Category-Level 6D Object Pose Recovery in Depth Images

Caner Sahin(✉) and Tae-Kyun Kim

ICVL, Imperial College London, London, UK
c.sahin14@imperial.ac.uk

Abstract. Intra-class variations, distribution shifts among source and target domains are the major challenges of category-level tasks. In this study, we address category-level full 6D object pose estimation in the context of depth modality, introducing a novel part-based architecture that can tackle the above-mentioned challenges. Our architecture particularly adapts the distribution shifts arising from shape discrepancies, and naturally removes the variations of texture, illumination, pose, etc., so we call it as "Intrinsic Structure Adaptor (ISA)". We engineer ISA based on the followings: (i) "Semantically Selected Centers (SSC)" are proposed in order to define the "6D pose" at the level of categories. (ii) 3D skeleton structures, which we derive as shape-invariant features, are used to represent the parts extracted from the instances of given categories, and privileged one-class learning is employed based on these parts. (iii) Graph matching is performed during training in such a way that the adaptation/generalization capability of the proposed architecture is improved across unseen instances. Experiments validate the promising performance of the proposed architecture using both synthetic and real datasets.

Keywords: Category-level · 6D object pose · 3D skeleton · Graph matching · Privileged one-class learning

1 Introduction

Accurate 3D object detection and pose estimation, also known as 6D object pose recovery, is an essential ingredient for many practical applications related to scene understanding, augmented reality, control and navigation of robotics, *etc.* While substantial progress has been made in the last decade, either using depth information from RGB-D sensors [1–8] or even estimating pose from a single RGB image [9–12], improved results have been reported for instance-level recognition where source data from which a classifier is learnt share the same statistical distributions with the target data on which the classifiers will be tested. Instance-based methods cannot easily be generalized for category-level tasks, which inherently involve the challenges such as distribution shift among source and target domains, high intra-class variations, and shape discrepancies between objects, *etc.*

L. Leal-Taixé and S. Roth (Eds.): ECCV 2018 Workshops, LNCS 11129, pp. 665–681, 2019.
https://doi.org/10.1007/978-3-030-11009-3_41

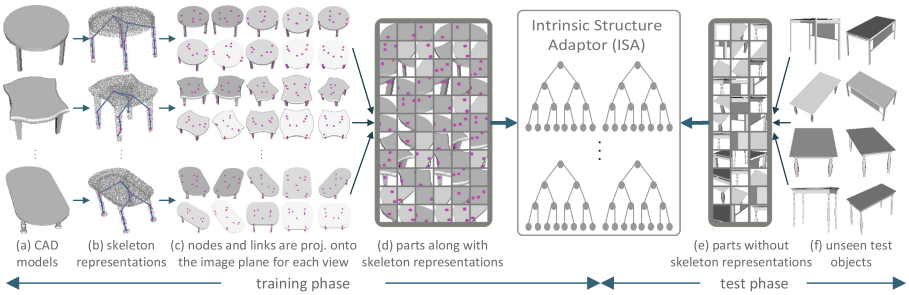

Fig. 1. Intrinsic Structure Adaptor (ISA) is trained based on parts extracted from instances of a given category. CAD models in (a) are represented with skeletons in (b). Nodes and links are projected onto the image plane in (c) for each view. Parts along with skeletal representations in (d) are fed into the forest. In the test, appearances of the parts (e) that are extracted from depth images of unseen instances (f) are used in order to hypothesise 6D pose.

At the level of categories, Sliding Shapes (SS) [13], an SVM-based method enlarging search space to 3D, detects objects in the context of depth modality naturally tackling the variations of texture, illumination, and viewpoint. The detection performance of this method is further improved in Deep Sliding Shapes (Deep SS) [14], where more powerful representations encoding geometric shapes are learned in ConvNets. These two methods run sliding windows in the 3D space mainly concerning 3D object detection rather than full 6D pose estimation. The system in [15], inspired by [13], further estimates detected and segmented objects' rotation around the gravity axis using a CNN. The system is the combination of individual detection/segmentation and pose estimation frameworks. Unlike these methods, we aim to directly hypothesise full 6D poses in a single-shot operation. The ways the methods above [13–15] address the challenges of categories are relatively naive. Both SS and the method in [15] rely on the availability of large scale 3D models in order to cover the shape variance of objects in the real world. Deep SS performs slightly better against the categories' challenges, however, its effort is limited to the capability of ConvNets.

In this study, we engineer a dedicated architecture that directly tackles the challenges of categories while estimating objects' 6D. To this end, we utilize *3D skeleton structures*, derive those as shape-invariant features, and use those as privileged information during the training phase of our architecture. *3D skeleton structures* are frequently used in the literature in order to handle shape discrepancies [16–19]. We introduce "Intrinsic Structure Adaptor (ISA)", a part-based random forest architecture, for full 6D object pose estimation at the level of categories in depth images. ISA works in the 6D space. It neither requires a segmented/cropped image as in [15], nor asks for 3D bounding box proposals as in [14]. Unlike [13,14], instead of running sliding windows, ISA extracts parts from the input depth image, and feeding all those down the forest, directly votes for the 6D pose of objects. Its training phase is processed so that the challenges

of the categories can successfully be tackled. 3D skeleton structures are used to represent the parts extracted from the instances of given categories, and privileged learning is employed based on these parts. Graph matching is performed during the splitting processes of random forest in such a way that the adaptation/generalization capability of the proposed architecture is improved across unseen instances. Note that, unlike [13–15], this is one-class learning, and a single classifier is learnt for all instances of the given category. Figure 1 depicts the whole system of our architecture. To summarize, our main contributions are as follows:

Contributions. "Semantically Selected Centers (SSC)" are proposed in order to define the "6D pose" at the level of categories. 3D skeleton structures, which we derive as shape-invariant features, are used to represent the parts extracted from the instances of given categories, and privileged one-class learning is employed based on these parts. Graph matching is performed during training in such a way that the adaptation/generalization capability of the proposed architecture is improved across unseen instances.

2 Related Work

A number of methods have been proposed for 3D object detection and pose estimation, and for skeleton representations. For the reader's convenience, we only review 6D case for instance-level object detection and pose estimation, and keep category-level detection broader.

2.1 Object Detection and Pose Estimation

Instance-Level (6D): State-of-the-art methods for instance-level 6D object pose estimation report improved results tackling the problem's main challenges, such as occlusion and clutter, and texture-less objects, *etc.* The holistic template matching approach, Linemod [20], estimates cluttered object's 6D pose using color gradients and surface normals. It is improved by discriminative learning in [21], and later been utilized in a part-based random forest method [22] in order to provide robustness across occlusion. Occlusion aware features [23] are further formulated, and more recently feature representations are learnt in an unsupervised fashion using deep convolutional networks [1,24]. The studies in [2,3] cope with texture-less objects. Whilst these methods fuse data coming from RGB and depth channels, a local belief propagation based approach [25] and an iterative refinement architecture [26,27] are proposed in depth modality [28]. 6D pose estimation is recently achieved from RGB only [9–12]. Despite being successful, instance-based methods cannot easily be generalized for category-level tasks, which inherently involve the challenges such as distribution shift among source and target domains, high intra-class variations, and shape discrepancies between objects, *etc.*

Category-Level: At the level of categories, several studies combine depth data with RGB. Depth images are encoded into a series of channels in [29] in such a way that R-CNN, the network pre-designed for RGB images, can represent that encoding properly. The learnt representation along with the features extracted from RGB images are then fed into an SVM classifier. In another study [30], annotated depth data, available for a subset of categories in Imagenet, are used to learn mid-level representations that can be fused with mid-level RGB representations. Although promising, they are not capable enough for the applications beyond 2D.

Sliding Shapes (SS) [13], an SVM-based method, hypothesises 3D bounding boxes of the objects, and naturally tackles the variations of texture, illumination, and viewpoint, since it works in depth images. However, hand-crafted features used by the method, being unable to reasonably handle the challenges of categories, limit the method's detection performance across unseen instances. Deep Sliding Shapes (Deep SS) [14], the method based on 3D convolutional neural networks (CNN), learns more powerful representations for encoding geometric shapes further improving SS. However, the improvement is architecture-wise, and Deep SS encodes a 3D space using Truncated Signed Distance Functions (TSDF), similar to SS. Although promising, both methods concentrate on hypothesising 3D bounding boxes, running sliding windows in the 3D space. Our architecture, ISA, works in the 6D space. Instead of running sliding windows, it directly votes for the 6D pose of the objects by passing the parts extracted from the input depth image down all the trees in the forest. The system in [15], inspired by [13], further estimates detected and segmented objects rotation around gravity direction using a CNN, which is trained using pixel surface normals. A relative improvement is observed in terms of accuracy, however, the system is built integrating individual detection/segmentation and pose estimation frameworks. ISA neither requires a segmented/cropped image as in [15], nor asks for 3D bounding box proposals as in [14].

Despite being proposed to work in large-scale scenarios, the methods [13–15] do not have specific designs that can explicitly tackle the challenges of categories. SS relies on the availability of large scale 3D models in order to handle distribution shifts arising from shape discrepancies. Deep SS learns powerful 3D features from the data via a CNN architecture, however, the representation used to encode a 3D space is similar to the one used in SS, that is, the improvement on the feature representation arises from the CNN architecture. Gupta et al. [15] use objects' CAD models at different scales in order to cover the shape variance of the objects in the real world while estimating objects' rotation and translation. Unlike these methods, ISA is a dedicated architecture that directly tackles the challenges of the categories while estimating objects 6D. It employs graph matching during forest training based on the parts represented with skeleton structures in such a way that the adaptation/generalization capability is improved across unseen instances.

2.2 Skeleton Representation

Skeletal structures have frequently been used in the literature, particularly to improve the performance of action/activity recognition algorithms. Baek et al. [31] consider the geometry between scene layouts and human skeletons and propose kinematic-layout random forests. Another study [17] utilizes skeleton joints as privileged information along with raw depth maps in an RNN framework in order to recognise actions. The study in [18] shows that, one can effectively utilize 3D skeleton structures for overcoming intra-class variations, and for building a more accurate classifier, advocating the idea, domain invariant features increase generalization, stated in [19].

3 Proposed Architecture

This section presents the technologies top of which the proposed architecture, ISA, is based on. We firstly define the "pose" for the category-level 6D object pose estimation problem, and demonstrate the dataset and annotations discussing shape-invariant feature representations. We next present privileged one-class learning where we employ graph matching, and lastly we describe the test step, category-level 6D object pose estimation.

3.1 Pose Definition: Semantically Selected Centers (SSC)

A method designed for 6D pose estimation outputs the 3D position and 3D rotation of an object of interest in camera-centered coordinates. According to this output, it is important to precisely assign the reference coordinate frame to the interested object. When the method is proposed for instance-level 6D object pose estimation tasks, the most common approach is to assign the reference coordinate frame to the center of mass (COM) of the object's model. At the level of instances, source data from which a classifier is learnt share the same statistical distributions with the target data on which the classifiers will be tested, that is, training and test samples are of the same object. Hence, instance-level 6D pose estimators output the relative orientation between the COM of the object and the camera center. At the level of categories, in turn, this 6D pose definition cannot be directly utilised, since significant distribution shifts arise between training and test data.

An architecture engineered for the category-level 6D object pose estimation problem should hypothesise 6D pose parameters of unseen objects. Objects from the same category typically have similar physical sizes [14]. However, investigations over 3D models of the instances demonstrate that each instance has different COM, thus making the utilization of conventional 6D pose definition to malfunction for category-level tasks. In such a case, we reveal Semantically Selected Centers (SSC), which allow us to redefine the "6D pose" for the category-level 6D object pose estimation problem. For every category we define only one SSC performing the following procedure:

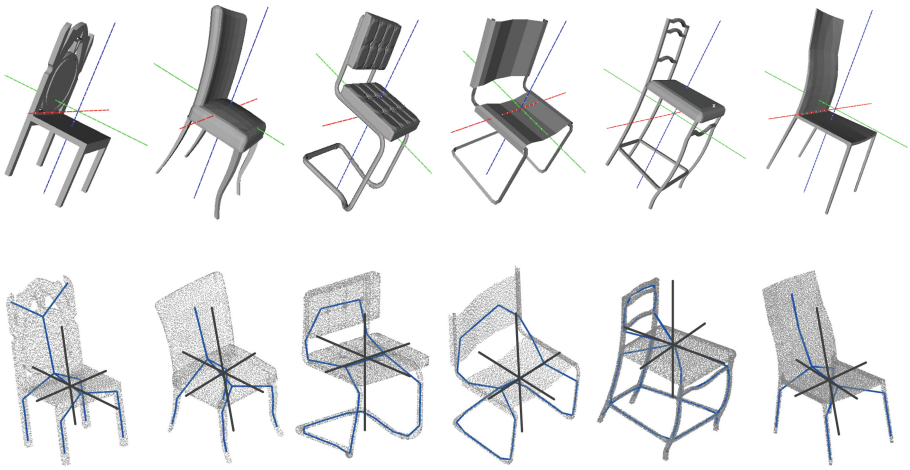

Fig. 2. Semantically Selected Centers (SSC): top row shows centers of mass of the instances, while the bottom row depicts SSCs of the corresponding instances (views best describing the difference selected).

- For each instance, skeletal graph representation is extracted, and the COM is found over 3D model. 3D distances between the nodes of the representation and the COM is computed.
- Between all instances, the skeleton nodes are topologically matched, and the most repetitive node is determined.
- In case there are more than 1 repetitive node computed, the SSCs are determined by interpolating between the repetitive nodes.

Note that, this repetitive node is also the one closest to the COMs of the instances. As the last step, we assign reference coordinate frames to the related parts of the objects given in the category. Figure 2 shows SSCs for the chair category. Despite the fact that COMs of the models are individually different, the 6D pose of each chair is defined with respect to Semantically Selected Centers (bottom row of the figure).

The metric proposed in [20], Average Distance (AD), is designed to measure the performance of instance-level object detectors. In order to evaluate our architecture, we modify AD making this metric work at the level of categories via the Semantically Selected Centers (SSC) of the instances of the given category. $M_c^{SSC_i}$ is the 3D model of the instance i that belongs to the category c, and the set of $M_c^{SSC_i}$ of the test instances form $\mathcal{M}_c$: $\mathcal{M}_c = \{M_c^{SSC_i} | i = 1, 2, ...\}$. $X_c^{SSC_i}$ is the point cloud of the model $M_c^{SSC_i}$. Having the ground truth rotation R and translation T, and the estimated rotation $\tilde{R}$ and translation $\tilde{T}$, we compute the average distance over $X_c^{SSC_i}$:

$$\omega_i = avg||(RX_c^{SSC_i} + T) - (\tilde{R}X_c^{SSC_i} + \tilde{T})||. \tag{1}$$

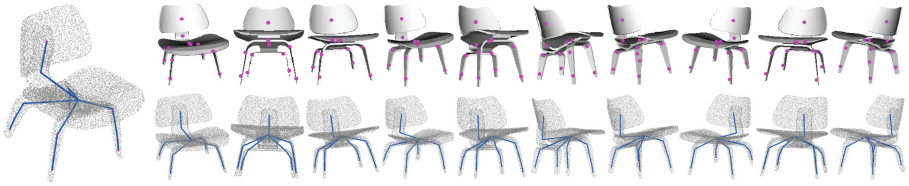

Fig. 3. Skeletal graph representation: skeleton nodes are determined with respect to model coordinate frame. Skeletal nodes and links are projected onto the image plane for each viewpoint at which a synthetic depth image is rendered.

ω_i calculates the distance between the ground truth and estimated poses of the test instance i. The detection hypothesis that ensures the following inequality is considered as correct:

$$\omega_i \leq z_{\omega_i} \Phi_i \tag{2}$$

where Φ_i is the diameter of $M_c^{SSC_i}$, and z_{ω_i} is a constant that determines the coarseness of an hypothesis that is assigned as correct.

3.2 Dataset and Part Representations

The training dataset $\mathcal{S}$ involves synthetic data that are of c_s instances of a given category. Using the 3D CAD models of these c_s instances, we render foreground synthetic depth maps from different viewpoints and generate annotated parts in order to form $\mathcal{S}$:

$$\begin{aligned} \mathcal{S} &= \{\mathcal{P}_i | i = 1, 2, ..., c_s\} \\ \mathcal{P}_i &= \{\cup_{j=1}^{n} P_j\} = \{\cup_{j=1}^{n} (\mathbf{c}_j, \Delta\mathbf{x}_j, \theta_j, \mathbf{a}_j, \mathbf{s}_j, D_{P_j})\} \end{aligned} \tag{3}$$

where $\mathcal{P}_i$ involves the set of parts $\{P_j | j = 1, 2, ..., n\}$ that are extracted from the synthetic images of the object instance i. $\mathbf{c}_j = (c_{x_j}, c_{y_j}, c_{z_j})$ is the part centre in $[px, px, m]$. $\Delta\mathbf{x}_j = (\Delta x, \Delta y, \Delta z)$ presents the 3D offset between the centre of the part and the SSC of the object, and $\theta_j = (\theta_r, \theta_p, \theta_y)$ depicts the 3D rotation parameters of the point cloud from which the part P_j is extracted. $\mathbf{a}_j$ describes the vector of the skeletal link angles. $\mathbf{s}_j$ is the skeletal node offset matrix representation, and D_{P_j} is the depth map of the part P_j.

We next briefly mention how we derive $\mathbf{a}_j$ and $\mathbf{s}_j$ based on skeletal graph representation extracted from 3D model of an instance.

Derivation of $\mathbf{a}_j$ and $\mathbf{s}_j$. The algorithm in [32] is utilized in order to extract the skeletal graph of an instance from its 3D model. Once the skeletal graph is extracted, we next project both the nodes and the links onto the image plane for every viewpoint at which synthetic depth maps are rendered. At each viewpoint, we measure the angles that the links of the graph representation make with the x direction, and stack them into the vector of skeletal link angles $\mathbf{a}_j$ (see Fig. 3). All of the parts extracted at a specific viewpoint have the same representation $\mathbf{a}$. The distances between the centre $\mathbf{c}_j$ of each part P_j and skeleton nodes are

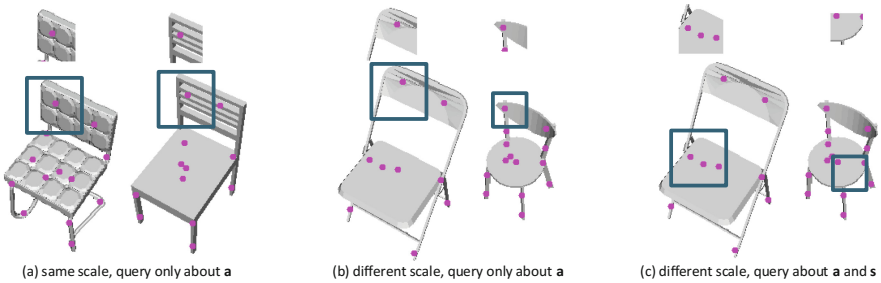

Fig. 4. Parts in (a) and (b) are topologically at the same location, having the same **a**. Parts in (c) are topologically at different location, having the same **a**, the case which is undesired. Hence the parts are further questioned with **s**, the representation that removes mismatches.

measured in image pixels along x and y, and in metric coordinates along z direction in order to derive the skeletal node offset matrix $\mathbf{s}_j$:

$$\mathbf{s}_j = [\Delta x_{j_i}, \Delta y_{j_i}, \Delta z_{j_i}]_{s_n \times 3}, \quad i = 1, 2, ..., s_n. \tag{4}$$

Figure 3 shows an example skeletal graph and its projection onto 2D image plane for several viewpoints. In this representation, we compute 19 nodes in total and project onto the image plane 11 of those.

We next discuss how to handle shape discrepancies between the parts extracted from the instances of a given category using the representations **a** and **s**.

Privileged Data: Shape-Invariant Skeleton Representations. We start our discussion by firstly representing the parts with **a**. The study in [14] states that objects from the same category typically have similar physical size, however, the appearances of the objects are relatively different. Figure 4(a) depicts the parts extracted from 2 different objects, belonging to the same category. Despite the fact that both parts have different shapes in depth channel, their representations **a** are the same, tackling the discrepancy in shape.

There are also cases where some instances are relatively larger in the given category. The vector of skeletal link angles, **a**, readily handles the scale variation between the instances. In Fig. 4(b), the objects from which the parts extracted are different in both shape and in scale, however, the parts have the same representations **a**. One drawback of this representation is that it is not sufficient enough to match topologically correct parts. In Fig. 4(c), the parts are semantically at different locations of the objects, however, they have the same **a**. Hence, we additionally represent the parts with the skeletal node offset matrix **s**. **s** along with **a** are used to adapt the intrinsic structures of the instances while topologically constraining the structures. In Fig. 4(c), when we query about **s**, in addition to **a**, the mismatch between the parts disappears, since both parts have different skeletal node offset matrix representation **s**.

3.3 Privileged One-Class Learning

ISA, being a part-based random forest architecture, is the combination of randomized binary decision trees. Employing one-class learning, it is trained only on positive samples, rather than explicitly collecting representative negative samples. The learning scheme is additionally privileged. The part representations **a** and **s** are only available during training, and not required during testing. This is achieved by using them in the split criteria (Eq. 7), but not in the split function (Eq. 5). We use the dataset $\mathcal{S}$ in order to train ISA employing simple depth comparison features (2-pixel test) in the split nodes. At a given pixel $\mathbf{w}$, the features compute:

$$f_\psi(D_P, \mathbf{w}) = D_P(\mathbf{w} + \frac{\mathbf{u}}{D_P(\mathbf{w})}) - D_P(\mathbf{w} + \frac{\mathbf{v}}{D_P(\mathbf{w})}) \tag{5}$$

where $D_P(\mathbf{w})$ is the depth value of the pixel $\mathbf{w}$ in part P, and the parameters $\psi = (\mathbf{u}, \mathbf{v})$ depict offsets $\mathbf{u}$ and $\mathbf{v}$. Each tree is constructed by using a randomly selected subset $\mathcal{W} = \{P_j\}$ of the annotated training parts $\mathcal{W} \subset \mathcal{S}$. Starting from the root node, a group of splitting candidates $\{\phi = (\psi, \tau)\}$, where ψ is the feature parameter and τ is the threshold, are randomly produced. The subset $\mathcal{W}$ is partitioned into left $\mathcal{W}_l$ and right $\mathcal{W}_r$ by each ϕ:

$$\begin{aligned} \mathcal{W}_l(\phi) &= \{\mathcal{W} | f_\theta(D_P, \mathbf{w}) < \tau\} \\ \mathcal{W}_r(\phi) &= \mathcal{W} \setminus \mathcal{W}_l. \end{aligned} \tag{6}$$

The ϕ that best optimizes the following entropy is determined:

$$\begin{aligned} \phi^* &= \arg\max_\phi (Q(\phi)) \\ Q &= Q_1 + Q_2 + Q_3 \end{aligned} \tag{7}$$

where Q_1, Q_2, and Q_3 are the 6D pose entropy, the skeletal link angle entropy, and the skeletal node offset entropy, respectively. Each tree is grown by repeating this process recursively until the forest termination criteria are satisfied. When the termination conditions are met, the leaf nodes are formed and they store votes for both the object center $\Delta\mathbf{x} = (\Delta x, \Delta y, \Delta z)$ and the object rotation $\theta = (\theta_r, \theta_p, \theta_y)$.

Matching Skeletal Graphs. When we build ISA, our main target is to provide adaptation between the instances, and to improve the generalization across unseen objects. Apart from the data used to train the forest, the quality functions we introduce play an important role for these purposes. The quality function Q_1, optimizing data with respect to only 6D pose parameters, is given below:

$$Q_1 = log(|\Sigma^{\Delta\mathbf{x}}| + |\Sigma^\theta|) - \sum_{i \in (L,R)} \frac{\mathcal{S}_i}{\mathcal{S}} log(|\Sigma_i^{\Delta\mathbf{x}}| + |\Sigma_i^\theta|) \tag{8}$$

where $|\Sigma^{\Delta\mathbf{x}}|$, $|\Sigma^\theta|$ show the determinants of offset and pose covariance matrices, respectively. $\mathcal{S}_i$ depicts the synthetic data sent either to the left L or to the right

R child node. In case the architecture is trained only using parts extracted from 1 instance, Q_1 successfully works. We train ISA using multiple instances, targeting to improve the adaptation/generalization capability across unseen instances. In order to achieve that, we propose the following quality function in addition to Q_1:

$$Q_2 = log(|\Sigma^{\mathbf{a}}|) - \sum_{i \in (L,R)} \frac{\mathcal{S}_i}{\mathcal{S}} log(|\Sigma_i^{\mathbf{a}}|) \tag{9}$$

where $|\Sigma^{\mathbf{a}}|$ shows the determinant of the skeletal link angle covariance matrix. This function measures the similarity of the parts regarding the angles that the links of the skeleton representations make with the x direction. The main reason why we use this function is to handle shape discrepancies in depth channel between parts, even if the parts are extracted from relatively large scale objects. Let's suppose that if all parts under query are extracted from topologically same locations of the instances, the combination of Q_1 and Q_2 would be sufficient. On the other hand, the combination of these two functions is not sufficient, since the parts are extracted from the complete structures of the instances. In such a scenario, the parts coming from topologically different locations, but with similar $\mathbf{a}$ are tend to travel to the same child node, if the features used in the split function fails to correctly separate the data. Hence, we require the following function that prevents this drawback:

$$Q_3 = log(|\Sigma^{\mathbf{s}}|) - \sum_{i \in (L,R)} \frac{\mathcal{S}_i}{\mathcal{S}} log(|\Sigma_i^{\mathbf{s}}|) \tag{10}$$

where $|\Sigma^{\mathbf{s}}|$ shows the determinant of the skeletal node offset covariance matrix. The main reason why we use Q_3 is to prevent topologic mismatches in between the parts extracted from different instances of the given category.

3.4 Category-Level 6D Object Pose Estimation

Given a category of interest c, and a depth image I^t in which an unseen instance of the interested category exists, the proposed architecture, ISA, targets to maximize the joint posterior density of the object position $\Delta\mathbf{x}$ and the rotation θ:

$$(\Delta\mathbf{x}, \theta) = \arg\max_{\Delta\mathbf{x},\theta} p(\Delta\mathbf{x}, \theta | I^t, c). \tag{11}$$

Since ISA is based on parts, and the parts extracted from I^t are passed down all the trees by the split function in Eq. 5, we can calculate the probability $p(\Delta\mathbf{x}, \theta | I^t, c)$ for a single tree T aggregating the conditional probabilities $p(\Delta\mathbf{x}, \theta | P, c)$ for each part P:

$$p(\Delta\mathbf{x}, \theta | I^t, c; T) = \sum_i p(\Delta\mathbf{x}, \theta | P_i, c, D_{P_i}; T). \tag{12}$$

In order to hypothesise the final pose parameters, we average the probabilities over all trees using the information stored in the leaf nodes for a given forest F:

$$p(\Delta\mathbf{x}, \theta | I^t, c; F) = \frac{1}{|F|} \sum_{t}^{|F|} \sum_{i} p(\Delta\mathbf{x}, \theta | P_i, c, D_{P_i}; T_t). \tag{13}$$

Please note that the above pose inference is done using a single depth image, not skeletons and their representations.

4 Experiments

In order to validate the performance of the proposed architecture, we conduct experiments on both synthetic and real data.

Synthetic Dataset. Princeton ModelNet10 dataset [33] contains CAD models of 10 categories, and in each category, the models are divided into train and test. We use the CAD models of the test instances of four categories, *bed*, *chair*, *table*, and *toilet*, and render depth images from different viewpoints, each of which is 6D annotated and occlusion/clutter-free. Each category involves 264 images of unseen objects, and there are 1320 test images in total. We compare ISA and instance-based Linemod on the synthetic dataset.

Real Dataset. RMRC [34], involving cluttered real depth images of several object categories, is the dataset on which we test and compare our architecture with the state-of-the-art methods [13–15]. The images in this dataset are annotated only with 3D bounding boxes.

Evaluation Protocols. The evaluation protocol used for the experiments conducted on the synthetic dataset is the one proposed in Subsect. 3.1. We make use of the evaluation metric in [13] when we compare ISA with the state-of-the-art methods on real data.

4.1 Experiments on Synthetic Data

The main reason why we conduct experiments first on synthetic data is to demonstrate the intrinsic structure adaptation performance of the proposed algorithm in order to have a better understanding on its behaviour across unseen instances.

Training ISA. We employ one-class privileged training using only positive synthetic samples and train the classifiers based on parts extracted from the depth images of the instances in the given categories. Note that, the data related to skeletal representation is only available during training, and in the test phase, the parts reach the leaf nodes using depth appearances in order to vote for a

6D pose. The models from which the depth images are synthesised are sorted through the training part of ModelNet10. The number of the instances, the number of the viewpoints from which synthetic depth images are rendered, and the number of the parts used during training are shown in Table 1.

We train 16 different forests each 4 of which are individually trained using the quality functions Q_1, $Q_1\&Q_2$, $Q_1\&Q_3$, and $Q_1\&Q_2\&Q_3$ per category. The instances used to train the forests are shown in Fig. 5.

Linemod Templates. Since Linemod is an instance-based detector, the templates method uses are of the object instance on which the the method is tested. Hence, in order to fairly compare Linemod detector with ISA, we employ the following strategy: on the test images of a given category (*e.g.* chair), we run the Linemod detector using the templates of each training instance (for chair, we run Linemod detector 28 times using 89 templates of each of 28 training instances, see Table 1). We sort the recall values, and report 3 different numbers: Linemod (min) represents the lowest recall obtained by any of the training instances, Linemod (max) depicts the highest recall obtained by any of the training instances, and Linemod (all) shows the mean of recall values obtained by all of the training instances.

Test. Unseen test instances are shown in Fig. 6. The resultant recall values are depicted in Table 2 (left). A short analysis on the table reveals that the ISAs based on the 6D pose entropy Q_1 demonstrate the poorest performance. Thanks to the utilization of the skeletal link angle entropy Q_2, in addition to the 6D quality function, the classifiers reach higher recall values. In case the skeletal node offset entropy Q_3 is used along with the 6D pose entropy, there is a relative improvement if we compare with the classifiers trained only on 6D pose entropy. The combined utilization of 6D pose, skeletal link angle, and skeletal node offset entropies performs best on average.

For the *bed* category, separately using Q_2 and Q_3 along with Q_1 demonstrates approximately the same performance when the classifiers are trained only on the quality function Q_1. However, the combined utilization of Q_1, Q_2, and Q_3 shows the best performance. For the *chair* category, the forest trained on Q_1 and Q_3 generates the highest recall value, describing the positive impact of using the skeleton node offset entropy. Unlike the bed category, exploiting the skeletal link angle entropy Q_2 along with $Q_1\&Q_3$ relatively degrades the performance of ISA. For the *table* category, one can observe that the skeletal link angle entropy and the skeletal node offset entropy contribute the same to the classifiers in order to generalize across unseen instances. Training the forests using both $Q_1\&Q_2$ and $Q_1\&Q_3$ gives rise 3% improvement with respect to the quality function Q_1 only. For the *toilet* category, using Q_1 along with Q_2 outperforms other forests. Despite the fact that adding the last term Q_3 into the combined quality function relatively decreases the recall value, the resultant performance is still better that the classifier trained Q_1 only. Figure 7 depicts sample hypotheses of unseen instances with ground truth poses in red. The forests based on $Q_1\&Q_2\&Q_3$

Table 1. Numbers on training samples

	Bed	Chair	Table	Toilet
#instances	2	28	8	7
#view (per inst.)	89	89	89	89
#parts (total)	∼600k	∼1m	∼900k	∼800k

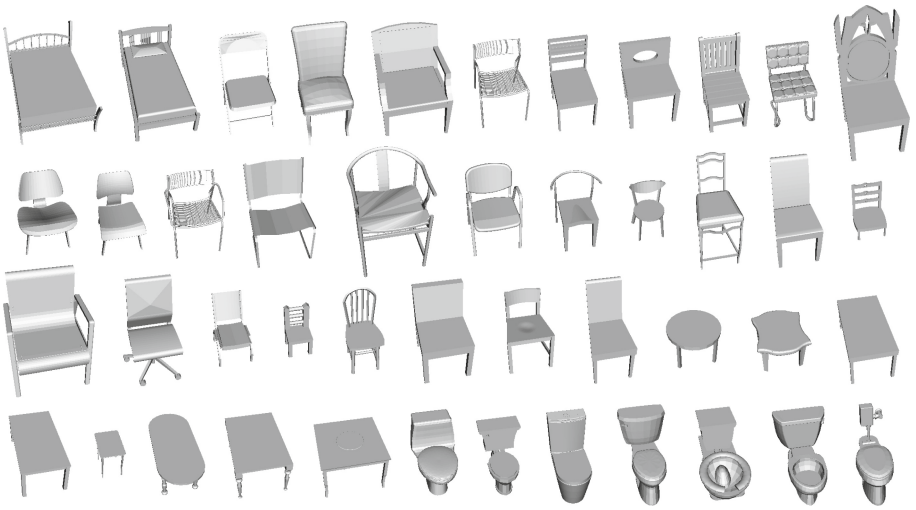

Fig. 5. Instances used to train a separate ISA for each category. These training instances are used to generate templates for testing Linemod.

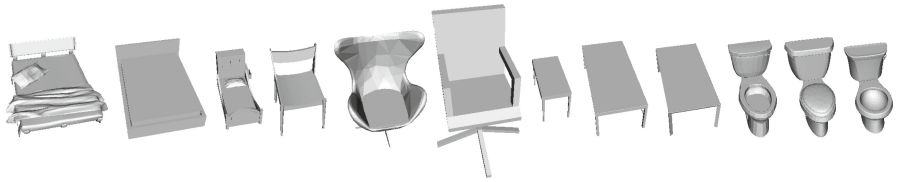

Fig. 6. Unseen object instances on which ISA and Linemod are tested

hypothesise the green estimations which are considered as true positive, and the forests based on Q_1 hypothesise the blue estimations which are considered as false positive. Note that, both 3D position and 3D orientation of an estimation are used when deciding whether the object is correctly estimated.

In Table 2 (left), we report recall values for the Linemod detector. Using the templates of each training instance of the given category, we run Linemod, and sort the recall values. According to the Linemod (min) recall values, Linemod worst performs on the *chair* category, whilst it shows best performance on the

Table 2. (left) Comparison on 6D object pose using the evaluation metric in Subsect. 3.1. (right) Comparison on 3D object detection using the evaluation metric in [13].

Method	bed	chair	table	toilet	average
ISA (Q_1)	39	40	50	80	52.25
ISA (Q_1 & Q_2)	41	37	53	89	55.0
ISA (Q_1 & Q_3)	39	46	53	82	55.0
ISA (Q_1 & Q_2 & Q_3)	46	42	52	87	56.75
Linemod (min)	58	5	9	27	25
Linemod (max)	62	51	69	83	66
Linemod (all)	60	32	37	58	47

Method	input channel	bed	chair	table	toilet	mean
Sliding Shapes [13]	depth	33.5	29	34.5	67.3	41.075
[15] on instance seg.	depth	71	18.2	30.4	63.4	45.75
[15] on estimated model	depth	72.7	47.5	40.6	72.7	58.375
Deep Sliding Shapes [14]	depth	83.0	58.8	68.6	79.2	72.40
ISA based on Q_1&Q_2&Q_3	depth	52.0	36.0	46.5	67.7	50.55

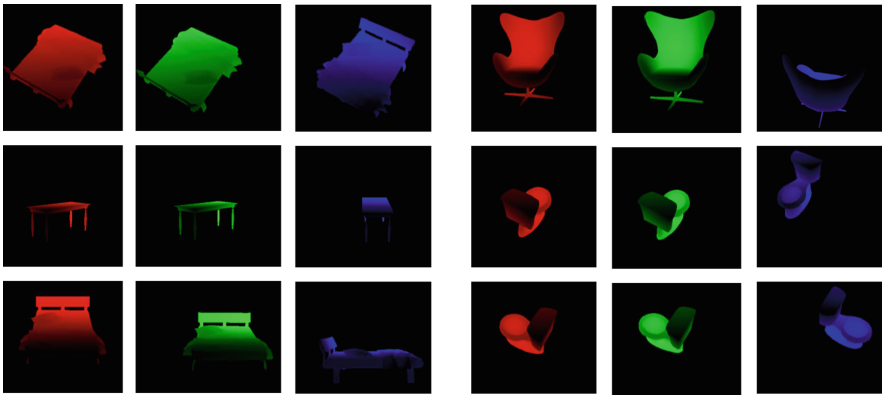

Fig. 7. Sample results generated by ISA on synthetic data: (for each triplet) each row is of per viewpoint, red is ground truth, green is estimation based on the quality function Q_1&Q_2&Q_3, blue is estimation based on the quality function Q_1 only. (Color figure online)

toilet category. The maximum recall value that Linemod achieve is of the *toilet* category. When we compute the mean for all recall values, Linemod best performs on the *bed* category.

4.2 Experiments on Real Data

Table 2 (right) depicts the comparison on 3D object detection. A short analysis on the table reveals that our architecture demonstrate 50% average precision. The highest value ISA reaches is on the *toilet* category, mainly because of the limited deviation in shape in between the instances. ISA next best performs on *bed*, with 52% mean precision. The accuracy on both the categories *bed* and *table* are approximately the same. Despite the fact that all forests used in the experiments undergo relatively a naive training process, the highest number of the instances during training are used for the chair category. However, ISA worst performs on this category, since the images in the test dataset have strong challenges of the instances, such as occlusion, clutter, and high diversity from the shape point of view. We lastly present sample results in Fig. 8. In these figures,

the leftmost images are the inputs of our architecture, and the 2^{nd} and the 3^{rd} columns demonstrate the estimations of the forests based on $Q_1\&Q_2\&Q_3$ and Q_1 only, respectively.

Fig. 8. Sample results generated by ISA on real data: (for each triplet) each row is for per scene. First column depicts depth images of scenes. Estimations in the middle belong to ISAs trained using $Q_1\&Q_2\&Q_3$, and hypotheses on the right are of ISAs trained on Q_1 only.

5 Conclusion

In this paper we have introduced a novel architecture, ISA, for category-level 6D object pose estimation from depth images. We have designed the proposed architecture in such a way that the challenges of the categories, intra-class variations, distribution shifts among source and target domains, can successfully be tackled while the 6D pose of unseen objects are estimated. To this end, we have engineered ISA based on the following technologies: We have firstly presented Semantically Selected Centers (SSC) for the category-level 6D object pose estimation problem. We next have utilized 3D skeleton structures and derived those as shape-invariant features. Using these features, we have represented the parts extracted from the instances of given categories, and employed privileged one-class learning based on these parts. We have performed graph matching during training so that the adaptation capability of the proposed architecture is improved across unseen instances. Experiments conducted on test images validate the promising performance of ISA. In the future, we are planning to improve the performance of ISA approaching the problem from transfer learning point of view.

References

1. Doumanoglou, A., Kouskouridas, R., Malassiotis, S., Kim, T.K.: Recovering 6D object pose and predicting next-best-view in the crowd. In: CVPR (2016)
2. Brachmann, E., Krull, A., Michel, F., Gumhold, S., Shotton, J., Rother, C.: Learning 6D object pose estimation using 3D object coordinates. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8690, pp. 536–551. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10605-2_35
3. Krull, A., Brachmann, E., Michel, F., Yang, M.Y., Gumhold, S., Rother, C.: Learning analysis-by-synthesis for 6D pose estimation in RGB-D images. In: ICCV (2015)
4. Wohlhart, P., Lepetit, V.: Learning descriptors for object recognition and 3D pose estimation. In: CVPR (2015)
5. Hodaň, T., et al.: BOP: benchmark for 6D object pose estimation. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11214, pp. 19–35. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01249-6_2
6. Michel, F., et al.: Global hypothesis generation for 6D object pose estimation. In: CVPR (2017)
7. Balntas, V., Doumanoglou, A., Sahin, C., Sock, J., Kouskouridas, R., Kim, T.K.: Pose guided RGBD feature learning for 3D object pose estimation. In: ICCV (2017)
8. Sock, J., Kasaei, S.H., Lopes, L.S., Kim, T.K.: Multi-view 6D object pose estimation and camera motion planning using RGBD images. In: 3rd International Workshop on Recovering 6D Object Pose (2017)
9. Brachmann, E., Michel, F., Krull, A., Yang, M., Gumhold, S., Rother, C.: Uncertainty-driven 6D pose estimation of objects and scenes from a single RGB image. In: CVPR (2016)
10. Kehl, W., Manhardt, F., Tombari, F., Ilic, S., Navab, N.: SSD-6D: making RGB-based 3D detection and 6D pose estimation great again. In: CVPR (2017)
11. Rad, M., Lepetit, V.: BB8: a scalable, accurate, robust to partial occlusion method for predicting the 3D poses of challenging objects without using depth. In: ICCV (2017)
12. Tekin, B., Sinha, S.N., Fua, P.: Real-time seamless single shot 6D object pose prediction. arxiv (2017)
13. Song, S., Xiao, J.: Sliding shapes for 3D object detection in depth images. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8694, pp. 634–651. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10599-4_41
14. Song, S., Xiao, J.: Deep sliding shapes for amodal 3D object detection in RGB-D images. In: CVPR (2016)
15. Gupta, S., Arbelez, P., Girshick, R., Malik, J.: Aligning 3D models to RGB-D images of cluttered scenes. In: CVPR (2015)
16. Garcia-Hernando, G., Kim, T.K.: Transition forests: learning discriminative temporal transitions for action recognition. In: CVPR (2017)
17. Shi, Z., Kim, T.K.: Learning and refining of privileged information-based RNNs for action recognition from depth sequences. In: CVPR (2017)
18. Lin, Y.Y., Hua, J.H., Tang, N.C., Chen, M.H., Liao, H.Y.M.: Depth and skeleton associated action recognition without online accessible RGB-D cameras. In: CVPR (2014)
19. Ben-David, S., Blitzer, J., Crammer, K., Pereira, F.: Analysis of representations for domain adaptation. In: NIPS (2007)

20. Hinterstoisser, S., et al.: Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes. In: Lee, K.M., Matsushita, Y., Rehg, J.M., Hu, Z. (eds.) ACCV 2012. LNCS, vol. 7724, pp. 548–562. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-37331-2_42
21. Rios-Cabrera, R., Tuytelaars, T.: Discriminatively trained templates for 3D object detection: a real time scalable approach. In: ICCV (2013)
22. Tejani, A., Tang, D., Kouskouridas, R., Kim, T.-K.: Latent-class hough forests for 3D object detection and pose estimation. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8694, pp. 462–477. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10599-4_30
23. Bonde, U., Badrinarayanan, V., Cipolla, R.: Robust instance recognition in presence of occlusion and clutter. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8690, pp. 520–535. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10605-2_34
24. Kehl, W., Milletari, F., Tombari, F., Ilic, S., Navab, N.: Deep learning of local RGB-D patches for 3D object detection and 6D pose estimation. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9907, pp. 205–220. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46487-9_13
25. Zach, C., Penate-Sanchez, A., Pham, M.: A dynamic programming approach for fast and robust object pose recognition from range images. In: CVPR (2015)
26. Sahin, C., Kouskouridas, R., Kim, T.K.: A learning-based variable size part extraction architecture for 6D object pose recovery in depth images. J. Image Vis. Comput. **63**, 38–50 (2017)
27. Sahin, C., Kouskouridas, R., Kim, T.K.: Iterative hough forest with histogram of control points for 6 DoF object registration from depth images. In: IROS (2016)
28. Sock, J., Kim, K., Sahin, C., Kim, T.K.: Multi-task deep networks for depth-based 6D object pose and joint registration in crowd scenarios. In: BMVC (2018)
29. Gupta, S., Girshick, R., Arbeláez, P., Malik, J.: Learning rich features from RGB-D images for object detection and segmentation. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8695, pp. 345–360. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10584-0_23
30. Hoffman, J., Gupta, S., Leong, J., Guadarrama, S., Darrell, T.: Cross-modal adaptation for RGB-D detection. In: ICRA (2016)
31. Baek, S., Shi, Z., Kawade, M., Kim, T.K.: Kinematic-layout-aware random forests for depth-based action recognition. In: BMVC (2017)
32. Cao, J., Tagliasacchi, A., Olson, M., Zhang, H., Su, Z.: Point cloud skeletons via Laplacian based contraction. In: Shape Modeling International Conference (SMI) (2010)
33. Wu, Z., et al.: 3D ShapeNets: a deep representation for volumetric shape modeling. In: CVPR (2015)
34. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from RGBD images. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7576, pp. 746–760. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33715-4_54