



저작자표시 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.
- 이차적 저작물을 작성할 수 있습니다.
- 이 저작물을 영리 목적으로 이용할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#) 

석 사 학 위 논 문

퍼지 - 픽셀 기반 모노 카메라 탐사 로버의
상대적 위치 추정

The seal of Pusan National University is a large, light blue circular emblem in the background. It features a central green shield with a white crown on top and the Korean characters '부산' (Busan) inside. Below the shield is the year '1946'. The outer ring of the seal contains the text 'PUSAN NATIONAL UNIVERSITY' in English and '부산대학교' in Korean.

이 나 현

부산대학교 대학원
전기전자컴퓨터공학과

2018년 2월

퍼지 – 픽셀 기반 모노 카메라 탐사 로봇의 상대적 위치 추정

이 논문을 공학석사 학위논문으로 제출함

이 나 현

부산대학교 대학원

전기전자컴퓨터공학과

지도교수 이장명

이 나 현 의 공학석사 학위논문을 인준함

2017년 12월 23일

위원장	최 준 영	인
-----	-------	---

위 원	이 장 명	인
-----	-------	---

위 원	박 성 경	인
-----	-------	---

차 례

표 차 례.....	ii
그 림 차 례.....	iii
1 서론.....	1
1.1 연구의 배경과 목적.....	1
2 제한 조건 및 위치인식 알고리즘.....	3
2.1 탐사 로버의 CPU 성능에 의한 비용적 문제.....	3
2.2 제한 조건에 적합한 로버의 위치인식 알고리즘.....	4
3 객체 인식 알고리즘.....	5
3.1 객체 인식 알고리즘.....	5
3.2 B-Spline 보간법.....	6
3.3 ASIFT.....	9
3.3.1 Scale Invariant Feature Transform(SIFT).....	9
3.3.2 Affine Scale Invariant Feature Transform(ASIFT).....	11
3.4 매칭 이상치 제거.....	13
4 퍼지테이블 - 픽셀기반 위치추정 알고리즘.....	17
4.1 퍼지 추론 시스템.....	17
4.2 입-출력 변수 및 소속함수(Membership Function).....	18
4.3 퍼지 규칙.....	21
4.4 퍼지 추론 및 비퍼지화.....	23
5 실험 및 결과.....	25
5.1 실험 방법.....	25
5.2 실험 결과.....	27
6 결론.....	29

표 차 례

[표 1] 퍼지 규칙 표.....	22
[표 2] 실험 결과.....	28



그림 차례

[그림 1] 로버 위치인식 알고리즘.....	4
[그림 2] 객체 인식 알고리즘	5
[그림 3] B-Spline 보간법	6
[그림 4] B-Spline 보간 적용 전, 후 결과.....	8
[그림 5] SIFT 알고리즘 2단계	10
[그림 6] SIFT 알고리즘 3단계	10
[그림 7] 어파인 맵	11
[그림 8] ASIFT 알고리즘 과정.....	12
[그림 9] 카메라 축 기울기 변화 상황에서의 SIFT 수행 결과.....	12
[그림 10] 카메라 축 기울기 변화 상황에서의 ASIFT 수행 결과	13
[그림 11] 2차 분포 필터 적용 전 영역 추출 결과	15
[그림 12] 2차 필터 적용 후 영역 추출 결과	16
[그림 13] 퍼지 추론 시스템 블록도	17
[그림 14] 회전에 따른 가로 세로 길이 변화.....	18
[그림 15] y/x 의 소속 함수	18
[그림 16] 거리에 따른 객체 크기 차이	19
[그림 17] 픽셀 수의 소속함수	19
[그림 18] 거리의 소속함수.....	20
[그림 19] 입출력 퍼지 규칙	21
[그림 20] Mamdani 퍼지 추론	23
[그림 21] 퍼지 룩업 테이블 작성 시 실험 환경.....	25
[그림 22] 두 번째 실험 환경	26
[그림 23] 70cm 거리 영상의 퍼지 추론 결과	27

퍼지 – 픽셀 기반 모노 카메라 탐사 로버의 상대적 위치 추정

이 나 현

부산대학교 대학원 전기전자컴퓨터공학과

요약

우주 탐사 로버의 경우 안정성을 고려하여 저성능 CPU를 사용하며 안정성을 만족하는 고성능 CPU는 비용적으로 매우 큰 부담이 된다. 본 논문에서는 저성능 CPU를 가진 탐사 로버에 적합한 원거리 탐사 위치 인식 방식을 제안하며, 연속적이고 많은 양의 영상 촬영이 필요한 Visual Odometry 방식 대신에 한 개의 영상만으로 특정 객체 인식을 통해 상대적인 거리를 추정하여 로버의 위치를 추정하는 퍼지-픽셀 기반 상대적 위치 추정 알고리즘을 제안한다. 객체 인식 시에는 카메라 축 기울기 변화에 강인한 Affine SIFT 알고리즘을 사용하며, ORSA와 특징점 분포를 이용한 2차 분포 필터를 통해 이상치를 제거한다. 객체 인식 후 인식한 객체의 픽셀 수와 가로, 세로 길이 비를 미리 작성해 두었던 퍼지 룩업 테이블에 적용시켜 거리를 추출하여 상대적 위치를 추정한다. 그리고 제안한 알고리즘은 로버가 정면에 위치할 경우, 측면에 위치 할 경우 등 여러 상황에서의 정확도를 실험을 통해 검증한다.

1 서론

1.1 연구의 배경과 목적

우주는 매우 방대하고 아직 잘 알려지지 않은 공간이며 꾸준한 탐사 작업이 필요한 환경이다. 달이나 화성과 같은 특수환경에서 활동하는 탐사 로버의 임무 기능은 크게 생존기능과 탐사기능으로 나뉘어지는데 생존기능은 환경 적응, 이동, 통신, 동면 등과 같은 임무이며, 탐사기능은 영상촬영, 표토층 분석, 암석 분석 등과 같은 임무이다[1][2].

위의 생존 및 탐사 기능의 수행을 위해서는 로버의 이동이 필요하며, 이동 방법에는 자율주행 방법과 비 자율주행 방법이 있다. 자율주행 방법은 우주 환경에서의 실시간 데이터 처리가 가능해야 하기 때문에 중량적, 비용적 조건이 만족되어야 한다. 그리고 안정적인 이동을 위해서는 로버의 위치인식, 장애물 감지 및 회피 등의 기술이 필요하다. 위치인식의 경우, 주행 명령을 직접적으로 전송하는 비 자율주행 방식을 사용하더라도 달 표면 특성 상 바퀴의 Slip현상으로 인해 전송한 주행 거리와 실제 주행 거리가 동일하지 않을 가능성이 높아 위치인식이 반드시 필요하다.

탐사 로버에 사용하는 위치인식을 위한 센서로는 카메라, 레이저센서 등 여러 센서가 있다. 그 중에서 카메라는 탐사 로버에 공통적으로 사용되는 센서이다. 로버에 따라 사용되는 카메라의 종류와 개수, 부착 위치 등이 다르지만 카메라로 얻은 영상은 생존기능과 탐사기능 두 가지에 다 사용될 수 있으므로 카메라는 탐사 로버에 필수적으로 사용된다.

탐사 로버에서 일반적으로 사용하는 위치인식 알고리즘은 스테레오 카메라로부터 영상을 획득하여 Visual Odometry 알고리즘을 사용한다.[3][4] 이 알고리즘은 Optical flow를 이용하는 알고리즘으로써 이전 영상과 다음 영상의 비교를 통해 밝기 패턴의 위치 변화를 속도벡터로 나타내어 이동량과 방향을 측정하고 위치인식을 하는 방식이다. 이 방식은 다른 추가적인 장치가 필요하지 않고 전방 영상만으로도 위치인식이 가능하다는 장점이 있다. 그러나 이전 영상과 현재 영상의 차이를 이용하기 때문에 영상 촬영 간격에 비해 움직임이 클 경우 정확도가 떨어지게 된다. 그래서 이 알고리즘은 짧은 시간 간격의 영상 촬영이 필요하다. 그러나 탐사 로버의 경우 CPU 성능이 곧 비용

과 직결되므로, 많은 영상 저장이 필요한 Visual Odometry와 같은 알고리즘은 비용적 부담이 크다. 본 논문에서는 저성능 CPU를 가지고도 영상을 이용한 위치인식을 가능하게 하는 새로운 알고리즘을 제안한다. 2장에서는 한국형 탐사 로버의 CPU 성능으로 인한 비용적 문제에 대해 세부적으로 서술하고, 이에 적합한 위치인식 방식에 대해 서술한다. 그리고 3장, 4장에서는 제안한 새로운 위치인식 알고리즘을 두 장에 걸쳐 서술한다. 5장에서는 실제 실험을 통해 결과와 성능을 확인하고, 마지막 6장에서 결론으로써 마무리한다.



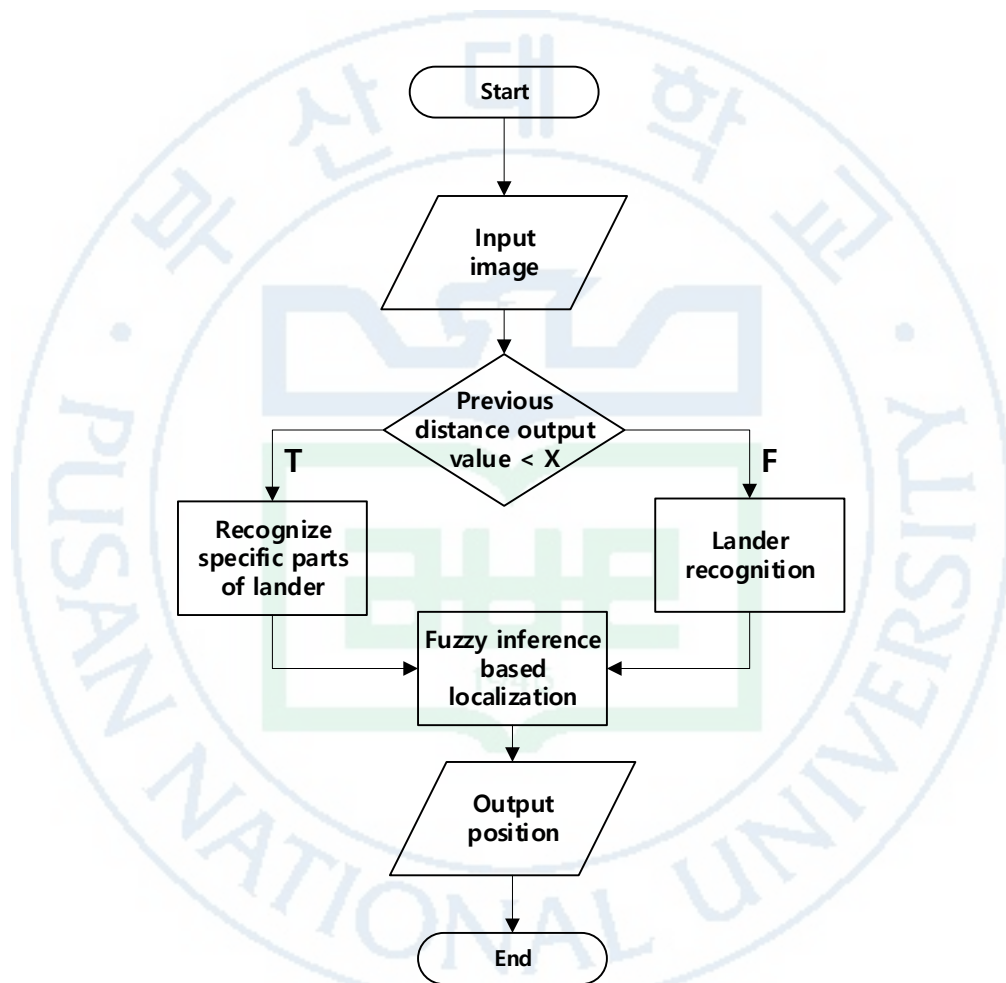
2 제한 조건 및 위치인식 알고리즘

2.1 탐사 로버의 CPU 성능에 의한 비용적 문제

우주 환경에서 사용되는 CPU는 일반적으로 사용되는 CPU와 동일하지 않다. 우주용 CPU는 일반적인 CPU에 비해 매우 낮은 사양을 가지고 있는데, 그럼에도 우주용 CPU를 사용하는 이유는 환경 변화에 강인하다는 점 때문이다. 지상에서 사용하는 일반적인 CPU들은 전압과 전류 변동에 극히 취약하고, 냉각장치가 없으면 상온에서도 쉽게 고장이 나며, 방사선에도 취약하다. 그런데 화성의 환경은 온도가 20℃에서 -140℃까지 변화하고, 달의 경우는 130℃에서 -130℃까지 변화한다. 그리고 달이나 화성은 대기가 없거나 매우 미미하기 때문에 태양전파를 지구에서보다 훨씬 직접적으로 강하게 받게 된다. 이렇듯 우주 환경은 지구의 환경과 매우 다르고 변화가 극심하다. 이러한 안전성의 이유로 낮은 사양에도 고가인 우주용 CPU를 사용해야만 한다. 상대적으로 고사양의 우주용 CPU를 사용하는 것은 비용적으로 큰 부담이 되므로 최대한 고사양이 필요하지 않은 알고리즘 제작이 요구되었다. 그래서 본 논문에서는 실시간 비전 연산을 하지 않고 우주 환경에서 영상을 촬영 후, 지상으로 전송하여 지상에서 비전 연산을 수행하는 원격 탐사 방식을 사용하고, 필요한 영상의 수와 화질을 최소화하는 알고리즘을 제안하였다.

2.2 제한 조건에 적합한 로버의 위치인식 알고리즘

본 논문에서 제안하는 로버의 위치인식 알고리즘은 로버 카메라의 pan/tilt를 조정하여 항상 착륙선 방향을 바라볼 수 있도록 유지하는 것이 필요하다. 이렇게 pan/tilt를 조절하여 항상 착륙선을 바라보게 유지할 경우 영상에서 보이는 착륙선의 방면에 따라 상대적인 로버의 위치를 알 수 있게 된다.



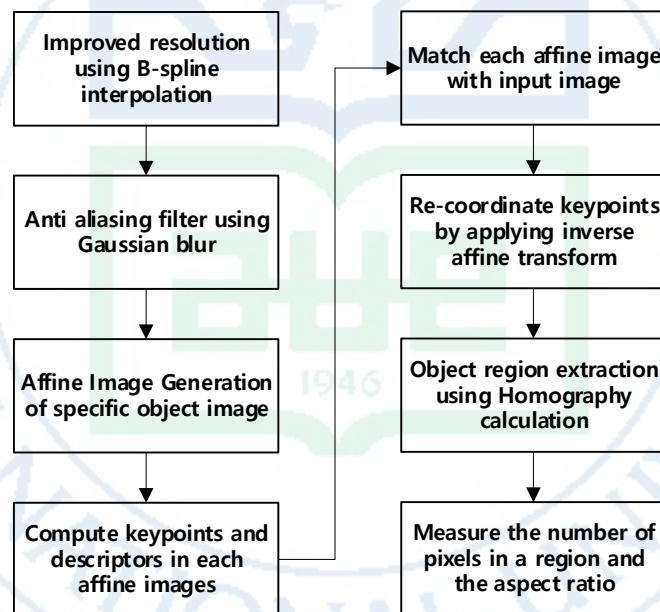
[그림 1] 로버 위치인식 알고리즘.

착륙선과 가까울 경우에는 착륙선 전체를 인식할 수 없으므로 착륙선의 일부분을 인식한다. 일정 거리 이상이 되어 착륙선 전체를 인식 가능하게 되면 착륙선 전체를 인식하여 룩업 테이블 기반으로 로버의 상대적인 위치를 인식하게 된다. 다음 장에서는 착륙선을 인식하기 위한 객체 인식 알고리즘에 대해 서술한다.

3 객체 인식 알고리즘

3.1 객체 인식 알고리즘

2장에서 제안한 로버의 위치인식 알고리즘의 경우 착륙선의 특정 부분이나 전체를 인식해야 한다. 두 경우의 차이는 실제 로버가 촬영한 영상과 매칭시키는 입력 영상에 차이가 있을 뿐, 방식에서는 동일하다. 그러므로 그림 2는 그림 1에서의 착륙선의 특정 부분 인식/ 착륙선 인식 부분을 자세히 나타낸 것이다.



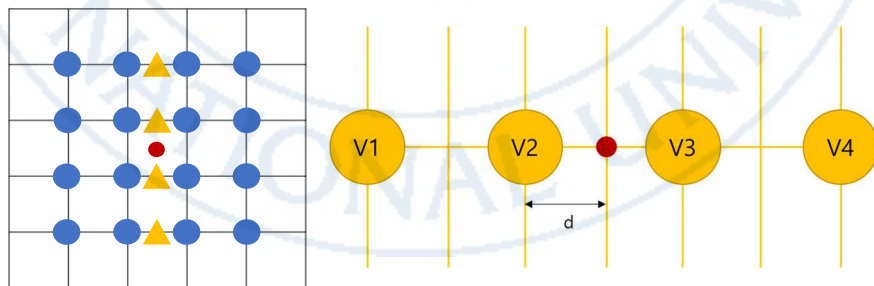
[그림 2] 객체 인식 알고리즘

먼저 착륙선을 인식하기 위해서는 입력 영상의 해상도를 향상시키는 과정이 필요하다. 고화질의 영상을 획득할 수 있다면 당연히 좋겠지만 달탐사 로버의 특성상 낮은 화소의 영상을 획득하기 때문에 해상도를 향상시킬 필요가 있다. 그래서 본 논문에서는 B-Spline 보간법을 이용하여 해상도를 향상시켰다.

3.2 B-Spline 보간법

B-Spline 보간법은 전체 구간을 소구간 별로 나누어 저차수의 다항식으로 나타내어 보간하는 방법으로, 인접한 16개의 화소값과 거리에 따른 가중치의 곱을 사용하여 결정한다. [5][6] Spline 보간법은 일반적인 다항식 보간법에 비해 계산이 쉽고 차수가 높아질수록 값이 크게 요동치는 Runge 현상이 상대적으로 적게 발생한다는 것이다. 보간법에서의 차수는 데이터 수가 많아질수록 높아지는데 영상의 경우 픽셀 수가 데이터가 되므로 차수가 높아질 수 밖에 없다. 이러한 측면에서 Spline 보간법은 매우 효과적이고 효율적인 보간법 이라고 할 수 있다. B-Spline 보간의 식은 수식 (1)과 같다.

$$f(x) = \begin{cases} \frac{1}{2}|x|^3 - |x|^2 + \frac{2}{3} & 0 \leq |x| < 1 \\ -\frac{1}{6}|x|^3 + |x|^2 - 2|x| + \frac{4}{3} & 1 \leq |x| < 2 \\ 0 & 2 \leq |x| \end{cases} \quad (1)$$



[그림 3] B-Spline 보간법

본 논문에서는 달 탐사로 획득한 영상의 해상도를 640 x 480 pixels로 가정하고, 해상도를 B-Spline 보간법을 사용하여 800 x 600 pixels로 향상시켰다. B-Spline 보간법은 수식을 총 5번 적용한다. 그림 3을 참조하면 주변의 4픽셀(파란색 원)에 대해 거리에 대한 가중치를 적용하여 보간된

픽셀(삼각형)을 각각 구한다. 총 16개의 픽셀을 참조하기 때문에 이 과정에서 4번의 수식을 적용한다. 이후 4개의 보간된 픽셀(삼각형)을 이용하여 수식을 적용시켜 최종적으로 보간된 픽셀(빨간색 원)을 구한다. 거리에 대한 가중치를 적용하는 식은 아래의 수식(2)와 같다.

$$\begin{aligned}
 v = & \left\{ -\frac{1}{6}(1+d)^3 + (1+d)^2 - 2(1+d) + \frac{4}{3} \right\} v_1 \\
 & + \left\{ \frac{1}{2}d^3 - d^2 + \frac{2}{3} \right\} v_2 \\
 & + \left\{ \frac{1}{2}(1-d)^3 - (1-d)^2 + \frac{2}{3} \right\} v_3 \\
 & + \left\{ -\frac{1}{6}(2-d)^3 + (2-d)^2 - 2(2-d) + \frac{4}{3} \right\} v_4
 \end{aligned} \tag{2}$$

식 (2)를 d 에 관해서 정리하면 수식(3)과 같다.

$$\begin{aligned}
 v = & \left(-\frac{1}{6}v_1 + \frac{1}{2}v_2 - \frac{1}{2}v_3 + \frac{1}{6}v_4 \right) d^3 \\
 & + \left(\frac{1}{2}v_1 - v_2 + \frac{1}{2}v_3 \right) d^2 \\
 & + \left(-\frac{1}{2}v_1 + \frac{1}{2}v_3 - v_4 \right) d \\
 & + \frac{1}{6}v_1 + \frac{2}{3}v_2 + \frac{1}{6}v_3
 \end{aligned} \tag{3}$$

B-Spline 보간을 이용하여 영상의 해상도를 향상시켜 특정 부분을 확대시켜 확인해 본 결과 B-Spline 보간을 사용하여 해상도를 향상시킨 결과가 사용하지 않고 해상도를 향상시킨 결과에 비해 더 부드럽게 영상이 이어지는 것을 확인할 수 있었다.



[그림 4] B-Spline 보간 적용 전, 후 결과

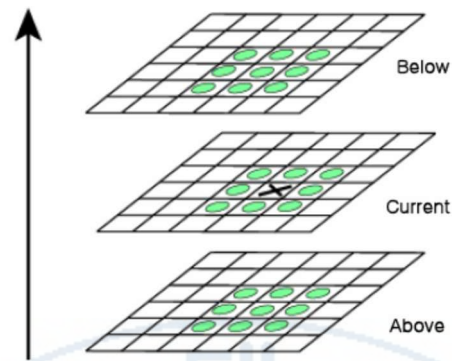


3.3 ASIFT

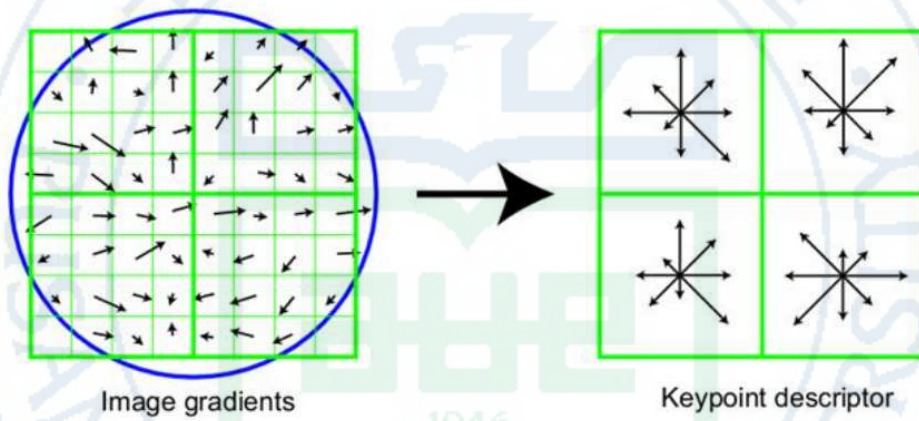
3.3.1 Scale Invariant Feature Transform(SIFT)

본 논문에서 제안한 위치인식 알고리즘은 착륙선을 인식하여 착륙선의 위치를 인식하여 상대적인 로버의 위치를 계산하는 알고리즘이다. 그러므로 해상도를 향상시킨 후 착륙선을 인식하기 위해 지상에서 미리 촬영한 착륙선 영상과 실제 로버가 촬영한 영상에서 각각 특징점을 검출한 후 두 영상의 특징점을 매칭함으로써 착륙선을 인식한다. 그러므로 객체 인식을 위해서는 특징점을 검출하고 매칭하는 알고리즘이 반드시 필요하다.

SIFT 알고리즘은 기본적으로 영상에서 코너점과 같은 식별하기 좋은 특징점들을 선택한 후에 각 특징점을 중심으로 주변 픽셀을 4x4블록으로 나누고 각 블록에 속한 픽셀들의 gradient 방향과 크기에 대한 히스토그램을 구한 후 이 값들을 128차원의 벡터에 일렬로 대입한 것이다. SIFT는 크게 크기 불변, 조명 불변, 회전 불변, 잡음 불변의 4가지 불변성을 특징으로 삼고 있으며, 4단계로 이루어져 있다. 1단계는 서로 다른 분산 값을 갖는 가우시안 함수를 적용한 영상들의 차이를 이용해 얻어진 DoG(Difference of Gaussian)를 통해 로컬 극값을 구하며, 2단계에서는 그림 5와 같이 구해진 로컬 극값들 중에서 인근 8개와 위 아래 영상의 18개 (총 26개) 점과 비교하여 주변 화소보다 크거나 작으면 특징점으로 선정된다. 3단계는 그림 6과 같이 선정된 특징점들의 주변 픽셀을 4x4 블록으로 나누고 각 블록에 포함된 픽셀들의 방향과 크기에 대한 히스토그램을 구하는 것이다. 그리고 4단계에서는 이 특징점들을 128차원의 벡터로 생성하는 것이다. 이러한 SIFT 알고리즘은 영상의 크기 변화에 강인하며, 지역적인 특징을 사용하기 때문에 객체가 다른 객체에 의해 특정 부분이 가려지는 현상에도 강인하다는 장점이 있다. [7][8]



[그림 5] SIFT 알고리즘 2단계



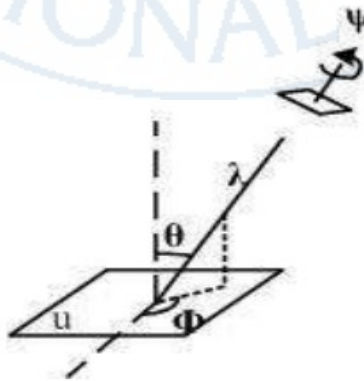
[그림 6] SIFT 알고리즘 3단계

3.3.2 Affine Scale Invariant Feature Transform(ASIFT)

그런데 SIFT 알고리즘은 매칭을 수행하는 두 영상의 카메라 축 기울기 변화가 있을 경우 매칭이 현저히 떨어지게 되는 문제가 있다. 하지만 탐사 로버의 경우 어느 방면에서 착륙선을 인식할지 알 수 없으므로 카메라 축 기울기 변화가 큰 상황에 놓이게 된다. 이를 해결하기 위해 SIFT에 Affine 변환을 적용한 ASIFT(Affine SIFT)를 사용하여 카메라 축 기울기 변화에 강인한 특징점 검출 및 매칭을 수행하였다. [9][10]

기존의 SIFT 알고리즘은 x축 이동, y축 이동, 회전, 크기의 총 4가지 변수만을 고려하여 특징점을 검출한다. 하지만 영상 간의 동일한 특징점 위치 변화는 공간 상의 각도 변화에 대한 요소도 존재한다. ASIFT는 이 공간 상의 각도 변화에 대한 요소를 반영하기 위해 촬영 공간에서 위도(Latitude), 경도(Longitude)의 위치변화를 기존의 SIFT 알고리즘에 추가로 고려한다. 위도와 경도를 고려하기 위해 어파인 맵을 생성하는데, 이 어파인 맵 A 에 대한 행렬식은 수식 4와 같다. 수식 4에서 λ 는 zoom에 해당하고 ψ 는 회전에 대한 변수이다. 그리고 위도와 경도는 그림 7에서 각각 θ , Φ 를 의미한다.

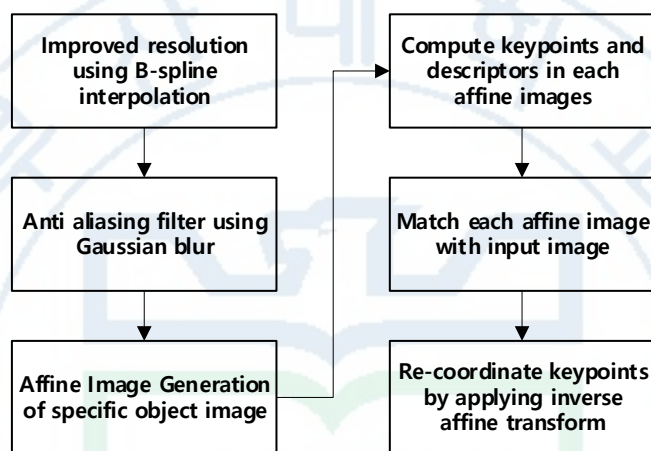
$$A = \lambda \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} t & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\Phi & -\sin\Phi \\ \sin\Phi & \cos\Phi \end{bmatrix} \quad (4)$$



[그림 7] 어파인 맵

다음으로 ASIFT의 알고리즘의 과정을 살펴보면 그림 8과 같이 ASIFT 알고리즘은 위도 θ , 경도

Φ 를 변경해가며 다양한 변환된 이미지를 획득하고 각각의 변환된 이미지의 keypoints와 descriptors를 추출하여 저장한다. 이후 저장된 keypoints를 로버가 촬영한 영상에서 추출한 keypoints와 매칭시킨다. 그리고 최종적으로 어파인 역변환을 적용하여 원래의 입력 이미지의 keypoints 좌표로 변환시켜 저장한다. 그림 9, 그림 10은 카메라 축 기울기 변화가 있는 상황에서 SIFT 적용 결과와 ASIFT 적용 결과를 나타낸다.



[그림 8] ASIFT 알고리즘 과정



[그림 9] 카메라 축 기울기 변화 상황에서의 SIFT 수행 결과



[그림 10] 카메라 축 기울기 변화 상황에서의 ASIFT 수행 결과

두 결과를 비교하면 SIFT 알고리즘에 비해 ASIFT 알고리즘 적용 시 매칭 점의 개수가 확연히 더 많은 것을 확인할 수 있었다.

3.4 매칭 이상치 제거

위의 ASIFT 알고리즘에는 이상치를 제거하기 위한 ORSA(Optimized RANSAC) 필터를 포함하고 있다. [11][12] ORSA는 일반적인 RANSAC 필터에서 NFA(Number of False Alarm)를 도입하여 표준 편차와 정상치 사이의 균형을 유지하는 필터이다. 그러나 ORSA 필터를 적용시키더라도 이상치가 완전히 제거되지는 않는다. 매칭된 특징점의 개수가 아주 많을 경우에는 적은 수의 이상치가 문제가 되지 않지만 거리가 멀거나 카메라 축 기울기 변화가 커서 매칭된 특징점의 수가 많지 않을 경우에는 적은 수의 이상치에도 크게 영향을 받아 인식한 물체의 영역 추출 시 올바른 영역 추출 수행이 되지 않는 경우가 빈번하게 발생한다. 그래서 본 논문에서는 추가적으로 이상치를 제거할 수 있는 2차 분포 필터를 적용하였다.

2차 분포 필터는 ORSA 필터를 거친 결과 매칭 점들의 분포도에 따라 정상치와 이상치를 구분한다. 이 필터는 표준 편차의 값에 따라 상, 하 문턱 값을 다르게 설정하는데, 여기서 상, 하 문턱

값은 정상치와 이상치를 판별하는 기준이다. 즉, 하 문턱 값과 상 문턱 값 사이의 값을 정상치로 판별하고 이외의 값은 이상치로 판별하여 제거하는 것이다. 표준 편차의 값을 구하는 식은 수식 (5)와 같다.

$$\sigma = \sqrt{\frac{(x_1 - m)^2 + (x_2 - m)^2 + \dots + (x_n - m)^2}{n}} \quad (5)$$

상, 하 문턱 값을 설정할 때는 ORSA 적용 후의 결과 매칭점들의 평균, 표준 편차, 표준 오차를 이용한다. 표준 오차를 계산하는 식은 수식 (6)과 같고, 상, 하 문턱 값에 대한 식은 수식 (7)과 같다.

$$e = \frac{\sigma}{\sqrt{n}} = \frac{\sqrt{(x_1 - m)^2 + (x_2 - m)^2 + \dots + (x_n - m)^2}}{\sqrt{n}} \quad (6)$$

$$\begin{aligned} \text{high threshold} &= m + \frac{\sigma}{2} + e * k \\ \text{low threshold} &= m - \frac{\sigma}{2} - e * k \end{aligned} \quad (7)$$

여기서 k값은 표준 편차의 값에 따라 다르게 설정된다. 이렇게 k값을 다르게 설정한 이유는 결과 매칭 점들의 밀집도에 따라 정상치까지 제거되거나, 이상치가 제거되지 않는 상황을 방지하기 위해서이다. 밀집도가 높을 경우 표준 편차 값과 표준 오차 값이 작아 정상치의 범위가 매우 좁아지게 된다. 그래서 필터 적용 시에 정상치까지 제거되는 결과가 발생할 수 있다. 반대로 밀집도가 낮을 때는 이상치가 제거되지 않는 결과가 발생할 수 있다.

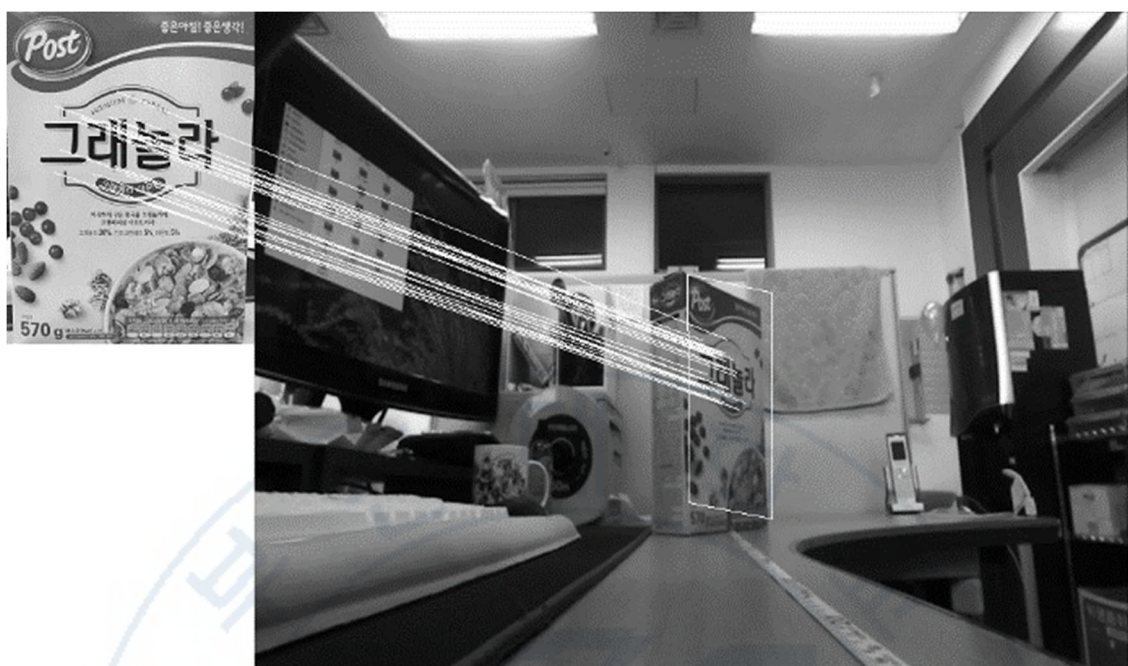
밀집도에 따른 k 값은 여러 영상에 적용을 통해 적합한 값을 선정하였고, 선정된 k 값은 수식 (8)

과 같다. 그림 8의 결과에서 영역을 추출한 결과가 그림11와 같고, 2차 분포 필터를 적용하여 영역을 추출한 결과는 그림 12과 같다. 결과를 비교했을 때 2차 분포 필터 적용 후가 훨씬 더 올바른 영역을 추출하는 것을 확인할 수 있었다.

$$\begin{cases} \sigma \geq 50 & \rightarrow k=1 \\ 20 \leq \sigma < 50 & \rightarrow k=5 \\ \sigma < 20 & \rightarrow k=10 \end{cases} \quad (8)$$



[그림 11] 2차 분포 필터 적용 전 영역 추출 결과

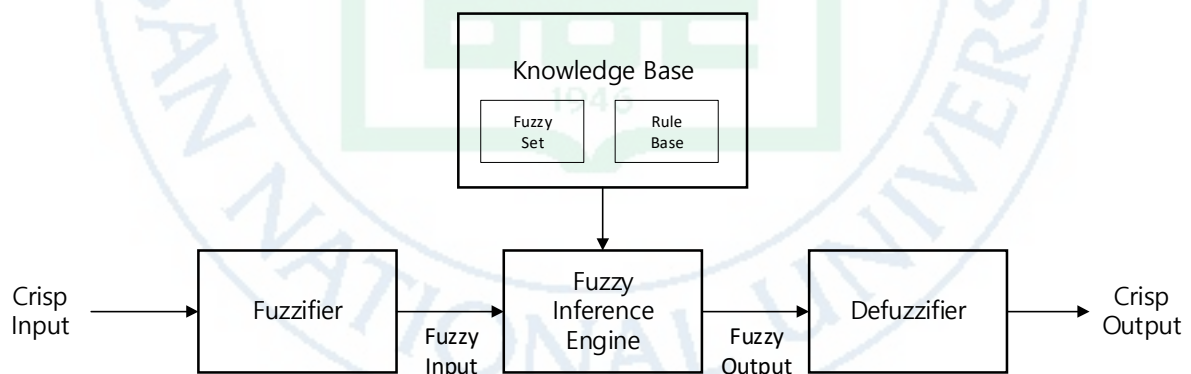


[그림 12] 2차 필터 적용 후 영역 추출 결과

4 퍼지테이블 - 픽셀기반 위치추정 알고리즘

4.1 퍼지 추론 시스템

퍼지 추론 시스템(Fuzzy Inference System)은 언어나 생각과 같이 추상적인 개념을 다룰 수 있고 입력과 출력의 관계가 비선형이며 물리적 의미를 알 수 없는 경우에도 모델링이 가능하다는 장점을 가지고 있다. 따라서 패턴 인식, 고장 진단, 자동제어, 로봇틱스 등 다양한 분야에서 응용되고 있다. FIS는 퍼지화(Fuzzifier), 지식베이스(Knowledge Base), 퍼지 추론(Fuzzy Inference Engine), 비퍼지화(Defuzzifier)로 구성되며, 기본 블록도는 그림 13과 같다. 퍼지화는 입력되는 센서 값을 퍼지 관계와 퍼지 논리의 추론 규칙을 이용하여 퍼지 값으로 생성하며 지식베이스는 퍼지 규칙 및 멤버쉽 함수를 정의한다. 퍼지 추론은 퍼지 출력을 계산하는 추론을 적용하며 비퍼지화는 퍼지 출력 값을 Crisp 출력으로 변환시킨다. [13][14][15]



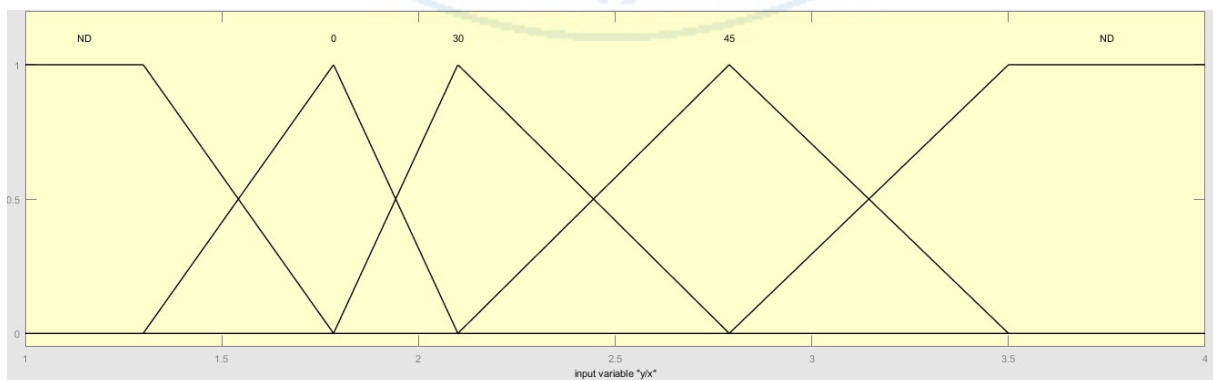
[그림 13] 퍼지 추론 시스템 블록도

4.2 입-출력 변수 및 소속함수(Membership Function)

거리 추정을 위한 퍼지제어기를 설계하기 위해 필요한 입력과 출력의 변수를 결정해야 한다. 본 논문에서는 2가지 입력을 사용하며 그림 15, 17는 입력에 대한 소속 함수를 보여준다. 그림 15은 입력에 대한 소속 함수로써 영상에서 인식된 객체의 가로 세로 비(y/x)를 의미한다. 영상에서 인식된 객체의 가로 세로 비는 객체가 회전 되어 있는 각도에 따라 값이 달라진다. 그림 14는 객체가 0°, 30°, 45°일 때의 영상을 나타낸 것으로 사진과 같이 각도에 따라 길이가 달라지는 것을 확인할 수 있다.



[그림 14] 회전에 따른 가로 세로 길이 변화

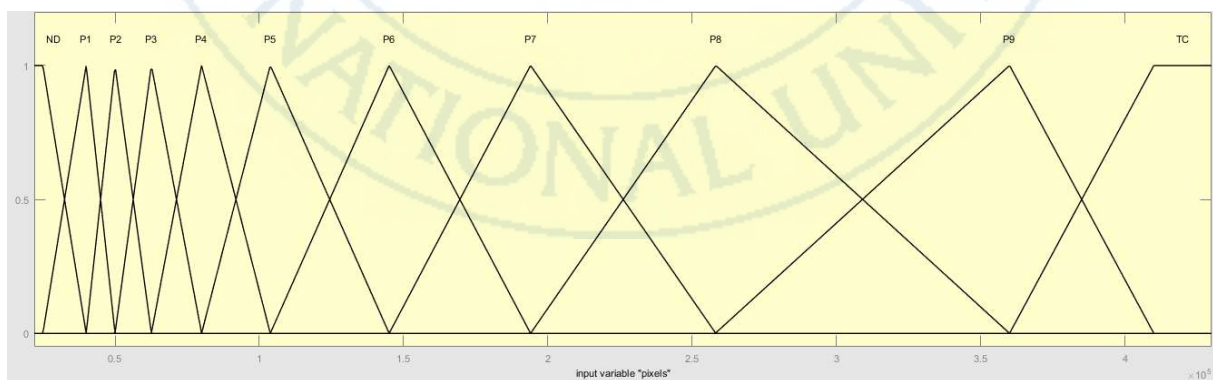


[그림 15] y/x 의 소속 함수

그림 17는 입력에 대한 소속 함수로써 영상에서 인식된 객체가 차지하는 픽셀 수를 의미한다. 영상에서 인식된 객체의 픽셀 수는 카메라와 객체 사이의 거리에 따라 값이 달라진다. 그림 16는 객체와 카메라 사이의 거리가 40cm, 80cm 일 때의 영상을 나타낸 것으로 카메라와 객체 사이의 거리가 달라짐에 따라 객체의 크기가 변화하는 것을 확인할 수 있고 이는 곧 객체가 차지하는 픽셀 수가 달라진다는 것을 의미한다.

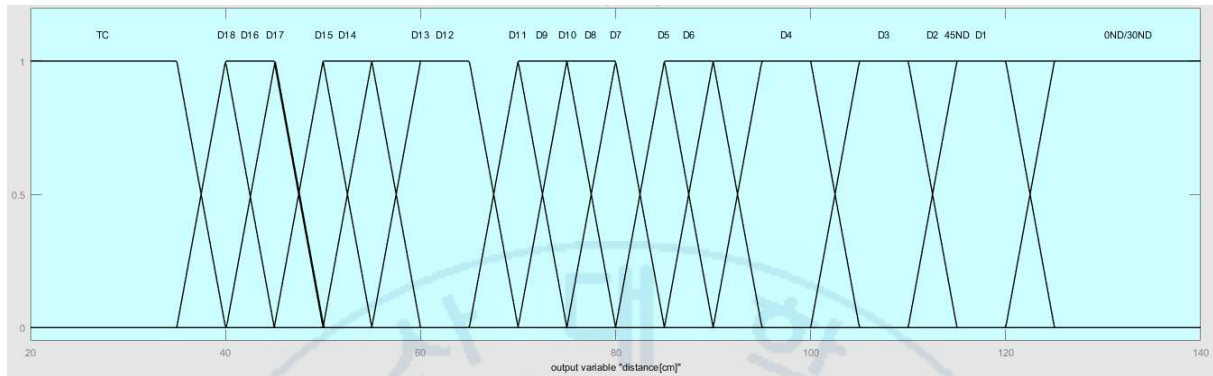


[그림 16] 거리에 따른 객체 크기 차이



[그림 17] 픽셀 수의 소속함수

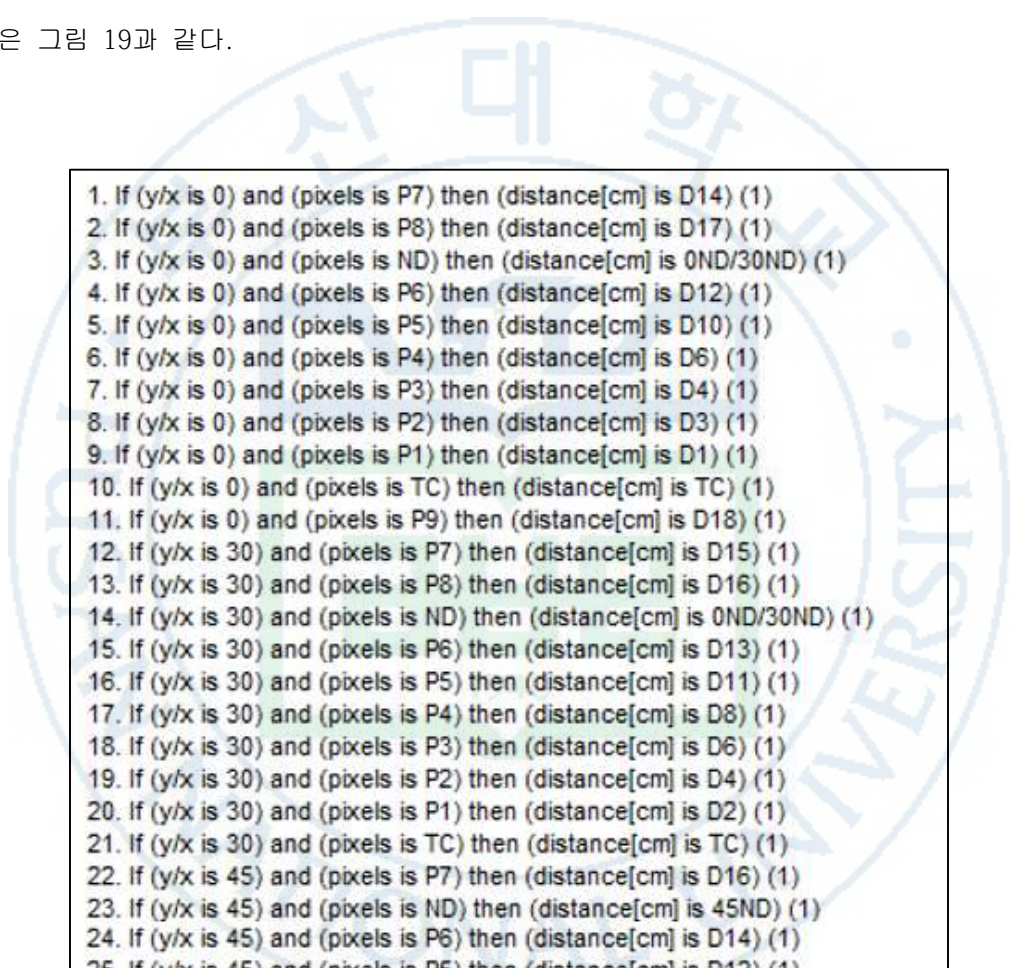
그림 18은 출력에 대한 소속 함수로써 두 입력으로 얻어진 거리를 의미한다.



[그림 18] 거리의 소속함수

4.3 퍼지 규칙

FIS에서 퍼지 규칙의 일반화 형태는 'If-Then 형식'의 언어적 규칙으로 표현된다. 보통 If문을 전제라고 하고 Then문을 결론이라고 한다. 그리고 If-Then문에 퍼지 집합을 도입한 것을 퍼지 규칙(Fuzzy Rule)이라고 표현한다. 본 논문에서는 두 개의 입력과 한 개의 출력을 가지며 시스템의 퍼지 규칙은 그림 19과 같다.

- 
1. If (y/x is 0) and (pixels is P7) then (distance[cm] is D14) (1)
 2. If (y/x is 0) and (pixels is P8) then (distance[cm] is D17) (1)
 3. If (y/x is 0) and (pixels is ND) then (distance[cm] is 0ND/30ND) (1)
 4. If (y/x is 0) and (pixels is P6) then (distance[cm] is D12) (1)
 5. If (y/x is 0) and (pixels is P5) then (distance[cm] is D10) (1)
 6. If (y/x is 0) and (pixels is P4) then (distance[cm] is D6) (1)
 7. If (y/x is 0) and (pixels is P3) then (distance[cm] is D4) (1)
 8. If (y/x is 0) and (pixels is P2) then (distance[cm] is D3) (1)
 9. If (y/x is 0) and (pixels is P1) then (distance[cm] is D1) (1)
 10. If (y/x is 0) and (pixels is TC) then (distance[cm] is TC) (1)
 11. If (y/x is 0) and (pixels is P9) then (distance[cm] is D18) (1)
 12. If (y/x is 30) and (pixels is P7) then (distance[cm] is D15) (1)
 13. If (y/x is 30) and (pixels is P8) then (distance[cm] is D16) (1)
 14. If (y/x is 30) and (pixels is ND) then (distance[cm] is 0ND/30ND) (1)
 15. If (y/x is 30) and (pixels is P6) then (distance[cm] is D13) (1)
 16. If (y/x is 30) and (pixels is P5) then (distance[cm] is D11) (1)
 17. If (y/x is 30) and (pixels is P4) then (distance[cm] is D8) (1)
 18. If (y/x is 30) and (pixels is P3) then (distance[cm] is D6) (1)
 19. If (y/x is 30) and (pixels is P2) then (distance[cm] is D4) (1)
 20. If (y/x is 30) and (pixels is P1) then (distance[cm] is D2) (1)
 21. If (y/x is 30) and (pixels is TC) then (distance[cm] is TC) (1)
 22. If (y/x is 45) and (pixels is P7) then (distance[cm] is D16) (1)
 23. If (y/x is 45) and (pixels is ND) then (distance[cm] is 45ND) (1)
 24. If (y/x is 45) and (pixels is P6) then (distance[cm] is D14) (1)
 25. If (y/x is 45) and (pixels is P5) then (distance[cm] is D12) (1)
 26. If (y/x is 45) and (pixels is P4) then (distance[cm] is D9) (1)
 27. If (y/x is 45) and (pixels is P3) then (distance[cm] is D7) (1)
 28. If (y/x is 45) and (pixels is P2) then (distance[cm] is D5) (1)
 29. If (y/x is 45) and (pixels is TC) then (distance[cm] is TC) (1)

[그림 19] 입출력 퍼지 규칙

입력은 퍼지화 과정을 통해 퍼지에서 정의되는 퍼지 규칙에 따라 표 1과 같이 퍼지 규칙 표로 나

타낼 수 있다. 여기서 ND는 Not Detect로 검출되지 않음을 나타내며 0, 30, 45는 각도를 나타낸다.

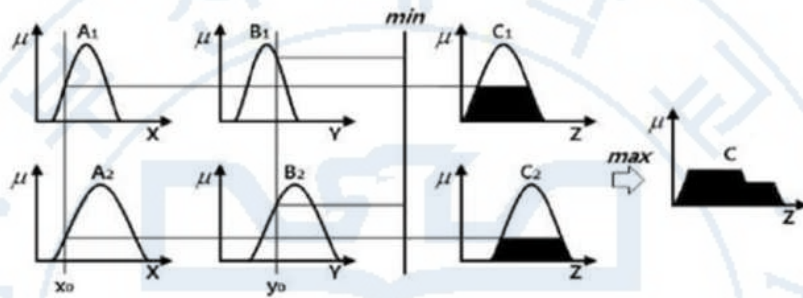
TC는 Too Close로 40cm 미만의 아주 가까운 거리를 나타낸다.

[표 1] 퍼지 규칙 표

	ND	0	30	45
ND	X	X	X	X
P1	X	D1	D2	X
P2	X	D3	D4	D5
P3	X	D4	D6	D7
P4	X	D6	D8	D9
P5	X	D10	D11	D12
P6	X	D12	D13	D14
P7	X	D14	D15	D16
P8	X	D17	D16	X
P9	X	D18	D16	X
TC	X	X	X	X

4.4 퍼지 추론 및 비퍼지화

퍼지 추론은 FIS에서 가장 중요한 요소로서, 실제 입력을 제어 규칙을 이용해서 퍼지 추론을 통해 계산한다. 제어 규칙을 이용해서 퍼지 추론을 통해 출력을 계산한다. 본 논문에서는 Mamdani 퍼지 추론을 적용하며, 그림 20은 Mamdani 퍼지 추론의 절차를 보여준다.



[그림 20] Mamdani 퍼지 추론

Mamdani 퍼지 추론을 이용하여 나온 최종 결과는 아래와 같이 최대-최소합성(max-min composition)에 의해서 구할 수 있다.

$$\alpha_1 = \mu_{A_1}(x_0) \text{ and } \alpha_2 = \mu_{A_2}(x_0)$$

(9)

$$\mu_c(z) = [\alpha_1 \wedge \mu_{c_1}(z)] \vee [\alpha_2 \wedge \mu_{c_2}(z)]$$

퍼지 추론한 결과를 사용하기 위해서는 물리적인 양으로 변환을 해야 한다. 본 논문에서는 비퍼지화를 하는 방법으로 아래와 같이 무게중심법을 사용한다.

$$z_{COA} = \frac{\int_z \mu_c(z) z dz}{\int_z \mu_c(z) dz} \quad (10)$$



5 실험 및 결과

5.1 실험 방법

본 실험의 목적은 지상에서 미리 획득 해놓은 데이터로 룩업테이블을 작성하고, 달 탐사 로버가 달 환경에서 촬영하여 전송한 영상을 룩업테이블에 적용시켜 로버의 위치를 파악하기 위함이다. 본 논문에서는 이를 모의적으로 실험하기 위해 두 환경에서 데이터를 획득하고, 첫 번째 환경에서 작성한 룩업 테이블에 두 번째 환경에서 획득한 데이터를 적용시켜 정확도를 비교한다.

첫 번째로 퍼지 룩업 테이블의 기준이 되는 데이터를 획득하기 위해 객체를 그림 21의 환경에서 40cm부터 120cm까지 5cm 간격으로 0°, 30°, 45° 각각의 영상을 촬영한다. 이후 각각의 영상을 알고리즘을 적용하여 인식된 객체의 가로, 세로 비 데이터와 픽셀 수 데이터를 획득한 후 퍼지 룩업 테이블을 작성한다. 이후 그림 22의 환경에서 새롭게 확보한 영상에 알고리즘을 적용한 후 얻어진 데이터를 이전에 작성한 퍼지 룩업 테이블에 입력 값으로 대입하여 상대적 위치 결과를 출력한다.



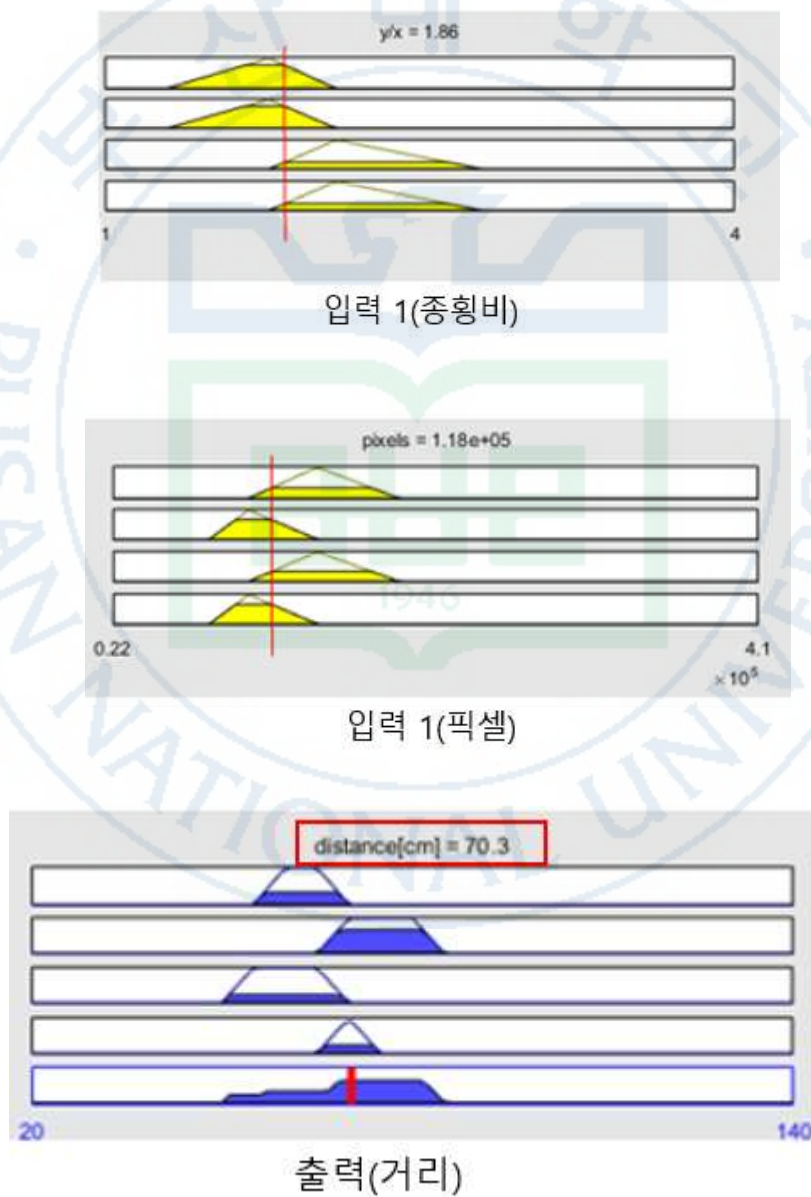
[그림 21] 퍼지 룩업 테이블 작성 시 실험 환경



[그림 22] 두 번째 실험 환경

5.2 실험 결과

그림 23은 70cm 거리 영상의 퍼지 추론 결과를 나타낸다. 70cm 거리에서 확보한 영상을 객체 인식 알고리즘을 통해 얻은 가로, 세로 비와 픽셀 수를 입력으로 주어 퍼지 추론 결과를 얻었더니 70.3cm가 출력으로 나왔다. 이와 같이 첫 번째 실험과 동일하게 5cm 간격 거리와 0°, 30°, 45° 각도 별로 실험을 진행하여 거리 결과를 출력하여 평균 오차와 최대 오차를 계산하였다.



[그림 23] 0°, 70cm 거리 영상의 퍼지 추론 결과

[표 2] 실험 결과

각도 \ 오차[cm]	평균 오차	최대 오차
0°	1.38cm	3.3cm
30°	2.01cm	3.8cm
45°	1.3cm	3.3cm

표 2는 21cm X 29cm 크기의 객체를 이용하여 각 각도 별로 촬영한 영상을 알고리즘에 적용시켰을 때의 오차를 나타낸다. 거리 별로 오차가 다르게 측정되었으나 모두 오차가 4cm 이내로 측정되는 것을 확인할 수 있었다. 또한 최대 인식 가능 거리는 0°일 때 120cm, 45°일 때 85cm였다. 이 알고리즘은 영상에서의 객체 인식을 기반으로 하는 위치 인식 알고리즘 방식이기 때문에 인식 가능 거리에 한계가 있다는 단점이 있으나 추후 착륙선이라는 큰 객체를 사용하게 될 경우 인식 가능 거리가 늘어날 것이다. 거리 별 픽셀 수를 비교해 보았을 때 거리가 k 배 늘어날 경우 픽셀 수는 k^2 배 감소하는 관계를 가지고 있다. 착륙선의 크기가 1.5m X 2.5m ~ 1.5m X 3m 일 경우 면적이 대략 61.6배 ~ 73.9배 차이가 난다. 그러므로 실제 착륙선을 가지고 실험할 경우 대략 0°일 때 9.4m ~ 10.3m, 45°일 때 6.7m ~ 7.3m 정도의 거리를 인식 가능할 것으로 예상된다.

6 결론

본 논문은 저 성능 CPU를 가진 탐사 로버에 적합한 원격 탐사 위치 인식 방식을 제안한다. 저 성능 탐사 로버에 적합한 위치 인식 방식은 비 자율주행 방식으로 탐사 로버가 촬영한 영상을 지상에서 처리하는 원격 탐사 방식이며, 지상에서 영상 처리 시에 근거리일 때와 원거리일 때 인식하는 부분을 다르게 설정하는 알고리즘이다. 인식하는 부분을 다르게 설정하는 이유는 근거리일 때는 로버가 착륙선 전체를 촬영할 수 없으므로 착륙선의 특정 부분만을 인식하도록 한다. 또한 본 논문에서 제안하는 퍼지-픽셀 기반 위치 인식 알고리즘은 저 성능 CPU에 적합하게 한 영상만으로도 상대적인 위치 추정이 가능한 알고리즘이다. 이때 한 영상을 이용해 상대적 위치 추정을 하기 위해서는 비교를 위한 지상에서의 데이터가 필요하다. 본 논문에서는 21cm X 29cm 크기의 객체를 이용하여 지상에서 40cm ~ 120cm까지 5cm 간격으로 0°, 30°, 45°의 각도 별로 영상을 획득한다. 획득된 영상들을 해상도 향상 알고리즘, 객체 인식 알고리즘을 적용시켜 인식된 객체의 가로, 세로 길이 비와 픽셀 수를 추출하여 퍼지 룩업 테이블을 작성한다. 이후 다른 환경에서 동일하게 영상을 확보하고 알고리즘을 적용시켜 가로, 세로 길이 비와 픽셀 수를 추출하고, 이를 룩업 테이블 기반의 퍼지 추론에 적용하여 거리 결과를 출력한다. 이 때 사용하는 해상도 향상 알고리즘은 달 탐사 로버에 부착되는 CPU의 저성능으로 인해 저해상도의 영상밖에 얻을 수 없어 객체 인식을 향상을 위해 필수적인 부분이다. 본 논문에서는 해상도 향상을 위한 알고리즘으로 B-Spline 보간법을 사용한다. 향상된 해상도의 영상에 적용되는 객체인식 알고리즘은 로버의 이동으로 인한 카메라의 시점 변화에도 강인한 ASIFT 알고리즘을 사용하며 ORSA 필터와 2차 분포 필터를 통해 이상치를 제거하여 올바른 객체의 영역을 추출한다. 이러한 실험 과정을 거쳐 위치인식 정확도를 확인해 본 결과 최대 오차가 4cm 미만의 거리 출력 결과를 얻을 수 있었다. 본 논문에서 제안한 알고리즘은 지상에서 데이터를 획득하여 퍼지 룩업 테이블을 작성해 둘 경우 실제 달 환경에서는 한 개의 영상으로도 로버의 착륙선에 대한 상대적인 위치를 추정할 수 있다는 장점을 가지고 있다. 그러므로 연속적이고 많은 영상 촬영이 불가능한 저성능 달 탐사 로버의 경우 이 알고리즘은 매우 유용하게 사용될 것이며, 위치 인식 가능 범위는 착륙선의 크기를 고려해 볼 때 0°

일 때 9.4m ~ 10.3m, 45°일 때 6.7m ~ 7.3m 정도의 거리를 인식 가능할 것으로 예상된다. 이는 동일한 원격 탐사 방식을 사용하는 과거의 화성 탐사 로버 Sojourner의 이동 반경이 착륙선으로부터 12m 이내였다는 것을 감안하면 단일 센서로 얻을 수 있는 적합한 인식 가능 범위로 생각 된다. 또한 본 논문에서는 한 방향의 영상 데이터를 가지고 실험을 진행하였지만 실제 적용 시에는 30° 간격으로 12방면의 영상 데이터를 가지고 진행한다면 어떤 방면에서 로버가 착륙선을 촬영하더라도 10m 정도의 거리 측정이 가능할 것으로 예상된다.

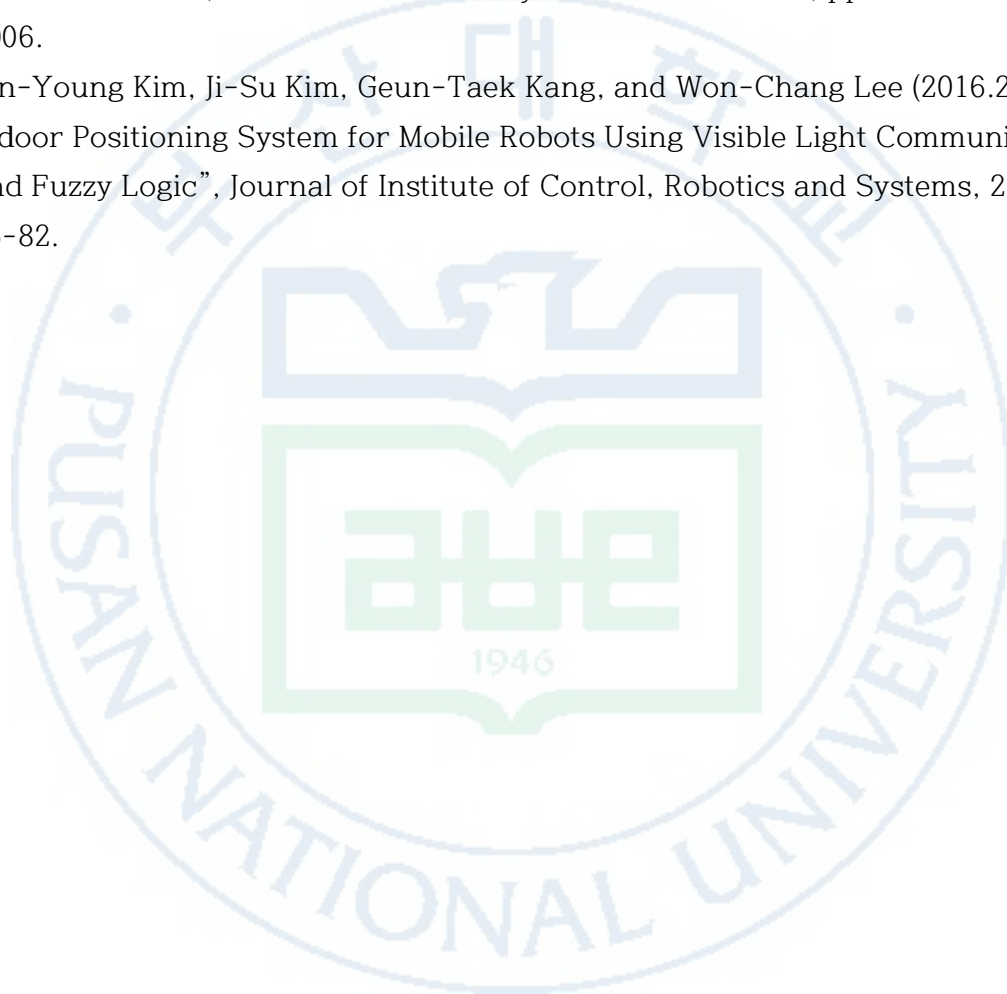


참고 문헌

- [1] S.-M. An and Y.-G. Lee, "Trend of Rover Development and Mission Analysis of Lunar Exploration Rovers," Korean Society for Aeronautical & Space Sciences(KSAS) Conference 2008, pp 1301-1304, November 2008.
- [2] W.-S. Eom, Y.-G. Kim, J.-H. Lee, K.-H. Choi and E.-S. Shim. (2012). "Development Trend of Unmanned Planetary Exploration Rover by NASA", Aerospace Industry Technology Trend, 10(2), 102-111.
- [3] M. Maimone, 2007, Two Years of Visual Odometry on the Mars Exploration Rovers, Jour. Field Robotics: Special Issue on Space Robotics 24 (3) : 169 ~ 186
- [4] K. Konolige, M. Agrawal and J. Sola, "Large-Scale Visual Odometry for Rough Terrain," International Symposium on Research in Robotics, 2007.
- [5] Y.-H. Joo and H.-J. Yang, "3D Image Process by Template Matching and B-Spline Interpolations", Korean Institute of Intelligent Systems, vol 19,no. 5, pp. 683-688, 2009
- [6] M. Unser, "Splines—A perfect fit for signal and image processing," IEEE Signal Processing Mag., vol. 16, pp. 22-38, June 1999.
- [7] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", Int. Journal of computer Vision, 60(2), 91-110, 2004
- [8] K. Mikolajczyk and C. Schmid, "Indexing based on scale invariant interest points", Proc. of the 8th Int Conf. on Computer Vision, pp. 525-531, 2001
- [9] Guoshen Yu, and Jean-Michel Morel, "ASIFT: An Algorithm for Fully Affine Invariant Comparison", Image Processing On Line, 1, pp. 11-38, 2011.
- [10] Y.-M. Kim, C.-W. Park, K.-T. Park and Y.-S. Moon, "Marker Detection by Using Affine-SIFT Matching Points for Marker Occlusion of Augmented Reality", The Institute of Electronics and Information Engineers-CI, 48(2), 55-65.
- [11] Lionel Moisan, Pierre Moulon, and Pascal Monasse, Automatic Homographic Registration of a Pair of Images, with A Contrario Elimination of Outliers, Image

Processing On Line, 2 (2012), pp. 56-73.

- [12] O.Chum, J.Matas and J.Kittler, "Locally Optimized RANSAC," In Proceedings of deutsche Arbeitsgemeinschaft für Mustererkennung, Germany, vol.2781, pp.236-243, 2003.
- [13] J. Jantzen, "Foundations of Fuzzy Control", John Wiley & Sons, 2007.
- [14] M. Trabia, L. Z. Shi, and N. E. Hodge, "A fuzzy logic controller for autonomous wheeled vehicles," Advanced Robotic Systems International, pp. 175-200. Dec. 2006.
- [15] Jun-Young Kim, Ji-Su Kim, Geun-Taek Kang, and Won-Chang Lee (2016.2). "An Indoor Positioning System for Mobile Robots Using Visible Light Communication and Fuzzy Logic", Journal of Institute of Control, Robotics and Systems, 22(2), 75-82.



Estimation of Relative Position of Fuzzy-Pixel Based Mono Camera Exploration Rover

Na-hyun Lee

Department of Electrical and Computer Engineering
The Graduate School Pusan National University

Abstract

In the case of space exploration rover, a low-performance CPU is used because of stability. A high-performance CPU satisfying stability is very expensive. In this paper, we propose a remote sensing localization method suitable for rover with a low performance CPU. Instead of using a continuous and large amount of image capturing visual odometry method, we propose a fuzzy - pixel based relative position estimation algorithm that estimates the position of a pixel. Affine SIFT algorithm robust to camera axis slope change is used for object recognition, and ORSA and quadratic distribution filter using feature point distribution are used to remove outliers. After the object is recognized, the number of pixels of the recognized object and the horizontal and vertical length ratio are applied to the previously created fuzzy lookup table to extract the distance and estimate the relative position. And the proposed algorithm verifies the accuracy of various situations such as when the rover is located on the front side and when it is located on the side.