

Lidar-Inertial Odometry 를 활용한 극한 환경에서의 비디오 안정화 기술 개발

Development of Video Stabilization Technology in Extreme Environments Using Lidar-inertial Odometry

윤영인¹, 서형석¹, 신관준¹, 임종우^{1*}

(Yeong-In Yoon¹, Hyeong-Seok Suh¹, Gwan-Jun Shin², and Jong-Woo Lim^{1,*})

¹Department of Mechanical Engineering, Seoul National University

²Interdisciplinary Program in Artificial Intelligence, Seoul National University

Abstract: This study proposes designing and applying a video stabilization algorithm using LIDAR-inertial odometry (LIO). The performance of the video stabilization function was validated in narrow, low-light environments using a small robot. A unique camera pose estimation method based on LIO has been introduced and integrated with the camera system in this work to not only overcome the limitations of image feature-based camera pose estimation but also address the issues arising from camera sensors. Thus, this strategy allows the practical implementation of a robust camera stabilization algorithm even in extreme environments. The proposed approach is resilient to the differing scan rates of the LIDAR system and the camera, and it enhances usability by filtering noise from the LIO. Furthermore, it requires less computational resources compared to traditional image feature extraction-based methods. This technology can substantially improve the operational convenience for robot operators, especially in scenarios where robots equipped with various sensors are deployed in extreme environments.

Keywords: video stabilization, image perspective transformation, lidar-inertial odometry

I. 서론

최근 로보틱스 분야에서는 카메라를 활용한 객체 인식과 깊이 추정 등 다양한 연구가 활발하게 이루어지고 있다[1-2]. 하지만 시스템의 급격한 움직임으로 인해 발생하는 카메라의 흔들림은 이러한 알고리즘의 성능과 활용에 한계를 초래할 수 있다. 비디오 안정화 기술은 움직임이 많은 카메라 장착 시스템에서 카메라 시스템의 자세를 추정하고, 이를 컴퓨터 비전 기술의 원근 변환(perspective transform)을 이용하여 급격한 움직임과 로봇의 뒤틀림으로 인한 기울어진 화면을 보정하는 기술이다. 이 기술은 불안정한 환경에서 안정적인 영상 품질을 보장할 수 있기 때문에 드론, 차량, 로봇 등 다양한 분야에서 필수적인 기능으로 자리잡고 있다. 흔히 짐벌(gimbal)이라고도 불리며, 카메라 시스템의 자세를 추정하여 수평 상태 혹은 급격한 이동을 억제시키고 안정적인 시스템 자세를 유지하도록 하는 방식으로 안정화된 비디오 프레임을 생성해낼 수 있다. 대부분의 SW 기반 비디오 안정화 기술은 이미지 기반 특징점 추출 방식[3-4]이나 이미지 인페인팅 방식[5-6], 딥러닝 기반[7-10]의 방식으로 이루어지고 있다. 이미지 기반 특징점 추출을 통한 카메라 시스템의 자세 추정 방식은 이미지를 이용하여 간편하게 구성할 수 있지만, 컴퓨터 자원 소모가 크고, 특징

점이 없는 단조로운 환경이나 빛이 부족한 극한 환경에서는 모션 블러나 롤링 셔터 왜곡 등 카메라 센서 자체의 한계로 해결하기 어려운 문제점이 발생한다. 카메라의 움직임으로 인해 손상된 이미지의 일부를 주변 픽셀의 정보를 이용하여 복원하는 인페인팅 방식은 매우 동적인 환경에서는 아티팩트로 인하여 이미지를 적절히 복원하기 어렵다는 단점이 존재한다. 앞선 방식의 한계점을 해결하기 위한 딥러닝 기반의 접근법도 제시되었으나, 이미지 처리 특성상 제한된 컴퓨팅 환경에서 구동하는 데 어려움이 많다.

따라서 본 논문에서는 LIO (Lidar Inertial Odometry)를 활용한 비디오 안정화 기술을 제안한다. 라이다 센서와 관성 측정 센서가 장착된 로봇에서 LIO 알고리즘을 통해 로봇과 카메라 시스템의 자세를 추정한 뒤, 이를 비디오 안정화에 사용하는 방식이다. 기존의 이미지 기반 비디오 안정화 기법에 비해 컴퓨터 자원 소모가 적으며, 이미지 특징점이 없거나 부족한 환경에서도 원활하게 작동할 수 있다. 알고리즘은 서로 다른 주기의 라이다, IMU, 카메라에도 강건하게 작동할 수 있도록 설계되었으며 협소 공간 저조도 환경에서 소형 로봇으로, 일반적인 조도 환경에서 Hand-held 실험으로 성능을 검증하였다.

* Corresponding Author

Manuscript received October 22, 2024; revised November 18, 2024; accepted November 23, 2024

윤영인: 서울대학교 기계공학부 대학원생(yoonyi2017@snu.ac.kr, ORCID[®] 0009-0004-5290-8261)

서형석: 서울대학교 기계공학부 대학원생(hyeongaa@snu.ac.kr, ORCID[®] 0009-0002-5688-3466)

신관준: 서울대학교 협동과정 인공지능전공 대학원생(shinkansan@snu.ac.kr, ORCID[®] 0000-0002-3691-4707)

임종우: 서울대학교 기계공학부 교수(jongwoo.lim@snu.ac.kr, ORCID[®] 0000-0002-2814-4765)

※ 이 연구는 2024년도 산업통상자원부 및 산업기술평가관리원(KEIT) 연구비 지원에 의한 연구임(과제번호: 20018110).

II. 제안 방법

1. Lidar-Inertial Odometry (LIO)

LIO는 IMU의 각속도 및 선형 가속도와 라이다의 포인트 클라우드 정보를 융합하여 로봇의 자세 및 위치를 추정하는 기술이다. 현재 LIO 시스템은 라이다와 IMU 데이터의 융합 방식에 따라 크게 약결합(loosely coupled) 방식과 강결합(tightly coupled) 방식으로 나뉜다.

약결합 방식에서는 각 센서로부터 독립적으로 상태 추정을 수행한 후, 가중치 기반으로 두 결과를 통합한다. 대표적인 예로 LOAM [11]과 LEGO-LOAM [12]은 IMU 데이터를 사용하여 라이다 스캔 왜곡을 보정하고 초기 위치 추정값을 제공한다. 이 방식은 센서 데이터의 등록과 융합 과정을 별도로 수행하여 계산 복잡도를 줄일 수 있지만, 센서 간의 상호 보완적인 정보를 충분히 활용하지 못하는 단점이 존재한다.

반면, 강결합 방식에서는 센서 데이터의 측정 단계부터 서로 융합하여 동시에 상태 추정을 수행한다. 이러한 방식은 크게 최적화 기반 접근법과 필터링 기반 접근법으로 구분될 수 있다. LIO-SAM [13]은 팩터 그래프(factor graph)를 이용하여 IMU preintegration 결과를 라이다 스캔의 de-skewing에 반영하고, 이를 통해 팩터 그래프의 포즈 팩터(pose factor)를 제공한다. FAST-LIO [14]는 반복 칼만 필터(iterated Kalman filter)를 사용하여 라이다와 IMU 데이터를 실시간으로 융합하고, 생성된 맵에 대해 스캔 정합을 효과적으로 수행한다. 이러한 강결합 방식은 센서 간의 깊이 있는 상호보완적 융합을 가능하게 하여 더 정밀한 추정을 제공하지만, 계산량이 많다는 단점이 있다.

FAST-LIO2 [15]는 피처 추출(feature extraction)을 사용하지 않고, 대신 ikd-Tree 자료구조를 활용하여 라이다의 원시 데이터(raw measurements)를 직접 처리함으로써 계산량을 줄이고, 동시에 odometry의 정확도를 향상시켰다. 이 방법을 통해 계산 효율성을 높이면서도 성능을 유지할 수 있었다. 이번 실험에서는 FAST-LIO2를 사용하며 성능을 확인하였다.

2. Odometry Filtering

FAST-LIO2 결과로 얻을 수 있는 Odometry는 정확도가

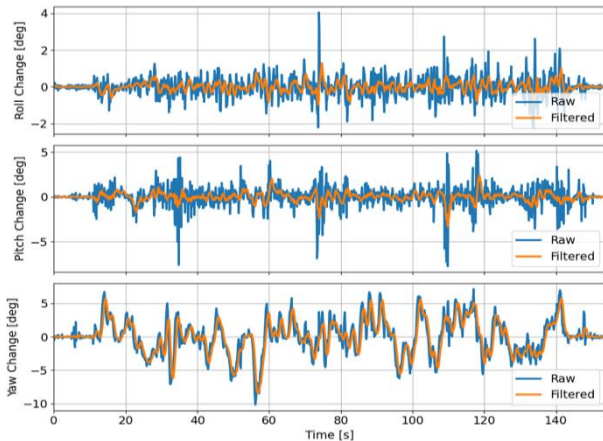


그림 1. FAST-LIO2 Odometry의 시간에 따른 roll, pitch, yaw angle 변화 필터링 후 결과.

Fig. 1. Changes in roll, pitch, and yaw angles over time in the FAST-LIO2 odometry and the results after odometry filtering.

높지만, Odometry roll, pitch, yaw 각도 변화량에는 그림 1과 같이 Oscillation이 존재한다. 비디오 안정화 과정에서 해당 roll, pitch, yaw 각도를 카메라 시스템의 자세로 바로 사용할 경우 성능 저하를 일으킬 수 있다.

따라서 이를 평활화하기 위한 방법으로 이동 평균 필터를 사용하여 FAST-LIO2 결과 Odometry 값에 대한 후처리를 진행하였다. 이동 평균 필터의 특성 상, 윈도우의 크기가 클수록 데이터가 평활화되는 정도는 크지만, 데이터 변화에 반응성이 떨어질 수 있다. 즉, 보다 부드럽게 비디오 안정화가 가능하나 빠른 움직임을 가지는 로봇 시스템 등에서는 반응성이 떨어지는 문제가 나타날 가능성이 있다. 따라서 이를 복합적으로 고려한 윈도우 사이즈 선정이 필요하다.

3. Asynchronous Sensor Data Processing

본 연구에서 제안하는 알고리즘은 라이다, IMU, 카메라의 세 가지 센서의 정보에 주로 의존한다. 라이다와 IMU 센서 값을 통해 FAST-LIO2가 선행되고, FAST-LIO2 결과 Odometry와 카메라의 이미지 정보를 활용하여 비디오 안정화 작업을 수행한다. 이 때, 라이다(10 Hz), IMU (200 Hz) 데이터를 활용한 FAST-LIO2 결과 Odometry는 라이다 센서의 데이터 주기에 맞게 10 Hz로 출력된다. 카메라 센서 주기는 이보다 빠른 20 Hz로, 카메라 이미지가 FAST-LIO2의 Odometry보다 빠르게 갱신되어 중간 프레임 이미지는 건너뛰거나 비디오 안정화에 사용할 수 없게 될 가능성이 존재한다. 이와 같은 문제에 강건하게 대응하기 위하여 LIO 알고리즘 결과 Odometry 쿼터니언 값에 대한 구면 선형 보간법(spherical linear interpolation, SLERP)을 사용한 외삽을 진행하여 중간 이미지 프레임에서의 자세를 계산하였다.

$$\theta = \cos^{-1}(q(t_1) \cdot q(t_2)), \quad t' = \frac{t_3 - t_1}{t_2 - t_1}, \quad (1)$$

$$q(t_3) = \frac{\sin((1-t')\theta)}{\sin(\theta)} q(t_1) + \frac{\sin(t'\theta)}{\sin(\theta)} q(t_2)$$

시간 t_1, t_2, t_3 ($t_1 < t_2 < t_3$)에서의 로봇의 자세 쿼터니언을 각각 $q(t_1), q(t_2), q(t_3)$ 라고 하면 (1)과 같이 t_1, t_2 에서의 쿼터니언을 이용하여 t_3 에서의 쿼터니언을 외삽할 수 있고, 이를 중간 이미지 프레임들의 자세로 비디오 안정화 계산에 활용하였다. 이러한 방식의 알고리즘은 어떠한 센서 조합 Odometry 알고리즘에도 적용될 수 있으며, 이를 통해 이미지 센서 주기에 맞는 안정화 이미지 프레임을 계산할 수 있다.

4. Image Stabilization

실험 환경에서 사용된 로봇의 좌표계는 그림 2와 같다.

FAST-LIO2 Odometry는 IMU 좌표계를 기준으로 하기 때문에 IMU 좌표계를 Reference 좌표계 $R : \{O_R, X_R, Y_R, Z_R\}$ 로,

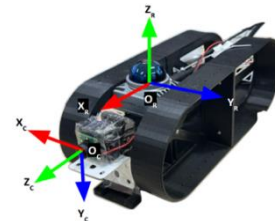


그림 2. 실험에 사용된 로봇의 IMU, 카메라 센서의 좌표계.
Fig. 2. Coordinate systems of the IMU and camera sensors used in the experiment.

카메라를 $C : \{O_C, X_C, Y_C, Z_C\}$ 좌표계로 정의하였다. 로봇이 움직이면서 자세에 따라 두 좌표계는 각각 고정된 지면 좌표계 $W : \{O_W, X_W, Y_W, Z_W\}$ 에 대한 회전이 발생할 수 있다. 이 때, 좌표계의 축이 로봇의 진행방향 및 중력과 나란하도록 좌표계를 회전시킨 수평 방향의 좌표계를 각각 $R_H : \{O_{R_H}, X_{R_H}, Y_{R_H}, Z_{R_H}\}$, $C_H : \{O_{C_H}, X_{C_H}, Y_{C_H}, Z_{C_H}\}$ 로 정의하였다. IMU, 카메라 각각의 수평 방향 좌표계에 대하여 회전 변환만이 작용하였기 때문에 $O_R \equiv O_{R_H}$, $O_C \equiv O_{C_H}$ 의 관계가 성립한다.

FAST-LIO2 결과 Odometry 출력 값을 통해 시작 기준 좌표계에 대한 회전 정보를 쿼터니언 형태로 얻을 수 있다 ($q = q_w + q_x i + q_y j + q_z k$).

$$\phi = \tan^{-1} \left(\frac{2(q_w q_x + q_y q_z)}{1 - 2(q_x^2 + q_y^2)} \right) \quad (2)$$

$$\theta = \sin^{-1} (2(q_w q_y - q_x q_z)) \quad (3)$$

$$\psi = \tan^{-1} \left(\frac{2(q_w q_z + q_x q_y)}{1 - 2(q_y^2 + q_z^2)} \right) \quad (4)$$

(2)-(4)의 식을 사용하여 쿼터니언 값을 x, y, z 축 방향의 회전인 roll(ϕ), pitch(θ), yaw(ψ) 각도로 변환하여 현재 기준 좌표계의 자세 정보를 나타내는데 사용할 수 있다. 앞선 기준 좌표계와 수평 방향 좌표계 사이의 관계를 고려하여, 초기 roll, pitch 각도를 0으로 가정하면,

$$\mathbf{R}_R^{R_H} = \mathbf{R}_z(\psi) \mathbf{R}_y(\theta) \mathbf{R}_x(\phi) = \mathbf{R}_y(\theta) \mathbf{R}_x(\phi) \quad (5)$$

수평 좌표계의 정의에 의하여 $\psi = 0$ 이므로 기준 좌표계와 그에 해당하는 수평 좌표계 사이의 회전 행렬을 (5)와 같이 구할 수 있다.

비디오 안정화를 위해, 대부분의 방식은 넓은 화각의 카메라를 사용한 뒤 이를 잘라내어 안정화된 프레임을 생성한다. 해당 연구에서도 먼저 카메라 렌즈 왜곡이 있는 넓은 화각의 이미지를 보정하여 왜곡이 없는 새로운 이미지로 변환한다.

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (6)$$

기존의 카메라의 내부 파라미터 행렬은 (6)와 같이 x, y 방향 초점거리 f_x, f_y 와, 이미지 중심 좌표 c_x, c_y 로 정의된다. 카메라 캘리브레이션 과정[16-17]을 통해 이와 같은 행렬을 구할 수 있다. 카메라 렌즈 왜곡을 보정하기 위해서는 (7), (8)과 같이 왜곡된 각 픽셀의 좌표 (x, y) 를 보정된 좌표 $(x_{undistorted}, y_{undistorted})$ 로 변환해야 한다.

$$x_{undistorted} = x \cdot (1 + k_1 \cdot r^2 + k_2 \cdot r^4 + k_3 \cdot r^6) + (2p_1 \cdot x \cdot y + p_2 \cdot (r^2 + 2x^2)) \quad (7)$$



그림 3. 이미지 왜곡 보정 전(좌), 후(우).

Fig. 3. Image distortion correction, before (left) and after (right).

$$y_{undistorted} = y \cdot (1 + k_1 \cdot r^2 + k_2 \cdot r^4 + k_3 \cdot r^6) + (p_1 \cdot (r^2 + 2y^2) + 2p_2 \cdot x \cdot y) \quad (8)$$

이때 $r^2 = x^2 + y^2$ 이고, k_1, k_2, k_3 는 방사 왜곡 계수, p_1, p_2 는 접선 왜곡 계수를 의미하며 카메라 캘리브레이션 과정을 통해 구할 수 있다. 왜곡이 보정된 좌표를 사용하여 새로운 카메라 내부 파라미터 행렬 \mathbf{K}' 을 구할 수 있다(9).

$$\mathbf{K}' = \begin{bmatrix} f'_x & 0 & c'_x \\ 0 & f'_y & c'_y \\ 0 & 0 & 1 \end{bmatrix} \quad (9)$$

그림 3과 같이 왜곡이 없는 이미지를 생성한 뒤, 이를 센서 간의 기하학적 관계를 고려하여 수평 방향 좌표계로의 변환이 필요하다. 해당 과정은 [18]과 동일한 방식으로 진행하였다. 기존 이미지 평면 I 에서 이미지 픽셀의 동차 좌표계 표현 $\mathbf{p}'_I = [u_i \ v_i \ 1]^T$ 과 해당 픽셀에 해당하는 3차원 좌표의 동차 좌표계 표현 $\mathbf{p}'_W = [x_W \ y_W \ z_W \ 1]^T$ 사이에는 (10)의 관계식이 성립한다.

$$\mathbf{p}'_I = \mathbf{K} [\mathbf{R}_W^C |^C t_W] \mathbf{p}'_W \quad (10)$$

\mathbf{K} 는 카메라 내부 파라미터 행렬을, $[\mathbf{R}_W^C |^C t_W] \in \mathbb{R}^{3 \times 4}$ 은 고정된 지면 좌표계와 카메라 좌표계 사이의 회전 ($\mathbf{R}_W^C \in \mathbb{R}^{3 \times 3}$)과 병진($^C t_W \in \mathbb{R}^{3 \times 1}$)을 결합한 결합 행렬을 의미한다. 마찬가지로 이를 수평 카메라 좌표계에 적용하면 (11)과 같다.

$$\mathbf{p}'_{I_H} = \mathbf{K} [\mathbf{R}_W^{C_H} |^{C_H} t_W] \mathbf{p}'_W \quad (11)$$

$\mathbf{R} \in SO(3)$ 인 회전행렬 \mathbf{R} 에 대하여 (12)의 관계식이 성립하고,

$$\begin{aligned} \mathbf{R}_A^B &= \mathbf{R}_A^T = \mathbf{R}_B^A, \\ \mathbf{R}_A^B \mathbf{R}_B^C &= \mathbf{R}_A^C \end{aligned} \quad (12)$$

이를 이용하면,

$$\begin{aligned} \mathbf{p}'_W &= [\mathbf{R}_C^{C_H} \mathbf{R}_W^{C_H} |^{C_H} t_W]^{-1} \mathbf{K}^{-1} \mathbf{p}'_I \\ &= [\mathbf{R}_W^{C_H} \mathbf{R}_C^{C_H} |^{C_H} t_W]^{-1} \mathbf{K}^{-1} \mathbf{p}'_I \\ &= [\mathbf{R}_C^{C_H} \mathbf{R}_W^{C_H} |^{-C_H} t_W] \mathbf{K}^{-1} \mathbf{p}'_I. \end{aligned} \quad (13)$$

(11)에 (13)를 대입하면,

$$\begin{aligned} \mathbf{p}'_{I_H} &= \mathbf{K} [\mathbf{R}_W^{C_H} |^{C_H} t_W] [\mathbf{R}_C^{C_H} \mathbf{R}_W^{C_H} |^{-C_H} t_W] \mathbf{K}^{-1} \mathbf{p}'_I \\ &= \mathbf{K} [\mathbf{R}_W^{C_H} \mathbf{R}_C^{C_H} |^{C_H} t_W -^{C_H} t_W] \mathbf{K}^{-1} \mathbf{p}'_I \\ &= \mathbf{K} [\mathbf{R}_C^{C_H} | 0] \mathbf{K}^{-1} \mathbf{p}'_I. \end{aligned} \quad (14)$$

따라서, 기존 이미지 평면 I 의 동차 좌표 \mathbf{p}'_I 와 수평 이미지 평면 I_H 의 동차 좌표 \mathbf{p}'_{I_H} 사이의 호모그래피 변환 행렬은 $\mathbf{H}'_{I_H} = \mathbf{K} [\mathbf{R}_C^{C_H} | 0] \mathbf{K}^{-1}$ 로 정의할 수 있고, $\mathbf{R}_C^{C_H}$ 는 카메라 좌표계와 수평 카메라 좌표계 간의 회전 변환을 의미한다.

$$\mathbf{R}_C^{C_H} = \mathbf{R}_{R_H}^{C_H} \mathbf{R}_R^R \mathbf{R}_C^R \quad (15)$$

$$= (\mathbf{R}_C^R)^T \mathbf{R}_R^{R_H} \mathbf{R}_C^R \quad (16)$$



그림 4. 비디오 안정화 알고리즘 적용 예시 전(좌), 후(우).

Fig. 4. Stabilized image example, before (left) and after (right).

R_C^{CH} 는 회전 행렬의 특성을 활용하면 (15)와 같이 풀어서 표현할 수 있다. IMU와 카메라가 강제로 연결되어 있으므로, IMU 수평 좌표계와 카메라 수평 좌표계 간의 회전행렬은 IMU 좌표계와 카메라 좌표계 간의 회전행렬로 (16) 나타낼 수 있다 ($R_{RH}^{CH} = R_R^C$). IMU(라이다) 좌표계와 카메라 좌표계 간의 회전행렬은 외부 파라미터 캘리브레이션 과정을 통해 계산할 수 있다[19-21]. 따라서 IMU-카메라 회전 행렬과 (5)를 이용하여 (16), (14)를 계산할 수 있고, 원본 이미지를 수평 좌표계로 변환이 가능하며 그 결과는 그림 4와 같다.

III. 실험 및 결과

1. 실험 환경

실험은 라이다-카메라 리그 장치를 탑재한 협소공간 주행 로봇(그림 5)을 활용하여 진행되었다. 총 세 개의 환경에서의 성능을 검증하였으며(그림 6), 두 개의 저조도-협소 실험 환경 한 개의 Hand-held 실험에서의 비디오 안정화 성능을 평가하였다. 저조도-협소 실험 환경은 암막으로 조명이 완전히 차단된 암실에서 카메라 전방에 부착된 조명만에 의존하며 협소한 공간을 주행하는 시나리오이며, Hand-held 실험은 일반적인 조명 환경에서 센서 리그를 직접 들고 복도를 이동하는 시나리오이다. 구체적인 실험 환경 및 데이터 구성은 표 1과 같다.

라이다-카메라 리그 시스템의 구성으로 라이다는 Livox사의 MID360 라이다를 사용하였고, 카메라는 RasPi-CAM과 e-CAM24_CUNX, 150도 대각 시야각을 갖는 렌즈를 사용하였다. 이를 처리하는 임베디드 보드는 Nvidia Jetson Orin NX 보드를 사용하였고, 센서 데이터 드라이버와 비디오 안정화 코드는 ROS 기반 소프트웨어로 작성되었다.

2. 평가 방법

비디오 안정화의 경우 안정화 대상 영상에 대한 정확한 정답 값을 갖고 있는 경우가 드물다. 카메라 별 렌즈의 시야각도 다르고, 안정화된 결과의 경우 절대적인 Roll, Pitch, Yaw가 제시되기가 어렵기 때문이다. 또한 안정화 알고리즘은 실제 카메라 시스템 자세를 추종하면서 작동하는데, 추종성을 높이면 반대로 안정성이 떨어져 때문에 영상 안정화 결과에 대한 평가는 다소 주관적일 수 있다. 따라서 가공된 이미지의 변화나 가공의 정도를 상대적인 지표로 평가하는 방법이 주를 이루고 있다.

대부분의 비디오 안정화 연구에서는 영상을 안정화하기 전에 취득된 카메라 포즈 데이터의 필터링의 정도를 정성적으로 보고, 안정화된 이미지의 잘림 영역의 정도(cropping ratio)와 이미지 뒤틀림 정도(distortion value), 이전 프레임과의



그림 5. 실험에 사용된 협소공간 주행로봇.
Fig. 5. Narrow-space navigation robot used in the experiment.



그림 6. 저조도-협소 및 Hand-held 실험 환경.
Fig. 6. Low-light and narrow-space, and hand-held experimental environments.

표 1. 실험 환경 및 데이터 구성.

Table 1. Experimental setup and data composition.

| | Env 1 | | Env 2 | Env 3 |
|---------------|----------|----------|-----------|----------|
| 카메라 | e-CAM | RasPi | e-CAM | e-CAM |
| 카메라 해상도 (WxH) | 1280x720 | 1280x720 | 1920x1080 | 1280x720 |
| 카메라 화각 (°) | 120 | 155 | 155 | 155 |
| 카메라 주기 (Hz) | 17 | 30 | 30 | 17 |
| 경로 길이 평균 (m) | 9.24 | 25.80 | 9.73 | 101.09 |
| 시퀀스 길이 평균 (s) | 106.16 | 240.59 | 123.40 | 168.71 |

비교에서 픽셀 단위 변화 정도(stability score)를 이용하여 평가한다[22].

한편, 특징점 추출이나 딥러닝 기반의 방식을 사용한 비디오 안정화 알고리즘은 소형 로봇의 컴퓨터 성능으로 계산하기 어려운 경우가 많다. 따라서 본 연구에서는 비디오 안정화 알고리즘의 계산 시간을 고려한 성능 점수(total score)를 제안한다. 성능 점수는 **(크롭 비율 점수 + 왜곡 점수 + 안정성 점수) / (1 프레임 이미지 처리 알고리즘 소요 시간)**으로 계산할 수 있으며, 이 수치가 높을수록 낮은 컴퓨터 성능으로 높은 안정화 성능을 달성하는 알고리즘으로 이해할 수 있다.

3. 실험 결과

- 정량적 성능 분석

본 연구에서는 비디오 안정화 기법의 성능을 기존의 특징점 매칭 기반의 방식[4]과 비교하였다(표 2). 해당 정량 성능 비교는 사전 취득된 데이터를 PC에서 테스트하는 방식으로

표 2. 세 가지 실험 환경에 대한 정량 성능 평가 결과.

Table 2. Quantitative performance evaluation results for the three experimental environments.

| Camera | Env 1 | | Env 2 | | Env 3 | | | |
|----------------------|--------------|--------------|-----------|--------------|--------------|--------------|--------------|--------------|
| | e-CAM24_CUNX | | RasPi-CAM | | e-CAM24_CUNX | | e-CAM24_CUNX | |
| Metric / Method | Feature | Odometry | Feature | Odometry | Feature | Odometry | Feature | Odometry |
| Cropping Ratio (↑) | 0.943 | 1 | 0.96 | 1 | 0.966 | 1 | 0.954 | 1 |
| Distortion Value (↑) | 0.451 | 0.911 | - | 0.854 | - | 0.812 | 0.996 | 0.812 |
| Stability Score (↑) | 0.777 | 0.846 | 0.568 | 0.618 | 0.884 | 0.731 | 0.576 | 0.463 |
| Time (ms) (↓) | 11.544 | 1.582 | 11.901 | 1.527 | 26.509 | 1.478 | 11.188 | 2.985 |
| Total Score (↑) | 0.188 | 1.743 | 0.128 | 1.619 | 0.07 | 1.721 | 0.226 | 0.762 |

진행되었다. 각 환경의 데이터셋은 30초 정도 길이를 가지며, 환경별로 3개의 실험 결과를 평균 낸 값으로 성능을 측정하였다.

저조도-협소 환경 1에서는 Odometry 기반의 방식이 특징점 매칭 기반의 방식에 비해 모든 점수에서 앞서는 결과를 확인할 수 있다. 그림 6에서 확인할 수 있듯, 조도가 낮고 바닥에 턱이 존재하여 소형 로봇 주행 시 많은 흔들림이 발생하게 된다. 이와 같은 환경에서 특징점 추출 기반의 방식은 순간적으로 큰 움직임이 발생하였을 때 특징점 매칭에 어려움을 겪을 수 있으며, 저조도 환경 또한 특징점 추출을 어렵게 할 수 있다. 한편, 두 개의 카메라 센서에서 모두 Odometry 기반의 방식이 특징점 매칭 방식에 비해 좋은 성능을 보인다. 이는 본 논문에서 제시하는 비디오 안정화 알고리즘의 성능이 센서에 크게 의존하지 않다는 것을 의미한다.

저조도-협소 환경 2에서는 안정성 점수를 제외한 평가 점수에서 Odometry 기반 방식이 높은 성능을 나타냈다. 환경 1과는 다르게 환경 2에서는 바닥에 턱이 존재하지 않아 로봇에 순간적으로 큰 회전이 발생하지 않는다. 이러한 특성으로 인해 특징점 매칭 기반의 방식이 Odometry 기반의 비디오 안정화 기법보다 높은 Stability Score를 기록한 것으로 볼 수 있다. 그러나 알고리즘 소요 시간 측면에서 Odometry 기반의 방식이 대략 18배 정도 빠르며, 이는 최종 성능 점수에도 큰 차이를 발생시킨다.

일반적인 조도 환경의 Hand-held 데이터는 환경 3은 특징점 추출 및 매칭이 비교적 용이한 환경으로, 특징점 매칭 기반의 방식이 Odometry 기반의 방식에 비해 Stability Score가 높은 것을 확인할 수 있다. 그러나 앞선 2개의 환경과 마찬가지로 Odometry 기반의 방식이 짧은 계산 시간으로 인해 최종적으로 높은 성능 점수를 기록하였다. 종합적으로, 제안하는 Odometry 기반의 비디오 안정화 알고리즘이 저조도-협소 극한 환경에서 기존 방식[4]의 성능을 뛰어넘는 결과를 나타낸다.

- 컴퓨팅 성능 분석

비디오 안정화 기법에 따른 계산 시간 및 CPU 사용률은 표

표 3. 비디오 안정화 기법에 따른 온보드 계산 시간 및 CPU 사용률.

Table 3. Computation time and CPU usage (onboard).

| | 계산 시간 (ms) | CPU 사용률 |
|------------------|---------------|---------|
| Feature Matching | 94.741 | 약 140% |
| Odometry | 30.495 | 약 120% |

3과 같다. 이때, 성능 비교는 Jetson Orin NX 온보드 상에서 진행되었다. 계산 시간은 이미지 100 프레임에 대한 평균 처리 속도를 의미하며, CPU 사용률은 6코어 600% CPU를 기준으로 한다. LIO의 Odometry 결과값을 받아 비디오 안정화를 수행하는 알고리즘의 특성 상, 이미지에서 특징점을 뽑고 매칭하는 방식에 비해 계산시간이 3배 이상 단축된다. 이는 딥러닝을 사용한 저지연 비디오 안정화 알고리즘을 복잡한 연산 처리에 강점을 가진 NVIDIA RTX 2080 Ti GPU에서 테스트한 결과[10] (대략 26 ms)와 대등한 프레임 처리 속도이다. 비록 LIO 알고리즘 자체의 계산 시간이 존재하나, Localization 알고리즘으로 LIO를 기존에 사용하고 있는 소형 로봇에서는 비디오 안정화에 사용되는 계산 시간을 크게 단축시켜 보다 부드러운 비디오 안정화가 가능하다 또한, CPU 사용률은 20% 가량 감소한다. CPU 사용률에 계산 시간만큼 큰 개선이 없는 것은 이미지 변환에 사용되는 행렬 연산에 의한 것으로, 특징점 추출을 제외하면서 추출 과정에서 발생하는 CPU 병목 현상을 제거하여 CPU 점유율이 대비 연산 속도로 상승으로 이어져, 이는 적절하게 연산 처리 속도를 조정함으로써 CPU 점유율을 줄일 수 있는 가능성을 확보하였다.

IV. 결론

본 연구에서는 LIO를 사용한 비디오 안정화 알고리즘을 제안한다. 특히, 조도가 낮고 협소한 극한 환경에서의 주행 로봇에서의 사용 가능성에 초점을 맞추고 있다. 3 가지 실제 환경에서의 테스트를 통해 알고리즘의 성능을 확인하였으며, 기존의 특징점 기반 비디오 안정화 기법[4]보다 극한 환경에서의 비디오 안정화 성능이 개선되었다.

LIO Odometry 결과를 이동 평균 필터를 통해 일차적으로 평활화 과정을 거친 뒤, 쿼터니언 간의 구면 선형 보간법을 통해 외삽을 진행함으로써 카메라 센서의 주기와 LIO 알고리즘의 계산 주기에 강건하게 알고리즘이 동작할 수 있도록 구성하였다. 실제 두 가지 카메라 센서로 진행된 실험에서의 성능을 통해 강건성을 검증하였다.

비디오 안정화에 사용될 현재 프레임의 보정된 카메라 자세 값을 얻은 뒤에는, 이를 가상의 수평 카메라 평면으로 변환하는 과정에 활용하여 비디오 안정화를 진행하였다. 이 과정을 통해 얻은 수평 이미지는 협소한 공간에서 로봇의 회전 움직임이 큰 경우 조작자에게 보다 나은 인지 상황을 제공할 수 있고, 이렇게 가공된 안정화된 이미지는 객체 검출 등의 이미지 기반 기술의 성능을 높이는 것에도 활용될 수 있다.

추가적으로, 제안하는 방식은 LIO 이외의 Odometry 알고리즘 과도 결합될 수 있으며, 비교적 적은 시스템 리소스 투자를 통해 비디오 안정화가 가능하다는 확장 가능성이 존재한다.

REFERENCES

- [1] J. Kim, H. Kim, J. Oh, X. Li, K. Jang, and H. Kim, "Distance measurement of tunnel facilities for monocular camera-based localization," *Journal of Institute of Control, Robotics and Systems (in Korean)*, vol. 29, no. 1, pp.7-14, Jan. 2023.
doi: 10.5302/JICROS.2023.22.0203
- [2] D. Lee, J. Kim, J. Oh, X. Li, C. Ryu, and H. Kim, "Precise compensation of estimated distance by pose estimation using camera-inu sensor fusion," *Journal of Institute of Control, Robotics and Systems (in Korean)*, vol. 28, no. 12, pp. 1161-1169, Dec. 2022.
doi: 10.5302/JICROS.2022.22.0133
- [3] K. Y. Lee, Y. Y. Chuang, B. Y. Chen, and M. Ouhyoung, "Video stabilization using robust feature trajectories," *2009 IEEE 12th International Conference on Computer Vision*, vol. 2009, pp. 1397-1404, Sep. 2009.
doi: 10.1109/ICCV.2009.5459297.
- [4] L. Kejrival and I. Singh, "A hybrid filtering approach of digital video stabilization for UAV using Kalman and low pass filter," *Procedia Computer Science*, vol. 93, pp. 359-366, 2016.
doi: <https://doi.org/10.1016/j.procs.2016.07.221>.
- [5] Y. Matsushita, E. Ofek, W. Ge, X. Tang, and H.-Y. Shum, "Full-frame video stabilization with motion inpainting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 7, pp. 1150-1163, Jul. 2006.
doi: 10.1109/TPAMI.2006.141.
- [6] Z. Peng, X. Ye, W. Zhao, T. Liu, H. Sun, B. Li, and Z. Cao, "3D multi-frame fusion for video stabilization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2024, pp. 7507-7516, Jun. 2024.
- [7] C. Li, L. Song, S. Chen, R. Xie, and W. Zhang, "Deep online video stabilization using IMU sensors," *IEEE Transactions on Multimedia*, vol. 25, pp. 2047-2060, 2022.
doi: 10.1109/TMM.2022.3142429.
- [8] Z. Shi, F. Shi, W.-S. Lai, C.-K. Liang, and Y. Liang, "Deep online fused video stabilization," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, vol. 2022, pp. 1250-1258, Jan. 2022.
- [9] J. Yu and R. Ramamoorthi, "Learning video stabilization using optical flow," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2020, pp. 8159-8167, Jun. 2020.
- [10] Z. Zhang, Z. Liu, P. Tan, B. Zeng, and S. Liu, "Minimum latency deep online video stabilization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, vol. 2023, pp. 23030-23039, Oct. 2023.
- [11] J. Zhang and S. Singh, "LOAM: Lidar odometry and mapping in real-time," *Robotics: Science and Systems*, vol. 2, no. 9, pp. 1-9, Jul. 2014.
- [12] T. Shan and B. Englot, "LeGO-LOAM: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 2018, pp. 4758-4765, Madrid, Spain, 2018.
doi: 10.1109/IROS.2018.8594299.
- [13] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "LIO-SAM: Tightly-coupled lidar inertial odometry via smoothing and mapping," *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 2020, pp. 5135-5142, Las Vegas, NV, USA, 2020.
doi: 10.1109/IROS45743.2020.9341176.
- [14] W. Xu and F. Zhang, "FAST-LIO: A fast, robust LiDAR-inertial odometry package by tightly-coupled iterated Kalman filter," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3317-3324, Apr. 2021.
doi: 10.1109/LRA.2021.3064227.
- [15] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "FAST-LIO2: Fast direct LiDAR-inertial odometry," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053-2073, Aug. 2022.
doi: 10.1109/TRO.2022.3141876.
- [16] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330-1334, Nov. 2000.
doi: 10.1109/34.888718.
- [17] J. Kannala and S. S. Brandt, "A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 8, pp. 1335-1340, Aug. 2006.
doi: 10.1109/TPAMI.2006.153.
- [18] M. Odelga, N. Kochanek, and H. H. Bühlhoff, "Efficient real-time video stabilization for UAVs using only IMU data," *2017 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*, vol. 2017, pp. 210-215, 2017.
doi: 10.1109/RED-UAS.2017.8101668.
- [19] C. Yuan, X. Liu, X. Hong, and F. Zhang, "Pixel-level extrinsic self-calibration of high-resolution LiDAR and camera in targetless environments," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7517-7524, Oct. 2021.
doi: 10.1109/LRA.2021.3098923.
- [20] L. Zhou, Z. Li, and M. Kaess, "Automatic extrinsic calibration of a camera and a 3D LiDAR using line and plane correspondences," *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 2018, pp. 5562-5569, Madrid, Spain, 2018.
doi: 10.1109/IROS.2018.8593660.
- [21] G. Pandey, J. McBride, S. Savarese, and R. Eustice, "Automatic targetless extrinsic calibration of a 3D LiDAR and camera by maximizing mutual information," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 26, no. 1, pp. 2053-2059, 2012.
doi: <https://doi.org/10.1609/aaai.v26i1.8379>.
- [22] J. Choi and I. S. Kweon, "Deep iterative frame interpolation for full-frame video stabilization," *ACM Transactions on Graphics*, vol. 39, no. 1, Feb. 2020.
doi: <https://doi.org/10.1145/3363550>.



윤영인

2022년 서울대학교 기계항공공학부 졸업. 2022년~현재 서울대학교 대학원 기계공학과 석박통합과정 재학 중. 관심분야는 로보틱스, 자율 주행, 모션 추정.



서형석

2024년 서울대학교 기계공학부 졸업. 2024년~현재 서울대학교 대학원 기계공학과 석사과정 재학 중. 관심분야는 SLAM, 로보틱스.



신관준

2022년 대구경북과학기술원 기초학부 졸업. 2022년~현재 서울대학교 대학원 협동과정 인공지능전공 석박통합과정 재학 중. 관심분야는 로보틱스, 자율 주행, 딥러닝.



임종우

1997년 서울대학교 컴퓨터공학부(학사). 2003년 University of Illinois at Urbana-Champaign (UIUC), Computer Science (M.S.). 2005년 University of Illinois at Urbana-Champaign (UIUC), Computer Science (Ph.D.). 2012년~2023년 한양대학교 컴퓨터공학부 교수. 2023년~현재 서울대학교 기계공학부 교수. 관심분야는 컴퓨터 비전, 로보틱스, 머신러닝.