2025 하계 세미나

3D Foundation Models and Their Applications

Sogang University
Vision & Display Systems Lab, Dept. of Electronic Engineering

Presented By
120240320 석사과정 신은호

Contents

- 논문 선정 이유

- Vggt: Visual geometry grounded transformer [CVPR 2025 Best Paper Award]

- MASt3R-SLAM: Real-time dense SLAM with 3D reconstruction priors [CVPR 2025]

논문 선정 이유

- 3D Foundation Models
  - 3D Scene understanding
    - 촬영을 통해 취득된 2D 이미지로부터 공간 이해에 대한 지속적 수요
      - Image matching, Depth estimation, SLAM…
    - 다양한 rule-based algorithm 통한 동작 수행 방식 존재
      - SIFT, SURF, FAST, Epipolar Geometry…
    - 2D image 기반 3D 공간 예측 방식 (2D matching → 3D point cloud)
      - Structure from Motion 방식 알고리즘 이용 (e.g. COLMAP, Reality Capture)
      - Feature Description → Epipolar Geometry → Triangulation → Bundle Adjustment
  - DUSt3R, MASt3R, VGGT… 3D Foundation model 방식 등장
    - MASt3R : 3D-2D matching algorithm
      - Image Matching을 3D point cloud → 2D matching의 downstream으로 해결
    - VGGT : Visual Geometry Grounded Transformer
      - Camera pose estimation, Depth Estimation, Tracking을 feed forward 방식으로 수행

- Vggt: Visual geometry grounded transformer [CVPR 2025 Best Paper Award]
  - 1. Visual Geometry Group, University of Oxford
  - 2. Meta AI

VGGT: Visual geometry grounded transformer

- Introduction
  - Scene의 key 3D attributes를 직접 추론하는 feed forward neural network
    - Camera parameters, point maps, depth maps, 3D point tracks
  - 기존 3D computer vision 연구가 단일 작업에 국한 혹은 특정 작업에 특화
    - 3D 재구성 기법의 경우 기본적으로 Bundle Adjustment와 같은 반복 최적화 기법 사용
    - 단안 깊이 추정 등 geometric 정보로 해결 불가한 작업 수행 ML(e.g. Depth Anything)
    - 순수 신경망을 통해 3D 작업을 직접 해결 할 수 있는가
    - VGGT의 경우 simple and efficient image reconstruction

VGGT: Visual geometry grounded transformer

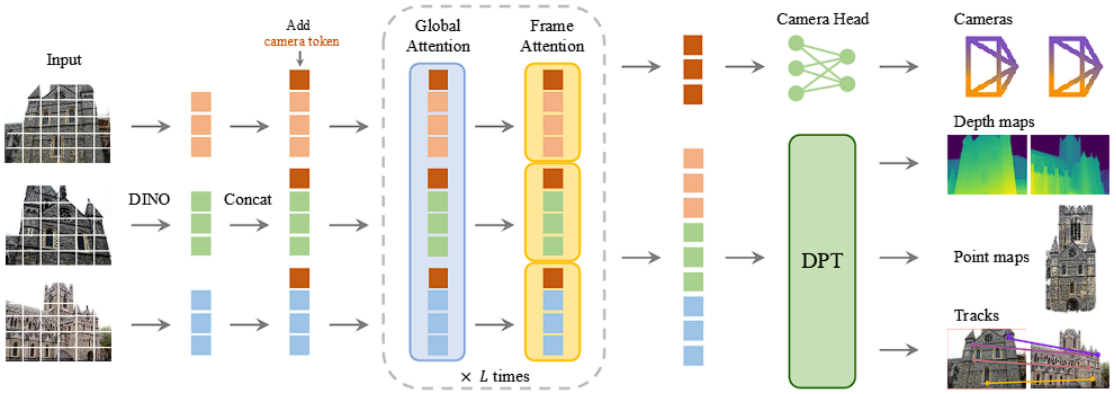
- Introduction
  - 최근 연구 DUSt3R, MASt3R의 경우 FFN 기반 3D Reconstruction 수행
    - 2장의 pair 이미지에 대해 처리 가능
  - VGGT의 경우 one, a few, hundreds of input view에 대해 처리 가능
    - 추가 postprocessing 없이도 기존 방법 대비 뛰어난 성능
    - 후처리가 필수인 DUSt3R, MASt3R, VGGSfM과 본질적으로 다른 접근
    - 3D 재구성을 위해 특수한 구조의 네트워크 설계 아닌 fairly standard large transformer
      - 특수한 3D inductive bias 없이 구성
    - GPT, CLIP, DINO, Stable Diffusion과 유사하게 다양한 작업에 파인튜닝이 가능
      - VGGT 추출 특징은 point tracking in dynamic videos, novel view synthesis에 높은 성능

VGGT: Visual geometry grounded transformer

- Method
  - Problem definition and notation
    - $f\left((I_i)_{i=1}^N\right) = (\mathbf{g}_i, D_i, P_i, T_i)_{i=1}^N$
      - 동일 장면을 관찰하고 있는 N장의 이미지 I
      - Camera parameter g (intrinsic and extrinsic $\mathbb{R}^9$)
      - Depth map D ∈ $\mathbb{R}^{H\times W}$, point map P ∈ $\mathbb{R}^{3\times H\times W}$, grid T ∈ $\mathbb{R}^{C\times H\times W}$ for point tracking
        - ✓g=[q,t,f], rotation quaternion q, translation vector t, FoV f
        - ✓Depth D ∈ $\mathbb{R}^+$, point map P는 viewpoint invariant하므로 g_1에서 정의
        - ✓Keypoint tracking의 경우 Track-Anypoint 방식
          - the network output track $\mathcal{T}^*(y_q) = (y_i)_{i=1}^N$
          - $\mathcal{T}\left((y_j)_{j=1}^M, (T_i)_{i=1}^M\right) = \left((y_{i,j})_{i=1}^N\right)_{j=1}^M$, feature map T
      - Over-complete Predictions
        - ✓VGGT가 예측하는 각 값들이 서로 독립적이지 않음
        - ✓서로 간 closed-form 관계로 연결되나, 이를 명시적으로 모두 예측
        - ✓결과적으로 성능이 크게 향상됨을 확인

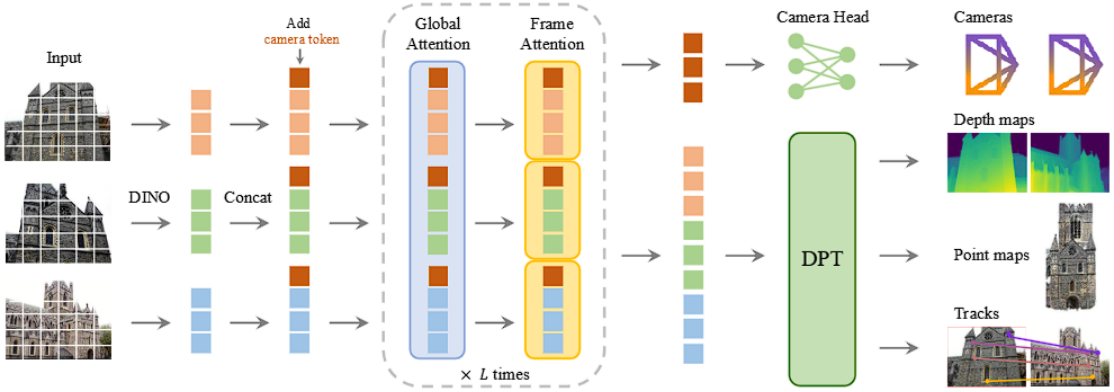
VGGT: Visual geometry grounded transformer

- Method
  - Feature Backbone
    - Following recent works in 3D deep learning, 최소한의 3D inductive bias 유지 설계
      - ∴ 모델을 대형 transformer로 구현
    - 각 입력 이미지 I는 DINO를 통해 patchify 후 K개의 토큰 $t \in \mathbb{R}^{K \times C}$
    - 모든 프레임의 이미지 토큰을 결합한 후 frame-wise & global self-attention 교대 적용
      - ∴ Alternating-Attention(AA) 기법 도입
        - ✓Frame 단위 self-attention : 프레임 내 토큰들 t_k^I에 attend
        - ✓Global self-attention : 모든 프레임의 토큰들 고려 t^I, 전역 정보 결합
          - Cross-attention 층 없이 self-attention만으로 구성



VGGT: Visual geometry grounded transformer

- Method
  - Prediction heads
    - 각 I_i에 대해 image token t_i^I camera token $t_i^g \in \mathbb{R}^{1\times C}$, register token $t_i^R \in \mathbb{R}^{4\times C}$ 결합
      - Register token reference 'Vision transformers need registers'[ICLR 2024]
    - $\left(t_i^I, t_i^g, t_i^R\right)_{i=1}^N$은 AA transformer 구조에 입력되어 $\left(\hat{t}_i^I, \hat{t}_i^g, \hat{t}_i^R\right)_{i=1}^N$ 출력
      - First frame에 대해 camera token과 register token은 different set of learnable token
        - ✓모든 3D 예측을 첫번째 카메라 좌표계에서 정의할 수 있도록 수행
      - Transformer 는 frame별 self-attention을 포함하기 때문에 frame 특화 출력 token
      - 최종 출력 시 register token은 버려지며 image & camera token이 예측에 사용



VGGT: Visual geometry grounded transformer

- Method
  - Prediction head
    - Camera Prediction
      - 4개의 self-attention 층과 linear layer 를 통과시켜 예측
    - Dense Prediction
      - Image token은 DPT 계층을 거쳐 고해상도 feature map $F_i \in \mathbb{R}^{C \times H \times W}$로 변환
      - F는 3x3 convolution layer를 통해 Depth map & Point map으로 mapping
        - ✓Aleatoric Uncertainty를 예측하며 모델의 confidence를 나타내고 손실함수에 사용
    - Tracking
      - CoTracker2[ECCV 2024] architecture 사용
        - ✓입력 프레임 간의 시간적 순서 가정하지 않으므로, 단순 이미지 집합 사용 가능

VGGT: Visual geometry grounded transformer

- Training
  - Training Loss
    - $L = L_{camera} + L_{depth} + L_{pmap} + \lambda L_{track}$
    - Camera loss
      - Huber loss $L_{camera} = \sum_{i=1}^{N} \|\hat{g}_i - g_i\|_{\epsilon}$

$$\text{Huber Loss} = \sum_{n=1}^{N} l_n,$$

$$\text{where } l_n = \begin{cases} \frac{1}{2}(x_n - y_n)^2, & \text{if } |x_n - y_n| < \delta \\ \delta\left(|x_n - y_n| - \frac{1}{2}\delta\right), & \text{otherwise} \end{cases}$$

      - 미분 가능하며 outlier 영향을 덜 받는 일차식 가까운 형태
    - Depth loss
      - DUSt3R을 따르며, aleatoric-uncertainty loss와 mono depth에서 사용되는 gradient 항
      - $L_{\text{depth}} = \sum_{i=1}^{N} \|\Sigma_i^D \odot (\hat{D}_i - D_i)\| + \|\Sigma_i^D \odot (\nabla\hat{D}_i - \nabla D_i)\| - \alpha \log \Sigma_i^D$
    - Point map loss
      - Depth loss와 동일하게 정의
    - Tracking loss
      - $L_{\text{track}} = \sum_{j=1}^{M} \sum_{i=1}^{N} \|y_{j,i} - \hat{y}_{j,i}\|$
      - CoTracker2와 동일하게 point의 해당 프레임 관측 여부 위한 visibility loss 동시 사용

VGGT: Visual geometry grounded transformer

- Training
  - Ground Truth Coordinate Normalization
    - Scene의 scale이나 global reference frame이 바뀌어도 이미지 자체는 영향을 받지 않음
      - 다양한 변형은 모두 타당한 3D 재구성 결과
        - ✓불확정성(ambiguous reconstruction) 제거 필요
        - ✓Canonical frame 정의 후, 해당 정규화를 학습하도록 유도
    - DUSt3R 을 따라
      - 모든 데이터를 첫번째 카메라 좌표계로 변환
      - Point map상의 모든 3D 점들의 원점으로부터의 평균 유클리드 거리 계산
      - Camera translation, point map, depth map 정규화
      - 단, 모델의 출력물은 별도로 normalization 하지 않음

VGGT: Visual geometry grounded transformer

- Training
  - Implementation Details
    - 1.2 B parameter (1.2B x 4 byte = 약 4.8GB)
    - AdamW optimizer 사용, 160,000 iteration 수행, cosine learning rate scheduler 사용
    - Color jittering, Gaussian blur, grayscale augmentation 수행
    - 64개 A100 GPU 9일간 수행
    - (그 외 생략)
  - Training Data
    - Co3Dv2,BlendMVS, DL3DV, MegaDepth,Kubric, WildRGB, ScanNet, HyperSim, Mapillary, Habitat, Replica, MVS-Synth, PointOdyssey, Virtual KITTI,Aria Synthetic Environments, Aria Digital Twin, Objaverse 등 artist-created asset
    - Indoor and outdoor 환경과 synthetic and real-world scenario로 다양하게 구성
    - 3D 주석 정보는 센서기반 측정, 합성 엔진, SfM 기법을 통해 획득
    - 전체 구성의 경우 MASt3R의 학습 데이터 규모 및 다양성과 유사

VGGT: Visual geometry grounded transformer

- Experiments
  - Camera Pose Estimation
    - CO3Dv2, RealEstate 10K dataset 수행
      - Fast3R과 유사한 속도를 가지며 뚜렷한 성능 향상
      - Re10K의 경우 학습되지 않은 dataset으로 generalization 입증
      - BA를 결합할 경우 추가적인 성능 향상 가능

Methods	Re10K *(unseen)* AUC@30 ↑	CO3Dv2 AUC@30 ↑	Time
Colmap+SPSG [92]	45.2	25.3	∼ 15s
PixSfM [66]	49.4	30.1	> 20s
PoseDiff [124]	48.0	66.5	∼ 7s
DUSt3R [129]	67.7	76.7	∼ 7s
MASt3R [62]	76.4	81.8	∼ 9s
VGGSfM v2 [125]	78.9	83.4	∼ 10s
MV-DUSt3R [111] ‡	71.3	69.5	∼ 0.6s
CUT3R [127] ‡	75.3	82.8	∼ 0.6s
FLARE [156] ‡	78.8	83.3	∼ 0.5s
Fast3R [141] ‡	72.7	82.5	∼ **0.2s**
Ours (Feed-Forward)	85.3	88.2	∼ **0.2s**
Ours (with BA)	**93.5**	**91.8**	∼ 1.8s

VGGT: Visual geometry grounded transformer

- Experiments
  - Multi-view Depth Estimation
    - DTU dataset에서 Accuracy(Euclidean), Completeness, Overall(Chamfer distance)
    - 카메라 GT 정보를 사용하지 않고 작동하는 DUSt3R, VGGT
  - Point Map Estimation
    - DUSt3R와 MASt3R는 각각 장면당 약 10초가 소요되는 고비용 최적화 수행
    - 직접 예측된 포인트 맵보다, 예측된 depth map과 camera parameter로부터 3D unprojection을 통해 얻은 포인트들이 더 정확하다는 것도 관찰
    - 유화 그림(oil painting), 겹치지 않는 프레임(non-overlapping), 반복적이거나 균질한 질감(예: 사막) 등 어려운 out-of-domain 예제에서도 잘 작동

Known GT camera	Method	Acc.↓	Comp.↓	Overall↓
✓	Gipuma [40]	**0.283**	0.873	0.578
✓	MVSNet [144]	0.396	0.527	0.462
✓	CIDER [139]	0.417	0.437	0.427
✓	PatchmatchNet [121]	0.427	0.377	0.417
✓	MASt3R [62]	0.403	0.344	0.374
✓	GeoMVSNet [157]	0.331	**0.259**	**0.295**
✗	DUSt3R [129]	2.677	0.805	1.741
✗	Ours	**0.389**	**0.374**	**0.382**

Table 2. **Dense MVS Estimation on the DTU [51] Dataset.**

Methods	Acc.↓	Comp.↓	Overall↓	Time
DUSt3R	1.167	0.842	1.005	~ 7s
MASt3R	0.968	0.684	0.826	~ 9s
Ours (Point)	0.901	0.518	0.709	~ 0.2s
Ours (Depth + Cam)	**0.873**	**0.482**	**0.677**	~ 0.2s

Table 3. **Point Map Estimation on ETH3D**

VGGT: Visual geometry grounded transformer

- Experiments
  - Ablation studies
    - Alternating-Attention 설계 유효성 검증
      - Point map 추정 정확도는 장면의 기하 구조와 카메라 파라미터에 대한 모델의 통합적인 이해를 반영하므로, 구성 요소 분석의 평가 지표로 선택

ETH3D Dataset	Acc.↓	Comp.↓	Overall↓
Cross-Attention	1.287	0.835	1.061
Global Self-Attention Only	1.032	0.621	0.827
Alternating-Attention	**0.901**	**0.518**	**0.709**

    - Multi-task Learning
      - 하나의 네트워크가 여러 3D 요소들을 동시에 학습하는 것이 유리한지를 검증
        - ✓depth map과 camera parameter로 point map을 계산 가능

w. $\mathcal{L}_{camera}$	w. $\mathcal{L}_{depth}$	w. $\mathcal{L}_{track}$	Acc.↓	Comp.↓	Overall↓
✗	✓	✓	1.042	0.627	0.834
✓	✗	✓	0.920	0.534	0.727
✓	✓	✗	0.976	0.603	0.790
✓	✓	✓	**0.901**	**0.518**	**0.709**

VGGT: Visual geometry grounded transformer

- Experiments
  - Finetuning for Downstream Tasks
    - Feed-forward Novel View Synthesis
      - 기존 방법은 카메라 파라미터가 주어진 입력 이미지를 받아, 새로운 카메라 위치에 해당하는 이미지를 예측
      - LVSM을 따라 VGGT를 직접 RGB 이미지를 출력하도록 수정
      - 4개의 입력 뷰를 사용하고, Plücker ray를 이용해 타겟 시점 정보를 표현

Method	Known Input Cam	Size	PSNR ↑	SSIM ↑	LPIPS ↓
LGM [110]	✓	256	21.44	0.832	0.122
GS-LRM [154]	✓	256	29.59	0.944	0.051
LVSM [53]	✓	256	31.71	0.957	0.027
Ours-NVS*	✗	224	30.41	0.949	0.033

VGGT: Visual geometry grounded transformer

- Experiments
  - Finetuning for Downstream Tasks
    - Dynamic Point Tracking
      - Metric
        - ✓Occlusion Accuracy (OA) : occlusion 예측의 binary accuracy
        - ✓Mean proportion of visible points accuracy
        - ✓Average Jaccard (AJ) : 추적 및 가려짐 예측 모두 고려 평균 자카드 지수
      - 수정된 CoTracker2를 Kubric 데이터셋에서 전체 파인튜닝
      - 해당 태스크에 특화되어 있지 않음에도 우수한 성능

Method	Kinetics			RGB-S			DAVIS		
	AJ	δ^{vis}_{avg}	OA	AJ	δ^{vis}_{avg}	OA	AJ	δ^{vis}_{avg}	OA
TAPTR [63]	49.0	64.4	85.2	60.8	76.2	87.0	63.0	76.1	91.1
LocoTrack [13]	52.9	66.8	85.3	69.7	83.2	89.5	62.9	75.3	87.2
BootsTAPIR [26]	54.6	68.4	86.5	70.8	83.0	89.9	61.4	73.6	88.7
CoTracker [56]	49.6	64.3	83.3	67.4	78.9	85.2	61.8	76.1	88.3
CoTracker + Ours	**57.2**	**69.0**	**88.9**	**72.1**	**84.0**	**91.6**	**64.7**	**77.5**	**91.4**

VGGT: Visual geometry grounded transformer

- Discussion
  - Limitations
    - Fisheye or panorama image 처리 지원하지 않음
    - 입력 영상의 회전이 극단적인 경우 재구성 성능 저하
    - 소규모 non-rigid 움직임 장면 처리 가능하나, 크게 변형되는 경우 실패
    - But, 최소한의 구조 수정과 함께 대상 데이터셋에 대해 fine-tuning 통해 극복 가능
  - Runtime and Memory
    - NVIDIA H100 GPU Flash Attention v3 사용 수행

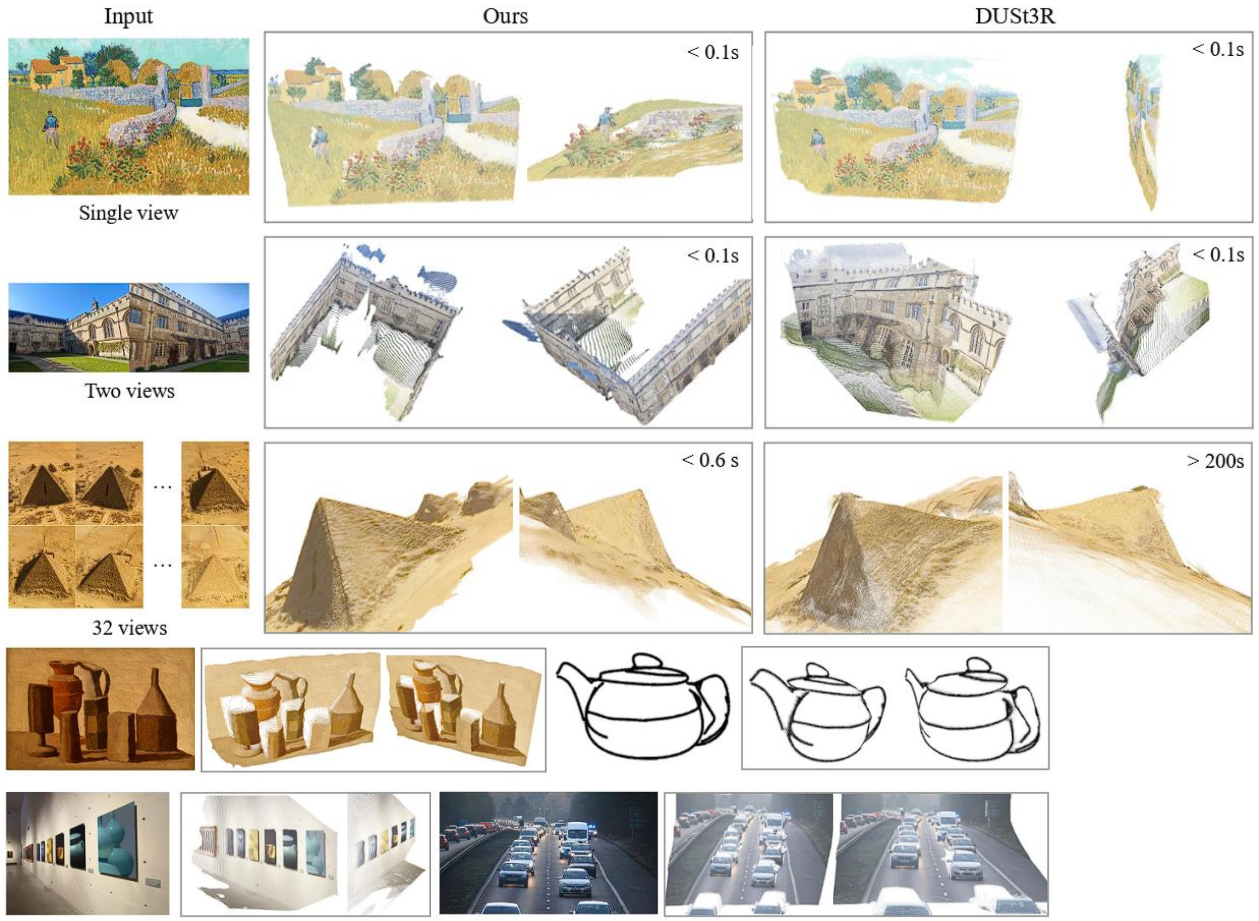
Resolution : 336 x 518

Input Frames	1	2	4	8	10	20	50	100	200
Time (s)	0.04	0.05	0.07	0.11	0.14	0.31	1.04	3.12	8.75
Mem. (GB)	1.88	2.07	2.45	3.23	3.63	5.58	11.41	21.15	40.63

Table 9. **Runtime and peak GPU memory usage across different numbers of input frames.** Runtime is measured in seconds, and GPU memory usage is reported in gigabytes.

VGGT: Visual geometry grounded transformer

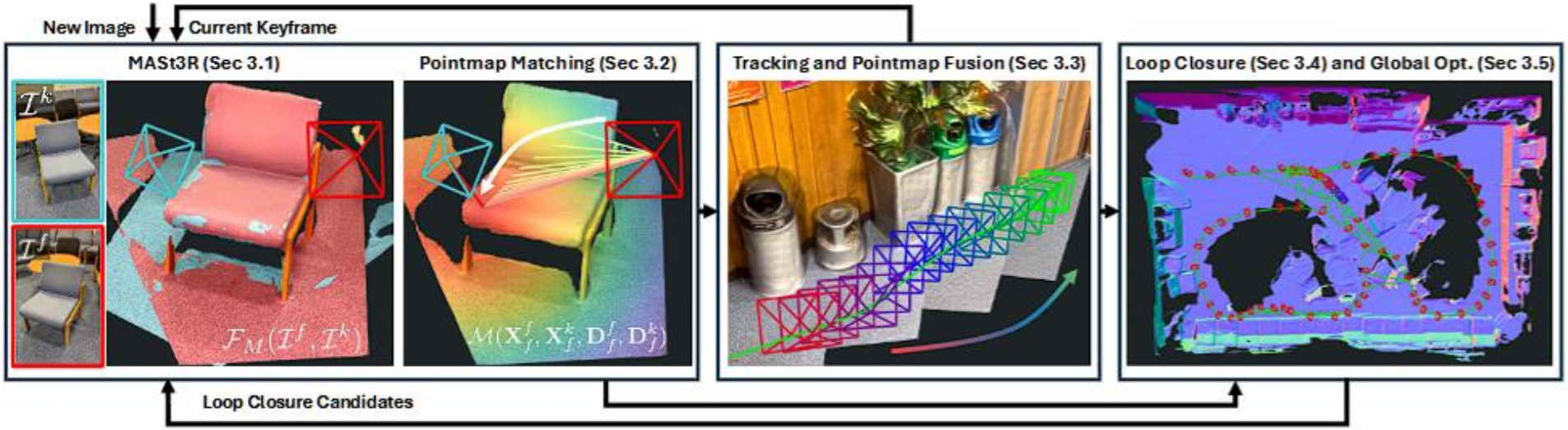
- Discussion
  - Single-view Reconstruction



- MASt3R-SLAM: Real-time dense SLAM with 3D reconstruction priors [CVPR 2025]
  - Imperial College London

MASt3R-SLAM

- Introduction
  - Robust and accurate visual SLAM
    - 하드웨어와 소프트웨어가 통합된 스택을 신중히 설계를 통해 가능
    - 그러나, plug-and-play 방식으로 쉽게 사용 가능하지 않음
      - Hardware 전문성과 calibration 요구
    - IMU와 같은 추가 센서 없이 minimal single camera setup에서 pose와 일관된 dense map제공 in-the-wild 환경 SLAM 시스템 존재하지 않음
    - Reliable dense SLAM system은 spatial intelligence의 중요한 발전이 될 것



MASt3R-SLAM

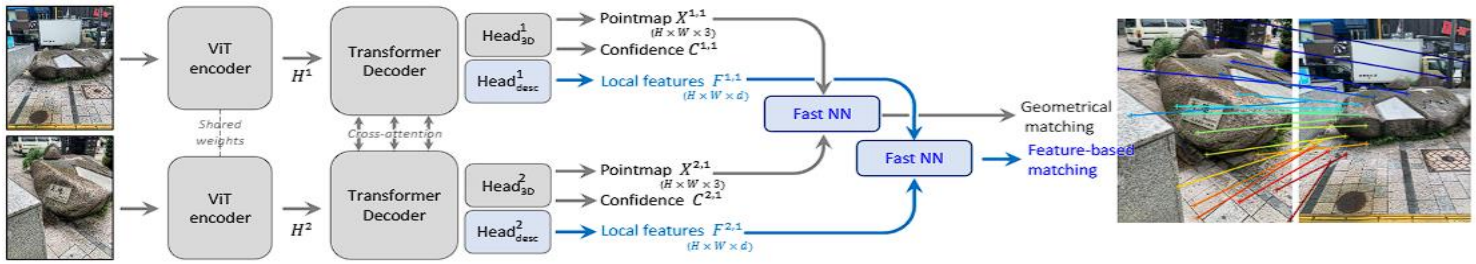
- Introduction
  - 2D image로부터의 SLAM 수행
    - Time-varying pose와 camera model, 3D scene geometry 추론 필요
    - Single-view prior
      - Monocular depth and normal 등 한 장의 이미지로부터 3D 기하를 예측
        - ✓모호성이 크고 view 간 일관성이 부족
    - Multi-view 기반 prior
      - Optical flow 등 모호성을 줄일 수 있음
      - Pixel 이동이 camera pose extrinsic과 camera model에 의존
        - ✓포즈와 기하 분리가 어려움
    - 3D geometry in a common coordinate frame 기반
      - 이미지 set 이용 pose, camera model, dense geometry 해결
    - Two-view 3D reconstruction prior
      - DUSt3R and MASt3R
      - 두 장 이미지로부터 common coordinate의 point map 출력

MASt3R-SLAM

- Introduction
  - Two-view 기반 3D 재구성 prior 사용 이유
    - SfM과 SLAM은 공간적 희소성을 활용해 불필요한 중복을 피함으로써 대규모 일관성을 유지
    - two-view 구조는 SfM의 기본 단위인 2-view 기하를 반영
    - 모듈화 된 구조를 통해 backend에서의 효율적인 판단과 robust consensus 가능
  - Main contribution
    - MASt3R 이용 2-view 기반 3D 재구성과 matching prior 기반 bottom-up 설계
      - SLAM system을 위한 순차적 데이터 입력 및 실시간 처리
      - Low-latency matching, map maintenance, large scale optimization 필요
        - ✓Frontend에서 point map local filtering, backend에서 대규모 global optimization
    - 고정된 카메라 모델이나 매개변수화 된 모델을 가정하지 않은 강건성
      - 모든 ray가 고유한 카메라 중심을 지나간다는 가정 외 제약 없음
        - ✓기존 연구의 경우 intrinsic calibration 주어진 상황 가정 다수 존재

MASt3R-SLAM

- Method
  - Preliminaries
    - DUSt3R's pipeline
      - Pair of images $I^i, I^j \in \mathbb{R}^{H\times W\times 3}$ 입력
      - Point map $X_i^i, X_i^j \in \mathbb{R}^{H\times W\times 3}$ 출력, Confidences $C_i^i, C_i^j \in \mathbb{R}^{H\times W\times 1}$
    - MASt3R's pipeline (additional head)
      - d-dimensional features for matching $D_i^i, D_i^j \in \mathbb{R}^{H\times W\times d}$
      - Corresponding confidence $Q_i^i, Q_i^j \in \mathbb{R}^{H\times W\times 1}$
      - $F_M(I^i, I^j)$로 MASt3R의 출력 정의



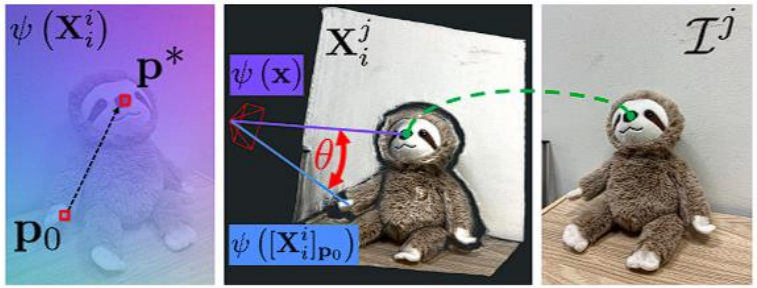
MASt3R pipeline. 회색 영역은 DUSt3R, 파란 head & Fast NN MASt3R's additional head

MASt3R-SLAM

- Method
  - Pointmap Matching
    - Tracking과 mapping을 위해 MASt3R의 pointmap과 features 기반 pixel matching
      - $m_{i,j} = M(X_i^i, X_i^j, D_i^i, D_i^j)$
    - 단순 brute-force matching 방식의 경우 quadratic complexity 발생
      - Global search over all possible pairs of pixel
      - DUSt3R의 경우 이를 회피하기 위해 3D point에 k-d tree 적용
        - ✓Pointmap 예측 오류 경우, 3D nearest-neighbor search 부정확
      - MASt3R의 경우 네트워크에서 high-dimensional feature 추가적으로 예측
        - ✓Wider baseline matching and coarse-to-fine scheme 제안
        - ✓Fine 탐색에 수 초의 실행 시간 소요, coarse 또한 k-d tree보다 느림
      - MASt3R-SLAM의 경우 최적화에서 영감을 얻어 local search 기반 접근 수행

MASt3R-SLAM

- Method
  - Pointmap Matching
    - Dense SLAM에서 주로 사용되는 projective data-association 사용
      - 해당 방식의 경우 closed-form projection 기반 parametric camera model 필요
        - ✓이전 프레임 3D point cloud, 현재 프레임 카메라 포즈로 변환 후 투영
      - 그러나, 본 연구의 경우 모든 ray가 단일한 카메라의 중심을 지난다는 단일 가정
        - ✓주어진 pointmap에 대해 generic camera model의 I^i with the rays $\psi(X_i^i)$
          - Generic camera calibration method 'Why having 10,000 parameters in your camera model is better than twelve' [CVPR 2020]
        - ✓각 point를 독립적으로 최적화하여 ray error 최소화
        - ✓$p^* = argmin_p \left\| \psi\left([X_i^i]_p\right) - \psi(x) \right\|^2, x \in X_i^j$



MASt3R-SLAM

- Method
  - Pointmap Matching
    - Normalized vector 사이의 Euclidean distance를 minimise하는 것은 normalized ray 사이의 각도를 최소화 하는 것과 동일하므로
      - $\|\psi_1 - \psi_1\|^2 = 2(1 - \cos\theta), \cos\theta = \psi_1^T \psi_2$
        - ✓'Why having 10,000 parameters in your camera model is better than twelve' [CVPR 2020] 유사한 방식으로 non-linear least squares form을 통해 Levenberg-Marquardt 알고리즘으로 업데이트 가능
          - LM 알고리즘 : 비선형 최소제곱 문제를 풀기 위한 고전적인 최적화 기법
          - 가우스-뉴턴 방식과 확률적 경사 하강법(SGD) $(J^T J + \lambda I)\Delta x = -J^T f$
          - (이하 생략)
      - 결론) 매칭 과정은 GPU에서 대규모 병렬처리가 가능하며, SLAM의 점진적 특성 활용 가능
        - ✓Tracking 과정은 2ms, 새롭게 추가되는 edge에 대해 a few ms 수행 가능
      - 포즈 추정값에 의존하지 않고 MASt3R의 출력만을 사용하여 unbiased 결과 제공

MASt3R-SLAM

- Tracking and Pointmap Fusion
  - Low-latency tracking in SLAM
    - Keyframe-based system, 현재 frame과 마지막 keyframe 사이 transformation 추정
      - 효율성을 위해 a single pass of the network 추구
    - Minimizing the 3D point error
      - $E_p = \sum_{m,n \in m_{f,k}} \left\| \frac{\tilde{X}^k_{k,n} - T_{kf} X^f_{f,n}}{w(q_{m,n}, \sigma^2_p)} \right\|_\rho$, $q_{m,n} = \sqrt{Q^f_{f,m} Q^k_{f,n}}$ (match confidence weighting)
      - Robustness를 위해 Huber norm $\|\blacksquare\|_\rho$ $w(\mathbf{q}, \sigma^2) = \begin{cases} \sigma^2/\mathbf{q} & \mathbf{q} > \mathbf{q}_{min} \\ \infty & \text{otherwise} \end{cases}$
      - 3D point error는 적절한 지표임에도, point map 예측에서 오류 발생 경우 왜곡 가능
        - ✓깊이 정보의 불일치는 상대적으로 자주 발생
    - 최종적으로 모든 예측 결과를 평균하여 하나의 point map으로 fuse
      - Pointmap은 초기 작은 baseline frame만을 사용하기 때문에 오류가 크고 신뢰도 낮음
      - 필터링을 통해 여러 시점 정보 통합

MASt3R-SLAM

- Method
  - Graph Construction and Loop Closure
    - Tracking 중 유효 매칭 수가 임계값 아래로 떨어질 경우 새로운 keyframe K_i 추가
      - K_i가 추가되면, 이전 keyframe K_{i-1}과 bidirectional edge 추가
      - 시간 순서대로 estimated pose를 제약
        - ✓그럼에도 drift가 발생할 수 있음
    - Small and large loop를 close하기 위해, MASt3R –SfM의 ASMK 사용
      - Aggregated Selective Match Kernel : feature 이용 image 간 graph 구성
  - Backend Optimization
    - 최적화 목표는 모든 pose와 기하 정보 간 전역적인 일관성을 달성하는 것
      - $E_p = \sum_{m,n \in m_{f,k}} \left\| \frac{\psi(\tilde{X}^k_{k,n}) - \psi(T_{kf} X^f_{f,n})}{w(q_{m,n}, \sigma^2_p)} \right\|_\rho$
      - Gauss-Newton 방법을 사용해 solve하되, dense하지 못하기 때문에 sparse Cholesky decomposition 수행, CUDA 구현으로 병목 현상 회피(이후 생략)

MASt3R-SLAM

- Method
  - Results
    - 다양한 real-world dataset 대상 시스템 평가
      - Localization을 위해 monocular SLAM benchmark 이용
    - Camera Pose Estimation

Table 1. Absolute trajectory error (ATE (m)) on TUM RGB-D [38].

		360	desk	desk2	floor	plant	room	rpy	teddy	xyz	avg
Calibrated	**ORB-SLAM3** [4]	X	0.017	0.210	X	0.034	X	X	X	**0.009**	-
	DeepV2D [42]	0.243	0.166	0.379	1.653	0.203	0.246	0.105	0.316	0.064	0.375
	DeepFactors [6]	0.159	0.170	0.253	0.169	0.305	0.364	0.043	0.601	0.035	0.233
	DPV-SLAM [22]	0.112	0.018	0.029	0.057	0.021	0.330	0.030	0.084	0.010	0.076
	DPV-SLAM++ [22]	0.132	0.018	0.029	0.050	0.022	0.096	0.032	0.098	0.010	0.054
	GO-SLAM [54]	0.089	**0.016**	0.028	0.025	0.026	0.052	**0.019**	0.048	0.010	0.035
	DROID-SLAM [45]	0.111	0.018	0.042	**0.021**	**0.016**	**0.049**	0.026	0.048	0.012	0.038
	Ours	**0.049**	**0.016**	**0.024**	0.025	0.020	0.061	0.027	**0.041**	**0.009**	**0.030**
Uncalibrated	**DROID-SLAM*** [45, 48]	0.202	0.032	0.091	0.064	0.045	0.918	0.056	0.045	0.012	0.158
	Ours*	0.070	0.035	0.055	0.056	0.035	0.118	0.041	0.114	0.020	0.060

    - Dense Geometry Evaluation

Table 3. Reconstruction Evaluation on 7-Scenes and EuRoC with all metrics in metres.

7-scenes	ATE	Accuracy	Completion	Chamfer
DROID-SLAM	0.049	0.115	**0.040**	0.077
Spann3R @20	N/A	0.069	0.047	0.058
Spann3R @2	N/A	0.124	0.043	0.084
Ours	**0.047**	0.074	0.057	0.066
Ours*	0.066	**0.068**	0.045	**0.056**
EuRoC	ATE	Accuracy	Completion	Chamfer
DROID-SLAM	**0.022**	0.173	**0.061**	0.117
Ours	0.041	**0.099**	0.071	**0.085**
Ours*	0.164	0.108	0.072	0.090

MASt3R-SLAM

- Method
  - Limitations
    - 프론트엔드에서 포인트맵(pointmap)을 필터링함으로써 정확한 지오메트리를 추정
      - 전체 글로벌 최적화 과정에서 모든 지오메트리를 정제(refine)하지 않음
    - DROID-SLAM은 bundle adjustment 통해 픽셀 단위의 깊이(depth) 최적화
      - MASt3R-SLAM의 경우 비일관적인(incoherent) 지오메트리를 허용
    - 예측에서 나타나는 geometry 일관성 유지하며, 3D에서 글로벌하게 정합시키는 방식을 real-time 수행하는 것은 흥미로운 방향이 될 것
    - MASt3R는 pinhole camera 이미지에만 학습되어 있기 때문에, 카메라 distortion 커질수록 geometry 예측 성능이 저하