# Geometric Correspondence Fields: Learned Differentiable Rendering for 3D Pose Refinement in the Wild

—

# Supplementary Material

Alexander Grabner[1,2], Yaming Wang[2], Peizhao Zhang[2], Peihong Guo[2], Tong Xiao[2], Peter Vajda[2], Peter M. Roth[1], and Vincent Lepetit[1]

[1] Graz University of Technology, Austria [2] Facebook Inc., USA
{alexander.grabner, pmroth, lepetit}@icg.tugraz.at
{wym, stzpz, phg, xiaot, vajdap}@fb.com

In the following, we provide additional details and experimental results of our novel 3D pose refinement approach. In Sec. 1, we present our evaluation setup and discuss different datasets. In Sec. 2, we formally describe our evaluated metrics. In Sec. 3, we give specific details on the implementation of our approach. In Sec. 4, we analyze the iterative refinement behavior of different methods. In Sec. 5, we present detailed quantitative 3D pose refinement results for individual object categories. In Sec. 6, we show additional qualitative results. Finally, we analyze failure cases of our approach in Sec. 7.

## 1 Datasets and Evaluation Setup

We evaluate our proposed 3D pose refinement approach on the challenging Pix3D [19] dataset. The Pix3D dataset provides in-the-wild RGB images with 3D pose, 3D model, and focal length annotations for objects of different categories. We follow the evaluation protocol of previous work [3] and perform experiments on categories which have more than 300 non-occluded and non-truncated samples (*bed*, *chair*, *sofa*, *table*). Further, we restrict the training and evaluation to samples marked as non-occluded and non-truncated because all evaluated refinement methods lack explicit mechanisms to deal with occlusions. In addition, this dataset does not provide information on which objects parts are occluded nor the extent of the occlusion. For each category, we use 50% of the samples for training and the other 50% for testing as in [3].

Other category-level datasets do not provide annotations with sufficient accuracy to both train and evaluate fine-grained 3D pose refinement methods. For example, the Comp [22] and Stanford [22] datasets only provide coarse 3D pose annotations. In addition to coarse 3D pose annotations, the ScanNet [2], Pascal3D+ [24], and ObjectNet3D [23] datasets also just provide approximate 3D model annotations. Moreover, the latter two datasets assume constant camera intrinsics for images captured with different cameras which further decreases the annotation quality [3]. As a consequence of this label noise, the training

of refinement methods results in models with poor accuracy, while quantitative evaluations are not representative of the true refinement performance due to the lack of accuracy in the annotations.

In contrast, instance-level datasets like LineMOD [8], YCB [1], T-LESS [10], or NOCS [21] provide accurate annotations but have many images with strong occlusions. Neither traditional refinement methods [14, 16, 25], nor differentiable rendering based refinement methods [12, 15, 18], nor our approach employ explicit mechanisms to deal with occlusions. For example, one issue across all evaluated methods is that they align renderings to real-world images but occlusions are only present in the real-world images while the renderings are always un-occluded. Also, simply training methods based on feed-forward CNNs on occluded objects is in practice not sufficient to handle occlusions [17]. However, we specifically plan to address occlusions in the future by predicting occlusion masks and correspondence confidences.

In addition, our approach is specifically designed for category-level 3D pose refinement using untextured 3D models. This task is very different from instance-level 3D pose estimation where exactly matching colored and textured 3D models are available. In this case, methods which leverage color and texture information of 3D models have a clear advantage and should be used instead.

For these two reasons, we did not evaluate our method on instance-level datasets like LineMOD or YCB which have many images with strong occlusions and provide 3D models with colors and textures that exactly match those of the objects in the RGB images.

## 2 Metrics

We follow the evaluation protocol of previous work [3] and report the median error ($MedErr$) of multiple geometric distances:

**Rotation:** The 3D rotation distance

$$e_R = \frac{\|\log(\mathbf{R}_{\text{gt}}^T \mathbf{R}_{\text{pred}})\|_F}{\sqrt{2}} \tag{1}$$

represents the minimal angle between the ground truth rotation matrix $\mathbf{R}_{\text{gt}}$ and the predicted rotation matrix $\mathbf{R}_{\text{pred}}$ [20].

**Translation:** The 3D translation distance

$$e_t = \frac{\|\mathbf{t}_{\text{gt}} - \mathbf{t}_{\text{pred}}\|_2}{\|\mathbf{t}_{\text{gt}}\|_2} \tag{2}$$

gives the relative error between the ground truth translation $\mathbf{t}_{\text{gt}}$ and the predicted translation $\mathbf{t}_{\text{pred}}$ [11].

**Pose:** The 3D pose distance

$$e_{R,t} = \underset{\mathbf{M}_i \in \mathcal{M}}{\text{avg}} \frac{d_{\text{bbox}}}{d_{\text{img}}} \cdot \frac{\|\text{transf}(\mathbf{M}_i, \mathcal{P}_{\text{gt}}) - \text{transf}(\mathbf{M}_i, \mathcal{P}_{\text{pred}})\|_2}{\|\mathbf{t}_{\text{gt}}\|_2} \tag{3}$$

represents the average normalized Euclidean distance of all transformed 3D model points in 3D space [9, 11]. Each 3D point $\mathbf{M}_i$ of the ground truth 3D model $\mathcal{M}$ is transformed using the ground truth 3D pose $\mathcal{P}_{\text{gt}}$ and the predicted 3D pose $\mathcal{P}_{\text{pred}}$. This distance is normalized by the relative size of the object in the image using the ratio between the ground truth 2D bounding box diagonal $d_{\text{bbox}}$ and the image diagonal $d_{\text{img}}$, and the L2-norm of the ground truth translation $\|\mathbf{t}_{\text{gt}}\|_2$.

**Projection:** The 2D projection distance

$$e_P = \underset{\mathbf{M}_i \in \mathcal{M}}{\text{avg}} \frac{\|\text{proj}(\mathbf{M}_i, \mathcal{P}_{\text{gt}}) - \text{proj}(\mathbf{M}_i, \mathcal{P}_{\text{pred}})\|_2}{d_{\text{bbox}}} \tag{4}$$

is the average reprojection error normalized by the ground truth 2D bounding box diagonal $d_{\text{bbox}}$ [22]. In this case, each 3D point $\mathbf{M}_i$ of the ground truth 3D model $\mathcal{M}$ is projected to the 2D image plane using the ground truth 3D pose $\mathcal{P}_{\text{gt}}$ and the predicted 3D pose $\mathcal{P}_{\text{pred}}$ subject to a camera model. In this work, we assume the camera intrinsics to be known.

## 3 Implementation and Training Details

To train our refinement network, we resize and pad images to a spatial resolution of $256 \times 256$ while maintaining the aspect ratio. In this way, we are able to combine images with different aspect ratios in the same training batch. For our mapping branches, we adapt a ResNet-50 [6, 7] architecture. We utilize all layers up to the end of the first stage but use a stride of 1 for all convolutional layers and discard max pooling layers. For our correspondence branch, we use a channel dimensionality of 64 for all convolutional layers except the output layer.

During training of our network, we additionally employ different forms of data augmentation like mirroring, affine transformations, and independent pixel augmentations like additive or multiplicative noise for the RGB image $I$.

To regularize our network, we use L2 weight decay with a factor of $1e^{-5}$. We train our network $f(\cdot)$ for 1500 epochs using the Adam optimizer [13] with an initial learning rate of $\eta = 1e^{-3}$. We use a batch size of 8 and decrease the learning rate by a factor of 5 after 1000 and 1400 epochs.

During inference of our system, we compute geometry-level gradients $\nabla \mathbf{m}_i$ from our predicted geometric correspondence fields. In this way, vertices belonging to self-occluded triangles [12] or to visible triangles which are masked out by our geometric attention module do not receive gradients. However, since the geometry is fixed, providing gradients for a subset of all vertices is sufficient to perform 3D pose refinement.

## 4 Iterative Refinement

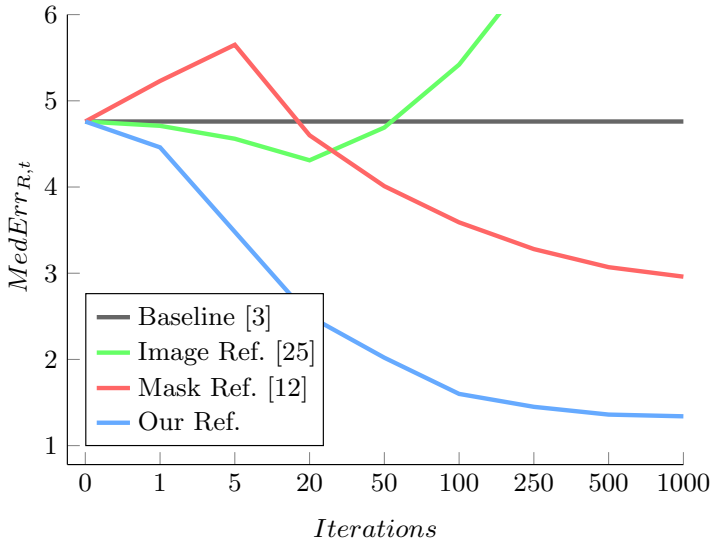


Fig. 1: Evaluation on 3D pose refinement after varying numbers of iterations. In contrast to other methods, our refinement achieves a consistent improvement upon the baseline, increasing with the number of iterations.

Figure 1 shows the performance of different refinement methods after varying numbers of iterations. In this experiment, we report the 3D pose metric $MedErr_{R,t}$ as a function of the number of 3D pose updates. The baseline does not perform iterative updates, thus, its $MedErr_{R,t}$ score is constant.

For *Image Refinement* [25], the accuracy increases until 20 iterations but then starts to decrease. After the first couple of coarse refinement steps, the predicted updates are not accurate enough to refine the 3D pose but start to jitter without further improving the 3D pose. Moreover, for many objects the prediction fails and the iterative updates cause the 3D pose to drift off which results in high $MedErr_{R,t}$ for large numbers of iterations.

For *Mask Refinement* [12], we observe an opposite effect. In the beginning, the accuracy decreases but then the performance increases. This is due to degenerated masks predicted from RGB images by Mask R-CNN [5]. The mask prediction often fails to capture fine-grained and thin structures, *e.g.*, ornaments and legs of a bed (see Figure 2). These degenerated masks cause large gradients during refinement and quickly pull the 3D pose away from the reasonable initial estimate predicted by the baseline (also see Figure 3, 2$^{\text{nd}}$ row). After five iterations the refinement on samples with correctly predicted masks counteracts this effect and the performance improves.

Table 1: Average computation times of different refinement methods for a single iteration using a Titan X GPU. For all evaluated methods, the execution time for computing a single 3D pose update is within the same order of magnitude.

| Method | Time per Iteration |
|---|---|
| Image Refinement [25] | 86.5 ms |
| Mask Refinement [12] | 29.7 ms |
| Our Refinement | 36.0 ms |

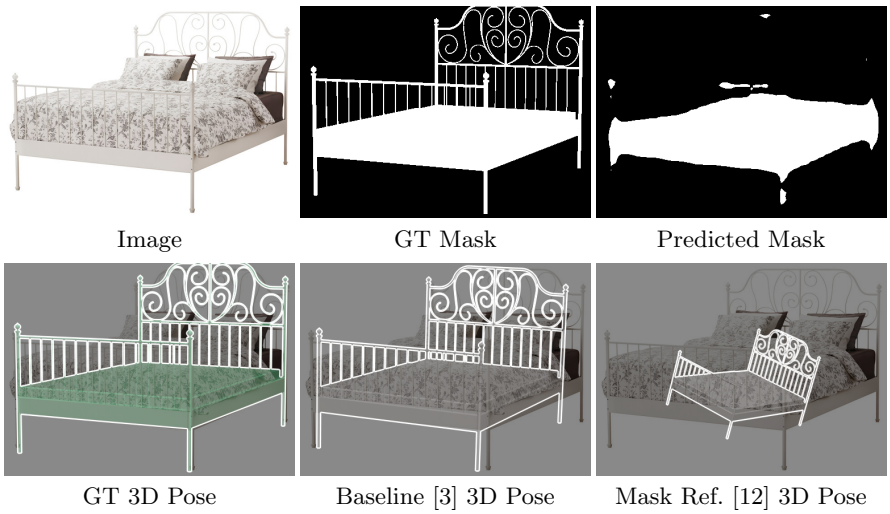


Fig. 2: Typical failure case of *Mask Refinement* [12]. Degenerated predicted masks (*top right*) cause large gradients during refinement and quickly pull the 3D pose away from reasonable initial estimates by the baseline (*bottom middle*). As a consequence, the 3D pose refinement using *Mask Refinement* fails (*bottom right)*. Also see Figure 3, 2$^{\text{nd}}$ row.

In contrast to other refinement methods, our approach achieves a consistent improvement upon the baseline, increasing with the number of iterations. As expected, the accuracy saturates for large numbers of iterations. Empirically, we achieve maximum accuracy by performing 1000 3D pose updates using the Adam optimizer [13] with a learning rate of $\eta = 0.05$.

Finally, Table 1 compares the computation times of different refinement methods. *Image Refinement* evaluates two large ResNet-style networks [6] during inference and, thus, this method is the slowest in our evaluation. *Mask Refinement* generates a mask rendering, computes a loss in the mask space, and performs the backward pass of its differentiable renderer in each iteration. This method is the fastest in our evaluation since the target mask predicted from the input RGB image by Mask R-CNN [5] does not change during refinement. It is only predicted once before the iterative process and the inference time of Mask R-CNN is not considered in this experiment.

Our refinement is only marginally slower than *Mask Refinement* as we just evaluate our efficient network branches in addition to the forward and backward pass of our differentiable renderer in each iteration. However, all evaluated refinement methods show comparable execution time within the same order of magnitude for computing a single 3D pose update.

## 5 Detailed Quantitative Results

Tables 2 (*GT 3D models*) and 3 (*retrieved 3D models*) show detailed quantitative results for individual object categories on Pix3D. For completeness, we additionally report the detection accuracy $Acc_{D_{0.5}}$ which gives the percentage of objects for which the intersection over union between the ground truth 2D bounding box and the predicted 2D bounding box is larger than 50% [24]. We do not make 3D pose predictions for objects which are not detected by the baseline [3]. However, the reported $MedErr$ metrics are computed over all samples, both detected and not detected.

Our refinement outperforms the baseline as well as competing refinement methods across all metrics. In fact, we do not only increase the *mean* performance over all categories but also achieve state-of-the-art results for each individual category. Using both ground truth (see Table 2) and retrieved (see Table 3) 3D models, we improve the performance compared to other methods by a large margin for each evaluated category.

Table 2: Detailed 3D pose refinement results for individual categories on Pix3D. In this experiment, we provide the **ground truth 3D model** for refinement. We outperform existing methods across all categories by a large margin.

| | | **Detection** | **Rotation** | **Translation** | **Pose** | **Projection** |
|---|---|---|---|---|---|---|
| Method | Category | $Acc_{D_{0.5}}$ | $MedErr_R$ $\cdot 1$ | $MedErr_t$ $\cdot 10^2$ | $MedErr_{R,t}$ $\cdot 10^2$ | $MedErr_P$ $\cdot 10^2$ |
| Baseline [3] | bed | 99.0 | 5.07 | 6.68 | 5.18 | 3.42 |
| Image Refinement [25] | | | 4.65 | 5.45 | 4.60 | 3.38 |
| Mask Refinement [12] | | | 3.03 | 4.04 | 3.07 | 2.00 |
| Our Refinement | | | **2.40** | **1.84** | **1.45** | **1.28** |
| Baseline [3] | chair | 95.2 | 7.36 | 5.49 | 3.90 | 3.32 |
| Image Refinement [25] | | | 7.10 | 5.31 | 3.68 | 3.34 |
| Mask Refinement [12] | | | 4.42 | 4.89 | 3.17 | 1.79 |
| Our Refinement | | | **2.96** | **1.77** | **1.23** | **1.17** |
| Baseline [3] | sofa | 96.5 | 4.40 | 4.96 | 3.78 | 2.57 |
| Image Refinement [25] | | | 4.30 | 3.87 | 3.15 | 2.54 |
| Mask Refinement [12] | | | 2.97 | 2.89 | 2.25 | 1.54 |
| Our Refinement | | | **2.28** | **1.36** | **1.19** | **1.08** |
| Baseline [3] | table | 94.0 | 10.18 | 7.72 | 6.17 | 5.54 |
| Image Refinement [25] | | | 9.81 | 7.07 | 5.80 | 5.40 |
| Mask Refinement [12] | | | 3.81 | 4.44 | 3.34 | 2.27 |
| Our Refinement | | | **2.59** | **2.00** | **1.48** | **1.55** |
| Baseline [3] | *mean* | 96.2 | 6.75 | 6.21 | 4.76 | 3.71 |
| Image Refinement [25] | | | 6.46 | 5.43 | 4.31 | 3.67 |
| Mask Refinement [12] | | | 3.56 | 4.06 | 2.96 | 1.90 |
| Our Refinement | | | **2.56** | **1.74** | **1.34** | **1.27** |

Table 3: Detailed 3D pose refinement results for individual categories on Pix3D. In this experiment, we automatically **retrieve 3D models** for refinement using the method presented in [4]. We outperform existing methods across all categories by a large margin.

| | | **Detection** | **Rotation** | **Translation** | **Pose** | **Projection** |
|---|---|---|---|---|---|---|
| Method | Category | $Acc_{D_{0.5}}$ | $MedErr_R$ $\cdot 1$ | $MedErr_t$ $\cdot 10^2$ | $MedErr_{R,t}$ $\cdot 10^2$ | $MedErr_P$ $\cdot 10^2$ |
| Baseline [3] | bed | 99.0 | 5.07 | 6.68 | 5.18 | 3.42 |
| Image Refinement [25] | | | 4.65 | 5.86 | 4.41 | 3.38 |
| Mask Refinement [12] | | | 4.40 | 5.32 | 4.21 | 2.69 |
| Our Refinement | | | **2.95** | **2.75** | **2.18** | **1.83** |
| Baseline [3] | chair | 95.2 | 7.36 | 5.49 | 3.90 | 3.32 |
| Image Refinement [25] | | | 7.15 | 5.21 | 3.67 | 3.35 |
| Mask Refinement [12] | | | 7.22 | 6.32 | 4.53 | 3.31 |
| Our Refinement | | | **4.89** | **2.87** | **2.04** | **2.19** |
| Baseline [3] | sofa | 96.5 | 4.40 | 4.96 | 3.78 | 2.57 |
| Image Refinement [25] | | | 4.34 | 3.75 | 3.02 | 2.54 |
| Mask Refinement [12] | | | 3.33 | 3.00 | 2.34 | 1.63 |
| Our Refinement | | | **2.60** | **1.60** | **1.42** | **1.19** |
| Baseline [3] | table | 94.0 | 10.18 | 7.72 | 6.17 | 5.54 |
| Image Refinement [25] | | | 9.73 | 7.23 | 6.22 | 5.68 |
| Mask Refinement [12] | | | 6.92 | 6.34 | 5.52 | 4.85 |
| Our Refinement | | | **4.73** | **3.38** | **2.93** | **3.49** |
| Baseline [3] | *mean* | 96.2 | 6.75 | 6.21 | 4.76 | 3.71 |
| Image Refinement [25] | | | 6.47 | 5.51 | 4.33 | 3.74 |
| Mask Refinement [12] | | | 5.47 | 5.25 | 4.15 | 3.12 |
| Our Refinement | | | **3.79** | **2.65** | **2.14** | **2.18** |

## 6 Additional Qualitative Results

Figures 3, 4, and 5 show additional qualitative 3D pose refinement results for different methods complementary to those presented in the main paper. While other methods fail to predict fine-grained 3D poses in the wild, our approach precisely aligns 3D models to objects in RGB images which results in significantly improved 3D poses for objects of different categories. In many cases, our predicted 3D pose is visually indistinguishable from the ground truth 3D pose.

Figures 6 and 7 show additional qualitative results of our predicted geometric correspondence fields. Our predicted 2D displacement vectors are highly accurate for many different objects and scales in the wild. The illustrations also show our computed geometric attention masks, outlined in white underneath the 2D displacement vectors.

## 7 Additional Failure Cases

Figure 8 shows additional failure cases of our approach. In the presence of strong image noise, we cannot predict accurate geometric correspondence fields and, thus, our refinement fails. Also, if there are duplicate or ambiguous structures in the image our method sometimes predicts wrong correspondences and aligns the 3D model to unintended image parts.

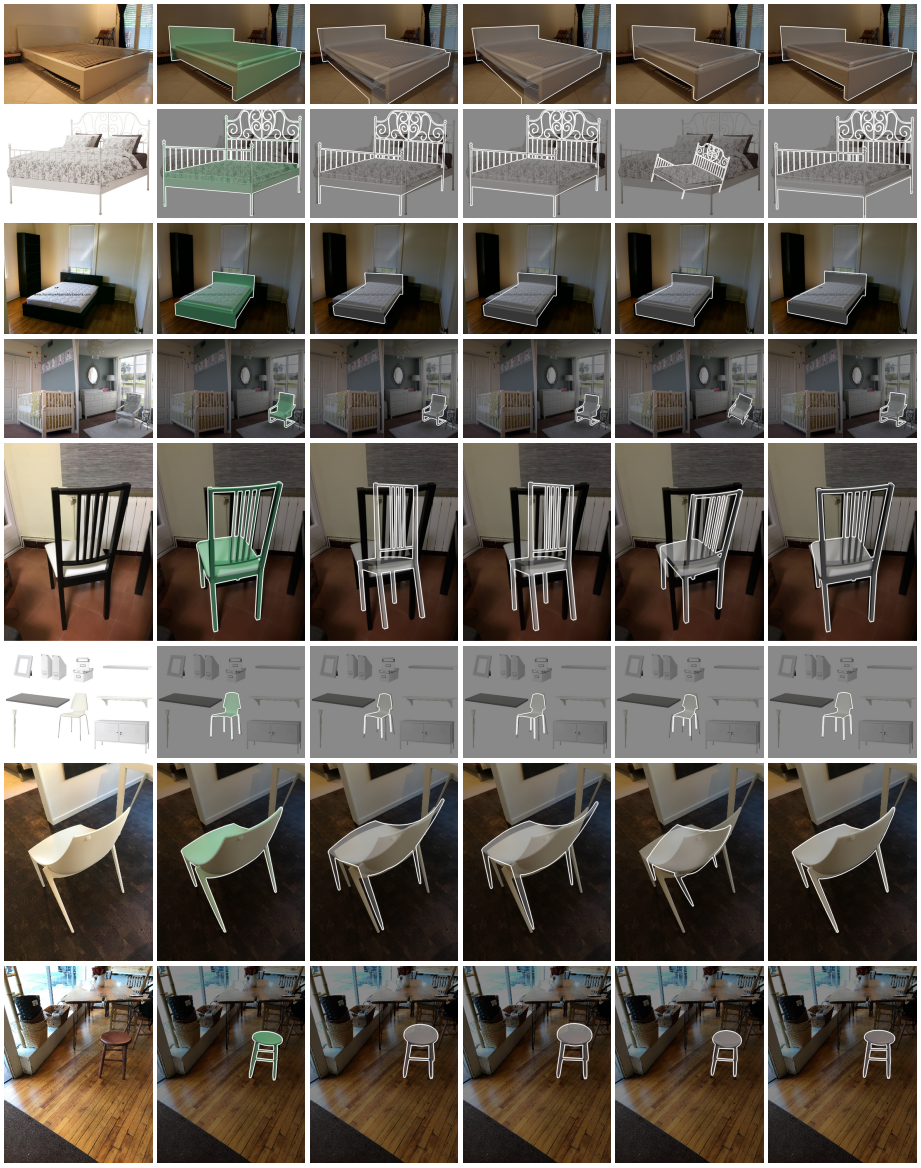


Fig. 3: Additional qualitative 3D pose refinement results for objects of different categories. We project the ground truth 3D model on the image using the predicted 3D pose. Our approach overcomes the limitations of previous methods and predicts fine-grained 3D poses which are in many cases visually indistinguishable from the ground truth. Best viewed in **digital zoom**.

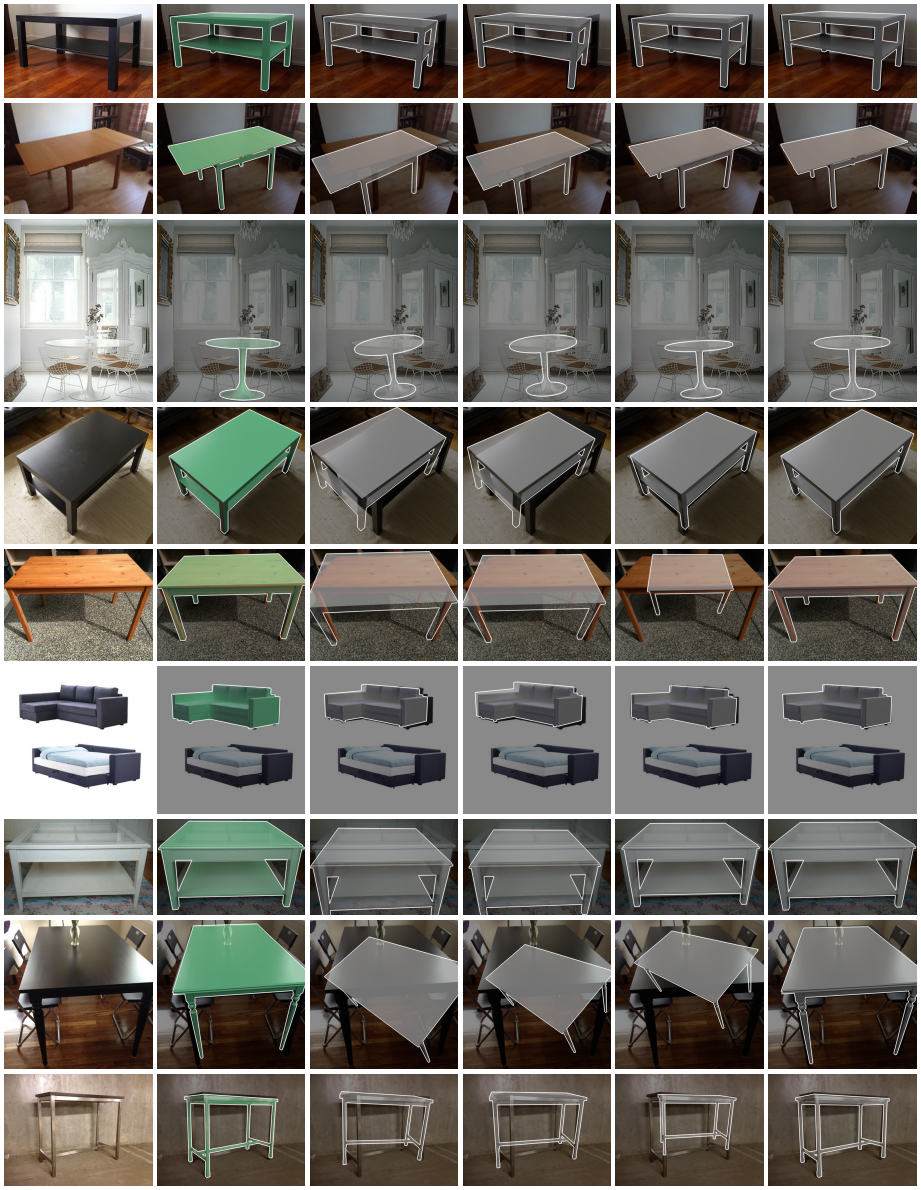


Fig. 4: Additional qualitative 3D pose refinement results for objects of different categories. We project the ground truth 3D model on the image using the predicted 3D pose. Our approach overcomes the limitations of previous methods and predicts fine-grained 3D poses which are in many cases visually indistinguishable from the ground truth. Best viewed in **digital zoom**.

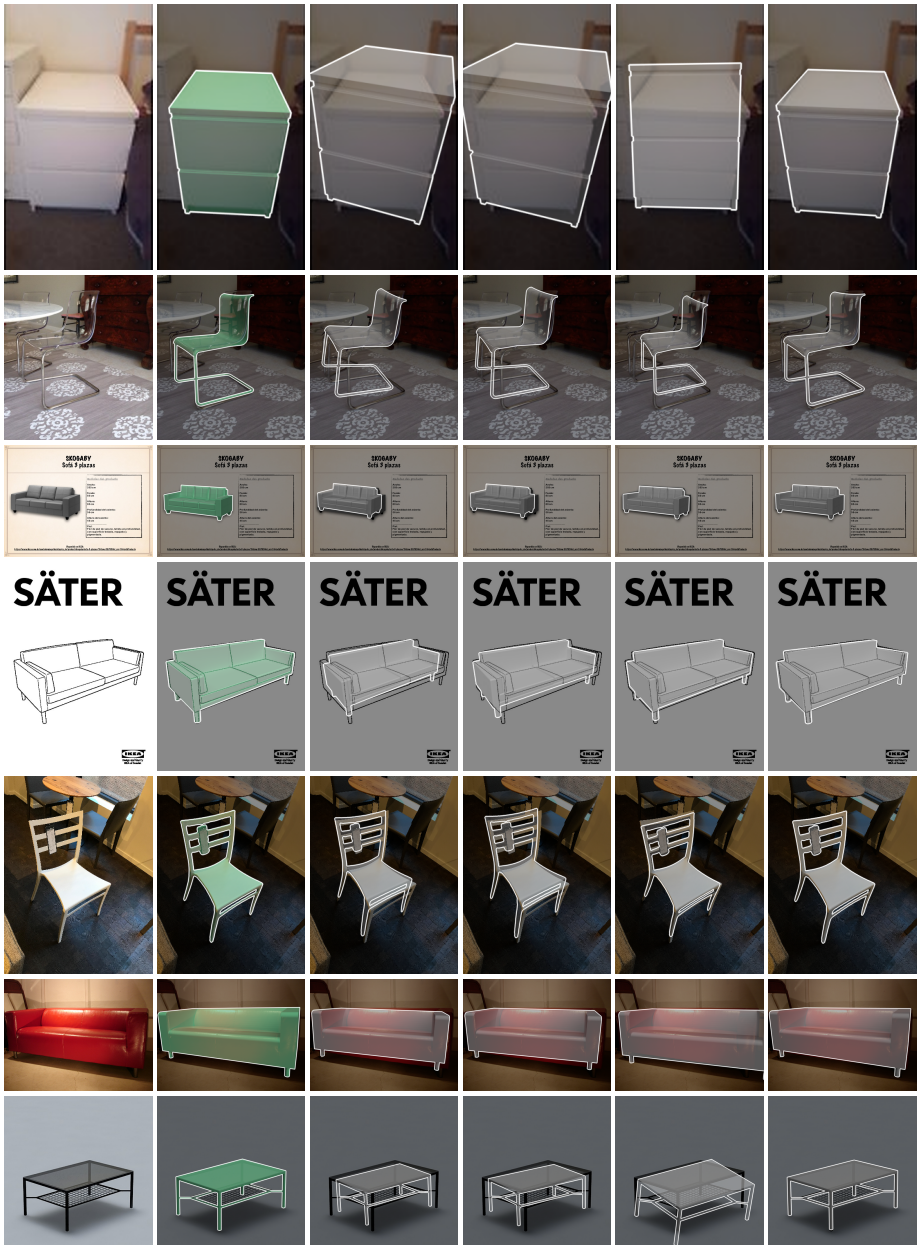


Fig. 5: Additional qualitative 3D pose refinement results for objects of different categories. We project the ground truth 3D model on the image using the predicted 3D pose. Our approach overcomes the limitations of previous methods and predicts fine-grained 3D poses which are in many cases visually indistinguishable from the ground truth. Best viewed in **digital zoom**.

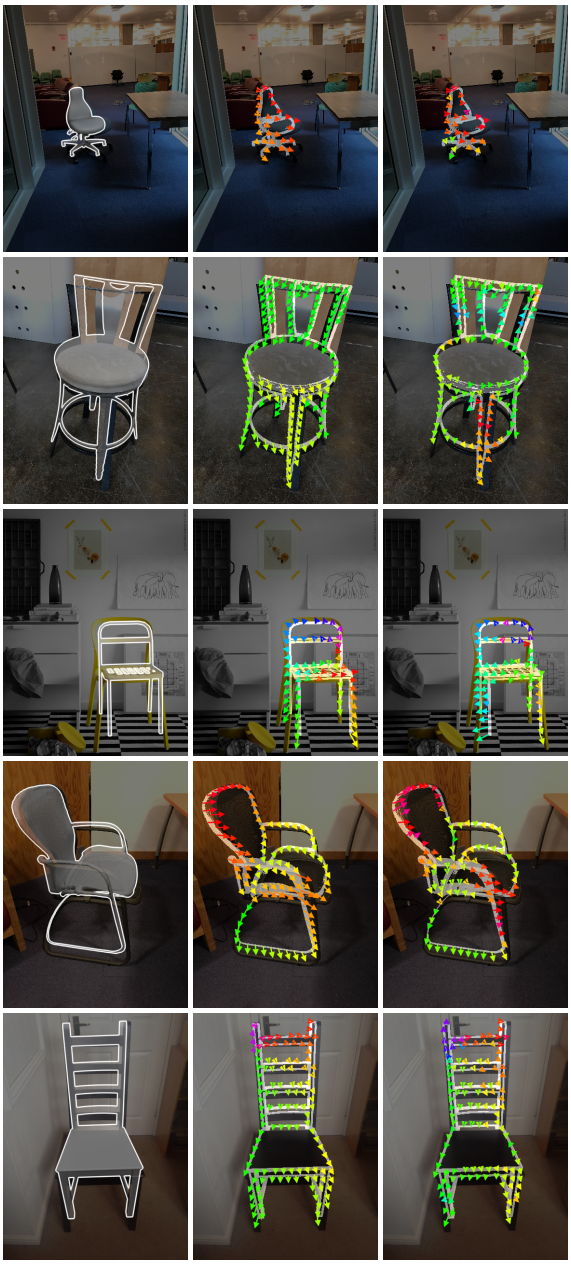


Fig. 6: Additional qualitative examples of our predicted geometric correspondence fields. Our predicted 2D displacement vectors are highly accurate for many different objects and scales. Best viewed in **digital zoom**.

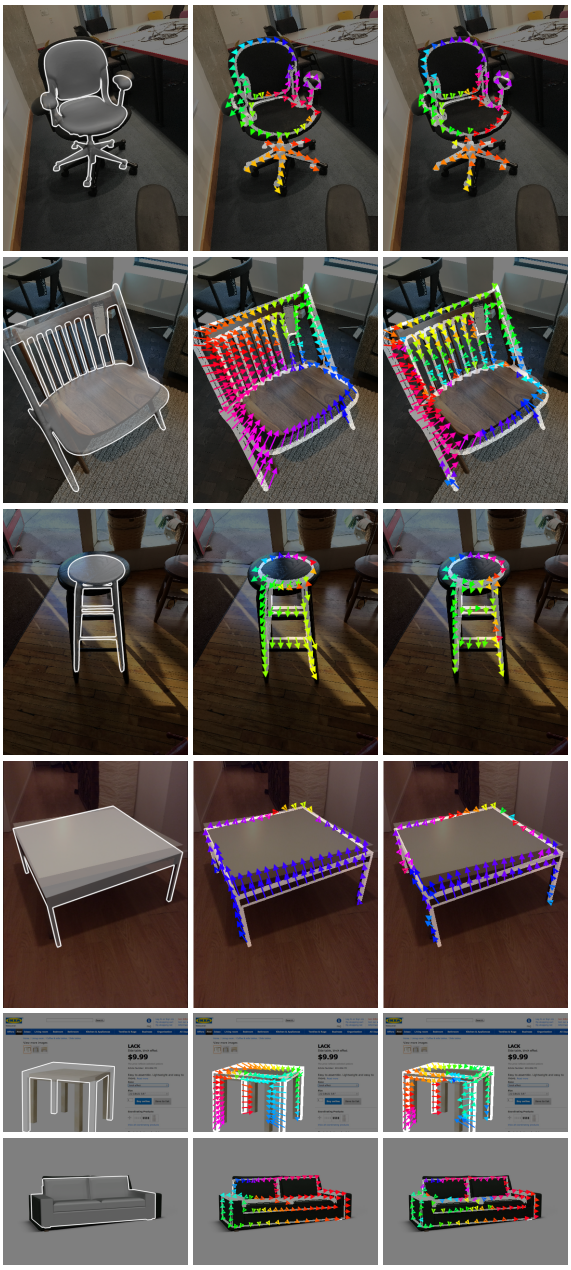


Fig. 7: Additional qualitative examples of our predicted geometric correspondence fields. Our predicted 2D displacement vectors are highly accurate for many different objects and scales. Best viewed in **digital zoom**.

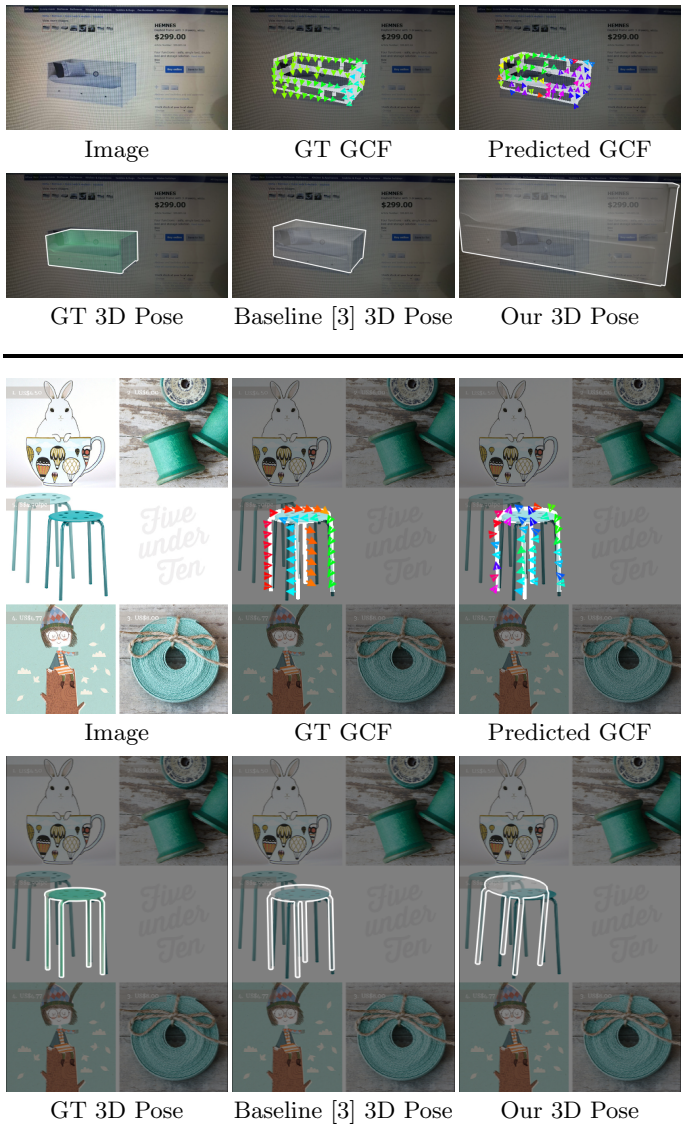


Fig. 8: Failure cases of our approach. In the presence of strong image noise (*top example*), we cannot predict accurate geometric correspondence fields (GCF) and, thus, our refinement fails. Also, if there are duplicate or ambiguous structures in the image our method sometimes predicts wrong correspondences and aligns the 3D model to unintended image parts (*bottom example*). Best viewed in **digital zoom**.

# Bibliography


[1] Calli, B., Singh, A., Walsman, A., Srinivasa, S., Abbeel, P., Dollar, A.: The YCB Object and Model Set: Towards Common Benchmarks for Manipulation Research. In: International Conference on Advanced Robotics. pp. 510–517 (2015)

[2] Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes. In: Conference on Computer Vision and Pattern Recognition. pp. 5828–5839 (2017)

[3] Grabner, A., Roth, P.M., Lepetit, V.: GP$^2$C: Geometric Projection Parameter Consensus for Joint 3D Pose and Focal Length Estimation in the Wild. In: International Conference on Computer Vision. pp. 2222–2231 (2019)

[4] Grabner, A., Roth, P.M., Lepetit, V.: Location Field Descriptors: Single Image 3D Model Retrieval in the Wild. In: International Conference on 3D Vision. pp. 583–593 (2019)

[5] He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: International Conference on Computer Vision. pp. 2980–2988 (2017)

[6] He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)

[7] He, K., Zhang, X., Ren, S., Sun, J.: Identity Mappings in Deep Residual Networks. In: European Conference on Computer Vision. pp. 630–645 (2016)

[8] Hinterstoisser, S., Cagniart, C., Ilic, S., Sturm, P., Navab, N., Fua, P., Lepetit, V.: Gradient Response Maps for Real-Time Detection of Textureless Objects. IEEE Transactions on Pattern Analysis and Machine Intelligence **34**(5), 876–888 (2011)

[9] Hinterstoisser, S., Lepetit, V., Ilic, S., Holzer, S., Bradski, G., Konolige, K., Navab, N.: Model Based Training, Detection and Pose Estimation of Texture-Less 3D Objects in Heavily Cluttered Scenes. In: Asian Conference on Computer Vision. pp. 548–562 (2012)

[10] Hodan, T., Haluza, P., Obdržálek, Š., Matas, J., Lourakis, M., Zabulis, X.: T-LESS: An RGB-D Dataset for 6D Pose Estimation of Texture-Less Objects. In: IEEE Winter Conference on Applications of Computer Vision. pp. 880–888 (2017)

[11] Hodaň, T., Matas, J., Obdržálek, Š.: On Evaluation of 6D Object Pose Estimation. In: European Conference on Computer Vision. pp. 606–619 (2016)

[12] Kato, H., Ushiku, Y., Harada, T.: Neural 3D Mesh Renderer. In: Conference on Computer Vision and Pattern Recognition. pp. 3907–3916 (2018)

[13] Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. arXiv:1412.6980 (2014)

[14] Li, Y., Wang, G., Ji, X., Xiang, Y., Fox, D.: DeepIM: Deep Iterative Matching for 6D Pose Estimation. In: European Conference on Computer Vision. pp. 683–698 (2018)
[15] Loper, M.M., Black, M.J.: OpenDR: An Approximate Differentiable Renderer. In: European Conference on Computer Vision. pp. 154–169 (2014)
[16] Manhardt, F., Kehl, W., Navab, N., Tombari, F.: Deep Model-Based 6D Pose Refinement in RGB. In: European Conference on Computer Vision. pp. 800–815 (2018)
[17] Oberweger, M., Rad, M., Lepetit, V.: Making Deep Heatmaps Robust to Partial Occlusions for 3D Object Pose Estimation. In: European Conference on Computer Vision. pp. 119–134 (2018)
[18] Palazzi, A., Bergamini, L., Calderara, S., Cucchiara, R.: End-to-End 6-DoF Object Pose Estimation through Differentiable Rasterization. In: European Conference on Computer Vision Workshops. pp. 1–14 (2018)
[19] Sun, X., Wu, J., Zhang, X., Zhang, Z., Zhang, C., Xue, T., Tenenbaum, J., Freeman, W.T.: Pix3D: Dataset and Methods for Single-Image 3D Shape Modeling. In: Conference on Computer Vision and Pattern Recognition. pp. 2974–2983 (2018)
[20] Tulsiani, S., Malik, J.: Viewpoints and Keypoints. In: Conference on Computer Vision and Pattern Recognition. pp. 1510–1519 (2015)
[21] Wang, H., Sridhar, S., Huang, J., Valentin, J., Song, S., Guibas, L.: Normalized Object Coordinate Space for Category-Level 6D Object Pose and Size Estimation. In: Conference on Computer Vision and Pattern Recognition. pp. 2642–2651 (2019)
[22] Wang, Y., Tan, X., Yang, Y., Liu, X., Ding, E., Zhou, F., Davis, L.S.: 3D Pose Estimation for Fine-Grained Object Categories. In: European Conference on Computer Vision Workshops (2018)
[23] Xiang, Y., Kim, W., Chen, W., Ji, J., Choy, C., Su, H., Mottaghi, R., Guibas, L., Savarese, S.: ObjectNet3D: A Large Scale Database for 3D Object Recognition. In: European Conference on Computer Vision. pp. 160–176 (2016)
[24] Xiang, Y., Mottaghi, R., Savarese, S.: Beyond Pascal: A Benchmark for 3D Object Detection in the Wild. In: IEEE Winter Conference on Applications of Computer Vision. pp. 75–82 (2014)
[25] Zakharov, S., Shugurov, I., Ilic, S.: DPOD: Dense 6D Pose Object Detector in RGB Images. In: International Conference on Computer Vision. pp. 1941–1950 (2019)