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

—

Supplementary Material

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

1 Additional Results

We present some qualitative results on the Linemod dataset [2] in Figure 1. We also present shape reconstructions of some objects obtained by creating meshes from implicit representation in Figure 2.

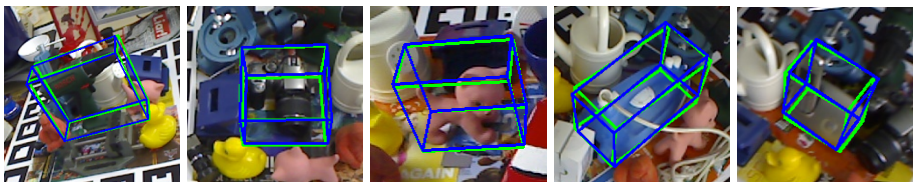

Fig. 1: Pose estimation results on LineMOD: The green and blue box indicate the bounding box projected with ground truth pose and predicted pose respectively

1.1 Reprojection

To generate more data for the deep learning pipeline and to make our network more robust towards unseen poses, we employ an approach to generate more data by augmenting rotations and translations on point clouds. We use RGB-D image and segmentation mask to generate colored point cloud and RGB image. However, since both input modalities, point cloud and image, are supposed to preserve the pose relationship when they undergo augmentations, we can only perform in-plane rotations on RGB-D images since the projected image cannot undergo out-of-plane rotations. To add out-of-plane rotation augmentations, we transform the RGB-D image to a colored point cloud and then re-project the point cloud after performing rotations on point clouds. This way, we can incorporate out-of-plane rotation augmentations into the training pipeline. We observe an increase in accuracy of about 1.5% on total accuracy by adding out-of-plane augmentations.

Fig. 2: Shape reconstruction results of some objects in first row along with their ground truth meshes with color in second row.

1.2 Training protocol

We employ a ResNet-18 [1] based image encoder and PointNet [3] based point cloud encoder to predict a global embedding. The feature decoder and correspondence decoder employ a similar architecture with a multi-layer-perceptron that takes the global embedding and a 3D point input to predict the corresponding point features and correspondence points respectively. The shape network has a similar architecture as the correspondence decoder except for the output layer and input layer. In stage 1, we train the network for 300 epochs to learn the shape and encoder-decoder networks using the assembled point cloud and labeled samples. In stage 2, we freeze the shape network to train the encoder-decoder pipeline using shape-based losses from the shape network and feature aligning chamfer loss for unlabelled samples for 500 epochs. In stage 3, we reconstruct the mesh from its implicit representation. Then, we predict the poses for unlabeled data and refine the estimated poses using the reconstructed shape. We then train the encoder-decoder with labels for all the samples for 1000 epochs. We employ a learning rate of 0.0005 for all the networks in all the stages. We use images of resolution 200×200 and point clouds with 1000 points during training.

1.3 Architecture

The detailed architecture of our network is presented in Figure 3. Our network takes in a segmented point cloud and segmented color image to predict 3D correspondences which are used to estimate the 6D pose. A feature decoder and shape network are employed only during training to improve shape alignment. The encoder consists of a point cloud encoder and an image encoder. To process a color image, we employ a ResNet-18 to extract a latent vector from the color image, I, with width, W, and height, H. The point cloud, X, with n points along with its color, C, and normal, N, are concatenated to form a $n \times 9$ dimensional

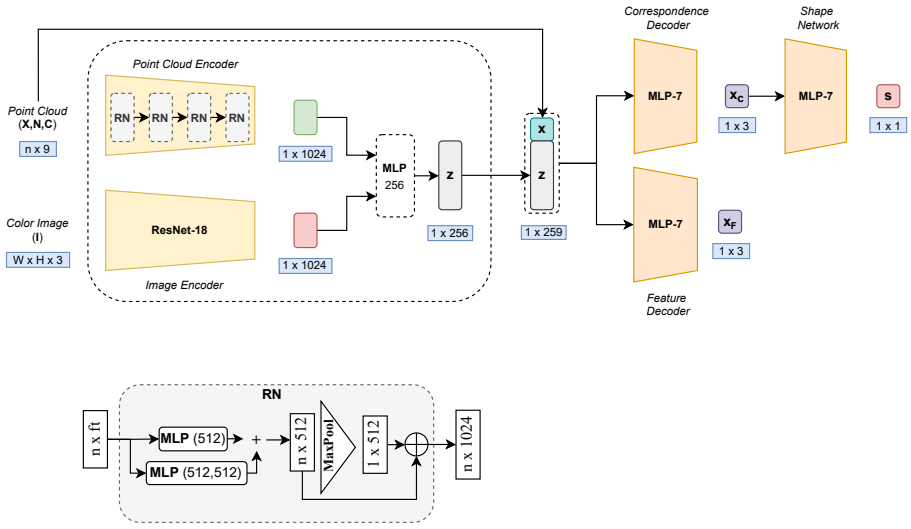

Fig. 3: The architecture of our pipeline. The network comprises two encoders (Point Cloud Encoder, Image Encoder) to encode point clouds and color image and two decoders (Correspondence Decoder, Feature Decoder) to predict point-wise correspondences and point-wise features. The shape network learns the shape of the object in the canonical pose by taking correspondences as input to predict the corresponding SDF values. The blue boxes indicate the dimension of the vector/matrix. RN represents the resnet blocks used in the point cloud encoder which comprises resnet connections to point cloud features. The latent code, z, predicted by the encoder is concatenated with each 3D point in the point cloud to pass through both the decoders. The encoder takes the full point cloud, while the decoders and shape network only takes input involving a single 3D single point at a time. We only show the decoder pipeline with a single 3D point to create a simpler figure. In reality, each point is concatenated with the latent vector, z, to pass through the decoder. Similarly, the shape network also takes all the correspondences with a single 3D point input at a time.

input which is passed through a point cloud encoder to extract a latent vector. The point cloud encoder is composed of 4 ResNet (RN) blocks containing MLP blocks and MaxPool layers which map the $n \times ft$ dimensional input matrix to an output of size $n \times 1024$. Both latent vectors are concatenated to pass through another MLP to predict a 256 dimensional latent vector, z. The predicted latent vector, z, is concatenated with each point in the point cloud to pass through a correspondence decoder, a seven-layer MLP, to predict its correspondence, x_C. The same input is passed through the feature decoder which is also a seven-layered MLP to predict per point feature x_F. The correspondence decoder and feature decoder have similar architecture comprising six fully connected layers with 512 dimensional output and a final fully connected layer with an output of dimension three. The predicted correspondences, x_C, are passed through the shape network to predict the corresponding SDF value, s. The shape network comprises a seven-layer MLP with six fully connected layers with 512 dimensional outputs and a final fully connected layer with an output of dimension one. The MLP used in RN blocks and decoder employ a ReLu activation function.

References

1. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385 (2015)
2. 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)
3. 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)