2024 여름 세미나

Hand Pose Estimation

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

Presented By
MinSuh Song

Outline

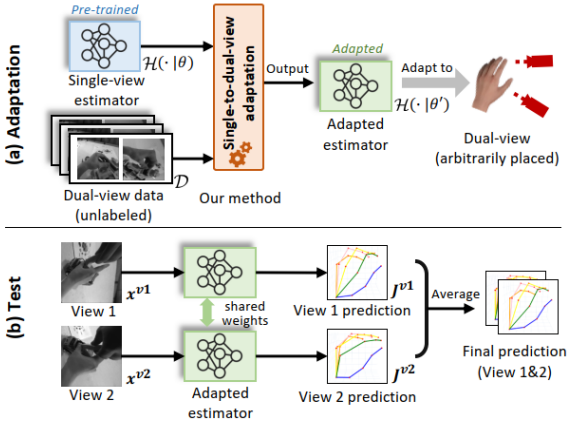
- Ruicong Liu, Takehiko Ohkawa, et al. **“S2DHand: Single-to-Dual-View Adaptation for Egocentric 3D Hand Pose Estiation.”** CVPR, 2024

- Zicong Fan, Maria Parelli, et al. **“HOLD: Category-agnostic 3D Reconstruction of Interacting Hands and Objects from Video.”** CVPR, 2024

S2D Hand: Single-to-Dual-View Adaptation for Egocentric 3D Hand Pose Estimation

S2DHand

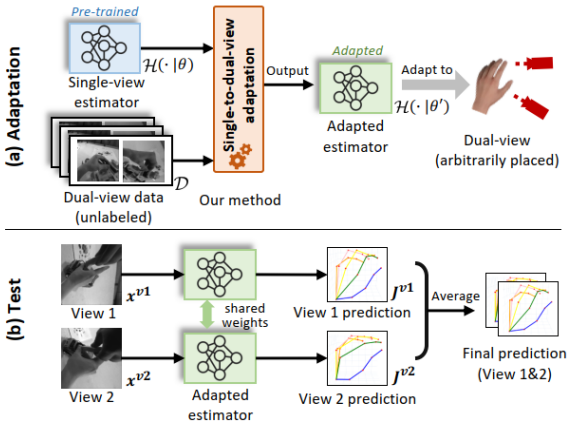
- Abstract
  - Single view image만으로 hand pose estimation을 수행하는 것에는 accuracy 측면에서 한계점이 존재
    - 하나의 시점(view)를 추가하여 더 정확하게 hand pose estimation을 수행하고자 함
  - 기존 multi-view method들의 문제점
    - Training 과정에서 image에 대한 multi-view annotation이 필요
    - Test 과정에서 camera parameter가 달라지면 모델을 적용할 수 없음
  - 본 논문은 pretrained 된 single-view estimator를 dual view로 적응시키는 새로운 방법으로 S2DHand를 제안



<S2DHand의 pipeline>

S2DHand

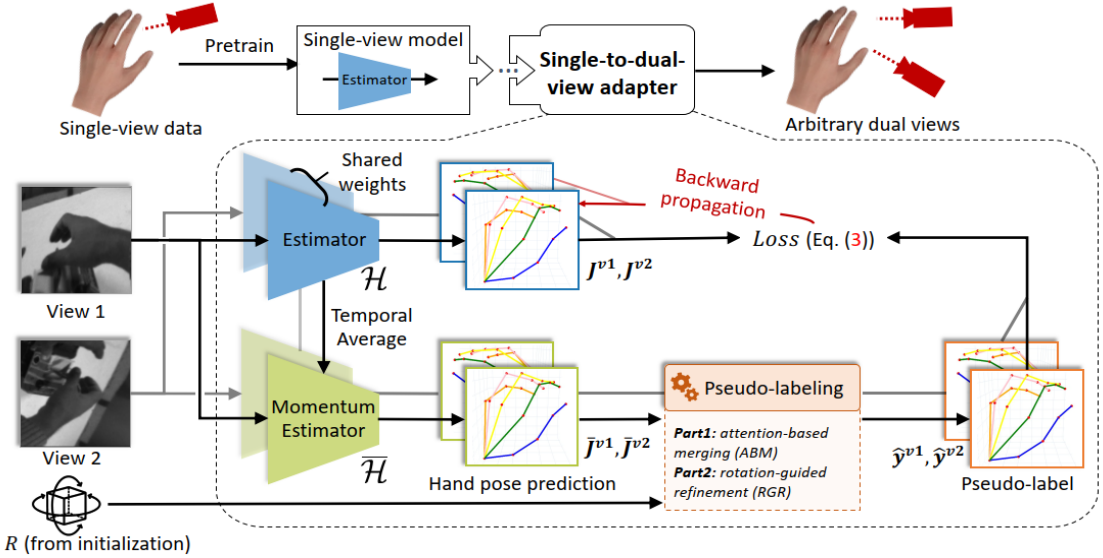
- Introduction
  - 기존의 pretrained된 single-view estimator를 baseline으로 사용
    - 본 논문에서는 $DetNet$[1] 을 사용
  - Single-view estimator는 adaptation 과정을 거쳐 adapted estimator로 조정
    - 동일한 이미지에 대해 서로 다른 시점의 dual view input pair를 통해 학습을 진행
  - Testing 과정에서 adapted estimator는 서로 다른 view에 대한 joint prediction을 생성
    - 이는 이후에 combined되어 최종 output을 생성



<S2DHand의 pipeline>

S2DHand

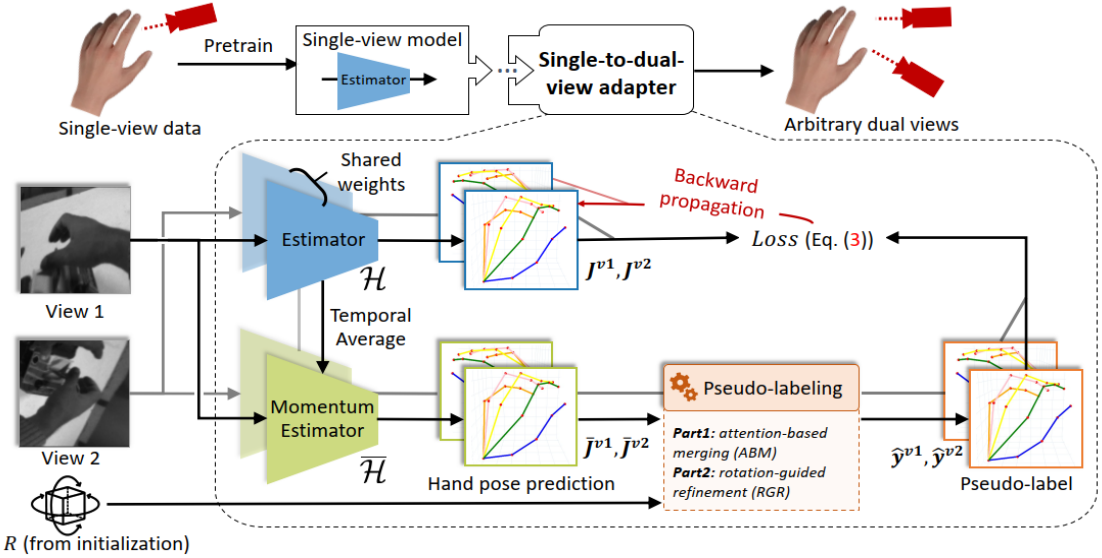
- Proposed Method
  - Initialization step
    - Initialization의 목적은 두 시점 간의 rotation matrix R을 추정하기 위함
      - R은 두 카메라 좌표계를 맞추어 주는데 필요
    - Unlabeled dual-view data $D = \{x_i^{v1}, x_i^{v2}|_{i=1}^{N}\}$를 single-view estimator H에 입력하여 joints (J^{v1}, J^{v2})를 생성
    - 이를 이용하여 초기 rotation matrix R을 계산



<S2DHand의 전체적인 구조>

S2DHand

- Proposed Method
  - Initialization step
    - $R(0) = \frac{1}{N}\sum_{i=1}^{N} rot(J_i^{v1}, J_i^{v2})$
      - rot함수는 두 3D joints 로부터 3*3 rotation matrix를 생성
    - Rotation matrix는 학습을 진행함에 따라 momentum estimator의 영향을 받아 update
      - $R^{(T)} = \eta_R R^{(T-1)} + (1-\eta_R)\cdot\frac{1}{B}\sum_{i=1}^{B} rot(\bar{J}_i^{v1}, \bar{J}_i^{v2})$



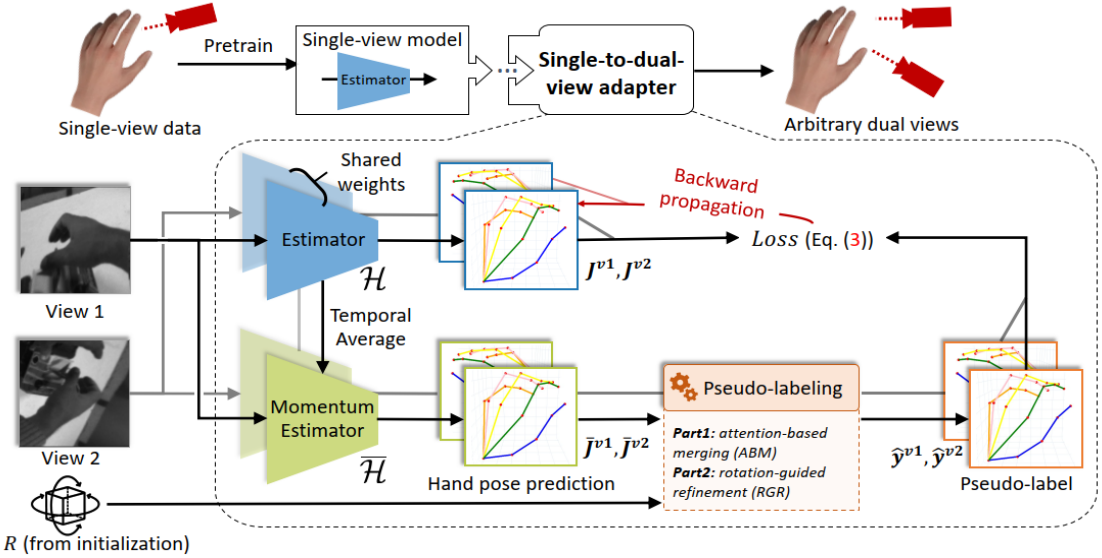
<S2DHand의 전체적인 구조>

S2DHand

- Proposed Method
  - Single-to-dual-view adaptation
    - S2DHand architecture는 baseline single-view model $\mathcal{H}(\cdot|\theta)$과 그것의 momentum version $\bar{\mathcal{H}}(\cdot|\bar{\theta})$으로 구성
    - $\bar{\mathcal{H}}$의 parameter $\bar{\theta}$는 temporal moving average를 통해 update

$$\bar{\theta}^{(T)} = \eta_\theta \bar{\theta}^{(T-1)} + (1-\eta_\theta)\theta$$

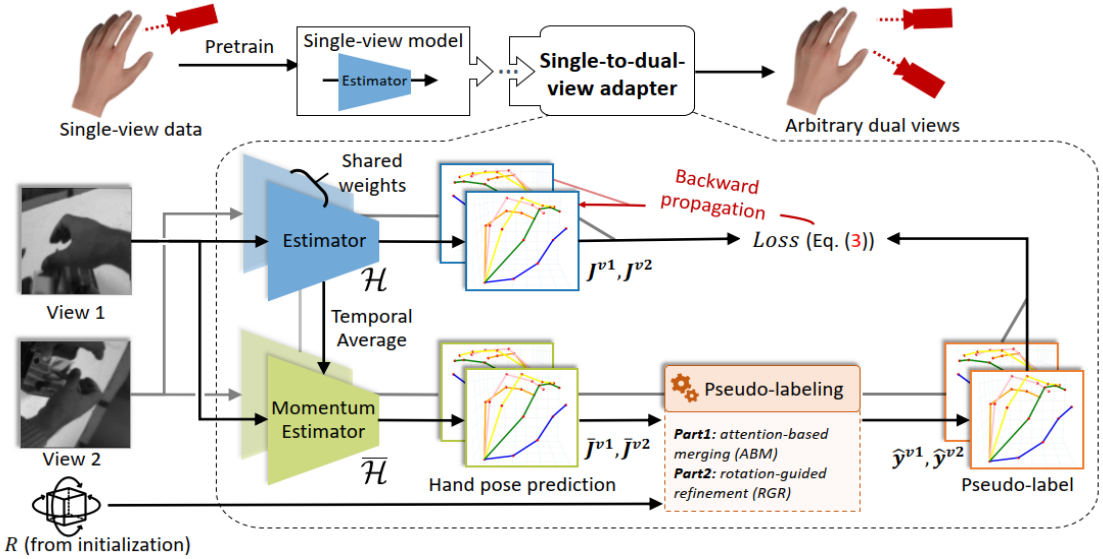
✓η_θ는 0.99로 초기화



<S2DHand의 전체적인 구조>

S2DHand

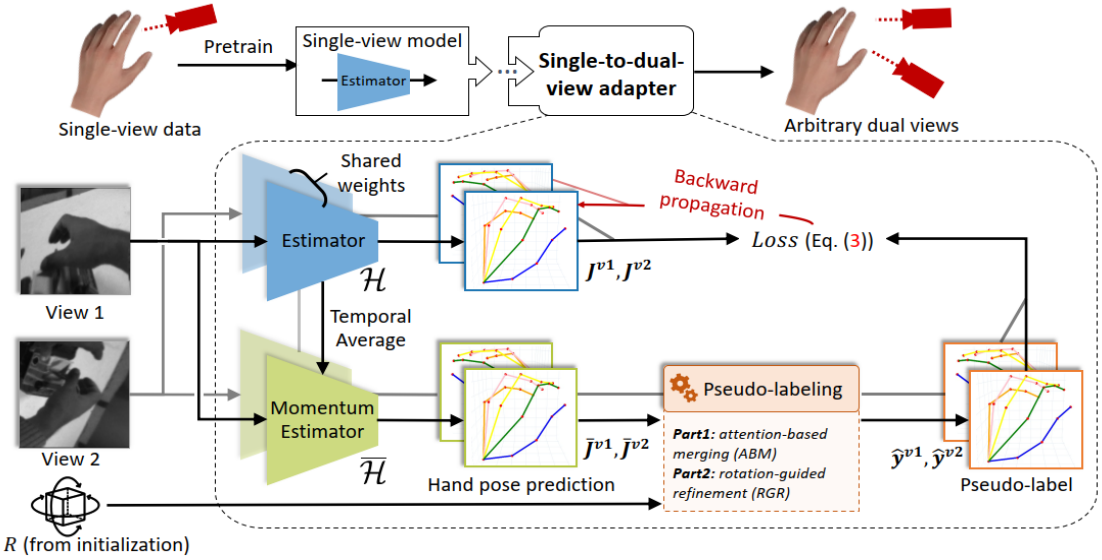
- Proposed Method
  - Single-to-dual-view adaptation
    - Momentum estimator의 목적은 pseudo-labels를 생성하여 기존 single view estimator H를 학습시키기 위함
    - $D = \{x_i^{v1}, x_i^{v2}|_{i=1}^{N}\}$를 momentum estimator $\bar{H}$에 입력하여 joints $(\bar{J}^{v1}, \bar{J}^{v2})$를 생성



<S2DHand의 전체적인 구조>

S2DHand

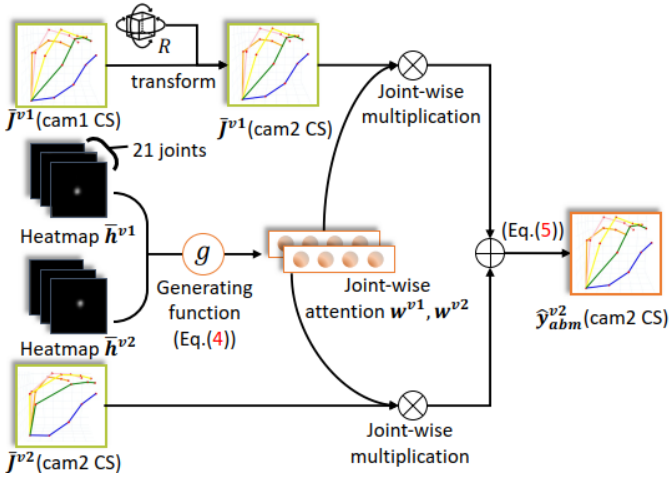
- Proposed Method
  - Single-to-dual-view adaptation
    - Joints $(\bar{J}^{v1}, \bar{J}^{v2})$는 pseudo-labeling module을 통해 pseudo-label $(\hat{y}^{v1}, \hat{y}^{v2})$로 출력
    - 기존 single-view estimator가 추정한 (J^{v1}, J^{v2})와의 loss를 최소화하는 방향으로 학습을 진행
    - $\mathcal{L} = \|J^{v1} - \hat{y}^{v1}\|_2 + \|J^{v2} - \hat{y}^{v2}\|_2$



<S2DHand의 전체적인 구조>

S2DHand

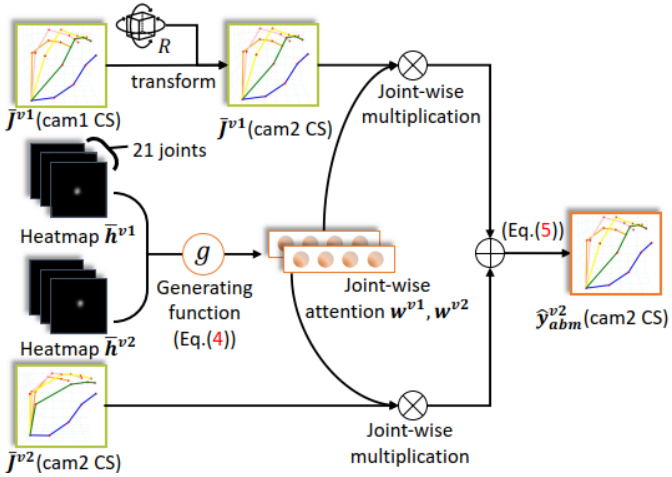
- Proposed Method
  - Pseudo-labeling: attention-based merging
    - 두 시점에서 예측된 손의 3D pose는 같은 좌표계로 변환하였을 때, 동일해야 함
      - 예를 들어 손목을 기준으로 변환하였을 때, $R\bar{J}^{v1} = \bar{J}^{v2}$가 되어야 함
    - Pseudo-label 계산 시, 단순히 $(R\bar{J}^{v1} + \bar{J}^{v2})/2$ 로 계산을 하면 occlusion을 고려하지 못함
    - 각 joints의 prediction confidence를 나타내는 attention weight ω를 계산
      - Momentum estimator $\bar{H}$로부터 생성된 2D heatmap $\bar{h}$에서 계산
      - Heatmap의 각 픽셀은 해당 joint가 위치에 존재할 확률을 나타냄



<Attention-based merging의 pipeline>

S2DHand

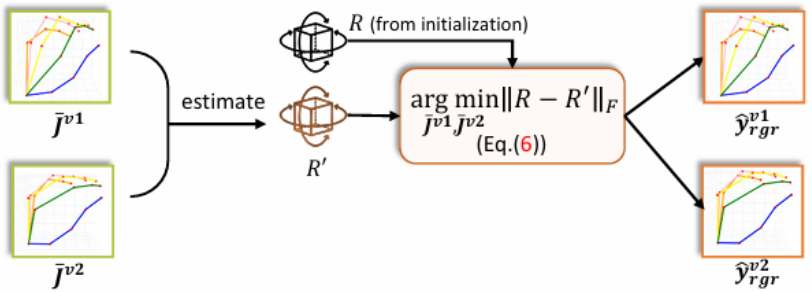
- Proposed Method
  - Pseudo-labeling: attention-based merging
    - $\omega_j^v = \frac{\beta \max(h_j^v)}{\sum_{v \in \{v1, v2\}} \beta \max(h_j^v)}$
    - Attention weight를 사용하여 두 view의 prediction을 결합해서 pseudo-label 생성
      - $\hat{y}_{abm}^{v1} = w^{v1}\bar{J}^{v1} + w^{v2}R^T\bar{J}^{v2}$
      - $\hat{y}_{abm}^{v2} = w^{v1}R\bar{J}^{v1} + w^{v2}\bar{J}^{v2}$



<Attention-based merging의 pipeline>

S2DHand

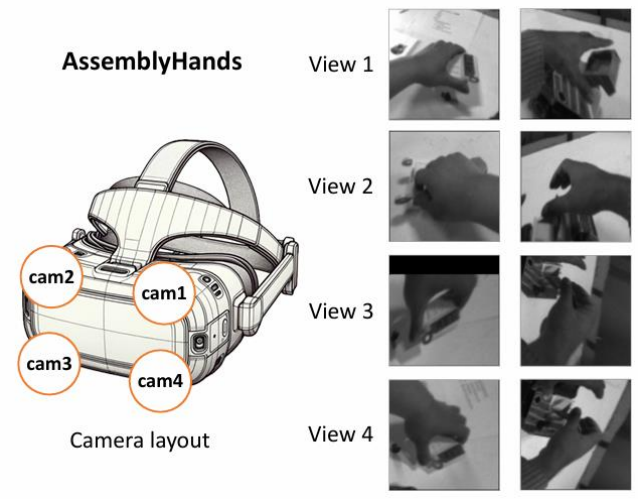
- Proposed Method
  - Pseudo-labeling: Rotation-guided Refinement
    - Single-to-dual view adaptation이 완료된 후에도 rotation matrix는 초기값인 R과 동일해야 함
    - Momentum estimator 로부터 생성된 3D joints $(\bar{J}^{v1}, \bar{J}^{v2})$을 통해 rotation matrix 계산
      - $R' = rot(\bar{J}^{v1}, \bar{J}^{v2})$
    - R과 R'의 차이가 최소가 되게 하는 방향으로 학습을 진행
      - $\hat{y}^{v1}_{rgr}, \hat{y}^{v1}_{rgr} = \underset{\bar{J}^{v1}, \bar{J}^{v2}}{argmin} \|R - rot(\bar{J}^{v1}, \bar{J}^{v2})\|_F$
    - 최종 pseudo-label $\hat{y} = \alpha \hat{y}_{abm} + (1-\alpha)\hat{y}_{rgr}$
      - α의 초기값은 0.7



<Attention-based merging의 pipeline>

S2DHand

- Experiment
  - 실험 환경
    - 그림과 같은 VR기기 2대를 이용하여 서로 다른 4개의 view를 이용하여 실험을 진행
      - Assembly Hands dataset은 두 부분으로 나누어서 Headset1, Headset2로 나누어 사용
    - Dataset
      - D_{ah}: Assembly Hands
      - D_{syn}: GANerated Hands + Rendered Handpose
    - Evaluation Metrics
      - Mean per joint position error (MPJPE)
        - ✓Mono-M
        - ✓Dual-M



<S2DHand의 실험 도구>

S2DHand

- Experiment
  - Results
    - In-dataset $(D_{ah} \rightarrow D_{ah})$
      - Baseline model이 Assembly Hands로 pretrained 되고, Assembly Hands로 adaptation 진행
    - Cross-dataset $(D_{syn} \rightarrow D_{ah})$
      - Baseline model이 synthetic dataset으로 pretrained되고, Assembly Hands로 adaptation 진행
    - Baseline으로 설정한 DetNet보다 adaptation 과정을 거친 S2DHand가 더 좋은 성능을 보임

Camera pair	Method	In-dataset ($\mathcal{D}_{ah} \rightarrow \mathcal{D}_{ah}$)				Cross-dataset ($\mathcal{D}_{syn} \rightarrow \mathcal{D}_{ah}$)			
		$\mathcal{D}_{ah} - Headset1$		$\mathcal{D}_{ah} - Headset2$		$\mathcal{D}_{ah} - Headset1$		$\mathcal{D}_{ah} - Headset2$	
		Mono-M	Dual-M	Mono-M	Dual-M	Mono-M	Dual-M	Mono-M	Dual-M
$cam\ 1, 2$	Baseline	43.00	39.20	54.71	52.38	67.93	60.48	70.32	62.26
	S2DHand	**31.01** ▼ 27.9%	**31.36** ▼ 20.0%	**45.52** ▼ 16.8%	**45.14** ▼ 13.8%	**63.46** ▼ 6.6%	**59.32** ▼ 1.9%	**70.09** ▼ 0.3%	**60.97** ▼ 2.1%
$cam\ 1, 3$	Baseline	25.00	23.29	22.59	21.08	57.79	51.42	64.00	60.25
	S2DHand	**19.73** ▼ 21.1%	**19.92** ▼ 14.5%	**17.90** ▼ 20.8%	**17.68** ▼ 16.1%	**50.84** ▼ 12.0%	**47.55** ▼ 7.5%	**61.81** ▼ 3.4%	**58.34** ▼ 3.2%
$cam\ 1, 4$	Baseline	24.90	22.70	16.73	14.91	52.71	46.55	54.32	50.57
	S2DHand	**20.88** ▼ 16.1%	**20.87** ▼ 8.1%	**14.64** ▼ 12.5%	**14.29** ▼ 4.2%	**46.05** ▼ 12.6%	**42.50** ▼ 8.7%	**46.59** ▼ 14.2%	**45.66** ▼ 9.7%
$cam\ 2, 3$	Baseline	17.96	15.23	17.10	15.08	53.36	48.42	52.84	48.84
	S2DHand	**14.97** ▼ 16.6%	**14.44** ▼ 5.2%	**14.42** ▼ 15.7%	**14.20** ▼ 5.8%	**40.26** ▼ 24.6%	**39.32** ▼ 18.8%	**43.61** ▼ 17.5%	**42.88** ▼ 12.2%
$cam\ 2, 4$	Baseline	22.09	19.84	23.24	20.96	59.44	54.32	61.13	57.41
	S2DHand	**17.98** ▼ 18.6%	**17.75** ▼ 10.5%	**18.31** ▼ 21.2%	**18.41** ▼ 12.2%	**50.59** ▼ 14.9%	**49.41** ▼ 9.0%	**52.45** ▼ 14.2%	**51.48** ▼ 10.3%
$cam\ 3, 4$	Baseline	16.83	15.77	19.93	18.08	45.82	42.34	49.84	48.99
	S2DHand	**16.36** ▼ 2.8%	**15.55** ▼ 1.4%	**19.25** ▼ 3.4%	**17.80** ▼ 1.5%	**39.46** ▼ 13.9%	**37.43** ▼ 11.6%	**44.04** ▼ 11.6%	**42.88** ▼ 12.5%
Overall	Baseline	24.96	22.67	25.72	23.75	56.18	50.59	58.74	54.72
	S2DHand	**20.16** ▼ 19.2%	**19.98** ▼ 11.9%	**21.67** ▼ 15.7%	**21.25** ▼ 10.5%	**48.44** ▼ 13.8%	**45.92** ▼ 9.2%	**53.11** ▼ 9.6%	**50.37** ▼ 7.9%

<S2DHand 적용 후 향상된 실험 결과>

S2DHand

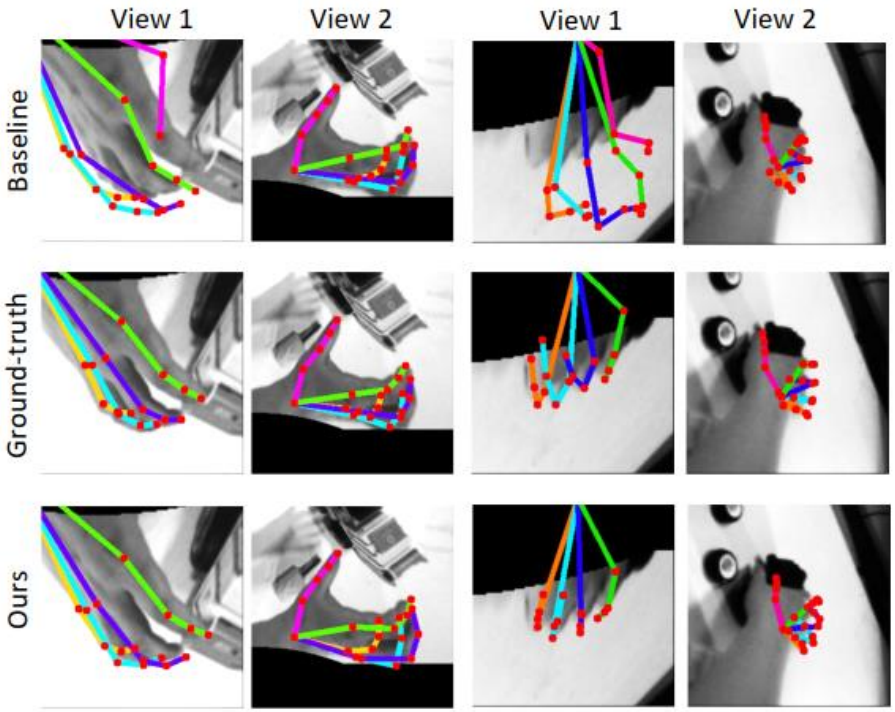
- Experiment
  - Results
    - Cross-dataset setting에서 기존의 다른 adaptation method들과 성능 비교
      - ADDA, DAGEN, RegDA, SFDAHPE
    - 더 낮은 에러율을 보이고, adaptation 과정에서 source dataset을 필요로 하지 않는 source-free
      - Source-free: 모델이 source dataset에서 학습된 후, 다른 데이터셋에서도 잘 동작하도록 하는 방법

$\mathcal{D}_{syn} \rightarrow \mathcal{D}_{ah}$	SF	$\mathcal{D}_{ah} - Headset1$		$\mathcal{D}_{ah} - Headset2$	
		Mono-M	Dual-M	Mono-M	Dual-M
Source Only		56.18	50.59	58.74	54.72
Fine-tune*		45.03	38.11	47.75	42.19
ADDA [36]	✗	56.90	48.48	57.87	51.39
DAGEN [11]	✗	55.37	49.72	57.62	53.17
RegDA [17]	✗	51.41	47.85	54.75	51.50
SFDAHPE [32]	✓	54.06	49.11	57.22	53.39
S2DHand (Ours)	✓	**48.44**	**45.92**	**53.11**	**50.37**

<S2DHand와 기존 methods와의 성능 비교표>

S2DHand

- Experiment
  - Baseline과 비교한 정성적 결과

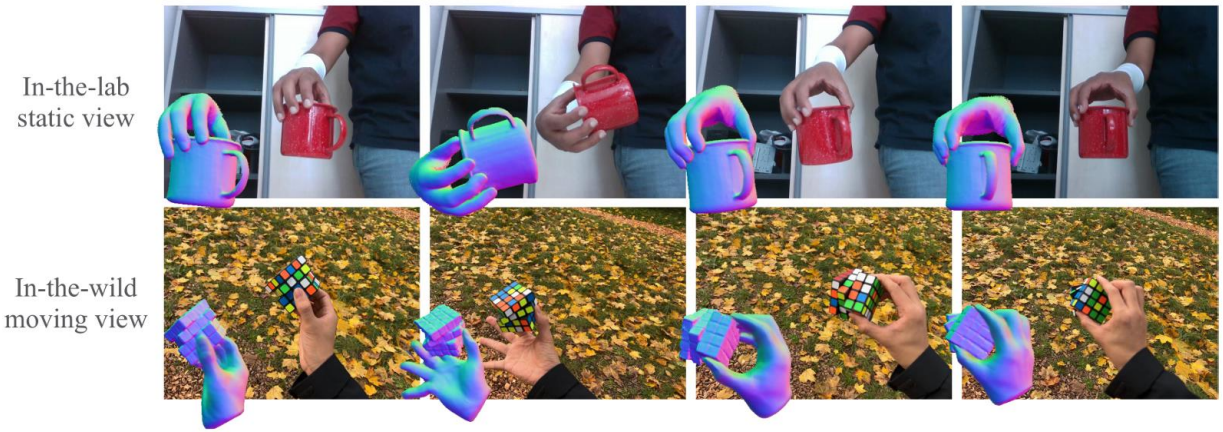


< 기존 모델과 HandOccNet의 정성적 평가>

HOLD: Category-agnostic 3D Reconstruction of Interacting Hands and Objects from Video

HOLD

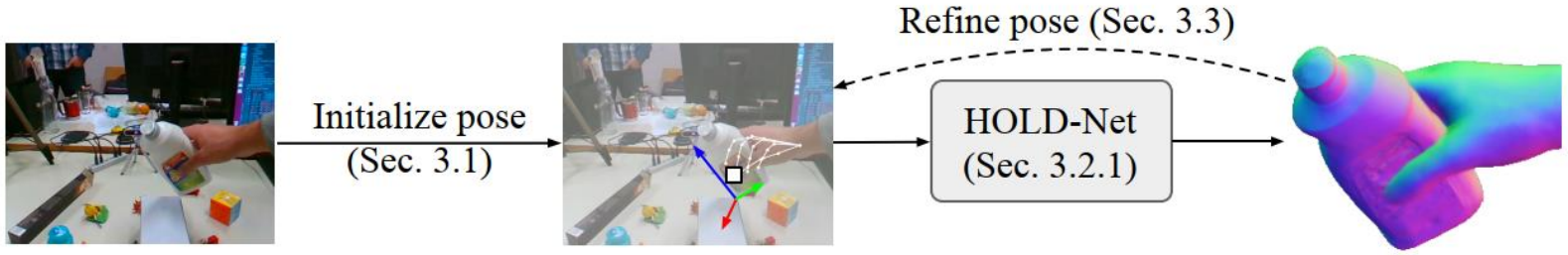
- Abstract
  - Video (sequence of frames)에서 3D hand-object estimation을 수행
  - 기존 hand-object interaction을 다루는 논문들의 한계점
    - Pre-scanned object template 혹은 대량의 3D-object data를 필요로 함
      - In-the-wild scenarios에서 일반화하기 힘듦
  - 본 논문은 monocular video에서 category-agnostic method를 사용하여 hand-object interaction을 reconstruct하는 HOLD를 제안



<HOLD의 정성적 결과>

HOLD

- Proposed Method
  - Off-the-shelf estimator를 이용하여 hand와 object의 pose를 출력
    - Hand pose estimation으로는 METRO[1] 모델 사용
    - Object pose estimation으로는 HLoc[2] 를 사용하여 structure-from-motion (SFM) 수행
  - 이를 통해 HOLD-Net을 적은 epoch으로 학습
  - HOLD-Net의 결과로 출력된 shape을 바탕으로 hand-object interaction constraints를 추가하여 pose를 refine
  - Refine된 pose를 이용하여 HOLD-Net을 fully train



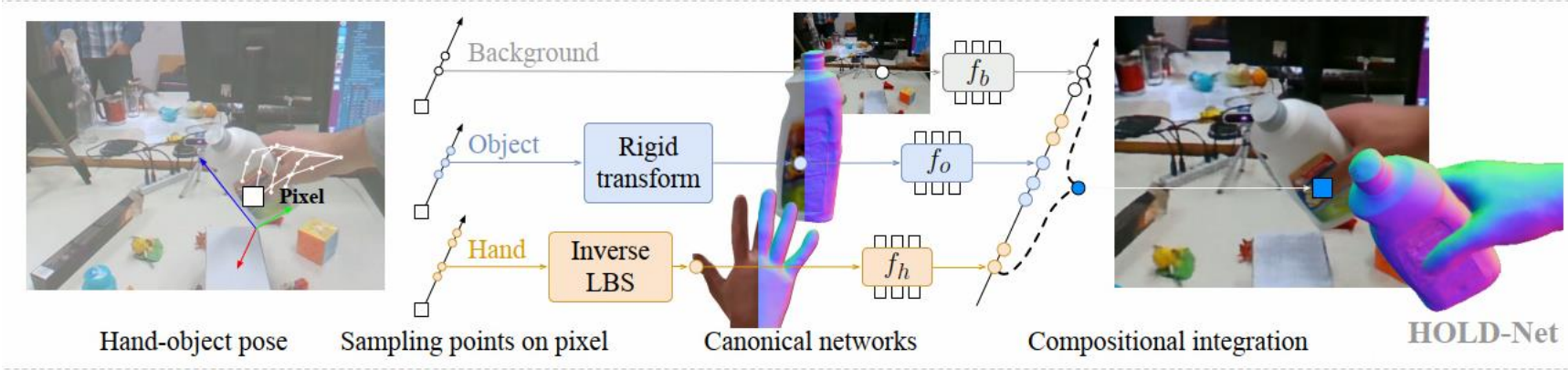
<HOLD의 mechanism>

HOLD

- Proposed Method
  - Hand Model
    - Hand Model을 하나의 network로 정의
    - METRO[1] 모델을 사용해서 hand poses θ, shape β, global rotation R_h, translation t_h를 얻음
    - Observation space의 hand points(x')들을 canonical space(x)로 변환하기 위해 Inverse Linear Blend Skinning(Inverse LBS)을 적용

$$x = \left(\sum_{i=1}^{nb} w_i(x') \cdot B_i\right)^{-1} x'$$

✓B_i는 bone transformation, $w_i(x')$는 skinning weights

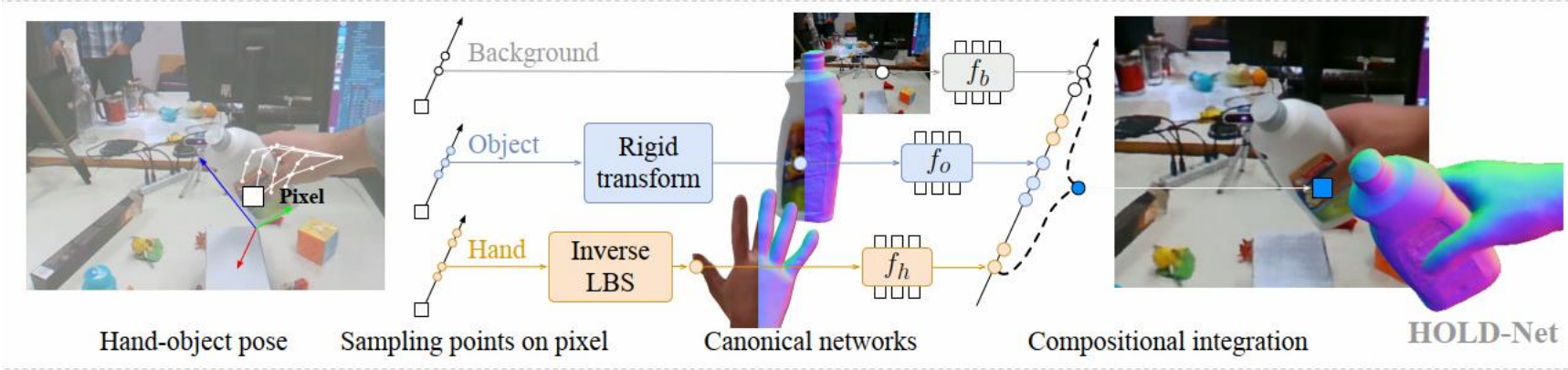


<HOLD-Net의 전체적인 구조>

HOLD

- Proposed Method
  - Hand Model
    - 여기서 손의 canonical point x를 MLP(f_h)에 입력하여 손 표면과의 signed distance(d)와 color(c)를 추정

$$f_h(x) = (d, c)$$



<HOLD-Net의 전체적인 구조>

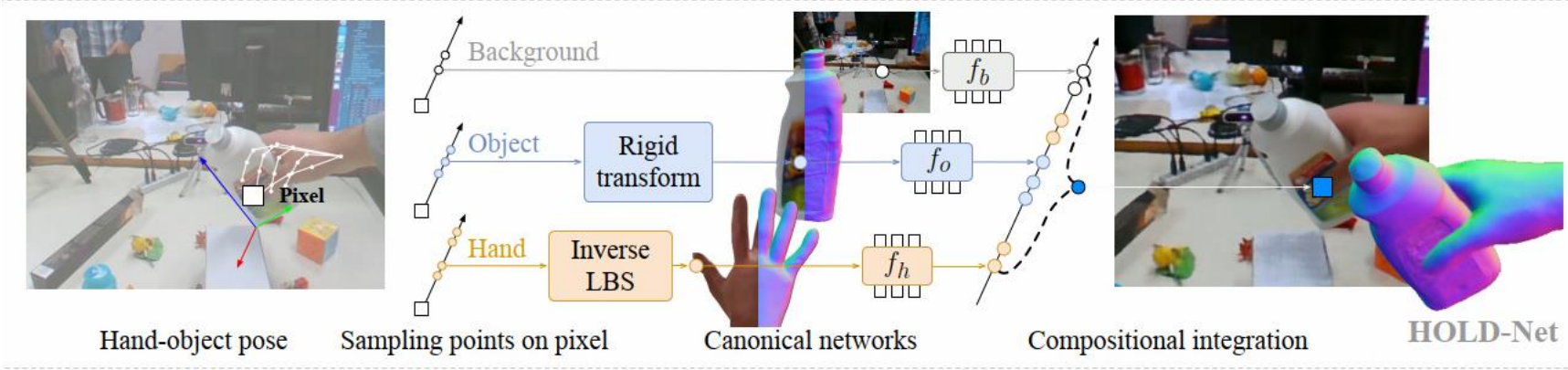
HOLD

- Proposed Method
  - Object Model
    - HLoc[1)] 모델을 사용해서 object scale s, rotation R_o, translation t_o를 얻음
    - Object model은 observation space에서 canonical space로 변환할 때 Rigid transform을 이용

      ∴ $x = (sR_o)^{-1} \cdot (x' - t_o)$
    - Hand model과 마찬가지로 canonical point x를 MLP(f_o)에 입력하여 distance와 color를 추정

      ∴ $f_o(x, z_o) = (d, c)$

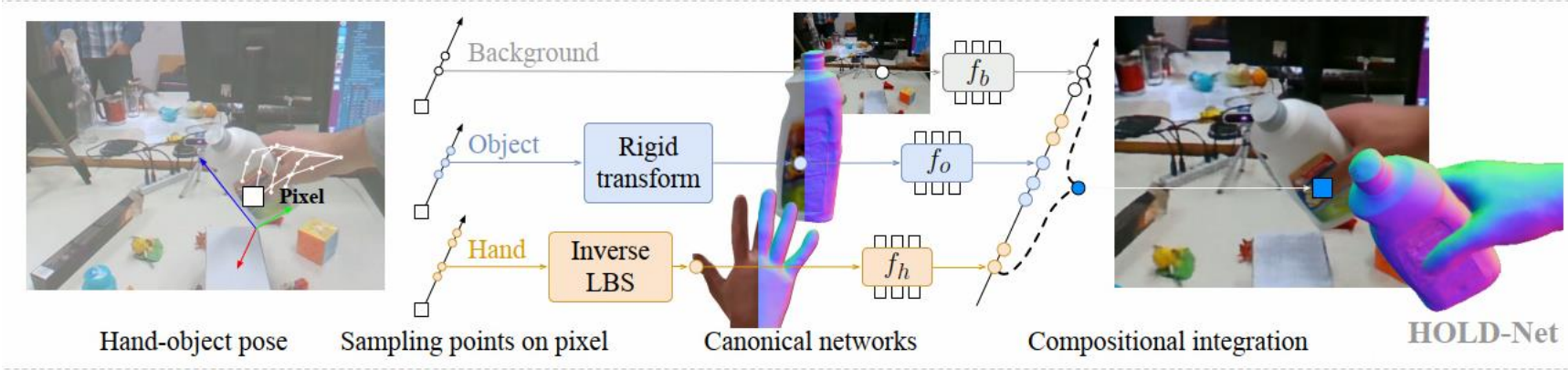
      ✓z_o는 물체의 occlusion이나 shadow를 표현하기 위한 parameter



<HOLD-Net의 전체적인 구조>

HOLD

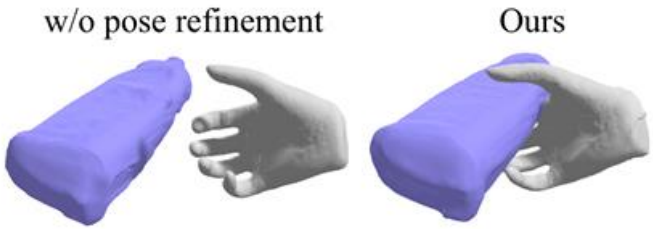
- Proposed Method
  - Compositional Integration
    - Hand와 object의 signed distance를 volume rendering을 위한 density σ로 변환하는데 scaled Laplace 분포의 cumulative distribution function을 사용
      - $C_F(r) = \sum_{i=1}^{2n} \tau_i c_i$
        - ✓ $\tau_i = \exp(-\sum_{j<i} \sigma_j \delta_j)(1 - \exp(-\sigma_i \delta_i))$
      - $C(r) = C_F(r) + (1 - M_F(r))C_B(r)$
    - Inverse LBS를 통해 canonical space로 변환된 hand model과, rigid transform으로 변환된 object model을 합치는 과정



<HOLD-Net의 전체적인 구조>

HOLD

- Proposed Method
  - Pose refinement
    - 초기 pose estimation은 noise가 많아 종종 부정확하기에 refinement가 필요
      - 아래 그림과 같이 손과 물체 사이의 거리에 오차가 발생
    - Initial pose를 바탕으로 적은 epoch 동안 training하여 hand와 object의 초기 3D 형상을 추정
    - 이를 바탕으로 contact loss를 최소화하여 hand와 object가 연결되도록 함
      - $\mathcal{L}_{contact} = \sum_i \min_j \left\| V_{tips}^i - V_o^j \right\|$
      - 물체와 접촉하는 hand tips의 vertices와 object vertices 사이의 거리를 minimize
    - Pose refinement 결과 생성된 refined pose parameter θ와 $\{\beta, R_h, t_h, R_o, t_o, s\}$를 이용하여 HOLD-Net을 fully train



<Pose refinement를 제외한 실험의 정성적 결과>

HOLD

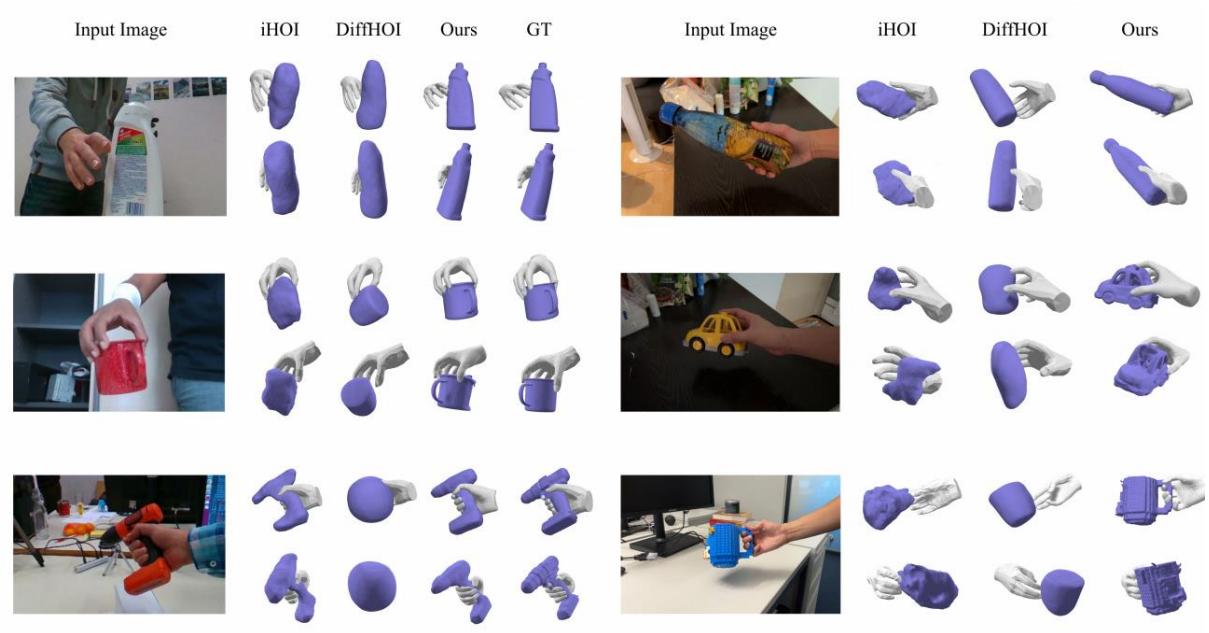
- Experiments
  - Datasets
    - HO3D-v3
      - In-the-lab dataset
      - 물체를 손으로 잡고 있는 image dataset
  - Evaluation metrics
    - Mean per joint position error (MPJPE)
    - Chamfer distance
      - 두 점들의 집합 간의 유사성을 측정하는 지표

	MPJPE [mm] ↓	CD [cm^2] ↓	F10 [%] ↑	CD_h [cm^2] ↓
iHOI† [65]	38.4	3.8	75.8	41.7
DiffHOI [66]	32.3	4.3	68.8	43.8
Ours	**24.2**	**0.4**	**96.5**	**11.3**

<기존 연구들과 HOLD와의 성능차이 비교표>

HOLD

- Experiments
  - Results
    - HOLD와 기존의 SOTA hand-object reconstruction methods의 차이 비교
      - HOLD가 더 정교한 object shape, hand pose를 추정
      - 다양한 objects에 대해서도 성공적으로 수행



<기존 연구들과 HOLD와의 정성적 평가 비교>

감사합니다