PoseFromGraph: Compact 3-D Pose Estimation using Graphs

Meghal Dani
dani.meghal@tcs.com
TCS Research
Delhi, India

Additya Popli
additya.popli@research.iiit.ac.in
IIIT Hyderabad
Telangana, India

Ramya Hebbalaguppe
ramya.hebbalaguppe@tcs.com
TCS Research
Delhi, India

ABSTRACT

With the rising need for reliable and real-time pose estimation in resource constrained environments such as smartphones, IoT devices, and head mounts, requires a compact and accurate pose estimation framework. To this end, we propose PoseFromGraph, a light weight 3D pose estimation framework that first generates a compact graph representation to estimate the pose of generic objects and subsequently overlays the 3D model utilizing the estimated pose. The inputs to the network are: a graph obtained by skeletonizing the 3D meshes using the prairie-fire analogy and the RGB image, and the output is the 3D pose of the object. The introduction of 3D shapes to the architecture makes our model category-agnostic. Unlike computationally expensive multi-view geometry and point-cloud based representations to estimate pose, our approach uses a message passing network to incorporate local neighborhood information at the same time maintaining global shape property in a graph by optimizing a neighborhood preserving objective. PoseFromGraph surpasses the state-of-the-art [29] pose estimation methods in terms of accuracy achieving 84.43% on the Pascal3D dataset, and at the same time yields 11× and 4× reduction in the space and time complexity respectively.

CCS CONCEPTS

• **Computing methodologies** → *Computer vision.*

KEYWORDS

Node embedding, skeletonisation, pose estimation, 3D registration, internal overlay

ACM Reference Format:
Meghal Dani, Additya Popli, and Ramya Hebbalaguppe. 2020. PoseFromGraph: Compact 3-D Pose Estimation using Graphs. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Augmented Reality (AR) enhances users' perception through seamless interaction with virtual objects in a real scene. One such example of enhanced perception are guided surgeries, where a medical practitioner can benefit from accurate 3D model registration. Though modern AR toolkits such as Vuforia [2] and MetaIO [1] (now acquired by Apple Inc.) provide support for these features, they are typically opaque in terms of the underlying algorithms and their implementation techniques, and hence leave very little scope for customization [35]. Thus, we work in the direction of democratisation of object registration process and aim to provide a novel technique for efficiently performing object registration, where we estimate the 3-dimensional pose of a physical object to overlay digital content such as annotations, 3D models, and animations for an immersive user experience.

Our method utilizes 3D shape information along with RGB images. Previous works in the literature only employ 3D shapes for rigid body objects in the form of meshes, point clouds or multi-view images [29], [36], [22]. To the best of our knowledge, this is the first work that uses compact graphs in an unsupervised manner to represent 3D shape information in generic rigid body objects. We leverage EdgeConv [25], a differentiable and permutation invariant module. It incorporates local graph information by learning embeddings for nodes in the graph and keeps global shape information intact. We then make use of a combination of our custom network and standard convolution neural networks (CNNs) to match this representation with the visual scan data of the object.

In a real-world scenario, we need systems that interact with objects of various geometries. Our intuition is that 3D shapes capture latent semantic and compositional information. Using these 3D shapes not only promotes accurate registration but is also useful to make it category-agnostic. We claim that the general shape information of an object can be captured with a few nodes and need not be a huge set of points on a point cloud which tends to be noisy. Moreover, graphs provide concise, informative, and easily computable information as compared to point-clouds and multi-view [21] [23]. With the desirable properties such as invariance to rotation, translation, and scaling make graphs an apt choice for representing 3D objects and subsequent matching process.

The major contributions of our work are:

1. To the best of our knowledge, our approach, PoseFromGraph is the first work in pose estimation that employs unsupervised graph representation to represent 3D shapes for generic rigid body objects, facilitating a category-agnostic framework.
2. We surpass the state-of-the-art accuracy [29] by achieving 84.43% on the standard Pascal3D Dataset.
3. PoseFromGraph reduces the space and memory by factors 4× and 11× respectively w.r.t. SOA[29] making it suitable for on-device AR applications.

2 RELATED WORK

Many vision based algorithms have been developed for pose estimation. Perspective-n-Point (PnP) algorithm [20] used for 3D to 2D correspondences in pose estimation though, gained lot of interest among researchers [28, 34] suffers from heavy online run-time computation cost and does not deal with outliers efficiently [11]. PnP with RANdom SAmple Consensus (RANSAC) [32] in loop was developed to deal with outliers. The method is robust with textured

Conference'17, July 2017, Washington, DC, USA
https://doi.org/10.1145/nnnnnnn.nnnnnnn

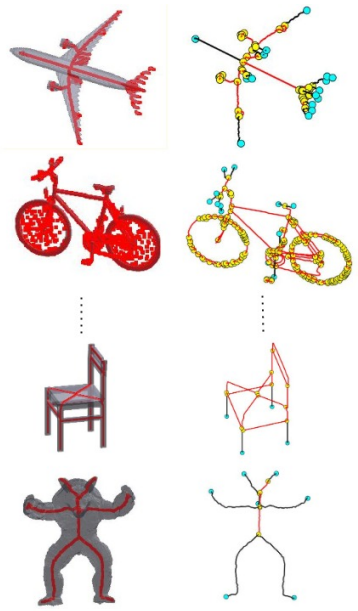

Figure 1: Illustration of skeletons generated and their respective graphs using prairie-fire analogy [5]. The blue and yellow nodes represent the leaf and non-leaf nodes respectively; edges are represented in red color. The bottom-most representation is on unseen data (Armadillo object [19])

objects but fails in case of occluded scenes. Later iterative closest point (ICP) [4] algorithm became popular for registration in varying fields. Being successful relatively, this method is susceptible towards converging to local minima [6], [28] and, therefore, a close initial manual alignment is necessary for convergence. Deformable Part Model (DPM) [10] was also introduced to be very efficient. Recently, CNNs [18] have been shown to outperform DPM [10] based methods for recognition tasks [8, 12, 15]. DPMs explicitly model part appearances and their deformations whereas, the CNN architecture allows such relations to be captured implicitly using a hierarchical convolutional structure. Since then, multitude of work [13, 17, 22, 36] has been done in development of CNN models to learn correspondence of 3D model to 2D image. PoseCNN [28] was one of the initial works in this domain and delivered promising results. Few researchers made use of segmentation masks instead of annotation data [24]. But these binary masks lacked texture, color and other relevant information which presents difficulties to resolve ambiguities in the mask representation [24]. They rely heavily on segmentation accuracy and do not do well with occluded image datasets.

These approaches were category dependent i.e., they could identify pose only for those objects on which the model has been trained on. Y. Xiao et al. [29] introduced a category agnostic model that made use of 3D shape information to define pose. The model is complex and computationally heavy as it requires multiview and point-cloud for 3D registration. Our work proposes a compact graph representation and a lighter model that can be used on frugal devices and have improved performance in memory constrained environment. Graphs have been studied well in humans [26, 30, 31] but not in rigid body objects. In addition to the compact representation, accuracy of our approach surpasses the S.O.A technique Pose From Shape [29].

3 PROPOSED ARCHITECTURE

We propose PoseFromGraph, a light-weight pose estimation approach that utilises 3D shapes and images to extract rich features using deep neural networks. Figure 2 shows the detailed architecture of PoseFromGraph; for each given RGB image and its corresponding CAD model, we extract and concatenate features using a convolutional neural network(CNN) and a graph neural network(GNN).

3.1 Skeletonization and Graph Pruning

Meshes of the 3D models are voxelized and converted to skeletons via thinning [16]. These skeletons provide a compact and expressive shape abstraction to 3D shapes and volumes. As shown in Figure 1, the graph representations are generated and pruned using a threshold value of 10, i.e., short branches with less than 10 voxels between two nodes are removed. The graph pruning reduces the number of nodes in a graph by 41.67% [Additya: Is it not more than 41%? from point cloud to our graph?] on an average on PASCAL3D dataset preserving the topological information of original object and visual parts. The significant feature here is that these graphs are invariant under Euclidean Transformations such as rotations and translations and devoid of unwanted noise in the graph in the form of parallel edges, loops and short branches. This makes them an optimal choice of representation for 3D objects [21], [23], beating the traditional representational methods (e.g pointcloud or multiviews [21, 23]) both in terms of accuracy and space complexity.

3.2 Feature Extraction and Fusion

We make use of two separate feature extractors; a GNN or message passing network to encode the shape of the object and a CNN to encode the RGB image. Unlike previous works [22, 29, 36], we use GNNs because: (i) graphs are considered an ideal choice for representing data without a well-defined structure and (ii) GNNs with message passing algorithms have proven to be effective for pose-estimation and related tasks if a graphical representation is available [26, 30, 31].

Section 3.1 shows how [16] is employed generate nodes for the input graph for our network. The initial feature vector for each of these nodes is a three-dimensional vector consisting of the x, y, and z coordinates of the corresponding point in the given point cloud.

$$x_i' = \square_{max}\{h_\theta(x_i||x_j - x_i)|j \in N(i)\} \quad (1)$$

$$g_i = \square_{sum}\{x_j|j \in V_i\} \quad (2)$$

This graph is then fed to the graph network consisting of multiple message passing layers in order to accumulate surrounding information for each node in the network, as described in Equation 1, where x_i and x_i' refer to the current and updated feature vectors respectively for the i^{th} node, $N(i)$ refers to its neighbourhood which consists of its k nearest neighbours in the feature space and $||$ represents concatenation of feature vectors.

Finally, the node-level features of the all the nodes are summed to obtain the graph-level features as shown in Equation 2. For RGB

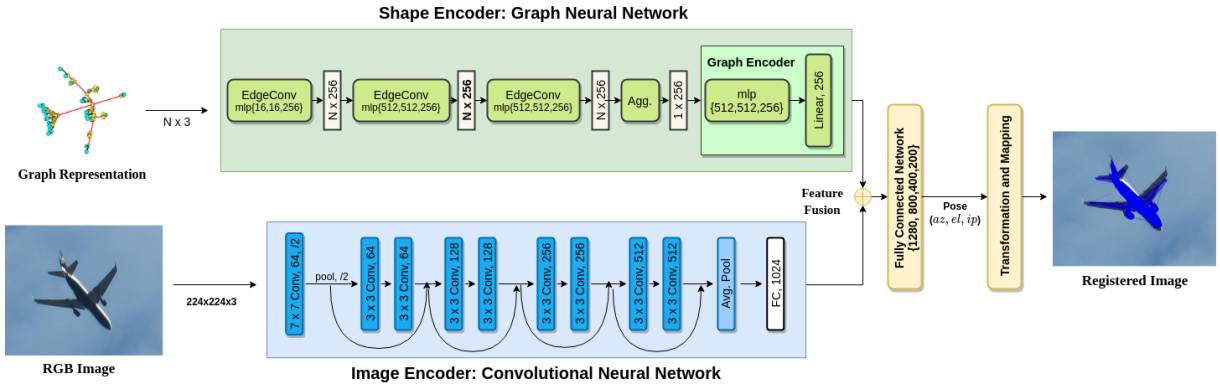

Figure 2: Proposed architecture PoseFromGraph comprises of a shape encoder (green block) and an image encoder (blue block); the shape encoder takes graph of 3D model as input and extracts graph features using EdgeConv, a message passing network [25] and image encoder takes RGB image as input and feeds it to a ResNet-18 block. Features extracted are concatenated and passed to FCN layers for final pose estimation. Then, CAD model projected onto the RGB image with respect to the estimated pose.

* Annotation file in the dataset contains the information about RGB images and their corresponding CAD models.

images, we utilize ResNet-18 [14], a standard CNN model for extraction the image features (a vector of dimension 1024). The features are then fused (concatenated) to obtain the final feature vector. The image and shape features encode complementary information and thus help in better pose estimation [29].

3.3 Pose Estimation

The concatenated feature vector (as explained in 3.2) is passed to a Fully Connected Network (FCN) for pose estimation with each layer (1280-800-400-200) followed by Batch Normalisation and ReLU activation. Finally, we obtain the output pose of object with respect to the RGB image in the reference frame in term of azimuth (az), elevation (el), and in-plane rotation (ip) angles. Each angle $\phi \in \{az, el, ip\}$ is divided into b bins uniformly such that our model outputs l labels $\in (0, b-1)$ and δ offsets $\in [-1, 1]$. Thus, our training data consists of 3 inputs to the network $(img_i, graph_i, ori_i)$, where, img_i is input image, $graph_i$ is input graph of 3D model and ori_i is orientation given in the annotation file. The Euler angles ori_i are converted to bin labels l encoded as one-hot vectors and relative offsets δ within the bins.

Our network combines the outputs from pose regression and pose classification i.e., offsets and probabilities respectively. For classification, we make use of the standard Cross Entropy Loss $\mathcal{L}_{\text{CE}}$ and Smooth $L1$ loss (Huber Loss) for regression $\mathcal{L}_{\text{L1}}$. The final loss, $\mathcal{L}_{\text{Cla-Reg}}$, is a summation of $\mathcal{L}_{\text{CE}}$ and $\mathcal{L}_{\text{L1}}$:

$$\mathcal{L}_{\text{Cla-Reg}} = \sum_{i=1}^{N} \sum_{\phi} \mathcal{L}_{\text{CE}}(l_{i,\phi}, prob_{\phi}(img_i, graph_i)) + \mathcal{L}_{\text{L1}}(\delta_{i,\phi}, reg_{\phi,l_{i,\phi}}(img_i, graph_i)) \quad (3)$$

where $prob_{\phi}(img_i, graph_i))$ refer to the predicted probabilites for parameterised by ϕ w.r.t. the input image and shape representation followed and $reg_{\phi,l_{i,\phi}}(img_i, graph_i))$ correspond to the regressor predicted offset.

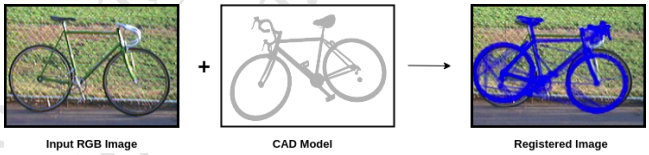

Figure 3: Result of pose estimation on bicycle on [27] Dataset

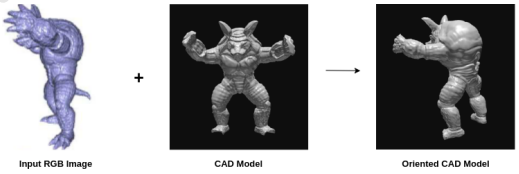

Figure 4: A sample illustration of pose estimation using PoseFromGraph on unseen data, in our case, Armadillo

4 PERFORMANCE EVALUATION

4.1 Dataset

For the purpose of training our pose estimator we use a state-of-the-art dataset, Pascal3D+ [37], which comprises of images, annotations and CAD models for 12 rigid body object categories including aeroplane, bicycle, car, etc. The dataset is compilation of training and validation set images from PASCAL VOC 2012 [9] and ImageNet [7].

4.2 Baseline and Evaluation

We regard PoseFromShape [29] as the baseline that generates PointClouds, random Multiviews and feeds them to a ResNet-18 model [14] for pose estimation. We report our results in terms of accuracy denoted by $Accuracy_{\pi/6}$ where $\pi/6$ is the threshold value. The prediction is considered correct if the error of the estimated pose is within this threshold. We first evaluate our method on seen categories, i.e, objects belonging to the categories present in the training

Table 1: Details on computational efficiency, memory footprint, and number of parameters: Note the reduction in inference time and GPU memory required.

	Test Time (per instance)	Number of Parameters
Baseline[29]	0.26 sec	23,009,664
PoseFromGraph	0.06 sec	15,049,192

Table 2: Performance Comparison in terms of $Accuracy_{\pi/6}$ on Pascal3D+ [37] dataset.

Algorithm	Category-Agnostic	$Accuracy_{\pi/6}$
Tulsiani et. al[36]	×	76.00%
Su et. al* [22]	×	82.00%
Kundu et. al[17]	×	74.00%
Grabner et. al[33]	√	81.33%
PoseFromShape [29]	√	**82.66%**
PoseFromGraph	√	**84.43%**

* Not trained on ImageNet data but trained on ShapeNet Renderings.

data. Following the testing protocol of [29], our method achieves state-of-the-art accuracy of 84.43% (Table 2), with computation time and memory requirements reduced by approximately 4× and 11×, as shown in Table 1. These results are on par with category-specific approaches discussed [17, 22, 36]. Some of the models developed [17] make use of select category of objects instead of entire dataset, making them biased. Even though underlying idea is competitive, they fail to generalize on a wide variety of objects.

In addition to Table 2, we also show visually the pose estimation on a RGB image from the dataset in Figure 3. The registration looks reasonable owing to accurate pose estimation. To illustrate the unseen categories, we show a sample on Armadillo images obtained from Stanford 3D scanning repository [19] on which the PoseFromGraph (our network) is not trained on and our framework learns the underlying structure. The graph generated (as shown in Figure 1) is meaningfully capturing the significant parts including well defined nodes for head, ears, hands and legs. Adding to it, the pose estimation (as shown in Figure 4) is quite accurate even when we consider completely different category of objects (i.e., Armadillo unlike rigid body objects). This demonstrates the robustness of graph generation procedure that encodes the shape accurately and also the pipeline developed.

5 FUTURE WORK AND CONCLUSION

We present a compact neural network architecture for deep pose estimation termed PoseFromGraph, mainly aimed at AR applications. We demonstrated the benefits of this approach in terms of computational efficiency, model parameters, with a negligible trade-off in accuracy. We show PoseFromGraph is promising for pose estimation on generic unseen objects without the need for re-training the network. To facilitate the compactness of neural networks, we utilize graphs to represent 3D shapes. This work pushes forward the state-of-the-art and open several avenues for future research in pose estimation utilizing graphs.

In the future, we intend to work on extending our work to compress neural networks via re-engineering/revising graph pruning and feature extractor modules. To see the light of our research work problems where a 3D object registration to a live feed is essential, a compact model is the need of the hour. We aim to achieve this through additional techniques using off the shelf model compression techniques such as ONNX [3] and TensorFlowLite frameworks.

REFERENCES

[1] [n.d.]. Metaio. https://en.wikipedia.org/wiki/Metaio. Accessed: 2020-06-11.
[2] [n.d.]. Model Target Generator. https://library.vuforia.com/articles/Solution/model-target-generator-user-guide.html. Accessed: 2020-06-11.
[3] J. Bai, F. Lu, K. Zhang, et al. 2019. Onnx: Open neural network exchange. GitHub repository (2019).
[4] P J Besl and N D McKay. 1992. Method for registration of 3-D shapes. In Sensor fusion IV: control paradigms and data structures, Vol. 1611. International Society for Optics and Photonics, 586–606.
[5] H Blum et al. 1967. A transformation for extracting new descriptors of shape. Models for the perception of speech and visual form 19, 5 (1967), 362–380.
[6] Brounstein et al. 2011. Towards real-time 3D US to CT bone image registration using phase and curvature feature based GMM matching. In MICCAI. Springer, 235–242.
[7] Deng et al. 2009. Imagenet: A large-scale hierarchical image database. In 2009 IEEE CVPR. Ieee, 248–255.
[8] Donahue et al. 2013. A deep convolutional activation feature for generic visual recognition. arXiv preprint arXiv:1310.1531 1 (2013).
[9] Everingham et al. [n.d.]. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.
[10] Felzenszwalb et al. 2009. Object detection with discriminatively trained part-based models. IEEE TPAMI 32, 9 (2009), 1627–1645.
[11] Ferraz et al. 2014. Very fast solution to the PnP problem with algebraic outlier rejection. In CVPR. 501–508.
[12] Girshick et al. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE CVPR. 580–587.
[13] Georgakis et al. 2019. Learning local rgb-to-cad correspondences for object pose estimation. In IEEE ICCV. 8967–8976.
[14] He et al. 2016. Deep residual learning for image recognition. In Proceedings of the IEEE CVPR. 770–778.
[15] Krizhevsky et al. 2012. Imagenet classification with deep convolutional neural networks. In Adv Neural Inf Process Syst. 1097–1105.
[16] Kollmannsberger et al. 2017. The small world of osteocytes: connectomics of the lacuno-canalicular network in bone. New J. Phys 19, 7 (2017), 073019.
[17] Kundu et al. 2018. iSPA-Net: Iterative Semantic Pose Alignment Network. In ACM-MM. 967–975.
[18] LeCun et al. 1989. Backpropagation applied to handwritten zip code recognition. Neural Comput. 1, 4 (1989), 541–551.
[19] Levoy et al. 2005. The Stanford 3D scanning repository. URL http://www-graphics. stanford. edu/data/3dscanrep 5 (2005), 7.
[20] Lepetit et al. 2009. Epnp: An accurate o (n) solution to the pnp problem. IJCV 81, 2 (2009), 155.
[21] Natali et al. 2011. Graph-based representations of point clouds. Graphical Models 73, 5 (2011), 151–164.
[22] Su et al. 2015. Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In Proceedings of the IEEE ICCV. 2686–2694.
[23] Tagliasacchi et al. 2016. 3d skeletons: A state-of-the-art report. In Computer Graphics Forum, Vol. 35. Wiley Online Library, 573–597.
[24] Wu et al. 2018. Real-time object pose estimation with pose interpreter networks. In 2018 IEEE/RSJ IROS. IEEE, 6798–6805.
[25] Wang et al. 2019. Dynamic graph cnn for learning on point clouds. Acm Transactions On Graphics (tog) 38, 5 (2019), 1–12.
[26] Wang et al. 2020. Global Relation Reasoning Graph Convolutional Networks for Human Pose Estimation. IEEE Access 8 (2020), 38472–38480.
[27] Xiang et al. 2014. Beyond PASCAL: A Benchmark for 3D Object Detection in the Wild. In WACV.
[28] Xiang et al. 2017. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. arXiv preprint arXiv:1711.00199 (2017).
[29] Xiao et al. 2019. Pose from Shape: Deep Pose Estimation for Arbitrary 3D Objects. arXiv preprint arXiv:1906.05105 (2019).
[30] Zhang et al. 2019. Human pose estimation with spatial contextual information. arXiv preprint arXiv:1901.01760 (2019).
[31] Zhao et al. 2019. Semantic graph convolutional networks for 3D human pose regression. In IEEE CVPR. 3425–3435.

[32] M A Fischler and R C Bolles. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM 24, 6 (1981), 381–395.
[33] A Grabner, P M Roth, and V Lepetit. 2018. 3d pose estimation and 3d model retrieval for objects in the wild. In Proceedings of the IEEE CVPR. 3022–3031.
[34] F Hagelskjær and A. G. Buch. 2019. PointPoseNet: Accurate Object Detection and 6 DoF Pose Estimation in Point Clouds. arXiv preprint arXiv:1912.09057 (2019).
[35] E Marchand, H Uchiyama, and F Spindler. 2015. Pose estimation for augmented reality: a hands-on survey. IEEE T VIS COMPUT GR 22, 12 (2015), 2633–2651.
[36] S Tulsiani and J Malik. 2015. Viewpoints and keypoints. In Proceedings of the IEEE CVPR. 1510–1519.
[37] Y Xiang, R Mottaghi, and S Savarese. 2014. Beyond pascal: A benchmark for 3d object detection in the wild. In IEEE WACV. IEEE, 75–82.