Inverse Neural Rendering for Explainable Multi-Object Tracking

Julian Ost* Tanushree Banerjee* Yuval Bahat Mario Bijelic Felix Heide

Princeton University

Abstract

Today's most successful methods for image understanding tasks rely on feed-forward neural networks. While this approach has allowed for empirical accuracy, efficiency, and task adaptation via fine-tuning, it comes with fundamental disadvantages. Existing networks often struggle to generalize across different datasets, even on the same task. Moreover, these networks ultimately reason about high-dimensional scene features, which are challenging to analyze. This is true especially when attempting to predict 3D information based on 2D images. We propose to recast 3D multi-object tracking from RGB cameras as an Inverse Rendering (IR) *problem, by optimizing through a differentiable rendering pipeline over the latent space of pre-trained 3D object representations that best represent object instances in a given input image. To this end, we optimize an image loss over generative latent spaces that inherently disentangle shape and appearance properties. Our method is not only a new take on tracking, but also enables examining the reconstructed objects, reasoning about failure situations, and resolving ambiguous cases. We validate the generalization capabilities of our method by training on synthetic data only and assessing camera-based 3D tracking on the nuScenes and Waymo datasets. Both these datasets are completely unseen to our method and do not require fine-tuning.*

1. Introduction

Today's most successful image understanding methods employ feed-forward neural networks for performing vision tasks, including segmentation [11, 37, 41], object detection [17, 33, 39, 57–59, 91], object tracking [9, 30, 53, 61, 72, 85, 90] and pose estimation [69, 80]. Typically, these approaches learn network weights using large labeled datasets. At inference time, the trained network layers sequentially process a given 2D image to make a prediction. Despite being a successful approach across disciplines, from robotics to health, and effective in operating at real-time rates, this approach also comes with several limitations: (i) Networks trained on data captured with a specific camera/geography *generalize poorly*, (ii) these networks typically rely on high-dimensional internal feature representations which are *often not interpretable*, making it hard to identify and reason about failure cases, and, (iii) it is challenging to explicitly enforce 3D geometrical constraints, consistency, and priors in the predictions.

We focus on multi-object tracking as a task that must tackle all these challenges. Accurate multi-object tracking is essential for safe robotic planning. While approaches using LiDAR point clouds (and camera image input) are successful as a result of the explicitly measured depth [13, 30, 40, 53, 73, 81, 85], camera-based approaches to 3D multi-object tracking have only been studied recently [9, 18, 23, 44, 47, 54, 70, 76, 82, 90]. Monocular tracking methods, typically consisting of independent detection, 3D dynamic model, and matching modules, often struggle as the errors in the distinct modules tend to accumulate. Moreover, wrong poses in the detections can lead to ID switches in the matching process.

We propose an alternative approach that recasts visual inference problems as inverse rendering (IR) tasks, jointly solving them at test time by optimizing over the latent space of a generative object representation. Specifically, we combine object reconstruction through the inversion of an object class prior with a 3D object tracking pipeline. This approach allows us to simultaneously reason about an object's 3D shape, appearance, and three-dimensional trajectory from monocular image input only. The location, pose, shape, and appearance parameters corresponding to the anchor objects are then iteratively refined via test-time optimization to minimize the distance between their corresponding rendered objects and the given input image. Rather than directly predicting scene and object attributes, we optimize over an efficient representation to synthesize an image that best explains the observed image.

Our method hinges on an efficient rendering pipeline and generative object representation at its core. While the approach is not tied to a specific object representation, we

*equal contribution

adopt GET3D [16] as the generative object prior, that *is only trained on synthetic data* to synthesize textured meshes and corresponding images with an efficient differentiable rendering pipeline. Note that popular volumetric representations [55, 62] do not exploit class-specific priors and require expensive volume sampling.

Our proposed method builds on the strong implicit geometry priors embedded in our rendering forward model, solving *different prediction tasks simultaneously*. For instance, multi-object tracking, shape and texture retrieval, and object pose estimation – typically considered disjoint tasks – are jointly solved by our method by optimizing over object rendering parameters such as 3D location and pose. Our method outputs object pose as a byproduct, merely by learning to represent objects of a given class. Recovering object attributes as a result of inverse rendering also provides *interpretability "for free"*: once our proposed method detects an object at test time, it can extract the parameters of the corresponding representation alongside the reconstructed input view. This ability allows for reasoning about failure cases.

We validate that our method naturally exploits 3D geometry priors and *generalizes across unseen domains and unseen datasets*. After training solely on simulated data, we test on nuScenes [7] and Waymo [67] datasets, and although untrained, we find that our method is on par with existing 3D multi-object tracking methods [23, 70, 72, 90] on monocular image data.

In summary, we make the following contributions.

- We introduce an inverse rendering method for 3D-grounded monocular multi-object tracking. Instead of formulating tracking as a feed-forward prediction problem, we propose to solve it as an inverse image fitting problem optimizing over the latent embedding space of neural scene representations.
- We investigate the interpretability of our method using the generated image produced by our method during test-time optimization.
- Out method is only trained on synthetic data. We validate the generalization capabilities of our method by evaluating on unseen automotive datasets.

Scope and Limitations While facilitating inverse rendering, the iterative optimization in our method makes it slower than classical object-tracking methods based on feed-forward networks. We hope to address this limitation in the future by accelerating the forward and backward passes with adaptive level-of-detail rendering techniques.

2. Related Work

Object Tracking is a challenging visual inference task that requires the detection and association of multiple objects. Specific challenges include highly dynamic scenes with partial or full occlusions, changes in appearance, and varying illumination conditions [66, 77, 84]. In this section, we first review classical methods as well as deep-learning-based detection and association methods. Next, we discuss 3D scene representations and inverse rendering.

3D Object Tracking. An extensively investigated line of work proposes tracking by detection, i.e., to solve the task by first detecting scene objects and then learning to find the associations between the detected objects over multiple frames [3, 5, 6, 8, 25, 74, 75]. In addition to association, 3D tracking requires the estimation of object pose. Since directly predicting 3D object pose is challenging [24], most existing 3D tracking methods rely on some explicit depth measurements in the form of Lidar point clouds [1, 15, 85], hybrid camera-lidar measurements [24] or stereo information[18, 51]. Weng *et al.* [72] proposed a generic tracking method that combines a 3D Kalman filter and the Hungarian algorithm for matching on an arbitrary object detector.

Only recent work [9, 23, 44, 76, 90] tackles monocular 3D tracking. Hu *et al.* [23] relies on similarity across different viewpoints to learn rich features for tracking. DEFT [9] jointly trains the feature extractor for detection and tracking using the features to match objects between frames. In contrast, Marinello *et al.* [44] use an off-the-shelf tracker and enhance image features with 3D motion and bounding box information. Zhou *et al.* [90] rely on a minimal input of two frames and predicted heatmaps to perform simultaneous detection and tracking. Some 3D tracking methods rely on motion models [12, 46, 60] such as the Kalman Filter [26]. Recent methods also make use of optical flow predictions [42], learned motion models metrics [82], long short-term memory modules (LSTM) [9, 23, 44] and more recently transformer modules [54, 70]. All the above methods rely on a feed-forward image encoder backbone to predict object features. Departing from this approach, we propose a multi-object tracking method that directly optimizes a consistent three-dimensional reconstruction of objects and 3D motion via an inverted graphics pipeline.

3D Scene Representations, Generation and Neural Rendering. A growing body of work addresses joint 3D reconstruction and detection from monocular cameras. Existing methods have exploited different geometrical priors [43] for this task, including meshes [2], points [34], wire frames [21], voxels [79] CAD models or implicit functions [52] signed distance functions (SDFs) [87]. Early approaches in neural rendering represent the scene explicitly by, e.g., encoding texture or radiance on the estimated scene geometry [68] or using volumetric pixels (Voxels) [65]. Other methods represent 3D scenes *implicitly*. This includes the successful NeRF method [45] and variants that have been extended to dynamic scenes [52, 56, 86]. To allow the handling of semi-transparent objects, these representation models refrain from explicitly representing object surfaces. Signed distance fields represent surfaces of watertight objects as a zero level-set [14, 29, 55] modeling a Signed Distance Function (SDF). Adding textures to surface models allows for disentangling object shape from appearance [32, 78]. In recent years ideas from generative imaging models, such as GANs [27, 28], VAEs and diffusion models [22, 48] have been applied to the 3D domain [14, 16, 20, 62]. Generative models can either be used for pure generation [62] or provide prior knowledge for downstream tasks. Starting from a good prior can drastically improve the efficiency of inverse tasks, such as IR. While Gina3D [62] provides a prior on in-the-wild objects its volumetric rendering pipeline adds another layer of complexity sampling the full volume. We therefore rely on GET3D [16] generating a mesh as a prior object model and renders through rasterization, profiting from from graphic pipelines optimized over decades.

Inverse Rendering. Inverse rendering methods conceptually "invert" the graphics rendering pipeline, which generate images from 3D scene descriptions, and instead estimate the 3D scene properties, i.e., geometry, lighting, depth, and object poses based on input images. Recent work [38, 71, 83] successfully achieved joint optimization of a volumetric model and unknown camera poses from a set of images merely by back-propagating through a rendering pipeline. Another area of inverse rendering focuses on material and lighting properties [19, 49, 50], to find a representation that best models the observed image.

To the best of our knowledge, we present the first method that employs an inverse rendering approach for multi-object 3D tracking, *without any feed-forward prediction* of object features – only given 2D image input.

3. Tracking by Inverse Rendering

We cast object tracking as a test-time inverse rendering problem that fits generated multi-object scenes to the observed image frames. First, we discuss the proposed scene representation we fit. Next, we devise our rendering-based test-time optimization at the heart of the proposed tracking approach. We employ an object-centric scene representation. We model the underlying 3D scene for a frame observation as a composition of all object instances without the background scene.

Object Prior. To represent a large, diverse set of instances per class, we define each object instance o as a sample from a distribution $\boldsymbol{O}$ over all objects in a class, that is

$$\boldsymbol{O} \sim f(\boldsymbol{o}), \tag{1}$$

where f is a learned function over a known prior object distribution. Here, the prior distribution is modeled by a differentiable generative 3D object model $o_p = G(z_p)$, that maps a latent embedding z_p to an object instance o_p, the object p. In particular, the latent space comprises two disentangled spaces $\boldsymbol{z}_S$ and $\boldsymbol{z}_T$ for shape S and texture T.

Given an object-centric camera projection $\mathbf{P}_c = \mathbf{K}_c\mathbf{T}_c$, where $\mathbf{K}_c$ is the camera intrinsic matrix, $\mathbf{T}_c = [\mathbf{R}_c|\mathbf{t}_c]$ is composed of rotation $\mathbf{R}$ and translation $\mathbf{t}$ of the camera c, a differentiable rendering method $R(o_p, c)$, such as rasterization for meshes or volumetric rendering for neural fields, renders an image $I_{c,p}$, a 2D observation of the 3D object o_p. While our method is general, implementation details of the generator and rendering method are provided in the implementation section.

Scene Composition. We model a multi-object scene as a scene graph composed of transformations in the edges and object instances in the leaf nodes, similar to Ost *et al.* [52]. Object poses are described by the homogeneous transformation matrix $\mathbf{T}_p \in \mathbb{R}^{4x4}$ with the translation $\mathbf{t}_p$ and orientation $\mathbf{R}_p$ in the reference coordinate system. The camera pose $\mathbf{T}_c \in \mathbb{R}^{4x4}$ is described in the same reference coordinate system. The relative transformation of the camera c and each object instance o can be computed through edge traversal in the scene graph as

$$\mathbf{T}_{c,p} = \text{diag}\left(\frac{1}{s_p}\right)\mathbf{T}_p\mathbf{T}_c^{-1}, \tag{2}$$

where the factor s_p is a scaling factor along all axes to allow a shared object representation of a unified scale. This canonical object scale is necessary to represent objects of various sizes, independent of the learned prior on shape and texture. The object centric projection $\mathbf{P}_{c,p} = \mathbf{K}_c\mathbf{T}_{c,p}$ is used to render the RGB image $I_{c,p} \in \mathcal{R}^{H\times W\times 3}$ and mask $M_{c,p} \in [0,1]^{H\times W}$ for each object/camera pair.

Individual rendered RGB images are ordered by object distance $\|\mathbf{t}_{c,p}\|$, such that $p = 1$ is the shortest distance. Using the Hadamard Product of the non-occluded mask γ_p all N_o, object images are composed into a single image

$$\begin{aligned}
\hat{I}_c &= \sum_{k=1}^{N_o} R\left(G\left(\mathbf{z}_{S,p}, \mathbf{z}_{T,p}\right), \mathbf{P}_{c,p}\right) \circ \gamma_p, \text{ where} \\
\gamma_p &= \max\left(\left(\mathbf{M}_{c,p} - \sum_{q=1}^{p}\mathbf{M}_{c,q}\right), \mathbf{0}^{H\times W}\right),
\end{aligned} \tag{3}$$

where instance masks are generated in the same fashion.

3.1. Inverse Multi-Object Scene Rendering.

We invert the differentiable rendering model defined in Eq. 3 by optimizing the set of all object representations in

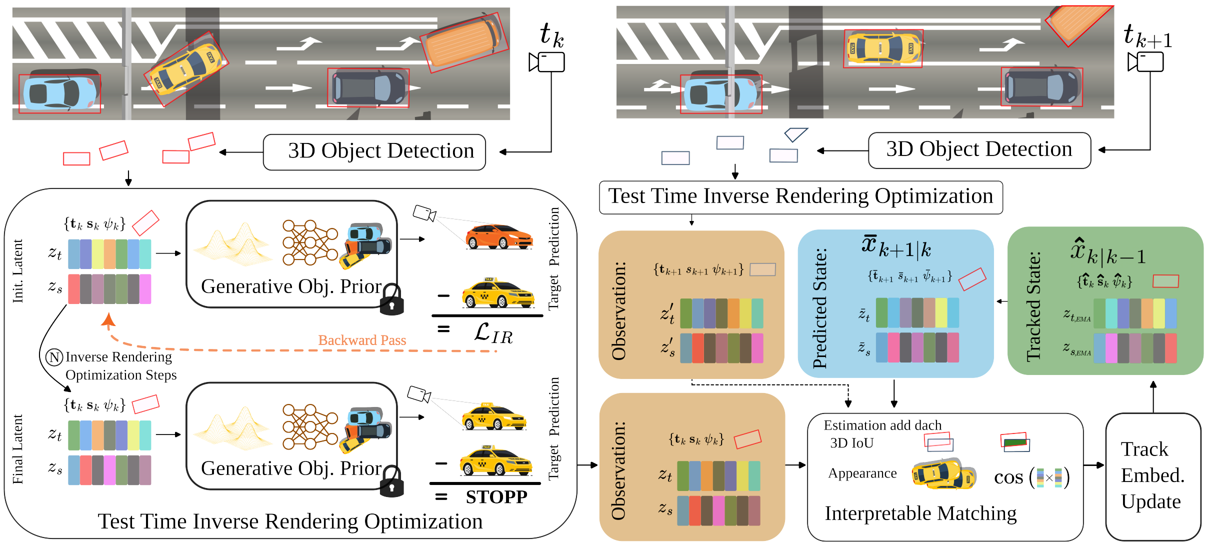

Figure 1. **Inverse Rendering for Monocular Multi-Object Tracking.** For each 3D Detection, we initialize the embedding codes of an object generator $\mathbf{z}_S$ for shape and $\mathbf{z}_T$ for texture. This prior trained model is frozen and only the embedding representation of both modalities together with the pose and size are optimized through inverse rendering to best fit the image observation. Inverse-rendered embeddings and refined object locations are provided to the matching stage to match predicted states of tracked objects of the past and the new observations. Matched tracklets are updated, unmatched detections and tracklets discarded before predicting states in the next step.

a given image I_c with gradient-based optimization. We assume that, initially, each object o_p is placed at a pose $\hat{\mathbf{T}}_{c,p}$ and scaled with $\hat{s}_p$ near its underlying location. We represent object orientations in their respective Lie algebraic form $\mathfrak{so}(3)$. We further sample an object embedding $\hat{\mathbf{z}}_{S,p}$ and $\hat{\mathbf{z}}_{T,p}$ in the respective latent embedding space.

For in-the-wild images, I_c is not just composed of sampled object instances but other objects and the scene background. Since our goal for tracking is the reconstruction of all object instances of specific object classes, a naïve ℓ_2 image matching objective of the form $\|I_c - \hat{I}_c\|_2$ is noisy and challenging to solve with vanilla stochastic gradient descent methods. To tackle this issue, we optimize visual similarity in the generated object regions instead of the full image. We optimize only on rendered RGB pixels and minimize

$$\mathcal{L}_{RGB} = \| \left(I_c - \hat{I}_c \right) \circ \hat{M}_{I_c} \|_2, \text{ with}$$
$$\text{with } \hat{M}_{I_c} = \min \left(\sum M_{c,p}, \mathbf{1} \right).$$

The mask of all foreground/object pixels $\hat{M}_{I_c}$ is computed as the sum over all object masks $M_{c,p}$ in the frame rendered by camera c. We employ a learned perceptual similarity metric [88] (LPIPS) on image patches of each object, that is

$$\mathcal{L}_{perceptual} = \text{LPIPS}_{patch} \left(I_c, \hat{I}_{c,p} \right). \tag{4}$$

The combined loss function of our method is

$$\mathcal{L}_{IR} = L_{RGB} + \lambda \mathcal{L}_{perceptual}, \tag{5}$$

which we optimize for shape and appearance latent codes, position, rotation, and scale, that is

$$\hat{\mathbf{z}}_{S,p}, \hat{\mathbf{z}}_{S,p}, \hat{s}_p \hat{\mathbf{t}}_p, \hat{\mathbf{R}}_p = \arg\min \left(\mathcal{L}_{IR} \right). \tag{6}$$

Instead of using vanilla stochastic gradient descent methods, we propose an alternating optimization schedule of distinct properties that includes aligning z_S before z_T, to reduce the number of total optimization steps. A detailed implementation and validation of all design choices of the optimization are presented in the Supplementary Material.

3.2. 3D Tracking via Inverse Rendering

Next, we describe the proposed method for tracking multiple dynamic objects with the inverse rendering approach from above. The approach tracks objects in the proposed representation across video frames and is illustrated in Fig. 1. For readability, we omit p and the split of $\mathbf{z}$ into $\mathbf{z}_S$ and $\mathbf{z}_T$ in the following.

Initial Object and Pose Estimation. Common to tracking methods, we initialize with a given initial 3D detection on image $I_{c,k}$, and we initialize object location $\mathbf{t}_k = [x, y, z]_k$, scale $s_k = \max(\mathrm{w}_k, \mathrm{h}_k, \mathrm{l}_k)$ using the detected bounding box dimensions and heading ψ_k in frame k. We then find an optimal representation z_k, and a refined location and rotation of each object o via the previously introduced inverse rendering pipeline for multi-object scenes. The resulting location, rotation, and scale together form the observation vector

$$\mathbf{y}_k = [\mathbf{t}_k, s_k, \psi_k]. \tag{7}$$

Figure 2. Tracking via Inverse Neural Rendering on nuScenes [7]. From left to right, we show (i) observed images from diverse scenes at timestep $k = 0$; (ii) an overlay of the optimized generated object and its 3D bounding boxes at timestep $k = 0, 1, 2$ and 3. The color of the bounding boxes for each object corresponds to the predicted tracklet ID. We see that even in such diverse scenarios, our method does not lose any tracks and performs robustly across all scenarios, although the dataset is unseen.

Prediction. While not confined to a specific dynamics model, we use a linear state-transition model $\mathbf{A}$, for the objects state $\mathbf{x}_k = [x, y, z, s, \psi, w, h, l, x', y', z']_k$, and a forward prediction using a Kalman Filter [26], a vanilla approach in 3D object tracking [72]. The predicted state in frame k given the object tracked in $k-1$ is

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{A}\hat{\mathbf{x}}_{k-1|k-1} \text{ and } \mathbf{P}_{k|k-1} = \mathbf{A}\mathbf{P}_{k-1|k-1}\mathbf{A}^T + \mathbf{Q} \tag{8}$$

is the predicted *a priori* covariance matrix modeling the uncertainty in the predicted state.

Interpretable Latent Matching. In the matching stage, all optimal object representations o_p in frame k are matched with *tracked* and *lost* objects from $k-1$. Objects are matched based on appearance and location with a weighted affinity score

$$A = w_{IoU} A_{IoU} + w_z A_z + w_c D_{centroid}, \tag{9}$$

where A_{IoU} is the IoU computed over the predictions of tracked object predictions $\mathbf{x}_{k|k-1}$ and refined observations. Here, the object affinity A_z is computed as the cosine distance of tracked object latent embeddings $\mathbf{z}$. In addition to that the Euclidean distance between the center $D_{centroid}$ adds additional guidance. We add no score for unreasonable distant tracked objects and detections.

We compute the best combination of tracked and detected objects using the Hungarian algorithm [35], again a conventional choice in existing tracking algorithms. Matched tracklet and object pairs are kept in the set of *tracked* objects and the representation of the corresponding detections is discarded, while unmatched detections are added as new objects. Unmatched tracklets are set to *lost* with a lost frame counter of one. Objects that were not detected in previous frames are set to *tracked* and their counter is reset to 0. Objects with a lost frame count higher than lifespan N_{life}, or outside of the visible field, are removed.

Track and Embedding Update. In the update step, we refine each object embedding $\mathbf{z}$ and motion model $\mathbf{y}_k$ given the result of the matching step. Embeddings are updated through an exponential moving average

$$\mathbf{z}_{k,EMA} = \beta\mathbf{z}_k + (1-\beta)\mathbf{z}_{k-1,EMA} \text{ with } \beta = \frac{2}{T-1} \tag{10}$$

over all past observations of the object, where T is the number of observed time steps of the respective instance. The observation $\mathbf{y}_k$ is used to update the Kalman filter. The optimal Kalman gain

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}^T(\mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^T + R)^{-1} \tag{11}$$

is updated to minimize the residual error of the predicted model and the observation. The observation $\mathbf{y}_k$ is used to estimate the object state as

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{y}_k - \mathbf{H}\hat{\mathbf{x}}_{k|k-1}) \tag{12}$$

and with

$$\mathbf{P}_k = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{H}\mathbf{P}_{k|k-1} \tag{13}$$

the *a posteriori* of the covariance matrix is updated.

Figure 3. Without changing the model or training on the dataset, our proposed method can generalize well to the Waymo Open Driving Dataset [67]. Similar to Fig 2, from left to right, we show (i) observed images from diverse scenes from the dataset at timestep $k = 0$; (ii) an overlay of the closest generated object and predicted 3D bounding boxes at timestep $k = 0, 1, 2$ and 3. The color of the bounding boxes for each object corresponds to the predicted tracklet ID. Our method does not lose any tracks even on a different unseen dataset in diverse scenes, validating that the approach generalizes.

3.3. Implementation Details

Representation Model. We employ the GET3D [16] architecture as object model G. Following StyleGAN [27, 28] embeddings z_T and z_S are mapped to intermediate style embeddings w_S and w_T in a learned $\boldsymbol{W}$-space, which we optimize over instead of $\boldsymbol{Z}$-space. Style embeddings condition a generator function that produces tri-planes representing object shapes as Signed Distance Fields (SDFs) and textures as texture fields. We deliberately *train our generator on synthetic data only*, see experiments below. Differentiable marching tetrahedra previously introduced in DMTet [63] extract a mesh representation and Images are rendered with a differentiable rasterizer [36].

Optimization. To solve Eq. 5, we propose an optimization schedule, that first optimizes a coarse color, and then jointly optimizes the shape and the positional state of each object. As a backbone of the learned perceptual loss, we utilize a pre-trained VGG16 [64] and utilize individual output feature map similarities at different points of the optimization. We find that color and other low-dimensional features are represented in the initial feature maps and those are better guidance for texture than high-dimensional features as outputs of the later blocks. These features have a more informative signal for shape and object pose. We use the average of the first and second blocks in the optimization for $\mathbf{z}_T$, while the combined perceptual similarity loss guides the optimization of $\mathbf{z}_T$ and the pose.

We initialize all object embeddings with the same fixed values inside the embedding space, take two optimization steps solely on color utilizing the described loss, and then freeze the color for the joint optimization of the shape and pose. We regularize out-of-distribution generations with

$$\mathcal{L}_{embed} = \alpha_T \mathbf{z}_T + (1-\alpha_T)\mathbf{z}_T^{avg} + \alpha_S \mathbf{z}_S + (1-\alpha_S)\mathbf{z}_S^{avg} \tag{14}$$

that minimizes a weighted distance in each dimension with respect to the average embedding $\mathbf{z}_S$ or $\mathbf{z}_T$ respectively. For optimization, we use the ADAM optimizer [31]. The final loss function combines the RGB, perceptual cost 5 and the regularization 14 with $\lambda = 10$, $\alpha_T = 0.7$ and $\alpha_S = 0.7$. We freeze color after two steps of optimization and optimize the shape and scale for three more steps, adding translation and rotation only in the last two steps.

4. Experiments

In the following, we assess the proposed method. Having trained our generative scene model solely on simulated data, we test the generalization capabilities on the nuScenes [7] and Waymo [67] – both datasets are unseen by the method. We analyze generative outputs of the test-time optimization and compare against existing 3D multi-object tracking methods [23, 70, 72, 90] on monocular image data.

4.1. Single-Shot Object Retrieval and Matching

Although trained only on ShapeNet [10], our method is capable of fitting to observed objects in real datasets that match the vehicle type, color, and overall appearance, effectively making our method dataset-agnostic. We analyze the generations during optimization in the following.

Optimization. Given an image observation and coarse detections, our method aims to find the best 3D representation, including pose and appearance, solely through inverse rendering. In Fig. 4 we analyze this iterative optimization process, following a scheduled optimization as described in Sec. 3.3. We observe that the object's color is inferred in only two steps. Further, we can observe that even though the initial pose is incorrect, rotation and translation are optimized jointly through inverse rendering together with the shape and scale of the objects, recovering from sub-optimal

Method	Modality	AMOTA ↑	AMOTP (m) ↓	Recall ↑	MOTA ↑	No Training on Dataset
PF-Track [54]	Camera	0.622	0.916	0.719	0.558	×
QTrack [82]	Camera	0.692	0.753	0.760	0.596	×
QD-3DT [23]	Camera	0.425	1.258	0.563	0.358	×
CenterTrack [90] (Vision [89])	Camera	0.202	1.195	0.313	0.134	×
AB3DMOT + CP [89]	Camera	0.387	1.158	0.506	0.284	✓
Inverse Neural Rendering (ours) + CP [89]	Camera	0.248	1.140	0.485	0.193	✓

Table 1. **Qualitative Evaluation for Camera-only Multi-Object Tracking**. Quantitative results on "cars" in the test split of the nuScenes tracking dataset [7]. IR-based rendering achieves comparable quality to AB3DMOT [72] on all metrics, while outperforming CenterTrack [90] and similar quality as QD-3DT [23]. Only very recent transformer-based methods, such as PF-Track [54] and the metric learning approach of Q-Track achieve a higher score. However, these methods require end-to-end training on each dataset.

Figure 4. **Optimization Process.** From left to right, we show (i) the observed image, (ii) the rendering predicted by the initial starting point latent embeddings, (iii) the predicted rendered objects after the texture code is optimized (iv) the predicted rendered objects after the translation, scale, and rotation are optimized, and (v) the predicted rendered objects after the shape latent code is optimized. The ground truth images are faded to show our rendered objects clearly. Our method is capable of refining the predicted texture, pose, and shape over several optimization steps, even if initialized with poses or appearance far from the target – all corrected through inverse rendering.

initial guesses. The shape representation close to the observed object is reconstructed in just 5 steps.

Our generative model does not predict specular textures and instead is restricted to diffuse reflectance. As such, it tends to reconstruct darker or lighter textures compensating for shadows from the environment and reflections of the sky. Prediction of reflectance and the integration and reconstruction of realistic environmental lighting is an exciting topic for future work.

4.2. Evaluation

To provide a fair comparison of 3D multi-object tracking methods using monocular inputs, we compare against existing methods by running all our evaluations with the method reference code. We only evaluate methods, that consider past frames, but have no knowledge about future frames, which is a different task. While our method does not store the full history length of all images, we allow such memory techniques for other methods. We only consider purely mono-camera-based tracking methods following a two-staged detect and track approach. We note that, in contrast to our method, the *baseline methods we compare to are finetuned on the respective training set*. For all methods, we use CenterPoint [85] as the detection method. We compare to CenterTrack [90] as an established learning-based baseline, the very recent PFTrack [54], a transformer-based tracking method, Qtrack [82] as a metric learning method, and QD-3DT [23] as an LSTM-based state tracker combined with image feature matching. We also compare to

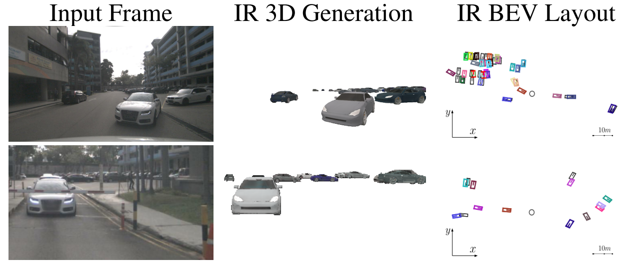

Figure 5. **Layout Generation Through Inverse Rendering.** From left to right, we show (i) observed image from a single camera, (ii) test-time optimized inverse rendered (IR) objects of class "car", and (iii) Bird's Eye View (BEV) layout of the scene. In the BEV layout, black boxes represent ground truth BEV boxes, and the colored boxes represent our predicted BEV boxes. The bottom shows a zoomed-in region at a 60 m distance (see BEV layout). Even in this setting, our method accurately recovers the 3D location, orientation, size, coarse appearance, and shape of the objects.

AB3DMOT [72] that builds on an arbitrary 3D detection algorithm and combines it with a modified Kalman filter to track the state of each object. This method is the most similar to our method in the sense that the Kalman filter parameters are not tuned for each dataset.

Validation on nuScenes. Tab. 1 reports quantitative results on the test split of the nuScenes tracking dataset [7] on the car object class for all six cameras. We list results for the multi-object tracking accuracy (MOTA) [4] metric, the AMOTA [72] metric, average multi-object tracking precision (AMOTP) [72] and recall of all methods. Our IR-based method achieves comparable results with the general tracker AB3DMOT [72] on all metrics. Surprisingly, established end-to-end trained baselines CenterTrack [90], which use the same vision-only detection backbone as in our approach, perform worse than our method in all metrics. Additionally, learning-based methods such as the end-to-end LSTM-based method QD-3DT [23] perform on par. Only the most recent transformer-based methods such as PF-Track [54] and the QTrack, which employ a quality-based association model on a large set of learned metrics, such as heatmaps and depth, achieve higher scores. Note again, that these methods, in contrast to the proposed method, have seen and trained on this dataset.

We visualize the rendered objects predicted by our tracking method in Fig. 2. We show an observed image from a single camera at time step $k = 0$, followed by rendered objects overlayed over the observed image at time step $k = 0, 1, 2$ and 3 along with their respective bounding boxes, with color-coded tracklets. We see that our method does not lose any tracks in challenging scenarios in diverse scenes shown here, from dense urban areas to suburban traffic crossings, and handles occlusions and clutter effectively. By visualizing the rendered objects as well as analyzing the loss values, our method allows us to reason about and explain success and failure cases effectively, enabling explainable 3D object tracking. The rendered output images provide interpretable inference results that explain successful or failed matching due to shadows, appearance, shape, or pose. For example, the blue car in the IR inference in Fig. 5 top row was incorrectly matched due to an appearance mismatch in a shadow region. A rendering model including ambient illumination may resolve this ambiguity, see further discussion in the Supplementary Material.

Fig 5 shows the inverse rendered scene graphs in isolation and birds-eye-view tracking outputs on a layout level. Our method accurately recovers the object poses, instance types, appearance, and scale. As such, our approach directly outputs a 3D model of the full scene, i.e., layout and object instances, along with the temporal history of the scene recovered through tracking – a rich scene representation that can be directly ingested by downstream planning and control tasks, or simulation methods to train downstream tasks. As such, the method also allows us to reason about the scene by leveraging the 3D information provided by our predicted 3D representations. The 3D locations, object orientations, and sizes recovered from such visualizations can not only enable us to explain the predictions of our object tracking method, especially in the presence of occlusions or ID switches but also be used in other downstream tasks that require rich 3D understanding, such as planning.

Method	AMOTA ↑	Recall ↑	MOTA↑
No Schedule	0.102	0.224	0.110
$\mathcal{L}_{RGB}$ - Eq. 4	N/A	N/A	N/A
$\mathcal{L}_{perceptual}$ - Eq. 4	0.100	0.251	0.101
$\mathcal{L}_{IR}$ - Eq.5	0.103	0.236	0.112
$\mathcal{L}_{IR}$ & $\mathcal{L}_{embed}$ - Eq.14	**0.112**	**0.264**	**0.113**

Table 2. **Ablation Experiments on Optimization Schedule and Loss Components.** Ablations were run on a small subset of the nuScenes [7] validation set. $\mathcal{L}_{RGB}$ fails due to the optimizer fitting objects to the background instead, increasing the size of each object resulting in out of memory.

Validation on Waymo. Next, we provide qualitative results from the 3D tracking on the validation set of the Waymo Open Driving Dataset [67] in Fig. 3. The *only public results* on the provided test set are presented in QD-3DT [23], achieving non-interpretable results. While the size of the dataset and its variety is of high interest for all autonomous driving tasks, Hu et al. [23] conclude that vision-only test set evaluation is not representative of a test set developed for surround view lidar data on partial unobserved camera images only. As such, we provide here qualitative results in Fig. 3, which validate that the method achieves tracking of similar quality on all datasets, providing a generalizing tracking approach. We show that our method does not lose any tracks on Waymo scenes in diverse conditions.

4.3. Ablation Experiments

As ablation experiments, we analyze the optimization schedule and loss function components, applying them to a subset of scenes from the nuScenes validation set. We deliberately select this smaller validation set due to its increased difficulty. Our findings reveal a crucial insight: the strength of our method lies not in isolated loss components but in their synergetic integration. Specifically, the amalgamation of pixel-wise, perceptual, and embedding terms significantly enhances AMOTA, MOTA, and Recall metrics. Moreover, the absence of an optimization schedule led to less robust matching. However, the core efficacy of our method remained intact, thanks to its reliance on consistent distance and IoU criteria for object matching. This nuanced understanding underscores the importance of component interplay in our method.

5. Conclusion

We investigate inverse neural rendering as an alternative to existing feed-forward tracking methods. Specifically, we recast 3D multi-object tracking from RGB cameras as an inverse test-time optimization problem over the latent space of pre-trained 3D object representations that, when rendered, best represent object instances in a given input image. We optimize an image loss over generative latent spaces that inherently disentangle shape and appearance properties. This approach to tracking also enables examining the reconstructed objects, reasoning about failure situations, and resolving ambiguous cases – rendering object layouts and loss function values provides interpretability "for free". We validate that the method has high generalization capabilities, and without seeing a dataset, performs on par with existing tracking methods. In the future, we hope to investigate not only object detection with inverse rendering but broad, in-the-wild object classes via conditional generation methods – towards unlocking analysis-by-synthesis in vision with generative neural rendering. '

References

[1] Claudia Álvarez-Aparicio, Ángel Manuel Guerrero-Higueras, Francisco Javier Rodríguez-Lera, Jonatan Ginés Clavero, Francisco Martín Rico, and Vicente Matellán. People detection and tracking using lidar sensors. *Robotics*, 8(3):75, 2019.

[2] Deniz Beker, Hiroharu Kato, Mihai Adrian Morariu, Takahiro Ando, Toru Matsuoka, Wadim Kehl, and Adrien Gaidon. Monocular differentiable rendering for self-supervised 3d object detection. In *European Conference on Computer Vision*, pages 514–529. Springer, 2020.

[3] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. Tracking without bells and whistles. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 941–951, 2019.

[4] Keni Bernardin, Alexander Elbs, and Rainer Stiefelhagen. Multiple object tracking performance metrics and evaluation in a smart room environment. In *Sixth IEEE International Workshop on Visual Surveillance, in conjunction with ECCV*. Citeseer, 2006.

[5] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE international conference on image processing (ICIP)*, pages 3464–3468. IEEE, 2016.

[6] Michael D Breitenstein, Fabian Reichlin, Bastian Leibe, Esther Koller-Meier, and Luc Van Gool. Robust tracking-by-detection using a detector confidence particle filter. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1515–1522. IEEE, 2009.

[7] 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.

[8] Jinkun Cao, Xinshuo Weng, Rawal Khirodkar, Jiangmiao Pang, and Kris Kitani. Observation-centric sort: Rethinking sort for robust multi-object tracking. *arXiv preprint arXiv:2203.14360*, 2022.

[9] Mohamed Chaabane, Peter Zhang, J Ross Beveridge, and Stephen O'Hara. Deft: Detection embeddings for tracking. *arXiv preprint arXiv:2102.02267*, 2021.

[10] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.

[11] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014.

[12] SY Chen. Kalman filter for robot vision: a survey. *IEEE Transactions on industrial electronics*, 59(11):4409–4420, 2011.

[13] Yilun Chen, Zhiding Yu, Yukang Chen, Shiyi Lan, Anima Anandkumar, Jiaya Jia, and Jose M Alvarez. Focalformer3d: Focusing on hard instance for 3d object detection. 2023.

[14] Gene Chou, Ilya Chugunov, and Felix Heide. Gensdf: Two-stage learning of generalizable signed distance functions. In *Proc. of Neural Information Processing Systems (NeurIPS)*, 2022.

[15] Ayush Dewan, Tim Caselitz, Gian Diego Tipaldi, and Wolfram Burgard. Motion-based detection and tracking in 3d lidar scans. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 4508–4513. IEEE, 2016.

[16] Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojcic, and Sanja Fidler. Get3d: A generative model of high quality 3d textured shapes learned from images. In *Advances In Neural Information Processing Systems*, 2022.

[17] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[18] Mariia Gladkova, Nikita Korobov, Nikolaus Demmel, Aljoša Ošep, Laura Leal-Taixé, and Daniel Cremers. Directtracker: 3d multi-object tracking using direct image alignment and photometric bundle adjustment. *arXiv preprint arXiv:2209.14965*, 2022.

[19] Yuan-Chen Guo, Di Kang, Linchao Bao, Yu He, and Song-Hai Zhang. Nerfren: Neural radiance fields with reflections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18409–18418, 2022.

[20] Zekun Hao, Arun Mallya, Serge Belongie, and Ming-Yu Liu. Gancraft: Unsupervised 3d neural rendering of minecraft worlds. *arXiv preprint arXiv:2104.07659*, 2021.

[21] Tong He and Stefano Soatto. Mono3d++: Monocular 3d vehicle detection with two-scale 3d hypotheses and task priors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8409–8416, 2019.

[22] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[23] Hou-Ning Hu, Yung-Hsu Yang, Tobias Fischer, Trevor Darrell, Fisher Yu, and Min Sun. Monocular quasi-dense 3d object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[24] Kemiao Huang and Qi Hao. Joint multi-object detection and tracking with camera-lidar fusion for autonomous driving.

In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6983–6989. IEEE, 2021.

[25] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. Tracking-learning-detection. *IEEE transactions on pattern analysis and machine intelligence*, 34(7):1409–1422, 2011.

[26] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960.

[27] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[28] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020.

[29] Petr Kellnhofer, Lars C Jebe, Andrew Jones, Ryan Spicer, Kari Pulli, and Gordon Wetzstein. Neural lumigraph rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4287–4297, 2021.

[30] Aleksandr Kim, Aljoša Ošep, and Laura Leal-Taixé. Eagermot: 3d multi-object tracking via sensor fusion. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11315–11321. IEEE, 2021.

[31] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

[32] Lukas Koestler, Daniel Grittner, Michael Moeller, Daniel Cremers, and Zorah Lähner. Intrinsic neural fields: Learning functions on manifolds. *arXiv preprint arXiv:2203.07967*, 2, 2022.

[33] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven L Waslander. Joint 3d proposal generation and object detection from view aggregation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8. IEEE, 2018.

[34] Jason Ku, Alex D Pon, and Steven L Waslander. Monocular 3d object detection leveraging accurate proposals and shape reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11867–11876, 2019.

[35] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[36] Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. Modular primitives for high-performance differentiable rendering. *ACM Transactions on Graphics (TOG)*, 39(6):1–14, 2020.

[37] Yi Li, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei. Fully convolutional instance-aware semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2359–2367, 2017.

[38] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5741–5751, 2021.

[39] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[40] Zhijian Liu, Haotian Tang, Alexander Amini, Xingyu Yang, Huizi Mao, Daniela Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. *arXiv*, 2022.

[41] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[42] Jonathon Luiten, Tobias Fischer, and Bastian Leibe. Track to reconstruct and reconstruct to track. *IEEE Robotics and Automation Letters*, 5(2):1803–1810, 2020.

[43] Jiageng Mao, Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. 3d object detection for autonomous driving: A review and new outlooks. *arXiv preprint arXiv:2206.09474*, 2022.

[44] Nicola Marinello, Marc Proesmans, and Luc Van Gool. Triplettrack: 3d object tracking using triplet embeddings and lstm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4500–4510, 2022.

[45] 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, 2020.

[46] Hieu Tat Nguyen and Arnold WM Smeulders. Fast occluded object tracking by a robust appearance filter. *IEEE transactions on pattern analysis and machine intelligence*, 26(8): 1099–1104, 2004.

[47] Pha Nguyen, Kha Gia Quach, Chi Nhan Duong, Ngan Le, Xuan-Bac Nguyen, and Khoa Luu. Multi-camera multiple 3d object tracking on the move for autonomous vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2569–2578, 2022.

[48] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021.

[49] Merlin Nimier-David, Delio Vicini, Tizian Zeltner, and Wenzel Jakob. Mitsuba 2: A retargetable forward and inverse renderer. *Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, 38(6), 2019.

[50] Merlin Nimier-David, Zhao Dong, Wenzel Jakob, and Anton Kaplanyan. Material and Lighting Reconstruction for Complex Indoor Scenes with Texture-space Differentiable Rendering. In *Eurographics Symposium on Rendering - DL-only Track*. The Eurographics Association, 2021.

[51] Aljoša Osep, Wolfgang Mehner, Markus Mathias, and Bastian Leibe. Combined image-and world-space tracking in traffic scenes. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1988–1995. IEEE, 2017.

[52] Julian Ost, Fahim Mannan, Nils Thuerey, Julian Knodt, and Felix Heide. Neural scene graphs for dynamic scenes. In

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 2856–2865, 2021.

[53] Ziqi Pang, Zhichao Li, and Naiyan Wang. Simpletrack: Understanding and rethinking 3d multi-object tracking. *arXiv preprint arXiv:2111.09621*, 2021.

[54] Ziqi Pang, Jie Li, Pavel Tokmakov, Dian Chen, Sergey Zagoruyko, and Yu-Xiong Wang. Standing between past and future: Spatio-temporal modeling for multi-camera 3d multi-object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[55] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[56] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. *Proceedings of the IEEE International Conference on Computer Vision*, 2021.

[57] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 918–927, 2018.

[58] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[59] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[60] Samuel Scheidegger, Joachim Benjaminsson, Emil Rosenberg, Amrit Krishnan, and Karl Granström. Mono-camera 3d multi-object tracking using deep learning detections and pmbm filtering. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 433–440. IEEE, 2018.

[61] Sarthak Sharma, Junaid Ahmed Ansari, J Krishna Murthy, and K Madhava Krishna. Beyond pixels: Leveraging geometry and shape cues for online multi-object tracking. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3508–3515. IEEE, 2018.

[62] Bokui Shen, Xinchen Yan, Charles R. Qi, Mahyar Najibi, Boyang Deng, Leonidas Guibas, Yin Zhou, and Dragomir Anguelov. Gina-3d: Learning to generate implicit neural assets in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4913–4926, 2023.

[63] Tianchang Shen, Jun Gao, Kangxue Yin, Ming-Yu Liu, and Sanja Fidler. Deep marching tetrahedra: a hybrid representation for high-resolution 3d shape synthesis. *Advances in Neural Information Processing Systems*, 34:6087–6101, 2021.

[64] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[65] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhöfer. Deepvoxels: Learning persistent 3d feature embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[66] Arnold WM Smeulders, Dung M Chu, Rita Cucchiara, Simone Calderara, Afshin Dehghan, and Mubarak Shah. Visual tracking: An experimental survey. *IEEE transactions on pattern analysis and machine intelligence*, 36(7):1442–1468, 2013.

[67] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2446–2454, 2020.

[68] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering. *ACM Transactions on Graphics*, 38 (4):1–12, 2019.

[69] Chen Wang, Danfei Xu, Yuke Zhu, Roberto Martín-Martín, Cewu Lu, Li Fei-Fei, and Silvio Savarese. Densefusion: 6d object pose estimation by iterative dense fusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3343–3352, 2019.

[70] Shihao Wang, Yingfei Liu, Tiancai Wang, Ying Li, and Xiangyu Zhang. Exploring object-centric temporal modeling for efficient multi-view 3d object detection. *arXiv preprint arXiv:2303.11926*, 2023.

[71] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. Nerf–: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021.

[72] Xinshuo Weng, Jianren Wang, David Held, and Kris Kitani. 3d multi-object tracking: A baseline and new evaluation metrics. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10359–10366. IEEE, 2020.

[73] Xinshuo Weng, Yongxin Wang, Yunze Man, and Kris M Kitani. Gnn3dmot: Graph neural network for 3d multi-object tracking with 2d-3d multi-feature learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6499–6508, 2020.

[74] Nicolai Wojke and Alex Bewley. Deep cosine metric learning for person re-identification. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 748–756. IEEE, 2018.

[75] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)*, pages 3645–3649. IEEE, 2017.

[76] Jialian Wu, Jiale Cao, Liangchen Song, Yu Wang, Ming Yang, and Junsong Yuan. Track to detect and segment: An online multi-object tracker. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12352–12361, 2021.

[77] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking: A benchmark. In *Proceedings of the IEEE con-*

ference on computer vision and pattern recognition, pages 2411–2418, 2013.

[78] Fanbo Xiang, Zexiang Xu, Milos Hasan, Yannick Hold-Geoffroy, Kalyan Sunkavalli, and Hao Su. Neutex: Neural texture mapping for volumetric neural rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7119–7128, 2021.

[79] Yu Xiang, Wongun Choi, Yuanqing Lin, and Silvio Savarese. Data-driven 3d voxel patterns for object category recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1903–1911, 2015.

[80] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017.

[81] Xinge Zhu Qingqiu Huang Yilun Chen Hongbo Fu Xuyang Bai, Zeyu Hu and Chiew-Lan Tai. TransFusion: Robust Lidar-Camera Fusion for 3d Object Detection with Transformers. *CVPR*, 2022.

[82] Jinrong Yang, En Yu, Zeming Li, Xiaoping Li, and Wenbing Tao. Quality matters: Embracing quality clues for robust 3d multi-object tracking. *arXiv preprint arXiv:2208.10976*, 2022.

[83] Lin Yen-Chen, Pete Florence, Jonathan T Barron, Alberto Rodriguez, Phillip Isola, and Tsung-Yi Lin. inerf: Inverting neural radiance fields for pose estimation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1323–1330. IEEE, 2021.

[84] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *Acm computing surveys (CSUR)*, 38(4):13–es, 2006.

[85] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11784–11793, 2021.

[86] Wentao Yuan, Zhaoyang Lv, Tanner Schmidt, and Steven Lovegrove. Star: Self-supervised tracking and reconstruction of rigid objects in motion with neural rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13144–13152, 2021.

[87] Sergey Zakharov, Wadim Kehl, Arjun Bhargava, and Adrien Gaidon. Autolabeling 3d objects with differentiable rendering of sdf shape priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12224–12233, 2020.

[88] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 586–595, 2018.

[89] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. In *arXiv preprint arXiv:1904.07850*, 2019.

[90] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *European Conference on Computer Vision*, pages 474–490. Springer, 2020.

[91] 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.