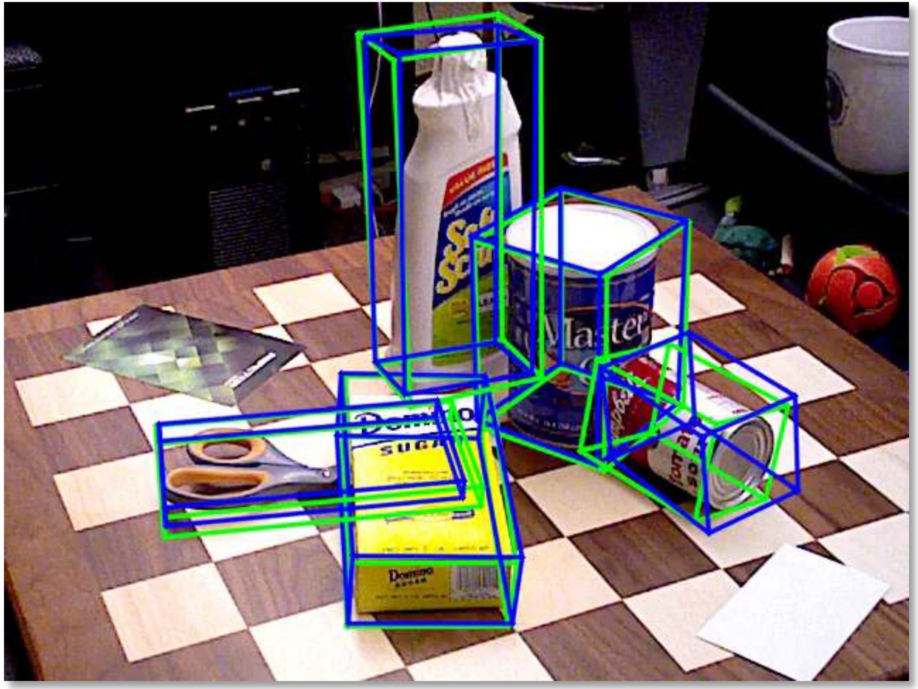
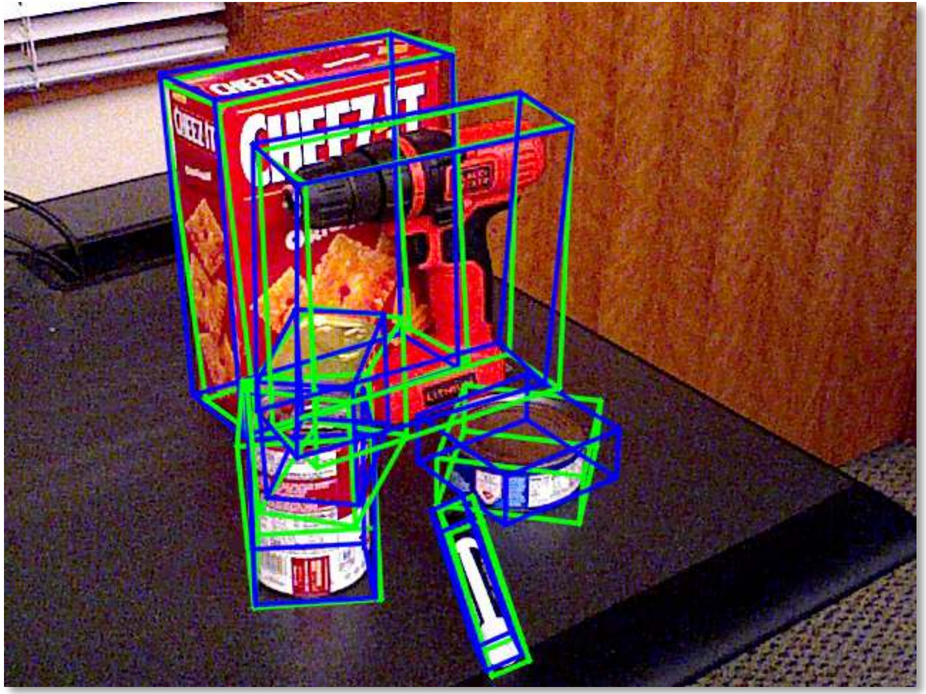
Deep Learning for Augmented Reality

Vincent Lepetit

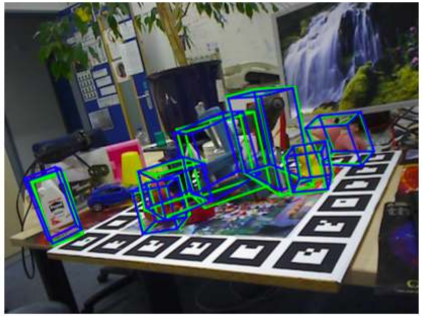
École des Ponts
ParisTech


Master

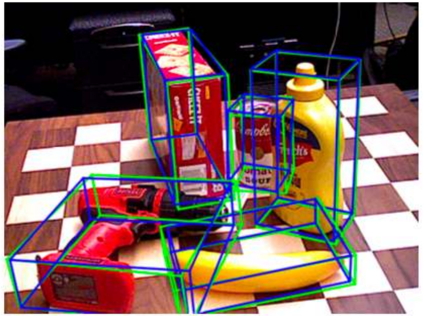

CHEEZ-IT

Typical Datasets

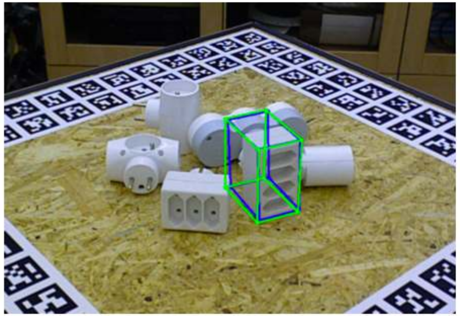
LineMOD



Occluded-LineMOD



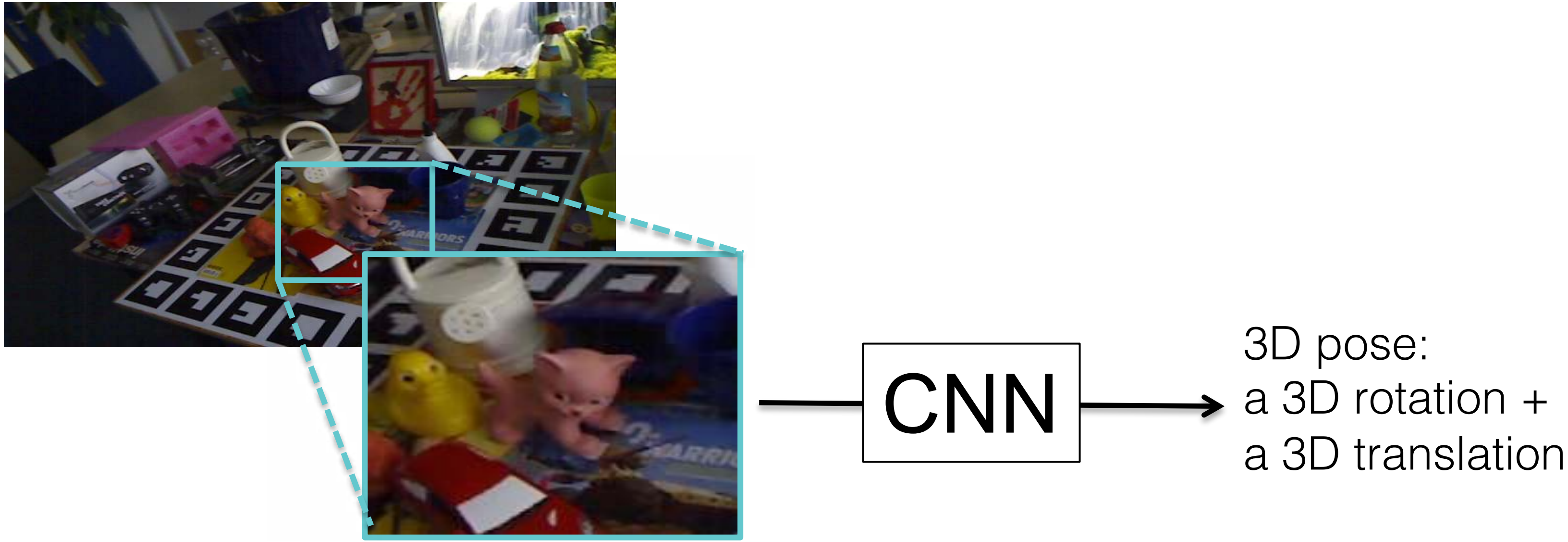
YCB-Video



T-LESS

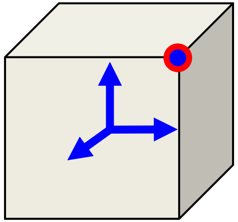
École des Ponts
ParisTech

Predicting a 3D Pose



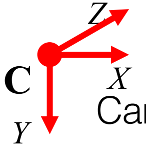
Object Coordinates to Camera Coordinates

Object coordinate system



$$\mathbf{M}_\mathrm{w} = \begin{pmatrix} X_\mathrm{w} \\ Y_\mathrm{w} \\ Z_\mathrm{w} \end{pmatrix}$$

$$\mathbf{M}_\mathrm{c} = \begin{pmatrix} X_\mathrm{c} \\ Y_\mathrm{c} \\ Z_\mathrm{c} \end{pmatrix}$$



Camera coordinate system

$$\begin{pmatrix} X_\mathrm{c} \\ Y_\mathrm{c} \\ Z_\mathrm{c} \end{pmatrix} = \begin{pmatrix} \mathbf{R} & \mathbf{T} \end{pmatrix} \begin{pmatrix} X_\mathrm{w} \\ Y_\mathrm{w} \\ Z_\mathrm{w} \\ 1 \end{pmatrix}$$

$$= \begin{pmatrix} R_{11} & R_{12} & R_{13} & T_1 \\ R_{21} & R_{22} & R_{23} & T_2 \\ R_{31} & R_{32} & R_{33} & T_3 \end{pmatrix} \begin{pmatrix} X_\mathrm{w} \\ Y_\mathrm{w} \\ Z_\mathrm{w} \\ 1 \end{pmatrix}$$

Parameterizing the Rotation Matrix

Possible Parameterizations of the Rotation Matrix

Rotation in 3D space has only 3 degrees of freedom.
Using the nine elements as its parameters would not be a good idea.

Possible parameterizations:

- Euler Angles;
- Quaternions;
- Exponential Map;
- ...

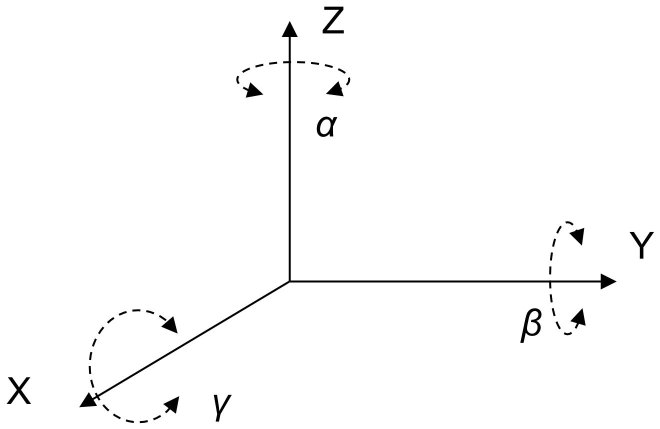
All have singularities, can be avoided by locally reparameterizing the rotation.

Euler Angles

Rotation defined by angles of rotation around the X-, Y-, and Z- axes.

Different conventions. For example:

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



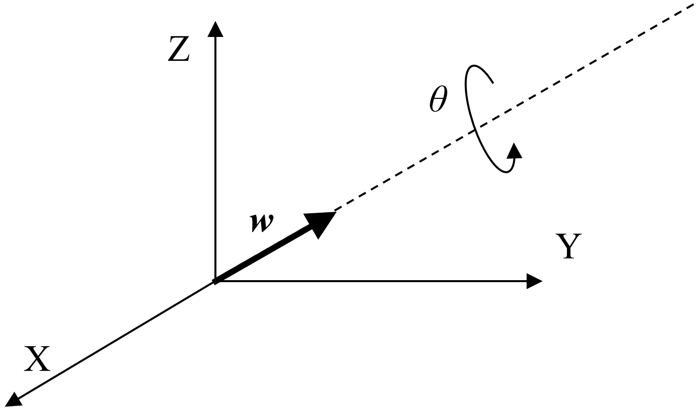
A Unit Quaternion

Quaternions are hyper-complex numbers that can be written as the linear combination $a+bi+cj+dk$, with $i^2 = j^2 = k^2 = ijk = -1$.

Can also be interpreted as a scalar plus a 3- vector: $(a, \mathbf{v})$.

A rotation about the unit vector $\boldsymbol{w}$ by an angle θ can be represented by the unit quaternion:

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



A Unit Quaternion

Quaternions are hyper-complex numbers that can be written as the linear combination $a+bi+cj+dk$, with $i^2 = j^2 = k^2 = ijk = -1$.
Can also be interpreted as a scalar plus a 3- vector: $(a, \mathbf{v})$.
A rotation about the unit vector $\boldsymbol{w}$ by an angle θ can be represented by the *unit quaternion*: $q = \left(\cos\frac{\theta}{2}, \boldsymbol{w}\sin\frac{\theta}{2}\right)$

To rotate a 3D point $\mathbf{M}$: write it as a quaternion p = (0, $\mathbf{M}$), and take the rotated point p' to be

$$p' = q.p\bar{q} \quad \text{with} \quad \bar{q} = \left(\cos\frac{\theta}{2}, -\boldsymbol{w}\sin\frac{\theta}{2}\right)$$

No gimbal lock.

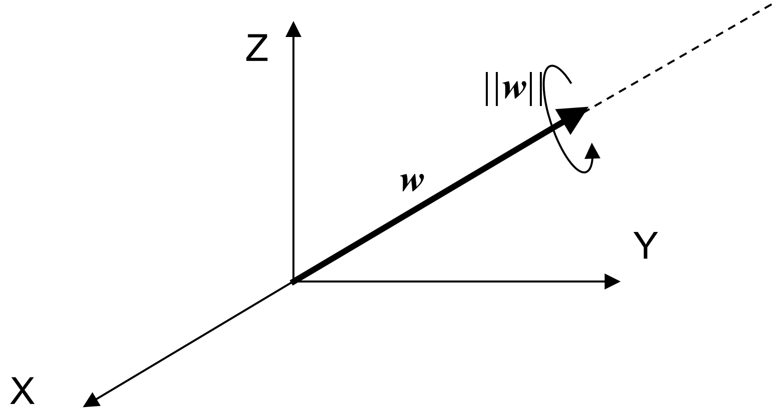
Parameterization of the rotation using the 4 coordinates of a quaternion $\boldsymbol{q}$:

1. No constraint and rotation performed using $\frac{q}{\|q\|}$ → singularity: kq yields the same rotation whatever the value of $k > 0$;
2. Additional constraint: norm of q must be constrained to be equal to 1, for example by adding the quadratic term $K(1 - \|q\|^2)$.

Exponential Maps

No gimbal lock;
No additional constraints;
Singularities occur in a region that can easily be avoided.

Parameterization by a 3D vector $\boldsymbol{w} = [w_1, w_2, w_3]^T$: Rotation around the axis of direction $\boldsymbol{w}$ of an amount of $||\boldsymbol{w}||$



École des Ponts ParisTech

Rodrigues' Formula

$$\boldsymbol{\Omega} = \begin{bmatrix} 0 & -w_z & w_y \\ w_z & 0 & -w_x \\ -w_y & w_x & 0 \end{bmatrix}$$

The rotation matrix is given by:

$$\begin{aligned} \mathbf{R}(\boldsymbol{\Omega}) &= \exp(\boldsymbol{\Omega}) = \mathbf{I} + \boldsymbol{\Omega} + \frac{1}{2!}\boldsymbol{\Omega}^2 + \frac{1}{3!}\boldsymbol{\Omega}^3 + \ldots \\ &= \mathbf{I} + \frac{\sin\theta}{\theta}\boldsymbol{\Omega} + \frac{(1-\cos\theta)}{\theta^2}\boldsymbol{\Omega}^2 \quad \text{(Rodrigues' formula)} \end{aligned}$$

Not singular for small values of θ even if we divide by θ (see Taylor expansions).

6D / 2 vectors

$$e_1' = \frac{e_1}{||e_1||_2}$$

$$e_3' = \frac{e_1' \wedge e_2}{||e_2||_2}$$

$$e_2' = e_3' \wedge e_1',$$

Zhou, Y., Barnes, C., Lu, J., Yang, J., Li, H.: On the continuity of rotation representations in neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 5745–5753

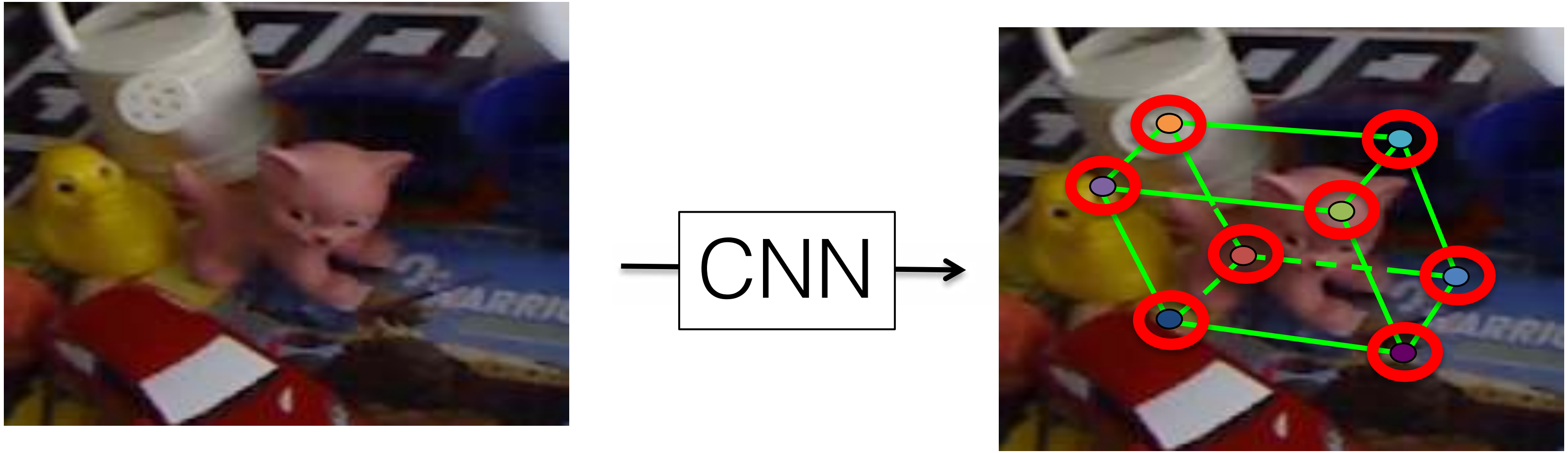
Predicting the Pose: Loss

$$\min_{\Theta} \sum_{(I,\mathbf{p}_{\mathrm{GT}})} \|\mathbf{p}_{\mathrm{GT}} - f_1(I;\Theta)\|^2$$

I → f_1 → $\mathbf{p}$

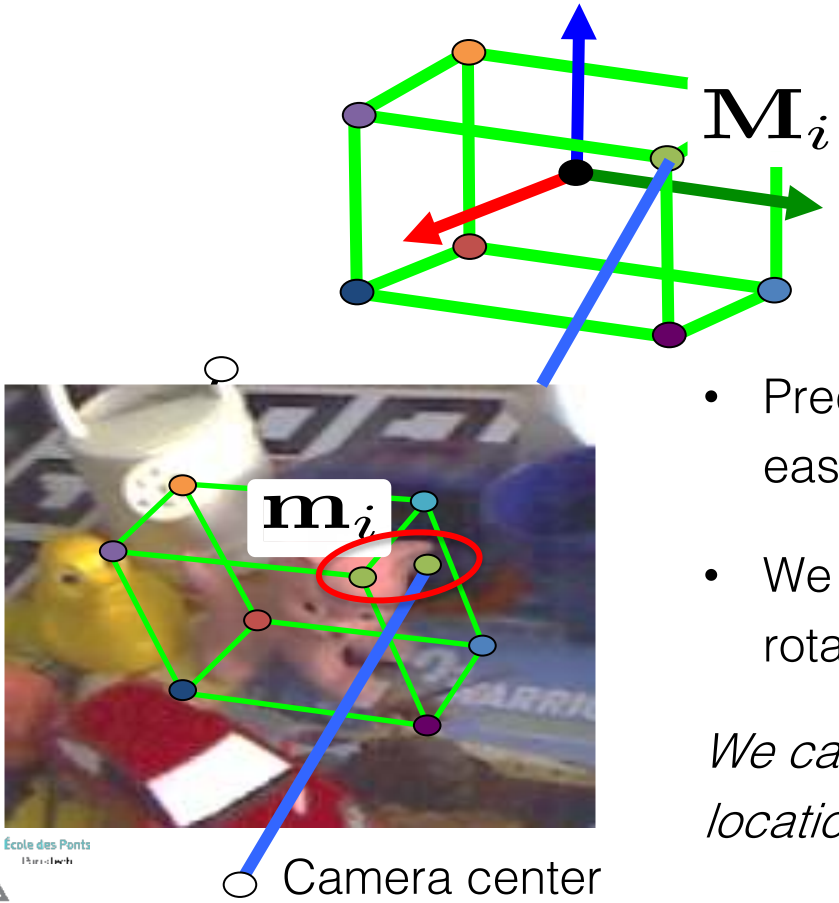
Alternative Pose Representations (1)

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.

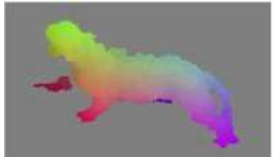
3D Pose Estimation from Correspondences



- Predicting 2D locations from an image is an easier regression task;
- We do not need a representation of the 3D rotation;

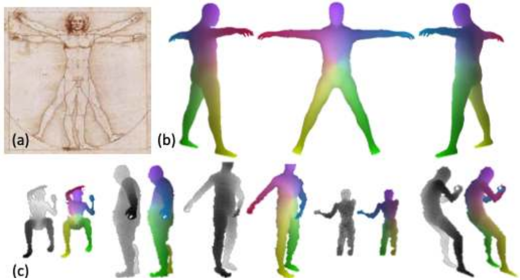
We can compute the 3D pose from these 2D locations.

Alternative Pose Representations (2)

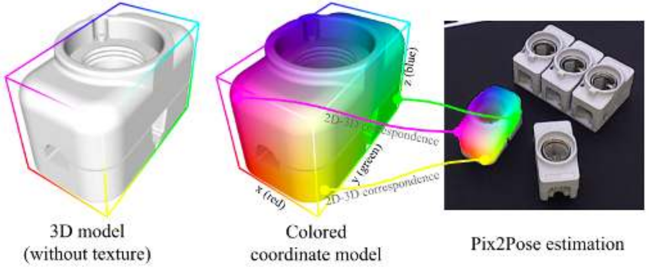


E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother. Learning 6D Object Pose Estimation using 3D Object Coordinates. ECCV 2014.

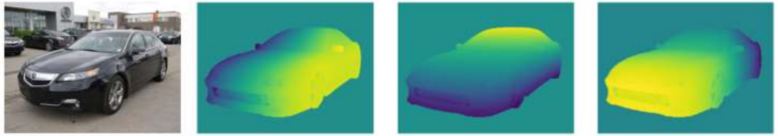
Normalized Object Coordinate Space for Category-Level 6D Object Pose and Size Estimation. Wang et al., CVPR 2019.



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

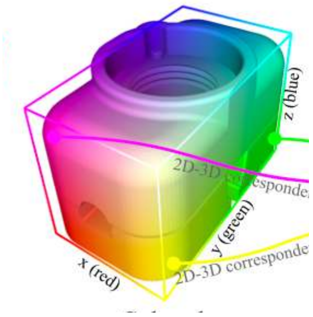


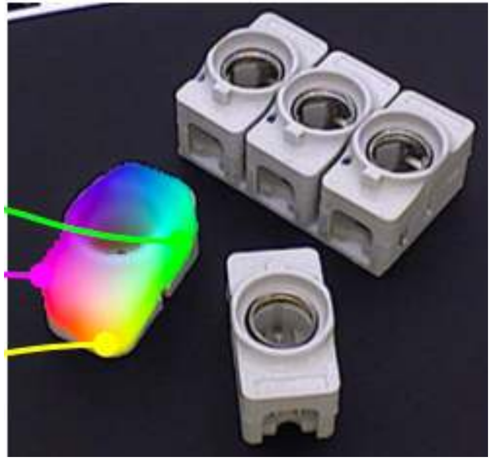
Pix2Pose: Pixel-Wise Coordinate Regression of Objects for 6D Pose Estimation. Park et al., CVPR 2019.



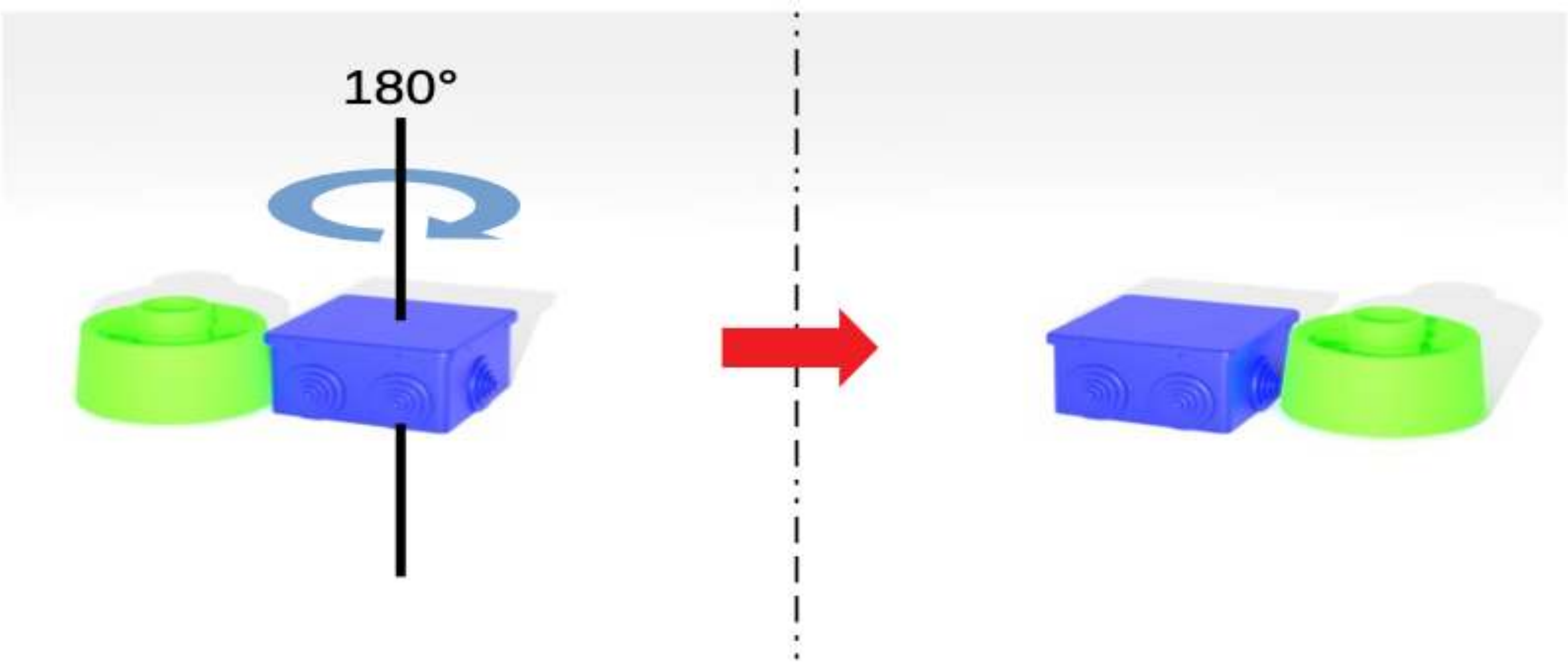
location fields. Wang et al., ECCV 2018

How to use 3D coordinate maps

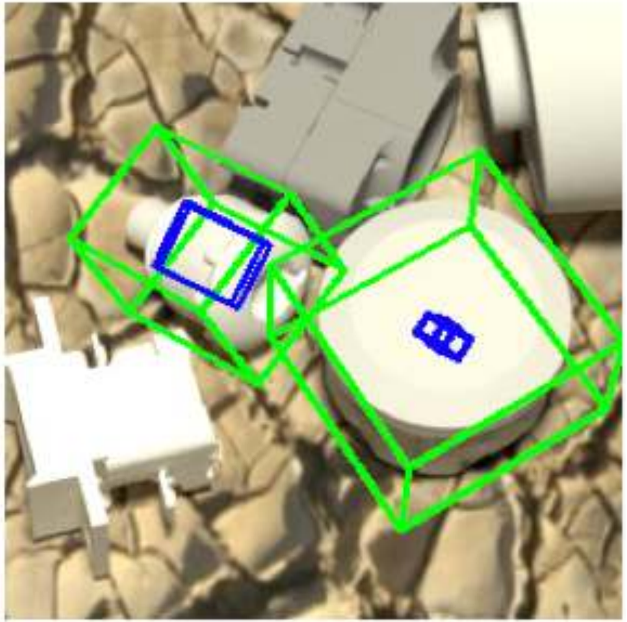


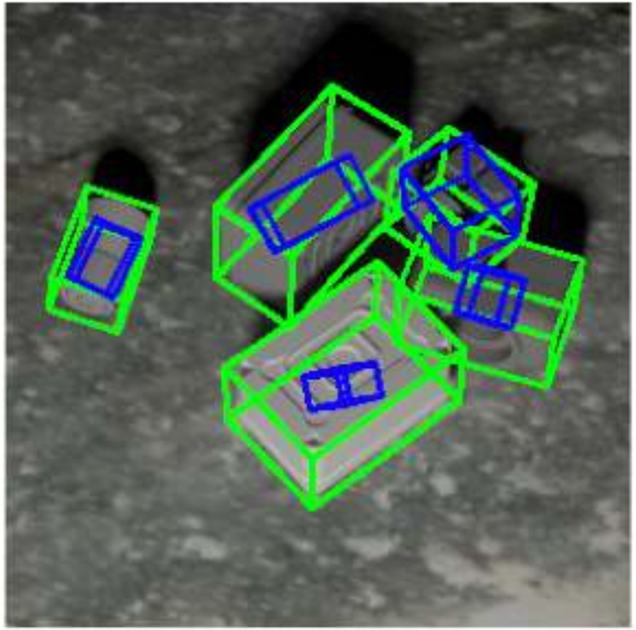


Symmetrical Objects



Symmetrical Objects Can Be Problematic





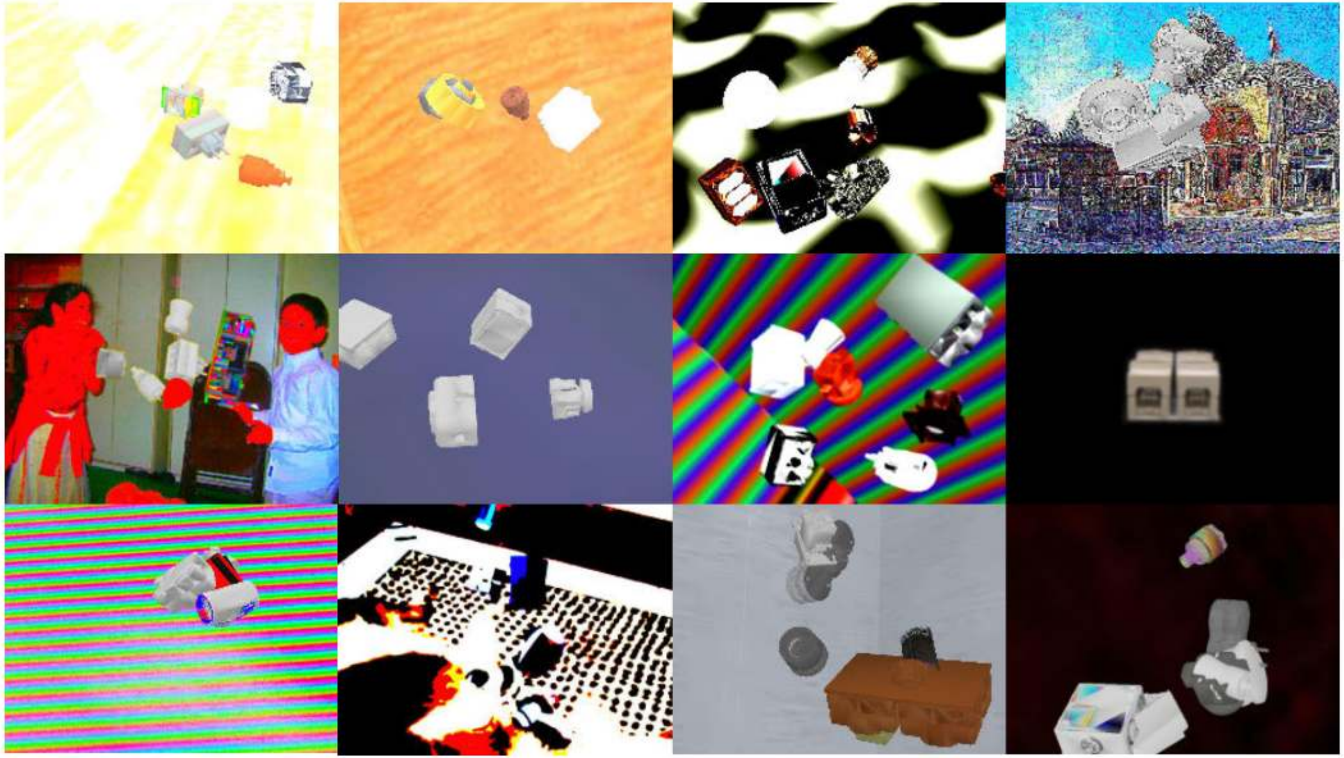
Symmetrical Objects: Solution

$$D_l(T_1, T_2) = \min_{S \in S(l)} \frac{1}{|\mathcal{X}_l|} \sum_{\mathrm{x} \in \mathcal{X}_l} ||T_1 S \mathrm{x} - T_2 \mathrm{x}||_2$$

Training Set:
About 200 Real Images + ...

… Data Augmentation (1)

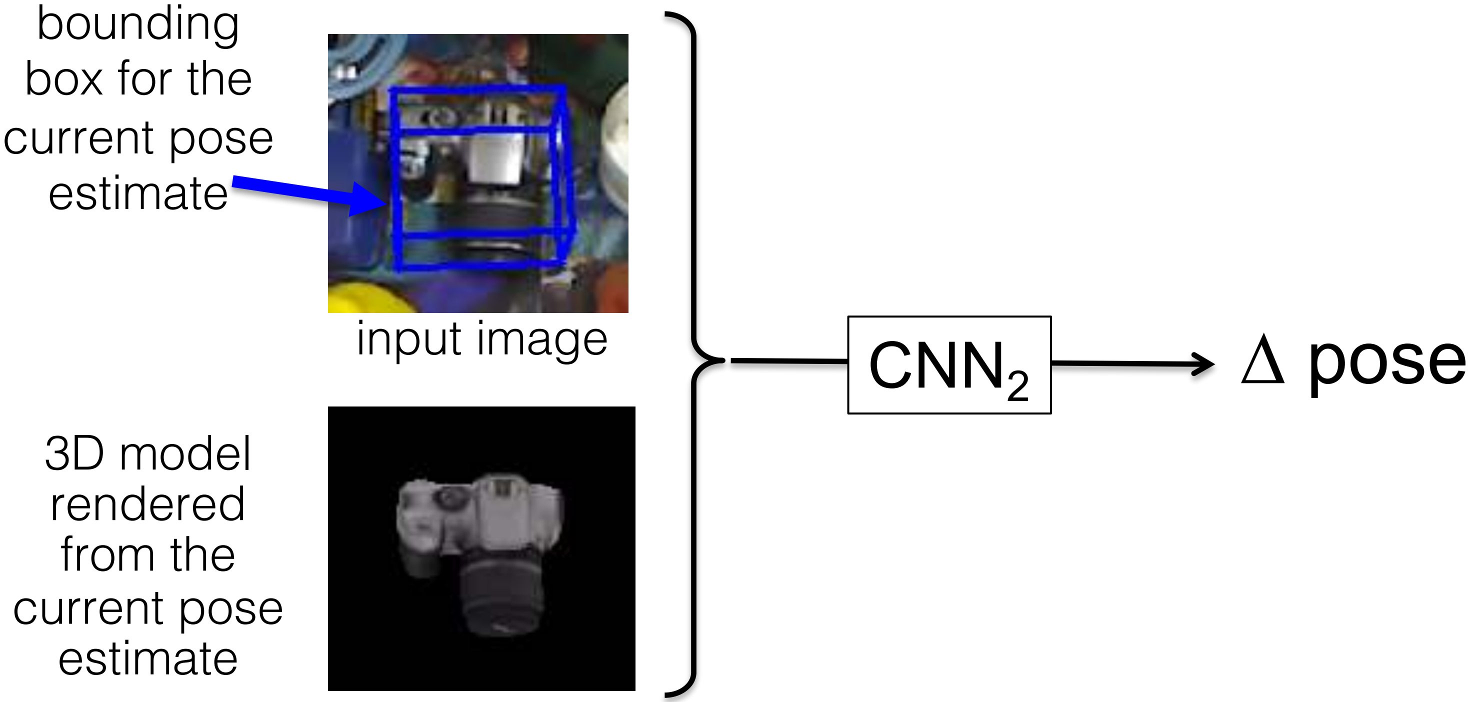
Data Augmentation and Domain Randomization



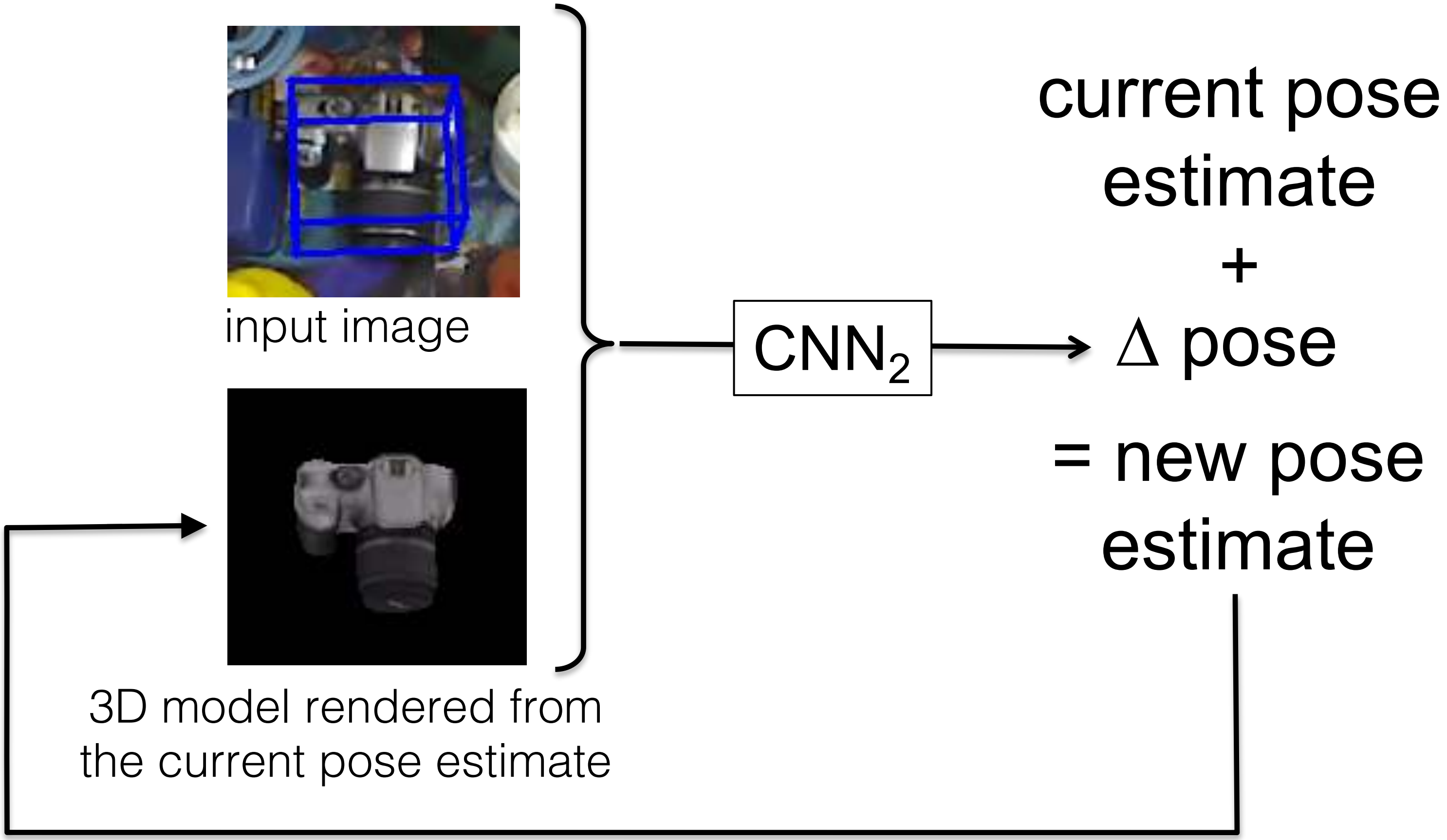
École des Ponts
ParisTech

How Domain Randomization Works

Refining the Pose

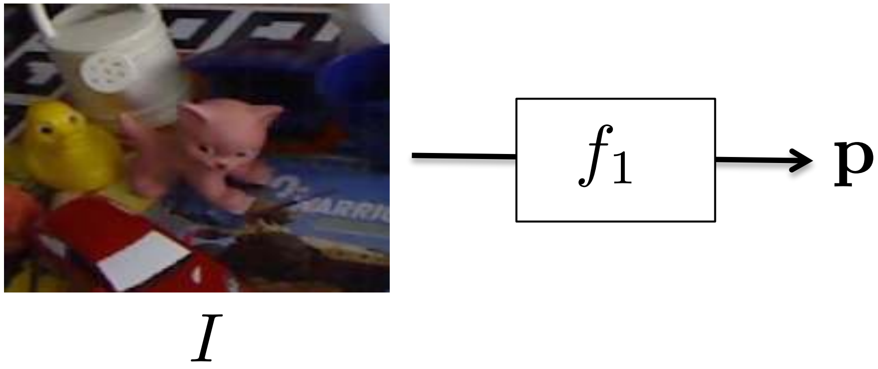


Refining the Pose



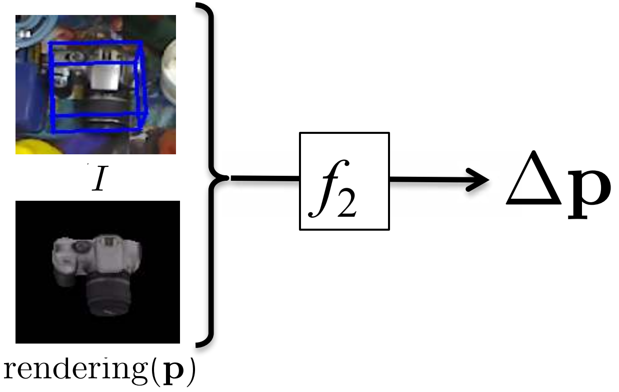
Predicting the Pose: Loss

$$\min_{\Theta} \sum_{(I,\mathbf{p}_{\mathrm{GT}})} \|\mathbf{p}_{\mathrm{GT}} - f_1(I;\Theta)\|^2$$



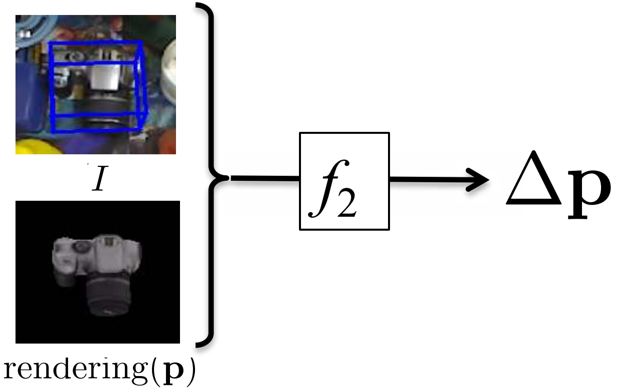
Refining the Pose: Loss

$$\Delta \mathbf{p} = f_2(I, \text{rendering}(\mathbf{p}); \Omega)$$



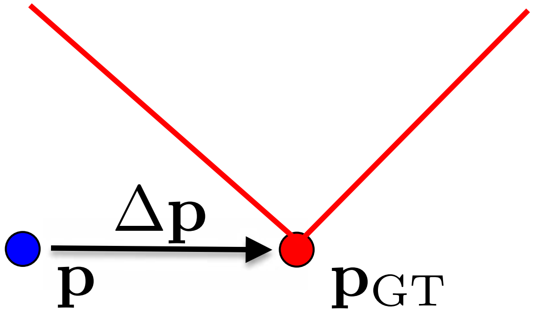
Refining the Pose: Loss

$$\Delta \mathbf{p} = f_2(I, \text{rendering}(\mathbf{p}); \Omega)$$



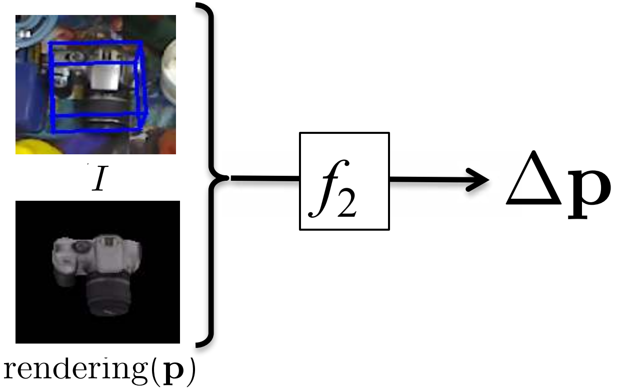
$$\min_{\Omega} \sum_{(I, \mathbf{p}_{\text{GT}})} \sum_{\mathbf{p} \in \mathcal{N}(\mathbf{p}_{\text{GT}})} \| \mathbf{p} - \mathbf{p}_{\text{GT}} \|$$

$$\text{with } \mathbf{p}' = \mathbf{p} + f_2(I, \text{rendering}(\mathbf{p}); \Omega)$$



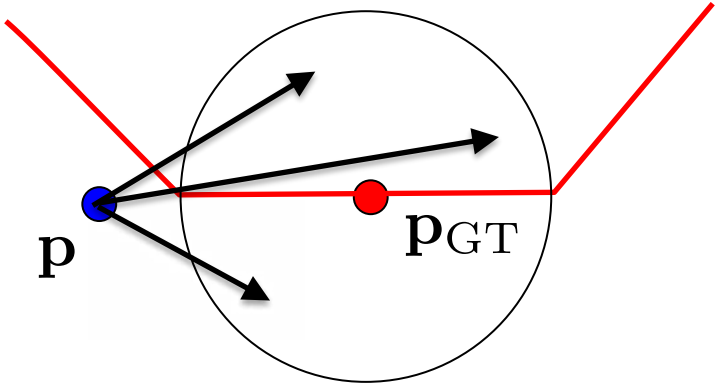
Refining the Pose: Loss

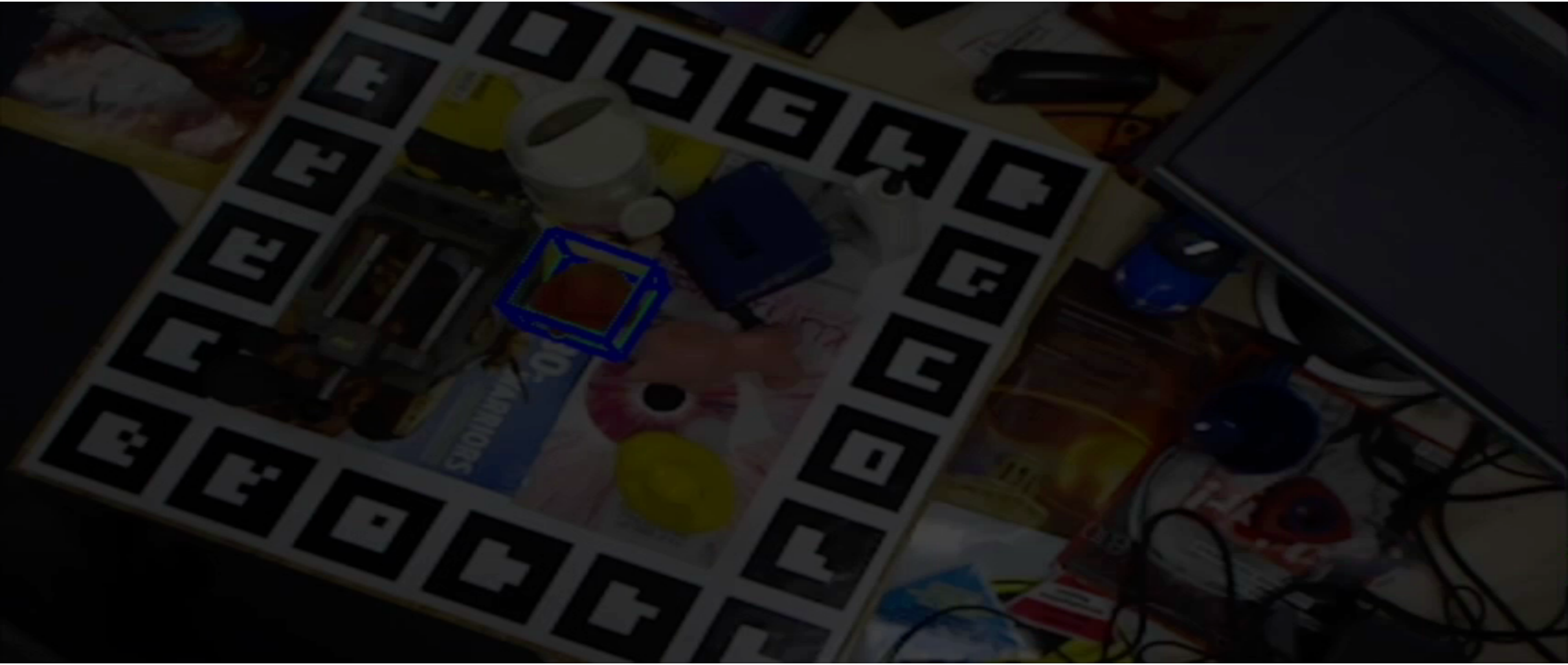
$$\Delta \mathbf{p} = f_2(I, \text{rendering}(\mathbf{p}); \Omega)$$



$$\min_{\Omega} \sum_{(I, \mathbf{p}_{\text{GT}})} \sum_{\mathbf{p} \in \mathcal{N}(\mathbf{p}_{\text{GT}})} \max(0, \|\mathbf{p}' - \mathbf{p}_{\text{GT}}\| - \lambda \|\mathbf{p} - \mathbf{p}_{\text{GT}}\|)$$

$$\text{with } \mathbf{p}' = \mathbf{p} + f_2(I, \text{rendering}(\mathbf{p}); \Omega)$$





Object Detection: How?

Object Detection: How?

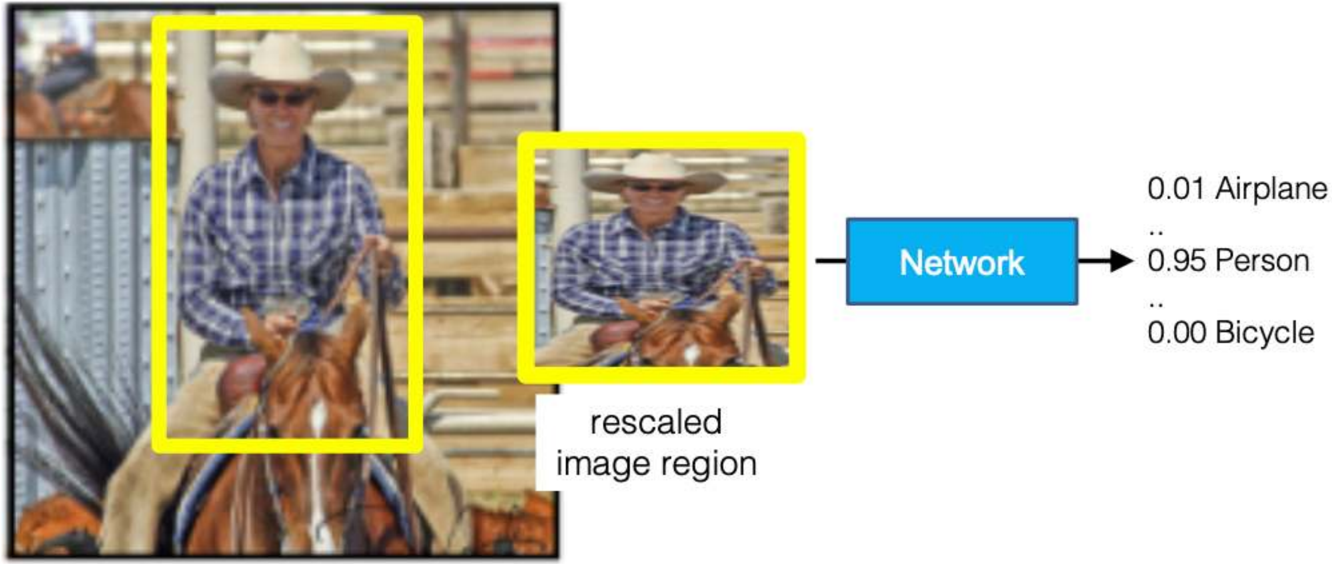
R-CNN (1)

Input Image

Region Proposals
No learning (yet)

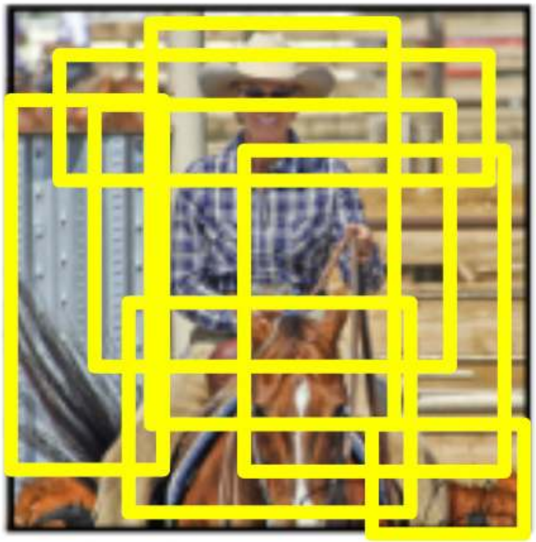
R.B. Girshick et al. "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation". In: *Conference on Computer Vision and Pattern Recognition*. 2014, pp. 580–587.

R-CNN (2)


rescaled
image region
Network
0.01 Airplane
..
0.95 Person
..
0.00 Bicycle

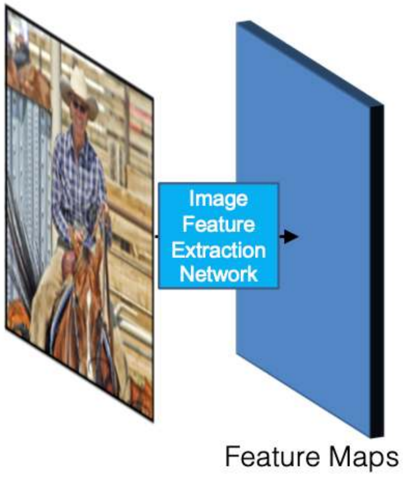
R-CNN (3)

Problem: In practice, many region proposals. R-CNN inefficient since image locations are processed many times.



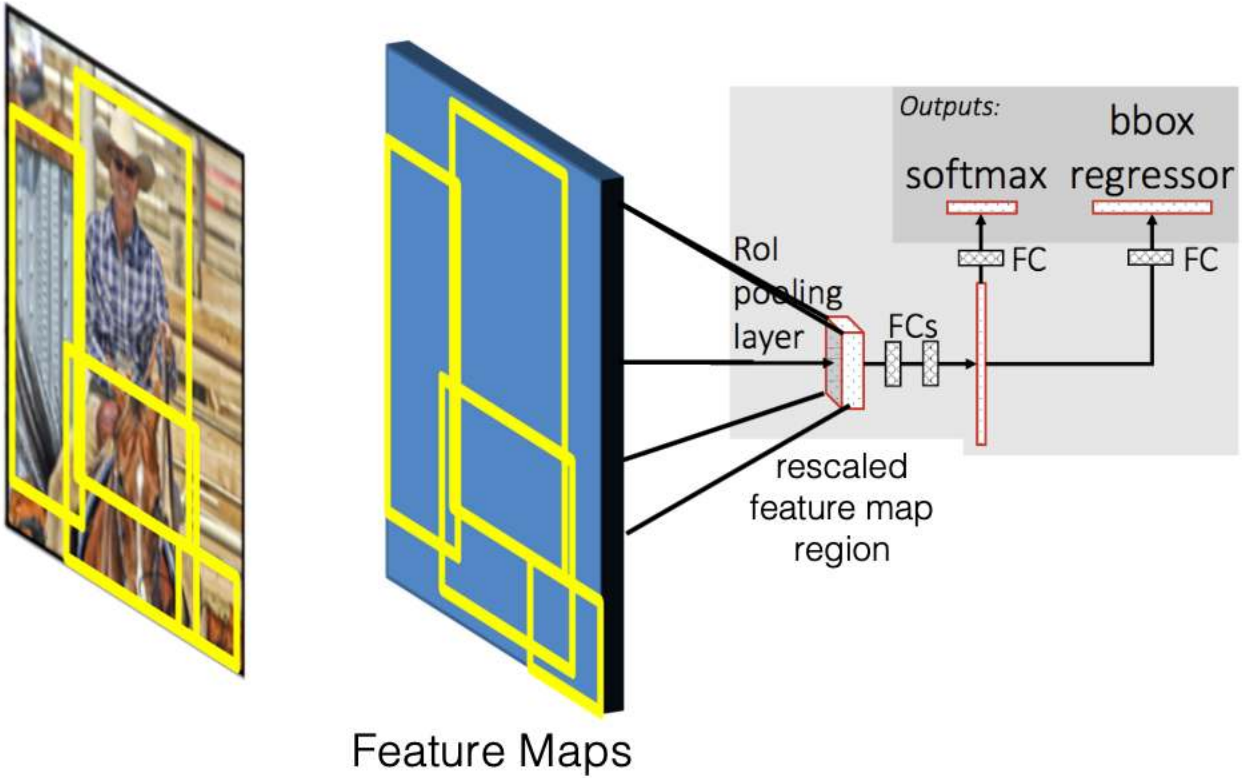
Fast-RCNN (1)

Convolutions are applied only once to the image to extract image features: Much faster.

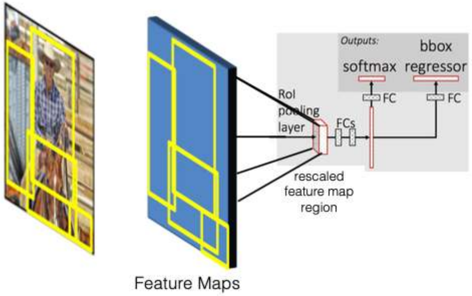


Ross B. Girshick. "Fast R-CNN". In: *International Conference on Computer Vision*. 2015.

Fast-RCNN (2)


Outputs:
softmax
bbox regressor
FC
FC
RoI pooling layer
FCs
rescaled feature map region
Feature Maps

Fast R-CNN: Loss Function (1)



Loss function for 1 region:

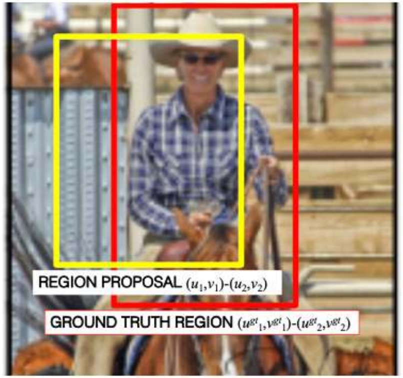
$$\mathcal{L}(\Theta) = -\log p_c(f_{\text{cl}}(\mathbf{x};\Theta)) + \lambda 1_{[c \geq 1]} \mathcal{L}_{\text{bbox}} \,, \qquad (15)$$

where:

- $\mathbf{x}$ is the rescaled region in the feature maps;
- c is the true class for $\mathbf{x}$. $c = 0$ corresponds to the background.
- $\mathcal{L}_{\text{bbox}}$ is a loss term to refine the region bounding box (see next slide);
- λ is a weight.

École des Ponts
ParisTech

Fast R-CNN: Loss Function (2)



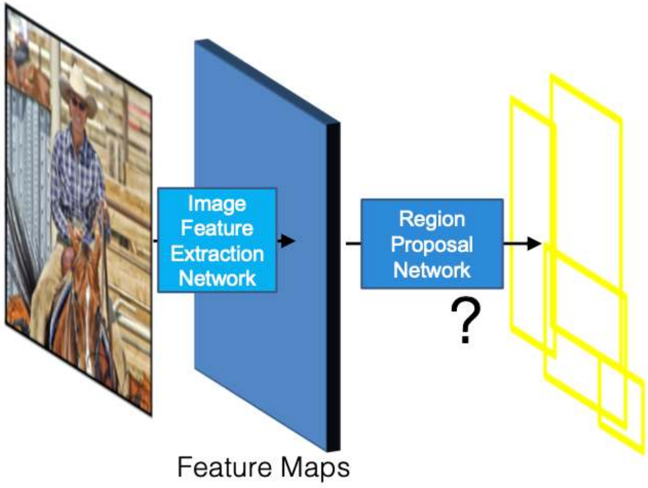
$\mathcal{L}_{\text{bbox}}$ is a loss term to refine the region bounding box:

$$\mathcal{L}_{\text{bbox}} = (u_1 + f_{\text{bbox}}(\mathbf{x};\Theta)[0] - u_1^{gt})^2 + (v_1 + f_{\text{bbox}}(\mathbf{x};\Theta)[1] - v_1^{gt})^2 + ..$$

where $(u_1, v_1) \times (u_2, v_2)$ are the coordinates of the region bounding box, and $(u_1^{gt}, v_1^{gt}) \times (u_2^{gt}, v_2^{gt})$ are the coordinates of the region bounding box.

Faster R-CNN

Learns to predict the region proposals:



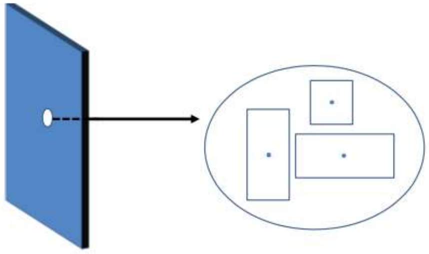
Challenges: the number of regions varies with the image, each region has a different size.

S. Ren et al. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In: *Advances in Neural Information Processing Systems*. 2015.

Faster R-CNN: Region Proposal Network

For each 2D location in the feature maps: consider 3 "anchor boxes" and predict for each anchor box:

- If the anchor box overlaps with an object;
- An offset to adapt the anchor box.

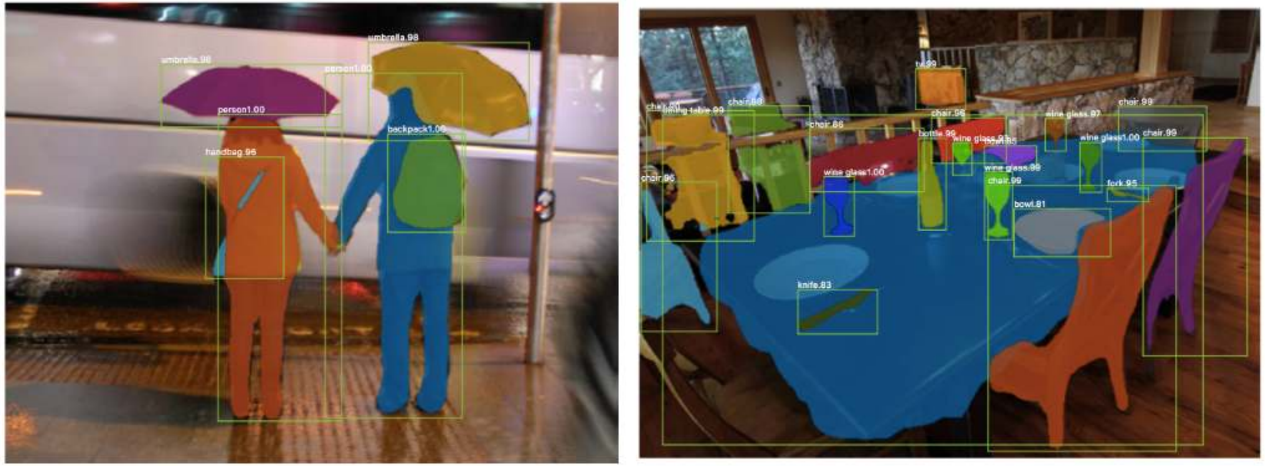


Loss function for 1 image $\mathbf{x}$ with $\mathcal{B} = \{B_i\}_i$ ground truth bounding boxes:

$$\mathcal{L}(\Theta_2) = -\sum_{A \in \mathcal{A}} \log p_{c(A,\mathcal{B})}(g(\mathbf{x};\Theta_2)[A]) + \lambda c(A,\mathcal{B})\mathcal{L}_{\text{bbox}}\,, \qquad (16)$$

with $c(A,\mathcal{B}) = 1$ if Anchor box A overlaps with at least one bounding box B_i in $\mathcal{B}$, 0 otherwise.

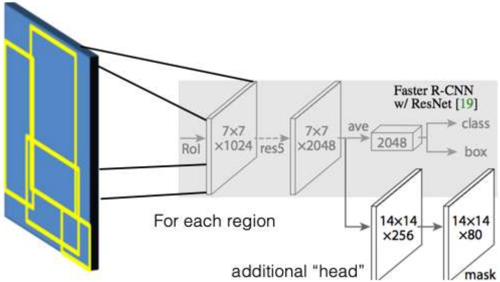
Mask-RCNN



Kaiming He et al. “Mask R-CNN”. In: *International Conference on Computer Vision*. 2017.

Mask-RCNN (2)

In addition to predicting the class and the “delta bounding box”, predict a binary mask for each possible class.



Loss function for one region:

$$\mathcal{L}(\Theta) = -\log p_c(f_{\text{cl}}(\mathbf{x};\Theta)) + \lambda 1_{[c\geq 1]}(\mathcal{L}_{\text{bbox}} + \lambda_2 \mathcal{L}_{\text{mask}}) \,, \qquad (17)$$

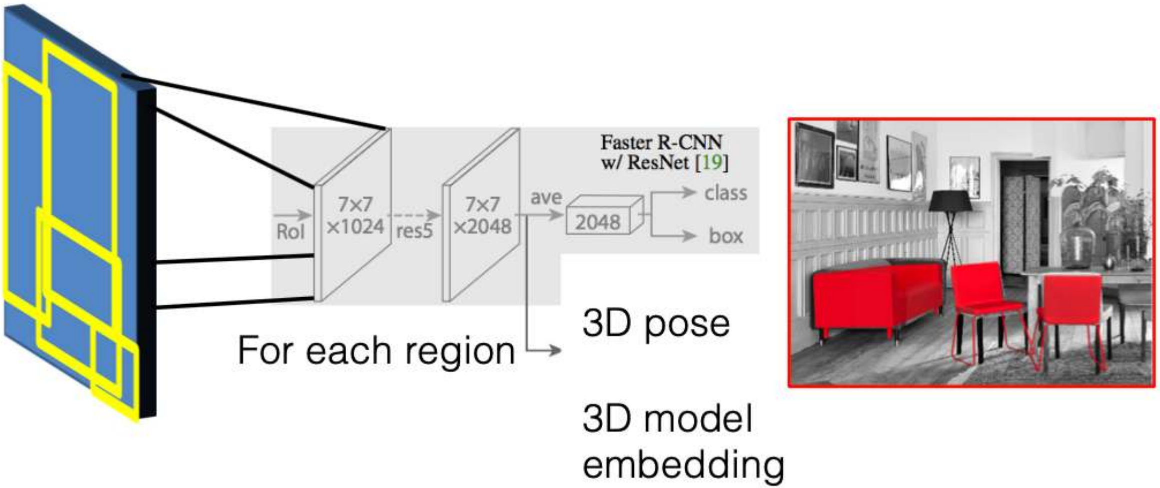
where:

- c is the true class for $\mathbf{x}$. $c=0$ corresponds to the background.
- $\mathcal{L}_{\text{mask}}$ is a loss term to refine the region bounding box:

$$\mathcal{L}_{\text{mask}} = \| f_{\text{mask}}(\mathbf{x};\Theta)[c] - m\|^2$$

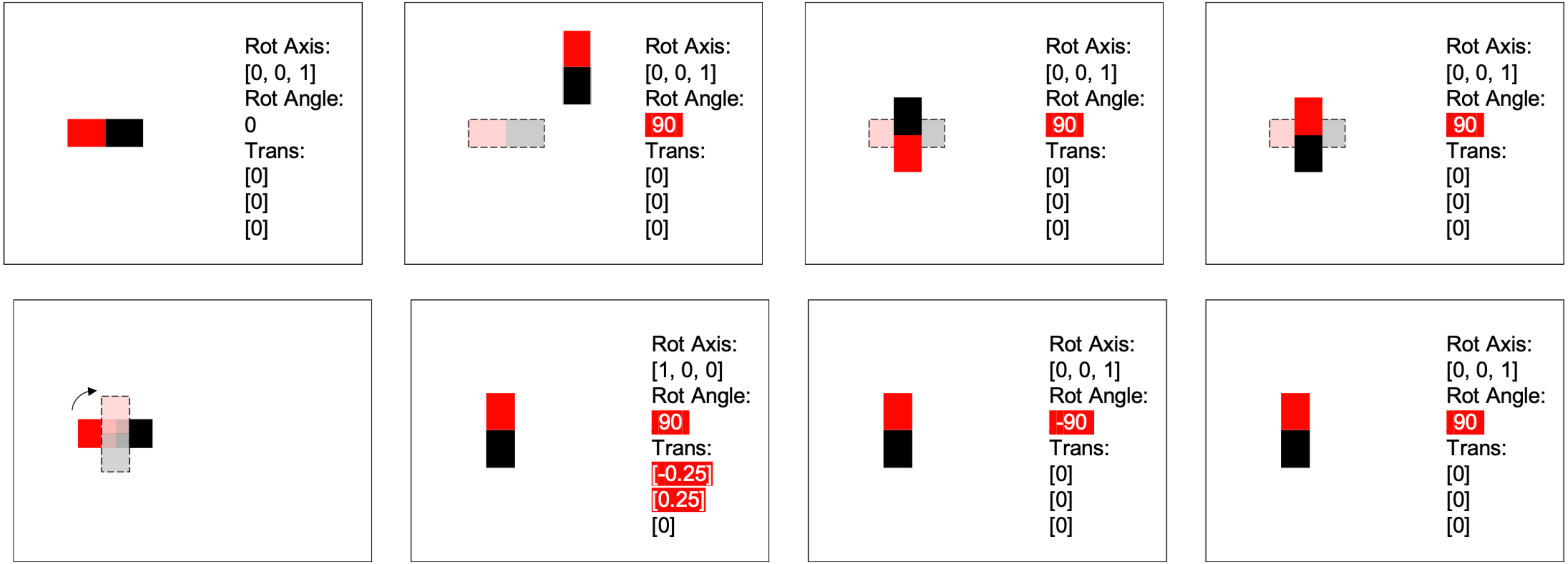
- m is the ground truth mask for the region.

Mask-RCNN: Extension to 3D



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

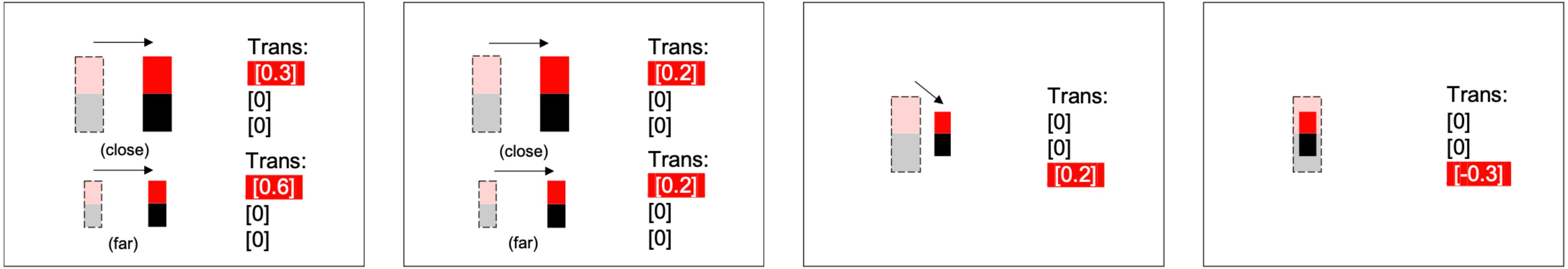
DeepIM: Decoupled Coordinates (R)



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

École des Ponts
ParisTech

DeepIM: Decoupled Coordinates (T)



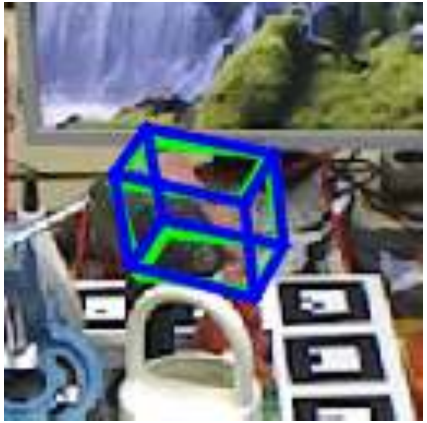
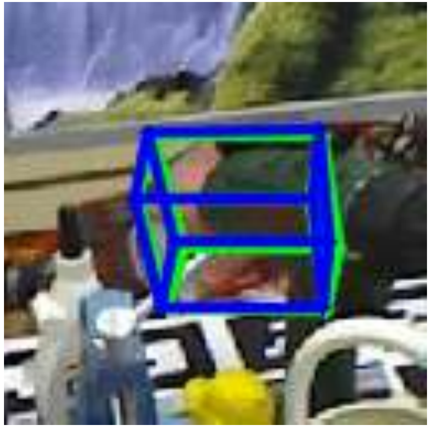
(a) Camera coord. xy-plane translation

(b) Disentangled coord. xy-plane translation

(c) Camera coord. z-axis translation

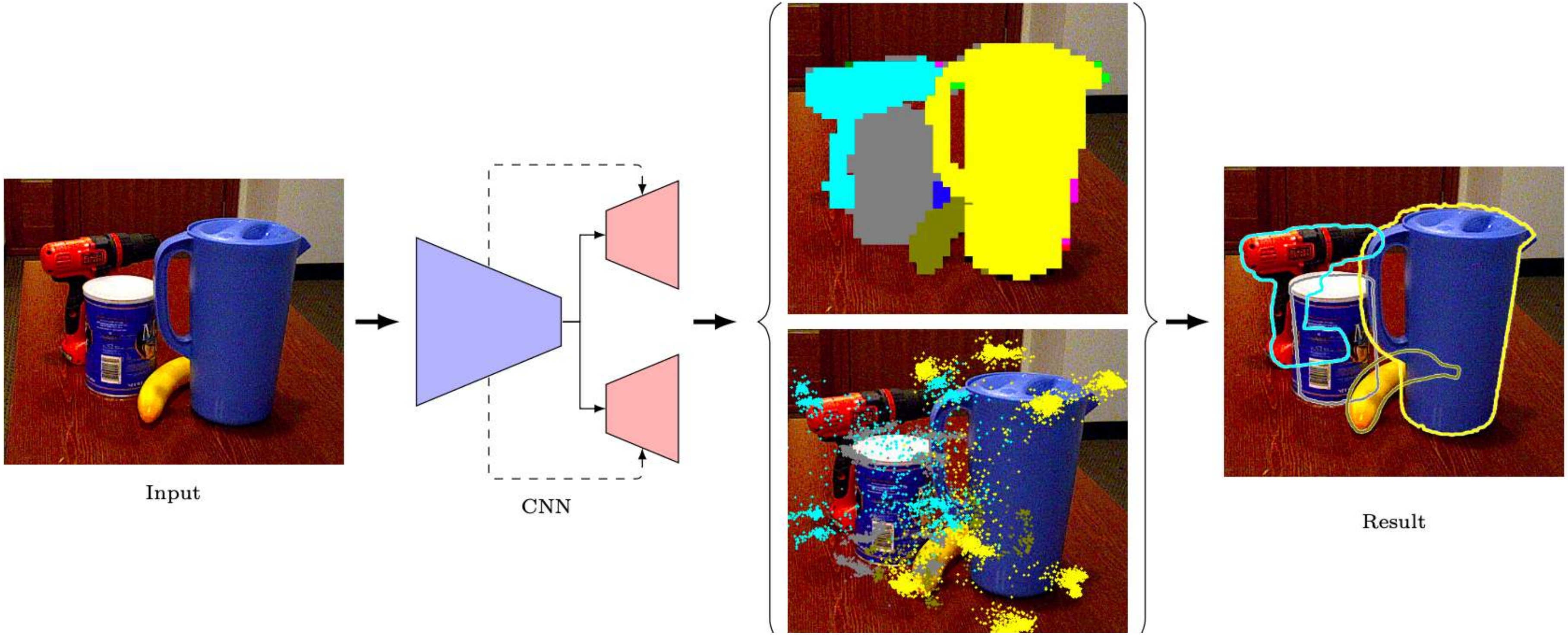
(d) Disentangled coord. z-axis translation

Dealing with Partial Occlusion

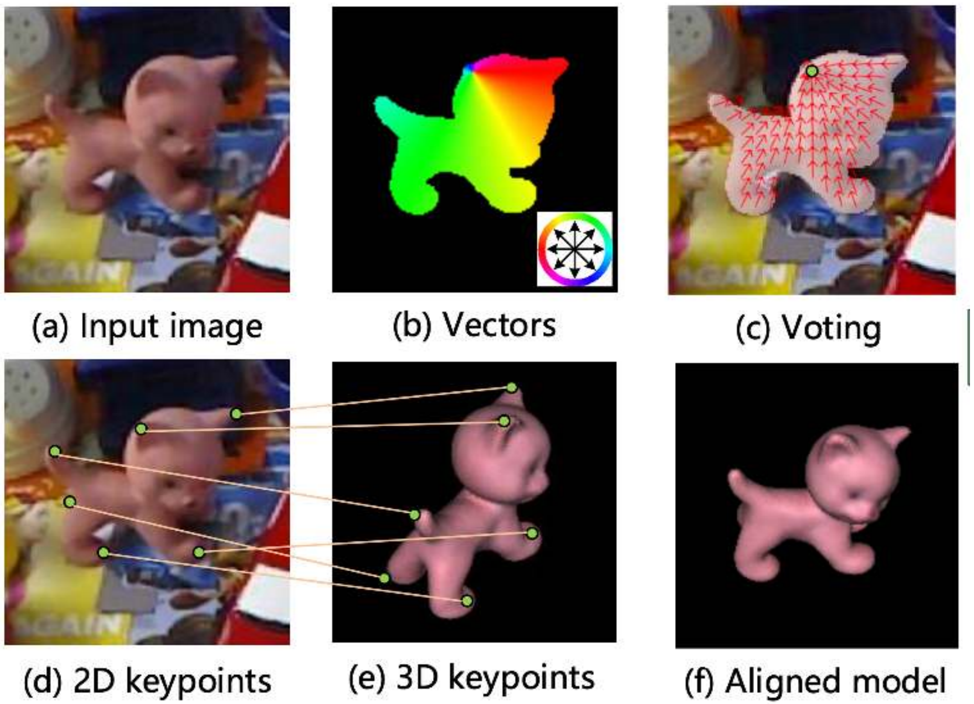
Avoid Occlusions in the Input

Voting for the corners



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

Voting for the corners



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

Conclusion

- Training set;
- Pose representation;
- Symmetrical objects;
- Partial occlusions.

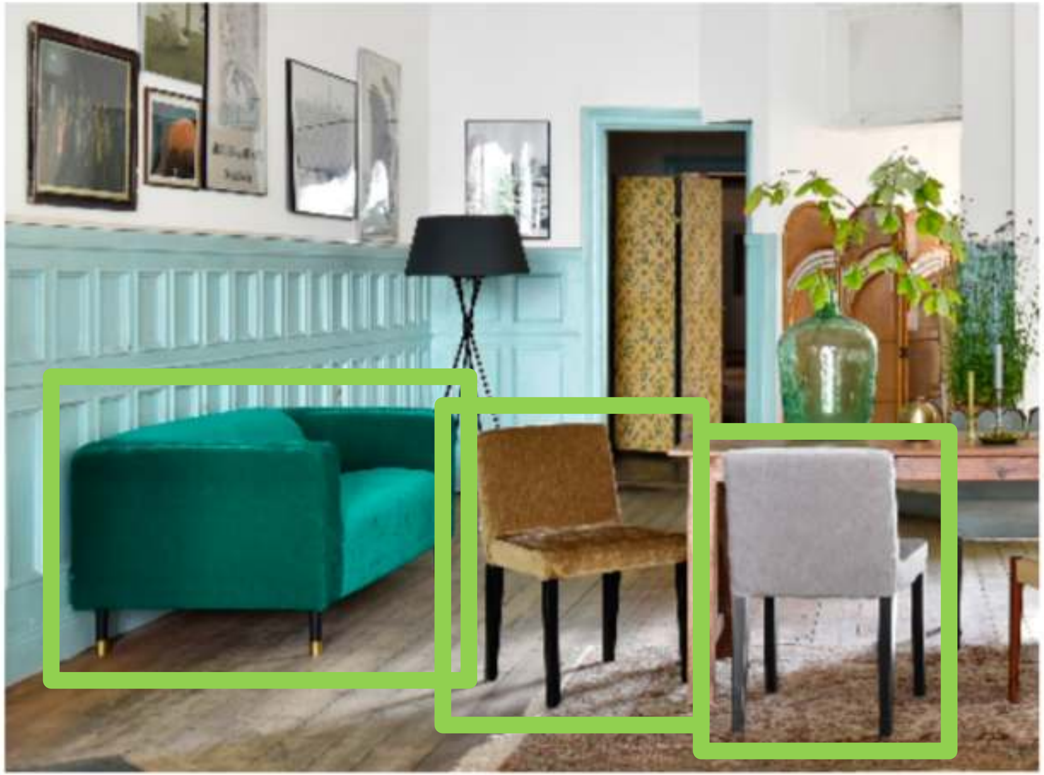
Object Categories

École des Ponts
ParisTech

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. 52 citations. Patented.

3D Pose Prediction for Object Categories



2D bounding boxes from Mask-RCNN

École des Ponts ParisTech

Qualitative Results

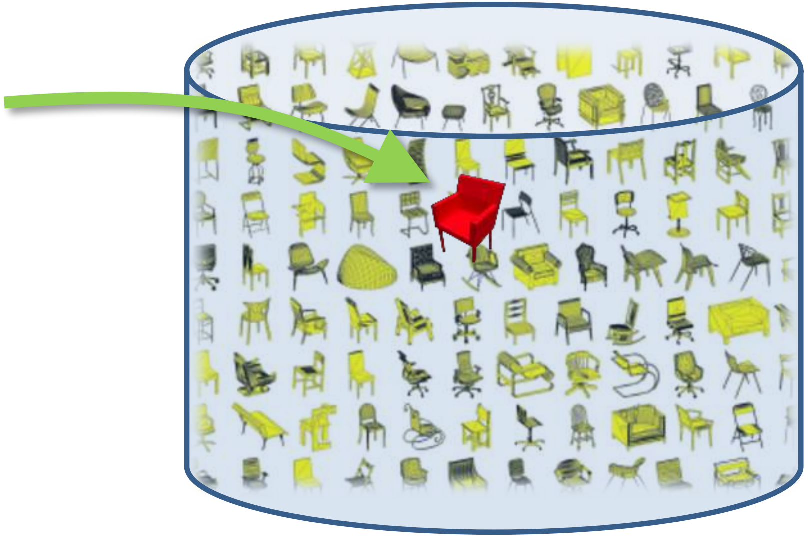
3D Geometry Retrieval for Object Categories

École des Ponts
ParisTech

3D Model Retrieval for Object Categories

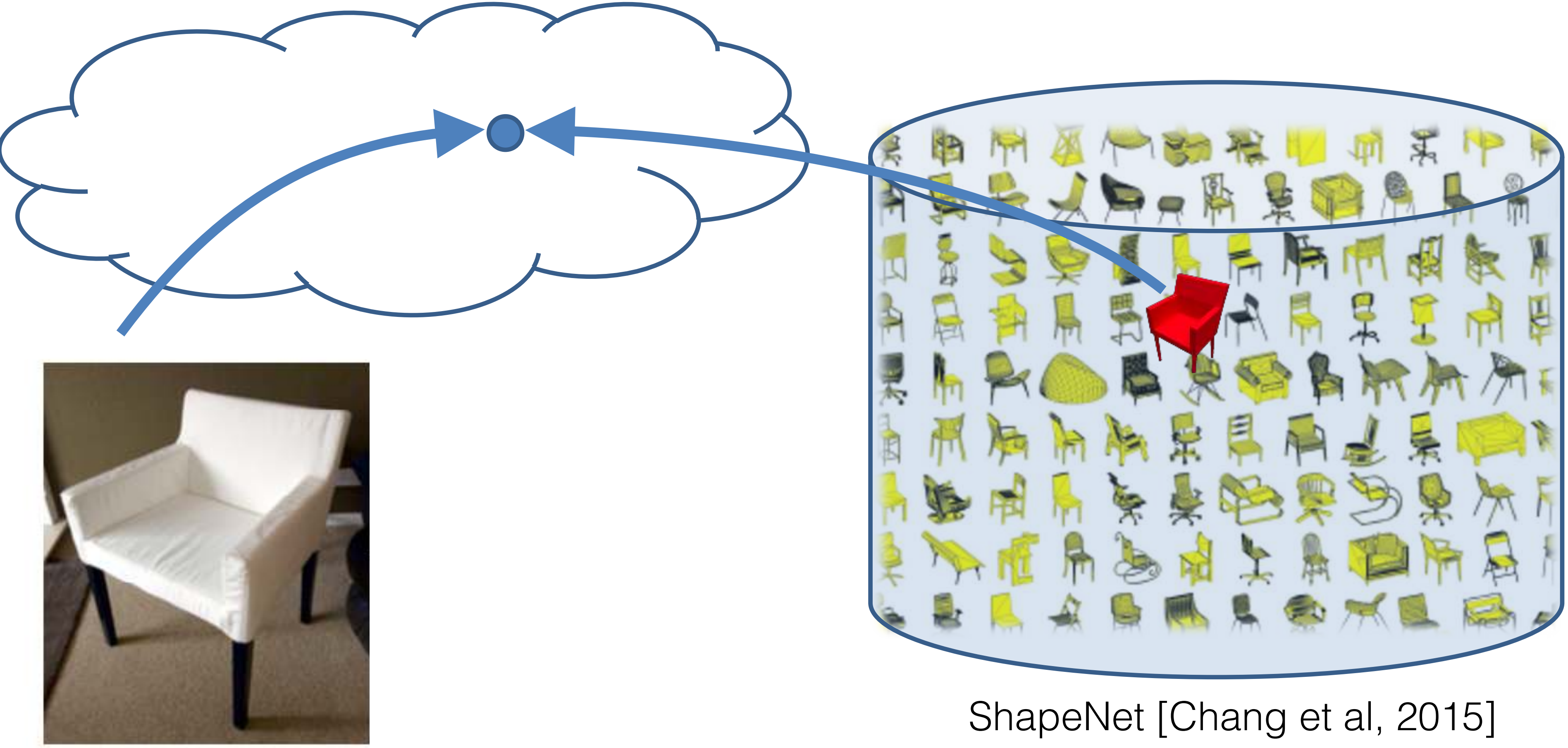
Possible options: Predicting a point cloud, voxels, 3D planes, ..

We look for a man-made 3D model similar to the object.

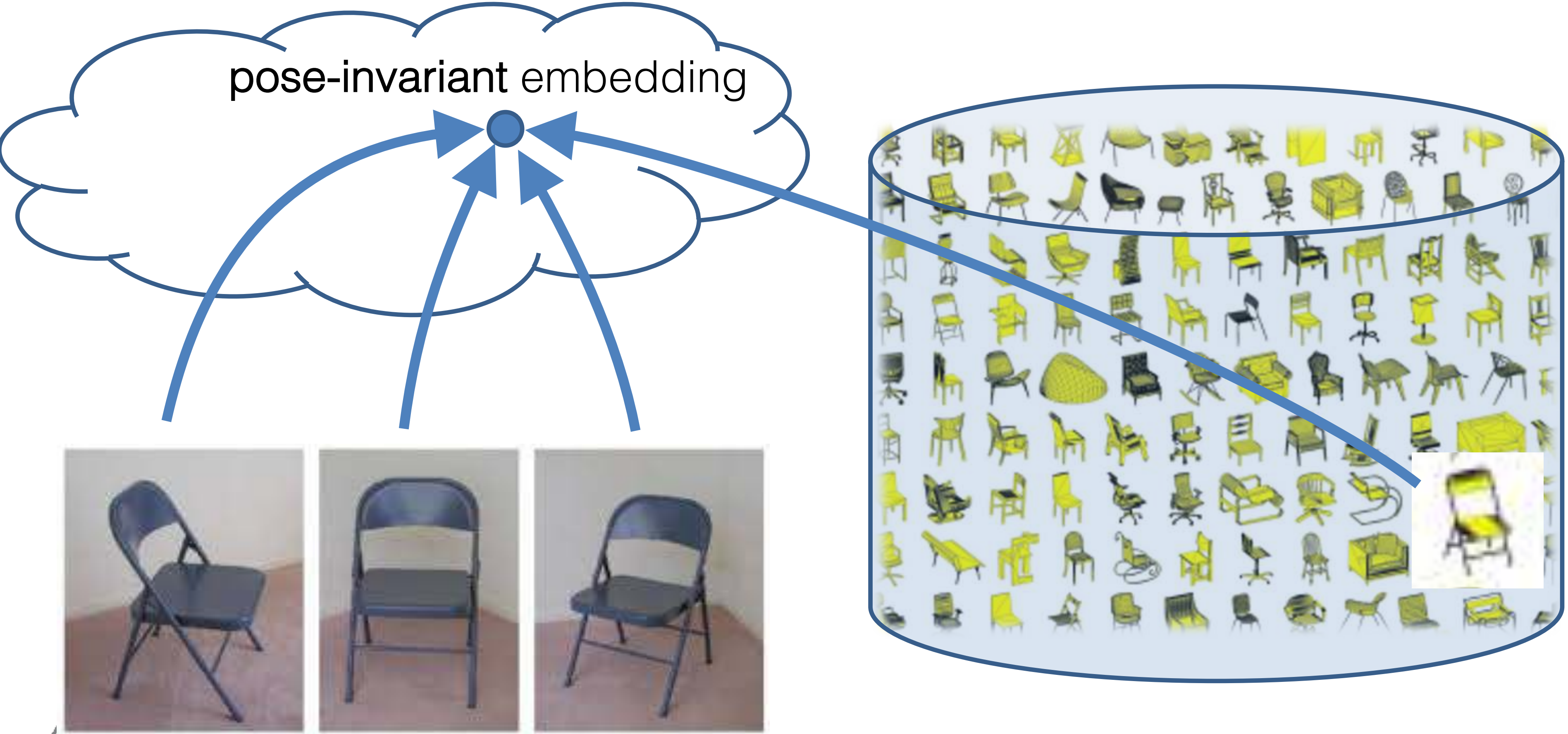


ShapeNet [Chang et al, 2015]

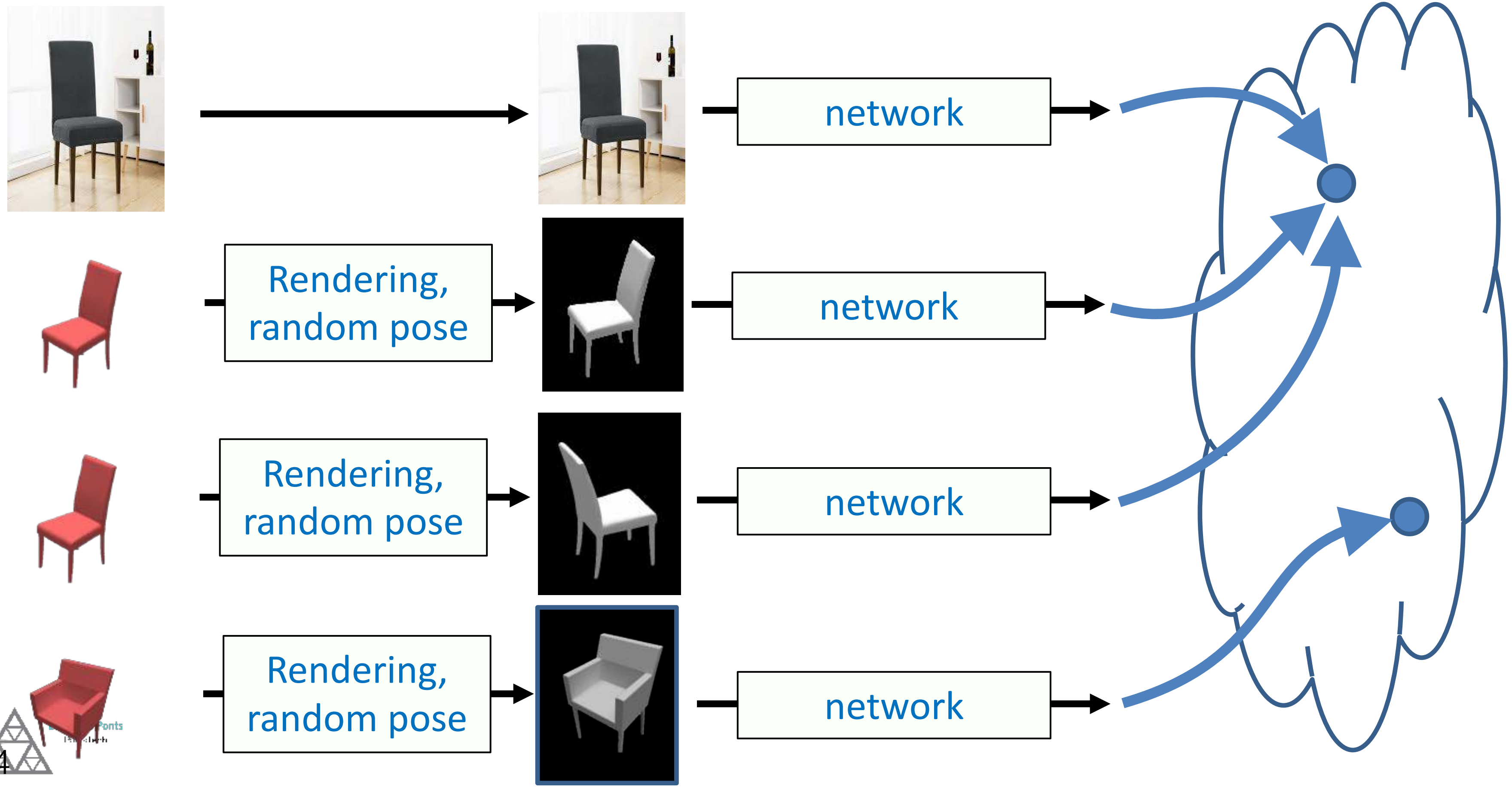
shared embedding space



ShapeNet [Chang et al, 2015]

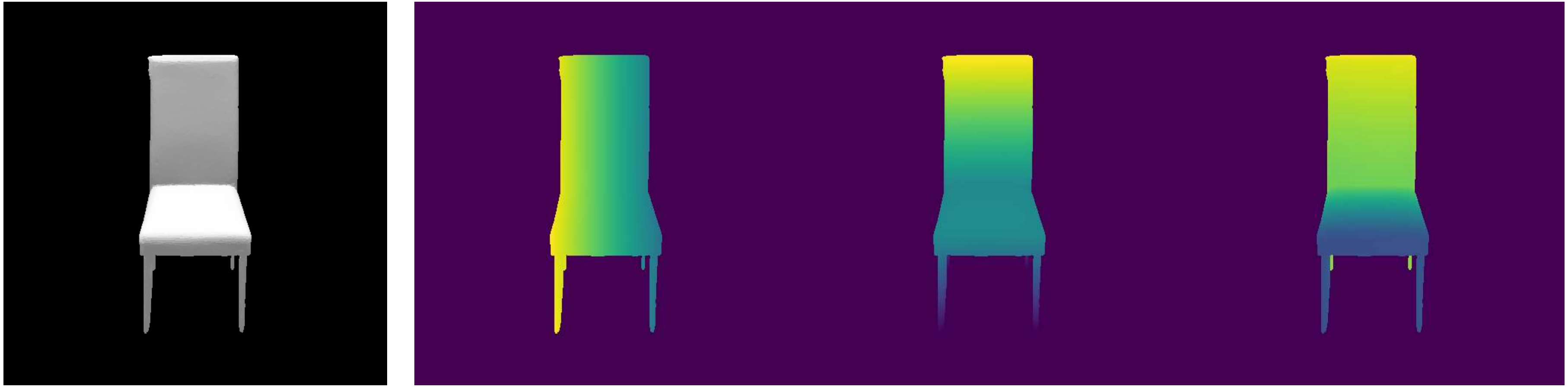

pose-invariant embedding

Pose Invariant Embeddings & Metric Learning



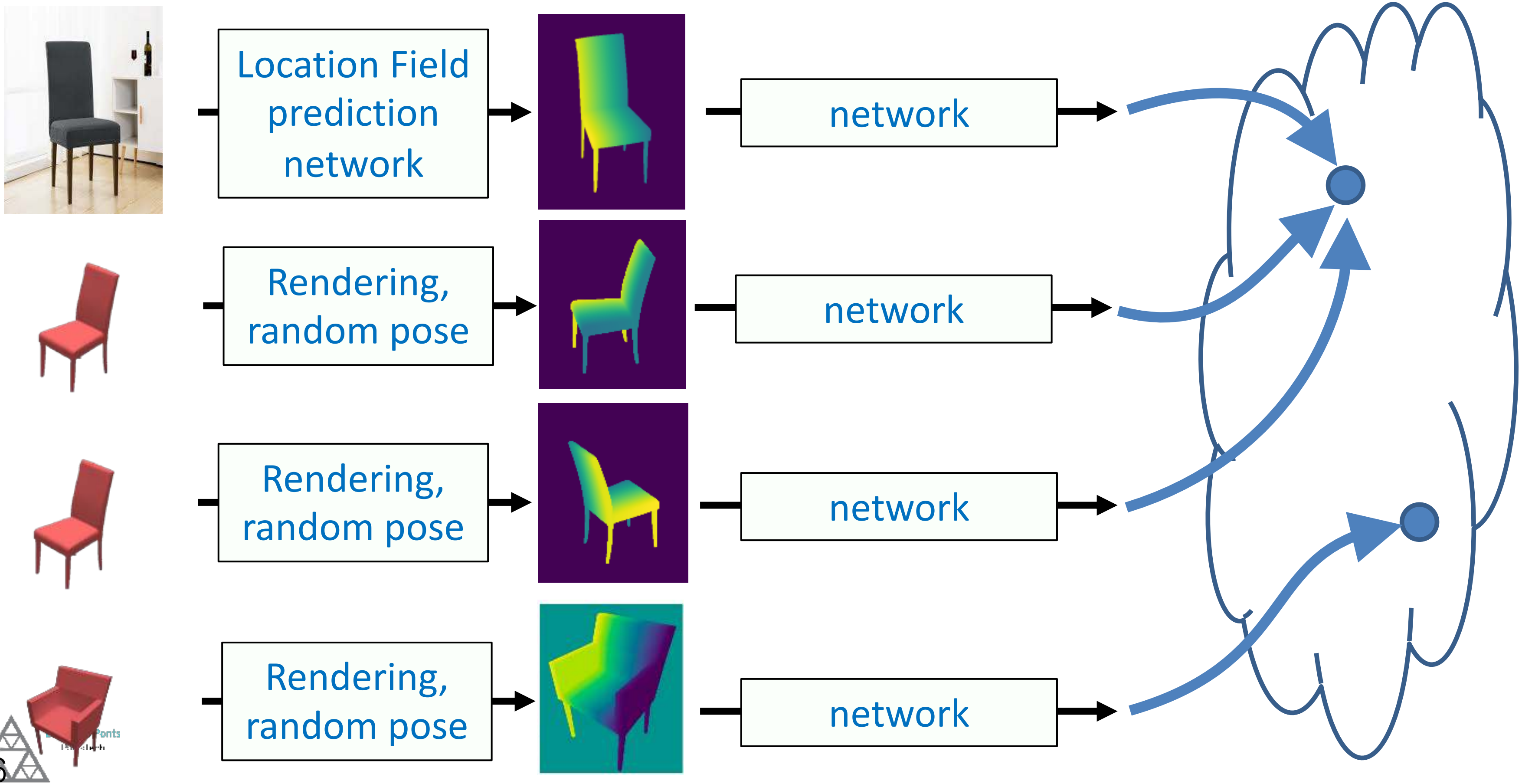
Location Fields/Object Coordinates/..

For each pixel: the 3D coordinates on the object's surface, in the object's coordinate system:

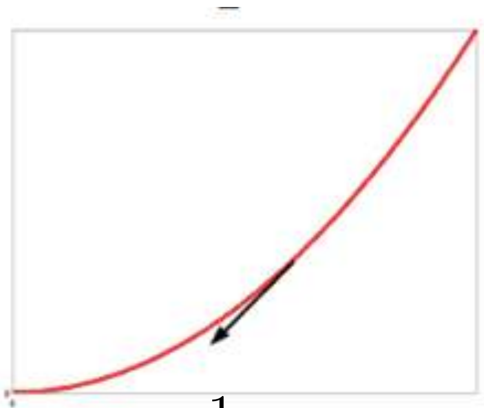


RGB X Y Z

Pose Invariant Embeddings & Metric Learning



Pose Invariant Embeddings & Metric Learning: Loss (called contrastive loss)



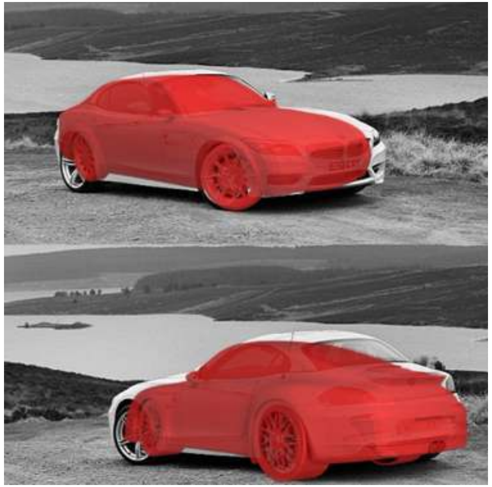
$$L_{\text{Similar}}(x_1, x_2) = \frac{1}{2} D_w^2(x_1, x_2)$$

$$L_{\text{Dissimilar}}(x_1, x_2) = \tfrac{1}{2}[\max(0, m - D_w(x_1, x_2))]^2$$

$$D_w(x_1, x_2) = \|G_w(x_1) - G_w(x_2)\|$$

Qualitative Results

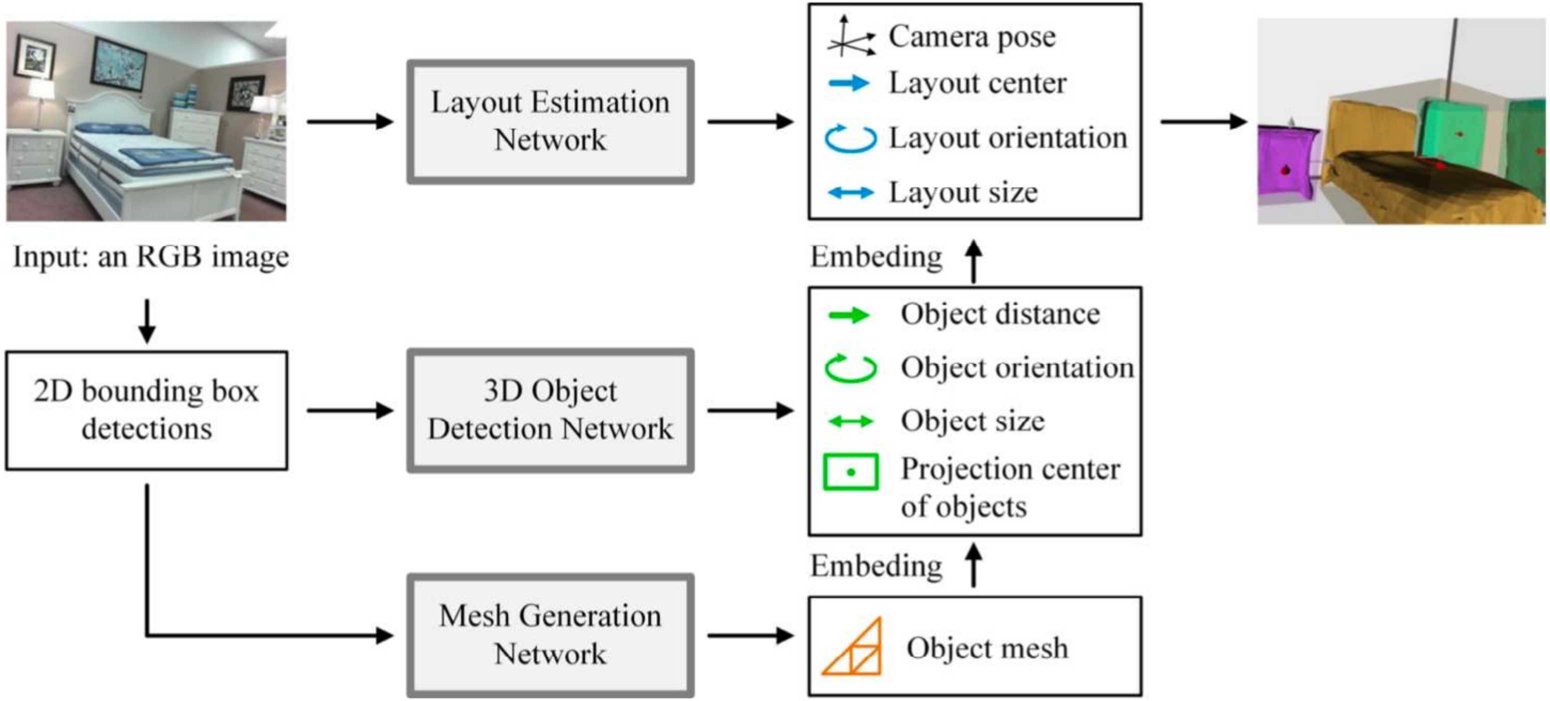




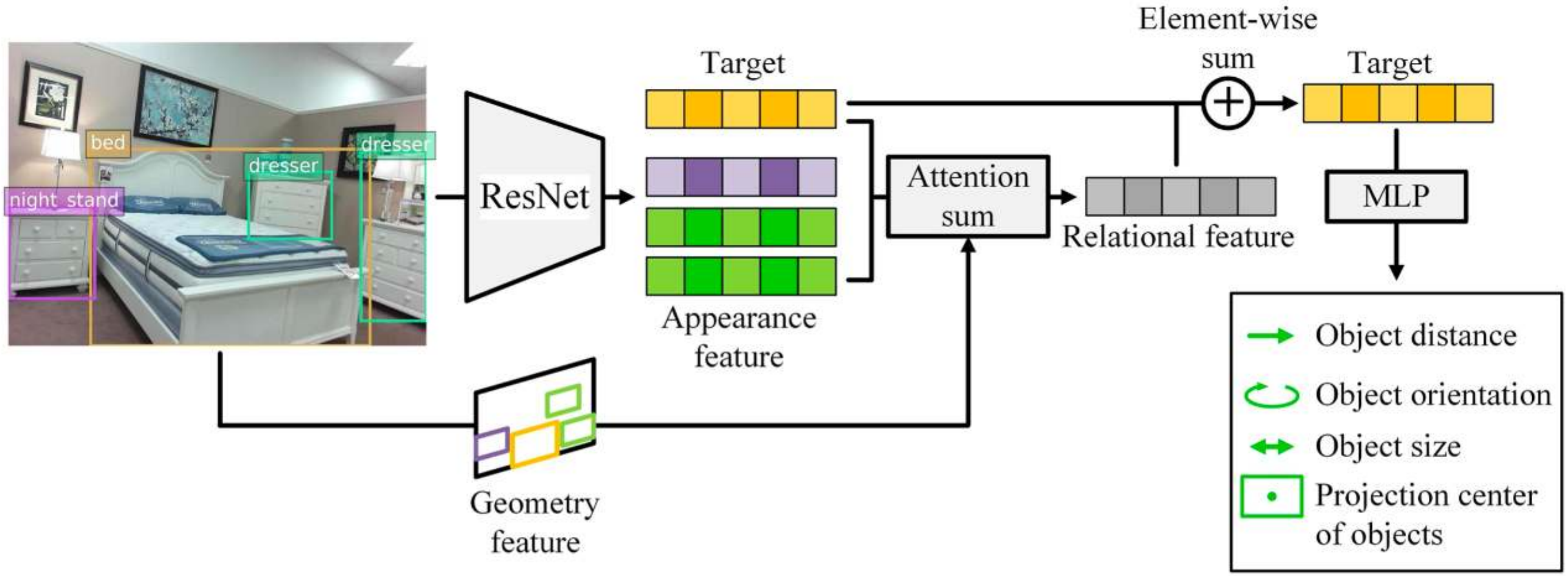

Total3DUnderstanding

[Total3DUnderstanding: Joint Layout, Object Pose and Mesh Reconstruction for Indoor Scenes from a Single Image. Yinyu Nie, Xiaoguang Han, Shihui Guo, Yujian Zheng, Jian Chang, Jian Jun Zhang. CVPR 2020]

Total3DUnderstanding



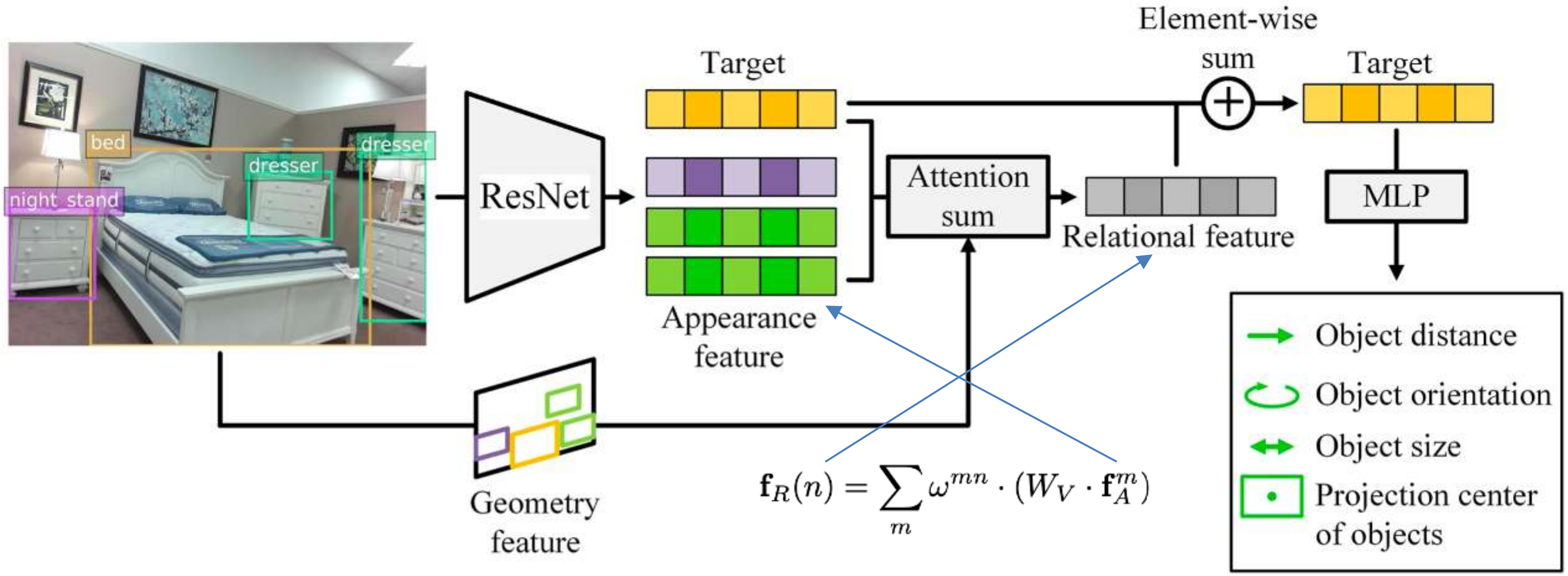
Total3DUnderstanding: 3D object pose prediction



Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation Networks for Object Detection. CVPR 2018.

Analogy with NLP: the appearance features correspond to the embeddings of the words of a sentence, the geometry features correspond to the positions of the words in the sentence. → Use attention!

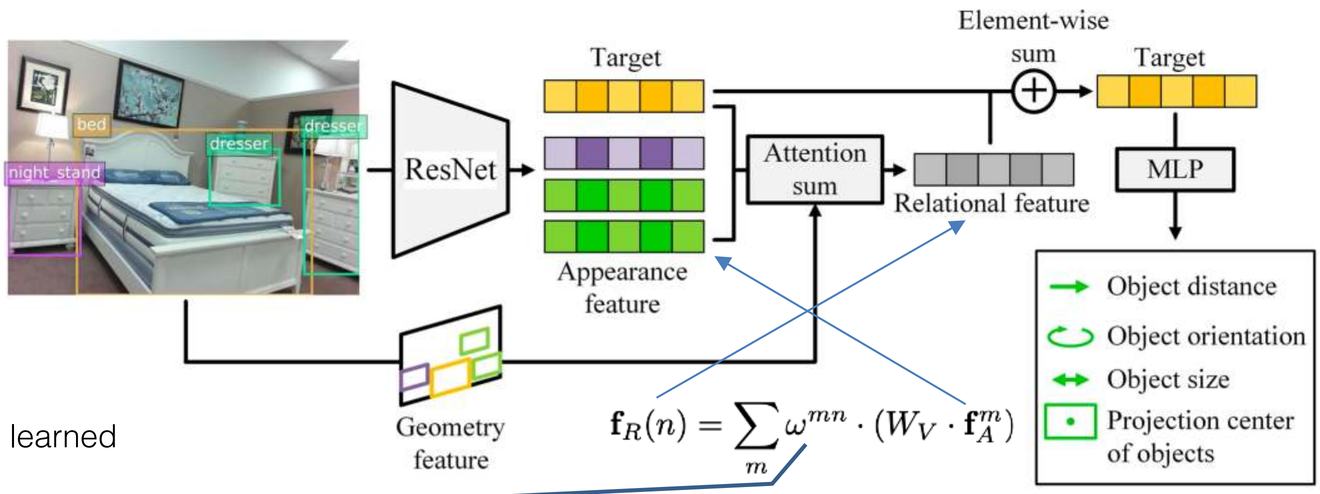
Total3DUnderstanding: 3D object pose prediction



$$\mathbf{f}_R(n) = \sum_m \omega^{mn} \cdot (W_V \cdot \mathbf{f}_A^m)$$

Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation Networks for Object Detection. CVPR 2018.

Total3DUnderstanding: 3D object pose prediction



$$\mathbf{f}_R(n) = \sum_m \omega^{mn} \cdot (W_V \cdot \mathbf{f}_A^m)$$

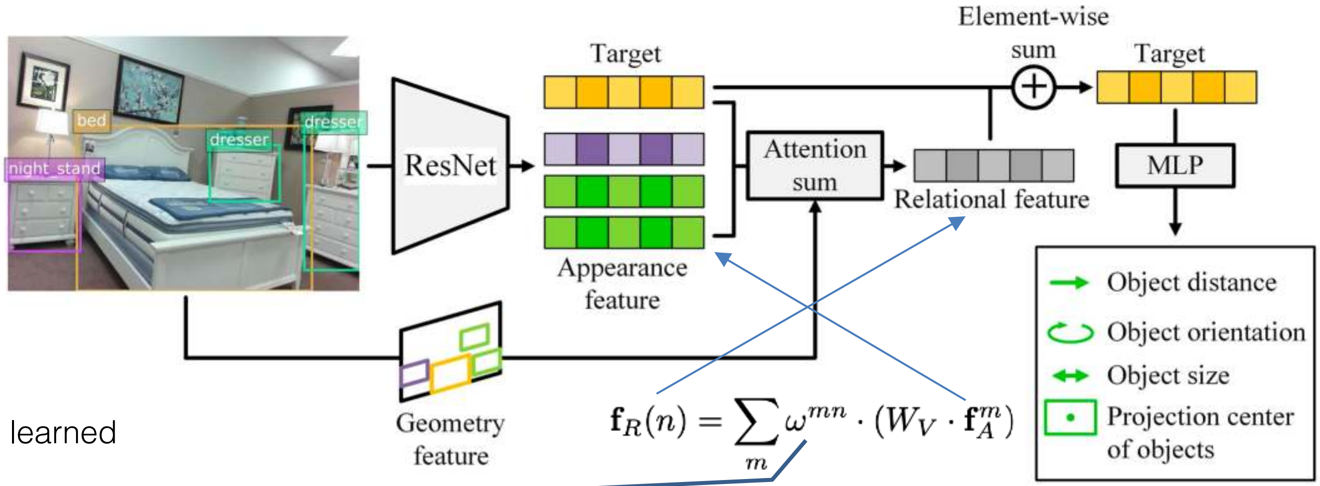
W_V, W_K, W_Q, W_G are learned

$$\omega^{mn} = \frac{\omega_G^{mn} \cdot \exp(\omega_A^{mn})}{\sum_k \omega_G^{kn} \cdot \exp(\omega_A^{kn})}$$

Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation Networks for Object Detection. CVPR 2018.

École des Ponts ParisTech

Total3DUnderstanding: 3D object pose prediction



$$\mathbf{f}_R(n) = \sum_m \omega^{mn} \cdot (W_V \cdot \mathbf{f}_A^m)$$

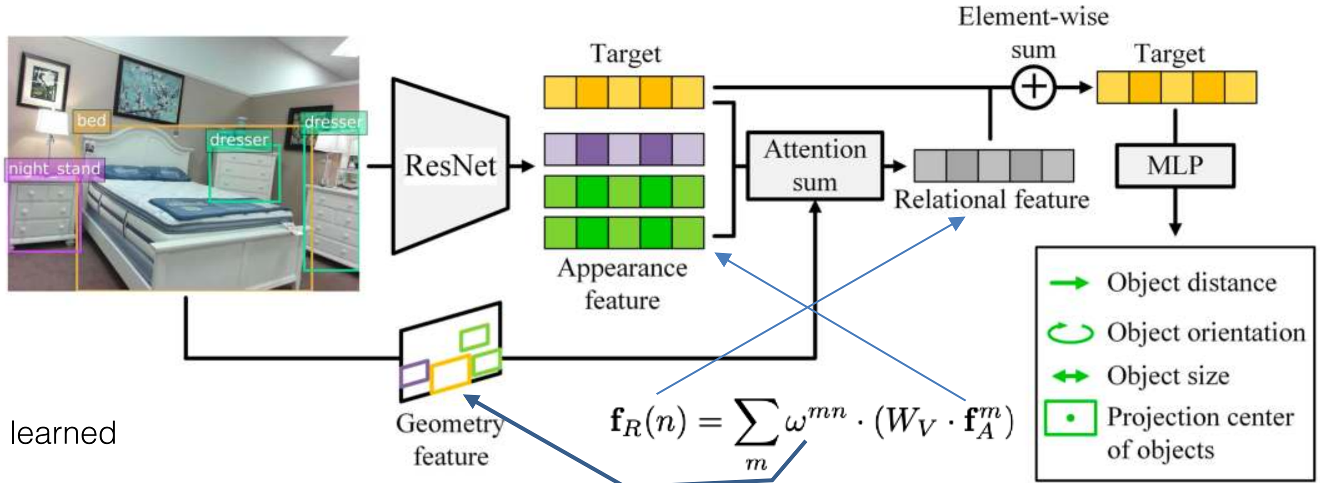
W_V, W_K, W_Q, W_G are learned

$$\omega^{mn} = \frac{\omega_G^{mn} \cdot \exp(\omega_A^{mn})}{\sum_k \omega_G^{kn} \cdot \exp(\omega_A^{kn})}, \qquad \omega_A^{mn} = \frac{dot(W_K \mathbf{f}_A^m, W_Q \mathbf{f}_A^n)}{\sqrt{d_k}}$$

École des Ponts ParisTech

Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation Networks for Object Detection. CVPR 2018.

Total3DUnderstanding: 3D object pose prediction



$$\mathbf{f}_R(n) = \sum_m \omega^{mn} \cdot (W_V \cdot \mathbf{f}_A^m)$$

W_V, W_K, W_Q, W_G are learned

$$\omega^{mn} = \frac{\omega_G^{mn} \cdot \exp(\omega_A^{mn})}{\sum_k \omega_G^{kn} \cdot \exp(\omega_A^{kn})}$$

$$\omega_A^{mn} = \frac{dot(W_K \mathbf{f}_A^m, W_Q \mathbf{f}_A^n)}{\sqrt{d_k}}$$

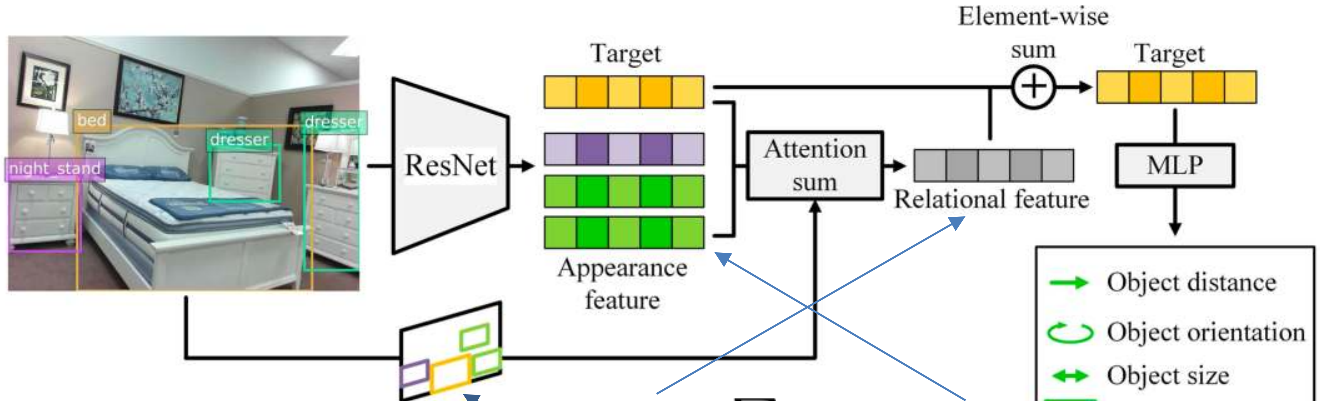
$$\omega_G^{mn} = \max\{0, W_G \cdot \mathcal{E}_G(\mathbf{f}_G^m, \mathbf{f}_G^n)\}$$

$$\left(\log\left(\frac{|x_m - x_n|}{w_m}\right), \log\left(\frac{|y_m - y_n|}{h_m}\right), \log\left(\frac{w_n}{w_m}\right), \log\left(\frac{h_n}{h_m}\right)\right)^T$$

$\mathcal{E}_G$: spatial encoding

Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation Networks for Object Detection. CVPR 2018.

Total3DUnderstanding: 3D object pose prediction



$$\mathbf{f}_R(n) = \sum_m \omega^{mn} \cdot (W_V \cdot \mathbf{f}_A^m)$$

W_V, W_K, W_Q, W_G are learned

$$\omega^{mn} = \frac{\omega_G^{mn} \cdot \exp(\omega_A^{mn})}{\sum_k \omega_G^{kn} \cdot \exp(\omega_A^{kn})}$$

$$\omega_A^{mn} = \frac{dot(W_K \mathbf{f}_A^m, W_Q \mathbf{f}_A^n)}{\sqrt{d_k}}$$

$$\omega_G^{mn} = \max\{0, W_G \cdot \mathcal{E}_G(\mathbf{f}_G^m, \mathbf{f}_G^n)\}$$

$$\left(\log\left(\frac{|x_m - x_n|}{w_m}\right), \log\left(\frac{|y_m - y_n|}{h_m}\right), \log\left(\frac{w_n}{w_m}\right), \log\left(\frac{h_n}{h_m}\right)\right)^T$$

$\mathcal{E}_G$: spatial encoding

École des Ponts ParisTech

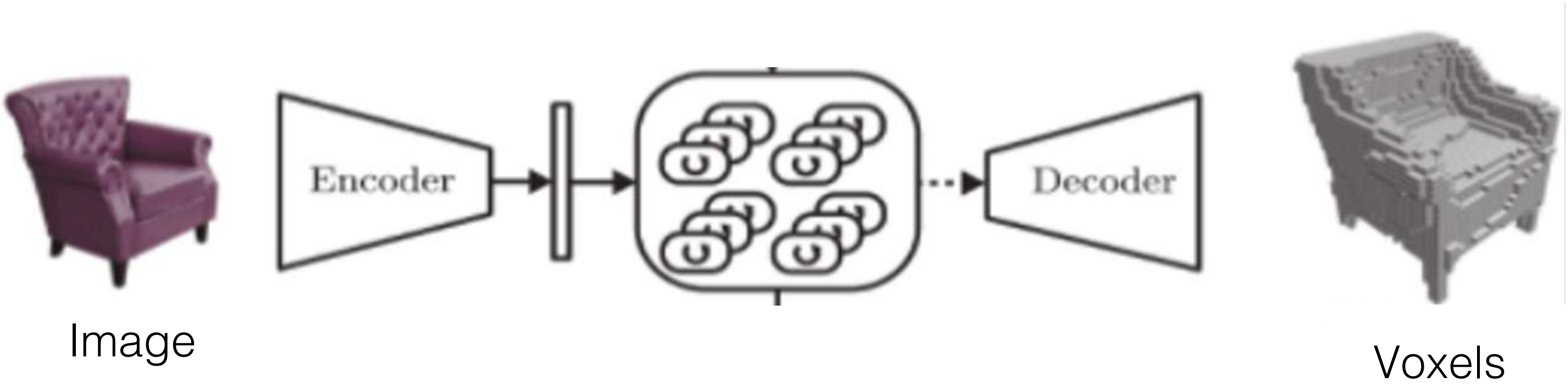
Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation Networks for Object Detection. CVPR 2018.

Training data

SUN-RGBD dataset, ~10'000 images annotated manually,
2000+ hours for initial annotations + time for error correction

3D model prediction from an image

Voxels



Choy, C. B., Xu, D., Gwak, J., Chen, K., & Savarese, S. 3D-R2N2: A unified approach for single and multi-view 3D object reconstruction, ECCV 2016

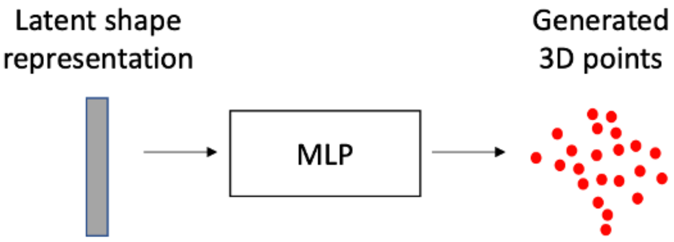
Unstructured Points

Image

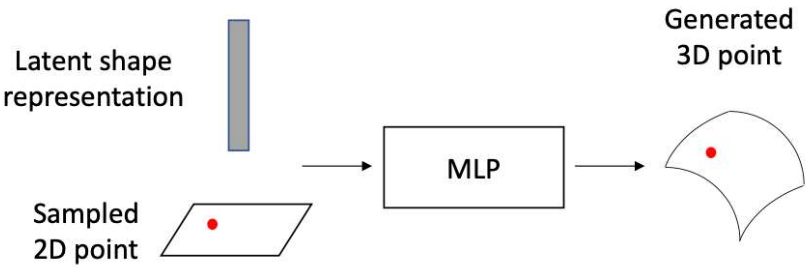
Points

Choy, C. B., Xu, D., Gwak, J., Chen, K., & Savarese, S. 3D-R2N2: A unified approach for single and multi-view 3D object reconstruction, ECCV 2016

AtlasNet

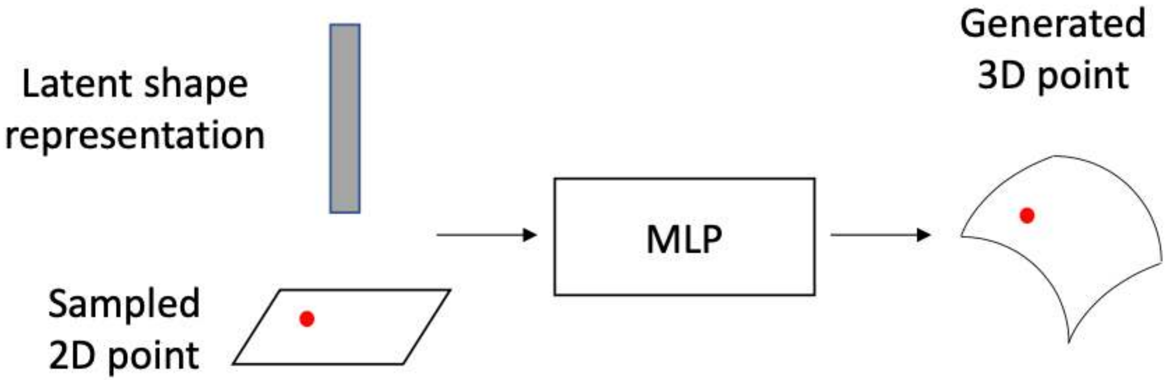


(a) Points baseline.



AtlasNet

AtlasNet



Latent shape representation
Sampled 2D point
MLP
Generated 3D point

AtlasNet: Qualitative comparisons

(a) Input (b) 3D-R2N2 (c) HSP (d) PSG (e) Ours

École des Ponts
ParisTech