영상 매칭 및 자세 추정을 이용한 무인 차량의 위치 추정
UGV Localization based on Scene Matching and Pose Estimation

복윤수 황영배 권인소
Yunsu Bok Youngbae Hwang In So Kweon
한국과학기술원 전자전산학과 전기 및 전자공학 전공, 대전 305-701
(발표자 연락처 : ysbok@rcv.kaist.ac.kr)

ABSTRACT Autonomous localization is very important for unmanned ground vehicles (UGVs). In outdoor environment, a GPS signal is available for localization. But there are many regions where the GPS signal is unstable. A vehicle cannot utilize the GPS signal if the signal is jammed or the device is out of work. We propose a method for localizing the vehicle based on scene matching and pose estimation to support unstable GPS signal. Our goal is localizing vehicle quickly and accurately using pre-built image database of the places where the vehicle is expected to pass. We perform two steps for this work. The first step is searching the DB image closest to a current image. Features invariant to scale difference and rotation such as SIFT and SURF are widely used for scene matching in robotics community. In this paper, we compare the performance of the features in scene matching. We use the tree structure based searching algorithm to make the searching process faster. The second one is localizing the vehicle using the current image and matched DB images. Among the methods estimating relative pose between two images, we use homography based method and perspective 3-point algorithm based method. We perform experiments using outdoor images, and estimate the location of the vehicle. The results show that the proposed method works well in outdoor environments.

Keyword : Localization, Scene matching

1. 서 론

무인 차량(Unmanned Ground Vehicle, UGV)이 주어진 역할을 수행하기 위해서 필요한 필수 기능 중의 하나는 자신의 위치를 추정하는 것이다. 무인 차량의 위치를 추정하기 위하여 INS(Inertial Navigation System), 레이저 거리 센서(Laser Range Finder, LRF), CCD 카메라 등 여러 가지 센서가 사용되고 있으며, 그 중에서도 실외 환경에서 가장 사용하기 용이한 것으로는 GPS(Global Positioning System)가 있다. GPS 는 위성을 이용하여 지구상에서의 현재 위치를 알려주기 때문에 절대적인 위치 추정이 가능하고, 따라서 상대적인 위치 추정의 문제점인 오차 누적(error accumulation)의 문제가 거의 없다는 것이 장점이다. 하지만 GPS 는 위성 신호를 수신해야 하기 때문에 건물 부근이나 숲 속 등의 환경에서는 GPS 신호가 상당히 불안정하다. 또한 전파방해를 받거나 GPS 장치가 고장난 경우에도 데이터를 얻을 수 없기 때문에 GPS 에만 의존하여 위치 추정을 할 경우 문제가 발생한다.

본 논문에서는 실외에서의 GPS 신호가 불안정한 경우를 대비하여 다른 센서를 이용하여 위치 추정을 수행한다. 앞에서 나열한 여러 가지 센서들 중 가격이 저렴하면서도 3 차원 위치 추정이 가능한 CCD 카메라를 이용한다. 무인 차량이 지나갈 장소에 대한 영상 데이터베이스(DB)를 구축해 놓으면, 실제 주행 시에 획득하는 영상과 DB 와의 매칭을 통하여 대강의 위치를 알아내고, 영상 사이의 상대적인 자세 추정을 통하여 차량의 위치를 추정할 수 있기 때문이다.

영상을 이용하여 차량의 위치 추정을 수행하기 위해 앞서 설명한 바와 같이 전체 과정을 두 단계로 나눈다. 첫 번째 단계는 현재 영상과 가장 가까운 데이터베이스 영상을 검색하는 것이다. 영상 매칭은 최근에 세계 여러 곳에서 연구하고 있으며, ICCV(International Conference on Computer Vision) 2005 에서는 영상 매칭을 이용한 위치 추정을 contest 의 주제로 선정하기도 했다. 로보틱스 및 비전 분야에서 영상 매칭에 주로 사용되는 특징점은 SIFT(Scale Invariant Feature Transform)[1]이다. 앞에서 언급한 ICCV contest 에서 1 위를 차지한 팀 역시 SIFT 를 사용하였다[2]. 하지만 SIFT 는 계산량이 많아서 추출 속도가 실시간과는 조금 거리가 있다는 단점이 있다. 최근에는 SIFT 와 거의 비슷한 성능을 보이면서 추출 속도가 빠른 SURF(Speeded Up Robust Feature)[3]가 제안되었다. 본 논문에서는 두 가지의 특징점을 모두 사용하여 실외 환경에서의 영상 사이의 매칭을 수행하고 성능을 비교한다.

두 번째 단계는 매칭된 데이터베이스 영상과 현재 영상을 이용하여 차량의 현재 위치를 추정하는 것이다. 영상 사이의 위치 관계를 계산하는 방법들 중에서, 본 논문에서는 호모그래피(homography)를 이용하는 방법과 P3P 알고리즘(Perspective 3-Point Algorithm)을 이용하는 방법을 사용한다. 각각의 방법을 사용해서 위치 추정 결과를 얻고 이를 비교한다.

본 논문은 다음과 같이 구성되어 있다. 2 장에서는 첫 번째 단계인 DB 와의 영상 매칭을 위해 사용한 특징점 및 매칭 방법론에 대하여 설명하고, 영상 매칭에서의

SIFT 와 SURF 의 성능을 분석한다. 3 장에서는 두 번째 단계인 영상 간의 상대적인 위치 관계 추정 방법을 소개하고, 실제 영상을 이용한 결과의 예를 보인다. 4 장에서는 실외 환경에서의 실험 결과를 표시한다. 5 장에서는 결론을 제시한다.

2. 영상 매칭 (Scene Matching)

영상 매칭은 앞에서 언급한 대로 현재 영상과 가장 가까운 DB 영상을 검색하는 과정이다. 두 영상 사이의 상대적인 위치 관계를 알아내기 위해서는 공통 부분이 존재해야 하기 때문이다.

2.1 특징점 추출

비교할 영상이 연속된 영상 시퀀스의 두 영상처럼 변화가 작다면 SAD(Sum of Absolute Difference), NCC(Normalized Cross Correlation) 등의 윈도우 기반 매칭 방법으로도 충분하지만 실제로는 그렇지 않은 경우가 대부분이다. 윈도우 기반 매칭의 가장 큰 단점은 영상이 스케일 변화, 회전 등이 있을 때 매칭이 잘 되지 않는다는 것이다. 따라서 영상 매칭에는 스케일, 회전 등에 불변인 특징점들을 추출해서 특징점들의 기술자(descriptor)를 비교하는 방법이 주로 이용된다.

비전 분야에서 가장 많이 이용되는 불변 특징량으로는 SIFT(Scale Invariant Feature Transform)[1]가 있다. 또한 최근에 제안된 새로운 불변 특징량으로는 SURF (Speeded Up Robust Feature)[2]가 있다.

두 가지 특징량의 공통점은 스케일 변화 및 회전에 불변이라는 것이다. 그 이유는 특징점을 추출하는 방법에서 찾을 수 있다. 특징점을 추출하기 위하여 우선 스케일 공간(scale space)를 만든다. 스케일 공간은 영상의 크기를 여러 가지로 조정하여 놓은 것으로, 크게 획득한 영상과 작게 획득한 영상을 모두 고려 대상으로 하기 때문에 스케일 변화에 불변인 특징량을 얻을 수 있다. 스케일이 작아진 영상을 얻는 방법은 가우시안 커널(Gaussian kernel)을 이용하는 것이다. 영상과의 convolution 을 수행할 가우시안 커널의 분산(variance)을 크게 할수록 작은 영상을 만드는 효과를 얻을 수 있다.

SIFT 의 경우에는 분산이 어느 정도 커지면 원본 영상의 크기를 줄이고 다시 가우시안 커널과의 convolution 을 수행한다. 다음에는 서로 이웃한 영상 간의 차이(Difference of Gaussian, DoG)를 계산한다. Scale space 에서의 DoG 의 극점(local extrema)이 특징점으로 선택된다. 이렇게 선택된 점들은 스케일 변화에 불변인 특성을 갖게 된다. 위치가 결정된 특징점들에게 회전에 불변인 특성을 부여하기 위하여 특징점에서의 gradient 의 방향을 계산한다. SIFT 의 descriptor 는 특징점 주변 영역에서의 orientation histogram 이다.

그림 1 은 분산이 서로 다른 가우시안 커널과의 convolution 으로 생성한 스케일 공간 영상의 예이며, 그림 2 는 그림 1 영상들의 차이값이다.

그림 1. 스케일 공간(scale space)의 예
(가우시안의 분산을 변화시키면서 얻은 영상)

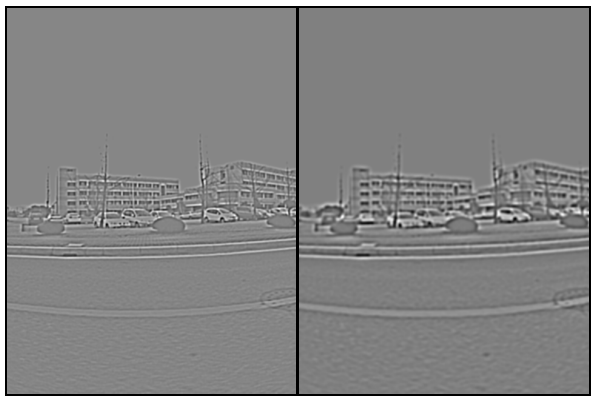

그림 2. DoG (그림 1 의 영상들의 차이)

SURF 는 DoG 를 이용하는 것이 아닌 Laplacian 을 이용한다. 분산을 변화시키면서 Laplacian 을 계산하고 극점을 찾는 것인데, SURF 는 SIFT 의 단점인 계산 시간이 오래 걸린다는 것을 극복하기 위하여 추출 과정을 단순화시켰다. Laplacian 의 계산을 정수로 양자화하여 box filtering 을 적용한 것이다. 스케일 변화에 강인한 특성을 얻기 위해서는 Laplacian 의 분산을 변화시키면서 convolution 을 한 영상을 얻어야 하는데, 이를 근사화하기 위하여 box 의 크기를 일정 간격으로 조정한다. 그 결과 추출 속도는 SIFT 보다 빠르다. SURF 의 회전 불변

특성을 위한 orientation 은 SIFT 와 비슷하지만 약간 다른 방법인 Haar wavelet 을 사용하며, descriptor 로는 주변 영역을 분할하고 샘플링하여 샘플링된 픽셀값들의 분포를 표현하는 몇 개의 숫자들을 추출한다.

640×480 영상에서 특징점을 추출하는 시간을 측정한 결과는 표 1 과 같다. 사용한 PC 의 CPU 속도는 Intel 2.40GHz, RAM 은 3GB 이다. SURF 가 SIFT 보다 약 4 배 정도 빠른 것을 확인할 수 있다.

표 1. 특징점 추출 시 소요 시간

SURF	104.4 ms
SIFT	420.42ms

2.2 데이터베이스 영상 검색

데이터베이스의 모든 영상으로부터 특징점을 추출한 다음, 현재 영상으로부터 마찬가지로 특징점을 추출하고 이를 데이터베이스의 모든 특징점과 비교한다. 가장 가까운 특징점이 추출된 DB 영상에 투표하여 득표수가 가장 많은 영상이 현재 영상과 가장 가까운 DB 영상이다.

두 특징점의 비슷한 정도를 결정하는 데는 기술자(descriptor)를 이용한다. 현재 영상에서 추출한 i 번째 특징점의 기술자를 d_i, DB 의 j 번째 영상에서 추출한 k 번째 특징점의 descriptor 를 d_{jk} 라 하면, 가장 가까운 특징점 d_{mn} 을 나타내는 index m, n 은 다음과 같이 결정한다.

$$(m,n) = \arg\min_{(j,k)}\left(\left\|d_{jk} - d_i\right\|\right) \qquad (1)$$

현재 영상의 모든 특징점에 대하여 결정한 (m,n)의 집합을 M 이라 하면 가장 가까운 DB 영상은 식 (2)와 같이 찾는다.

$$(\text{closest DB}) = \max_j\left(\sum_k a_{jk}\right) \qquad (2)$$

$$a_{jk} = \begin{cases} 1 & (j,k) \in M \\ 0 & (j,k) \notin M \end{cases}$$

영상 매칭의 정확도를 알아보기 위하여 몇 가지 가정을 하였다. 차량에서 너무 멀리 있는 점들은 위치를 추정하는 데 도움이 되지 않고 너무 가까이 있는 점들은 대응점을 거의 찾을 수 없기 때문에 적당한 거리에 있는 점들을 사용해야 하며, 이 거리를 10m 로 설정하였다. 카메라의 렌즈는 2.8mm 를 사용하였으며, 카메라를 옆으로 돌린 상태에서 전진하였기 때문에 시야각은 세로 방향 시야각인 약 65°이다. 따라서 10m 거리의 점들이 보이는 구간의 길이는 2·tan(65/2)·10=12.74(m)이다.

4 장에서 설명하겠지만 데이터베이스 영상 1 장당 평균 이동 거리는 약 3.58m 이다. 따라서 index 가 1 만큼 차이나는 두 장의 데이터베이스 영상에서 겹치는 부분은 (12.74-3.58)/12.74=72(%)이다. 차이가 2 일 경우에는 겹치는 부분이 약 44%가 된다. 본 논문에서는 영상이 60% 이상 겹쳐야 제대로 된 매칭이라고 가정한다. 따라서 매칭되어야 하는 영상과 실제로 매칭된 영상의 index 의 차이가 1 이하일 때 제대로 된 매칭이라고 할 수 있다.

실험용 영상과 매칭되어야 하는 영상과 실제로 매칭된 영상의 index 차이를 수집하였다. 그 결과는 표 2 와 같다. Index 의 차이가 1 이하인 영상의 비율은 SURF 의 경우 89.6%, SIFT 의 경우 92.9%이다. SIFT 가 SURF 보다 조금 더 좋은 성능을 보이기는 하지만 크게 차이가 나지는 않는다.

표 2. 매칭 정확도

차이	SURF	SIFT
0	151	149
1	100	111
2	12	9
3	2	1
4	2	0
>5	13	10

본 논문에서는 실험용 영상으로부터 얻은 특징점들과 가장 가까운 특징점을 데이터베이스의 모든 특징점 중에서 전체 검색(full search)으로 찾기 때문에 시간이 오래 걸린다. 계산 시간을 단축하기 위하여 KD tree 를 사용한다[5]. 현재 영상의 특징점 1 개당 평균 검색 시간을 KD tree 를 사용하기 전과 후로 나누어 표 3 에 기록하였다. SIFT 가 SURF 보다 특징점의 위치를 계산하는 시간 및 기술자를 계산하는 시간이 모두 오래 걸리기 때문에 같은 영상에서 더 많은 시간을 필요로 한다.

표 3. DB 에서의 특징점 검색 시 소요 시간

Feature	SURF	SIFT
Number of Features	59372	63009
Dimension	64	128
Time (Normal)	8.043ms	17.516ms
Time (KD tree)	4.199ms	11.771ms

비교해야 할 특징점의 수가 워낙 많기 때문에 시간이 오래 걸린다. 이를 해결하기 위하여 검색할 데이터베이스의 범위를 좁히는 방법을 선택한다. 차량의 위치를 INS, odometer 등의 다른 센서들을 이용하여 어느 정도

의 오차 범위 내에서 알 수 있다고 가정한다.

실험용 영상과 매칭되어야 할 데이터베이스 영상으로부터 앞뒤로 6 장의 영상을 추가하여 총 13 장의 DB 영상 중 가장 가까운 한 장을 선택하도록 한다. 이번에도 마찬가지로 매칭되어야 하는 영상과 실제로 매칭된 영상의 index 차이를 조사하였다. 그 결과는 표 4 와 같다.

표 4. DB 중 일부만을 사용했을 때의 매칭 정확도

차이	SURF	SIFT
0	137	151
1	97	98
2	17	11
3	8	0
4	6	5
>5	15	15

차이가 1 이하인 영상의 비율이 SURF 의 경우 83.6%, SIFT 의 경우 88.9%이다. 정확도가 약간 떨어진 것으로도 해석할 수 있는데, 그 이유는 매칭될 DB 영상의 수가 많은 경우 잘못된 매칭으로 인한 투표가 여러 곳으로 분산되어 큰 영향을 주지 않는 반면, DB 영상의 수가 적을 경우에는 잘못된 한두 장의 영상으로 몰릴 가능성이 있기 때문이다. 그림 3 은 잘못된 투표 결과의 예이다.

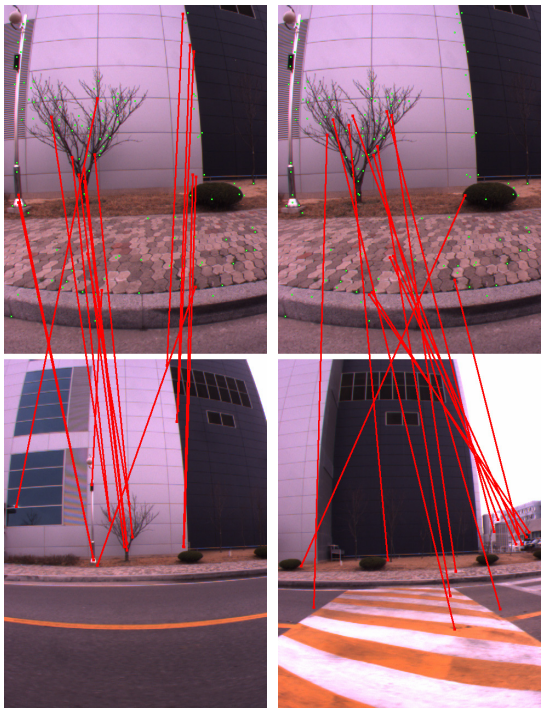

그림 3. 잘못된 투표 결과의 예
(좌) 매칭되어야 하는 영상 (우) 실제로 매칭된 영상

매칭에 소요된 시간을 측정한 결과는 표 5 와 같다. 검색해야 하는 DB 특징점의 수가 상당히 줄어들었기 때문에 검색 시간이 단축된 것을 확인할 수 있다.

표 5. 일부만을 사용했을 때의 매칭 소요 시간

Feature	SURF	SIFT
Number of Features	2831	3006
Dimension	64	128
Time (KD tree)	0.209ms	0.460ms

3. 데이터베이스 영상 검색

영상 사이의 상대적인 위치 관계를 계산하는 방법은 여러 가지가 제안되어 왔다. 본 논문에서는 그 중에서도 기본이라고 할 수 있는 호모그래피(homography)와 P3P 알고리즘을 이용한다.

3.1 호모그래피(Homography) 기반의 위치 추정

렌즈에 의해 발생하는 원형 왜곡(radial distortion)을 고려하지 않을 때, 일반적으로 두 영상 간의 관계는 하나의 행렬로 표현할 수 있다. 실외 환경에서는 대응점들이 건물 벽 등과 같은 하나의 평면 상에 여러 개가 존재하거나, 대응점들의 카메라가 보는 방향으로의 거리 차이가 카메라와의 거리 차이보다 작아서 하나의 평면 상에 존재하는 것으로 근사화할 수 있다. 3 차원 상의 하나의 점 Q가 두 영상으로 투영된 점을 각각 $\hat{q}$, $\hat{q}'$ 이라 하면 투영된 위치 사이의 관계는 다음과 같다.

$$\lambda q' = Hq \tag{3}$$

행렬 H 를 호모그래피(homography)라고 하며, 두 영상 사이의 위치 관계 $[R \quad t]$와의 관계는 다음과 같다. N 은 대응점들이 속해 있는 평면과 수직인 벡터(normal vector)이며, d 는 카메라와 평면 사이의 거리이다. (그림 4 참조)

$$H = R + \frac{1}{d} t N^T \tag{4}$$

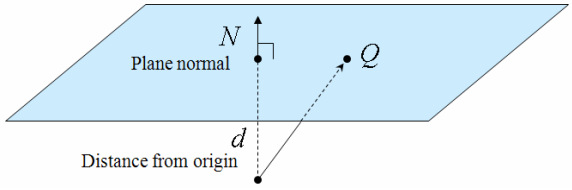

그림 4. 호모그래피(Homography)를 나타내는 평면

영상 사이의 대응점은 SURF 와 SIFT 를 각각 사용하여 2.2 절과 같은 방법(식 1, 2)으로 매칭한다. H 를 계산하는 방법론은 다음과 같다. 4 쌍의 대응점으로부터

아래의 식을 계산하면 SVD(Singular Value Decomposition)를 이용하여 해를 구할 수 있다.

$$\lambda q' = Hq \tag{5}$$

$$\begin{bmatrix} q^T & 0 & -q'_x q^T \\ 0 & q^T & -q'_y \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} = 0 \tag{6}$$

$$\left(H = \begin{bmatrix} h_1 & h_2 & h_3 \end{bmatrix}^T\right)$$

이 방법은 단일 카메라 영상에 기반한 방법이기 때문에 위 식으로부터 호모그래피 H 를 계산하고 $[R \quad t]$ 를 추정한다고 해도 병진 이동 t 의 스케일은 알 수 없다. t 의 스케일을 알아내기 위해서는 2 장 이상의 DB 영상이 필요하다. 따라서 본 논문에서는 매칭된 DB 영상 외에도 앞뒤로 2 장씩의 DB 영상을 추가하여 총 5 장의 영상을 사용한다.

H 를 계산하고, 이로부터 $[R \quad t]$ 를 계산하면, 현재 영상은 매칭된 DB 영상을 지나고 t 와 평행한 직선 상에 있게 된다. 상대적인 위치를 알고 있는 5 장의 DB 영상으로부터 이러한 직선을 얻게 되면 그 교점이 현재 영상의 위치가 되는 것이다.

DB 영상의 위치를 p, 현재 영상의 위치를 q, t 의 방향을 v 라 하면 다음과 같은 식이 성립한다.

$$v \times (p - q) = [v]_\times (p - q) = 0 \tag{7}$$

$$\begin{bmatrix} [v]_\times & -[v]_\times p \end{bmatrix} \begin{bmatrix} q \\ 1 \end{bmatrix} = 0 \tag{8}$$

위의 식은 SVD 를 이용하면 해를 구할 수 있다. 이 방법을 이용하여 위치를 추정한 예는 그림 5 와 같다. 점들은 DB 영상의 위치, 직선은 $[R \quad t]$ 로부터 얻은 현재 영상의 방향, 원은 추정한 현재 위치이다.

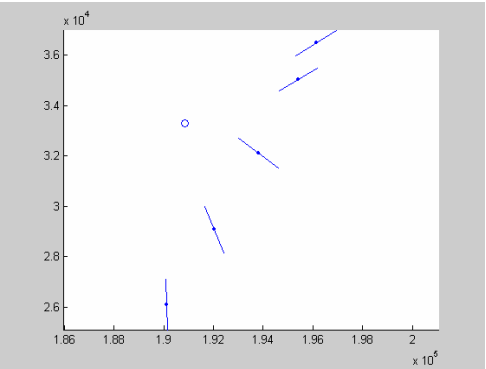

그림 5. 위치 추정 결과

3.2 P3P 알고리즘 (Perspective 3-Point Algorithm)

월드 좌표계(world coordinate) 상에서의 좌표를 알고 있는 3 차원 점들이 영상에 투영된 위치를 알면 월드 좌표계 상에서의 영상의 위치를 알아낼 수 있다. 대표적인 예로는 P3P 알고리즘이 있다. P3P 알고리즘은 그림 6 에서 3 개의 점 사이의 거리 a, b, c, 그리고 이 점들이 투영된 위치로부터 계산한 카메라 광선(camera ray) 사이의 각도 α, β, γ 를 알고 있을 때, 카메라 좌표계의 중심으로부터 각 점들까지의 거리 s_1, s_2, s_3 를 구하는 알고리즘이다.

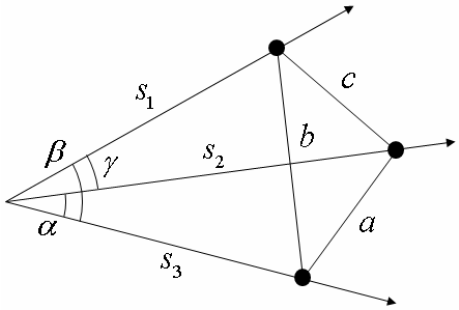

그림 6. P3P 알고리즘(3-Point Algorithm)

그림 6 에 코사인 제 2 법칙을 적용하면 3 개의 식을 얻을 수 있다.

$$s_2^{\,2} + s_3^{\,2} - 2s_2 s_3 \cos\alpha = a^2 \tag{9}$$

$$s_1^{\,2} + s_3^{\,2} - 2s_1 s_3 \cos\beta = b^2 \tag{10}$$

$$s_1^{\,2} + s_2^{\,2} - 2s_1 s_2 \cos\gamma = c^2 \tag{11}$$

이 연립방정식의 해를 계산하는 방법은 여러 가지가 제안되었다[6]. 본 논문에서는 비교적 단순하면서도 해가 정확한 Grunert 의 풀이 방법을 사용하였다.

4. 실험 결과

앞에서 설명한 시스템의 성능을 테스트하기 위하여 실외 영상 시퀀스를 사용하였다. 카메라는 Point Gray Research 사의 Flea 를, 렌즈는 Avenir 사의 2.8mm 렌즈를 사용하였다. 데이터베이스 및 실험용 영상의 크기는 640×480 이며, 데이터베이스는 약 700m 길이의 구간으로부터 2200 장을 획득하여 10 장 간격으로 샘플링하였다. 실험용 영상은 같은 장소에서 DB 영상의 이동 경로보다 약 2~3m 정도 안쪽으로 이동한 경로로부터 2800 장을 획득하여 역시 10 장 간격으로 샘플링하였다. 그림 7 은 실험용 영상 및 이에 대응하는 DB 영상의 예이다.

데이터베이스 영상의 경우 정확한 위치 정보를 얻어야 하기 때문에 어느 정도의 오차를 항상 가지고 있는 GPS 대신 오차가 누적되기는 하지만 짧은 구간에서는

상당히 정확한 상대적 위치 추정 방법을 사용하였다. 데이터베이스 영상의 위치를 계산하기 위하여 사용한 방법은 카메라-레이저 융합 센서[4]이다. 이 센서를 이용하여 데이터베이스 영상의 위치를 추정한 결과는 그림 8 과 같다. 샘플링하지 않은 데이터베이스 영상 시퀀스 전체의 병진이동 거리는 평균 357.66mm, 표준편차 87.18mm 이다. 그림 8 은 데이터베이스 영상들의 위치를 점으로 표시한 그래프이다.

그림 7. 실험용 영상 및 DB 영상의 예

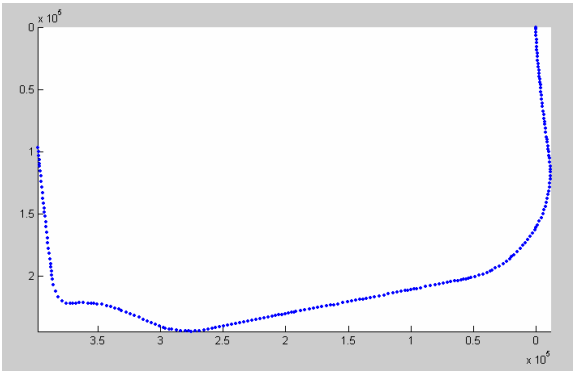

그림 8. DB 영상의 위치

그림 9~12 는 각각 SURF 와 SIFT, 호모그래피와 3 점 알고리즘을 이용한 위치 추정 결과이다. DB 를 선으로, 실험 영상을 점으로 나타냈다. 모든 영상은 차량을 움직이면서 획득한 것이기 때문에 결과는 급격하게 변하지 않는 직선 또는 완만한 곡선이어야 한다. 이러한 관점에서 살펴보면 4 개의 그래프 중 그림 12 가 가장 좋은 결과를 보인다고 할 수 있다.

위치 추정 결과의 오차의 원인으로는 여러 가지가 있을 수 있다. SURF 와 SIFT 등의 불변 특징량들은 특징점이 같은 곳에서 일정하게 나타나기는 하지만 위치 자체는 정확하지 않다. 또한 outlier 를 제거하기 위하여 RANSAC[7]을 수행하는데 다 제거되지 않은 outlier 가 결과에 영향을 미치기도 한다.

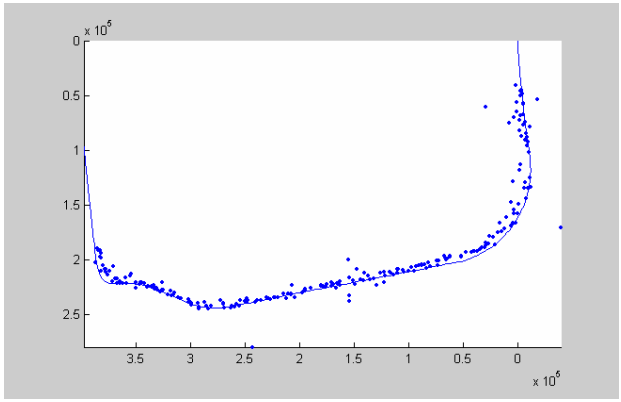

그림 9. SURF, 호모그래피를 이용한 결과

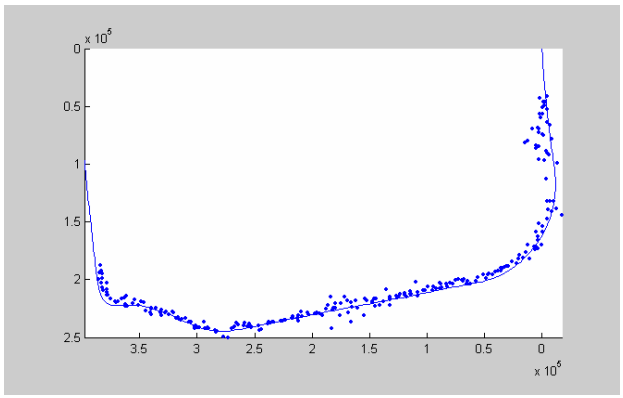

그림 10. SIFT, 호모그래피를 이용한 결과

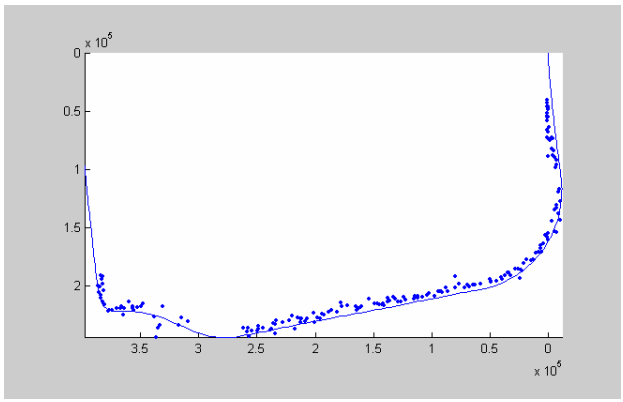

그림 11. SURF, 3 점 알고리즘을 이용한 결과

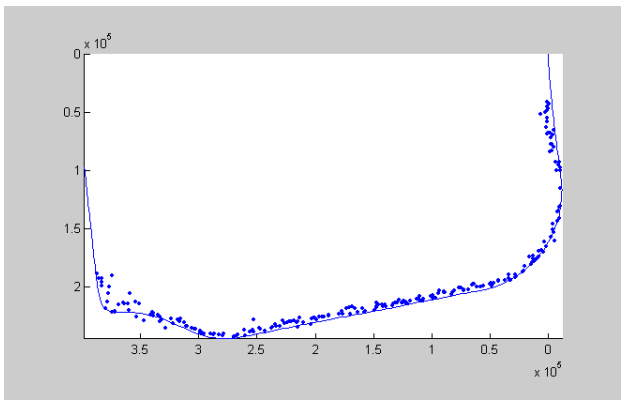

그림 12. SIFT. 3 점 알고리즘을 이용한 결과

전체 과정의 수행 시간은 표 6 과 같다. 계산 속도는 SURF 가 SIFT 보다 빠르며, P3P 가 호모그래피보다 빠른 것을 확인할 수 있다. 호모그래피가 느린 이유는 영상 5 장에 대한 호모그래피를 모두 구하기 때문이다. 또한 2 장에서의 측정 시간보다 영상 매칭 부분에 많은 시간이 소요되었는데, 그 이유는 KD tree 를 구축하는 시간이 포함되었기 때문이다. 2.2 절에서 설명한 부분 검색 방법은 DB 의 특징점 수는 적지만 tree 구조를 매번 생성해야 한다는 단점이 있다. 이는 시스템의 속도와 직접적인 관련이 있는데, DB 를 업데이트하지 않으면 계산 속도가 빨라지고, 계산속도가 빠를수록 DB 를 업데이트하는 회수가 줄어든다.

표 6. 각 과정별 계산 시간 (단위 : ms)

Method	Homography		P3P Algorithm	
Feature	SURF	SIFT	SURF	SIFT
Feature Extraction	146.52	554.98	146.52	554.98
Scene Matching	153.68	385.30	153.68	385.30
Feature Matching	154.07	268.33	48.65	87.50
RANSAC (1 iteration)	11.756	12.647	0.025	0.063

4. 결 론

본 논문에서는 영상을 이용하여 무인 차량의 위치 추정을 수행하였다. 실외 환경에서는 GPS 정보를 얻을 수 있지만 신호가 불안정한 환경이 다수 존재하기 때문에 이를 대체할 위치 추정 방법이 필수적이다. 위치 추정을 위하여 차량이 지나갈 장소에 대한 영상 DB 를 만들고, 실제 주행 시 획득한 영상과 DB 영상을 이용하여 위치를 추정하였다.

이러한 일을 수행하기 위한 전체 과정을 2 단계로 나누었다. 첫 번째 단계는 현재 영상과 가장 가까운 DB 영상을 검색하는 것이다. 스케일 변환 및 회전에 불변인 SIFT, SURF 두 가지의 특징량을 사용하여 영상 매칭을 수행하고 성능을 분석하였다. 계산 속도 개선을 위하여 KD tree 를 사용하였다. 두 번째 단계는 매칭된 DB 영상과 현재 영상을 이용하여 위치를 추정하는 것이다. 호모그래피를 이용하는 방법과 3 점 알고리즘을 이용하는 방법을 모두 구현하였다. 호모그래피 기반 방법에서 매칭된 DB 영상 1 장만으로는 정확한 위치를 알 수 없기 때문에 매칭된 DB 영상의 앞뒤로 2 장씩의 영상을 추가로 사용하여 현재 영상이 위치하는 직선들을 찾은 다음 이들의 교점을 구하였다.

구현한 알고리즘을 검증하기 위하여 실외 영상을 이용하여 실험하였다. SIFT 및 SURF 를 이용하여 영상 매칭을 수행한 결과 80% 이상의 성공률을 보였다. 계산 속도에서는 SURF 와 P3P 의 조합이 가장 뛰어났으며, 이 때의 계산 속도는 약 3Hz 정도였다. 이 정도로는 아직 실시간 시스템이라고 할 수는 없기 때문에 속도를 개선시킬 방안을 모색할 것이다.

이러한 영상 매칭 과정과 매칭된 DB 영상을 이용한 위치 추정 과정을 조합하여 실험용 영상의 위치 추정을 수행하였으며, 이로부터 만족할 만한 결과를 얻을 수 있었다. 하지만 실제 시스템으로 구현할 때 매칭이 되지 않는 20%는 문제가 될 수 있다. 이러한 부분은 제안된 알고리즘 내에서 해결하기 어려운 문제이므로 알고리즘 밖에서 해결책을 찾는 것이 좋다. 데이터베이스를 구축할 때 같은 장소이더라도 거리를 달리하면서 여러 장을 획득하거나, 영상이 연속적이라는 조건 등을 이용하여 제한을 두는 방법 등으로 해결하는 것이 가능하다.

본 논문에서 사용한 영상 데이터는 위치에 대한 GPS 등의 ground truth 가 없어서 직접적인 비교를 할 수 없었다. 앞으로의 테스트에서 추가해야 할 부분이다.

참고문헌

[1] D. G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints”, International journal of Computer Vision, 60(2), pp. 91-110, 2004.

[2] W. Zhang, J. Kosecka, “Image Based Localization in Urban Environments”, in Proceedings of the International Symposium on 3D Data Processing, Visualization, and Transmission, pp. 33-40, 2006.

[3] H. Bay, T. Tuytelaars, L. V. Gool, “SURF: Speeded Up Robust Features”, in Proceedings of the European Conference on Computer Vision, 2006.

[4] Y. Bok, Y. Hwang, I. S. Kweon, “Accurate Motion Estimation and High-Precision 3D Reconstruction by Sensor Fusion”, in Proceedings of the IEEE International Conference on Robotics and Automation, pp. 4721-4726, 2007.

[5] http://www.cs.umd.edu/~mount/ANN/

[6] R. M. Haralick, C. N. Lee, K. Ottenberg, M. Nölle, “Review and Analysis of Solutions of the Three Point Perspective Pose Estimation Problem”, International Journal of Computer Vision, 1994.