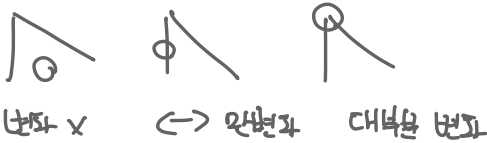


Visual SLAM 구조

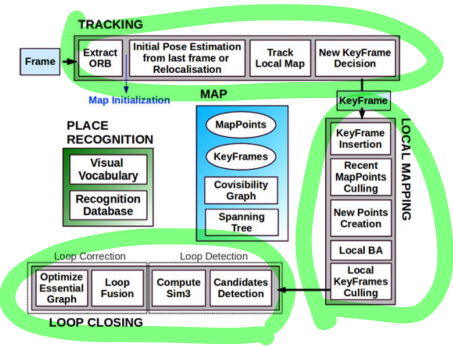
- Monocular : 1개 카메라
- Stereo : 2개 고정 카메라
- RGB-D : depth 정보 얻을 수 있는 Cam

Harris Corner (key point 찾는법)

⇒ image 의 한부분이동시키면 코너의 강도 모든 방향으로 이동하여도 변함
다 보자.



⇒ 3개의 thread 이용해서 Speed Good



- Tracking : 현재까지 특징 (기준점, ORB Feature)
- Local Mapping : key frame 정보를 mapping
- Loop Closing : 다시 현재 위치로 돌아오는 것

⇒ PnP 문제 풀기 카메라 내리 특징점

④ Bas of words (문서 간을 분류하는 데 사용)

(1) 영상 처리에서 feature 추출 후 Clustering 거쳐
feature를 대표할 수 있는 코드 및 코드 생성

=> 0.1 3개 thread에 다 활용 (keyframe 유사도, 등)

④ Bundle Adjustment

(Motion only BA => Tracking thread (카메라 포즈 최적화)
Local BA => Local map에 대한 카메라와 맵포인트 최적화
Full BA => Loop closing 이후 모든 카메라와 맵포인트 최적화

④ Covisibility Graph

=> key frame 간의 맵포인트 공유 관계를 나타낸 그래프

BA 를 필요 없을때 버린다 (계산 버리기 하기 위해)

Key frames (Next View selection) 으로 좋은 image 골라 느낌!

⇒ 같은 feature 로 tracking, map, relocal, loop closing 한다.
(한방에 한다)

- ↳ lsb → dense slam
- ↳ orb → feature slam (Feature 뽑아 작업)
 - Direct slam (Pixel 그대로 작업)

* heuristic
⇒ 노가다 해서 정해놓은
기준값 찾았다.

- (Covisibility Graph
- Essential Graph

ORB 가 Sift, Surf 보다 빠르다 사용한다.

tracking ⇒ feature tracking 의미 (대부분 바깥에서)

⇒ 카메라 vs 2D/C-1 ⇒ $\frac{\text{카메라}}{(\text{mms})}$ 단위는
Scale 을 모른다 (거리)

- ↳ 그래서 stereo 카메라 사서 다듬는다.
 - ↳ 단점 error 코- 엄청