

차량의 모션계측을 위한 RANSAC 의존 없는 스테레오 영상 거리계

Stereo Visual Odometry without Relying on RANSAC for the Measurement of Vehicle Motion

송 광 열, 이 준 응*
(Gwang-Yul Song¹ and Joon-Woong Lee^{1,*})

¹Department of Industrial Engineering, Chonnam National University

Abstract: This paper addresses a new algorithm for a stereo visual odometry to measure the ego-motion of a vehicle. The new algorithm introduces an inlier grouping method based on Delaunay triangulation and vanishing point computation. Most visual odometry algorithms rely on RANSAC in choosing inliers. Those algorithms fluctuate largely in processing time between images and have different accuracy depending on the iteration number and the level of outliers. On the other hand, the new approach reduces the fluctuation in the processing time while providing accuracy corresponding to the RANSAC-based approaches.

Keywords: visual odometry, feature detection, 3D-to-2D motion estimation, inlier grouping, stereo vision

I. 서론

연속적인 입력영상을 통해 카메라의 모션을 파악하는 것을 영상 거리계(VO: Visual Odometry)라 한다[1]. 여기에서 모션은 자세변환과 위치변환으로 묘사된다. 따라서 VO를 구현하면 카메라가 장착된 차량이나 로봇 등의 자세와 이동경로를 추종할 수 있다. VO는 차량이나 로봇과 같은 모바일 에이전트(mobile agent)의 자세와 이동경로뿐만 아니라 이들 에이전트와 다른 운동을 하는 물체의 검출 및 이전 시점의 영상과 현 시점의 영상의 융합 등에 활용된다.

VO라는 용어는 Nister 등의 논문 "Visual Odometry" [3]에 의해 처음 등장했다. 이 용어는 일정 시간에 걸쳐 휠의 회전수를 누적시킴으로써 차량의 움직임을 점차적으로 계측하는 휠 오도메트리(wheel odometry)에 대한 유사성으로 선택된 것이다. VO도 차량의 움직임이 차량에 장착된 카메라의 영상에 야기시키는 변화의 분석을 통해 차량의 자세를 점차적으로 계측함으로써 작동한다. VO가 효과적으로 작동하려면 대상 환경에 충분한 조명과 분명한 움직임이 추출될 수 있을 만큼 충분한 텍스처를 가진 정적 장면(static scene)이 있어야 한다. 또한 연속적인 영상 프레임들이 충분한 장면 겹침을 가지고 있어야 한다. 현재 발표된 VO 알고리즘들은 전자의 환경적 영향은 여전히 해소하지 못했지만, 하드웨어의 발달에 힘입어 처리속도를 높일 수 있어서 후자의 영향은 상당부분 극복하였다.

차량의 모션을 계측하는 연구로 앞서서도 언급한 휠 거리계가 있고, GPS (Global Positioning System), IMU (Inertial

Measurement Unit), 레이저 스캐너 기반의 거리계[16]도 있다. 이들 각각은 모두 장단점이 있다. 휠 거리계는 지형이나 바퀴 미끄러짐 등의 영향을 받고, GPS 기반 거리계는 신호가 단절되는 음영지역이나 낮은 신호 입력 주기로 인해 급격한 자세변화가 발생한 상황에서는 모션을 추종하기 어려운 단점이 있으며, IMU 기반 거리계의 경우 가속도량의 누적오차가 크다는 단점이 있다. 레이저 스캐너 기반 거리계는 VO와 유사하게 연속된 레이저 스캐너 데이터의 매칭을 통해 움직임을 계측한다. 최근에는 VO와 다른 센서 기반의 거리계를 융합하는 연구결과도 발표되고 있다[8-10].

VO는 단안 VO와 스테레오 VO로 나뉘며[1], 지금까지의 대부분의 연구가 그렇듯 본 논문도 스테레오 VO 문제를 다룬다. 단안 VO에 비해 스테레오 VO가 많이 행해진 이유는 상대적으로 3차원 데이터를 더 쉽게 계산할 수 있기 때문이다. 참고논문 [1]에 따르면 연속된 두 영상 I_{t-1} 과 I_t 사이에서 대응되는 특징점들이 2차원 영상에서 규정되느냐 3차원에서 규정되느냐에 따라 모션계측은 세 가지로 분류되는데 1) 2D-to-2D, 2) 3D-to-3D, 3) 3D-to-2D가 그것이다. 각각의 특징은 참고문헌 [1]에 잘 기술되어 있다. 여기에서 2D는 영상에서의 점(point)을 의미하고, 3D는 공간상의 3차원 점을 의미한다. 본 연구는 3D-to-3D와 3D-to-2D를 혼용하는 모션 계측법에 바탕을 두고 있다.

VO 연구에서 가장 까다로운 부분이 아웃라이어(outlier) 제거다[2]. 단안 VO든 스테레오 VO든 대부분의 VO는 연속된 영상에서 특징점(point feature)을 추출하고 이 특징점들의 전후 영상간 대응관계(correspondence)를 통해 모션을 계측한다. 그러나 대응점들 가운데는 시점(viewpoint)변화, 조명변화, 스케일(scale)변화, 자세변화, 모션블러(motion blur) 등 여러 요인에 의해 아웃라이어(outlier)가 포함될 수 밖에 없고, 이러한 아웃라이어들은 정확한 모션계측을 심각하게 방해한다. 특히 본 연구는 스테레오 VO를 추구하고 있기 때문에 전후 영상

* Corresponding Author

Manuscript received October 1, 2014 / revised November 18, 2014 / accepted December 16, 2014

송광열, 이준응: 전남대학교 산업공학과

(skyclass@nate.com/joonlee@chonnam.ac.kr)

* 이 논문은 2014년도 전남대학교 학술연구비 지원에 의하여 연구되었음.

간의 오매칭(miss matching)뿐만 아니라 좌우 영상간의 오매칭도 이 문제를 증폭시킨다. 정확한 모션계측을 위해 이와 같은 아웃라이어를 제거하기 위한 여러 방법들이 있지만, 그 가운데 표준적인 것이 RANSAC (RANDOM SAmple Consensus) [4]이다. RANSAC의 취지는 데이터 점들 가운데 임의의 (random) 취해진 집합들로부터 모델 가설들(model hypotheses)을 계산하고 남은 다른 데이터 점들로 이 가설들이 정확한지 확인하는 것이다. 이 확인 과정에서 가장 큰 합의(consensus)를 이루는 가설, 즉 인라이어(inlier) 수를 가장 많이 갖는 가설을 해(solution)로 선택한다. 그러나 RANSAC은 모델 가설을 위한 반복(iteration) 횟수에 따라 정확도와 처리시간이 달라진다[2]. 대개 반복횟수가 증가하면 정확도는 높아지지만 처리시간이 길어지므로 영상마다 처리시간 예측이 어렵다는 단점이 있다. 특히, 잡음이 많은 영상에서는 아웃라이어들의 수도 많기 때문에 처리시간이 더욱 길어진다. 반복 횟수 N 은 다음 식으로 결정된다.

$$N = \frac{\log(1-p)}{\log(1-(1-\epsilon)^s)} \quad (1)$$

여기에서 s 는 모델이 설명될 수 있는 데이터 점의 수이고, ϵ 은 데이터 점들 속에 있는 아웃라이어들의 비율이며, p 는 성공확률로서 N 번 반복중에서 적어도 하나는 참값이 되는 비율인데, 대개 0.95나 0.99로 설정된다. 3D-to-2D 모션 계측법에서 s 는 3이다. RANSAC 구현시 매 시행마다 데이터 점들을 인라이어와 아웃라이어로 분류하여 이전 시행보다 인라이어의 수가 많은 모델을 저장하고, ϵ 을 새로 계산하여 반복횟수 N 을 갱신한다.

RANSAC의 반복횟수를 줄이기 위한 방법으로 Nguyen 등이 제안한 한 점(one point) RANSAC을 들 수 있다[5]. 본 연구는 아예 RANSAC에 의존하지 않고 차량의 모션을 계속할 수 있는 스테레오 VO를 제안한다. 제안된 방법은 RANSAC에 의한 방법보다 처리시간 변동폭을 크게 줄이는 장점을 갖는데, 이 점이 RANSAC에 의한 방법과 다른 점이다. 제안된 방법에서는 아웃라이어의 영향을 줄이기 위해 순환매칭(cyclic matching)법을 이용하여 영상간 매칭 오류가 있는 픽셀들을 제거하고, 델로나이 삼각형화(DT: Delaunay Triangulation) [11]를 이용하여 추출된 특징점들을 인라이어와 아웃라이어로 나누는 근거를 제시한다. 이어지는 논문의 구성은 다음과 같다. 단원 II는 제안된 VO 알고리즘의 구성, 단원 III은 영상의 특징점 추출과정, 단원 IV는 논문에서 제안한 모션계측 과정을 설명한다. 단원 V는 실험결과, 그리고 단원 VI에서 결론을 맺는다.

II. 제안된 VO 알고리즘의 구성

그림 1은 제안된 알고리즘의 전체적인 흐름을 보인 것이다. 알고리즘 시작시점에서 입력된 좌우 한 쌍으로 이루어진 스테레오 영상 각각에 Harris 코너 검출기[6]를 적용하여 특징점을 추출하고, 좌측 영상을 기준으로 좌우 영상간에 추출된 특징점들의 매칭을 실시한다. 이후 입력된 좌우 한 쌍의 현 스테레오 영상마다 좌측 영상은 좌측 영상끼리 우측 영상은 우측 영상끼리 각각 이전 영상의 특징점들과 대응이 이루어

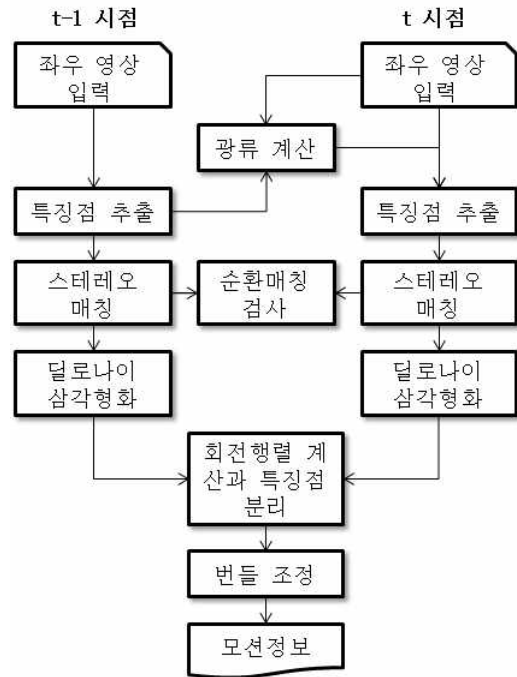


그림 1. VO 알고리즘 구성.

Fig. 1. Structure of the proposed VO algorithm.

어지는 점(pixel)을 광류(optical flow) 계산을 통해 찾는다. 이와 같이 광류 계산을 통해 이전 영상의 특징점과 대응이 이루어진 현재 영상의 픽셀을 현재 영상의 특징점으로 한다. 이 특징점들을 대상으로 현재의 좌우 영상간 스테레오 매칭을 실시한다. 이와 같이 좌우 영상은 스테레오 매칭을 하고 전후 영상은 광류를 통한 매칭을 하므로 이전 시점의 좌측 영상에서 현재 시점의 우측 영상에 도달하는 길은 두 가지가 있다. 이 두 길을 통해 우측 영상에 도달한 지점이 동일 지점인지를 검사한다. 이 검사를 순환매칭 검사라 한다. 이 검사에서 동일 지점에 이르지 못한 이전 영상의 특징점은 모션계측 대상에서 배제한다.

이전 영상과의 광류계산에 의해 검출된 특징점들만으로는 차량의 모션으로 인하여 영상이 바뀌었기 때문에 현재영상의 특징점들을 다 설명하지 못한다. 따라서 현재의 스테레오 영상에 Harris 코너 검출기를 적용하여 코너점들을 찾되 광류계산으로 얻어진 특징점에 가까운 코너점들은 제거하고 남은 코너점들을 특징점 집합에 포함시킨다.

다음은 순환매칭 검사를 통과한 이전 시점의 왼쪽 영상의 특징점들과 이들에 대응된 현 영상의 왼쪽 영상의 특징점들을 대상으로 영상마다 DT를 적용하여 삼각형망을 형성한다. 이전 영상과 현 영상 사이에 서로 대응되는 삼각형들의 꼭지점의 3차원 점을 계산한 후, 이전 영상의 3개의 3차원 점을 현 영상의 3차원 점으로 변환시킨 회전행렬을 추정하고, 이 회전행렬을 이용하여 특징점들을 인라이어와 아웃라이어로 분리한다. 이 분리는 본 논문에서 제안하는 소실점 보팅(vanishing point voting) 기법에 의해 이루어진다. 즉, 회전행렬이 동일하면 서로 다른 3차원 점들은 동일한 소실점을 갖는다. 이 원리를 토대로 보팅이 가장 많이 이루어진 소실점을 생성한 특징점들을 인라이어들로 간주한다. 이와 같은 방법

으로 선택된 인라이어들 가운데 아웃라이어가 섞일 수 있고, 또한 이 선택은 이동(translation)을 고려하지 않았기 때문에 필터링 과정이 필요하다. 이를 위해 본 연구에서는 마지막 과정으로 번들 조정(bundle adjustment)을 실시한다. 즉, 인라이어로 선택된 이전 영상의 특징점들의 3차원 점을 현재의 좌우 영상으로 투영하면 이 투영점들에 대응하는 현 특징점들과의 위치 차이를 최소화 하는 회진행렬과 병진벡터(translation vector)가 있을 것이다. 바로 이 회진행렬과 병진벡터를 구하면 차량의 모션이 계측된다.

III. 특징점 추출

1. 특징점 추출

본 연구에서 특징점 추출은 두 가지 방법으로 이루어진다. 첫째, Harris 코너 검출기의 적용이다. 둘째, 광류(optical flow) 계산에 의해 이전 영상의 특징점과 대응되는 픽셀을 현 영상에서 찾아 특징점으로 삼는 것이다. 단원 II에서 언급하였듯 제안된 알고리즘에서 Harris 코너 검출기는 두 가지 역할을 한다. 첫째, 이전 영상의 특징점이 존재하지 않는 알고리즘 시작단계에서 처음으로 입력된 좌우 영상에 적용한 Harris 코너 검출기가 추출한 코너점들은 모두 특징점으로 삼는다. 둘째, 알고리즘 시작단계를 제외하고는 항상 이전 영상에서 추출된 특징점들이 존재하므로 새로 영상이 입력되었을 경우 이전 영상의 특징점들 각각에 대응되는 픽셀을 광류계산으로 찾아내어 이 픽셀들을 현 영상의 특징점으로 삼는다. 그러나 이 때 현 영상은 자차량의 모션에 의해 이전 영상과는 다른 영상이기 때문에 이 방법으로 찾아진 특징점들만으로는 현 영상을 완전히 표현할 수는 없다. 따라서 이를 보완하기 위해 현 영상에 대해서도 Harris 코너 검출기를 적용하여 코너점을 추출하는데, 이미 광류로 결정된 특징점 주위의 코너는 배제한다.

2. Harris 코너 검출기

특징점 추출에 Harris 코너 검출기를 이용한 것은 위치정확성과 반복성이 뛰어나며[2], 코너 추출시 사용한 헤시안 행렬(Hessian matrix)을 광류계산시에도 사용할 수 있기 때문에 처리시간을 단축할 수 있어서다. Harris 코너 검출은 두 단계를 걸쳐 수행된다. 첫 단계는 모든 픽셀에 대해 헤리스 코너값을 계산하는 것으로 식 (2)에 의한다. 식 (2)의 $h(p)$ 는 픽셀 $p = (x, y)$ 에서의 헤리스 코너값으로 헤시안 행렬 G 에 의해서 계산되며, $\det(G)$ 는 헤시안 행렬의 행렬식이고, $\text{trace}(G)$ 는 헤시안 행렬의 대각요소들의 합이며, k_h 는 Harris 코너의 값을 조절하기 위한 상수로 보통 0.03정도를 설정한다. 식 (3)은 헤시안 행렬로서 식 내의 I_x 와 I_y 는 각각 영상의 수평, 수직축의 밝기값 그레디언트(gradient) 요소이며 3x3 소벨연산자를 이용하여 계산한다. 그리고 헤시안 행렬 G 는 5x5 영역 내에서 그레디언트 요소를 합산하여 구한다. 둘째 단계는 계산된 헤리스 코너값을 이용하여 코너점을 선별하는 것이다. 보통의 경우 임계값을 이용하지만, 영상에 따라서 임계값 설정이 어렵고 최대한 코너점이 많이 선택되도록 본 논문에서는 지역 최대값을 뽑는 방법인 식 (4)를 이용한다. 식 (4)는 해당 픽셀 p 의 5x5 크기의 윈도우 영역 $w(p)$ 내에서 p 의 이웃 픽셀들 \tilde{p} 와 Harris 코너값을 비교하여 p 의 값이 가장

크면 p 를 코너점으로 선택한다[6].

$$h(p) = \det(G(p)) + k_h \times \text{trace}^2(G(p)) \quad (2)$$

$$G(p) = \begin{bmatrix} \sum I_x^2(p) & \sum I_x(p)I_y(p) \\ \sum I_x(p)I_y(p) & \sum I_y^2(p) \end{bmatrix} \quad (3)$$

$$H(p) = \begin{cases} 1 & \text{if } (h(p) > h(\tilde{p}), \tilde{p} \in w(p), \tilde{p} \neq p) \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

3. 광류계산

짧은 시간 내에 연속적으로 영상을 얻는 경우 영상간 카메라 시점(view point) 차가 크지 않기 때문에 작은 베이스라인 문제로 가정할 수 있다[17]. 이런 상황에서 영상간 대응점 탐색은 광류계산을 통해서 빠르게 수행할 수 있다. 따라서 이전 영상의 특징점 \mathbf{p}^{t-1} 에 대응되는 현 영상의 특징점 \mathbf{p}^t 를 식 (5)와 같이 표현한다. 이 식의 \mathbf{u} 는 변위량으로 KLT (Kanade-Lucas-Tomasi) 방법[12]을 이용한 광류계산으로 얻는다.

$$\mathbf{p}^t = \mathbf{p}^{t-1} + \mathbf{u} \quad (5)$$

본 논문에서 특징점을 코너점으로 선택을 했으므로 이와 같이 이전 영상의 특징점에 대응하는 현 영상의 픽셀을 광류계산으로 탐색하는 방법은 광류계산시 나타나는 어퍼처(aperture) 문제를 최소화할 수 있다. 이외에 코너점을 찾기 위해 밝기값의 그레디언트를 계산하므로 광류 계산시 필요로 하는 그레디언트를 별도로 계산할 필요가 없다.

4. 스테레오 매칭

본 연구에서는 차량 모션계측에 3D 데이터를 사용하는데, 3D 데이터 추출을 위해 좌우 영상에서 추출된 특징점들에 대해 좌측 영상의 특징점을 기준으로 일정 크기의 윈도우를 이용하여 좌우 영상간 특징점 매칭을 한다. 이 때 영상은 교정(rectification)된 영상이지만, 매칭의 정확도를 높일 목적으로 매칭의 범위를 차량의 진동과 교정 에러 등을 감안하여 수직으로는 스캔라인과 스캔라인의 이웃 행까지 포함하며, 수평으로는 최대시차를 넘지 않는 영역 내로 설정한다. 이와 같이 좌영상의 특징점 $p_l = (x_p, y_p)$ 에 대응되는 우영상의 특징점 $p_r = (x_p, y_p)$ 을 찾기 위해 계산하는 매칭 비용을 식 (6)으로 정의한다. 위에서 언급한 바와 같이 p_r 의 탐색범위는 $\{0 \leq x_p - x_p \leq D \text{ and } |y_p - y_p| \leq 1\}$ 이다. 여기서 D 는 최대시차로서 영상의 해상도와 카메라 베이스라인 등에 의해 결정된다.

$$C(p_r) = \sum_{\substack{p' \in \Omega(p_l) \text{ and} \\ p'' \in \Omega(p_r)}} |I_x^l(p') - I_x^r(p'')| + |I_y^l(p') - I_y^r(p'')| \quad (6)$$

여기에서 $\Omega(p_l)$ 과 $\Omega(p_r)$ 은 각각 특징점 p_l 과 p_r 을 중심으로 한 11x11 크기의 윈도우이며, I_x 와 I_y 는 각각 수평, 수직의 밝기값 그레디언트이고, 위 첨자 l 과 r 은 각각 좌영상과 우영상을 의미한다. p_l 의 대응점으로는 이 비용들 가운데 최소의 비용을 생성한 p_r 이 선택된다.

스테레오 매칭이 이루어진 특징점 $p_l = (x_p, y_p)$ 의 3D 점 $P = (X, Y, Z)$ 는 다음과 같이 계산된다.

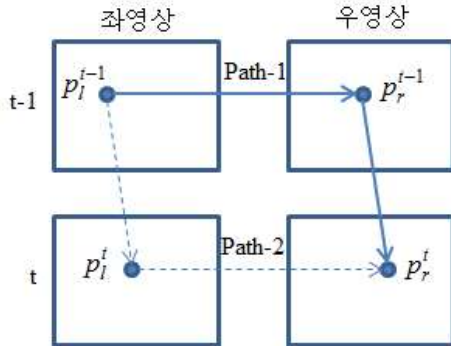


그림 2. 순환매칭.

Fig. 2. Cyclic matching.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \frac{(x-c_u) \cdot b}{d} \\ \frac{(y-c_v) \cdot b}{d} \\ \frac{f \cdot b}{d} \end{bmatrix} \quad (7)$$

여기에서 f 는 초점거리, b 는 베이스라인, d 는 p_l 의 시차(disparity)이며, (c_u, c_v) 는 카메라 영상의 원점(principal point)이다.

5. 순환매칭

광류계산에 의해 전후 영상간의 특징점들을 매칭시키고, 식 (6)을 이용하여 좌우 영상간의 특징점들을 매칭시키면 시점 $t-1$ 의 좌영상에서 시점 t 의 우영상에 도달하는 길은 그림 2에 보인 바와 같이 Path-1과 Path-2 두 가지가 생긴다. 이 말은 시점 $t-1$ 의 좌영상의 어떤 특징점에서 출발하여 다른 세 영상을 거쳐 다시 이 점으로 돌아올 수 있다는 뜻이다.

문제는 매칭 오류가 있을 수 있다는 점이다. 순환매칭 검사는 이 오류가 생긴 특징점을 찾기 위한 방법으로 식 (8)에 의해 행해진다.

$$\|p'_{s,r} - p'_{o,r}\| < \tau \quad (8)$$

여기에서 $p'_{s,r}$ 는 식 (6)에 의해 p'_l 와 매칭이 이루어진 우영상의 특징점이고, $p'_{o,r}$ 은 광류계산에 의해 p'_r 과 매칭이 이루어진 점이며, τ 는 임계값으로서 실험적으로 결정한 값이다. 논문에서 τ 는 1로 설정하였다. p'_l 이 식 (8)를 만족하면 순환매칭 검사를 통과한 특징점이 되지만 만족하지 못하면 모션계측에서 배제된다.

IV. 모션 계측

1. 특징점 분리

정적인(static) 배경에서 검출된 특징점들은 자차량의 모션에 대해 모두 동일한 움직임을 갖겠지만, 동적인 배경에서 추출된 특징점들은 제 각각의 움직임을 보일 것이다. 이 말은 정적인 배경에서 추출된 특징점들의 전후 영상에서의 움직임을 알면 자차량의 모션을 정확히 계측할 수 있다는 것이다. 따라서 정적인 배경에서 추출된 특징점들을 인라이어(inlier)라 하고 그 외의 특징점들을 아웃라이어(outlier)라 한다.

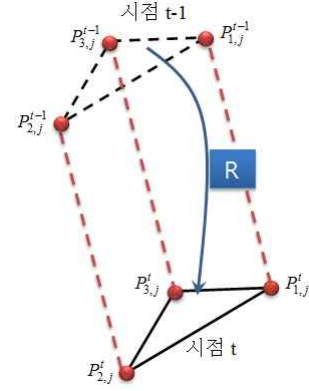


그림 3. 회전에 의한 두 영상 사이의 3개의 3D 점의 변환.

Fig. 3. Transformation of three 3D points between two images by a rotation.

본 논문에서는 인라이어와 아웃라이어 분리를 위해 딜로나이 삼각형화(DT)를 이용한다. DT는 평면위의 점들을 삼각형으로 연결하여 공간을 분할할 때, 이 삼각형의 내각의 최소값이 최대가 되도록 분할을 수행하는 기법이다[11]. DT를 이용하면 추출된 특징점의 움직임을 계산하는 데에 필요한 최소 점인 세 점을 효과적으로 분할할 수 있으며, 이 세 점은 동일 선상에 있지 않기 때문에 항상 삼각형의 움직임을 계산할 수 있다.

이전 영상과 현 영상 사이에 서로 대응되는 특징점들의 3D 점을 식 (7)에 의해 얻은 후, 이전 영상에서 구축된 삼각형들의 세 꼭지점들의 3D 점들이 현 영상의 3D 점들로 변환되는 회전행렬을 식 (9)에 의해 계산하는데[7], 그림 3은 이때 서로 대응되는 3D 점들의 관계를 보인 것이다.

$$R_j = \min_{\tilde{R} \in SO(3)} \sum_{i=1}^3 \|\tilde{R}P_{i,j}^{t-1} - P_{i,j}^t\|^2 \quad (9)$$

여기에서 $P_{i,j}^t$ 는 시점 t 에서 얻어진 삼각형 j 의 i 번째 3D 점이다. 식 (9)를 전개한 과정을 나타낸 식 (10)에서 $\sum_i \|P_{i,j}^{t-1}\|^2$ 과 $\sum_i \|P_{i,j}^t\|^2$ 은 상수이므로, R_j 는 식 (11)과 같이 모델링 할 수 있다.

$$\begin{aligned} & \sum_i (\|\tilde{R}P_{i,j}^{t-1}\|^2 - 2(\tilde{R}P_{i,j}^{t-1}, P_{i,j}^t) + \|P_{i,j}^t\|^2) \\ &= \sum_i \|P_{i,j}^{t-1}\|^2 - 2\text{trace}(\tilde{R}^T \sum_i (P_{i,j}^{t-1})^T \cdot P_{i,j}^t) + \sum_i \|P_{i,j}^t\|^2 \quad (10) \\ &\approx -2\text{trace}(\tilde{R}^T \sum_i (P_{i,j}^{t-1})^T \cdot P_{i,j}^t) \end{aligned}$$

$$R_j = \max_{\tilde{R} \in SO(3)} \text{trace}(\tilde{R}^T K_j) \quad (11)$$

여기에서 $K_j = \sum_i (P_{i,j}^{t-1})^T \cdot P_{i,j}^t$ 이다. 이때 K_j 의 SVD (Singular Value Decomposition) [14]가 $K_j = U \Sigma V^T$ 와 같이 되고, R_j 가 식 (12)과 같이 되었을 때, 식 (11)를 만족시킨다. 또한 $\text{rank}(K) > 1$ 이고, U 와 V 가 직교행렬이므로 $\det(UV^T) = \pm 1$ 이 되므로, 식 (12)는 $SO(3)$ 그룹의 조건을 만족한 유효해가 된다.

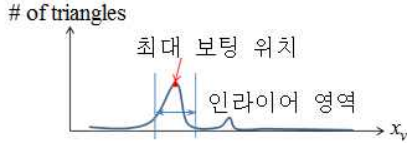


그림 4. 특징점의 인라이어와 아웃라이어 분리.
Fig. 4. Division of features into inlier and outlier.

$$R_j = U \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det(UV^T) \end{pmatrix} V^T = UV^T \quad (12)$$

추정된 회전행렬을 이용한 특징점 분리에 소실점(vanishing point)을 이용한다. 카메라의 이동은 소실점의 위치를 바꾸지 않지만 회전은 소실점의 위치를 바꾼다. 따라서 회전행렬이 동일하면 서로 다른 3D 점들도 동일한 소실점을 갖는다. 특징점들을 인라이어와 아웃라이어로 분리하는 데에 이 원리를 활용하기 위해 소실점의 위치를 계산한다. 보통의 경우 차량의 움직임을 평면기반의 회전으로 모델링 할 수 있다는 Nguyen 등[5]의 연구 결과에 따라 소실점의 위치 변화를 하나의 수식으로 나타낼 수 있다. 식 (13)의 x_v 는 영상에서 소실점의 수평 좌표를 나타낸다. 그리고 r_{ij} 는 회전행렬의 (i, j) 번째 요소이며, f 는 초점거리이고, $round(\cdot)$ 는 반올림 함수이다.

$$x_v = \lim_{Z \rightarrow \infty} f \frac{r_{11}X + r_{12}Y + r_{13}Z}{r_{31}X + r_{32}Y + r_{33}Z} \approx round \left(f \frac{r_{13}}{r_{33}} \right) \quad (13)$$

이와 같은 방법으로 삼각형마다 소실점을 계산한 후 어떤 소실점에 가장 많은 삼각형들이 관련되는지 파악하기 위해 보팅(voting)기법에 기반하여 소실점의 위치마다 삼각형의 수를 누적한다. 그림 4는 보팅 결과를 나타낸 그래프인데, 이 그래프에서 누적이 가장 많이 이루어진 소실점 위치를 찾고, 이 소실점의 위치에 가까이에 보팅된 삼각형들도 최대 보팅된 삼각형들과 유사한 회전행렬을 가진 것으로 보고, 이 최대 위치 주변으로 일정 폭을 정해 이 범위에 보팅된 삼각형들을 구성한 특징점들을 인라이어로 선택한다. 이 때 특징점들이 중복으로 인라이어로 선택되는 것은 피한다.

2. 번들 조정

모든 삼각형들에 대해서 식 (13)에 의해 계산된 소실점 위치에 보팅(voting)이 끝난 후, 차량의 모션은 인라이어로 선택된 특징점들을 이용하여 계산한다. 이를 번들 조정(bundle adjustment)이라 한다. 번들 조정 과정을 도입한 이유는 세 점만을 이용하여 차량의 움직임을 계산하는 경우[6]보다 많은 특징점들의 데이터를 이용하는 경우가 잡음의 영향을 줄일 수 있어서 모션계산이 더욱 정확해지기 때문이다. 번들 조정에는 최적화 기법을 사용하는데, 본 논문에서는 오매칭 데이터의 영향을 최소화 할 수 있는 M-추정법(M-estimator) [13]을 선택하였다. 이를 이용하여 모션 파라미터인 회전 R 과 이동 T 까지 고려하여 모델링한 것이 식 (14)이다.

$$\min \sum_{i=1}^N \eta(e_i^{t-1}) (\|p'_{i,l} - \pi_l(P_i^{t-1}; R, T)\|^2 + \|p'_{i,r} - \pi_r(P_i^{t-1}; R, T)\|^2) \quad (14)$$

$$\eta(e) = \frac{1}{1 + |e|/c}, \quad c = 1.3998 \quad (15)$$

$\eta(e)$ 는 M-추정법의 'Fair' 함수에 의한 반복 계산시의 가중치로, 초기값은 1로 설정한다[13]. 식 (14)에서 π_l 과 π_r 은 각각 좌우 카메라의 투영행렬이며, N 은 인라이어로 선택된 특징점들의 수이고, i 는 인라이어 특징점의 인덱스이다. P_i^{t-1} 는 이 특징점의 3D 점이다. $p'_{i,l}$ 와 $p'_{i,r}$ 는 각각 현 시점의 좌우 영상에서의 P_i^{t-1} 에 대응될 특징점들이다. k 는 최적화 수행 횟수이며, 반복적인 계산에 의해서 최소 에러를 갖는 차량의 회전 R 과 이동 T 를 계산한다. 이에 대한 상세 과정은 부록에 소개한다.

3. 차량모션예측

식 (15)에서 구한 R, T 를 이용하여 t 시점의 차량의 모션 M'_v 를 구하는 것은 식 (16)과 같다. 수식에서 R_v, T_v 는 차량의 모션을 나타내는 회전행렬과 이동벡터이다. 식 (16)에서 M_v^{t-1} 는 이전 시점의 차량의 모션으로, 최초의 모션은 식 (17)과 같이 정한다.

$$M'_v = \begin{bmatrix} R_v & T_v \\ 0 & 1 \end{bmatrix} = M_v^{t-1} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}^{-1} \quad (16)$$

$$M_v^0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (17)$$

V. 실험결과

1. 차량의 모션과 영상의 변화

그림 5에 화살표로 표현한 바와 같이 연속된 영상에서의 정적 배경의 위치 변화는 카메라의 모션에 의해 야기된다. 이 변화를 알아내면 카메라의 모션을 계산할 수 있다. 만약에 카메라가 차량에 장착되어 있고, 카메라와 차량간의 기하학적 관계를 알면 식 (16)에 의해 카메라의 모션으로부터 차량의 모션을 계산할 수 있다.

2. 인라이어 검출과정

그림 6은 본 연구에서 제안한 특징점 추출부터 인라이어 검출까지의 과정을 보인 예다. 그림 6(a)는 이 예시에 사용된



그림 5. 자차량 모션에 의한 영상의 변화.
Fig. 5. Change on the images by the motion of ego-vehicle.



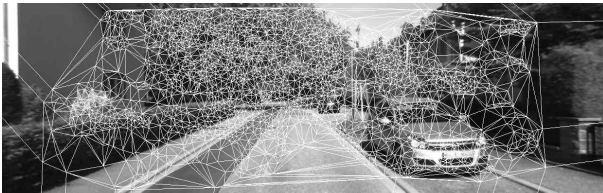
(a) Images captured at time t-1 and t.



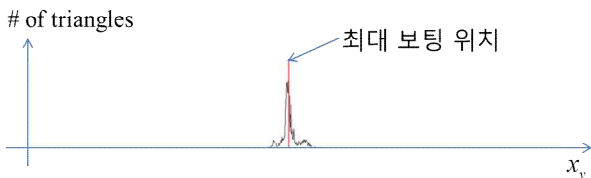
(b) Extraction of feature points.



(c) Computation of optical flow.



(d) Triangles by Delaunay triangulation.



(e) Distribution of triangles for the location of vanishing point x_v .



(f) Optical flow extracted by inliers.

그림 6. 인라이어 검출과정.

Fig. 6. Selection of inliers.

연속한 두 영상(상단부: t-1시점 영상, 하단부: t시점 영상)이다. 그림 6(b)는 t-1시점의 영상에 Harris 코너 검출기를 적용하여 추출한 특징점을 보인 것이다. 그림 6(c)는 본문의 식 (5)에 의해 계산된 광류를 t시점 영상에 표시한 것으로서, 광류는

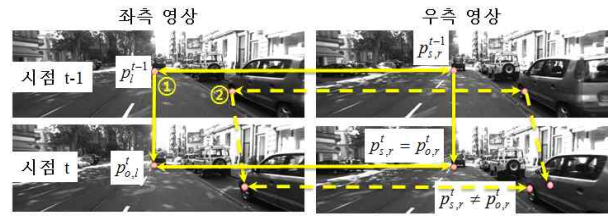


그림 7. 순환매칭 검사의 개요.

Fig. 7. Concept of cyclic matching inspection.

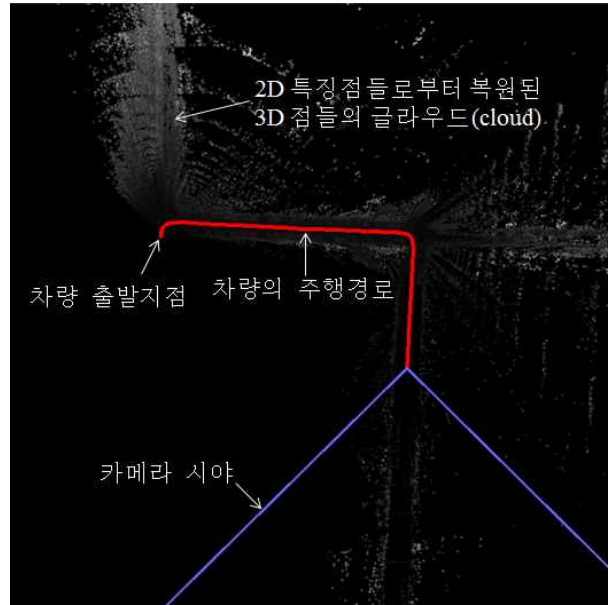


그림 8. 제안된 알고리즘에 의해 복원된 차량주행경로.

Fig. 8. Vehicle trajectory recovered by the proposed algorithm.

그림 6(b)에 보인 t-1시점 영상의 특징점들에 대응이 되는 픽셀들을 t시점 영상에서 탐색함으로써 얻은 것이다. 그림 6(d)는 순환매칭 검사를 통과한 특징점들을 가지고 DT를 적용하여 구축한 삼각형들을 표시한 것이다. 그림 7은 순환매칭 검사에 대한 개념을 실제 영상을 통해 설명한 것으로서, 그림 7의 좌측 상단에 있는 영상에서 특징점 ①번은 이 검사를 통과했지만, ②번은 검사를 통과하지 못했다. 따라서 이후 과정에서 이 특징점은 배제된다. 그림 6(e)는 DT로 얻은 삼각형 각각으로부터 본문의 식 (9)에 의해 예측된 회전행렬을 통해 계산된 소실점들의 수평위치상의 자신의 소실점 위치에 보팅(voting)된 삼각형들의 분포도를 보인 것인데, 가장 많은 삼각형들이 보팅된 위치도 함께 표시하였다. 이 최대 보팅 위치를 낳은 삼각형들을 구성하는 특징점들이 바로 본 연구에서 언급한 인라이어가 된다. 그림 6(f)는 바로 이 인라이어들의 광류를 표현한 것이다.

3. 경로추종

제안된 알고리즘에 의해 복원된 차량주행경로를 그림 8에 표현하였다. 최초 위치에서 출발한 차량은 짧게 직선주행을 하다 우측으로 회전한 후 직선주행을 이어갔고, 다시 우측으로 회전한 후 직선주행을 했다. 그림 8은 복원된 주행경로와 2D특징점들로부터 3차원 복원된 점들의 클라우드(cloud)를 함께 보여 주고 있다.

다음 실험결과는 본 연구에서 제안한 방법과 RANSAC에 기반한 방법을 KITTI [15] 데이터 셋에 적용하여 복원된 차량주행경로를 KITTI [15]에서 제공한 참 경로(ground truth)와 비교해 본 것이다. 그림 9(a)는 KITTI의 데이터 셋1의 참 경로를 나타낸 것이고, 그림 9(b)는 본 연구에서 제안한 방법에 의해 복원된 경로이고, 그림 9(c)는 RANSAC을 기반으로 한 방법에 의해 복원된 경로를 표시한 것이다. 그림 9(d)는 KITTI [15] 데이터 셋2에서 제공된 참 경로, 본 연구에 의해 복원된 경로 그리고 RANSAC 기반의 방법에 의해 복원된 경로 이 세가지를 동시에 나타낸 것이다. 그림 9(d)를 보면 제안된 방법이 회전모션 복원은 별 문제가 없지만 이동모션 시 목적지까지 도달하기 이전에 방향을 바꿈으로써 이동에 있어서의 에러가 상대적으로 크다는 것을 알 수 있다.

표 1에 제시된 에러 데이터는 KITTI [15] 데이터 셋에 적용하여 얻은 차량주행경로 데이터의 평균을 취해 산출된 것이다. 여기에서 이동모션의 에러는 1m당 위치 오차의 비율을 뜻하며, 회전모션의 에러는 1m당 회전 오차 각도를 뜻한다. 앞서 그림 9를 설명할 때 이미 언급하였듯이 제안된 방법의 이동모션에 대한 에러가 RANSAC에 기반한 방법에 비해 상대적으로 더 크다는 것을 알 수 있다.

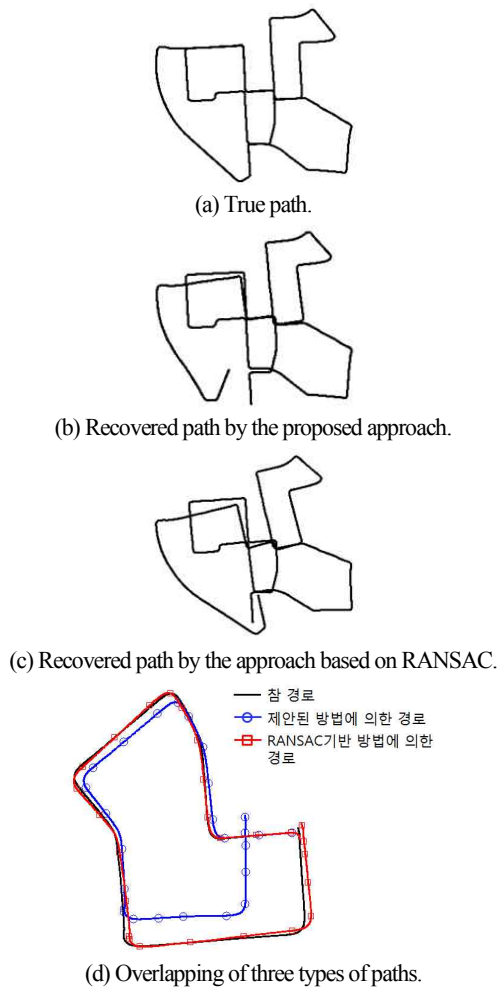


그림 9. 복원된 차량주행경로와 참 경로의 비교.

Fig. 9. Comparison of recovered vehicle trajectory with the ground truth.

표 1. 예측 모션의 에러율.

Table 1. Error rate of estimated motion.

	이동모션 에러(%)	회전모션 에러(deg/m)
RANSAC에 의한 방법	2.3	0.011
제안된 방법	4.4	0.012

표 1에 제시된 제안된 방법의 이동모션 에러발생 원인은 다음과 같다. 그림 10(a)와 (b)의 원내에 보인 바와 같이 영상의 측면에 위치하면서 광류가 클 경우 이에 해당된 픽셀들이 제안된 방법에서는 RANSAC에 의한 방법에 비해 인라이어 대상에서 탈락하는 경향이 크다. 그 이유를 알아보기 위해 그림 10(c)의 원내에 보인 인라이어에서 탈락한 세 점을 분석하였다. 기본적으로 DT는 인접한 세 점을 택하므로 세 점이 영상의 넓은 영역에 분포하지 않고, 스테레오 매칭에 의한 3차원 위치 계산결과에도 부정확성이 존재한다. 따라서 그림 10(c)의 세 점의 광류는 실제로는 이동모션에 의한 것이지만, 그림 10(d)에 보인 바와 같은 회전 모션과 유사한 음



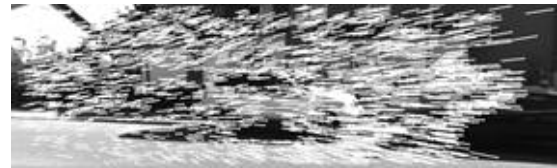
(a) Inliers extracted by DT.



(b) Inliers extracted by RANSAC.



(c) Optical flow of three points removed by DT.



(d) Example of optical flows at a rotating motion of a vehicle.



(e) Distribution of triangles built by DT for the location of vanishing point x_v .

그림 10. 제안된 방법의 이동모션 에러의 원인 분석.

Fig. 10. Analysis of error sources of translation motion for the proposed algorithm.

표 2. 알고리즘 처리시간.

Table 2. Processing time of motion estimation algorithms.

	데이터 셋 1		데이터 셋 2	
	평균 처리 시간 (msec)	최대 처리 시간 (msec)	평균 처리 시간 (msec)	최대 처리 시간 (msec)
RANSAC	18.3	72.2	46.2	4942.8
제안된 방법	27.6	56.0	29.9	58.1

직입으로 계산되었다. 그 결과 그림 10(e)에 보인 바와 같이 이 세 점의 보팅위치가 최대 보팅위치와는 떨어진 곳이었으며, 결국 인라이어 대상에서 제외되었다. 이와 같이 인라이어여야 할 특징점들이 탈락하면서 차량의 이동벡터의 크기 추정에 에러가 생긴 것이다. 그러나 이동 모션의 크기를 차량의 차속에 의해 보정하면 에러를 크게 줄일 수 있다.

서론에서 언급하였듯이 RANSAC에 기반한 VO는 처리시간에 대한 변동폭이 크다는 단점을 갖는다. 표 2에 제시한 데이터는 이 점을 명확히 밝히고 있다. 특히 KITTI [15]의 데이터 셋2에 적용된 결과는 그 변동폭이 매우 컸음을 알 수 있다. 이와 같은 상황을 알고리즘을 적용하기 이전에 미리 알 수 없기 때문에 VO를 활용하는 문제에서 한 프레임에 처리하는 데에 소요되는 시간을 예측하기 어렵다.

표 2에 제시된 알고리즘 처리시간의 측정은 Intel Core i7@3.4GHz 컴퓨터에서 수행하였다.

VI. 결론

본 논문은 차량의 모션을 예측하기 위한 스테레오 VO를 구현한 알고리즘을 제안하였다. 제안된 알고리즘은 다른 VO 알고리즘들과는 다르게 RANSAC에 의존하지 않았기 때문에 RANSAC이 갖는 단점을 줄였다는 점이 큰 특징이다. 알고리즘은 DT를 적용하여 특징점들을 세 점씩 묶어 회전행렬을 구하고, 이를 통해 소실점의 영상상의 위치 변화를 계산하여 영상 특징점들의 인라이어 그룹과 아웃라이어 그룹을 분리하였다. 그리고 인라이어 특징점들을 대상으로 번들 조정 과정을 통해 자차량의 모션 파라미터인 회전과 이동량 계산의 정확도를 높일 수 있었다.

제안된 알고리즘은 실험 결과에서 보였듯이 안정적으로 차량의 주행경로를 추종하였다.

부록

본문의 식 (14)를 이용해 차량의 모션 R과 T를 계산하는 과정을 소개한다. 카메라의 좌우 투영행렬은 다음과 같다.

$$\begin{aligned} \pi_l(\mathbf{P}_i^{t-1}; \mathbf{R}, \mathbf{T}) &= \mathbf{A}(\mathbf{R}\mathbf{P}_i^{t-1} + \mathbf{T}) \\ \pi_r(\mathbf{P}_i^{t-1}; \mathbf{R}, \mathbf{T}) &= \mathbf{A} \left(\mathbf{R}\mathbf{P}_i^{t-1} + \mathbf{T} - \begin{pmatrix} b \\ 0 \\ 0 \end{pmatrix} \right) \end{aligned} \quad (\text{A1})$$

여기에서

$$\mathbf{A} = \begin{bmatrix} f & 0 & c_v \\ 0 & f & c_u \\ 0 & 0 & 1 \end{bmatrix}$$

이고, b 는 좌우 카메라의 원점간 거리이며, R은 회전행렬, T는 병진벡터로서 각각 다음과 같다.

$$\begin{aligned} \mathbf{R} &= \mathbf{R}_x(\phi)\mathbf{R}_y(\theta)\mathbf{R}_z(\psi) = \\ & \begin{bmatrix} \cos\theta\cos\psi & & -\cos\theta\sin\psi \\ \cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi & \cos\phi\cos\psi - \sin\phi\sin\theta\sin\psi & \\ \sin\phi\sin\psi - \cos\phi\sin\theta\cos\psi & \sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi & \end{bmatrix} \\ & \begin{bmatrix} \sin\theta \\ -\sin\phi\cos\theta \\ \cos\phi\cos\theta \end{bmatrix} \end{aligned}$$

이고, $\mathbf{T} = [x \ y \ z]^T$ 이다. 여기에서

$$\begin{aligned} \mathbf{R}_x(\phi) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}, \quad \mathbf{R}_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}, \\ \mathbf{R}_z(\psi) &= \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \end{aligned}$$

식 (14)를 풀기 위한 모션 상태벡터는 다음과 같이 정의한다.

$$\mathbf{M} = [\phi \ \theta \ \psi \ x \ y \ z]^T \quad (\text{A2})$$

여기에서 ϕ 는 틸트(tilt)각, θ 는 요(yaw)각, ψ 는 롤(roll)각이고, x, y, z 는 각각 회전된 좌표계에서의 각 축방향의 이동거리이다. 상태벡터 M을 구하려면 식 (14)의 에러를 최소화하는 벡터를 찾으면 된다. 이를 찾기 위해서 식 (14)를 상태벡터 M에 대해서 미분하면 식 (A3)와 같이 전개가 되고, 수식은 M-추정법의 가중치 η , 자코비안 J, 잔차 Φ 로 이루어진다.

$$\begin{aligned} \frac{\partial e}{\partial \mathbf{M}} &= 2\eta \left(\left(-\frac{\pi_l(\mathbf{P}_i^{t-1}; \mathbf{R}, \mathbf{T})}{\partial \mathbf{M}} \right)^T (\mathbf{p}'_{i,l} - \pi_l(\mathbf{P}_i^{t-1}; \mathbf{R}, \mathbf{T})) \right. \\ & \quad \left. + \left(-\frac{\pi_r(\mathbf{P}_i^{t-1}; \mathbf{R}, \mathbf{T})}{\partial \mathbf{M}} \right)^T (\mathbf{p}'_{i,r} - \pi_r(\mathbf{P}_i^{t-1}; \mathbf{R}, \mathbf{T})) \right) \\ &= 2\eta \left(\left(-\mathbf{A} \frac{\partial(\mathbf{R}\mathbf{P}_i^{t-1} + \mathbf{T})}{\partial \mathbf{M}} \right)^T (\mathbf{p}'_{i,l} - \pi_l(\mathbf{P}_i^{t-1}; \mathbf{R}, \mathbf{T})) \right. \\ & \quad \left. + \left(-\mathbf{A} \frac{\partial(\mathbf{R}\mathbf{P}_i^{t-1} + \mathbf{T})}{\partial \mathbf{M}} \right)^T (\mathbf{p}'_{i,r} - \pi_r(\mathbf{P}_i^{t-1}; \mathbf{R}, \mathbf{T})) \right) \\ &= 2\eta \left(\left(-\mathbf{A} \frac{\partial(\mathbf{R}\mathbf{P}_i^{t-1} + \mathbf{T})}{\partial \mathbf{M}} \right)^T \left((\mathbf{p}'_{i,l} - \pi_l(\mathbf{P}_i^{t-1}; \mathbf{R}, \mathbf{T})) \right. \right. \\ & \quad \left. \left. + (\mathbf{p}'_{i,r} - \pi_r(\mathbf{P}_i^{t-1}; \mathbf{R}, \mathbf{T})) \right) \right) \\ &= 2\eta \left(\left(-\mathbf{A} \frac{\partial(\mathbf{R}\mathbf{P}_i^{t-1} + \mathbf{T})}{\partial \mathbf{M}} \right)^T (\mathbf{p}'_{i,l} + \mathbf{p}'_{i,r} - \mathbf{A}(\mathbf{R}\mathbf{P}_i^{t-1} + \mathbf{T})) \right. \\ & \quad \left. - \mathbf{A} \left(\mathbf{R}\mathbf{P}_i^{t-1} + \mathbf{T} - \begin{pmatrix} b \\ 0 \\ 0 \end{pmatrix} \right) \right) \\ &= 2\eta \left(-\mathbf{A} \left(\frac{\partial(\mathbf{R}\mathbf{P}_i^{t-1} + \mathbf{T})}{\partial \mathbf{M}} \right)^T (\mathbf{p}'_{i,l} + \mathbf{p}'_{i,r} - 2\mathbf{A}(\mathbf{R}\mathbf{P}_i^{t-1} + \mathbf{T})) \right) \end{aligned}$$

$$-A \begin{pmatrix} b \\ 0 \\ 0 \end{pmatrix} \quad (A3)$$

$$= 2\eta(-J^T \Phi) \propto -\eta J^T \Phi$$

수식의 최적화는 가우스 뉴턴법[14]을 이용하는데, 가우스 뉴턴법에 의해서 최적해를 찾는 방법은 식 (A4)와 같다.

$$M^{k+1} = M^k + \sum \eta (J^T J)^{-1} J^T \Phi \quad (A4)$$

여기에서

$$J(M) = A \begin{bmatrix} \frac{\partial R}{\partial \phi} P_i^{t-1} & \frac{\partial R}{\partial \theta} P_i^{t-1} & \frac{\partial R}{\partial \psi} P_i^{t-1} & \frac{\partial T}{\partial x} & \frac{\partial T}{\partial y} & \frac{\partial T}{\partial z} \end{bmatrix} \in \mathfrak{R}^{3 \times 6}$$

이고,

$$\Phi(M) = p'_{i,l} + p'_{i,r} - 2A(RP_i^{t-1} + T) - A \begin{pmatrix} b \\ 0 \\ 0 \end{pmatrix}$$

실제 프로그램에서 좌측 영상의 투영행렬 π_i , 자코비안 J 의 계산은 영상의 픽셀기준으로 하기 때문에 식 (A5), (A6)에 의해서 행한다.

$$\pi_i(P_i^{t-1}; R, T) = A(RP_i^{t-1} + T) = \begin{bmatrix} f \frac{X'}{Z'} + c_u \\ f \frac{Y'}{Z'} + c_v \end{bmatrix} \in \mathfrak{R}^{2 \times 1} \quad (A5)$$

$$J = \frac{\partial A(RP_i^{t-1} + T)}{\partial M} = \begin{bmatrix} f \frac{1}{(Z')^2} \left(\frac{\partial X'}{\partial M} Z' - X' \frac{\partial Z'}{\partial M} \right) \\ f \frac{1}{(Z')^2} \left(\frac{\partial Y'}{\partial M} Z' - Y' \frac{\partial Z'}{\partial M} \right) \end{bmatrix} \in \mathfrak{R}^{2 \times 6} \quad (A6)$$

여기에서

$$P_i^{t-1} = \begin{bmatrix} X^{t-1} & Y^{t-1} & Z^{t-1} \end{bmatrix}^T,$$

$$P' = RP_i^{t-1} + T = \begin{bmatrix} X' & Y' & Z' \end{bmatrix}^T,$$

$$\begin{bmatrix} \frac{\partial X'}{\partial M} \\ \frac{\partial Y'}{\partial M} \\ \frac{\partial Z'}{\partial M} \end{bmatrix} = \frac{\partial (RP_i^{t-1} + T)}{\partial M}$$

$$= \begin{bmatrix} \frac{\partial R}{\partial \phi} P_i^{t-1} & \frac{\partial R}{\partial \theta} P_i^{t-1} & \frac{\partial R}{\partial \psi} P_i^{t-1} & \frac{\partial T}{\partial x} & \frac{\partial T}{\partial y} & \frac{\partial T}{\partial z} \end{bmatrix} \in \mathfrak{R}^{3 \times 6}$$

REFERENCES

[1] D. Scaramuzza and F. Fraundorfer, "Visual odometry part I: the first 30 years and fundamentals," *IEEE Robotics and Automation Magazine*, vol. 18, no. 4, pp. 80-92, 2011.

[2] D. Scaramuzza and F. Fraundorfer, "Visual odometry part II: matching, robustness, optimization, and applications," *IEEE Robotics and Automation Magazine*, vol. 19, no. 2, pp. 78-90,

2012.

[3] D. Nister, O. Naroditsky, and J. Bergen, "Visual odometry," *Proc. of CVPR*, pp. 652-659, 2004.

[4] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 12, pp. 381-395, 1981.

[5] V. C. Nguyen, M. B. Heo, and G. I. Jee, "1-point ransac based on robust visual odometry," *Journal of the Korean GNSS Society*, vol. 2, no. 1, pp. 81-89, 2013.

[6] Y. Ma, "An invitation to 3D vision," Springer Science & Business Media, 2004.

[7] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239-256, 1992.

[8] M. Agrawal and K. Konolige, "Real-time localization in outdoor environments using stereo vision and inexpensive gps," *18th International Conference on Pattern Recognition*, vol. 3, pp. 1063-1068, 2006.

[9] K. Ohno, T. Tsubouchi, B. Shigematsu, and S. Yuta, "Differential gps and odometry-based outdoor navigation of a mobile robot," *Advanced Robotics*, vol. 18, no. 6, pp. 611-635, 2004.

[10] P. Corke, "An inertial and visual sensing system for a small autonomous helicopter," *Journal of Robotic Systems*, vol. 21, no. 2, pp. 43-51, 2004.

[11] M. de Berg, O. Cheong, M. Kreveld, and M. Overmars, "Computational geometry: algorithms and applications," Springer-Verlag, 2008.

[12] J. Y. Bouguet, "Pyramidal Implementation of the Lucas Kanade Feature Tracker," OpenCV Documentation, Intel Corporation, Microprocessor Research Labs, 1999.

[13] W. J. J. Rey, "Introduction to robust and quasi-robust statistical methods," Springer, Berlin, Heidelberg, 1983.

[14] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, "Numerical recipes 3rd edition: the art of scientific computing," Cambridge University Press, 2007.

[15] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *International Journal of Robotics Research*, pp. 1229-1235, 2013.

[16] S. U. Lee and B. K. Kim, "3D simultaneous localization and map building (SLAM) using a 2D laser range finder based on vertical/horizontal planar polygons," *Journal of Institute of Control, Robotics and Systems (in Korean)*, vol. 20, no. 11, pp. 1153-1163, 2014.

[17] Y. W. Choi, K. S. Choi, J. W. Choi, and S. G. Lee, "Localization using ego motion based on fisheye warping image," *Journal of Institute of Control, Robotics and Systems (in Korean)*, vol. 20, no. 1, pp. 70-77, 2014.

송 광 열

제어·로봇·시스템학회 논문지 제17권 제4호 참조.

이 준 응

제어·로봇·시스템학회 논문지 제19권 제6호 참조.