2026 동계 세미나

Zero-shot 6D object pose estimation

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

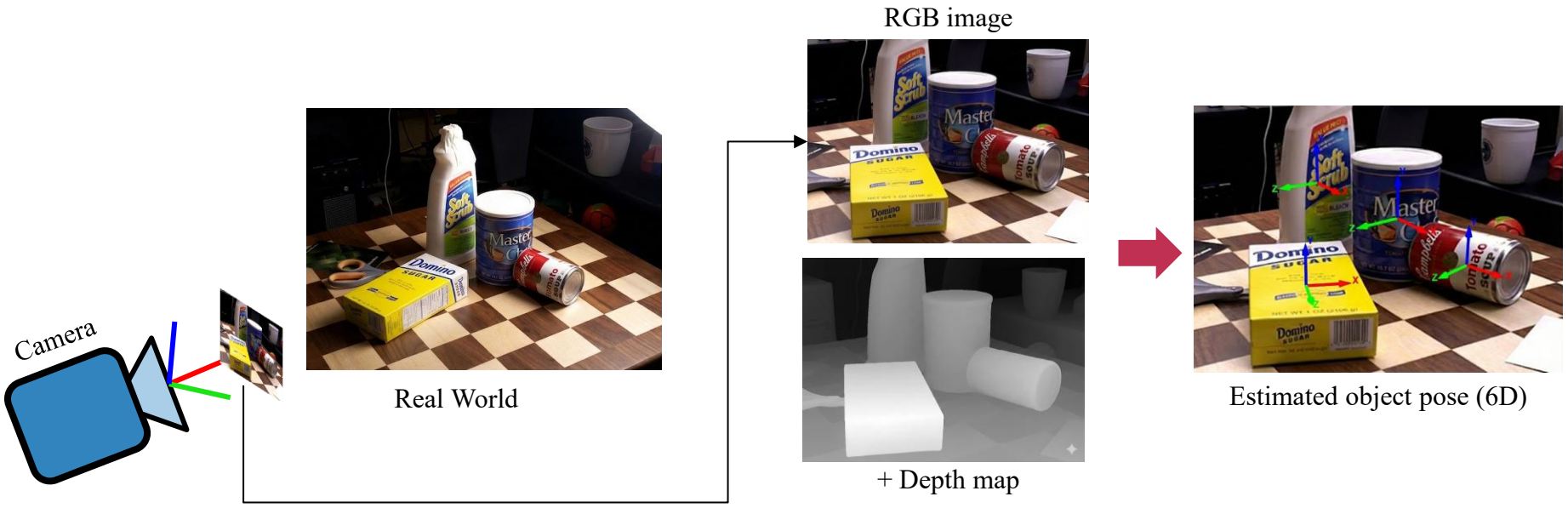
Presented By
김현빈

Contents

- Introduction
  - 6D pose estimation
- Paper Review
  - ZeroPose: CAD-Prompted Zero-shot Object 6D Pose Estimation in Cluttered Scenes (TCSVT 2024)
  - FreeZe: Training-free zero-shot 6D pose estimation with geometric and vision foundation models (ECCV 2024)

Introduction

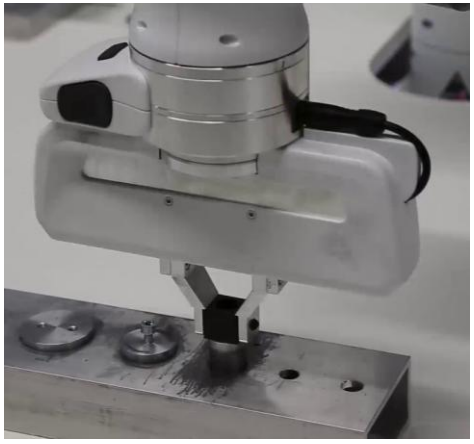
- 6D object pose estimation
  - **카메라 좌표계에서** 물체의 6D pose를 추정하는 문제
    - **6D pose**: 3D translation (x, y, z) + 3D rotation (roll, pitch, yaw)
    - **Input**: RGB image or RGB-D image
    - **Output**: Object pose (Rotation, translation) in **camera coordinate system**



https://www.youtube.com/watch?v=ojpJIXIiTYE
https://zju3dv.github.io/onepose_plus_plus/

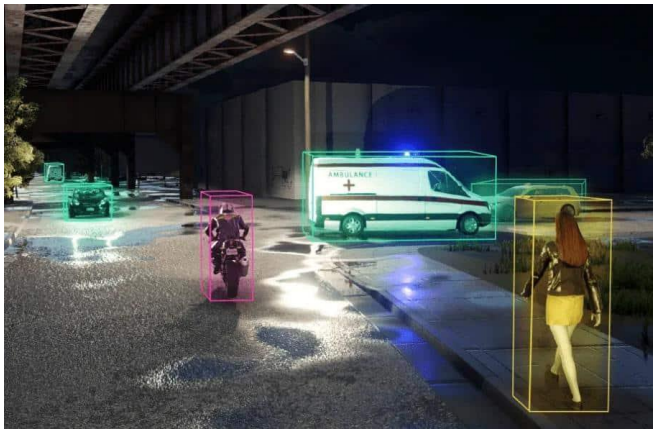
Introduction

- 6D pose in Real-World Application
  - Robotic Manipulation
    - Grasping, pick and place
    - Peg in hole, assembly
  - AR / VR
  - Autonomous driving



Estimated Object Poses

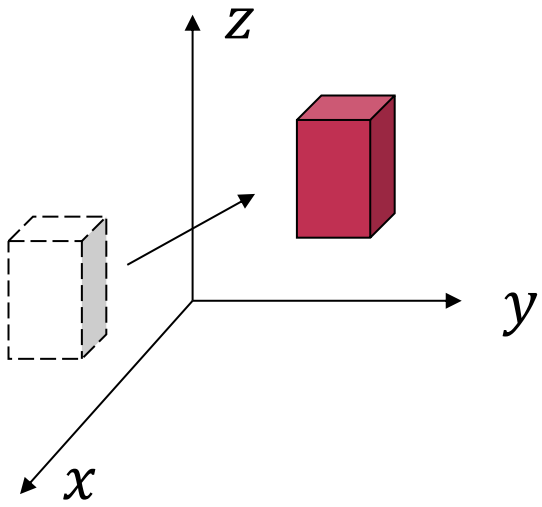
AR Demo



Introduction

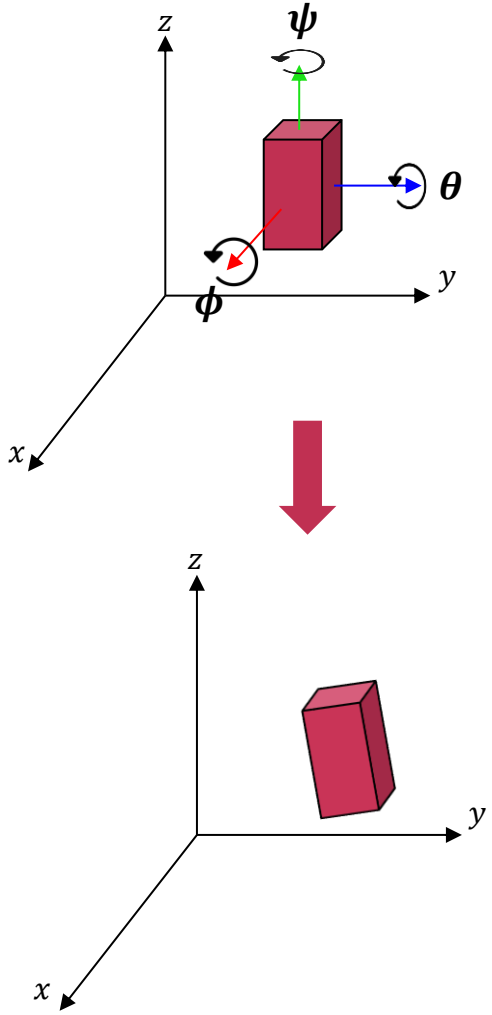
- 6D pose estimation
  - 3D translation
    - 고정된 축(x,y,z)을 기준으로 object가 얼만큼 이동하는지를 표현
      - 6D pose estimation에서는 camera 좌표계의 x,y,z 축이 기준
      - $x' = x + t_x,\ y' = y + t_y,\ z' = z + t_z$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$



Introduction

- 6D pose estimation
  - 3D rotation
    - 고정된 축(x,y,z)을 기준으로 얼만큼 회전하는지를 표현
      - 6D pose estimation에서는 camera 좌표계의 x,y,z 축이 기준
        - ✓ $R = R_z(\psi) \cdot R_y(\theta) \cdot R_x(\phi)$
      - Roll(R_x) : Rotation about fixed x-axis ϕ
        - ✓ $R_x \cdot P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi & 0 \\ 0 & \sin\phi & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$
      - Pitch(R_y): Rotation about fixed y−axis θ
        - ✓ $R_y \cdot P = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$
      - Yaw(R_z): Rotation about fixed z−axis ψ
        - ✓ $R_z \cdot P = \begin{bmatrix} \cos\psi & -\sin\psi & 0 & 0 \\ \sin\psi & \cos\psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$



Introduction

- 6D pose estimation
  - What is 6D pose estimation?
    - 객체 좌표계를 카메라 좌표계로 변환하는 **카메라 extrinsic**을 추정하는 문제
      - 회전(Rotation)과 이동(Translation) 으로 구성
        - ✓$\mathbf{p}_c = \mathbf{R}\mathbf{p}_o + \mathbf{t}$
          - $\mathbf{p}_o$:객체 좌표계(Object Coordinate System)의 점
          - $\mathbf{p}_c$:카메라 좌표계(Camera Coordinate System)의 점
      - 보통 Translation과 rotation은 따로 쓰이지 않고, 하나의 변환 행렬로 묶어서 사용

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

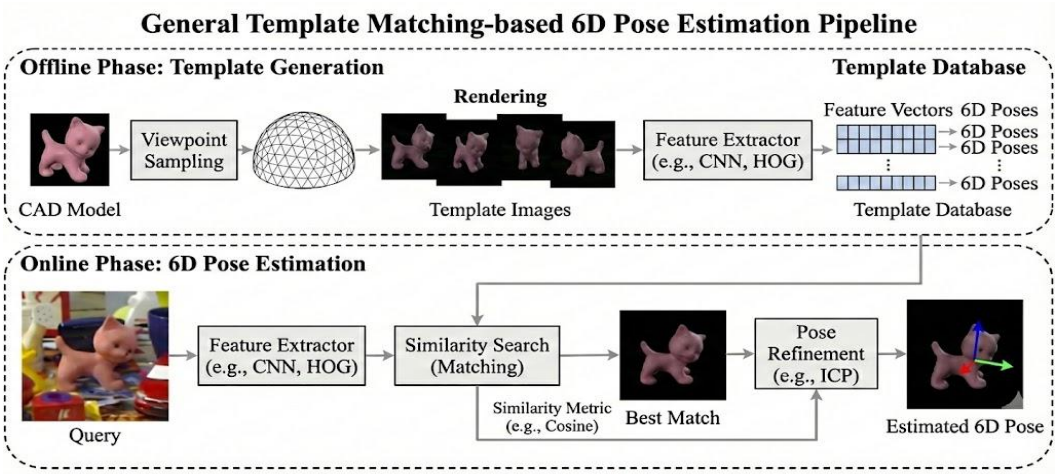
- 6D pose estimation의 목적은 객체 모델과 실제 관측 장면을 가장 잘 일치시키는 $(\mathbf{R}, \mathbf{t})$를 찾는 것

Pipeline of 6D pose estimation

- 6D Pose Estimation에 필요한 정보 (Inputs & Prior Knowledge)
  - Input
    - 센서입력 : RGB or RGB-D image (RGB + depth map)
  - Given / Prior Knowledge
    - 객체 모델 정보 (Model)
      - 3D model: CAD, mesh, point cloud
      - 중간 표현: Template, keypoint, NOCS
    - Camera parameters (일반적으로 사전에 캘리브레이션하여 획득)
      - Intrinsic K: Projection/PnP에 필요
  - 방법 분류에 따른 차이
    - Model-based(with CAD)
      - template/feature matching → PnP/registration → ICP
    - Category-level / Model-free (=CAD-free)
      - CAD 없이도 가능하지만, 보통 다른 중간표현(NOCS, keypoint, canonical space)을 사용

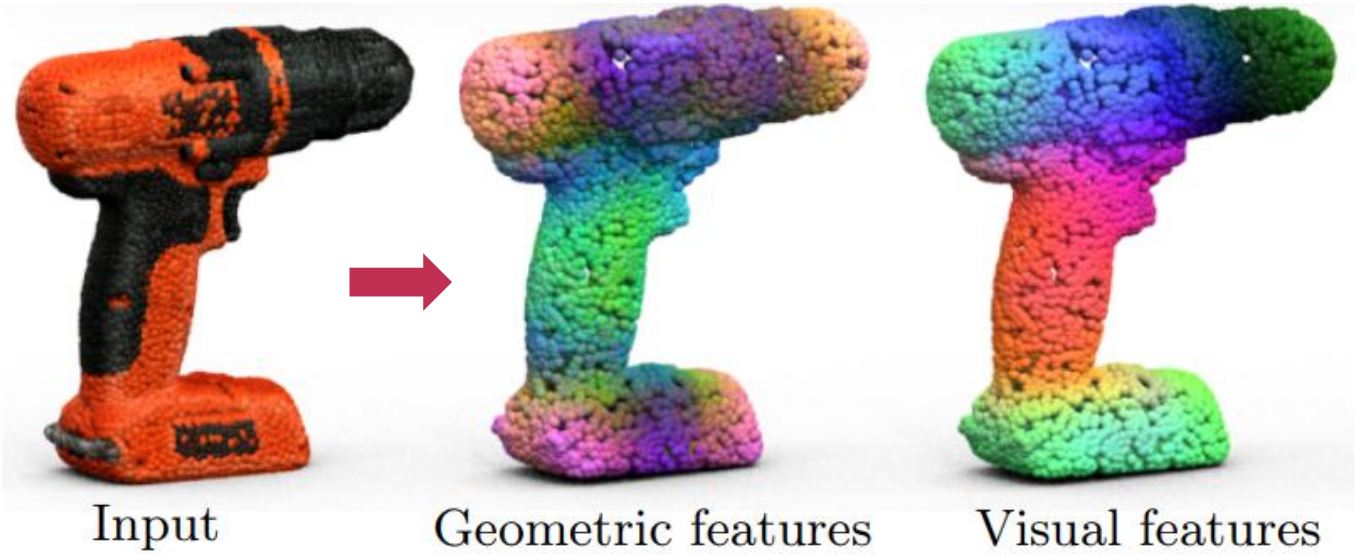
Pipeline of 6D pose estimation

- Template matching
  - 미리 다양한 pose에서 생성한 템플릿 중 입력과 가장 유사한 템플릿의 pose를 선택
  - Offline phase: Template 생성
    - CAD 모델을 여러 시점에서 렌더링하여 템플릿 이미지를 생성
    - 각 템플릿에서 feature를 추출하고, 해당 템플릿에 대응의 6D pose와 함께 DB에 저장
  - Online phase: Pose 추정
    - 입력 이미지에서 feature를 추출한 뒤, DB의 템플릿과 유사도 검색을 수행
    - 가장 유사한 템플릿의 pose를 초기 pose로 사용
    - RGB-D가 있으면 ICP로 refinement하여 최종 pose를 정밀화



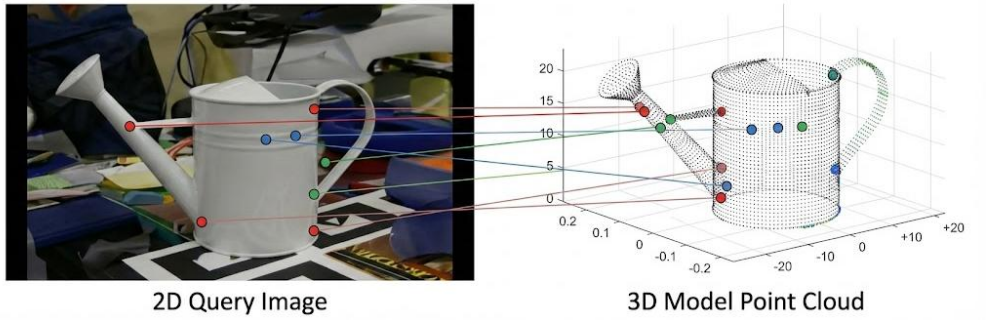
Pipeline of 6D pose estimation

- Feature matching
  - 입력에서 feature를 추출하고, 3D 모델과의 correspondence를 추정한 뒤 (R,t)를 계산
    - Feature 추출
      - Visual feature
        - ✓Image로부터 색,질감,엣지 등 외관 정보를 표현
      - Geometric feature
        - ✓ Depth map 혹은 point cloud로부터 점 주변의 형상 정보를 표현

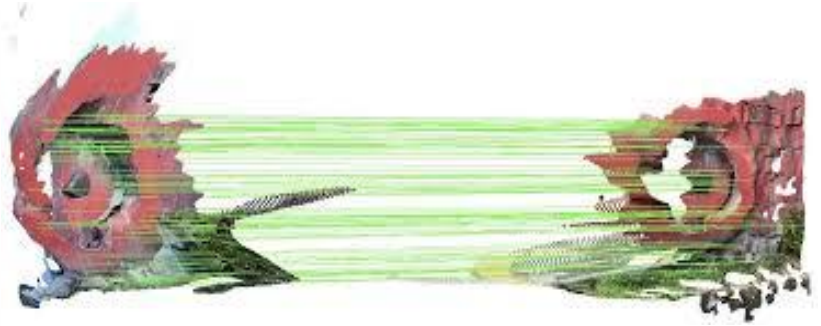


Pipeline of 6D pose estimation

- Feature matching
  - 입력에서 feature를 추출하고, 3D 모델과의 correspondence를 추정한 뒤 (R,t)를 계산
    - Correspondence 추정 (대응점 추정)
      - 추출된 feature 유사도를 비교하여 대응점을 생성
      - Outlier을 줄이기 위해 RANSAC 등의 필터링 방법을 사용
    - Pose 계산
      - 2D–3D 혹은 3D–3D 대응점으로부터 (R,t)를 기하학적으로 계산
        - ✓PnP solve: 2D–3D → (R,t)
        - ✓3D rigid registration solve: (3D–3D → (R,t))



2D-3D correspondence



3D-3D correspondence

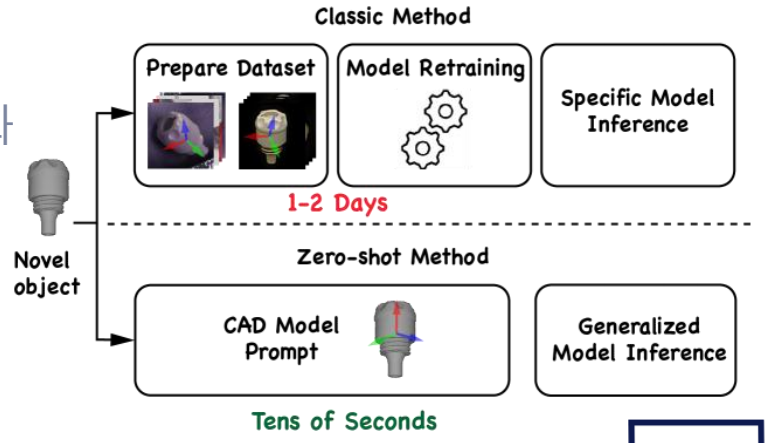
Paper Review

ZeroPose: CAD-Prompted Zero-shot Object 6D Pose Estimation in Cluttered Scenes (TCSVT 2024)

Jianqiu Chen, Zikun Zhou, Mingshan Sun, Tianpeng Bao, Rui Zhao, Liwei Wu, Zhenyu He

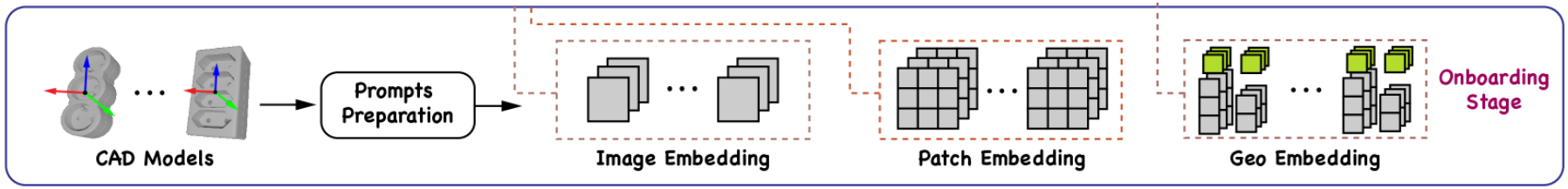
Introduction

- Motivation
  - Why zero-shot 6D pose?
    - 새 객체가 추가될 때마다 데이터 수집/라벨링 + 재학습이 필요 → 현장 적용 비용이 큼
  - Limitations of prior Zero-shot 6D-pose estimation
    - Render-and-compare
      - 다양한 pose로 CAD를 온라인 렌더링하고 입력 이미지와 비교
        - ✓후보 수가 많아질수록 연산량이 커지고 속도 느려짐
      - 일부 방법은 ROI(박스/마스크) 입력을 가정 → 완전 자동화 어려움
  - ZeroPose
    - SAM + CAD 매칭
      - 객체 후보의 위치 추정 및 마스크 생성 자동화
    - 3D feature matching 기반 pose estimation
      - 렌더링 의존도를 낮춰 효율 개선



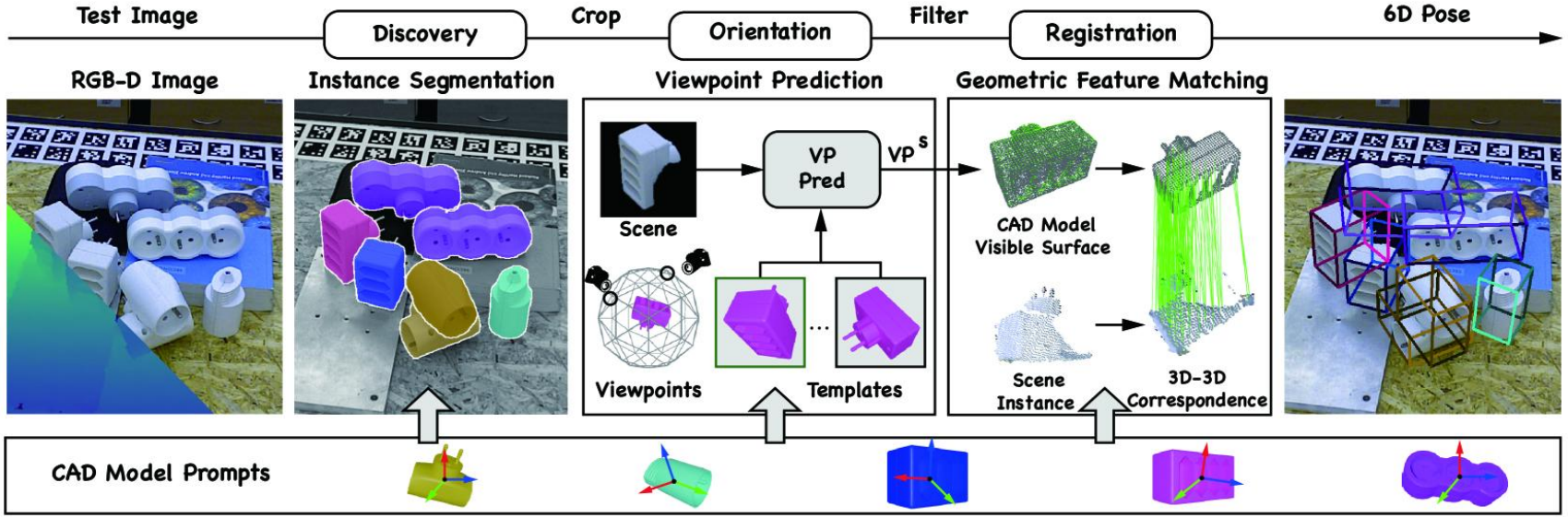
Method

- Onboarding Stage
  - 새 객체(CAD)가 추가될 때, 추론에 필요한 prompt를 미리 구축하는 단계
    - Visual prompt preparation (with ImageBind)
      - CAD 모델을 여러 시점에서 렌더링한 후, image, patch feature를 추출하여 저장
      - 이후 SAM proposal과 유사도 비교를 통해 instance/orientation 후보 선정에 사용
    - Geometric prompt preparation (with GeoTranformer)
      - CAD에서 point cloud을 샘플링한 후, 각 3D 점에 대해 geometric feature를 추출해 저장
      - 이후 3D–3D correspondence matching 에 사용



Method

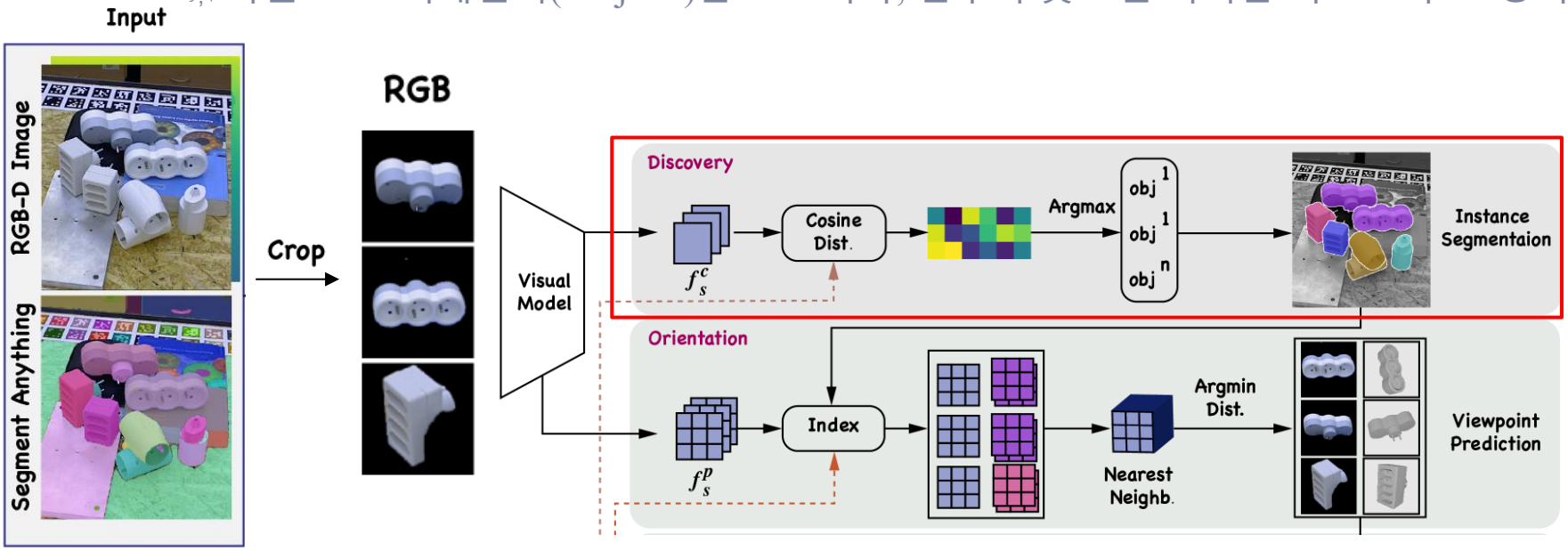
- Overview



- Discovery
  - 객체 후보 마스크를 생성하고, 각 후보가 어떤 객체인지 결정
- Orientation
  - 입력 ROI와 CAD 템플릿을 비교하여 가장 가까운 viewpoint 선택
- Registration
  - 3D feature matching으로 correspondence를 추정한 후, rotation, translation을 계산

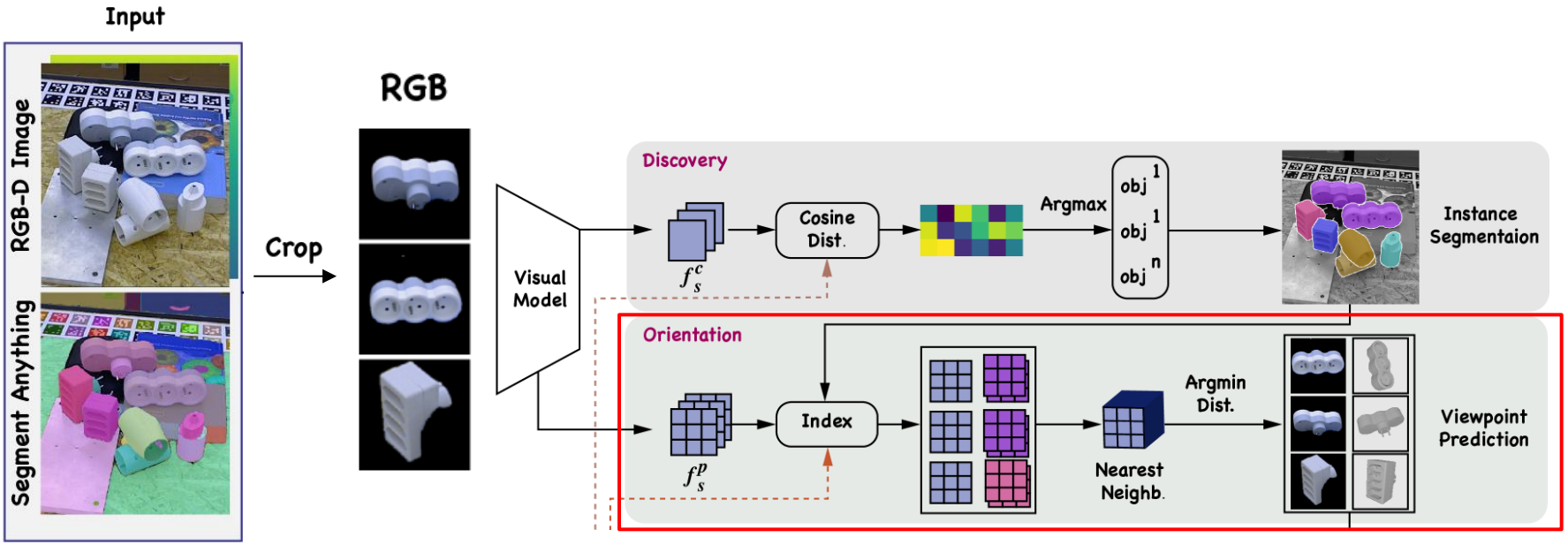
Method

- Discovery (instance segmentation & object identification)
  - 장면에서 객체 후보를 찾고, 어떤 객체인지 식별하는 단계
    - SAM으로 객체 후보 마스크 생성
    - 각 마스크를 crop해서 ImageBind로 instance visual feature 추출
    - 마스크별로 얻은 feature와 CAD 템플릿들의 image feature들을 cosine similarity로 비교
      - 어떤 CAD 객체인지("obj id")를 고르거나, 점수가 낮으면 버리는 식으로 후보 정리



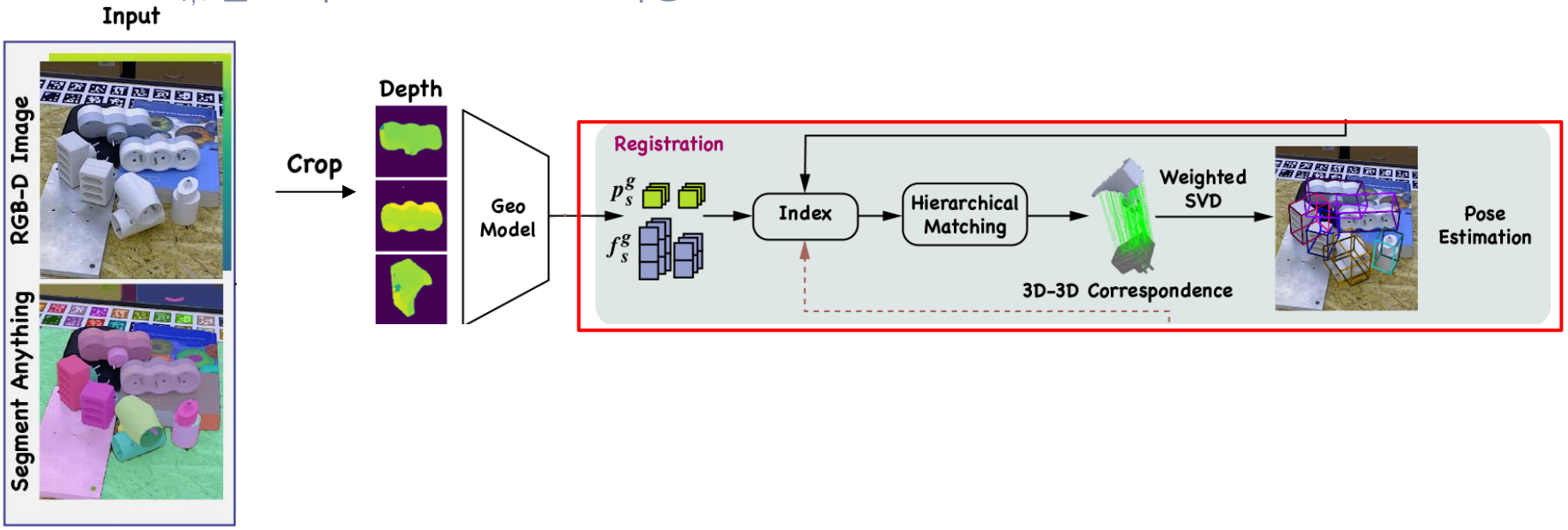
Method

- Orientation (coarse viewpoint initialization)
  - 선택된 ROI와 CAD 템플릿을 비교해 initial viewpoint를 결정
    - Discovery로 객체 ID를 결정한 후, 해당 객체의 viewpoint별 patch feature를 로드
    - ROI의 patch feature와 각 viewpoint 템플릿의 patch feature간 cosine similarity를 계산
    - 유사도가 최대인 viewpoint를 선택 → initial viewpoint로 사용



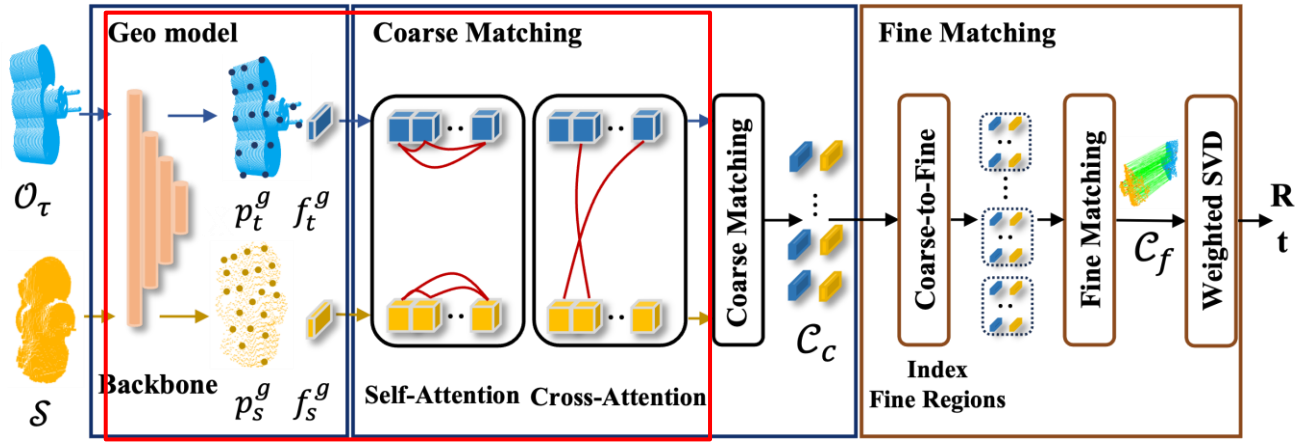
Method

- Registration (3D–3D matching → solve pose)
  - Point cloud 간의 3D–3D correspondence를 추정하고, 최종 (R,t)를 계산
    - Depth 기반 scene point cloud와 CAD(model) point cloud에서 geometric feature를 추출
    - Hierarchical matching으로 3D–3D 대응점 추출
    - (Weighted) SVD rigid registration을 수행해 6D pose 계산
      - 필요 시 ICP로 refinement 가능

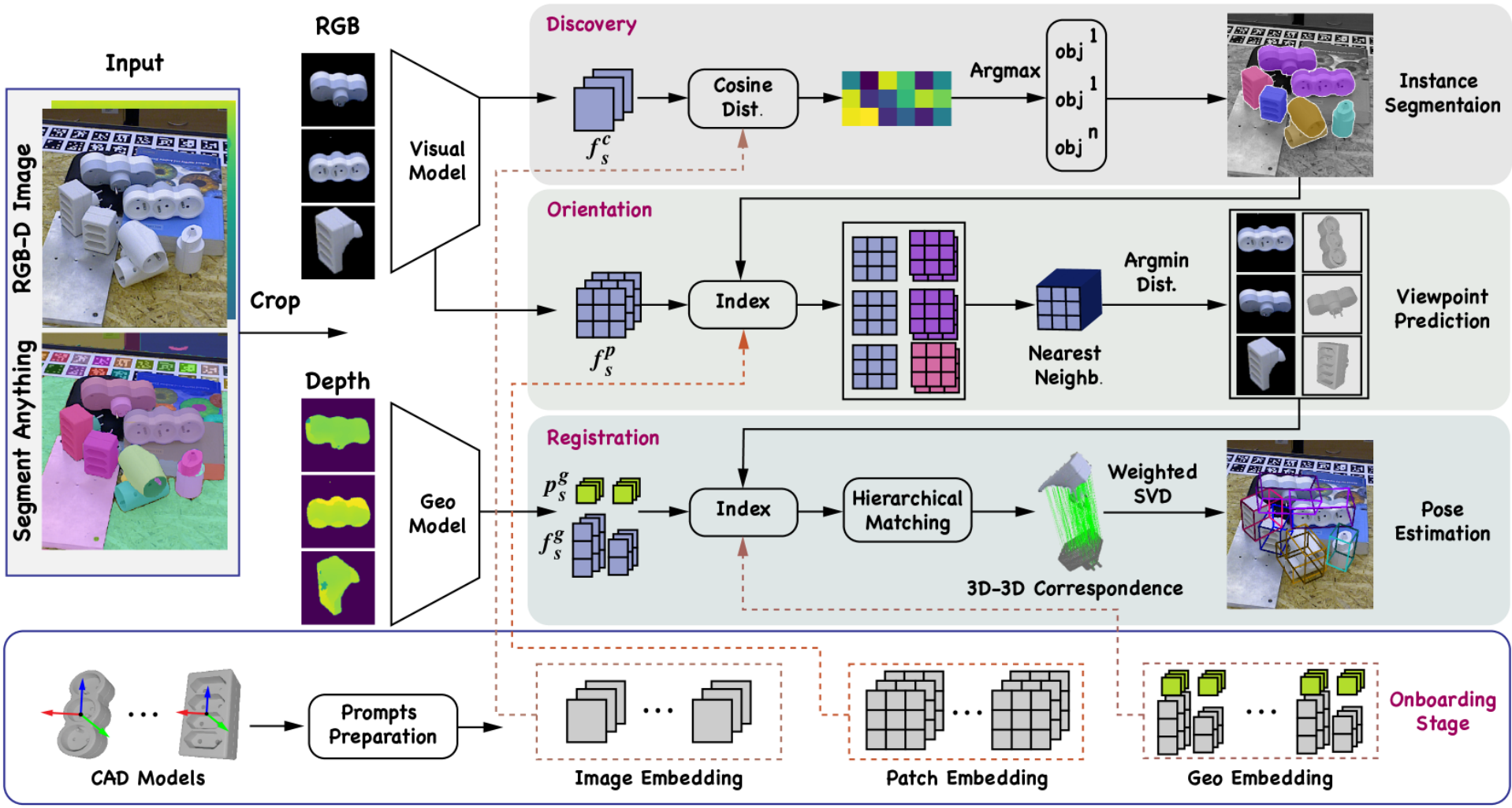


Method

- Registration (3D–3D matching → solve pose)
  - Hierarchical matching
    - GeoTransformer로 scene/template의 coarse/fine feature를 추출
    - Coarse matching
      - Coarse feature 간 유사도를 비교하여 coarse correspondence $\mathcal{C}_c$생성
    - Fine matching
      - $\mathcal{C}_c$ 의 receptive field를 이용해 관련 fine-level 후보 영역을 indexing
      - fine-level 후보 영역에서 dense correspondence $\mathcal{C}_f$ 생성
    - Weighted SVD: $\mathcal{C}_f$ 를 이용해 최종 pose (R,t) 계산



Method


Input
RGB
RGB-D Image
Segment Anything
Crop
Depth
Visual Model
Geo Model
Discovery
Cosine Dist.
Argmax
obj 1
obj 1
obj n
Instance Segmentaion
Orientation
Index
Argmin Dist.
Nearest Neighb.
Viewpoint Prediction
Registration
Index
Hierarchical Matching
Weighted SVD
3D-3D Correspondence
Pose Estimation
CAD Models
Prompts Preparation
Image Embedding
Patch Embedding
Geo Embedding
Onboarding Stage

Experiment

- Benchmark: BOP 7 core datasets
  - LM-O, T-LESS, TUD-L, IC-BIN, ITODD, HB, YCB-V

- Metrics
  - Object discovery/instance segmentation
    - mAP
  - Pose estimation
    - BOP AR(average recall)
      - 여러 pose 오차 함수 + 여러 임계값에서의 Recall을 평균낸 점수

- Extra setting
  - 기존 zero-shot 일부는 정확한 ROI 마스크 입력을 가정
    - 공정 비교를 위해 Mask R-CNN 마스크를 동일하게 제공한 setting도 함께 비교

Experiment

- SOTA comparison

	Method	Object Discovery		Pose Estimation		BOP7 Datasets								
		Zero-shot	Inst. level	Zero-shot	Refinement	LM-O	T-LESS	TUD-L	IC-BIN	ITODD	HB	YCB-V	Mean	Time (s)
1	CosyPose [54]	✗	✓	✗	✓	63.3	64.0	68.5	58.3	21.6	65.6	57.4	57.0	0.5
2	CDPNv2 [55]	✗	✓	✗	✓	63.0	43.5	79.1	45.0	18.6	71.2	53.2	53.4	1.5
3	SurfEmb [17]	✗	✓	✗	✓	*76.0*	*82.8*	85.4	*65.9*	*53.8*	86.6	79.9	*75.8*	9.0
4	Coupled [56]	✗	✓	✗	✓	73.2	82.0	85.8	60.6	47.2	*87.3*	82.9	74.1	-
5	MegaPose [10]	✗	✓	✓	–	18.7	19.7	20.5	15.3	8.00	18.6	13.9	16.2	25.6
6	OVE6D [12]	✗	✓	✓	–	49.6	52.3	-	-	-	-	57.5	-	-
7	Ours	✗	✓	✓	–	**58.3**	**55.9**	**86.9**	**53.2**	**33.8**	**69.3**	**70.1**	**61.1**	**1.8**
8	OVE6D [12]	✗	✓	✓	✓	62.7	54.6	-	-	-	-	58.7	-	-
9	GCPose [57]	✗	✓	✓	✓	65.2	**67.9**	92.6	-	-	-	75.2	-	-
10	MegaPose [10]	✗	✓	✓	✓	58.3	54.3	71.2	37.1	40.4	75.7	63.3	57.2	93.3
11	Ours	✗	✓	✓	✓	**66.3**	63.0	***94.9***	**52.0**	**44.2**	**82.0**	***84.1***	**69.5**	**48.3**

- Pose AR (BOP7 mean) & runtime 비교
  - MegaPose: mean AR 16.2, time 25.6s
  - ZeroPose: mean AR 61.1, time 1.8s
- Refiner 포함 setting에서 비교
  - MegaPose: mean AR 57.2, time 93.3s
  - ZeroPose: mean AR 69.5, time 48.3s
- Object-specific method 대비 유사 성능을 보이면서, time efficiency도 높음

Experiment

- Latency & Memory (Onboarding + Inference cost)

	Onboarding Time (s)			Memory (MB)	BOP5 Datasets						
Template Num.	Render	Feat Extraction	Total	Template Feat.	LM-O	T-LESS	TUD-L	IC-BIN	YCB-V	Mean	Time (s)
6	0.3	0.1	0.4	0.02	29.7	11.6	45.2	19.8	32.0	27.6	0.240
42	26.5	0.5	27.0	0.16	37.7	34.7	46.0	20.6	57.4	39.3	0.242
162	102.2	2.3	104.5	0.64	37.8	35.7	45.4	20.8	56.0	39.2	0.251
642	405.1	7.2	412.3	2.51	37.4	36.4	44.7	20.5	56.0	38.9	0.289

- viewpoint 템플릿 수가 늘수록 onboarding time 및 메모리 증가
  - 42 tempates: total onboarding 27.0s, memory 0.16MB, inference time 0.242s
  - 642 templates: total onboarding 412.3s, memory 2.51MB, inference time 0.289s

- 과도한 template density는 성능은 큰차이 없고, 효율만 안좋아지는 것을 확인

Paper Review

FreeZe: Training-free zero-shot 6D pose estimation with geometric and vision foundation models (ECCV 2024)

Andrea Caraffa, Davide Boscaini, Amir Hamza, Fabio Poiesi

Introduction

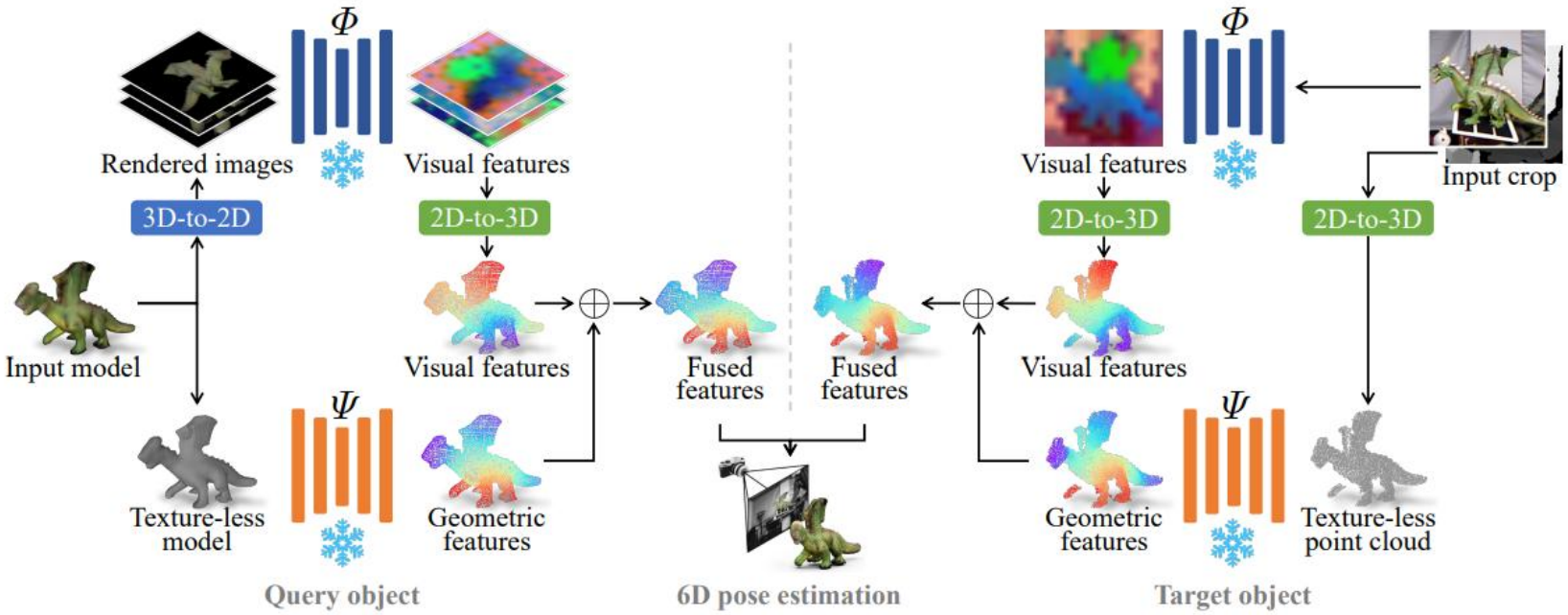
- Motivation
  - Limitations of prior Zero-shot 6D-pose estimation
    - 기존 model-based zero-shot 방법은 대규모 합성 데이터 + 장시간 학습에 의존
    - 렌더링 품질/도메인 갭에 민감하여 일반화가 흔들릴 수 있음
  - Key question
    - Task-specific training 없이도 정확한 6D 포즈 추정이 가능한까?
  - FreeZe
    - 대규모 데이터로 학습된 foundation model feature를 그대로 활용(Training-free)
    - Geometric(3D) + Visual(2D) feature를 결합하면, 별도 학습 없이도 robust correspondence를 얻을 수 있음

Method

- Key idea
  - Extract feature with training-free foundation models
    - Geometric FM (GeDi) + Vision FM (DINOv2) 를 추가 학습 없이 사용
  - 2D→3D feature lifting & fusion
    - RGB image에서 얻은 visual patch feature를 depth로 3D point에 투영
    - 3D point로부터 얻은 GeDi geometric feature와 결합하여 point-level descriptor 생성
  - Pose estimation & refinement
    - Point-level descriptor로 3D–3D correspondence 생성
    - RANSAC registration으로 초기 (R,t) 추정
    - (Optional) ICP + symmetry-aware refinement로 정밀화

Method

- Overall pipeline



Method

- Feature extraction
  - Geometric feature (GeDi)
    - GeDi를 geometric encoder로 사용
      - 각 3D point에서 반경 r = 0.3D/0.4D (D: 물체의 지름) 이내 이웃의 32d feature 추출
      - 각 feature를 concat하여 geometric feature 추출 (32d + 32d → concat(64d))
  - Visual feature (DINOv2)
    - DINOv2(ViT-G)를 vision encoder로 사용
      - 224×224 이미지로부터 → Visual feature (1536 ×16 × 16)
      - 이후 PCA로 차원 축소

Method

- 2D→3D feature lifting & fusion
  - Step 1) ROI/crop
    - Segmentation 마스크로 물체 ROI만 남기고 배경 제거
  - Step 2) ROI 이미지에서 2D visual feature 추출
    - DINOv2로 ROI 이미지의 patch feature map 추출 (16×16)
      - Bilinear interpolation으로 pixel-level feature map으로 변환
  - Step 3) Depth를 이용해 2D pixel-3D point 대응 생성
    - ROI 내 각 픽셀 (u, v)의 depth $z = D(u, v)$와 intrinsics K로 3D 점을 복원

$$X = zK^{-1}\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

Method

- 2D→3D feature lifting & fusion
  - Step 4) Pixel feature를 3D point에 할당 (lifting)
    - 대응되는 픽셀 feature를 그대로 3D point에 부착하여 point-level visual descriptor 생성
  - Step 5) ROI 이미지에서 2D visual feature 추출
    - 각 3D point에 대해 visual+geometric을 결합하여 최종 descriptor 생성
      - 두 feature의 스케일 차이를 없애기 위해 normalization 후 concat

$$f(X) = \left[\frac{g(X)}{\| g(X) \|} \;\middle|\; \frac{v(X)}{\| v(X) \|} \right]$$

X: 3D point, g(X) : geometric feature, $v(X)$: visual feature

$f(X)$: 두 정보를 합친 최종 point-level descriptor

    - 이 fused descriptor로 3D–3D correspondence를 더 안정적으로 추정할 수 있음

Method

- Pose estimation & refinement
  - RANSAC registration (3D–3D feature matching)
    - query에서 3점(triplet) 샘플링 → fused feature로 target에 NN 매칭
    - 대응점으로 pose (R, t)계산 → inlier(오차 $< \tau$인 샘플) 수 최대인 pose 선택
    - 여러 mask 후보가 있는 경우: 각 후보별로 수행 후 inlier 최대 후보 선택
  - ICP refinement
    - RANSAC 초기 pose T_c를 ICP로 point-level 정밀화 → T_f
  - SAR (Symmetry-Aware Refinement)
    - 대칭 회전 후보 R_s들에 대해 Chamfer distance로 후보를 평가/선별
    - $T_f \circ R_s$로 렌더링 → 입력 crop과의 DINOv2 patch cosine similarity로 최종 후보 선택

Experiment

- Benchmark: BOP 7 core datasets
  - LM-O, T-LESS, TUD-L, IC-BIN, ITODD, HB, YCB-V

- Metrics
  - Pose estimation
    - BOP AR(average recall)
      - 여러 pose 오차 함수 + 여러 임계값에서의 Recall을 평균낸 점수

- Evaluation setting
  - FreeZe는 pose estimation이 핵심 → 실험에서 보통 mask/ROI prior를 입력으로 사용
    - Ex: CNOS masks / SAM6D masks 등
  - 비교 시 같은 prior(ROI) / 같은 refinement 조건에서 성능 비교를 강조

Experiment

- Quantitative Results (SOTA Comparison)

	Method	Training free	Input	Prior	Refin.	BOP Dataset LM-O	T-LESS	TUD-L	IC-BIN	ITODD	HB	YCB-V	Mean
1	MegaPose [26]		RGB	CNOS		22.9	17.7	25.8	15.2	10.8	25.1	28.1	20.8
2	ZS6D [3]	✓	RGB			29.8	21.0	-	-	-	-	32.4	-
3	GigaPose [35]		RGB			29.9	27.3	30.2	23.1	18.8	34.8	29.0	27.6
4	FoundPose [40]	✓	RGB			39.7	33.8	46.9	23.9	20.4	50.8	45.2	37.3
5	SAM6D [31]		RGBD			57.0	38.2	69.8	41.5	41.4	66.9	73.2	55.4
6	FreeZe (ours)	✓	RGBD			**64.7**	**49.3**	**86.1**	**44.3**	**49.2**	**75.7**	**78.7**	**64.0**
7	MegaPose [26]		RGB	CNOS	✓	56.0	50.8	68.7	41.9	34.6	70.6	62.0	54.9
8	GigaPose [35]		RGB		✓	59.9	**57.0**	63.5	46.7	39.7	72.2	66.3	57.9
9	FoundPose [40]	✓	RGB		✓	61.0	**57.0**	69.3	47.9	40.7	72.3	69.0	59.6
10	ZeroPose [6]		RGBD		✓	53.8	40.0	83.5	39.2	52.1	65.3	65.3	57.0
11	MegaPose [26]		RGBD		✓	62.6	48.7	85.1	46.7	46.8	73.0	76.4	62.8
12	SAM6D [31]		RGBD		✓	63.5	46.3	80.0	46.5	54.3	71.1	80.0	63.2
13	FreeZe (ours)	✓	RGBD		✓	**69.0**	52.0	**93.6**	**49.9**	**56.1**	**79.0**	**85.3**	**69.3**
14	SAM6D [31]		RGBD	SAM6D		62.7	42.0	77.7	**50.4**	45.5	68.9	74.3	60.2
15	FreeZe (ours)	✓				**67.6**	**50.0**	**88.1**	48.7	**52.0**	**76.1**	**77.4**	**65.7**
16	SAM6D [31]		RGBD	SAM6D	✓	68.7	49.8	87.4	**56.1**	57.7	75.4	82.8	68.2
17	FreeZe (ours)	✓			✓	**71.6**	**53.1**	**94.9**	54.5	**58.6**	**79.6**	**84.0**	**70.9**

- 다양한 prior/조건에서 FreeZe가 상위 성능을 보고
  - SAM6D prior + refinement 조건에서 mean AR 70.9
  - 추가 학습 없이 foundation model feature(GeDi+DINOv2)만으로 높은 AR 달성

Experiment

- Computational times

Method	Input data RGB	D	Localization prior CNOS	SAM6D	AR	Time (s)
MegaPose [26]	✓		✓		65.9	4.59
MegaPose [26]	✓			✓	68.4	6.06
SAM6D [31]	✓	✓	✓		82.0	0.98
SAM6D [31]	✓	✓		✓	90.3	3.94
FreeZe (ours)	✓	✓	✓		93.6	2.72
FreeZe (ours)	✓	✓		✓	94.9	4.79

- 동일 HW에서 MegaPose / SAM6D / FreeZe 비교
  - 기존 model 대비 비슷한 computational time을 가지면서 높은 성능을 보이는 것을 확인

Experiment

- Ablation study
  - Feature encoder ablation
    - Geometric encoder: GeDi > FCGF
    - Vision encoder: DINOv2 > CLIP
    - Multi-scale & fusion: MS(Multi-Scale)-GeDi가 일반 GeDi보다 안정적

		Geometric encoder		Vision encoder		Prior	ICP	AR
		Method	Radius	Method	Backbone			
Geometry	1	FPFH	0.3	-	-	m		20.7
	2	FCGF	-	-	-	m		20.8
	3	GeDi	0.3	-	-	m		**55.3**
	4	GeDi	0.2	-	-	m		52.6
	5	GeDi	0.3	-	-	m		55.3
	6	GeDi	0.4	-	-	m		55.0
	7	MS-GeDi	(0.2, 0.3)	-	-	m		56.2
	8	MS-GeDi	(0.3, 0.4)	-	-	m		**56.5**
Vision	9	-	-	RGB	-	m		15.1
	10	-	-	CLIP	ViT-L	m		30.5
	11	-	-	DINOv2	ViT-L	m		**36.8**
	12	-	-	DINOv2	ViT-S	m		31.9
	13	-	-	DINOv2	ViT-B	m		33.7
	14	-	-	DINOv2	ViT-L	m		36.8
	15	-	-	DINOv2	ViT-G	m		**39.6**
Prior	16	MS-GeDi	(0.3, 0.4)	DINOv2	ViT-G	bb		64.7
	17	MS-GeDi	(0.3, 0.4)	DINOv2	ViT-G	bb	✓	68.6
	18	MS-GeDi	(0.3, 0.4)	DINOv2	ViT-G	m		64.7
	19	MS-GeDi	(0.3, 0.4)	DINOv2	ViT-G	m	✓	**69.0**

Experiment

- Ablation study
  - Pose refinement ablation

	Refin.	LM-O	T-LESS	TUD-L	IC-BIN	ITODD	HB	YCB-V	Mean
1	None	64.7	49.3	86.1	44.3	49.2	75.7	78.7	64.0
2	+ICP	**69.0**	**52.0**	**93.6**	47.3	**56.1**	78.4	84.9	68.8
3	+SAR	**69.0**	**52.0**	**93.6**	**49.9**	**56.1**	**79.0**	**85.3**	**69.3**

- SAR은 대칭이지만 texture가 구분되는 객체가 포함된 데이터셋에서 효과적인 것을 확인
  - IC-BIN(+2.6), HB(+0.6), YCB-V(+0.4)

- 반면 LM-O, T-LESS, TUD-L, ITODD에서는 ICP만 사용한 경우 대비 이득이 거의 없음

Conclusion

- 6D pose estimation
  - 카메라 좌표계 기준에서 물체의 자세를 추정하는 문제
  - Template matching
    - 여러 viewpoint로 렌더링해 만든 템플릿 후보와 입력을 비교해 가장 유사한 pose를 선택
  - Feature matching
    - 입력과 모델에서 feature를 추출해 correspondence를 만들고, PnP(2D–3D) / 3D registration(SVD) 으로 pose를 계산
- ZeroPose
  - SAM segment-CAD 매칭 후, 객체 patch feature 비교로 initial viewpoint 생성
  - 이후 GeoTransformer 기반 3D–3D hierarchical matching → weighted SVD로 pose를 추정하며, multi-hypothesis + Chamfer로 최종 후보를 선택
- FreeZe (training-free zero-shot)
  - Foundation model로 2D/3D feature를 추출
  - 2D→3D lifting & fusion하여 point-level descriptor를 구성
  - RANSAC registration으로 pose를 계산한 후, ICP + SAR로 정밀화

감사합니다