

공학학사학위논문

물체인식을 위한 움직이는 탐사로봇 개발
(Exploration Robot for Object Recognition)

2014 년 8 월

서울대학교 공과대학

전기정보공학부

김 주 용

물체인식을 위한 움직이는 탐사로봇 개발
(Exploration Robot for Object Recognition)

지도교수 오 성 회

이 논문을 공학 학사학위 논문으로 제출함

서울대학교 공과대학

전기정보공학부

김 주 용

김주용 의 학사 학위 논문을 인준함

2014 년 8 월

지 도 교 수

(인)

초록

본 논문은 특징 기반 접근법을 사용한 물체인식의 성능을 개선할 수 있는 물체인식 알고리즘을 제안하고, 이를 사용한 저비용의 탐사로봇을 구현한다. SIFT 특징 값만을 사용하여 특징점을 비교할 경우 낮은 해상도의 카메라나 멀리 있는 물체의 SIFT 특징으로 매칭할 경우 인식의 성능이 좋지 않은 문제점이 있다. 본 논문에서는 SIFT 특징 간의 거리함수로 Hellinger distance를 사용하고 특징점 근처의 색상, 공간정보를 사용하여 SIFT 특징 간의 매칭 성능을 높였다. 제안한 물체인식 알고리즘은 기존의 방법과 비교하여 성능이 개선되었음을 확인하였으며, 제작한 탐사로봇이 실험환경에서 물체를 인식하는 실험으로 그 성능을 검증하였다.

주요어: 물체인식, 특징 기반 접근법, SIFT, Hellinger Distance, HSV Histogram, Undirected Graph

목차

초록	i
제 1 장 서론	1
1.1 연구의 배경	1
1.2 논문의 구성	1
제 2 장 본론	2
2.1 SIFT 특징의 거리함수	2
2.2 SIFT 특징을 사용한 물체인식 알고리즘	3
2.3 인식률을 높이기 위한 공간, 색상정보 사용	6
2.4 인식한 물체의 위치 계산	8
2.5 탐사роб트의 물체인식	9
제 3 장 결과	11
3.1 실험환경	11
3.2 물체인식 알고리즘 결과	11
3.3 실험결과	13
제 4 장 토의	15
제 5 장 결론	17
참고문헌	18

표 목차

표 2.1	Cost matrix 매칭 알고리즘 성능 표	5
표 2.2	Cost matrix 매칭 알고리즘 연산 시간 표	5
표 3.1	색상, 공간정보를 사용한 SIFT 특징의 매칭 결과	13

그림 목차

그림 2.1	SIFT 특징의 거리함수 성능 비교를 위한 물체-실험환경 이미지 쌍	3
그림 2.2	Hellinger distance와 Euclidean distance를 SIFT 특징의 거리함수로 사용하였을 때의 Precision-recall curve	3
그림 2.3	SIFT 특징 매칭과 Cost matrix 그림	4
그림 2.4	Cost matrix의 매칭 알고리즘 비교를 위한 물체-실험환경 이미지 쌍	4
그림 2.5	Cost matrix의 매칭 알고리즘 성능 비교	5
그림 2.6	SIFT 특징 주변의 크기와 방향을 정규화하여 얻은 이미지 조각	6
그림 2.7	SIFT 특징 주변의 이미지를 크기와 방향을 정규화하여 얻은 이미지 조각	7
그림 2.8	인식률을 높이기 위한 공간정보의 사용	8
그림 2.9	인식한 물체의 위치 계산	8
그림 2.10	탐사로봇의 상태도	10
그림 3.1	탐사로봇과 그 구성	11
그림 3.2	물체인식 알고리즘의 결과와 기존의 방법과의 비교	12
그림 3.3	색상, 공간정보를 사용한 SIFT 특징의 매칭 결과	13
그림 3.4	물체인식 탐사로봇의 실험결과	14

제 1 장 서론

1.1 연구의 배경

컴퓨터 비전 분야에서 물체인식(Object recognition)이란 이미지나 동영상에서 물체를 찾고 그 물체가 무엇인지 찾는 과정을 말한다. 인간은 적은 노력으로도 다른 위치, 각도로 놓이거나 심지어 부분적으로 가려진 경우에도 물체를 인식하지만 컴퓨터 비전 분야에서 물체인식은 매우 어려운 문제로 여겨진다.

이를 위해서 여러 접근법이 제안되었는데 그 중에서 특징 기반(Feature-based) 접근법은 이미지에서 국소적인 특징점(Keypoint)을 검출하고 특징(Feature)으로 추출하여 이를 미리 학습한 특징과 대조하는 방식으로 물체를 인식한다. 인식하고자 하는 물체가 가진 특징과 유사한 특징을 이미지의 어떤 부분에서 많이 가지고 있다면 그 부분에 찾고자 하는 물체가 있는 것으로 인식하는 것이다. 이러한 특징을 추출하는 과정은 특징점 검출(Keypoint detection)과 특징 기술(Feature Description)으로 나뉘게 된다. 물체인식에서 좋은 성능을 보이는 특징점은 물체의 위치, 크기의 조명의 변화도 쉽게 검출되는 것이며 좋은 특징은 그 값이 이런 변화에 크게 바뀌지 않는다. David Lowe에 의해 고안된 알고리즘인 SIFT(Scale-Invariant Feature Transform)는 특징점 검출과 특징 기술을 둘 다 포함하는 알고리즘으로 확대, 회전 등의 왜곡에 대해 강건한(robust) 것으로 알려져 있다.

본 논문에서는 SIFT 특징의 거리함수로 Hellinger distance를 사용하여 이미지 간의 SIFT 특징을 비교하고 대응된 SIFT 특징점의 위치, 색상 정보를 사용하는 물체인식 알고리즘을 제안한다. 이 방식으로 특징 기반 접근법에서 동일한 SIFT 특징점을 대응시키는 성능을 향상시켜 물체인식의 성능을 개선하고, 낮은 해상도를 가진 카메라로 비교적 먼 거리의 물체를 인식할 수 있게 한다. 본 논문에서는 이 알고리즘을 로봇에 응용하기 위해 물체인식 알고리즘을 이용해서 원하는 물체를 찾는 저비용 탐사로봇을 구현하고 실험환경에서 탐사로봇의 성능을 검증한다.

1.2 논문의 구성

본 논문은 크게 6장으로 구성되었다. 2장에서는 SIFT 특징의 거리함수로써 사용한 Hellinger distance와 SIFT 특징점 주변의 색상정보, 이미지 상의 SIFT 특징점 공간정보를 사용한 물체인식 알고리즘과 이를 이용한 탐사로봇의 물체인식 방법에 대해서 설명한다. 3장에서는 물체인식 알고리즘과 탐사로봇의 성능을 실험으로 검증하고 4장에서는 결과를 분석하고 논문의 구현에서의 세부사항을 논의한다. 마지막 5장에서는 결론을 제시한다.

제 2 장 본론

2.1 SIFT 특징의 거리함수

SIFT 특징은 128차원의 벡터로 표현된다. 이는 특징점 주변을 크기, 방향을 고려하여 잘라낸 영역을 4×4 구간으로 나누고 각각의 구간내의 픽셀들의 Gradient에 대한 8구간 히스토그램을 모아 정규화(normalization)한 것이다. 이 SIFT 특징 간의 유사함을 비교하기 위해서는 SIFT 특징의 거리함수가 필요한데, 기존에 많이 사용하는 거리함수로 다음의 유클리드 거리(Euclidean distance)를 많이 사용한다.

$$d_{euclidean}(P, Q)^2 = \sum_{d=1}^D (P(i) - Q(i))^2 \quad (2.1)$$

여기서 P 와 Q 는 비교하고자 하는 두 SIFT 특징을, $P(i)$ 는 SIFT 특징 P 의 i 번째 성분을 나타낸다. 하지만 SIFT 특징은 앞에서 언급한 바와 같이 특징점 주변의 픽셀 값의 Gradient에 대한 히스토그램이므로 유사함을 나타내는 거리함수는 정규화된 히스토그램(또는 확률분포) 간의 유사성을 나타내는 다른 척도가 더 좋을 것임을 기대할 수 있다.

SIFT 특징의 거리함수로써 우리는 확률분포 간의 유사성을 나타내는 거리함수인 Hellinger distance를 제안한다. Hellinger distance는 두 이산확률분포 P 와 Q 에 대해 다음과 같이 정의된다.

$$H(P, Q) = \sqrt{1 - BC(P, Q)} \quad (2.2)$$

여기서

$$BC(P, Q) = \sum_{d=1}^D \sqrt{P(i)Q(i)} \quad (2.3)$$

는 두 히스토그램의 Bhattachayya 계수다.

SIFT 특징의 거리함수로써 Euclidean distance와 Hellinger distance의 성능을 비교하기 위해서 물체의 이미지로부터 추출한 SIFT 특징들과 동일한 물체를 포함한 실험환경의 이미지로부터 추출한 SIFT 특징들의 유사함을 각각의 거리함수로 계산하여 동일한 부분을 나타내는 SIFT 특징을 얼마나 잘 찾는지를 확인해보았다. 각각의 거리함수에 대하여 같은 SIFT 특징으로 판정하기 위한 기준치(threshold)값을 다르게 하여 얻은 SIFT 특징쌍의 정확률(precision: 같은 것으로 판정한 SIFT 특징쌍 중에서 실제로 동일한 부분을 나타낸 SIFT 특징쌍의 비율)과 재현율(recall: 참값으로 지정한 전체 SIFT 특징쌍 중에서 같은 것으로 판정한 SIFT 특징쌍의 비율)을 계산하고 이를 그림 2.2의 정확률-재현율 곡선(precision-recall curve)로 나타내었다.

실험에는 5개의 물체-실험환경 이미지 쌍으로 부터 얻은 전체 약 4000개의 SIFT 특징과 그 중에서 수동으로 표시한 451개의 SIFT 특징쌍 참값(ground truth)을 데이터로 사용하였다. 실험에 사용된 이미지



(a) 테스트 데이터 1

(b) 테스트 데이터 2

그림 2.1: SIFT 특징의 거리함수 성능 비교를 위한 물체-실험환경 이미지 쌍

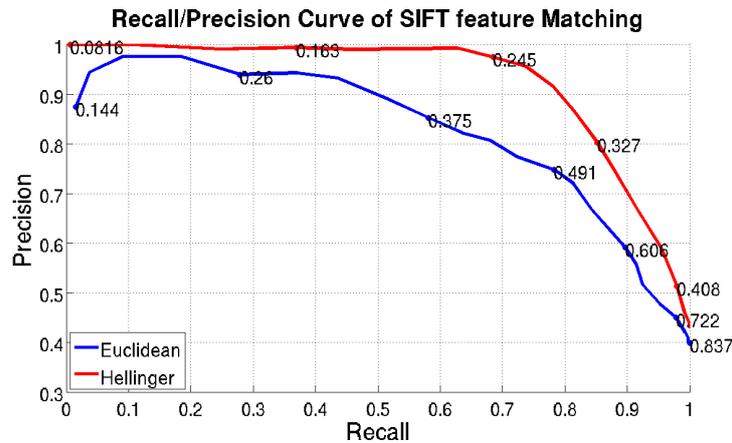


그림 2.2: Hellinger distance와 Euclidean distance를 SIFT 특징의 거리함수로 사용하였을 때의 Precision-recall curve

쌍들 중 일부를 그림 2.1에 나타내었다. 실험의 결과로 얻은 그림 2.2의 정확률-재현율 곡선(precision-recall curve)에서 볼 수 있듯이 Hellinger distance를 거리함수로 사용하였을 때 더 높은 정확률과 재현율을 보이는 것을 알 수 있다. 즉, SIFT 특징간의 거리함수로 Euclidean distance보다 Hellinger distance를 사용하였을 때 물체인식이 더 좋은 성능을 발휘하게 될 것이다.

2.2 SIFT 특징을 사용한 물체인식 알고리즘

특징기반 접근의 물체인식에서는 물체의 이미지와 카메라로 촬영한 이미지에서 각각 추출한 SIFT 특징들 간의 매칭이 필요하다. 본 논문에서 하고자 하는 것이 촬영한 이미지 상에서 물체가 있는지 찾는 것이기 때문에 물체 이미지로부터 추출한 SIFT 특징 각각에 대해 촬영한 이미지의 SIFT 특징들 중에서 가장 유사한 것을 찾아야 한다. 여기서 잘못된 매칭을 막기 위해 Hellinger distance가 특정 기준치(threshold)값보다 작은 대응만을 찾아야 하며 그렇지 않은 경우에는 촬영한 이미지 상에서 유사한 SIFT 특징이 없는 것으로 본다. 또한 물체 이미지에서 추출한 SIFT 특징 각각이 물체의 다른 부분을 나타내므로 촬영한 이미지에서 대응되는

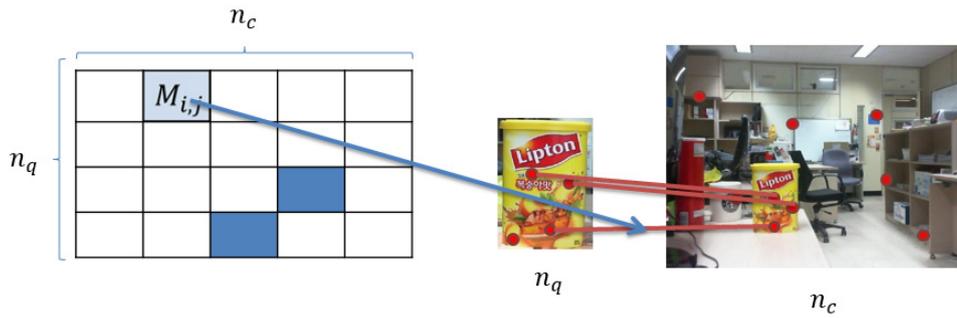


그림 2.3: SIFT 특징 매칭과 Cost matrix 그림

SIFT 특징은 서로 다른 것이어야 한다.

예를 들어, 찾고자 하는 물체의 이미지에서 n_q 개의 SIFT 특징이 검출되고 카메라로 촬영한 이미지에서 n_c 개의 SIFT 특징이 검출되었다고 하면 가능한 SIFT 특징의 대응은 총 $n_q n_c$ 개가 있다. 물체와 촬영한 이미지의 SIFT 특징 간의 Hellinger distance를 $n_q \times n_c$ 의 Cost matrix M 으로 표현하면 문제는 이 행렬의 각 행에서 최대 하나의 성분을 열의 중복 없이 선택하는 것이 되며 이때 선택된 각 성분이 기준치값보다 클 경우 매칭에서 탈락시킨다. 그림 2.3에서 SIFT 특징의 매칭과 cost matrix의 성분 선택을 그림으로 표현하였다. 그림에서 나타내었듯이 모든 행에서 성분을 선택하지 않을 수도 있다.

이렇게 행렬에서 각 행에서 한 성분씩을 선택하는 방법으로는 두 가지가 있다. 첫째로 단순히 각 행에서 가장 값이 작은 성분을 선택한 뒤에 같은 열이 선택된 성분들이 있을 경우 값이 더 작은 쪽을 선택하는 방법이 있다(이 방법을 one-to-one 이라 하자). 둘째로는 조합 최적화 알고리즘인 Hungarian method를 사용하는 방법인데 이는 각 행에서 열의 중복 없이 선택한 성분의 합이 최소가 되게 하는 알고리즘이다. 첫 번째 방법은 물체 이미지와 촬영한 이미지에서 각각 n_q, n_c 개의 SIFT 특징이 나온 경우에 각 행에 대해 최소가 되는 원소를 선택하는 연산에 의해 $O(n_q n_c)$ 의 시간 복잡도를 가지고, 두 번째 방법인 Hungarian method는 $n \times n$ 행렬에 대해 $O(n^3)$ 의 시간 복잡도를 가진다.

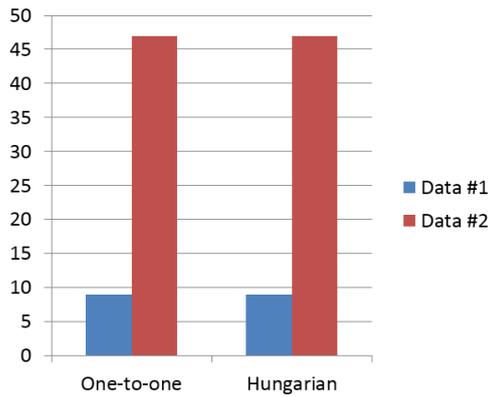
두 매칭 방법을 비교하기 위해서 물체-실험환경 테스트 이미지 쌍의 SIFT 특징으로 만든 Hellinger dis-



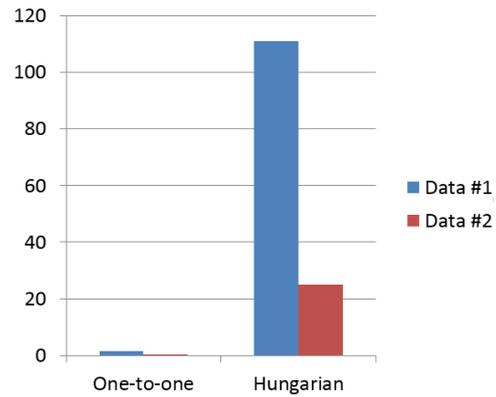
(a) 테스트 데이터 1

(b) 테스트 데이터 2

그림 2.4: Cost matrix의 매칭 알고리즘 비교를 위한 물체-실험환경 이미지 쌍



(a) SIFT 특징의 매칭 수 그래프



(b) 연산 시간 그래프

그림 2.5: Cost matrix의 매칭 알고리즘 성능 비교

표 2.1: Cost matrix 매칭 알고리즘 성능 표

	One-to-one	Hungarian
Data #1	9	47
Data #2	9	47

표 2.2: Cost matrix 매칭 알고리즘 연산 시간 표

	One-to-one	Hungarian
Data #1	110.85	1.49
Data #2	25.139	0.253

(단위 : ms)

tance의 cost matrix에 대해 두 매칭 방법을 연산 시간과 매칭의 성능 측면에서 측정, 비교하였다. 실험은 두 개의 테스트 이미지 쌍에 대해서 이루어졌으며 그림 2.4에 나타내었다. 각각의 이미지 쌍에 대하여 만들어진 cost matrix에 대해 두 매칭 알고리즘을 수행하고 연산 시간을 측정한다. 매칭 개수는 알고리즘을 수행한 뒤 동일한 기준치값으로 매칭을 탈락시킨 뒤 올바르게 매칭한 SIFT 특징 쌍의 수를 확인한다.

표 2.1, 2.2와 그림 2.5에 실험 결과를 나타내었다. 올바르게 매칭한 SIFT 특징의 수는 두 매칭 알고리즘에 대해 차이가 없었다. 이는 Hellinger distance에 기반하여 SIFT 특징을 매칭하는 경우 물체의 이미지에서 추출한 각각의 SIFT 특징에 대해 촬영한 이미지의 SIFT 특징 중 가장 유사한 특징만을 독립적으로 선택하여도 다른 물체의 SIFT 특징과 함께 고려하는 것과 성능이 비슷함을 보여준다. 반면 연산 시간에서는 각각 물체의 SIFT 특징에 대해 독립적으로 선택하는 경우가(one-to-one) 물체의 SIFT 특징간에 서로 고려하며 매칭하는 경우인 Hungarian method에 비해 크게 적은 연산 시간을 소요함을 알 수 있다. 따라서 SIFT 특징을 매칭하는데 사용할 알고리즘으로 단순히 각 행에서 가장 값이 작은 성분을 선택한 뒤 중복을 제거하는 방식(one-to-one)을 사용하도록 한다.

2.3 인식률을 높이기 위한 공간, 색상정보 사용

낮은 해상도를 가진 카메라에서 또는 먼 거리에 있는 물체를 인식할 때 그 성능을 높이기 위해서 앞 절에서 논의한 SIFT 특징 간의 비교 외에 다른 정보를 사용하여 특징점을 매칭시킨다. 이를 위해서 SIFT 특징의 비교에서 기준치값을 높게 하여 더 많은 SIFT 특징이 매칭되게 하는 동시에(SIFT 특징 간에 같은 특징으로 인식할 Hellinger distance의 최대 거리가 높아지므로 기준이 더 약해지고, 동일한 SIFT 특징쌍과 그렇지 않은 SIFT 특징쌍을 모두 더 많이 검출한다.) 매칭 후에 특징점 간의 다른 정보를 사용하여 올바르게 매칭되는 특징점의 수를 높이고자 한다.

첫째로, 특징점 근처의 색상정보를 비교한다. SIFT 특징은 검출된 특징점 주변의 밝기(Intensity)의 국소적인 변화 정보만을 사용하여 128차원의 벡터를 만들어낸다. 즉, 특징점 주변의 색이 다르지만 국소적인 밝기 변화의 형태가 비슷한 두 특징점 간에는 SIFT 특징의 Hellinger distance가 작다. 따라서 낮은 성능의 카메라, 또는 먼 거리의 물체의 SIFT 특징에 대해 인식률을 높이기 위해 특징점 근처의 색상정보를 사용할 수 있을 것이다.

SIFT 특징점의 검출은 그 결과로 특징점 근처의 영역에 대한 방향과 크기를 가지는데, 이를 이용해 특징점 주변의 이미지를 그 크기 비율과 방향에 맞게 잘라내어 30px×30px의 크기로 정규화한 이미지를 만든다. 그림 2.6은 앞 절에서 다른 SIFT 특징의 매칭에서 대응된 두 쌍의 SIFT 특징에 대해 자른 이미지의 조각의 예를 보여준다. 첫 번째 이미지쌍에서 볼 수 있듯이 올바른 특징점이 매칭된 경우엔 같은 모양을 보여주며 두번째 이미지쌍에서는 국소적으로 밝기 변화의 모양이 비슷하나 색상이 다를 수 있다.

이렇게 잘라낸 SIFT 특징 주변의 이미지 간의 색상정보를 나타내는데에는 HSV 색상모델의 히스토그램을 사용한다. HSV 색상 모델은 색을 색상(Hue), 채도(Saturation), 밝기(Value)의 값으로 표현한 것으로 RGB 모델보다 밝기 변화에 더 강인한 것으로 알려져 있다. HSV 히스토그램은 다음의 방법([2], [3])으로 생성한다. 히스토그램은 색이 있는 픽셀에 대한 구간과 색이 없는 픽셀에 대한 구간으로 나뉜다. 채도와 밝기가 특정

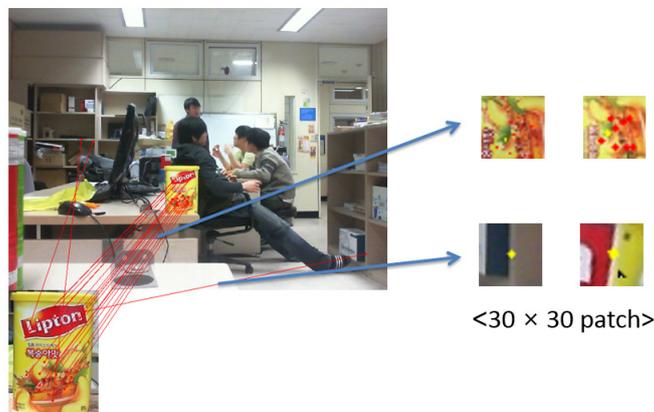


그림 2.6: SIFT 특징 주변의 크기와 방향을 정규화하여 얻은 이미지 조각

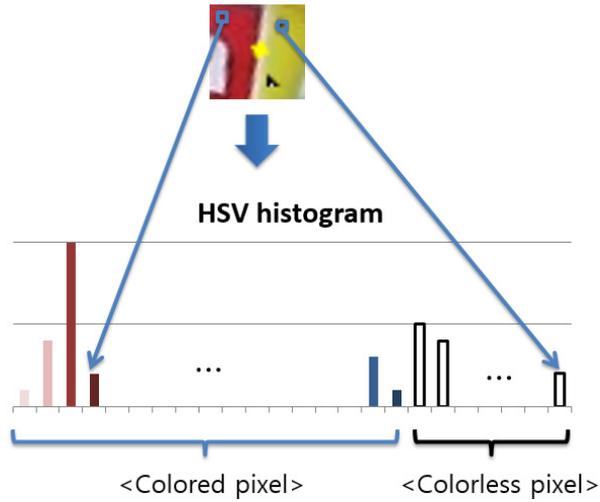


그림 2.7: SIFT 특징 주변의 이미지를 크기와 방향을 정규화하여 얻은 이미지 조각

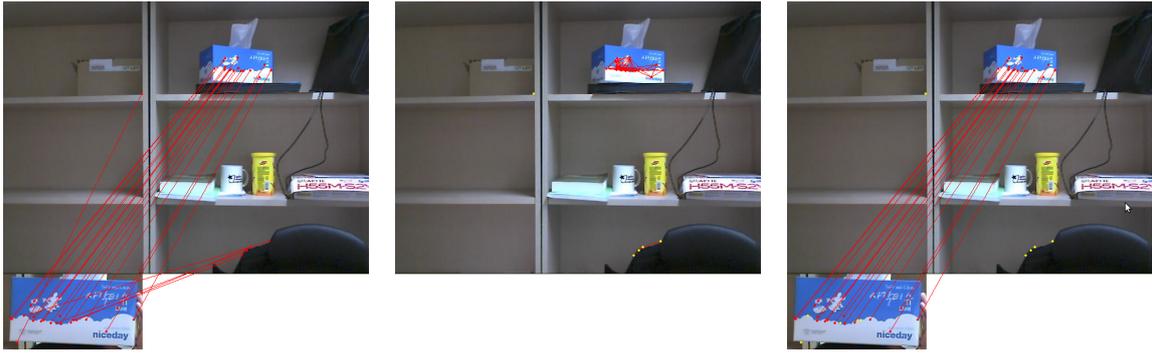
기준값보다 큰 픽셀은 색이 있는 픽셀로 분류되어 전체 색상을 10등분하고 각 색상에 대해 채도를 5등분한 50개의 구간 중에 하나에 속하게 된다. 그렇지 않은 픽셀은 색이 없는 픽셀로 분류되어 전체 밝기를 10등분한 10개의 구간 중에 하나에 속하게 된다. 히스토그램은 SIFT 특징점 주변을 맞추어 자른 30px×30px 이미지의 900개 픽셀에 대하여 총 60개 구간에 대해 집계하고 정규화하여 얻어진다. 그림 2.7에서 한 특징점 주변의 이미지에 대한 히스토그램의 예를 보여준다. 이렇게 얻어진 HSV 히스토그램 간의 비교는 앞에서 언급한 Hellinger distance로 계산하여 특정 기준값보다 큰 SIFT 특징점의 쌍은 다른 것으로 판단하여 제외한다.

둘째로, SIFT 특징점의 공간정보를 사용한다. 촬영한 이미지에 찾고자 하는 물체가 있다면, 이미지 속 물체의 SIFT 특징은 한 곳에 모여있게 된다. 또, 이때 이미지 상에 나타난 물체의 SIFT 특징의 크기는 이미지 상에 나타난 물체의 크기에 비례하게 될 것이다. 이를 이용해 남은 SIFT 특징점들 중에서 물체가 될 SIFT 특징점의 집합을 찾는다.

촬영한 이미지의 SIFT 특징에서 그룹을 만드는 과정은 다음과 같다. 이미지 상의 SIFT 특징점을 꼭지점(vertex)으로 하는 무향 그래프(undirected graph)를 그리는데, 이 때 그래프의 변(edge)은 변이 있는 두 SIFT 특징(꼭지점)이 인접함을 나타낸다. 임의의 두 SIFT 특징점에 대해서 각각의 SIFT 특징 중에서 하나라도 그 크기가 둘 사이의 거리보다 클 경우 두 특징점을 변으로 잇는다.

이렇게 만들어진 SIFT 특징점의 그래프에서 연결성분(connected component: 임의의 두 꼭지점이 경로로 연결된 부분그래프)들을 찾고 그 중에서 가장 큰 연결성분의 꼭지점을 물체가 있을 SIFT 특징점의 집합으로 한다. 이렇게 얻은 가장 큰 연결성분이 3개 이상의 꼭지점을 가지고 있을 때 물체를 인식하였다고 판단한다.

그림 2.8은 공간정보를 사용하여 물체로 판단할 SIFT 특징의 집합을 찾는 과정을 보여준다. 왼쪽은 Hellinger distance를 사용하여 SIFT 특징점을 대응시키고, 색상정보까지 사용하여 얻은 SIFT 특징점 매칭을 나타낸다. 가운데 이미지는 이 SIFT 특징점으로 연결성분을 만드는 것을 보여준다. 오른쪽에서 가장 큰



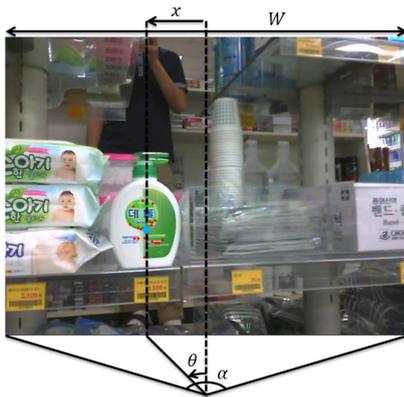
(a) 공간정보 사용 전의 SIFT 특징 매칭 (b) 공간정보를 사용한 SIFT 특징점의 그 래프 (c) 공간정보 사용 후의 SIFT 특징 매칭

그림 2.8: 인식률을 높이기 위한 공간정보의 사용

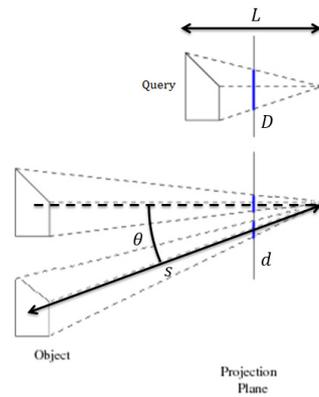
연결성분의 SIFT 특징점들을 실제 물체 이미지의 SIFT 특징을 대응시킨 결과를 보여준다.

2.4 인식한 물체의 위치 계산

탐사로봇은 공간상에서 움직이면서 물체를 찾고 물체를 발견하였을 때 물체 이미지와 같은 크기로 물체를 찍는 위치에 도달하는 것을 목표로 한다. 이를 위해 촬영한 이미지에서 물체를 발견하였을 때 현재 로봇에 대한 물체의 위치를 계산하여야 한다. 카메라의 가로 방향 화각과 물체 이미지를 촬영할 때 물체와 카메라와의 거리를 알고 있는 경우에 카메라에 대한 물체의 수평방향 위치(거리와 방향)를 계산할 수 있는데, 그 계산방법은 다음과 같다.



(a) 이미지 상의 물체 위치와 정면방향에 대한 각도



(b) SIFT 특징의 평균 크기비와 물체와의 거리

그림 2.9: 인식한 물체의 위치 계산

카메라로 촬영한 이미지는 카메라 앞의 공간을 투영 평면(projection plane)에 원근투영(perspective projection)한 것이다. 그러므로 이미지 상에서 인식한 물체의 수평 위치는 물체의 정면방향에 대한 각도의 탄젠트에 비례한다. 따라서 카메라로 촬영한 이미지의 양 끝 사이의 각도, 즉 화각을 알고 있으면 인식한 물체의 정면방향에 대한 각도를 계산할 수 있다. 그림 2.9(a)에서

$$\theta = \arctan(\tan(\alpha/2) \times \frac{x}{W/2}) \quad (2.4)$$

여기서 α 는 카메라의 수평 방향 화각, W 는 이미지의 전체 폭, x 는 가운데를 기준으로 한 물체의 수평 위치이다.

앞 절에서 언급하였듯이 SIFT 특징점 검출에서 그 크기가 얻어진다. 물체의 이미지와 촬영한 이미지의 SIFT 특징이 올바르게 매칭되었다면, 물체의 SIFT 특징과 매칭된 촬영한 이미지 상의 SIFT 특징의 크기는 카메라와 물체 사이의 거리에 반비례하게 나타날 것이다. 즉, SIFT 특징의 크기비 r_{SIFT} 는 물체 이미지와 촬영한 이미지에서 각각 나타난 물체의 폭 D , d 와 다음의 관계를 가진다.

$$r_{SIFT} = \frac{d}{D} \quad (2.5)$$

이를 이용해 매칭된 SIFT 특징 간의 크기비의 평균으로 물체와 카메라 사이의 거리를 구한다. 그림 2.9(b)에서 정면방향에 대해 θ 의 각도에 놓여있는 물체와 카메라 사이의 거리는

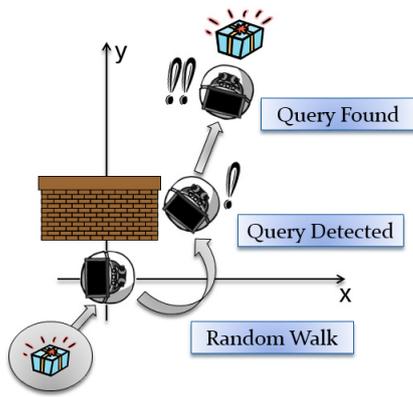
$$s = L \times \frac{D}{d} \times \frac{1}{\cos \theta} = \frac{L}{r_{SIFT} \cos \theta} \quad (2.6)$$

이다. 여기서 L 은 물체 이미지를 찍을 때 물체와 카메라 사이의 거리이다.

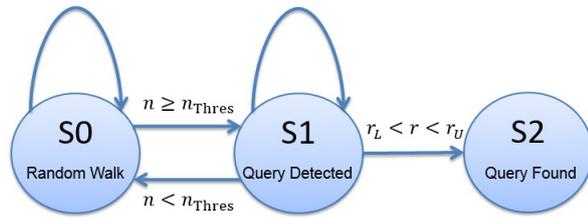
2.5 탐사로봇의 물체인식

탐사로봇은 그림 2.10(a)와 같이 랜덤하게 공간상에서 움직이면서 물체를 발견하고 물체 이미지와 같은 크기의 이미지를 찍는 위치까지 다가가는 것을 목표로 한다. 물체를 찾기 시작할 때부터 물체 앞으로 다가갈 때 같은 크기의 이미지를 찍는 위치에 가는데 까지를 간단한 상태도(State diagram)로 나타낼 수 있다. 그림 2.10(b)에서 그 상태도를 나타내었다.

로봇은 세 가지의 상태 중 하나를 가질 수 있는데 첫 번째 상태 S_0 는 물체를 찾을 때까지 랜덤하게 공간을 움직이는 상태(Random Walk), 두 번째 상태 S_1 는 물체를 발견했을 때의 상태, 마지막 상태 S_2 는 물체 앞으로 다가가서 크기가 같은 이미지를 촬영했을 때의 상태이다. 로봇은 S_0 에서 시작해서 물체를 찾아 랜덤하게 돌아다니다 물체를 발견하면 S_1 으로 전이한다. S_1 에서는 앞 절에서 논의한 바와 같이 발견한 물체의 위치를 계산해서 물체를 같은 크기로 찍을 수 있는 위치로 점차 이동한다. S_1 에서 물체와 같은 크기의 이미지를 촬영하는 위치에 오게 되면 S_2 로 전이하여 작동을 종료하게 된다.



(a) 탐사로봇의 물체인식



(b) 탐사로봇의 상태도

그림 2.10: 탐사로봇의 상태도

상태간의 전이에 기준이 되는 입력은 매칭된 물체의 SIFT 특징의 수와 매칭된 SIFT 특징 간의 평균 크기 비이다. S0에서 촬영한 이미지에서 매칭된 물체의 SIFT 특징의 수가 기준치값 이상이 되면 S1으로 전이하고 S1에서 촬영한 이미지에서 기준치값(실험에서는 3) 미만의 SIFT 특징이 매칭될 경우 S0로 전이한다. S1에서 매칭된 SIFT 특징쌍의 평균 크기비가 1 근처의 값(실험에서는 0.95에서 1.05 사이의 값)을 가지게 되면 S2로 전이하게 된다.

제 3 장 결과

3.1 실험환경

이 장에서는 앞에서 제안한 방법을 사용한 물체인식 알고리즘과 이를 이용한 탐사로봇의 성능을 검증한다. 물체인식 알고리즘은 탐사로봇에서 사용하는 카메라를 사용해 물체의 이미지가 주어졌을 때 촬영한 이미지 속에서 물체인식한 결과를 기존의 방법과 비교함으로써 검증한다. 탐사로봇은 물체의 이미지가 주어졌을 때 실험환경에서 물체를 탐색하는 실험으로 물체인식 알고리즘과 인식한 물체의 위치계산이 응용되는 것을 확인한다.

그림 3.1에 실험에서 만들어 사용한 움직이는 탐사로봇을 나타내었다. 제작한 탐사로봇이 그림 3.1(a)에 나와있다. 탐사로봇은 저비용으로 제작되었으며 세 부분으로 되어있는데, 로봇 플랫폼 iRobot Create(그림 3.1(b)) 위의 HP Mini 110 Netbook(그림 3.1(c))과 Logitech C250 웹캠(그림 3.1(d))으로 구성되어있다. iRobot Create와 웹캠이 넷북에 연결되어 있으며 넷북에서 웹캠으로 촬영한 이미지로 물체를 인식하고 iRobot Create를 제어한다.



(a) 탐사로봇



(b) iRobot Create



(c) HP Mini 110 Netbook



(d) Logitech C250

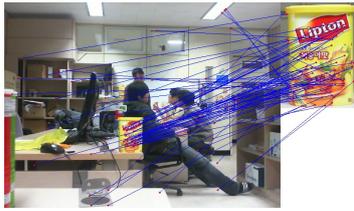
그림 3.1: 탐사로봇과 그 구성

3.2 물체인식 알고리즘 결과

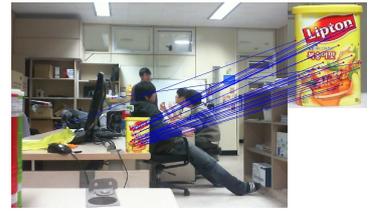
본론에서 제시한 물체인식 알고리즘이 물체의 이미지와 촬영한 이미지의 SIFT 특징을 매치하는 각각의 과정과 기존의 방법에서 SIFT 특징을 매칭한 결과를 3.2에 나타내었다. 물체의 이미지와 카메라로 촬영한 이미지가 그림 3.2(a)와 같이 있을 때 가장 먼저 Hellinger distance를 사용하여 물체의 이미지와 촬영한 이미지의 SIFT 특징을 비교하여 매치한다. 실험에서는 L1-정규화된 SIFT 특징에 대해 가장 가까운 SIFT 특징간의 거리가 0.40 이하일 때 매치된 것으로 인식한다. SIFT 특징의 비교만을 통해 인식한 결과를 그림 3.2(b)에 나타내었다. 이 단계에서 80개의 SIFT 특징쌍이 매칭되었다. 약한 기준을 사용하여 SIFT 특징을



(a) 촬영한 이미지와 인식할 물체



(b) Hellinger distance를 거리함수로 사용해 SIFT 특징을 매칭한 결과



(c) (b)의 결과에 색상정보를 사용하여 매칭한 결과



(d) (c)의 결과에 공간정보를 사용하여 물체를 인식한 결과



(e) 기존의 방법(David Lowe의 예제)으로 인식한 결과

그림 3.2: 물체인식 알고리즘의 결과와 기존의 방법과의 비교

매칭하였기 때문에 많은 수의 옳은 매칭(true positive)뿐 아니라 틀린 매칭(false positive)도 포함되었다. 그 다음으로 2.3절에서 언급한 방식으로 색상정보를 사용해 앞의 매칭에서 색상이 다른 것을 제외한다. 그 결과는 그림 3.2(c)와 같다. 많은 수의 틀린 매칭이 이 단계에서 제외되며 이 단계에서 33개의 매칭만이 남는다. 마지막으로 이 결과에 공간정보를 사용하여 가장 크게 모여 있는 SIFT 특징들을 모아 물체를 인식한다. 그 결과는 그림 3.2(d)와 같으며 다른 것들에 대해 멀리 떨어진 SIFT 특징들이 제외되었음을 볼 수 있다. 전체 물체인식 알고리즘의 결과 31개의 SIFT 특징쌍이 매치되었다. 그림 3.2(e)에 David Lowe의 예제로 SIFT 특징을 매치한 결과를 나타내었는데 SIFT 특징값의 비교만으로 매치하는 이 방법으로는 18개의 SIFT 특징쌍이 매치되었다. 단순히 SIFT 특징값만을 사용하는 것보다 다른 정보를 함께 사용하였을 때 성능이 더 좋은 것을 알 수 있다.

추가로 물체의 이미지가 선명하지 않거나 물체가 카메라에 대해 비스듬하게 놓인 경우에 본 논문에서 후처리를 했을 때와 후처리 없이 SIFT 특징값만을 비교하였을 때의 성능을 비교하였다. 다른 각도로 놓여 있고 카메라의 해상도에 비해 비교적 멀리 떨어져 있는 물체를 인식한 결과를 그림 3.3에 나타내었다. 그림 3.3(a)에 SIFT 특징값만을 비교하여 SIFT 특징을 매치한 결과를 나타내었다. 이 때 SIFT 특징만을 사용하여 매치하기 때문에 높은 정확률(precision)을 가지는 낮은(강한) 기준치값을 사용하였다. 결과에서 볼 수 있듯이 다른 각도로 놓인 물체에 대해 적은 수의 SIFT 특징만이 매칭되었음에도 불구하고 잘못된 매칭이 있음을 알 수 있다. 이에 반해 색상, 공간정보를 사용하여 매칭한 경우(그림 3.3(b))에는 더 좋은 성능을 보인다. 그림



(a) SIFT 특징의 매칭만을 사용하여 물체를 인식한 결과



(b) 색상, 공간정보를 사용한 후처리를 사용한 물체 인식 결과

그림 3.3: 색상, 공간정보를 사용한 SIFT 특징의 매칭 결과

표 3.1: 색상, 공간정보를 사용한 SIFT 특징의 매칭 결과

	기존 방법	논문의 방법
올바른 매칭의 수	2	10
틀린 매칭의 수	1	1

3.3(b)의 노란 점은 약한 기준치값으로 SIFT 특징을 비교하여 처음에는 매칭되었으나 후처리를 통해 제외된 SIFT 특징점을 나타낸다. 두 경우의 매칭 결과를 표 3.1에 나타내었다.

3.3 실험결과

본 논문에서 제안한 물체인식 알고리즘을 사용하여 물체인식을 위한 탐사로봇을 제작하였다. 3.1절에서 언급한 바와 같이 로봇 플랫폼과 넷북과 웹캠으로 이루어진 로봇을 구성하고 여기에 물체의 이미지를 받아 물체를 찾고 같은 크기의 사진을 촬영하는 프로그램을 작성하였다. 프로그램의 흐름은 2.5절에서 서술한 바와 같이 랜덤하게 주변을 탐색하며 물체를 찾고 물체를 인식한 뒤에는 물체의 위치를 계산하여 주어진 물체의 이미지와 동일한 크기로 물체를 찍는 위치로 이동하게 작성되었다.

장애물이 놓인 실험환경에서 물체를 찾는 탐사로봇의 물체인식 과정과 결과를 그림 3.4에 나타내었다. 물체의 이미지(그림 3.4(e))가 주어진 탐사로봇이 장애물에 물체가 가려진 초기 위치(그림 3.4(a))에서 물체의 탐색을 시작한다. 먼저 랜덤하게 네 방향으로 이동(그림 3.4(b))하며 주변을 45도 간격으로 촬영하며 물체를



(a) 탐사robot의 초기 위치. 장애물이 있다.



(b) 랜덤하게 이동하며 주변을 탐색하는 탐사robot



(c) 물체를 인식하여 다가가는 탐사robot



(d) 최종적으로 물체를 찾은 탐사robot



(e) 주어진 물체의 이미지



(f) 탐사robot이 찾은 물체의 이미지

그림 3.4: 물체인식 탐사robot의 실험결과

찾는다. robot이 물체를 발견하면 그림 3.4(c)와 같이 물체의 위치를 계산하여 다가간다. 물체를 같은 크기로 찍을 수 있는 위치에 오게 되면 그림 3.4(d)와 같이 탐사를 종료한다. 그림 3.4(f)에 탐사robot이 최종적으로 찾은 물체의 이미지를 나타내었다.

제 4 장 토의

논문에서 제시한 대로 물체의 이미지와 촬영한 이미지의 SIFT 특징점을 매칭함에 있어 SIFT 특징값의 차이에 대한 기준치값을 약하게 하여 매칭한 뒤 잘못된 매칭을 색상, 공간정보를 사용하여 제외하는 방법은 멀리서 보이는 물체나 해상도가 낮은 카메라 또는 비스듬한 방향의 물체에 대해서 SIFT 특징점의 매칭 성능을 높일 수 있는 방법으로 사용할 수 있다. 이러한 경우 사진의 노이즈로 인해 옳은 특징점의 매칭일지라도 SIFT 특징값 간에 큰 거리를 가질 수 있다. 이 경우 SIFT 특징의 거리에 대한 기준치값을 약하게 두어 재현율을 높이고 틀린 SIFT 특징의 매칭은 색상정보를 사용해 제외하는 방법으로 정확률을 높여 전체적인 알고리즘의 성능을 높인다. 실제 실험에서 낮은 성능의 카메라에 대해 상대적으로 멀리 그리고 비스듬히 놓인 물체에 대해 개선된 SIFT 특징의 매칭을 볼 수 있으며 그림 3.3은 그러한 경우에 대한 실험결과를 보여준다. 그림 3.3(b)의 결과에 노란 점이 많은데 이는 본 논문에서 SIFT 특징간의 거리에 대한 기준치값을 충분히 낮추어 많은 SIFT 특징쌍을 매칭하기 때문이다. 하지만 이 점들 중 대부분은 특징 주변 영역에 대한 HSV 히스토그램의 비교로 제외된다.

물체인식 알고리즘을 구현하는 방법에서 고려해야 할 사항이 두 가지가 있다. 첫 번째는 SIFT 특징간의 Hellinger distance로 매칭시키는 과정에서 SIFT 특징쌍이 바르게 매칭되었다고 판정할 기준치값과 그 이후에 HSV 히스토그램으로 SIFT 특징쌍을 제외시킬 기준치 값을 정하는 것이다. 본 논문의 구현에서는 이를 여러 시도를 통해 경험적으로 정하였다. SIFT 특징 간의 Hellinger distance에 대한 기준은 0.3 이하인 경우 SIFT 특징쌍이 옳게 매칭된 것으로 보았다. SIFT 특징 간의 색상정보를 비교할 때는 정규화된 HSV 히스토그램 간의 Hellinger distance가 $\frac{2}{3}$ 이상이면 색상이 다른 것으로 보고 SIFT 특징 쌍의 매칭에서 제외하였다. 이 기준값을 정하는 것은 성능을 최대화하기 위해 함께 변화시키면서 고려해야 하는데, 본 논문에서는 경험적으로 결정하였지만 많은 수의 데이터와 다양한 기준값에 대한 실험으로 최적의 값을 찾아 알고리즘의 성능을 향상시킬 수 있을 것이다.

두 번째는 공간정보를 사용해 SIFT 특징점의 그래프를 그리는 과정에서 SIFT 특징의 크기와 특징점 사이의 공간적 거리의 기준을 정하는 것이다. 2.3절에서 설명한 SIFT 특징점의 그래프를 그리는 과정에서 두 SIFT 특징점을 잇는 변은 두 특징점의 거리가 그 크기보다 클 경우 그린다고 설명하였다. 하지만 단순히 SIFT 특징의 크기에 대한 정비례로 둘 사이의 거리와 비교할 경우 한 SIFT 특징점이 크기가 작은 경우 같은 물체의 SIFT 특징은 연결성분으로 잇지 못하고 큰 경우 멀리 있어 관련이 없는 특징점까지 연결성분으로 잇게 되었다. 이를 피하기 위해서 본 논문의 구현에서는 둘 사이의 거리와 비교하는 SIFT 특징의 크기에 대한 함수를 정비례가 아닌 SIFT 특징의 크기의 제곱근에 비례하는 함수로 하여 이보다 거리가 짧은 경우 SIFT 특징점 간의 변을 이었다. 이 또한 위의 문제점을 피하면서 성능을 개선할 다른 방법을 찾아 사용할 수

있을 것이다.

탐사로봇의 구현에 있어 특이한 점으로는 물체를 찾는 시간을 단축하기 위해 스레드 프로그래밍을 사용한 것이다. 넷북의 제한된 연산능력 때문에 SIFT 특징을 추출하는 시간이 걸리는데 랜덤하게 공간을 탐색하는 상태(Random Walk) 동안 실시간으로 물체를 찾는 시간을 단축하기 위해서 스레드 프로그래밍을 하였다. 랜덤하게 이동함과 동시에 이동하기 전에 촬영한 사진에 대한 SIFT 특징 추출 및 매칭을 하는 스레드를 실행시켜 물체를 인식하는 방식이다. 리눅스에서 작동하는 C 프로그램으로 Pthread API를 사용해서 작성하였다.

제 5 장 결론

본 논문에서는 특징 기반 접근법의 물체인식의 성능을 개선할 수 있는 물체인식 알고리즘을 제안하고, 이를 로봇 플랫폼과 넷북과 웹캠으로 이루어진 로봇에 적용한 물체인식을 위한 저비용의 탐사로봇을 구현하였다. 물체인식 알고리즘은 SIFT 특징을 사용한 특징 기반 접근법에 촬영한 이미지에서 얻을 수 있는 SIFT 특징점의 색상, 공간정보를 사용하여 그 성능을 높였다. 본 논문에서 제안한 물체인식 알고리즘은 실험을 통해 기존의 방법과 비교하여 성능을 개선함을 확인하였고, 제작한 탐사로봇이 실험환경에서 물체를 인식하는 실험으로 그 성능을 검증하였다. 본 논문에서 제안한 물체인식 방법은 낮은 성능을 가진 카메라를 사용하여 특징 기반 접근법으로 물체인식을 해야 하는 경우 유용하게 사용할 수 있을 것이다.

참고문헌

- [1] Lowe, David G. “Object recognition from local scale-invariant features.“ *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*. Vol. 2. IEEE, 1999.
- [2] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, “Color-based probabilistic tracking“ *Proc. European Conference on Computer Vision* pp. 661–675, 2002.
- [3] R. Hess and F. A., “Discriminatively trained particle filters for complex multi-object tracking,“ *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009.
- [4] Kim, Junyoung, et al. “Actionable topological mapping for navigation using nearby objects.“ *Automation Science and Engineering (CASE), 2012 IEEE International Conference on*. IEEE, 2012.
- [5] iRobot, <http://www.irobot.com>
- [6] OpenCV, Open Source Computer Vision software library, <http://opencv.org/>