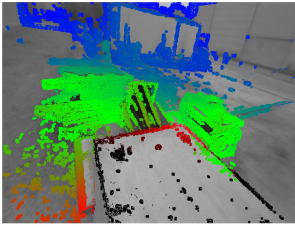


# LSD-SLAM

↳ feature 추출없이 바로 사본 자체에서 Semi-Dense 사용해 카메라에 강제된 물체 거리 가능해 SLAM에 적용.

+ (ORB는 feature로 과정 추가해야지만 그래도 좋음)

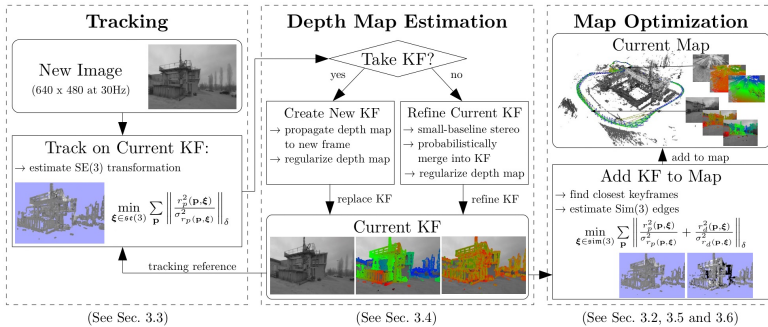
→ Semi dense의 의미? (물체는 GPU 필요했지만 CPU만으로 Semi하면 가능)



⇒ 원래는 모든부분 사본으로 표현되던데 거칠고 메모를 표현하지만, 이는 시간이 오래걸리기 때문에 X인 부분은 거칠기만으로 처리하고, 일부분만 표현 그래서 dense 한 semi가 붙은 것이다.

→ 이미지 밝기 강도에 따른 error 값 최소화

## LSD SLAM 전체 구조 (key frame 되면 depth 계산)



→ 이미지 밝기 tracking, depth map est 동일  
이 keyframe에서 추출된 Depth map은 Global map에 추가

Fig. 3: Overview over the complete LSD-SLAM algorithm.

→ 단일 trajectory에 error가 생기면 map 계산에도 error 전파  
→ scale 문제, loop closing 안됨

질문? ORB는 feature 지만 어떤 방법으로 바르래 특징점의 정확도 ↓ 되는것?

↳ 소도카는 관측을 정확도는 높임

↳ tracking 부분에서는 depth 뿐 intensity 만으로도 최적화

\* (depth, intensity) 외곽을 줄이는 코드로 최적화

\* Loop Closing  $\Rightarrow$  global 못찾고 local minimum에  
바라다 못가서 init 값 중 하나인데 loop closing  
은 init set 아래서 LSD에서는 거의 LC 실패



view point variant

(intensity, depth는 빛도  
같은 장소라도 앞 뒤 다르기 때문  
전부 intensity, depth 다 달라지고,  
빛도 달라서 거의 (정) 안됨)

\* Intensity, dep 쓰기 때문에 error 상이하여 들쭉날쭉  
해결을 위하여 이전 frame, 이전 depth 고려해 scale도 맞추고  
한다. (feature는 scale 최적화 안함, feature은 특징점만 extrac  
이기 때문)

\* 즉 depth, scale 몰라 힘들.

∴ CNN 학습 시켜서 depth 알아내는것 자동으로 해서  
depth 알려줘서 성능 더 올려줄 (CNN SLAM)

\* Camera 2개 안보고 1개쓰는 이유? 저렴, 1개면 반속 ↓ 이기때문