Leveraging Features & Geometry for Generalized Gaussian Splatting

2026년도 동계 세미나

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

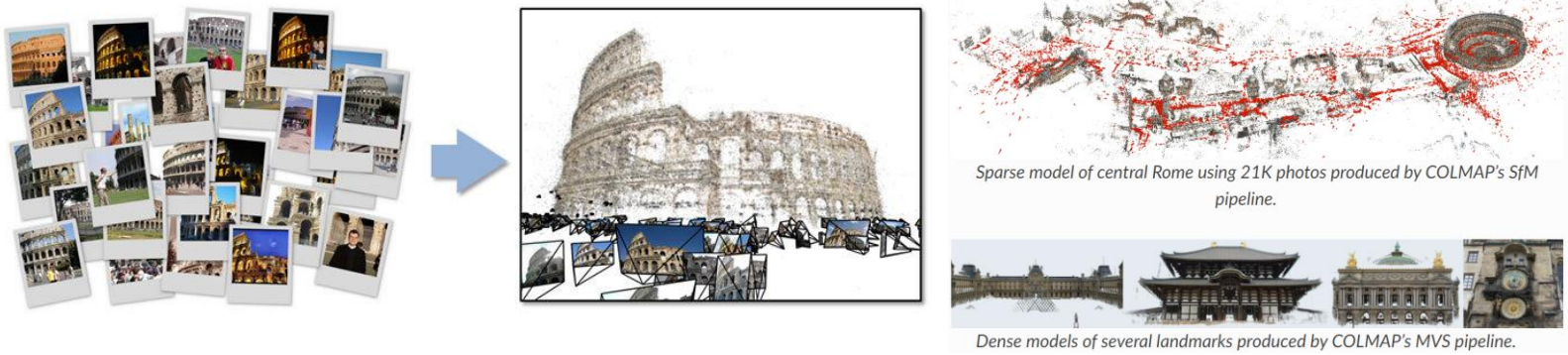
Presented By
남준식

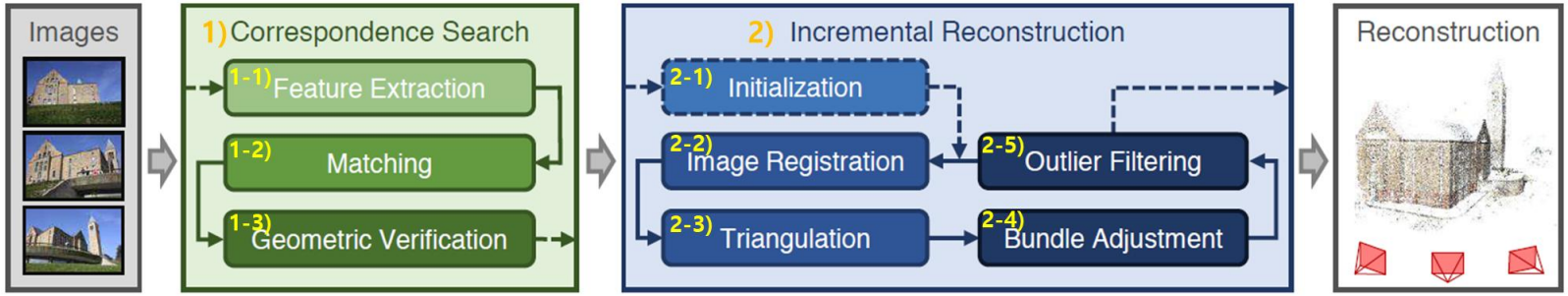
Outline

- Background
  - 3D Reconstruction & Feed forward 3DGS
- 논문 선정의 이유
- Feat2GS: Probing Visual Foundation Models with Gaussian Splatting
  - CVPR 2025
- MoGe: Unlocking Accurate Monocular Geometry Estimation for Open-Domain Images with Optimal Training Supervision
  - CVPR 2025

Background

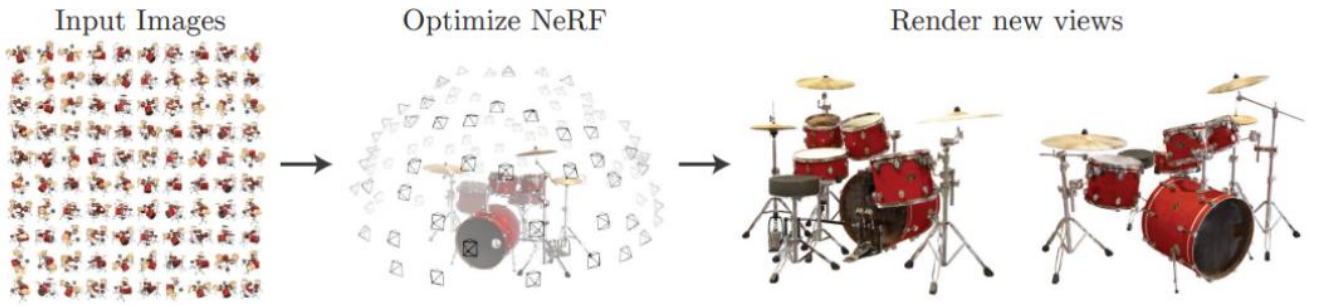
- **Classic Reconstruction**
  - SfM(Structure from Motion) / MVS(Multi-View Stereo)
    - Multi-view에서 geometry를 복원하는 classic pipline
      - 장점 : GT 없이도 Reconstruction이 가능
      - 단점 : Texture/Dynamic scene/Occlusion 에 취약

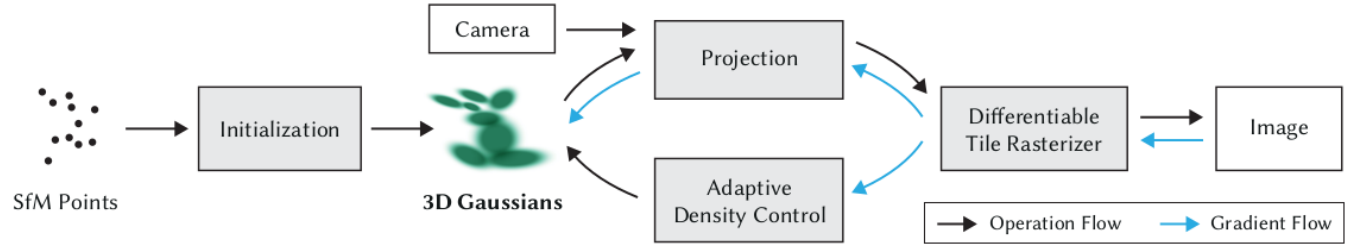




Background

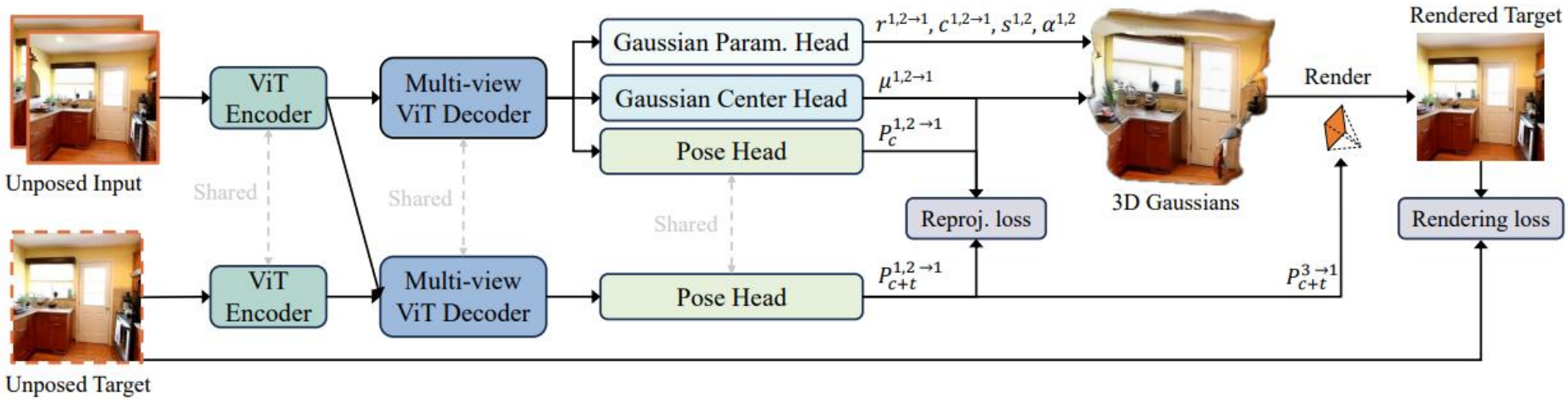
- **Neural Rendering**
  - NeRF / 3DGS
    - Rendering 기반 최적화로 3D를 만들고 NVS로 검증하는 pipeline
    - NeRF : 3D 공간은 MLP를 사용하여 표현
      - 고품질 이지만, inference 속도가 느리고 최적화에 필요한 비용이 크다
    - 3DGS : 3D Gaussian attribute를 사용하여 3D 공간을 표현
      - 매우 빠른 rendering 속도를 보이며, explicit 파라미터로 표현되기에 분석이 용이





Background

- **Feed Forward Gaussian Splatting**
  - Gaussian Splatting 의 단점
    - 기존의 3DGS은 SfM과 같은 방법으로 얻은 Initial Point Cloud가 필요
    - Iterative algorithm(Densify & Prune)을 사용해서 최종 Gaussian Splatting을 진행
  - Image-to-3D Gaussian Splatting을 목표로 하는 Feed Forward 기반의 방법에 대한 연구 진행
    - 3D-aware Model (Dust3R, Mast3R , etc …) 의 feature를 활용하여 3D Gaussian Attribute를 regression
    - 입력 이미지만으로 3D reconstruction을 한 다음, 3D reconstruction 결과를 통해서 Camera Pose를 추정
    - 단일/소수 뷰 입력에서는 occlusion, scale/depth ambiguity 때문에 3D 일관성(geometry consistency) 확보가 어렵고, 결과적으로 렌더링 아티팩트가 쉽게 발생



Background

- 논문 선정의 이유
  - 3D GT 데이터의 취득에 많은 비용이 들고 data의 다양성도 제한적임
  - 2D image로 부터 3D representation을 만들고 검증하는 방법이 중요해짐

- Feat2GS
  - Off-the shelf VFM (Visual Foundation Model) 에서의 feature를 활용하기 이전에 각 feature의 3D 특성을 살펴보기 위함
  - 3D Dataset으로 학습되지 않은 VFM에 대해서 3D task에서의 활용 가능성을 보여줌
    - VFM의 feature가 실제로 3D 정보를 얼마나 담는지를 보여준 논문

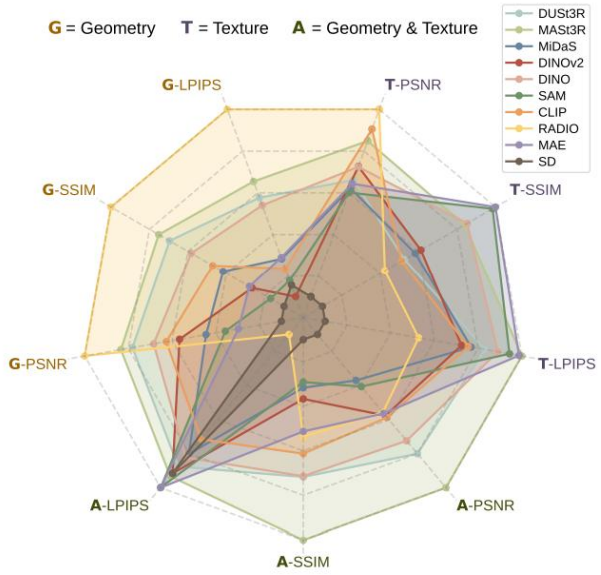
- MoGe
  - 단일 이미지에서 Open-domain 3D는 scale/depth ambiguity에 의해 supervision이 충돌하는 문제가 존재
  - MoGe는 단일 이미지에서도 open-domain geometry를 안정적으로 뽑는 학습 방법 제안

" Feat2GS: Probing Visual Foundation Models with Gaussian Splatting."

[CVPR, 2025]

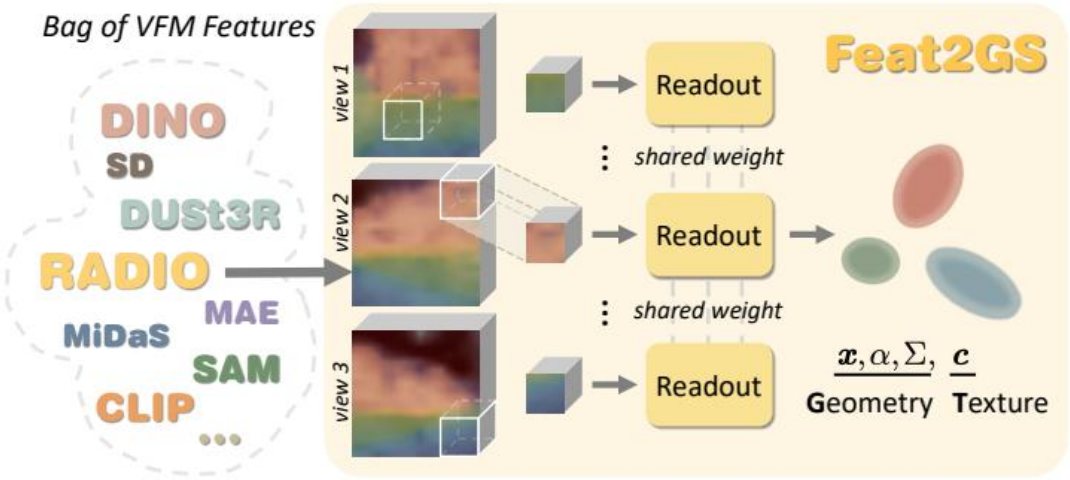
Introduction

- 3D awareness of VFM feature
  - 현재 VFM은 downstream task에서 중요한 요소로 사용되고 있음
  - VFM을 3D-downstream task에 적용하기 위해서는 VFM Feature에 대한 탐구가 중요함
  - 3D-awareness에 대한 평가는 Depth/Normal/2D Matching/Tracking과 같은 방법으로 이루어짐
    - VFM feature의 texture awareness와 Multi-view dense consistency를 평가하기에는 적절하지 않음
    - 논문에서는 2D multi-view data를 통해서 NVS task를 진행함으로써 평가를 진행
      - 이때 논문에서 제안하는 GTA scheme을 통해서 feature 특성을 파악



Method

- Feat2GS
  - Gaussian Attribute estimation
    - VFM을 frozen 하여 feature map만을 추출
    - PCA를 사용하여 feature의 channel 수를 일치시킨 다음, shallow MLP를 통해 attribute 추정
      - PCA output dim (= 256) → 256 → out dim(Gaussian attributes)
  - GTA probing schemes
    - Feature의 특성을 알아보기 위한 schemes
    - Gaussian attributes 중에서 geometry, texture 에 해당하는 attribute 만을 추정하여 feature 특성을 확인
    - Geometry(G), Texture(T), All(A) 로 나누어서 진행



Feat2GS	-Geometry	-Texture	-All	InstantSplat [22]
Feature-Readout	$\boldsymbol{x}, \alpha, \Sigma$	$\boldsymbol{c}$	$\boldsymbol{x}, \boldsymbol{c}, \alpha, \Sigma$	-
Free-Optimize	$\boldsymbol{c}$	$\boldsymbol{x}, \alpha, \Sigma$	-	$\boldsymbol{x}, \boldsymbol{c}, \alpha, \Sigma$

$$\{\boldsymbol{x}_i, \alpha_i, \Sigma_i\} = g_\Theta^{(G)}(\mathbf{f}_i) \quad \text{- Geometry}$$

$$\{\boldsymbol{c}_i\} = g_\Theta^{(T)}(\mathbf{f}_i) \quad \text{- Texture}$$

$$\{\boldsymbol{x}_i, \alpha_i, \Sigma_i, \boldsymbol{c}_i\} = g_\Theta^{(A)}(\mathbf{f}_i) \quad \text{- All}$$

$$\min_{\Theta, \boldsymbol{T}} \|\mathcal{R}(g_\Theta(\mathbf{f}), \boldsymbol{T}) - \mathcal{I}\| \quad \text{- Training Loss}$$

Experiments

- Experimental setup
  - VFM으로는 2D 와 3D 데이터로 각각 학습된 10 개의 모델을 비교
  - VFM 이외에 IUVRGB를 feature로써 사용
    - Image index(I), Pixel coordinates(UV), colors(RGB)를 concat하여 feature로 사용
  - Dataset으로는 Multi-view dataset(LLFF, DTU, etc ..)를 사용
    - 각 dataset은 2 ~ 7개의 view를 sparse view가 되도록 sampling 하여 사용
    - Test view는 training view에서 가장 거리가 먼 view를 사용
  - Metric
    - PSNR, SSIM, LPIPS를 사용

VFM	Arch.	Channel	Supervision	Dataset
DUSt3R [94]	ViT-L/16	1024	Point Regression	3D DUSt3R-Mix
MASt3R [49]	ViT-L/16	1024	Point Regression	3D MASt3R-Mix
MiDaS [70]	ViT-L/16	1024	Depth Regression	3D MiDaS-Mix
DINOv2 [64]	ViT-B/14	768	Self Distillation	2D LVD-142M
DINO [9]	ViT-B/16	768	Self Distillation	2D ImageNet-1k
SAM [44]	ViT-B/16	768	Segmentation	2D SA-1B
CLIP [69]	ViT-B/16	512	Contrastive VLM	2D WIT-400M
RADIO [72]	ViT-H/16	1280	Multi-teacher Distillation	2D DataComp-1B
MAE [33]	ViT-B/16	768	Image Reconstruction	2D ImageNet-1k
SD [75]	UNet	1280	Denoising VLM	2D LAION

Dataset	Scene Type	Complexity	View Range	Views
LLFF [60]	Indoor	Simple	Small	2
DTU [1]	Indoor Object	Simple	Small	3
DL3DV [52]	Indoor / Outdoor	Moderate	Medium	5-6
Casual	Daily Scenario	Moderate	Medium	4-7
MipNeRF360 [4]	Unbounded	Moderate	360	6
MVimgNet [111]	Outdoor Object	Moderate	180-360	2-4
T&T [46]	Indoor / Outdoor	High	Large	6

Experiments

- Quantitative Results
  - 대부분의 VFM Model에서 Geometry Scheme 을 사용한 경우 가장 성능이 좋았음
  - Feature 에서 Texture 를 바로 출력으로 얻는 경우 성능이 떨어지는 결과를 보임
    - VFM의 feature 에서는 Texture-awareness가 부족한 것은 확인할 수 있음

	LLFF									DL3DV									Casual								
	Geometry			Texture			All			Geometry			Texture			All			Geometry			Texture			All		
Feature	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
DUSt3R	19.88	.7442	.2123	19.01	.7120	.2262	19.87	.7190	.2691	19.64	.7338	.3196	18.01	.6815	.3219	19.39	.7360	.3458	19.29	.6562	.3580	17.54	.5693	.3750	19.19	.6556	.4050
MASt3R	19.89	.7447	.2123	19.01	.7115	.2261	19.99	.7250	.2657	19.64	.7334	.3188	18.07	.6813	.3211	19.41	.7373	.3464	19.30	.6550	.3576	17.59	.5708	.3722	19.37	.6588	.4027
MiDaS	19.81	.7420	.2154	19.00	.7129	.2261	19.86	.7142	.2733	19.47	.7271	.3311	17.94	.6796	.3224	19.22	.7291	.3493	19.24	.6545	.3612	17.52	.5693	.3757	18.96	.6516	.4073
DINOv2	19.77	.7345	.2226	19.04	.7133	.2254	19.91	.7163	.2637	19.47	.7293	.3288	18.00	.6805	.3223	19.27	.7317	.3479	19.42	.6524	.3698	17.64	.5701	.3754	19.21	.6535	.4023
DINO	19.81	.7423	.2140	18.98	.7121	.2260	19.97	.7212	.2744	19.60	.7324	.3209	17.97	.6790	.3219	19.41	.7359	.3476	19.24	.6513	.3614	17.50	.5683	.3756	19.10	.6566	.4056
SAM	19.72	.7354	.2181	18.98	.7133	.2260	19.76	.7144	.2629	19.48	.7297	.3271	17.97	.6822	.3218	19.20	.7272	.3459	19.32	.6469	.3704	17.52	.5725	.3736	19.19	.6569	.3981
CLIP	19.78	.7378	.2221	19.02	.7113	.2276	19.74	.7136	.2822	19.53	.7295	.3304	18.05	.6771	.3235	19.22	.7310	.3563	19.21	.6552	.3719	17.46	.5669	.3743	19.05	.6582	.4084
RADIO	19.73	.7402	.2207	19.06	.7101	.2301	19.56	.6999	.3252	19.48	.7313	.3139	18.03	.6748	.3254	19.20	.7316	.3654	19.54	.6545	.3465	17.52	.5666	.3748	18.67	.6533	.4216
MAE	19.75	.7363	.2183	19.00	.7128	.2249	19.92	.7209	.2612	19.54	.7288	.3248	17.98	.6821	.3207	19.34	.7310	.3448	19.03	.6502	.3690	17.51	.5691	.3758	19.18	.6547	.3974
SD	19.62	.7293	.2234	18.85	.7100	.2297	19.78	.7121	.2656	19.31	.7251	.3276	17.79	.6784	.3260	19.10	.7282	.3500	19.24	.6483	.3649	17.38	.5698	.3789	18.86	.6505	.4053
IUVRGB	15.55	.5765	.3986	19.75	.7303	.2262	15.38	.6175	.4308	14.78	.6326	.4541	18.75	.7023	.3250	14.05	.6431	.4386	13.17	.5454	.5248	17.88	.5927	.3846	13.71	.5917	.4955

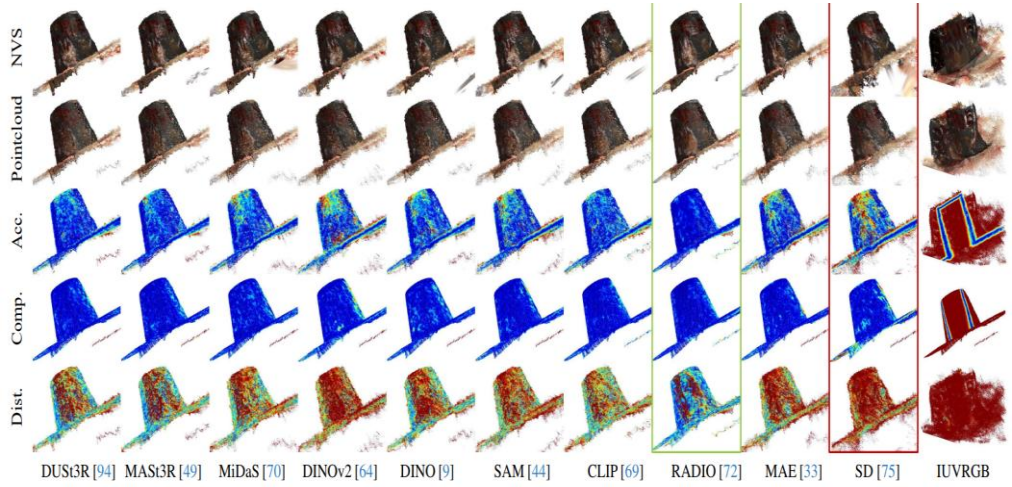
	MipNeRF 360									MVImgNet									Tanks and Temples (T&T)								
	Geometry			Texture			All			Geometry			Texture			All			Geometry			Texture			All		
Feature	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
DUSt3R	20.82	.5008	.3795	19.10	.4489	.3816	21.02	.5048	.4752	19.47	.6004	.3073	16.88	.5348	.3334	19.43	.5937	.3674	18.85	.6458	.3715	17.53	.6222	.3328	18.61	.6477	.4023
MASt3R	20.92	.5093	.3745	19.21	.4540	.3803	20.92	.5054	.4749	19.49	.6008	.3032	16.91	.5350	.3337	19.49	.5983	.3637	18.80	.6428	.3703	17.68	.6238	.3319	18.76	.6512	.3991
MiDaS	20.89	.5059	.3815	19.05	.4509	.3813	20.84	.5004	.4795	19.35	.5900	.3222	16.82	.5336	.3343	19.34	.5910	.3672	18.53	.6374	.3798	17.64	.6238	.3333	18.32	.6428	.4039
DINOv2	20.81	.4946	.3953	19.05	.4495	.3821	20.75	.4924	.4684	19.35	.5896	.3246	16.88	.5359	.3344	19.43	.5943	.3674	18.71	.6432	.3772	17.58	.6214	.3348	18.43	.6443	.4064
DINO	20.91	.5054	.3769	19.18	.4545	.3795	20.83	.5010	.4772	19.44	.5982	.3071	16.90	.5394	.3329	19.41	.5952	.3683	18.75	.6416	.3733	17.66	.6233	.3330	18.61	.6467	.4030
SAM	20.73	.4913	.3945	19.14	.4556	.3775	20.75	.4949	.4639	19.23	.5899	.3188	16.84	.5346	.3346	19.29	.5915	.3649	18.65	.6421	.3780	17.49	.6217	.3338	18.43	.6425	.4029
CLIP	20.80	.4982	.3913	19.28	.4543	.3807	20.88	.4984	.4773	19.41	.5945	.3098	16.96	.5362	.3358	19.37	.5969	.3695	18.92	.6463	.3729	17.81	.6226	.3316	18.75	.6515	.4052
RADIO	20.87	.5100	.3620	19.35	.4550	.3819	20.91	.5067	.5127	19.54	.6105	.2949	16.99	.5373	.3366	19.60	.5955	.3946	19.19	.6612	.3480	17.84	.6225	.3321	19.01	.6574	.4109
MAE	20.82	.4992	.3884	19.14	.4572	.3781	20.79	.4995	.4668	19.23	.5909	.3142	16.84	.5355	.3328	19.25	.5914	.3680	18.65	.6395	.3758	17.55	.6234	.3333	18.49	.6451	.4000
SD	20.71	.4962	.3985	18.89	.4472	.3839	20.59	.4929	.4672	19.08	.5881	.3185	16.63	.5313	.3389	19.06	.5838	.3660	18.69	.6422	.3772	17.32	.6217	.3374	18.55	.6467	.4020
IUVRGB	16.45	.4075	.5910	19.96	.4797	.3911	16.41	.4187	.5929	14.83	.5069	.4648	17.84	.5568	.3431	15.38	.5362	.4699	15.29	.5846	.4736	18.60	.6526	.3396	15.17	.5948	.4718

Experiments

- Qualitative Results

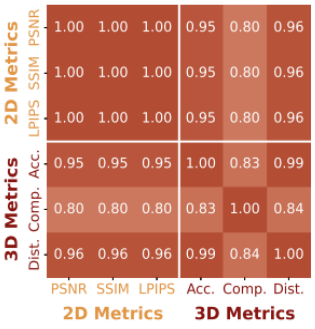
Experiments

- NVS correlates with 3D metrics
  - 3D metric으로 측정한 결과와 2D metric 간의 correlation을 측정한 결과
    - 3D metric 결과와 2D metric 결과를 비교한 결과 높은 연관성을 보이는 것을 확인
    - 2D rendering 품질이 올라가면 GT point cloud 와의 3D metric 또한 상승
  - Completeness 쪽 correlation이 상대적으로 낮게 측정
    - 2D NVS metric이 Occlusion과 같은 3D problem을 덜 민감하게 반응하는 것을 보여줌



	2D Metrics			3D Metrics		
Feature	PSNR↑	SSIM↑	LPIPS↓	Acc.↓	Comp.↓	Dist.↓
DUSt3R	21.36	.7772	.2195	2.439	1.316	6.955
MASt3R	21.44	.7792	.2177	2.321	1.286	6.557
MiDaS	21.09	.7712	.2254	2.934	1.412	8.230
DINOv2	21.01	.7695	.2277	3.101	1.337	8.588
DINO	21.40	.7783	.2187	2.440	1.316	6.885
SAM	20.93	.7660	.2304	3.176	1.339	8.785
CLIP	21.26	.7752	.2215	2.357	1.209	6.739
RADIO	21.78	.7871	.2042	1.886	1.326	5.431
MAE	20.96	.7666	.2289	2.963	1.337	8.374
SD	20.76	.7638	.2343	4.334	1.603	11.594
IUVRGB	16.09	.6825	.3134	13.015	16.957	46.671

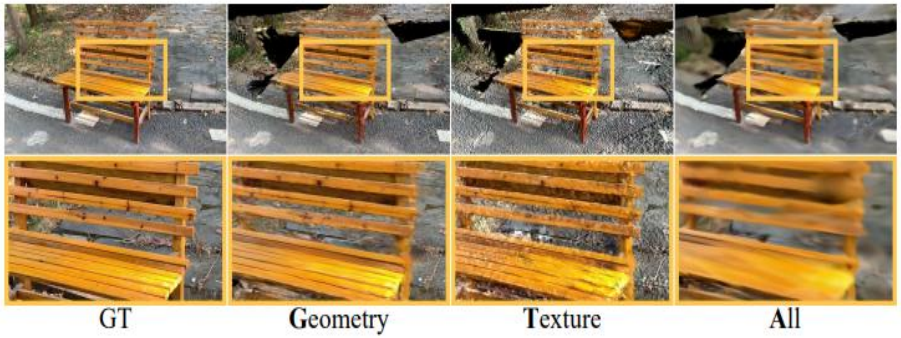
(a) 2D Metrics *vs.* 3D Metrics

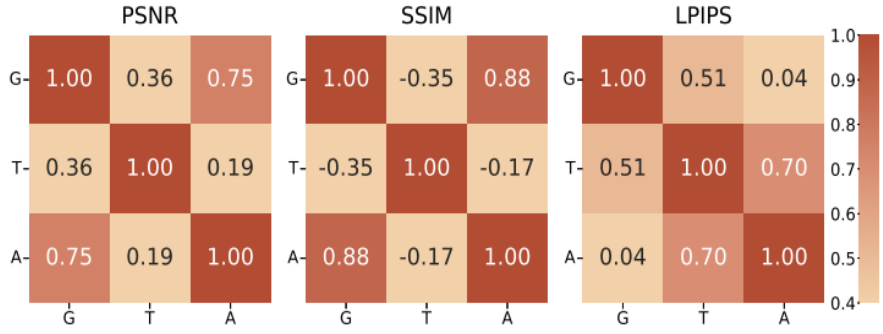


(b) Correlation Matrix

Experiments

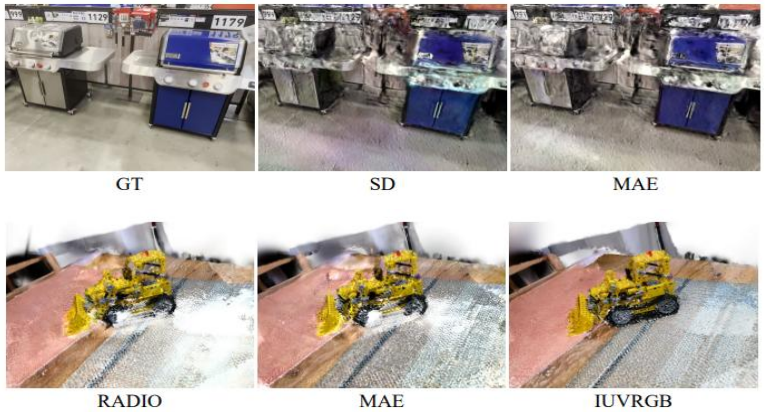
- GTA Modes comparison
  - Texture mode 에서 geometry가 망가지는 것을 확인
    - VFM feature가 geometry에 대해서 강한 Prior로써 동작하는 것을 보여줌
  - All mode 에서는 geometry와 비교하여 blur가 강한 것을 확인
    - VFM feature에서 high-frequency information이 부족한 것을 보여주는 부분
    - 논문에서는 VFM의 texture awareness가 부족하다고 해석
  - 2D Metric에 대한 분석
    - PSNR, SSIM의 경우, 3D reconstruction에서 Geometry reconstruction 성능과 연관성이 존재
    - LPIPS의 경우, Texture/appearance 성능과 크게 연관성 있음을 보여줌





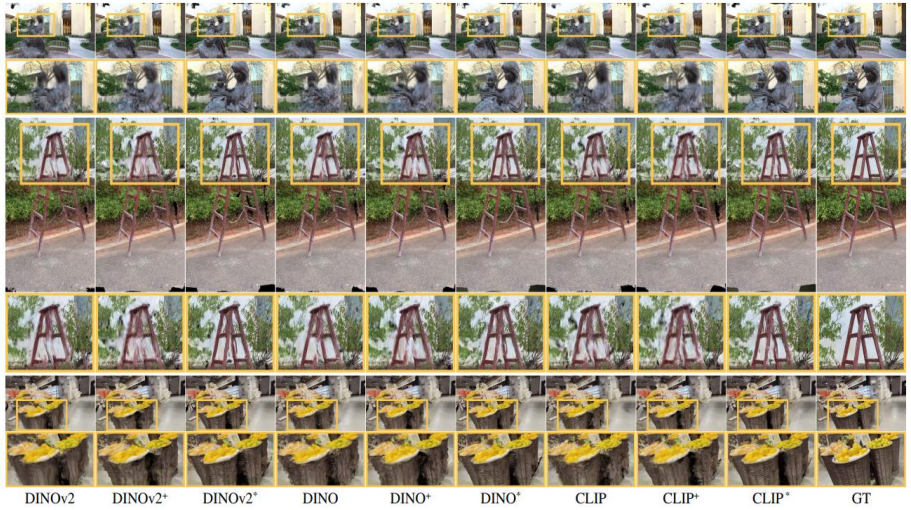
Experiments

- Geometry awareness & texture awareness of VFM
  - Geometry awareness
    - Supervision 으로 3D, 2.5D data로 학습한 모델이 blur가 적은 3D reconstruction 결과를 보임
    - Depthmap prediction(MiDaS)와 3D supervision(DUst3R, MASt3R) 결과를 비교한 결과 3D supervision이 우수함
      - MiDaS는 Relative Depth를 추정하기에, view/ distance에 따라 결과 값이 달라지기 때문
        - ✓View에 따라서 gauge 가 다르게 추정하는 것이 가능하기에 발생함
  - Texture awareness
    - MAE 나 SD와 같은 appearance 관련하여 학습한 모델 역시 texture-awareness가 부족한 것을 확인
      - 두 모델이 High-frequency 정보가 noise 처럼 학습 되는 것 때문



Experiments

- Feature upsampling^{+} vs Fine Tuning*
  - VFM에서 얻는 feature에서 high-frequency texture를 담기 위한 방법
    - Feature upsampling 과 feature fine-tuning(warmup stage에서 feature optimize)각각에 대해서 진행
  - Feature를 upsampling 하는 것 보다는 Fine tuning 하는 것이 시각적 품질이 우수함
    - 단순히 feature upsampling의 경우 feature 에서 손실된 high-freq 성분을 채우지 않고 interpolation 에 의해 채우기에, blur 정도가 심함
    - Fine tuning 방법의 경우 어떤 VFM이든 성능이 상승하는 것을 보여줌
      - Scene-specific 하게 feature가 학습되기 때문에, 해당 Scene에서의 고주파 정보를 학습하는 것이 가능해지기 때문

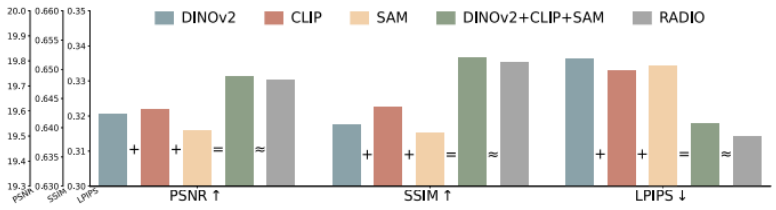


	All Datasets								
	Geometry			Texture			All		
Feature	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
DINOv2	19.59	.6406	.3364	18.03	.5951	.3291	19.50	.6388	.3760
DINOv2^{+}	19.67	.6480	.3202	18.10	.5950	.3291	19.58	.6443	.3894
DINOv2*	19.78	.6552	.2962	18.18	.5968	.3232	19.80	.6614	.3247
DINO	19.63	.6452	.3256	18.03	.5961	.3282	19.55	.6427	.3793
DINO^{+}	19.72	.6485	.3207	18.03	.5941	.3291	19.64	.6465	.3839
DINO*	19.74	.6557	.2918	18.09	.5949	.3235	19.69	.6630	.3154
CLIP	19.61	.6436	.3331	18.10	.5947	.3289	19.50	.6416	.3832
CLIP^{+}	19.68	.6466	.3222	18.09	.5941	.3286	19.63	.6468	.3842
CLIP*	19.70	.6540	.2959	18.19	.5962	.3242	19.67	.6599	.3199

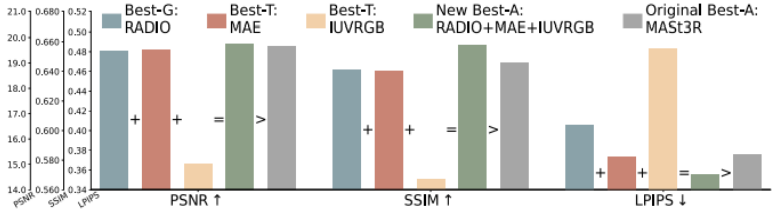
Experiments

- Feature Concatenation
  - VFM의 특성을 합쳐서 사용한 결과를 비교
  - VFM + IUVRGB 모델을 사용하여 Gaussian attribute를 추정한 결과 성능이 크게 증가
    - VFM을 단일로 사용하는 것 보다 여러 Model을 Concat 하는 것이 더 좋은 것을 보여줌
  - VFM이 많아지게 될수록 오히려 중복된 정보로 인해 성능이 개선이 미미

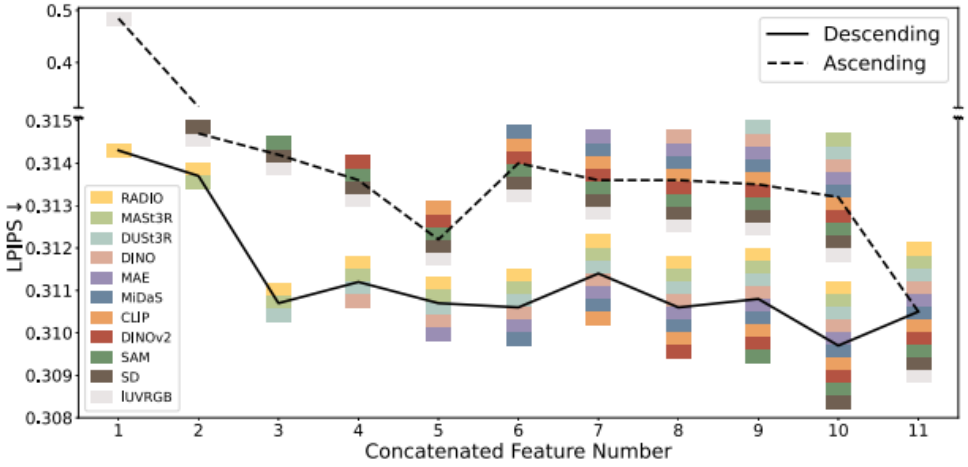
Method	All Datasets		
	PSNR ↑	SSIM ↑	LPIPS ↓
InstantSplat [22]	18.87	0.6044	0.3039
Feat2GS w/ RADIO	19.73	0.6513	0.3143
Feat2GS w/ concat all	**19.80**	0.6545	0.3105
Feat2GS w/ DUSt3R	19.66	0.6469	0.3247
Feat2GS w/ DUSt3R*	19.75	**0.6561**	**0.2928**



(a) Concatenated Features *vs.* RADIO



(b) Concatenated Features *vs.* MASt3R



" MoGe: Unlocking Accurate Monocular Geometry Estimation for Open-Domain Images with Optimal Training Supervision"

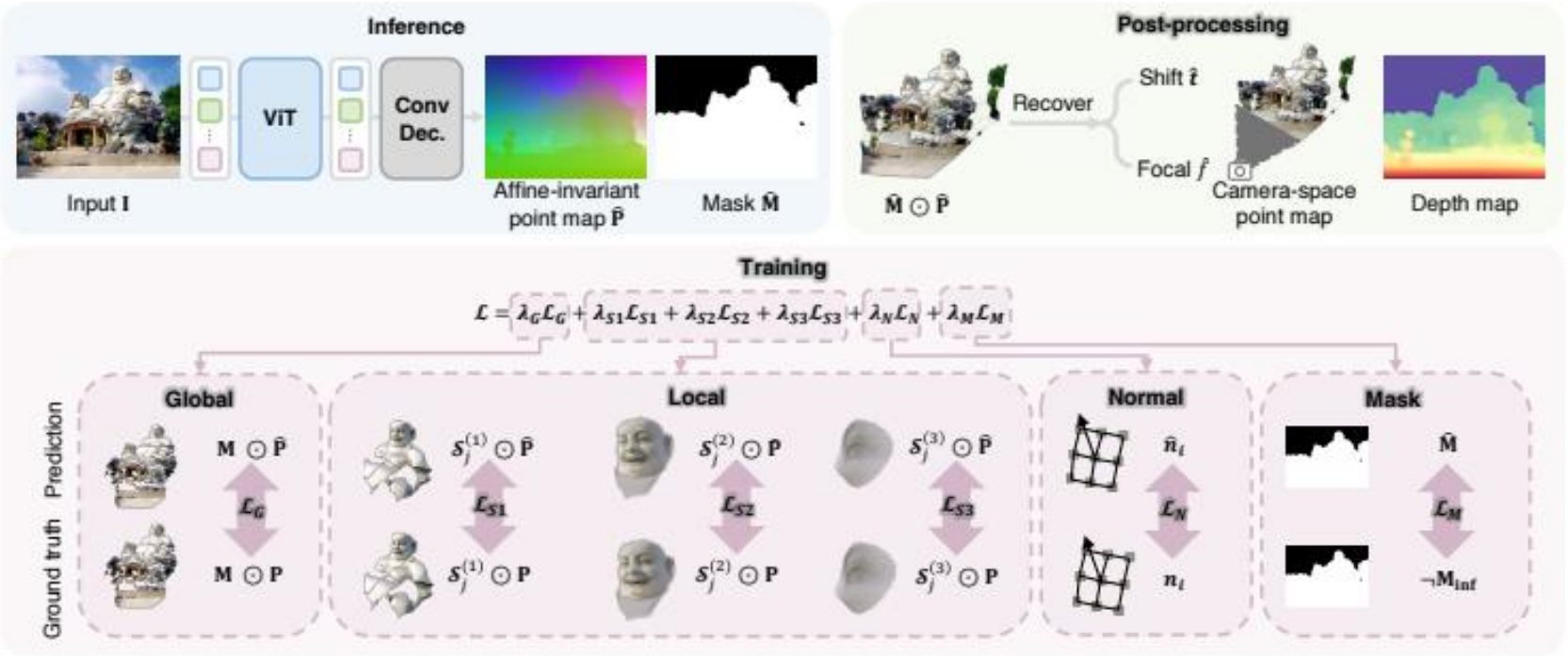
[CVPR, 2025]

Introduction

- Monocular Geometry Estimation(MGE)
  - Scale Ambiguity Problem
    - Monocular setting 에서 물체의 실제 depths (Z value)를 추정하기 어려움
    - Projection 될 때, Intrinsic parameter(Focal) 과 Z value가 곱해지기에, Optimal Value의 추정이 어려움
    - $u = f_x \cdot \frac{X}{Z} + c_x$, $v = f_y \cdot \frac{Y}{Z} + c_y$, $X = \frac{u - c_x}{f_x} Z$, $Y = \frac{v - c_y}{f_y} Z$
      - Projection 할 때, $\frac{X}{Z}, \frac{Y}{Z}$ 로 반영되기에, scale이 곱해져도 동일한 값으로 반영
  - Recent Study
    - DepthMap Estimation Method (Camera Parameter 추정 → 3D Lifting)
      - Foundation Depths Estimation Model을 사용해서 Depth 및 카메라 parameter 추정 후 해당 3D를 바탕으로 복원
    - Direct 3D reconstruction Method (3D Lifting → Camera Parameter 추정)
      - Dust3R, Mast3R과 같이 image로 부터 3D reconstruction을 진행한 다음 down stream으로 camera parameter 추정

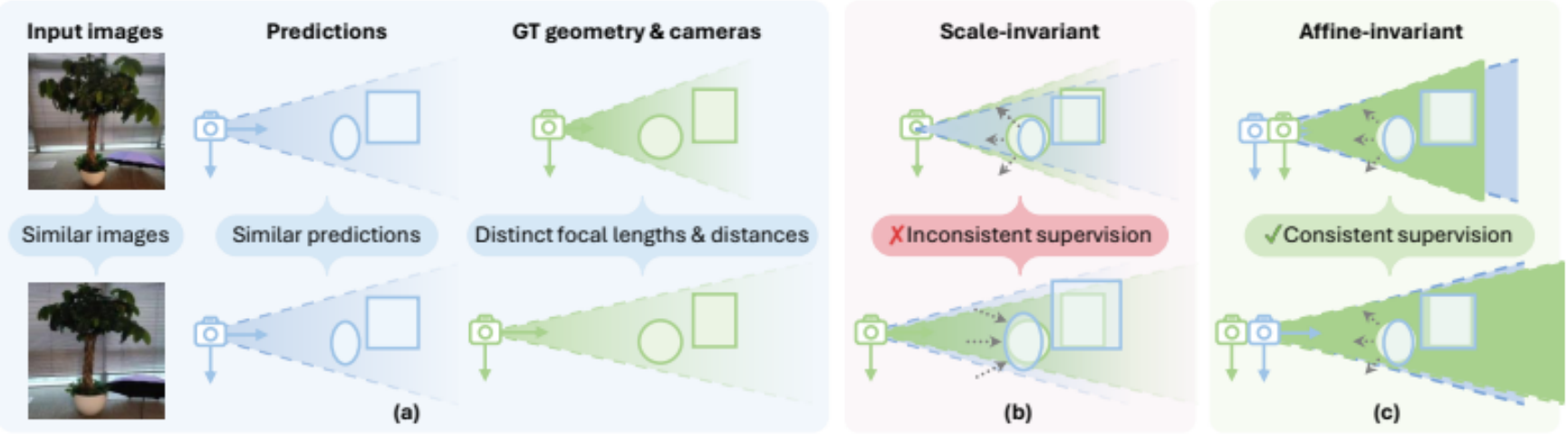
Introduction

- MoGe (Monocular Geometry estimation)
  - Single Image-to-3D points Model ($F_\theta : I \rightarrow P$)
    - Camera parameter는 Downstream으로 예측, Open-Domain에서 사용하기 위함
  - Affine-invariant Point map을 이용하여 Robust한 3D point map을 생성
    - $\mathbf{P} \simeq s\mathbf{P} + t, \forall s, \forall t$



Method

- Affine-invariant Point map
  - Scale-invariant $P \simeq sP$
    - Similar images라도 거리 초점 차이 (f, d)가 다르면 GT가 달라짐
    - scale-only 정렬은 supervision이 서로 충돌하는 문제가 존재
  - Affine-invariant $P \simeq sP + (0,0,t_z)$
    - scale + depth shift로 카메라 거리 차이를 보정 → GT ↔ pred 정렬 안정
    - t_z로 depth offset을 흡수 하여, shape-position의 결합 문제 완화



Method

- Training Objectives
  - Global point map supervision

$$\mathcal{L}_G = \sum_{i \in \mathcal{M}} \frac{1}{z_i} \| s\hat{\mathbf{p}}_i + \mathbf{t} - \mathbf{p}_i \|_1 \quad (s^*, \mathbf{t}^*) = \underset{s,\mathbf{t}}{\operatorname{argmin}} \sum_{i \in \mathcal{M}} \frac{1}{z_i} \| s\hat{\mathbf{p}}_i + \mathbf{t} - \mathbf{p}_i \|_1$$

- Optimal s* ,t* 를 구하고 Pred Point map을 최적화 하는 구조
- Outlier에 Robust 하기 위해서 s*, t*를 L1-Loss를 통해서 Optimize
  - L1-Loss의 경우 Optimal point 근처에서는 미분이 안되기에, Gradient Descent로는 수렴이 느림
  - Linear Programming(Simplex, Interior-point) 와 같은 방법으로 최적화
    - ✓ LP 의 경우 $O(N^3)$의 시간 복잡도를 가지기에, $N = H \times W \times T$ 에서는 계산이 많음
    - ✓ 논문에서는 이를 최적화 하기 위한 $O(N^2 \log N)$의 Optimal Solver(ROE alignment)를 제안
      - Breaking Point 근처에서 Optimal Point 가 생김을 이용

Method

- Training Objectives
  - ROE Alignment(Robust, Optimal, Efficient Alignment)
    - L1 Loss property

      ⁘ $\min_t \sum_i w_i || t - a_i ||_1$에서 Optimal point는 특정 index k 에서 $t = a_k$에서 존재.

      ✓ k를 고정한다면 t 를 a_k 로 치환하는 것이 가능

    - L1 Loss의 특성을 적용하면 다음과 같음

$$(s^*, \mathbf{t}^*) = \underset{s,\mathbf{t}}{\operatorname{argmin}} \sum_{i \in \mathcal{M}} \frac{1}{z_i} \|s\hat{\mathbf{p}}_i + \mathbf{t} - \mathbf{p}_i\|_1 \longrightarrow \min_s \sum_{i \in M} \frac{1}{z_i} ||s\hat{p}_i + p_i + z_k - s\hat{z}_k||_1$$

⁘ 이때, Outlier 영향을 줄이기 위해, Loss 계산 범위를 제한 ($\min(\cdot, \tau)$)

Algorithm 1 Overview of ROE alignment

function SOLVESUBPROBLEM()
  Enumerate breakpoints of the piecewise linear function.
  Find extrema among the breakpoints by their derivatives.
  return $s^{(k)}$ with smallest objective value $l^{(k)}$ at extrema.
end function

for index $k = 1$ to N **do** ▷ *parallel computation*
  Formulate subproblem by substituting t_z with $z_k - s^{(k)}\hat{z}_k$.
  Solve scale $s^{(k)}$ and $l^{(k)}$ via SOLVESUBPROBLEM().
  Obtain translation $t_z^{(k)}$ as $z_k - s^{(k)}\hat{z}_k$.
end for
Select optimal s^* and t_z^* with smallest function value $l^{(k)}$.

Algorithm 2 ROE alignment subproblem *w/o* truncation

input: arrays $\hat{X}[1..n]$, $X[1..n]$, $W[1..n]$
output: optimal scale s^* and objective value l^* to Eq. 11

function SOLVESUBPROBLEM($\hat{X}, X, W$)
  sort arrays $\hat{X}, X, W$ by $X[i]/\hat{X}[i]$
  $Q[1..n] \leftarrow$ accumulated sum of $W * \hat{X}$
  $D[0..n] \leftarrow \{-Q[n]\} \cup \{2 \cdot Q[i] - Q[n]\}_{i=1}^n$
  $i^* \leftarrow$ the first i s.t. $D[i-1] \leq 0 \leq D[i]$
  $s^* \leftarrow X[i^*]/\hat{X}[i^*]$
  $l^* \leftarrow$ objective function value at s^*.
  return s^*, l^*.
end function

Method

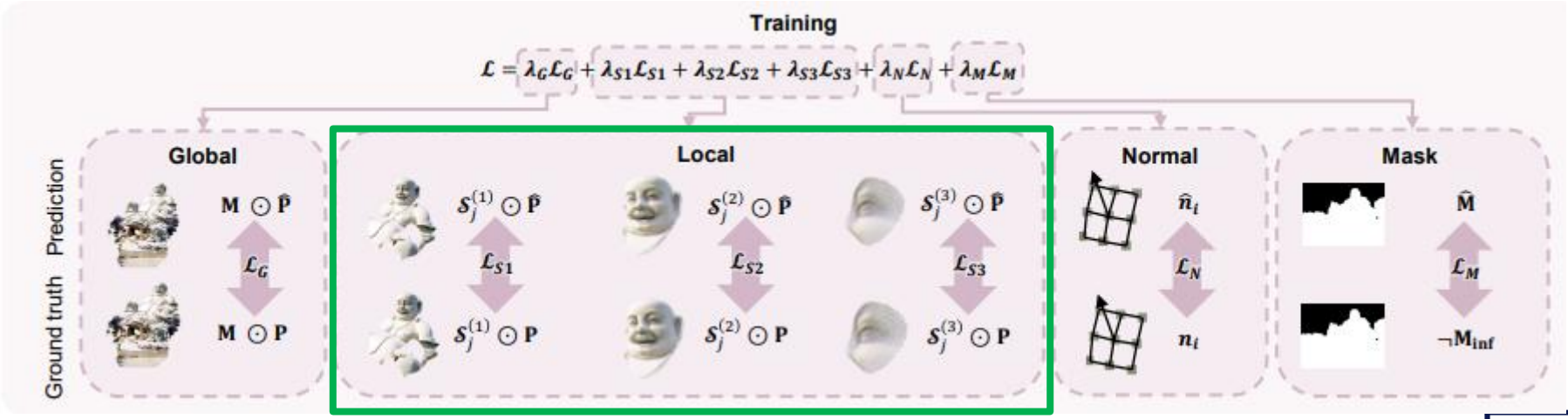
- Training Objectives
  - Multi-scale local geometry loss
    - Global s, t을 적용하는 경우 Local geometry에서의 alignment가 맞지 않는 문제가 있음
    - GT point에 대해서 3D space의 대각선 길이에 비례하여 ($\frac{1}{4}$,$\frac{1}{16}$,$\frac{1}{64}$), Local region을 지정
    - 해당 Local region 에서 scale과 translation를 맞추기 위함 ROE alignment를 통해 local scale ,translation 계산

$$\mathcal{S}_j = \{i \mid \|\mathbf{p}_i - \mathbf{p}_j\| \le r_j, i \in \mathcal{M}\}$$

$$r_j = \alpha \cdot z_j \cdot \frac{\sqrt{W^2+H^2}}{2 \cdot f}$$

,

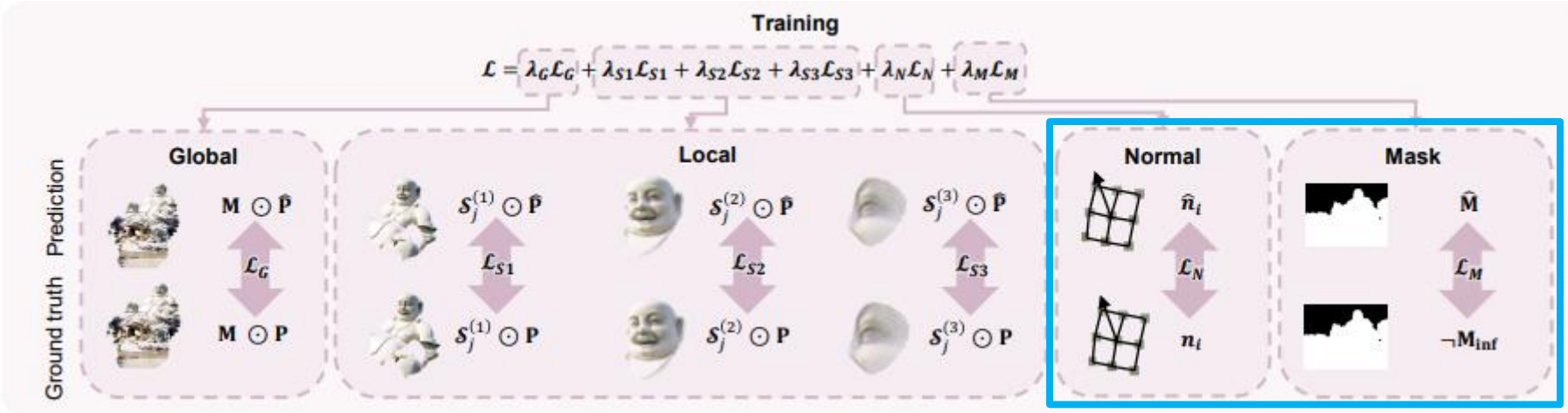
$$\mathcal{L}_{S(\alpha)} = \sum_{j \in \mathcal{H}_\alpha} l_{\mathcal{S}_j} = \sum_{j \in \mathcal{H}_\alpha} \sum_{i \in S_j} \frac{1}{z_i} \left\| s_j^* \hat{\mathbf{p}}_i + \mathbf{t}_j^* - \mathbf{p}_i \right\|_1$$



Multi-scale local geometry loss

Method

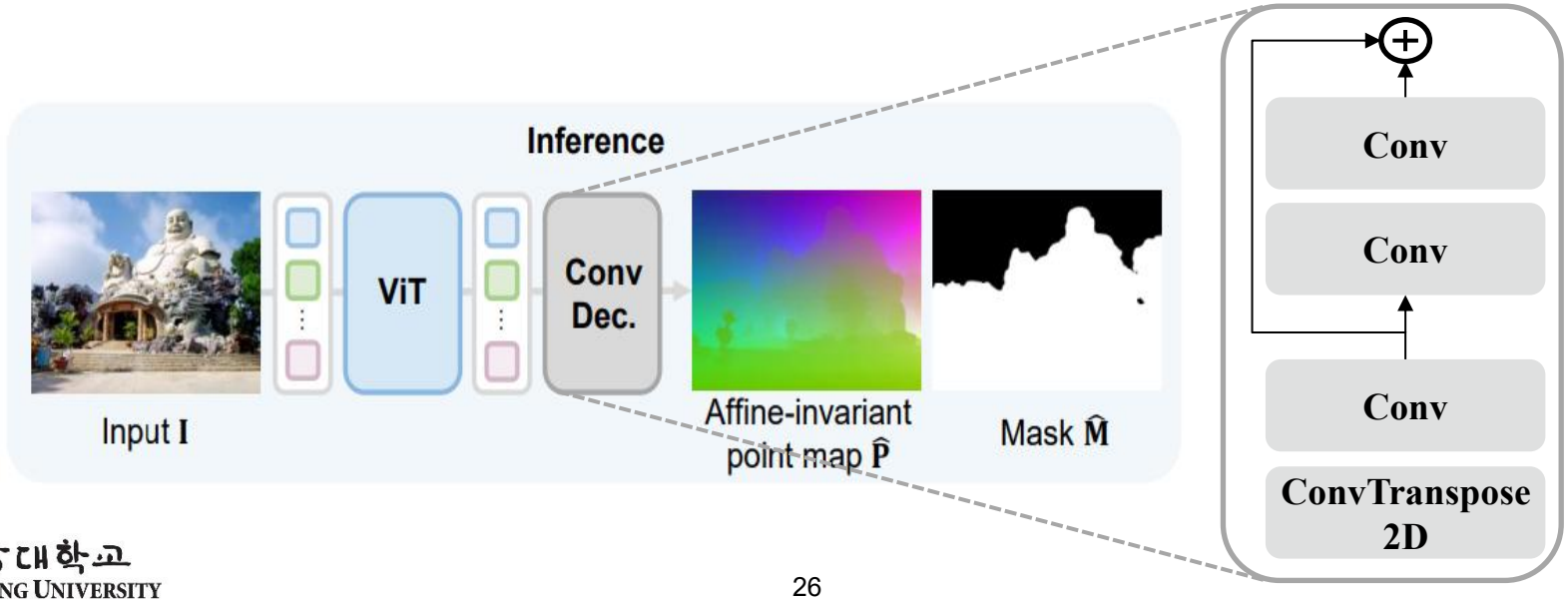
- Training Objectives
  - Normal Loss $\mathcal{L}_N = \sum_{i \in \mathcal{M}} \angle(\hat{\mathbf{n}}_i, \mathbf{n}_i)$
    - Surface quality를 높이기 위해서 사용
    - Predicted normal의 경우 인접한 point와의 cross product를 통해서 계산
  - Mask Loss $\mathcal{L}_M = \|\hat{\mathbf{M}} - (1 - \mathbf{M}_{\text{inf}})\|_2^2$
    - MoGe의 Mask head의 학습을 위해서 사용
    - 하늘과 같은 depth가 ∞ 인 영역의 경우 학습 안정성을 저해하기에, masking 하기 위함
    - SegFormer[NeurIPS 2021]의 결과를 GT로 사용



Normal & Mask loss

Method

- MoGe Architecture
  - DINOv2 ViT encode를 frozen 하여 사용
    - DINOv2의 마지막 4개의 layer의 output에 대해서 1x1 Conv 이후 summation 하여 feature map으로 사용
  - Conv Decoder
    - ConvTranspose(2x upsample) → Conv → ResBlock 으로 구성된 stage를 3번 사용
    - 매 단계마다 feature에 UV(2-channel)를 concat하여 position cue로 활용



Experiments

- Experiments Setting
  - Baseline model
    - Monocular point map estimation (LeReS, Unidepths, Dust3R)
    - Depth map estimation(Zoedepth, MiDas v3, Geowizard, Metric3D v2, DA v1, DA v2)
  - Point map estimation
    - NYUv2, KITTI, ETH3D, iBims-1,. Sintel, GSO, DDAD, DIODE
      - Sintel의 Sky region 및 DIODE의 boundary artifacts 는 제거 하여 사용
  - Metric
    - Point Map metric $\longrightarrow$ $Rel_p = \frac{||\hat{p} - p||_2}{||p||_2}$, $\delta^p = \frac{||\hat{p} - p||_2}{\min(||p||_2, ||\hat{p}||_2)} < 0.25$
    - Depth Metric $\longrightarrow$ $Rel_d = \frac{||\hat{z} - z||_2}{||z||_2}$, $\delta^d = \max(\frac{z}{\hat{z}}, \frac{\hat{z}}{z}) < 1.25$
    - FOV Metric
      - Vertical FOV가 45° 이상인 NYUv2/ ETH3D/ iBim-1 에서 진행
      - 이미지 중심을 기준으로 Crop하여 FOV 변형하여 평균 오차 및 중앙값 비교

Name	Domain	#Frames	Type	Weight
A2D2[21]	Outdoor/Driving	196K	C	0.8%
Argoverse2[62]	Outdoor/Driving	1.1M	C	7.4%
ARKitScenes[3]	Indoor	449K	B	8.6%
DIML-indoor[14]	Indoor	894K	D	4.8%
BlendedMVS[69]	In-the-wild	115K	B	12.0%
MegaDepth[35]	Outdoor/In-the-wild	92K	B	5.6%
Taskonomy[74]	Indoor	3.6M	B	14.1%
Waymo[51]	Outdoor/Driving	788K	C	6.4%
GTA-SfM[57]	Outdoor/In-the-wild	19K	A	2.8%
Hypersim[45]	Indoor	75K	A	5.0%
IRS[58]	Indoor	101K	A	5.6%
KenBurns[40]	In-the-wild	76K	A	1.6%
MatrixCity[34]	Outdoor/Driving	390K	A	1.3%
MidAir[19]	Outdoor/In-the-wild	423K	A	4.0%
MVS-Synth[27]	Outdoor/Driving	12K	A	1.2%
Spring[37]	In-the-wild	5K	A	0.7%
Structured3D[76]	Indoor	77K	A	4.8%
Synthia[47]	Outdoor/Driving	96K	A	1.2%
TartanAir[60]	In-the-wild	306K	A	5.0%
UrbanSyn[25]	Outdoor/Driving	7K	A	2.1%
ObjaverseV1[12]	Object	167K	A	4.8%

< Training Dataset >

Experiments

- Quantitative Results
  - Point map estimation
    - 기존의 point map 방법인 Scale-invariant point map 과 비교하여 Affine-invariant point map이 우수한 것을 확인할 수 있었음
    - Local point map의 경우, 한 장면에 여러 객체가 있는 조건이 필요하기에, NYUv2, KITTI, GSO 는 평가에서 제외

Method	NYUv2		KITTI		ETH3D		iBims-1		GSO		Sintel		DDAD		DIODE		Average		
	$Rel^p\downarrow$	$\delta_1^p\uparrow$	$Rel^p\downarrow$	$\delta_1^p\uparrow$	$Rel^p\downarrow$	$\delta_1^p\uparrow$	$Rel^p\downarrow$	$\delta_1^p\uparrow$	$Rel^p\downarrow$	$\delta_1^p\uparrow$	$Rel^p\downarrow$	$\delta_1^p\uparrow$	$Rel^p\downarrow$	$\delta_1^p\uparrow$	$Rel^p\downarrow$	$\delta_1^p\uparrow$	$Rel^p\downarrow$	$\delta_1^p\uparrow$	Rank↓
Scale-invariant point map																			
LeReS	16.9	76.0	31.6	28.4	17.1	75.8	18.5	72.2	14.7	76.0	38.6	30.6	32.0	39.4	27.6	46.4	24.6	55.6	3.94
DUSt3R	5.53	97.1	15.2	87.9	10.7	90.6	6.18	95.4	4.54	99.3	34.8	50.3	21.4	70.1	12.4	86.7	13.8	84.7	2.75
UniDepth	5.33	**98.4**	5.96	**98.5**	18.5	77.6	5.29	**97.4**	6.58	99.6	33.0	48.9	**11.4**	90.2	12.3	91.0	12.3	87.7	2.09
Ours	**4.86**	**98.4**	**5.47**	97.4	**4.58**	**98.9**	**4.63**	97.1	**2.58**	**100**	**22.3**	**69.5**	12.3	**90.3**	**6.58**	**94.5**	**7.91**	**93.3**	**1.22**
Affine-invariant point map																			
LeReS	9.51	91.4	26.1	49.1	14.7	79.6	11.0	88.6	8.91	95.2	29.7	55.5	29.4	46.7	15.1	80.1	18.1	73.3	3.94
DUSt3R	4.45	97.4	12.7	83.3	7.27	95.0	5.04	96.0	3.07	99.6	30.3	56.6	19.7	71.2	8.97	88.7	11.4	86.0	2.94
UniDepth	3.93	**98.4**	**4.29**	**98.6**	12.2	89.6	4.65	**98.0**	2.99	99.8	28.5	58.4	**10.3**	90.5	8.56	90.9	9.43	90.5	1.81
Ours	**3.68**	98.3	4.86	97.2	**3.57**	**99.0**	**3.61**	97.3	**1.14**	**100**	**16.8**	**77.8**	10.5	**91.4**	4.37	**96.4**	**6.07**	**94.7**	**1.31**
Local point map																			
LeReS	-	-	-	-	9.32	91.9	8.57	93.2	-	-	13.3	84.8	10.7	88.9	11.6	88.2	10.7	89.4	3.80
DUSt3R	-	-	-	-	6.05	94.8	5.44	95.9	-	-	11.8	87.0	9.24	90.8	7.32	93.1	7.97	92.3	2.30
UniDepth	-	-	-	-	8.61	92.6	5.92	96.0	-	-	13.4	84.3	8.18	92.0	9.95	90.0	9.21	91.0	2.90
Ours	-	-	-	-	**3.21**	**98.1**	**4.16**	**96.8**	-	-	**8.63**	**92.7**	**6.74**	**94.3**	**4.78**	**96.3**	**5.50**	**95.6**	**1.00**

< Quantitative Result for point map estimation >

Experiments

- Quantitative Comparison
  - Depth estimation
    - Depth 예측 결과 outlier의 비율이 baseline 모델과 비교하여 크게 줄어듬
  - FOV estimation
    - Learning-based camera calibration 방법인 Perspective Fields 와 Wild Camera를 추가로 비교
    - MoGe가 baseline 모델과 비교하여 FOV estimation에서 성능이 좋은 것을 확인

Method	NYUv2		KITTI		ETH3D		iBims-1		GSO		Sintel		DDAD		DIODE		Average		
	$\text{Rel}^{\text{d}}\downarrow$	$\delta_1^{\text{d}}\uparrow$	Rel.↓	$\delta_1^{\text{d}}\uparrow$	$\text{Rel}^{\text{d}}\downarrow$	$\delta_1^{\text{d}}\uparrow$	Rel.↓	$\delta_1^{\text{d}}\uparrow$	$\text{Rel}^{\text{d}}\downarrow$	$\delta_1^{\text{d}}\uparrow$	$\text{Rel}^{\text{d}}\downarrow$	$\delta_1^{\text{d}}\uparrow$	$\text{Rel}^{\text{d}}\downarrow$	$\delta_1^{\text{d}}\uparrow$	$\text{Rel}^{\text{d}}\downarrow$	$\delta_1^{\text{d}}\uparrow$	$\text{Rel}^{\text{d}}\downarrow$	$\delta_1^{\text{d}}\uparrow$	Rank↓
Scale-invariant depth map																			
LeReS	12.1	82.6	19.2	64.8	14.2	78.4	14.0	78.8	13.6	77.9	30.5	52.1	26.5	52.0	18.2	69.6	18.5	69.5	7.31
ZoeDepth	5.62	96.3	7.27	91.9	10.4	87.3	7.45	93.2	3.23	99.9	27.4	61.8	17.0	72.8	11.3	85.2	11.2	86.1	5.50
DUSt3R	4.40	97.1	7.81	90.6	6.04	95.7	4.98	95.8	3.27	99.5	31.1	57.2	18.6	73.3	8.91	88.8	10.6	87.2	5.00
Metric3D V2	4.69	97.4	4.00	98.5	3.84	98.5	4.23	97.7	2.46	99.9	20.7	69.8	7.41	94.6	**3.29**	**98.4**	6.33	94.3	2.07
UniDepth	3.86	**98.4**	**3.73**	**98.6**	5.67	97.0	4.79	97.4	4.18	99.7	28.3	58.8	10.1	**90.5**	6.83	92.8	8.43	91.6	3.00
DA V1	4.77	97.5	5.61	95.6	9.41	88.9	5.53	95.8	5.49	99.3	28.3	56.7	13.2	81.5	10.3	87.5	10.3	87.9	5.67
DA V2	5.03	97.3	7.23	93.7	6.12	95.5	4.32	**97.9**	4.38	99.3	23.0	65.2	14.7	78.0	7.95	90.0	9.09	89.6	4.06
Ours	**3.44**	**98.4**	4.25	97.8	**3.36**	**98.9**	**3.46**	97.0	**1.47**	**100**	**19.3**	**73.4**	**9.17**	**90.5**	4.89	94.7	**6.17**	**93.8**	**1.62**
Affine-invariant depth map																			
Marigold	4.63	97.3	7.29	93.8	6.08	96.3	4.35	97.2	2.78	99.9	21.2	75.0	14.6	80.5	6.34	94.3	8.41	91.8	2.25
GeoWizard	4.69	97.4	8.14	92.5	6.90	94.0	4.50	97.1	2.00	99.9	17.8	76.2	16.5	75.7	7.03	92.7	8.44	90.7	2.69
Ours	**2.92**	**98.6**	**3.94**	**98.0**	**2.69**	**99.2**	**2.74**	**97.9**	**0.94**	**100**	**13.0**	**83.2**	**8.40**	**92.1**	**3.16**	**97.5**	**4.72**	**95.8**	**1.00**
Affine-invariant disparity map																			
MiDaS V3.1	4.58	98.1	6.25	94.7	5.77	96.8	4.73	97.4	1.86	**100**	21.3	73.1	14.5	82.6	6.05	94.9	8.13	92.2	3.69
DA V1	4.20	98.4	5.40	97.0	4.68	98.2	4.18	97.6	1.54	**100**	20.1	77.6	12.7	86.9	5.69	95.7	7.31	93.9	2.31
DA V2	4.14	98.3	5.61	96.7	4.71	97.9	3.47	**98.5**	1.24	**100**	21.4	72.8	13.1	86.4	5.29	96.1	7.37	93.3	2.56
Ours	**3.38**	**98.6**	**4.05**	**98.1**	**3.11**	**98.9**	**3.23**	98.0	**0.96**	**100**	**18.4**	**79.5**	**8.99**	**91.5**	**3.98**	**97.2**	**5.76**	**95.2**	**1.06**

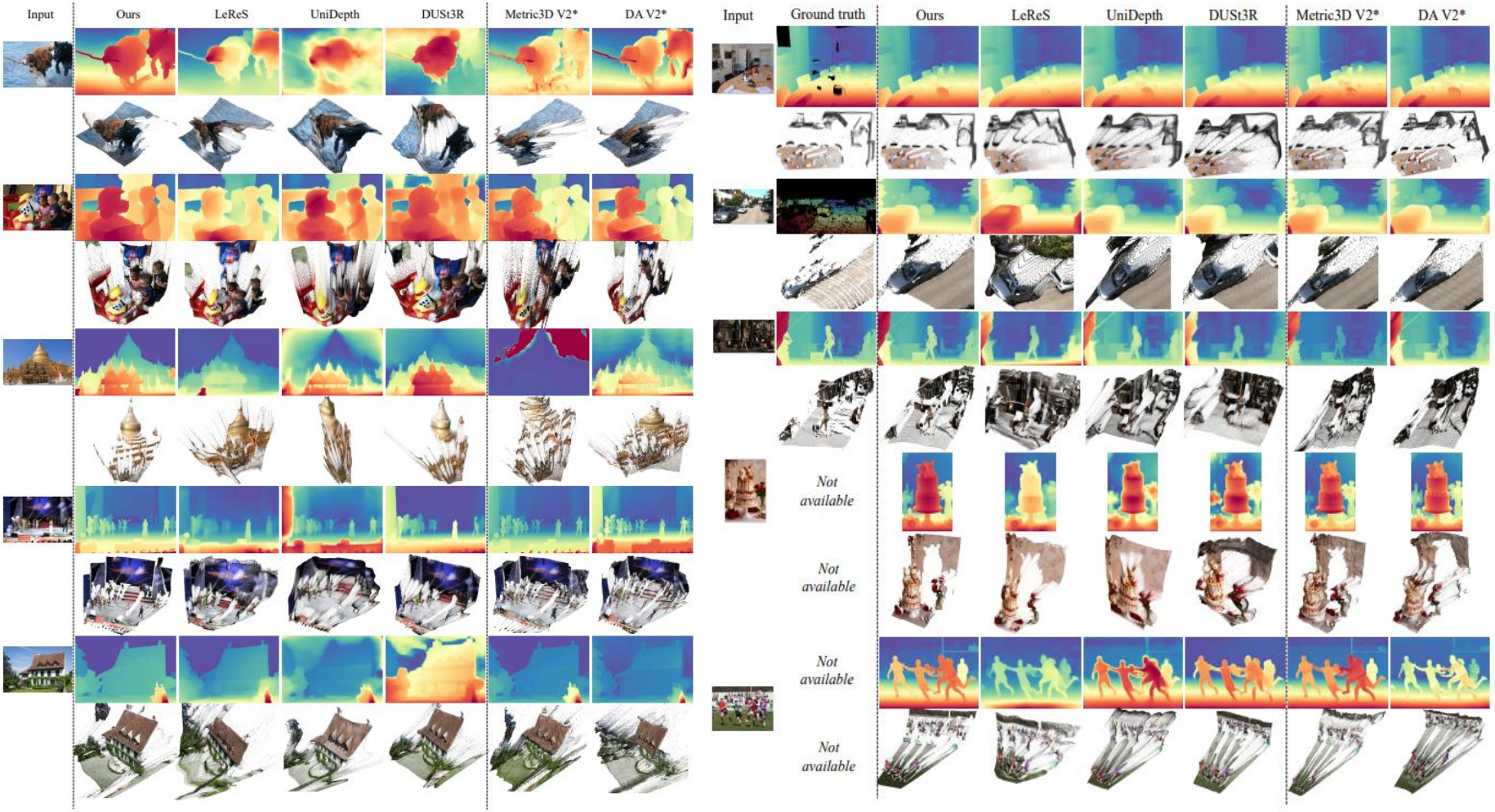
< Quantitative Result for depth estimation >

Method	NYUv2		ETH3D		iBims-1		Average		
	Mean↓	Med.↓	Mean	Med.↓	Mean↓	Med.↓	Mean↓	Med.↓	Rank↓
Perspective	5.38	4.39	13.6	11.9	10.6	9.30	9.86	8.53	5.00
WildCam	3.82	3.20	7.70	5.81	9.48	9.08	7.00	6.03	3.00
LeReS	19.4	19.6	8.26	7.19	18.4	17.5	15.4	14.8	5.53
DUSt3R	**2.57**	**1.86**	5.77	3.60	3.83	2.53	4.06	2.66	1.67
UniDepth	7.56	4.31	10.7	9.96	11.9	5.96	10.1	6.74	4.50
Ours	3.41	3.21	**2.50**	**1.54**	**2.81**	**1.89**	**2.91**	**2.21**	**1.50**

< Quantitative Result for FOV estimation >

Experiments

- Qualitative Comparison



< Qualitative Results >

Experiments

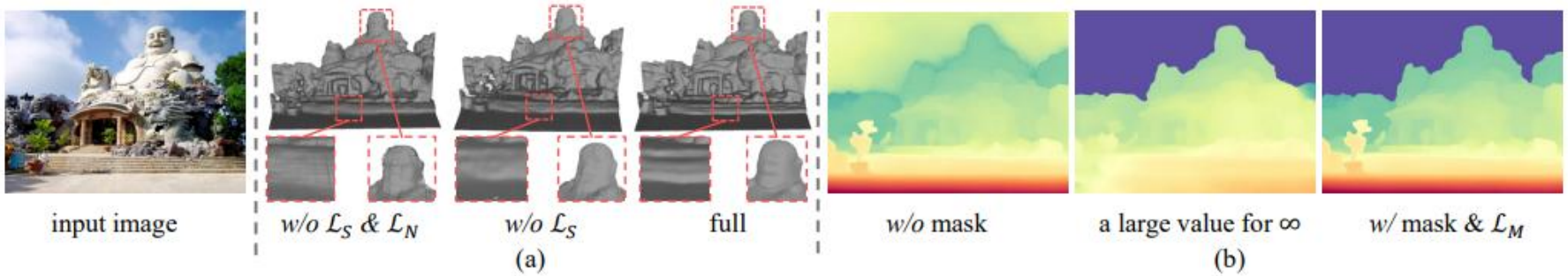
- Ablation Study
  - GT 랑 pred를 1:1로 비교하면 scale ambiguity 때문에 supervision 충돌이 발생
    - Affine-invariant point map을 사용하면 이를 완화 할 수 있음을 보여줌
  - Alignment method
    - ROE alignment 방법이 기존의 방법보다 우수한 것을 보여줌

L2 affine : $(s^*, \mathbf{t}^*) = \underset{s,\mathbf{t}}{\operatorname{argmin}} \sum_{i \in \mathcal{M}} \frac{1}{z_i} \left\| s\hat{\mathbf{p}}_i + \mathbf{t} - \mathbf{p}_i \right\|_2$

Med. affine : $\frac{p - t(P)}{s(P)}$, where $t(P) = (0, 0, median(Z))$, $s(P) = \frac{1}{N} \sum_{i=1}^{N} ||p_i - t(P)||_1$

Ablation	Point						Depth				Disparity	
	Scale-inv.		Affine-inv.		Local		Scale-inv.		Affine-inv.		Affine-inv.	
	$\text{Rel}^p\downarrow$	$\delta_1^p\uparrow$	$\text{Rel}^p\downarrow$	$\delta_1^p\uparrow$	$\text{Rel}^p\downarrow$	$\delta_1^p\uparrow$	$\text{Rel}^d\downarrow$	$\delta_1^d\uparrow$	$\text{Rel}^d\downarrow$	$\delta_1^d\uparrow$	$\text{Rel}^d\downarrow$	$\delta_1^d\uparrow$
SI-Log depth	11.2	88.7	9.09	90.6	9.19	91.2	8.94	90.1	7.27	92.6	8.23	92.1
Affine-inv. depth	29.9	51.4	29.0	52.7	12.2	86.0	28.9	52.7	**6.18**	**93.9**	15.9	76.6
ROE scale-inv.	10.3	89.8	8.34	91.6	8.59	91.9	8.27	90.9	6.73	93.2	7.90	92.6
L2 affine-inv.	13.5	84.2	10.3	88.2	9.48	91.0	11.1	85.7	8.03	91.2	9.37	90.5
Med. affine-inv.	10.9	89.0	8.97	90.7	9.44	90.7	9.10	89.8	7.50	92.4	8.74	91.8
ROE affine-inv.	**9.84**	**90.3**	**7.88**	**92.1**	**7.62**	**93.3**	**7.91**	**91.2**	6.29	93.7	**7.43**	**93.2**
Full *w/o* trunc.	9.81	90.5	7.91	91.7	**7.12**	**93.8**	7.92	**91.3**	6.31	93.5	7.45	93.1
Full w/o $\mathcal{L}_S$	9.98	90.3	7.94	**92.1**	7.47	93.4	7.94	91.2	6.30	93.6	7.47	93.2
Full	**9.78**	**90.6**	**7.83**	**92.1**	7.16	**93.8**	**7.82**	**91.3**	**6.20**	**93.7**	**7.30**	**93.3**

Table 4. Quantitative ablation study results. All experiments are conducted with a ViT-Base encoder. The first six rows are trained with global loss only.



감사합니다.