Coupled Iterative Refinement for 6D Multi-Object Pose Estimation

Supplemental Material

1 Additional inputs to the GRU

The correlation features and hidden state input to the GRU are described in the main paper. The additional inputs to the GRU are described in this section.

Context features Following the procedure described in RAFT [9], we use a Resnet[3]-based feature extractor to construct context features and an initial hidden state for every image (Fig. 1b). The hidden state is updated with every GRU iteration, while the context features remain unchanged.

Depth features For each image pair $(i, 0)$, we have inverse-depth maps $\mathbf{z}_i^{-1}$ and $\mathbf{z}_0^{-1}$. We use $\mathbf{x}_{i\to 0}$ to index $\mathbf{z}_0^{-1}$, producing a new inverse-depth map $\mathbf{z}_{i\to 0}^{-1}$. The depth residuals $(\mathbf{z}_0^{-1} - \mathbf{z}_{i\to 0}^{-1})$ implied by the induced correspondences $\mathbf{x}_{i\to 0}$ are provided as input to the GRU before each update. This gives the GRU more information on how well depth has been aligned. Since $\mathbf{x}_{i\to 0}$ are real numbers, we use bilinear interpolation. This procedure is identical for correspondences from image 0 to i.

Solver Residuals The solver residuals from the previous solver iteration are fed to the GRU. These residuals are calculated by taking the difference between the current induced correspondences $\mathbf{x}_{i\to 0}^{(t)}$ and the previous revised correspondences $\mathbf{x}_{i\to 0}^{\prime(t-1)}$. This allows our model to detect outliers easily as the pixels with unusually high residuals, and thus prune them in the next update.

2 RGB-Only Optimization

In the RGB-D setting, depth for the input image is given and stays fixed for each pixel regardless of the current pose estimate. In the RGB-Only setting, depth is obtained by rendering the object at the current pose estimate $\mathbf{G}_0$. See Fig. 1a.

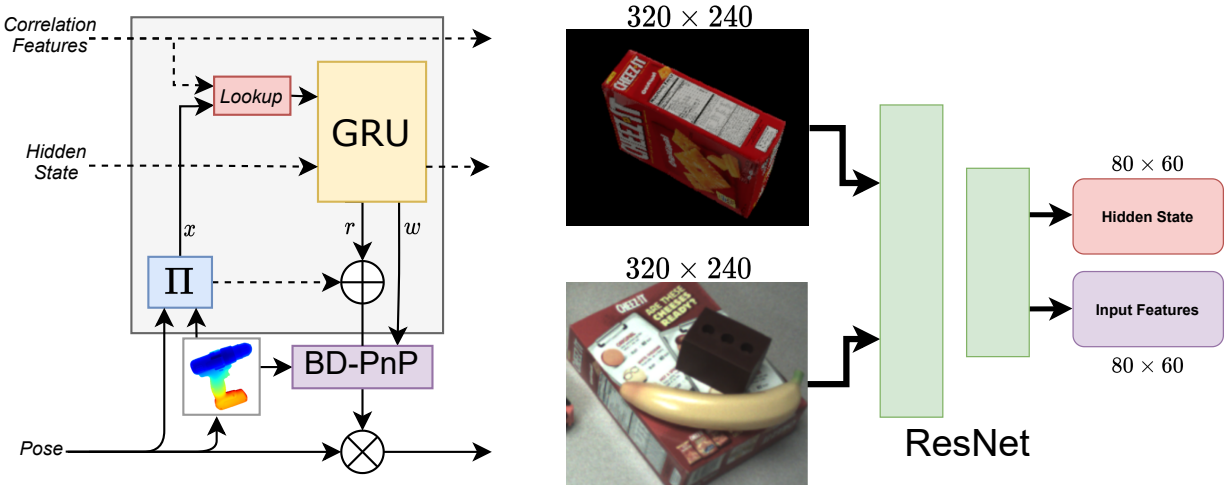
$$\mathbf{x}_0^{RGBD} = \begin{bmatrix} x_0 \\ y_0 \\ d_0 \end{bmatrix} \quad \mathbf{x}_0^{RGB} = \begin{bmatrix} x_0 \\ y_0 \\ \mathcal{R}(\mathbf{G}_0) \end{bmatrix} \tag{1}$$

In our objective function, $\mathbf{G}_0$ maps the points between images. In the RGB setting, it also produces the depth for points $\mathbf{x}_0^{RGB}$ in the input image.

$$\textbf{RGBD}: \quad \mathbf{E}(\mathbf{G}_0) = \sum_{i=1}^{N} \left|\left|\mathbf{x}'_{i\to 0} - \Pi(\mathbf{G}_0\mathbf{G}_i^{-1}\Pi^{-1}(\mathbf{x}_i))\right|\right|^2_{\Sigma_{i\to 0}} + \sum_{i=1}^{N} \left|\left|\mathbf{x}'_{0\to i} - \Pi(\mathbf{G}_i\mathbf{G}_0^{-1}\Pi^{-1}(\mathbf{x}_0^{RGBD}))\right|\right|^2_{\Sigma_{0\to i}} \tag{2}$$

$$\textbf{RGB}: \quad \mathbf{E}(\mathbf{G}_0) = \sum_{i=1}^{N} \left|\left|\mathbf{x}'_{i\to 0} - \Pi(\mathbf{G}_0\mathbf{G}_i^{-1}\Pi^{-1}(\mathbf{x}_i))\right|\right|^2_{\Sigma_{i\to 0}} + \sum_{i=1}^{N} \left|\left|\mathbf{x}'_{0\to i} - \Pi(\mathbf{G}_i\mathbf{G}_0^{-1}\Pi^{-1}(\mathbf{x}_0^{RGB}))\right|\right|^2_{\Sigma_{0\to i}} \tag{3}$$

In the RGB setting, the depth of $\mathbf{x}_0^{RGB}$ is a function of the pose $\mathbf{G}_0$ in Eq. 3. However, it is difficult to treat it as such in the optimization since our differentiable solver requires calculating the derivative of the Jacobian for the rendering function $\mathcal{R}$. Instead, we treat the depth in the image as a variable to be optimized over jointly with the pose $\mathbf{G}_0$. The resulting optimized pose and depth may be inconsistent with one another since they were treated as separate variables in the optimization. Therefore, we discard the depth update produced by the solver and produce a new depth map by rendering the updated pose. This ensures that the depth is a function of the pose.



(a) RGB Pose Update Module

(b) Context features and hidden state initialization

3 Implementation Details:

Training Schedule: Our method is implemented in Pytorch [8]. All models are initialized from scratch with random weights and trained for 100K steps. During training, we use the AdamW [7] optimizer. Final models are trained with a batch size of 12 on two RTX-3090tis. Ablation experiments are trained with a batch size of 4. We use an exponential learning rate schedule with a linear increase to 3×10^{-4} over 10000 steps and a 50% drop every subsequent 20000 steps. We do not use any weight decay. We use the full resolution images for training: 640×480 for YCB-V, 720×540 for T-LESS, and 640×480 for LM-O. In the inner update loop, the correspondence field, confidence weights, depth, hidden state, etc. are maintained at 80×60 spatial resolution, which is $\frac{1}{4}$ of the resolution of the 320×240 input crop. We downsample the input depth by subsampling and use strided convolutions for the image features.

Image augmentation: We use the same image augmentation as [6] on all three datasets, specifically contrast, hue, sharpness, gaussian blur, and brightness. The depth images in the training data are sparse, so we fill in the gaps using bilinear interpolation.

4 Accuracy Metrics

Our ablation experiments and main results follow the evaluation protocol used in the BOP Challenge [5]. Here, we formally define the error metrics used.

Visible Surface Discrepancy (VSD)

$$e_{\text{VSD}} = \underset{p \in \hat{V} \cup \bar{V}}{\text{avg}} \begin{cases} 0 & \text{if } p \in \hat{V} \cap \bar{V} \wedge |\hat{D}(p) - \bar{D}(p)| < \tau \\ 1 & \text{otherwise,} \end{cases}$$

where $\hat{D}(p)$ and $\bar{D}(p)$ are the depth maps obtained by rendering the object at the predicted pose and ground-truth pose, respectively. $\hat{V}$ and $\bar{V}$ are visibility masks obtained by comparing each depth map with the sensor depth. VSD treats indistinguishable poses as identical. VSD Recall is the percent of VSD scores less than 10 thresholds ranging from 0.05 to 0.5, with the misalignment tolerance τ ranging from 5% to 50% of the object's diameter.

Maximum Symmetry-Aware Surface Distance (MSSD)

$$e_{\text{MSSD}} = \min_{\mathbf{S} \in S_O} \max_{\mathbf{x} \in V_O} \|\hat{\mathbf{P}}\mathbf{x} - \bar{\mathbf{P}}\mathbf{S}\mathbf{x}\|_2,$$

	YCB-V [2]		T-LESS [4]		LM-O[1]	
	MSPD Recall	MSSD Recall	MSPD Recall	MSSD Recall	MSPD Recall	MSSD Recall
Bidirectional (depth as variable) PnP	0.833	**0.751**	**0.649**	**0.474**	**0.793**	**0.609**
Unidirectional (depth as variable) PnP [render to image]	**0.834**	0.750	0.639	0.456	0.792	0.593
Unidirectional (depth as variable) PnP [image to render]	0.750	0.630	0.480	0.234	0.645	0.379
Multiview renders	**0.833**	**0.751**	**0.649**	**0.474**	**0.793**	**0.609**
Single render	0.814	0.727	0.619	0.429	0.772	0.569
Predicting per-pixel confidence weights	**0.833**	**0.751**	**0.649**	**0.474**	**0.793**	**0.609**
Uniform confidence	0.694	0.598	0.568	0.377	0.765	0.568
Revised approach (depth as variable)	**0.833**	**0.751**	**0.649**	**0.474**	**0.793**	**0.609**
Depth as variable but do not discard depth update	0.744	0.674	0.520	0.326	0.725	0.497
Use Gradient Clipping	0.833	0.751	0.649	0.474	0.793	0.609
No Gradient Clipping	Training diverges causing NaNs					
Bidirectional context features	0.833	0.751	**0.649**	**0.474**	0.793	**0.609**
Using unidirectional context features	**0.848**	**0.764**	0.647	0.452	**0.807**	0.608
Pose + Flow Loss	**0.833**	**0.751**	**0.649**	**0.474**	**0.793**	**0.609**
Flow Loss Only	0.760	0.666	0.589	0.378	0.741	0.521
Pose Loss Only	0.246	0.190	0.263	0.166	0.271	0.093
Depth-augmented PnP (predicting depth revisions)	**0.842**	**0.765**	0.647	0.465	**0.803**	**0.616**
No depth augmentation (no depth revisions)	0.833	0.751	**0.649**	**0.474**	0.793	0.609

Table 1: Additional ablation experiments using our method for RGB-Only input. We evaluate our method on a held-out subset of training images. Initial poses are generated by randomly perturbing the ground truth pose. Options used in our full RGB method are bolded.

	YCB-V [2]		T-LESS [4]		LM-O [1]	
	MSPD Recall	MSSD Recall	MSPD Recall	MSSD Recall	MSPD Recall	MSSD Recall
Bidirectional context features	**0.924**	**0.955**	**0.685**	**0.582**	**0.828**	**0.788**
Unidirectional context features	0.910	0.944	**0.685**	**0.582**	0.826	0.784

Table 2: Additional ablation experiments using our method for RGB-D input. We evaluate our method on held-out training images. Initial poses are generated by randomly perturbing the ground truth pose. Options used in our full method are bolded.

where S_O are the rotation symmetries of object O and V_O are its vertices. $\hat{\mathbf{P}}$ and $\bar{\mathbf{P}}$ are the predicted and ground truth poses. MSSD is useful for robotic manipulation where the maximum surface deviation is related to the chance of a successful grasp. Compared to the average distance used in ADD/ADI, the maximum distance is less dependant on the sampling density of vertices. MSSD Recall is the percent of MSSD scores less than 10 thresholds ranging from 5% to 50% of the object's diameter.

Maximum Symmetry-Aware Projection Distance (MSPD)

$$e_{\text{MSPD}} = \min_{\mathbf{S}\in S_O} \max_{\mathbf{x}\in V_O} \|\text{proj}(\hat{\mathbf{P}}\mathbf{x}) - \text{proj}(\bar{\mathbf{P}}\mathbf{S}\mathbf{x})\|_2,$$

where V_O, S_O, $\hat{\mathbf{P}}$, $\bar{\mathbf{P}}$ are defined above. MSPD evaluates the perceivable discrepancy, which is important for augmented reality applications. Like MSSD, MSPD also measures the maximum distance instead of the average in order to be robust to the sampling density of vertices. MSPD Recall is the percent of MSPD scores less than 10 thresholds ranging from 5 to 50 (measured in pixels).

5 Additional Ablations

Gradient Clipping Treating depth as a variable in the optimization leads to unstable behavior after several thousand training steps. We solve this problem by clipping the gradients of the input to the GRU to a maximum of 0.01 in the middle of the backward pass.

Discarding Depth Update In the RGB setting, we jointly optimize the pose and depth together. We then discard the depth update and generate a new depth map by rendering the updated pose (see Sec 2). This works

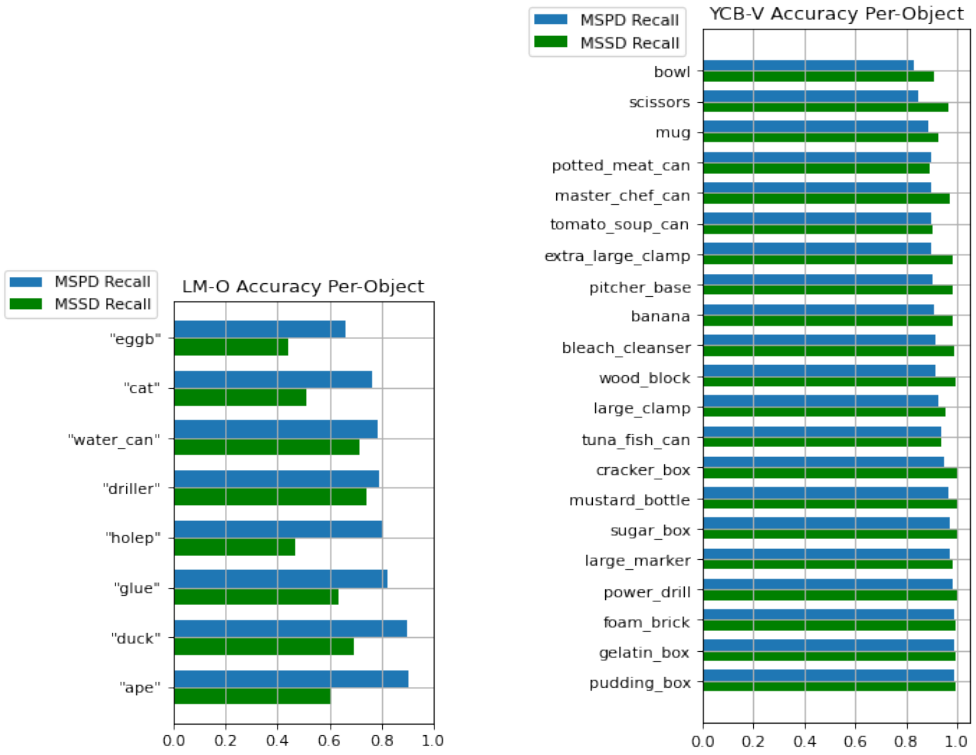
better than jointly optimizing pose and depth together but applying the depth update to the previous estimate instead of discarding it.

RGB-Only Bidirectional (depth as variable) PnP In the RGB setting, bidirectional PnP is helpful on T-LESS while benign on LM-O and YCB-V. On YCB-V, our method converges to an accurate pose even when only using correspondences from the input image to the rendered image (Tab. 1). This indicates that the induced flow is sufficiently accurate even when using a depth *approximation* in the mapping function Π .

RGB-Only Depth-augmented PnP In the RGB setting, depth residuals are helpful on YCB-V and LM-O but not on T-LESS (Tab. 1). The inconsistent benefit could arise from the fact that the depth in the image is generated from the current pose estimate, and therefore predicting the appropriate depth residuals in the GRU is more challenging.

Bidirectional Context features There is a significant domain gap between the input images and the rendered images. This is due to, among other things, innaccurate meshes resulting from imperfect scans, significant lighting differences and different reflectance properties. We evaluate the importance of extracting contextual features from both images, contrary to RAFT [9] which operates in a single image domain. Tabs. 1 & 2 show that using contextual information from only the source image is sufficient to establish accurate correspondences.

Figure 1: Additional qualitative results on the YCB-V Dataset. Our method incorrectly orients the bowl and spam in the far left image.



(a) Linemod (Occluded) Accuracy

(b) YCB-V Accuracy

Figure 2: Accuracy Per-Object on the Linemod (Occluded) and YCB-V Datasets

Figure 3: Additional qualitative results on the T-LESS Dataset. Our method incorrectly positions the purple object in the far right image.

Figure 4: Qualitative results on the Linemod (Occluded) Dataset

Licenses: Linemod and T-LESS licensed under CC BY 4.0. Linemod (Occluded) licensed under CC BY-SA 4.0. YCB-V licensed under MIT. Cosypose models licensed under MIT.

References

[1] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. Learning 6d object pose estimation using 3d object coordinates. In *European conference on computer vision*, pages 536–551. Springer, 2014. 3

[2] Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. The ycb object and model set: Towards common benchmarks for manipulation research. In *2015 international conference on advanced robotics (ICAR)*, pages 510–517. IEEE, 2015. 3

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[4] Tomáš Hodan, Pavel Haluza, Štepán Obdržálek, Jiri Matas, Manolis Lourakis, and Xenophon Zabulis. T-less: An rgb-d dataset for 6d pose estimation of texture-less objects. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 880–888. IEEE, 2017. 3

[5] Tomas Hodan, Frank Michel, Eric Brachmann, Wadim Kehl, Anders GlentBuch, Dirk Kraft, Bertram Drost, Joel Vidal, Stephan Ihrke, Xenophon Zabulis, et al. Bop: Benchmark for 6d object pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 19–34, 2018. 2

[6] Yann Labbé, Justin Carpentier, Mathieu Aubry, and Josef Sivic. Cosypose: Consistent multi-view multi-object 6d pose estimation. In *European Conference on Computer Vision*, pages 574–591. Springer, 2020. 2

[7] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 2

[8] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 2

[9] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pages 402–419. Springer, 2020. 1, 4