You Only Look at One: Category-Level Object Representations for Pose Estimation From a Single Example (Supplementary Material)

Walter Goodwin, Ioannis Havoutis, Ingmar Posner
Oxford Robotics Institute
University of Oxford
firstname@robots.ox.ac.uk

7 Appendix

In this supplementary material, we cover certain aspects of our implementation in more detail. We present plots visualising the first descriptors extracted for various object categories. We describe the extension of our method to fitting 9-D transforms (rather than the 7-D similarity transform) between object instances. Finally, we give details of our experiments on a Panda robot arm with a RealSense camera, which demonstrated that the method could run at 15fps while providing pose estimates for previously unseen objects.

7.1 Implementation details

7.1.1 Descriptor dimensionality reduction

In Section 3.3, we describe our approach to using principal components analysis (PCA) to reduce descriptor dimensionality. In Section 4.1.1, we show that empirically, such dimensionality reduction actually improves performance up to a point (an over 10-fold decrease in dimensionality from 384-D raw ViT features to 32-D descriptors improves performance considerably for pose estimation over the CO3D dataset). In this appendix, we provide illustrative examples of the descriptors resulting from PCA. Fig. A.1 shows the first three principal components for several frames from each of 20 CO3D categories. The components onto which these frames' descriptors have been projected have been, in each case, calculated on a distinct reference sequence from the same category. Descriptor plots are masked by a threshold on ViT saliency for clarity. Fig. A.2 also shows the projection of sequence descriptors onto the first three principal components, but shows the result of using principal components calculated on descriptors from a single reference sequence (the left-most sequence for each category) on five other distinct sequences drawn from the same category. The cross-instance generalisation of the DINO ViT features, and of descriptors derived from these, can be seen in the consistent colouring (in the space of the first three principal components) of key object parts across diverse instances.

7.1.2 Fast depth completion

In both our static datasets (Section 4.1) and robot manipulator settings Section 4.3, depth images are incomplete. For fast and scene-agnostic depth completion, we take inspiration from [38], which proposes a simple sequence of kernel-based filters for fast depth completion. We use a similar sequence of kernels to process our depth images. Of particular importance is that there are no holes in the depth image, as these can cause numerical instabilities when back-projected points are used to estimate rigid-body transforms in our method. As a final stage, we fill any remaining holes with a dilation operation with a very large kernel size, and set any remaining empty values to the mean

6th Conference on Robot Learning (CoRL 2022), Auckland, New Zealand.

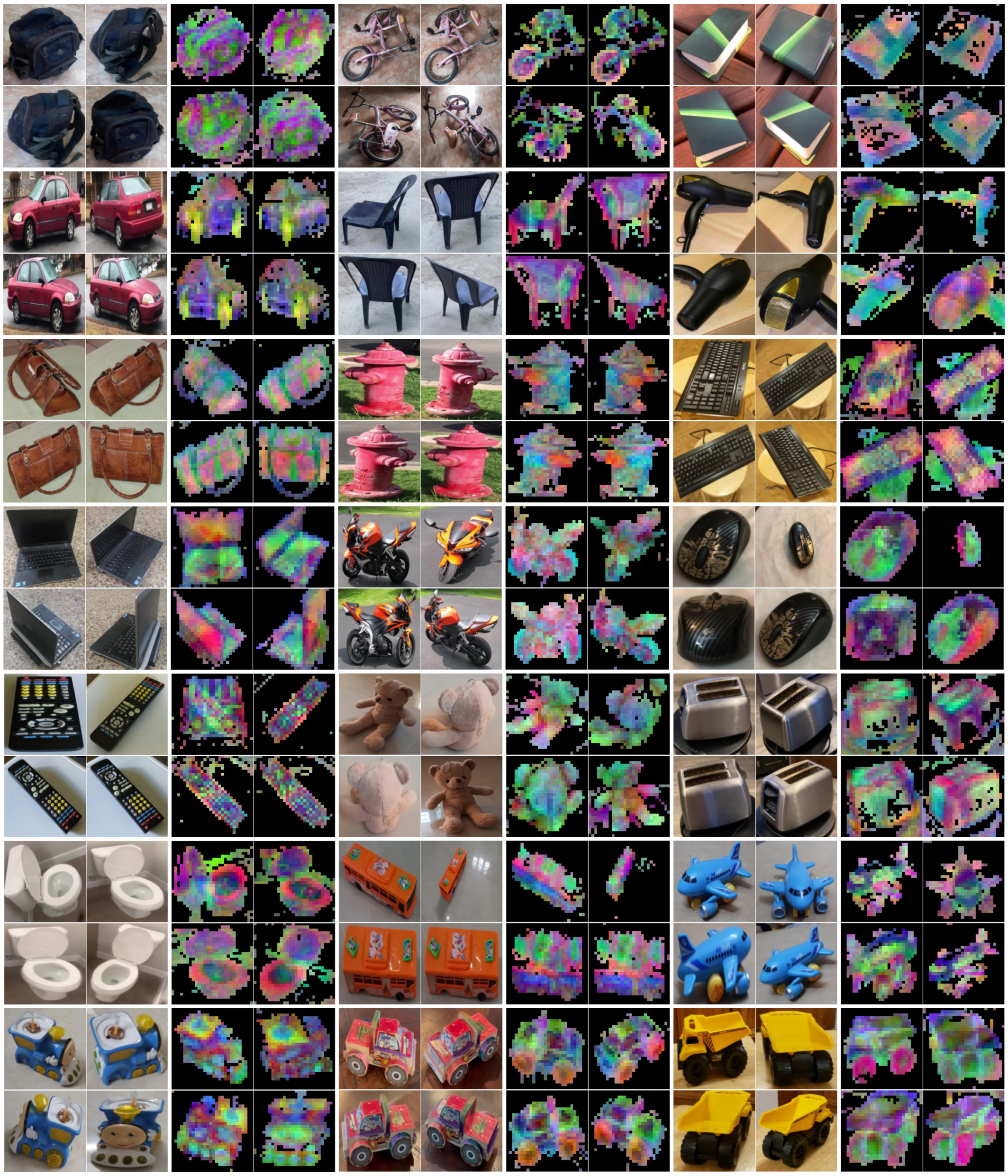

Figure A.1: Examples from the CO3D dataset, showing the first three principal components for target sequences from each of 20 considered categories. For each example, the principal components have been calculated on a *different* reference sequence. PCA feature maps are masked by a threshold on saliency computed from the ViT attention maps.

depth over the image. Depth processing takes 4.1 ms. An example result on a sequence from the RealSense camera can be seen in Fig. A.4.

7.1.3 TEASER++ baseline

For the TEASER++ [13] baseline experiments reported in Section 4, we use the official implementation from `https://github.com/MIT-SPARK/TEASER-plusplus` with all parameters at defaults.

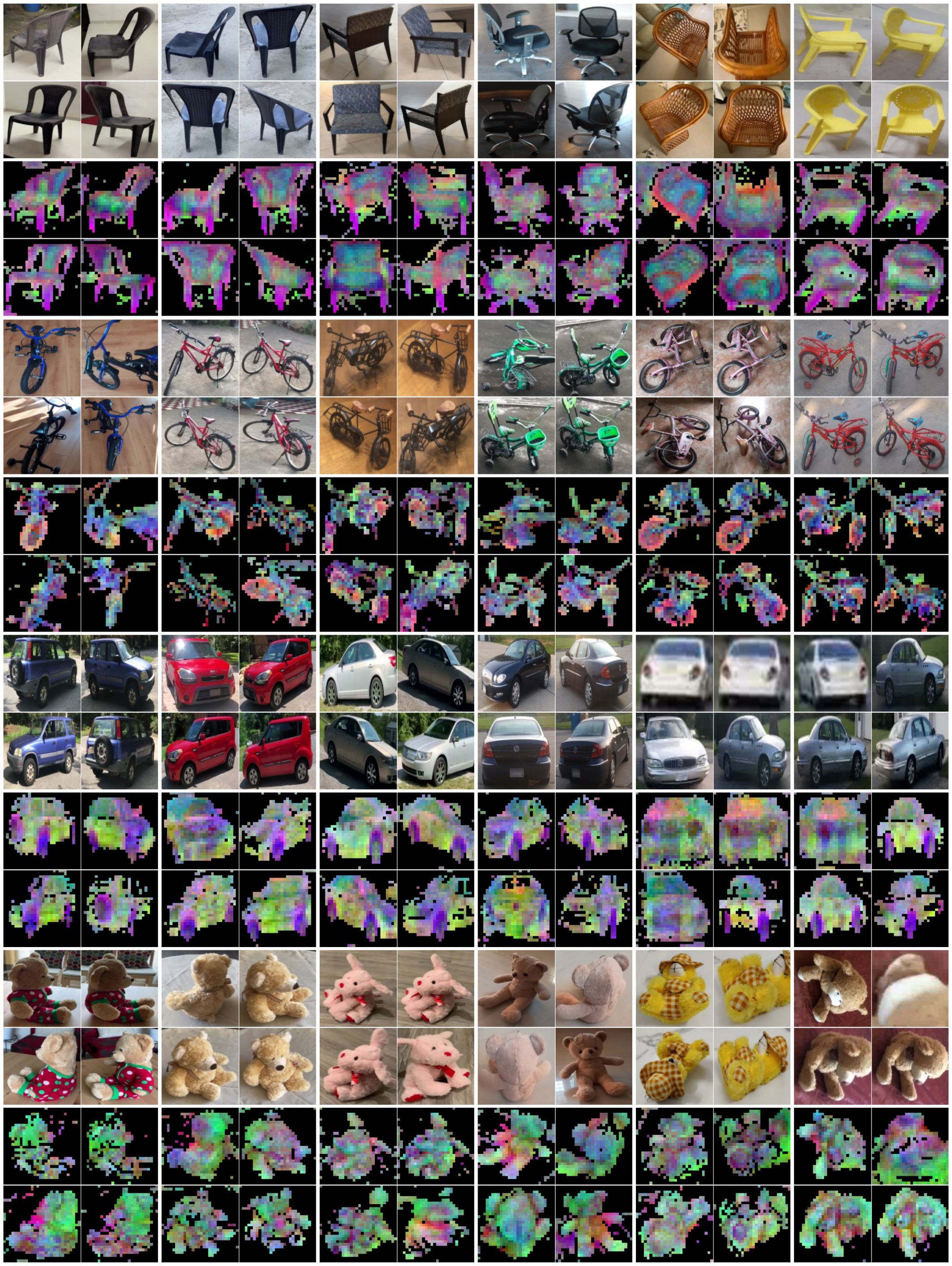

Figure A.2: Visualising the first three principal components calculated from the features of a single reference sequence (leftmost images for each category) on five further sequences from the same category. Consistent colouring of corresponding parts can be seen. This property of invariance to varying instances within a category is exploited by our method for robust cross-instance correspondence estimation, used in our pose estimation method.

7.2 Additional experiments

7.2.1 3D cyclical distance for correspondence matching

In Section 3.2, we describe the cyclical distance metric for selecting strong correspondences from descriptor similarity matrices, which is introduced in [17]. This cyclical distance metric for selecting correspondences can be thought of as a spatial prior: correspondences that form a cycle from reference to target back to reference image patch (by nearest neighbours) which arrives at a *close* location in patch-space to the original location are more likely to be meaningful, because a close patch location could very likely contain the same part of an object as the original patch. Patch distances, though, are a 2D measure, while in this work we have access to depth maps. We experiment with a 3D version of the cyclical distance measure in which correspondences are ranked based on the distance *in actual 3D space* between the original patch and the final patch. We found a small improvement from this (+0.6% on Acc30 across CO3D in a 10 reference, 10 target image setting), but we do not believe this result to be statistically significant.

7.2.2 Replacing 7D similarity transform with 9D affine transform

As noted in Section 5, the use of a 7D similarity transform (1D isotropic scale, and 6D pose) to model the relationship between correspondences found between two object instances from a category is almost certainly over-prescriptive. In this section, we how a 9D affine transform can be used instead with very few changes to the method. This is a more general model, and has promise to be more suitable for certain categories. In 9D setting, rather than a single isotropic scaling parameter λ, we seek a vector $\mathbf{s} \in \mathbb{R}^3$ which models separate scalings for each dimension.

Otherwise following Section 3.4.2, we seek a rotation $\hat{\mathbf{R}}$, translation $\hat{\mathbf{t}}$ and scaling $\hat{\mathrm{s}}$ that, for K corresponding 3D points $\{\mathbf{u}_k, \mathbf{v}_k\}_{k=1:K}$ satisfy the following:

$$(\mathbf{s}, \hat{\mathbf{R}}, \hat{\mathbf{t}}) = \underset{(\mathbf{s},\mathbf{R},\mathbf{t})}{\operatorname{argmin}} \sum_{k=1}^{K} \mathbf{v}_k - (\operatorname{diag}(\mathbf{s})\mathbf{R}\mathbf{u}_k + \mathbf{t}) \tag{3}$$

In the 7D case (λ scaling rather than $\mathbf{s}$), Umeyama's method gives a fast and closed form solution that scales well with the number of points as it is based on the singular value decomposition of the covariance matrix of the two correspondence matrices [34].

A similar method has been proposed in [39] to calculate the 9D transform described above. We refer the reader to that paper for further details. We implemented this method to evaluate whether finding a 9D transform might lead to better pose estimates by allowing for a more accurate category-level model of spatial correspondence. As with Umeyama's method, the most time-consuming component of this algorithm is computing the singular value decomposition of the covariance matrix, and we are still able to run 1,000 RANSAC trials in a few milliseconds with this method and CUDA acceleration.

While we did not find this method to improve pose estimation performance on aggregate over the categories used in this work (a -0.5% drop in Acc30 across CO3D in a 10 reference, 10 target image setting), this may be because the categories considered tend to have objects whose shape relationship is captured acceptable by a single scaling factor.

Finally, while this work's evaluation is on pose estimation, the underlying methodology and finding of robust category-level correspondences could be applied to the challenging setting of category-level grasping. In this context, there is good evidence that the ability to infer non-uniform scaling is important for transferring grasps between items within a category. Recent works have motivated the use of the 9D transform described here in representing a category-level canonical space for objects [40, 41], extending Normalised Object Canonical Space (NOCS) [6] to Non-Uniform NOCS (NUNOCS).

7.2.3 RANSAC inlier threshold

We use RANSAC to find an optimal transform as described in Section 3.4.2. For the results in Table 1, we used an inlier threshold of 0.2 for all categories, in order to match the conditions in [17]. All objects in the dataset are at the same scale, measuring about 6 units long on their longest side. This threshold is thus intuitively quite restrictive: a point on the reference object under the estimated transform must not be more than about 3.5% of the object's overall size away from the corresponding point on the target object, or else it will not be counted as an inlier. We experimented with various inlier thresholds, and found that setting this parameter to be higher, based on the above intuition, has a positive effect. The results in Table 2, for instance, use a threshold of 0.5 (c.f. 72.1% Acc@30°for the 30-vs-30 image results, vs 69.8% Acc@30°in Table 1).

7.3 Further results

7.3.1 Translation estimation performance

Table A.1 reports results for estimating the translation component of 6D pose.

	All Categories				Per Category (Acc@0.5), %					
Method	Med. Err (↓)	Acc@1.0 (↑)	Acc@0.5	Acc@0.2	B'pack	Car	Chair	Keyboard	Laptop	M'cycle
Ours-U	0.586	76.1	46.9	19.9	25	**97**	27	80	96	85
Ours-UC	0.548	78.3	49.4	19.1	37	84	36	81	99	84
Ours-UCD	0.498	81.6	52.6	20.7	**39**	91	35	**86**	99	85
Ours-UCD+	**0.489**	**81.8**	**53.9**	**22.3**	36	90	**40**	82	**100**	**90**

Table A.1: Pose estimation accuracy (translation). We report Acc@δ, with $\delta \in \{1.0, 0.5, 0.2\}$, giving the percentage of estimates that fall within a certain maximum threshold on Euclidean distance from ground truth, and the median error 'Med. Err' (per category, then averaged). These numbers are absolute, as every category in the CO3D dataset is at the same scale, with the longest side scaled to 2π. Thus, a translation error of 1.0 is approximately 1/6 the longest side of the object. Suffixes on our method ablations: **C**: consensus by largest inlier group **U**: Umeyama's method; rigid body solution using best-view correspondences. **D**: descriptor dimensionality reduction (to 32 components). Methods use 10 reference and target views. **+** indicates 30 views used.

7.3.2 Visual examples of inlier correspondences following RANSAC

Fig. A.3 visualises a key part of the pose estimation process - the inlier set following RANSAC - on an example reference-target object pair from each of the 20 CO3D categories considered.

7.4 Robot experiments

In Section 4.3, we describe the setting for deploying our pose estimation method on a Panda robot arm with a wrist-mounted RealSense camera. In this appendix, we expand on several fully automated pre-processing steps that are important implementation details for this real-world setting. For videos of real-time one-shot pose estimation, we refer the reader's attention to the accompanying video.

7.4.1 Attention maps for object detection

In our experiments on the CO3D dataset, object detection is an upstream process, and we operate on images closely cropped to the objects of interest. We show that we are able to achieve the same accurate object detection and cropping in a fully autonomous setting (running pose estimation on a robot arm at 15Hz) through running ViT inference on each image twice. This process is shown in Fig. A.4.

In the first pass, the whole field-of-view image from the wrist-mounted RealSense camera is processed. A threshold (0.05) on the attention map from the first pass is used to produce a binary segmentation mask, from which the largest connected component is taken to be the object of interest. The resulting bounding box is up-sampled to the original image size, and expanded by 10% so

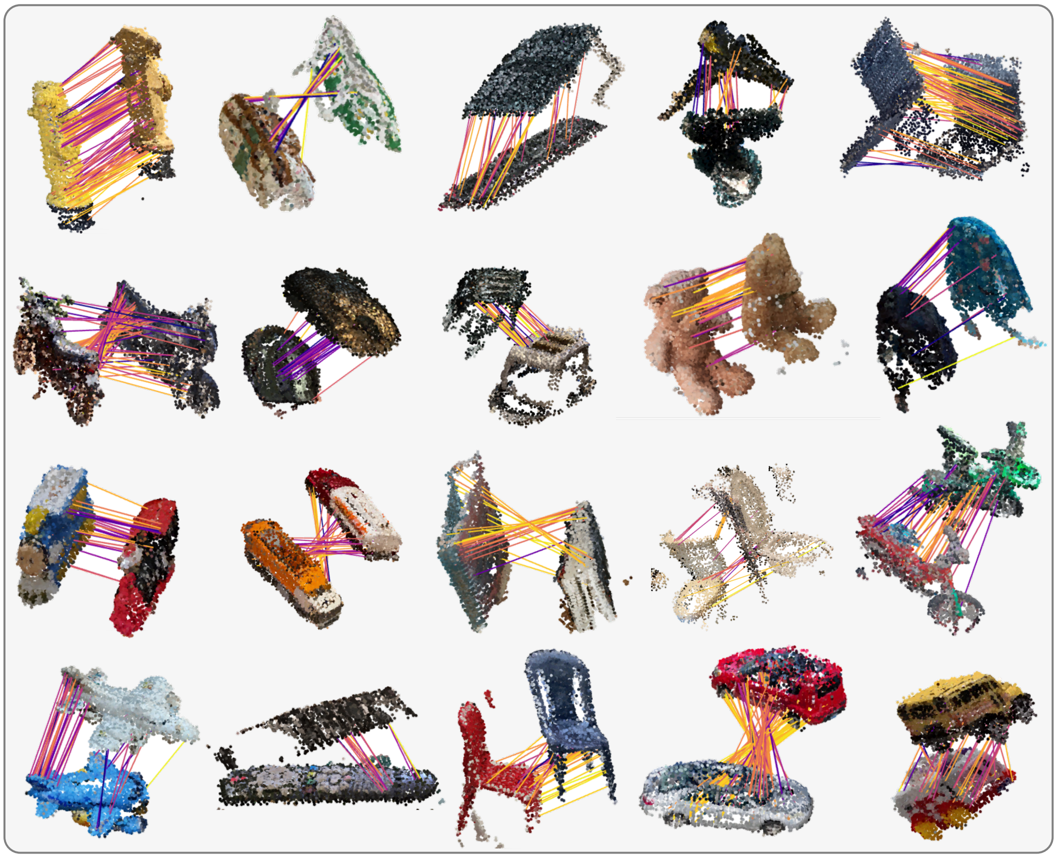

Figure A.3: An example reference-target object pair from each of 20 CO3D categories used for pose estimation. Point clouds are created from the images and depth maps as described in Section 7.4.2. Point clouds are rendered with respect to the camera viewpoint of the first frame in their respective sequences, with an added fix offset to avoid overlap. Lines between the objects show the *inlier set* of correspondences following the RANSAC rigid body solution (Section 3.4.2). As in Fig. 2, line colour shows correspondence similarity (🟨 =higher, 🟪 =lower). Categories, from left to right, top to bottom: *hydrant, handbag, keyboard, hairdryer, laptop, motorbike, mouse, toaster, teddybear, backpack, toy train, toy bust, book, toilet, bicycle, toy plane, remote, chair, car, toy truck*.

as to be sure to capture the whole object. This box is used to produce a closely cropped image whose ViT features are used for correspondences and pose estimation.

7.4.2 Bounding boxes for visualising pose estimates

Although we describe our method as being *one-shot* pose estimation because of its use of a reference sequence to describe a target object category, we design our system to require no manual labelling or manual processing of this reference object, such that the whole method could in practice be deployed as fully autonomous.

Our method estimates a 7 DoF rigid body transforming $(\hat{\lambda}, \hat{\mathbf{R}}, \hat{\mathbf{t}})$ between a category's reference object, and a target object, as described in Section 3.4.2, where 6D pose is given by SO(3) rotation $\hat{\mathbf{R}}$ and translation $\hat{\mathbf{t}}$. While this captures object pose from a mathematical point of view, it does not immediately offer a way of *visualising* the pose estimates. To facilitate this in a fully automated pipeline, we fit an oriented bounding box to the *reference* object for a category, and transform this by the estimated 7 DoF rigid body transform to visualise pose estimates for the target objects.

The process of fitting an oriented bounding box to a captured reference object is shown in Fig. A.4. The first stages of close cropping are described in Section 7.4.1. Subsequently, masking and point cloud outlier removal steps produce a filtered point cloud, to which an oriented bounding box is

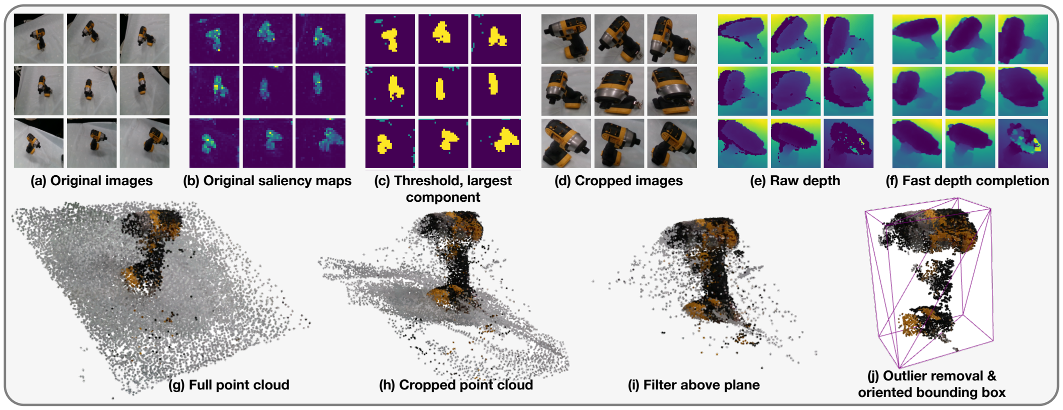
(a) Original images
(b) Original saliency maps
(c) Threshold, largest component
(d) Cropped images
(e) Raw depth
(f) Fast depth completion
(g) Full point cloud
(h) Cropped point cloud
(i) Filter above plane
(j) Outlier removal & oriented bounding box

Figure A.4: Point cloud processing for fitting oriented bounding boxes to reference objects. **(a)** Original images from RealSense camera, 720x1280 resolution centre-cropped to 720x900 then resized to 224x224 for ViT processing. **(b)** Saliency aggregated from the ViT attention maps. **(c)** A threshold of > 0.05 on saliency produces a binary foreground mask - here, the largest connected component is shown in yellow (other 'foreground' in blue). **(d)** Images cropped to the box described by the largest connected component. **(e)** Raw depth from the RealSense D435i, following same crop. **(f)** Depth maps following fast inpainting process. **(g)** Point cloud from backprojecting original images. **(h)** Point cloud using saliency-based cropping. **(i)** Point cloud following plane detection. **(j)** Final point cloud following outlier removal with a KNN criteria, and oriented bounding box fitting.

fitted based on the PCA of the convex hull of the filtered point cloud. This is a fast approximation to the minimum-volume bounding box, which tends to produce intuitive, easily interpreted bounding boxes around object volumes [42]. We use Python bindings to the Open3D library to perform this final stage [43].

For outlier removal, we use a fast K-nearest neighbours based approach using Pytorch3D [44]. For an inlier point, we require that its 10 nearest neighbours be within $5 \times 10^{-5}\,\mathrm{m}$.

References

[1] K. Wada, S. James, and A. J. Davison. ReorientBot: Learning Object Reorientation for Specific-Posed Placement. In *ICRA*, 2022.

[2] E. Sucar, K. Wada, and A. Davison. NodeSLAM: Neural Object Descriptors for Multi-View Shape Reconstruction. *Proceedings - 2020 International Conference on 3D Vision, 3DV 2020*, pages 949–958, 2020.

[3] W.-C. Ma, A. J. Yang, S. Wang, R. Urtasun, and A. Torralba. Virtual Correspondence: Humans as a Cue for Extreme-View Geometry. In *CVPR*, 2022.

[4] C. Wang, D. Xu, Y. Zhu, R. Martin-Martin, C. Lu, L. Fei-Fei, and S. Savarese. DenseFusion: 6D object pose estimation by iterative dense fusion. In *CVPR*, 2019.

[5] Y. He, W. Sun, H. Huang, J. Liu, H. Fan, and J. Sun. PVN3D: A deep point-wise 3D keypoints voting network for 6DoF pose estimation. In *CVPR*, 2020.

[6] H. Wang, S. Sridhar, J. Huang, J. Valentin, S. Song, and L. Guibas. Normalized Object Coordinate Space for Category-Level 6D Object Pose and Size Estimation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019-June, 2019.

[7] C. Wang, R. Martin-Martin, D. Xu, J. Lv, C. Lu, L. Fei-Fei, S. Savarese, and Y. Zhu. 6-PACK: Category-level 6D Pose Tracker with Anchor-Based Keypoints. In *ICRA*, 2020.

[8] Y. Lin, J. Tremblay, S. Tyree, P. A. Vela, and S. Birchfield. Single-stage Keypoint-based Category-level Object Pose Estimation from an RGB Image. 2021. URL `http://arxiv.org/abs/2109.06161`.

[9] P. R. Florence, L. Manuelli, and R. Tedrake. Dense Object Nets: Learning Dense Visual Object Descriptors By and For Robotic Manipulation. In *Conference on Robotic Learning*, 2018.

[10] L. Manuelli, W. Gao, P. Florence, and R. Tedrake. kPAM: KeyPoint Affordances for Category-Level Robotic Manipulation. 2019.

[11] A. Simeonov, Y. Du, A. Tagliasacchi, J. B. Tenenbaum, A. Rodriguez, P. Agrawal, and V. Sitzmann. Neural Descriptor Fields: SE(3)-Equivariant Object Representations for Manipulation. 2021.

[12] A. Ahmadyan, L. Zhang, A. Ablavatski, J. Wei, and M. Grundmann. Objectron: A Large Scale Dataset of Object-Centric Videos in theWild with Pose Annotations. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 2021.

[13] H. Yang, J. Shi, and L. Carlone. Teaser: Fast and certifiable point cloud registration. *IEEE Transactions on Robotics*, 37, 2021.

[14] B. Wen and K. Bekris. BundleTrack: 6D Pose Tracking for Novel Objects without Instance or Category-Level 3D Models. In *IEEE International Conference on Intelligent Robots and Systems*, 2021.

[15] I. Shugurov, F. Li, B. Busam, and S. Ilic. OSOP: A Multi-Stage One Shot Object Pose Estimation Framework. *arXiv preprint*, 2022.

[16] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. Emerging Properties in Self-Supervised Vision Transformers. In *ICCV*, apr 2021.

[17] W. Goodwin, S. Vaze, I. Havoutis, and I. Posner. Zero-Shot Category-Level Object Pose Estimation. *arXiv preprint*, 2022.

[18] X. Chen, Z. Dong, J. Song, A. Geiger, and O. Hilliges. Category Level Object Pose Estimation via Neural Analysis-by-Synthesis. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 12371 LNCS. Springer Science and Business Media Deutschland GmbH, 2020.

[19] K. Chen and Q. Dou. SGPA: Structure-Guided Prior Adaptation for Category-Level 6D Object Pose Estimation. In *ICCV*, 2021.

[20] A. Wang, A. Kortylewski, and A. Yuille. NeMo: Neural Mesh Models of Contrastive Features for Robust 3D Pose Estimation. In *ICLR*, 2021.

[21] M. Tian, M. H. Ang, and G. H. Lee. Shape Prior Deformation for Categorical 6D Object Pose and Size Estimation. In *ECCV*, 2020.

[22] D. Chen, J. Li, Z. Wang, and K. Xu. Learning canonical shape space for category-level 6D object pose and size estimation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 2020.

[23] J. Shi, H. Yang, and L. Carlone. Optimal Pose and Shape Estimation for Category-level 3D Object Perception. In *Robotics: Science and Systems XVII*, 2021.

[24] Y. Xiao, X. Qiu, P. A. Langlois, M. Aubry, and R. Marlet. Pose from Shape: Deep pose estimation for arbitrary 3D objects. In *30th British Machine Vision Conference 2019, BMVC 2019*, 2019.

[25] C. Sahin and T. K. Kim. Category-level 6D object pose recovery in depth images. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2019.

[26] A. Grabner, P. M. Roth, and V. Lepetit. 3D Pose Estimation and 3D Model Retrieval for Objects in the Wild. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018.

[27] S. Gupta, P. Arbeláez, R. Girshick, and J. Malik. Inferring 3D Object Pose in RGB-D Images. 2015.

[28] Y. Xiao, Y. Du, and R. Marlet. PoseContrast: Class-Agnostic Object Viewpoint Estimation in the Wild with Pose-Aware Contrastive Learning. In *2021 International Conference on 3D Vision (3DV)*. IEEE, 2021.

[29] W. Chen, X. Jia, H. J. Chang, J. Duan, L. Shen, and A. Leonardis. FS-Net: Fast Shape-based Network for Category-Level 6D Object Pose Estimation with Decoupled Rotation Mechanism. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2021.

[30] J. Sun, Z. Wang, S. Zhang, X. He, H. Zhao, G. Zhang, and X. Zhou. OnePose: One-Shot Object Pose Estimation without CAD Models. *arXiv preprint*, 2022.

[31] S. Lu, R. Wang, Y. Miao, C. Mitash, and K. Bekris. Online Object Model Reconstruction and Reuse for Lifelong Improvement of Robot Manipulation. In *ICRA*, 2022.

[32] Y. He, Y. Wang, H. Fan, J. Sun, and Q. Chen. FS6D: Few-Shot 6D Pose Estimation of Novel Objects. In *CVPR*, 2022.

[33] K. Aberman, J. Liao, M. Shi, D. Lischinski, B. Chen, and D. Cohen-Or. Neural best-buddies: Sparse cross-domain correspondence. *ACM Transactions on Graphics*, 2018.

[34] S. Umeyama. Least-Squares Estimation of Transformation Parameters Between Two Point Patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1991.

[35] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox. PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes. In *Robotics: Science and Systems XIV*, 2018.

[36] J. Reizenstein, R. Shapovalov, P. Henzler, L. Sbordone, P. Labatut, and D. Novotny. Common Objects in 3D: Large-Scale Learning and Evaluation of Real-life 3D Category Reconstruction. In *ICCV*, 2021.

[37] J. L. Schonberger and J.-M. Frahm. Structure-from-Motion Revisited. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[38] J. Ku, A. Harakeh, and S. L. Waslander. In defense of classical image processing: Fast depth completion on the CPU. In *Proceedings - 2018 15th Conference on Computer and Robot Vision, CRV 2018*, 2018.

[39] J. L. Awange, K. H. Bae, and S. J. Claessens. Procrustean solution of the 9-parameter transformation problem. *Earth, Planets and Space*, 2008.

[40] B. Wen, W. Lian, K. Bekris, and S. Schaal. CaTGrasp: Learning Category-Level Task-Relevant Grasping in Clutter from Simulation. In *ICRA*, 2022.

[41] B. Wen, W. Lian, K. Bekris, and S. Schaal. You Only Demonstrate Once: Category-Level Manipulation from Single Visual Demonstration. In *RSS*, 2022.

[42] Y. Wu, O. P. Jones, and I. Posner. Obpose: Leveraging canonical pose for object-centric scene inference in 3d, 2022.

[43] Q.-Y. Zhou, J. Park, and V. Koltun. Open3d: A modern library for 3d data processing, 2018. URL http://arxiv.org/abs/1801.09847.

[44] N. Ravi, J. Reizenstein, D. Novotny, T. Gordon, W. Lo, J. Johnson, and G. Gkioxari. Accelerating 3d deep learning with pytorch3d. *CoRR*, abs/2007.08501, 2020. URL https://arxiv.org/abs/2007.08501.