3D Gaussian Feature Fields based 6D Pose Estimation

2026년 동계 세미나

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

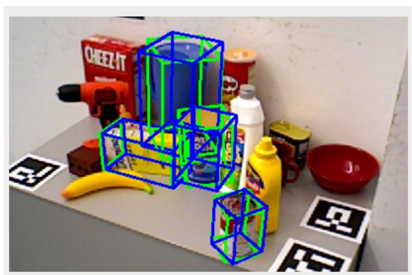
Presented By
장순원

Outline

- Introduction
  - 6D object pose estimation
  - 3D Gaussian Splatting & Feature Field
- From Sparse to Dense: Camera Relocalization with Scene-Specific Detector from Feature Gaussian Splatting
- Gaussian Splatting Feature Fields for (Privacy-Preserving) Visual Localization
- Conclusion

Introduction

- 6D object pose estimation
  - 2D view로부터 Camera 좌표계↔ Object 좌표계간 translation 및 rotation 추정
  - Detailed Tasks
    - Zero-shot, Model-free, Category-level pose estimation



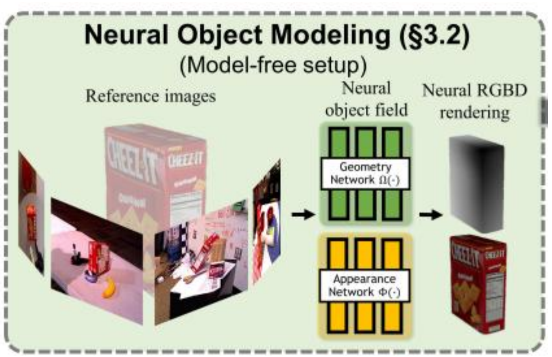
Query Image

- 3D CAD Model
- Camera intrinsic

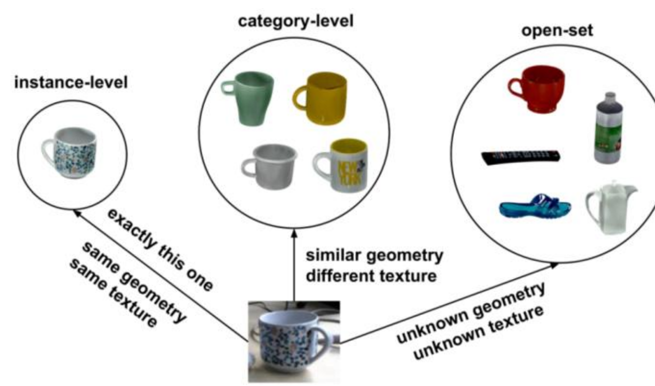
Zero-shot

$$s \cdot \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = K [R \mid t] \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}$$

6DoF parameter estimation



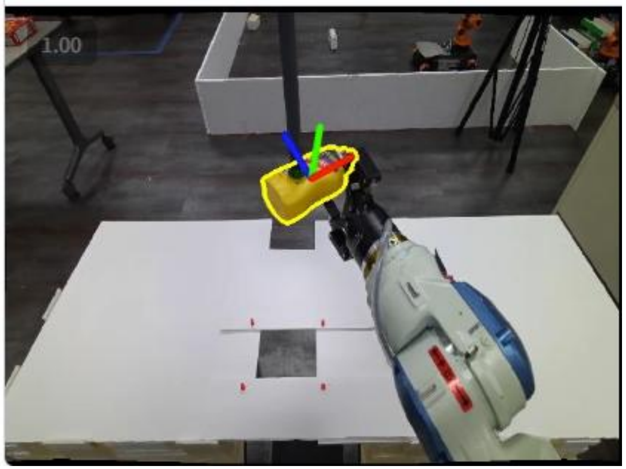
Model-free[1] (Unified)

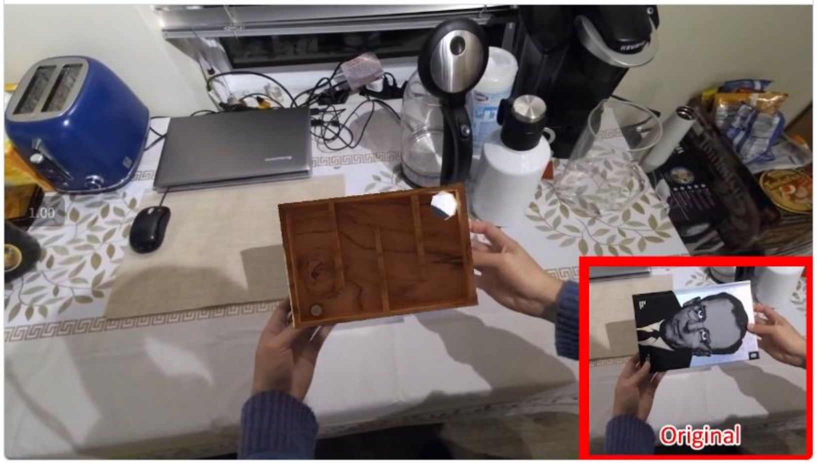


Category-level

Introduction

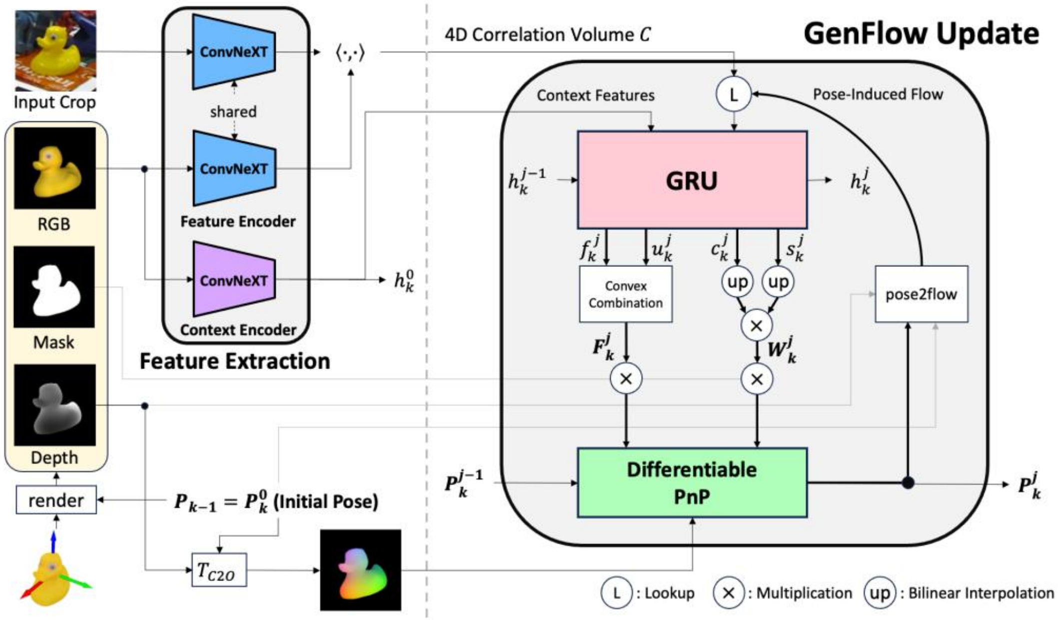
- 6D object pose estimation
  - Application
    - Robot manipulation
      - Object 좌표계 기준으로 잡아야 하는 부분 표현, Camera를 통해 robot 좌표계로 변환
    - AR/VR
      - Physically consistent AR overlay
    - Autonomous Systems
      - Object-level 3D understanding for planning & interaction





Introduction

- Render-and-compare method
  - 2D-2D relative pose estimation 기반
    - Rendered image와 query image를 deep network에 입력하여 residual pose estimation
    - Initial pose update 후 residual pose estimation 및 update 반복

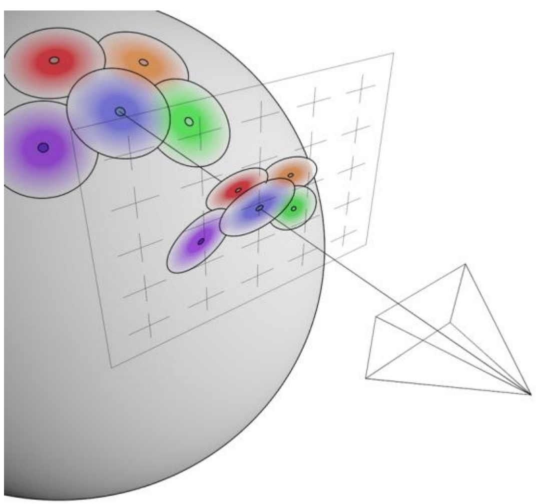


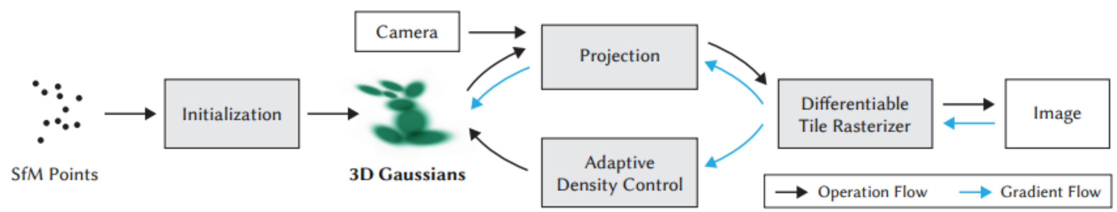
< GenFlow[1] >

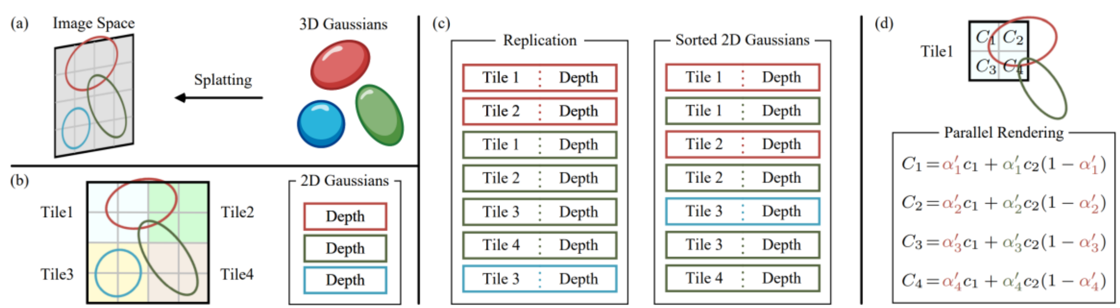
Introduction

- 3D Gaussian Splatting
  - Explicit scene representation by 3D gaussian modeling
    - Mean, Covariance(Scale, Rotation) , Color, Opacity
  - Tile-based rasterization
    - Projection, Depth sorting, Alpha blending

$$F_s = \sum_{i \in \mathcal{N}} f_i \alpha_i T_i, \quad T_i = \prod_{j=1}^{i-1} (1 - \alpha_j)$$

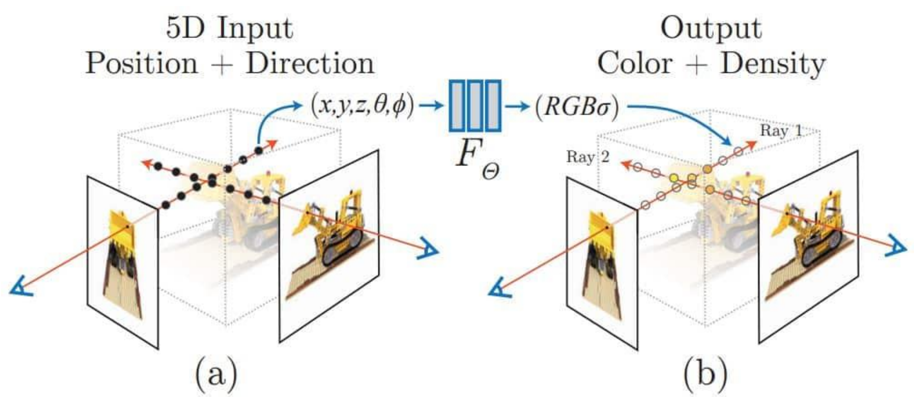


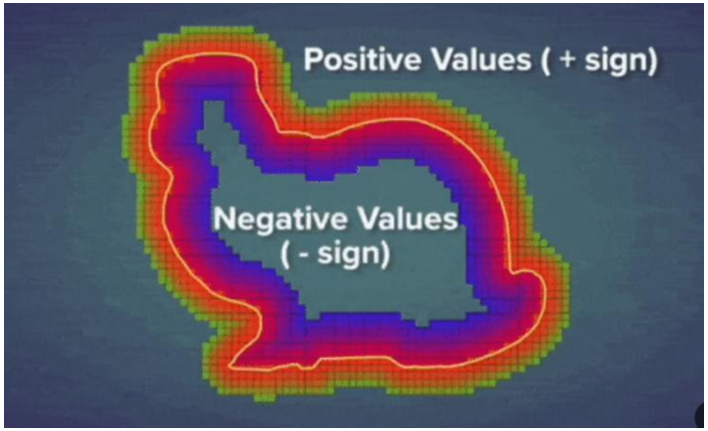




Introduction

- Feature Field
  - Field
    - 공간의 모든 위치 x에 어떤 값 $f(x)$ 을 연속적으로 정의한 함수
      - Radiance field, Temperature field, etc.
  - Feature field
    - 공간의 각 위치에 semantic feature vector를 할당한 field





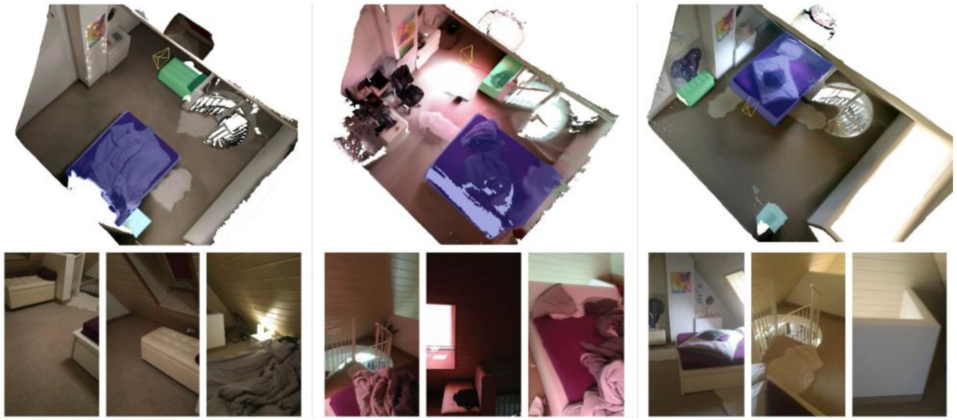
< Neural Radiance Field >

< Surface Distance Field >

- From Sparse to Dense: Camera Relocalization with Scene-Specific Detector from Feature Gaussian Splatting (CVPR 2025)

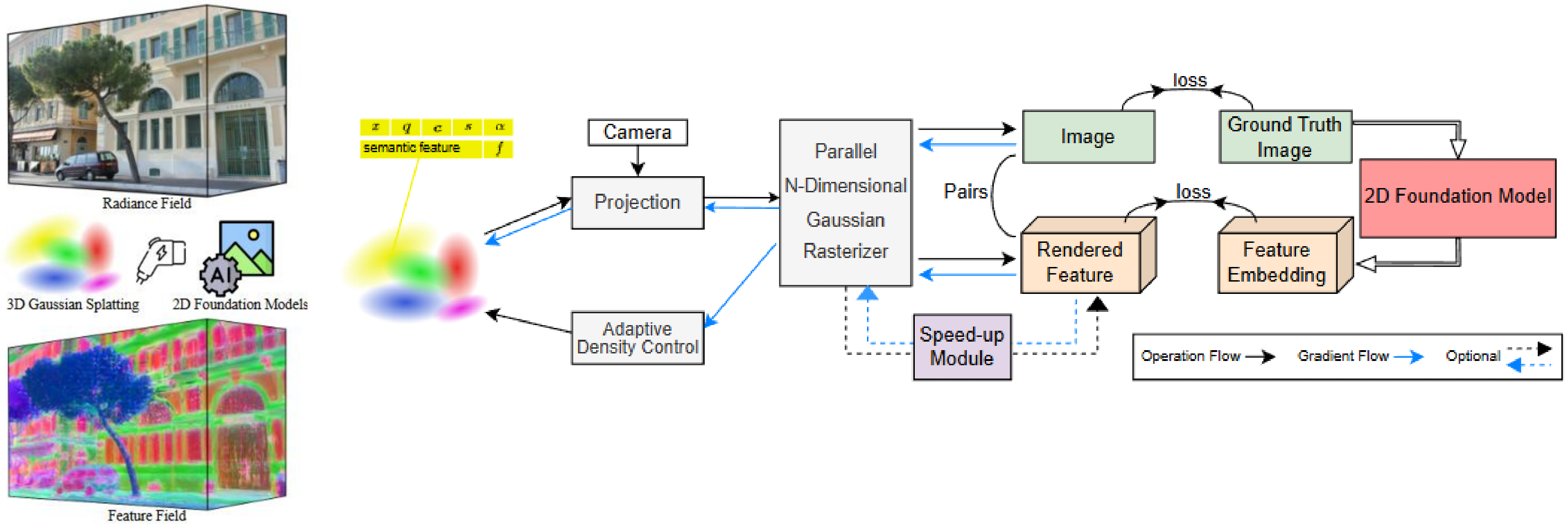
STDLoc

- Preliminaries
  - Camera Re-localization
    - Estimating 6DoF camera pose to pre-built 3D scene map
    - Challenges
      - Appearance change, Viewpoint difference, Perceptual aliasing
    - Application
      - VR, Autonomous Driving, Robots navigation, etc.



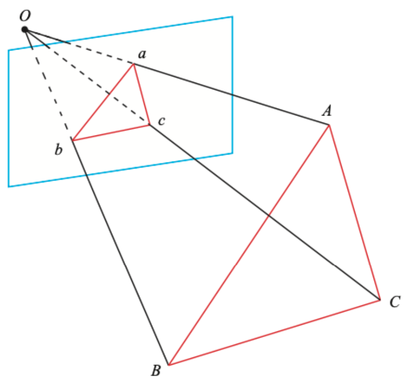
STDLoc

- Preliminaries
  - Feature 3DGS
    - 3D Gaussian에 semantic feature 인자 추가 부여
    - 2D foundation model의 feature를 3D Gaussian으로 distillation
      - Rendered image와 Rendered feature에 대해 joint optimization
    - SAM의 feature를 distillation하여 3D 공간상에서 segmentation 수행



STDLoc

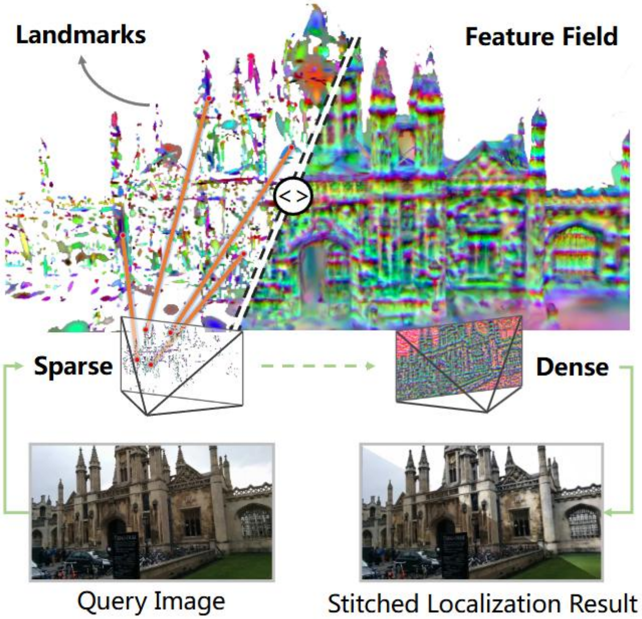
- Introduction
  - Previous approaches
    - 2D-3D correspondence establish → PnP
      - Degrade in weak texture
    - Dense matching with Point Cloud
      - Too much computation
    - Direct regression
      - Target scene scale 변화에 대응 어려움
      - Weak texture의 indoor scene에서 좋은 성능
    - Novel view synthesis
      - Scene representation으로 사용 (NeRF feature, rendering inversion)
      - Data augmentation
      - Iterative optimization (Render and compare)



$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & \gamma & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

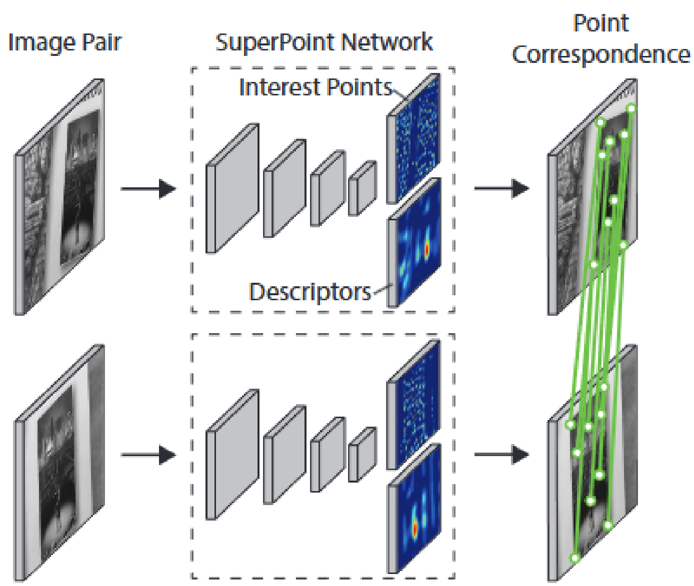
STDLoc

- Introduction
  - End-to-end localization pipeline
    - Initial pose estimation을 위한 별도의 방법 불필요
  - Feature field distillation 활용
    - Radiance + Feature를 포함한 3D scene prior 학습
  - 2D-3D matching 기반
    - Sparse to Dense correspondence matching
  - Scene-specific detector
    - Landmark-aware key point detection
    - Weak-texture 환경에서도 높은 recall

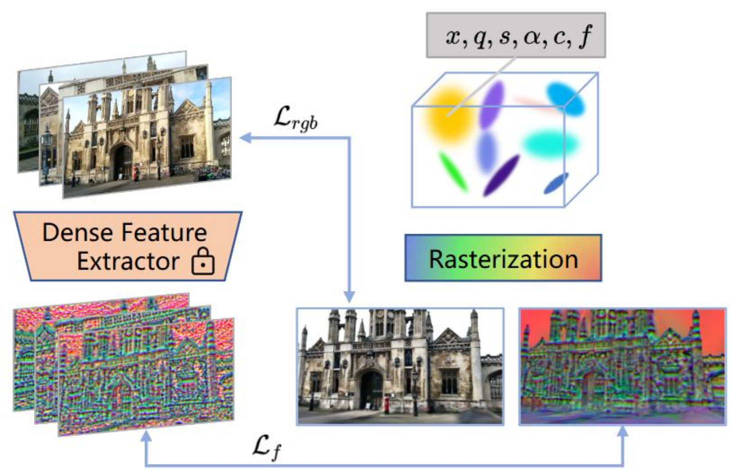


STDLoc

- Feature Gaussian Training
  - Feature 3DGS의 2d foundation model distillation 적용
    - RGB loss + Feature field loss
  - SuperPoint[1)] feature 적용
    - Key point extraction and matching network의 feature distillation하여 2D-3D matching 가능



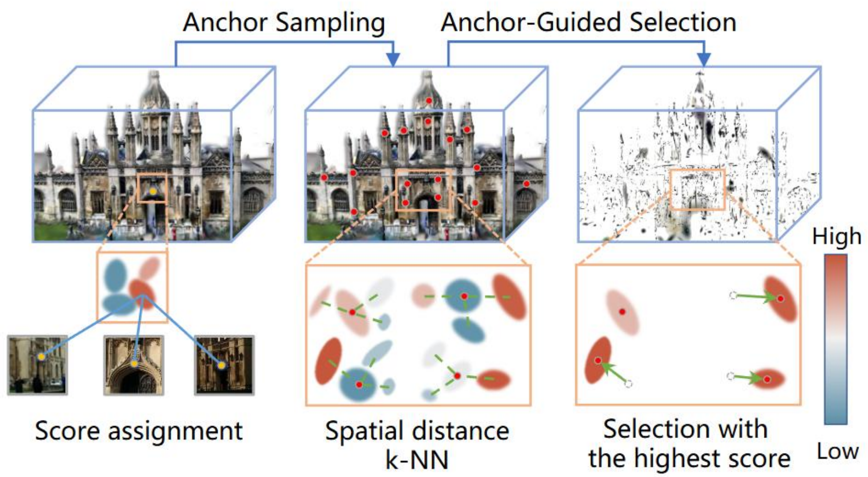
< Superpoint >



< STDLoc optimization >

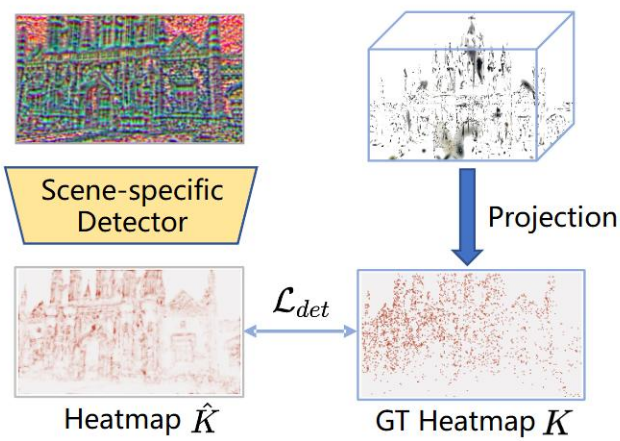
STDLoc

- Matching-Oriented Sampling
  - Scene landmark Gaussian selection
    - Known pose의 training image를 통해 scene별로 선정
    - Gaussian feature와 해당 gaussian center projection 위치의 2D feature 비교
      - 모든 training image에서의 similarity 평균이 score가 됨
    - 전체 coverage를 위해 사전에 지정한 anchor 근처에서 landmark 선정
      - Anchor의 k-nearest neighbor 중 가장 높은 score의 gaussian 선정
      - Rich texture 부분에 landmark가 집중되는 것을 방지



STDLoc

- Scene-Specific Detector
  - 2D feature의 landmark detector 학습
    - “4-layer CNN with SiLU activation”을 통해 확률분포 형태의 heatmap 출력
      - Feature gaussian의 sampled landmark일 확률을 나타냄
    - Sampled gaussian의 projection(center)을 GT map으로 제공하여 self-supervised
    - Binary cross-entropy 적용
    - Inference시에는 estimated heatmap에 Non-Maximum Suppression (NMS) 적용
      - Sparse gaussian과 Spare feature을 비교하여 적은 연산량으로 coarse pose 추정



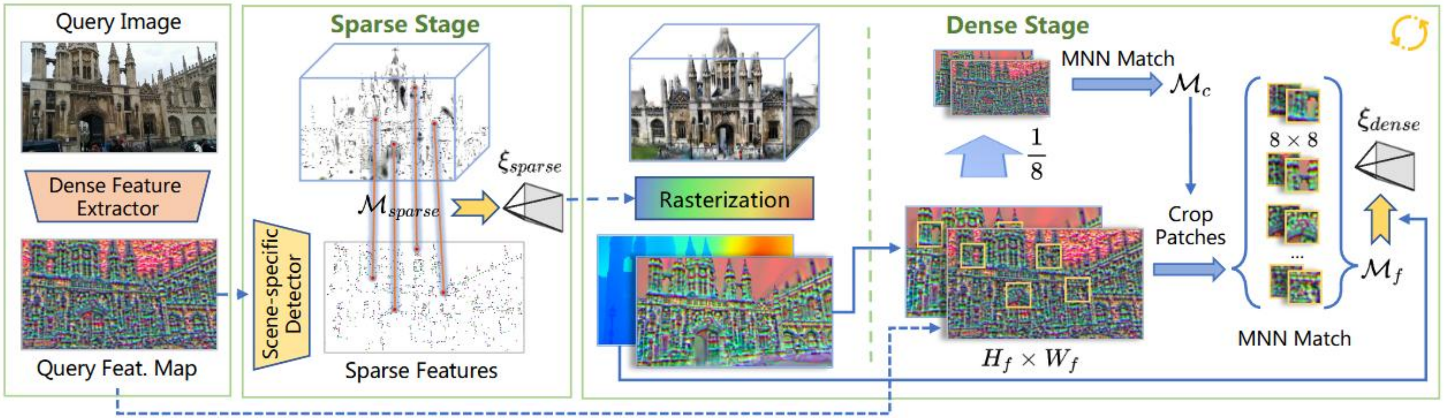
< NMS example >

STDLoc

- Sparse-to-Dense Localization
  - Sparse Stage
    - Sparse gaussian, Sparse feature matching → PnP + RANSAC
  - Dense Stage
    - Sparse stage의 initial pose로 feature map rendering
    - 1/8 down-sampled feature map에서 먼저 coarse matching
      - Correlation matrix → Dual softmax → Mutual Nearest Neighbor (MNN)

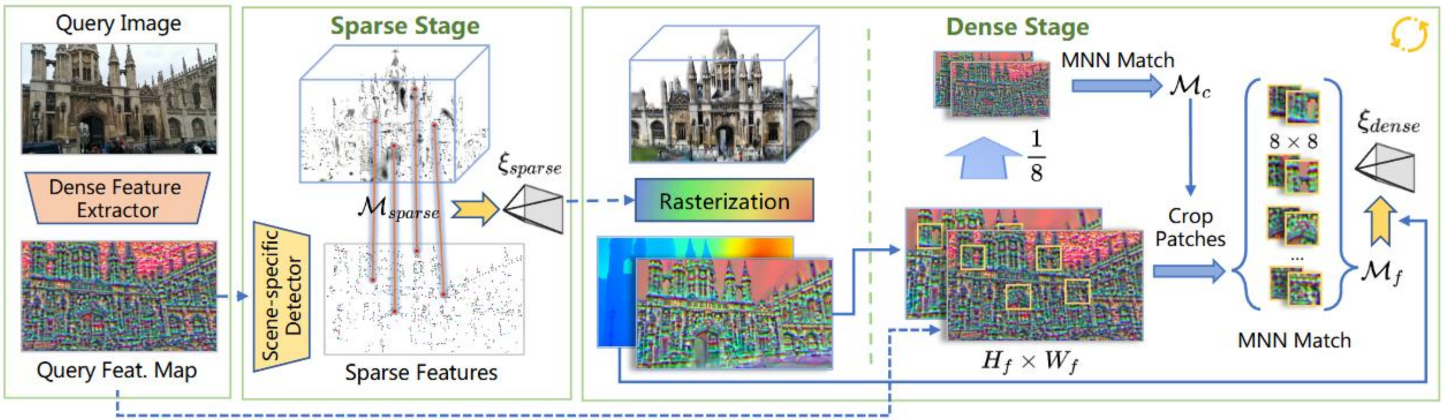
$$\mathcal{P}_c = \text{softmax}(\frac{1}{\tau}\mathcal{S}_c)_{row} \cdot \text{softmax}(\frac{1}{\tau}\mathcal{S}_c)_{col}$$

서로가 서로의 최근접 이웃일 때만 매칭



STDLoc

- Sparse-to-Dense Localization
  - Dense Stage
    - Full resolution feature map에서 coarse matching 결과 refine
    - 주변 8x8 patch에서만 matching 다시 수행하여 최종 2D-2D correspondence 추정
      - Correlation matrix → Dual softmax → Mutual Nearest Neighbor (MNN)
    - Rendered depth map을 통해 2D-3D correspondence로 변환 후 PnP + RANSAC
    - Dense stage를 iterative하게 반복하여 성능 향상



STDLoc

- Experiments
  - Training & Localization Details
    - Loss
      - RGB loss: L1 + SSIM
      - Feature loss: L1
    - Feature rasterization 안정화
      - Gaussian feature render 전후 L2 normalize
    - Training setup
      - 30k iteration, RTX4090 기준 scene별 90min
    - Dense stage iteration
      - ~4 iteration
    - Feature map resolution
      - Max side 640

STDLoc

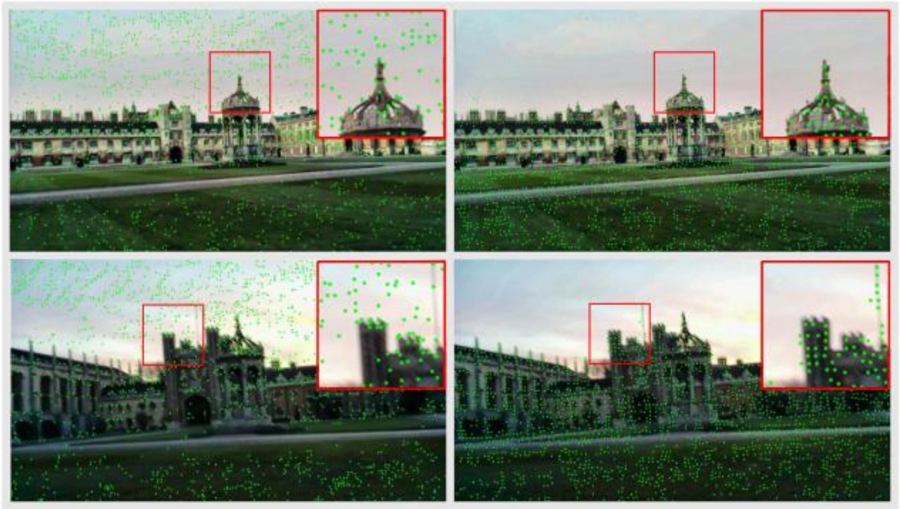
- Experiments
  - Quantitative Results
    - Sparse stage의 initial pose만으로도 높은 recall 달성

Stage	Err.↓[cm/°]	5m 10°↑
Image Retrieval	586/7.9	48.2%
Sparse	13.8/0.21	**99.6%**
Sparse + Dense (RGB)	15.3/0.3	99.4%
Sparse + Dense (Feat.)	**10.1/0.14**	99.4%

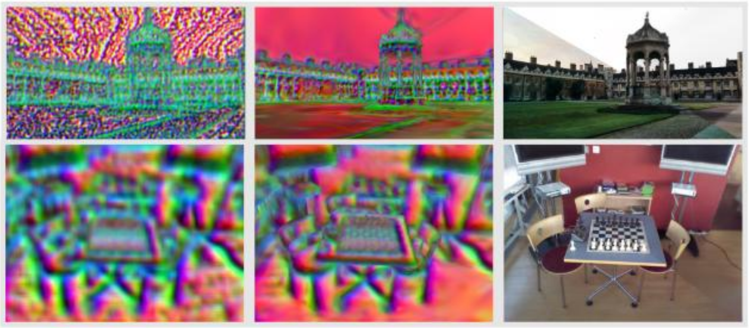
	Method	Chess	Fire	Heads	Office	Pumpkin	RedKitchen	Stairs	Avg.↓[cm/°]
FM	AS (SIFT)	3/0.87	2/1.01	1/0.82	4/1.15	7/1.69	5/1.72	4/1.01	3.71/1.18
FM	HLoc (SP+SG)	2.39/0.84	2.29/0.91	1.13/0.77	3.14/0.92	4.92/1.30	4.22/1.39	5.05/1.41	3.31/1.08
FM	DVLAD+R2D2	2.56/0.88	2.21/0.86	0.98/0.75	3.48/1.00	4.79/1.28	4.21/1.44	4.60/1.27	3.26/1.07
SCR	DSAC*	0.50/0.17	0.78/0.29	0.50/0.34	1.19/0.35	1.19/0.29	0.72/0.21	2.65/0.78	1.07/0.35
SCR	ACE	0.55/0.18	0.83/0.33	0.53/0.33	1.05/0.29	1.06/0.22	0.77/0.21	2.89/0.81	1.10/0.34
SCR	NBE+SLD	0.6/0.18	0.7/0.26	0.6/0.35	1.3/0.33	1.5/0.33	0.8/**0.19**	2.6/0.72	1.16/0.34
SCR	NeuMap	2/0.81	3/1.11	2/1.17	3/0.98	4/1.11	4/1.33	4/1.12	3.14/0.95
NeRF/GS	DFNet+NeFeS$_{50}$	2/0.57	2/0.74	2/1.28	2/0.56	2/0.55	2/0.57	5/1.28	2.43/0.79
NeRF/GS	CROSSFIRE	1/0.4	5/1.9	3/2.3	5/1.6	3/0.8	2/0.8	12/1.9	4.43/1.38
NeRF/GS	PNeRFLoc	2/0.80	2/0.88	1/0.83	3/1.05	6/1.51	5/1.54	32/5.73	7.29/1.76
NeRF/GS	NeRFMatch	0.95/0.30	1.11/0.41	1.34/0.92	3.09/0.87	2.21/0.60	1.03/0.28	9.26/1.74	2.71/0.73
NeRF/GS	**STDLoc (Ours)**	**0.46/0.15**	**0.57/0.24**	**0.45/0.26**	**0.86/0.24**	**0.93/0.21**	**0.63/0.19**	**1.42/0.41**	**0.76/0.24**

STDLoc

- Experiments
  - Qualitative Analysis



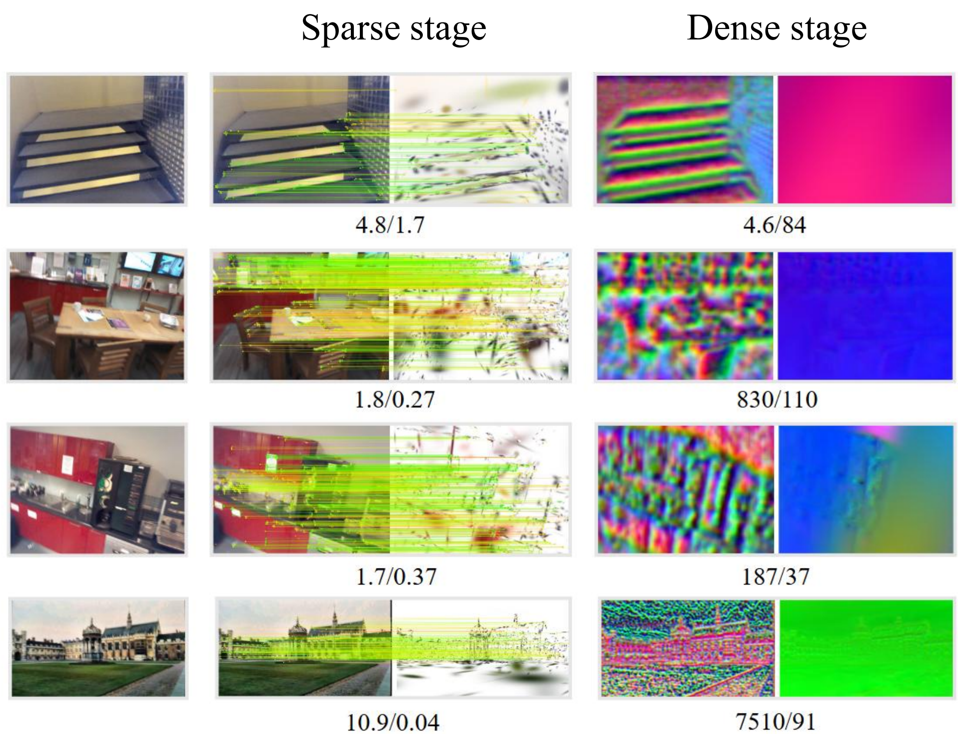
a) SuperPoint Detector.   b) Scene-specific Detector.



(left) query feature map
(middle) rendered feature map
(right) stitched image

STDLoc

- Experiments
  - Qualitative Analysis – failure cases
    - Floater로 인한 artifact로 인해 dense stage에서 오히려 성능이 크게 떨어지는 경우

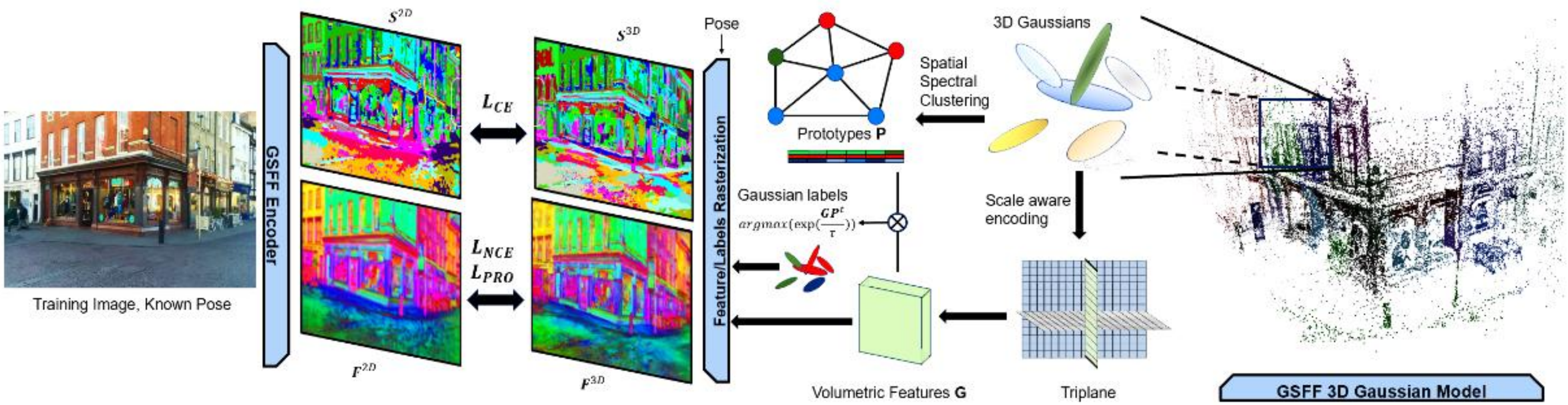


< translation error(cm) / rotation error(°) >

- Gaussian Splatting Feature Fields for (Privacy-Preserving) Visual Localization (CVPR 2025)

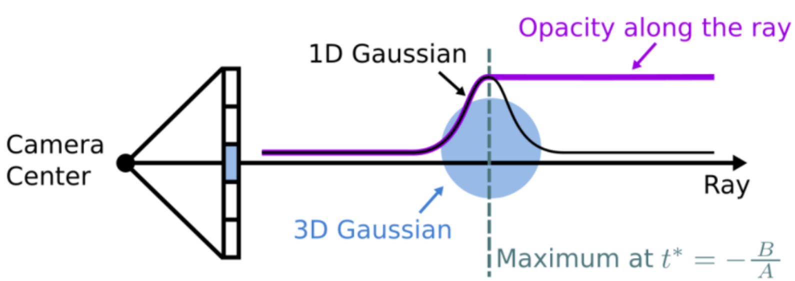
GSFFs

- Introduction
  - Gaussian Feature Fields
    - 논문에서 제안하는 새로운 feature field
    - Geometry structure 반영 & Pose 변화에 강건한 표현을 학습하도록 유도
  - Joint optimization of image encoder and feature field
  - Privacy preserving localization 고려



GSFFs

- Scene Representation
  - Gaussian Opacity Fields[1)] (GOF)
    - 렌더링 품질에 더해 surface 추정과 3D 지점 querying이 가능하도록 GS 학습
    - Ray Tracing based Volume rendering
      - Ray-gaussian intersection & Ray에 대한 Gaussian의 contribution 평가
      - Contribution, view-dependent color를 alpha blending
        - ✓View-dependent color는 spherical harmonics coefficient를 학습을 통해 구현



$$c_k = \sum_{l=0}^{L} \sum_{m=-l}^{l} C_{l,m,k} Y_l^m(v)$$

contribution $\mathcal{E}(\mathcal{G}_k, \mathbf{o}, \mathbf{r}) = \mathcal{G}_k^{1D}(t^*)$

intersection

$$\mathbf{c}(\mathbf{o}, \mathbf{r}) = \sum_{k=1}^{K} \mathbf{c}_k \, \alpha_k \, \mathcal{E}(\mathcal{G}_k, \mathbf{o}, \mathbf{r}) \prod_{j=1}^{k-1} (1 - \alpha_j \, \mathcal{E}(\mathcal{G}_j, \mathbf{o}, \mathbf{r}))$$

GSFFs

- Scene Representation
  - Gaussian Opacity Fields (GOF)
    - Depth distortion regularization
      - ray에서 Gaussian contribution의 깊이 분산에 penalty 부여
      - 같은 ray에 기여하는 Gaussian이라면 서로 비슷한 depth를 가지도록 제약

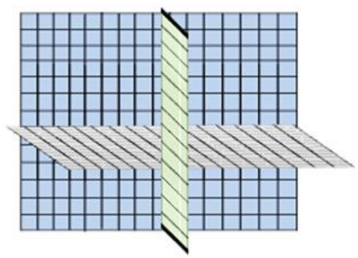
$$\mathcal{L}_d = \sum_{i,j} \omega_i \omega_j |t_i - t_j|$$

    - Normal consistency regularization
      - Gaussian의 normal과 Rendered depth를 통해 얻은 normal이 같아지도록 제약
        - ✓ray-Gaussian intersection에서의 접평면의 normal을 Gaussian normal로 정의
      - intersection 평면이 실제 표면 접평면과 평행해지도록 학습이 유도됨
        - ✓모든 ray에서 gaussian이 주장하는 normal은 동일하도록
        - ✓Floaters 감소, Surface가 두꺼워지는 현상 감소

$$\mathcal{L}_n = \sum_{i} \omega_i (1 - \mathbf{n}_i^{\top} \mathrm{N})$$

GSFFs

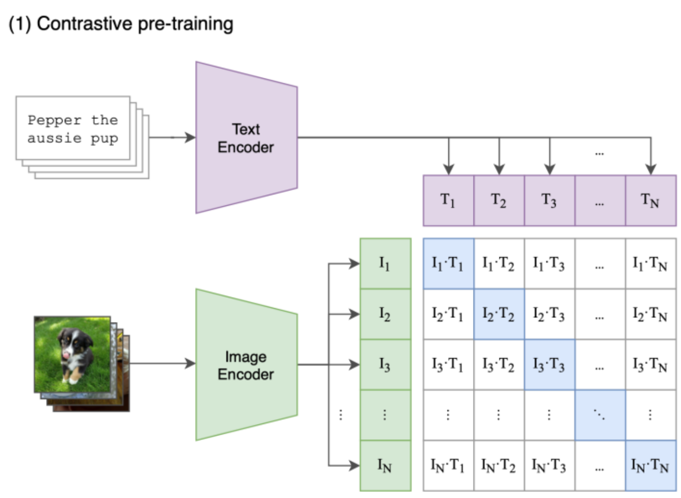
- Scene Representation
  - Scale-aware feature encoding (Volumetric Feature Extraction)
    - Triplane grid feature 할당
      - Parameter 효율성
      - Spatial inductive bias, Scale-aware encoding
    - Radial Basis Function (RBF) kernel
      - 각 3D Gaussian을 triplane에 2D ($K(\mathbf{x},\mathbf{y}) = \exp(-\gamma\|\mathbf{x}-\mathbf{y}\|_2^2)$
        - ✓2D Gaussian을 가중치로 해당 평면의 grid로부터 feature 가중합
      - 세 평면에서 얻은 feature의 평균을 해당 3D Gaussian의 volumetric feature로 사용
    - 3D Gaussian의 물리적 스케일을 feature에 직접 반영
    - 3D Gaussian에 volumetric feature 할당 후 GOF와 동일한 rasterization 과정 수행
      - Color term를 feature로 대체한 후 ray tracing + alpha blending



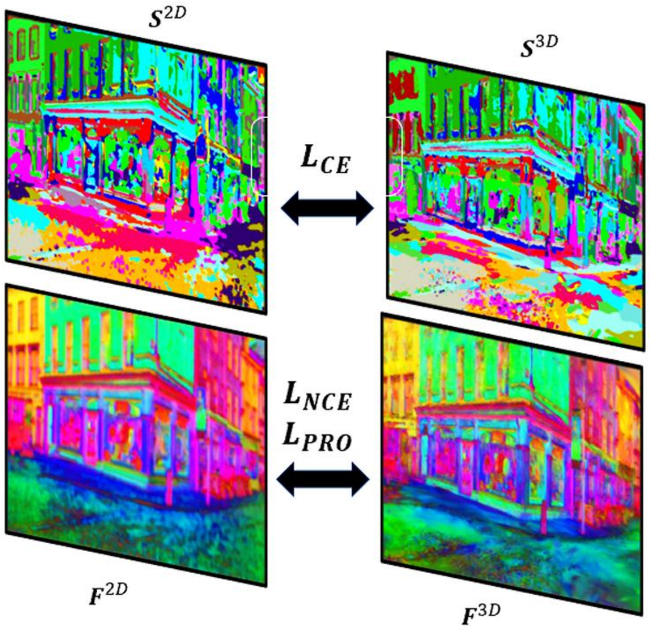
Triplane

GSFFs

- Scene Representation
  - Contrastive loss
    - 2D image encoder feature map과 Rendered feature map 간 contrastive loss 적용
    - Self-supervised manner로 학습 가능



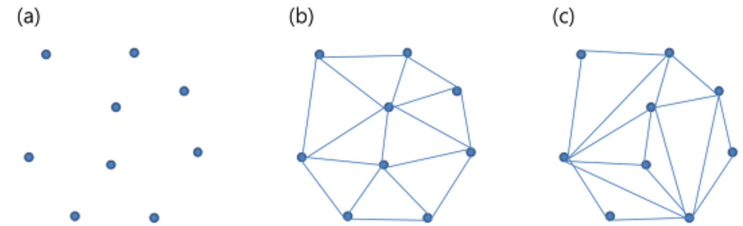
< CLIP contrastive loss >



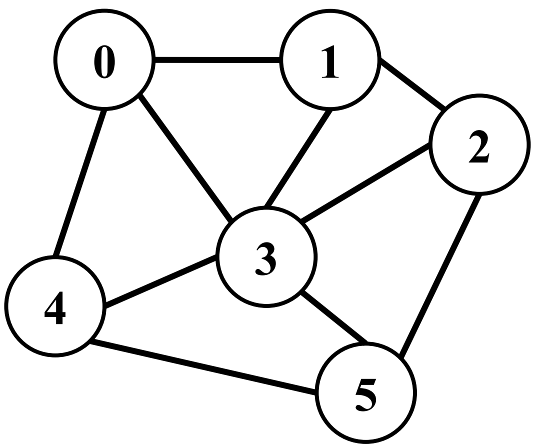
$$L_{NCE} = -\frac{1}{2HW}\sum_{u\in I}\log\left(\frac{\exp\left(\mathbf{F}_u^{3D}\cdot\mathbf{F}_u^{2D}/\tau\right)^2}{A}\right)$$

$$A = \left(\sum_{j=1}^{N}\exp(\mathbf{F}_u^{3D}\cdot\mathbf{F}_j^{2D}/\tau)\right)\left(\sum_{j=1}^{N}\exp(\mathbf{F}_u^{2D}\cdot\mathbf{F}_j^{3D}/\tau)\right)$$

GSFFs



- Prototypical Feature Regularization
  - Spatial prototypes clustering
    - Transform to graph (Delaunay triangulation 들로네 삼각분할)
      - Gaussian center의 삼각분할 후 그래프로 간주함 (center=node, 연결선=edge)
    - Spectral clustering - Laplacian L of the sparse adjacency matrix
      - $L = D - A$ 로 정의되며, 그래프의 전체적인 energy, smoothness 반영



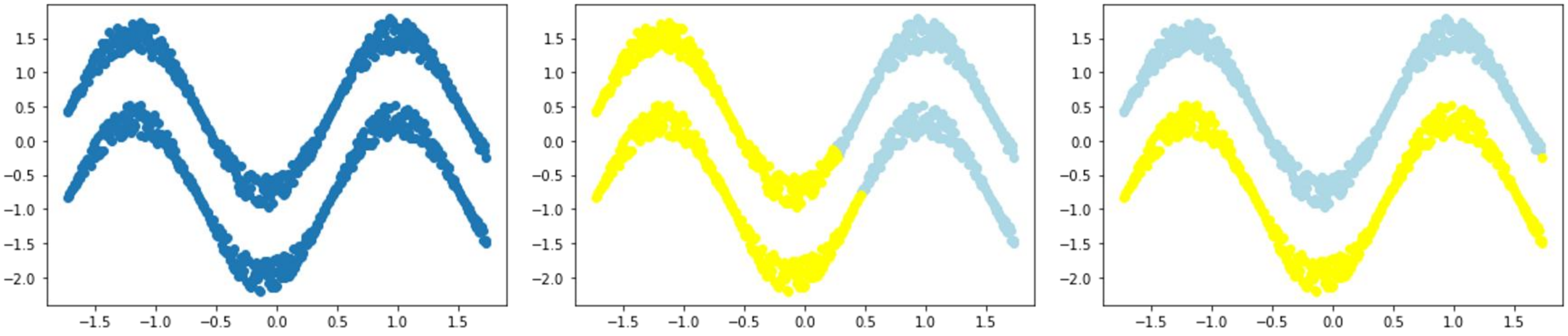
< Graph example >

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix} \quad D = \begin{pmatrix} 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 \end{pmatrix}$$

$$L = \begin{pmatrix} 3 & -1 & 0 & -1 & -1 & 0 \\ -1 & 3 & -1 & -1 & 0 & 0 \\ 0 & -1 & 3 & -1 & 0 & -1 \\ -1 & -1 & -1 & 5 & -1 & -1 \\ -1 & 0 & 0 & -1 & 3 & -1 \\ 0 & 0 & -1 & -1 & -1 & 3 \end{pmatrix}$$

GSFFs

- Prototypical Feature Regularization
  - Spatial prototypes clustering
    - Spectral clustering - Cluster with Eigenvector
      - Laplacian의 eigenvector를 통해 선형 변환 후 k-mean clustering
    - Prototype feature
      - 각 cluster를 대표하는 단일 feature vector
      - Cluster k에 속하는 모든 Gaussian의 volumetric feature 평균



< (left) data (middle) k-means clustering (right) spectral clustering >

GSFFs

- Prototypical Feature Regularization
  - Prototypical loss
    - 2D feature, Rendered Feature pair가 동일한 prototype에 가까워지도록 학습
    - Pixel마다 optimal transport를 통해 어떤 cluster의 prototype할당할지 동적으로 결정
    - Spatial prior를 feature space에 반영하도록 가이드
    - 학습안정화 + 이후 privacy preserving localization을 위해 사용

$$L_{PRO} = -\frac{1}{N}\sum_{n=1}^{N}\log\left(\frac{\exp\left(\mathbf{F}_n^{3D}\mathbf{p}_n^t + \mathbf{F}_n^{2D}\mathbf{p}_n^t\right)/\tau)}{B}\right)$$

  - Multi-view consistency
    - 다양한 시점에서 동일한 3D point를 관찰했다면 일관된 feature가 되도록 guide
    - 랜덤하게 다른 view에서 관찰된 point의 feature로 대체하여 loss 적용

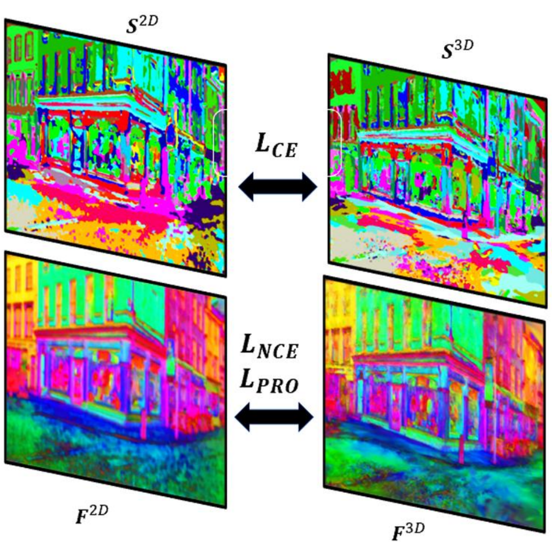
GSFFs

- Feature-based Localization
  - Pose refinement
    - 잘 정의된 Feature field를 통해 간단한 방식으로 pose refinement 가능
    - Initial pose estimation
      - Retrieve closest database image
      - Pose refinement 만을 다루며, pose initialize는 아예 다른 방식으로 수행
    - Backpropagation을 통한 pose update
      - Parameterize 6DoF pose
      - L2 loss 계산 후 pose update

$$P^* = \min_{P \in SE(3)} \|\mathbf{F}^{\mathrm{2D}} - \mathbf{F}^{\mathrm{3D}}(P, \mathcal{G})\|_2^2$$

GSFFs

- Privacy-preserving localization
  - Optimizing segmentation
    - Privacy preserving을 위해 texture 정보를 모두 삭제하고 추론하는 것이 목표
    - Pixel별로 clusters에 대한 logit만을 추정
      - Segmentation map 추정용 head를 병렬로 학습
    - Clustered Gaussian을 통해 GT label 제공하여 CE loss 적용
    - 추론 시 2D segmentation map과 3D label을 통해 backpropagation



$$L_{CE} = -\sum_{u \in I} \mathbb{1}_u \cdot \left(\log(\mathbf{S}_u^{2D}) + \log(\mathbf{S}_u^{3D}) \right)$$

GSFFs

- Experiments
  - Results

	Model	Chess	Fire	Heads	Office	Pumpkin	Redkitchen	Stairs
SBM	HLoc [60]	0.8/0.11/100	0.9/0.24/99	0.6/0.25/100	1.2/0.20/100	1.4/0.15/100	1.1/0.14/99	2.9/0.80/72
	DSAC*[4]	0.5/0.17/100	0.8/0.28/99	0.5/0.34/100	1.2/0.34/98	1.2/0.28/99	0.7/0.21/97	2.7/0.78/92
	ACE [6]	0.5/0.18	0.8/0.33	0.5/0.33	1.0/0.29	1/0.22	0.8/0.2	2.9/0.81
	ACE + GS-CPR [41]	0.5/0.15	0.6/0.25	0.4/0.28	0.9/0.26	1.0/0.23	0.7/0.17	1.4/0.42
RBM	NeFeS [14] (DFNet [13])	2/0.79	2/0.78	2/1.36	2/0.60	2/0.63	2/0.62	5/1.31
	MCLoc [74]	2/0.8	3/1.4	3/1.3	4/1.3	5/1.6	6/1.6	6/2.0
	SSL-Nif [56] (DV)	1/0.22/93	0.8/0.28/91	0.8/0.49/71	1.7/0.41/81	1.5/0.34/86	2.1/0.41/75	6.5/0.63/49
	NeRFMatch [99] (DV)	0.9/0.3	1.3/0.4	1.6/1.0	3.3/0.7	3.2/0.6	1.3/0.3	7.2/1.3
	GSplatLoc [68] (GSplatLoc)	**0.43/0.16**	1.03/0.32	1.06/0.62	1.85/0.4	1.8/0.35	2.71/0.55	8.83/2.34
	GS-CPR [41] (DFNet)	0.7/0.20	0.9/0.32	0.6/**0.36**	1.2/0.32	**1.3/0.31**	0.9/0.25	**2.2/0.61**
	GSFFs-PR Feature (DV)	**0.4**/0.19/95	**0.6/0.26**/98	**0.5/0.36**/94	**1.0/0.31**/97	**1.3**/0.38/85	**0.6/0.23/93**	25.1/0.63/32
	GSFFs-PR Privacy (DV)	0.8/0.28/96	0.8/0.33/94	1.0/0.67/90	1.5/0.51/90	2.0/0.50/78	1.2/0.33/85	28.2/0.98/29

	KC (F)	OH (F)	KC (P)	OH (P)
Coarse only	37.5/0.80	670/1.10	92.2/1.55	82.5/1.50
Fine only	22.6/0.32	55.1/0.77	28.0/0.45	151/0.96
k-means	25.2/0.33	27.6/0.50	37.2/0.64	52.3/0.84
No SOpt	23.5/0.30	23.4/0.45	31.0/0.45	30.8/0.57
No MVR	20.0/0.28	22.5/0.49	29.6/0.51	29.1/0.49
No SE	23.5/0.31	21.9/0.42	27.0/0.42	26.0/0.47
GSFFs-PR	17.9/0.27	21.4/0.41	24.3/0.39	25.6/0.49

Triplane resolution=256

Triplane resolution=1024

No spectral segmentation

Segmentation Optimization

Multi-View Regularization

Scale-aware feature Encoding

Conclusion

- STDLoc
  - Feature field의 pose estimation로의 기본적인 활용
  - Dynamic scene handling 어려움
- GSFFs
  - Reliance on Initial Pose Estimate
  - Dynamic scene handling 어려움
- Transfer to 6d object pose estimation
  - Weak texture / Occlusion
  - Relatively small 3d information