Jing Zhu

Personal Information

Status:	**PH.D. STUDENT**
Program:	**Computer Science and Engineering**
School:	**Tandon School of Engineering, New York University**
Period:	**From 2016-01 to 2020-05**

Biography

I was a Ph.D. student at New York University and advised by Professor Yi Fang. During my Ph.D. period, I was a research assistant in NYU Multimedia and Visual Computing Lab. I am broadly interested in 3D Computer Vision and Deep Learning. Now I am a research scientist at Apple Inc..

Research Project: Global geometric feature learning: Depth Image and 3D Mesh

Description

3D shape retrieval has become an important topic in computer vision field with a wide range of applications in engineering, manufacturing, product design, and the medical field. Although cross-domain shape retrieval has received many attentions for years, most of them are on sketch-based or image-based shape retrieval. Recently, with the increase of the depth image datasets, researchers start to look at the depth image-based shape retrieval problem. We propose to learn a domain-invariant representation for depth image-based shape retrieval using two discriminative neural networks, one for each domain. By projecting the cross-domain data into same feature space, the comparison between depth image and 3D model can be conducted directly. To validate the performance of our proposed method, we comprehensively evaluate our algorithm on one large depth image dataset and two 3D model datasets by conducting experiments with various settings. In all experiments, our method outperforms the state of the arts.

Method

We use two deep neural networks in our proposed model, one for the depth images and the other for 3D shapes. To connect the two networks, we define a loss function with constraints on both inter-class and intra-class margin, which maps the distinctive inputs into the same target space by minimizing the (intraclass) difference between cross-domain data within the same category while maximizing the (inter-class) variation among data from different categories. Finally, the outputs of the leaned networks are considered as the domain-invariant representation for given cross-domain data, and relevant 3D shape can be retrieved by directly comparing the output features from the networks.

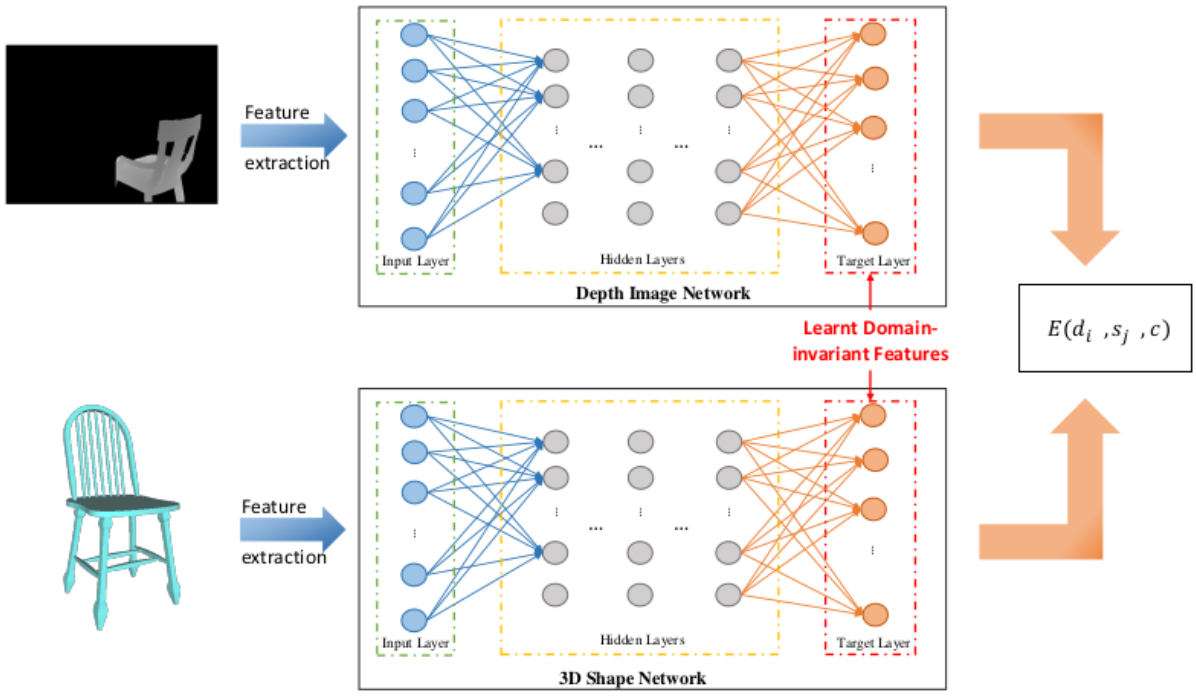


Figure 1: The structure of our proposed method, where two neural networks with same architecture, one for each domain, are used to handle the cross-domain issues. The domain-invariant feature can be learned by connecting the two networks with a loss function on the outputs.

Results

The experimental results on popular datasets, where depth images are from NYU Depth V2 dataset and 3D models come from SHREC 2014 database and ModelNet dataset, suggest that our proposed model significantly outperform other state-of-the-art approaches. Once the networks are trained, we can perform efficient shape retrieval on given depth image queries since it only requires some matrix computation.

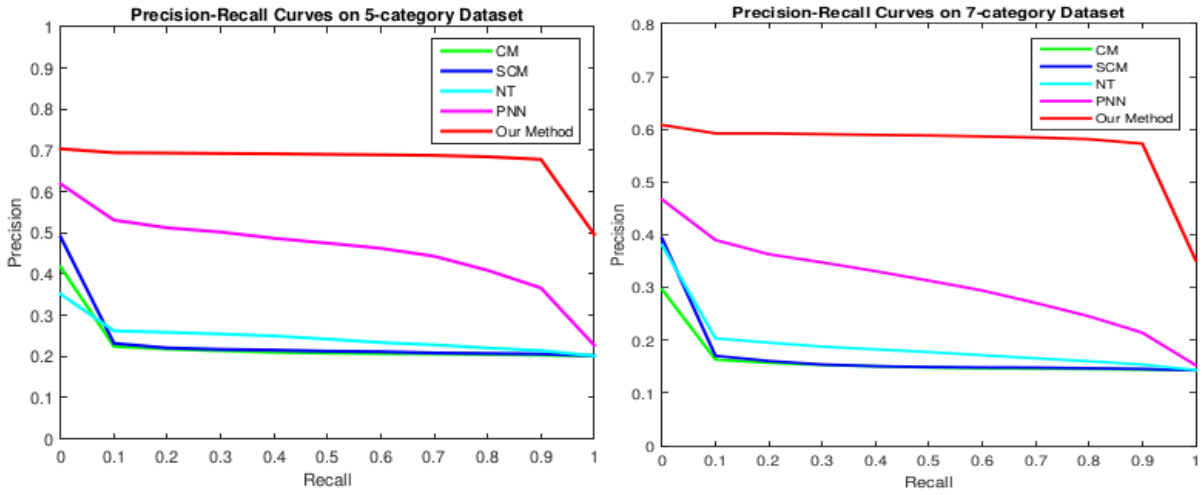


Figure 2: Precision-Recall plot for performance comparison of state-of-the-art methods on NYU Depth V2 dataset and SHREC 2014 benchmark. The left plot shows the comparisons on 5-category dataset and the right one shows the comparisons on 7-category dataset.

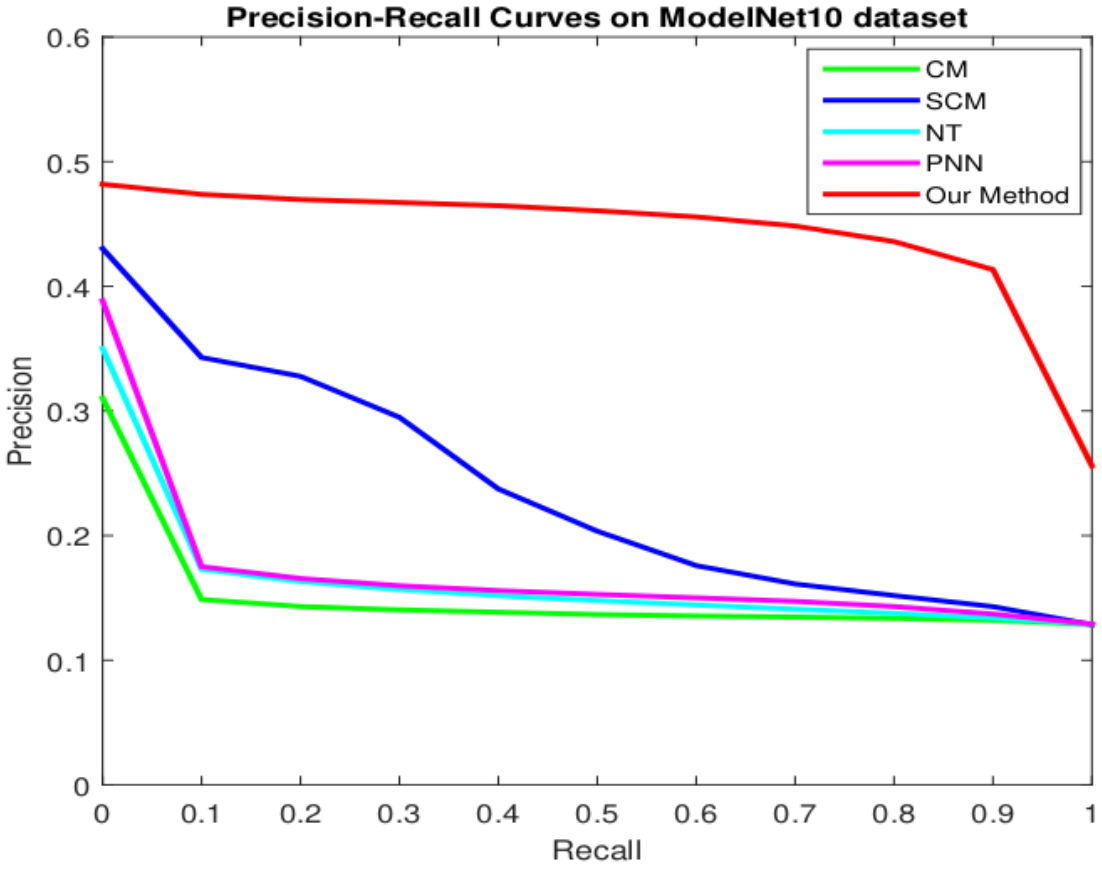


Figure 3: Precision-Recall plot for performance comparison of state-of-the-art methods on NYU Depth V2 dataset and ModelNet 10 dataset.

Research Project: Local geometric feature learning: RGB Image and 3D Voxelized Model

Description

In early 3D model generation systems, new models were usually generated by mixing up several parts from the existing models. With the emergence of depth sensors, such as Microsoft Kinect and 3D LiDAR, it becomes possible to reconstruct 3D models from lower-cost captured RGB-D images or point clouds. However, processing the sensor captured images or point clouds is kind of complicated and time-consuming, especially in some state-of-the-art methods that infer 3D models from multi-view images or depth maps. In this work, we consider constructing a generative model that could effectively synthesize high-quality 3D models without any image or depth map inputs. We propose to learn a GAN-based 3D model generator from 2D images and 3D models simultaneously. The experimental results demonstrate that our proposed framework can synthesize high-quality 3D models.

Method

We build our framework on 3D generative adversarial networks with an enhancer network for better training a 3D model generator. The enhancer contains two deep convolutional neural networks, and learns features from images in an adversarial manner. The high-level learned image features from the enhancer are fed into the 3D model generator for better generation. We train the two networks together, so that our 3D model generator can be learned from 3D data and 2D data simultaneously. Once the framework has been trained, given a random vector, the enhancer first generates corresponding high-level image features, and then the 3D model generator can synthesize a volumetric 3D model based on the image features.

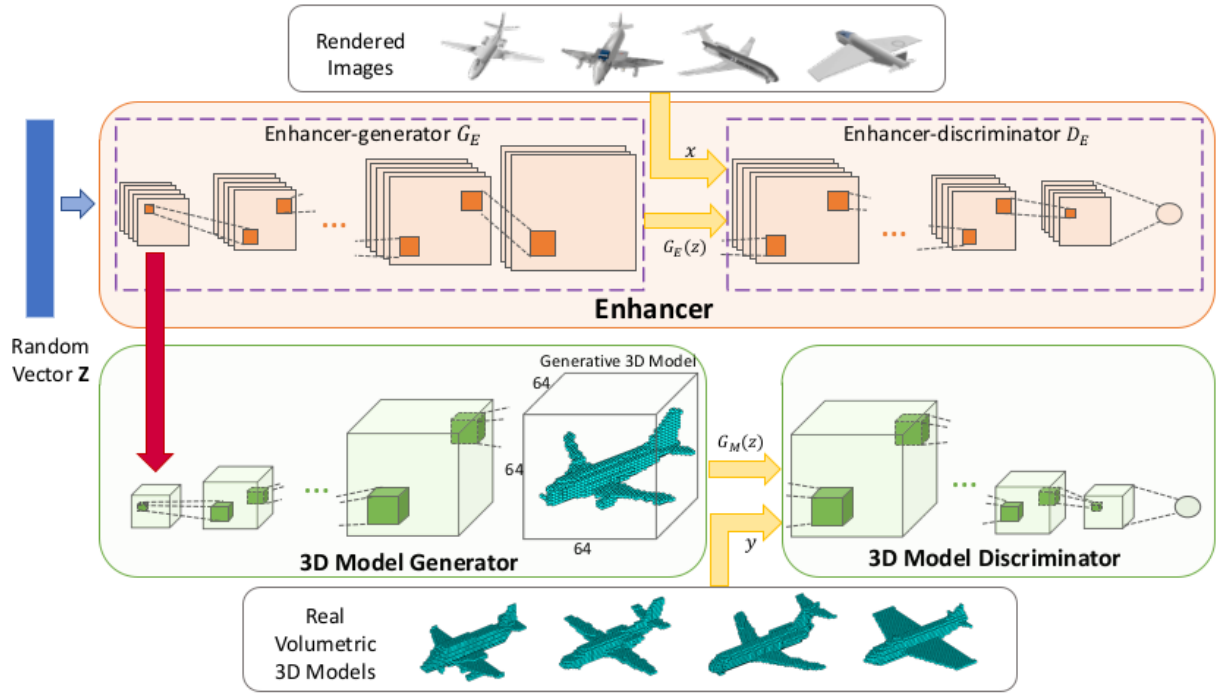
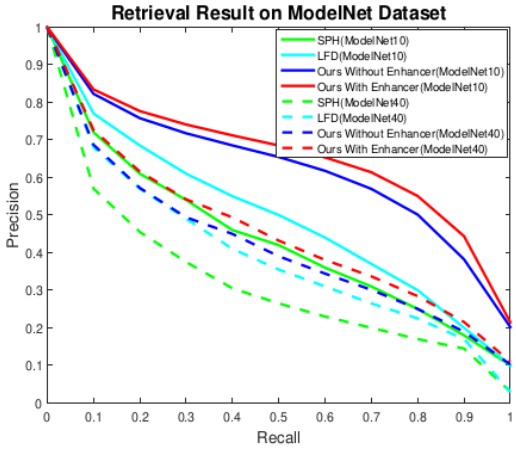


Figure 4: The framework of our proposed method for training. It consists of three parts: an enhancer, a 3D model generator and a 3D model discriminator. The enhancer contains two deep neural networks and learns features from rendered images via an adversarial manner. The 3D model generator is trained on 3D data with the 3D model discriminator. By feeding the outputs from the first layer of the enhancer into the 3D model generator, the learned high-level image feature from enhancer can be utilized for better training a 3D model generator.

Results

To comprehensively validate our proposed framework, we conduct three different experiments on large-scale 3D model datasets, including 3D model generation, shape classification and shape retrieval. The experimental results demonstrate that our proposed framework can synthesize high-quality 3D models. Moreover, the unsupervised shape features learned by our framework can achieve superior performance over most of the state-of-the-art methods for shape classification and shape retrieval on ModelNet



Precision-Recall plots for shape retrieval comparison of state-of-the-art methods on two benchmarks (ModelNet10 and ModelNet40) of ModelNet dataset.

dataset.

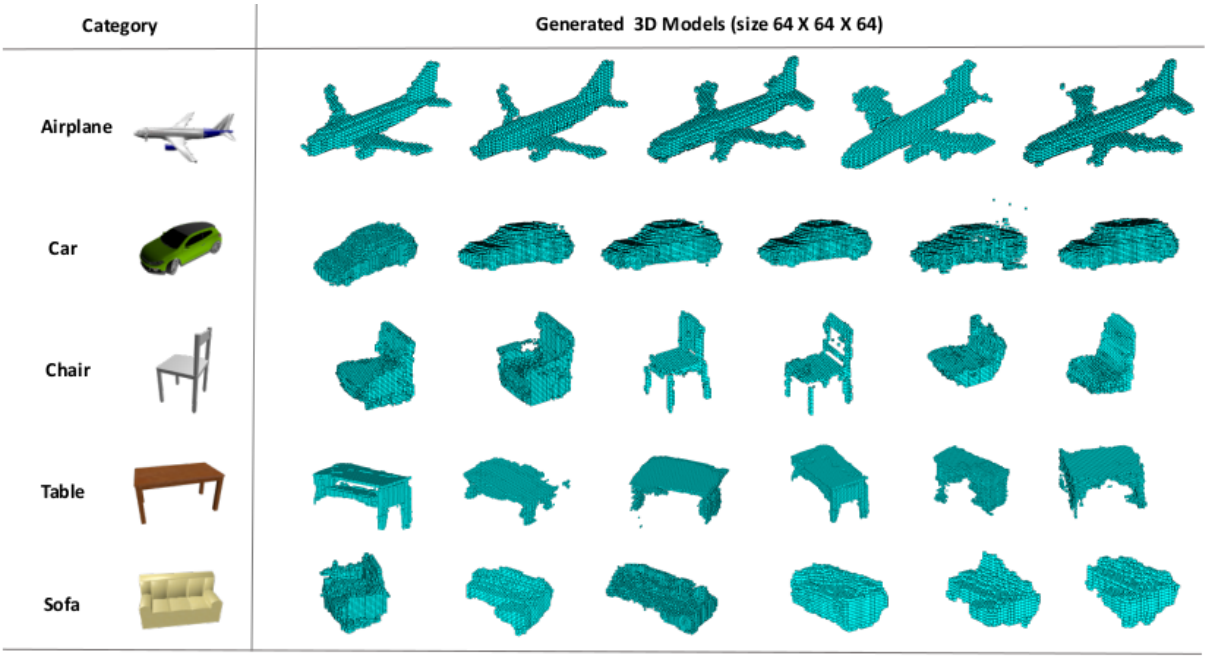


Figure 5: Examples of 3D models generated by our proposed method, one row for each category (e.g. airplane, car, chair, table, sofa).

Method	Supervised?	ModelNet10 (%)	ModelNet40 (%)
3D ShapeNets[222]	✓	83.54	77.32
VoxNet [141]	✓	92.00	83.00
Geometry Image [193]	✓	88.40	83.90
PointNet [155]	✓	77.60	–
GIFT [8]	✓	92.35	83.10
FusionNet [85], fine-tuned	✓	93.11	90.80
SPH [108]	×	79.79	68.23
LFD [27]	×	79.87	75.47
VConv-DAE [184]	×	80.50	75.50
3D-GAN [220]	×	91.00	83.30
Our Method w/o Enhancer	×	**88.88**	**85.53**
Our Method with Enhancer	×	**91.63**	**87.85**

Figure 6: Performance comparison of shape classification with state-of-the-art methods on two benchmarks (ModelNet10 and ModelNet40) of ModelNet dataset.

Research Project: Local geometric feature learning: Depth Image and 3D Voxel

Description

A good local geometric descriptor enables a wide range of applications, such as semantic segmentation, point matching, and scene reconstruction. Though many methods have been proposed to learn descriptors from multiple formats of 3D data, such as 3D mesh, 3D scan data – RGB-D images, volumetric point patches, most of them focused on learning a global descriptor. However, how to obtain a good local descriptor remains a challenging but interesting computer vision task. In this paper, we aim to learn a robust local 3D descriptor that can be used as representations to match the sample points in the fragments, and then we can compute the rigid transformation matrix (including rotations, scales, translations) between matching points in any two fragments, finally align the two fragments using the computed transformation.

Method

We introduce an siamese-network-based adversarial enhancer with an opposite loss to the local descriptor generator. That is to say, during the training process, the adversarial enhancer is learning to minimize the distances between the generator- learned features for non-match pairs while to maximize the distances for match pairs. To compete with the enhancer and win the two-player game, it enforces the local descriptor generator to learn a robust and powerful descriptor for giving volumetric point patches with very small distances between the learned features for match pairs but very large differences between learned features for the non- match pairs. As a result, the performance of the local descriptor generator can be improved.

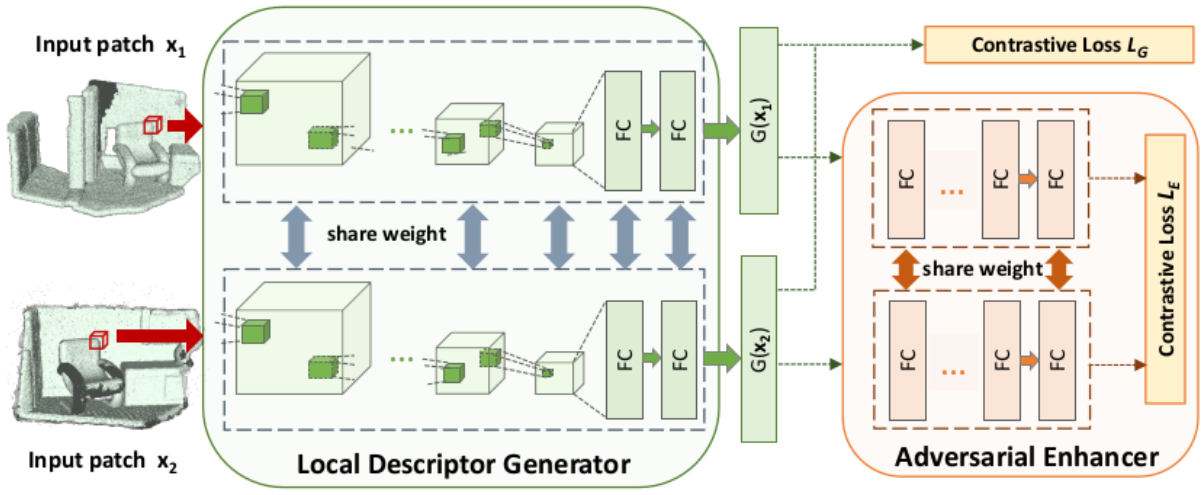


Figure 7: The framework of our proposed method. It consists of two parts: a local descriptor generator and an adversarial enhancer. The local descriptor gener- ator contains a siamese deep convolutional neural networks, while the adversarial enhancer has a siamese networks with only simple fully-connected layers. Optimiz- ing with the contrastive loss L E which is opposite to the loss in the local descriptor, the enhancer is introduced to boost the generator in adversarial manner.

Results

To comprehensively validate our proposed framework, we conduct two different experiments on large-scale RGB-D reconstruction datasets, including keypoint matching and geometry registration. The experimental results demonstrate that our method can learn a robust representation for local 3D volumetric point patches to solve classic local matching problems. Moreover, though our model is trained on fewer samples for less time, it outperforms the state-of-the-arts methods with lower keypoint matching error and higher geometry registration precisions.

Method	Recall (%)	Precision (%)
FPFH [173] + RANSAC	44.2	30.7
Spin-Images [103] + RANSAC	51.8	31.6
3DMatch [240] + RANSAC	66.8	40.1
Ours without Adversarial Enhancer + RANSAC	**69.1**	**40.5**
Ours with Adversarial Enhancer + RANSAC	**72.0**	**42.9**

Figure 8: The precision and recall comparisons of geometry registration with state-of-the-art methods on real-world scan scenes constructed from the SUN3D and 7-scenes datasets.

Method	Recall (%)	Precision (%)
Super 4PCS [144]	17.8	10.4
FPFH [173]	44.9	14.0
Variant FPFH [32]	59.2	19.6
FPFH [173] + RANSAC	46.1	19.1
Spin-Images [103] + RANSAC	52.0	21.7
3DMatch [240] + RANSAC, fine-tuned	65.1	25.2
Ours without Adversarial Enhancer + RANSAC	**58.6**	**25.3**
Ours with Adversarial Enhancer + RANSAC	**60.3**	**28.3**

Figure 9: The precision and recall comparisons of geometry registration with state- of-the-art methods on the synthetic scenes in the augmented ICL-NUIM dataset.

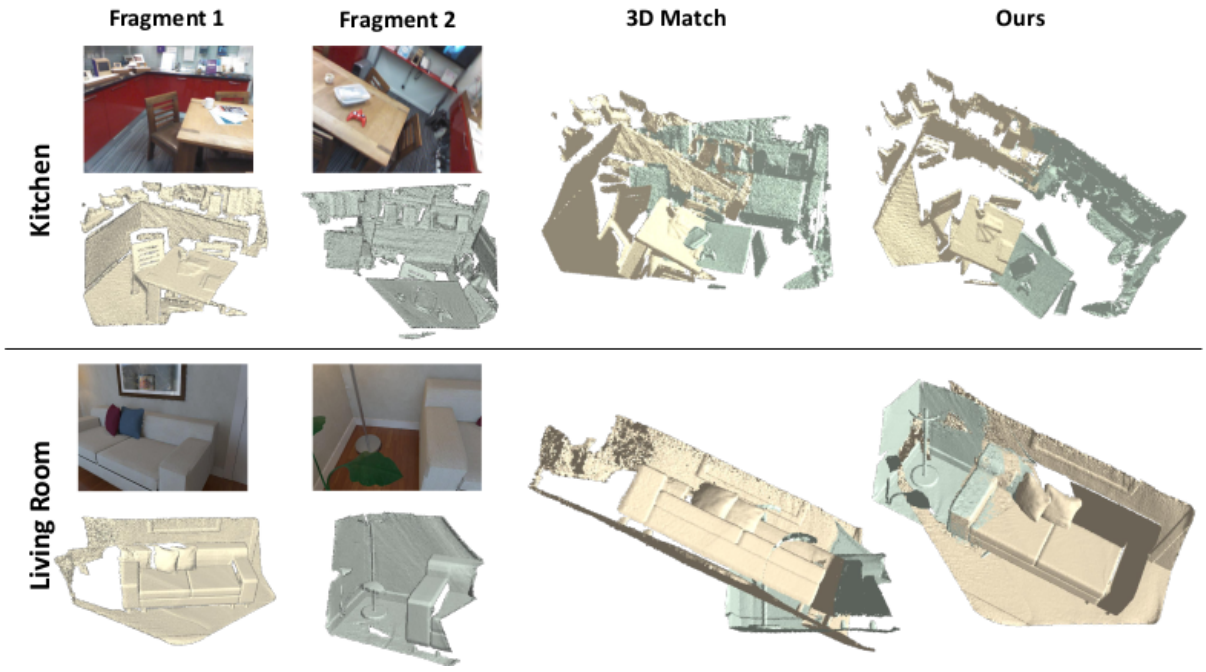


Figure 10: Examples of fragment alignment results using our learned 3D local descriptor. For comparison, we also provide the alignment results using the descriptors generated from the state-of-the-art 3DMatch.

Research Project: Regional geometric feature learning: RGB Image and Distance

Description

With the advances in the field of computer vision, visual environment perception, which includes object classification, detection, segmentation and distance estimation, has become a key component in the development of autonomous driving cars. However, object-specific distance estimation has attracted little attention from the computer vision community. In this work, we focus on addressing the interesting but challenging object-specific distance estimation problem for autonomous driving. Ours is the first work to develop an end-to-end learning-based approach that directly predicts distances for given objects in the RGB images. To facilitate the training and evaluation on this task, we construct the extended KITTI and nuScenes (mini and full) object-specific distance datasets. The experiment results demonstrate that our proposed method achieves superior performance over alternative approaches with less inference time.

Method

We build a Vanilla Model that extracts features from RGB images, then utilizes ROI pooling to generate a fixed-size feature vector for each object, and finally feeds the ROI features into a distance regressor to predict a distance for each object. Though our Vanilla-Model is able to provide promising prediction, it still does not fulfill the precision requirement for autonomous driving. Therefore, we create a close-object- focused(COF)-Model for more precise distance estimation, particularly for objects close to the camera. Specially, in the COF-Model, we design a keypoint regressor to predict part of the 3D keypoint coordinates (X, Y). Together with the predicted distance (Z), it forms a complete 3D keypoint (X, Y, Z). Leveraging the camera projection matrix, we define a projection loss between the projected 3D point and the ground truth keypoint on image to enforce a correct prediction.

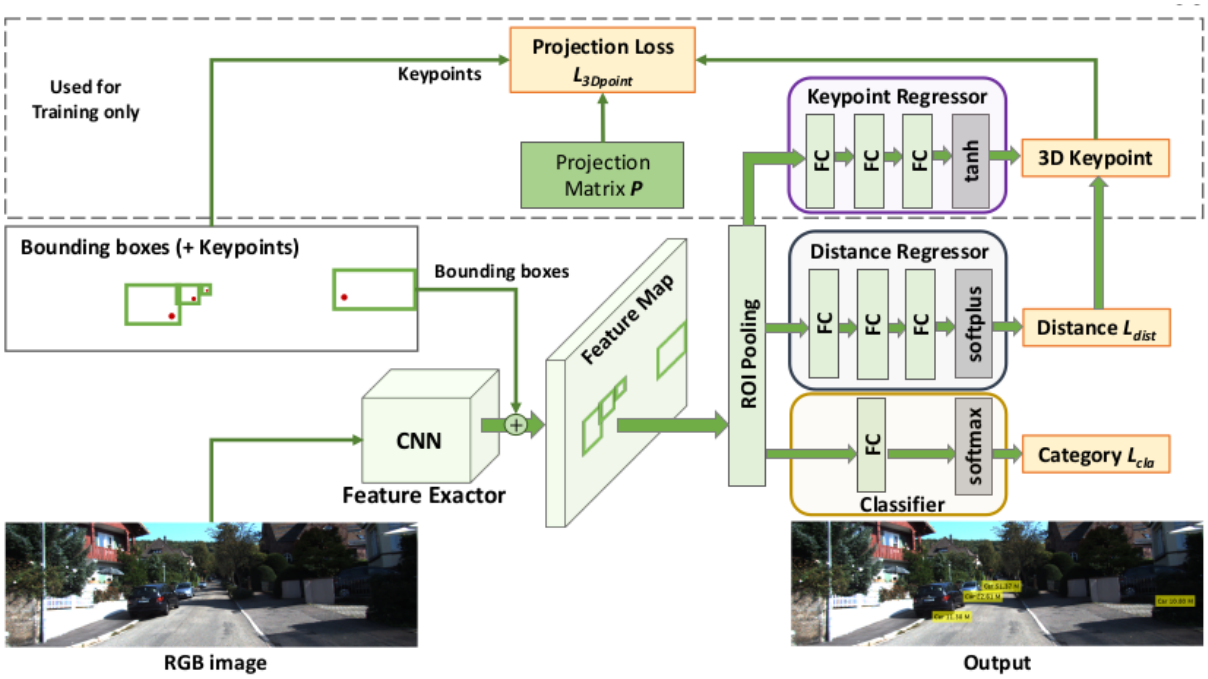


Figure 11: The framework of our close-object-focused (COF)-Model, which con- tains four parts, a feature extractor to generate a feature map for the whole RGB image, a keypoint regressor to predict a keypoint position on 3D coordinate, a distance regressor to directly predict a distance , and a multiclass classifier to predict the category label. The outputs of the keypoint regressor and distance regressor compose a 3D keypoint, which will be projected back to the image plane using the camera projection matrix. A projection loss is defined between the projected keypoint and the ground truth keypoint to enforce a better distance estimation.

Results

To validate our proposed methods, we construct an extended dataset based on the public available KITTI object detection dataset and the newly released nuScenes (mini and full) dataset by computing the distance for each object using its corresponding LiDAR point cloud and camera parameters. The experimental results on our constructed object- specific distance dataset demonstrate that our deep-learning-based models can successfully predict distances for given objects with superior performance over alternative approaches.

	Method	higher is better			lower is better			
		$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$	Abs Rel	Squa Rel	RMSE	RMSE_{log}
Car	SVR [69]	0.345	0.595	0.823	1.494	47.748	18.970	1.494
	IPM [208]	0.701	0.898	0.954	0.497	1290.509	237.618	0.451
	DORN (res101) [56]	0.647	0.824	0.929	0.248	1.711	6.614	0.348
	SAGN (res50) [229]	0.189	0.409	0.683	0.401	5.560	14.529	0.628
	Vanilla-Model (res50)	0.782	0.927	0.964	0.178	0.843	4.501	0.415
	Vanilla-Model (vgg16)	0.846	**0.947**	**0.981**	**0.150**	**0.618**	3.946	**0.204**
	COF-Model (res50)	0.796	0.924	0.958	0.188	0.843	4.134	0.256
	COF-Model (vgg16)	**0.848**	0.934	0.962	0.161	0.619	**3.580**	0.228
Pedestrian	SVR [69]	0.129	0.182	0.285	1.499	34.561	21.677	1.260
	IPM [208]	0.688	0.907	0.957	0.340	543.223	192.177	0.348
	DORN (res101) [56]	0.789	0.902	0.958	0.170	0.890	3.972	0.269
	SAGN (res50) [229]	0.265	0.548	0.773	0.345	2.827	8.268	0.565
	Vanilla-Model (res50)	0.649	0.896	0.966	0.247	1.315	4.166	0.335
	Vanilla-Model (vgg16)	0.578	0.861	0.960	0.289	1.517	4.724	0.312
	COF-Model (res50)	0.734	**0.963**	**0.988**	0.188	0.807	3.806	0.225
	COF-Model (vgg16)	**0.747**	0.958	0.987	**0.183**	**0.654**	**3.439**	**0.221**
Cyclist	SVR [69]	0.226	0.393	0.701	1.251	31.605	20.544	1.206
	IPM [208]	0.655	0.796	0.915	0.322	9.543	19.149	0.370
	DORN (res101) [56]	**0.810**	**0.962**	**0.984**	**0.149**	**0.704**	**4.735**	**0.200**
	SAGN (res50) [229]	0.297	0.614	0.842	0.319	4.653	14.275	0.506
	Vanilla-Model (res50)	0.744	0.938	0.976	0.196	1.097	4.997	0.309
	Vanilla-Model (vgg16)	0.740	0.942	0.979	0.193	0.912	4.515	0.240
	COF-Model (res50)	0.766	0.947	0.981	0.173	0.888	4.830	0.225
	COF-Model (vgg16)	0.768	0.947	0.974	0.188	0.929	4.891	0.233

Figure 12: The comparisons of object-specific distance estimation on the validation subset of constructed KITTI-based dataset.

	Method	higher is better			lower is better			
		$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$	AR	SR	RMSE	RMSE_{log}
Barrier	SVR [69]	0.007	0.008	0.009	9.998	196.476	204.734	2.365
	IPM [208]	0.670	0.901	0.965	0.252	30.362	36.813	0.302
	Vanilla-Model(res50)	0.572	0.851	0.952	0.283	2.378	7.492	0.327
	Vanilla-Model(vgg16)	0.525	0.795	0.898	0.401	3.893	8.153	0.428
	COF-Model(res50)	0.767	0.933	0.976	0.190	1.240	5.695	0.238
	COF-Model(vgg16)	**0.796**	**0.941**	**0.976**	**0.173**	**0.941**	**4.853**	**0.225**
Bicycle	SVR [69]	0.001	0.001	0.001	9.519	1842.093	202.612	2.321
	IPM [208]	0.485	0.732	0.869	0.443	22.644	27.742	0.445
	Vanilla-Model(res50)	0.404	0.699	0.897	0.416	6.360	12.491	0.405
	Vanilla-Model(vgg16)	0.457	0.777	0.916	0.392	4.354	9.517	0.388
	COF-Model(res50)	0.505	0.809	0.962	0.321	3.914	10.002	0.328
	COF-Model(vgg16)	**0.571**	**0.862**	**0.978**	**0.275**	**2.855**	**8.779**	**0.294**
Bus	SVR [69]	0.167	0.176	0.185	6.333	1476.812	152.789	2.090
	IPM [208]	0.489	0.723	0.847	0.738	915.818	241.888	0.586
	Vanilla-Model(res50)	0.556	0.874	0.965	0.225	2.823	12.650	0.306
	Vanilla-Model(vgg16)	0.550	0.831	0.935	0.308	3.510	11.572	0.363
	COF-Model(res50)	0.677	0.900	0.973	0.177	2.271	11.838	0.267
	COF-Model(vgg16)	**0.740**	**0.928**	**0.980**	**0.163**	**1.542**	**9.088**	**0.232**
Car	SVR [69]	0.060	0.061	0.063	6.740	1084.027	168.587	2.119
	IPM [208]	0.605	0.860	0.950	0.625	22526.891	1182.829	0.406
	Vanilla-Model(res50)	0.714	0.924	0.978	0.193	1.545	7.153	0.249
	Vanilla-Model(vgg16)	0.688	0.875	0.933	0.270	2.240	7.204	0.321
	COF-Model(res50)	0.835	**0.962**	**0.991**	0.137	0.837	5.564	0.187
	COF-Model(vgg16)	**0.851**	0.956	0.990	**0.137**	**0.753**	**5.010**	**0.186**

Figure 13: Comparison of object-specific distance estimation on the nuScenes-based full dataset.