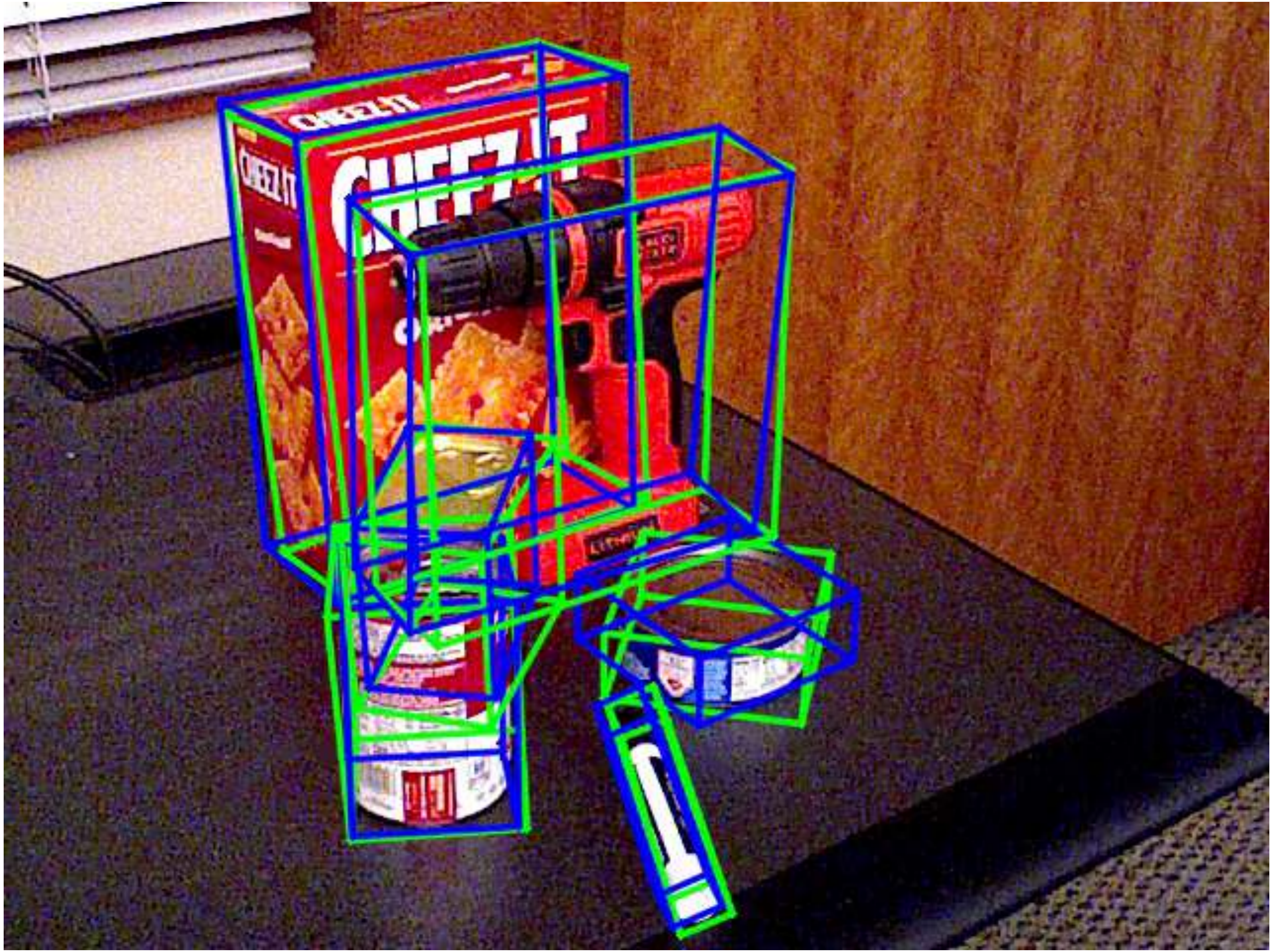
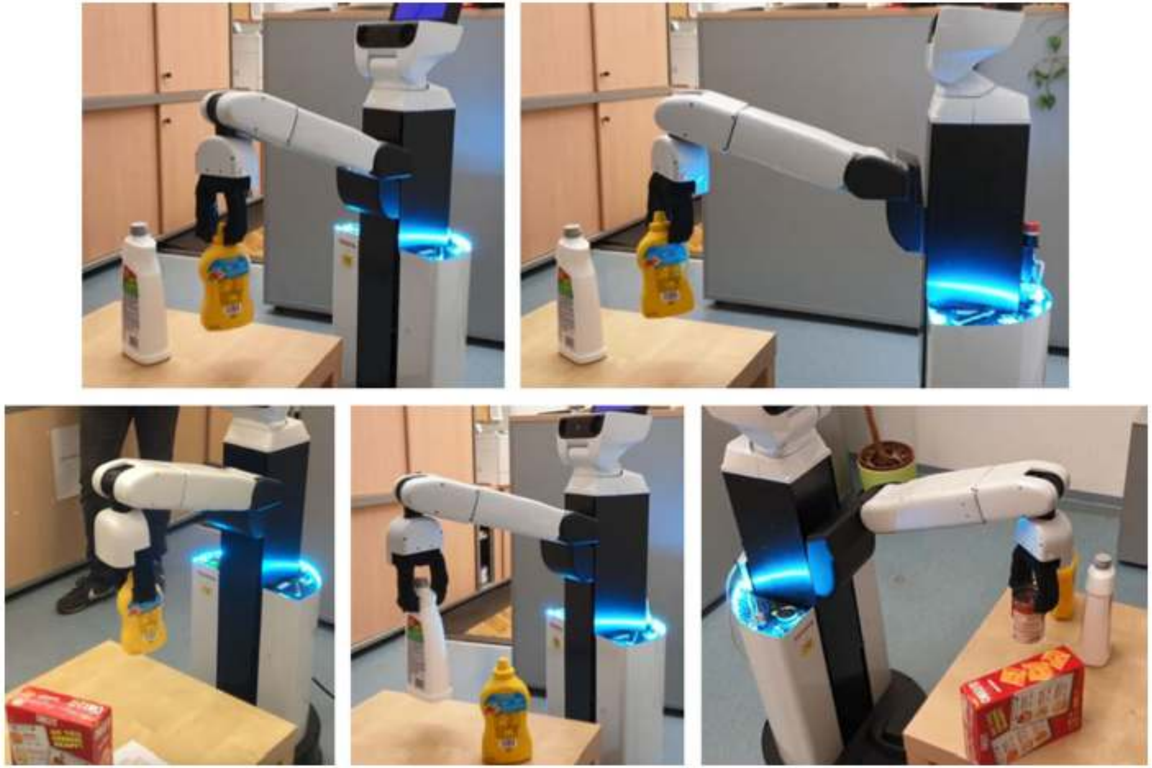
3D Object Detection and Pose Estimation

Vincent Lepetit

École des Ponts
ParisTech



possible applications



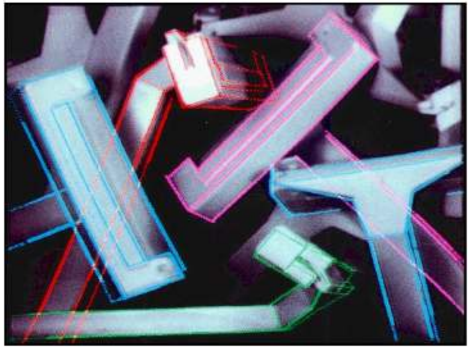
[Vincze et al, 2020]

École des Ponts
ParisTech

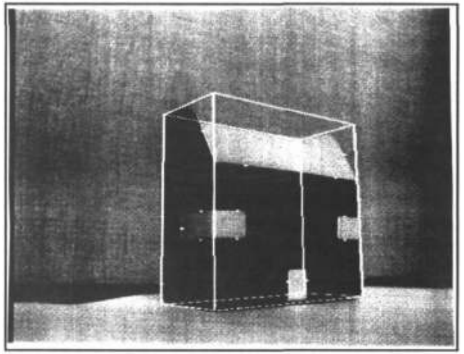
possible applications

[Petit et al, ISMAR 2013]

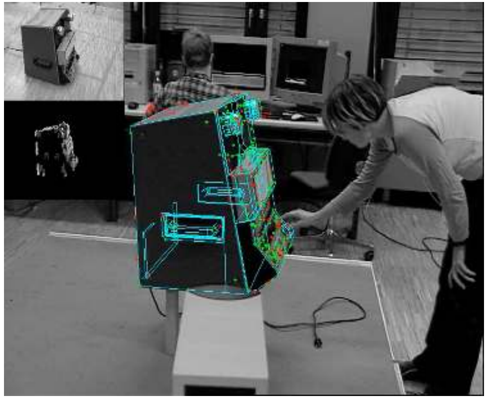
a bit of history



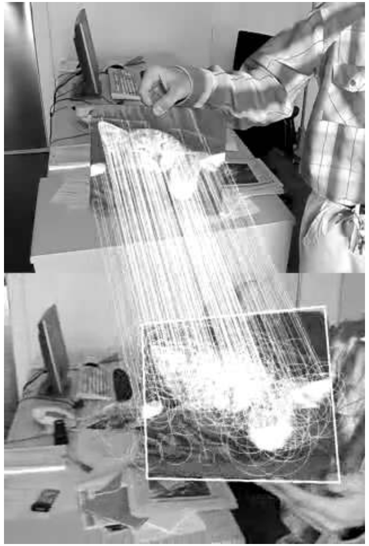
[Lowe, 1987]



[Harris&Stennet, 1990]



[Vacchetti et al, CVPR 2003]



[Lepetit et al, CVPR 2004]

École des Ponts
ParisTech

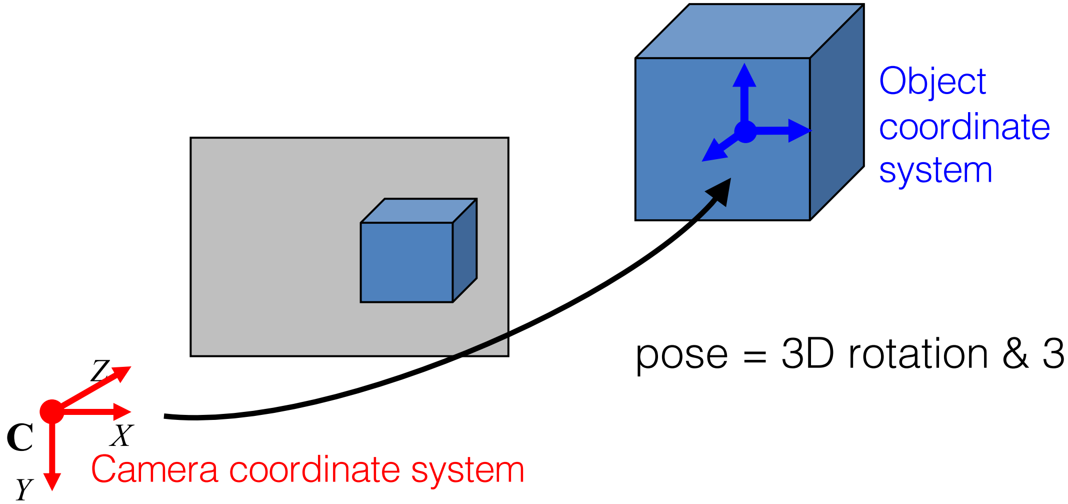
more modern take

training set: (many) annotated real images

→ deep network → object pose

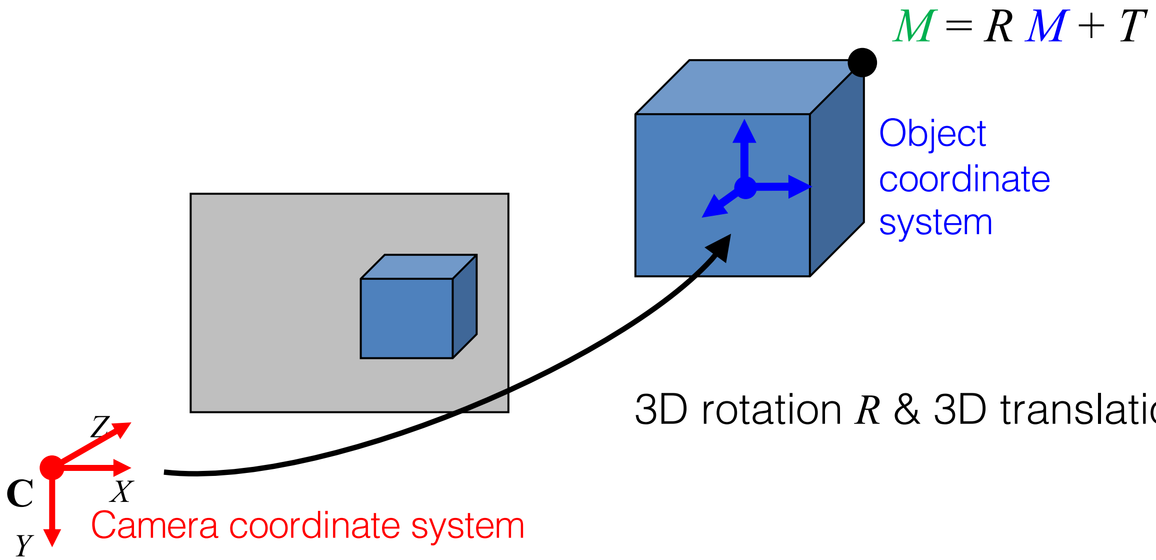
training set: About 200 annotated real images

3D Pose / 6D Pose



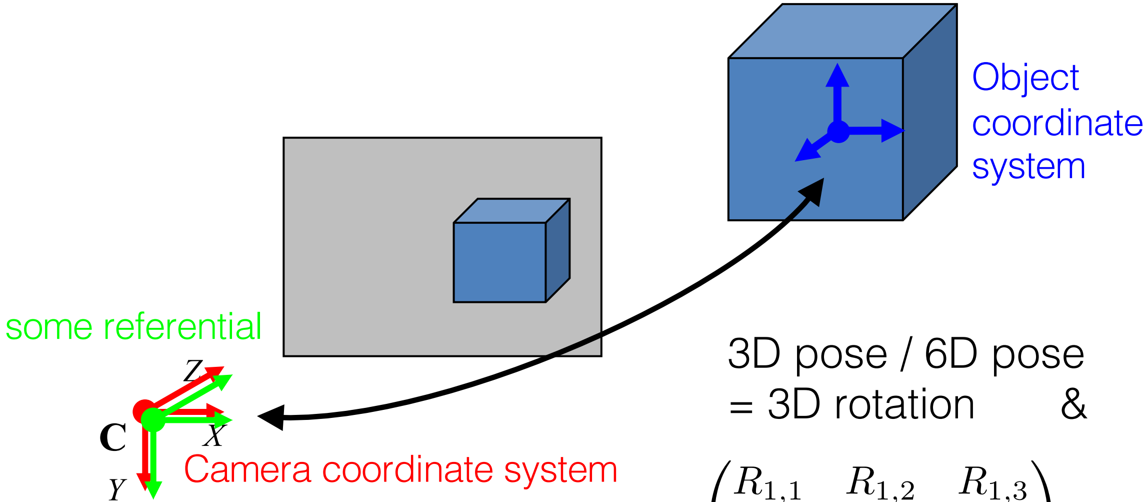
pose = 3D rotation & 3D translation

3D Pose / 6D Pose



3D rotation R & 3D translation T?

3D Pose / 6D Pose



3D pose / 6D pose
= 3D rotation & 3D translation

$$\begin{pmatrix} R_{1,1} & R_{1,2} & R_{1,3} \\ R_{2,1} & R_{2,2} & R_{2,3} \\ R_{3,1} & R_{3,2} & R_{3,3} \end{pmatrix} \qquad \begin{pmatrix} T_1 \\ T_2 \\ T_3 \end{pmatrix}$$

loss for pose prediction

For the 3D translation, simply the Euclidean distance between prediction and ground truth:

$$\mathcal{L}_T = \|T - \hat{T}\|^2$$

For the 3D rotation, geodesic distance:

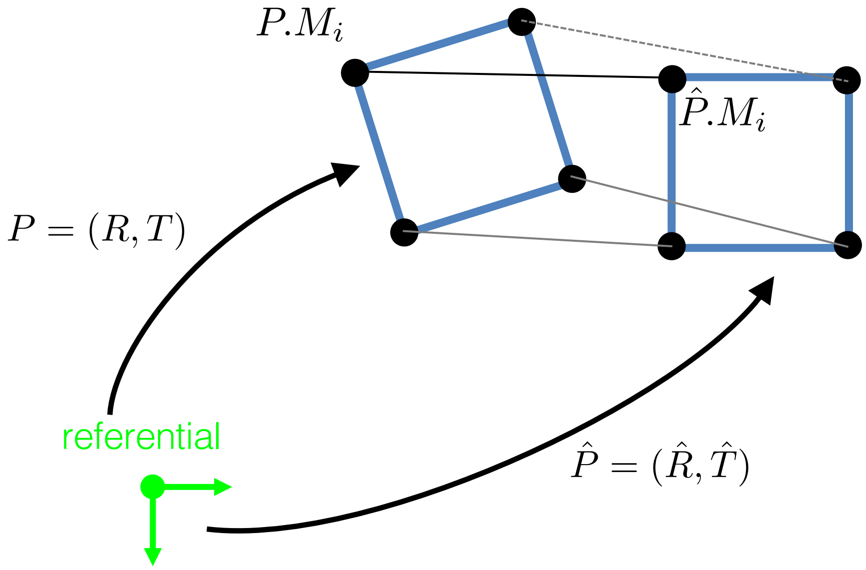
$$\begin{aligned}\mathcal{L}_R &= \|\log(R\hat{R}^\top)\|_F \\ &= \cos^{-1}(tr(R\hat{R}^\top) - 1)/2)\end{aligned}$$

For the full 3D pose:

$$\mathcal{L}_{\text{pose}} = \mathcal{L}_T + \gamma\mathcal{L}_T$$

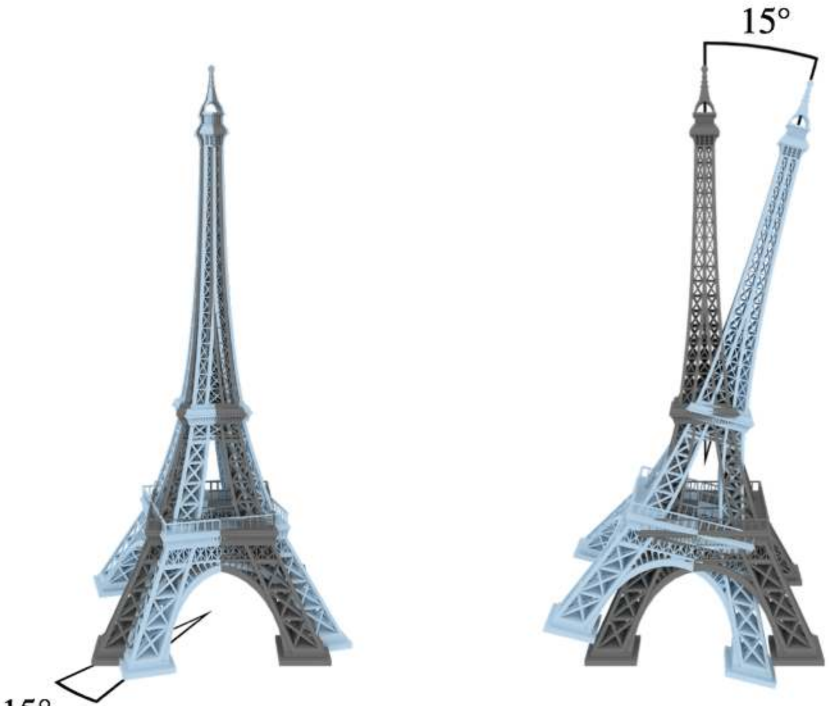
École des Ponts
ParisTech

alternative loss for pose prediction



$$\begin{cases} \mathcal{L} = \sum_i \|P.M_i - \hat{P}.M_i\|^2 \\ \\ P.M_i = RM_i + T \\ \hat{P}.M_i = \hat{R}M_i + \hat{T} \end{cases}$$

alternative loss for pose prediction



(a) d=9.2m (b) d=18.9m

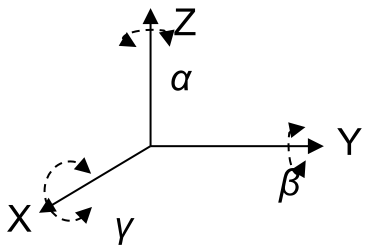
R. Brégier et al. Defining the Pose of Any 3D Rigid Object and an Associated Distance. IJCV, June 2018.

possible parameterizations of the rotation matrix

- Directly the rotation matrix (ie 9 values);

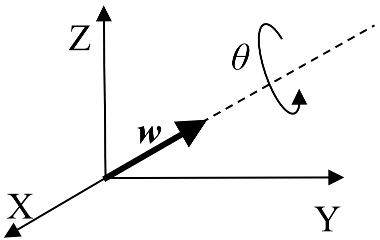
- Euler angles (3 values):

$$\mathbf{R} = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{bmatrix}$$

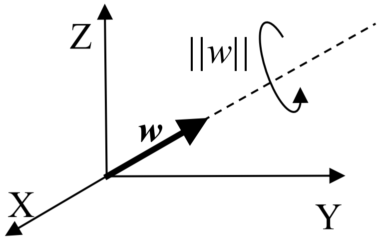


- A unit quaternion (4 values):

$$q = \left(\cos\frac{\theta}{2}, \boldsymbol{w}\sin\frac{\theta}{2}\right)$$

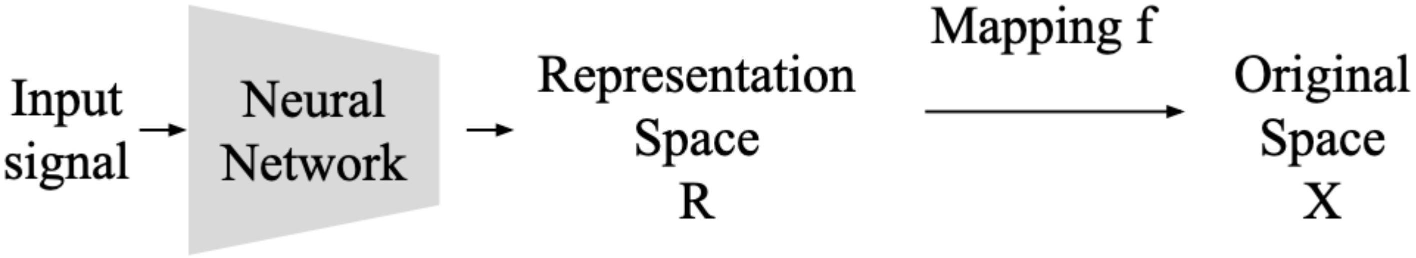


- An Exponential Map (a unit 3-vector, 3 values):



- ...

the problem with these representations

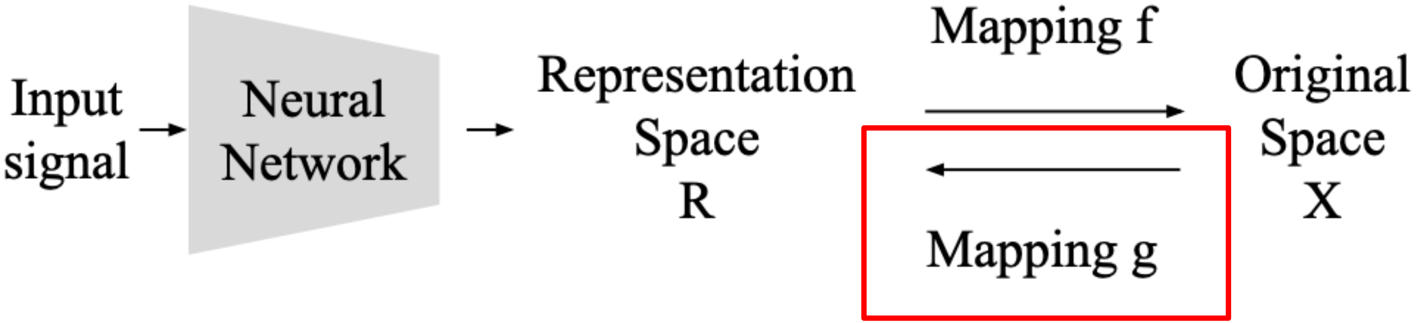


for example, Euler angles

the 3x3 rotation matrix itself

On the Continuity of Rotation Representations in Neural Networks. Zhou et al. CVPR 2018.

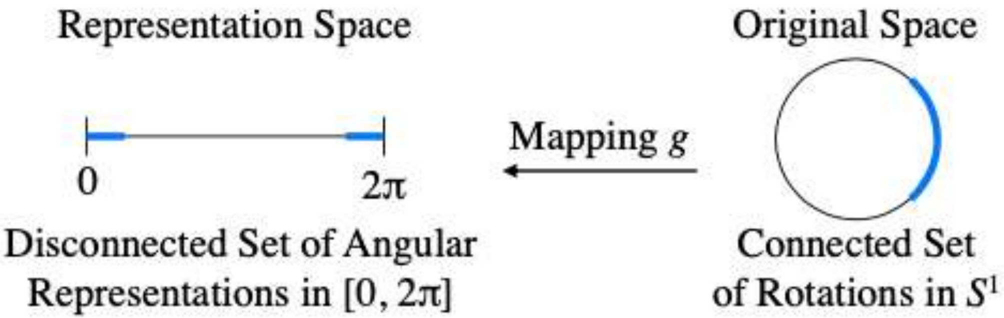
the problem with these representations



Needed for training the network (in back-propagation).
Not continuous for these rotation representations

On the Continuity of Rotation Representations in Neural Networks. Zhou et al. CVPR 2018.

discontinuities of g



proposed solution

- 2 3-vectors (6 values): e_1, e_2

$$e'_1 = \frac{e_1}{||e_1||_2}$$
$$e'_3 = \frac{e'_1 \wedge e_2}{||e_2||_2}$$
$$e'_2 = e'_3 \wedge e'_1,$$

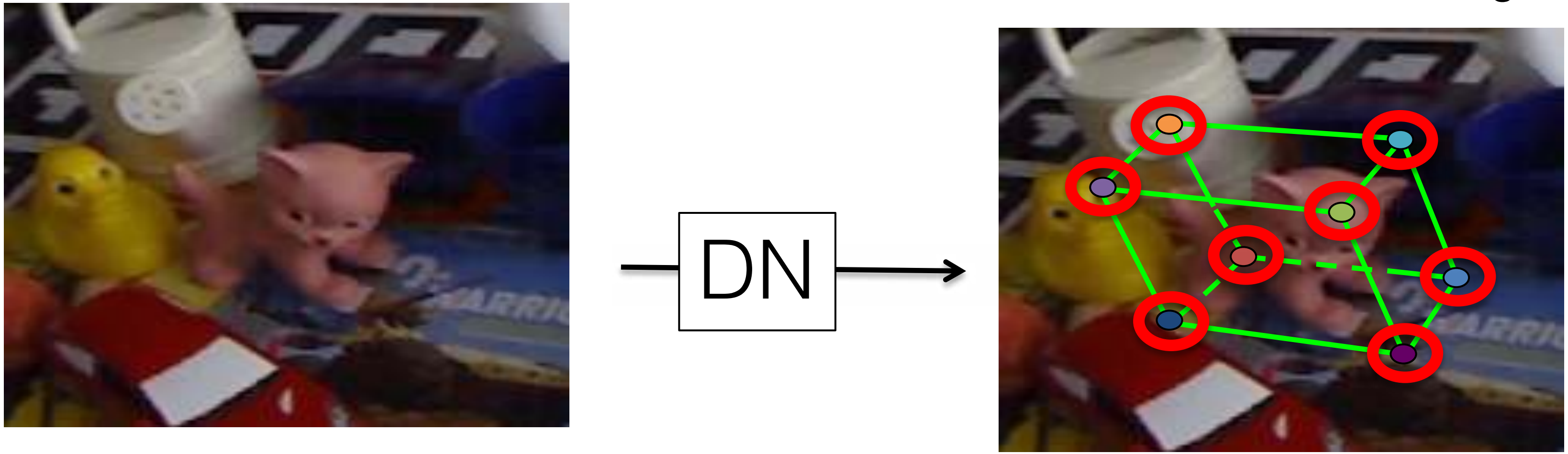
$$R = \begin{pmatrix} e'_1 & e'_2 & e'_3 \end{pmatrix}$$

It is then possible to define a $g(R) = (e_1, e_2)$ function that is continuous.

On the Continuity of Rotation Representations in Neural Networks. Zhou et al. CVPR 2018.

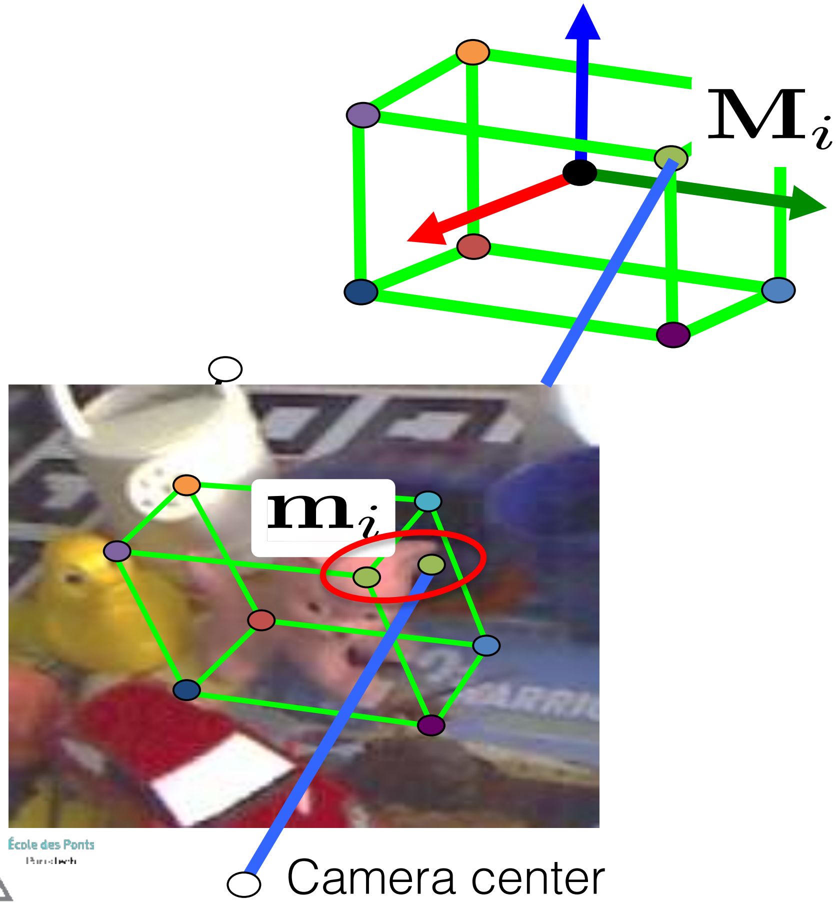
alternative predictions (1)

the 2D projections of the 8 corners of the 3D bounding box

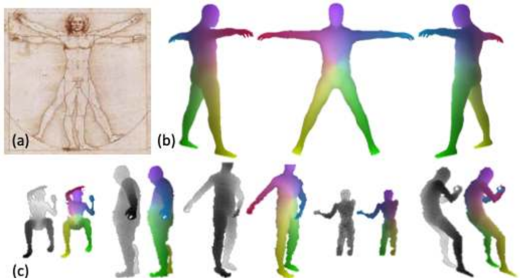


BB8: A Scalable, Accurate, Robust to Partial Occlusion Method for Predicting the 3D Poses of Challenging Objects without Using Depth. Mahdi Rad and Vincent Lepetit. ICCV 2017.

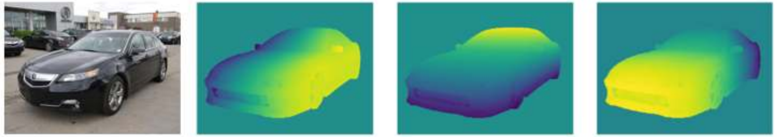
3D pose estimation from correspondences



Alternative predictions (2)

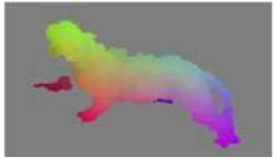


Taylor et al. The Vitruvian Manifold: Inferring Dense Correspondences for One-Shot Human Pose Estimation. CVPR 2012.

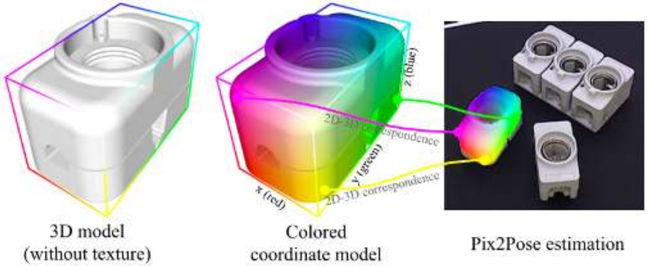


Location Fields.   Wang et al., ECCV 2018

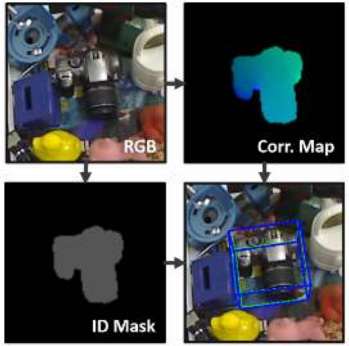
Normalized Object Coordinate Space for Category-Level 6D Object Pose and Size Estimation. Wang et al., CVPR 2019.



E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother. Learning 6D Object Pose Estimation using 3D Object Coordinates. ECCV 2014.



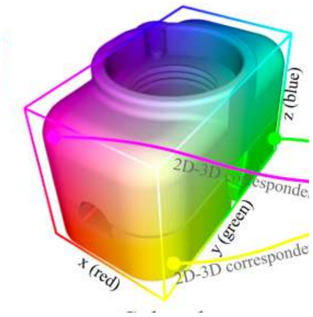
Pix2Pose: Pixel-Wise Coordinate Regression of Objects for 6D Pose Estimation. Park et al., CVPR 2019.

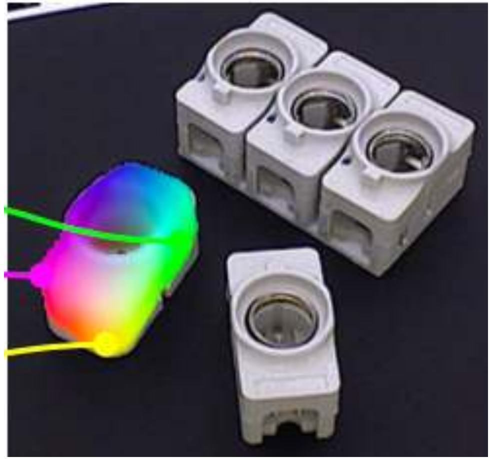


DPOD: 6D Pose Object Detector and Refiner. Zakharov et al. ICCV 2019.

École des Ponts
ParisTech

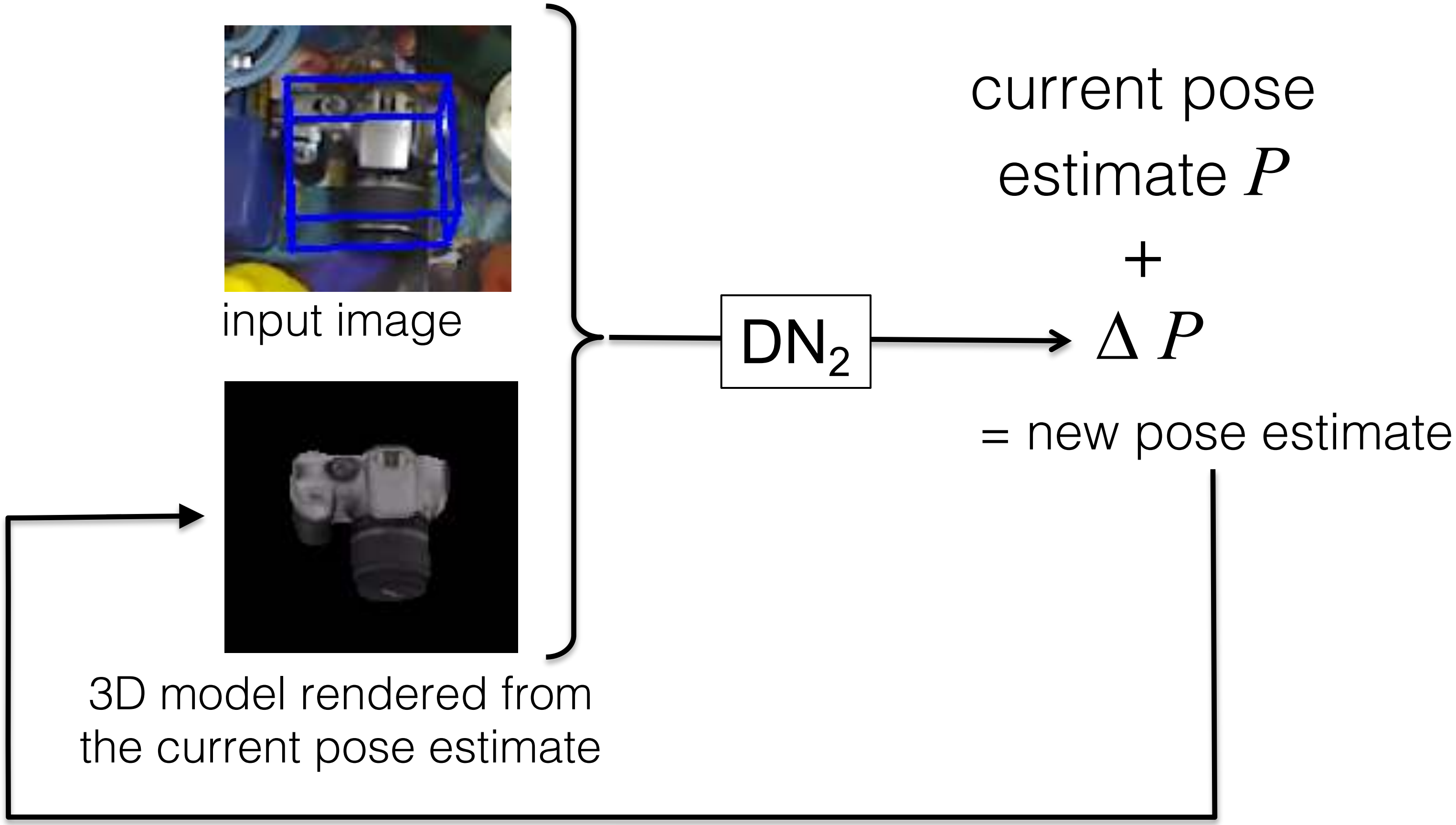
how to use 3D coordinate maps





pose refinement

refining the pose



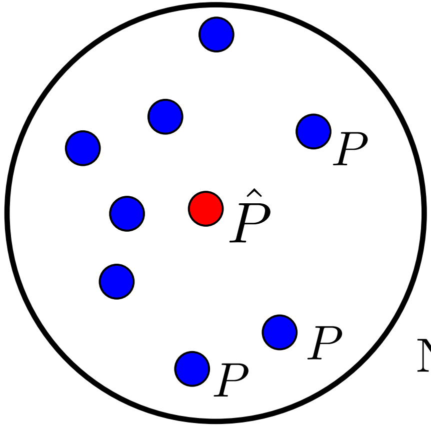
refining the pose, why it works

I

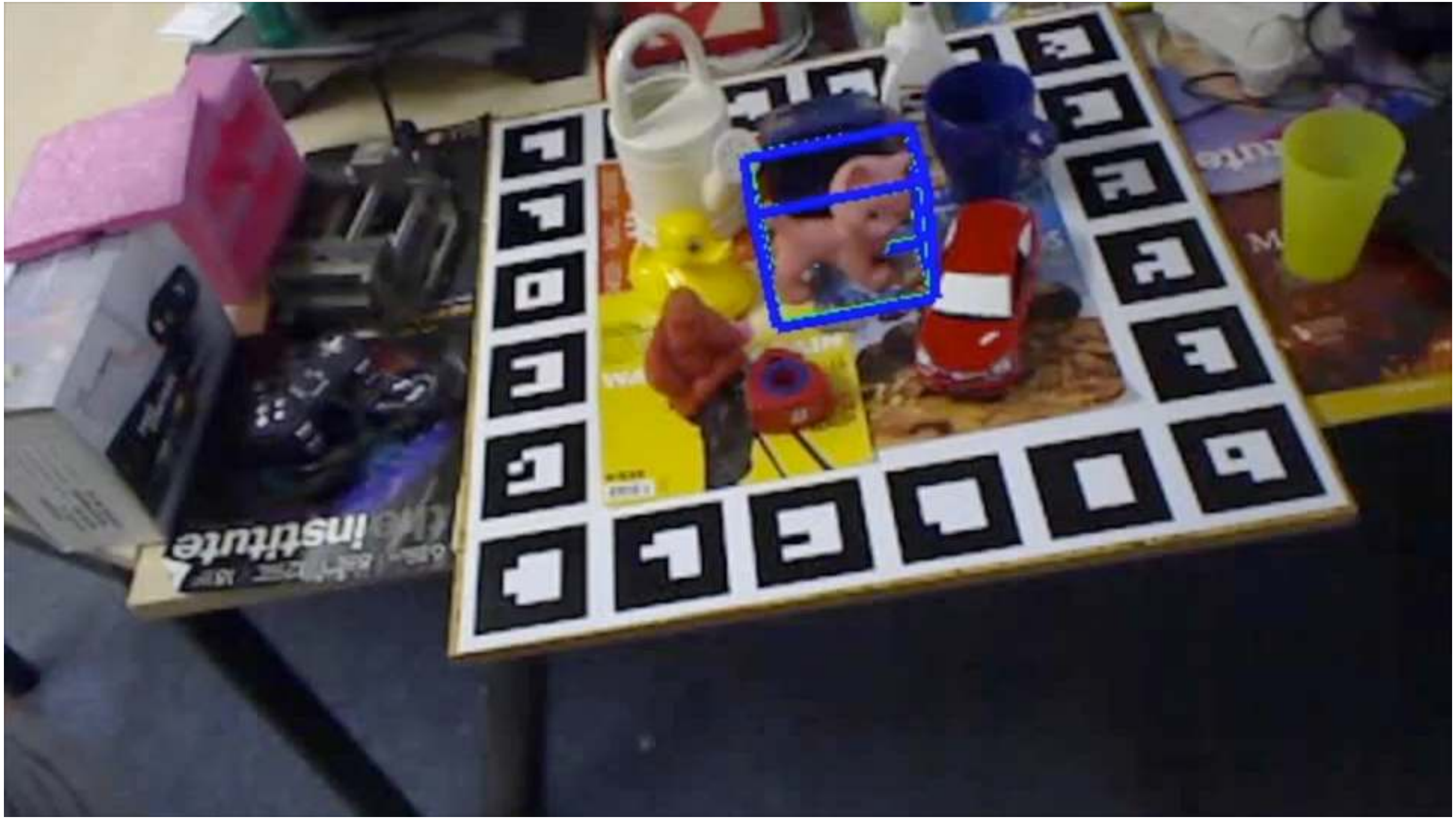
rendering(P)

$$\left.\begin{matrix} I \\ \text{rendering}(P) \end{matrix}\right\} \longrightarrow \boxed{f_2} \longrightarrow \Delta P$$

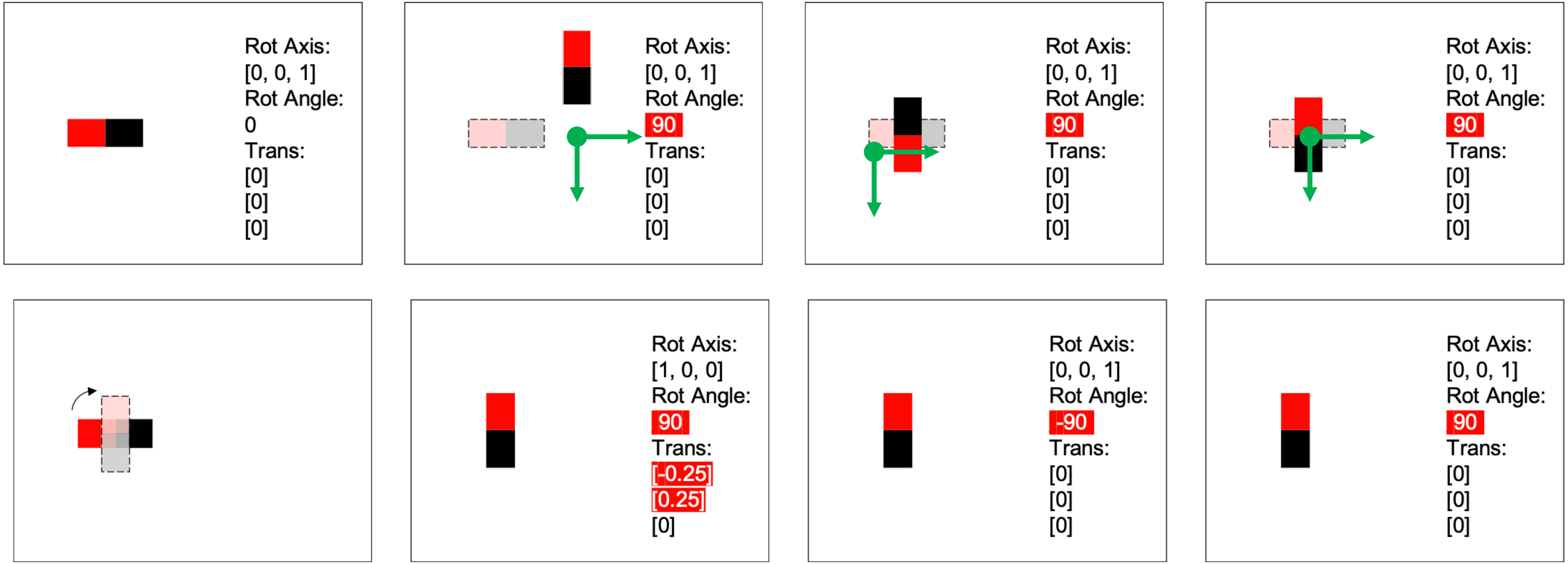
$$\mathcal{L}_2 = \sum_{P \in \text{Neighb}(\hat{P})} \mathcal{L}(\overbrace{f_2(I, \text{rendering}(P))}^{\Delta P}.P, \hat{P})$$



Neighb$(\hat{P})$



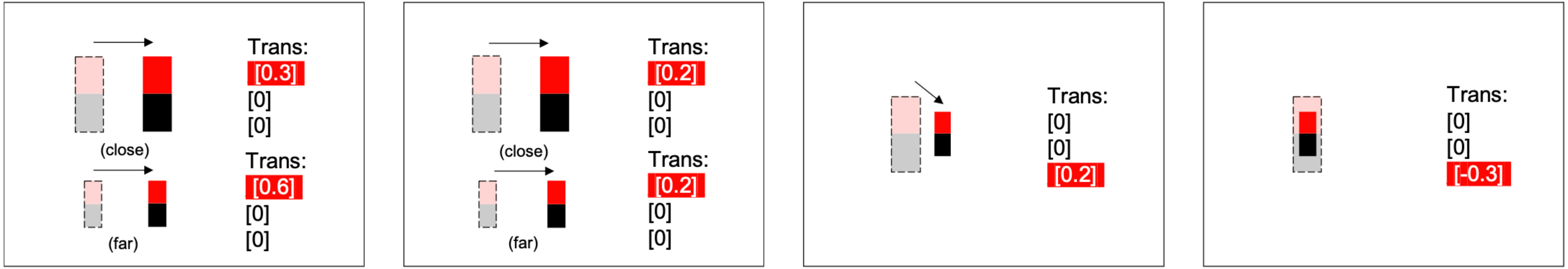
DeepIM: Decoupled Coordinates



(a) Initial pose (b) Camera coord. (c) Model coord. (d) disentangled coord.

DeepIM: Deep Iterative Matching for 6D Pose Estimation. Li et al. 2019

DeepIM: Decoupled Coordinates (T)



(a) Camera coord. xy-plane translation

(b) Disentangled coord. xy-plane translation

(c) Camera coord. z-axis translation

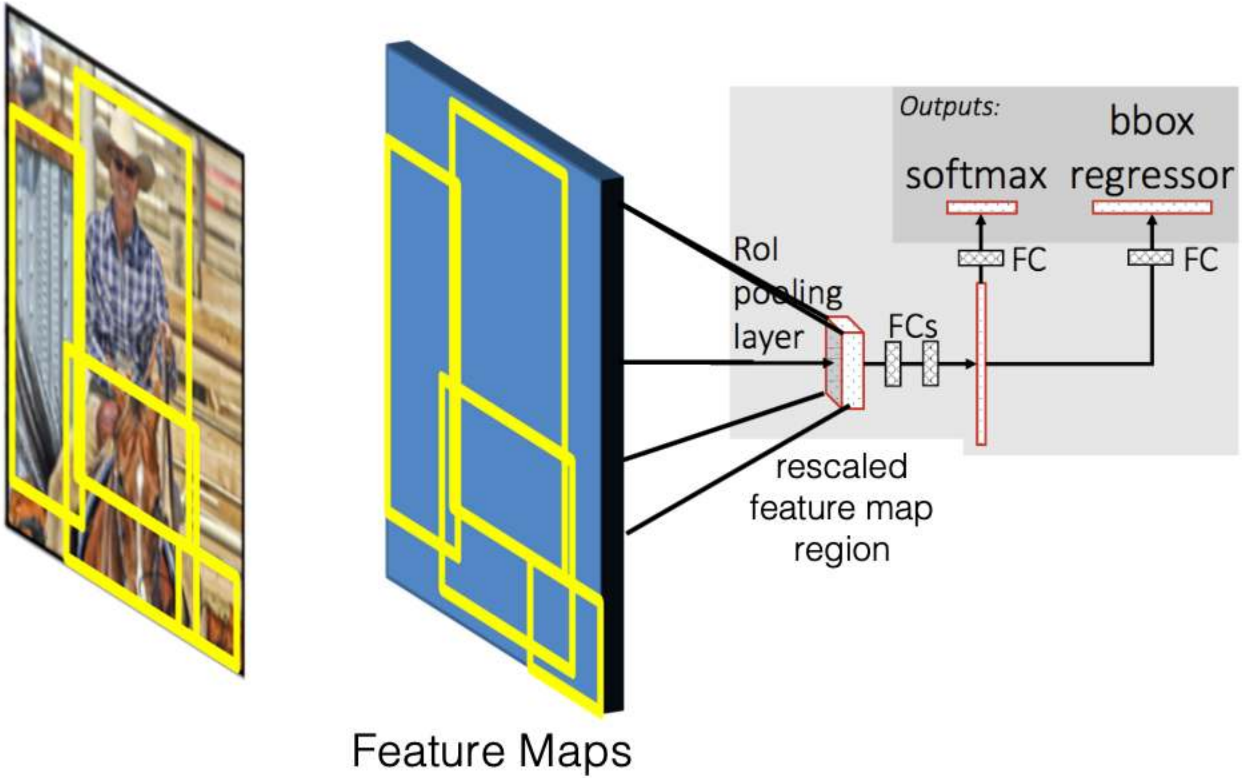
(d) Disentangled coord. z-axis translation

2D detection

→ deep network → 3D pose

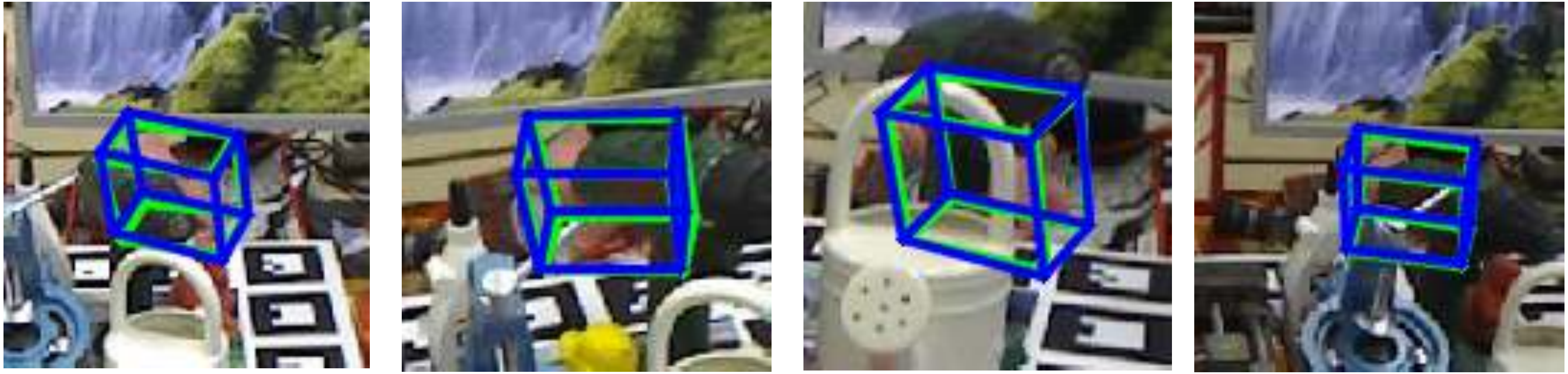
SSD-6D: Making RGB-Based 3D Detection and 6D Pose Estimation Great Again. Wadim Kehl, Fabian Manhardt, Federico Tombari, Slobodan Ilic, Nassir Navab. ICCV 2017.

Fast-RCNN / Mask-RCNN / Detectron2



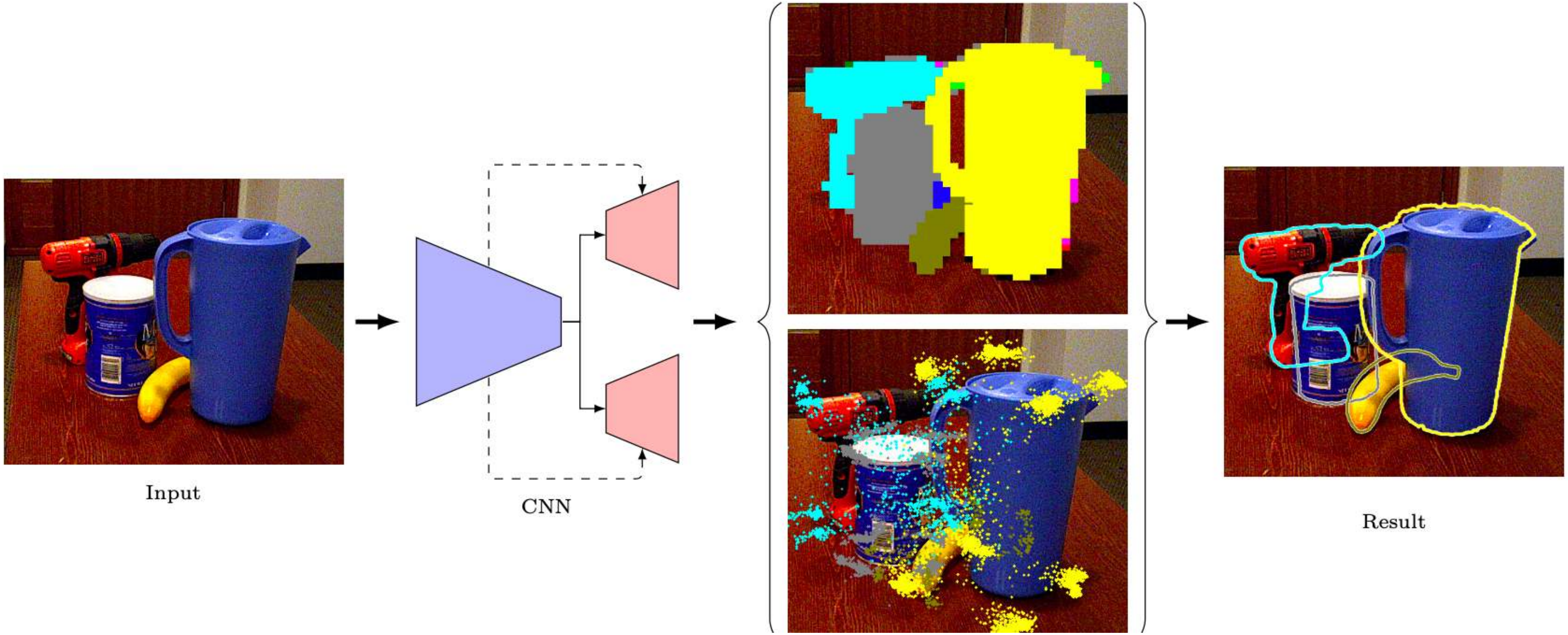
École des Ponts
ParisTech

Dealing with Partial Occlusion



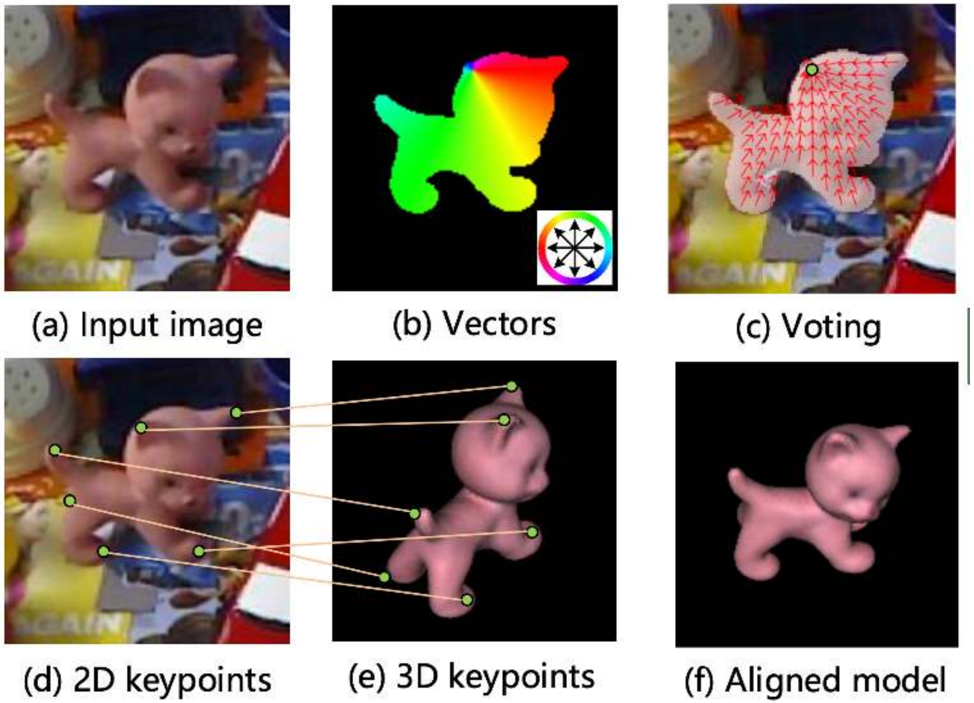
Avoid Occlusions in the Input

Voting for the pose



Segmentation-driven 6D Object Pose Estimation. Hu et al. CVPR 2019.

Voting for the pose



PVNet: Pixel-wise Voting Network for 6DoF Pose Estimation. Peng et al. CVPR 2019.

Training Set:
About 200 Real Images + ...

… Data Augmentation (1)

Dwibedi et al. Cut, Paste and Learn: Surprisingly Easy Synthesis for Instance Detection. ICCV 2017.

Data Augmentation and Domain Randomization

Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World. Tobin et al. IROS 2017.

CosyPose: Consistent multi-view multi-object 6D pose estimation. Labbé et al. ECCV 2020.

How Domain Randomization Works

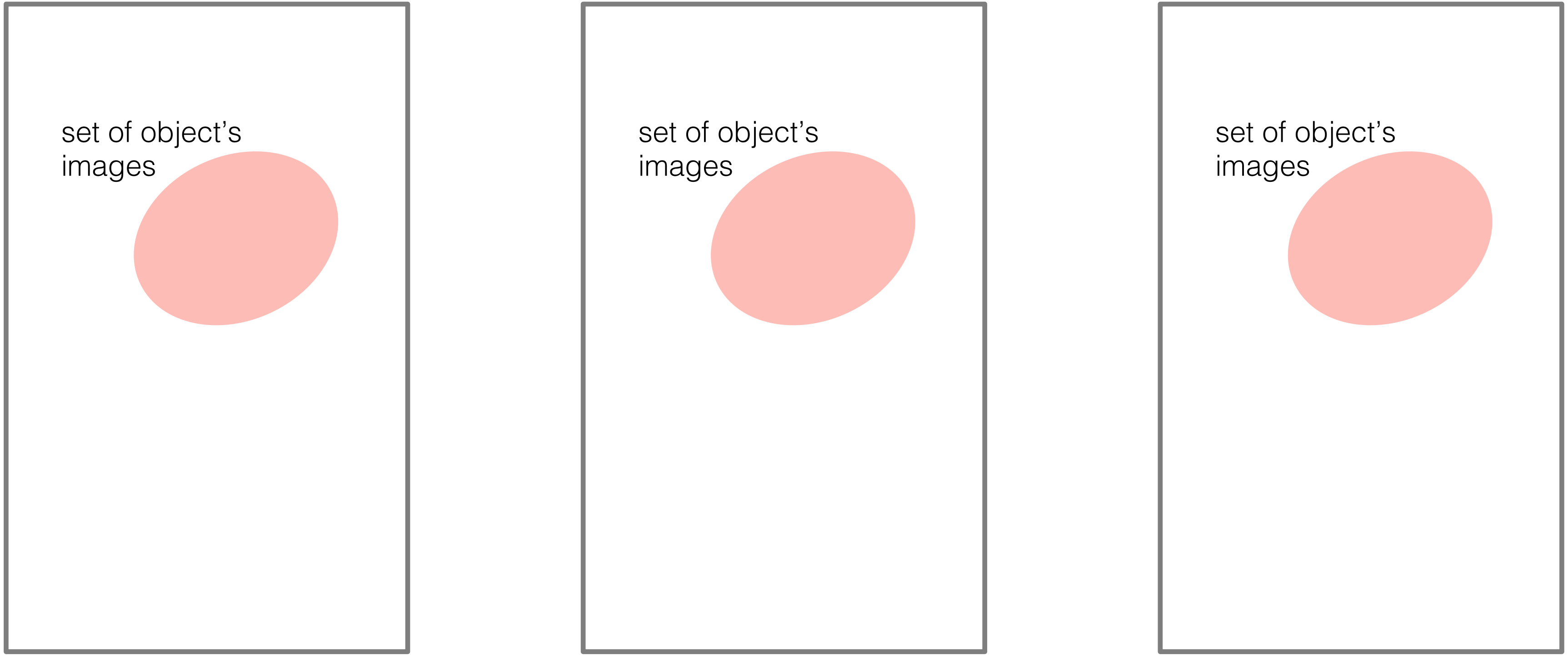


image space image space image space

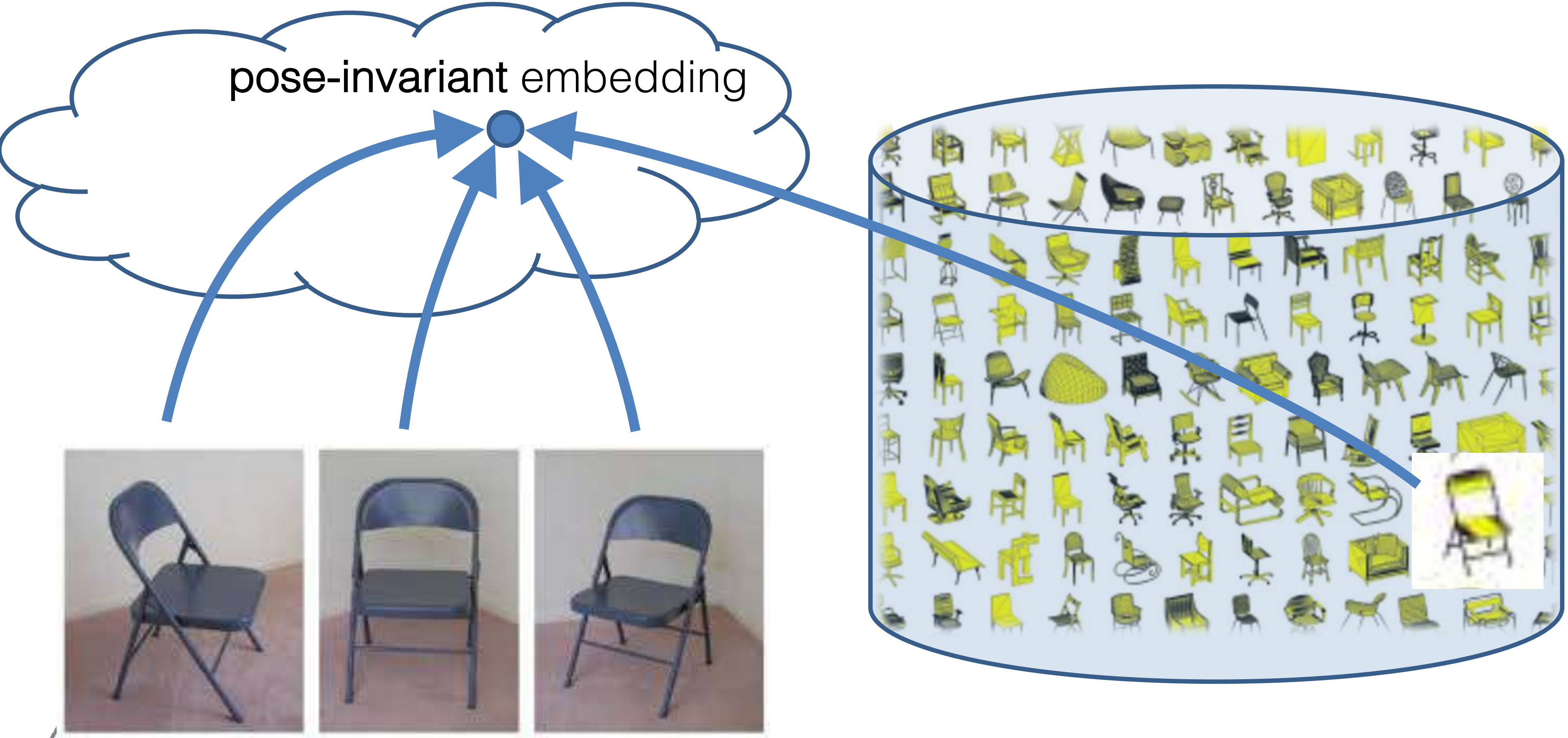
limiting the need for training data and for training time

- Considering object categories;
- Few-shot learning;
- ...

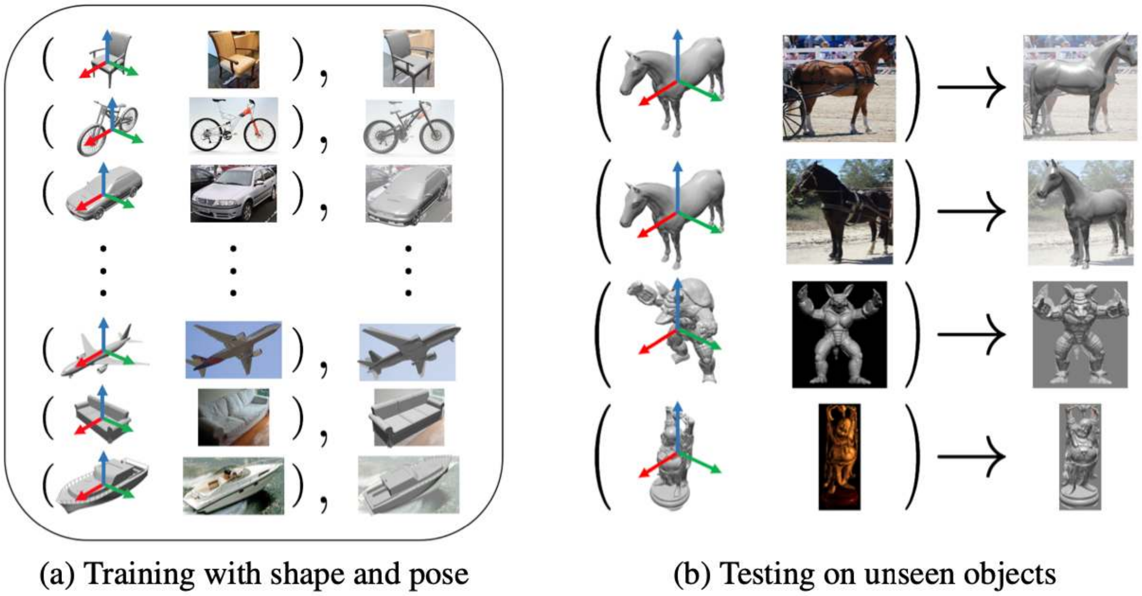
3D Pose Prediction for Object Categories

3D Pose Estimation and 3D Model Retrieval for Objects in the Wild. Alexander Grabner, Peter M. Roth, and Vincent Lepetit. CVPR 2018.

Pix3D: Dataset and Methods for Single-Image 3D Shape Modeling. Sun et al, 2018.


pose-invariant embedding

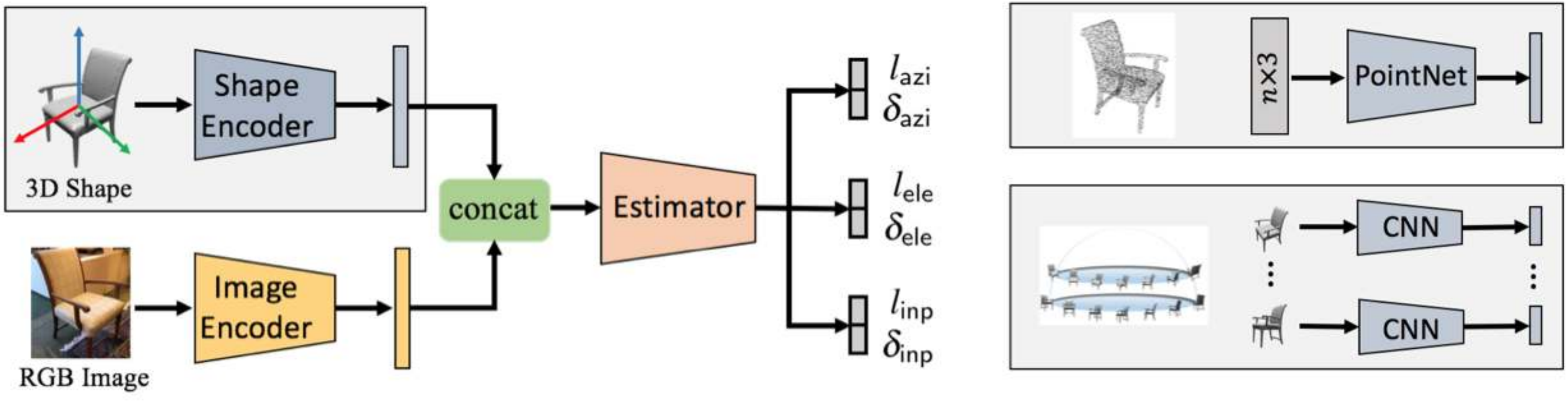
Pose from Shape



Pose from Shape: Deep Pose Estimation for Arbitrary 3D Objects. Xiao et al. BMVC 2019.

École des Ponts
ParisTech

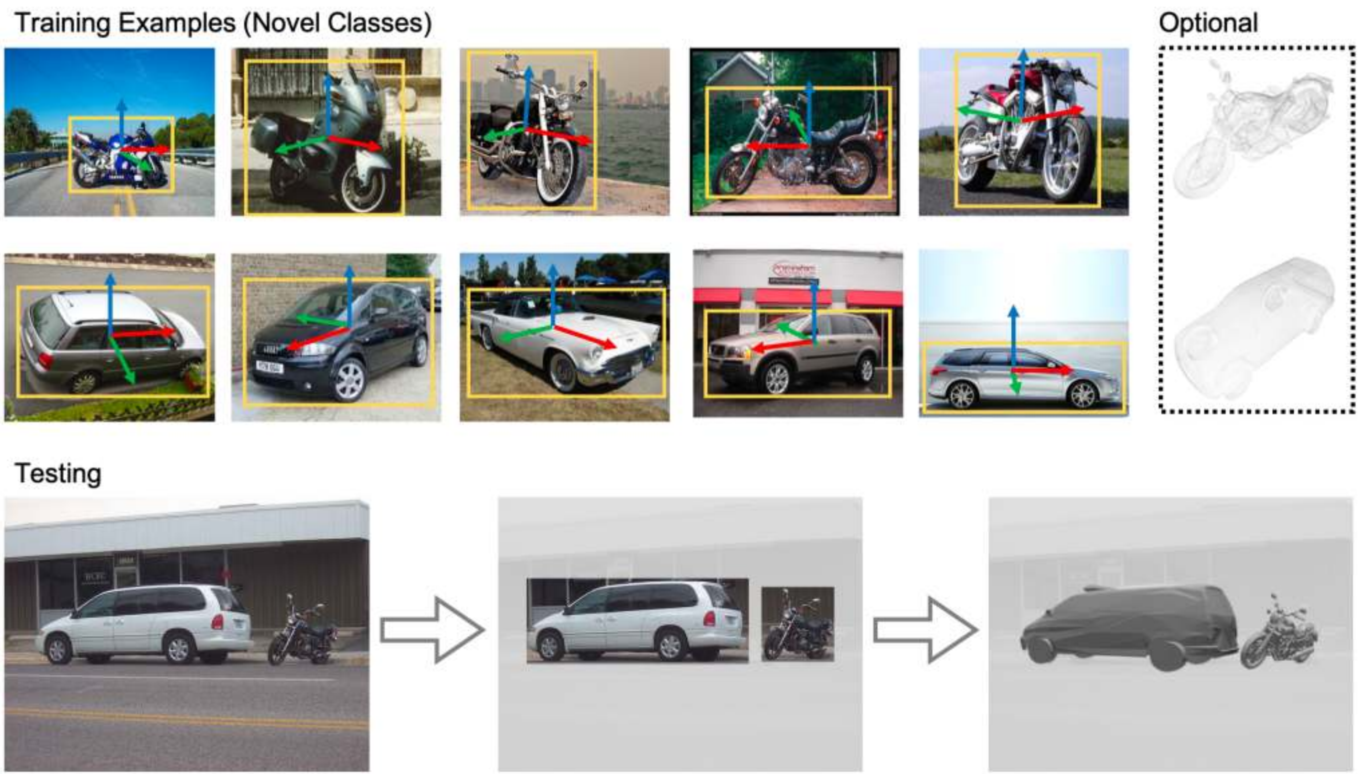
Pose from Shape



(a) Our pose estimation approach (b) Two possible shape encoders

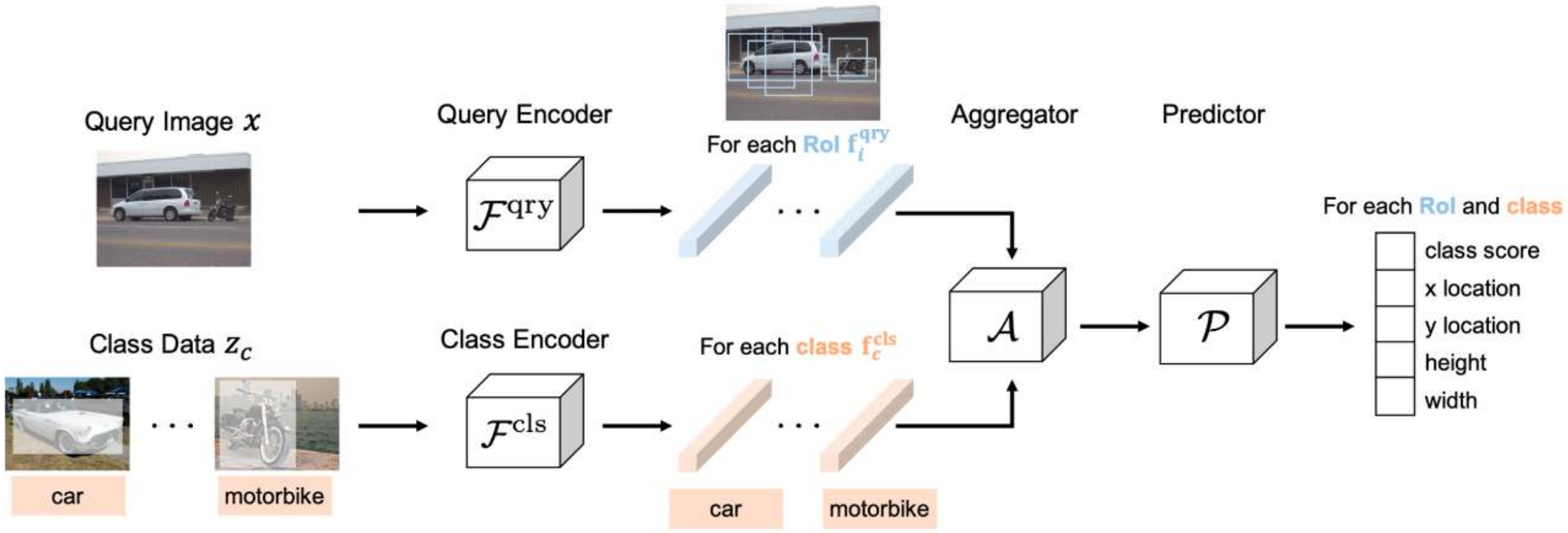
Pose from Shape: Deep Pose Estimation for Arbitrary 3D Objects. Xiao et al. BMVC 2019.

few-shot learning for 3D scene understanding



Few-shot Object Detection and Viewpoint Estimation for Objects in the Wild. Yang Xiao, Vincent Lepetit, Renaud Marlet. arXiv 2020.

few-shot learning for 3D scene understanding



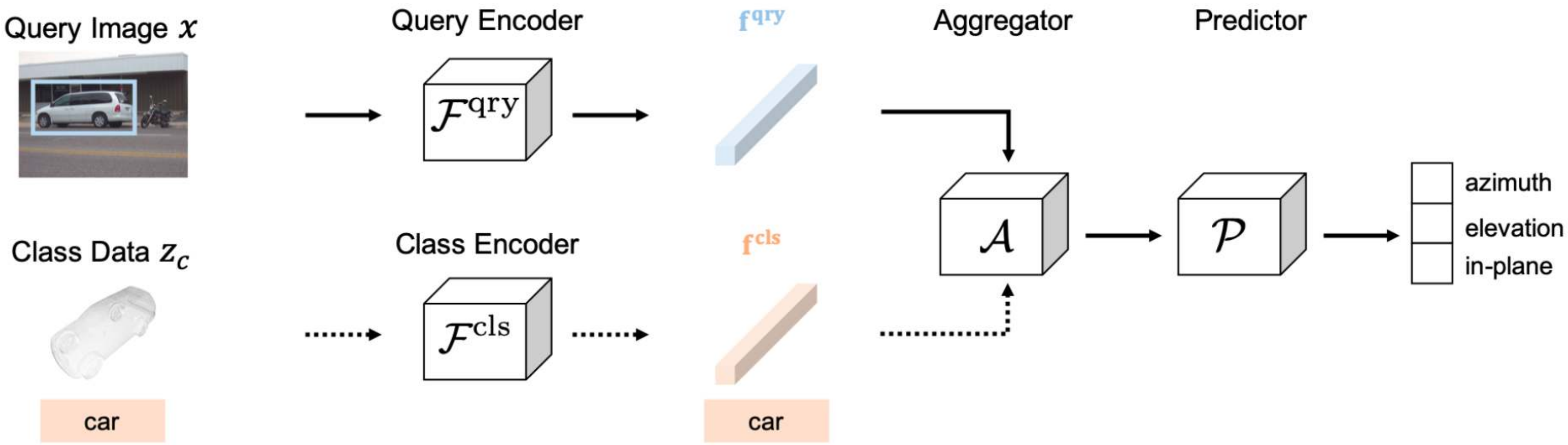
$$\text{cls}_{i,c} = \frac{\alpha \mathcal{A}(\mathrm{f}_i^{\text{qry}}, \mathrm{f}_c^{\text{cls}})^\top \mathrm{w}_c}{\|\mathcal{A}(\mathrm{f}_i^{\text{qry}}, \mathrm{f}_c^{\text{cls}})\| \|\mathrm{w}_c\|}$$

$$\mathcal{A}(\mathrm{f}^{\text{qry}}, \mathrm{f}^{\text{cls}}) = \mathrm{f}^{\text{qry}} \otimes \mathrm{f}^{\text{cls}}$$

Few-shot Object Detection and Viewpoint Estimation for Objects in the Wild. Yang Xiao, Vincent Lepetit, Renaud Marlet. arXiv 2021.

École des Ponts ParisTech

few-shot learning for 3D scene understanding



$$(\text{azi}, \text{ele}, \text{inp}) = \mathcal{P}\Big(\mathcal{A}(\mathrm{f}^{\text{qry}}, \mathrm{f}^{\text{cls}})\Big)$$

$$\text{with } \mathrm{f}^{\text{qry}} = \mathcal{F}^{\text{qry}}(\text{crop}(\text{img}(x), \text{box}(x))), \text{ and}$$

$$\mathrm{f}^{\text{cls}} = \mathcal{F}^{\text{cls}}(z_c), \; c = \text{cls}(x)$$

Few-shot Object Detection and Viewpoint Estimation for Objects in the Wild. Yang Xiao, Vincent Lepetit, Renaud Marlet. arXiv 2020.