**KUSARF 자료조사**

**주제: Visual SLAM (feature based)의 심화**

1. **Visual SLAM이 무엇인가?**

SLAM이란 localization과 mapping을 동시적으로 하는 task를 의미하고, 실내 항법 뿐만 아니라 자율주행이 앞으로 나아가야 할 방향 중 고정밀 지도 작성에는 SLAM을 할 필요가 있다.

Visual SLAM은 이미지 센서를 통해 얻는 데이터를 가지고 SLAM을 구현하는 기술을 의미한다. 이는 카메라가 주일 것이다. 즉, 카메라에서 얻은 영상을 가지고 실시간으로 현재 위치를 추정하고, 그곳의 map도 작성해야 한다는 것이다. 보통 Visual SLAM의 접근법은 이미지의 특징점을 이용하여 구현하는 방식이 대표적이다. 픽셀의 intensity 값을 이용하여 접근하는 경우도 있지만 여기서 제외하도록 한다.

* **특징점?**

특징점이란 간단히 말해서, 이미지에서 의미있는 정보를 담고 있는 픽셀이라고 할 수 있다. 어떤 장면에서 찍든, 특징점은 구분 가능해야 할 것이다. 물체의 형태 또는 크기 그리고 위치가 변해도 특징점은 인식이 쉽게 돼야하고, 카메라의 시점이나 조도가 변해도 인식할 수 있어야 할 것이다.

대표적으로 SIFT 알고리즘이나 ORB feature extrator, harris corner 등이 있다. 구체적인 이론은 생략한다. Opencv 라이브러리를 이용하면 쉽게 사용할 수 있다.

텍스트, 표지판이(가) 표시된 사진

자동 생성된 설명

그림 출처: <https://darkpgmr.tistory.com/131>

* **특징점 매칭**

완전히 다른 장면인 경우가 아닌, 얼마 차이 안나는 frame을 얻었다고 가정해보자. 예를 들면 아래와 같다.



이미지 출처: <http://www.gisdeveloper.co.kr/?p=6824>

다른 사진이라도 같은 영역이 있을 때, 특징점을 가지고 잘 매칭이 된다면 공통 구역을 찾아낼 수 있을 것이다. 즉, 특징점 매칭이랑 특징 묘사자의 유사도를 정의하고 서로 다른 이미지에서 등장한 유사한 특징끼리 서로 짝을 지어주는 과정이다. 보통, 유사도 측정은 특징 묘사자 벡터 거리 측정 방식이 주를 이루게 된다. 이를 수행하면 다른 위치에서 같은 landmark를 바라볼 때, 이를 인지할 수 있을 것이다. 마찬가지로 opencv 예제를 쉽게 찾을 수 있다.

* **사진 측량**

위처럼, 다양한 위치에서 같은 landmark를 바라본 후, 그 landmark의 위치를 추정할 수 있는 기술이라고 이해하면 쉽다. 대표적인 Triangulation은 삼각 측량법으로 예시는 아래 그림과 같다.

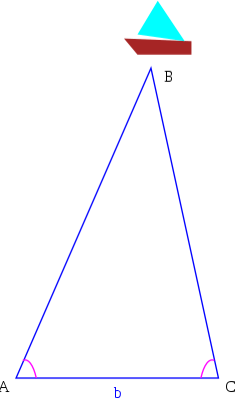


사진 출처: <https://en.wikipedia.org/wiki/Triangulation>

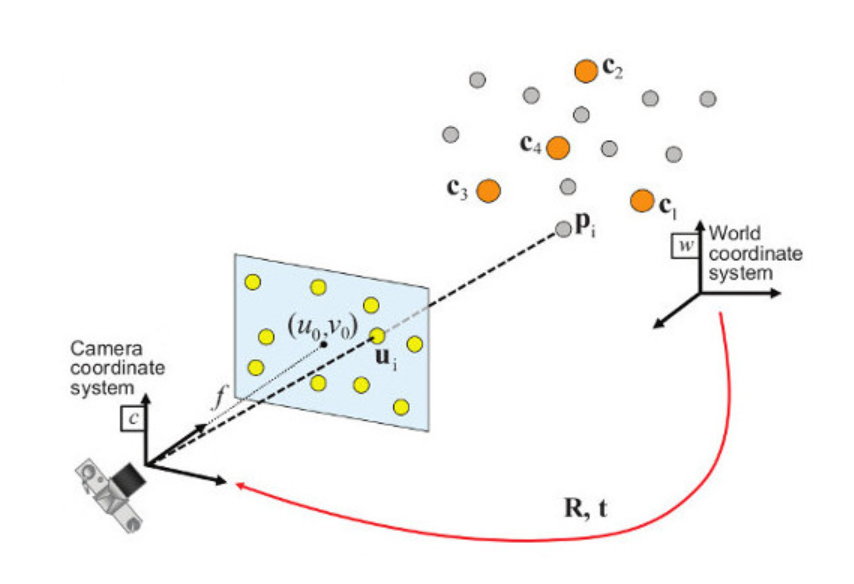
즉, 2D에서 3D를 찾는 과정이다. 조금 더 깊게 들어가려면 아래 블로그를 참고하면 좋을 듯 하다.

<https://m.blog.naver.com/dnjswns2280/222127017232>

* **Reprojection error?**

3D plane에서 2D plane으로 투영되는 것을 projection이라고 흔히 칭한다면 reprojection은 이의 반대로 3D plane 좌표로 투영하는 것을 의미할 것이다. Reprojection error란 추정된 3D 점이 얼마나 실제와 똑같이 했느냐에 대한 성능 지표이다. SLAM에서는 이를 최소화시킬 필요가 있다

* **3D point와 이에 대응하는 2D point를 이용한 camera 자세 및 위치 추정**



출처: <https://docs.opencv.org/4.x/dc/d2c/tutorial_real_time_pose.html>

텍스트이(가) 표시된 사진

자동 생성된 설명

PnP 알고리즘은 카메라의 intrinsic matrix와 카메라가 찍은 특징점에 대한 3D point와 이에 해당하는 2D projection을 가지고 있다면 이에 맞는 [R|t] 행렬(회전 및 translation matrix)을 구하는 것이다. 위의 출처에 opencv가 제공하는 예제를 올렸기에 참고하면 좋을 것이다.

카메라의 intrinsic matrix는 카메라 calibration으로 얻을 수 있으며, 마찬가지로 opencv에서 쉽게 수행할 수 있다. 링크는 아래에 첨부하였다.

링크: <https://docs.opencv.org/4.x/dc/dbb/tutorial_py_calibration.html>

이렇게 카메라가 이동하면서 특징점을 토대로 매칭을 해가면서 내 현재 카메라 위치를 추정하고 추정된 특징점의 위치로는 map 작성을 할 수 있다. 이것이 visual slam의 기본적인 동작 원리이다.

1. **SLAM 알고리즘에서 많이 쓰이는 용어들**

* **Loop closure detection**

쉽게 말해 이전에 방문한 곳인지를 찾는 문제이다. 유사도를 판별하여 진행하게 될 것이다.

.

* **Bundle adjustment**

특징점들의 위치로 추정 가능한 3D landmark position과 카메라 프레임 간의 3D relative motion을 동시에 최적화하는 과정이다. 보통은 SLAM에서 sliding window 최적화 방식이나, 따로 thread를 돌려 loop closure가 일어났을 때 최적화하게 된다.

Global 하게 수행하는 경우에는 다른 thread에서 map과 pose를 모두 최적화한다.

최적화를 할 때 중요한 요소로 작용하는 것은 cost function이다. 바로 이 때 사용하는 것이 3페이지에서 언급했던 reprojection error이다. 보통 SLAM에서 initialization 후에 추정해가는 방식을 이용하는데 이 과정에서 error는 생길 수 밖에 없다.

조금 더 깊은 내용은 아래 블로그에서 참고하면 좋다.

링크: <http://www.cv-learn.com/20210313-ba/>

* **Pose graph optimization**

Pose graph가 graph 자료구조를 바탕으로 만들어진 것이다. Graph는 G(V,E)로 표현할 수 있고 vertex와 edge로 표현된다. 여기서 vertex를 current 위치로 설정하고 edge를 둘 사이의 자세로 표현한 것이 pose graph이다. Pose graph optimization 과정은 모든 vertex를 최적화하는 과정이다.

자세한 설명은 아래 블로그에서 참고하면 좋다.

링크: <https://edward0im.github.io/engineering/2020/09/08/pose-graph-optimization/>

* **Place recognition**

다른 시간에 얻어진 같은 장소의 다른 view를 matching 하는 task를 의미한다. learning으로도 꽤 많이 접근한다.

참고 링크: <https://paperswithcode.com/task/visual-place-recognition>

1. **(Additional) 대표적 visual SLAM 알고리즘 ORB-SLAM 초기버전 논문 리뷰**

* **Abstract**

ORB-SLAM은 크고 작은 실내 또는 실외 환경에서 실시간으로 작동하는 feature-based SLAM system이다. 심한 motion clutter에도 강건하다. 또한 폭 넓은 baseline loop closing과 relocalization, 완전 자동화된 초기화 기능을 포함한다고 설명한다. 최근의 훌륭한 알고리즘에 기반하여 모든 SLAM 과제에 사용하는 시스템을 디자인하였다. 이것은 tracking, mapping, relocalization, loop closing을 포함한다. 이 당시에, monocular SALM 접근법 중 SOTA를 달성했다고 설명한다. monocular SLAM 접근법은 Visual SLAM의 일종이다. Visual SLAM은 크게 3가지가 있으며 이는 아래와 같다.

1. Monocular: 1개의 카메라

2. Stereo: 2개의 고정 카메라

3. RGB-D: 카메라 + depth

사실상 Stereo 카메라로도 depth를 추정할 수 있지만 RGB-D 기반이 더 정확할 것이다.

* **Introduction**

BA(Bundle Adjustment)는 강한 매치 네트워크와 좋은 초기 추정이 제공되면 sparse한 기하학적 reconstruction 뿐 아니라 정확한 카메라 위치 추정을 제공한다고 알려져있다. 그러나 오랫동안 실시간 적용에서 연산이 너무 많다고 여겨졌다. Visual SLAM은 환경을 reconstruct 하는 동안 카메라 경로를 추정하는 목표를 가진다. 또한 실시간 SLAM 알고리즘을 위해서는 아래를 따르는 BA를 제공해야 한다고 설명한다.

1) Corresponding observations of scene features (map points) among a subset of selected frames (keyframes).

(keyframe 사이에 대응하는 map point들의 관찰)

2) As complexity grows with the number of keyframes, their selection should avoid unnecessary redundancy.

(keyframe의 수에 따라 복잡도가 증가하기에 불필요한 중복을 피하기.)

3) A strong network configuration of keyframes and points to produce accurate results, that is, a well spread set of keyframes observing points with significant parallax and with plenty of loop closure matches.

(정확한 결과 제공을 위한 keyframe들과 point들의 강한 연결 배열 )

4) An initial estimation of the keyframe poses and point locations for the nonlinear optimization.

(비선형 최적화로 초기 자세 및 위치 추정)

5) A local map in exploration where optimization is focused to achieve scalability.

(최적화가 확장성을 이루는 것에 초점을 둔 탐색을 한 local map)

6) The ability to perform fast global optimizations (e.g., pose graph) to close loops in real time.

(실시간으로 loop를 close 하는 빠른 global 최적화 알고리즘을 수행하기 위한 능력)

이 논문에서 PTAM(Parallel Tracking And Mapping)의주요 아이디어를 가져오고 아래와 같은 주요 contribution을 가지고 있다.

1) Use of the same features for all tasks: tracking, mapping, relocalization, and loop closing

또한 ORB feature를 사용한다고 나와있다. ORB는 FAST keypoint 추출 방법과 BRIEF descriptor가 합쳐진 기술이며 GPU 없이 실시간 작업을 수행할 수 있었다.

2) Real-time operation in large environments.

여기서 covisibility graph 사용으로 tracking과 mapping을 global map size와 독립적인 지역적인 covisible 영역에서 초점 맞추어 수행할 수 있다고 언급한다.

covisibility graph는 아래 tistory를 참고하면 간단한 개념을 알 수 있다.

<https://elecs.tistory.com/310>

3) Real-time loop closing based on the optimization of a pose graph that we call the Essential Graph.

여기서 pose graph를 언급하고 이를 optimization 하여 real-time loop closing을 하게 된다.

4) Real-time camera relocalization with significant invariance to viewpoint and illumination.

(보는 곳과 빛에 불변한 실시간 카메라 재위치파악)

5) A new automatic and robust initialization procedure based on model selection

(model 선택에 기반한 자동화되고 강건한 초기화 절차 수립)

6) A survival of the fittest approach to map point and keyframe selection

(map point와 키프레임 선택에 가장 알맞은 접근법)

인기있는 데이터셋에 대해 많은 평가를 진행했다고 하며 direct method를 제치고 좋은 성능을 달성했다고 한다. direct method는 feature-based가 아니라 pixel의 intensity를 기반으로 직접적으로 수행하는 방식이다. loop closing과 relocalization 방법은 이전에 있었던 방법 기반이다. 여기서는 초기화 방법을 추가하고 모든 building block들을 세부적으로 묘사하며 실험 검증을 수행한다고 설명한다.

* **Related Work**

A. Place Recognition

Place recognition task란, 같은 장소의 다른 view를 봤을 때 matching 하는 task이다.

image-to-image 매칭 방법이 scale better라고 설명한다. 이는 apperance-based 방식이며 이 방식 내에서 FAB-MAP과 같은 확률적 접근법과 같은 bags of words는 그들의 높은 효율성 때문에 주목을 받았다. DBoW2가 처음으로 bag of binary words를 사용하였고 이것은 feature extraction에서 필요한 시간을 SURF와 SIFT에 비해 현저히 줄일 수 있었다. 여기서는 ORB를 가지고 DBoW2에 사용된 bag of words를 제안한다고 나와있다. ORB는 이진 특징 기술자이고 회전과 스케일에 불변하며 빠른 것 등에 장점이 있다. 4개의 다른 데이터셋에서 recognizer의 높은 recall과 강건함을 볼 수 있었고 시간도 39ms 미만으로 상당히 적게 걸렸다. 이번 연구에서, covisibility 정보와 데이터베이스에 물을 때 몇가지 가설을 return 함으로써 향상된 plcae recognizer를 사용하였다.

B. Map Initialization

Monocular SLAM은 하나의 이미지로부터 depth 추출이 불가능하기 때문에 초기 map을 만드는 절차가 필요하게 된다. 이 논문에서 planar 장면에 homography와 nonplanar 장면에 기초적 matrix 사이에 모델 선택에 기반한 새로운 자동화된 접근법을 주게 된다. 여기서 바람직하지 않은 상황에 가까운 기초적 matrix를 선택할 위험을 고려하는 휴리스틱한 초기화 알고리즘을 개발하게 된다. planar case에서 만일 twofold ambiguity가 발생한다면 초기화를 삼가하게 된다. 즉, method가 유니크한 solution을 낼 때까지 초기화를 딜레이하게 된다.

C. Monocular Simultaneous Localization and Mapping

filtering을 사용해서 Monocular SLAM을 접근하는 방식은 computation 낭비에 결함이 있었다. keyframe에 기반한 접근들은 오직 선택된 frame을 사용하여 map을 추정하기 때문에 cost가 크지만 정확한 BA 최적화가 가능하게 된다. 또한 keyframe에 기반한 기법들이 filtering에 기반한 기법보다 같은 계산량에 비교하면 더 정확하다고 한 논문을 설명하고 있다. 이 keyframe 기반 기법 중 가장 대표적인 방법은 PTAM이며 이는 camera traking과 mapping을 parallel thread로 분리하면서 작은 환경에서 real time application이 더 가능해졌다.

다음으로 GPU에서 실행된 optical flow에 기반한 front-end 방식을 설명한다. Loop closure들은 pose graph optimization으로 해결하였고 이번 work에서도 7-DoF pose graph optimization을 loop closing에 사용하게 된다.

이 시스템에서 covisibility에 기반한 local map을 사용하고 covisibility graph로 부터 pose graph를 얻고 그것들을 완전히 다시 고안된 front-end와 back-end에서 적용한다. 또다른 차이는 loop detection을 위해 구체적인 feature를 사용하는 것 대신에, 같은 트래킹되고 매핑된 feature에 대해 place recognition을 수행한다고 나와있다.

CD-SLAM은매우 완전한 SLAM 시스템을 제안하지만 map initialization에 대해 언급이 없었다.

ORB feature를 사용한 work, LSD-SLAM 같은 잘 알려진 방식도 소개하고 있으며 약간의 단점들을 다 언급하고 있다.

여기서 제안한 keyframe selection 방법은 가능한 빨리 keyframe들을 삽입하고 중복된 것들 나중에 제거함으로써 어려운 시나리오에서 전례없는 강건함을 가졌다.

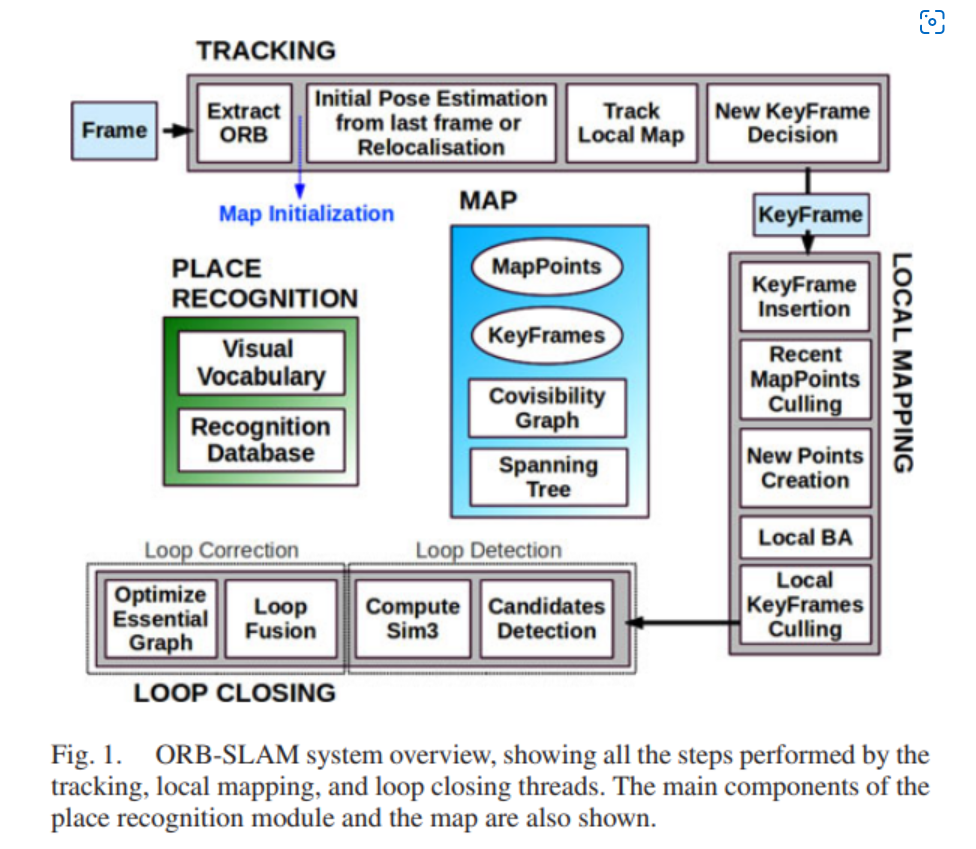
* **System Overview**

A. Feature Choice

이 시스템에서 주된 아이디어 중 하나는 mapping과 tracking에 의해 사용됐던 feature들이 frame-rate relocalization과 loop detection을 수행하기 위한 place recognition에서 사용된다는 것이다. 이것은 시스템을 효율적으로 만들고 depth interpolation을 할 필요가 없다고 설명한다. 또한 이미지 당 33ms 미만의 feature 추출 시간과 회전 불변 성질이 필요했다. SIFT와 SURF, BRIEF 같은 것은 이것 때문에 바로 제외된다. 그래서 ORB를 선택했고 이것은 viewpoint에 좋은 invariance 성질을 가지고 계산이 상당히 빨랐다.

B. Three Threads: Tracking, Local Mapping, and Loop Closing

아래 그림은 시스템 아키텍처이다.



이 아키텍처를 보면 tracking, local mapping, loop closing task가 각각 병렬로 3개의 thread를 통해 돌아감을 볼 수 있다.

tracking - 모든 프레임을 가지고 카메라 localization과 새로운 keyframe이 삽입될 때를 결정한다. 처음 이전 프레임과 initial feature matching을 수행하고 motion-only BA를 사용함으로써 pose를 최적화한다. 만일 tracking을 occulusion과 같은 상황에 따라 잃게 된다면 place recognition 모듈이 global relocalization을 수행하기 위해 사용된다. 일단 카메라 자세와 특징점 매칭의 초기 추정을 가지고 있다면 local visible map가 철회되고 local map point들을 가진 match들이 reprojection에 의해 탐색된다. 그리고 카메라 자세가 다시 모든 매치들과 함께 최적화된다.

local mapping - 이 과정은 새로운 keyframe들을 가지고 카메라 자세의 surrounding에 optimal한 reconstruction을 수행하기 위해 local BA를 수행하게 된다. 새로운 keyframe에서 매칭되지 않은 ORB의 새로운 대응점은 연결된 keyframe에서 탐색된다. 만들어진 약간의 시간 후에 tracking 동안 모아진 정보에 기반하여 급박한 point 고르기가 오직 높은 퀄리티를 가진 포인트만 남기기 위해 수행된다. local mapping은 또한 중복된 keyframe을 고르는 역할도 하게 된다.

loop closing - 이 단계에서 매번 새로운 프레임으로 loop를 탐색하게 된다. loop가 감지되면 loop에서 모아진 drift에 대해 알리는 similarity transformation을 계산하게 된다. 그리고 loop의 both side들이 정렬되고 복제된 포인트들은 fuse 된다. 그 후에 similarity constraint들에 기반한 pose graph optimization이 global한 일관성을 가지기 위해 수행된다.

C. Map Points, Keyframes, and Their Selection

각각의 map point는 아래와 같은 사항들을 저장한다.

1) world 좌표계에서 3-D 위치 정보

2) 바라보는 방향

3) 대표적인 ORB descriptor (해밍 거리로 association이 이루어짐.)

4) point가 관찰될 수 있는 최대 그리고 최소 거리

또한 각각의 keyframe은 아래와 같은 사항들을 저장한다.

1) 카메라 자세

2) 카메라 내부 파라미터들 (focal length, principal point 등)

3) 프레임에서 추출된 모든 ORB 특징들

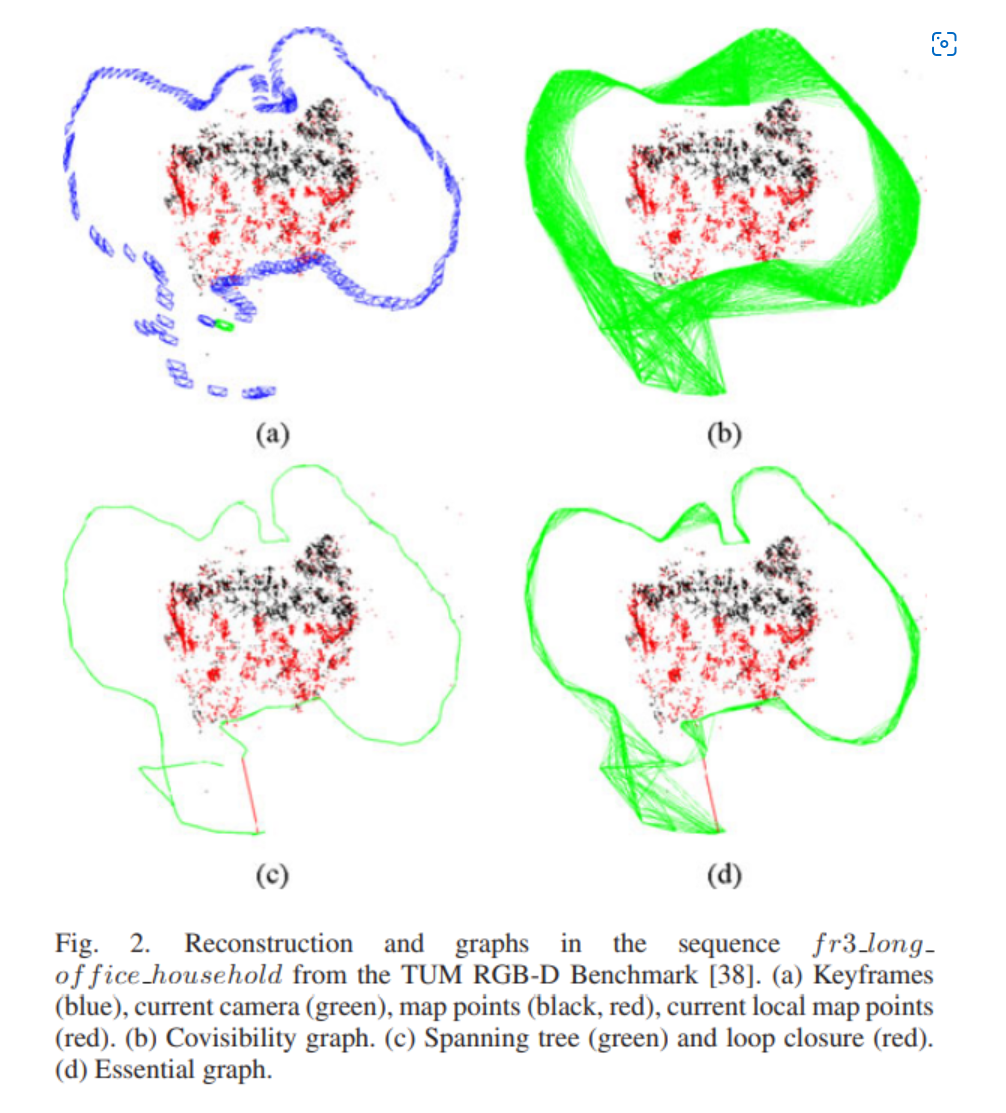
Map point들과 keyframe들은 일반적인 정책으로 생성되지만 후에 매우 exigent한 culling 메커니즘이 중복되는 keyframe들과 잘못 매치되거나 tracking 될 수 없는 map point들을 탐지하여 탐색 동안 유연한 맵 확장이 가능하진다. 추가적으로 PTAM과 비교하여 outlier들을 거의 포함하지 않고 있다고 설명한다.

D. Covisibility Graph and Essential Graph

각각의 노드가 keyframe이고 두 keyframe 사이 에지는 그들이 같은 map point들을 적어도 15개 공유하고 가중치 theta를 가지면서 존재하게 된다. loop를 정확하게 하기 위해서 pose graph optimization을 수행하게 된다. 또한 covisibility graph에 의해 제동되는 모든 에지들을 포함하지 않기 위해서 Essential Graph라는 것을 제안한다. 이것은 모든 노드들을 유지하지만 더 적은 에지를 가지게 된다. 시스템은 초기 keyframe으로부터 스패닝 트리를 구축하고 이것은 최소 에지 수를 가진 covisibility graph의 연결된 subgraph를 제동한다.

새로운 keyframe이 삽입됐다면, 가장 많은 point observation을 공유하는 keyframe에 연결된 tree에 포함된다. 그리고 keyframe이 앞서 언급된 culling policy에 의해 지워졌다면, 시스템은 그 keyframe에 영향 받은 link를 업데이트 한다.

그래서 Essential graph는 스패닝 트리, 높은 covisibility를 가진 covisibility graph로부터의 에지의 subset, loop closure edge들을 포함하게 된다. 아래 그림은 covisibility graph, 스패닝 트리, associated essential graph의 예시를 보여준다.

​

E. Bags of Words Place Recognition

시스템은 임데딩된 bags of words place recognition 모듈을 가진다. 그리고 이것은 loop detection과 relocalization을 수행하게 된다. Visual word들은 descriptor 공간의 discretization이다. 이것은 visual vocabulary로 알려져있다. 이 vocabulary는 수많은 이미지 셋들로부터 추출된 ORB descriptor를 가지고 offline으로 만들어진다.

만일 이미지들이 충분하다면 같은 vocabulary는 좋은 성능을 가지는 다른 환경에서 사용될 수 있다.

keyframe들 사이에 visual overlap이 존재하기 때문에 높은 score를 특별히 가지는 keyframe이 존재하지 않을 것이다. 그래서 DBoW2는 시간 축에서 가까운 이미지에게 점수를 더 줌으로써 overlapping을 고려하였다.다만 이는 한계가 있고 이것 대신에 covisibility graph에 연결된 keyframe를 가지고 grouping 하게 된다.

또한 bag of words를 사용하면 추가적인 이득이 또 있는데 ORB 특징의 두 set 사이에 대응점을 계산하려고 할 때 특정한 level에서 vocabulary tree에 같은 노드에 속하는 특징들에게만 제약을 걸 수 있다. 이 trick을 loop 검출과 relocalization, 그리고 새로운 point들의 triangulation 계산을 위한 매치들을 찾는데 사용된다.

* **Automatic Map Initialization**

map initialization의 목적은 두 프레임 사이에 상대적인 자세를 계산하는 것이다. 여기서 두 개의 기하학적 모델을 병렬로 계산하는 것을 제안한다. 이는 planar scene을 가정하는 호모그래피와 nonplanar scene을 가정하는 fundamental matrix이다. 그 후에 모델을 선택하기 위해서 휴리스틱을 사용하고 선택된 모델을 위한 구체적인 방법으로 relative pose를 복구하게 된다. 알고리즘은 아래와 같은 순서를 가지게 된다.

1) Find initial correspondences : 이 단계에서는 ORB 특징점을 현재 frame에서 뽑은 다음 reference frame에서 match들을 찾게 되는데 매치들이 충분하지 않으면 reference match를 reset 하게 된다.

2) Parallel computation of two models : 여기서는 병렬로 homography와 fundamental matrix를 계산하게 된다. 이는 아래와 같이 나타난다.

텍스트이(가) 표시된 사진

자동 생성된 설명

H\_cr = homography이며 F\_cr은 fundamental matrix이다. x\_c, x\_r은 각각 current frame, reference frame의 feature를 의미하게 된다.

위 식은 RANSAC scheme를 가진 eight-point 알고리즘이나 normalized DLT 등으로 풀게 된다.

iteration 수는 미리 고정돼야 한다.

eight for the fundamental matix, and four of them for the homography라고 나와있다.

각 iteration에서 모델에 대한 score를 계산하게 되는데 이는 아래와 같다.

텍스트이(가) 표시된 사진

자동 생성된 설명

d\_cr^2과 d\_rc^2은 한 프레임부터 다른 프레임까지의 symmetric transfer error이며 T\_M은 카이제곱 검정 95%에 기반한 outlier rejection threshold이다.

만약 충분한 inlier가 없다면 다시 1단계로 돌아가게 된다.

​3) Model selection : 만일 scene이 planar, nearly planar, low parallax를 가지는 상황과 비슷하다면 homography로 설명가능하다. 반면에 nonplanar scene이라면 fundamental matrix로 설명될 수 있을 것이다. 그래서 scene에 따라서 잘 선택할 필요가 있으며 이 논문에서 robust 휴리스틱을 아래와 같이 계산하여 homography이냐 fundamental matix이냐를 선택하게 된다.

텍스트이(가) 표시된 사진

자동 생성된 설명

R\_H > 0.45 이면 homography이며 이외에는 fundamental matrix를 선택하게 된다.

4) Motion and structure from motion recovery : 우선 structure from motion을 알 필요가 있는데 쉽게 말해서 2차원 이미지 sequence로 부터 3d structure를 recovery하는 과정이라고 생각하면 된다. 여기서는, homography의 경우에 eight solution을 직접적으로 triangulate하는 것을 제안하게 된다. 만일에 분명한 winner solution이 발견되지 않으면 다시 1단계로 돌아가게 되고 initialization하지 않는다.

fundamental matrix일 경우에는 이것을 calibration matrix(K)를 사용하여 essential matrix로 변환하게 된다.

​텍스트이(가) 표시된 사진

자동 생성된 설명

그 다음에 네 개의 motion hypothese들을 가져온 뒤에 triangulate 한다.

​5) Bundle adjustment : initial reconstruction을 개선하기 위해서 full BA를 수행한다.

* **Tracking**

A. ORB Extraction

우선 8개의 scale level을 가진 FAST corner들을 추출하게 된다. homogeneous 한 분포를 보장하기 위해서, 그리드에서 각 scale level로 나누고 셀 당 최소 5개의 특징점을 추출하게 된다. 만일 충분한 corner들이 발견되지 않는다면 adapting threshold를 적용한다. 셀 당 남아있는 corner들의 양은 몇몇의 셀들이 코너들을 가지지 않으면 adapt되게 된다. 남아있는 FAST corner들로 방향과 ORB descriptor가 계산된다.

B. Initial Pose Estimation From Previous Frame

등속도 motion model을 사용하여 camera pose를 예측하고 guided search를 수행하게 된다. 만일에 match들이 충분히 발견되지 않았다면 last 프레임에서 더 많이 map point들을 탐색하게 된다. 그런 다음 발견된 대응점으로 최적화된다.

C. Initial Pose Estimation via Global Relocalization

tracking을 잃었을 때 frame을 bag of words로 변환하고 데이터베이스에 query하게 된다. 이를 통해 global relocalization을 수행하게 된다. 각 keyframe에서 map point와 연관된 ORB 대응점을 찾고 RANSAC iteration을 돌려 pnp 알고리즘을 사용하여 카메라 pose를 찾게 된다.

D. Track Local Map

카메라 pose 추정과 특징점 매치 initial set을 가지고 있으면 frame으로 map을 투영할 수 있고 더 많은 map point 대응점을 탐색할 수 있다. 큰 맵을 사용하면 복잡도가 커지기에 local map에 project 하게 된다.

local map은 (set of Keyframes(K\_1), neighbors to the keyframe(K\_1) --> K\_2, reference keyframe (K\_ref) --> 이는 K\_1에 포함된 것)을 포함하게 된다.

그리고 K\_1과 K\_2에서 발견된 map point는 아래와 같은 절차로 현재 프레임에서 탐색된다.

1) 현재 frame에서 map point projection을 계산하고 만일 image bound를 넘어가면 버린다.

2) 현재 보고있는 ray(v)와 map point가 평균적으로 보고있는 방향 (n)사이의 방향을 계산하게 된다. v\*n < cos(60도) 이면 버리게 된다.

3) map point로 부터 camera 센터까지의 거리 d를 계산하게 된다. 만일 map point의 scale invariace region을 벗어나면 버리게 된다. 여기서 범위는 [d\_min, d\_max]

4) d/d\_min을 계산하여 frame에서 scale을 계산한다.

5) frame에서 매칭되지 않은 ORB 특징점과 map point의 대표적 descriptor D을 비교한다. 그리고 best match를 가진 map point를 associate 시킨다. (예측된 scale, near x에 따라서)

카메라 pose는 frame에서 발견된 모든 map point들에 따라 최적화된다.

E. New Keyframe Decision

이제 tracking의 마지막 단계인데 만일 현재 프레임이 새로운 keyframe에서 소환될지 안될지를 결정하는 것이다.

새로운 keyframe을 삽입하기 위해서 아래와 같은 조건을 만족시켜야 한다.

1) 20 frame 이상이 last global relocalization으로부터 지나야 한다.

2) Local mapping이 idle 한 상태이거나 last keyframe 삽입으로부터 20 프레임이상이 지나가야 한다.

3) 현재 프레임이 최소 50 point들 이상 track 해야한다.

4) 현재 프레임이 reference keyframe (K\_ref)보다 90%의 point들 미만 track 해야한다.

4번 condition을 통해 최소 visual change를 도입하고 condition 1은 좋은 relocalization을 보장하게 해주고 condition 3 는 좋은 tracking을 보장한다.

* **Local Mapping**

A. Keyframe Insertion

처음, 새로운 노드 추가, 에지 업데이트들을 수행하여 covisibility graph를 업데이트한다. 그 후에 spanning tree를 업데이트 하게 된다. 그런 다음, bag of words를 계산하여 triangulating을 위한 data association에 도움을 주게 된다.

B. Recent Map Points Culling

map에서 유지된 Map point들은 생성 후 세 keyframe 동안 test를 거쳐야 한다, 이것은 trackable과 잘못된 triangulating을 막을 수 있다. 한 point는 아래와 같은 두 조건을 만족시켜야 한다.

1) tracking이 visible 하다고 예측한 frame들의 25% 이상에서 point를 찾아야 한다.

2) 만일에 map poimt 생성으로부터 한 keyframe 이상이 지났을 경우 적어도 세 keyframe에서 관찰돼야 한다.

C. New Map Point Creation

새로운 map point들은 covisibility graph에서 연결된 keyframe들로부터 ORB를 triangulating 함으로써 생성되게 된다. keyframe에서 매칭되지 않은 ORB는 다른 keyframe에서 매칭되지 않은 point에서 매칭을 탐색하게 된다.

D. Local Bundle Adjustment

local BA는 현재 진행되는 keyframe, covisibility graph에서 이것과 연결된 모든 keyframe들, 그 keyframe에 의해 발견된 모든 map point들을 최적화한다. outlier라고 표시된 관찰들은 middle과 최적화의 끝에서 버려지게 된다.

E. Local Keyframe Culling

local mapping은 중복 keyframe을 감지하고 그것들을 삭제하려고 한다. 이것은 복잡도를 줄일 수 있다. 여기서 모든 keyframe들이 적어도 3개 이상의 다른 keyframe의 map point들과 90% 이상의 covisibility 연결이 돼야한다.

​

* **Loop Closing**

A. Loop Candidates Detection

처음에, 새로운 keyframe (K\_i)의 bag of words와 covisibility graph에서 그것의 모든 이웃 사이의 similarity를 계산하고 가장 낮은 score s\_min을 남기게 된다. 그 후에 recognition 데이터베이스에 query 한 후에 s\_min보다 낮은 점수를 가진 keyframe들을 모두 제거한다. loop candidate를 승인하기 위해서는 연관이 있는 세 loop candidate들을 연속해서 감지해야 한다. K\_i에 비슷한 apperance를 가진 몇가지 place들이 있다면 몇가지 loop candidate들이 있을 것이다.

B. Compute the Similarity Transformation

monocular SLAM에서는 map이 drift 할 수 있는 7DoF가 있다. 3개의 translation, 3개의 rotation, scale factor이다. 그렇기에 현재 keyframe (K\_i)로부터 loop keyframe (K\_l) 까지 similarity transformation을 계산할 필요가 있다.

처음에 현재 keyframe에서 map point들과 연관된 ORB와 loop candidate keyframe들 사이에 대응점을 계산한다. 각 candidate에 RANSAC iteration을 수행하여 similarity transformation을 찾게 되고 충분한 inlier들을 가지게 되면 최적화하고 더 많은 대응하는 것의 search를 수행하게 된다.

C. Loop Fusion

loop correction에서 첫 번째 단계는 중복된 map point들을 fuse 하는 것이고 covisibility graph에 새로운 에지들을 삽입하는 것이다.

처음에 현재 keyframe pose (T\_iw)가 위에서 계산된 similarity transformation (S\_il)에 의해 보정되고 이 보정은 현재 keyframe의 이웃들에게도 전파되게 된다. loop keyframe과 그 이웃에서 보여진 map point들은 현재 keyframe에 project되며 그 이웃들과 매치들은 projection 주위에 좁은 구역에서 탐색된다.모든 매치된 map point들과 S\_il으 ㅣ계산에서 inlier에 있는 것들은 fuse된다.

D. Essential Graph Optimization

효율적으로 loop closing을 하기 위해서, pose graph 최적화를 수행한다. 최적화는 scale drift를 보정하기 위해 similarity transformation에 대해 수행되게 된다.

논문 본문: Mur-Artal, Raul, Jose Maria Martinez Montiel, and Juan D. Tardos. "ORB-SLAM: a versatile and accurate monocular SLAM system." IEEE transactions on robotics 31.5 (2015): 1147-1163.

1. **ORB-SLAM 실제 코드 돌린 결과**

나무, 실외, 식물이(가) 표시된 사진

자동 생성된 설명

point들을 기반으로 3D map을 작성할 수 있다. 빼곡하게 몰려있는 것을 obstacle로 인식할 수도 있고, 저 특징점들을 기반으로 내 위치가 어딨는지 알 수 있음을 볼 수 있다.

워낙 유명한 알고리즘이고 open code여서 github에서 쉽게 사용 가능하다.

링크: <https://github.com/raulmur/ORB_SLAM>

1. **Visual SLAM의 활용분야**

Visual SLAM을 통해 아래와 같이 실제 grid map 추상화도 가능하다.

텍스트이(가) 표시된 사진

자동 생성된 설명

이를 이용하면, path planning 알고리즘이 가능해져 실내 제어가 가능해질 수 있다.

장애물 회피도 가능해진다. 아래는 단일 전방 카메라만이 달려있는 드론으로 장애물 회피를 한 영상이다. (직접 구현)



Appendix 1. ORB-SLAM3 paper review

Link: <https://blog.naver.com/t1207a/222890100033>

Appendix 2. 3d visual slam map -> 2d grid map generation

Link: <https://blog.naver.com/t1207a/222871055705>