programmers

자율주행 데브코스

VSLAM 프로젝트 설명 및 가이드

Programmers Lecturer

장형기

hyunggi.chang95@gmail.com

programmers

Contents

programmers

1. Project Details

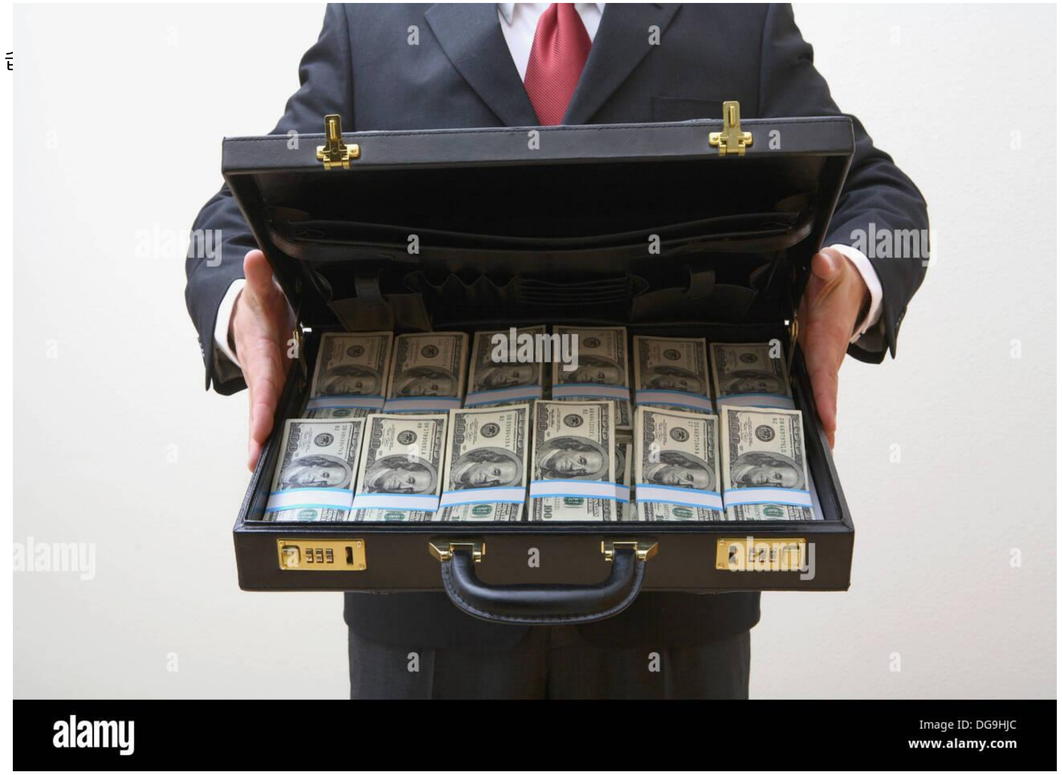
SLAM PoC Project

…as real as possible

- 당신은 자율주행 알고리즘 개발자/매니저입니다.

- 어느 날, 벤츠/폭스바겐/아우디에서 연락이 왔습니다.
  - “5년 후 양산을 목표로, **SLAM이 들어간 자율주행 시스템**을 만들고 싶습니다. 개발을 맡아주시겠습니까?”

- 계약금에 대해 들으신 대표님께서는 (당신의 의견은 묻지도 않고) 덜컥 개발 과제를 수락해버리셨습
  - “자네들, 조금 촉박하겠지만 2주만에 SLAM이 들어간 자율주행 시스템을 만들어줘야겠어.”
  - "오, 맙소사!”



SLAM PoC Project

…as real as possible

- 우리의 고객은 벤츠 (알잘딱깔센, 슬래임) / 폭스바겐 (스리슬램, Slam Dunk) / 아우디 (Visual설렘, S2LAM) 입니다.

- 당신은 고객사의 제품 개발 책임자 (장형기 강사)와 미팅을 가졌습니다.
  - “반갑습니다, 현재 우리는 **카메라를 이용한 오픈소스 SLAM**을 이용해서 여러 가지 제품 가능성을 알아보고 있습니다.”
  - “**우리가 개발하려는 목표**에 맞게, **좀 더 좋은 VSLAM 알고리즘**을 만들어주세요.”

SLAM PoC Project

…as real as possible

- 벤츠의 요구사항 - 알잘딱깔센, 슬래임
  - 우리는 주차를 할 때, 주차장의 모습을 유저에게 실시간으로 보여주고 싶습니다.
  - ProSLAM을 사용해서 얻은 지도를 보니, 이게 어디를 보여주는 건지도 모르겠더군요.
  - **1. 주변 환경을 예쁘게 보여줄 수 있는 VSLAM 알고리즘과, 2. 주변 환경을 잘 보여주는 시각화 툴을** 만들어주세요. (실시간 시각화가 아니어도 됩니다)

- 폭스바겐의 요구사항 - 스리슬램, Slam Dunk
  - 우리는 VSLAM 엔지니어가 없습니다. 앞으로 채용하려고 하는데, 이를 위해서 미래에 사용할 VSLAM 프레임워크가 필요합니다.
  - PTAM의 코드를 한번 본적이 있는데, 너무 예전 코드이고, 아키텍처도 이상하더군요.
  - **1. 다양한 알고리즘을 원하는 대로 사용해보고 실험을 해볼 수 있는, 모듈러 아키텍처를 가진 VSLAM 프레임워크**을 만들고, 2. **이 프레임워크를 다른 유명한 Open-Source 슬램과 비교해서 성능 비교**를 해주세요. (성능이 더 좋아야하는건 아닙니다)

- 아우디의 요구사항 - Visual설렘, S2LAM
  - 우리는 VSLAM으로 주행 중 자동차의 위치 정보를 뽑아서, 우리의 딥러닝 시스템의 Ground truth 정보로 사용하고 싶습니다.
  - ORB-SLAM을 사용하는데, CPU를 너무 많이 사용하기도 하고 속도도 너무 느리더군요.
  - **1. 적은 계산량으로 빠르게 돌면서 자동차의 pose 정보를 뽑는 알고리즘**을 만들어주세요.

SLAM PoC Project

…as real as possible

- 공통 요구사항
  - 프로젝트 종료 시기에 맞춰 정해진 **개발 목표를 달성**시키기 (**기술 개발**) (+150점)
  - 프로젝트 종료 시기에 맞춰 개발 내용 및 사용한 프레임워크에서 사용한 알고리즘들에 대해 **발표** (**이론 이해**) (+150점)
  - **중간 점검** 시기에 맞춰 개발 중인 내용 보고 (**프로젝트 매니징**) (+150점)

- 보너스 포인트
  - KITTI 데이터셋에서 돌 수 있게 프레임워크 개량 (+50점)
  - PC 웹캠 / 리얼센스 카메라로 실시간 데모가 가능하게 개량 (+100점)
  - 자이카에서 돌 수 있게 프레임워크 개량 (+100점)
  - 정확도 개선 (+50점)
  - 속도 개선 (+50점)
  - 오프라인 시각화 가능 (+50점)
  - 실시간 시각화 가능 (+100점)
  - 아키텍처 / 알고리즘 재사용성 개선 (+100점)
  - 안정성 확보 - CI/CD 및 유닛테스트 (+100점)
  - 다른 팀에게도 도움이 될 수 있는 자료 정리 및 공유 (+50점)
  - 오픈소스를 참고해 직접 VSLAM 파이프라인을 설계 및 구현 – 최소 2 모듈 이상 변경 (+150점)
  - 고객의 갑작스러운 요구사항 1 달성 (+100점)
  - 고객의 갑작스러운 요구사항 2 달성 (+200점)

SLAM PoC Project

…as real as possible

- 제약 조건
  - 중간 점검 시기는 **6월 17일 (금) -** 고객과 논의 후 변경 가능.
  - 최종 데드라인은 **6월 24일 (금) -** 변경 불가능
  - 공통 요구사항 3개 중 2개 이상 실패 시 보너스 포인트는 전부 무효
  - 모든 고객과의 논의 사항은 ‘매니저’ 또는 ‘매니저 + 엔지니어 1명’만 가능 (출장)
    - 최대 하루에 한번
  - OS 종속성 - Ubuntu 버전은 자유. ROS 없이도 돌아갈 수 있어야함.

programmers

2. Roles

매니저

프로젝트 매니저 (또는 팀의 대표자)

- 책임
  - 고객의 요구사항을 정확하게 이해하고, 이를 엔지니어링 요구사항으로 변환
  - 개발 일정 수립 및 모니터링
  - 팀원들에게 업무 분배
  - 고객과의 소통 및 목표 조정
  - ‘고객이 원하는 것은 무엇이고, 우리 팀이 그걸 어떻게 준비해야할까?’ 라는 문제를 해결하는 사람

- 어울리는 사람의 특징
  - 계획성있고, 임기응변이 좋은 사람
  - 사교성이 좋고 팀원들을 잘 이끌 수 있는 사람

- 어울리지 않는 사람의 특징
  - ‘난 개발할거야!!’

아키텍트

시스템 설계자 및 품질 보증인

- 책임
  - VSLAM 파이프라인 및 프레임워크 아키텍처 설계
  - 코드의 품질 확인 – 버그, 디자인, 재사용성, 속도, 정확도
  - ‘우리 코드가 엔지니어링 요구사항을 다 지켜내는가?’ 라는 문제를 해결하는 사람

- 어울리는 사람의 특징
  - 영상처리/Frontend/Backend에 대해 깊지는 않지만 넓은 이해도를 가진 사람
  - 꼼꼼한 성격
- 어울리지 않는 사람의 특징
  - ‘에이 뭐 이정도 쯤이면 버그 없겠지’
  - ‘코드 리뷰? 시간도 없는데 그걸 왜 하는데?’

영상처리 알고리즘 엔지니어

센서부터 이미지 처리까지

- 책임
  - VSLAM의 인풋인 영상을 다루는 엔지니어
  - 영상에서 좋은 feature 매치가 나오는 알고리즘을 설계 및 구현
  - 빠르고, 정확한 알고리즘 구현

- 어울리는 사람의 특징
  - OpenCV를 기가 막히게 잘 다룸
  - 속도, 정확도 측정에 도가 틈

- 어울리지 않는 사람의 특징
  - ‘어… 돌아는 가는데요… 가끔 터져요…’

프론트엔드 알고리즘 엔지니어

정확하게 초기 포즈를 추정

- 책임
  - 영상처리의 결과물을 기반으로 Motion estimation 알고리즘의 설계 및 구현을 담당
  - 다양한 알고리즘 파라미터 튜닝
  - 빠르고 정확한 알고리즘 구현

- 어울리는 사람의 특징
  - OpenCV / Eigen을 기가 막히게 잘 다룸
  - 실험을 즐김

- 어울리지 않는 사람의 특징
  - ‘슬램 너무 어렵다 ㅎ…’

백엔드 알고리즘 엔지니어

포즈와 랜드마크 위치 보정

- 책임
  - 그래프 기반 포즈 / 랜드마크 위치 보정 알고리즘 설계 및 튜닝

- 어울리는 사람의 특징
  - 새로운 라이브러리 사용을 두려워하지 않음
  - 슬램 마스터의 자질이 보임

- 어울리지 않는 사람의 특징
  - ‘그래프 슬램? 처음 들어보는데?’

programmers

3. Tips / Q&A

Tips

… 2 weeks is not enough

- 2주는 굉장히 짧은 시간입니다.
  - 어떻게 하면 개발 효율성을 높일 수 있을까?
  - 어떻게 하면 기존 목표를 전부 빠르게 달성할 수 있을까?
  - 어떻게 하면 리스크를 줄일 수 있을까?

Tips

개발 효율성 높이기

- 올바른 툴 사용하기
- 도움을 구하기
- 임팩트 있는 작업을 주로 하기

- ~~10배 더 타자를 빠르게 치기~~
- ~~24시간 코드 치기~~

Tips

목표 달성

- 현실적인 플래닝
- 팀원들간의 효율적인 소통
- 전략적으로 일하기 + 전략적으로 쉬기

Tips

리스크 줄이기

- 투명한 소통
- 플랜 B 만들어두기

programmers

끝