous office navigation. The approach enables a robot to utilize all its sensor information, both positional and feature-based, in order to robustly track its location. A simple path planner and action selection heuristics are used to direct the robot's goal heading. Advantages of this approach include the ability to account for uncertainty in the robot's initial position, actuator uncertainty, sensor noise, and uncertainty in the interpretation of the sensor data. Also, by integrating topological and metric information, the approach easily deals with uncertainty arising from incomplete descriptions of the environment.

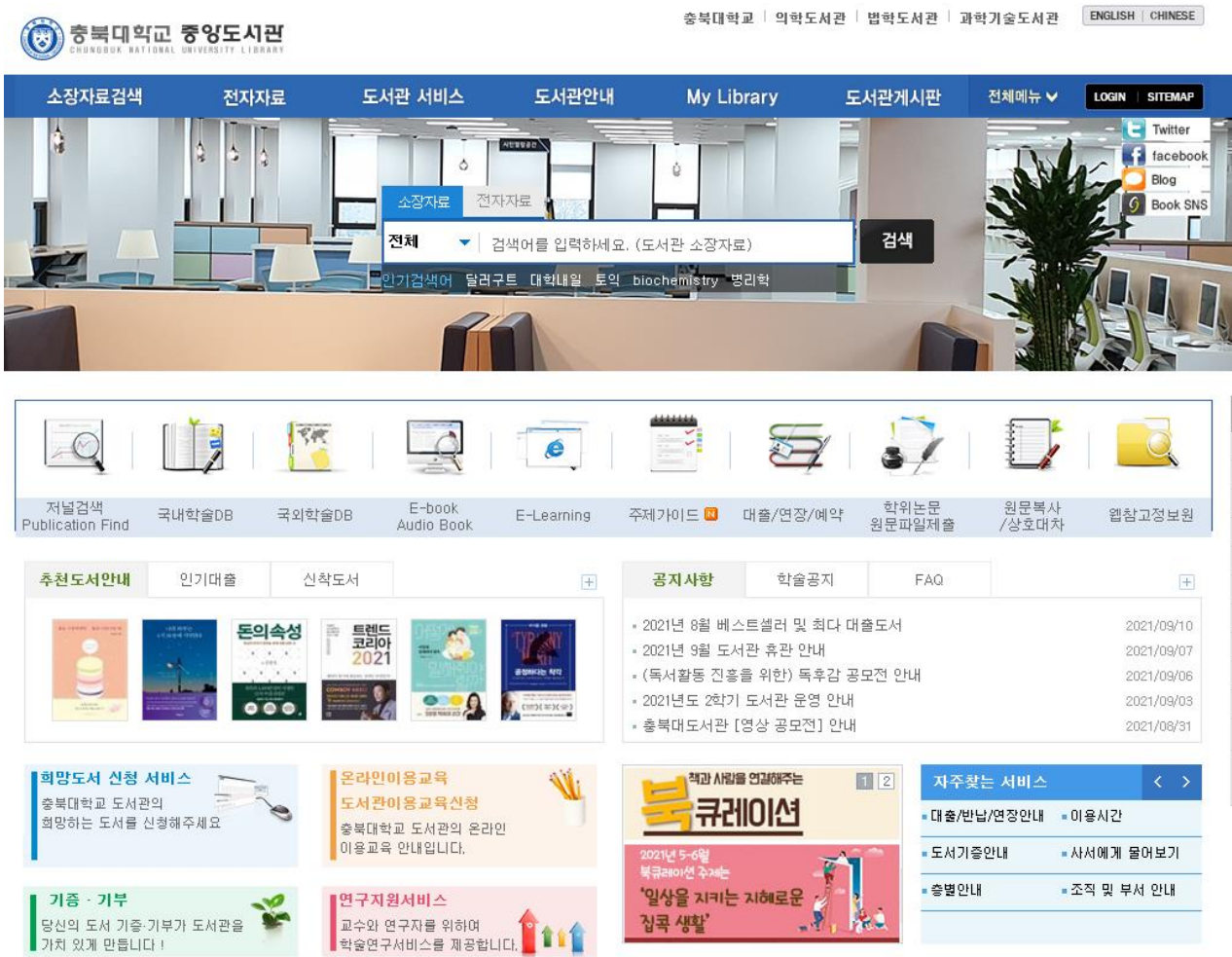
We are extending this work in several directions. We have implemented methods, based on EM learning techniques, that passively refine metric map information as well as the sensor and action models, and will be testing it with Xavier. In addition, we are developing improved learning techniques that are more resistant to violations of the Markov assumption.

본 논문의 장점 및 novelty 설명

후속 연구에 대한 방향성 제시

# 논문 글쓰기 방법 (참고문헌)

- 참고 문헌 정리 스타일은 각 논문마다 형식이 매우 다양함
- 산업인공지능학과의 석사학위 참고문헌 스타일은  
충북대학교의 학위논문 양식에 맞춰 작성하면 됨
- 충북대학교 도서관에서 학위 논문을 3~5개 정도 다운 받아  
관련 양식을 참고할 것 (원문 다운로드 가능)  
(이 때, 가능하면 담당 지도교수님 졸업생의 논문을 다운 받아  
볼 것을 권장함)



# 참고 자료

(동영상)

- Cyrill Stachniss, "How to a research paper": <https://www.youtube.com/watch?v=sSnxXQgUCyY>
- (주)드림셀파, 논문쓰는법: 논문 목차와 조작적 정의  
: <https://www.youtube.com/watch?v=8rURsveES5w>

(논문)

- Simmons, Reid, and Sven Koenig. "Probabilistic robot navigation in partially observable environments." IJCAI. Vol. 95. 1995.



# 수업 과제 및 Q&A

---

## 1. 수업 과제

- e-class에 올라가 있는 "논문 분석 과제.pptx" 다운 받아 논문 읽고 작성함
- 작성된 결과물을 e-class에 업로드

## 2. Q&A

- 질문 사항은 어규호 초빙교수 ([gyuho.eoh@cbnu.ac.kr](mailto:gyuho.eoh@cbnu.ac.kr))에게 문의
- 프로젝트 시간을 활용하여 문의 가능 (사전 문의)