BB8 : A Scalabel, Accurate, Robust to Partial Occlusion Method for Predicting the 3D Poses of Challenging Objects without Using Depth(ICCV 2017)

KIST

송명하

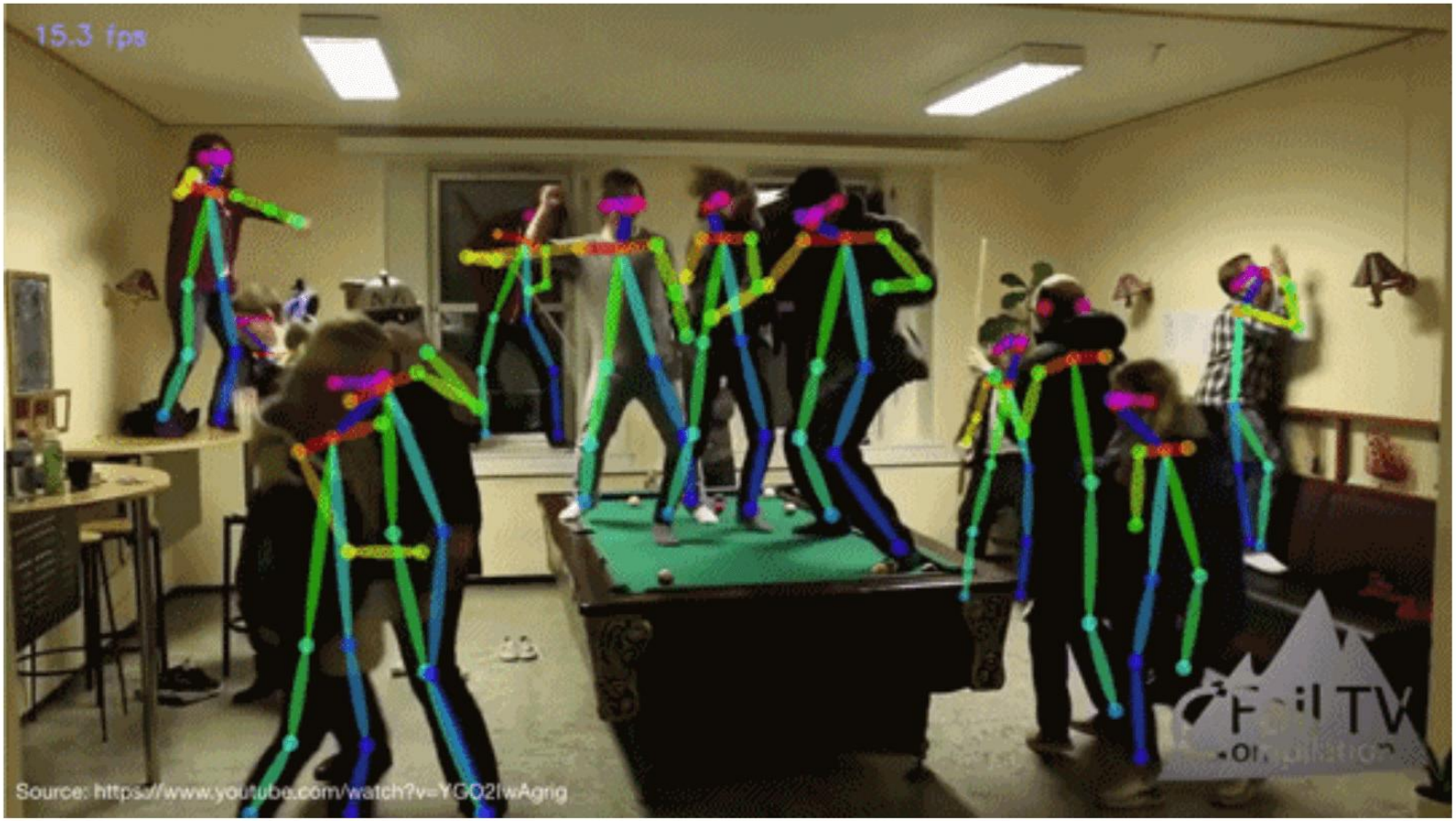
Content

1. Introduction(What is Object Pose Estimation)
2. Related Work
3. Proposed Method
4. Experiments
5. Conclusion

What is Object Pose Estimation

아 Pose 그거 ㅋㅋ

이거잖아ㅋㅋ



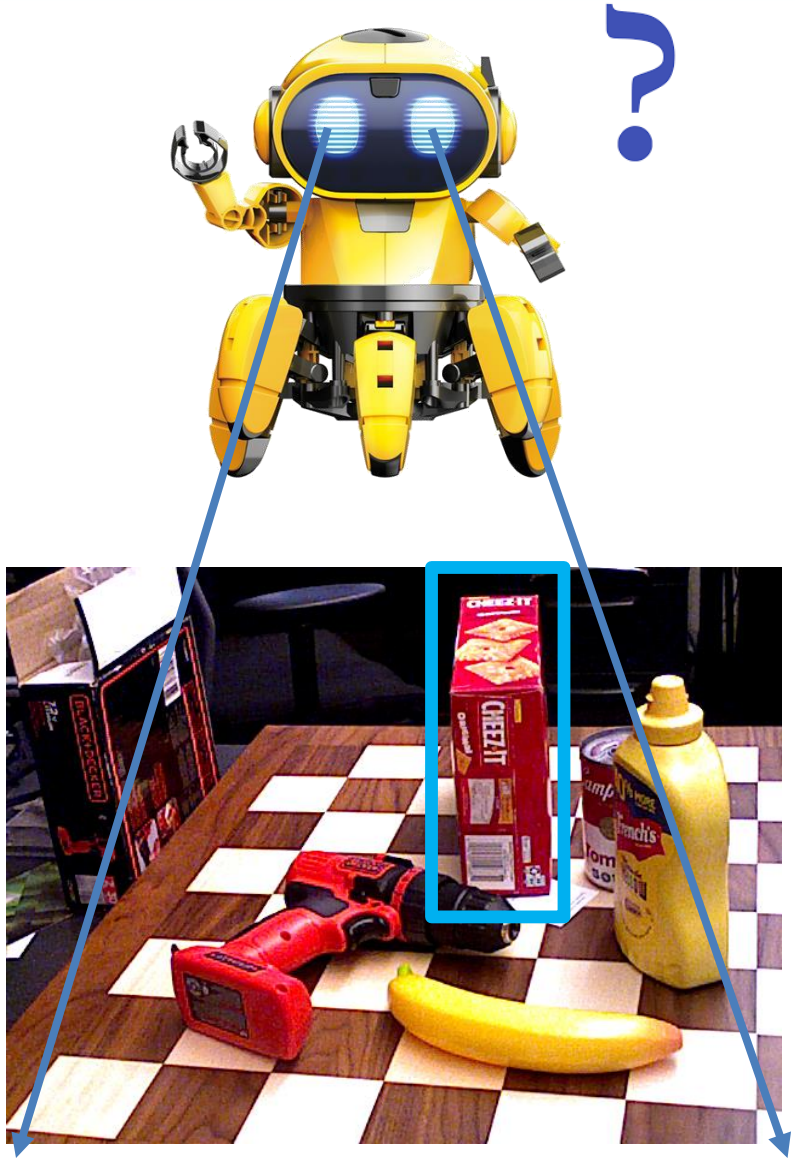
What is Object Pose Estimation

What is Object Pose Estimation

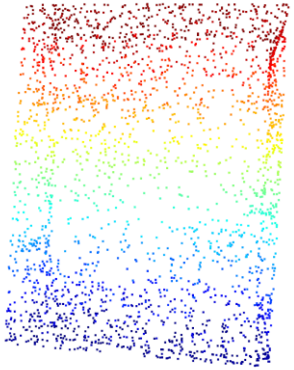
What is Object Pose Estimation

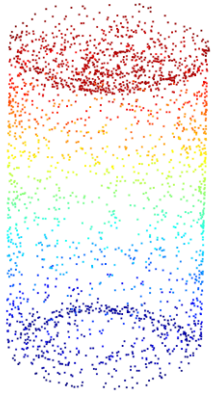
What is Object Pose Estimation

What is Object Pose Estimation

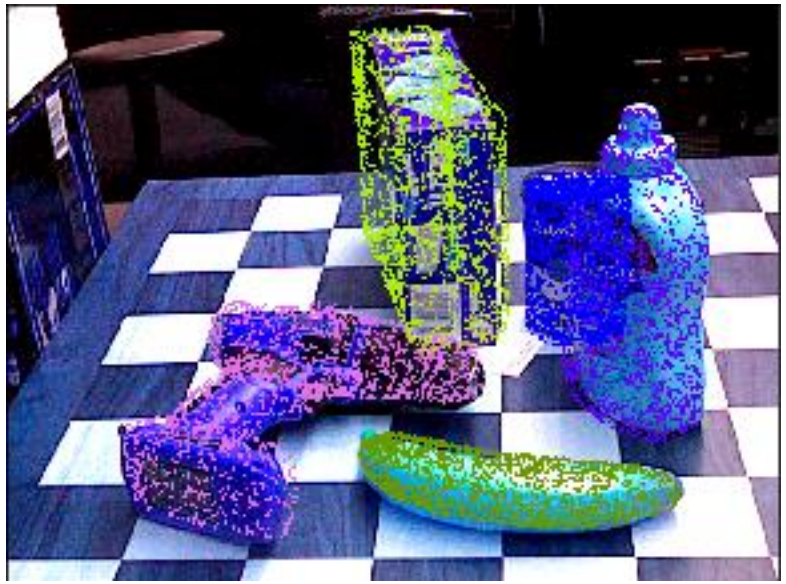


What is Object Pose Estimation





```
array([[[-0.09398018,  0.64188197, -0.80094869, -0.82608921],
        [ 0.9402408 ,  0.75827657, -0.58462179,  0.54925076],
        [ 0.32728237,  0.11409544, -0.12922327,  0.12609354],
        [ 0.02148714,  0.14903675, -0.0668179 ,  0.02688122]],
```



```
[[-0.71619143,  0.52527327,  0.50278352,  0.49635643],
 [-0.2921976 , -0.54315395, -0.53955106,  0.60320668],
 [ 0.63379064,  0.6550358 , -0.67534741,  0.62431801],
 [ 0.78362206,  0.9398721 ,  0.89724764,  0.81292044]]]
```

1. 포즈를 찾는 것.

2. Occlusion

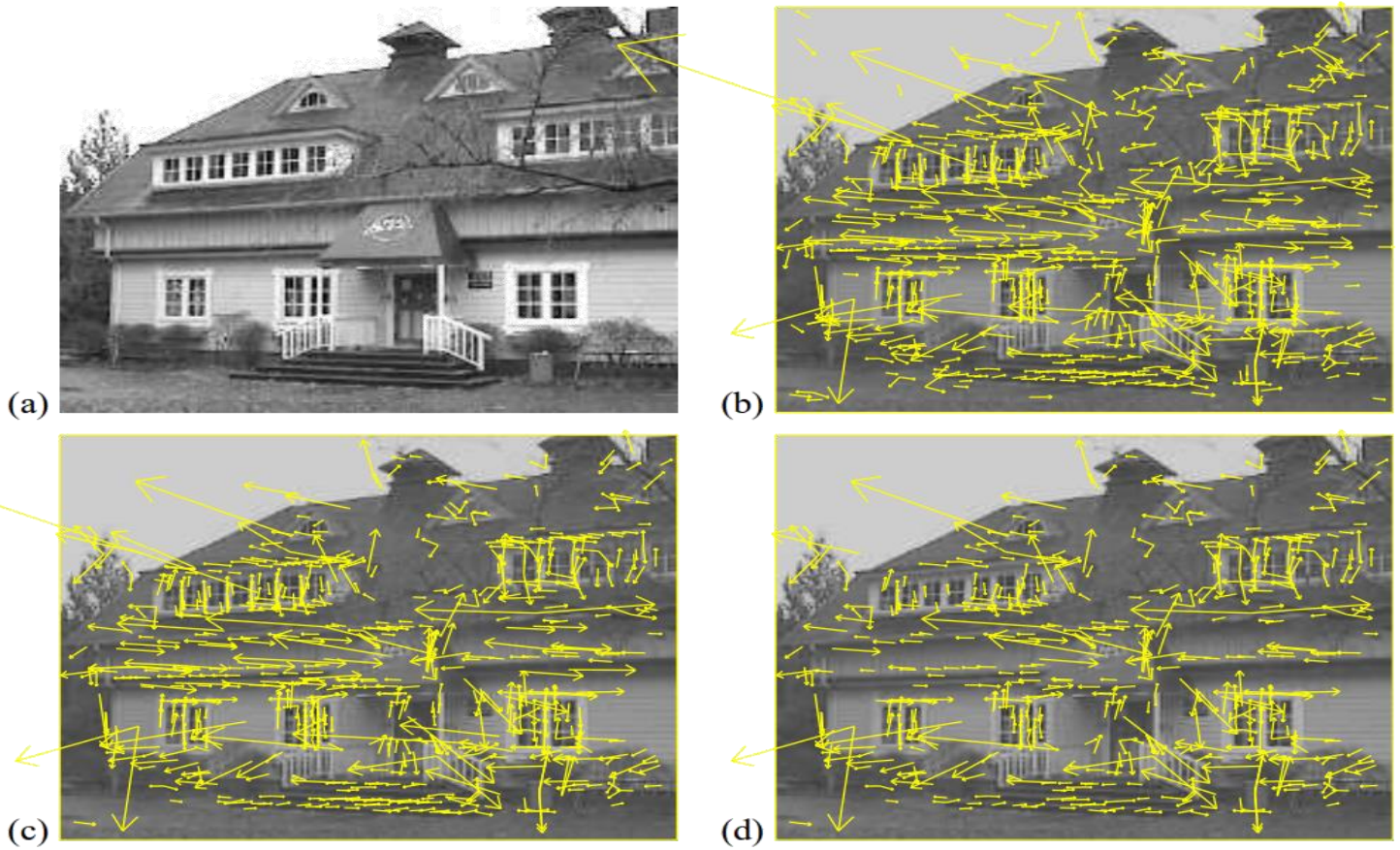
3. Symmetric Object

2. Occlusion

2. Occlusion

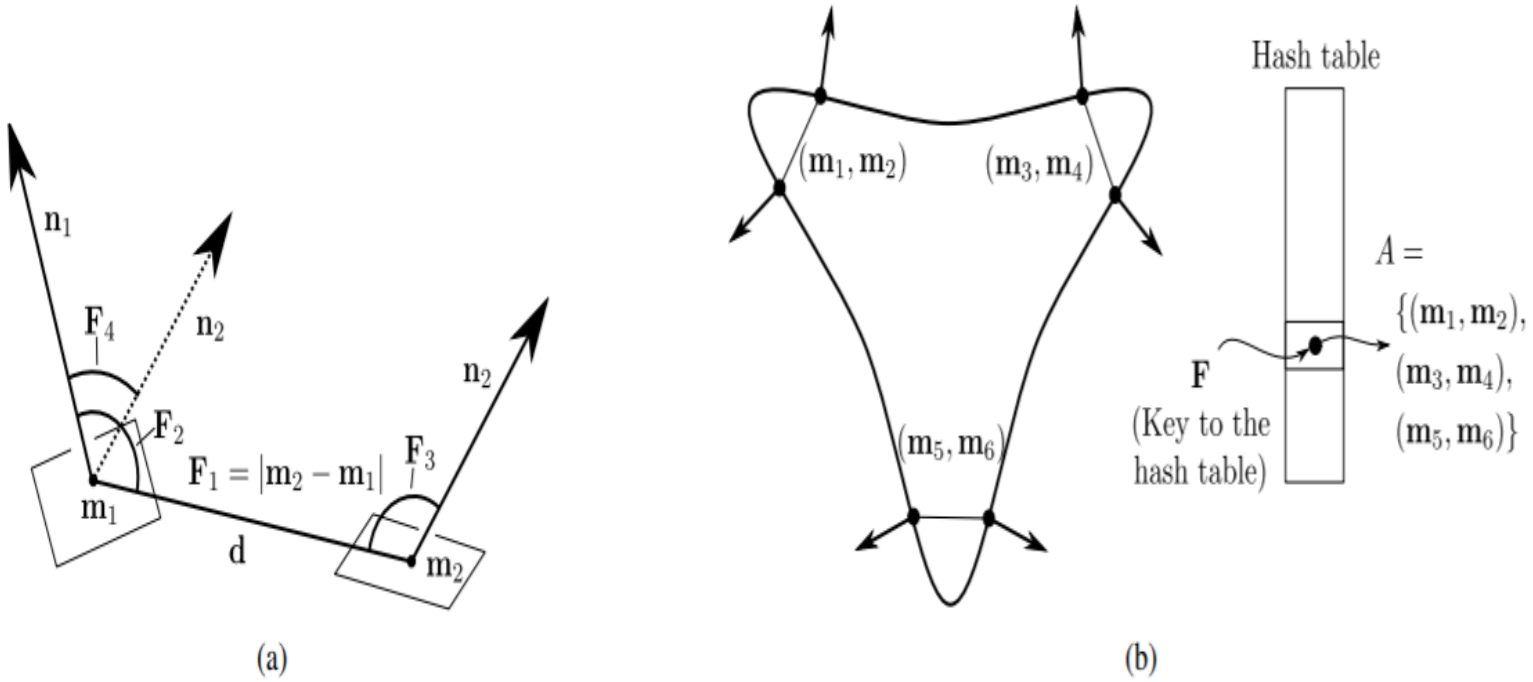
3. Symmetric Object

Related work



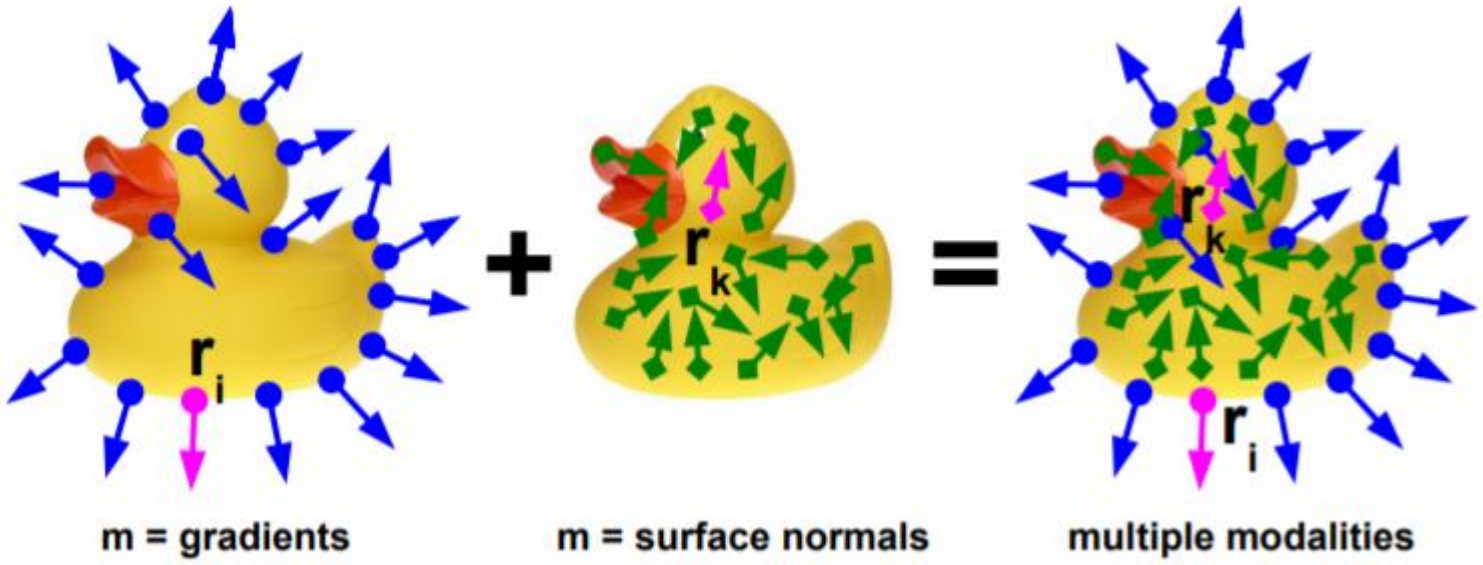
Distinctive Image Features from Scale-Invariant Keypoints(2004)

Related work



Model Globally, Match Locally: Efficient and Robust 3D Object Recognition (CVPR2010)

Related work



Multimodal Templates for Real-Time Detection of Texture-less Objects in Heavily Cluttered Scenes(ICCV2011)

Related work

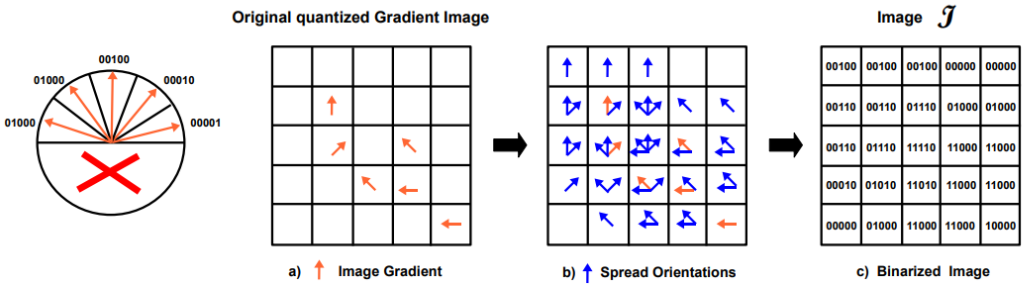
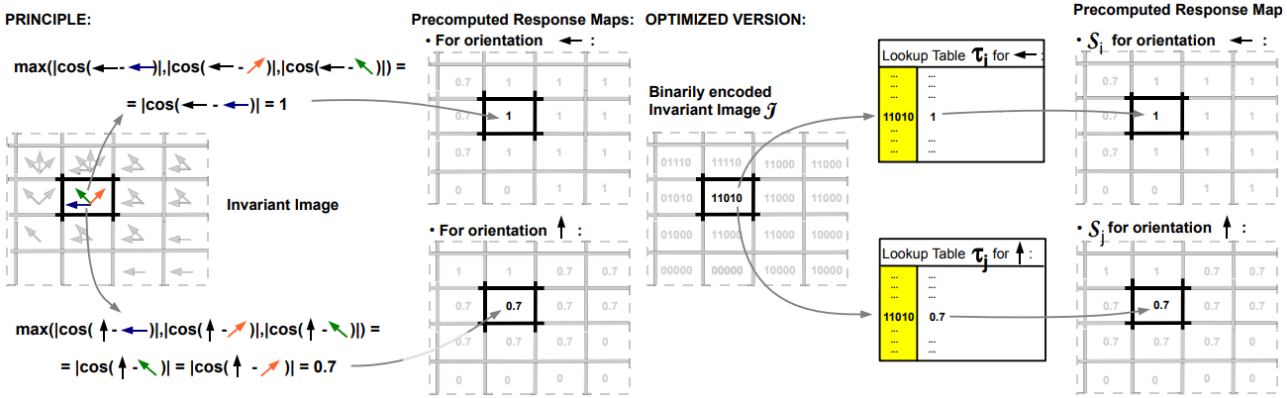


Fig. 4: Spreading the gradient orientations. **Left:** The gradient orientations and their binary code. We do not consider the direction of the gradients. **a)** The gradient orientations in the input image, shown in orange, are first extracted and quantized. **b)** Then, the locations around each orientation are also labeled with this orientation, as shown by the blue arrows. This allows our similarity measure to be robust to small translations and deformations. **c)** $\mathcal{J}$ is an efficient representation of the orientations after this operation, and can be computed very quickly. For this figure, $T = 3$ and $n_o = 5$. In practice, we use $T = 8$ and $n_o = 8$.



Gradient Response Maps for Real-Time Detection of Texture-Less Objects (PAMI 2012)

Related work

2가지 방법으로 보편화…

1) 8개의 3D Bounding Box 꼭지점을 찾고 PnP algorithm

2) Directly predict 6D(3D rotation + 3D translation)

Related work

2가지 방법으로 보편화…

1) 8개의 3D Bounding Box 꼭지점을 찾고 PnP algorithm

2) Directly predict 6D(3D rotation + 3D translation)

Proposed Method

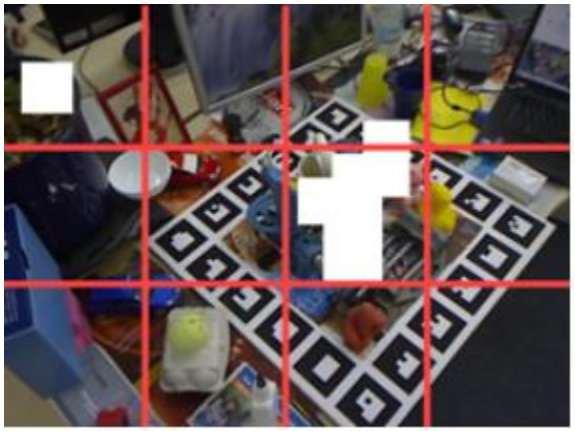
1. Localizing the Objects in 2D

2. Predicting the 3D Pose

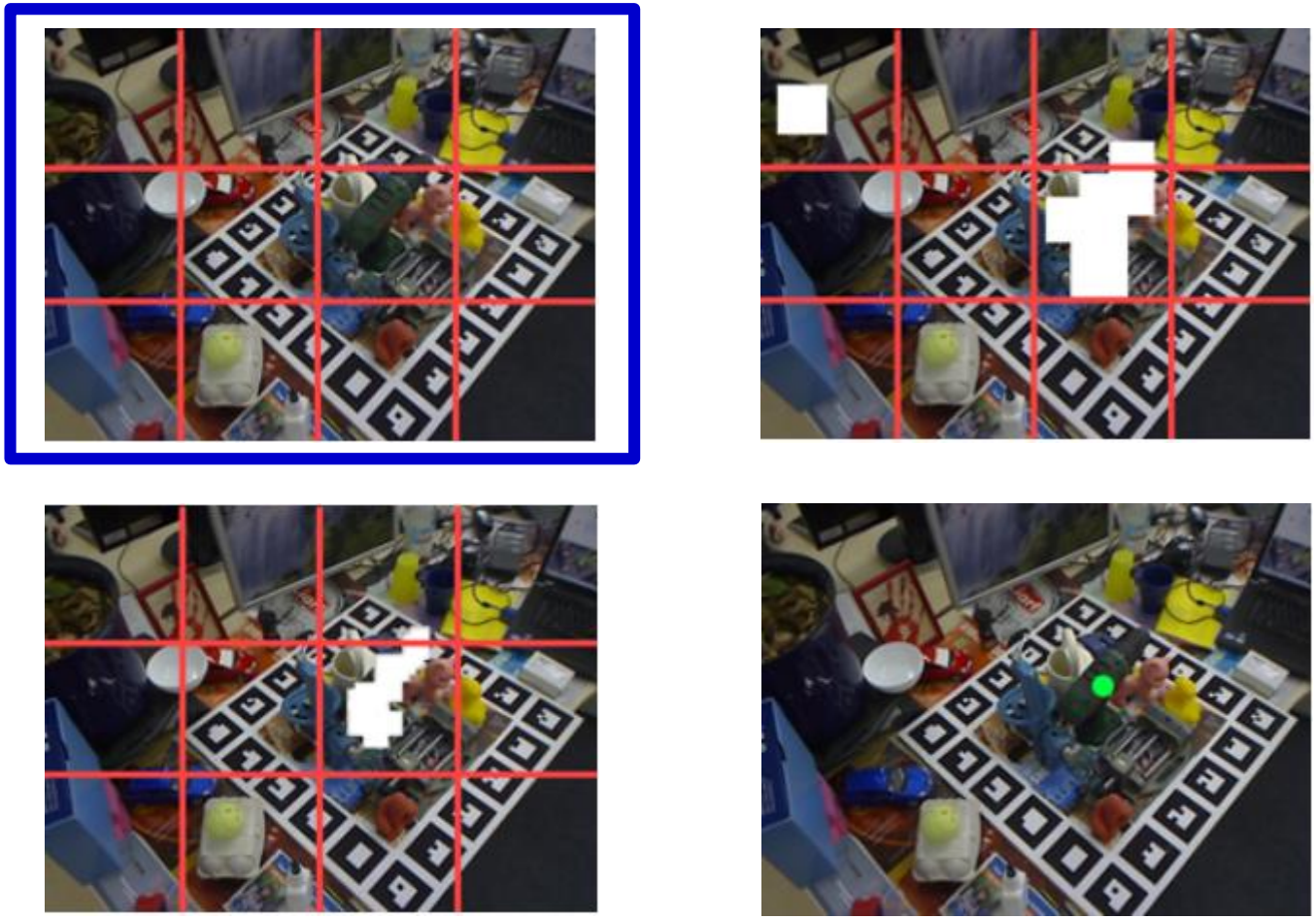
3. Handling Objects with an Axis of Symmetry

4. Refining the Pose

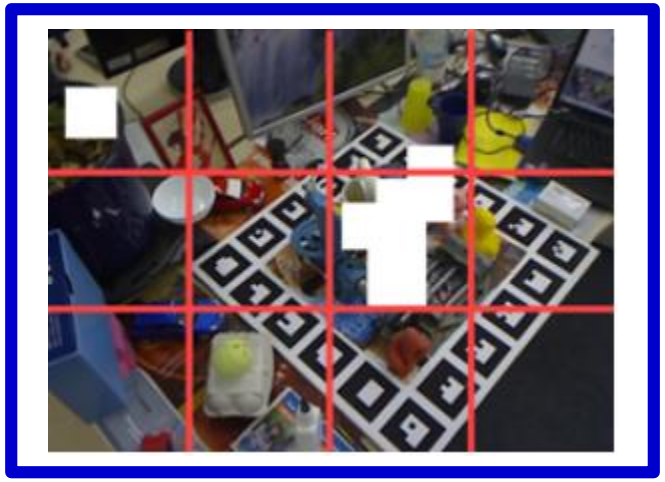
Proposed Method – Localizing the Objects in 2D



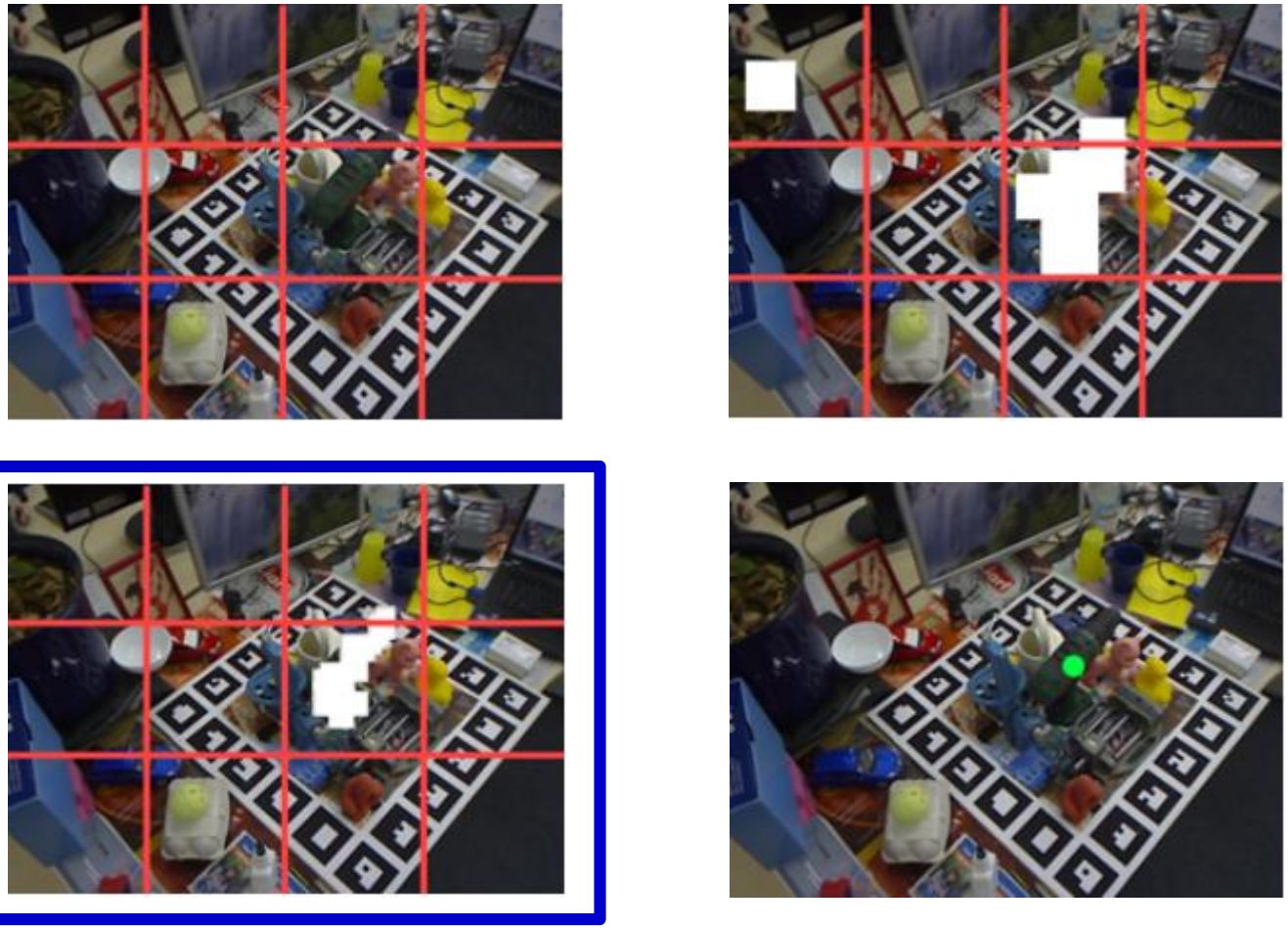
Proposed Method – Localizing the Objects in 2D



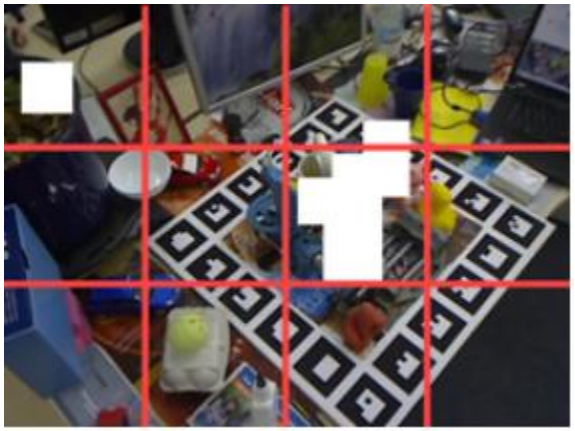
Proposed Method – Localizing the Objects in 2D

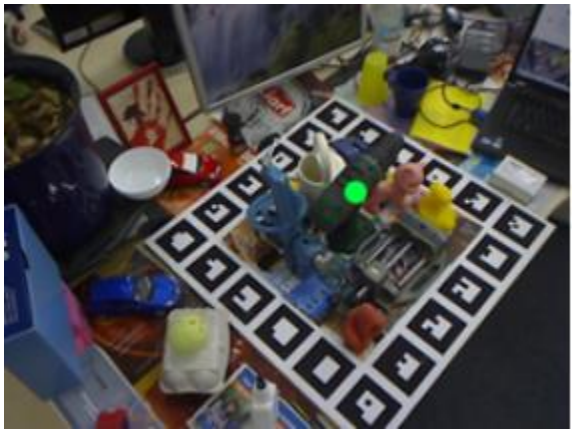


Proposed Method – Localizing the Objects in 2D

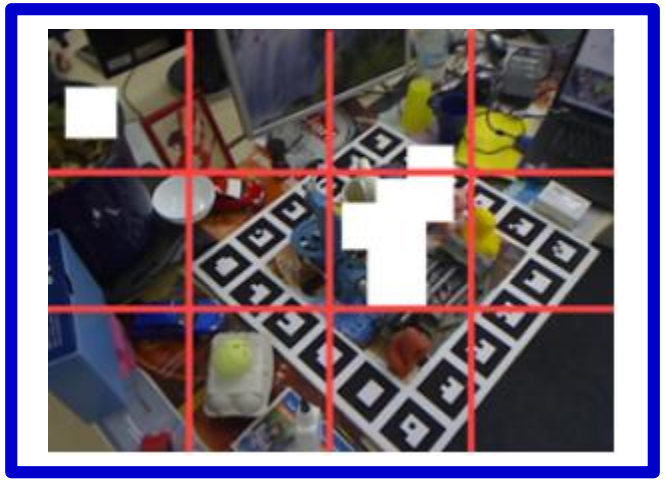


Proposed Method – Localizing the Objects in 2D

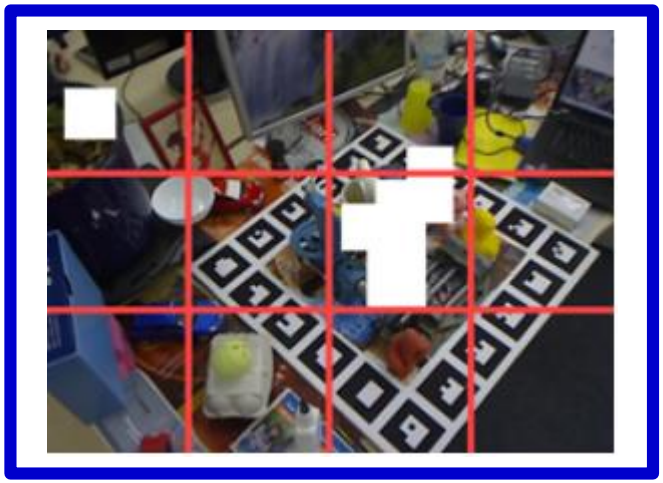




Proposed Method – Localizing the Objects in 2D

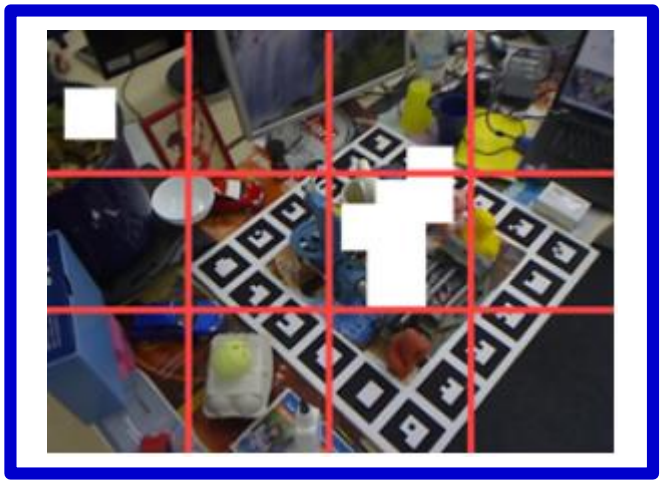


Proposed Method – Localizing the Objects in 2D



VGG에서 FC빼고 필요한 만큼의 output을 설정한 후 fine-tune;

Proposed Method – Localizing the Objects in 2D



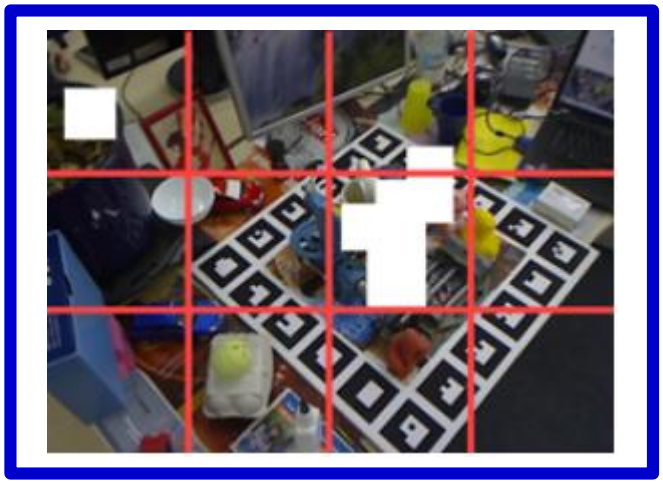
$$\sum_{(J,S,o)\in\mathcal{T}_s} \|(f_\phi^1(J))[o] - S\|^2$$

$\mathcal{T}_s$ **Training set made of image region**

$(f_\phi^1(J))[o]$ **The output of network**

S **Curresponding segmentations for object o**

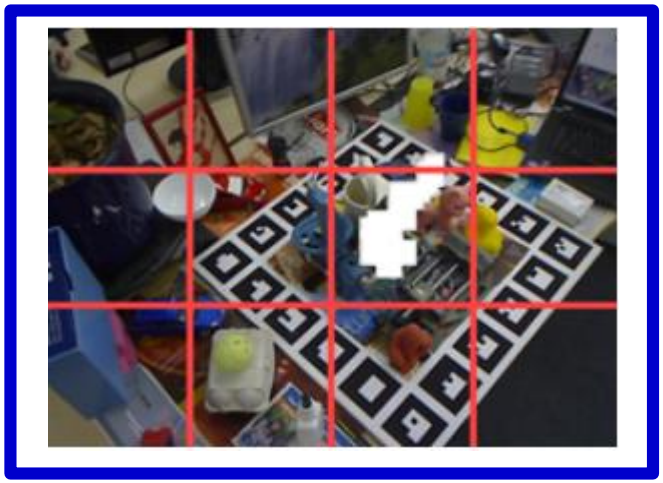
Proposed Method – Localizing the Objects in 2D



$$\sum_{(J,S,o)\in\mathcal{T}_s} \|(f_\phi^1(J))[o] - S\|^2$$

$$s_{1,o}(J) = (f_\phi^1(J))[o] > \tau_1 \ ,$$

Proposed Method – Localizing the Objects in 2D



$$s_{2,o}(P) = (f_{\psi}^{2}(P))[o] > \tau_2 \ ,$$

Two Convolutional layers and 2 Pooling layers

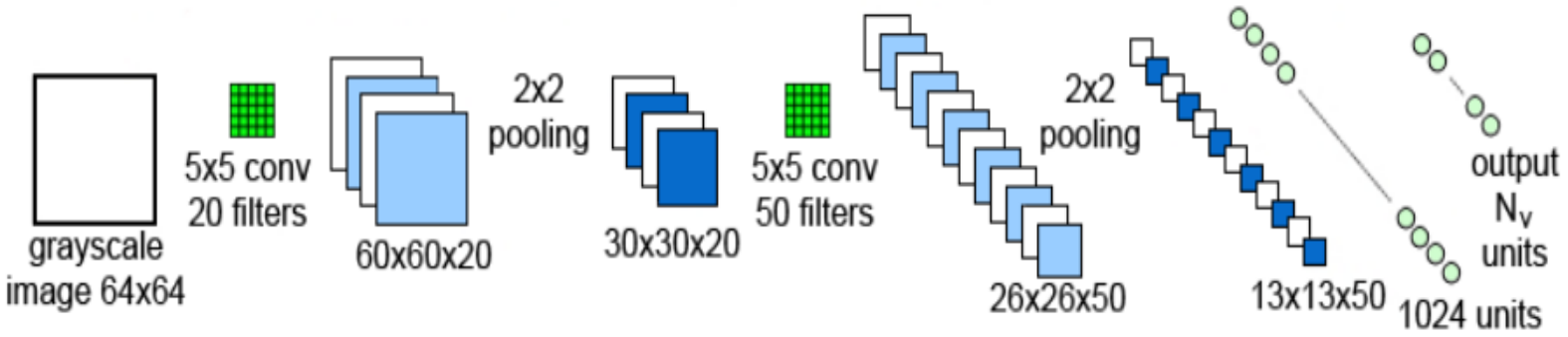
Proposed Method – Localizing the Objects in 2D

$$s_{2,o}(P) = (f_{\psi}^{2}(P))[o] > \tau_2 \ ,$$

Two Convolutional layers and 2 Pooling layers

Proposed Method – Predicting the 3D Pose

$$\sum_{(W,\mathbf{e},\mathbf{t},o)\in\mathcal{T}} \sum_{i} \|\mathrm{Proj}_{\mathbf{e},\mathbf{t}}(\mathbf{M}_i^o) - m_i((g_\Theta(W))[o])\|^2 ,$$



A Novel Representation of Parts for Accurate 3D Object Detection and Tracking in Monocular Images(2015)

CNN

Proposed Method – Predicting the 3D Pose

$$\sum_{(W,\mathbf{e},\mathbf{t},o)\in\mathcal{T}} \sum_{i} \left\| \mathrm{Proj}_{\mathbf{e},\mathbf{t}}(\mathbf{M}_i^o) - m_i((g_\Theta(W))[o]) \right\|^2 ,$$

$\mathcal{T}$ is a training set made of image windows W

$\mathbf{M}_i^o$ are the 3D coordinates of the corners of the bounding box of object o in t he object coordinate system

$\mathrm{Proj}_{\mathbf{e},\mathbf{t}}(\mathbf{M}_i^o)$ Projects the 3D point M on the image from the pose defined by e and t.

$m_i((g_\Theta(W))[o])$ Returns the two components of the output of g_Θ Corresponding to the predicted 2D coordinates of the i-th corner for object

Proposed Method – Predicting the 3D Pose

P3P [edit]

When $n = 3$, the PnP problem is in its minimal form of P3P and can be solved with three point correspondences. However, with just three point correspondences, P3P yields up to four real, geometrically feasible solutions. For low noise levels a fourth correspondence can be used to remove ambiguity. The setup for the problem is as follows.

Let P be the center of projection for the camera, A, B, and C be 3D world points with corresponding images points u, v, and w. Let $X = |PA|$, $Y = |PB|$, $Z = |PC|$, $\alpha = \angle BPC$, $\beta = \angle APC$, $\gamma = \angle APB$, $p = 2\cos\alpha$, $q = 2\cos\beta$, $r = 2\cos\gamma$, $a' = |AB|$, $b' = |BC|$, $c' = |AC|$. This forms triangles PBC, PAC, and PAB from which we obtain a sufficient equation system for P3P:

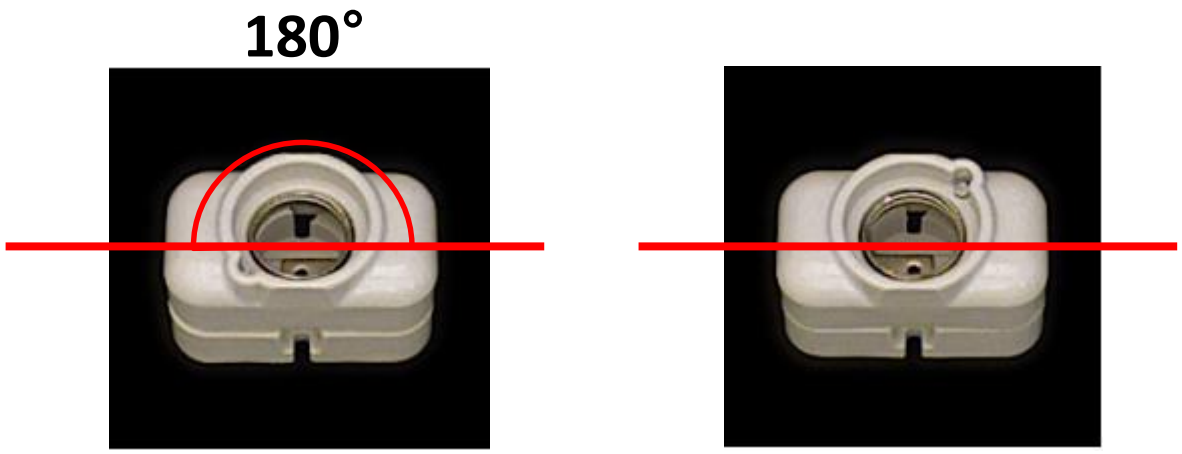
$$\begin{cases} Y^2 + Z^2 - YZp - b'^2 = 0 \\ Z^2 + X^2 - XZq - c'^2 = 0 \\ X^2 + Y^2 - XYr - a'^2 = 0 \end{cases}.$$

Solving the P3P system results in up to four geometrically feasible real solutions for R and T. The oldest published solution dates to 1841[5]. A recent algorithm for solving the problem as well as a solution classification for it is given in the 2003 *IEEE Transactions on Pattern Analysis and Machine Intelligence* paper by Gao, et al.[6] An open source implementation of Gao's P3P solver can be found in OpenCV's *calib3d* module in the *solvePnP* function.[7] Several faster and more accurate versions have been published since, including Lambda Twist P3P [8] which achieved state of the art performance in 2018 with a 50 fold increase in speed and a 400 fold decrease in numerical failures. Lambdatwist is available as open source in OpenMVG and at https://github.com/midjji/pnp.

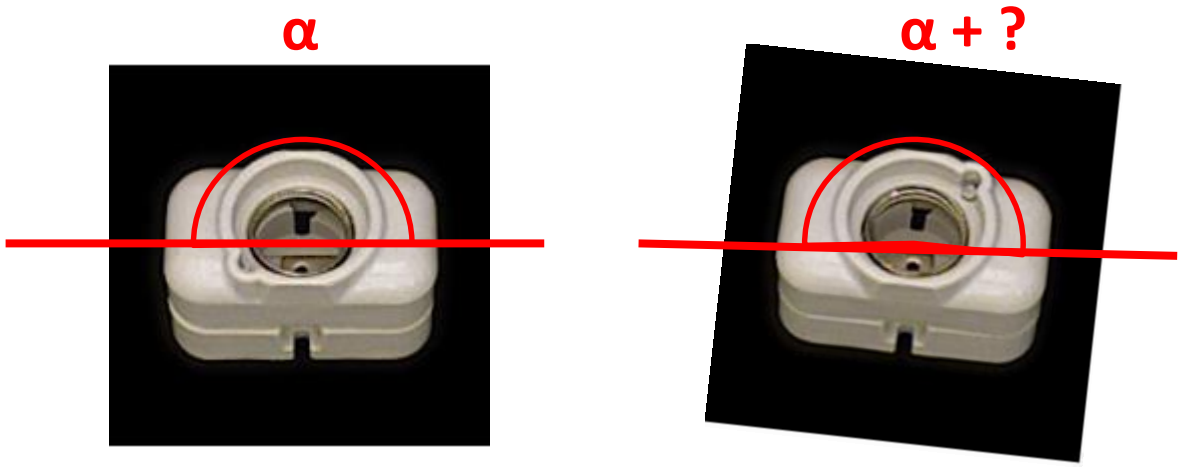
Proposed Method – Handling Objects with an Axis of Symmetry

Proposed Method – Handling Objects with an Axis of Symmetry

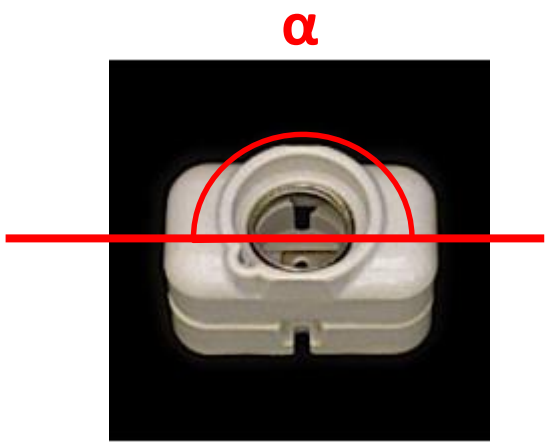
Proposed Method – Handling Objects with an Axis of Symmetry



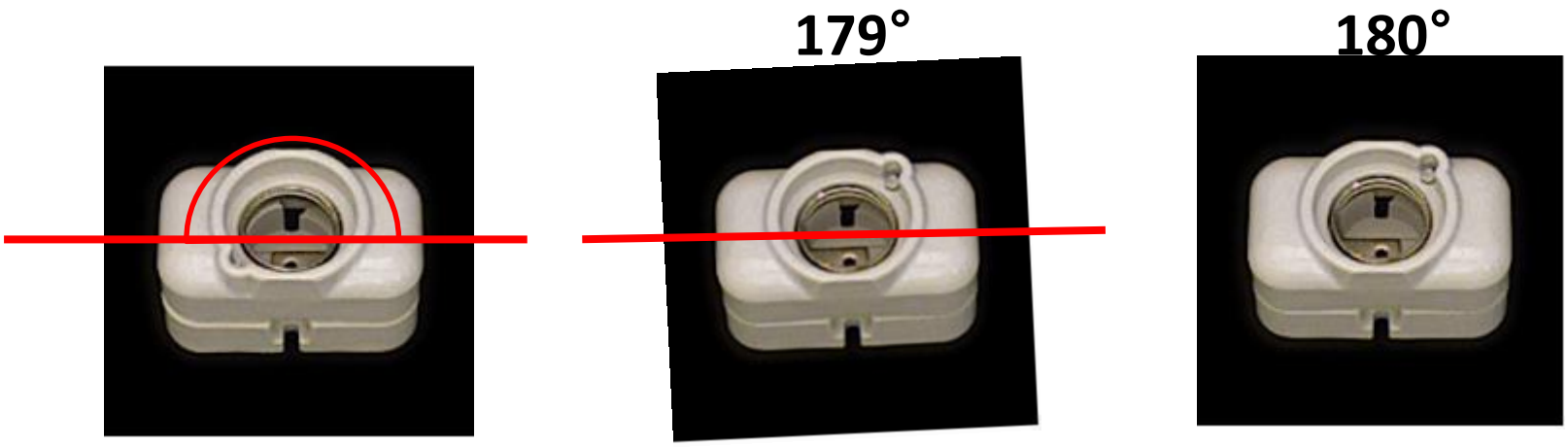
Proposed Method – Handling Objects with an Axis of Symmetry



Proposed Method – Handling Objects with an Axis of Symmetry



Proposed Method – Handling Objects with an Axis of Symmetry



Proposed Method – Handling Objects with an Axis of Symmetry

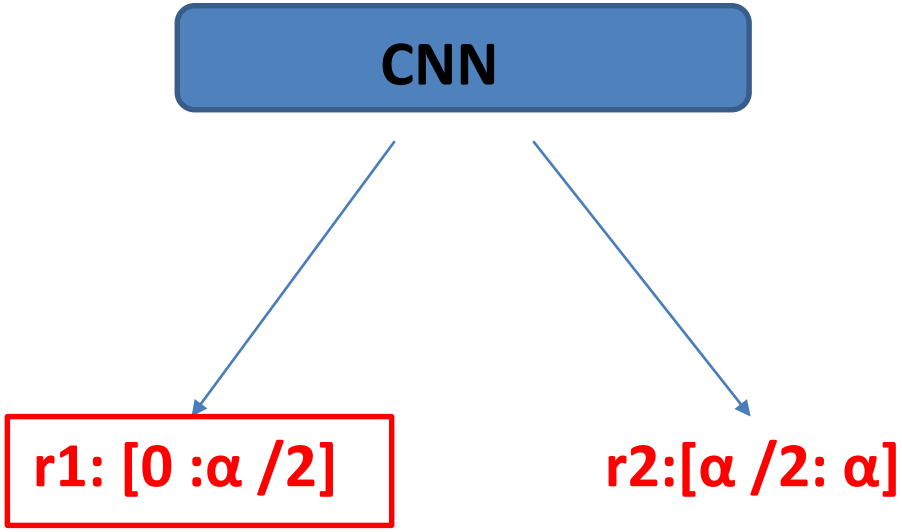
β : rotation angle

r1: [0 :α /2]

r2:[α /2: α]

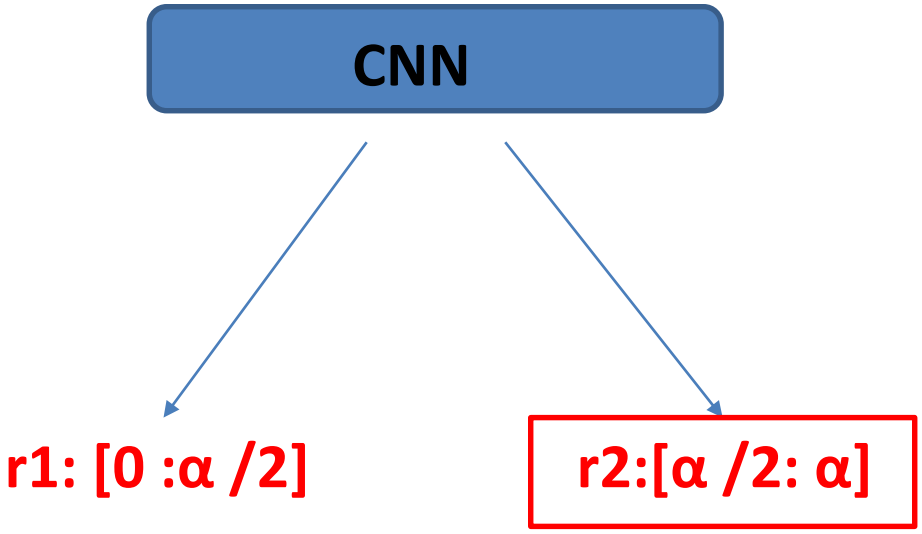
Proposed Method – Handling Objects with an Axis of Symmetry

β : rotation angle



Proposed Method – Handling Objects with an Axis of Symmetry

β : rotation angle

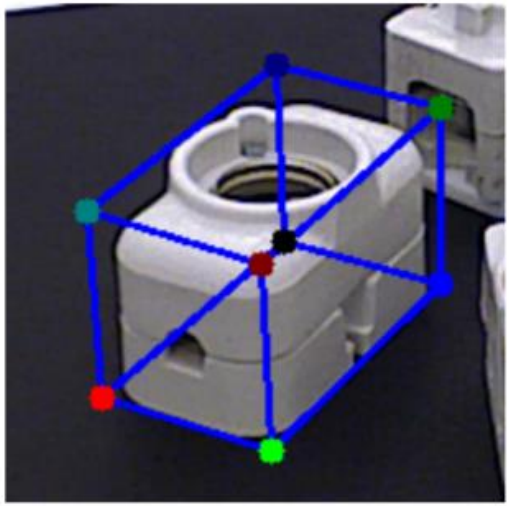


Proposed Method – Handling Objects with an Axis of Symmetry

β : rotation angle

Proposed Method – Handling Objects with an Axis of Symmetry

β : rotation angle

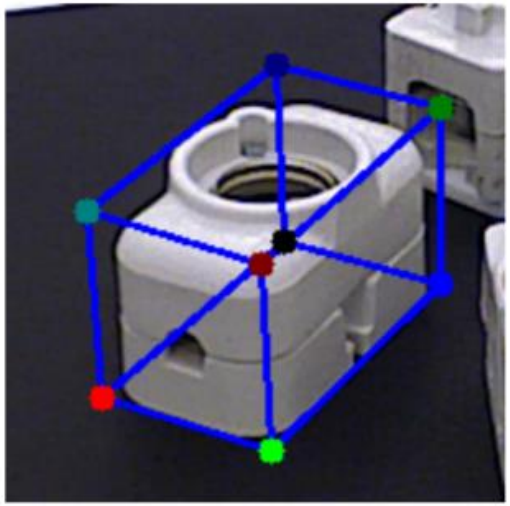


r1: [0 :α /2]

CNN

Proposed Method – Handling Objects with an Axis of Symmetry

β : rotation angle

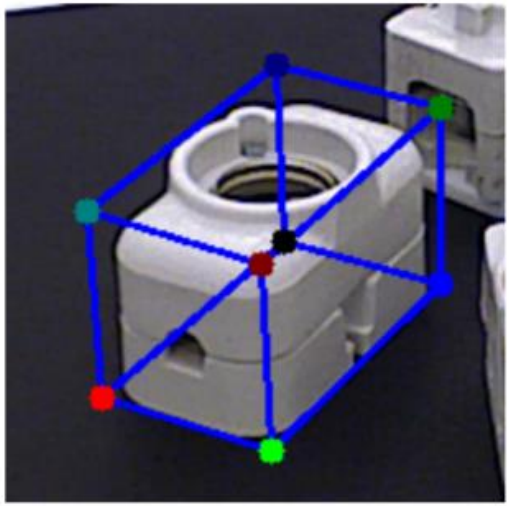


r1: [0 :α /2]

CNN

Proposed Method – Handling Objects with an Axis of Symmetry

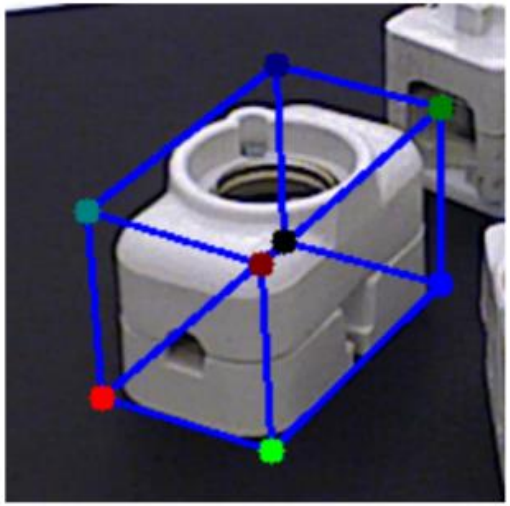
β : rotation angle

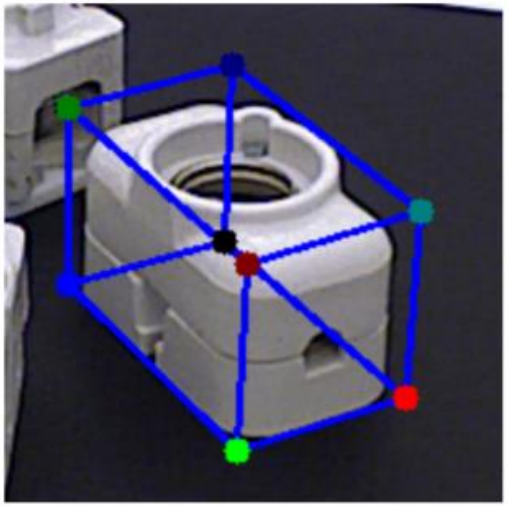


r1: [0 :α /2]

Proposed Method – Handling Objects with an Axis of Symmetry

β : rotation angle





r1: [0 :α /2]

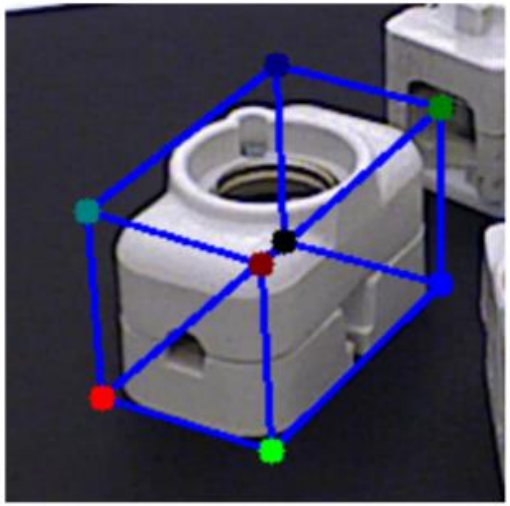
r2:[α /2: α]

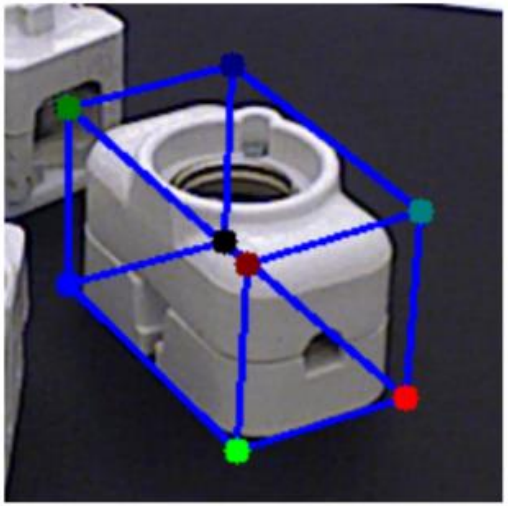
CNN

CNN

Proposed Method – Handling Objects with an Axis of Symmetry

β : rotation angle





r1: [0 :α /2]

r1: [0 :α /2]

CNN

CNN

Proposed Method – Refining the Pose

CNN

4개 또는 6개의 channels을 사용하기 때문에 VGG는 사용하지 못하고 각 Object마다 CNN을 구성.

Generate depth map, Binary mask or color rendering

Proposed Method – Refining the Pose

4개 또는 6개의 channels을 사용하기 때문에 VGG는 사용하지 못하고 각 Object마다 CNN을 구성.

Generate depth map, Binary mask or color rendering

Training a Feedback Loop for Hand Pose Estimation (ICCV2015)

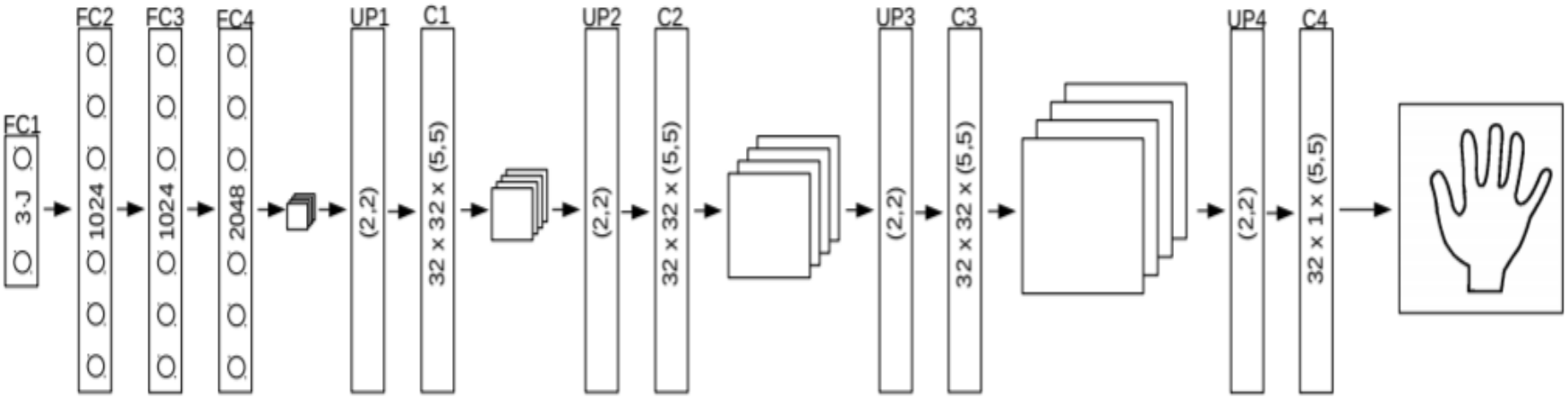
Proposed Method – Refining the Pose

$$\widehat{\mathbf{p}}^{(0)} = \mathrm{pred}(\mathcal{D}_{\mathrm{input}}) \quad .$$

Proposed Method – Refining the Pose

4개 또는 6개의 channels을 사용하기 때문에 VGG는 사용하지 못하고 각 Object마다 CNN을 구성.

Generate depth map, Binary mask or color rendering



Training a Feedback Loop for Hand Pose Estimation (ICCV2015)

Proposed Method – Refining the Pose

$$\widehat{\mathbf{p}}^{(0)} = \mathrm{pred}(\mathcal{D}_{\mathrm{input}}) \quad .$$

$$\mathcal{D}_{\mathrm{synth}} = \mathrm{synth}(\mathbf{p}) \quad .$$

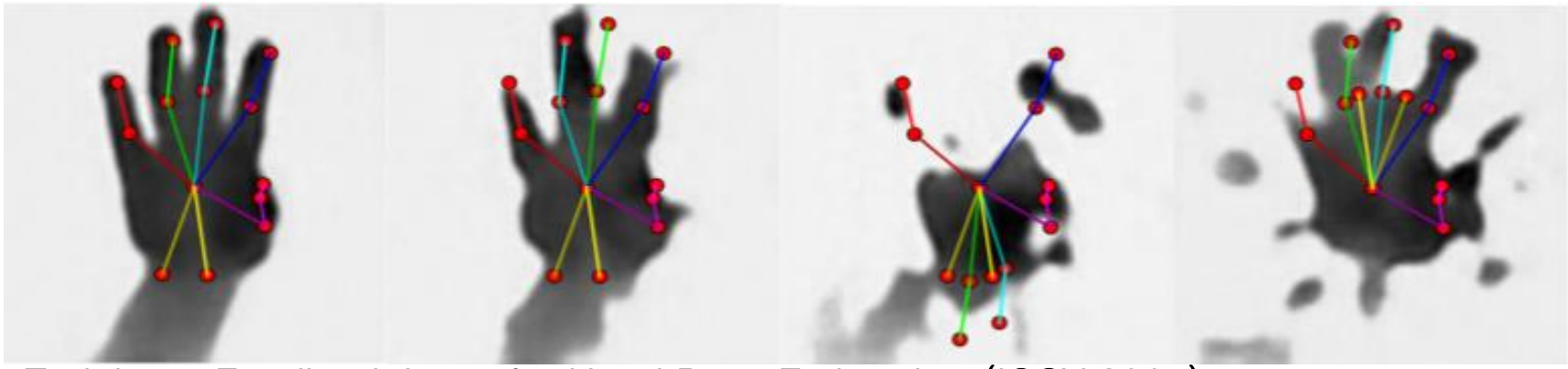
$$\widehat{\mathbf{p}} = \arg\min_{\mathbf{p}} \|\mathcal{D}_{\mathrm{input}} - \mathrm{synth}(\mathbf{p})\|^2 \quad .$$

Proposed Method – Refining the Pose

$$\widehat{\mathbf{p}}^{(0)} = \mathrm{pred}(\mathcal{D}_{\mathrm{input}}) \quad .$$

$$\mathcal{D}_{\mathrm{synth}} = \mathrm{synth}(\mathbf{p}) \quad .$$

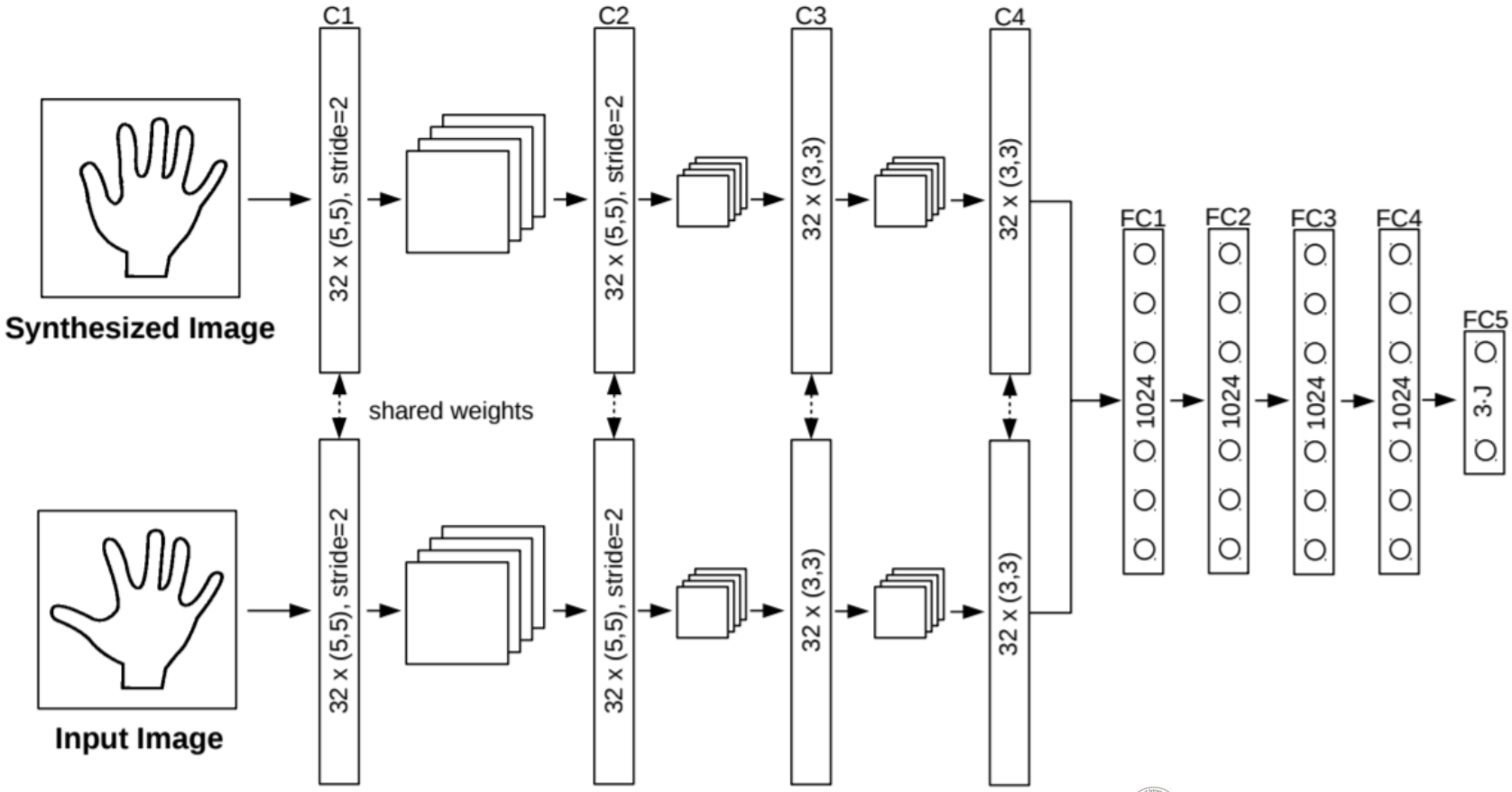
$$\widehat{\mathbf{p}} = \arg\min_{\mathbf{p}} \|\mathcal{D}_{\mathrm{input}} - \mathrm{synth}(\mathbf{p})\|^2 \quad .$$



Training a Feedback Loop for Hand Pose Estimation.(ICCV 2015)

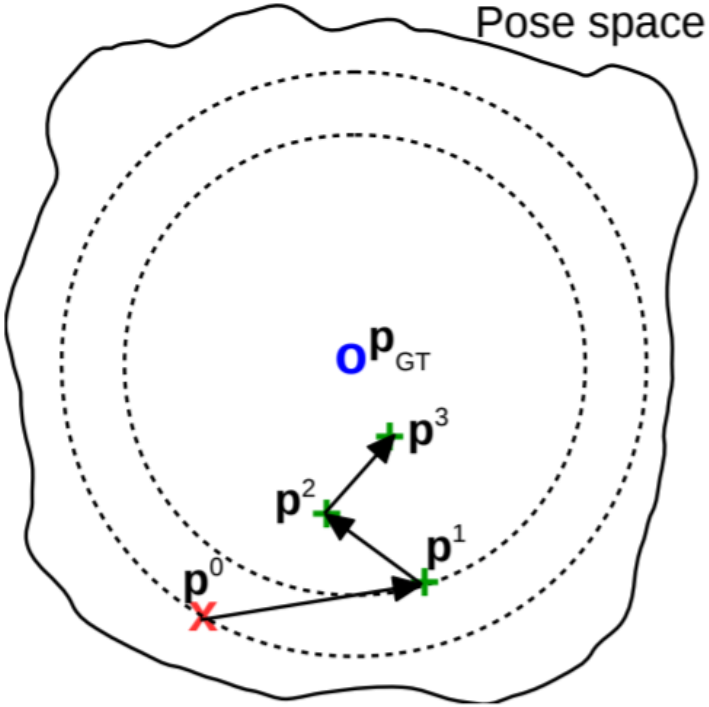
Proposed Method – Refining the Pose

$$\widehat{\mathbf{p}}^{(i+1)} \leftarrow \widehat{\mathbf{p}}^{(i)} + \mathrm{updater}(\mathcal{D}_{\mathrm{input}}, \mathrm{synth}(\widehat{\mathbf{p}}^{(i)})) \quad .$$



Proposed Method – Refining the Pose

$$\widehat{\mathbf{p}}^{(i+1)} \leftarrow \widehat{\mathbf{p}}^{(i)} + \mathrm{updater}(\mathcal{D}_{\mathrm{input}}, \mathrm{synth}(\widehat{\mathbf{p}}^{(i)})) \quad .$$



Proposed Method – Refining the Pose

$$\sum_{(W,\mathbf{e},\mathbf{t})\in\mathcal{T}} \sum_{(\hat{\mathbf{e}},\hat{\mathbf{t}})\in\mathcal{N}(\mathbf{e},\mathbf{t})} \sum_{i} \|\text{Proj}_{\mathbf{e},\mathbf{t}}(\mathbf{M}_i^o) - \text{Proj}_{\hat{\mathbf{e}},\hat{\mathbf{t}}}(\mathbf{M}_i^o) - m_i(h_\mu(W, \text{Render}(\hat{\mathbf{e}}, \hat{\mathbf{t}})))\|^2$$

h_μ **Denotes the CNN…….**

Proposed Method – Refining the Pose

$$\sum_{(W,\mathbf{e},\mathbf{t})\in\mathcal{T}} \sum_{(\hat{\mathbf{e}},\hat{\mathbf{t}})\in\mathcal{N}(\mathbf{e},\mathbf{t})} \sum_{i} \|\text{Proj}_{\mathbf{e},\mathbf{t}}(\mathbf{M}_i^o) - \text{Proj}_{\hat{\mathbf{e}},\hat{\mathbf{t}}}(\mathbf{M}_i^o) - m_i(h_\mu(W, \text{Render}(\hat{\mathbf{e}}, \hat{\mathbf{t}})))\|^2$$

h_μ **Denotes the CNN…….**

$$\hat{\mathbf{v}} \leftarrow \hat{\mathbf{v}} + h_\mu(W, \text{Render}(\hat{\mathbf{e}}, \hat{\mathbf{t}})) \ .$$

$\hat{\mathbf{v}}$ **Projections of the corners** $\hat{\mathbf{v}} = [\ldots \hat{\mathbf{m}}_i^\top \ldots]^\top$

Experiments

LINEMOD

Occlusioned- Object

T-LESS

Experiments

LINEMOD(2008)

Ape, benchvise, bowl,can,cat,cup,driller,duck,glue, holepuncher,iron,lamp,phone,cam,eggbox

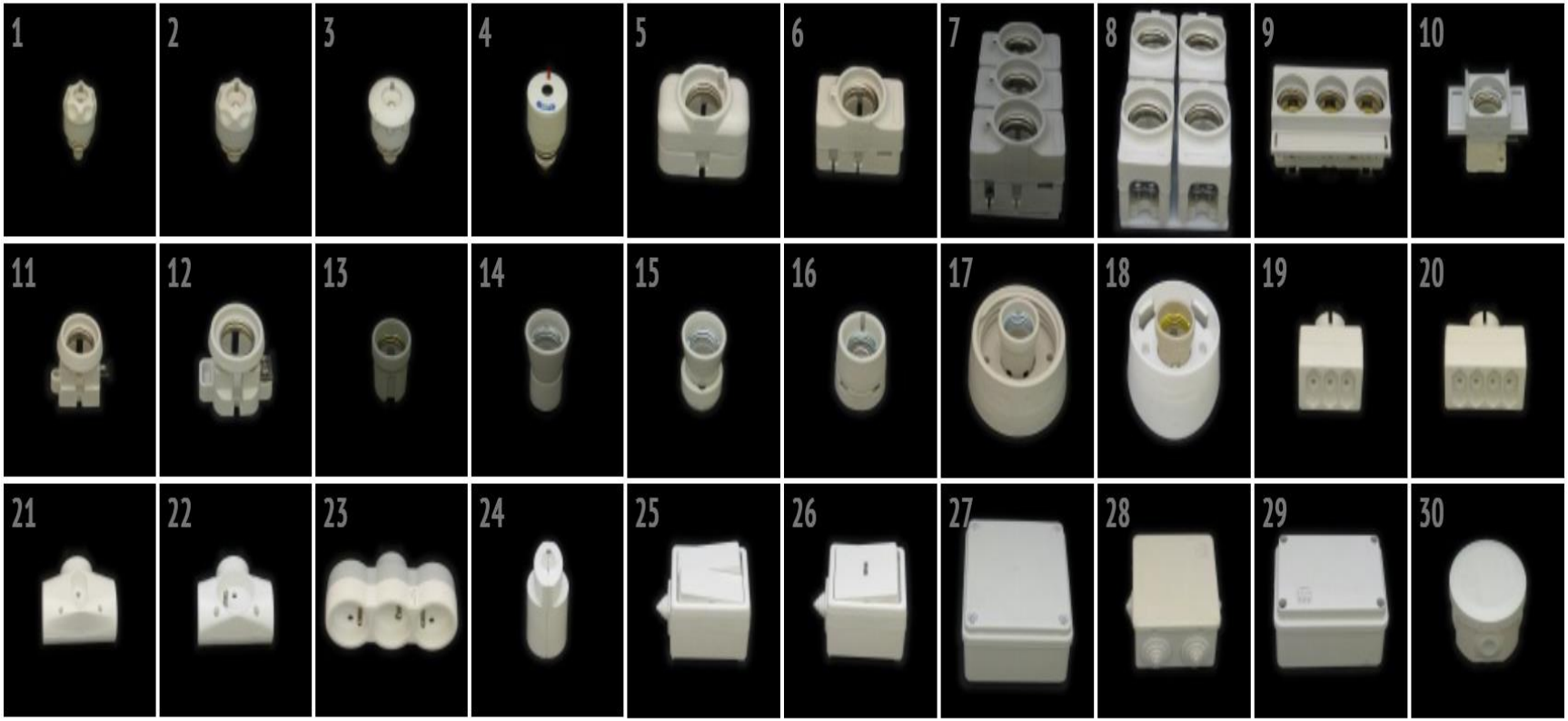
Experiments

Occlusioned- Object(2015)

Ape, benchvise, bowl,can,cat,cup,driller,duck,glue, holepuncher,iron,lamp,phone,cam,eggbox

Experiments

T-LESS(2015)



Experiments

Evaluation Metrics

1) 2D Projections : 추정된 pose와 GT pose의 projection된 점 간의 2D distance의 평균이 5픽셀 미만인 경우 올바른 것으로 판단.

2) 6D Pose : transform된 3D vertices 간의 평균 거리가 객체의 직경의 10%보다 작은 경우 올바른 포즈로 판단.

$$\frac{1}{|\mathcal{V}|}\sum_{\mathbf{M}\in\mathcal{V}}\|\mathrm{Tr}_{\hat{\mathbf{e}},\hat{\mathbf{t}}}(\mathbf{M})-\mathrm{Tr}_{\bar{\mathbf{e}},\bar{\mathbf{t}}}(\mathbf{M})\|_2 \quad \frac{1}{|\mathcal{V}|}\sum_{\mathbf{M}_1\in\mathcal{V}}\min_{\mathbf{M}_2\in\mathcal{V}}\|\mathrm{Tr}_{\hat{\mathbf{e}},\hat{\mathbf{t}}}(\mathbf{M}_1)-\mathrm{Tr}_{\bar{\mathbf{e}},\bar{\mathbf{t}}}(\mathbf{M}_2)\|_2 \ .$$

3) 5cm 5° : translation error는 5cm 미만, rotation error는 5 °미만으로 각각 설정하여 계산한 것.

Experiments - LINEMOD(2008)

Sequence	Direct	BB	Mask Ref.	RGB Ref.
Ape (*)	91.2	96.2	97.5	**97.7**
Bench Vise	61.3	80.2	90.1	**91.5**
Camera	43.1	82.8	82.5	**86.3**
Can	62.5	85.8	90.2	**91.5**
Cat (*)	93.1	97.2	**98.6**	**98.6**
Driller (*)	46.5	77.6	83.4	**83.6**
Duck	67.9	84.6	94.0	**94.1**
Egg Box	68.2	90.1	92.0	**93.2**
Glue	69.3	93.5	94.2	**95.8**
Hole Puncher	78.2	91.7	95.2	**97.4**
Iron	64.5	79.0	79.5	**85.0**
Lamp	50.4	79.9	**83.6**	83.5
Phone	46.9	80.0	85.6	**88.9**
average	64.9	85.4	89.7	**91.3**

Experiments - LINEMOD(2008)

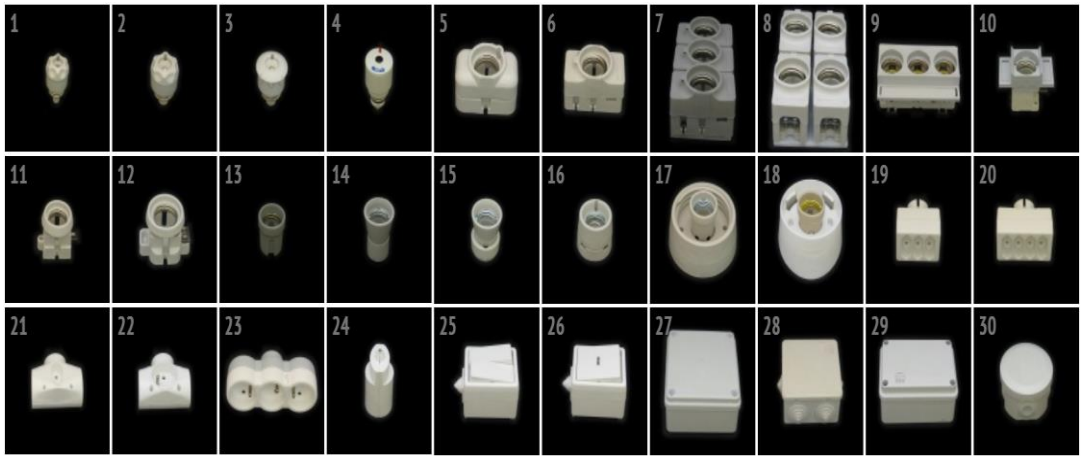
Metric	2D Projection			6D Pose		5cm 5°	
Sequence	[2]	w/o	w/Ref.	[2]	w/Ref.	[2]	w/Ref.
Ape	85.2	95.3	**96.6**	33.2	**40.4**	34.4	**80.2**
Bench Vi.	67.9	80.0	**90.1**	64.8	**91.8**	40.6	**81.5**
Camera	58.7	80.9	**86.0**	38.4	**55.7**	30.5	**60.0**
Can	70.8	84.1	**91.2**	62.9	**64.1**	48.4	**76.8**
Cat	84.2	97.0	**98.8**	42.7	**62.6**	34.6	**79.9**
Driller	73.9	74.1	**80.9**	61.9	**74.4**	54.5	**69.6**
Duck	73.1	81.2	**92.2**	30.2	**44.3**	22.0	**53.2**
Egg Box	83.1	87.9	**91.0**	49.9	**57.8**	57.1	**81.3**
Glue	74.2	89.0	**92.3**	31.2	**41.2**	23.6	**54.0**
Hole P.	78.9	90.5	**95.3**	52.8	**67.2**	47.3	**73.1**
Iron	83.6	78.9	**84.8**	80.0	**84.7**	58.7	**61.1**
Lamp	64.0	74.4	**75.8**	67.0	**76.5**	49.3	**67.5**
Phone	60.6	77.6	**85.3**	38.1	**54.0**	26.8	**58.6**
average	73.7	83.9	**89.3**	50.2	**62.7**	40.6	**69.0**
Bowl	-	97.0	**98.9**	-	**60.0**	-	**90.9**
Cup	-	93.4	**94.8**	-	**45.6**	-	**58.4**

Experiments - T-LESS(2015)

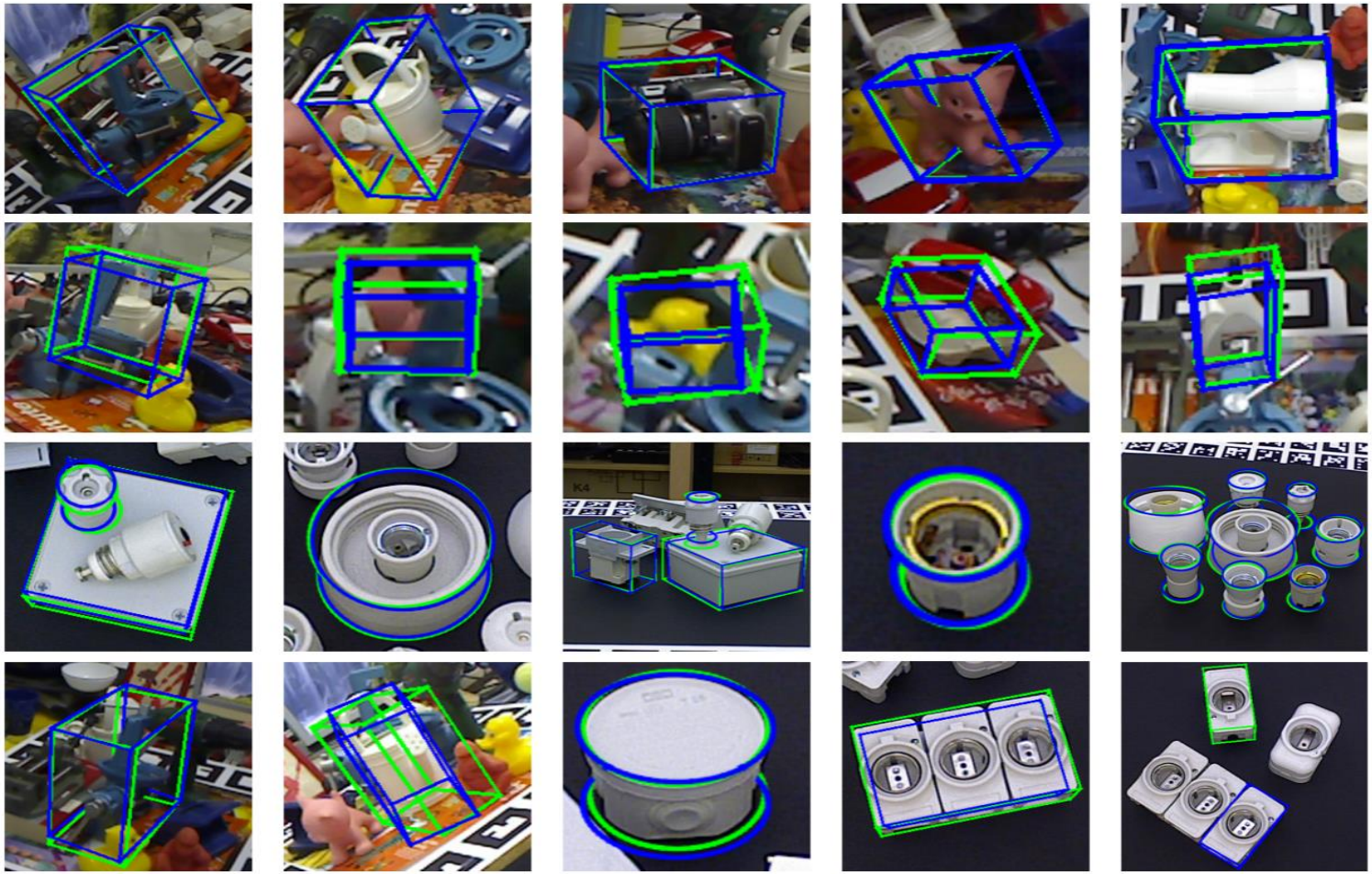
Scene ID: [Obj. IDs]	6D Pose	Average
1: [2, 30]	50.8, 55.4	53.1
2: [5, 6]	56.5, 55.6	56.1
4: [5, 26, 28]	68.7, 53.3, 40.6	54.3
5: [1, 10, 27]	39.6, 69.9, 50.1	53.2
7: [1, 3, 13, 14, ...	42.0, 61.7, 64.5, 40.7, ...	
7: ... 15, 16, 17, 18]	...39.7, 45.7, 50.2, 83.7	53.5

Experiments - T-LESS(2015)

Scene ID: [Obj. IDs]	6D Pose	Average
1: [2, 30]	50.8, 55.4	53.1
2: [5, 6]	56.5, 55.6	56.1
4: [5, 26, 28]	68.7, 53.3, 40.6	54.3
5: [1, 10, 27]	39.6, 69.9, 50.1	53.2
7: [1, 3, 13, 14, ...	42.0, 61.7, 64.5, 40.7, ...	
7: ... 15, 16, 17, 18]	...39.7, 45.7, 50.2, 83.7	53.5



Experiments



Conclusion

1) Deep Learning Network를 이용한 3D point regression에 대해서 remarkable 한 능력과 가능성을 확인.

2) 3D pose Estimation에서 SOTA다

3) Symmetric object에 대해서 기존의 방법들 보다 더 잘함.

End