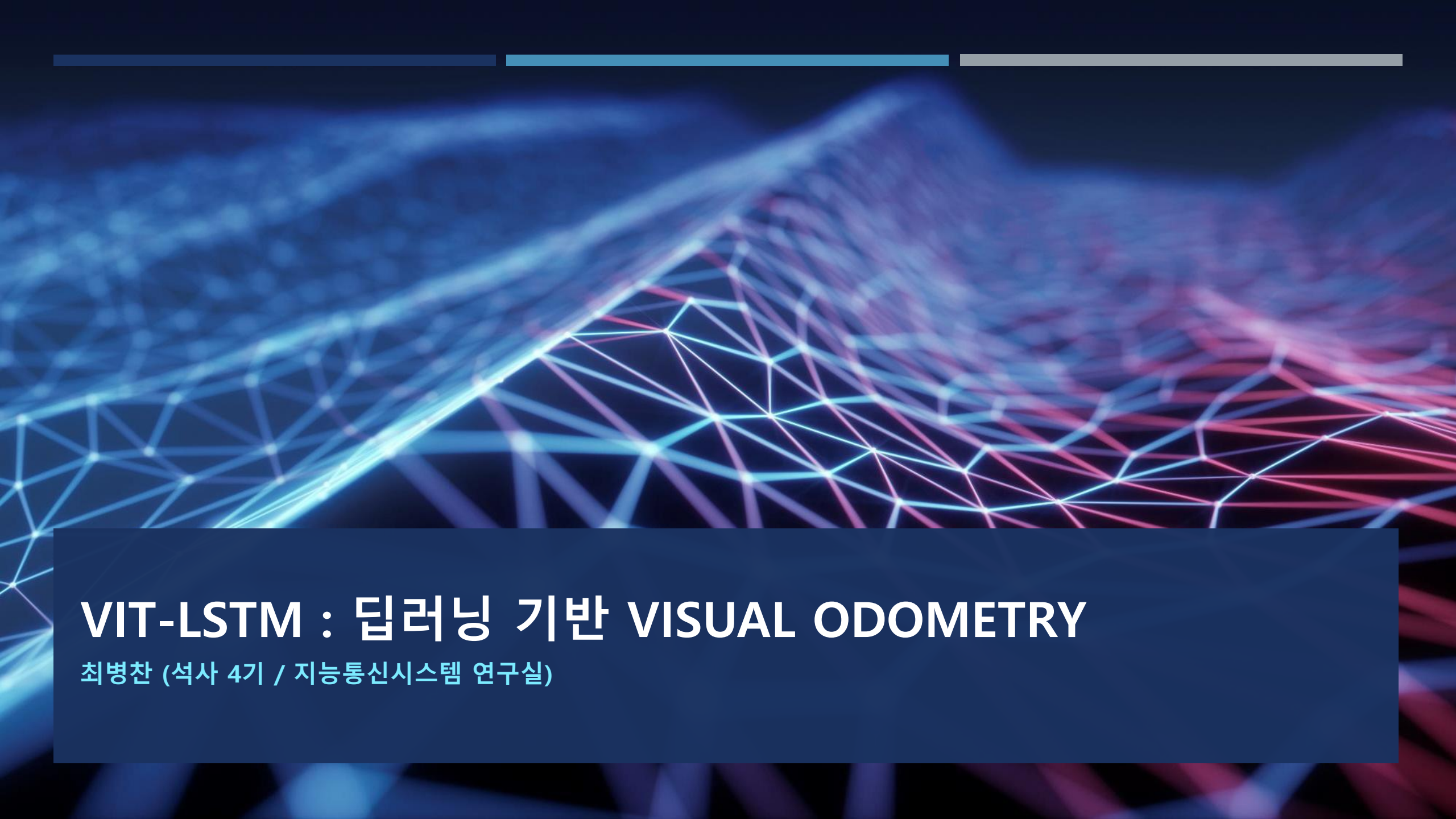


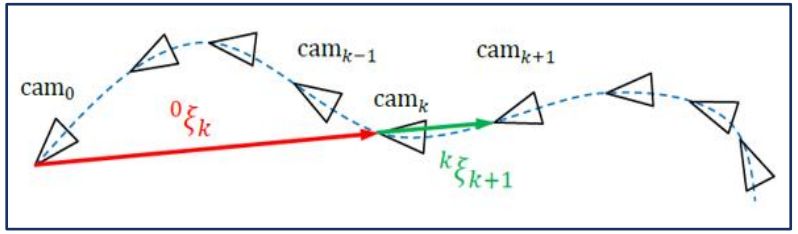
VIT-LSTM : 딥러닝 기반 VISUAL ODOMETRY

최병찬 (석사 4기 / 지능통신시스템 연구실)

목 차

VISUAL ODOMETRY 소개

VISUAL ODOMETRY 소개



[Visual Odometry (VO)란?]

- Camera를 이용하여 차량 또는 로봇의 이동경로 및 자세를 추정하는 방식
  - **Monocular VO : Camera 1개만 사용한 VO**
  - **Stereo VO : Stereo Vision을 이용한 VO**
  - **Multi-Camera VO : 다중 Camera의 Panorama 이미지를 이용한 VO**
- Camera만을 사용하거나 Camera + 외부센서 (IMU, Wheel Encoder)를 Sensor Fusion하여 이동경로 및 자세를 추정할 수 있음

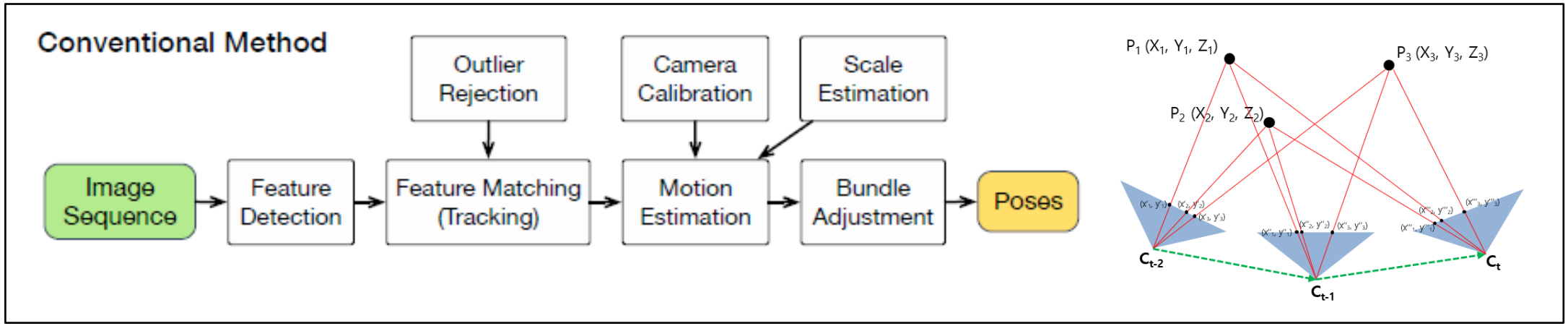
[VO 등장배경]

- 화상 탐사 로봇의 위치 및 자세 추정을 위해 카메라만을 사용한 Odometry 기법 등장
- 지구와 다른 **우주 공간에서 중력에 의존적인 IMU 및 Wheel Encoder를 사용할 수 없기에** 로봇의 자세 추정을 위해 Computer Vision을 사용함

VISUAL ODOMETRY 소개

[Classical Visual Odometry]

- 연속된 이미지에서 Feature를 추출하고, 동일 Feature Keypoint에 대한 Epipolar Geometry (Multi-view Geometry) 기법을 사용하여 카메라의 Rotation과 Translation을 산출함.
- 연속된 카메라 이미지에서 동일 Feature Keypoint에 대한 Rotation과 Translation에 대한 Error를 최소화하기 위해 Reprojection Error을 최소화하게 만드는 Rotation과 Translation을 Bundle Adjustment와 같은 비선형 최적화 기법을 사용함.

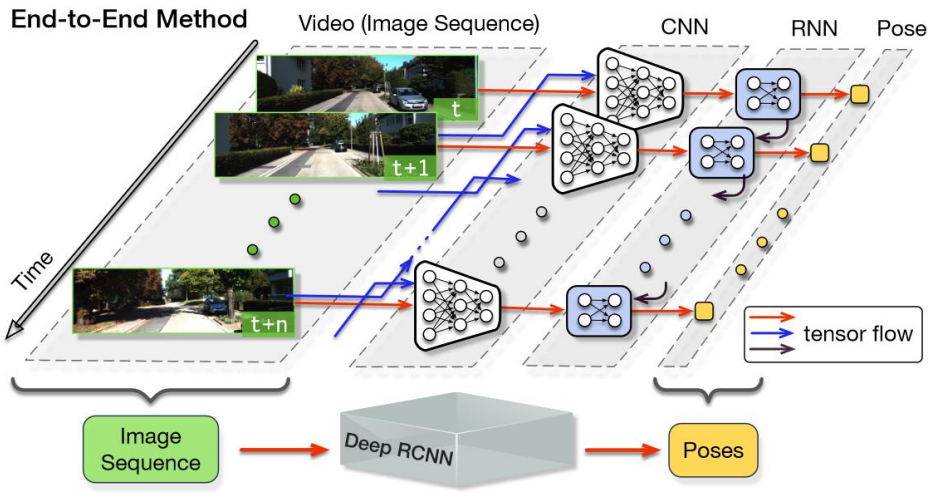


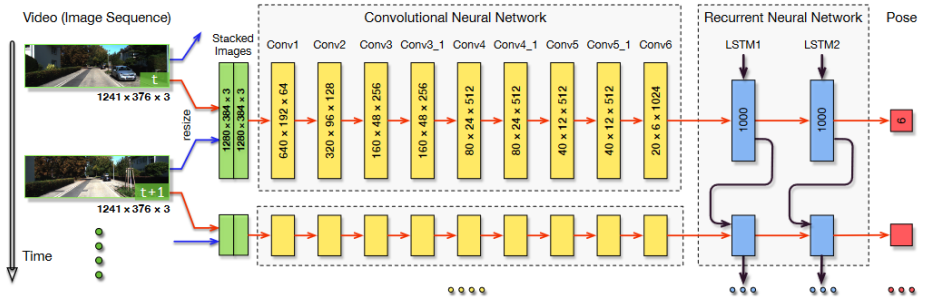
VISUAL ODOMETRY 소개

[Deep Learning 기반 Visual Odometry]

- 현재 널리 알려져 있는 CNN (Convolutional Neural Network)의 Feature Output은 카메라 Pose 또는 이미지 이동에 관한 정보가 아닌 Recognition • Detection에 관한 정보를 내포하기에 Classification에 특화되어 사용됨.
- Deep Neural Network를 사용한 VO를 구현하기 위해서는 CNN이 출력하는 Feature 데이터를 추적하여 시간에 따른 변화 추세 (Rotation, Translation)를 산출할 수 있는 기법이 요구됨.
- RNN (Recurrent Neural Network)와 같이 시간에 따른 데이터의 변화를 학습하여 Prediction하는 기법을 VO에 사용함.

Conventional VO	→	Deep Learning-based VO
Feature Extraction	→	**CNN (Convolutional Neural Network)**
Feature Tracking		
Pose Estimation	→	**RNN (Recurrent Neural Network)**
Pose Optimization		





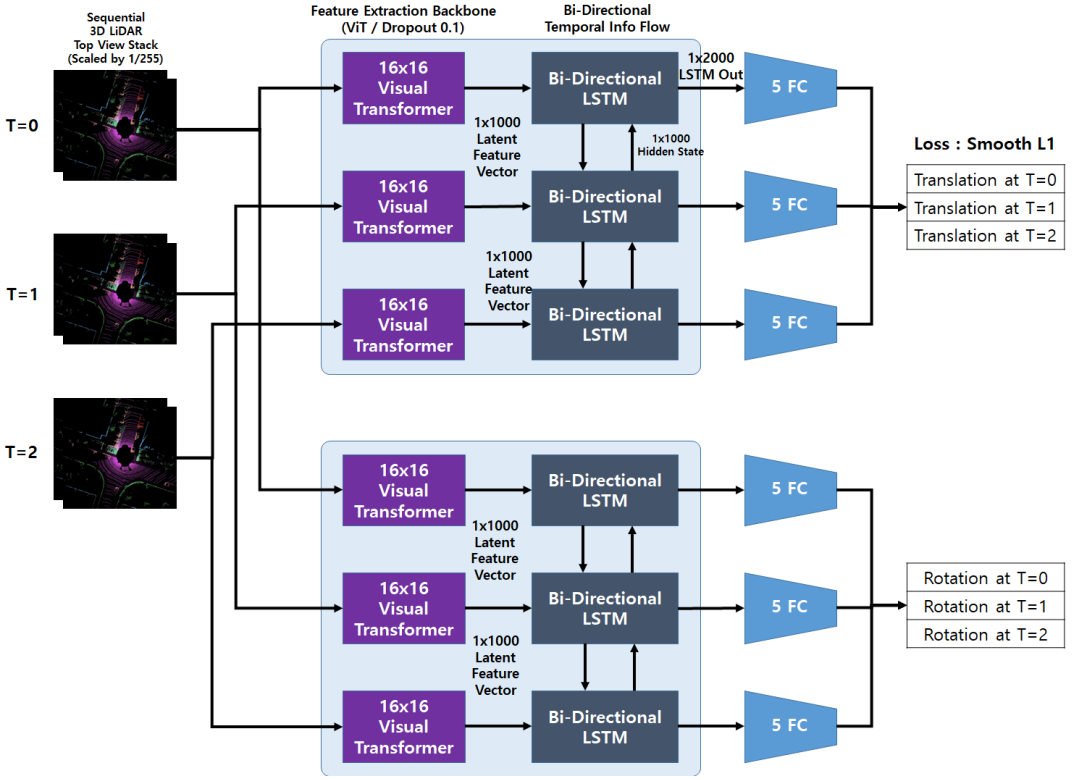
DeepVO : Towards End-to-End Visual Odometry with Deep Recurrent Convolutional Neural Networks

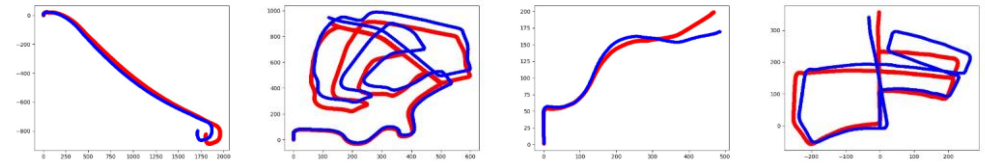
VIT-LSTM 소개

VIT-LSTM 소개

[ViT-LSTM : Visual Transformer + LSTM]

- **입력 데이터 : 3D LiDAR Top View 이미지 Sequence**
  - 전방 카메라 이미지 보다 로봇의 이동과 회전을 일관성 있게 표현하기 위해 사용
- **출력 데이터 : Translation & Rotation Sequence**
  - N Timestep 동안의 매 순간마다 로봇의 이동과 회전 정보
- **Feature Extraction : ViT**
  - CNN 보다 큰 Kernel을 가지고 학습할 수 있는 네트워크를 선정하여 고속 이동의 이미지 변화를 감지함
- **Sequential Regression : Bi-Directional LSTM**
  - Timestep간 정보를 교류하기 위해서 사용함
- **Final Regression : Fully Connected Layer**
  - 최종 Output의 목표 형태로 구성하기 위해 사용함
- **결과 : 200m 주행에 대해 수 cm 단위 위치 추정 정확도**
  : 해당 성능을 가진 네트워크 저장
  : Reproducibility 검증 진행





필요 사항

필요 사항

[논문 작성]

- Classical 기법 + Neural Network 기법과의 비교
  - 평가 지표 : ATE (Absolute Trajectory Error)
  - Classical 기법 (SuMA++, ViSO, RTAB, ORB-SLAM, etc)와 Trajectory 비교
  - 다른 Neural Network 기법은 재구현이 매우 힘들기 때문에 다른 논문에서 제시한 평가 지표로만 비교해야함

- Visual Transformer를 포함한 관련 기술 및 Related Work 내용 작성

- 6DOF Groundtruth에 대한 문제점 지적하는 내용 작성 / Relative 6DOF 학습의 한계 관련 내용 작성

- 플랫폼별 좌표계 차이를 고려한 좌표계 통일 방법 관련 내용 작성

필요 사항

[SW 구현]

- AI 학습 구현 : PyTorch, Tensorboard, timm (Network Design, Tensor Control, Dataloader, Data Augmentation)
- 영상 처리 : OpenCV-Python (Image I/O, Optical Flow, Multi-View Geometry)
- Dataset 관리 : HDF5 Python
- 수학적 연산 : Numpy, SciPy

[관련 이론 및 기법]

- AI : Backpropagation, Normalization, Optimizer (Momentum), Attention
  - : Issues (Overfitting, Gradient Vanishing, Gradient Explosion)
  - : Issue Handling (Dropout, Data Augmentation, Normalization, Gradient Clipping)
  - : Data Distribution Analysis (Correlation Matrix, Box Plot)
- Robotics : Odometry, Pose Transformation (Translation & Rotation), Euler Angle-Quaternion

감사합니다