FS-Net Supplementary Material

Wei Chen et. al.
School of Computer Science, University of Birmingham
United Kindom
wxc795@cs.bham.ac.uk

A. Overview

This document provides more details about our FS-Net. Section B describes the details of the 3D deformation mechanism and deformed examples. Section C provides more quantitative results of the FS-Net on NOCS-REAL [3] dataset and comparison with state-of-the-art method.

B. 3D Deformation Mechanism

As stated in Section 3.5 of the paper, the 3D deformation mechanism is box-cage based and the deformations are applied in a canonical space. In the canonical coordinate system, every box edge is parallel to an axis (shown in Figure A). This property makes the 3D deformation calculation easier. For example, when we need to elongate/shrink the mug along Y axis by n times. We enlarge the distance between surface $S_{1,2,3,4}$ and surface $S_{5,6,7,8}$ by n times. Since these two surfaces are parallel to the XZ-plane, the x and z coordinates are unchanged. Then points coordinates are changed from $[\mathbf{x}, \mathbf{y}, \mathbf{z}]$ to $[\mathbf{x}, n\mathbf{y}, \mathbf{z}]$. The calculations are similar when we need to elongate/shrink the mug along X or Z axis by n times:

$$[\mathbf{x}, n\mathbf{y}, \mathbf{z}] = \mathbb{F}_x([\mathbf{x}, \mathbf{y}, \mathbf{z}]), \quad (1)$$
$$[n\mathbf{x}, \mathbf{y}, \mathbf{z}] = \mathbb{F}_y([\mathbf{x}, \mathbf{y}, \mathbf{z}]), \quad (2)$$
$$[\mathbf{x}, \mathbf{y}, n\mathbf{z}] = \mathbb{F}_z([\mathbf{x}, \mathbf{y}, \mathbf{z}]), \quad (3)$$

where $\mathbb{F}_{x,y,z}$ is the elongate/shrink operation along corresponding axis.

Further, if the object is the mug or bowl, we may need to change the top or bottom size to generate new shapes (shown in Figure B). In this case, assuming we enlarge the bottom along X axis by n times, then from bottom to top, the coordinates are changed as:

$$\mathbf{x}_{new} = (1 + (n-1)\frac{l}{L})\mathbf{x}, \quad (4)$$

where l is the distance from a point to the top surface, i.e. $S_{1,2,3,4}$ in Figure A. L is the height of the object. Please note, all the edges are keep straight while deformation.

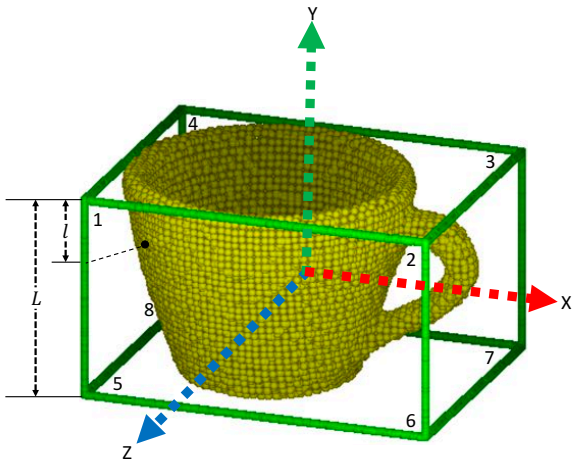

Figure A. **3D object model**. We assume that the center of 3D bounding box is the origin point of the coordinate. The surface is represented by its four corners. For example, the top surface is represented by $S_{1,2,3,4}$.

C. Experimental Results

C.1. Detailed Results

We report the specific category pose estimation results under different metrics in Table A. We also provide the rotation recovered by one/two vectors in Figure C. We can see that the bounding boxes are well aligned in the recovered vector direction.

Table A. **Category-Level results.** Object-wise experiments with different metrics.

Category	IoU_{75}	5°5 **cm**	10°5 **cm**	10°10 **cm**
Bottle	0.4710	0.4219	0.8134	0.8755
Bowl	0.9810	0.5916	0.9793	0.9793
Camera	0.5882	0.0176	0.1457	0.1480
Can	0.6334	0.4055	0.7820	0.8141
Laptop	0.3805	0.1659	0.5570	0.6859
Mug	0.7534	0.0874	0.3698	0.3706
Average	0.6345	0.2816	0.6078	0.6455

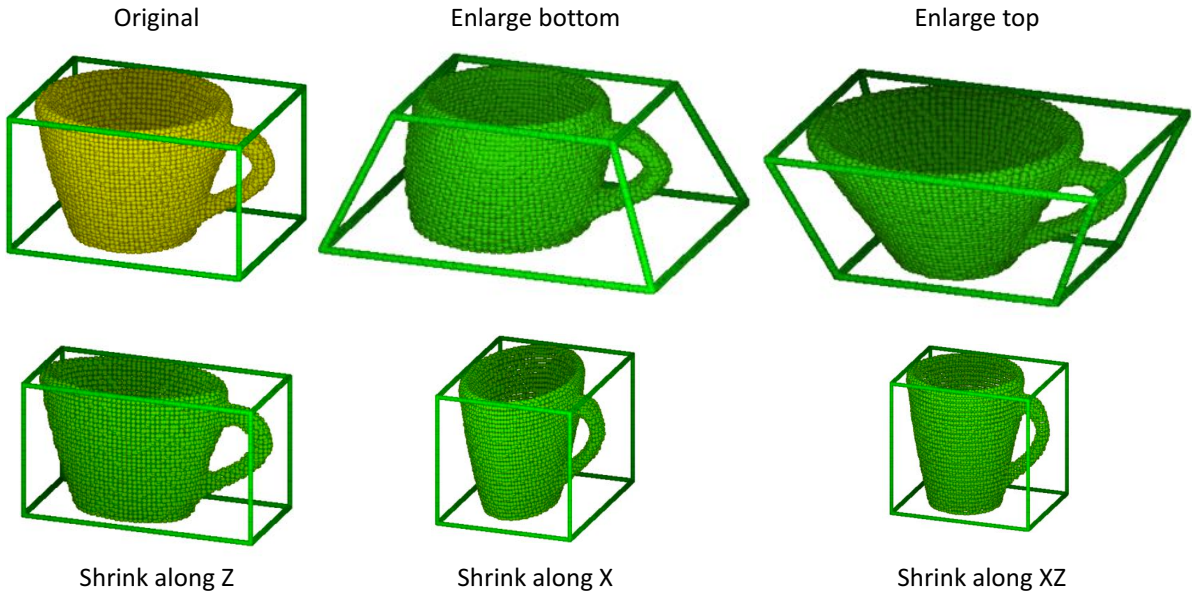

Figure B. **Examples of different deformations.** We assume that the XYZ axis are the same as Figure A. The upper right corner is the original point cloud with corresponding box-cage. The rest are the deformed box-cages and point clouds. The deformation operations are described on the top or bottom of the pictures.

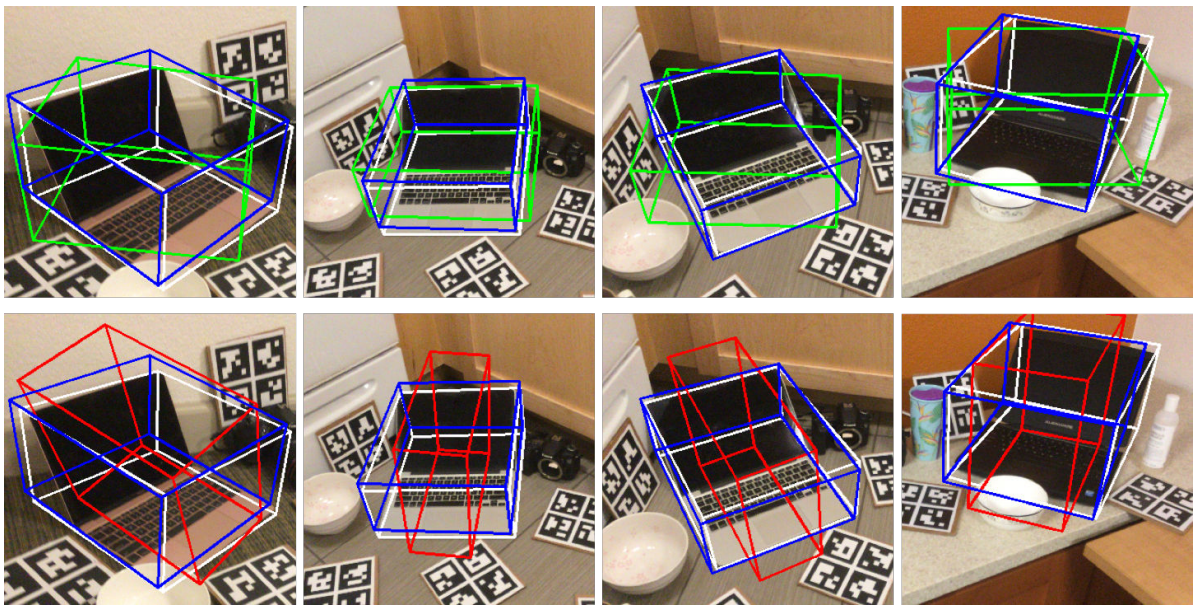

Figure C. **Rotation recovered by different vectors.** The white boxes are the ground truth. Blue boxes are the rotation recovered by two estimated vectors. The green and red boxes are the rotation recovered by estimated green vector and estimated red vector (see Figure 4 in the paper), respectively. For better illustration, we use ground truth object size to calculate the final 3D bounding box.

C.2. Comparison with State-Of-The-Art

We compare FS-Net with the state-of-the-art method Shape-Prior [1], which utilized point cloud for category-level 6D object pose estimation. Shape-Prior [1] estimated the object size and 6D pose from dense-fusion feature [2], while we estimate the pose from point cloud feature. Figure D shows that our FS-Net is robust to color and shape variation, and can handle some failure cases of Shape-Prior. For Shape-Prior, we use the predicted results provided on their website: https://github.com/mentian/object-deformnet.

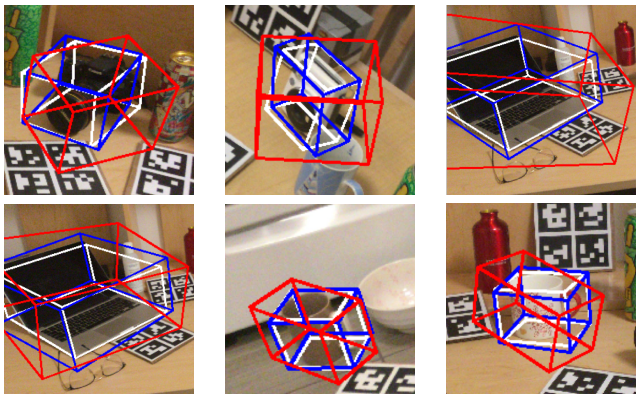

Figure D. **Qualitative comparison with Shape-Prior**. The white boxes are the ground truth. Blue boxes are our results. Red boxes are the poses predicted by Shape-Prior [1]

References

[1] Meng Tian, Marcelo H Ang Jr, and Gim Hee Lee. Shape prior deformation for categorical 6d object pose and size estimation. *arXiv preprint arXiv:2007.08454*, 2020. 2, 3

[2] Chen Wang, Danfei Xu, Yuke Zhu, Roberto Martín-Martín, Cewu Lu, Li Fei-Fei, and Silvio Savarese. Densefusion: 6d object pose estimation by iterative dense fusion. 2019. 2

[3] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2642–2651, 2019. 1