Shape-Constraint Recurrent Flow for 6D Object Pose Estimation

Yang Hai [1],   Rui Song [1],   Jiaojiao Li [1],   Yinlin Hu [2]

[1] State Key Laboratory of ISN, Xidian University,   [2] MagicLeap

Abstract

Most recent 6D object pose methods use 2D optical flow to refine their results. However, the general optical flow methods typically do not consider the target's 3D shape information during matching, making them less effective in 6D object pose estimation. In this work, we propose a shape-constraint recurrent matching framework for 6D object pose estimation. We first compute a pose-induced flow based on the displacement of 2D reprojection between the initial pose and the currently estimated pose, which embeds the target's 3D shape implicitly. Then we use this pose-induced flow to construct the correlation map for the following matching iterations, which reduces the matching space significantly and is much easier to learn. Furthermore, we use networks to learn the object pose based on the current estimated flow, which facilitates the computation of the pose-induced flow for the next iteration and yields an end-to-end system for object pose. Finally, we optimize the optical flow and object pose simultaneously in a recurrent manner. We evaluate our method on three challenging 6D object pose datasets and show that it outperforms the state of the art significantly in both accuracy and efficiency.

1. Introduction

6D object pose estimation, *i.e.*, estimating the 3D rotation and 3D translation of a target object with respect to the camera, is a fundamental problem in 3D computer vision and also a crucial component in many applications, including robotic manipulation [8] and augmented reality [34]. Most recent methods rely on pose refinement to obtain accurate pose results [16, 31, 52]. Typically, they first synthesize an image based on the rendering techniques [9, 38] according to the initial pose, then estimate dense 2D-to-2D correspondence between the rendered image and the input based on optical flow networks [46]. After lifting the estimated 2D optical flow to 3D-to-2D correspondence based on the target's 3D shape, they can obtain a new refined pose using Perspective-n-Points (PnP) solvers [27].

Although this paradigm works well in general, it suffers from several weaknesses. First, the general optical flow

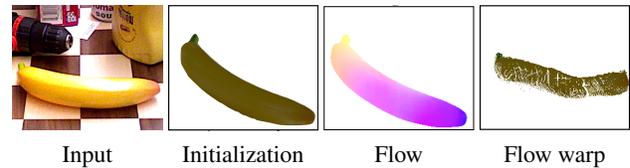| Input | Initialization | Flow | Flow warp |

Figure 1. **The problem of optical flow in 6D pose estimation.** Given an initial pose, one can estimate the dense 2D-to-2D correspondence (optical flow) between the input and the synthetic image rendered from the initial pose, and then lift the dense 2D matching to 3D-to-2D correspondence to obtain a new refined pose by PnP solvers (PFA-Pose [16]). However, the flow estimation does not take the target's 3D shape into account, as illustrated by the warped image based on the estimated flow in the last figure, which introduces significant matching noise to pose solvers and is suboptimal for 6D object pose estimation.

networks they use are mainly built on top of two assumptions, *i.e.*, the brightness consistency between two potential matches and the smoothness of matches within a local neighbor [1]. These assumptions, however, are too general and do not have any clue about the target's 3D shape in the context of 6D object pose estimation, making the potential matching space of every pixel unnecessarily large in the target image. Second, the missing shape information during matching often results in flow results that do not respect the target shape, which introduces significant matching noise, as shown in Fig. 1. Third, this multi-stage paradigm trains a network that relies on a surrogate matching loss that does not directly reflect the final 6D pose estimation task [17], which is not end-to-end trainable and suboptimal.

To address these problems, we propose a shape-constraint recurrent matching framework for 6D object pose estimation. It is built on top of the intuition that, in addition to the brightness consistency and smoothness constraint in classical optical flow solutions [2, 35], the dense 2D matching should comply with the 3D shape of the target. We first build a 4D correlation volume between every pixel of the source image and every pixel of the target image, similar to RAFT [46]. While, instead of indexing from the correlation volume according to the current flow during the iteration, we propose indexing the correlation volume based on
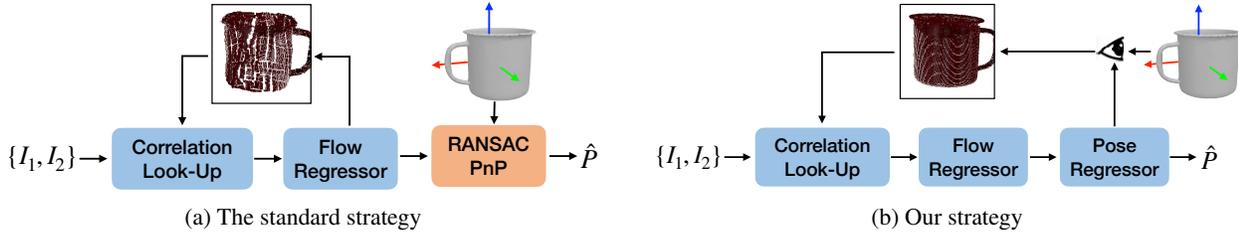
(a) The standard strategy   (b) Our strategy

Figure 2. **Different pose refinement paradigms. (a)** Most pose refinement methods [16] rely on a recurrent architecture to estimate dense 2D flow between the rendered image I_1 and the real input image I_2, based on a dynamically-constructed correlation map according to the flow results of the previous iteration. After the convergence of the flow network and lifting the 2D flow to a 3D-to-2D correspondence field, they use PnP solvers to compute a new refined pose $\tilde{P}$. This strategy, however, has a large matching space for every pixel in constructing correlation maps, and optimizes a surrogate matching loss that does not directly reflect the final 6D pose estimation task. **(b)** By contrast, we propose optimizing the pose and flow simultaneously in an end-to-end recurrent framework with the guidance of the target's 3D shape. We impose a shape constraint on the correlation map construction by forcing the construction to comply with the target's 3D shape, which reduces the matching space significantly. Furthermore, we propose learning the object pose based on the current flow prediction, which, in turn, helps the flow prediction and yields an end-to-end system for object pose.

a pose-induced flow, which is forced to contain only all the 2D reprojections of the target's 3D shape and reduces the matching space of the correlation map construction significantly. Furthermore, we propose to use networks to learn the object pose based on the current flow prediction, which facilitates the computation of the pose-induced flow for the next iteration and also removes the necessity of explicit PnP solvers, making our system end-to-end trainable and more efficient, as shown in Fig. 2(b).

We evaluate our method on the challenging 6D object pose benchmarks, including LINEMOD [14], LINEMOD-Occluded [25], and YCB-V [50], and show that our method outperforms the state of the art significantly, and converges much more quickly.

2. Related Work

Object pose estimation, has shown significant improvement [36, 47, 50] after the utilization of deep learning techniques [13, 51]. While most of them still follow the traditional paradigm, which consists of the establishment of 3D-to-2D correspondence and the PnP solvers. Most recent methods create the correspondence either by predicting 2D points of some predefined 3D points [18, 21, 36, 37] or predicting the corresponding 3D point for every 2D pixel location within a segmentation mask [3, 10, 29, 42, 48, 53]. On the other hand, some recent methods try to make the PnP solvers differentiable [4, 5, 17]. However, the accuracy of these methods still suffers in practice. We use pose refinement to obtain more accurate results in this work.

Object pose refinement, usually relies on additional depth images [29, 45, 47, 50], which is accurate but the depth images are hard to obtain in some scenarios and even inaccessible in many applications [21, 40]. Most recent refinement methods use a render-and-compare strategy without

any access to depth images and achieve comparable performance [16, 23, 26, 28, 31, 33, 37, 52, 53]. These methods, however, usually formulate pose refinement as a general 2D-to-2D matching problem and do not consider the fact that the dense 2D matching should comply with the 3D shape of the target, which is suboptimal in 6D object pose estimation. On the other hand, most of them rely on numerical PnP solvers [27] as their post processing and optimize a surrogate matching loss that does not directly reflect the final 6D pose estimation task during training. By contrast, we propose a recurrent matching framework guided by the 3D shape of the target, which transforms the constraint-free matching problem into a shape-constraint matching problem. Furthermore, we propose to learn the object pose from the intermediate matches iteratively, making our method end-to-end trainable and producing more accurate results.

Optical flow estimation, whose goal is to obtain the matching of every pixel from the source image to the target image, has been widely studied for a long time [1]. Classically, it is formulated as an energy optimization problem, which is usually based on the assumption of brightness consistency and local smoothness [1, 6, 19, 20, 39]. Recently, the learning-based methods inspired by those traditional intuitions have shown great progress in estimating flow in large-displacement and occlusion scenarios [1, 11, 22, 24, 30, 43, 44]. Especially, RAFT [46], which introduces a recurrent deep network architecture for optical flow, has shown significant improvement over the previous learning-based methods further. However, these optical flow methods are general and can not utilize the target's prior knowledge, making them suboptimal when used in 6D object pose estimation. By contrast, we propose embedding the target's 3D shape prior knowledge into the optical flow framework for 6D object pose estimation, which reduces the matching space significantly and is much easier to learn.
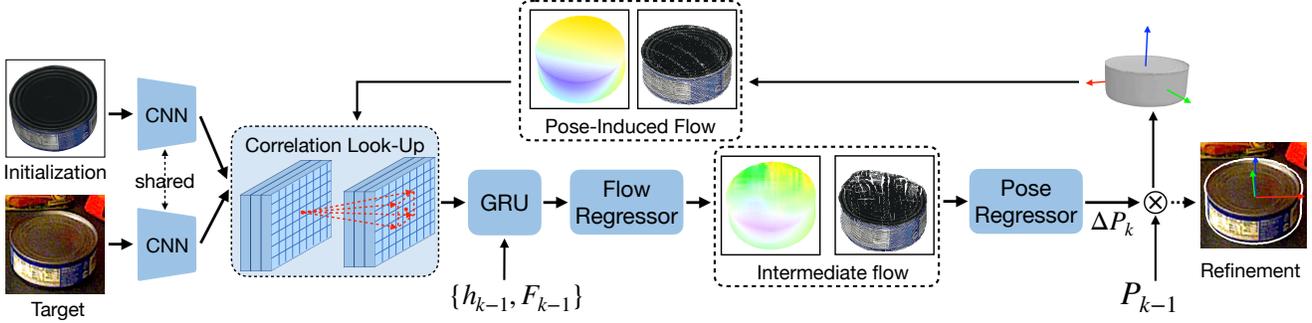
Figure 3. **Overview of our shape-constraint recurrent framework.** After building a 4D correlation volume between the rendered image and the input target image, we use GRU [7] to predict an intermediate flow, based on the predicted flow F_{k-1} and the hidden state h_{k-1} of GRU from the previous iteration. We then use a pose regressor to predict the relative pose ΔP_k based on the intermediate flow, which is used to update the previous pose estimation P_{k-1}. Finally, we compute a pose-induced flow based on the displacement of 2D reprojection between the initial pose and the currently estimated pose P_k. We use this pose-induced flow to index the correlation map for the following iterations, which reduces the matching space significantly. Here we show the flow and its corresponding warp results in the dashed boxes. Note how the intermediate flow does not preserve the shape of the target, but the pose-induced flow does.

3. Approach

Given a calibrated RGB input image and the 3D model of the target, our goal is to estimate the target's 3D rotation and 3D translation with respect to the camera. We obtain a pose initialization based on existing pose methods [21, 50], and refine it using a recurrent matching framework. We focus on the refinement part in this paper. We first have an overview of our framework and then discuss the strategy of reducing the search space of matching by imposing a shape constraint. Finally, we present the design of learning object pose based on optical flow to make our matching framework end-to-end trainable.

3.1. Overview

Given an input image and the initial pose, we synthesize an image by rendering the target according to the initial pose, and use a shared-weight CNN to extract features for both the rendered image and the input image, and then build a 4D correlation volume containing the correlations of all pairs of feature vectors between the two images, similar to PFA [16, 46]. However, unlike the standard strategy that indexes the correlation volume for the next iteration without any constraints, we use a pose-induced flow for the indexing, which embeds the target's shape information implicitly.

We first predict an intermediate flow based on the constructed correlation map. Then we use a pose regressor to predict an intermediate pose based on the intermediate flow. After that, we compute the pose-induced flow based on the displacement of 2D reprojection between the initial pose and the currently estimated pose. We use this pose-induced flow to index the correlation map for the next iteration, which reduces the matching space significantly. On the other hand, the pose regressor based on the intermedi-

ate flow removes the need for RANSAC-PnP and produces an end-to-end system for object pose. Fig. 3 shows the overview of our framework.

3.2. Shape-Constraint Correlation Space

We first obtain a 4D correlation volume $\mathbf{C} \in \mathbb{R}^{H \times W \times H \times W}$ based on the dot product between all pairs of feature vectors from image features from different pyramid levels [46]. The standard lookup operation generates a correlation feature map by indexing from the correlation volume, which maps the feature vectors at location $\mathbf{x} = (u, v)$ in I_1 to the corresponding new location in I_2: $\mathbf{x}' = (u, v) + f(u, v)$, where $f(u, v)$ is the currently estimated flow. This standard lookup operation works well in general, but does not consider the fact that all the matches should comply with the shape of the target in 6D object pose estimation, making its matching space unnecessarily large.

To address this, we embed the target's 3D shape into the lookup operation, generating a shape-constraint location in constructing the new correlation map:

$$\mathbf{x}' = (u_k, v_k) + f(u_k, v_k; \mathbf{K}, \mathbf{S}, \mathbf{P}_0, \mathbf{P}_k),\ 1 \le k \le N,\ (1)$$

where N is the number of iterations, $\mathbf{K}$ is the intrinsic camera matrix, $\mathbf{S}$ is the target's 3D shape, $\mathbf{P}_0$ and $\mathbf{P}_k$ are the initial pose and the currently estimated pose of the target, respectively. We call the flow fields $\mathbf{f}(u_k, v_k; \mathbf{K}, \mathbf{S}, \mathbf{P}_0, \mathbf{P}_k)$ pose-induced flow.

More specifically, given a 3D point $\mathbf{p}_i$ on the target's mesh, we use the perspective camera model to get its 2D location $\mathbf{u}_{i0}$ under the initial pose $\mathbf{P}_0$,

$$\lambda_i \begin{bmatrix} \mathbf{u}_{i0} \\ 1 \end{bmatrix} = \mathbf{K}(\mathbf{R}_0 \mathbf{p}_i + \mathbf{t}_0), \qquad (2)$$
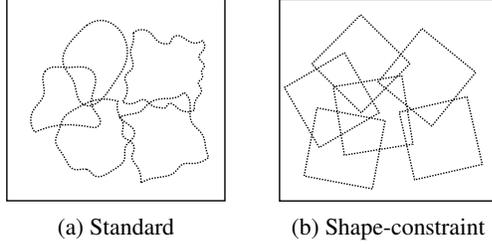
(a) Standard (b) Shape-constraint

Figure 4. **Illustration of shape-constraint correlation space. (a)** The standard index operation has no constraint in constructing the correlation map, which has unnecessarily large search space for matching in 6D object pose estimation. **(b)** By contrast, we force it to contain only all the 2D reprojections of the target's 3D shape (illustrated as rectangles), reducing the matching space significantly.
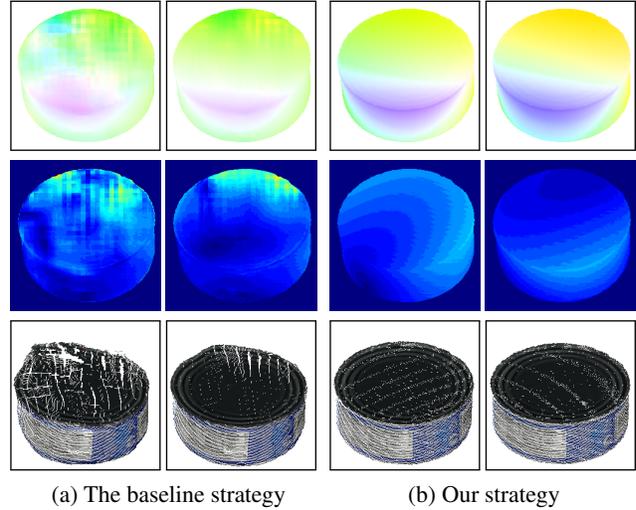

(a) The baseline strategy (b) Our strategy

Figure 5. **Comparison of the predicted flow.** From top to bottom, we show the predicted flow, the flow error map, and the corresponding warp images for both the baseline PFA and our method. We show the results with 2 and 8 iterations from left to right for each method. The baseline can not preserve the target's shape during the iterations, and our method preserves it after every iteration.

where λ_i is a scale factor, and $\mathbf{R}_0$ and $\mathbf{t}_0$ are the rotation matrix and translation vector representing the initial pose $\mathbf{P}_0$. Similarly, we obtain a new 2D location $\mathbf{u}_{ik}$ of the same 3D point $\mathbf{p}_i$ under the currently estimated pose $\mathbf{P}_k$. Then the pose-induced flow $f = \mathbf{u}_{ik} - \mathbf{u}_{i0}$, which represents the displacement of 2D reprojection of the same 3D point between the initial pose and the currently estimated pose. We compute 2D flow only for the 3D points on the visible surface of the mesh. Fig. 4 illustrates the advantages of this strategy.

3.3. Learning Object Pose From Optical Flow

The pose-induced flow relies on the current pose prediction. In principle, the pose can be obtained by a PnP solver based on the current intermediate flow [16]. This strategy, however, is not easy to be stably differentiable during training [17, 49]. Instead, we propose using networks to learn the object pose.

We learn a residual pose $\Delta\mathbf{P}_k$ based on the current intermediate flow, which updates the estimated pose iteratively: $\mathbf{P}_k = \mathbf{P}_{k-1} \otimes \Delta\mathbf{P}_k$. Note that the intermediate flow does not preserve the target's shape, as shown in Fig. 3.

For the supervision of the residual pose, we first encode the residual rotation $\Delta\mathbf{R}$ into a six-dimensional representation [54], and parameterize the residual translation $\Delta\mathbf{T}$ as a form of 2D offsets and scaling on the image plane [28].

After predicting the current pose $\mathbf{P}_k$, we compute the pose-induced flow based on the initial pose and $\mathbf{P}_k$, as discussed in the previous section. The pose-induced flow, which is used to construct the correlation map for the next iteration, preserves the targets's shape, as shown in Fig. 5.

3.4. Implementation Details

Our method works in a render-and-compare manner [23, 28, 52] to learn the difference between the input image I_1 and I_2. For I_1, we use Pytorch3D [38] to render the target according to the initial pose $\mathbf{P}_0$, with a fixed image reso-

lution of 256×256. For I_2, we crop the region of interest from the raw input image based on $\mathbf{P}_0$ and resize it to the same resolution as I_1.

We use GRU [7, 46] for the recurrent modeling

$$h_k = \mathrm{GRU}(C_k, F_{k-1}, h_{k-1}; \Theta), \quad (3)$$

where, C_k is the constructed correlation map for the current iteration, F_{k-1} is the intermediate flow from the previous iteration, and h_k and Θ are the hidden state feature and network parameters of the GRU structure, respectively.

For the pose regressor, we concatenate the intermediate flow and the hidden state feature of GRU, and use two networks to predict $\Delta\mathbf{R}$ and $\Delta\mathbf{T}$, respectively. These two small networks have the same architecture except for the dimension of the final output layer, and consist of three convolutional layers and two fully connected layers for each.

We predict both the optical flow and the object pose iteratively. To supervise them in each iteration, we use a simple exponentially weighting strategy [46] in our loss

$$\mathcal{L} = \sum_{k=1}^{N} \gamma^{N-k} (\mathcal{L}_{pose}^k + \alpha \mathcal{L}_{flow}^k) \quad (4)$$

where γ is the exponential weighting factor, and α is the parameter balancing the object pose loss $\mathcal{L}_{pose}$ and the optical flow loss $\mathcal{L}_{flow}$. We use $N = 8$, $\gamma = 0.8$ and $\alpha = 0.1$ in this work.

To compute the object pose loss $\mathcal{L}_{pose}$, we randomly select 1k 3D points from the surface of the object's 3D mesh,

Dataset	PoseCNN	PVNet	SO-Pose	DeepIM	RePose	RNNPose	PFA	**Ours**
LM	63.3	86.3	96.0	88.6	96.1	<u>97.4</u>	95.8	**99.3**
LM-O	24.9	40.8	62.3	55.5	51.6	60.7	<u>65.3</u>	**66.4**
YCB-V	21.3	-	56.8	53.6	62.1	<u>66.4</u>	62.8	**70.5**

Table 1. **Comparison against the state of the art in ADD-0.1d.** Our method outperforms the competitors by a large margin.

and then calculate the distance between these points transformed by the ground truth pose and the predicted pose, respectively. For the optical flow loss $\mathcal{L}_{flow}$, we first compute the ground truth flow based on the initial pose and the ground truth pose, by the geometry reasoning as discussed in Section 3.2. Then we use the L1 loss to capture the end-point error between the ground truth flow and the intermediate flow. We only supervise the pixels within the mask of the rendered target, and discard the pixels under occlusion.

We train our model using AdamW [32] optimizer with a batch size of 16, and use an adaptive learning rate scheduler based on One-Cycle [41], starting from 4e-4. We typically train the model for 100k steps. During training, we randomly generate a pseudo initial pose around the ground truth pose of the input image, and render the reference image I_1 according to the pseudo initial pose on the fly.

4. Experiments

In this section, we evaluate our method systematically. We first introduce our experiment settings and then demonstrate the effectiveness of our method by comparing with the state of the art. Finally, we conduct extensive ablation studies to validate the design of our method. Our source code is publicly available at https://github.com/YangHai-1218/SCFlow.

4.1. Experiment Setup

Datasets. We evaluate our method on three challenging datasets, including LINEMOD ("LM") [14], LINEMOD-Occluded ("LM-O") [25], and YCB-V [50]. LINEMOD contains 13 sequences, each containing a single object annotated with accurate ground-truth poses. LINEMOD-Occluded has 8 objects which is a subset of the LM objects. Its test set is one of the sequences in LM, which contains all the annotations of the 8 objects in the scene. There is no standard experiment setting on LM and LM-O. Some previous methods [10,48,52] use different training settings for LM and LM-O, and some methods train a separated model for every single object [23, 36]. For consistency and simplicity, we train a single model for all the objects on both LM and LM-O, and for each sequence, we use about 15% of the RGB images for training, resulting in a total of 2.4k images. YCB-V is a more challenging dataset containing 21

Method	Avg.	MSPD	VSD	MSSD
YCB-V (Real+PBR)				
Ours	**0.826**	**0.860**	<u>0.778</u>	<u>0.840</u>
PFA	0.795	0.844	0.743	0.797
CIR	<u>0.824</u>	0.852	**0.783**	0.835
CosyPose	0.821	<u>0.850</u>	0.772	**0.842**
SurfEmb	0.781	-	-	-
YCB-V (PBR)				
Ours	**0.651**	<u>0.769</u>	**0.556**	**0.626**
PFA	0.615	0.739	0.521	0.585
SurfEmb	<u>0.647</u>	**0.773**	<u>0.548</u>	<u>0.620</u>
CosyPose	0.574	0.653	0.516	0.554
LM-O (PBR)				
Ours	**0.682**	<u>0.842</u>	**0.532**	**0.674**
PFA	<u>0.674</u>	0.818	<u>0.531</u>	<u>0.673</u>
CIR	0.655	0.831	0.501	0.633
SurfEmb	0.647	**0.851**	0.497	0.640
CosyPose	0.633	0.812	0.480	0.606

Table 2. **Refinement comparison in BOP metrics.** We compare our method with the state-of-the-art refinement methods, and our method achieves the best accuracy in different settings in most metrics.

objects and 130k real images in total, which is captured in cluttered scenes. Besides, we conduct some ablation studies with the BOP synthetic datasets [15] that include the same objects as those in LM and YCB-V but generated with physically-based rendering (PBR) techniques [9]. We train our model only with the real data if not explicitly stated.

Evaluation metrics. We use the standard ADD(-S) metric to report the results, which is based on the mean 3D distance between the mesh vertices transformed by the predicted pose and the ground-truth pose, respectively. We report most of our results in ADD-0.1d, which is the percentage of correct predictions with a distance error less than 10% of the mesh diameter. In some ablation studies, we report ADD-0.05d, which uses a threshold of 5% of the model diameter. Some recent methods [12, 26, 31] only report their results in BOP metrics [15]. For comparing with them, we report some of our results in BOP metrics which

Method	CIR	CosyPose	SurfEmb	PFA	**Ours**
Timing	11k	20	1k	37	**17**

Table 3. **Efficiency comparison.** We run all the methods on the same workstation, and report the running time in milliseconds in processing an image containing one object instance. Our method is the most efficient one among all the competitors.

include the Visible Surface Discrepancy (VSD), the Maximum Symmetry-aware Surface Distance (MSSD), and the Maximum Symmetry-aware Projection Distance (MSPD). We refer readers to [15] for the detailed metric definition.

4.2. Comparison to the State of the Art

We compare our method with most state-of-the-art methods, including PoseCNN [50], PVNet [36], SO-Pose [10], DeepIM [28], RePose [23], RNNPose [52], and PFA [16]. We use the results of PoseCNN as our pose initialization by default. For PFA, we use its official code which relies on online rendering and has only one view for correspondence generation, producing slightly better results than that in its paper. Nevertheless, our method outperforms most of them by a large margin, as shown in Table 1.

Furthermore, we compare with the recent refinement methods CIR [31], CosyPose [26], and SurfEmb [12], which only have results in BOP metrics. Since all of them use the first stage of CosyPose as their pose initialization, we use the same initialization in this experiment for a fair comparison. We report the results on YCB-V and LM-O, and on LM-O we only report the results with PBR training images following the standard BOP setting. As shown in Table 2, our method achieves the best accuracy in most metrics.

Running time analysis. We evaluate our method on a workstation with an NVIDIA RTX-3090 GPU and an Intel-Xeon CPU with 12 2.1GHz cores. We run most of the recent refinement methods on the same workstation. Since different refinement methods use very different strategies for pose initialization, we only report the running time of pose refining, as shown in Table 3. Our method takes only 17ms on average to process an object instance and is much more efficient than most refinement methods. Thanks to the removal of the need for RANSAC-PnP, our method is more than twice faster than PFA.

4.3. Ablation Study

Robustness to different pose initialization. Our pose refinement method is general and can be used with most pose methods with their results as pose initialization. To verify the generalization ability of our method, we evaluate our method with pose initialization from the results of WDR [21] and PoseCNN [50], respectively. For WDR, we

Method	LM	LM-O	YCB-V
WDR	60.2	37.9	27.5
w/ PFA	**95.8**	65.3	62.8
w/ **Ours**	**95.8**	**71.1**	**65.5**
PoseCNN	63.3	24.9	21.3
w/ PFA	99.2	60.5	61.9
w/ **Ours**	**99.3**	**66.4**	**70.5**

Table 4. **Comparison with different pose initialization.** We compare our refinement method with the baseline PFA with different pose initialization, including WDR and PoseCNN. Our method consistently outperforms PFA in each setting.
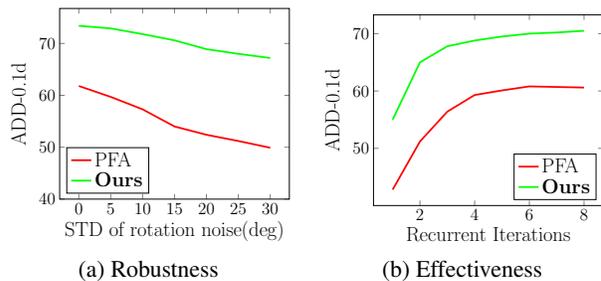

(a) Robustness (b) Effectiveness

Figure 6. **Ablation study on YCB-V. (a)** We compare our method and PFA with the results of PoseCNN as pose initialization with random pose errors in different levels. Our method is much more robust than PFA, especially in scenarios with heavy initialization noise. **(b)** We evaluate the methods with different recurrent iterations during inference, and our method outperforms PFA after only 2 iterations, and improves further with more iterations.

Shape-constraint lookup	Pose regressor	ADD 0.05d	ADD 0.1d
-	-	33.5	61.9
-	✓	34.0	63.6
✓	-	46.1	67.6
✓	✓	**50.4**	**70.5**

Table 5. **Ablation study of different components on YCB-V.** We evaluate the two key components of our method, including the shape-constraint lookup operation and the pose regressor. For the notation "-" for them, we use the standard constraint-free lookup operation based on the intermediate flow, and the RANSAC-PnP, respectively. The first row is the baseline PFA. Our shape-constraint lookup boosts the performance, and the pose regressor increases the performance further.

obtain its results trained only on the synthetic data, similar to PFA [16]. For PoseCNN, we use its pre-generated results, which was trained on real images. As shown in Table 4, our method improves the results of pose initialization significantly and outperforms PFA in most settings.

Furthermore, we study the robustness of our method with the results of PoseCNN as pose initialization with random pose errors in different levels, as shown in Fig. 6(a). Our method is much more robust than PFA. Especially, the performance drop of PFA with heavy noise can be nearly 11.9% in ADD-0.1d, and our method's accuracy only decreases by about 6.2% in the same condition and is still higher than that of PFA obtained with little initial noise.

Effect of different number of iterations. We evaluate our method with different number of iterations during inference. As shown in Fig. 6(b), our method performs on par with PFA after the very first iteration, and outperforms it significantly after only 2 iterations. Our method improves further with more iterations, and outperforms PFA by over 11.5% in the end.

Evaluation of different components. We study the effect of different components of our method, including the shape-constraint lookup operation guided by the pose-induced flow, and the pose regressor based on the intermediate flow, as shown in Table 5. The first row is the baseline method PFA, which does not have any constraints in the correlation map construction and relies on RANSAC-PnP. Our shape-constraint lookup operation boosts the performance, demonstrating the effectiveness of embedding targets' 3D shape. The RANSAC-PnP, even equipped with the shape-constraint lookup during the recurrent optimization, still suffers in producing accurate pose results, which is caused by the surrogate loss that does not directly reflect the final object pose. By contrast, our pose regressor is end-to-end trainable, which does not suffer from this problem and can benefit from simultaneously optimizing the optical flow and object pose.

Evaluation with different training data. To study the effect of different training data, we report the results of our method trained with four data settings, including pure PBR images, PBR images with additional 20 real images for each object ("PBR+20"), pure real images, and a mixture of all PBR images and real images. As shown in Table 6, more data generally results in more accurate pose estimates. While we report most results of our method trained only on the real images to be consistent with other methods. On the other hand, we find that, on LM, the models trained with only real images perform even better than those trained with a mixture of the real and PBR images, which we believe is caused by the distribution difference between the PBR and real data on LM. Note that the results of PFA are different from that in Table 1 since here we use the same results of PoseCNN as pose initialization for both methods for fair comparison. Nevertheless, our method consistently outperforms PFA in all different settings.

Training analysis. We study the properties of our method during training. We report the pose accuracy and flow loss for both the baseline PFA and our method in different steps

Dataset	LM		LM-O		YCB-V	
	0.05d	0.1d	0.05d	0.1d	0.05d	0.1d
PBR	65.5	95.0	27.6	50.9	8.8	28.9
	72.5	**96.8**	**28.9**	**52.9**	**10.9**	**36.5**
PBR+20	77.3	97.7	28.2	54.5	24.0	49.6
	81.5	**98.5**	**36.2**	**63.3**	**33.2**	**61.9**
Real	89.5	99.2	30.6	60.5	33.5	61.9
	92.9	**99.3**	**39.3**	**66.4**	**50.4**	**70.5**
Mixed	77.3	97.7	37.7	64.5	38.0	61.6
	90.9	**99.3**	**44.6**	**67.0**	**51.2**	**73.2**

Table 6. **Comparison with different training data.** We compare our method with the baseline PFA in four different data settings. In each setting, the **first row** is PFA, and the **second row** is ours. Our method outperforms PFA in all settings.
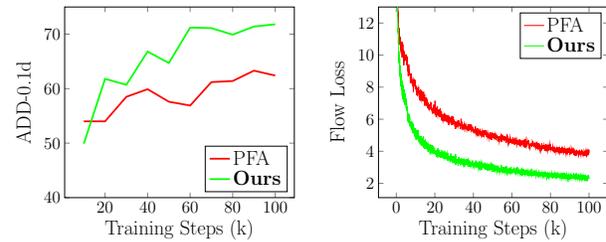

Figure 7. **Training analysis on YCB-V.** We report the pose accuracy and flow loss during training for both the baseline PFA and our method. Our method performs equally well as the fully-trained PFA after only 20k training steps, and outperforms it significantly with more training steps.
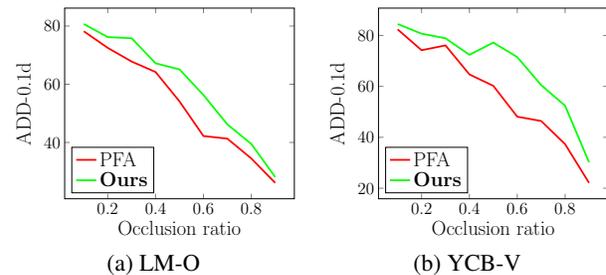

(a) LM-O   (b) YCB-V

Figure 8. **Performance with different occlusion ratios.** Our method consistently outperforms the baseline PFA in different occlusion ratios, demonstrating the effectiveness of our shape-constraint strategy.

during training, as shown in Fig. 7. At the beginning, our method performs less well than PFA. However, after only 20k training steps, our method outperforms PFA with the same training steps and performs equally well as the fully-trained PFA. Furthermore, our method produces much better results than PFA in both pose accuracy and flow loss with more training steps.

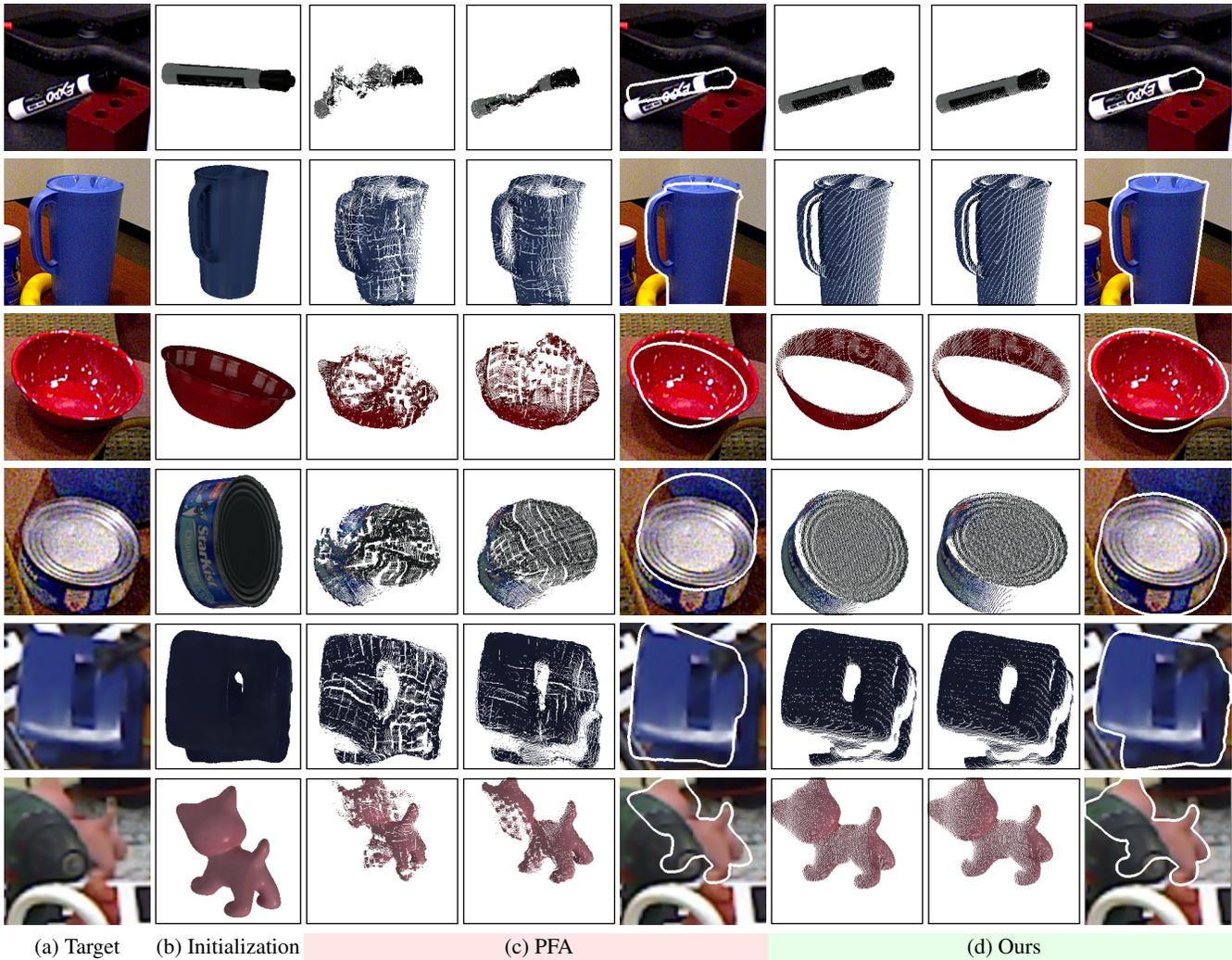| (a) Target	(b) Initialization	(c) PFA	(d) Ours

Figure 9. **Qualitative results.** We visualize the initialization by rendering the target according to the initial pose. For both PFA and our method, we show the flow wrap results with 2 and 8 iterations from left to right, respectively. PFA can not preserve the target's shape even after 8 iterations. By contrast, our method preserves the shape after every iteration and produces more accurate pose results by the end.

Performance with different occlusion ratios. We compare our method with the baseline PFA in scenarios with different occlusion ratios. As shown in Fig. 8, although the performance of both methods decreases with the increase of occlusion rations, our method is more robust than PFA and outperforms it in every setting of different occlusion ratio, either on LM-O or YCB-V, thanks to our shape-constraint design, which implicitly embeds the target's 3D shape information into our model and is more robust to occlusions. We show some qualitative results in Fig. 9.

5. Conclusion

We have introduced a shape-constraint recurrent matching framework for 6D object pose estimation. We have first analyzed the weaknesses of the standard optical flow net-

works and introduced a new matching framework that contains only all the 2D reprojections of the target's 3D shape in constructing the correlation map, which reduces the matching space significantly. Furthermore, we have proposed learning the object pose based on the current estimated flow and simultaneously optimizing the object pose and optical flow in an end-to-end recurrent manner. We have demonstrated the advantages of our method with extensive evaluation on three challenging 6D object pose datasets. It outperforms the state of the art significantly, and converges much more quickly.

References

[1] M.J. Black and P. Anandan. A Framework for the Robust Estimation of Optical Flow. In *International Conference on Computer Vision*, 1993. 1, 2

[2] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A Naturalistic Open Source Movie for Optical Flow Evaluation. In *European Conference on Computer Vision*, 2012. 1

[3] Ming Cai and Ian Reid. Reconstruct Locally, Localize Globally: A Model Free Method for Object Pose Estimation. In *Conference on Computer Vision and Pattern Recognition*, 2020. 2

[4] Bo Chen, Alvaro Parra, Jiewei Cao, Nan Li, and Tat-Jun Chin. End-to-End Learnable Geometric Vision by Backpropagating PnP Optimization. In *Conference on Computer Vision and Pattern Recognition*, 2020. 2

[5] Hansheng Chen, Pichao Wang, Fan Wang, Wei Tian, Lu Xiong, and Hao Li. EPro-PnP: Generalized End-to-End Probabilistic Perspective-n-Points for Monocular Object Pose Estimation. In *Conference on Computer Vision and Pattern Recognition*, 2022. 2

[6] Qifeng Chen and Vladlen Koltun. Full Flow: Optical Flow Estimation by Global Optimization over Regular Grids. In *Conference on Computer Vision and Pattern Recognition*, 2016. 2

[7] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. *arXiv preprint arXiv:1409.1259*, 2014. 3, 4

[8] Alvaro Collet, Manuel Martinez, and Siddhartha S Srinivasa. The MOPED Framework: Object Recognition and Pose Estimation for Manipulation. *The international journal of robotics research*, 30(10):1284–1306, 2011. 1

[9] Maximilian Denninger, Martin Sundermeyer, Dominik Winkelbauer, Youssef Zidan, Dmitry Olefir, Mohamad Elbadrawy, Ahsan Lodhi, and Harinandan Katam. BlenderProc. *arXiv preprint arXiv:1911.01911*, 2019. 1, 5

[10] Yan Di, Fabian Manhardt, Gu Wang, Xiangyang Ji, Nassir Navab, and Federico Tombari. SO-Pose: Exploiting Self-Occlusion for Direct 6D Pose Estimation. In *International Conference on Computer Vision*, 2021. 2, 5, 6

[11] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning Optical Flow with Convolutional Networks. In *International Conference on Computer Vision*, 2015. 2

[12] Rasmus Laurvig Haugaard and Anders Glent Buch. SurfEmb: Dense and Continuous Correspondence Distributions for Object Pose Estimation with Learnt Surface Embeddings. In *Conference on Computer Vision and Pattern Recognition*, 2022. 5, 6

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Conference on Computer Vision and Pattern Recognition*, 2016. 2

[14] Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary Bradski, Kurt Konolige, and Nassir Navab. Model Based Training, Detection and Pose Estimation of Texture-Less 3D Objects in Heavily Cluttered Scenes. In *Asian Conference on Computer Vision*, 2012. 2, 5

[15] Tomáš Hodaň, Martin Sundermeyer, Bertram Drost, Yann Labbé, Eric Brachmann, Frank Michel, Carsten Rother, and Jiří Matas. BOP Challenge 2020 on 6D Object Localization. *European Conference on Computer Vision Workshops*, 2020. 5, 6

[16] Yinlin Hu, Pascal Fua, and Mathieu Salzmann. Perspective Flow Aggregation for Data-Limited 6D Object Pose Estimation. In *European Conference on Computer Vision*, 2022. 1, 2, 3, 4, 6

[17] Yinlin Hu, Pascal Fua, Wei Wang, and Mathieu Salzmann. Single-Stage 6D Object Pose Estimation. In *Conference on Computer Vision and Pattern Recognition*, 2020. 1, 2, 4

[18] Yinlin Hu, Joachim Hugonot, Pascal Fua, and Mathieu Salzmann. Segmentation-Driven 6D Object Pose Estimation. In *Conference on Computer Vision and Pattern Recognition*, 2019. 2

[19] Yinlin Hu, Yunsong Li, and Rui Song. Robust Interpolation of Correspondences for Large Displacement Optical Flow. In *Conference on Computer Vision and Pattern Recognition*, 2017. 2

[20] Yinlin Hu, Rui Song, and Yunsong Li. Efficient Coarse-to-Fine Patchmatch for Large Displacement Optical Flow. In *Conference on Computer Vision and Pattern Recognition*, 2016. 2

[21] Yinlin Hu, Sebastien Speierer, Wenzel Jakob, Pascal Fua, and Mathieu Salzmann. Wide-Depth-Range 6D Object Pose Estimation in Space. In *Conference on Computer Vision and Pattern Recognition*, 2021. 2, 3, 6

[22] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks. In *Conference on Computer Vision and Pattern Recognition*, 2017. 2

[23] Shun Iwase, Xingyu Liu, Rawal Khirodkar, Rio Yokota, and Kris M. Kitani. RePOSE: Fast 6D Object Pose Refinement via Deep Texture Rendering. In *International Conference on Computer Vision*, 2021. 2, 4, 5, 6

[24] Shihao Jiang, Dylan Campbell, Yao Lu, Hongdong Li, and Richard Hartley. Learning To Estimate Hidden Motions With Global Motion Aggregation. In *International Conference on Computer Vision*, 2021. 2

[25] Alexander Krull, Eric Brachmann, Frank Michel, Michael Ying Yang, Stefan Gumhold, and Carsten Rother. Learning Analysis-by-Synthesis for 6D Pose Estimation in RGB-D Images. In *International Conference on Computer Vision*, 2015. 2, 5

[26] Y. Labbe, J. Carpentier, M. Aubry, and J. Sivic. CosyPose: Consistent Multi-View Multi-Object 6D Pose Estimation. In *European Conference on Computer Vision*, 2020. 2, 5, 6

[27] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An Accurate o (n) Solution to the PnP Problem. *International Journal of Computer Vision*, 81(2):155–166, 2009. 1, 2

[28] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. DeepIM: Deep Iterative Matching for 6D Pose Estimation. In *European Conference on Computer Vision*, 2018. 2, 4, 6

[29] Zhigang Li, Gu Wang, and Xiangyang Ji. CDPN: Coordinates-Based Disentangled Pose Network for Real-Time RGB-Based 6-DoF Object Pose Estimation. In *International Conference on Computer Vision*, 2019. 2

[30] Lahav Lipson, Zachary Teed, and Jia Deng. Raft-Stereo: Multilevel Recurrent Field Transforms for Stereo Matching. In *International Conference on 3D Vision*, 2021. 2

[31] Lahav Lipson, Zachary Teed, Ankit Goyal, and Jia Deng. Coupled Iterative Refinement for 6D Multi-Object Pose Estimation. In *Conference on Computer Vision and Pattern Recognition*, 2022. 1, 2, 5, 6

[32] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*, 2018. 5

[33] Fabian Manhardt, Wadim Kehl, Nassir Navab, and Federico Tombari. Deep Model-Based 6D Pose Refinement in RGB. In *European Conference on Computer Vision*, 2018. 2

[34] Eric Marchand, Hideaki Uchiyama, and Fabien Spindler. Pose Estimation for Augmented Reality: a Hands-on Survey. *IEEE transactions on visualization and computer graphics*, 22(12):2633–2651, 2015. 1

[35] Nikolaus Mayer, Eddy Ilg, Philip Häusser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation. In *Conference on Computer Vision and Pattern Recognition*, 2016. 1

[36] Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. PVNet: Pixel-Wise Voting Network for 6DoF Pose Estimation. In *Conference on Computer Vision and Pattern Recognition*, 2019. 2, 5, 6

[37] Mahdi Rad and Vincent Lepetit. Bb8: A Scalable, Accurate, Robust to Partial Occlusion Method for Predicting the 3D Poses of Challenging Objects Without using Depth. In *International Conference on Computer Vision*, 2017. 2

[38] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3D Deep Learning with Pytorch3D. *arXiv preprint arXiv:2007.08501*, 2020. 1, 4

[39] Jerome Revaud, Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. Epicflow: Edge-Preserving Interpolation of Correspondences for Optical Flow. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1164–1172, 2015. 2

[40] Petter Risholm, Peter Ørnulf Ivarsen, Karl Henrik Haugholt, and Ahmed Mohammed. Underwater Marker-Based Pose-Estimation with Associated Uncertainty. In *International Conference on Computer Vision Workshops*, 2021. 2

[41] Leslie N Smith and Nicholay Topin. Super-Convergence: Very Fast Training of Neural Networks using Large Learning Rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, volume 11006, page 1100612. International Society for Optics and Photonics, 2019. 5

[42] Yongzhi Su, Mahdi Saleh, Torben Fetzer, Jason Rambach, Nassir Navab, Benjamin Busam, Didier Stricker, and Federico Tombari. ZebraPose: Coarse to Fine Surface Encoding for 6DoF Object Pose Estimation. In *Conference on Computer Vision and Pattern Recognition*, 2022. 2

[43] Xiuchao Sui, Shaohua Li, Xue Geng, Yan Wu, Xinxing Xu, Yong Liu, Rick Siow Mong Goh, and Hongyuan Zhu. CRAFT: Cross-Attentional Flow Transformers for Robust Optical Flow. In *Conference on Computer Vision and Pattern Recognition*, 2022. 2

[44] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for Optical Flow using Pyramid, Warping, and Cost Volume. In *Conference on Computer Vision and Pattern Recognition*, 2018. 2

[45] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel. Implicit 3D Orientation Learning for 6D Object Detection from RGB Images. In *European Conference on Computer Vision*, 2018. 2

[46] Zachary Teed and Jia Deng. RAFT: Recurrent All-Pairs Field Transforms for Optical Flow. In *European Conference on Computer Vision*, 2020. 1, 2, 3, 4

[47] 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. In *Conference on Computer Vision and Pattern Recognition*, 2019. 2

[48] Gu Wang, Fabian Manhardt, Federico Tombari, and Xiangyang Ji. GDR-Net: Geometry-Guided Direct Regression Network for Monocular 6D Object Pose Estimation. In *Conference on Computer Vision and Pattern Recognition*, 2021. 2, 5

[49] Wei Wang, Zheng Dang, Yinlin Hu, Pascal Fua, and Mathieu Salzmann. Robust Differentiable SVD. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 4

[50] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes. In *Robotics: Science and Systems*, 2018. 2, 3, 5, 6

[51] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated Residual Transformations for Deep Neural Networks. In *Conference on Computer Vision and Pattern Recognition*, 2017. 2

[52] Xu Yan, Lin Junyi, Zhang Guofeng, Wang Xiaogang, and Li Hongsheng. RNNPose: Recurrent 6-DoF Object Pose Refinement with Robust Correspondence Field Estimation and Pose Optimization. In *Conference on Computer Vision and Pattern Recognition*, 2022. 1, 2, 4, 5, 6

[53] Sergey Zakharov, Ivan Shugurov, and Slobodan Ilic. DPOD: 6D Pose Object Detector and Refiner. In *International Conference on Computer Vision*, 2019. 2

[54] Yi Zhou, Connelly Barnes, Lu Jingwan, Yang Jimei, and Li Hao. On the Continuity of Rotation Representations in Neural Networks. In *Conference on Computer Vision and Pattern Recognition*, 2019. 4