WeLSA: Learning To Predict 6D Pose From Weakly Labeled Data Using Shape Alignment

Shishir Reddy Vutukur[1,2], Ivan Shugurov[1,2], Benjamin Busam[1], Andreas Hutter[2], and Slobodan Ilic[1,2]

[1] Technical University of Munich
[2] Siemens Technology

Abstract. Object pose estimation is a crucial task in computer vision and augmented reality. One of its key challenges is the difficulty of annotation of real training data and the lack of textured CAD models. Therefore, pipelines which do not require CAD models and which can be trained with few labeled images are desirable. We propose a weakly-supervised approach for object pose estimation from RGB-D data using training sets composed of very few labeled images with pose annotations along with weakly-labeled images with ground truth segmentation masks without pose labels. We achieve this by learning to annotate weakly-labeled training data through shape alignment while simultaneously training a pose prediction network. Point cloud alignment is performed using structure and rotation-invariant feature-based losses. We further learn an implicit shape representation, which allows the method to work without the known CAD model and also contributes to pose alignment and pose refinement during training on weakly labeled images. The experimental evaluation shows that our method achieves state-of-the-art results on LineMOD, Occlusion-LineMOD and TLess despite being trained using relative poses and on only a fraction of labeled data used by the other methods. We also achieve comparable results to state-of-the-art RGB-D based pose estimation approaches even when further reducing the amount of unlabeled training data. In addition, our method works even if relative camera poses are given instead of object pose annotations which are typically easier to obtain.

Keywords: object pose estimation, shape alignment, weak supervision.

1 Introduction

6D object pose estimation is a crucial task in computer vision and robotics with applications for robotic grasping [37], augmented reality [3], and autonomous driving [23]. 6D pose estimation comprises estimating rotation and translation of an object from the camera coordinate system to the object coordinate system.

With the rise of deep learning, we witnessed a rapid improvement of learning-based pose estimation from RGB images [38,27,36,40,33,40,32,18,30] as well as from RGBD data [5,9,34,8]. A common problem of deep learning-based methods is their dependence on access to a large number of images with ground

truth labels. This is even more problematic for 6D pose estimation because pose labels cannot be manually annotated and, thus, require sophisticated labeling pipelines [17]. A possible solution would be to simulate synthetic scenes and train networks on rendered images. This, however, requires the availability of textured CAD models and significant time for simulation and rendering. As we can see from the results of the BOP Challenge [13], methods trained on synthetic data still lag behind their counterparts trained on real or mixed data. For RGBD-based methods [9,34,8], training on real data is also prevalent due to difficulty of realistic simulation of depth sensor noise [14]. To overcome these problems, we propose a novel training pipeline that uses a combination of a very small number of images with object pose or relative camera transformation labels and weakly-labeled images with only 2D segmentation masks. Besides, our training pipeline does not require access to CAD models.

Some approaches are proposed which do not require CAD model. For example, RLLG [4] proposes a training pipeline without CAD model by using labeled training RGB images to regress 2D-3D correspondences by implicitly reconstructing the model using multiview supervision. Similarly, we establish 3D-3D correspondences without CAD model by reconstructing the model implicitly by establishing correspondences in local reference frame of one of the training samples. RLLG [4] requires labeled data for the entire training data even though it doesn't require CAD model. To overcome the need of real training data and CAD model, Latent Fusion [26] proposes a generalized reconstruction pipeline from few labeled views of the unseen object which is then used to estimate pose for a given segmented image by iterative feature alignment. Even though it is a generalized network, the performance of Latent Fusion is not comparable to fully supervised approaches. Without requiring CAD model, our approach uses a combination of very few labeled data and weakly-labeled data and achieves accuracy on par with state-of-the-art supervised RGB-D approaches.

In this paper, we propose a novel deep learning architecture for pose estimation and a novel weakly supervised training pipeline. Since we don't use a CAD model, we use the reference frame of one of the labeled training sample as the canonical reference frame. During training, the network takes as input a point cloud of the object and outputs both implicit shape representation of the object and dense 3D-3D correspondences in the canonical reference frame defined by one of the labeled samples. Note that, since the CAD models are not available, the network is trained to learn the implicit shape representation of the object which is used for shape alignment and pose refinement. Thus, during training, the network learns to reconstruct the object. The predicted implicit shape is converted to the triangular mesh for ICP [1] refinement. During inference, we estimate pose using estimated 3D-3D correspondences and refine the pose using the shape reconstructed during training. The proposed training pipeline consists of three stages: 1) fully supervised training on fully labeled images; 2) fine-tuning on weakly labeled images; 3) auto-labeling weakly labeled images and then training on them in a fully supervised manner. The architecture and the training pipeline were evaluated on LineMOD, LineMOD-Occlusion and TLess

datasets and showed very competitive results despite having been trained using a fraction of labeled training images compared to other approaches.

To this end, we contribute:

1. A novel RGB-D pose estimation architecture for 6D pose estimation from point clouds and color image, which fuses features from color, depth and normals and performs simultaneous object reconstruction and 6D pose estimation from very few images with pose labels. Two key components can be distinguished: We propose a feature decoder to improve pose estimation by associating pose invariant features in the shape alignment pipeline, and a shape network to learn the shape of the object which contributes to pose refinement and shape alignment.

2. A weakly supervised novel pose estimation pipeline that simultaneously learns a pose estimation network and auto-label weakly-labeled data with segmentation masks using labeled samples with relative pose labels leveraging shape alignment and feature alignment losses without the need of CAD model. The pipeline achieves accuracy on par with fully supervised RGB-D approaches.

3. Novel shape-based and feature-based rotation invariant losses, are proposed which are suitable for weakly supervised training. The training setup makes it suitable to handle symmetric objects as our approach is shape alignment based. Our approach works even if only the relative poses are available instead of absolute pose for few samples.

2 Related Work

Advances in deep learning enabled a rapid development of pose estimation methods from monocular RGB images. One line of work presented in [21,38,5,36,18] treat pose estimation as a regression problem by directly predicting rotation and translation of the object. [40,32,27,20,22,30] treat it as a correspondence estimation problem by regressing 2D-3D correspondences between image pixels and the 3D model of the object. The 6D pose is then estimated using PnP [19] and RANSAC. Alternatively, [33,28,16] estimate a predefined set of sparse keypoints instead of dense correspondences, which has proven to be more robust to occlusions. We have opted for a dense correspondence-based approach as dense 3D-3D correspondences are required for shape alignment. Despite that, our approach is robust to occlusions on par with keypoint based approaches.

Pose estimation from RGBD data, which was traditionally solved using geometric methods, such as Point Pair Features [6], has also attracted attention from the deep learning community. Most of the methods, such as [10,34,32,8], rely on a common idea of fusing the features estimated separately from RGB using a CNN and from Depth using a variant of Pointnet [29]. Similarly to pose estimation from RGB, some methods [10,32,8] are trained to predict 3D-3D correspondences, while [34] directly outputs rotation and translation.

There have been several attempts to lessen the dependence of pose estimation methods on hard-to-annotate pose labels. [35,39,31] propose to pre-train a model on synthetic images with full annotations and then use the network to label real images, which are then used for training. A CAD-free and 6D labels-free method

that uses known bounding boxes, object size and multi-view constraints to train a direct pose regression was proposed in[21]. [41] proposes an approach to estimate domain invariant keypoints by training with labeled synthetic images and real data without pose labels. We propose a weakly-supervised approach using few labeled samples and weakly-labeled data to train a pose estimation pipeline and to label weakly-labeled data without requiring CAD model.

3 Method

3.1 Architecture

The proposed network takes a segmented point cloud and a segmented RGB image of the detected object and outputs 1) dense 3D-3D correspondences between the point cloud and the object model; 2) decoded rotation-invariant features; 3) implicit shape representation of the object. Figure 1 summarizes the architecture. Two separate encoders are used to map the input point cloud and the input image to their respective latent vector representations. Both vectors are stacked to form the vector z, which is used by the decoders. Vector z along with each 3D point in the point cloud are passed through two MLP decoders to predict point-wise 3D correspondences and point-wise feature values. The predicted correspondences are passed through an MLP shape network to predict their respective SDF values. The feature decoder and shape network are used only during the training of the network. We describe different components of our network in detail in the following section.

Encoders The encoder E consists of two main components. A PointNet based encoder [24] is used to extract features from the point cloud, $X \in \mathbb{R}^{K\times 3}$. Each x of the K points is concatenated with its corresponding normal, n, and color, c, of the normal vector, N, and color vector, C, to use as input to point cloud encoder. The network extracts features from the input point cloud and its features, normals, and color, and outputs an intermediate latent vector.

A ResNet CNN [7] encoder extracts features from an RGB image, I. The CNN predicts another intermediate latent vector from a color image. The intermediate latent vectors from Pointnet and CNN are concatenated to pass through a fully connected layer to produce a global latent vector, z of length d.

The full encoder is mathematically defined as:

$$z = E(I, X, N, C) \tag{1}$$

Correspondence Decoder Correspondence decoder is an MLP with 7 fully connected layers which takes the global latent vector z and a 3D point x and predicts its corresponding 3D point $x_C \in \mathbb{R}^3$ in the canonical reference frame. We denote the decoder with D which is mathematically defined as:

$$x_C = D(z, x) \tag{2}$$

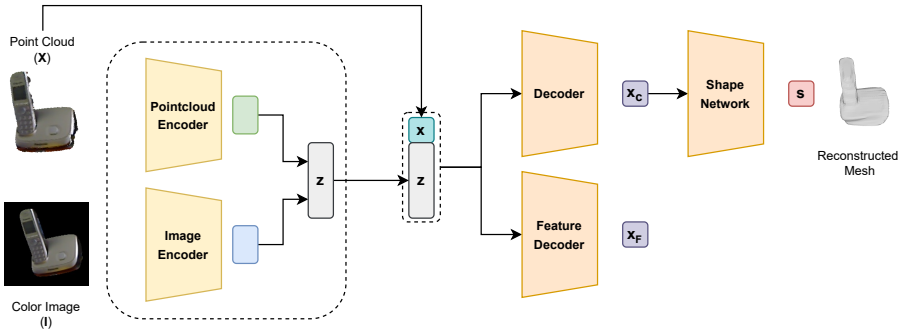

Fig. 1: Architecture: Our network takes in a segmented point cloud and segmented color image to predict 3D correspondences which are used to estimate 6D pose. A feature decoder and shape network are employed only during training to improve shape alignment. The encoder combines latent vectors from the point cloud encoder and image encoder to predict z. Each 3D point, x, in the point cloud, X, is concatenated with z to predict point-wise correspondences, x_C and pose invariant point-wise features, x_F using Correspondence decoder (Decoder) and Feature decoder respectively. The predicted correspondences are passed through the shape network to predict point-wise SDF value, s, which is used to improve shape alignment. The shape network learns implicit shape which is used to reconstruct shape used for pose refinement during training.

Feature Decoder The feature decoder, F, has the same architecture as the correspondence decoder and has the same input. It takes a concatenated vector of z and x to predict a per-point pose invariant feature, x_F. The decoder serves an auxiliary role and is used only during training. The motivation behind the feature decoder is to incorporate pose-invariant features. This allows us to define loss functions even for unlabeled data. Since both decoders share the same input, losses formulated on features impact the correspondence prediction. By associating shape alignment with feature loss, we are able to avoid false matches where the structure is similar but features are different. This is illustrated in Figure 4. In addition to explicit loss terms on shape alignment related feature losses, predicting pose invariant features for each point is an auxiliary task that improves the encoder performance.

$$x_F = F(z, x) \tag{3}$$

Shape Decoder The shape decoder SN, similarly to DeepSDF [25], learns implicit shape representation in the canonical object pose space. It consists of an MLP with 7 fully connected layers which takes in a correspondence 3D point, x_C, to predict its signed distance function value, s. The purpose of the decoder is twofold. First, it allows us to reconstruct the object during training and avoid

using CAD models. Second, it allows for shape alignment during training.

$$s = SN(x_C) \tag{4}$$

3.2 Training Pipeline and Loss Functions

We aim to estimate the 6D pose of an object from an RGB-D image which involves estimating 3D rotation and 3D translation. We train a network to estimate dense 3D-3D correspondences, which are then used to estimate the 6D pose. To facilitate shape alignment and pose refinement, an implicit object shape is predicted. The model is trained in a weakly-supervised manner on a mix of weakly labeled data and a very small amount of fully labeled data. Fully-labeled images are provided with ground truth poses and segmentation masks. Only object masks are available for weakly-labeled images. We refer to the fully-labeled samples as fewshot samples as we use very few of them in our pipeline.

We employ a three-stage pipeline to train the network. In Stage 1, all parts of the network are trained in a fully supervised manner on the fewshot samples. The shape encoder, trained separately from the rest of the network, implicitly reconstructs the object shape. The pre-trained shape encoder is frozen after this stage. In Stage 2, we train the network along with the feature decoder using both fewshot data and weakly-labeled images using shape and feature deviation losses. We also use the frozen shape network to facilitate shape alignment for weakly-labeled samples. In Stage 3, we first estimate poses for weakly-labeled data using the trained network from Stage 2. The estimated poses are refined using ICP with the triangular mesh reconstructed from the shape network. The network is then trained in a fully supervised manner using the refined poses.

We use the following terminology in the rest of the paper. The camera reference frame is the local reference frame in which each image is observed which is specific to each image. The canonical reference frame is the reference frame to which we find the pose for all the samples. The reference frame of the first sample in the fewshot samples is treated as the canonical reference frame. We use relative pose between frames to generate pose for fewshot samples to the canonical reference frame which is the camera reference frame of the first sample. We estimate pose for all our samples treating the first sample in fewshot samples as the canonical reference frame.

Stage 1 In this stage, we train the network with fewshot samples for which we know the pose labels. It is depicted in Figure 2. We use correspondence regression loss, L_C, to penalize L2 distance between ground-truth correspondences, X_G, and predicted correspondences. The ground-truth correspondences are obtained by transforming the input point cloud with ground truth rotation, R_G, and translation, T_G, to the canonical reference frame. Using relative pose between frames, we find the transformation to the first frame from every frame and use it as a ground truth pose. We train the encoder-decoder network using loss function, L_C which is defined as follows:

$$X_G = XR_G + T_G \quad L_C = \|X_C - X_G\|_2 \tag{5}$$

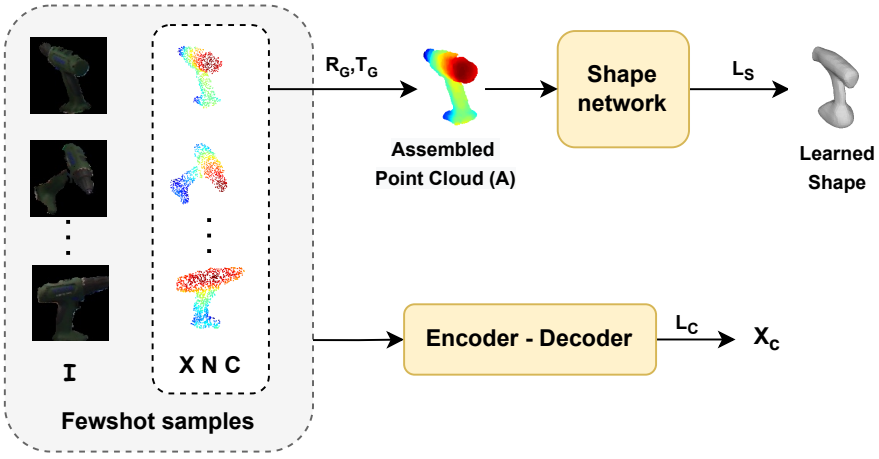

Fig. 2: Pipeline of Stage 1: Few shot samples comprises of RGB image, I, and point cloud, X, and its corresponding normals, N, and color, C. We use ground truth relative poses R_G, T_G to transform all the point clouds to the first image reference frame to get an assembled point cloud A. We train implicit shape network using A and loss, L_S. We also train the encoder-decoder using ground truth relative poses for fewshot samples using loss L_C.

We employ SDF loss, L_S, with the assembled point cloud, A and its normals, A_N to train the shape network. The assembled point cloud is obtained by combining the ground truth correspondences of the few shot point clouds. Since the assembled point cloud is supposed to lie on the object's surface, the SDF value should be zero for these points. To generate more samples to train the shape network and to learn shape better, we estimate point cloud normals and translate assembled point cloud along their per-point normals, A_N, by a scaling factor, P. For these samples, the SDF value should be equal to the scaling factor, P, and hence we formulate loss with the scaling factor as follows:

$$A' = A + PA_N$$
$$S = SN(A')$$
$$L_S = \|S - P\|_1$$

Stage 2 In this stage, the network is trained on both weakly-labeled and fewshot samples. It is depicted in Figure 3. We freeze the shape network because it has already learned the signed distance function of the shape and is used to formulate shape deviation loss for weakly-labeled data. We present the loss functions used in this stage below.

We predict color as a feature using the feature decoder. We use color as pose-invariant feature to improve pose alignment. We employ a feature loss, L_F,

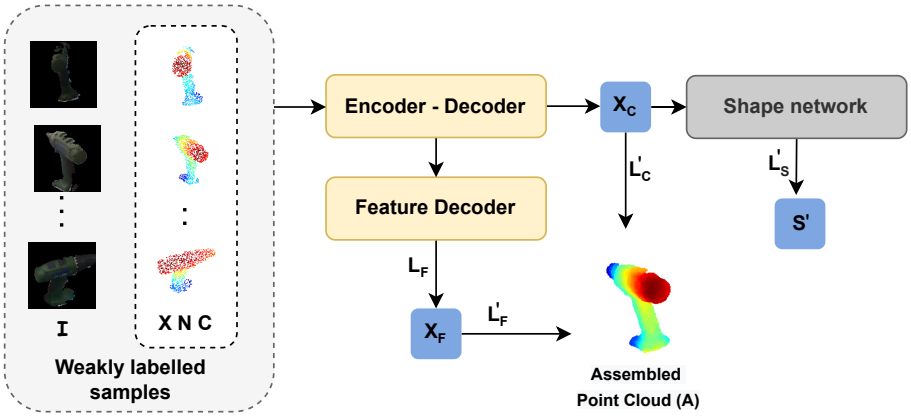

Fig. 3: Pipeline of Stage 2 : We train the encoder-decoder, feature decoder using a frozen shape network and assembled point cloud. Chamfer loss, L'_C, photometric loss, L'_F are formulated on correspondences, X_C and predicted features X_F. Correspondences are also passed through the frozen shape network to predict SDF, S' to formulate SDF loss, L'_S.

between predicted feature, X_F, and color of the input point cloud, C. This loss is applied to all the samples.

$$L_F = \|X_F - C\|_2 \tag{6}$$

We employ different loss functions for weakly-labeled samples as we do not know the pose for these samples. We employ a rigidity preserving loss, L'_R, between the input point cloud and predicted correspondences to predict the correspondences with the same structure as the input point cloud. We formulate a loss to preserve the inter-point euclidean distance between each pair of points between input point cloud, X, and predicted correspondences, X_C.

$$\begin{aligned} d^{ij} &= \|x^i - x^j\|_2 \\ d_C^{ij} &= \|x_C^i - x_C^j\|_2 \\ L'_R &= \sum_j \sum_i \|d^{ij} - d_C^{ij}\|_2 \end{aligned} \tag{7}$$

We estimate rotation and translation between input point cloud, X, and predicted correspondences, X_C, using the differentiable Kabsch algorithm [15]. So, the loss formulated using the rotation and translation can be back propagated through the network. We transform input normals, N, and point cloud, X, to canonical reference frame using estimated rotation and translation as follows:

$$N_T = NR \quad X_T = XR + T \tag{8}$$

We transform the input point cloud using R, T to constrain the shape alignment to rigid transform by parameterizing loss over R, T instead of predicted

correspondences directly. If we formulate the chamfer shape loss with predicted correspondences, the degenerate solution is that all predicted correspondences can collapse to one point on the target shape. Although we have a structure preserving loss, L'_R, transforming input point cloud and formulating loss on R and T eliminates the sensitivity of network on weights assigned to structure preserving and shape aligning losses. Since we employ structure preserving loss even for noisy point clouds and parameterize shape based losses over rigid transformation, the entire point cloud undergoes a rigid transformation avoiding the need to include RANSAC for correspondence filtering.

To register weakly-labeled samples to the canonical reference frame, we employ chamfers distance, L'_C, to register the predicted correspondences with the assembled point cloud. We employ chamfers distance between assembled point cloud, A, its colors, A_C, its normals, A_N, and transformed input, X_T, predicted color, X_F, transformed normals, N_T.

Chamfer distance penalizes the structure deviation by penalizing the distance between the closest points on a target and source shape. We further combine features and correspondences by penalizing the feature deviation among the closest points in addition to structure deviation. We use color and normal features along with 3D points in chamfer distance to align the point clouds better. To propagate loss to the encoder through the features based on chamfer distance, we need to associate the feature loss to the pose. We achieve this by adding a feature decoder, which uses the same input as the correspondence decoder. Since we use same the latent space for both correspondence decoder and feature decoder, the pose alignment is impacted by losses formulated on correspondences as well as the features. Thus, we formulate feature loss on colors to propagate loss to improve the encoder to predict better correspondences and align the shapes better. The photometric consistency loss, L'_F, penalizes the color deviation between closest points on the transformed input cloud and the assembled point cloud.

We estimate rotation between the input point cloud and correspondences to rotate normals to the canonical reference frame. Losses formulated on rotated normals, N_T, propagates through rotation matrix to correspondences and to the network. The normal consistency loss, L'_N, penalizes normal deviation between closest points on the transformed input cloud and the assembled point cloud.

$$\begin{aligned} idx &= \arg\min_{x \in A} ||x - x_T||_2 \\ L'_C &= ||A[idx] - x_T||_2 \\ L'_F &= ||A_C[idx] - x_F||_2 \\ L'_N &= ||A_N[idx] - n_T||_2 \end{aligned} \tag{9}$$

By adding features into the pipeline and formulating loss function combining correspondences and features, the network automatically learns to predict the pose better as it impacts both the feature and correspondence decoder.

We employ another shape penalizing loss formulated using the frozen shape network. The predicted correspondences, X_C, from the weakly-labeled data are passed through the shape network to predict SDF values, S, for the correspon-

dences. Since the correspondences are supposed to lie on the shape, the predicted SDF value should be 0 for the correspondences. Basically, the predicted SDF value measures how far the predicted correspondence is from the object surfaces. In the ideal case, when the correspondence is on the surface, SDF is 0. We formulate shape deviation SDF loss, L'_S, as follows:

$$L'_S = \|S\|_1 \tag{10}$$

The total loss, L_2, to train the stage 2 with coefficients β_1 and β_2 is as follows:

$$L_2 = \beta_1(L_C + L_F) + \beta_2(L'_R + L'_C + L'_F + L'_N + L'_S) \tag{11}$$

Stage 3 We train the network using the above losses until convergence in Stage 2. We observe that the predictions still need refinement to align perfectly with the shape. To further refine the pose predictions, we employ an ICP-based refinement pipeline to achieve exact alignment. We reconstruct the surface mesh of the object from the Shape network using the marching cubes algorithm. We project the mesh using the estimated pose from the predicted correspondences and then perform ICP with the input point cloud to find better alignment. After extracting the refined pose for all the weakly labeled samples, we employ correspondence loss, L_C, with the estimated refined poses as ground truth poses for weakly-labelled samples to improve the accuracy further. We train the third stage using estimated refined pose, R', T', for weakly-labelled data as follows:

$$X'_G = XR' + T' \quad L_3 = \|X_C - X'_G\|_2 \tag{12}$$

4 Results

We evaluate our approach on LineMOD [11], LineMOD-Occlusion [2] and TLess [12]. We employ ADD/ADD-S metric [11] for LineMOD, LineMOD-Occlusion and AR metric of BOP Challenge[13] for TLess. We demonstrate the effectiveness of our approach by comparing results we attained using a fraction of total data as labeled data in contrast to other approaches using full labeled data.

4.1 Training Data

We use ground truth masks to extract tight image patches containing the object and point clouds corresponding to the foregrounds of the object. For LineMOD, we sample 15% of data as training data similar to other approaches. Of the sampled training data, we consider ground truth training samples(1%) from the sampled data in a way that they cover the object from different viewpoints which are far apart from each other so that they cover the object to the maximal extent. This is essential for our approach as we need an approximate point cloud of the object from the ground truth samples. The rest (14%) is treated as weakly-labeled data with only segmentation masks. If the absolute pose is given for the dataset, we use the first image in the fewshot samples as the canonical reference frame and convert the absolute poses of other fewshot samples to relative poses.

Table 1: Results on LineMOD dataset along with state of the art RGB and RGBD approaches. * denotes discrete symmetric objects. GT data and Total Data refer to the amount of ground truth data and total data used for training respectively. ICP refers to results with ICP refinement using the reconstructed model. CAD refers to training data synthetically generated using a CAD model.

Method/ Object	**Dpod**		**Dpodv2**		**PVN3D**	**G2L-Net**		**Ours**				
RGB	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Depth	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
GT Data	-	15%	-	15%	15%	5%	15%	1%	1%	1%	15%	15%
Total Data	-	15%	-	15%	15%	5%	15%	5%	15%	15%	15%	15%
ICP	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✓
CAD	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗
Ape	37.2	53.2	62.14	80	97.3		96.8	94.2	97.2	98.2	99.1	99.3
Benchvice	66.7	95.3	88.39	99.7	99.7		9.1	94.9	97.9	99.9	99.5	99.5
Camera	24.2	90.3	92.51	99.2	99.6		98.2	94.1	97.9	99.3	98.8	98.8
Can	52.5	94.1	96.6	99.6	99.5		98	96.5	97.7	98.7	99.7	100
Cat	32.3	60.3	86.17	95.1	99.8		99.2	99.1	99.3	99.3	99.3	99.3
Driller	66.6	97.7	90.15	98.9	99.3		99.8	89.5	99.5	99.5	99.5	99.5
Duck	26.1	66	54.86	79.5	98.2		97.7	84.3	95.4	97.1	96.8	98.8
Egg box*	73.3	99.7	98.64	99.6	99.8		100	99.4	100	100	100	100
Glue*	74.9	93.8	95.4	99.8	100		100	99.9	99.8	99.8	99.9	99.9
Puncher	24.5	65.8	27	72.3	99.9		99	87.7	96.6	98.8	98.6	99
Iron	85	99.8	98.2	99.4	99.7		99.3	94.1	95.4	99.3	100	100
Lamp	57.2	88.1	91	96.3	99.8		99.5	90.2	99.1	99.1	99.1	99.1
Phone	29	74.2	74.3	96.8	99.5		98.9	98.5	99.9	99.9	99.9	99.9
Average	50	82.9	81.2	93.5	99.4	88.5	98.7	**94.03**	**98.1**	**99.1**	**99.2**	**99.4**

4.2 LineMOD Dataset

Most approaches use 15% of the total data as training data. Our training data consists of 1% samples with ground truth poses and 14% samples without known poses. We train DpodV2[32] segmentation network with 15% ground truth data whereas the pose pipeline uses 1% ground truth data. During training, the pose accuracy of labels generated for weakly-labeled training samples after stage 2 is 90.6% without ICP and 99.2% with ICP. The ICP accuracy indicates the efficiency of our labeling pipeline and the gap in accuracy justifies the need for refinement in stage 3. From table 1, we are able to achieve SOTA results using a fraction of ground truth pose data. We observe an improvement of 1.0% with ICP refinement (99.1%) using the generated mesh from shape network. We achieve closer to benchmark accuracy when we use 15% ground truth data (99.2%).

4.3 LineMOD-Occlusion Dataset

On LineMOD-Occlusion, we achieve close to SOTA results even though we train on data obtained from the LineMOD instead of LineMOD-Occlusion and with

Table 2: Results on LineMOD-Occlusion dataset along with state of the art RGB [38,40,33] and RGBD [9,8] approaches.

Method	**PoseCNN**	**Dpod**	**Hybrid Pose**	**PVN3D**	**FFB6D**	**Ours**
Average	24.9	47.3	47.5	63.2	66	**63.2**

only 1% labeled data. The results are presented in Table 2. Our method achieves the same accuracy as PVN3D [10] which shows that our network is capable of handling occluded objects on par with keypoint based approaches despite being a dense correspondence based approach.

4.4 TLess Dataset

We conducted experiments on TLess [12] to show the robustness of our method to occlusions and symmetries. We used 1.5% (20 samples) labeled training images to train the network compared to 100% ($\approx$ 1260) used by other approaches. During training, the AR score of labels generated for weakly-labeled training samples is 0.85 indicating that we are able to label most of the training samples correctly. The results on test set are presented in Table 3. We achieve a higher VSD score that measures shape alignment. This indicates that our shape matching is on par with other approaches despite using fewer labels. We observed a slightly low recall on MSSD, MSPD since there are some objects with minimal geometric differences leading to misalignments.

Table 3: Results on TLess along with supervised [27,32] approaches. GT data refer to the amount of ground truth training data. ICP refers to results with ICP refinement.

Method	**Pix2Pose**	**DpodV2**		**Ours**	
ICP	✓	✗	✓	✗	✓
GT data	100%	100%	100%	1.5%	1.5%
VSD	0.43	0.53	0.41	0.34	**0.46**
MSSD	0.54	0.55	0.51	0.28	0.36
MSPD	0.54	0.58	0.53	0.29	0.37
AR	0.51	0.56	0.49	0.30	0.4

5 Ablation Studies

5.1 Amount Of Training Data

We achieve 94% accuracy with just 5% of training data with 1% labeled data as shown in table 1. As expected, our accuracy increases with an increase in GT training data which is evident from the increase in training data from 1% to

15%. However, the increase in accuracy (1.1%) shows that our weakly-supervised approach is estimating very good poses for unlabelled samples during training even with very less labeled data. We observe that when we use only 5% of total data as training data with 1% ground truth data, we get better accuracy (94%) compared to G2L-Net with 5% ground truth training data(89%). To show that our architecture and training pipeline is on par with benchmark approaches, we evaluate our pipeline with full ground truth labels (15%) and observe that our accuracy(99.2%) is very close to the benchmark accuracy(99.4%).

(a) (b) (c) (d) (e)

Fig. 4: Failure cases on Driller without Feature Decoder: a) Input Image, b) Input Point cloud, c) 3D Object Model d) Bad Correspondences without feature decoder , e) Correct Correspondences with feature decoder.

5.2 Feature Decoder

In the LineMOD dataset, objects like driller which has a texture that is not constant, the performance increases by a significant margin when we use feature decoder. We especially added this module to solve some issues we encountered with the driller object. As shown in the Figure 4, the pose is predicted wrongly in the fourth image. The failure cases happen when naive chamfer loss, L'_C, without feature decoder based losses (L'_F, L_F) are used. The failure cases occur as the structure is very similar to the object in both the third and fourth images. The distinguishing feature between correctly aligned fourth image and wrongly aligned third is the color of the point cloud as the structure is very similar in both the scenarios. If the failed sample is not present in the ground truth training data during training, the pose is predicted wrongly when the feature decoder is not present in the pipeline. The pose is predicted correctly in the presence of the feature decoder even if the specific sample is not present in the training set. We observe an increase in accuracy of weakly-labeled samples (8.9%) on driller when feature decoder is added as shown in Table 4.

5.3 Shape Network

The SDF loss, L'_S, helps in aligning weakly-labeled samples to the shape. We observe a drop in pose accuracy by 1.5% when we remove SDF loss. The drop

Table 4: Ablation study on the contribution of different loss functions. We present the pose labeling accuracy of weakly-labeled training data on driller object after Stage 2 with combinations of chamfer loss, shape loss and feature decoder.

Losses	L'_C		L'_S	$L'_C + L'_S$	
Feature Decoder	✗	✓	✗	✗	✓
Average	89	96	87.5	90.5	99.4

in accuracy is not significant as the chamfer loss serves a similar purpose. However, the approach works even when the shape loss L'_S is used without chamfers distance L'_C as shown in Table 4. Besides, the reconstructed shape used for pose refinement of the weakly-labeled samples improves the unrefined accuracy of the pipeline by a significant amount (6.6%) from stage 2 (91.5%) to stage 3 (98.1%).

5.4 Influence of each training stage on the final performance

To show the significance of each stage, we evaluate the pose accuracy after each stage on the LineMOD dataset presented in Table 5. The accuracy (75.3%) is quite low after Stage 1 since it is only trained with around 10 fewshot samples. The accuracy after Stage 2 (91.5%) is higher since we incorporate different viewpoints ($\sim$ 160 samples) from weakly-labeled data, but it still needs improvement to achieve exact alignment. After Stage 3 (98.1%), the network achieves better accuracy as the estimated poses for weakly-labeled data are refined using reconstructed shape and thus network learns to predict more accurately.

Table 5: Accuracy on LineMOD after each stage of training

Training Stage	Stage 1	Stage 2	Stage 3
Average	75.3	91.5	98.1

6 Conclusion

We propose a novel weakly-supervised training pipeline for pose estimation, that does not require CAD models and requires a very small number of labeled images. The core idea is to develop a pipeline leveraging few fully labeled images to automatically label the rest of the images and then train on them. To achieve this, we propose novel rotation-invariant feature and shape -based losses used for weakly-supervised shape alignment. Absolute object poses can be replaced with relative camera transformations which are easier to obtain in practice without changes to the training pipeline. Experimental evaluation demonstrate the effectiveness of our pipeline despite using only a fraction of labeled training images.

Acknowledgements This work was partially funded by the German BMWK under grant GEMIMEG-II-01MT20001A.

References

1. Arun, K.S., Huang, T.S., Blostein, S.D.: Least-squares fitting of two 3-d point sets. IEEE Transactions on Pattern Analysis and Machine Intelligence (1987)
2. Brachmann, E., Krull, A., Michel, F., Gumhold, S., Shotton, J., Rother, C.: Learning 6d object pose estimation using 3d object coordinates. In: ECCV (2014)
3. Busam, B.: High Performance Visual Pose Computation. Ph.D. thesis, Technische Universität München (2021)
4. Cai, M., Reid, I.: Reconstruct locally, localize globally: A model free method for object pose estimation. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
5. Chen, W., Jia, X., Chang, H.J., Duan, J., Leonardis, A.: G2l-net: Global to local network for real-time 6d pose estimation with embedding vector features. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
6. Drost, B., Ulrich, M., Navab, N., Ilic, S.: Model globally, match locally: Efficient and robust 3d object recognition. In: 2010 IEEE computer society conference on computer vision and pattern recognition. pp. 998–1005. Ieee (2010)
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385 (2015)
8. He, Y., Huang, H., Fan, H., Chen, Q., Sun, J.: Ffb6d: A full flow bidirectional fusion network for 6d pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3003–3013 (2021)
9. He, Y., Sun, W., Huang, H., Liu, J., Fan, H., Sun, J.: Pvn3d: A deep point-wise 3d keypoints voting network for 6dof pose estimation. In: CVPR (2020)
10. He, Y., Sun, W., Huang, H., Liu, J., Fan, H., Sun, J.: Pvn3d: A deep point-wise 3d keypoints voting network for 6dof pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
11. Hinterstoisser, S., Lepetit, V., Ilic, S., Holzer, S., Bradski, G., Konolige, K., , Navab, N.: Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes (2012)
12. Hodaň, T., Haluza, P., Obdržálek, Š., Matas, J., Lourakis, M., Zabulis, X.: T-LESS: An RGB-D dataset for 6D pose estimation of texture-less objects. IEEE Winter Conference on Applications of Computer Vision (WACV) (2017)
13. Hodan, T., Melenovsky, A.: Bop: Benchmark for 6d object pose estimation: `https://bop.felk.cvut.cz/home/` (2019)
14. Jung, H., Brasch, N., Leonardis, A., Navab, N., Busam, B.: Wild tofu: Improving range and quality of indirect time-of-flight depth with rgb fusion in challenging environments. In: 2021 International Conference on 3D Vision (3DV). pp. 239–248. IEEE (2021)
15. Kabsch, W.: A solution for the best rotation to relate two sets of vectors. Acta Crystallographica Section A (1976)
16. Kaskman, R., Shugurov, I., Zakharov, S., Ilic, S.: 6 dof pose estimation of textureless objects from multiple rgb frames. In: European Conference on Computer Vision. pp. 612–630. Springer (2020)
17. Kaskman, R., Zakharov, S., Shugurov, I., Ilic, S.: Homebreweddb: Rgb-d dataset for 6d pose estimation of 3d objects. In: ICCV Workshops (2019)
18. Labbe, Y., Carpentier, J., Aubry, M., Sivic, J.: Cosypose: Consistent multi-view multi-object 6d pose estimation. In: Proceedings of the European Conference on Computer Vision (ECCV) (2020)

19. Lepetit, V., Moreno-Noguer, F., Fua, P.: Epnp: An accurate o(n) solution to the pnp problem. International Journal of Computer Vision (2009)
20. Li, F., Shugurov, I., Busam, B., Li, M., Yang, S., Ilic, S.: Polarmesh: A star-convex 3d shape approximation for object pose estimation. IEEE Robotics and Automation Letters **7**(2), 4416–4423 (2022)
21. Li, F., Shugurov, I., Busam, B., Yang, S., Ilic, S.: Ws-ope: Weakly supervised 6-d object pose regression using relative multi-camera pose constraints. IEEE Robotics and Automation Letters (2022)
22. Li, F., Yu, H., Shugurov, I., Busam, B., Yang, S., Ilic, S.: Nerf-pose: A first-reconstruct-then-regress approach for weakly-supervised 6d object pose estimation. arXiv preprint arXiv:2203.04802 (2022)
23. Manhardt, F., Kehl, W., Gaidon, A.: Roi-10d: Monocular lifting of 2d detection to 6d pose and metric shape. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2069–2078 (2019)
24. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3d reconstruction in function space. In: Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2019)
25. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
26. Park, K., Mousavian, A., Xiang, Y., Fox, D.: Latentfusion: End-to-end differentiable reconstruction and rendering for unseen object pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2020)
27. Park, K., Patten, T., Vincze, M.: Pix2pose: Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation. In: The IEEE International Conference on Computer Vision (ICCV) (2019)
28. Peng, S., Liu, Y., Huang, Q., Zhou, X., Bao, H.: Pvnet: Pixel-wise voting network for 6dof pose estimation. In: CVPR (2019)
29. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. arXiv preprint arXiv:1612.00593 (2016)
30. Shugurov, I., Li, F., Busam, B., Ilic, S.: Osop: A multi-stage one shot object pose estimation framework. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6835–6844 (June 2022)
31. Shugurov, I., Pavlov, I., Zakharov, S., Ilic, S.: Multi-view object pose refinement with differentiable renderer. IEEE Robotics and Automation Letters (2021)
32. Shugurov, I., Zakharov, S., Ilic, S.: Dpodv2: Dense correspondence-based 6 dof pose estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence (2021)
33. Song, C., Song, J., Huang, Q.: Hybridpose: 6d object pose estimation under hybrid representations. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (2020)
34. Wang, C., Xu, D., Zhu, Y., Martín-Martín, R., Lu, C., Fei-Fei, L., Savarese, S.: Densefusion: 6d object pose estimation by iterative dense fusion (2019)
35. Wang, G., Manhardt, F., Shao, J., Ji, X., Navab, N., Tombari, F.: Self6d: Self-supervised monocular 6d object pose estimation. In: The European Conference on Computer Vision (ECCV) (2020)
36. Wang, G., Manhardt, F., Tombari, F., Ji, X.: GDR-Net: Geometry-guided direct regression network for monocular 6d object pose estimation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)

37. Wang, P., Manhardt, F., Minciullo, L., Garattoni, L., Meier, S., Navab, N., Busam, B.: Demograsp: Few-shot learning for robotic grasping with human demonstration. In: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 5733–5740. IEEE (2021)
38. Xiang, Y., Schmidt, T., Narayanan, V., Fox, D.: Posecnn: A convolutional neural network for 6D object pose estimation in cluttered scenes. In: Robotics: Science and Systems (RSS) (2018)
39. Zakharov, S., Kehl, W., Bhargava, A., Gaidon, A.: Autolabeling 3d objects with differentiable rendering of sdf shape priors. In: IEEE Computer Vision and Pattern Recognition (CVPR) (June 2020)
40. Zakharov, S., Shugurov, I., Ilic, S.: Dpod: 6d pose object detector and refiner. In: International Conference on Computer Vision (ICCV) (2019)
41. Zhang, S., Zhao, W., Guan, Z., Peng, X., Peng, J.: Keypoint-graph-driven learning framework for object pose estimation. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)