On the Road to Large-Scale 3D Monocular Scene Reconstruction using Deep Implicit Functions

Thomas Roddick
University of Cambridge
tr346@cam.ac.uk

Benjamin Biggs
University of Cambridge
bjb56@cam.ac.uk

Daniel Olmeda Reino
Toyota Motor Europe
daniel.olmeda.reino@toyota-europe.com

Roberto Cipolla
University of Cambridge
rc10001@cam.ac.uk

Abstract

Autonomous driving relies on building detailed models of a vehicles surroundings, including all hazards, obstacles and other road users. At present, much of the autonomous driving literature reduces the world to a collection of parametric 3D boxes. While this framework is sufficient for many driving scenarios, other important scene details (e.g. overhanging structures, open car doors, debris, potholes etc.) are not modelled. Recently deep implicit functions have been shown to be suitable for representing fine grained details at arbitrarily high resolutions using images alone. However, they have predominantly been employed in constrained situations, such as reconstructing individual objects or small-scale scenes. In this work we explore the application of deep implicit functions to larger scenes in the context of real-world autonomous driving scenarios. In particular we focus on the challenging case where only monocular images are available at test time. While most implicit function networks rely on watertight meshes for training, these are not in general available for real world scenes. We therefore propose an alternative training scheme using LiDAR to provide approximate ground truth occupancy supervision. We also show that incorporating priors such as pre-detected object bounding boxes can improve the quality of reconstruction. Our method is evaluated on a real-world autonomous driving dataset.

1. Introduction

Understanding the 3D structure of a scene is an essential element of applications that interact with the real world, a prime example of which are autonomous vehicles. The ability of algorithms to capture complex detail within a scene determines the level to which the scene can be understood. Cuboidal bounding boxes are commonly used to capture the geometry of objects within a scene, such as cars and pedestrians. They encode the relative position, extent and pose of the object and enable applications such as tracking [13] and trajectory forecasting [24]. However, there are many instances where this coarse representation is insufficient. Consider for example a car with its door open; a truck with an overhanging load; or a pile of amorphous debris. Such variations are difficult to represent with a simple 3D bounding box, and yet to identify their extents accurately is critical for safely interacting with them. In this work, we take a first step towards building a more expressive representation by taking advantage of recent developments in 3D reconstruction using implicit function representations, which allows us to reconstruct the scene geometry at an arbitrary resolution, without the extreme memory constraints required by other dense reconstruction methods such as 3D occupancy grids [8]. Whilst previous works on implicit functions focus on reconstructing individual objects [20], people [31], or indoor scenes [33], we reconstruct entire large-scale traffic scenes, using only monocular camera images as input. Moreover, our method does not require 3D meshes nor synthetic data, and we propose a pipeline for generating ground truth from real sensors.

The uncertainty present in large-scale outdoor scenes often leads to scene reconstructions with blurred occupancy predictions. We propose to alleviate this uncertainty, for objects of known categories, by providing priors in the form of reference points. While cuboid bounding boxes are a minimal expression of an object geometry, we argue that they serve as priors to guide the implicit function towards a more accurate reconstruction. We study the impact of conditioning such representations with learnt shape priors, based on said 3D bounding boxes.

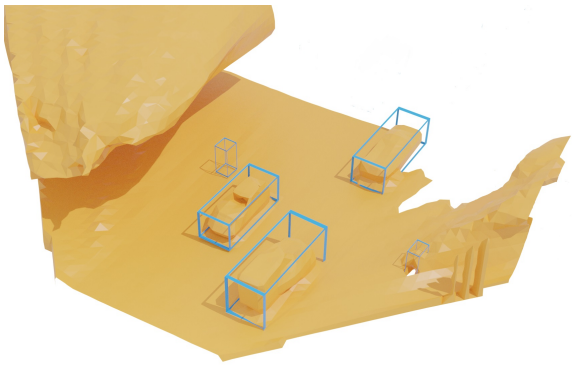

Figure 1. Overhead neural rendering of the geometry of a driving scene, produced with a single image from a front-facing camera of a vehicle. We condition the underlying implicit function representation with shape priors (3D cuboids) of objects of known categories in the scene.

Towards this goal, we present the following two contributions: 1) Works such as [20] and [31] rely on comprehensive, watertight 3D meshes as ground truth to train their implicit functions. For large scale 3D road scenes, no such data is available. However, we propose a pipeline for extracting the required training data automatically using LiDAR observations and coarse bounding boxes, which are easily obtainable and are freely provided by many recent autonomous driving datasets. We use this approach to build a large scale dataset based on the NuScenes dataset [2]. 2) 3D Shape Priors. Using the implicit occupancy function framework discussed above, we learn a generative shape model of common traffic agents such as cars and pedestrians. Given only the dimensions of a 3D bounding box as input, with no visual observations, the method is able to generate an approximate 3D mesh of a given object, even if that object has never been seen in the training data. This forms a vital element of our solution to the full scene reconstruction, allowing our implicit function to reconstruct parts of the scene which are completely hidden from view.

2. Related Work

3D reconstruction plays an important role in scene understanding and as such it has been extensively studied. Given the apparently continuous nature of 3D space, a recurrent question is what should be the underlying representation of 3D geometry. There have been multiple candidates for sparse and discrete quantization of the 3D space, including point [25, 26], mesh [10, 15] and voxel-based [5, 35, 27, 38] representations. Recent works propose using implicit functions [4, 20, 32] as a continuous representation of the 3D space.

Neural representations for scene reconstruction learn a mapping function from a 3D spatial location to a feature representation, which describes the geometry, and other properties [16], of a scene at that location. These representations are coupled with neural rendering engines which render and decode the features to form 2D images. This allows the supervision of the representation from images, and to generate novel views of the scene.

Occupancy Networks. A desirable property of a 3D geometric representation is the ability to sample at arbitrarily high resolutions and to be able to generalize to unseen points of view. In [20], occupancy networks are proposed as a way to implicitly represent 3D surfaces as the continuous decision boundary of a deep neural network classifier. They approximate an occupancy function, at every possible point in 3D, with a neural network that assigns to every location an occupancy probability. This representation, related to level set approaches, encodes the 3D geometry in a fixed memory, regardless of the sampling resolution. This approach is extended in [23] to convolutional occupancy networks. Saito et al. recover geometry and texture of humans at high resolution in [30, 31]. They define the surface as an implicit function, and align the pixel-level features to the 2D projection of the surface. This allows the learnt model to preserve the local details present in the image. This approach is extended in [12] by regularizing the reconstruction using the latent voxel feature representation and incorporating geometry-aligned shape features, as well as pixel-aligned features. In [36] the same principle is extended by combining global and local features.

NeRF [21] represents a 3D scene as a function of its coordinates and the radiance emitted from each point to the position of the camera. For each viewpoint, they march camera rays and accumulate colors and densities into an image, given the estimated radiance. Similarly, [17] learns the surface with differentiable ray-marching. However, in this case, the surface is reconstructed progressively, by having individual implicit fields for each voxel in an octree. In an iterative process, the voxels are pruned and adjusted to the underlying scene.

While there has been recent advances in reconstruction of highly-controlled, synthetic settings [37], fewer works tackle the more challenging setting of shape reconstruction in the wild. Single objects are reconstructed in [7], which builds on early work [22] on signed distance functions. It tests the reconstruction based on sparse LiDAR and optionally images from street scenes. However, differently from us, they exploit synthetic 3D meshes, and simulated LiDAR scans and images to build the representation. Our approach does not require synthetic data and is able to reconstruct entire outdoor scenes.

3D scenes. Neural implicit functions show expressive reconstructions of single objects with limited complexity and local shape variability. Recently, some works have shown significant advance in reconstruction of more complex indoor scenes[33]. In [14] it is suggested that the local geo-

metric forms of objects of different categories share similar features at a certain scale. They exploit this observation to reconstruct indoor scenes, by aggregating parts of decomposed objects and learning their common shape features.

3D object detection. In this work we study how object shape priors influence the reconstruction of an otherwise unconstrained scene. A common representation of 3D objects are 3D bounding boxes, which minimally encode position, dimensions and pose. We use as priors similar output as that of any of the recent examples of 3D object detection from monocular images [18, 1, 32, 29].

3. Method

3.1. Background: Deep Implicit Functions

The objective of our work is to recover a full 3D reconstruction of a large-scale outdoor scene using a single monocular image as input. We begin by representing the 3D structure of the scene in the form of an occupancy function f_θ, which maps a 3D point $\mathbf{x} \in \mathbb{R}^3$ to the probability that the point lies inside the solid surface of the scene

$$p(\hat{o}) = f_\theta(\mathbf{x}, \mathbf{z}) \in [0, 1] \tag{1}$$

where $\mathbf{z} \in \mathbb{R}^n$ is a conditioning feature vector corresponding to the point $\mathbf{x}$ and $\hat{o}$ is the occupancy:

$$\hat{o} = \begin{cases} 1 & \text{if x is inside the surface} \\ 0 & \text{otherwise} \end{cases}$$

In this work, the occupancy function f_θ takes the form of a neural network which is trained to minimise the binary cross entropy loss between the predicted and ground truth occupancy over a set of randomly sampled points $\{\mathbf{x}_i\}$:

$$\mathcal{L}(\hat{o}, o) = \sum_i o_i \log(f_\theta(\mathbf{x}_i, \mathbf{z_i}) + (1 - o_i) \log(1 - f_\theta(\mathbf{x}_i, \mathbf{z_i})). \tag{2}$$

At inference time the 3D surface of the scene can then be extracted by computing the level set corresponding to $f_\theta(\mathbf{x}, \mathbf{z}) = 0.5$.

3.2. Pseudo-ground truth occupancy from LiDAR

In order to train an implicit occupancy function $f_\theta(\mathbf{x}, \mathbf{z})$ as described in Section 3.1, it is necessary to sample arbitrary 3D points in the scene $\mathbf{x}$ and determine their ground truth occupancy o. For real-world outdoor scenes, watertight 3D meshes which can be used to query the ground truth occupancy state of the world, are not generally available. However, observations from a range sensor such as a LiDAR system, which are widely available in datasets designed for autonomous driving [3, 2], provide a strong prior estimate of the occupancy state of points which lie along the returned rays.

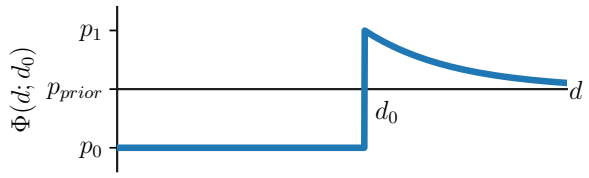

Figure 2. The inverse sensor model, which estimates the probability that a point $\mathbf{x}$ a distance d along a LiDAR ray is occupied, given the LiDAR return distance d_0.

Let us begin by considering a LiDAR ray r, which terminates at a solid surface a distance d_0 along the ray. For a point $\mathbf{x}$ which lies at a distance d along the ray, we can infer that if $d < d_0$, it is highly likely that the point $\mathbf{x}$ represents free space, since the lidar ray was unobstructed by any surface closer than d_0. Similarly, for points where $d > d_0$, there is a high likelihood that the point lies inside an object since we know that d_0 lies on the surface. We can encode this prior knowledge in the form of an *inverse sensor model* [34], which approximates the probability of occupancy $p(o) = \Phi(d; d_0)$ for a point a distance d along the ray, given an observed LiDAR return at distance d_0. For this work we adopt a simple heuristic inverse sensor model of the form

$$\Phi(d; d_0) = \begin{cases} p_0 & \text{if } d < d_0 \\ p_{prior} + (p_1 - p_{prior})\, e^{-\alpha(d - d_0)} & \text{otherwise} \end{cases} \tag{3}$$

In practice, real LiDAR point clouds exhibit noise due to reflection and other effects, so the terms $p_0 = p(o|d < d_0)$ and $p_1 = p(o|d = d_0)$ allow us to introduce some uncertainty into the estimated occupancy values. Meanwhile, the exponential decay factor $e^{-\alpha(d-d_0)}$ encodes the fact that for points which lie just beyond the surface, we can be reasonably confident that the point is inside the object. For points where $d >> d_0$, however, it is unclear whether the point is still inside the object or whether it has emerged from the occluded back face, so we set $p(o)$ to the prior probability of occupancy p_{prior}. Using this heuristic, we are able to train an implicit occupancy network as described in Section 3.1 by replacing the ground truth occupancy state o from (2) with the approximate occupancy probability $p(o)$.

3.3. LiDAR augmentation

The aim of this work is to reconstruct the full 3D geometry of the scene, including surfaces which are occluded or facing away from the camera. However, a limitation of the above approach is that typically the LiDAR point cloud is captured from the same perspective as the camera system, meaning that pseudo-ground truth can only be obtained for points which are visible to the camera. Furthermore,

LiDAR point clouds are typically sparse, with distant objects returning only a handful of points. Fortunately, many autonomous driving datasets consist of sequences captured from a moving vehicle, meaning that we can obtain multiple views of the same scene by combining LiDAR point clouds across time. Consider a trajectory T which consists of a sequence of ego-vehicle poses $T = \{P_t\}, P \in SO(3)$ at times t. We can map each LiDAR ray r_{it} captured at time t into the target frame at time t_0 by applying the rigid-body transformation $P_{t_0}^{-1} P_t$ to the start and end points of the ray. Aggregating across the entire sequence results in a dense set of rays which capture the surfaces of the scene from multiple perspectives.

Dynamic objects The above approach assumes the scene is static: a rarely satisfied assumption for traffic scenes. However, ground truth bounding boxes b_{nt} for most dynamic objects in the scene are available in the same autonomous driving datasets, in the form of the object pose $Q_{nt} \in SO(3)$ and dimensions $\mathbf{d}_n \in \mathbb{R}^3$ for object n at time t. We map each LiDAR ray r_{it} that intersects the bounding box b_{nt} to the target frame by applying the rigid-body transformation $P_{t_0}^{-1} Q_{nt_0} Q_{nt}^{-1} P_t$.

Symmetry constraints Despite aggregating rays across multiple frames as described above, many parts of the scene will still only be visible from one side. For objects such as vehicles with 3D bounding boxes provided, we can further densify the rays by taking advantage of simple bilateral symmetry constraints. For each ray which intersects a 3D bounding box, we append a duplicate ray which is a mirror image about the vertical plane.

3.4. Sampling strategies

A key element in the success of occupancy networks is the strategy used to sample points at training time; methods balance uniformly spanning the volume of interest against choosing points which provide useful training signal. In our setting the problem is particularly challenging since the ground truth occupancy is only defined for points which lie along LiDAR rays. We approach the problem by applying a weighted combination of the following strategies:

Surface sampling In order to maximise the discriminitive power of the network, the most useful training signal comes from points which lie close to the surface of objects. Although we do not know the true surface of the scene, the endpoints of the LiDAR rays r provide a sparse approximation. Given a ray r with start and end points $(\mathbf{s}, \mathbf{e})$, we sample points $\mathbf{x}$ by applying random Gaussian noise to the end point along the direction of the ray

$$\mathbf{x} = \mathbf{e} + \epsilon \frac{\mathbf{e} - \mathbf{s}}{|\mathbf{e} - \mathbf{s}|}, \quad \epsilon \sim N(0, \sigma) \tag{4}$$

Uniform sampling Sampling points exclusively around surfaces causes the method to overfit and fail in regions of large open space, which for road scenarios form the majority of the scene. Unfortunately, we cannot apply the uniform sampling strategy used by other works since in our case the occupancy is only defined along the LiDAR rays r. We instead obtain an approximation to uniform sampling as follows. We begin by clipping each ray to the visible volume of the scene, resulting in clipped rays $\bar{r} = (\bar{\mathbf{s}}, \bar{\mathbf{e}})$. For rays which intersect objects, we instead clip to the object bounding box b. We then randomly sample N points along each clipped ray, where N is proportional to the length $|\bar{\mathbf{e}} - \bar{\mathbf{s}}|$. The sampled points $\mathbf{x}$ are taken uniformly between the start and end point of the clipped ray:

$$\mathbf{x} = (1 - \eta)\bar{\mathbf{s}} + \eta \bar{\mathbf{e}}, \quad \eta \sim U(0, 1) \tag{5}$$

Sparse sampling For some parts of the scene, the number of LiDAR rays passing through a given region can be relatively small, for example regions in the sky where many of the returns are invalid. To encourage low occupancy probability in these regions, we first divide the scene into coarse voxels and compute the number of rays passing through each voxel. For voxels where the number of rays is below a certain threshold, we uniformly sample points over the voxel, and set the ground truth occupancy to a low value. This helps avoid 'cave' artifacts which can occur due to lack of supervision in the sky region.

Object-centric sampling In traffic scenes, much of the structure of the road, buildings etc. is relatively uniform, and provides little useful training information to our network. To encourage the network to focus more on objects of interest such as cars, pedestrians etc., we apply a simple inverse frequency weighting to the surface and uniform strategies described above, such that rays which intersect objects with fewer LiDAR returns are sampled more frequently. The weighting factor w_n for object n is given by $(N_n / N_T)^{-\gamma}$, where N_n is the number of rays intersecting object n, N_T is the total number of rays, and γ is a hyperparameter.

3.5. Pixel-aligned implicit function network

As described in Section 3.1, an occupancy network takes as input a 3D point in space $\mathbf{x}$ and a conditioning vector $\mathbf{z}$, and predicts the probability of occupancy $p(\hat{o})$ at that location. In order to reconstruct the fine-grained details of the scene, we adopt the pixel-aligned implicit function approach of Saito et al. [30] to obtain the conditioning vector $\mathbf{z}$ from a single-view input image $\mathbf{I}$. We begin by transforming the image into a dense feature representation by passing the image through a convolutional image encoder g resulting in a set of spatial feature maps $F = g(\mathbf{I})$. For each 3D

query point $\mathbf{x}$ we then obtain the conditioning vector $\mathbf{z}$ by sampling the feature maps at the location corresponding to the projection of $\mathbf{x}$ into the image:

$$\mathbf{z} = F(\pi(K\mathbf{x})), \tag{6}$$

where π is the 2D perspective projection operator and K is the 3×3 camera intrinsics matrix, which is assumed to be known at test time. The probability occupancy for the point $\mathbf{x}$ is then obtained by passing $\mathbf{x}$ and $\mathbf{z}$ through our fully-connected occupancy network f_θ. At inference time, we can reconstruct an entire scene by sampling the points $\mathbf{x}$ over a dense grid in space and applying the marching cubes algorithm [19] to generate the final output mesh.

3.6. Bounding box conditioning

One of the major challenges of reconstructing large-scale outdoor scenes from a single view is that there is a high degree of uncertainty in estimating the depth of surfaces in the scene. This uncertainty can manifest as low-quality reconstructions or missed objects where the network blurs the predicted occupancy probability over multiple possible depth values. We can alleviate some of this uncertainty and force the network to commit to a particular set of depth values by providing a set of known reference points within the scene. Recent work has demonstrated that it is possible to obtain accurate 3D bounding boxes of objects of interest such as cars and pedestrians from a single monocular image [18, 32, 1, 29]. We propose to leverage these predicted bounding boxes to provide fixed anchor points within the scene, allowing the network to resolve some of the depth uncertainty and exploit detailed prior knowledge of the shapes of common objects such as cars and pedestrians.

Let us consider a point $\mathbf{x}$ which lies within a predicted bounding box b_n. In addition to the image feature conditioning vector $\mathbf{z}$, which we rename $\mathbf{z}_{image}$, we additionally provide the network with a set of features $\mathbf{z}_{box}$ derived from the bounding box. Specifically, $\mathbf{z}_{box}$ consists of:

$\mathbf{z}_{local} \in \mathbb{R}^3$: The offset of the point $\mathbf{x}$ in the box's local coordinate system, obtained by multiplying $\mathbf{x}$ by the inverse of the object pose Q_n^{-1}.

$\mathbf{z}_{dim} \in \mathbb{R}^3$: The dimensions of the bounding box $\mathbf{d}_n$.

$\mathbf{z}_{class} \in \{0,1\}^{|C|}$: A one-hot encoding of the object category $c_n \in C$, e.g. car, pedestrian etc.

The final conditioning vector $\mathbf{z}$ is then simply the concatenation of the individual features $\mathbf{z} = (\mathbf{z}_{image}, \mathbf{z}_{local}, \mathbf{z}_{dim}, \mathbf{z}_{class})$. For points which do not lie inside any predicted bounding boxes we set $\mathbf{z}_{local}$ and $\mathbf{z}_{dim}$ to zero and $\mathbf{z}_{class}$ to the label corresponding to the background class.

4. Experiments

4.1. Dataset

We benchmark our approach using the NuScenes dataset [2]. NuScenes is a large-scale autonomous driving dataset consisting of 850 sequences captured over four international locations. Crucially, it provides multimodal data including six surround view camera images, LiDAR point-clouds, and densely-annotated ground bounding boxes which allow us to apply the LiDAR augmentation strategy described in Section 3.3. We focus our evaluation on the front-facing camera images only, and adopt the dataset splits of Roddick et al. [28] to avoid overfitting to the geometry of scenes which appear in both the default training and validation sets.

4.2. Implementation details

The architecture for our pixel-aligned implicit function network is inspired by the architecture of Saito et al. [30]. We use the same implicit function network which consists of a five-stage multi-layer perceptron with skip connections to the input features $\mathbf{z}$ before the 2nd, 3rd and 4th layers. Differently from Saito et al., we found that in our setting an image encoder based on a standard ResNet-50 network [11], pre-trained on ImageNet [6], was most effective. To obtain the image conditioning vector $\mathbf{z}_{image}$, we sample features from the 3rd, 4th and 5th stages of the image encoder network using the camera parameters K provided by the NuScenes dataset. To condition our method on 3D object bounding boxes, we apply the state-of-the-art monocular 3D object detector of Liu et al. [18], which is pretrained on the KITTI dataset [9] and then finetuned on NuScenes.

We train our method using a balanced version of the cross-entropy loss in (2) where the loss corresponding to positive ($p(o) > .5$) and $p(o) < .5$ negative samples are weighted according to $w_{pos} = N_T/N_{pos}$ and we $w_{neg} = N_T/N_{neg}$ respectively, where N_{pos} and N_{neg} are the number of positive and negative samples. For each training image we sample 10000 points, according to the surface, uniform and sparse sampling strategies described in Section 3.4, in the ratio $45 : 45 : 10$. We also apply object-centric sampling to the surface and uniform samplers with an exponent of $\gamma = 0.1$. For the inverse sensor model discussed in Section 3.2, we set $p_0 = 0$, $p_1 = 1$ and $p_{prior} = 0.5$ for simplicity, and select $\alpha = 0.01$ for the decay factor. We train the model using stochastic gradient descent with a learning rate of 0.1, momentum factor 0.9 and weight decay 10^{-4}.

4.3. Metrics

Our primary evaluation metric is the Chamfer-L1 distance as described by Mescheder et al. [20]. The Chamfer-L1 distance is the mean of an accuracy term, which mea-

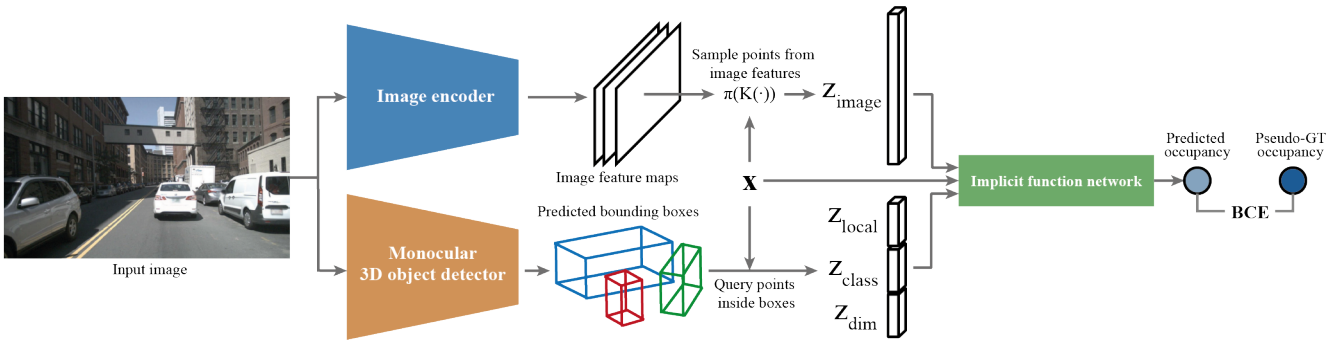

Figure 3. An overview of our deep implicit function-based approach. Our method accepts a single monocular image as input, and predicts the probability of occupancy $p(\hat{o})$ for 3D points in space $\mathbf{x}$. The system comprises three main components: (1) a convolutional image encoder which is sampled to provide image-based features $\mathbf{z}_{image}$. (2) a monocular 3D object detector which predicts bounding boxes for each object in the scene, providing bounding box features $\mathbf{z}_{local}, \mathbf{z}_{class}, \mathbf{z}_{dim}$, and (3) an implicit function network f_θ which predicts occupancy probability $p(\hat{o})$ given the 3D point $\mathbf{x}$ and conditioning vector $\mathbf{z}$.

sures the average distance from the predicted surface to the ground truth, and a completeness term, which measures the converse. In our setting, the true ground truth surface is unknown, so we approximate it using the aggregated LiDAR point clouds described in Section 3.3. We uniformly sample a set of N points from the surface of our predicted mesh, and compare them against a subset of N points sampled from the ground truth LiDAR point clouds.

Whilst this metric gives an overall indication of the quality of the scene reconstruction, for traffic scenes we observe that the vast majority of points on the mesh represent relatively uninteresting surfaces such as the road or walls. In order to capture the more relevant features of the scene, we additionally compute the above metrics at the individual object level. We achieve this by cropping the predicted mesh to a region equal to $1.5\times$ the dimensions of the bounding box for each object in the scene. We then provide the average metric computed over each object category (car, truck, pedestrian etc.) in the NuScenes dataset. We use a surface sampling rate of $N = 10000$ for evaluating the scene meshes and $N = 1000$ for each object.

4.4. Baselines

As the first work to tackle the problem of large-scale scene reconstruction on the NuScenes dataset, we consider two baseline methods to benchmark our performance:

Bounding box mesh In the absence of detailed information about the 3D shape of objects in the scene, a simple approximation to the scene reconstruction is to simply represent each object as its 3D bounding box. To provide a basic sanity-check, we apply the monocular object detector of Liu et al. [18] to each input image, and convert the resulting bounding boxes to a triangle mesh representation. To represent the ground surface, we place a simple plane at the minimum base height of all objects in the scene, or at a fixed height below the camera if no objects are present.

Pixel-aligned Implicit Function The basis for our approach is the Pixel-aligned Implicit Function (PIFu) approach of Saito et al. [30], which we provide as a baseline for our method. Since the LiDAR rays used to generate our ground truth training data do not provide explicit colour information, we train only the surface reconstruction subnetwork. In order to offer a fair comparison to our approach, we also adopt the same ResNet-50 image encoder as used in our approach.

4.5. Qualitative results

We begin by presenting qualitative examples of the reconstructions produced by our method evaluated on the NuScenes validation set. Figure 4.3 shows the 3D scene viewed from two views: the camera view and an overhead viewpoint. We annotate the reconstructions with the 3D bounding boxes predicted by our front-end object detector, which are used to condition the implicit function network. The final column of Figure 4.3 compares the reconstructed mesh against the ground truth LiDAR points used for evaluation. From these results it can be seen that our method is able to model complex scene geometry, accurately capturing the shapes of objects such as cars and correctly localise the boundaries of the road surface. It can additionally handle challenging edge-cases such as cluttered scenes (row 1), adverse weather conditions (row 3) and objects which are distant from the camera (row 4).

4.6. Comparison to baselines

We evaluate our box-conditioned implicit function approach alongside the two baseline methods on the NuScenes validation set. Table 1 presents the results of this evaluation according to the metrics discussed in Section 4.3. We also provide a qualitative comparison of the three methods in

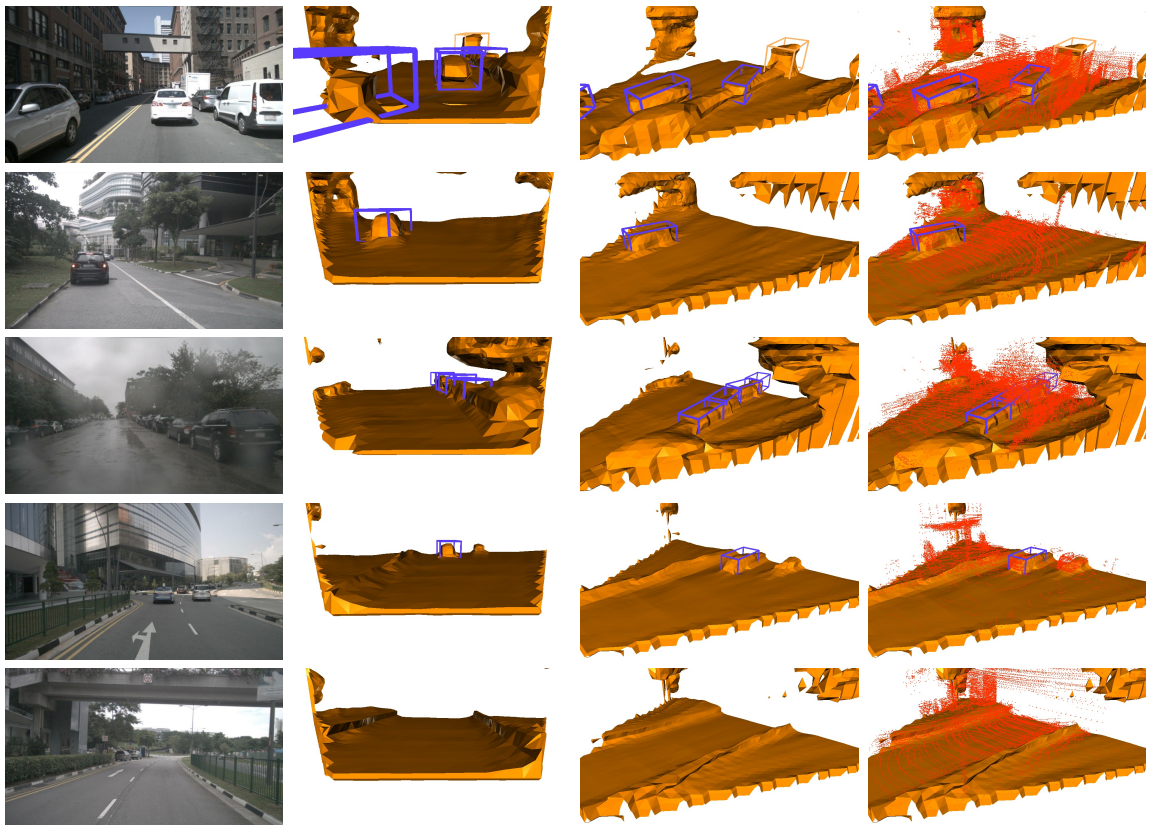

Image Front view Overhead view Ground truth LiDAR

Figure 4. Qualitative examples of our method on the NuScenes validation set. We show reconstructions produced by our method from the camera perspective and alternative side view. Predicted object bounding boxes, which are generated as part of our method, are shown in blue. We also show the set of densified LiDAR points in red, which provide the ground truth reference points used for evaluation.

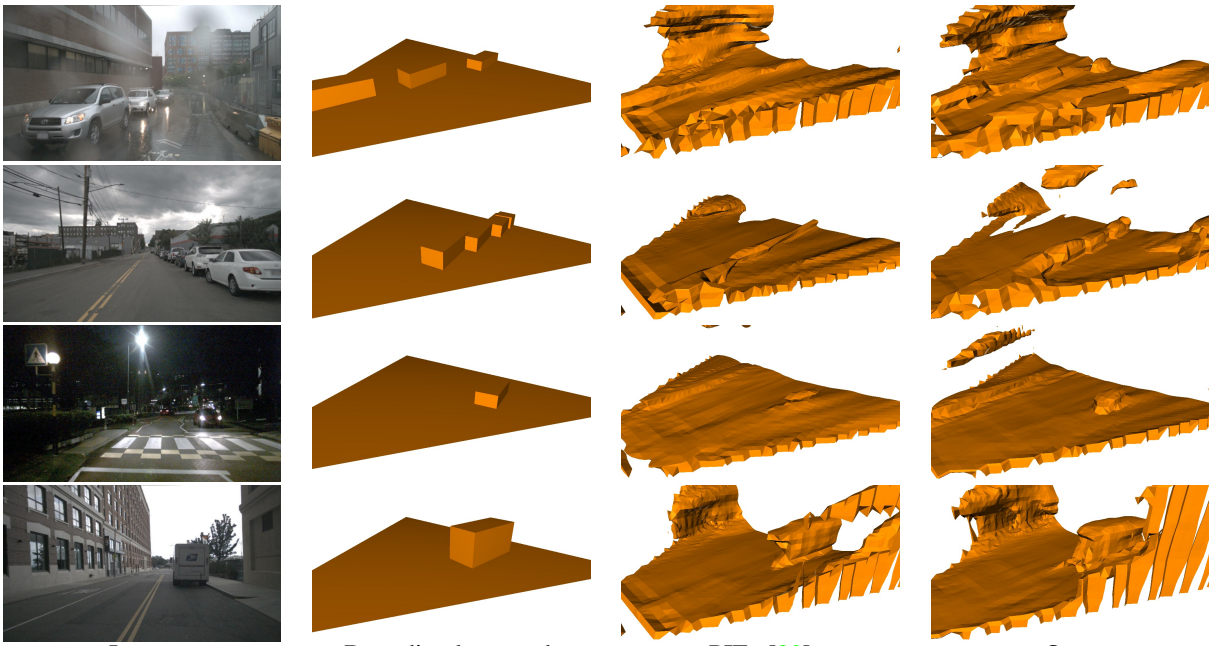

Image Bounding box mesh PIFu [30] Ours

Figure 5. Qualitative comparison of ours and baseline approaches on the NuScenes validation set.

Table 1. **Baseline comparisons** on the NuScenes validation set. We evaluate each metric both at the scene level, and averaged across each NuScenes object category. ↑/↓ indicates metrics where higher/lower scores are better..

Metric	Chamfer-L1 (m) ↓		Completeness (m) ↓		Accuracy (m) ↓		F1Score (%) ↑	
	Object	Scene	Object	Scene	Object	Scene	Object	Scene
Bounding box mesh	0.554	0.866	0.737	1.022	0.372	**0.710**	23.1	13.5
PIFu [30]	0.528	**0.680**	0.711	0.564	0.345	0.795	24.5	30.3
Ours (GT boxes)	0.389	0.753	0.518	0.659	0.260	0.847	41.6	26.9
Ours	**0.483**	0.710	**0.625**	**0.544**	**0.341**	0.876	**30.4**	**30.3**

Table 2. **Ablation study.** We investigate the impact of the four components of the conditioning vector $\mathbf{z}$ on the final reconstruction performace. ↑/↓ indicates metrics where higher/lower scores are better.

Conditioning vector $\mathbf{z}$				Chamfer-L1 (m) ↓		Completeness (m) ↓		Accuracy (m) ↓		F1Score (%) ↑	
Image	Local	Class	Dim	Object	Scene	Object	Scene	Object	Scene	Object	Scene
✓	✓	✓	✓	**0.483**	**0.710**	**0.625**	0.544	**0.341**	**0.876**	30.4	**30.3**
✓	✓	✓		0.506	0.766	0.661	**0.518**	0.351	1.015	29.2	25.3
✓	✓			0.500	0.828	0.644	0.530	0.355	1.126	**30.6**	25.2
	✓	✓	✓	0.579	1.219	0.766	0.722	0.392	1.716	22.1	15.4

Figure 4.3. Across the majority of the metrics our method outperforms both baseline methods, often by a considerable margin. Notably, our method performs particularly well when evaluated at the object level. The results in Figure 4.3 demonstrate that our method takes advantage of the explicit knowledge of predicted object locations, and thus produces results of significantly higher sharpness than the competitors. This observation emphasises the value of the bounding box feature conditioning described in Section 3.6. At the scene level, our method produces a slightly worse Chamfer-L1 score than PIFu, largely on account of a higher accuracy error, indicating that the method has a tendancy of over-estimating the geometry of background elements of the scene such as trees and buildings. However this is partially compensated for by the fact that the outperforms both baselines on the scene completeness metric, indicating that fewer elements of the background are missed.

To further understand the behaviour of our method, we seek to disentangle the bounding box conditioning performance from the accuracy of the underlying monocular 3D object detector by providing ground truth boxes at test time, which we present as an additional entry in Table 1. This analysis shows providing more accurate boxes at inference time dramatically improves the performance of the metric. Since our method is agnostic to the choice of front-end object detection architecture, these results provide optimism that future developments in monocular object detection will further improve the efficacy of our approach.

4.7. Ablation study

A key novelty of our approach is the addition of the bounding box conditioning vector $\mathbf{z}_{box}$ to the pixel-aligned feature encoding of Saito et al. [30] as discussed in Section 3.6. To understand the significance of each element of $\mathbf{z}_{box}$, we conduct an ablation experiment in which we systematically remove each component and evaluate the effect quantitatively. The results of this experiment are shown in Table 2.

We found that the component of the box encoding $\mathbf{z}_{box}$ most critical for performance was the encoding of the object dimensions $\mathbf{z}_{dim}$. Ablating this component resulted in a significant increase in error across most metrics. By contrast, further ablating the class encoding $\mathbf{z}_{class}$ resulted in a minor improvement in error, which we suspect was due to overfitting to the class label.

As a final ablation experiment, we additionally ablate the pixel-aligned image-features $\mathbf{z}_{image}$. As expected, doing so caused the reconstruction over the background regions of the scene to fail completely, since the network cannot exploit any local information in these regions. For the areas corresponding to objects however, this variant was still able to achieve surprisingly good reconstructions despite never having seen the object in an image. This suggests that the bounding boxes alone provide strong prior shape information for common objects in the scene.

5. Conclusions

We have presented a method for reconstructing large-scale traffic scenes. We draw inspiration from recent advances in implicit function representations, and propose to condition the reconstruction of known object categories on reference points in the form of cuboid bounding boxes. We conduct our experiments on a large autonomous driving dataset, and provide strong baselines, together with a pipeline for automatic generation of ground truth from real sensors. We have shown that our method, together with our sampling strategy and sensor modelling, is able to better approximate the shape of the scene, particularly in the context of autonomous driving.

References

[1] Garrick Brazil and Xiaoming Liu. M3d-rpn: Monocular 3d region proposal network for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9287–9296, 2019. 3, 5

[2] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 2, 3, 5

[3] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, et al. Argoverse: 3d tracking and forecasting with rich maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8748–8757, 2019. 3

[4] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5939–5948, 2019. 2

[5] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European conference on computer vision*, pages 628–644. Springer, 2016. 2

[6] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 5

[7] Shivam Duggal, Zihao Wang, Wei-Chiu Ma, Sivabalan Manivasagam, Justin Liang, Shenlong Wang, and Raquel Urtasun. Secrets of 3d implicit object shape reconstruction in the wild. *arXiv preprint arXiv:2101.06860*, 2021. 2

[8] Alberto Elfes et al. Occupancy grids: A stochastic spatial representation for active robot perception. In *Proceedings of the Sixth Conference on Uncertainty in AI*, volume 2929, page 6. Morgan Kaufmann, 1990. 1

[9] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012. 5

[10] Georgia Gkioxari, Jitendra Malik, and Justin Johnson. Mesh r-cnn. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9785–9795, 2019. 2

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[12] Tong He, John Collomosse, Hailin Jin, and Stefano Soatto. Geo-pifu: Geometry and pixel aligned implicit functions for single-view human reconstruction. *arXiv preprint arXiv:2006.08072*, 2020. 2

[13] Hou-Ning Hu, Qi-Zhi Cai, Dequan Wang, Ji Lin, Min Sun, Philipp Krahenbuhl, Trevor Darrell, and Fisher Yu. Joint monocular 3d vehicle detection and tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5390–5399, 2019. 1

[14] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, Thomas Funkhouser, et al. Local implicit grid representations for 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6001–6010, 2020. 2

[15] Angjoo Kanazawa, Shubham Tulsiani, Alexei A Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 371–386, 2018. 2

[16] Amit Pal Singh Kohli, Vincent Sitzmann, and Gordon Wetzstein. Semantic implicit neural scene representations with semi-supervised training. In *2020 International Conference on 3D Vision (3DV)*, pages 423–433. IEEE, 2020. 2

[17] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields, 2021. 2

[18] Yuxuan Liu, Yuan Yixuan, and Ming Liu. Ground-aware monocular 3d object detection for autonomous driving. *IEEE Robotics and Automation Letters*, 6(2):919–926, 2021. 3, 5, 6

[19] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 21(4):163–169, 1987. 5

[20] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019. 1, 2, 5

[21] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, pages 405–421. Springer, 2020. 2

[22] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019. 2

[23] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. *arXiv preprint arXiv:2003.04618*, 2, 2020. 2

[24] Tung Phan-Minh, Elena Corina Grigore, Freddy A Boulton, Oscar Beijbom, and Eric M Wolff. Covernet: Multimodal behavior prediction using trajectory sets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14074–14083, 2020. 1

[25] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 2

[26] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017. 2

[27] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions.

In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3577–3586, 2017. 2

[28] Thomas Roddick and Roberto Cipolla. Predicting semantic map representations from images using pyramid occupancy networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 5

[29] Thomas Roddick, Alex Kendall, and Roberto Cipolla. Orthographic feature transform for monocular 3d object detection. *British Machine Vision Conference*, 2019. 3, 5

[30] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2304–2314, 2019. 2, 4, 5, 6, 7, 8

[31] Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In *CVPR*, 2020. 1, 2

[32] Andrea Simonelli, Samuel Rota Bulo, Lorenzo Porzi, Manuel López-Antequera, and Peter Kontschieder. Disentangling monocular 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1991–1999, 2019. 2, 3, 5

[33] Vincent Sitzmann, Julien NP Martel, Alexander W Bergman, David B Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *arXiv preprint arXiv:2006.09661*, 2020. 1, 2

[34] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. 2006. 3

[35] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015. 2

[36] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. *arXiv preprint arXiv:1905.10711*, 2019. 2

[37] Xiaochen Zhao, Zerong Zheng, Chaonan Ji, Zhenyi Liu, Yirui Luo, Tao Yu, Jinli Suo, Qionghai Dai, and Yebin Liu. Vehicle reconstruction and texture estimation using deep implicit semantic template mapping. *arXiv preprint arXiv:2011.14642*, 2020. 2

[38] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2018. 2