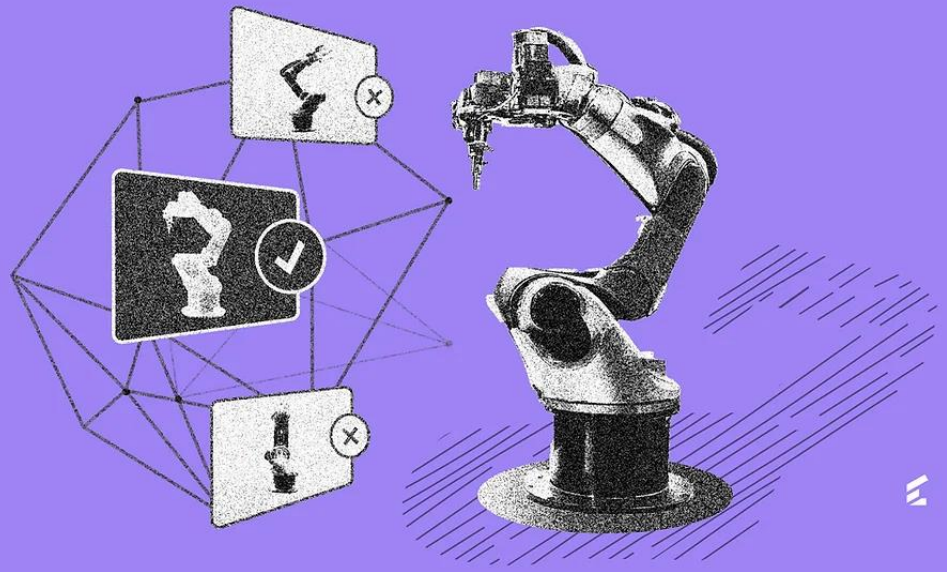
Boostcamp Robotics Study

# DeepIM: Deep Iterative Matching for 6D Pose Estimation!

SOYUN CHO



# Index

# 1. Abstract

**1.1 6D Pose Estimation**

**1.2** Matching rendered images - input image

**1.3 DeepIM: Deep Iterative Matching**

# 2. Introduction

## 2.1 Fig. 1

## 2.1 Fig. 1

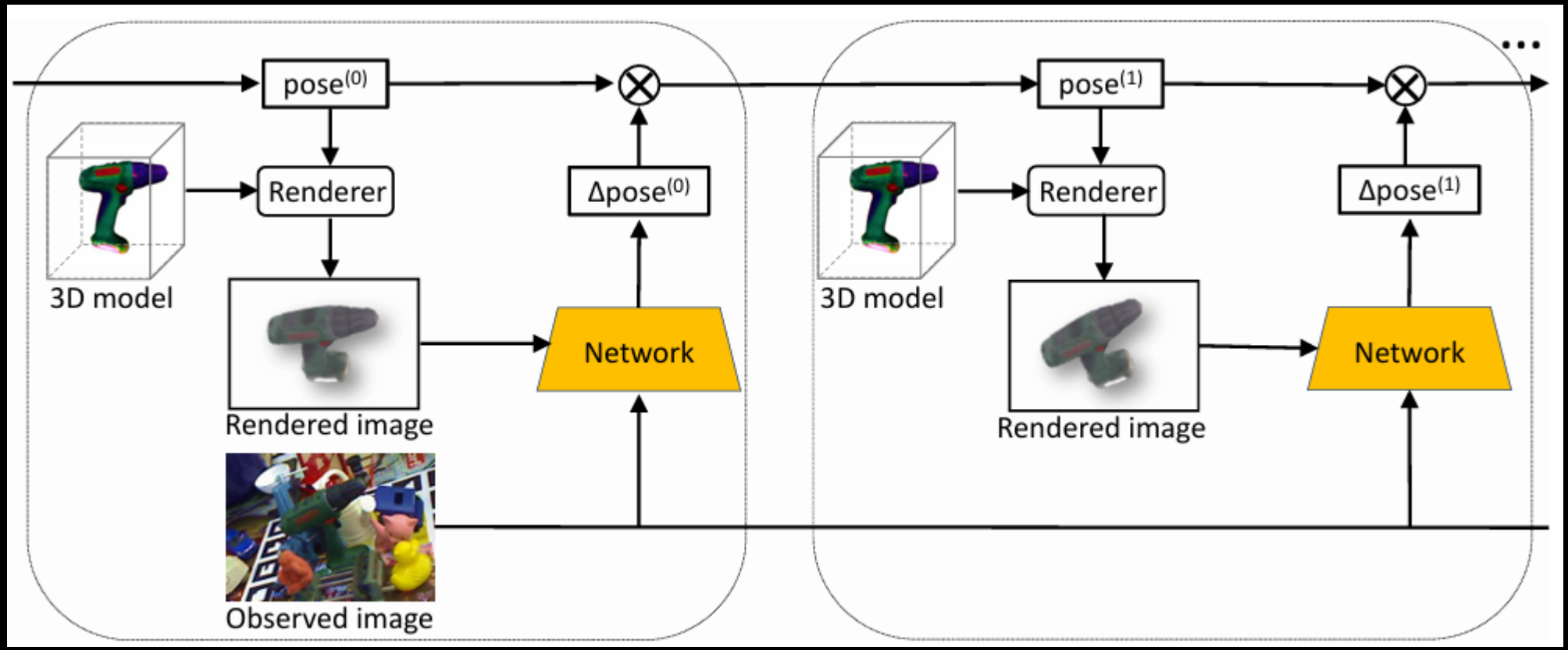



Fig. 1 illustrates the iterative matching procedure of our network for pose refinement.

## 2.1 Fig. 1

1. 이 네트워크가 상대 SE(3)변환을 예측하도록 학습함.

2. (PoseCNN등의 다른 포즈 추정 기법을 써서) 객체의 초기 6d 포즈 추정값을 주고, 객체의 3d 모델이 주어지면 -> Renderer가 객체의 렌더링된 이미지를 생성함.

3. 렌더링된 이미지와 관측된 이미지를 바탕으로 네트워크가 상대 SE(3)변환을 예측함 (=Δpose(0)).

4. 포즈를 업데이트(pose(1) = pose(0) * Δpose(0))하고 이를 반복함.

*SE(3)란?
: 3D 공간에서의 회전과 이동을 동시에 표현하는 수학적 구조
*상대 SE(3) 변환이란?
: 두 3D 좌표계 사이의 상대적인 위치를 표현하는 것. 즉, 초기 포즈과 관측된 이미지 사이의 상대적 회전 및 이동을 계산하는 것

## 3. Related work

### 3.1 RGB based 6D Pose Estimation

### 3.2 Depth based 6D Pose Estimation

### 3.3 RGB-D based 6D Pose Estimation

### 3.4 RGB vs RGB-D

# 3.1 RGB based 6D Pose Estimation

Goal: 추정 물체의 위치와 방향을 정확히 예측

- 전통적 방법:
  - 로컬 특징 매칭: 풍부한 텍스처 필요, 텍스처 없는 물체는 처리 어려움.
  - 템플릿 매칭: 텍스처 없는 물체 가능, 가림에 취약.

- 딥러닝 기반:
  - 키포인트 탐지 및 PnP 해결.
  - 물체 탐지 후 분류/회귀로 포즈 추정.
  - 오토인코더 기반 벡터 매핑.

  -> textureless, occlusion 문제 해결! 전통적 방법보다 높은 성능 제공.

# 3.2 Depth based 6D Pose Estimation

Goal: 두 포인트 클라우드의 정렬을 통해 물체의 6D 포즈를 추정
= geometric registration(기하학적 등록 문제)

- 접근법:
  - Local Refinement: 초기 포즈 필요, 로컬 미니멈 문제.
  - Global Registration: 초기 포즈 없이 가능, 계산 비용 높음.

- 딥러닝 기반:
  - 포인트 클라우드에 딥러닝 적용 → 포인트 특징 학습으로 성능 향상.

*Depth 데이터란?
: 각 픽셀에 대해 카메라와 물체 표면 사이의 거리를 나타내는 값으로 이루어진 이미지

# 3.3 RGB-D based 6D Pose Estimation

Goal: RGB와 깊이 데이터를 결합해 6D 포즈 추정 정확도 향상

•General Strategy:
- RGB 기반 초기 포즈 추정 → 깊이 기반 포즈 정제(ICP 등).

•Representative Studies:
- Hinterstoisser et al. (2012b): 템플릿 매칭 + ICP.
- Brachmann et al. (2014): 3D 좌표 회귀 + 최소자승법.
- PoseCNN (2018): 엔드 투 엔드 네트워크 + ICP.
- Wang et al. (2019): 반복적인 포즈 정제 네트워크.

*RGB-D 데이터란?
: RGB 데이터와 깊이 데이터를 동시에 포함하는 데이터

# 3.4 RGB vs RGB-D

**Performance Gap:**

- RGB 기반 방법은 여전히 RGB-D 기반 방법에 비해 성능 격차 존재.

**Challenges:**

- RGB 이미지만 사용하는 효과적인 포즈 정제 기법 부족.

**DeepIM의 Contribution:**

- 새로운 반복적 포즈 정제 네트워크.
- SE(3) 변환의 직접 회귀 및 기준 좌표계 사용.
- 기존 연구를 보완하며 새로운 물체 매칭 가능.

## 4. DeepIM Framework

### 4.1 High-resolution Zoom In

### 4.2 Network Structure

### 4.3 Disentangled Transformation Representation

### 4.4 Matching Loss

### 4.5 Training and Testing

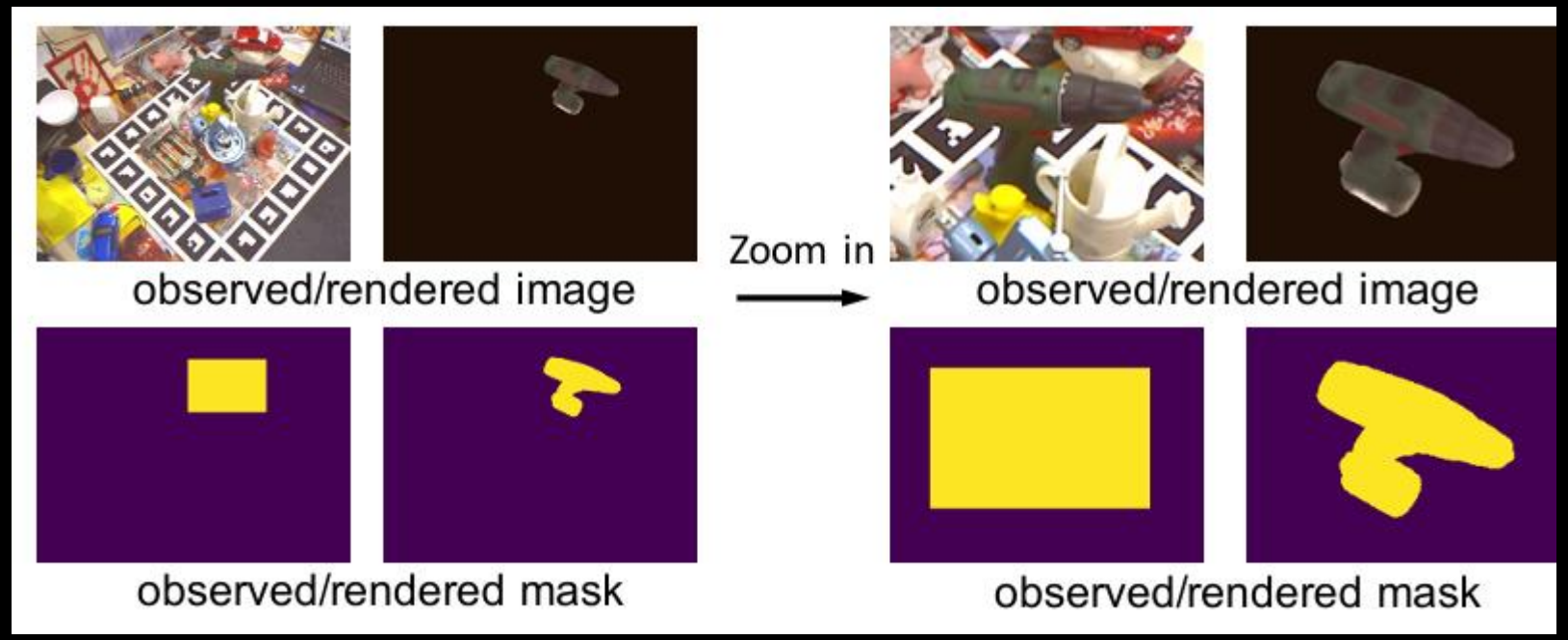
# 4.1 High-resolution Zoom In

**Problem: 물체가 작을 경우 유용한 특징 추출이 어려움.**
**Solution:**

- **전경 마스크와 확대된 바운딩 박스를 사용해 물체 영역을 잘라내고 업샘플링.**
- **종횡비 유지로 이미지 왜곡 방지.**

**Result:**

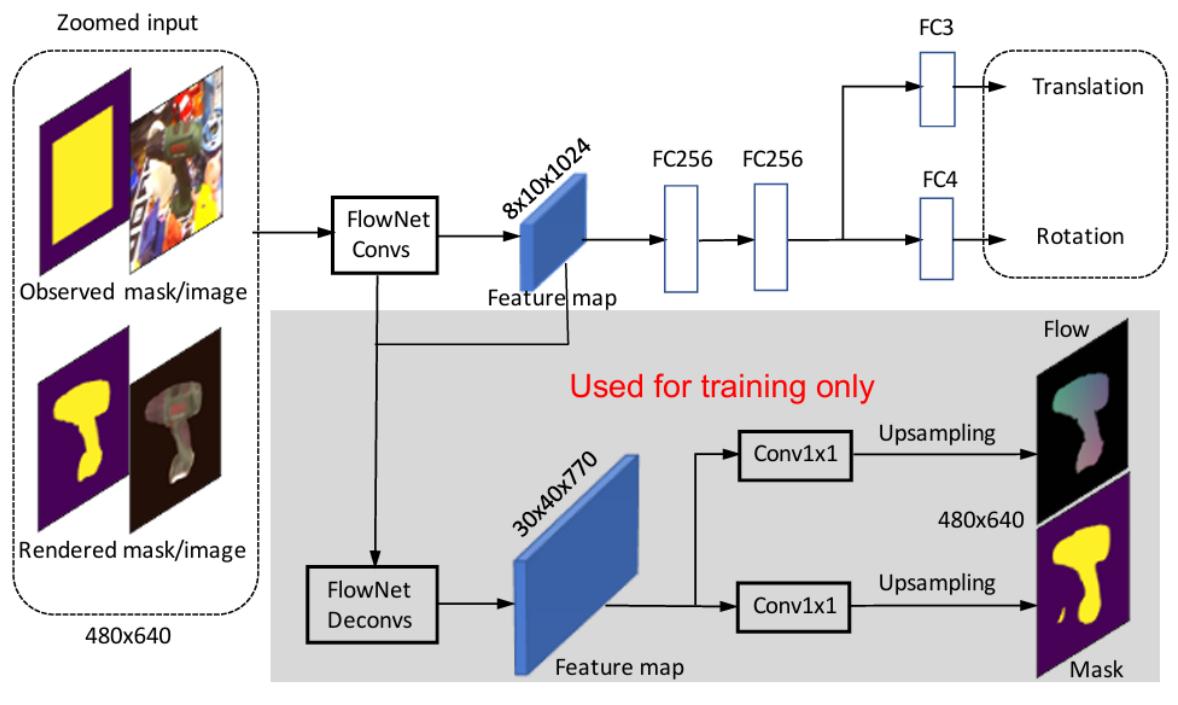
- **물체의 중심 정보와 디테일 확보 → 매칭 성능 향상.**




M rend : 렌더링 마스크
M obs : 관측된 마스크
u* d* l* r* : 관측된/렌더링된 마스크의 상단 하단 왼쪽 오른쪽 경계
Xc, Yc : 렌더링되ㄴ 이미지에서 객체 중심의 2d 정사영
R : 원래 이미지의 종횡비
ㅅ : 확대비율 (실험에선 1.4 고정)

$$
\begin{aligned}
x_{\text{dist}} &= \max(|l_{\text{obs}} - x_c|, |l_{\text{rend}} - x_c|, \\
&\qquad |r_{\text{obs}} - x_c|, |r_{\text{rend}} - x_c|), \\
y_{\text{dist}} &= \max(|u_{\text{obs}} - y_c|, |u_{\text{rend}} - y_c|, \\
&\qquad |d_{\text{obs}} - y_c|, |d_{\text{rend}} - y_c|), \\
\text{width} &= \max(x_{\text{dist}}, y_{\text{dist}} \cdot r) \cdot 2\lambda, \\
\text{height} &= \max(x_{\text{dist}}/r, y_{\text{dist}}) \cdot 2\lambda,
\end{aligned}
\tag{1}
$$

# 4.2 Network Structure




**Inputs: 관측 이미지, 렌더링 이미지, 관측/렌더링 마스크 (8채널)**
**FlowNetSimple Backbone:**
- **Optical Flow 기반으로 특징 추출 및 학습.**

**Pose Estimation Branch:**
- **3D 회전(Quaternion)과 3D 이동(Vector) 예측.**

**Auxiliary Branches:**
- **Optical Flow와 전경 마스크 예측으로 훈련 안정성 향상.**

*Optical Flow란?
: 두 연속된 이미지(프레임)간의 픽셀 이동방향과 속도를 나타내는 정보
*FlowNet이란?
: Optical Flow 문제를 해결하기 위한 Deep neural network 모델

# 4.3 Disentangled Transformation Representation

**Problem: 기존 표현은 회전과 이동이 상호작용하며, 물체 크기/거리 의존성이 높음.**

**Proposed Solution:**

- 회전: 카메라 축과 평행한 축 사용.
- 이동: 2D 이미지 기반 픽셀 이동 + 스케일 변화로 표현.

**Advantages:**

- 회전과 이동의 독립성 유지.
- 물체 크기와 거리와 무관한 학습 가능.
- 일반화 가능성 향상.

$$\mathbf{R}_{\text{tgt}} = \mathbf{R}_{\Delta}\mathbf{R}_{\text{src}}, \quad \mathbf{t}_{\text{tgt}} = \mathbf{R}_{\Delta}\mathbf{t}_{\text{src}} + \mathbf{t}_{\Delta}, \tag{2}$$

$$\begin{aligned} v_x &= f_x(x_{\text{tgt}}/z_{\text{tgt}} - x_{\text{src}}/z_{\text{src}}), \\ v_y &= f_y(y_{\text{tgt}}/z_{\text{tgt}} - y_{\text{src}}/z_{\text{src}}), \\ v_z &= \log(z_{\text{src}}/z_{\text{tgt}}), \end{aligned} \tag{3}$$

[RΔ|tΔ] : 상대적 회전 및 이동
소스객체포즈src
변환된타겟포즈tgt
단순표현; tΔ=(Δx,Δy,Δz)=ttgt−tsrc
Vx Vy : 객체가 이미지 x y축을 따라 몇 픽셀 이동해야 하는지
Vz : 객체의 크기 변화

# 4.3 Disentangled Transformation Representation

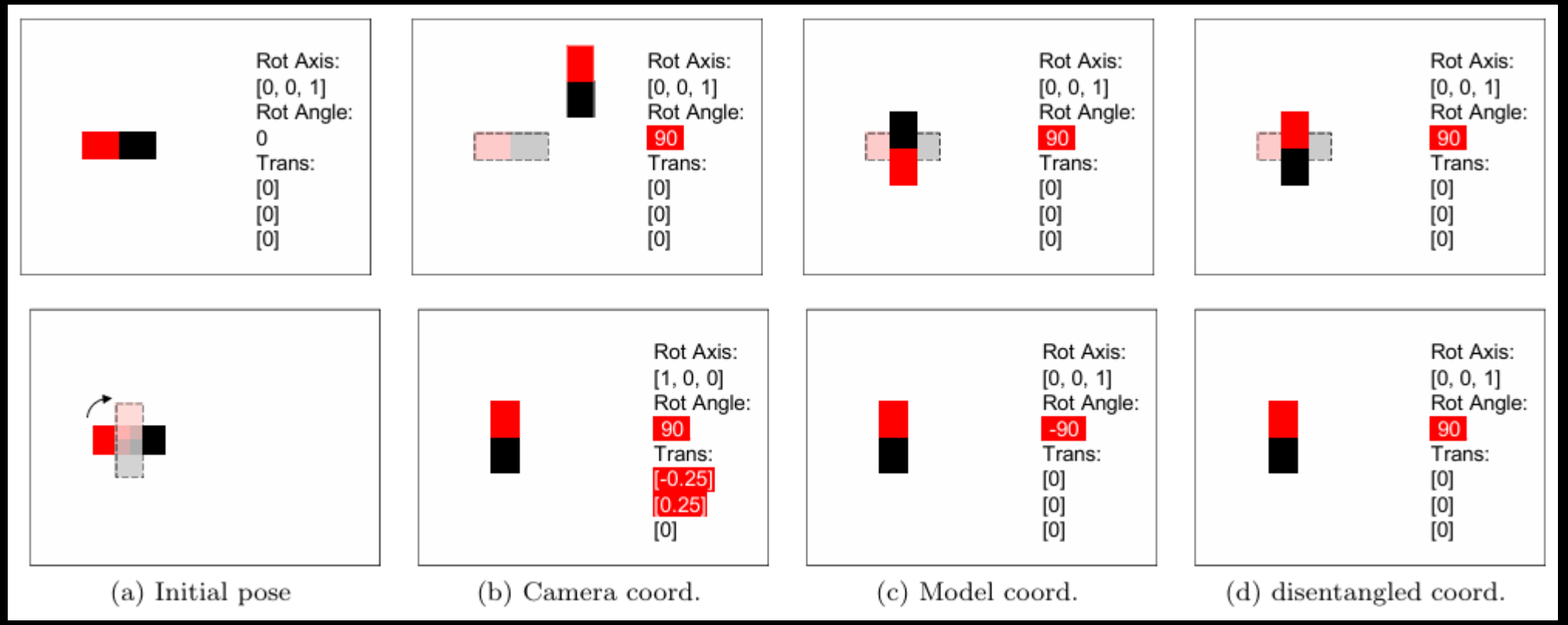
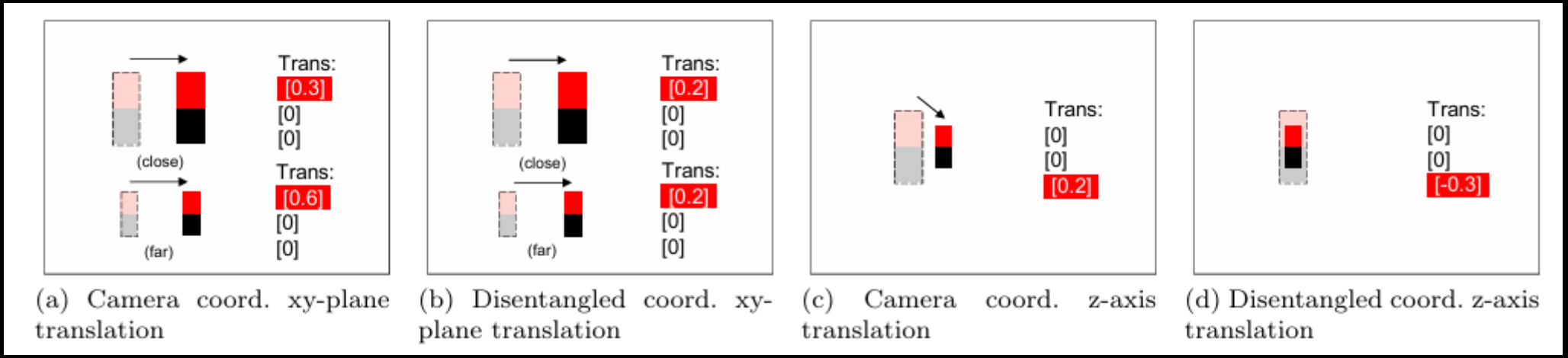



Fig. 4: 좌표계별 회전 차이

# 4.3 Disentangled Transformation Representation




(a) Camera coord. xy-plane translation (b) Disentangled coord. xy-plane translation (c) Camera coord. z-axis translation (d) Disentangled coord. z-axis translation

Fig. 5: 이동 표현 차이와 분리된 표현의 장점

# 4.4 Matching Loss

**Problem:** 회전과 이동 손실의 균형 조정이 어려움.

**Solution: Point Matching Loss**

- 3D 포인트를 변환 후 L1 거리로 평가.
- 이상치에 강건하고 훈련 안정성 보장.

**Result:**

- 정답 포즈와 추정된 포즈 간의 정확한 3D 매칭 평가.

$$L_{\text{pose}}(\mathbf{p}, \hat{\mathbf{p}}) = \frac{1}{n} \sum_{i=1}^{n} \|(\mathbf{R}\mathbf{x}_i + \mathbf{t}) - (\hat{\mathbf{R}}\mathbf{x}_i + \hat{\mathbf{t}})\|_1, \qquad (4)$$

정답 포즈 p = [R|t]
추정된 포즈 ^p = [ ^R|^t]

# 4.5 Training and Testing

**Training:**

- 노이즈 추가로 초기 포즈 생성 → 상대 변환 학습.
- 반복적 데이터 생성으로 테스트와 유사한 데이터 분포 학습.

**Testing:**

- 초기 추정값이 부정확해도 반복적 정제를 통해 성능 향상.
- 관측 이미지와 유사도가 점진적으로 증가.

**Result:**

- 반복적 훈련으로 테스트에서도 높은 정확도 달성.

# 5. Experiments

## 5.1 Purpose

## 5.2 Datasets

## 5.3 Results

## 6. Conclusion

### 6.1 DeepIM

### 6.2 Results

### 6.3 Future Directions

# 6. Conclusion

**DeepIM:**

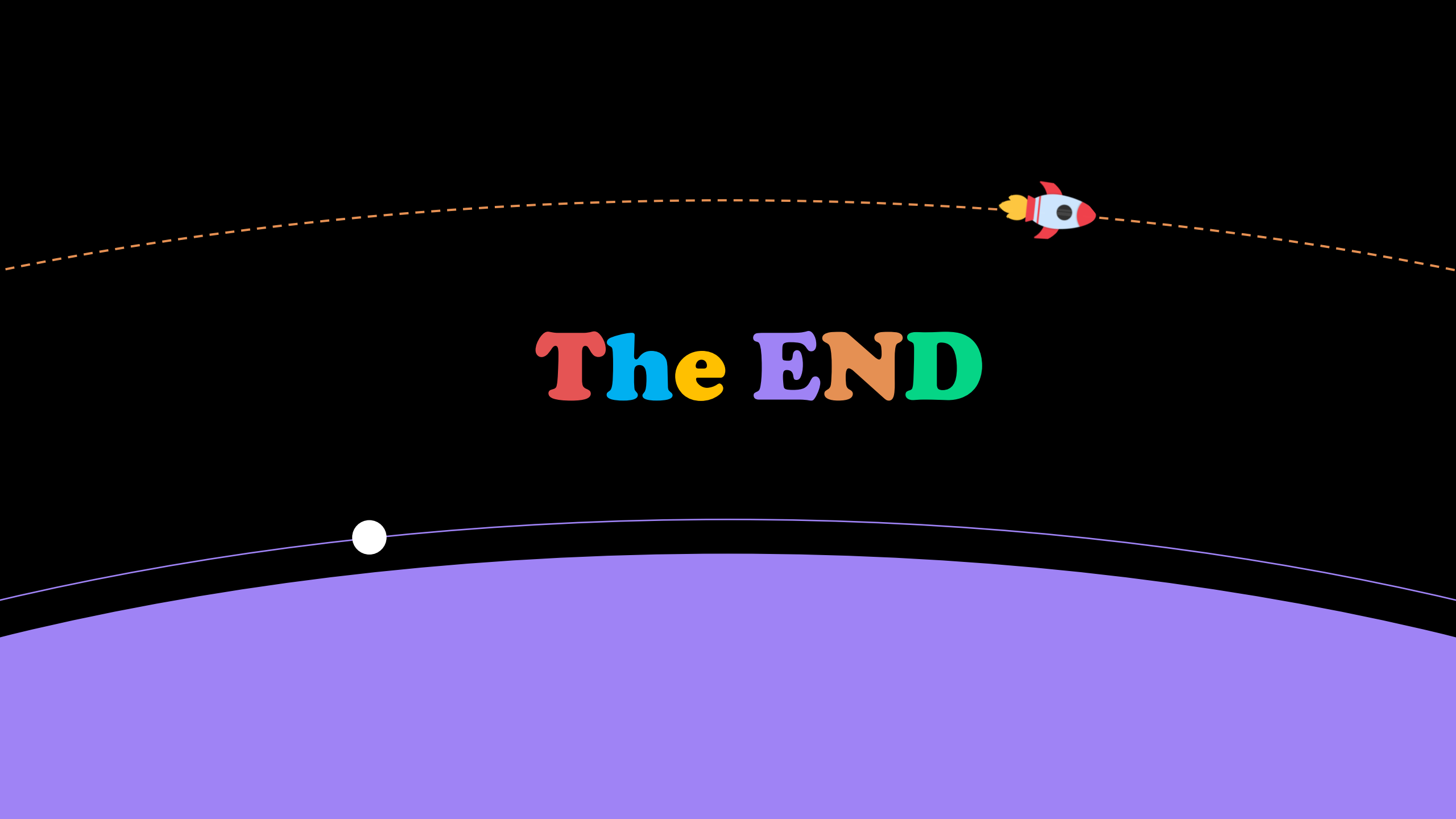
- 반복적 포즈 매칭 프레임워크, 컬러 이미지만 사용.
- 상대적인 포즈 변환을 출력하는 딥 뉴럴 네트워크 설계.

**Results:**

- 최신 RGB 기반 방법보다 우수한 성능.

**Future Directions:**

- 스테레오 확장으로 더 높은 정확도 기대.
- 로봇 조작 등 응용 가능성 확대.


The END