

특징점 기반 단안 영상 SLAM의 최적화 기법 및 필터링 기법 성능 분석

Performance Analysis of Optimization Method and Filtering Method for Feature-based Monocular Visual SLAM

전진석* · 김효중* · 심덕선†
(Jin-Seok Jeon · Hyo-Joong Kim · Duk-Sun Shim)

Abstract - Autonomous mobile robots need SLAM (simultaneous localization and mapping) to look for the location and simultaneously to make the map around the location. In order to achieve visual SLAM, it is necessary to form an algorithm that detects and extracts feature points from camera images, and gets the camera pose and 3D points of the features. In this paper, we propose MPROSAC algorithm which combines MSAC and PROSAC, and compare the performance of optimization method and the filtering method for feature-based monocular visual SLAM. Sparse Bundle Adjustment (SBA) is used for the optimization method and the extended Kalman filter is used for the filtering method.

Key Words : Camera image, Monocular SLAM, Feature point, Optimization, Kalman filter, RANSAC

1. 서론

오늘날 카메라 영상은 개인의 취미 활동뿐만 아니라 산업의 여러 분야에서 필수적인 요소로 자리 잡고 있다. 카메라는 컴퓨터비전의 주요 센서일 뿐만 아니라 드론, 자율주행 자동차와 같은 이동로봇의 센서로 최근 주목받고 있다. 이동 로봇은 자율 주행/비행을 위해서 로봇의 위치, 자세를 추정하고 주변의 지도를 동시에 작성하는 SLAM(simultaneous localization and mapping)을 수행해야 한다. 최근 로봇 SLAM에 GPS, 관성센서, 레이저, 소나 등 전통적인 센서 외에 카메라가 주요 센서로 자리잡고 있다.

컴퓨터비전 분야의 SfM(structure from motion)문제와 이동로봇의 SLAM문제는 오랜 기간 각각 발전해 왔는데, 동기는 달랐지만 근본적으로 같은 문제를 다루고 있다. SfM은 카메라의 2차원 영상으로부터 3차원 영상을 얻는게 목적이며 투사기하(projective geometry)와 최적화 기법을 사용하는데 비해, SLAM은 여러 센서를 사용해서 로봇의 위치와 주변 지도를 얻는 것이 목적이고 필터링 기법을 사용한다. 최근에는 SfM과 SLAM의 두 분야의 겹을 이어주는 연구가 많이 진행되었으며, 실시간 SfM문제는 영상 SLAM이라고도 한다. SfM의 특징점과 SLAM의 지도는 같은 개념인데, 필터링기법에서는 특징점이 상태식의 상태변수에 포함되

므로 특징점이 많아질수록 상태변수의 차수가 커지면서 계산량이 급증한다. SLAM의 필터링 기법보다는 SfM의 최적화 기법이 성능이 좋다는 결과가 발표되기도 했다[1].

특징점 기반 영상 SLAM에서는 영상에서 특징점을 추출하고 매칭하는 알고리즘이 필요하며, 이상점(outlier)을 제거하기 위해 RANSAC(RANdom SAmple Consensus)과 같은 알고리즘을 수행해야 한다. RANSAC에는 정확성, 계산의 신속성, 강인성을 향상시키는 여러 알고리즘들이 개발되어 있다[2].

본 논문에서는 RANSAC에서 매칭의 정확성을 향상시키는 MSAC(M-estimator)과 매칭의 계산 속도를 향상시키는 PROSAC(progressive SAC) 알고리즘을 결합시킨 MPROSAC을 제안하였다. 영상 SLAM에서의 카메라 자세 추정 방식에는 전체적으로 3가지의 방법(2D-2D, 3D-3D, 3D-2D)이 존재한다. 본 논문에서는 2D-2D와 3D-2D의 방법을 사용한 자세추정 방법에 대한 성능을 비교하였고, 영상 SLAM에서의 지도 작성을 위한 최적화 알고리즘인 번들조정(BA, bundle adjustment)과 필터링기법에서 많이 사용되는 칼만 필터의 성능을 비교하였다.

일반적으로 부유오차(drift)를 줄이기 위해서 루프 클로징(loop closing)을 하게 되는데, 본 논문에서는 루프 클로징 대신에 지역적인 지도 작성만 가지고 성능을 비교하였다.

논문 구성은 2장에서 좌표계 및 영상의 특징점에 대하여 기술하였고, 3장은 본 논문에서 제안한 MPROSAC 알고리즘에 대하여 기술하였다. 4장에서는 카메라 자세 추정방법에 대하여 기술하였고, 5장은 카메라 자세와 3차원 점에 대한 보정 방법에 대해 기술하였다. 마지막 6장에서는 4장과 5장에서 나타난 방법들의 성능을 비교하는 시뮬레이션을 실시하였다.

† Corresponding Author : School of Electrical and Electronics Engineering, Chung-Ang University, Seoul, Korea.

E-mail: dshim@cau.ac.kr

ORCID ID: <https://orcid.org/0000-0003-0141-9498>

* School of Electrical and Electronics Engineering, Chung-Ang University, Seoul, Korea.

Received : December 11, 2018; Accepted : December 23, 2018

2. 좌표계 및 영상의 특징점

2.1 카메라 좌표계와 월드 좌표계

카메라 좌표계와 월드 좌표계의 관계는 그림 1과 같이 월드 좌표계의 3차원 점으로부터 카메라 좌표계의 영상 평면의 2차원 점으로 변환하는 과정으로 볼 수 있으며 식 (1)과 같이 표현할 수 있다.

$$d \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & c & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix} \quad (1)$$

여기서 d 는 depth 값, u, v 는 정규화된 영상 평면에서의 특징점, f_x, f_y 는 초점거리, c 는 왜곡계수, u_0, v_0 는 광학축에서의 직선으로 내린 영상 평면에서의 주점, X_W, Y_W, Z_W 는 월드 좌표계에서의 3차원 점, $r_{ij} (i=1,2,3 j=1,2,3)$ 는 회전변환 행렬의 성분들, $t_k (k=1,2,3)$ 는 평행이동 벡터의 성분을 의미한다.

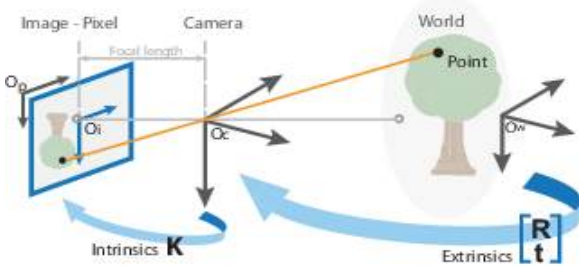


그림 1 월드좌표계와 카메라좌표계의 관계

Fig. 1 World coordinate and camera coordinate

2.2 영상의 특징점 추출, 검출, 및 매칭

본 논문에서는 특징점을 추출하는 알고리즘으로 SURF (speeded up robust features)를 사용하였다[3]. SURF는 영상의 적분 영상을 사용하여 계산 속도를 증가시키며, 박스 필터의 사이클을 증가함으로 인해서 스케일에 불변하게 된다. 또한 근사화한 헤시안 행렬을 사용하여 특징점 추출을 하게 되며, 식은 다음과 같다.

$$H = \begin{pmatrix} d_{xx}(\sigma) & d_{xy}(\sigma) \\ d_{xy}(\sigma) & d_{yy}(\sigma) \end{pmatrix} \quad (2)$$

$$d_{yx}(\sigma) = \frac{\partial}{\partial x} \left(\frac{\partial}{\partial y} (G(y,x,\sigma) * f(y,x)) \right) \quad (3)$$

식 (2)는 헤시안 행렬 H , 식 (3)은 가우시안 라플라시안 (Laplacian of Gaussian)이며, $G(x, y, \sigma)$ 는 가우시안 필터로 입력 영상 $f(y, x)$ 과 컨볼루션(convolution)한 후, y방향으로 미분하고 다시 x방향으로 미분하여 얻은 도함수 $d_{yx}(\sigma)$ 로 표현된다. 검출된 특징점이 회전불변하기 위해 방향(orientation)을 정의

한다. 이 방향은 하르 웨이블릿 응답(Haar wavelet response)에 대한 dx, dy 방향의 응답이 가장 큰 벡터로 정해지게 된다. 주 방향이 정의되면 4x4 윈도우를 씌우고 다시 하나의 윈도우에 대해 5x5 윈도우를 씌워서 각각을 dy, dx으로 히스토그램으로 표시하게 된다. 이를 식으로 표현하면 다음과 같다.

$$V = (\sum dx, \sum |dx|, \sum dy, \sum |dy|) \quad (4)$$

이 4가지 요소를 히스토그램으로 그려서 기술자(descriptor)로 표현하게 되며, 기술자는 $[n \times 64]$ 차원의 벡터 형식으로 추출된다. 여기서 n 은 특징점의 개수를 의미한다. 매칭을 하기 위해서 본 논문에서는 SSD(sum of squared differences)방식을 사용한다. 이 방법은 하나씩 전부 다 조사하는 전수조사 방법을 사용하고, 식으로 표현하면 다음과 같다.

$$SSD(\text{Sum of Squares Difference}) = \| D_i^k - D_j^{k+1} \| \quad (5)$$

여기서 D 는 기술자벡터(descriptor vector)를 의미하고, k 는 이미지의 순서를 의미한다. i, j 는 각각 이미지의 특징점의 인덱스를 의미한다. 매칭할 때 SSD를 대표적으로 사용한다.

매칭을 하게 되면 특징점 간에 이상점들이 발생하게 된다. 영상 사이의 이상점을 제거해 주기 위해 연구가 지속되어 왔으며 [4,5], 대표적으로 많이 사용되는 알고리즘이 RANSAC(RANdom SAMple Consensus)이다. 랜덤하게 샘플을 뽑아서 inlier인지 outlier인지 판단하는 알고리즘이다.

3. MPROSAC 알고리즘

본 절에서는 RANSAC의 개선된 알고리즘들에서[2] 속도 성능과 정확성을 동시에 향상시킬 수 있는 MPROSAC 알고리즘에 대해 기술한다.

3.1 제안하는 MSAC/PROSAC 결합 알고리즘

MSAC과 PROSAC은 RANSAC의 성능을 보완하기 위해 개선된 방법들이며, MSAC은 정확성, PROSAC은 계산 속도에 대한 성능을 개선시킨다. MSAC은 경계값(threshold)에 대한 inlier 판정 후, outlier가 inlier 집단에 들어갈 수 있기 때문에, 모든 inlier에 속한 집단에 대해 손실을 계산하여 정확도를 높게 된다. 손실(loss) 함수에는 Huber 함수를 사용한다. 식 (6)은 Huber 함수이다.

$$\rho(e^2) = \begin{cases} 0.5 * e^2, & e^2 < T^2 \\ 0.5 * T^2, & e^2 \geq T^2 \end{cases} \quad (6)$$

여기서 e 는 모델에 대한 오차값, T 는 경계값이다. PROSAC은 유도된 샘플링 방법으로, 초기에 매칭점을 샘플링하여 매칭 점수가 가장 높은 것부터 내림차순으로 정렬하여 가중치를 주는 방법이다. 이 정렬된 함수를 quality 함수라고 부른다. 다음은 quality 함수에 대한 식이다.

$$q(u_i) \geq q(u_j) \Rightarrow P(u_i) \geq P(u_j), i < j \quad (7)$$

$q()$ 은 quality 함수, P 는 확률, i, j 는 임의의 순서이다. 이 함수를 적용하기 위해서는 매칭방법을 다르게 하여 오차 순으로 함수를 정의한다. 매칭 방법으로는 가장 좋은 (first best) 매칭점과 두 번째로 좋은(second best) 매칭점에 대한 비율로 계산하며, 매칭비율은 식 (8)과 같다.

$$\frac{\text{First Best Matching Point}}{\text{Second Best Matching Point}} = \frac{\|D_i^k - D_j^{k+1}\|}{\|D_i^k - D_j'^{k+1}\|} \quad (8)$$

식 (8)은 식 (5)에 대해 가장 좋은 매칭점과 두 번째로 좋은 매칭점에 대한 비율이다. 이와 같이 quality 함수를 적용한 뒤, 이전의 샘플링 집합과 현재의 샘플링 집합을 랜덤하게 뽑는다. 이후로는 RANSAC의 알고리즘과 동일하게 진행한다. MPROSAC 알고리즘은 그림 3과 같다.

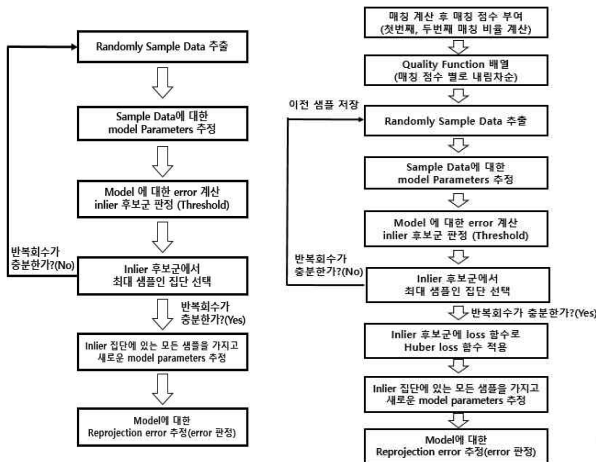


그림 2 RANSAC(좌)과 MPROSAC(우) 알고리즘
Fig. 2 RANSAC(left) and MPROSAC(right) algorithm

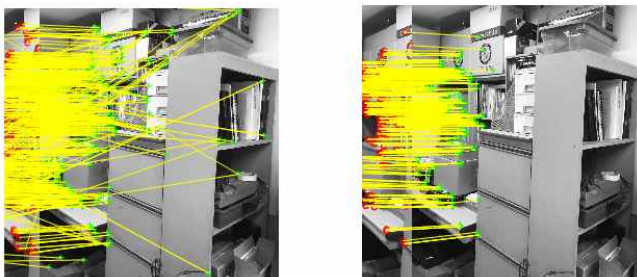


그림 3 MPROSAC 사용 전(좌)과 MPROSAC 사용 후(우)
Fig. 3 Before (left) and after (right) using MPROSAC

3.2 MPROSAC 결과 시뮬레이션

MPROSAC 알고리즘의 성능을 비교해보기 위해서 3.1절에 소개된 MPROSAC 알고리즘을 사용하여 매칭 전후의 사진을 그림 3

에 도시하였다. 사용 후에 outlier가 많이 줄어든 것을 볼 수 있다. [표 1]은 RANSAC, MSAC, PROSAC을 개별적으로 시뮬레이션 한 결과와 MPROSAC의 계산 속도와 정확도 성능을 비교하였다. [표 1]에서 알 수 있듯이 MPROSAC 사용 시에 매칭점의 개수가 증가하고, MSAC에 비해서 오차는 조금 증가했지만 계산속도가 30배 이상 빨라졌고, PROSAC에 비해서는 오차정확도와 계산 속도 성능이 모두 나아진 것을 볼 수 있다.

표 1 RANSAC, MSAC, PROSAC, MPROSAC 성능비교
Table 1 Performance comparison among RANSAC, MSAC, PROSAC, and MPROSAC

	매칭점 개수	오차[x,y](pix)	속도(sec)
RANSAC	31	[3.34 6.87]	14.12
MSAC	64	[2.47 4.35]	13.21
PROSAC	121	[4.32 7.44]	0.937
MPROSAC	189	[3.12 5.85]	0.432

4. 카메라 자세 추정 알고리즘

4.1 2D-2D 자세 추정 알고리즘

영상 사이에 매칭점 추출과 이상점 제거를 마치면, 두 영상 사이의 특징점들로부터 카메라의 자세를 추정할 수 있다. 여기에 사용하는 대표적인 알고리즘으로 8-point 알고리즘[6]이 있다.

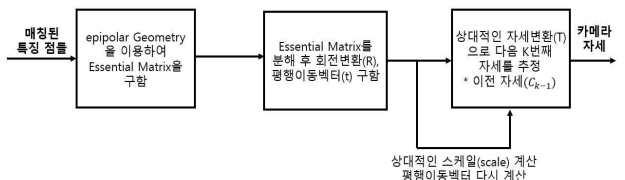


그림 4 2D-2D 카메라 자세 추정 알고리즘 블록도
Fig. 4 Block diagram for 2D-2D camera pose estimation

그림 4와 같이 매칭된 특징점들이 들어오면 에피폴라 (epipolar) 조건식을 이용하여 Essential 행렬을 구할 수 있다. 에피폴라 조건식은 다음과 같다.

$$\tilde{p}'^T E \tilde{p} = 0 \quad (9)$$

위에서 나온 \tilde{p}, \tilde{p}' 는 특징점, E 는 Essential 행렬이다. 식 (9)에 두 영상에서 매칭된 8개의 특징점들을 사용하면 SVD (singular value decomposition) 방법을 사용하여 Essential 행렬을 구할 수 있다. 또 Essential 행렬이 정해지면 식 (10)~식 (13)으로부터 평행이동 벡터 \hat{t} 와 회전변환 행렬 R 을 구할 수 있다. 평행이동벡터인 \hat{t} 은 반대칭행렬(skew-symmetric matrix)로 표현된다.

$$E_k = \hat{t}_k R_k \quad (10)$$

$$\hat{t} = U(\pm W) S U^T \quad (11)$$

$$R = U(\pm W^T) V^T \quad (12)$$

$$W = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (13)$$

4.2 상대적인 scale 계산 및 평행이동 벡터 보정

4.1절에서 구한 평행이동 벡터는 상대적인 scale에 대한 보정을 해 주어야 하며, 이를 위해서는 연속적인 3개의 영상이 필요하다. 이 과정에서 3차원 점을 구하고 2개의 3차원 점에 대한 특징점들에 비율로 상대적인 scale(크기)를 표현할 수 있다. 식은 다음과 같이 표현된다.

$$ratio = \frac{\|X_{k,(i,i+1)} - X_{k-1,(i,i+1)}\|}{\|X_{k,(i-1,i)} - X_{k-1,(i-1,i)}\|} \quad (14)$$

X_k, X_{k-1} 는 특징점 k번째, k-1번째의 3차원 점을 의미한다. $i-1, i, i+1$ 은 연속적인 영상을 의미한다.

4.3 3D-2D 자세 추정 알고리즘

3D-2D 자세 추정 알고리즘은 말 그대로 3차원 점과 2차원 점으로의 자세를 추정하는 알고리즘이며 그림 5와 같다.

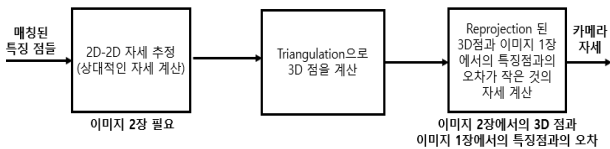


그림 5 3D-2D 카메라 자세 추정 알고리즘 블록도
Fig. 5 Block diagram for 3D-2D camera pose estimation

위의 알고리즘을 구현하기 위해서는 먼저 연속적인 3개의 영상에서 2D-2D 알고리즘으로 상대적인 자세를 추정하고 그것으로 인해 3차원 점을 계산하고 나서, 3D-2D 알고리즘을 실행할 수 있다. 최종적으로는 3차원 점을 2차원으로 투사시켜서 발생하는 오차를 최소화하는 자세를 찾는 것이 3D-2D 알고리즘이다. 식으로 나타내면 다음과 같다.

$$\operatorname{argmin} \sum_i \|p_k^i - \hat{p}_{k-1}^i\|^2 \quad (15)$$

3차원 점을 투사시킨 점이 \hat{p}_{k-1}^i 으로 표현되고, p_k^i 은 k번째 영상에서의 2차원 점을 의미한다.

4.4 삼각측량법(Triangulation)

3D-2D 알고리즘에서 2D-2D 알고리즘을 실행하고 난 후, 3

차원 점을 계산할 때 쓰는 방법은 삼각측량법(triangulation)으로 매칭점의 3차원 좌표를 계산할 수 있다. 카메라 좌표계에서부터 homogeneous 좌표계(동차 좌표계)를 표현하는 식은 다음과 같다.

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \frac{1}{Z} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}, \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \frac{1}{Z'} \begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} \quad (16)$$

x, y 는 영상 평면에 생기는 특징점이고, X, Y, Z 는 카메라 좌표계에서의 3차원 점을 의미한다. 월드좌표계의 3차원 점을 카메라 좌표계의 3차원 점으로 바꾸기 위해서는 다음과 같은 식을 정의할 수 있다.

$$P' = RP + t \quad (17)$$

P' 는 카메라 좌표계에서의 3차원 점, P 는 월드좌표계에서의 3차원 점, R, t 는 Extrinsic 행렬의 성분인 회전변환과 평행이동 벡터이다. 따라서 최종적으로 카메라 좌표계에서의 Z점을 구하고 그것을 월드 좌표계로 옮기기 위해 식 (17)을 쓰면 월드 좌표계의 3차원 점을 계산할 수 있다. 따라서 카메라 좌표계에서의 Z 식은 다음과 같다.

$$Z_1' = \frac{(r_1 - xr_3)t}{(r_1 - xr_3)x}, Z_2' = \frac{(r_2 - yr_3)t}{(r_2 - yr_3)x} \quad (18)$$

Z에 대한 식이 두 개 이상이 나오게 되므로 모든 성분을 더해서 평균을 내는 형식으로 사용된다.

5. 카메라 자세 및 3차원 점 보정 알고리즘

5.1 BA(번들조정, bundle adjustment) 방법

BA란 다발 묶음으로 모아서 보정하는 방법이라는 말로 지역적(local), 광역적(global) 방법으로 나뉜다. 보통 영상 odometry에선 지역적으로 사용하고, 영상 SLAM에서는 광역적으로 BA가 사용된다. 영상의 입력에서부터 BA까지의 블록도는 다음과 같다.

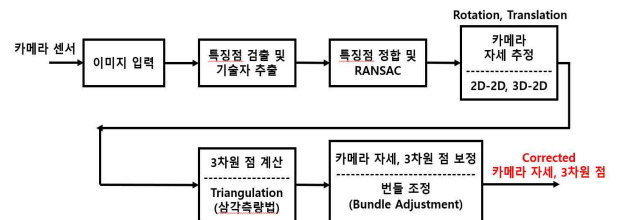


그림 6 BA 알고리즘 블록도
Fig. 6 Block diagram for BA algorithm

BA의 목적은 카메라의 자세와 3차원 점의 보정이다. BA는 재투사 오차를 최소화하는 자세와 3차원의 점을 구하는 알고리즘으로, 식 (19)을 최소화하는 자세와 3차원 점을 구한다.

$$\min \sum_{i=1}^n \sum_{j=1}^m d(Q(a_j, b_i), x_{ij})^2 \quad (19)$$

a_j 는 각각의 카메라 영상 j 번째를 벡터화, b_i 는 3차원의 점의 개수 i 에 대해 벡터화, $Q(\cdot)$ 는 j 번째 영상에서의 i 번째의 특징 점을 예측투사한 함수를 표현, $d(\cdot)$ 는 유클리디안 거리를 표현한 것이다. x_{ij} 는 측정치의 2차원 점을 의미한다.

BA에서의 입력은 비선형 함수이기 때문에 비선형 최소 자승의 한 종류인 Levenberg-Marquardt를 사용하였다. 식은 다음과 같다.

$$(J^T \sum_X^{-1} J + \mu I) \delta = J^T \sum_X^{-1} \epsilon \quad (20)$$

J 는 자코비안, δ 는 파라미터 벡터 P 에 대해 업데이트 변수, μ 는 Dumping 인자, $\epsilon = X - \hat{X}$ (재투사 오차), 이 오차에 대해 마할라노비스 거리(Mahalanobis distance) $\epsilon^T \sum_X^{-1} \epsilon$ 에 대해 LM 방식의 비선형 최소 자승 알고리즘으로 풀 것이 위의 식이다[7].

5.2 칼만 필터 방법(Kalman Filter)

본 논문은 필터링 방법으로 확장형 칼만필터(EKF, extended Kalman filter)를 사용하였다. 영상이 입력되면 초기 포즈 값을 넣어주고, 3차원 점을 계산함으로써 카메라 자세와 3차원 점을 보정하는 칼만필터의 입력이 결정된다. EKF 알고리즘은 그림7과 같다.

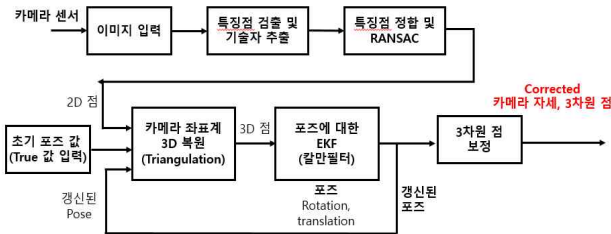


그림 7 확장 칼만필터 알고리즘 블록도

Fig. 7 Block diagram for Extended Kalman filter algorithm

EKF의 상태변수 x 와 측정식 I 는 다음과 같이 정의한다.

$$x = [t_x \dot{t}_x \ t_y \dot{t}_y \ t_z \dot{t}_z \ \alpha \ \dot{\alpha} \ \beta \ \dot{\beta} \ \gamma \ \dot{\gamma}] \quad (21)$$

$$x_k = Ax_{k-1} + w_k, \quad w_k = N(0, Q) \quad (22)$$

$$I_k = h(x_k) + n_k, \quad n_k = N(0, \lambda) \quad (23)$$

$$A = \text{diag} \left\{ \begin{bmatrix} 1 & \tau \\ 0 & 1 \end{bmatrix}, \dots, \begin{bmatrix} 1 & \tau \\ 0 & 1 \end{bmatrix} \right\} \quad (24)$$

$$I_k = [u_1, v_1, \dots, u_m, v_m]^T \quad (25)$$

$$h(x_k) = \left\{ \left[f_x \frac{X_1}{Z_1}, f_y \frac{Y_1}{Z_1}, \dots, f_x \frac{X_m}{Z_m}, f_y \frac{Y_m}{Z_m} \right]^T \right\} \quad (26)$$

위와 같이 상태변수는 이동 벡터의 x, y, z 값과 오일러 각도인 롤, 피치, 요 값을 상태변수로 설정하였다. 상태식의 A 행렬은 카메라가 등속도로 진행한다고 가정 하에 구했다. $h(\cdot)$ 함수는 영상 평면상 표현되는 초점거리와 카메라 좌표계의 3차원 점의 관계로 표현할 수 있다[8].

6. 카메라 자세와 3차원 점에 대한 시뮬레이션

6.1 카메라 자세추정 방법 비교

6.1.1 2D-2D 자세 추정방법의 성능

2D-2D 자세추정의 영상으로는 등속도로 이동하는 드론에 대한 영상[720x1280]을 사용하였다. 그림 8에서 보는 것과 같이 2D-2D 자세 추정의 scale 보정을 하지 않으면 오차가 x축 진행 방향으로 scale이 커지는 것을 확인하였다.

6.1.2 2D-2D 자세 추정방법 scale 보정 후 성능

6.1.1절에서 언급한 바와 같이 2D-2D 자세 추정에서 scale 보

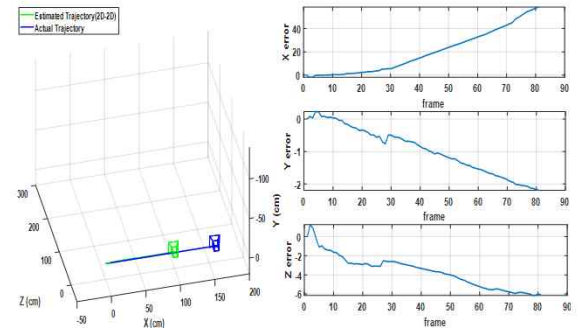


그림 8 2D-2D 자세 추정방법 결과

Fig. 8 Result of 2D-2D camera pose estimation

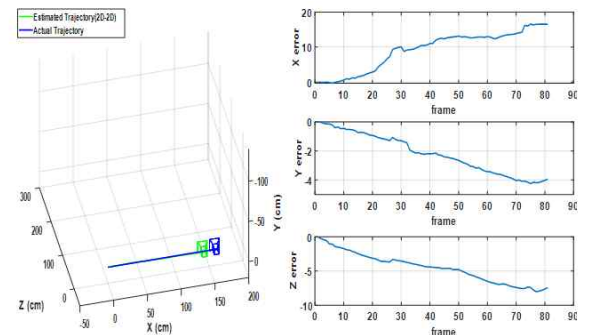


그림 9 Scale 보정 후 2D-2D 자세 추정방법 결과

Fig. 9 Result of 2D-2D camera pose estimation after scale correction

정을 하지 않으면 진행하는 방향으로의 거리가 멀어지게 된다. 4.2절의 알고리즘을 사용하여 상대적인 scale 보정을 해주어 얻은 결과는 다음과 같다. 그림8에 비해서 그림 9는 카메라의 위치가 잘 보정되었고 진행 방향인 x축 오차가 많이 줄어든 것을 확인할 수 있다.

6.1.3 3D-2D 자세 추정방법의 성능

자세 추정방법 중에서 정확도가 가장 높은 결과는 3D-2D 자세 추정방법이다. 6.1.1절의 2D-2D 자세추정 방법을 사용한 경우 x축 오차를 보면, 그림10의 결과에서 정확도가 좋아진 것을 알 수 있다.

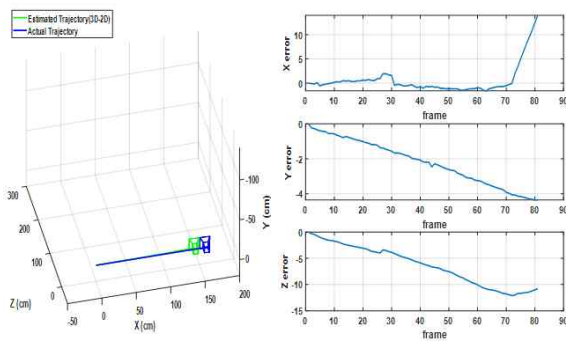


그림 10 3D-2D 자세 추정방법 결과
Fig. 10 Result of 3D-2D camera pose estimation

6.2 카메라 자세 및 3차원 보정 성능

6.2.1 3D-2D기반 번들조정 3차원 점 성능

6.1절의 자세 알고리즘에서 정확도가 가장 높은 3D-2D 알고리즘을 기반으로 카메라 자세와 3차원 점에 대하여 BA 알고리즘을 실행하였다. 그림 11을 보면 우측의 실제 영상의 물체의 윤곽이 좌측에 3차원 특징점으로 잘 드러나고 있음을 확인할 수 있다.

6.2.2 확장 칼만필터 3차원 점 성능

5.2절에서 언급한 EKF 알고리즘을 사용하여 3차원 점에 대한

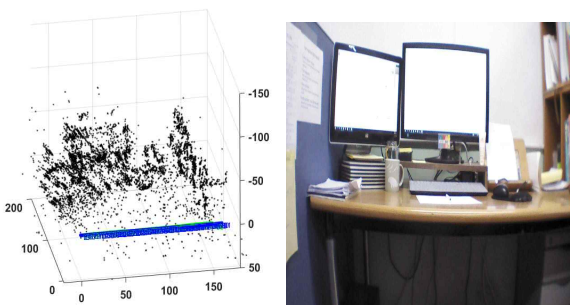


그림 11 3D-2D 기반 번들조정 후 3차원 점 성능 결과
Fig. 11 Result of 3D points after 3D-2D based BA

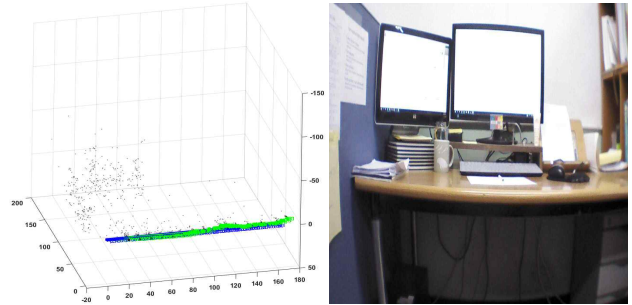


그림 12 확장 칼만필터 3차원 점 성능 결과
Fig. 12 Result of 3D points after EKF

성능을 얻은 결과를 그림 12에 도시하였다. 좌측 그림에서 3차원 점들이 희미하게 보이는데 오른쪽 물체의 윤곽을 알아보기 어렵다. 육안상으로도 그림 11과 그림 12에서 EKF가 BA보다 3차원 점을 보정하는 성능이 떨어진다는 것을 알 수 있었다.

7. 결 론

본 논문에서는 두 영상의 매칭 후 이상점을 제거하는 RANSAC 알고리즘 중에서 MSAC과 PROSAC의 장점을 결합한 MPROSAC 알고리즘을 제안하고 성능이 우수함을 확인하였다.

또 영상 SLAM의 알고리즘으로 필터링 방법인 EKF와 최적화 방법인 BA의 성능을 비교하였다. 크게 2가지의 성능을 비교, 분석하였는데, 첫째로는 자세 추정방법에 대한 성능 비교, 둘째로는 카메라 자세 추정 및 3차원 점 보정방법에 대한 성능을 비교하였다. 첫 번째의 결과에서는 2D-2D 자세 추정방법보다는 3D-2D 자세 추정방법이 정확도가 높다는 것을 알 수 있었고, 두 번째의 결과에서는 칼만필터를 사용한 3차원 점 보정 방법보다는 번들 조정의 방법으로 3차원 점을 보정하는 방법이 특징점의 정보와 정확도 면에서 뛰어나다는 것을 알 수 있었다.

감사의 글

본 논문은 2018년 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업(No. 2017R 1D 1A 1B03035207).

References

- [1] H. Strasdat, J. M. M. Montiel, A. J. Davison, "Visual SLAM: Why filter?", *Image and Vision Computing* 30, pp. 65- 77, 2012.
- [2] Sunglok Choi, Taemin Kim, Wonpil Yu, "Performance evaluation of RANSAC family", *British Machine Vision Conference, BMVC 2009, London, UK*, pp. 81.1-81.12,

7-10 September 2009.

- [3] Herbert Bay, Tinne Tuytelaars, Luc Van Gool, "SURF: Speeded Up Robust Features", *Computer Vision and Image Understanding*, Vol. 110, Issue 3, pp. 346-359, June 2008.
- [4] Martin A. Fischler and Robert C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography", *Communications of the Association for Computing Machinery*, Vol. 24, No. 6 pp. 381-395, June 1981.
- [5] P. H. S. Torr, A. Zisserman, "MLESAC: A new robust estimator with application to estimating image geometry", *Computer Vision and Image Understanding*, Vol. 78, pp. 138-156, 2000.
- [6] Richard I. Hartley, "In Defense of the Eight-Point Algorithm", *IEEE Transactions on Pattern Recognition and Machine Intelligence*, Volume 19, (6), pp. 580-593, June 1997.
- [7] MANOLIS I. A. LOURAKIS and ANTONIS A. ARGYROS, "SBA: A Software Package for Generic Sparse Bundle Adjustment", *ACM Transaction on Mathematical Software(TOMS)*, Vol. 36, No 1, Article 2, pp. 2:1-2:30, March 2009.
- [8] M.E. Ragab, and K.H. Wong, "Extended Kalman Filter Based Pose Estimation Using Multiple Cameras", *The CSE Dept., The Chinese University of Hong Kong, Internal report*, 16 May, 2007.
- [9] Jin-Seok Jeon, Hyo-Joong Kim, Duk-Sun Shim, MSAC/PROSAC Fusion Algorithm to Enhance SURF Performance, Conference on Information and Control Systems, pp. 276-277, 26 Oct, 2018.



김 효 중 (Hyo-Joong Kim)

2018년 중앙대학교 전자전기공학부 졸업(공학사), 현재 중앙대학교 전자전기공학부 석사과정 재학 중, 관심분야는 로봇 SLAM, 영상-관성 odometry, 로봇비전.



심 덕 선 (Duk-Sun Shim)

1984년, 1986년 서울대 제어계측공학과 졸업(공학사, 공학석사), 1993년 미시간대 항공우주공학과 졸업(공학박사), 1994년 미시간대 전기및컴퓨터공학과 포스트닥, 1995년 3월~현재 중앙대학교 전자전기공학부 교수, 2009년~2011년 JEET 에디터, 2014년~2017년 IJCAS 에디터, 2018년 대한전기학회 부회장, 정보 및 제어 부문회장, 관심분야는 제어, 위성항법, 관성항법, 로봇 SLAM, 영상-관성 odometry, 로봇비전.

저 자 소 개



전 진 석 (Jin-Seok Jeon)

2017년 단국대학교 전자전기공학부 졸업(공학사), 현재 중앙대학교 전자전기공학부 석사과정 재학중, 관심분야는 로봇 SLAM, 로봇비전, 위성항법.