

## 영상기반 자동항법을 위한 실시간 위치인식 및 지도작성<sup>§</sup>

임 현\* · 임종우\*\* · 김현진\*<sup>†</sup>

\* 서울대학교 우주항공공학과, \*\* 한양대학교 컴퓨터공학과

### Real-time Simultaneous Localization and Mapping (SLAM) for Vision-based Autonomous Navigation

Hyon Lim<sup>\*</sup>, Jongwoo Lim<sup>\*\*</sup> and H. Jin Kim<sup>\*<sup>†</sup></sup>

<sup>\*</sup> Dept. of Mechanical and Aerospace Engineering, Seoul Nat'l Univ.,

<sup>\*\*</sup> Division of Computer Science and Engineering, Hanyang Univ.

(Received July 3, 2014 ; Revised October 7, 2014 ; Accepted March 12, 2015)

**Key Words:** Simultaneous Localization and Mapping(동시간 위치인식 및 지도작성), Autonomous Navigation(자동항법)

**초록:** 본 논문에서는 넓은 지역에서의 영상기반 자동 항법을 위한 실시간 위치인식 및 지도작성 방법을 제안한다. 한 대의 카메라에서 입력된 영상으로부터 제안된 방법은 6 자유도 카메라 자세와 3 차원 특징 점 위치를 연속적으로 계산한다. 제안된 방법은 넓은 지역을 주행하며 촬영된 영상에 적용하여 그 위치와 환경지도를 성공적으로 작성하였다. 본 논문에서는 이진기술자(binary descriptor)와 수치-위상(metric-topological) 지도 표현법을 사용하여 GPU 나 영상의 축소 없이 실시간 성능과 광범위한 지역에서의 회귀점 검출(loop detection)을 하였다. 제안된 방법은 여러 환경에서 촬영된 영상과, 해당 영상의 GPS 기준값과 비교하여 평가하였다.

**Abstract:** In this paper, we propose monocular visual simultaneous localization and mapping (SLAM) in the large-scale environment. The proposed method continuously computes the current 6-DoF camera pose and 3D landmarks position from video input. The proposed method successfully builds consistent maps from challenging outdoor sequences using a monocular camera as the only sensor. By using a binary descriptor and metric-topological mapping, the system demonstrates real-time performance on a large-scale outdoor dataset without utilizing GPUs or reducing input image size. The effectiveness of the proposed method is demonstrated on various challenging video sequences.

### 1. 서 론

Simultaneous Localization and Mapping(SLAM; 동시간 위치인식 및 지도작성) 방법은 센서의 위치를 계산하는 동시에 그 센서 주변의 환경을 매개변수화 하는 방법이다. 이 문제는 지난 30 여 년간 로보틱스 분야에서 연구되어 왔으며 최근에는 컴퓨터 비전 분야에서도 활발히 연구되고 있다. 많은 자동화 시스템 그리고 로봇 시스템에서 카메라는 이제 찾아보기 어려운 센서가 아니다. 따라서 영

상을 이용한 SLAM 방법이 각광을 받고 있다. 그 결과로 Visual SLAM (VSLAM)은 영상센서를 이용하는 SLAM 의 한 방법으로, GPS 나 고가의 무거운 레이저 거리 측정기(Laser Range Finders; LRF)의 도움 없이도 자동 항법을 수행할 수 있는 거의 유일한 대안으로서 간주되어 왔다.

VSLAM 은 사용되는 카메라의 개수에 따라 크게 두 가지로 분류할 수 있다. 한 대의 카메라(monocular camera)와 두 대의 카메라(stereo vision) 혹은 그 이상의 카메라가 사용되지만 크게는 앞의 두 가지가 사용된다. 한 대의 카메라를 사용할 경우 두 대에 비하여 비용적인 측면이나 설치의 용이성 그리고 계산의 편의성이 증대된다. 또한 두 대에 비하여 한 대의 카메라는 언제나 절반 혹은

<sup>§</sup> 이 논문은 2014년도 대한기계학회 IT 융합부문 춘계학술대회(2014. 4. 24.-25., 서울대) 발표논문임

<sup>†</sup> Corresponding Author, [hjinkim@snu.ac.kr](mailto:hjinkim@snu.ac.kr)

© 2015 The Korean Society of Mechanical Engineers

그 이하의 비용을 필요로 한다. 특히, 설치 가능한 면적이 작은 경우 두 대의 카메라는 거의 한 대의 카메라와 마찬가지로 동작한다.<sup>(1)</sup>

한 대의 카메라를 사용시 장점에도 불구하고, 카메라의 특징으로 인하여 VSLAM 에서의 이용은 저조한 편이다. 그 이유는 카메라만 가지고는 거리를 알 수 없을 뿐만 아니라, 거리를 알려면 특정 관심점(interesting point)에 대하여 강건한 추적 알고리즘을 필요로 하기 때문이다.<sup>(2,3)</sup>

로봇과 관련된 연구에서 Visual Odometer (VO) 역시 활발히 연구가 진행되어왔다. VO 와 VSLAM 의 가장 큰 차이는 회귀점 검출(loop closing)이라고 할 수 있다. 이는 발산하는 오류를 막을 수 있는 유일한 대안으로 알려져 있다. 회귀점을 검출하여 표류(drift) 오차를 줄일 수 있으며, 이는 넓은 지역에서의 VSLAM 에 필수적인 요소이다. 따라서 실시간 VSLAM 을 위해서는 실시간 회귀점 검출이 필요하다.

본 논문에서는 한 대의 카메라를 이용한 VSLAM 방법을 새롭게 제안한다. 제안된 방법은 빠르게 동작하는 특징점 추적기와 이진 기술자(binary descriptor)

를 이용한 영상기반 회귀점 검출 방법을 포함한다. 또한 수치-위상 지도표현법(metric-topological representation)을 통하여 빠른 회귀점 최적화 방법을 제안한다.

본 논문의 구성은 다음과 같다. 제 2 장은 관련 연구를 고찰하고, 3 장은 제안된 방법에 대하여 자세히 설명한다. 제 4 장에서는 최적화 및 지도작성 방법에 대해서 설명하고, 5 장에서 실험 결과를 고찰한다. 제 6 장에서는 본 논문의 결론에 대해서 기술한다.

## 2. 관련 연구

### 2.1 한 대의 카메라를 이용한 VSLAM

초기의 monocular SLAM 에 관한 연구는 2003 년 Davison 등<sup>(2)</sup>에 의하여 제안되었다. 확장 칼만필터를 이용한 환경지도 및 자세추정을 수행하였다. 최대 약 100 여 개의 환경지도 내의 랜드마크를 다룰 수 있었다. 하지만 이 방법은 아주 부드러운 카메라 움직임에서만 잘 동작하였으며 영상 내 특징점은 영상의 절편(patch)을 이용함으로써 SIFT, SURF 및 BRIEF 기술자에 비교하여 강건하지 못한 단점이 있다.

Parallel Tracking and Mapping (PTAM) 방법은 2007 년 Klein 등<sup>(3)</sup>에 의하여 제안되었다. 두 개의 서로 다른 기능을 담당하는 모듈이 동시에 수행되기 때문에 Davison 등의 연구<sup>(2)</sup>에 비하여 더 많은 특징점을 다룰 수 있게 되었다. 하지만 PTAM 방법은 사용자와의 상호작용을 통한 초기화 작업이 필요한 단점이 있어, 자동 항법에는 적합하지 않은 방법이다. 또한 다룰 수 있는 특징점의 개수는 더 큰 환경에서는 부족한 편이다.

요약하면, 크게 두가지 방법이 제안되어 왔다. 첫번째는 필터링 방법을 이용한 상태변수 추정법이다.<sup>(2)</sup> 다른 방법은 Bundle adjustment 를 이용한 방법이다.<sup>(3,4)</sup> 이 두 방법에 대한 비교 정리는 Strasdat 등의 논문에서 평가되었다.<sup>(5)</sup>

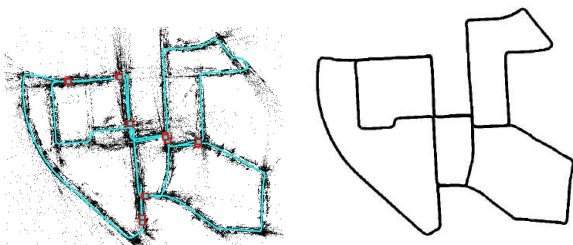
### 2.2 한 대의 카메라를 이용한 Visual Odometry

VSLAM 시스템과 함께, Visual Odometry (VO) 역시 활발히 연구되어 왔으며, 몇 가지 중대한 부분을 공유하고 있다. Nister 등은 2004 년 초기의 monocular VO 시스템을 제안하였다.<sup>(6)</sup> 이 방법은 RANSAC 기반의 3D-to-2D 자세추정 기법을 사용하였다. RANSAC 내에서의 계산량을 줄이기 위하여 5-pts 상대 자세 추정기법을 이용하였으며 이는 후에도 널리 사용되었다.<sup>(3)</sup>

Mouragnon 등은 2006 년 수 천개 수준의 랜드마크를 다룰 수 있는 실시간 영상기반 위치인식 및



(a) From monocular sequences, the proposed method builds 3D map successfully in a large-scale environment



(b) A scale consistent environment map with closed loops which is denoted by red square. Ground-truth trajectory is displayed on the right side for qualitative comparison

**Fig. 1** Input and outputs of the proposed method. The environment map and camera poses obtained by the proposed method using monocular sequences

지도작성에 대한 연구를 수행하였다.<sup>(7)</sup> 하지만 회귀점 인식 등 광범위 지역에서의 SLAM에 반드시 필요한 부분은 연구되지 않았다.

### 2.3 Graph-based Formulation

최근 그래프 기반의 SLAM 방법이 많이 연구되어 왔다. 이 방법은 로봇의 자세를 정점으로, 정점간 상대자세를 간선으로 표현하는 것이다. 간선은 정점간의 상대 자세들 뿐만 아니라, 랜드마크와 현재 정점간의 제약조건도 포함한다. Kümmerle 등은 비선형 최적화를 통한 그래프 기반 SLAM의 일반화 프레임워크를 제안하였다.<sup>(10)</sup> Lim 등은 metric-topological 표현법을 제안하였다.<sup>(11)</sup>

효율적인 회귀점 검출 후 최적화를 위하여 Strasdat 등은 double-window optimization 기법을 제안하였다.<sup>(12)</sup> 이 기법에서는 두 가지 최적화를 진행하게 되는데 하나는 랜드마크-자세에 관련한 제약조건이고 다른 하나는 자세-자세 제약조건이다. 이 두 조건의 활성화는 visibility 조건에 의하여 결정된다. 회귀점이 검출되었을 때에는 자세-자세 제약조건을 먼저 해결하는 편이 더 나은 수렴속도를 보장한다고 알려져 있다.<sup>(4)</sup>

### 2.4 회귀점 검출 및 최적화

본래 SLAM은 incremental 한 추정의 누적이기 때문에 특정 물리적인 지점의 재방문이 이루어지지 않게 되면 계속 그 값은 불확실해진다.<sup>(13)</sup> 따라서 카메라가 먼 거리를 계속 이동할 경우 이 표류현상(drift)은 더욱 두드러지게 된다. 이 문제를 극복하기 위하여 영상 유사성을 이용한 회귀점 검출 문제가 활발히 연구되어 왔다.<sup>(14,15)</sup> 이를 위하여 기존에는 SIFT나 SURF 같은 특징점 기술자가 이용되어 왔다. 하지만 이 기술자들은 매우 많은 연산을 요구하기 때문에 실시간으로 연산이 이루어지기 힘들다. 이를 극복하기 위하여 연산 속도가 빠른 이진 기술자<sup>(16)</sup>를 이용한 회귀점 검출법이 제안되었다.<sup>(17)</sup>

### 2.5 상대좌표 표현법

VSLAM에서의 좌표계는 다양하게 제안되어 왔다. 가장 일반적인 선택은 SLAM의 시작점을 원점으로 하는 하나의 global coordinates를 이용하는 것이다. 일부 연구자들은 현재 로봇의 위치를 원점으로 하는 robo-centric; ego-centric coordinates<sup>(18)</sup>를 제안하기도 하였다. 이는 현재 로봇의 위치가 원점으로 기술되고, 모든 랜드마크와 과거의 자세들은 이에 상대적으로 표현되는 것이다. 하지만 이 두 방법에는 공

통적인 문제가 있다. SLAM은 본래 점진적인 추정의 누적이기 때문에 이 누적을 극복하기에는 하나의 직렬화 된 좌표 표현법은 적합하지 않은 것이다. 따라서 본 논문에서는 유효한 상대좌표를 직접 최적화하는 방법을 사용한다. 유사한 방법으로 Mei 등의 Continuous Relative Representation (CRR)<sup>(16)</sup>이 있으나, 키프레임을 기술하기 위한 원점 선택 기준이 존재하지 않고 임의적이다.

## 3. 제안된 Visual SLAM 기법

### 3.1 실행 파이프라인

본 연구에서는 키포인트 추출을 위해서 FAST 코너<sup>(20)</sup>를 이용하였다. 매 프레임마다 FAST 코너가 추출되고 이전 이미지로부터 일정 영역 내에 존재하는 FAST 코너에 대해서 기술자 정합(descriptor matching)을 수행한다. 기술자 정합을 위하여 BRIEF 기술자<sup>(16)</sup>를 사용하였다. 그리고 특징점 추적을 위해 사용된 BRIEF 기술자는 또한 회귀점 검출에도 사용된다. 회귀점 검출을 위하여 binary visual words 방법을 사용하였다.<sup>(17)</sup> 이진 기술자는 추출 및 정합에 필요한 연산이 간단하여 회귀점 검출 역시 실시간으로 수행되었다.

일반적으로 회귀점 검출은 SIFT<sup>(21)</sup>나 SURF<sup>(22)</sup>와 같이 정교하지만 계산 시간이 많이 소요되는 특징점 기술자를 사용하였다. 주된 이유 중 하나는 이 특징점 기술자들이 반복성이 우수하고 정합이 잘 되기 때문이다. 하지만 이 기술자들은 프레임간의 특징점을 추적하는데는 잘 사용되지 않는다(보통은 Kanade-Lucas-Tomasi<sup>(23)</sup> 추적기를 사용한다). 이는 계산량이 많이 요구되기 때문이다. 즉, SIFT나 SURF를 이용하게 되면 특징점 추적기와는 별개로 관리해야 한다.

하지만 본 논문에서는 특징점 추적기는 이진 기술자(BRIEF)를 이용하고 이 때 사용된 이진 기술자를 이용하여 회귀점 역시 검출하는 방법을 제안한다. 이에 따라 회귀점 검출 및 최적화를 포함한 VSLAM을 실시간으로 수행할 수 있었다. 이때 사용된 영상은 1241x376 크기에 10 fps(frame per second)로 촬영되었다. 요약하면 다음의 과정을 매 프레임마다 수행한다.

- (1) 특징점 추출 : 새로운 영상으로부터 특징점(keypoints)을 추출한다.
- (2) 특징점 추적 : 이전 프레임의 특징점과 현재 프레임의 특징점을 정합한다.
- (3) 키프레임 추출 : 만약 이전 프레임으로부터 추적되어 온 특징점의 개수가 일정 이하이면 새로

운 특징점 추적을 위한 키프레임을 추가한다.

- (a) 키프레임 생성 : 새로운 키프레임의 이전 키프레임과의 상대 자세를 계산한다.
- (b) 회귀점 검출 : 2 단계에서 특징점 추적에 사용된 이전 기술자를 binary vocabulary tree 에 질의(query)하여 가장 유사한 기존 키프레임을 찾아낸다.
- (c) 국소 최적화(Local bundle adjustment) : 정해진 창 크기를 가지는 국소 최적화를 수행한다.
- (4) 자세 계산 : 만약 충분한 수의 특징점이 추적되고 있다면 3-pts 알고리즘<sup>(26)</sup>을 이용하여 현재 프레임의 자세를 계산한다.

3.2 특징점 추출 및 추적

본 연구에서는 키포인트 추출을 위해서 다해상도 FAST 코너를 추출하였다. 다해상도 FAST 코너는 영상 피라미드 전체에서 특징점을 추출하는 것으로 넓은 영역에서의 특징점 추적기 성능을 높여준다. 하지만 그만큼의 계산시간 역시 증가한다. 하지만 만약 계산 자원이 많지 않을 때에는 하나의 스케일에 대하여 FAST 코너를 사용해도 큰 무리는 없다. 영상은 7x4 의 격자로 나뉘어 FAST 코너를 추출하였다. 이는 영상 전역에 특징점을 분포함으로써 좀더 나은 자세 계산이 가능하기 때문이다.

프레임간의 특징점 추적을 위해서, 현재 프레임에서의 특징점은 이전 키프레임 특징점 좌표로부터 WxW 크기의 창 내에 존재할 경우 정합을 수행한다. 이를 위하여 256-비트의 BRIEF 기술자가 이용되었다. 이 방법은 불필요한 BRIEF 기술자 계산을 방지한다. 이 창 크기 내에서 BRIEF 기술자의 비교를 수행하며 가장 낮은 Hamming 거리값을 정합으로 간주한다. 하지만 가장 낮은 Hamming 거리값만을 이용할 경우 영상 특성에 따른 정합 오류가 발생할 수 있으므로, 가장 낮은 값(s0) 과 그 다음으로 낮은 Hamming 거리값(s1)의 비율(s0/s1)로서 정합을 찾는다. 이때 s0/s1 은 0.8 보다 작아야 한다고 가정하였다. 그리고 창 크기는 60x60 으로 설정하였다.

3.3 자세 추정

본 논문에서의 자세 추정은 미리 작성된 3 차원 랜드마크로부터 구하는 것과, 이러한 정보가 존재하지 않을 때 랜드마크를 초기화 하기 위한 상대 자세를 계산하는 두 가지 방법이 있다. 자세한 내용은 아래와 같다.

만약 SLAM 이 처음 수행된 경우 랜드마크는 존재하지 않는다. 랜드마크의 초기화를 위해서는 2D-to-2D 랜드마크 대응을 통하여 상대자세를 구

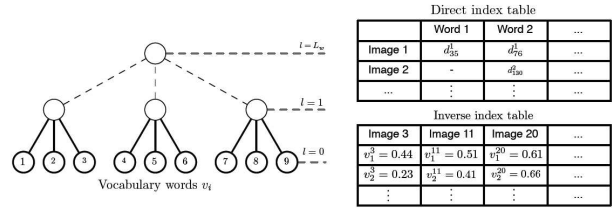


Fig. 2 Vocabulary tree with direct and inverse indices

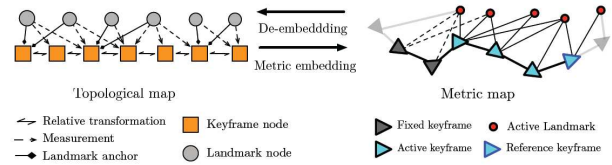


Fig. 3 Metric-topological map representation. We use both topological and metric map to represent the environment

해야 한다. 이때 5-pts 알고리즘<sup>(24)</sup>과 RANSAC<sup>(25)</sup>을 수행한다.

만약 SLAM 이 충분히 진행되어 3 차원 지도가 잘 작성된 경우, 각 3 차원 랜드마크들은 2 차원 특징점과의 정합을 통해 자세를 계산할 수 있다. 이렇게 2D-to-3D 정보가 존재할 경우 3-pts 알고리즘<sup>(26)</sup>은 자세 및 위치 (R,t)를 계산한다.<sup>(6)</sup> 이때 (R,t)는 Levenberg-Marquardt 알고리즘을 통하여 최적화한 값을 사용한다.

3.4 회귀점 검출

회귀점 검출은 먼저 장면인식(scene recognition) 으로부터 시작한다. 장면인식은 영상내 특징점을 Vocabulary tree 에 질의하여 초기 대응 프레임 번호를 출력한다. 현재 키프레임의 회귀점 검출을 위한 vocabulary tree 의 질의 결과는 다시 기하학적 검증을 통하여 검증을 수행한다. 이렇게 검증된 결과는 그래프상의 간선으로 추가된다.

회귀점 검출 후 정합을 위한 대응점을 찾는 데 있어 추가 계산을 방지하기 위하여 Direct Index 를 사용하였다.<sup>(17)</sup> Direct index 는 Vocabulary tree 의 Leaf node 가 추출된 영상에서의 특징점 번호를 기록해 놓은 것으로서 Vocabulary tree 의 질의 자체가 일종의 정합 기능까지 수행하는 것이다.

4. 환경맵의 표현 및 최적화

4.1 수치-위상지도 표현법

본 논문에서는 metric-topological 표현법을 사용하였다.<sup>(27)</sup> 이 표현법에서는 프레임간의 상대 자세 값은 간선으로 간주한다. 이와 같이 프레임간의 관계를 나타내는 위상지도(topological map)는 간선



의 적분을 통하여 수치지도(metric map)으로 변환된다(Fig. 3 참조).

4.2 그래프 최적화

자세와 랜드마크의 최적화를 위해서, 위상지도는 임베딩을 통하여 수치지도로 변환된다. 이때 그래프의 탐색은 깊이 우선탐색(breadth-first search)을 수행한다. 깊이 우선탐색에서는 우선순위 큐를 사용하게 되는데 이때 우선순위는 거리값이다.

최적화를 할 프레임의 개수는 일정 값으로 정해놓음으로서 실시간성을 확보한다. 수치 지도로 변환된 표현 상에서 bundle adjustment 를 수행한다. 수행 결과는 다시 위상지도로 변환되기 위하여 정점간의 상대좌표를 간선으로 인코딩 한다.

5. 실험 결과

본 논문에서는 제안된 알고리즘을 KITTI 데이터셋<sup>(1)</sup>에 평가하였다. 실험은 2.3 Ghz Intel Core i7 컴퓨터에서 수행되었으며, GPU 나 멀티스레딩 기법은 사용되지 않았다.

5.1 사용된 데이터셋

KITTI 데이터셋은 22 가지의 서로다른 데이터셋을 제공한다. 22 가지중 11 가지는 참조해(ground truth)를 제공하고 나머지는 평가용으로 서버에 결과를 전송하면 비교결과를 알려준다. 영상은 1241x376 크기를 가지는 흑백 영상이다. 촬영 당시의 속도는 0~90km/h 이다. 비선형 최적화는 Ceres-solver<sup>(28)</sup>를 활용하였으며, 해당 최적화 소프트웨어는 멀티 스레딩 기법을 사용하였다.

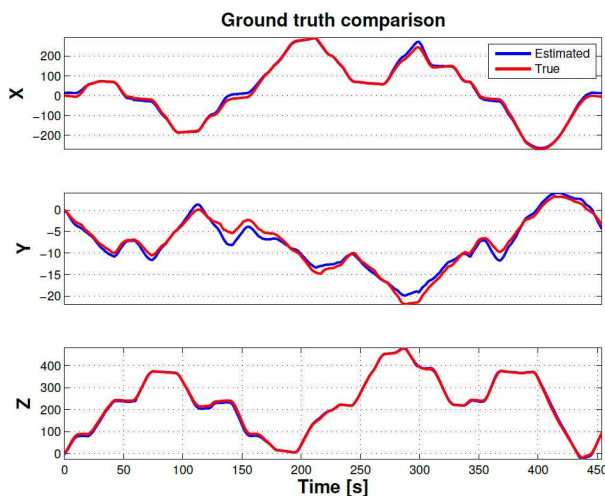


Fig. 4 Ground truth comparison result of sequence 0 of the KITTI dataset. The global arbitrary scale was obtained by the optimization process

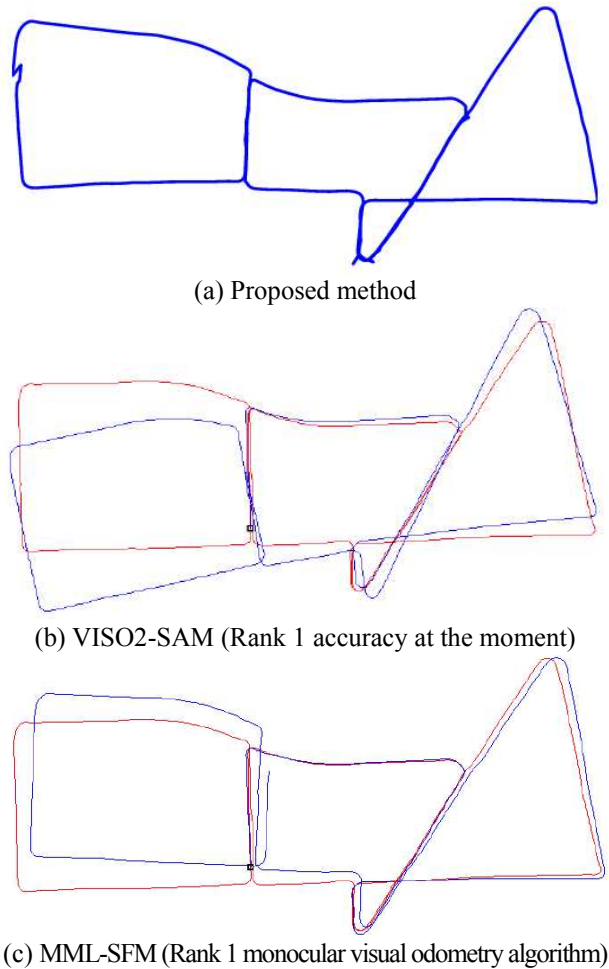


Fig. 5 Qualitative comparison with state-of-the-art method

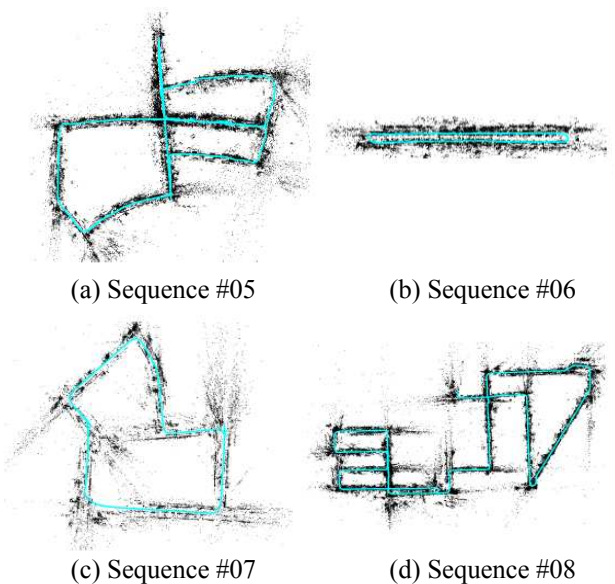


Fig. 6 Environment mapping result of KITTI dataset. Landmarks are denoted by black dots. Each number represents sequence identification number in KITTI dataset

## 5.2 참조값과의 비교

본 논문은 자동차의 높이 등 알려진 기하학적 요소를 이용하지 않는다. 따라서 한 대의 카메라로 계산한 자세값은 임의의 scale 에서도 만족한다. 즉, 얻어진 6 자유도 계산값을 참조값과 비교하기 위해서는 일정 변환 T 가 필요하다. 이 변환 T 를 SLAM 결과에 적용하여 정량적 분석을 수행할 수 있다. Fig. 4 는 시퀀스 0 에 대한 정량적 분석 결과이다. 계산된 오차값은  $2.9383 \pm 8.9118$  (평균  $\pm$  표준편차) 이다 (Fig. 4 참조). 참조값과의 정성적 비교 결과는 Fig. 5 와 같다.

## 6. 결 론

본 논문에서는 실시간 VSLAM 방법을 제안하였다. 하나의 통합된 이진 기술자의 이용으로 추가적인 계산 시간 없이 빠르게 회귀점 검출을 수행 할 수 있었다. 또한 회귀점 검출 이후 빠른 최적화를 위하여 metric-topological 표현법을 사용하였다. 제안된 알고리즘은 KITTI 데이터셋을 통하여 검증되었다.

## 후 기

본 연구는 미래창조과학부/한국연구재단 우수연구센터 육성사업의 부분 지원으로 수행되었음 (과제번호 NRF-2009-0083495).

## 참고문헌 (References)

- (1) Geiger, A., Lenz, P. and Urtasun, R., 2012, "Are We Ready for Autonomous Driving? the Kitti Vision Benchmark Suite," in *CVPR. IEEE*, pp. 3354-3361.
- (2) Davison, A. J., 2003, "Real-time Simultaneous Localisation and Mapping with a Single Camera," in *ICCV. IEEE*, pp. 1403-1410.
- (3) Klein, G. and Murray, D., 2007, "Parallel Tracking and Mapping for Small AR Workspaces," in *ISMAR. IEEE*.
- (4) Strasdat, H., Montiel, J. and Davison, A., 2010, "Scale Drift-Aware Large Scale Monocular Slam," in *Proceedings of Robotics: Science and Systems (RSS)*, p. 5.
- (5) Strasdat, H., Montiel, J. and Davison, A. J., 2010, "Real-time Monocular Slam: Why Filter?" in *ICRA. IEEE*, pp. 2657-2664.
- (6) Nist'er, D., Naroditsky, O. and Bergen, J., 2004, "Visual Odometry," in *CVPR IEEE, 2004.*, Vol. 1, pp. 1-652.
- (7) Mouragnon, E., Lhuillier, M., Dhome, M., Dekeyser, F. and Sayd, P., 2006, "Real Time Localization and 3D Reconstruction," in *CVPR IEEE*, Vol. 1, pp. 363-370.
- (8) Konolige, K. and Agrawal, M., 2008, "Frameslam: from Bundle Adjustment to Real-time Visual Mapping," *Robotics, IEEE Transactions on*, Vol. 24, No. 5, pp. 1066-1077.
- (9) Snavely, N., Seitz, S. M. and Szeliski, R., 2008, "Skeletal Graphs for Efficient Structure from Motion," in *CVPR*, Vol. 2.
- (10) Kümmerle, R., Grisetti, G., Strasdat, H., Konolige, K. and Burgard, W., 2011, "g2o: A General Framework for Graph Optimization," in *ICRA*.
- (11) Lim, J., Frahm, J.-M. and Pollefeys, M., 2011, "Online Environment Mapping," in *CVPR. IEEE*, pp. 3489-3496.
- (12) Strasdat, H., Davison, A. J., Montiel, J. and Konolige, K., 2011, "Double Window Optimisation for Constant Time Visual Slam," in *ICCV. IEEE*, pp. 2352-2359.
- (13) Engels, C., Stew'enius, H. and Nist'er, D., 2006, "Bundle Adjustment Rules," *Photogrammetric Computer Vision*, Vol. 2.
- (14) Cummins, M. and Newman, P., 2008, "Fab-map: Probabilistic Localization and Mapping in the Space of Appearance," *The International Journal of Robotics Research*, Vol. 27, No. 6, pp. 647-665.
- (15) Nister, D. and Stewenius, H., 2006, "Scalable Recognition with a Vocabulary Tree," in *CVPR. IEEE*.
- (16) Calonder, M., Lepetit, V., Strecha, C. and Fua, P., 2010, "Brief: Binary Robust Independent Elementary Features," *Computer Vision-ECCV 2010*, pp. 778-792.
- (17) Galvez-Lopez, D. and Tardos, J. D., 2011, "Real-time Loop Detection with Bags of Binary Words," in *IROS. IEEE*, pp. 51-58.
- (18) Williams, B. and Reid, I., 2010, "On Combining Visual Slam and Visual Odometry," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on. IEEE*, pp. 3494-3500.
- (19) Mei, C., Sibley, G., Cummins, M., Newman, P. and Reid, I., 2011, "Rslam: A System for Large-Scale Mapping in Constant-time Using Stereo," *International Journal of Computer Vision*, Vol. 94, No. 2, pp. 198-214.
- (20) Rosten, E. and Drummond, T., 2006, "Machine Learning for High-speed Corner Detection," in *ECCV. Springer*,
- (21) Lowe, D. G., 2004, "Distinctive Image Features from Scale-Invariant Keypoints," *IJCV*, Vol. 60, No. 2, pp. 91-110.
- (22) Bay, H., Tuytelaars, T. and Van Gool, L., 2006, "Surf: Speeded up Robust Features," in *ECCV. Springer*, pp. 404-417.
- (23) Lucas, B. D., Kanade, T., 1981, "An Iterative Image

- Registration Technique with an Application to Stereo Vision.” in *IJCAI*, Vol. 81, pp. 674~679.
- (24) Nist'er, D., 2004, “An Efficient Solution to the Five-point Relative Pose Problem,” *PAMI*, Vol. 26, No. 6, pp. 756~770.
- (25) Fischler, M. A. and Bolles, R. C., 1981, “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography,” *Communications of the ACM*, Vol. 24, No. 6, pp. 381~395.
- (26) Kneip, L., Scaramuzza, D. and Siegwart, R., 2011, “A Novel Parametrization of the Perspective-Three-point Problem for a Direct Computation of Absolute Camera Position and Orientation,” in *CVPR. IEEE*, pp. 2969~2976.
- (27) Lim, J., Frahm, J.-M. and Pollefeys, M., 2012, “Online Environment Mapping Using Metric-topological Maps,” *The International Journal of Robotics Research*, Vol. 31, No. 12, pp. 1394~1408.
- (28) Agarwal, S., Mierle, K. and Others, “Ceres Solver,” <https://code.google.com/p/ceres-solver/>.
- (29) Geiger, A., Ziegler, J. and Stiller, C., 2011, “Stereoscan: Dense 3d Reconstruction in Real-time,” in *Intelligent Vehicles Symposium (IV), IEEE*, pp. 963~968.