3D Scene Understanding from an Image

Vincent Lepetit
ENPC ParisTech

École des Ponts ParisTech

TU Graz

Qualcomm

Christian Doppler Forschungsgesellschaft

3D Scene Understanding from an Image

Applications

- Augmented/Virtual Reality
- Robotics
- …

Challenges

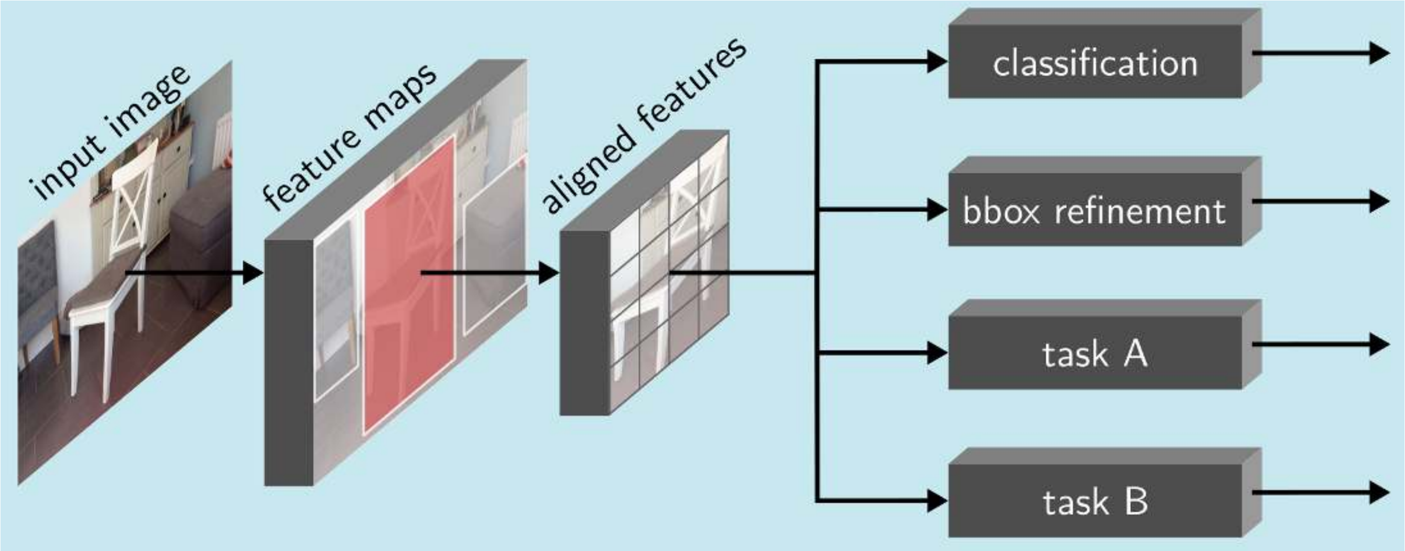
- 2D-3D information loss
- Appearance variation
- Pose ambiguities & symmetries
- Scarce training data
- Uncontrolled (“in the wild”) scenarios
- …

3D Pose Prediction for Object Categories

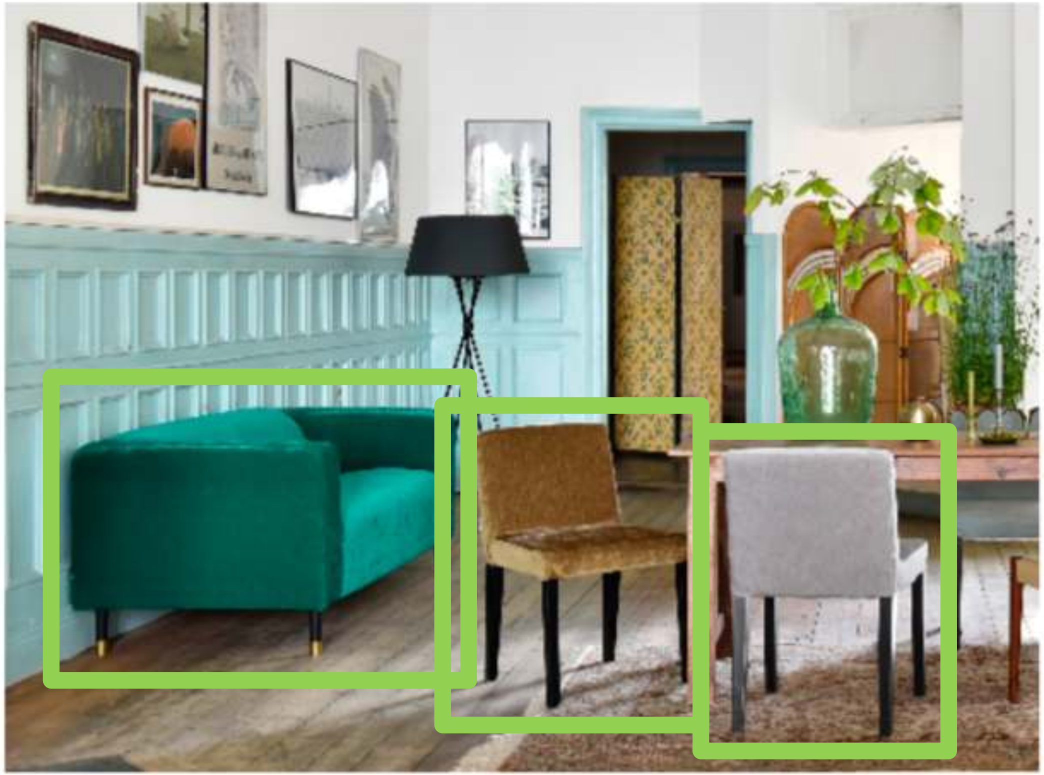
3D Pose Estimation and 3D Model Retrieval for Objects in the Wild.
Alexander Grabner, Peter M. Roth, and Vincent Lepetit. CVPR 2018.

Object Detection

Integrate the pose estimation into Mask R-CNN



3D Pose Prediction for Object Categories



2D bounding boxes from Mask-RCNN

3D Pose Prediction for Object Categories

3D Pose Prediction for Object Categories

3D pose of the object's bounding box

3D pose

3D Pose Prediction for Object Categories

Mask-RCNN head

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.

2D reprojections of the corners of the 3D bounding box

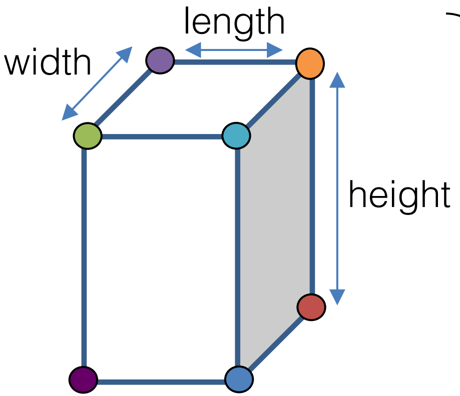
3D pose of the object's bounding box

3D pose

3D Pose Prediction for Object Categories

Mask-RCNN head

(length, width, height) of object's 3D bounding box

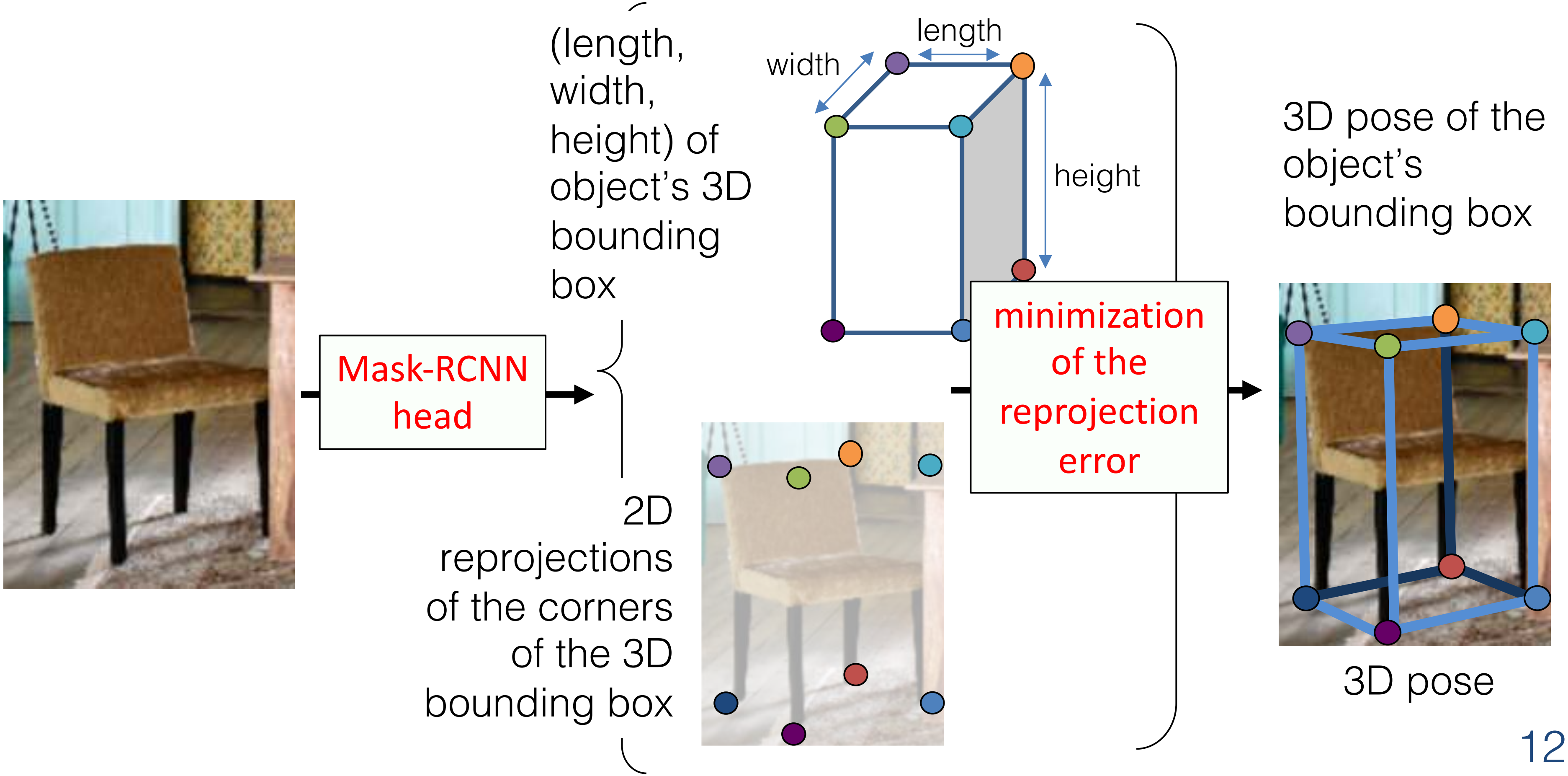


2D reprojections of the corners of the 3D bounding box

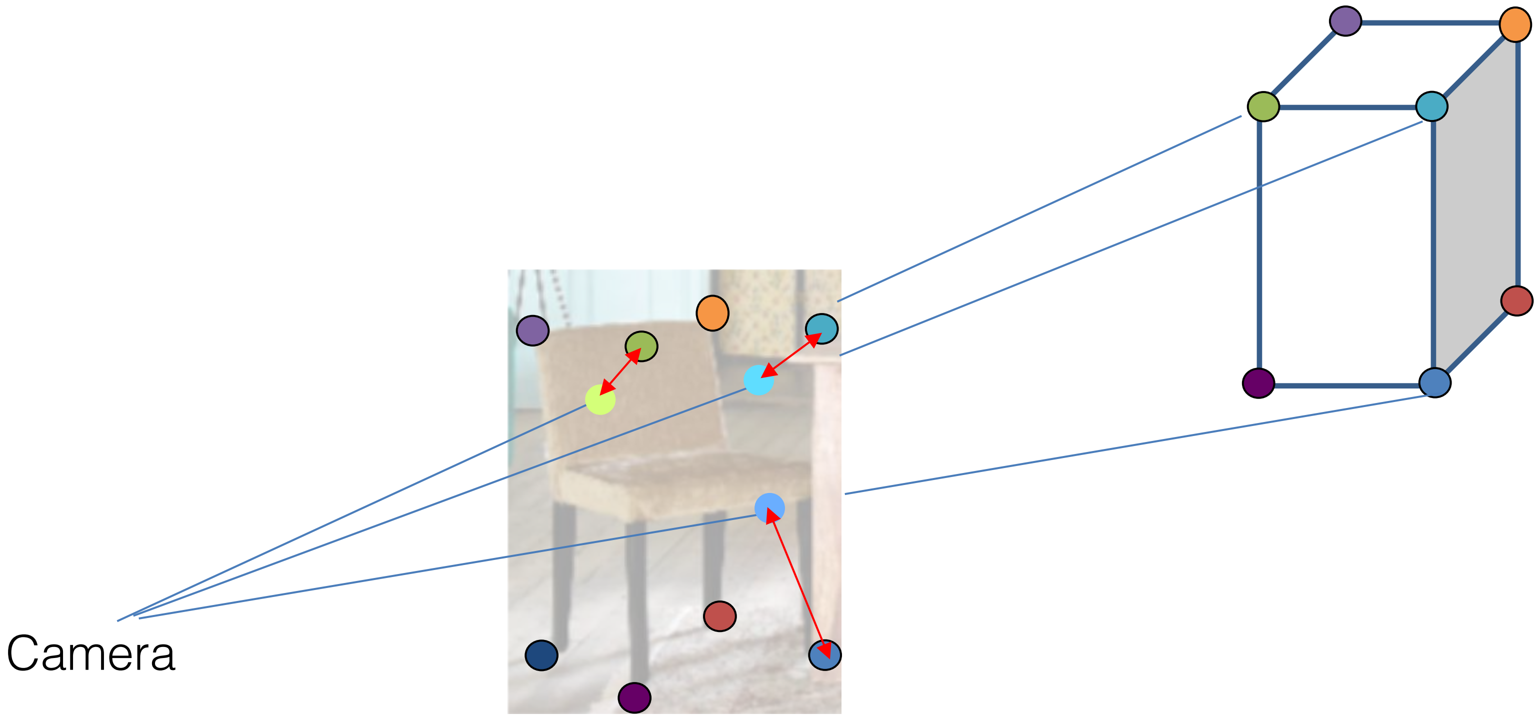
3D pose of the object's bounding box

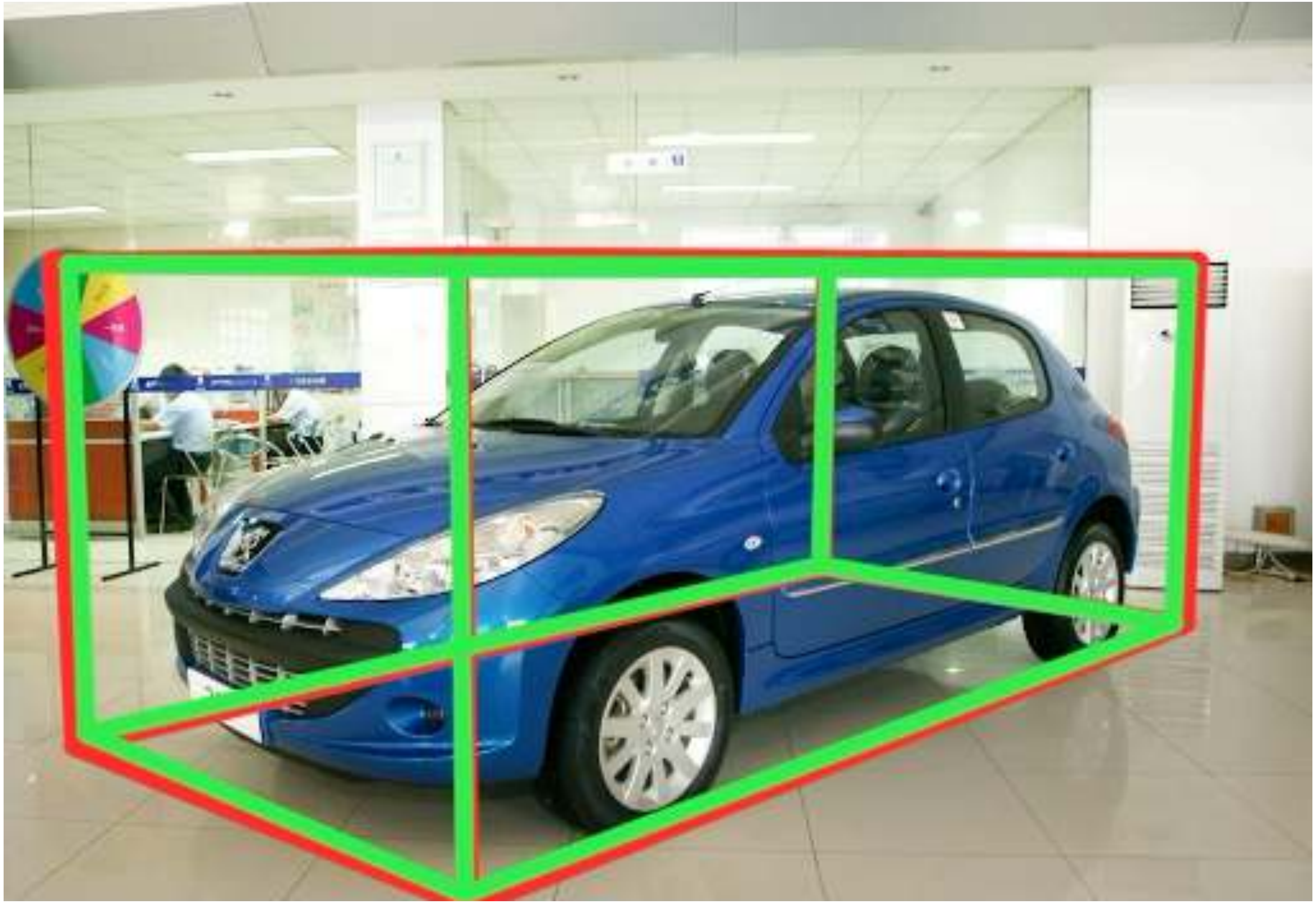
3D pose

3D Pose Prediction for Object Categories

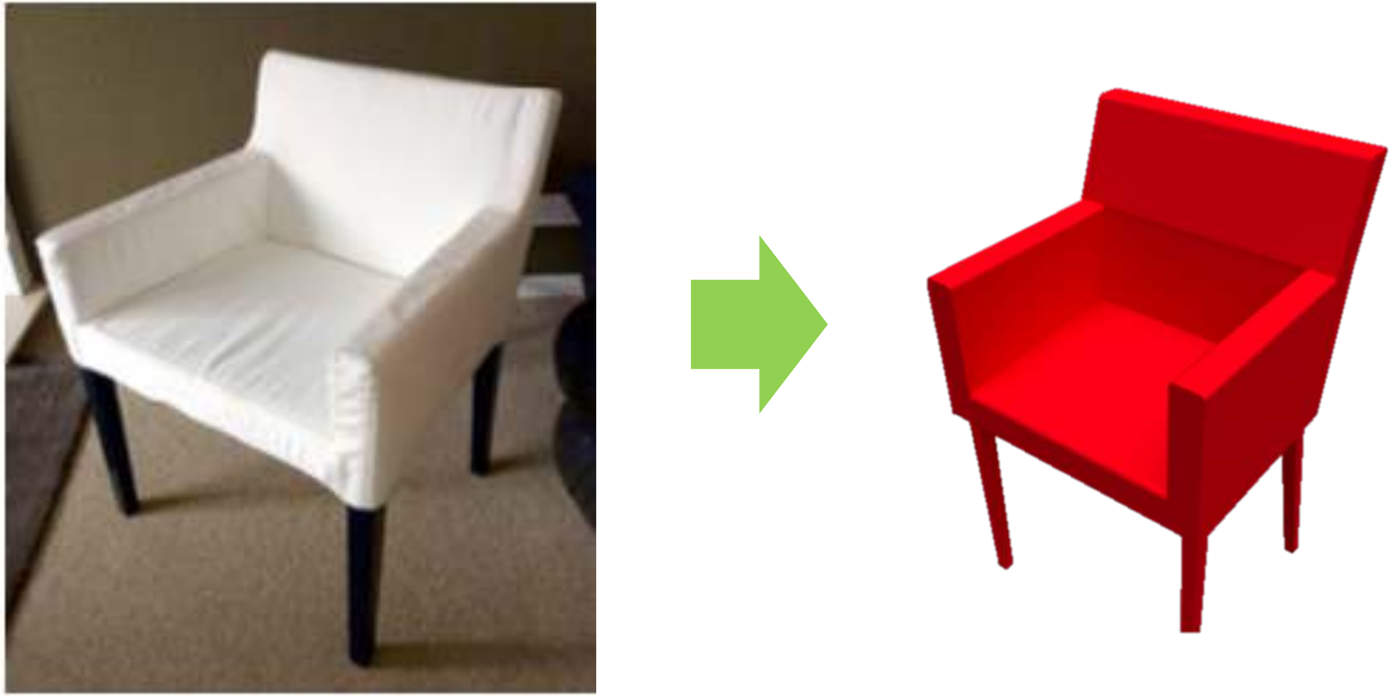


Minimization of the Reprojection Error





3D Geometry Retrieval for Object Categories

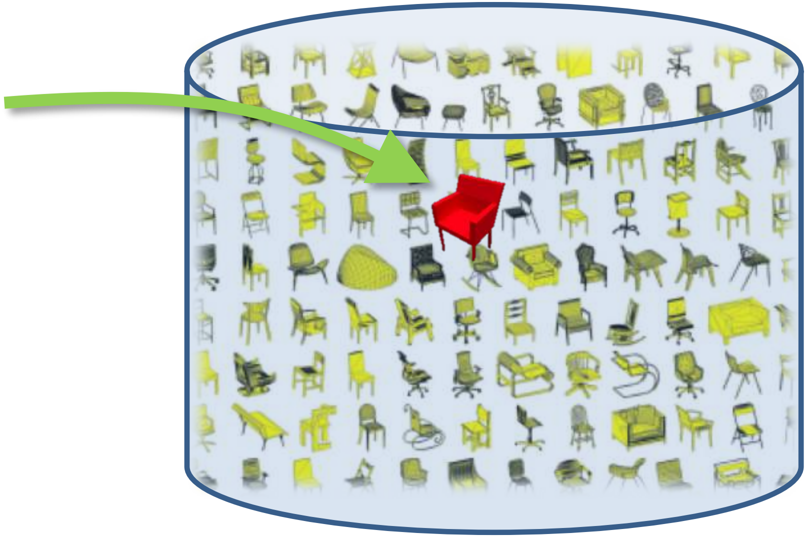


Location Field Descriptors: Single Image 3D Model Retrieval in the Wild. Alexander Grabner, Peter M. Roth, and Vincent Lepetit. 3DV 2019.

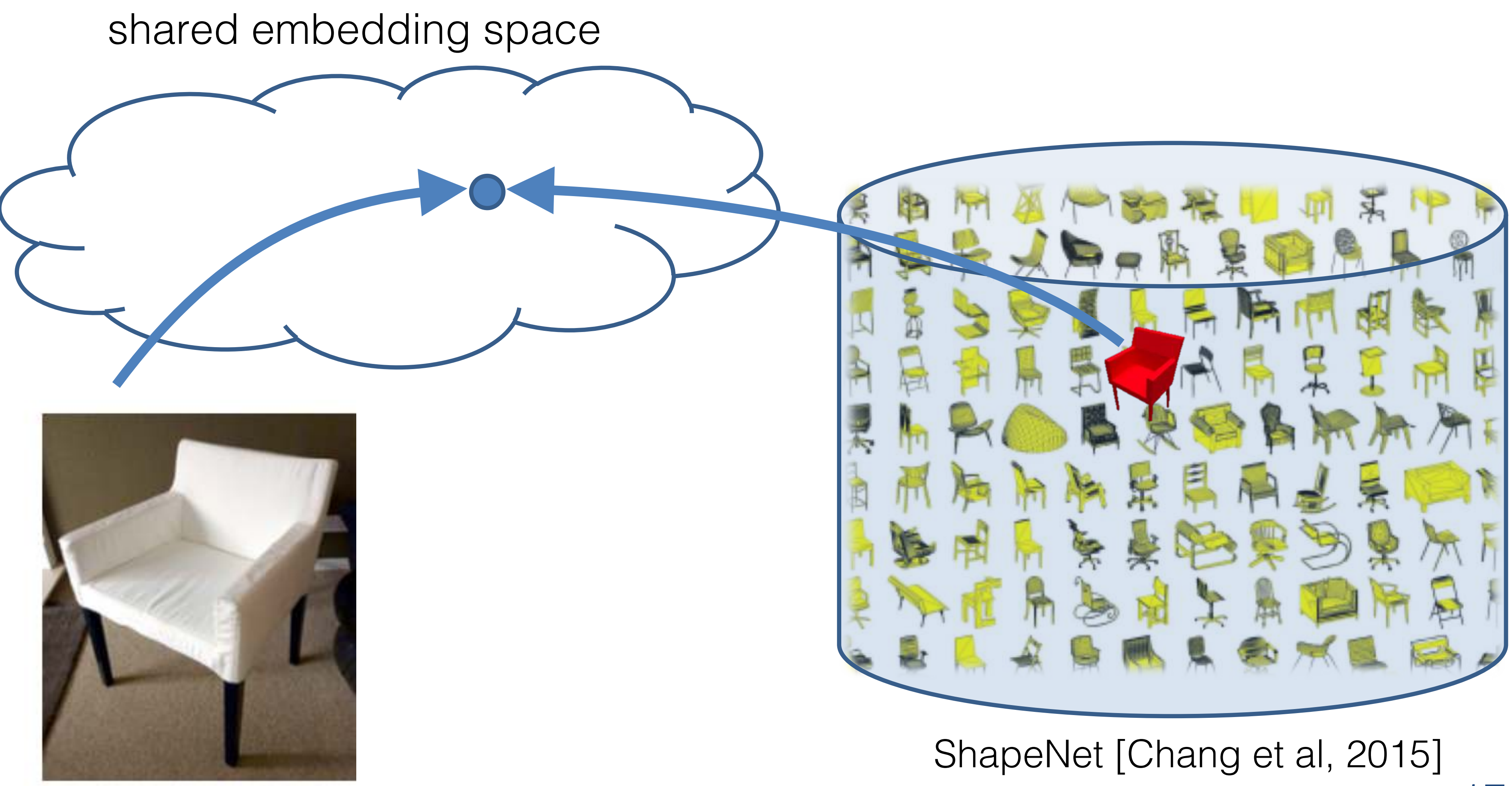
3D Model Retrieval for Object Categories

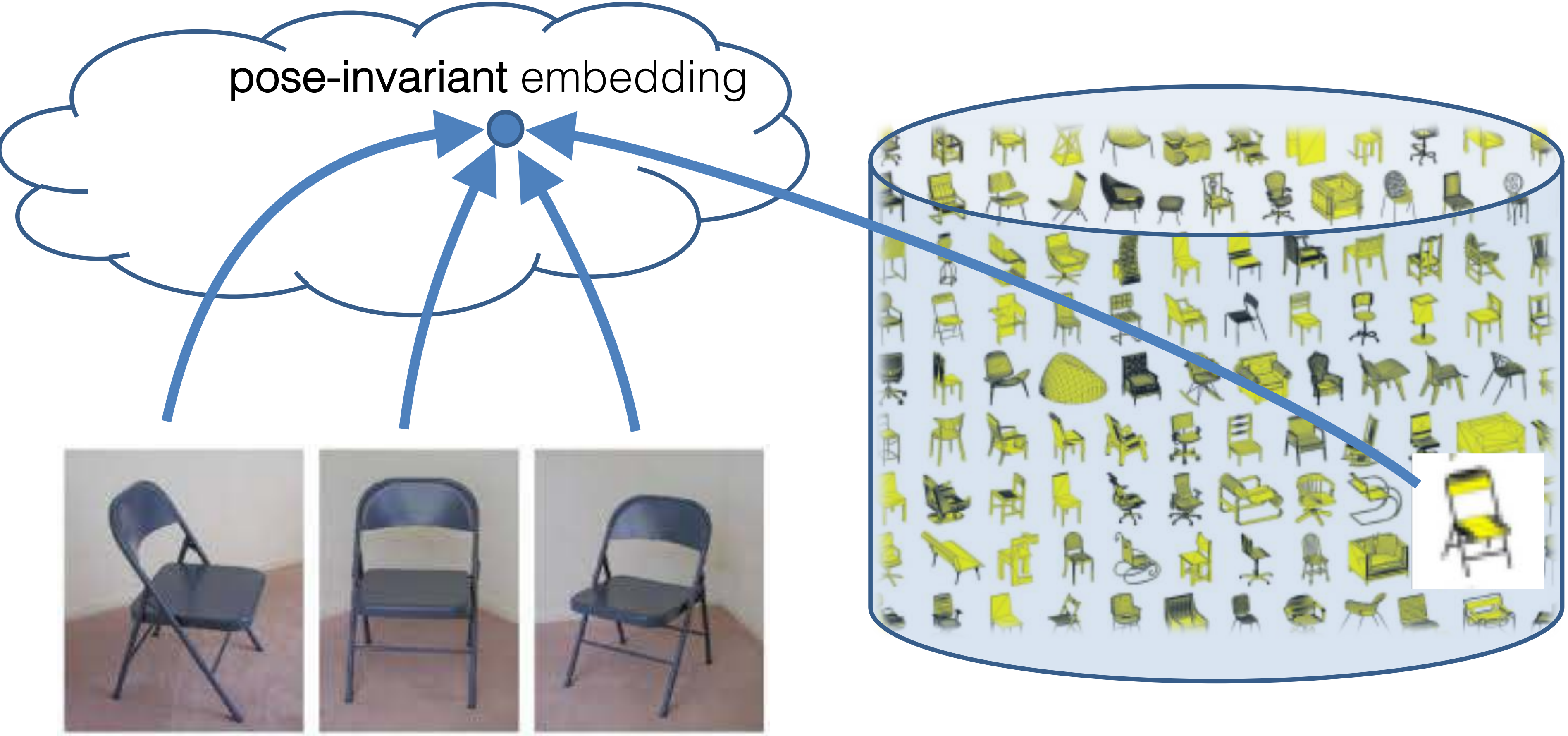
Possible options: Predicting a point cloud, voxels, 3D planes, ..

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

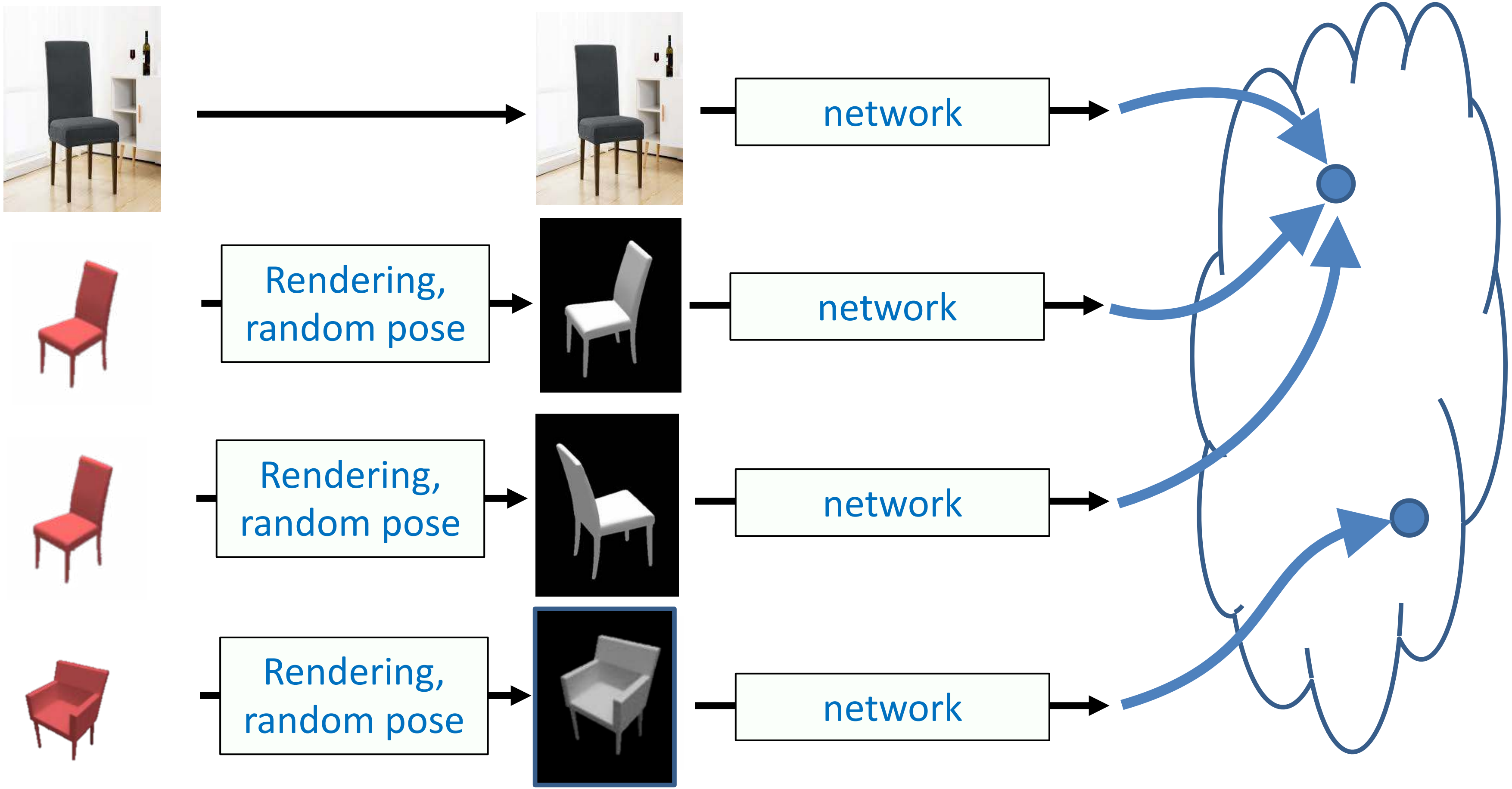


ShapeNet [Chang et al, 2015]


shared embedding space
ShapeNet [Chang et al, 2015]

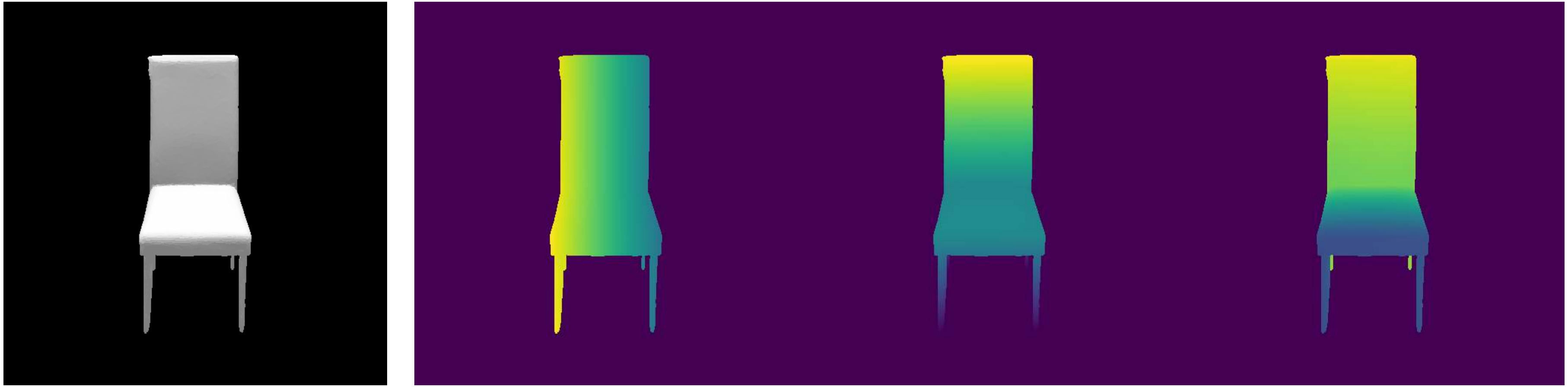

pose-invariant embedding

Pose Invariant Embeddings & Metric Learning



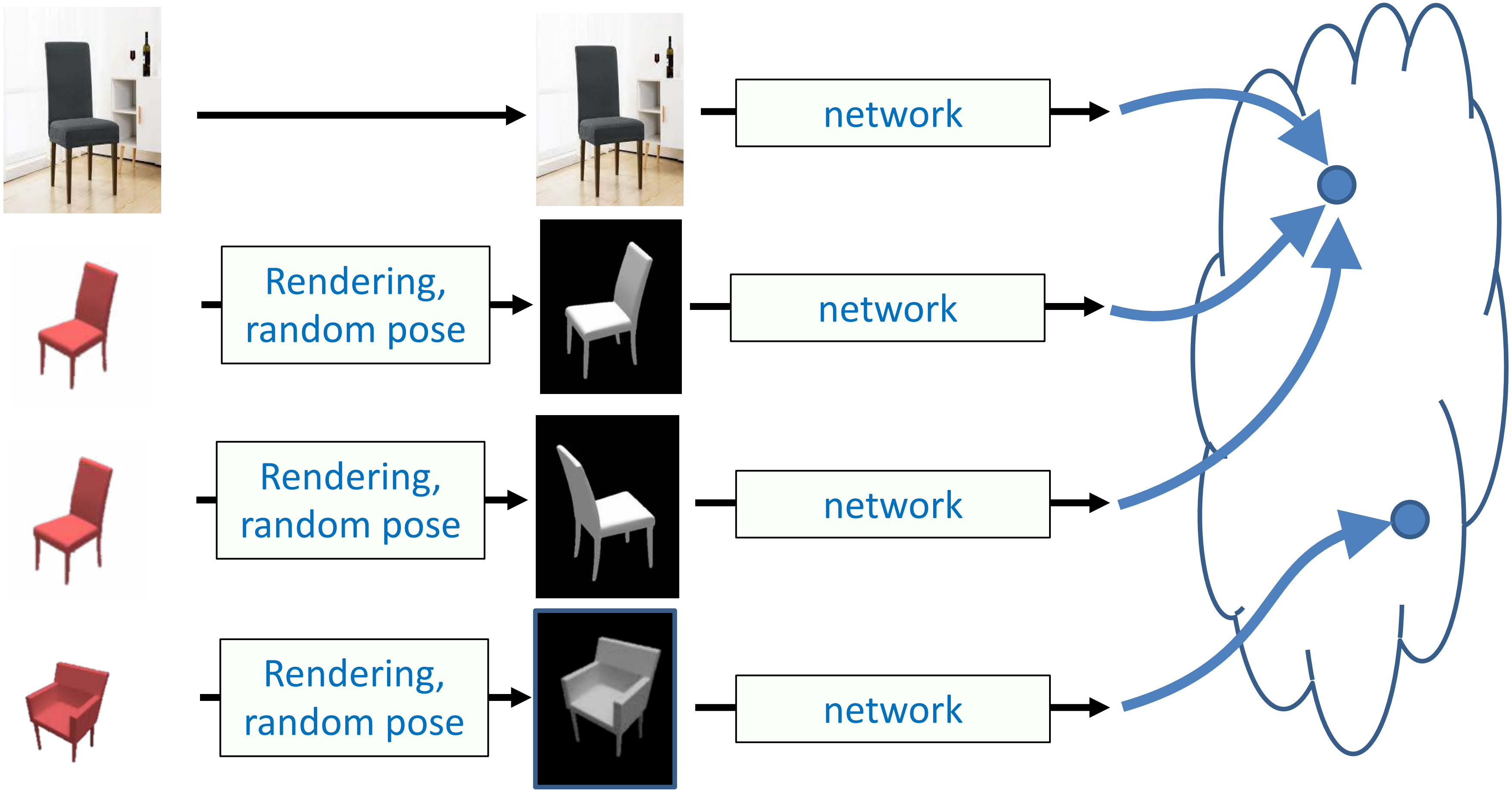
Location Fields/Object Coordinates/..

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

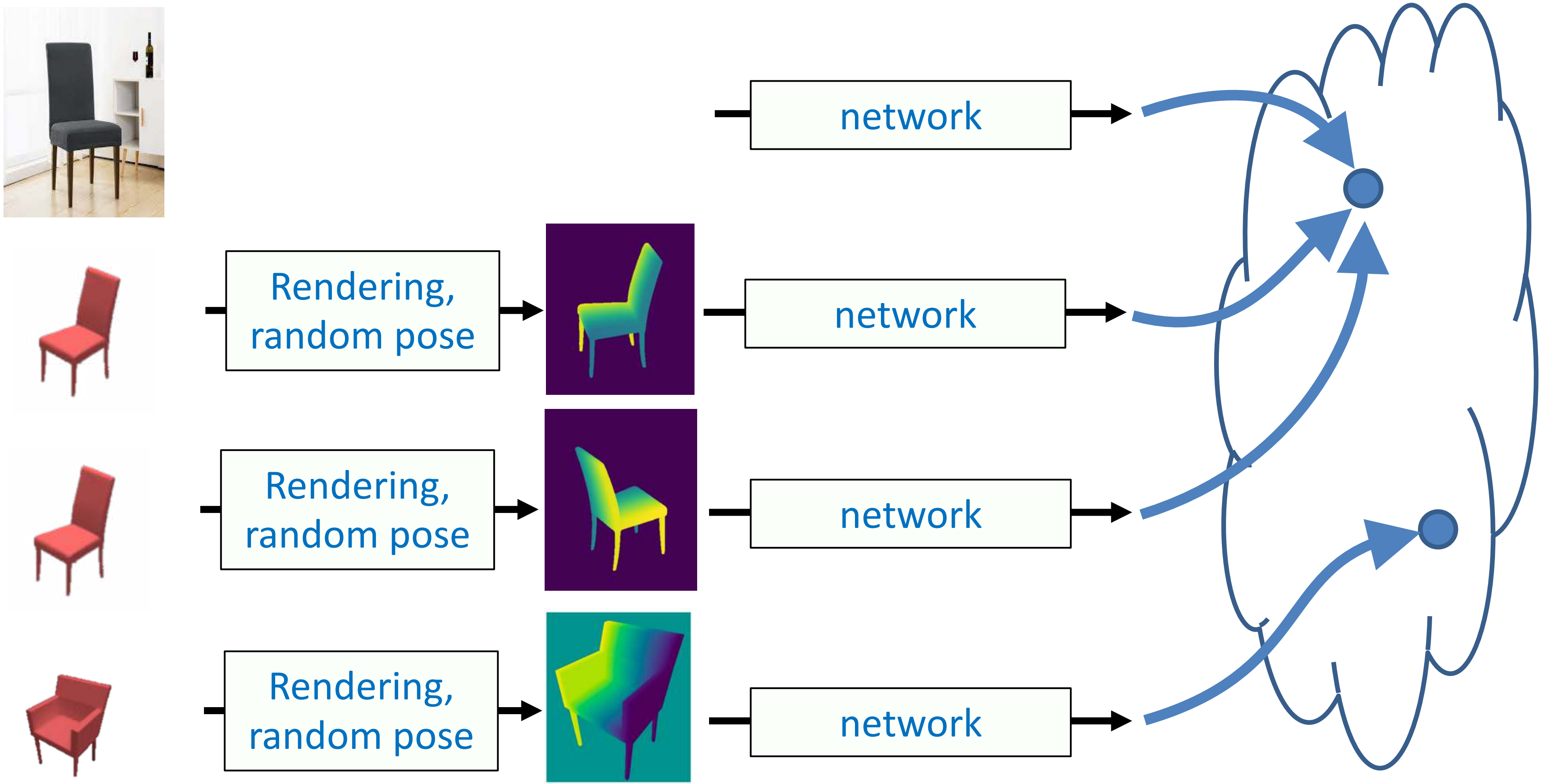


RGB X Y Z

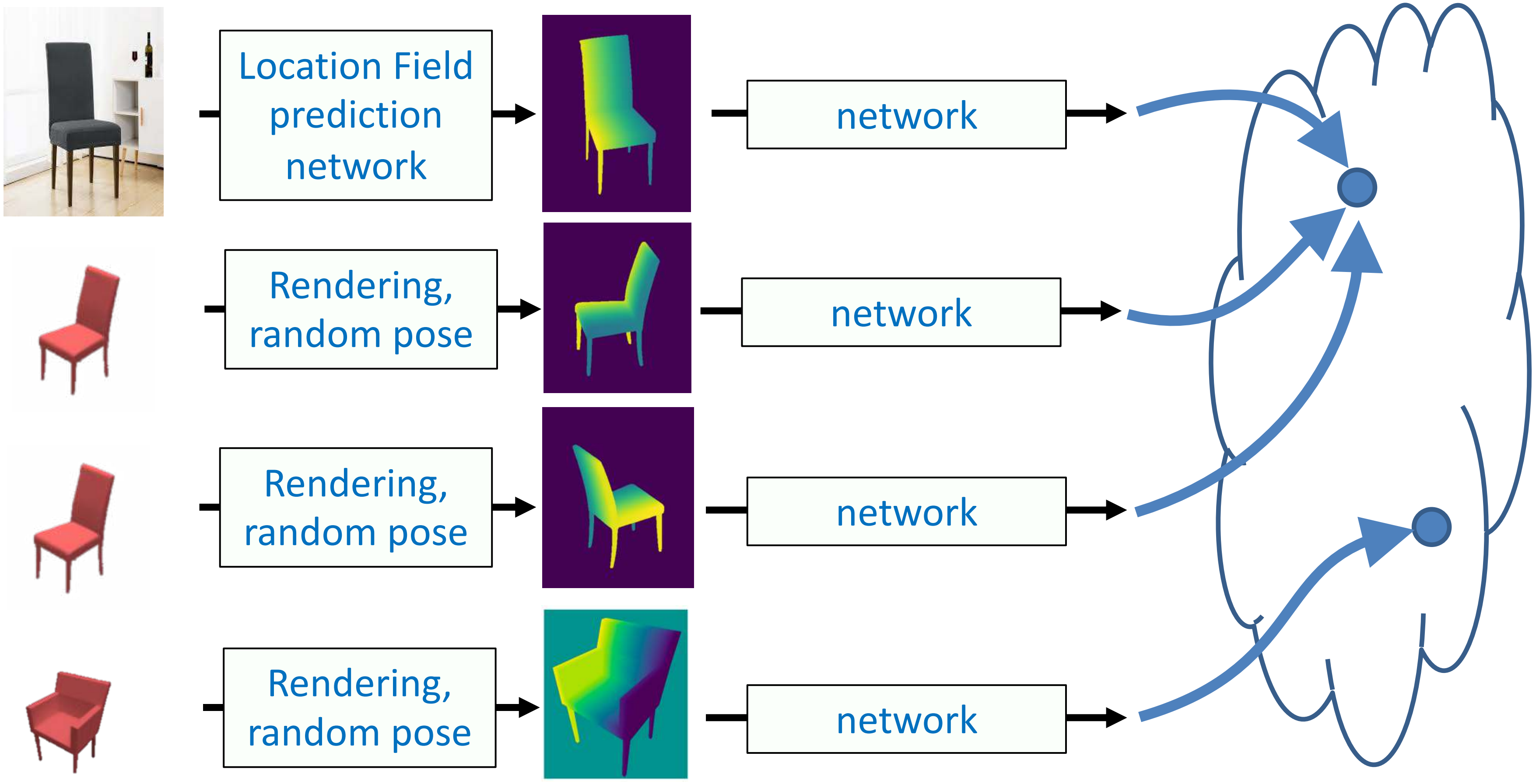
Pose Invariant Embeddings & Metric Learning



Pose Invariant Embeddings & Metric Learning

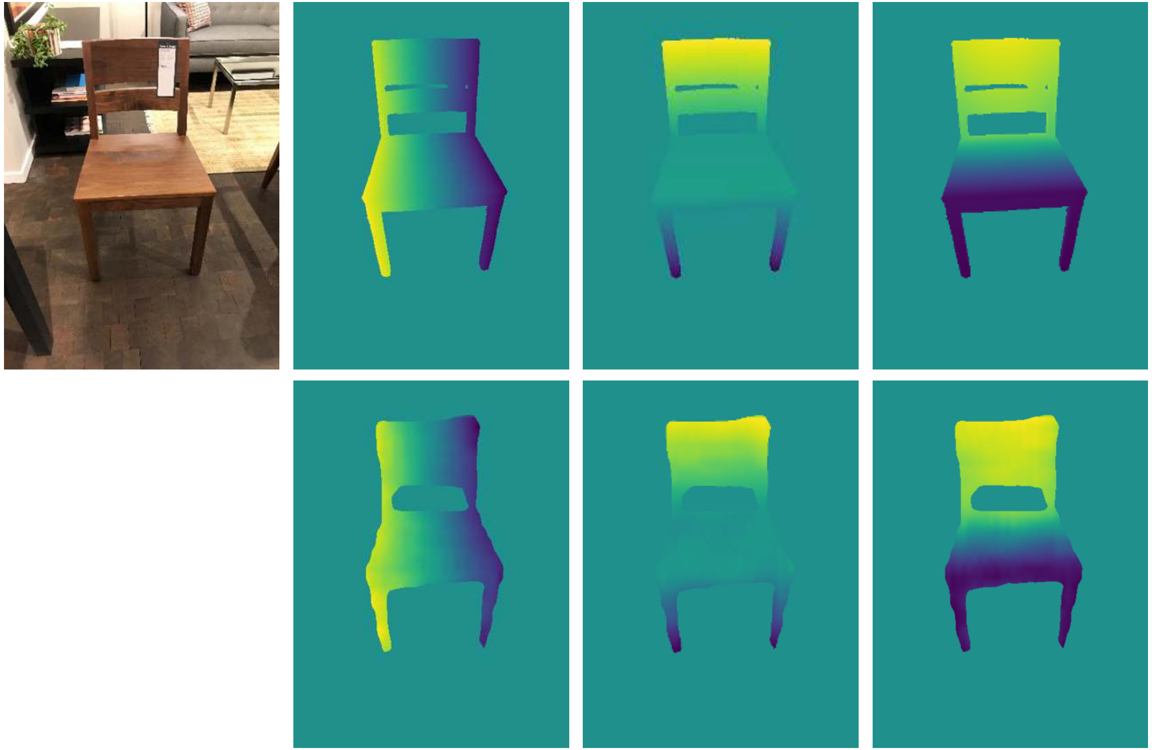


Pose Invariant Embeddings & Metric Learning



Predicted Location Fields

image



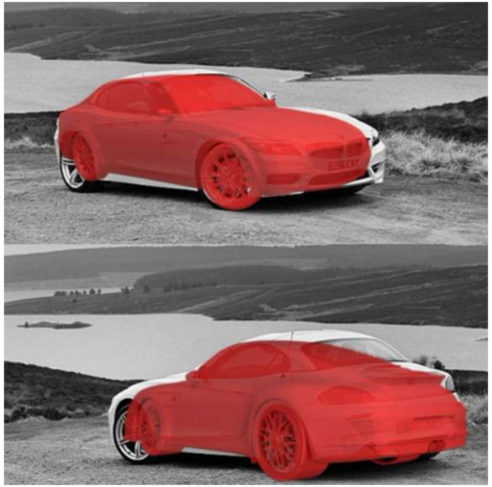
Ground Truth

Prediction

LF (X) LF (Y) LF (Z)

Results on Pix3D [Sun et al, 2018]






Limitation

3D Pose Refinement

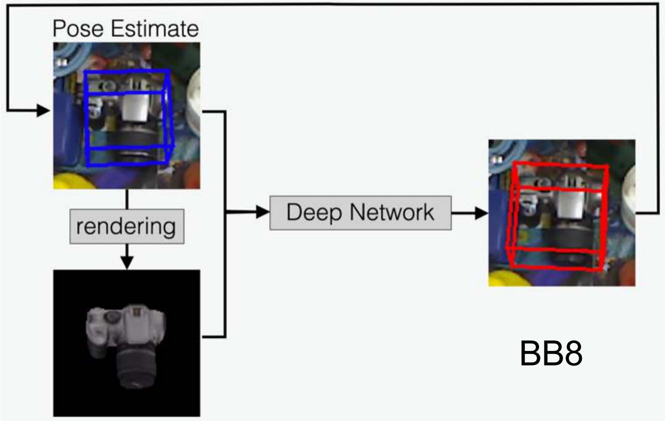
initial 3D pose

refined 3D pose

Geometric Correspondence Fields: Learned 3D Pose Refinement in the Wild. A. Grabner et al., ECCV 2020

Refinement Strategies

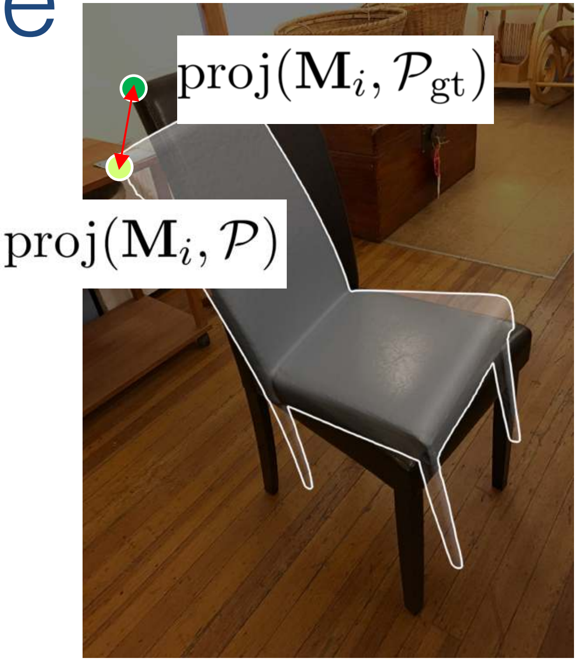
Option #1:
Compare images and renderings to predict 3D pose updates.



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.

Option #2:
Make a renderer differentiable and compute analytic 3D pose gradients.

Our Refinement Objective



Consider the geometric reprojection error:

$$e(\mathcal{P}) = \frac{1}{2}\sum_i \|\mathrm{proj}(\mathbf{M}_i, \mathcal{P}_{\mathrm{gt}}) - \mathrm{proj}(\mathbf{M}_i, \mathcal{P})\|_2^2$$

estimated
3D pose

ground truth
3D pose

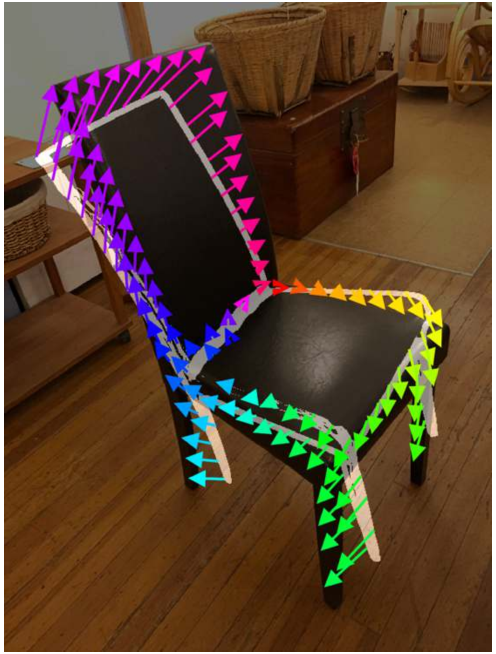
Gradient with respect to the 3D pose:

$$\left(\frac{\partial e(\mathcal{P})}{\partial \mathcal{P}}\right)(\mathcal{P}_{\mathrm{curr}}) = \sum_i \underbrace{\left[\frac{\partial \mathrm{proj}(\mathbf{M}_i, \mathcal{P})}{\partial \mathcal{P}}\right]^T}_{\text{analytic}} \underbrace{\left(\mathrm{proj}(\mathbf{M}_i, \mathcal{P}_{\mathrm{gt}}) - \mathrm{proj}(\mathbf{M}_i, \mathcal{P}_{\mathrm{curr}})\right)}_{\text{unknown}}$$

Our Refinement Objective

Consider the geometric reprojection error:

$$\underbrace{e(\mathcal{P})}_{\text{estimated 3D pose}} = \frac{1}{2}\sum_i \|\mathrm{proj}(\mathbf{M}_i, \underbrace{\mathcal{P}_{\mathrm{gt}}}_{\text{ground truth 3D pose}}) - \mathrm{proj}(\mathbf{M}_i, \mathcal{P})\|_2^2$$

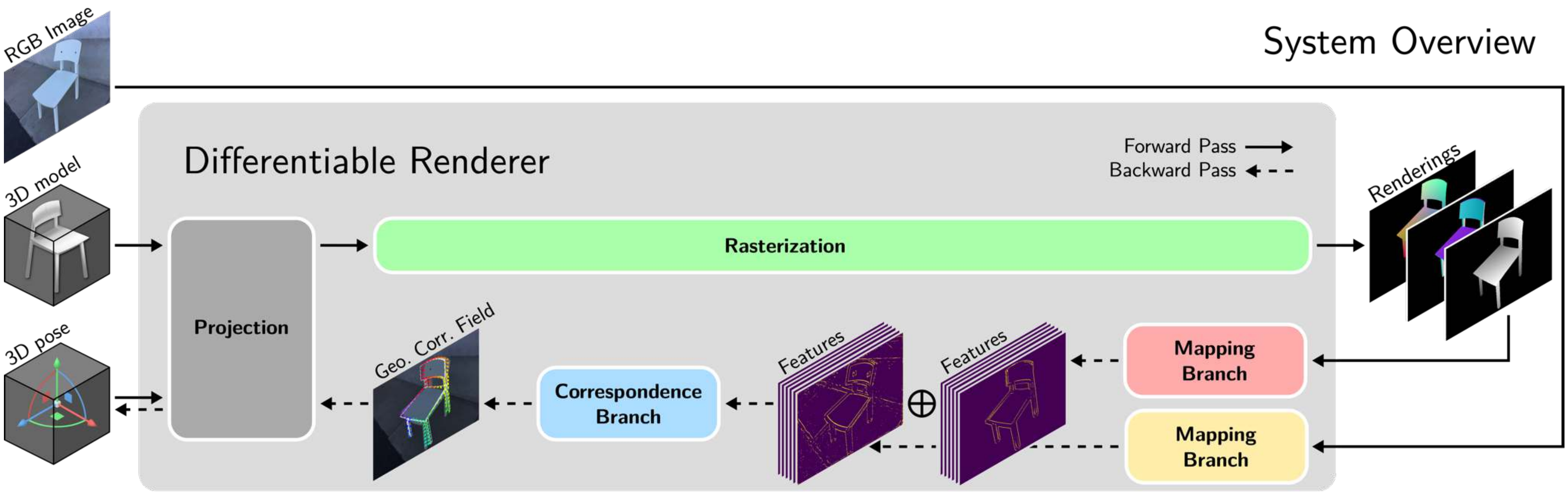


prediction

Gradient with respect to the 3D pose:

$$\left(\frac{\partial e(\mathcal{P})}{\partial \mathcal{P}}\right)(\mathcal{P}_{\mathrm{curr}}) = \sum_i \underbrace{\left[\frac{\partial \mathrm{proj}(\mathbf{M}_i, \mathcal{P})}{\partial \mathcal{P}}\right]^T}_{\text{analytic}} \underbrace{\left(\mathrm{proj}(\mathbf{M}_i, \mathcal{P}_{\mathrm{gt}}) - \mathrm{proj}(\mathbf{M}_i, \mathcal{P}_{\mathrm{curr}})\right)}_{\text{predicted}}$$

System Overview



Refinement Progress

Image | GT Pose | Initial Pose | Refined Pose

3D Annotations are Hard...

Pix3D dataset

How Can We Train a Deep Network for 3D Computer Vision?

1. On annotated real images;

annotation is difficult, time consuming, ..

Pix3D dataset

2. On synthetic images;

domain gap, content creation, ..

Structured3D dataset

3. Using self-learning;

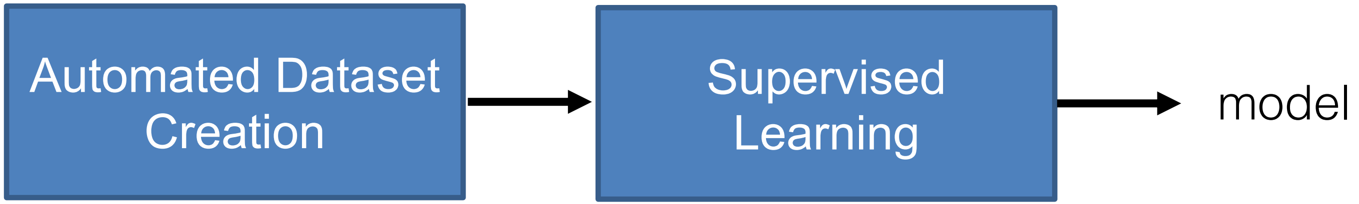
cool

How Can We *Evaluate* a Deep Network for 3D Computer Vision?

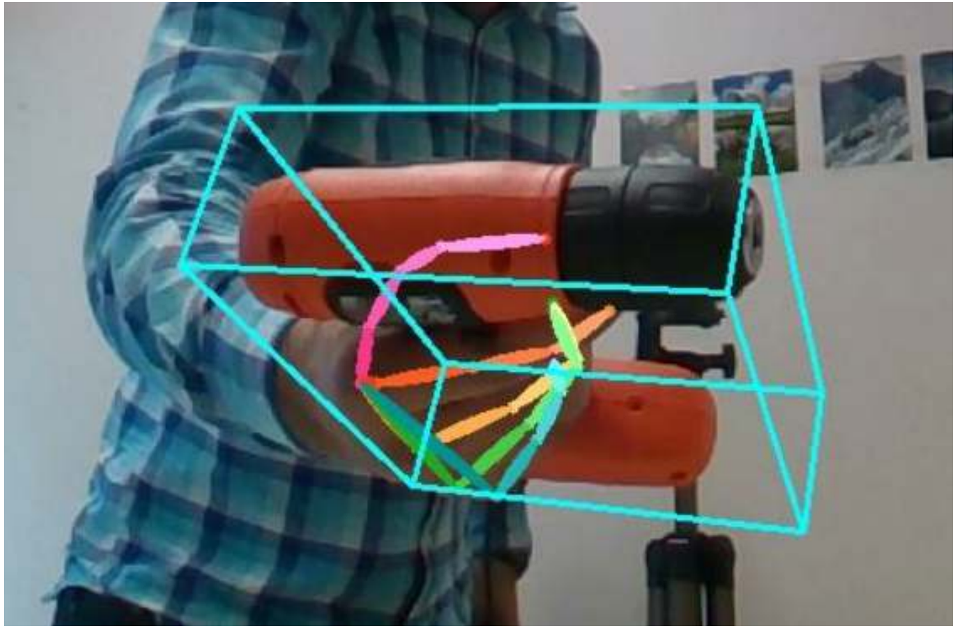
1. On *accurately* annotated real images;

Proposed Approach

A method for automatically creating a dataset of 3D annotations of real images that we can evaluate;

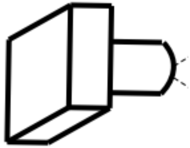


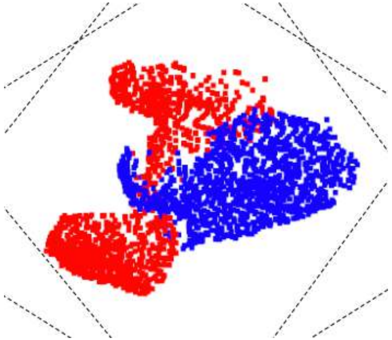
Automatically Creating a Dataset for 3D Hand+Object Pose Estimation



HOnnotate: A method for 3D Annotation of Hand and Object Poses. Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. CVPR 2020.

Automated Annotations





- 1 or more RGB-D cameras;
- temporal constraints.

Object 3D model from YCB [Xiang et al, 2018]

MANO model [Romero et al, 2017]

Bayesian Formulation

$$\max_{\{(\mathbf{p}_t^H, \mathbf{p}_t^O)\}_t} \prod_t \prod_c p((I_t^c, D_t^c) \mid \mathbf{p}_t^H, \mathbf{p}_t^O) \, p(\mathbf{p}_{t+1}^H, \mathbf{p}_{t+1}^O \mid \mathbf{p}_t^H, \mathbf{p}_t^O) \, p(\mathbf{p}_t^H, \mathbf{p}_t^O)$$

Bayesian Formulation

$$\max_{\{(\mathbf{p}_t^H,\mathbf{p}_t^O)\}_t} \prod_t \prod_c \overbrace{p((I_t^c, D_t^c) \mid \mathbf{p}_t^H, \mathbf{p}_t^O)}^{\text{RGB-D likelihoods}} \underbrace{p(\mathbf{p}_{t+1}^H, \mathbf{p}_{t+1}^O \mid \mathbf{p}_t^H, \mathbf{p}_t^O)}_{\text{temporal constraints}} \overbrace{p(\mathbf{p}_t^H, \mathbf{p}_t^O)}^{\text{physical constraints}}$$

hand pose at time t

object pose at time t

depth map from camera $\mathbf{c}$ at time t

color image from camera c at time t

RGBD Likelihood

$$\max_{\{(\mathbf{p}_t^H,\mathbf{p}_t^O)\}_t} \prod_t \prod_c \underset{\text{RGBD likelihoods}}{\underline{p((I_t^c, D_t^c) \mid \mathbf{p}_t^H, \mathbf{p}_t^O)}}\; p(\mathbf{p}_{t+1}^H, \mathbf{p}_{t+1}^O \mid \mathbf{p}_t^H, \mathbf{p}_t^O)\; p(\mathbf{p}_t^H, \mathbf{p}_t^O)$$

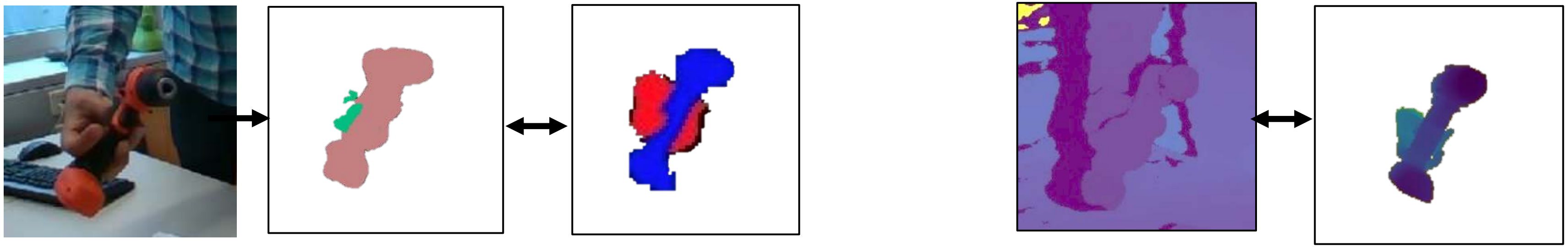


image I_t^c | observed masks from I_t^c | predicted masks for $\mathbf{p}_t^H, \mathbf{p}_t^O$ | observed depth D_t^c | predicted depth for $\mathbf{p}_t^H, \mathbf{p}_t^O$

- Efficient way to deal with occlusions hand/object;
- Gradient computed with differential renderer.

Physical Constraints

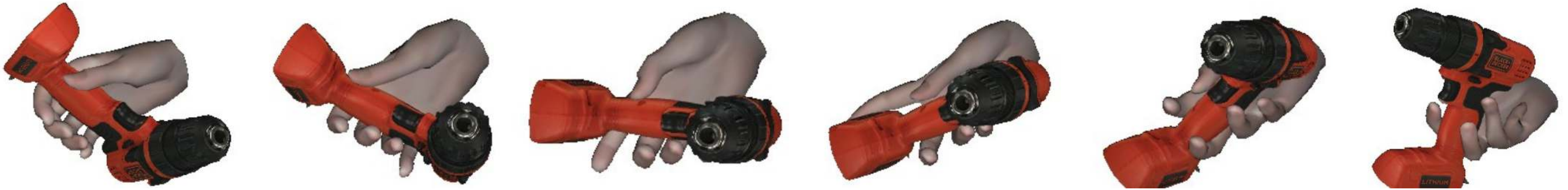
$$\max_{\{(\mathbf{p}_t^H,\mathbf{p}_t^O)\}_t} \prod_t \prod_c p((I_t^c, D_t^c) \mid \mathbf{p}_t^H, \mathbf{p}_t^O)\; p(\mathbf{p}_{t+1}^H, \mathbf{p}_{t+1}^O \mid \mathbf{p}_t^H, \mathbf{p}_t^O)\; \overbrace{p(\mathbf{p}_t^H, \mathbf{p}_t^O)}^{\text{physical constraints}}$$

joint angle constraints

no intersection constraint

Temporal Constraints

$$\max_{\{(\mathbf{p}_t^H, \mathbf{p}_t^O)\}_t} \prod_t \prod_c p((I_t^c, D_t^c) \mid \mathbf{p}_t^H, \mathbf{p}_t^O) \underbrace{p(\mathbf{p}_{t+1}^H, \mathbf{p}_{t+1}^O \mid \mathbf{p}_t^H, \mathbf{p}_t^O)}_{\text{temporal constraints}} p(\mathbf{p}_t^H, \mathbf{p}_t^O)$$



simple 0-order motion model

Optimization

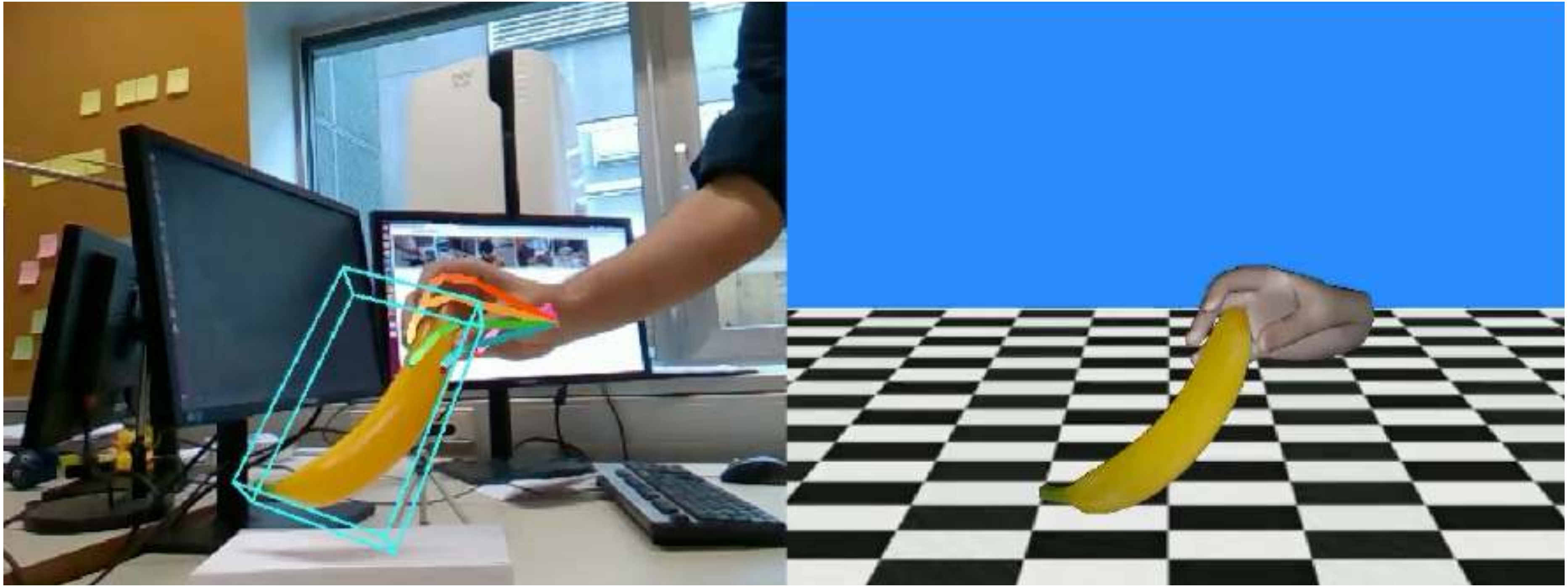
$$\max_{\{(\mathbf{p}_t^H,\mathbf{p}_t^O)\}_t} \prod_t \prod_c p((I_t^c, D_t^c) \mid \mathbf{p}_t^H, \mathbf{p}_t^O)\; p(\mathbf{p}_{t+1}^H, \mathbf{p}_{t+1}^O \mid \mathbf{p}_t^H, \mathbf{p}_t^O)\; p(\mathbf{p}_t^H, \mathbf{p}_t^O)$$

Negative log:

$$\begin{aligned}\min_{\{(\mathbf{p}_t^H,\mathbf{p}_t^O)\}_t} \sum_t \quad \sum_c\; & \alpha\|S_t^c - S(\mathbf{p}_t^H, \mathbf{p}_t^O)\|^2 + \beta\|D_t^c - D(\mathbf{p}_t^H, \mathbf{p}_t^O)\|^2 + \\ & \gamma E_{\text{joints}}(\mathbf{p}_t^H) + \delta E_{\text{inters}}(\mathbf{p}_t^H, \mathbf{p}_t^O) + \\ & \epsilon E_{\text{temp}}(\mathbf{p}_t^H, \mathbf{p}_t^O, \mathbf{p}_{t-1}^H, \mathbf{p}_{t-1}^O, \mathbf{p}_{t+1}^H, \mathbf{p}_{t+1}^O) + \\ & \eta E_{3D}(\{D_t^c\}_c, \mathbf{p}_t^H, \mathbf{p}_t^O)\end{aligned}$$

Optimized using Adam.

Automated 3D Annotations



Validating our Annotations

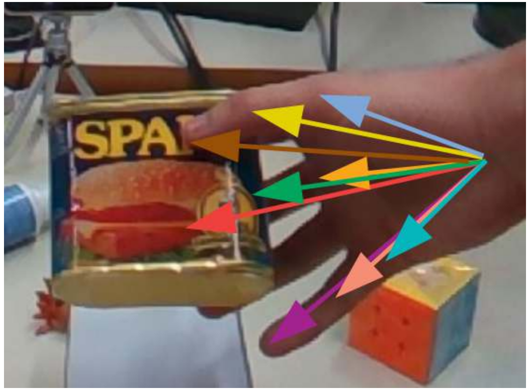
Manual annotations of 3D joints on 100 randomly selected time steps;

Done directly on the point cloud created from 4 cameras;

→ Mean error is 8mm with 4 cameras;
→ Mean error is 10mm with 1 camera.


SPA

Using our 3D Annotations for Single RGB Frame Prediction



MANO model [Romero et al, 2017]

We train a network to predict:
– 2D keypoint locations;
– Root relative joint directions.

+ MANO model fitted to these predictions

Using the Annotations for Single RGB Frame Prediction

(objects are unknown)

3D Scene Understanding from an Image

Alexander Grabner

Sinisa Stekovic

Shreyas Hampali

Madhi Rad

Peter Roth

Friedrich Fraundorfer

Qualcomm

Christian Doppler Forschungsgesellschaft

Thanks for listening!

Questions?