University of Modena and Reggio Emilia
XXXVIII cycle of the International Doctorate School in
Information and Communication Technologies (ICT)
Doctor of Philosophy dissertation in
Computer Engineering and Science

UNIMORE
UNIVERSITÀ DEGLI STUDI DI
MODENA E REGGIO EMILIA

Methodologies in Computer Vision for 3D Digitization and Interpretation

Davide Di Nucci

Supervisor: Prof. Rita Cucchiara
Co-Supervisor: Prof. Roberto Vezzani
PhD Programme Coordinator: Prof. Luigi Rovati

Review committee composed of:

Matteo Tomei, Prometeia s.p.a
Nicola Conci, University of Trento

Tesi di dottorato finanziata dall'Unione europea - Next Generation EU, Missione 4, componente 2 "Dalla Ricerca all'Impresa" – Investimento 3.3 "Introduzione di dottorati innovativi che rispondono ai fabbisogni di innovazione delle imprese e promuovono l'assunzione dei ricercatori dalle imprese".

Abstract

Processing and interpreting three-dimensional data remain fundamental challenges in Computer Vision and Computer Graphics, with applications spanning autonomous driving, robotics, augmented reality, and digital security. Accurate 3D reconstruction is essential for enabling machines to perceive and interact with the physical world, yet achieving high-fidelity results under real-world constraints such as sparse viewpoints, variable illumination, and complex object geometries—remains demanding. This doctoral thesis addresses these challenges by focusing on the reconstruction, representation, and authentication of 3D scenes and vehicles, proposing innovative solutions grounded in neural fields and differentiable rasterization techniques.

The research builds upon the Gaussian Splatting paradigm, a powerful alternative to traditional volumetric representations for real-time rendering and reconstruction. Unlike Neural Radiance Fields (NeRF), which require dense sampling and extensive optimization, Gaussian Splatting offers superior efficiency and scalability, making it particularly suitable for automotive scenarios where data acquisition is often limited to sparse camera views. The proposed methods extend this paradigm to handle outdoor environments and vehicle-specific constraints, introducing strategies for robust geometry recovery under challenging lighting conditions and partial occlusions. Additionally, the thesis explores the integration of parametric models to regularize shape estimation and improve pose accuracy, enabling the reconstruction of articulated structures and fine-grained details critical for downstream tasks such as simulation, inspection, and virtual prototyping.

Beyond algorithmic development, this work contributes to the research ecosystem through the creation of dedicated datasets tailored to real-world vehicle scenarios. These datasets capture diverse conditions—including varying illumination, weather, and occlusion patterns—providing a rigorous benchmark for evaluating reconstruction fidelity and generalization. Complementing this effort, the thesis introduces advanced evaluation metrics that go beyond traditional photometric error, incorporating measures of structural consistency, geometric plausibility, and perceptual realism to better reflect industrial and safety-critical requirements.

A distinctive dimension of this research addresses the emerging challenge of fake detection in 3D. As generative models become increasingly capable of producing photorealistic and geometrically plausible content, distinguishing authentic

reconstructions from synthetic or manipulated data is crucial for maintaining trust in digital communication and forensic analysis. The thesis proposes a principled framework for assessing visual authenticity in three dimensions. Practical applications include reliable indicators for identifying 3D deepfakes and malicious manipulations, which pose growing risks in domains such as automotive security, insurance fraud detection, and virtual asset verification.

Overall, the integration of Gaussian Splatting, parametric modeling, and authenticity assessment represents a promising direction for advancing the fidelity, robustness, and trustworthiness of 3D reconstruction systems. By addressing both technical and conceptual challenges, this work contributes to the scientific foundations of 3D vision while offering practical solutions for industrial deployment and digital security.

Contents

Chapter 1

Introduction

Three-dimensional (3D) digitization has become a cornerstone of modern computer vision systems, enabling applications that range from industrial inspection and robotics to virtual/augmented reality and digital content creation. Compared to purely 2D analysis, building an explicit (or implicit) 3D representation allows reasoning about geometry, scale, occlusions, and physical consistency, and it supports downstream tasks such as fine-grained measurement, semantic understanding, simulation, and interaction.

In recent years, *novel view synthesis* has substantially changed the way 3D content is acquired from images. Neural Radiance Fields (NeRF) demonstrate that a scene can be represented as a continuous function that maps 3D position and view direction to density and color, enabling high-quality rendering from unseen viewpoints given multi-view observations and accurate camera poses [114]. More recently, 3D Gaussian Splatting (3DGS) introduced an explicit set of 3D Gaussians optimized from images, achieving real-time rendering performance while preserving competitive visual quality [79]. These advances position neural rendering as an intermediate representation between 2D imagery and classical 3D pipelines, and they open the door to performing recognition and analysis directly "in the scene", rather than per-image.

Despite this progress, deploying neural rendering reliably in real-world settings still faces practical bottlenecks. First, these methods are highly sensitive to camera calibration and pose accuracy: errors in extrinsics can lead to blurry reconstructions, distorted geometry, or inconsistent appearance. Most pipelines therefore rely on Structure-from-Motion (SfM) systems such as COLMAP [142],

which can be computationally heavy and may struggle on challenging captures (e.g., reflective surfaces, motion blur, low texture, or sparse viewpoints). Second, many application domains (notably automotive inspection) involve strong specularities, thin structures, and partial occlusions, and they often provide only sparse or constrained views of the object. Third, evaluation in these settings is limited by the availability of targeted datasets, reliable ground-truth geometry, and metrics that reflect not only photometric fidelity but also geometric correctness.

Beyond reconstruction, *3D interpretation* increasingly requires modeling humans and their interactions with environments. A representative example is 3D gaze estimation and gaze target understanding, where the goal is to infer where a person is looking in 3D space. This problem is inherently multi-modal (multiple plausible gaze targets may exist), and it benefits from context such as body pose, scene layout, and surrounding objects. Recent progress in generative modeling—particularly diffusion models [59, 137]—provides a natural mechanism to represent and sample multiple hypotheses, which is attractive for uncertainty-aware 3D reasoning.

Finally, the same generative capabilities that improve reconstruction and editing also introduce new security and trust challenges. Text-driven editing of 3D representations is becoming increasingly realistic and multi-view consistent, for example with Gaussian-splat editing methods such as GaussCtrl [181] and instruction-driven optimization strategies such as Instruct-GS2GS [167] built upon image-editing foundations like InstructPix2Pix [10]. As 3D content becomes easier to manipulate, reliably distinguishing authentic reconstructions from edited or synthetic ones becomes important for forensics, safety-critical decision making, and asset verification.

1.1 Scope and contributions

This thesis investigates methodologies for *3D digitization and interpretation* under real-world constraints, with an emphasis on: (i) accurate and efficient camera pose estimation and optimization, (ii) robust reconstruction from sparse views in vehicle-centric scenarios, (iii) uncertainty-aware 3D gaze estimation, (iv) dataset design and benchmarking for neural rendering on vehicles, and (v) authenticity assessment and fake detection for edited 3D content.

The main contributions are:

- **Keypoint-based camera pose refinement for vehicle reconstruction.** We introduce a lightweight optimization strategy that leverages 2D semantic

keypoints to improve camera registration for vehicle scenes, reducing the reliance on expensive SfM refinement and improving downstream reconstruction quality [34, 142].

- **Sparse-view vehicle reconstruction with improved geometry recovery.** We propose a reconstruction pipeline tailored to outdoor vehicle captures that augments limited views and replaces fragile components of classical pose estimation with more robust priors, leveraging modern geometric vision tools such as DUSt3R [176].

- **Diffusion-based multi-hypothesis 3D gaze estimation.** We develop a 3D gaze estimation approach that models ambiguity via sampling, building on diffusion-based generation [59, 137] and leveraging contextual cues from body pose and scene objects.

- **CarPatch: a synthetic benchmark for radiance fields on vehicle components.** We present a dataset and evaluation protocol designed for vehicle inspection with neural rendering, including depth and component-level annotations to enable geometry-focused evaluation beyond appearance metrics [32].

- **Fake3D evaluation and detection for edited Gaussian scenes.** We introduce a benchmark derived from multi-view 3D content to study 3D fake detection under modern editing pipelines [181, 167], and we propose a detector that operates on 3D Gaussian representations rather than relying solely on 2D render artifacts.

Chapter 2

Literature review

This thesis covers 3D digitization and interpretation across four pillars: (i) camera registration for 3D reconstruction, (ii) neural scene representations for novel view synthesis, (iii) human-centric inference from monocular cues (3D pose and 3D gaze), and (iv) synthetic data and forensics for detecting edited/generated content. This chapter positions the thesis contributions within the most relevant literature.

2.1 3D Reconstruction: From Geometry to Neural Rendering

Classical 3D digitization is built on multi-view geometry, robust estimation, and global refinement. Projective geometry provides the foundations for camera models and epipolar constraints [56], while robust fitting is typically handled with RANSAC [44]. Modern SfM systems reconstruct camera trajectories and sparse structure at scale [153, 142, 1], and MVS produces dense geometry through depth estimation and fusion [144, 189]. Benchmarks such as Tanks and Temples highlight failure modes in real captures (reflectance, texture sparsity, and viewpoint gaps) that remain relevant today [87]. SLAM systems are widely used to recover motion online (e.g., ORB-SLAM and ORB-SLAM3) [117, 11], but both SfM and SLAM can degrade in sparse, outdoor, or low-texture scenarios.

Neural rendering reframed reconstruction as learning a continuous scene representation from posed images. NeRF models a radiance field optimized via differentiable rendering [114], and a large body of work improved quality and effi-

ciency for unbounded scenes and anti-aliasing [5, 6, 7, 113]. Efficiency-oriented designs move capacity into explicit structures: hash-grid encodings enable fast convergence (Instant-NGP) [116], voxel/grid optimization accelerates training (DVGO) [156], tensor factorization reduces memory (TensoRF) [16], and explicit voxel-like methods remove MLPs (Plenoxels) [46]. For large environments, scalable strategies such as Mega-NeRF partition the scene to maintain tractability [165]. In parallel, generalizable approaches condition view synthesis on input images to reduce per-scene optimization (e.g., PixelNeRF, IBRNet) [193, 174].

2.2 Pose Accuracy as the Bottleneck

Across both radiance fields and explicit primitives, pose accuracy is often the dominant factor controlling final quality: small errors produce ghosting, floaters, and part-level misalignment. This motivated methods that jointly optimize camera poses and the scene representation. BARF performs bundle-adjustment inside NeRF using a coarse-to-fine strategy to stabilize joint optimization [98]. iNeRF instead estimates pose by inverting a trained radiance field [191]. Several works target weak or missing priors: Nope-NeRF optimizes NeRF without pose priors [8], NeRF– studies reconstruction without known camera parameters [179], and SiNeRF explores sinusoidal parameterizations for joint pose estimation and reconstruction [183]. Self-calibrating radiance fields further consider intrinsics/extrinsics within the learning loop [74]. Local-to-global registration improves the stability of bundle-adjusting radiance fields by structuring the optimization process [19].

Sparse and noisy settings remain particularly fragile because photometric losses provide weak constraints and encourage entanglement between pose and appearance. SPARF studies NeRF training under sparse views and noisy poses [163]. Injecting geometric cues helps: CorresNeRF incorporates correspondence priors to guide optimization [89]. In category-specific scenarios, lightweight semantic structure can be even more effective: vehicles exhibit repeatable part layouts, and keypoints provide stable correspondences under appearance changes. This motivates the keypoint-driven refinement adopted in KRONC [35].

2.3 Gaussian Splatting and Sparse View Reconstruction

3D Gaussian Splatting (3DGS) represents scenes as Gaussians optimized via differentiable rasterization, achieving real-time rendering with high quality [79]. Its explicit primitive-based representation enables fast rendering and a growing ecosystem (e.g., open-source tooling) [190]. Recent work explores efficiency via compression and compact representations [119, 40, 115] and semantic extensions for scene understanding [65].

Sparse view settings remain challenging for Gaussians, similar to NeRF. DNGaussian proposes depth normalization strategies to stabilize sparse-view optimization [93], while SplatFields regularizes Gaussian features through a neural field for improved sparse 3D/4D reconstruction [112]. Other approaches add matching and structure-consistency priors to improve reconstruction under limited viewpoints [192, 131, 128]. These methods directly relate to the sparse-view vehicle reconstruction scenarios studied in this thesis.

2.4 Vehicles and Synthetic Benchmarks for Fine-Grained Evaluation

Vehicle reconstruction adds practical difficulties (specularities, thin structures, repeating patterns) but also offers strong priors (e.g., bounded objects and repeatable part semantics). Vehicle understanding benchmarks emphasize the importance of 3D structure and part-level cues [155]. In driving contexts, datasets and simulators such as KITTI and CARLA enable scalable evaluation under realistic conditions [48, 38], and recent neural rendering systems target autonomous-driving scenarios [161, 185, 207].

A key limitation of common evaluation practice is that global image metrics can hide localized failures on critical components (e.g., lights, mirrors, bumpers). CarPatch was introduced to measure radiance-field performance *per vehicle component* under controlled synthetic settings [33], enabling the component-level analysis adopted later in this thesis.

2.5 Human-Centric Interpretation: 3D Pose and 3D Gaze

Monocular 3D human pose estimation progressed from simple lifting baselines [110] to temporal and attention-based models that reduce jitter and better exploit video context. VideoPose3D showed strong temporal modeling with convolutional architectures [130], while transformer-based designs improved spatio-temporal reasoning [205, 195, 94]. Motion-guided objectives encourage temporal consistency [173]. Ambiguity remains intrinsic, motivating probabilistic and multi-hypothesis formulations [9, 92]. Diffusion models provide a natural sampling-based mechanism for multi-modality, and recent work applies diffusion to multi-hypothesis 3D pose estimation [59, 31, 61, 49, 146].

Gaze estimation has traditionally relied on eye/face appearance, supported by datasets such as MPIIGaze and ETH-XGaze [199, 198] and unconstrained benchmarks like Gaze360 and RT-GENE [78, 43]. However, appearance-only cues can be insufficient in real scenes and at distance [23]. Gaze following therefore leverages scene context to predict attended targets in images [136] and, more recently, object candidates and transformers for end-to-end target detection [164, 162]. Extending supervision to 3D with RGB-D enables metric targets and stronger geometric constraints [66, 67]. Body pose also provides informative context; Pose2Gaze studies eye–body coordination for gaze prediction from full-body pose [68], and "gaze from afar" approaches exploit temporal coordination among eye, head, and body [121]. These trends motivate the generative, multi-hypothesis viewpoint adopted later in the thesis for ambiguous 3D gaze inference.

2.6 From Diffusion Editing to 3D-Aware Fake Detection

Forensic detection has historically relied on artifacts and generator fingerprints, and datasets such as FaceForensics++ and Celeb-DF highlighted the importance of generalization and compression robustness [139, 96]. Classic cues include visual artifacts and frequency signatures [111, 45, 175], while transfer strategies improve robustness across domains [29, 51]. With diffusion models, recent detectors target diffusion-specific traces and aim for universal generalization [28, 27, 123, 177, 145], and contrastive approaches further separate diffusion outputs from real imagery [4]. Inconsistency signals such as perspective and lighting remain relevant

when edits are not fully 3D-consistent [42, 41].

Crucially, editing is increasingly performed in ways that enforce multi-view consistency, weakening purely 2D artifact-based detection. Instruction-guided image editing exemplifies the usability of such tools [10], and 3D Gaussian editing methods enable controllable, multi-view consistent manipulation directly in the representation [181, 18, 167, 57]. This shift motivates representation-level detection and provenance: recent work studies vulnerabilities and proposes watermarking for 3D Gaussian splats [70, 20, 73, 69, 71].

Chapter 3

Bundle Adjustment

3.1 Introduction

Recent view synthesis techniques, such as Neural Radiance Fields [114] and 3D Gaussian Splatting [80], have revolutionized the reconstruction of both synthetic and real-world scenes. Training only on a few dozen images with known camera poses, they are able to provide high-quality renderings of the scene from novel viewpoints. Their representations emerged as an intermediate domain between the realms of 2D and 3D on which executing standard computer vision tasks such as object detection [64] or segmentation [15], paving the way for a variety of applications [185, 100, 101, 26]. For instance, owning a NeRF model and directly applying downstream recognition to it allows for easier inspection and assessment [62], compared to conducting the same analysis across individual pictures. The *vehicle inspection* task [33] has recently gained attention for the benefits it can bring to automotive industries and service providers. Its purpose is to generate high-quality renderings of specific car instances from different perspectives starting from a collection of images. This is exactly what NeRF models try to achieve in their broadest formulation, although with a focus on vehicle instances. Facilitating their meticulous inspection without on-site check-up from experts could be extremely convenient for car manufacturers to determine eventual external defects, for insurance companies to estimate post-accident damages and repair costs, or for car rentals for liability assessment automation.

However, applying standard novel view synthesis approaches to vehicle reconstruction highlights the following limitations: (i) recent NeRF and Gaus-

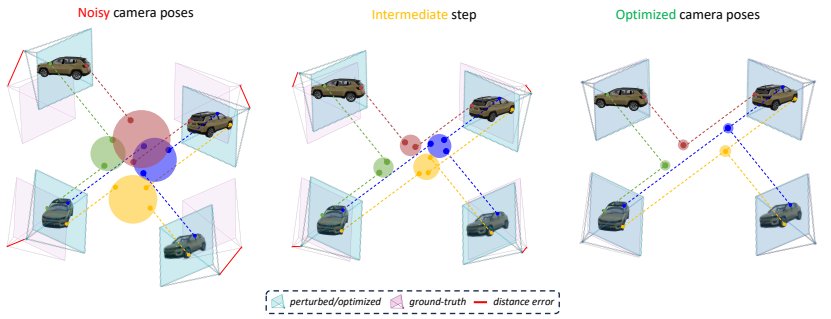


Figure 3.1: KRONC is a lightweight camera optimization algorithm for vehicle scenes which leverages 2D semantic keypoints. Keypoints are aligned in a common 3D world reference system, leading to precise camera registration.

sian Splatting methods still rely on classical Structure-from-Motion pipelines (*e.g.* COLMAP [143]) for camera parameters estimation, sometimes even exceeding the time and resource requirements of the actual downstream optimization [116, 156]; (ii) to the best of our knowledge, no dedicated datasets are available for comprehensive real-world vehicle reconstruction, with the evaluation still limited to synthetic scenes. Moreover, vehicles represent a well-studied and deeply modeled object category in the computer vision literature (for *e.g.* large-scale unbounded scene recognition and autonomous driving [48, 155, 38]), leading to a pool of established works and priors to be leveraged for *vehicle inspection*, too.

Among the others, generating high-quality car renderings from different perspectives has recently gained attention [33], striving to facilitate a meticulous inspection of vehicles post-accidents, thereby streamlining insurance and rental applications. This work specifically tackles the *vehicle inspection* task through novel view synthesis algorithms, by observing that: (i) recent NeRF methods achieve nearly instantaneous optimization of radiance fields, demonstrating state-of-the-art performance [116, 156]; (ii) most of them still rely on classical Structure from Motion (S*f*M) or Simultaneous Localization and Mapping (SLAM) pipelines for camera parameters estimation, often exceeding the time and resource requirements of the neural radiance field optimization; (iii) to the best of our knowledge, no dedicated datasets are available for high-fidelity and comprehensive real-world vehicle reconstruction. Motivated by these observations, in this work we propose an efficient algorithm for camera frame registration, which is able to break the dependency on heavy COLMAP-like pre-processing. Moreover, we release a

new benchmark (the KRONC-Dataset) of real-world vehicle scenes, with the aim of fostering novel view synthesis for *vehicle inspection*. To avoid S*f*M, recent works proposed Bundle-Adjusting NeRF [98, 19] by jointly reconstructing neural fields and registering camera frames. Their benefits come at the cost of integrating camera alignment in neural field optimization, which is not as straightforward as performing the two steps sequentially. We show that comparable performance can be obtained for vehicle reconstruction by keeping the two steps separated, exploiting a much lighter alternative to raw RGB pixels for camera optimization, *i.e.*2D keypoints, making computational overhead negligible. Our proposal combines the efficiency of bundle adjustment and the flexibility of stand-alone S*f*M packages, making it suitable for every downstream novel view synthesis technique not limited to NeRFs. As shown in Fig. 3.1, our KRONC algorithm projects semantically consistent keypoints from multiple views to a common 3D world's reference system and pushes them close together. Doing so, it tries to figure out both a reasonable configuration of cameras and meaningful depths for keypoints. Differently from incremental S*f*M and methods relying on pairwise image correspondences [163, 90], KRONC conducts global alignment, optimizing absolute camera positions depending on semantic keypoints shared between all viewpoints (without needing any matching algorithm).

On synthetic vehicle scenes our results show improved performance w.r.t. state-of-the-art bundle-adjustment methods, by adding the same camera noise to ground truth poses and attempting to restore a coherent disposition. On real scenes from the KRONC-dataset, we captured cars with mobile devices by performing a full 360° counterclockwise rotation around the car, which is the standard way of capturing scenes for large object reconstruction [159]. State-of-the-art bundle adjustment solutions struggle to converge in this setting. By coarsely initializing the poses of the cameras following a simple handcrafted circular trajectory, our keypoint-based registration method is able to find a good camera arrangement even when reducing the number of input images by 75%, while COLMAP performance rapidly drops. To sum up, our contributions encompass the following:

- We present the KRONC-dataset of real-world, high-quality car scenes, specifically devised for novel view synthesis in the context of *vehicle inspection*.
- We introduce an efficient keypoint-based camera registration (KRONC) algorithm to be executed before neural radiance field optimization, keeping the two separate steps and allowing for higher flexibility compared to bundle-adjusting NeRFs from noisy cameras.
- On real scenes, we leverage the typical behavior of capturing a scene by

Table 3.1: Summary of the KRONC-dataset: for each scene, we report the vehicle model, the number of images, and the average number of keypoints per image.

Env	**Env1**	**Env1**	**Env1**	**Env2**	**Env2**	**Env3**	**Env3**
Vehicle	Ford-Focus	Fiat-500L	Hyundai-i10	Fiat-500L	Toyota-Yaris	Toyota-Yaris	Hyundai-i10
Images	161	143	123	94	91	116	123
#Kpts	23	14	19	13	20	22	16

making cameras follow a circular trajectory, recovering a plausible pose configuration with a speedup reaching one order of magnitude w.r.t. COLMAP.

3.1.1 Dataset

Enhancing Vehicle Inspection Insights

Benchmark datasets like Tanks and Temples [87], LLFF [113], and the Unbounded 360° dataset [6] have played pivotal roles, serving as standards to evaluate the efficacy of cutting-edge NeRF (Neural Radiance Fields) methods in real-world scenarios. These datasets, widely referenced in literature, have significantly contributed to advancing the understanding and development of computational methods for scene reconstruction and rendering. However, these datasets, while efficient, lack specific information pertinent to vehicle inspection within real-world settings. Recognizing this significant gap, we introduce the KRONC dataset as an innovative solution tailored explicitly to address these challenges.

Dataset captures

The dataset has been collected by employing different devices in three distinct environments. For the first two environments (Env1 and Env2), the scenes have been captured using two standard smartphone cameras (OnePlus 7T and OnePlus Nord). For Env3, we adopted a DJI MINI 2 SE drone for taking pictures. Three different scenes belong to Env1 and two additional scenes come from Env2 and Env3, respectively, leading to a total of 7 scenes. Each scene represents a single vehicle captured from multiple viewpoints. To mimic user behavior in real use cases, we opted for capturing video clips by moving around the vehicle, following a circular path around each car, while maintaining a consistent distance throughout the registration. In each video, a single complete lap around the car has been performed, making the last frame roughly correspond to the first one. Each capture was intended to include the entire car body in the field of view of the camera. Note

that this represents the suggested way of capturing large bounded objects even from well-known 3D reconstruction services[1].

Original videos have been captured with a frame rate ranging from 30 to 60 fps, before being downsampled to 5 fps. Frames have been extracted and sub-sampled again to make data suitable for S*f*M pipelines and novel view synthesis processing. Both the original videos and the selected frames are available to download inside the public dataset for completeness and future fair comparisons.

Dataset metadata

To leverage car keypoints for camera extrinsic optimization (as will be detailed in Sec. 3.1.1), we automatically annotated semantic keypoints on each single frame of the KRONC-dataset, by adopting the OpenPifPaf [88] framework. Specifically, we used the ShufflenetV2K16 model [109] trained to predict the 66 distinct keypoints defined in ApolloCar3D [155]. Moreover, for vehicle inspection purposes, we provide car instance segmentation masks to make it possible to discard unnecessary background pixels. Image-wise mask predictions have been obtained through Mask2Former [21] with a Swin Large [106] backbone trained on the COCO panoptic dataset [85]. Masks isolate the vehicle from complex backgrounds, allowing to focus on vehicle reconstruction in presence of challenging environmental conditions, which however is not the case for the KRONC-dataset. Finally, each dataset underwent rigorous COLMAP [144] processing to estimate precise camera poses. This information can be used as an upper-bound reference for evaluating pose estimation methods, highlighting the remarkable precision achieved by COLMAP, especially when large volumes of images are available. Table 3.1 presents a summary of the scenes included in the KRONC-dataset along with the corresponding number of images per scene and the average number of keypoints detected per image.

Exploiting keypoint projections

In this section, we detail how vehicle semantic keypoints can benefit multi-view consistency and camera pose alignment, as a pre-processing step to improve downstream novel views synthesis algorithms (*e.g.*Neural Radiance Fields [114]).

The input of our algorithm is a set of N captures $\mathcal{I} = \{I_i\}_{i=1}^{N}$ of a scene representing a vehicle. Without loss of generality, we assume that the N images have been taken with the same camera, whose internal calibration parameters are

[1] `https://lumalabs.ai/`

known or have been previously calculated. Therefore, we can define a unique matrix $K \in \mathbb{R}^{3\times3}$ containing the intrinsic parameters, common to all the views.

Let $R_i \in SO(3)$, $\mathbf{t}_i \in \mathbb{R}^3$, be the extrinsic parameters (*i.e.*, rotation matrix and translation vector) of each image I_i with respect to a common world reference system. For images captured with a moving camera, these parameters are generally not available and should be estimated with computationally-intensive procedures such as S*f*M algorithms (*e.g.*COLMAP [143]). KRONC optimizes a noisy/coarse initial approximation of the extrinsic camera parameters. Differently from recent methods exploiting visual pairwise image correspondences [163, 90], we benefit from a much lighter global information shared between (potentially) all the captures, *i.e.*semantic 2D keypoint coordinates.

Projecting keypoints to the 3D world. Let $\{p^1, p^2, ..., p^J\}$ be a set of J semantic keypoints, meaningful for a class of interesting objects (vehicles, in our scenario). Each input image is required to be annotated with the 2D position of these keypoints. The estimation of the 2D keypoint coordinates is a common task in computer vision [155] and the corresponding algorithm remains outside the scope of this work. Therefore, let us define the available set of keypoints as $\mathcal{P} = \{p_i^j\}$, $p_i^j = (u_i^j, v_i^j, m_i^j, z_i^j)$, where (u_i^j, v_i^j) are the 2D coordinates of the j-th keypoint in the i-th image plane, $m_i^j \in [0, 1]$ is the visibility of the keypoint and z_i^j is the distance of the keypoint from the camera center. We introduce m_i^j as a consequence of potential occlusions, since we may observe only a subset of the J keypoints in each image. However, we assume that the number of views N is large enough to guarantee a certain degree of overlap between views, resulting in the same semantic keypoint p^j being visible in multiple captures. The additional z_i^j is required to back-project p_i^j from the 2D image plane to a common 3D world's reference frame XYZ as follows:

$$\begin{bmatrix} X_i^j \\ Y_i^j \\ Z_i^j \end{bmatrix} = \begin{bmatrix} R_i & \mathbf{t}_i \end{bmatrix} \begin{bmatrix} K^{-1} & \\ 0 \quad 0 & 1 \end{bmatrix} \begin{bmatrix} u_i^j \\ v_i^j \\ 1 \end{bmatrix} z_i^j. \tag{3.1}$$

Since both camera parameters R_i, $\mathbf{t}_i$ and keypoint's depth z_i^j in the camera's reference system are unknown or initialized with some noisy values, we need to find a suitable procedure to optimize them. In Sec. 3.1.2 we detail how these parameters are initialized for both synthetic and real vehicle scenes. In the remaining of this section, we describe how we optimize camera poses and keypoints' depths to ensure 2D re-projection consistency between captures.

3D centroids and re-projection consistency

The optimization of the camera poses is based on the following assumption: the 3D back-projections of the same semantic keypoint p^j from different views should lie on the same 3D point. However, if the extrinsic parameters and depths are affected by noise, a cluster of 3D points will be generated for a specific semantic keypoint. We aim to align each back-projected semantic keypoint p^j with its cluster's centroid. Taking into account a specific view, its extrinsic parameters will be optimal when the distances of its back-projected keypoints from the corresponding cluster centers are minimized. The same holds for each keypoint depth z_i^j. In our preliminary experiments, we empirically observed better results and convergence by minimizing the Euclidean distance between each keypoint and its cluster center both after re-projecting them onto each image plane and directly in the 3D space.

3D clusters and centroids re-projection. Formally, let's consider a semantic keypoint p^j at a time. Let M^j be the number of images where the j-th keypoint is visible, *i.e.*$M^j = \sum_i m_i^j$. We independently project all the keypoints p_i^j from these images to the common 3D world reference frame through Eq. 3.1, before computing their 3D centroid C^j as follows:

$$C^j = \begin{bmatrix} X_C^j \\ Y_C^j \\ Z_C^j \end{bmatrix} = \frac{1}{M^j} \sum_i \left(m_i^j \cdot \begin{bmatrix} X_i^j \\ Y_i^j \\ Z_i^j \end{bmatrix} \right). \tag{3.2}$$

The 3D cluster's centroid C^j can be re-projected into each i-th image I_i and compared to the corresponding annotated keypoint (if visible). The coordinates $(u_{C,i}^j, v_{C,i}^j)$ of the re-projected centroid can be computed as follows:

$$\begin{bmatrix} u_{C,i}^j \\ v_{C,i}^j \\ 1 \end{bmatrix} \propto \begin{bmatrix} & 0 \\ K & 0 \\ & 0 \end{bmatrix} \begin{bmatrix} & R_i & \mathbf{t}_i \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} X_C^j \\ Y_C^j \\ Z_C^j \\ 1 \end{bmatrix}. \tag{3.3}$$

For each image and for each visible keypoint, we aim to minimize the following optimization objective:

$$\mathcal{L}_i^j(R_i, \mathbf{t}_i, z_i^j) = \|(X_i^j, Y_i^j, Z_i^j) - (X_C^j, Y_C^j, Z_C^j)\|_2 + \lambda \|(u_i^j, v_i^j) - (u_{C,i}^j, v_{C,i}^j)\|_2 \tag{3.4}$$

where λ balances the magnitude of distances in the 3D world (as meters) and distances on the image plane (as pixels).

Algorithm 1: KRONC algorithm. Note that foreach statements here represent parallel operations in our implementation

Input : Images $\mathcal{I} = \{I_i\}_{i=1}^{N}$,
semantic keypoints $\mathcal{P} = \{p^j\}_{j=1}^{J}$
visibility m_i^j of keypoint p^j on image I_i
noisy R_i, $\mathbf{t}_i$, z_i^j, defining π_i projection,
function f mapping $R_i \in \mathbb{R}^{3\times3}$ to $\mathbf{r}_i \in \mathbb{R}^6$;
Output : Optimized R_i, $\mathbf{t}_i$, z_i^j;
Params : number of steps S,
learning rate η,
2D loss weight λ;

$\mathbf{r}_i = f(R_i)$;
for $s := 1 \rightarrow S$ **do**
  $R_i = f^{-1}(\mathbf{r}_i)$;
  $\mathcal{L} = 0$;
  foreach $p^j \in \mathcal{P},\ j \in \{1, ..., J\}$ **do**
    $C^j = \frac{1}{\sum_{i=1}^{N} m_i^j} \sum_{i=1}^{N} m_i^j \pi_i(p_i^j)$;
    foreach $I_i \in \mathcal{I},\ i \in \{1, ..., N\}$ **do**
      $\mathcal{L} = \mathcal{L} + m_i^j \left(\|\pi_i(p_i^j) - C^j\|_2 + \lambda \|p_i^j - \pi_i^{-1}(C^j)\|_2 \right)$;
    end
  end
  $\mathbf{r}_i = \mathbf{r}_i - \eta \frac{\partial \mathcal{L}}{\partial \mathbf{r}_i},\ i \in \{1, ..., N\}$;
  $\mathbf{t}_i = \mathbf{t}_i - \eta \frac{\partial \mathcal{L}}{\partial \mathbf{t}_i},\ i \in \{1, ..., N\}$;
  $z_i^j = z_i^j - \eta \frac{\partial \mathcal{L}}{\partial z_i^j},\ i \in \{1, ..., N\},\ j \in \{1, ..., J\}$;
end

Full optimization objective. The algorithm seeks to find the global minimum of the following loss, by concurrently optimizing keypoint's projections and back-projection for all the captures $\mathcal{I}$ and for all the semantic keypoints $\mathcal{P}$:

$$\min_{R_i, \mathbf{t}_i, z_i^j} \frac{1}{J} \sum_{j=1}^{J} \sum_{i=1}^{N} m_i^j \mathcal{L}_i^j(R_i, \mathbf{t}_i, z_i^j). \tag{3.5}$$

Although no constraints limit the direct optimization of translation embeddings

$\mathbf{t}_i \in \mathbb{R}^3$ and depth values $z_i^j \in \mathbb{R}$, the same does not hold for rotation matrices, which must preserve orthogonality. Inspired by recent works facing the same issue [163, 74], we adopt the 6D representation of [208], where the unnormalized first two columns of the rotation matrix are employed to represent a full rotation. Specifically, given the noisy rotation matrix for the i-th image $R_i = [\mathbf{a}_1\ \mathbf{a}_2\ \mathbf{a}_3] \in \mathbb{R}^{3\times3}$, we compute the corresponding initial rotation vector $\mathbf{r}_i = [\mathbf{a}_1^T, \mathbf{a}_2^T] \in \mathbb{R}^6$ by simply dropping the last column. At every optimization step, we first recover the full rotation matrix as $R_i = [\mathbf{b}_1\ \mathbf{b}_2\ \mathbf{b}_3] \in \mathbb{R}^{3\times3}$, where $\mathbf{b}_1 = N(\mathbf{a}_1)$, $\mathbf{b}_2 = N(\mathbf{a}_2 - (\mathbf{b}_1 \cdot \mathbf{a}_2)\mathbf{b}_1)$, $\mathbf{b}_3 = \mathbf{b}_1 \times \mathbf{b}_2$, and N denotes L2 normalization. Then, we compute our objective and update $\mathbf{r}_i$, $\mathbf{t}_i$ and z_i^j according to Eq. 3.5.

The optimization is carried out through several iterations. At each iteration, the new positions of the cluster centers are concurrently computed and the parameters are optimized in parallel using gradient descent, leading to almost real-time optimization on the latest GPU devices.

The KRONC algorithm is devised as an easy-to-implement and efficient camera alignment strategy to be executed before novel view synthesis methods. Note that it does not make any use of the raw RGB image values, but only exploits keypoints projections from 2D to 3D and vice versa. It does not jointly optimize for neural 3D representations and camera registration as other methods do [98, 19], allowing for seamless integration with every downstream method requiring accurate camera poses. KRONC is detailed in Algorithm 1.

3.1.2 Experimental evaluation

In this section, we present the experimental settings and the results obtained using KRONC for camera registration, followed by different state-of-the-art downstream novel view synthesis approaches. Performances are evaluated on synthetic and real-world vehicle scenes. In accordance with Barf [98], we apply Procrustes analysis to determine a 3D similarity transformation for aligning the optimized poses with the ground truth, before computing rotation and translation errors ϵ_R and $\epsilon_\mathbf{t}$, respectively. For novel view synthesis evaluation, we adopt common visual quality metrics, *i.e.* PSNR, SSIM [178], and LPIPS [197].

Synthetic vehicle scenes

We use the CarPatch dataset [33] as our benchmark for synthetic 3D vehicle reconstruction evaluation. We adopt the full version containing 8 scenes, each comprising 100 training and 200 test images with ground truth camera poses.

Table 3.2: Quantitative results averaged over the CarPatch scenes. We assign gold, silver, and bronze medals to the best three methods.

Method	**Poses**	$\epsilon_R(^\circ)\downarrow$	$\epsilon_{\mathbf{t}}$ (cm) $\downarrow$	PSNR ↑	SSIM ↑	LPIPS ↓	Runtime
TensoRF [16]	GT	-	-	34.74	0.973	0.043	35 min
DVGO [156]	GT	-	-	36.09	0.979	0.024	10 min
GaussianSplatting [80]	GT	-	-	34.86	0.982	0.014	5 min
Barf [98]	Noisy	7.67	49.38	17.46	0.870	0.142	12 h
L2G-NeRF [19]	Noisy	0.50	5.26	31.91	0.966	0.060	6 h
KRONC + TensoRF [16]	Noisy	0.65	3.06	33.80	0.971	0.042	35.5 min
KRONC + DVGO [156]	Noisy	0.65	3.06	34.03	0.975	0.029	10.5 min
KRONC + GaussianSplatting [80]	Noisy	0.65	3.06	34.38	0.982	0.014	5.5 min

Since KRONC requires the annotation of keypoints, we added them in the original CarPatch 3D Blender models, following the semantic convention defined in [155]. Then, we enriched the CarPatch scenes with ground truth 2D vehicle keypoints via Blender rendering. CarPatch keypoint annotations will be released together with the KRONC-dataset.

Implementation and experimental settings. We parametrize the camera poses with the SE(3) Lie algebra and assume known intrinsics. According to the Lego dataset setting of L2G [19], we synthetically perturb the camera poses creating noisy $R\mathbf{t}$ matrices. Noise values for R and $\mathbf{t}$ are sampled from normal distributions with standard deviation $\sigma_R = 4^\circ$ and $\sigma_{\mathbf{t}} = 0.5$ m, respectively. Similarly to COLMAP, we optimize the test poses together with train poses during camera optimization. This differs from L2G and Barf settings, where they perform test-time photometric pose optimization [99, 191] before evaluating view synthesis quality. Given the different ground truth camera distribution between the test set and the training set, we chose to partition each scene of the CarPatch training set into 80 images for training and 20 for testing. For an early plausible 3D keypoint back-projection (Eq. 3.1), we randomly initialize the z_i^j values from the range $[\frac{1}{2}\omega, \omega]$, where ω is the average L2 norm of the translation vectors of the initial camera poses in the scene. Different initialization methods are explored later in Sec. 3.1.2. As our method is designed to be plug-and-play, we demonstrate its versatility by evaluating the effect of optimized poses on various downstream novel view synthesis methods without modifying their original implementations. When selecting novel view synthesis architectures, we were driven by the best trade-off between training time and reconstruction quality, with the goal of developing a real-time system tailored for vehicle inspection. All the experiments are conducted using a single GeForce GTX 1080 Ti. For consistency, input resolution is fixed

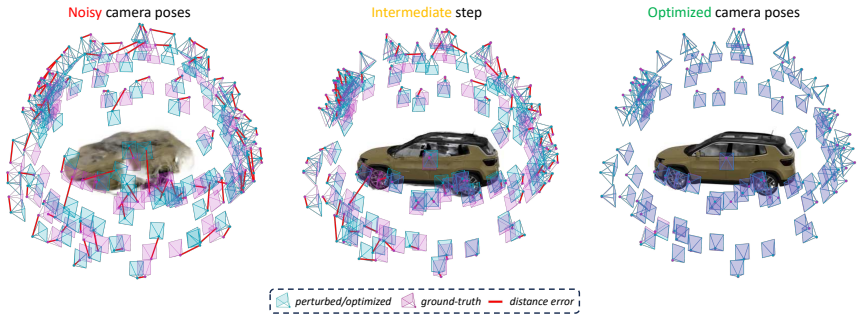


Figure 3.2: Camera arrangement starting from the noisy initialization (left) to the final KRONC prediction (right). Note how cameras align with ground-truth at the end.

to 400×400, as in L2G and Barf experimental settings. After a comprehensive assessment of various methods, we select the following baselines:

- **Gaussian Splatting [116]**: the experiments are conducted without altering the original settings. We train for 10k iterations before rendering test images.
- **TensoRF [16]**: we choose to employ the Nerfstudio [159] implementation for TensoRF. Our configuration involves a batch size of 4096 rays, a scale dimension of 0.5, and an initial learning rate set to 0.0001 with an exponential decay scheduler. Training lasts 10k iterations.
- **DVGO [156]**: this approach comprises a two-phases training process: an initial coarse training spanning 5k iterations, followed by a fine training of 10k iterations, intended to enhance the capability in grasping intricate scene details. We use a batch size of 8192, maintaining the default scene size.

Results. As shown in Table 3.2, KRONC is highly beneficial for 3D reconstruction architectures in synthetic scenarios. In terms of view synthesis quality metrics (PSNR, SSIM, LPIPS), all the selected baselines outperform Barf and L2G when using KRONC optimized poses, almost closing the gap with the visual quality obtained by training on ground truth poses. In terms of camera registration quality, relative rotation error increases by $\sim$30%, while relative translation error decreases by $\sim$42% compared to L2G. Both KRONC and L2G demonstrate

Table 3.3: KRONC results on CarPatch by varying rotation and translation noise.

$\sigma_R(^\circ)$	$\sigma_\mathbf{t}$ (cm)	$\epsilon_R(^\circ)\downarrow$	$\epsilon_\mathbf{t}$ (cm) $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
5	0.7×10^2	0.75	3.07	34.34	0.982	0.014
5	1.5×10^2	1.35	3.10	34.34	0.982	0.014
6	2.0×10^2	3.79	3.16	34.26	0.982	0.014
6	2.5×10^2	2.34	2.13	34.16	0.981	0.014
7	3.0×10^2	6.54	6.21	33.66	0.979	0.017

Table 3.4: KRONC performance on CarPatch using 2D loss, 3D loss, or both.

Loss	$\epsilon_R(^\circ)\downarrow$	$\epsilon_\mathbf{t}$ (cm) $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
2D Loss	0.75	3.10	33.93	**0.981**	0.144
3D Loss	0.68	3.19	33.95	**0.981**	**0.140**
(2D+3D) Loss	**0.65**	**3.06**	**34.48**	**0.981**	**0.140**

superior performance compared to Barf. The overall alignment achieved by KRONC closely approximates the ground truth camera poses, as visually depicted in Fig. 3.2. Moreover, while the additional overhead due to KRONC over the downstream novel view synthesis can be accurately quantified (30 seconds on a single GPU), the cost for camera registration on Barf/L2G can not be exactly assessed, since radiance fields and cameras are optimized together.

Additional analysis. We examine the robustness of our method in synthetic scenarios by introducing varying levels of noise to ground truth camera poses. This was accomplished by altering the normal distribution standard deviation used to randomly sample rotation and translation noise, σ_R and $\sigma_\mathbf{t}$, before adding it to the cameras. As shown in Table 3.3, results do not show significant deterioration even with a 7°, 3 m noise magnitude. Moreover, as explained in KRONC training loss is made up of two different components: one operating on the 2D image plane and the other in the 3D common space. Table 3.4 shows that their combination further improves performance compared to using them individually. All experiments adopt Gaussian Splatting [80].

Real-world vehicle scenes

To assess the performance of our method in the real domain, we use the proposed KRONC-dataset as our benchmark. Semantic keypoints information come from [88].

Implementation and experimental settings. Differently from the synthetic scenario, no ground truth camera poses are available in the KRONC-dataset. In this case, we run the COLMAP algorithm on each scene to retrieve a pseudo-ground truth to be used as our reference. Driven by what usually happens in real contexts and considering a reasonable dimension of the scene, we define a standard 4m radius circular trajectory, placing as many cameras as the number of vehicle images, forward-facing and with no tilt angle. We refer to this trajectory

Table 3.5: Quantitative results on the KRONC dataset. The GaussianSplatting [80] baseline trained with COLMAP poses and an optimized standard trajectory. The results in (·) are computed after masking out the background.

				Full scene(Masked vehicle)		
Env	**Vehicle**	**Init Pose**	**# Opt. Cameras**	PSNR ↑	SSIM ↑	LPIPS ↓
Env1	Ford-Focus	COLMAP	161/161	29.11 (28.37)	0.916 (0.959)	0.089 (0.036)
Env1	Fiat-500L	COLMAP	143/143	26.94 (25.12)	0.892 (0.930)	0.115 (0.061)
Env1	Hyundai-i10	COLMAP	123/123	29.38 (29.45)	0.918 (0.971)	0.097 (0.026)
Env2	Fiat-500L	COLMAP	94/94	28.15 (28.60)	0.922 (0.945)	0.117 (0.048)
Env2	Toyota-Yaris	COLMAP	91/91	28.90 (29.18)	0.936 (0.957)	0.115 (0.029)
Env3	Toyota-Yaris	COLMAP	116/116	31.00 (33.31)	0.948 (0.983)	0.065 (0.017)
Env3	Hyundai-i10	COLMAP	123/123	30.95 (31.11)	0.942 (0.974)	0.072 (0.025)
Env1	Ford-Focus	Trajectory	161/161	21,97 (23,41)	0.696 (0,888)	0.296 (0.081)
Env1	Fiat-500L	Trajectory	124/143	20.52 (21.56)	0.652 (0.835)	0.318 (0.121)
Env1	Hyundai-i10	Trajectory	121/123	21.46 (23.38)	0.666 (0.896)	0.296 (0.070)
Env2	Fiat-500L	Trajectory	67/94	16.94 (19.20)	0.660 (0.769)	0.359 (0.176)
Env2	Toyota-Yaris	Trajectory	90/91	17.68 (21.54)	0.727 (0.836)	0.348 (0.125)
Env3	Toyota-Yaris	Trajectory	116/116	19.06 (21.23)	0.601 (0.850)	0.396 (0.130)
Env3	Hyundai-i10	Trajectory	107/123	18.27 (22.88)	0.582 (0.892)	0.405 (0.091)

as our initial coarse camera configuration (the same for all real scenes), which we optimize using KRONC. We follow the LLFF dataset [113] train/test split protocol sampling one test image every 8 frames for each recording. We select Gaussian Splatting [80] as the 3D reconstruction baseline based on the results obtained on the synthetic scenario. Experiments are conducted with the same configuration described in Sec. 3.1.2, with an image resolution of 480×270.

Results. In Table 3.5, we assess the performance of our algorithm with respect to COLMAP camera registration. As a reference, the maximum PSNR achieved by training Gaussian Splatting with the initial coarse trajectory is 12.0 on the *Ford-Focus* scene. Bundle-adjustment methods (like L2G) are not able to converge in this inward-facing 360° setting with large rotations, as mentioned in their paper [19] and demonstrated by our preliminary experiments (starting from both identity transformation and our circular trajectory). L2G obtains a PSNR lower than 10.0 for all the KRONC-dataset scenes. KRONC is able to find a reasonable camera configuration, reaching a maximum PSNR of 23.41 on the *Ford-Focus* scene (with masked out background). We test the visual quality of the reconstruction using both full images and masked backgrounds with the Gaussian Splatting baseline. The performance drop compared to COLMAP is partly due to the keypoint detector recall, which may leave some viewpoints without

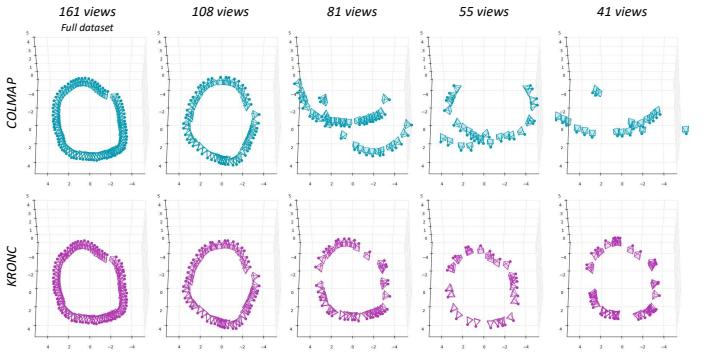


Figure 3.3: Comparison between COLMAP and KRONC for camera pose reconstruction on the KRONC-dataset's *Ford-Focus* using different subsets of the original full scene.

keypoint annotations, causing those poses to remain unadjusted by KRONC. In particular, this can be noted in the Env2 *Fiat-500L* scene, which has only 13 keypoints per image on average (according to Tab. 3.1), leading to almost 30% of the camera viewpoints being discarded in the optimization process. Even if the performance gap is noticeable, KRONC is $\sim$16 times faster than COLMAP, i.e. 30 seconds v.s. 8 minutes on a single GPU for the same number of images. It is worth noting that this comparison does not take into account the inference time needed for OpenPifPaf [88] keypoint extraction, which is 33 seconds on average over the KRONC-dataset scenes. This leads to an effective $\sim 8\times$ speedup.

Table 3.6: KRONC results with varying depth initialization on KRONC-dataset for masked cars (* indicates unoptimized depth).

Depth type	PSNR ↑	SSIM ↑	LPIPS ↓
DinoV2 [124]*	17.54	0.703	0.223
DinoV2 [124]	20.53	0.827	0.149
Random	**21.89**	**0.852**	**0.114**

Additional analysis. For real scenes, the z_i^j depth values are randomly initialized following the same approach described in Sec. 3.1.2 for synthetic scenes. Here we investigate the impact of depth initialization by considering all the KRONC-dataset. As an alternative, we provide results by initializing depths using predictions from DinoV2 [124]. In a preliminary experiment, these depths are not further optimized within the KRONC iterations, and are kept fixed. In a second scenario, we subsequently refine depths during the KRONC optimization process. As shown

in Table 3.6, the random initialization based on the scene scale obtains the best results in all the visual metrics.

Finally, we assess the robustness of the KRONC algorithm in a real-world scenario by sub-sampling the number of images used from the *Ford-Focus* scene within the KRONC-dataset. As illustrated in Fig. 3.3, COLMAP camera pose estimation capability rapidly degrades when reducing the number of images (*i.e.* decreasing image overlap), as already noted in [163]. In contrast, KRONC results demonstrate that by replacing pairwise matches with global reasoning via shared semantic keypoints and by coarsely initializing camera poses using some prior knowledge, robust registration can be achieved even with limited data.

Qualitative results

In Fig. 3.4 we show some qualitative samples obtained with the Gaussian Splatting baseline after KRONC camera optimization in both the synthetic and real scenarios. The proposed method is able to recover a camera configuration to obtain a high-quality reconstruction of the synthetic vehicles. Also in the more challenging real scenario KRONC confirms its robustness finding a consistent sub-optimal camera configuration for a realistic 3D vehicle reconstruction.

Figure 3.4: Qualitative results of KRONC followed by Gaussian Splatting on real scenes (first two rows) and synthetic ones (last three rows). Best viewed in color and zoom.

3.2 Conclusion

We presented both a new dataset and a state-of-the-art algorithm to foster research and applications on the *vehicle inspection* task. The KRONC-dataset represents the first collection of high-quality scenes of real vehicles, while the KRONC algorithm specifically tackles camera optimization using 2D keypoints as a pre-processing step for novel view synthesis. With almost no overhead, KRONC efficiently recovers camera poses, yielding reconstruction results comparable to those obtained with ground truth cameras for synthetic scenes. Similar observations have been demonstrated on the real scenes from the KRONC-dataset, by only assuming an initial circular trajectory of the cameras. Despite the advantages of the KRONC algorithm w.r.t. S*f*M and bundle-adjusting novel view synthesis approaches, it still has some limitations. Its performance on real-world scenes highly depends on the quality of predicted keypoints, when extracted with an automatic detection method, as demonstrated in Sec. 3.1.2. Moreover, it needs at least a rough initialization of the camera poses, being not able to converge to a good solution when starting from random values.

3.2.1 Reproducibility

Upon publication, we will release the complete KRONC-dataset together with detailed instructions for training all the considered novel view synthesis baselines with camera extrinsics estimated by KRONC.

Implementation details. The extrinsic parameters are optimized by disentangling rotation and translation. Since rotation and translation noises have different effects on vehicle visibility, optimizing both parameters in the same way is not trivial. All experiments have been run on a machine with an Intel Core i7-12700F and a NVIDIA GeForce GTX 1080 Ti. With this hardware configuration, the KRONC algorithm runs 10K iterations in 30 seconds on GPU. We use the Adam optimizer with a learning rate of 0.01 for the synthetic data and 0.001 for the real data. We apply a cosine annealing decay with a decay factor of 0.001. Being N the number of views for a scene and J the number of semantic keypoints, we optimize a 6D vector and a 3D vector for rotation and translation for each view. Moreover, a vector of J keypoint depths is optimized, leading to a total of $9N$+JN parameters. Considering a scene with 100 views and 66 keypoints, the KRONC algorithm optimizes only 7.5K parameters, making it suitable even for edge devices.

Camera noise. In all the synthetic scenario experiments, we introduce perturbations to the ground truth camera poses using additive noise. It's noteworthy that

Table 3.7: KRONC performances by sampling a different number of poses from the CarPatch dataset.

# poses	$\epsilon_R(^\circ)\downarrow$	ϵ_t (cm) $\downarrow$	PSNR ↑	SSIM ↑	LPIPS ↓
5	1.72	3.37	20.62	0.890	0.092
10	0.73	3.31	24.86	0.929	0.052
20	1.62	3.24	29.88	0.958	0.028
30	2.24	3.12	31.50	0.967	0.023
40	1.82	3.10	32.59	0.974	0.019
50	1.18	3.07	33.33	0.977	0.017
60	0.93	3.06	33.34	0.977	0.017
70	0.69	3.07	34.15	0.981	0.014

our strategy for adding noise differs from Barf [98], where ground-truth camera poses are perturbed using left multiplication, transforming cameras around the object's center. In this setting, the transformed cameras maintain their orientation toward the object's center, and the distances between the cameras and the object are not largely modified.

In contrast, our approach follows the perturbation strategy proposed by L2G-Nerf [19], which involves perturbing ground-truth camera poses using right multiplication, transforming cameras around themselves. This perturbation affects both camera viewing directions (which may not always face the object's center) and camera positions, consequently altering the distances between the cameras and the object.

Dataset. As described in Section 3, our dataset captures a diverse set of 7 vehicles across 3 distinct environments. Figure 3.8 showcases example captures from each environment, along with keypoint and mask annotations.

3.2.2 Additional quantitative results

The CarPatch [33] dataset provides ground-truth camera pose annotations, which can be thought of as an upper bound for KRONC optimization. In this section, we show additional ablation studies performed on the synthetic data.

KRONC vs state-of-the-art. Table 3.8 comprehensively details the performance of our method compared to the state-of-the-art on each scene of the CarPatch dataset. Our proposed method achieves performance comparable to L2G-NeRF in terms of rotation and translation metrics, while simultaneously establishing state-of-the-art results on PSNR, SSIM, and LPIPS metrics when combined with Gaussian Splatting.

Number of training poses. In Table 3.7, we show KRONC's robustness by

Table 3.8: Quantitative comparison of KRONC + Gaussian Splatting, Barf, and L2G-NeRF. The first two metrics show the results on the camera registration obtained using only the KRONC optimization.

Metric	Method	BMW	TESLA	SMART	MBZ$_1$	MBZ$_2$	FORD	JEEP	VOLVO
$\epsilon_{\mathbf{t}}$(cm) ↓	Barf [98]	53.87	73.68	37.08	76.05	56.70	24.37	13.41	59.89
	L2G-NeRF [19]	5.21	6.34	9.17	3.94	5.58	2.31	3.70	5.84
	KRONC	3.17	2.54	4.26	2.77	2.70	2.54	3.36	3.25
$\epsilon_R(^\circ)$ ↓	Barf [98]	15.38	7.08	5.15	13.60	7.69	2.99	2.27	7.27
	L2G-NeRF [19]	0.59	0.48	0.68	0.35	0.62	0.27	0.32	0.66
	KRONC	0.23	0.62	0.85	0.54	0.82	0.83	0.68	0.65
PSNR ↑	Barf [98]	17.88	13.43	17.51	12.63	14.83	21.08	27.16	15.19
	L2G-NeRF [19]	33.19	33.22	31.55	31.88	32.44	30.19	31.24	31.59
	KRONC + GaussianSplatting [80]	36.31	36.59	36.77	33.05	34.67	31.32	32.57	33.74
SSIM ↑	Barf [98]	0.879	0.827	0.912	0.827	0.844	0.868	0.942	0.858
	L2G-NeRF [19]	0.972	0.976	0.972	0.971	0.927	0.937	0.965	0.966
	KRONC + GaussianSplatting [80]	0.986	0.987	0.988	0.983	0.985	0.961	0.980	0.981
LPIPS ↓	Barf [98]	0.139	0.190	0.092	0.198	0.157	0.130	0.084	0.146
	L2G-NeRF [19]	0.052	0.056	0.043	0.054	0.048	0.098	0.069	0.057
	KRONC + GaussianSplatting [80]	0.012	0.010	0.009	0.011	0.012	0.027	0.015	0.014

Table 3.9: Performance comparison of KRONC + GaussianSplatting and L2G-NeRF on CarPatch dataset with different noise levels.

		KRONC+GaussianSplatting					L2G-NeRF				
$\sigma_R(^\circ)$	$\sigma_{\mathbf{t}}$ (cm)	$\epsilon_R(^\circ)$ ↓	$\epsilon_{\mathbf{t}}$ (cm) ↓	PSNR ↑	SSIM ↑	LPIPS ↓	$\epsilon_R(^\circ)$ ↓	$\epsilon_{\mathbf{t}}$ (cm) ↓	PSNR ↑	SSIM ↑	LPIPS ↓
5	0.7×10^2	0.75	3.07	34.34	0.982	0.014	0.51	6.25	31.64	0.965	0.062
5	1.5×10^2	1.35	3.10	34.34	0.982	0.014	8.19	78.0	18.51	0.876	0.129
6	2.0×10^2	3.79	3.16	34.26	0.982	0.014	14.38	177	15.64	0.844	0.171
6	2.5×10^2	2.34	2.13	34.16	0.981	0.014	24.39	269	12.02	0.793	0.252
7	3.0×10^2	6.54	6.21	33.66	0.979	0.017	31.29	348	11.19	0.778	0.267

varying the number of training views, keeping test views unaltered. Given a number of training views, results are averaged over all the scenes with that specific number of views. Our results showcase the method's capability to refine noisy poses even with limited data, leading to performance gains as the number of cameras increases.

Different noise levels. Our method, combined with Gaussian Splatting, demonstrates superior robustness to noise compared to the L2G-NeRF architecture, as shown in Table 3.9. While our method maintains accurate rotation and translation estimates across all noise levels tested, L2G-NeRF fails to reconstruct camera positions accurately when the translation noise exceeds 70cm.

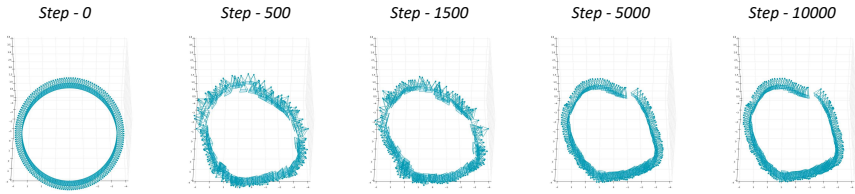


Figure 3.5: Optimization steps in a real scenario. On the left, a visualization of the initial circular trajectory. On the right, the optimized trajectory at each intermediate step.

Additional qualitative results

In this section, we show a qualitative comparison with respect to state-of-the-art approaches in the synthetic scenario. In the real-world scenes, we compare the quality of the reconstruction obtained with coarse or optimized camera trajectories.

KRONC vs state-of-the-art. Figure 3.7 presents a qualitative comparison among various methods utilized for reconstructing vehicles in the CarPatch dataset from noisy camera poses. Barf encounters challenges in accurately reconstructing vehicles, while L2G-NeRF demonstrates greater consistency in this task. Notably, leveraging KRONC alongside Gaussian Splatting (GS) leads to a more precise vehicle reconstruction, effectively capturing intricate details.

Trajectory optimization. Figure 3.5 illustrates the trajectory optimization process for real-world scenarios, as detailed in Section 5.2. The initial trajectory (left) starts as a generic circular path, which is progressively refined in the following iterations to achieve a reliable and reasonable camera registration (right).

Coarse vs optimized trajectory. In Figure 3.6, we present qualitative results illustrating KRONC's capability to reconstruct vehicles in a real-case scenario. Starting from the initialization of cameras, as detailed in 5.2, our method successfully achieves an enhanced vehicle reconstruction. This improvement is evident in both environments, with or without background.

Figure 3.6: Qualitative results of KRONC + Gaussian Splatting on the KRONC-dataset. The second and third columns showcase reconstructions using coarse and optimized trajectories, while the last two columns display reconstructions utilizing masked images.

Figure 3.7: Comparison of qualitative results across all scenes in the CarPatch dataset, showcasing vehicle reconstructions from Barf, L2G-NeRF, and KRONC + Gaussian Splatting.

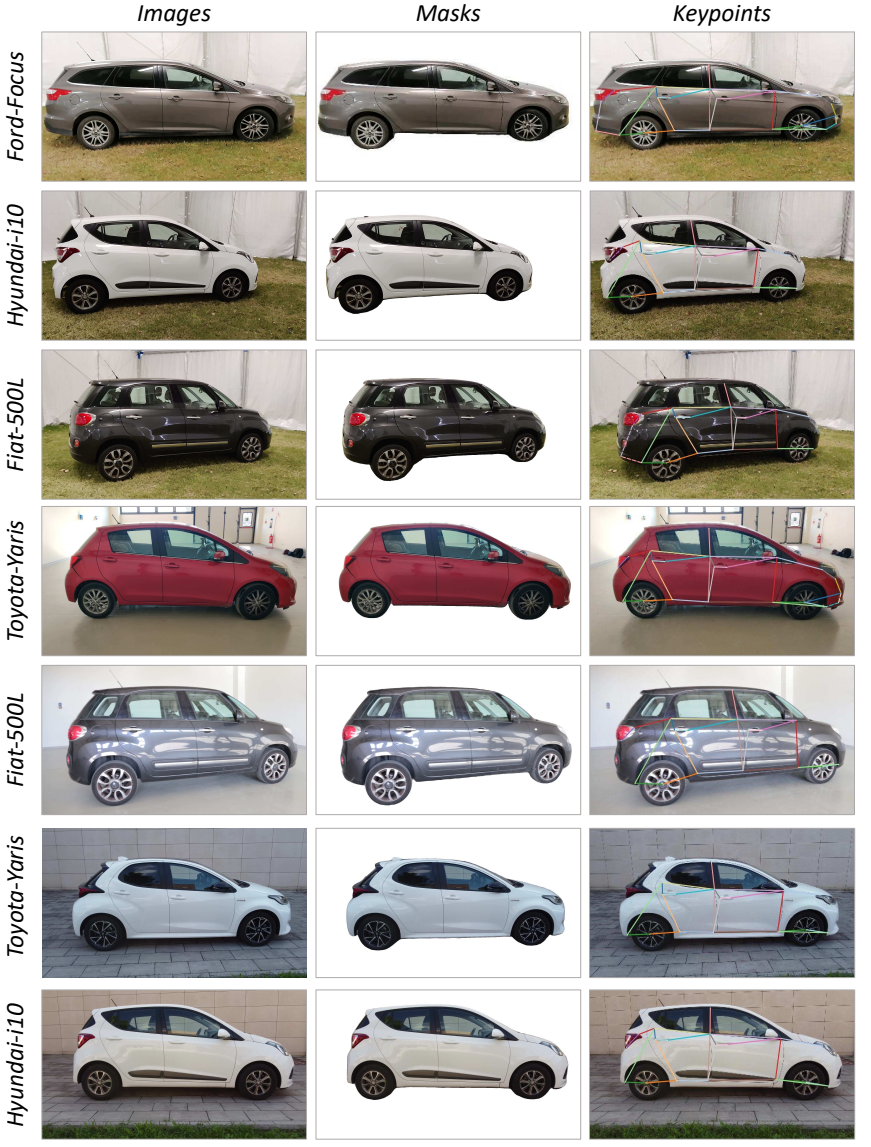


Figure 3.8: Overview of the KRONC-dataset showing the full-scene images, the segmented vehicles, and the predicted keypoints.

Chapter 4

3D Reconstruction

4.1 Introduction

Novel view synthesis and 3D scene representation have been driven by techniques such as Neural Radiance Fields (NeRF) [114] and Gaussian splatting (GS) [80] in the last few years. Both are capable of generating captures of complex scenes from input views with their corresponding camera poses: NeRF leverages an implicit representation through a neural network which models color and density of the scene as a function of the spatial coordinates and viewing directions; GS employs an explicit representation of the scene using a set of 3D Gaussian primitives with associated attributes (*e.g.*, color, opacity).

While the focus of these methods has often been on general (both synthetic and real-world) scenes or human-centered tasks, their application to specific object classes – *e.g.*, the vehicle domain – is still under-investigated. Accurate 3D vehicle reconstruction is essential for several critical applications. One primary motivation is automated *vehicle inspection* [33, 35], where precise 3D models enable detailed analysis of structural integrity, surface condition, and potential defects. In addition, 3D reconstruction facilitates tracking the status of vehicles over time, providing a digital record of wear, modifications, or damage, and supporting predictive maintenance. Autonomous driving is increasingly benefiting from novel view synthesis techniques, too [161, 207].

Beyond individual monitoring, these models can also enhance fleet management systems by providing actionable insights into vehicle health and optimizing operations. Building on these advantages, the public transportation sector stands

to benefit significantly from accurate reconstruction. Furthermore, in the context of smart cities, 3D models of public transportation vehicles can contribute to broader initiatives such as traffic management, urban planning, and environmental monitoring [165, 184, 104].

In practical scenarios, such as tracking a vehicle's condition over time (*e.g.*, daily inspections for buses), capturing images quickly and consistently is crucial. While video recordings enable rapid data collection, they are often constrained by significant storage and processing requirements. Human-operated data collection could be significantly faster and more reliable if it required only a predetermined set of sparse images. Fixed camera setups offer a cost-effective alternative for image acquisition but inherently limit the number of available views to the number of cameras deployed.

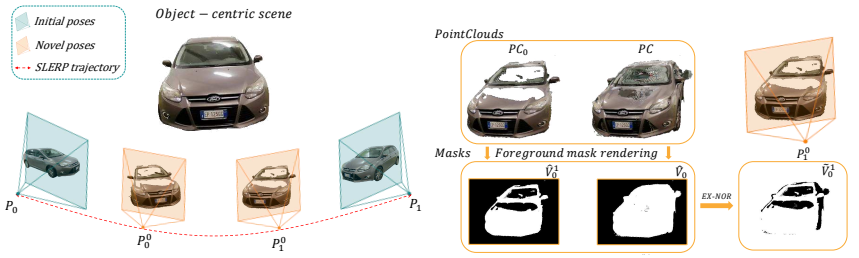
In this work, we tackle the challenge of 3D vehicle reconstruction under the constraint of limited sparse views. Specifically, starting with a handful of scene captures and their associated camera parameters, we leverage ground truth or estimated depth maps to project view-dependent point clouds into synthetic camera poses. These new poses are generated by systematically rotating and translating the original camera positions within a constrained range, effectively augmenting the available views for downstream Gaussian splatting training. Importantly, for these views, the photometric loss is applied exclusively to pixels with high-confidence re-projection.

Camera position estimation for real-world scenes is traditionally performed using standard structure-from-motion (S*f*M) pipelines, such as COLMAP [143]. Moreover, these methods often struggle or fail to converge when provided with only a few images or when image overlap is minimal [35, 163]. To address this limitation, we leverage the recently proposed DUSt3R architecture [176] for estimating both camera poses and an initial point cloud.

We show that our proposed method achieves results comparable or even better than the state of the art in forward-facing 360° vehicle scenes from the Carpatch [33] and KRONC [35] datasets, using as few as 4-8 images, without adding considerable computation time. To assess the effectiveness of our approach in large public transportation vehicle scenes, we also introduce the BRUM-dataset with both synthetic and realistic bus instances, highlighting promising results on it.

To sum up, our key contributions can be outlined as follows:

- We enhance Gaussian splatting robustness in sparse-view forward-facing 360° vehicle scenes. Our method (BRUM) synthesizes novel images from



(a) (b)

Figure 4.1: (a) illustrates an object-centric scene, where ground-truth camera poses are augmented along a SLERP (Sec. 4.2.1) trajectory to generate novel poses. (b) shows how to obtain the final foreground mask $\tilde{V}_0^1$, starting from point clouds and intermediate masks $\hat{V}_0^1$ and $\hat{V}_0$.

coarse point clouds, effectively augmenting the input data.

- We adopt DUSt3R as a replacement for the standard COLMAP and incorporate a masking strategy to exclude low-confidence pixels from the loss computation.
- We present the BRUM-dataset featuring 6 synthetic and 6 real public transportation vehicles. Our method achieves state-of-the-art performance on this dataset and other standard car inspection benchmarks.

4.1.1 The BRUM-dataset

Given the lack of data in literature representing public transportation vehicles, we publicly release the BRUM-dataset. This dataset comprises 12 scenes including 6 real-world 360° captures of buses and 6 synthetic bus models. For both settings, the buses are labeled as bus_1 through bus_6.

Synthetic. The BRUM-dataset consists of six distinct synthetic scenes, each featuring a unique 3D bus model. All the 3D models have been downloaded from Sketchfab[1] and 3DExport[2]. These scenes were generated using Blender[3],

[1] https://sketchfab.com
[2] https://3dexport.com
[3] http://www.blender.org

Figure 4.2: Overview of the BRUM-dataset.

inspired by the Google Blender dataset setup [114]. This approach allowed precise control over lighting conditions and camera viewpoints, providing a proof of concept for evaluating our method under challenging conditions. Each scene was designed with realistic rendering settings: the vehicle was positioned at the origin $(0, 0, 0)$, surrounded by nine lights with varying emission strengths to produce realistic shadows and reflections. Objects were resized to match real-world dimensions, the camera and lighting were configured to mimic real-world environments. Each scene includes 100 training images and 200 test images, accompanied by ground truth depth annotations and camera positions. For training, the camera was randomly positioned on a hemisphere above the ground, with rotation angles sampled uniformly. Instead, test images were captured with the camera at a fixed height, rotating around the Z-axis in steps of $\frac{2\pi}{\#test\_views}$ radians per frame.

Real-World. The BRUM-dataset also comes with six distinct real bus scenes, selected to represent a diverse range of designs and manufacturers. The data

collection was performed in a controlled environment using a DJI MINI 2 SE drone. For each bus, a 360° video was captured by flying multiple laps around the vehicle at varying altitudes. The initial lap was performed at eye level, followed by subsequent laps with the altitude gradually increased by approximately 3 meters, ensuring thorough coverage of each bus's structure. The original videos were recorded at 30 fps with a resolution of 1920×1080. For compatibility with novel view synthesis techniques, individual frames were extracted and sub-sampled. For each bus, a masked version of all frames is provided as mentioned in 4.2.3.

4.2 Method

4.2.1 Augmenting the available training views

The input to our algorithm is a set of N sparse images, denoted as $\mathcal{I} = \{I_i\}_{i=1}^{N}$, capturing an object-centric scene. We assume depth maps for each input image $\mathcal{D} = \{D_i\}_{i=1}^{N}$ being available, together with corresponding extrinsic camera parameters $\mathcal{P} = \{P_i\}_{i=1}^{N}$, with $P_i = [R_i, \mathbf{t}_i]$, where $R_i \in SO(3)$ and $\mathbf{t}_i \in \mathbb{R}^3$ are the rotation matrix and translation vector of camera i, respectively, relative to a common world coordinate system. We also suppose known camera intrinsic parameters, shared among all views.

Sampling novel poses. For the i^{th} camera, we propose to generate M different novel poses starting from its parameters, $P_i = [R_i, \mathbf{t}_i]$, by applying rotations and translations within a limited range. To facilitate this, we first identify the closest camera, indexed as k, among the other $N - 1$ cameras, characterized by its parameters $P_k = [R_k, \mathbf{t}_k]$. The pair of cameras P_i, P_k defines the starting and ending points for interpolation. To smoothly transition between these two poses, we adopt the Spherical Linear Interpolation (SLERP) [150] algorithm. SLERP interpolates between two unit quaternions along the shortest geodesic path on the unit sphere in $\mathbb{R}^4$, ensuring constant-speed rotation transitions. This approach maintains the object-centric nature of the scene while enabling precise and smooth interpolation for generating novel views.

P_i and P_k are first both converted to quaternions $\mathbf{q}_i$ and $\mathbf{q}_k$, then SLERP computes the intermediate quaternion $\mathbf{q}_h$ for interpolation parameter $h \in [0, 1]$ as follows:

$$\mathbf{q}_h = \frac{\sin((1-h)\theta)}{\sin\theta}\mathbf{q}_i + \frac{\sin(h\theta)}{\sin\theta}\mathbf{q}_k, \tag{4.1}$$

where θ is the angle subtended by the arc between $\mathbf{q}_i$ and $\mathbf{q}_k$. Finally, $\mathbf{q}_h$ is

converted back to $[R_h, \mathbf{t}_h]$. In practice, instead of relying solely on the nearest camera, we independently apply SLERP between the i^{th} camera and its two closest cameras. This enables interpolation along two distinct arcs, enhancing the diversity of the generated poses. The number of generated poses can be controlled by sampling various values of h, while constraining h within a specific range of values allows for limiting the deviation from the i^{th} camera. This results in a total of M new poses $\hat{\mathcal{P}}_i = \{\hat{P}_i^j\}_{j=1}^M$ sampled for camera i.

Generating novel views. Starting from poses $\hat{\mathcal{P}}_i$, by leveraging the RGB information in I_i and the depth D_i, we aim to produce synthetic novel views of the scene. Specifically, we first back-project pixels from I_i to 3D using the depth map D_i, obtaining a 3D point cloud PC_i:

$$PC_i = \pi_i^{-1}(I_i, D_i), \tag{4.2}$$

where π_i^{-1} represents the back-projection from the 2D image plane defined by P_i to the common 3D world's reference system. The point cloud PC_i is then warped to the novel views defined by camera poses $\hat{\mathcal{P}}_i$. Given a synthetic camera pose $\hat{P}_i^j \in \hat{\mathcal{P}}_i$, we project PC_i to its image plane as follows:

$$\hat{I}_i^j = \pi_j(PC_i), \tag{4.3}$$

where π_j is the projection from the common 3D world's reference system to the 2D image plane defined by $\hat{P}_i^j$. In practice, while we use the naïve back-projection for π^{-1} as a one-to-one mapping from 2D to 3D, we adopt the solution proposed by [180] for projection π. Specifically, a 3D point p from PC_i is splatted onto a 2D disk with radius r and center p_c, and its influence on a pixel u in $\hat{I}_i^j$ is inversely proportional to the 2D Euclidean distance between u and the disk's center p_c:

$$w(p,u) = \begin{cases} 0 & \text{if } \|p_c - u\|_2 > r \\ 1 - \frac{\|p_c - u\|}{r^2} & \text{otherwise,} \end{cases} \tag{4.4}$$

where $w(p,u)$ represents a weight quantifying how much the 3D point p affects pixel u. As in [180], the projected points are then stored in a z-buffer, sorted by their distance from the new camera pose $\hat{P}_i^j$ and only the K closest points are retained. Finally, alpha over-compositing is adopted for points accumulation. The weighting scheme helps ensure accurate novel views, particularly in cases of overlapping or occluded points, and guarantees smoother renderings.

Binary foreground masks $\hat{\mathcal{V}} = \{\hat{V}_i^j\}_{j=1}^M$ are also gathered, identifying the pixels that have been successfully projected from PC_i to $\hat{I}_i^j$.

4.2.2 Training objective

We can now exploit the original N images $\mathcal{I} = \{I_i\}_{i=1}^{N}$ together with the novel $N \times M$ images $\hat{\mathcal{I}} = \{\{\hat{I}_i^j\}_{j=1}^{M}\}_{i=1}^{N}$ generated from the original ones, for downstream Gaussian splatting training. We recall the original Gaussian splatting objective in the following equation:

$$\mathcal{L} = (1 - \lambda)\mathcal{L}_1(\tilde{I}, I) + \lambda\mathcal{L}_{SSIM}(\tilde{I}, I), \tag{4.5}$$

which is a combination of the L1 and SSIM [178] losses between rendered ($\tilde{I}$) and ground truth images (I).

First, for the 3D Gaussians optimization, we exclude pixels from the generated images that are not accurately projected from the 3D point cloud, as indicated by the foreground masks. However, in object-centric scenes - where objects are segmented, and the background is removed - simply masking out these pixels in the loss can inadvertently exclude background Gaussians from optimization. This happens because background points are absent from the 3D point cloud and are therefore not reported in the foreground masks. As a result, background Gaussians would remain unoptimized, leading to poor 3D reconstruction where the background is only initialized but never refined. To address this, we distinguish between background pixels and those that are actually not reprojected. For each foreground mask in $\hat{\mathcal{V}} = \{\{\hat{V}_i^j\}_{j=1}^{M}\}_{i=1}^{N}$, we compute its *exclusive nor* with the foreground mask obtained by rendering the complete point cloud $PC = \bigcup_{i=1}^{N} PC_i$ from the same camera pose:

$$\tilde{V}_i^j = 1 - (\hat{V}_i^j \oplus \hat{V}_i), \tag{4.6}$$

where $\hat{V}_i$ is the foreground of PC rendered from $\hat{P}_i^j$ and $\oplus$ is the XOR operator. We consider only pixels retained in $\tilde{V}_i^j$ in the final loss. This approach retains background pixels while excluding those not reprojected from PC_i, ensuring a more accurate optimization. More details in Figure 4.1.

Second, we associate a *reliability* measure with each pixel in the generated images. Beyond masking-out pixels that are incorrectly projected from 3D, we apply weights to the remaining pixels using the influence values computed in Eq. 4.4. Intuitively, a pixel largely affected by multiple 3D points is considered more reliable, and its weight in the loss function increases. Formally, for pixel u in $\hat{I}_i^j$, its loss weight becomes:

$$w_K(u) = \sum_{k=1}^{K} w(p_k, u), \tag{4.7}$$

where p_k is the k^{th} point from point cloud PC_i among the K used for alpha over-compositing, as mentioned in Sec. 4.2.1. This leads to weights $\hat{W}_i^j$ for image $\hat{I}_i^j$, after computing the above equation for all the pixels. $\hat{W}_i^j$ is further normalized as:

$$\tilde{W}_i^j = \frac{\hat{W}_i^j - \min(\hat{W}_i^j)}{\max(\hat{W}_i^j) - \min(\hat{W}_i^j)} \tag{4.8}$$

Finally, as the SSIM loss evaluates the structural similarity at the image level rather than on a per-pixel basis, and given that generated images are often less reliable and may lack consistent overall structure, we exclude the SSIM loss from the optimization process for generated images.

Our overall objective for generated images $\hat{\mathcal{I}}$ in Gaussian splatting training is:

$$\mathcal{L} = \begin{cases} (1-\lambda)\mathcal{L}_1(\tilde{I}_i, I_i) + \lambda\mathcal{L}_{SSIM}(\tilde{I}_i, I_i) & ,\forall\, I_i \in \mathcal{I} \\ \dfrac{\sum_u \tilde{V}_i^j(u) \cdot \tilde{W}_i^j(u) \cdot |\tilde{I}_i^j(u) - \hat{I}_i^j(u)|}{\sum_u \tilde{V}_i^j(u)} & ,\forall\, \hat{I}_i^j \in \hat{\mathcal{I}}, \end{cases} \tag{4.9}$$

where the standard Gaussian splatting loss is applied to the original images, while the weighted and masked L1 loss is used for the generated images. Here $\tilde{I}_i$ and $\tilde{I}_i^j$ denote Gaussian splatting renderings from P_i and P_i^j, respectively.

4.2.3 Preprocessing for real-world scenes

For real-word scenes, obtaining camera poses $\mathcal{P}$ and depth maps $\mathcal{D}$ is often challenging due to the limited accessibility of precise sensors and the expense associated with specialized imaging devices. Standard Gaussian splatting and NeRF pipelines usually rely on structure-from-motion as a pre-processing step to estimate both camera parameters and an initial point cloud. However, commonly used S*f*M methods, such as COLMAP [143], often struggle to converge in sparse-view scenarios with a limited number of images.

To address this challenge, we adpot DUSt3R [176] in place of COLMAP. DUSt3R processes image pairs using a shared ViT [37] encoder to extract representations, which are then fed into two Transformer-based decoders [170] equipped with cross-attention layers. These decoders predict two aligned point clouds with associated per-point confidence through regression heads. A subsequent global optimization step allows DUSt3R to determine, for each image I_i, the corresponding camera pose P_i, point cloud PC_i, and scale-aligned depth map D_i (derived from the z-coordinate of the predicted point cloud), as well as a confidence map C_i.

Since our focus is vehicle reconstruction, we further refine the images by leveraging Segment Anything [86] to remove backgrounds. This step generates a set of segmentation masks, $\mathcal{F} = \{F_i\}_{i=1}^{N}$.

The overall pipeline is kept consistent with Sec. 4.2.1 and 4.2.2, except for two modifications:

- Each image I_i is initially filtered using the segmentation mask, yielding $I_i = I_i \odot F_i$.
- The point cloud PC_i, obtained from Eq. 4.2, is refined based on both the segmentation mask and the confidence map, as follows:

$$PC_i = \{p_k \in PC_i \mid F_i(u_k) = 1 \text{ and } C_i(u_k) > c\}, \tag{4.10}$$

  where u_k is the 2D pixel location corresponding to the 3D point p_k in PC_i, and c is a fixed confidence threshold.

Table 4.1: Quantitative results averaged over the BRUM-dataset and CarPatch scenes.

	BRUM-dataset				CarPatch			
Method	PSNR ↑	SSIM ↑	LPIPS ↓	AVGE ↓	PSNR ↑	SSIM ↑	LPIPS ↓	AVGE ↓
3DGS [80]	21.08	0.861	0.132	0.073	22.42	0.867	0.122	0.063
DNGaussians [93]	17.33	0.617	0.187	0.129	21.23	0.824	0.144	0.077
SplatFields [112]	23.80	0.887	0.118	0.055	23.39	0.889	0.116	0.056
BRUM	**24.65**	**0.917**	**0.083**	**0.043**	**25.57**	**0.911**	**0.079**	**0.040**

4.3 Experiments

All experiments were conducted on a TITAN RTX GPU.

Competitor Implementation. We compare BRUM's performance in both synthetic and real-world scenarios against Gaussian Splatting and two state-of-the-art architectures designed for sparse input settings: DNGaussians [93] and SplatFields [112]. To ensure fair comparisons, we introduced minor yet essential modifications to these methods to better align them with the requirements of our use case. Specifically, for SplatFields, the original implementation proposes scaling the initial learning rate by a constant value. However, we found that dynamically adjusting the learning rate based on the characteristics of each training

Table 4.2: Quantitative results averaged over the KRONC and BRUM-dataset scenes.

	KRONC				BRUM-dataset			
Method	PSNR ↑	SSIM ↑	LPIPS ↓	AVGE ↓	PSNR ↑	SSIM ↑	LPIPS ↓	AVGE ↓
3DGS [80]	19.44	0.764	0.153	0.094	19.65	0.734	0.192	0.102
DNGaussians [93]	17.09	0.704	0.243	0.137	17.93	0.701	0.258	0.132
SplatFields [112]	17.81	0.739	0.201	0.119	17.74	0.682	0.207	0.125
BRUM	**20.37**	**0.797**	**0.143**	**0.084**	**20.62**	**0.762**	**0.179**	**0.091**

scene was necessary to achieve optimal performance in our experimental setup. In the case of DNGaussians, instead of optimizing parameters individually for each scene, we employed the default settings provided for the LEGO scene in the Blender dataset. This approach ensured the reproducibility of the results over different scenarios.

Evaluation Metrics. We evaluate our method using standard visual quality metrics: Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM) [178], and Learned Perceptual Image Patch Similarity (LPIPS) [197], which assess reconstruction fidelity, structural consistency, and perceptual quality, respectively. To provide a more intuitive and comprehensive comparison, we also report the Average Error (AVGE) [120]. This metric integrates multiple aspects of visual quality by combining $MSE = 10^{-\text{PSNR}/10}$, $\sqrt{1 - \text{SSIM}}$, and LPIPS as a geometric mean, capturing both pixel-wise and perceptual errors in a unified score.

Rendering. For the rendering process, we use the `PointsRenderer` module from PyTorch3D. The number of points contributing to a pixel's color (*i.e.* the K parameter introduced in Sec. 4.2.1) is set to 16 for both synthetic and real-world datasets to ensure smooth and consistent outputs. The r parameter (Eq. 4.4), which controls the projected size of points, is assigned a value of 0.003 for synthetic data and 0.1 for real-world data, allowing for an optimal balance between preserving fine details and minimizing overlapping effects. These parameter configurations facilitate high-quality renderings with minimal visual artifacts.

4.3.1 Results on synthetic scenes

Implementation Details For synthetic evaluation, we use the BRUM-dataset and the CarPatch [33] dataset as benchmarks to assess 3D reconstruction performance. To simulate a sparse input scenario during training, we sampled 4 images from

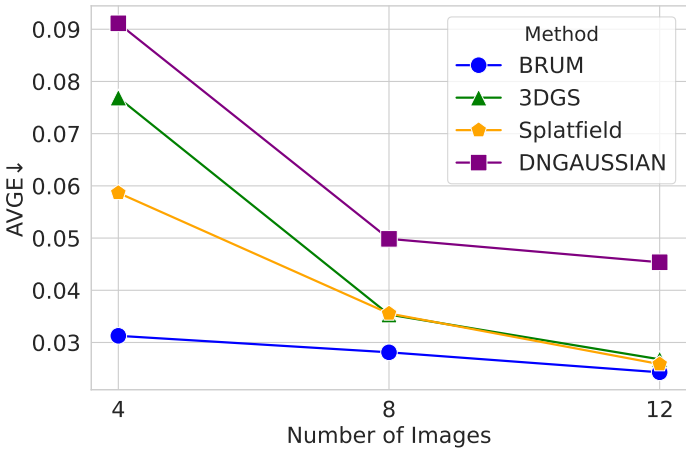


Figure 4.3: AVGE results by varying number of training images.

the training dataset. All images have a resolution of 800×800 and are chosen from distinct viewpoints around the vehicles to ensure comprehensive visibility of the object's structure. The h factor of SLERP (Eq. 4.1) varies from 0.025 to 0.975 (included) with a step size of 0.025. This results in a total of 156 novel views for a scene with only 4 images. Ground-truth (GT) camera poses and GT depth maps are employed during the image augmentation phase (Sec. 4.2.1) to enhance rendering accuracy. To ensure fair comparison, SplatFields is evaluated using GT camera poses, while DNGaussians employs also GT depth maps directly, bypassing the DPT [134] computation suggested in the original work. Given that all these methods utilize the 3DGS codebase, we conducted experiments in the synthetic scenario by initializing the point cloud for 3DGS with random values. This approach ensures a fair comparison of their performance under identical initial conditions.

Main Results. As demonstrated in Table 3.2, BRUM exhibits significant advantages in the context of novel view synthesis in challenging sparse-view scenarios. The results clearly demonstrate the effectiveness of BRUM compared to other state-of-the-art methods. For the BRUM-dataset and CarPatch datasets, BRUM consistently improves all the considered metrics. These results highlight BRUM's ability to preserve perceptual quality and geometric accuracy, while demonstrating scalability and generalization across diverse scenes. Notably, 3DGS, while competitive in some cases, falls short in preserving perceptual details (higher LPIPS)

and maintaining geometric accuracy (higher AVGE) compared to BRUM.

Additional Analysis. Figure 4.3 compares the average error (AVGE) of BRUM with other approaches under varying numbers of input views. The results demonstrate that our method consistently outperforms the others, particularly in sparse view scenarios with 4 or 8 input images, highlighting its capability to reconstruct 3D scenes effectively even in under-constrained conditions. As the number of input images increases to 12, the AVGE converges with that of SplatField and 3DGS, showcasing its robustness and scalability to denser input settings. In terms of runtime, our approach introduces an extra preprocessing overhead ranging from 25 seconds to 10 minutes, depending on the number of rendered images. For sampling and rendering novel views, it does not affect memory usage, which remains consistent. BRUM strikes a balance by achieving superior reconstruction quality compared to 3DGS while maintaining a reasonable computation time.

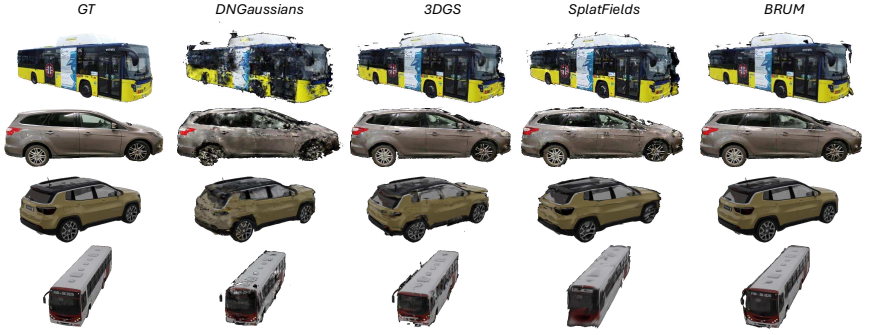


Figure 4.4: Qualitative comparison of 3D reconstruction methods for DNGaussian, 3DGS, SplatFields, and BRUM.

4.3.2 Results on real-world scenes

Implementation details. For the real-world scenario, we evaluate our method on the BRUM-dataset and KRONC-dataset [35]. Unlike the synthetic case, real-world datasets lack ground-truth camera poses and depth maps. We adopt DUSt3R as a preprocessing step to estimate these parameters, as mentioned in section 4.2.3. However, the uncertainties in this estimation process pose challenges for accurate 3D reconstruction. We relax the 4-view constraint and train with 8 views to compensate for inaccuracies in camera pose and depth estimation.

Table 4.3: Quantitative results for different interpolation factors, computed on the Ford scene of the KRONC dataset.

h	PSNR ↑	SSIM ↑	LPIPS ↓	AVGE ↓
0.05	21.01	0.824	0.125	0.075
0.1	**21.18**	**0.827**	**0.114**	**0.071**
0.3	21.13	0.827	0.120	0.073
0.5	20.95	0.823	0.119	0.074

To minimize rendering distortions caused by imprecisions in the depth maps estimated by DUSt3R, the h factor in SLERP (Eq. 4.1) for real-world scenes is restricted to a narrower range compared to synthetic scenes, while maintaining a fixed step size of 0.025. Specifically, for the KRONC-dataset h is fixed at 0.1 for all experimentis while for the BRUM-dataset is fixed to 0.08. During the preprocessing step (Sec. 4.2.3), we examine the influence of DUSt3R's confidence parameter c. For the KRONC-dataset, the optimal c is found to be 0, allowing all points predicted by DUSt3R to be utilized. In contrast, for the BRUM-dataset, the optimal c is determined to be 1.5. In the following sections, we analyze how varying h and c influence reconstruction metrics. To meet DUSt3R's requirements, all input images were downsampled to 512×256.

Main Results. As shown in Table 4.2, BRUM exhibits strong performance in real-world scenarios, though slightly less pronounced than in the synthetic setting (Table 4.1). Real-world datasets present additional challenges, including noise, reflections, and inaccuracies in estimated depth maps and camera poses. Despite these obstacles, BRUM achieves the highest scores across both datasets. Notably, neither DNGaussians nor SplatFields were originally designed to handle sparse 360° forward-facing real scenes in their formulations. This explains their lower performance highlighting the difficulty of our task.

Additional analysis. The results in Table 4.3 provide valuable insights into the effect of the maximum interpolation factor h on BRUM's performance in real-world scenarios. Here, h represents the upper limit for interpolation, while the step size remains fixed at 0.025. Higher values of h indicate moving farther away from the ground truth camera position, resulting in the generation of more synthetic views. For smaller values of h, such as 0.05 and 0.1, BRUM achieves superior preservation of image quality compared to the ground truth. However, as h increases (*e.g.*, 0.3 and 0.5), AVGE experiences a marginal increase. This suggests that larger h values may amplify distortions caused by inaccuracies in depth map and camera pose estimations.

Table 4.4: Performance of BRUM evaluated with varying c (Eq. 4.10) on the bus_1 scene of the BRUM-dataset.

c	PSNR ↑	SSIM ↑	LPIPS ↓	AVGE ↓
0	17.82	0.754	0.200	0.118
0.5	17.81	0.755	0.199	0.118
1	17.67	0.752	0.201	0.120
1.5	**18.41**	**0.789**	**0.167**	**0.103**
2	18.13	0.787	0.169	0.106
2.5	18.18	0.785	0.170	0.106

Table 4.5: Quantitative results for various configurations. The best-performing corresponds to BRUM, as described in Sec. 4.2.1, where the SSIM loss is excluded during training.

$\mathcal{L}_{SSIM}$	XNOR	$\hat{W}$	PSNR ↑	SSIM ↑	LPIPS ↓	AVGE ↓
✓	✗	✓	20.07	0.796	0.148	0.087
✓	✓	✗	20.46	0.798	0.131	0.081
✓	✓	✓	20.95	0.808	0.128	0.077
✗	✓	✓	**21.18**	**0.827**	**0.116**	**0.072**

Table 4.4 examines the impact of c (Eq. 4.10) on reconstruction performance using the bus_1 scene from the BRUM-dataset. Higher c values remove more points from the DUSt3R point cloud, leaving larger image portions unprojected. Lower c values (*e.g.*, 0 or 0.5) fail to filter depth estimation uncertainties, while higher values (*e.g.*, 2 or 2.5) overly prune the point cloud, causing information loss. A value of $c = 1.5$ provides the best balance, improving reconstruction accuracy.

Finally, Table 4.5 evaluates the impact of BRUM's key components, as outlined in Section 4.2.1. The most significant degradation occurs when the XNOR operation from Eq. 4.6 is removed, and the original foreground mask $\hat{V}$ is used. Additionally, omitting the weighting factor $\tilde{W}$ or reintroducing $\mathcal{L}_{SSIM}$ for generated images in Eq. 4.9 results in noticeable performance drops. The complete method achieves the best metrics, validating the importance of each component.

4.3.3 Qualitative results

In Fig. 4.4, we compare qualitative results on the BRUM, KRONC, and CarPatch datasets. The proposed method accurately reconstructs vehicles in both synthetic and real domains, preserving fine details and producing outputs closer to ground truth data.

4.4 Conclusion

This section addresses the challenge of 3D vehicle reconstruction from sparse views. By leveraging depth maps and the DUSt3R architecture, BRUM significantly enhances the robustness of Gaussian Splatting in scenarios with limited input data. We introduce a novel dataset combining synthetic and real-world public transport vehicles, achieving high-quality reconstructions across challenging scenes. The scalability and adaptability of our approach make it well-suited for practical deployment in resource-constrained environments.

Chapter 5

Synthetic Data Generation

5.1 The *CarPatch* dataset

In this chapter, we detail the source data and the procedure exploited for generating our *CarPatch* dataset. In particular, we describe how we gathered 3D models, set up the blender scenes, and designed the image capture process.

5.1.1 Synthetic 3D models and scene setup

All the 3D models included in *CarPatch* scenes have been downloaded from Sketchfab[1], a large collection of free 3D objects for research use. Table 5.1 provides a detailed list of all the starting models used. Each of them has been edited in Blender to enhance its realism; specifically, we improved the materials, colors, and lighting in each scene to create a more challenging environment.

The scenes have been set up accordingly to the Google Blender dataset [114]. The lighting conditions and rendering settings were customized to create a more realistic environment. The vehicle was placed at the center of the scene at position (0,0,0), with nine lights distributed around the car and varying emission strengths to create shadows and enhance reflections on the materials' surfaces. To improve realism, we resized objects to match their real-world size. The camera and lights were placed in order to provide an accurate representation of the environment, making the scenes similar to real-world scenarios.

[1] `https://sketchfab.com`

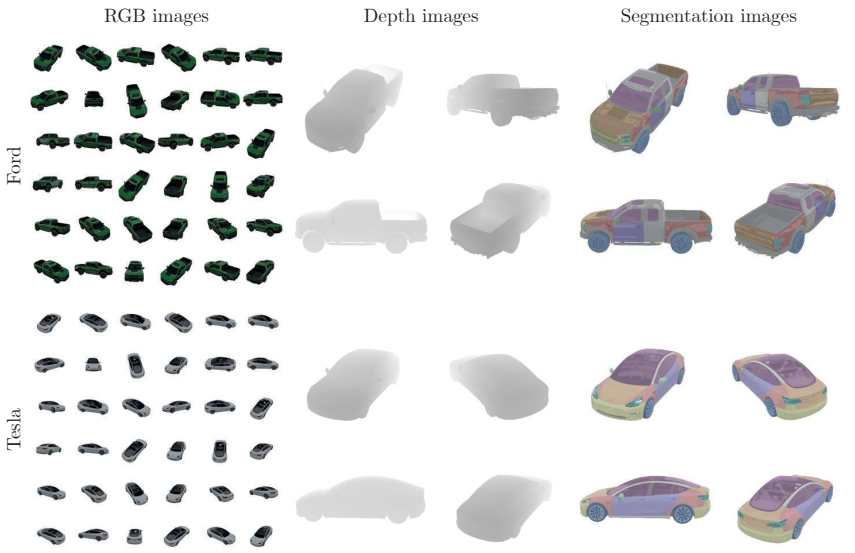


Figure 5.1: Sample RGB images (left), depth data (center), and segmentation masks (right) from *CarPatch*, for different car models.

5.1.2 Dataset building

The dataset was built using the Python interface provided in Blender, allowing us to control objects in the environment. For each rendered image, we captured not only the RGB color values but also the corresponding depth map, as well as the pixel-wise semantic segmentation masks for eight vehicle components: bumpers, lights, mirrors, hoods/trunks, fenders, doors, wheels, and windows. Examples of these segmentation masks can be seen in Fig. 5.1. Please note that all the pixels belonging to a component (*e.g.* doors) are grouped into the same class, regardless of the specific component location (*e.g.* front/rear/right/left door). The bpycv[2] utility has been used for collecting additional metadata, enabling us to evaluate NeRF models on the RGB reconstruction and depth estimation of the overall vehicle as well as each of its subparts.

For the rendering of training images, the camera randomly moved on the hemisphere centered in (0,0,0) and above the ground. The camera rotation angle

[2] `https://github.com/DIYer22/bpycv`

Table 5.1: Summary of the source 3D models from which our dataset has been generated, including their key features.

Model name	Acronym	#Triangles	#Vertices	#Textures	#Materials
Tesla Model	TESLA	684.3k	364.4k	22	58
Smart	SMART	42.8k	26.4k	0	31
Ford Raptor	FORD	257.1k	156.5k	12	50
BMW M3 E46	BMW	846.9k	442.4k	7	39
Mercedes GLK	MBZ_1	1.3M	741.4k	0	15
Mercedes CLS	MBZ_2	1.0M	667k	0	18
Volvo S90	VOLVO	3.3M	1.7M	56	44
Jeep Compass	JEEP	334.7k	189.6k	7	39

was sampled from a uniform distribution before each new capture. For building the test set, the position of the camera was kept at a fixed distance from the ground and rotated around the Z-axis with a fixed angle equal to $\frac{2\pi}{\#test\_views}$ radians before each new capture.

In order to guarantee the fairness of the current and future comparisons, we explicitly provide four different versions of each scene, by varying the number of training images (40, 60, 80, and 100 images, respectively). Different versions of the same scene have no overlap in training camera poses, while the test set is always the same and contains 200 images for each scene.
We release the code for dataset creation and metrics evaluation at `https://github.com/davidedinuc/carpatch`.

To sum up, our contributions encompass the following:

- We introduce *CarPatch*, a synthetic dataset tailored to radiance-field evaluation on vehicle components, providing aligned RGB images, depth maps, and pixel-wise semantic masks for eight meaningful car parts.

- We design a reproducible Blender-based data generation pipeline that controls scene layout, lighting, and camera trajectories, and we release the code for dataset creation and metrics evaluation to support fair and repeatable comparisons.

- We establish a benchmark on *CarPatch* by evaluating representative NeRF-based methods under a consistent experimental protocol, reporting results both at vehicle level and at component level to reflect the requirements of

vehicle inspection.

- Beyond appearance metrics (PSNR, SSIM, LPIPS), we propose depth-driven evaluation through D-RMSE and SN-RMSE to better quantify the quality of the reconstructed 3D geometry, and we analyze performance trends w.r.t. the number and distribution of training views.

5.2 Benchmark

This section presents the selection and testing of various recent NeRF-based methods [116, 16, 156] on the presented *CarPatch* dataset, with a detailed description of the experimental setting for each baseline. Additionally, we assess the quality of the reconstructed vehicles in terms of their appearance and 3D surface reconstruction, utilizing depth maps generated during volume rendering.

5.2.1 Compared methods

To overcome challenges related to illumination and reflective surfaces during the process of reconstructing vehicles, it is crucial to choose an appropriate neural rendering approach. We tested selected approaches on *CarPatch* without modifying the implementation details available in the original repositories, whenever possible. However, some parameters had to be adjusted in order to fit our models (which are larger compared to reference dataset meshes) to the scene. All tests were performed on a GeForce GTX 1080 Ti. After considering various NeRF systems, we have selected the following baselines:

- **Instant-NGP [116].** Since the original implementation of Instant-NGP is in CUDA, we decided to use an available PyTorch implementation[3] of this approach in order to have a fair comparison with the other approaches. In our experiments, a batch size of 8192 was maintained, with a scene scale of 0.5 and a total of 30,000 iteration steps.

- **TensoRF [16].** In our setting, a batch of 4096 rays was used. Additionally, we increased the overall scale of the scene from 1 to 3.5. These adjustments were made after experimentation and careful consideration of the resulting reconstructions. Training lasts 30,000 iterations.

[3] `https://github.com/kwea123/ngp_pl`

Table 5.2: Quantitative results on the *CarPatch* test set for each vehicle model.

Method	Metric	BMW	TESLA	SMART	MBZ_1	MBZ_2	FORD	JEEP	VOLVO	*Avg*
iNGP [116]		39.48	39.46	39.57	36.87	39.15	33.67	35.00	35.93	37.39
DVGO [156]	PSNR↑	39.91	39.89	40.34	37.45	39.37	33.82	35.32	36.28	37.80
TensoRF [16]		40.68	39.92	40.38	38.07	40.84	34.33	34.87	36.77	**38.23**
iNGP [116]		0.985	0.987	0.988	0.985	0.987	0.959	0.978	0.979	0.981
DVGO [156]	SSIM↑	0.987	0.988	0.990	0.987	0.988	0.964	0.980	0.981	0.983
TensoRF [16]		0.989	0.987	0.99	0.989	0.991	0.966	0.975	0.982	**0.984**
iNGP [116]		0.029	0.029	0.02	0.028	0.024	0.062	0.036	0.032	0.032
DVGO [156]	LPIPS↓	0.022	0.022	0.014	0.019	0.020	0.051	0.029	0.022	**0.025**
TensoRF [16]		0.023	0.026	0.017	0.02	0.017	0.051	0.039	0.027	0.028
iNGP [116]		0.640	0.369	0.377	0.496	0.500	0.406	0.558	0.674	0.503
DVGO [156]	D-RMSE↓	0.561	0.353	0.305	0.437	0.454	0.339	0.469	0.561	**0.435**
TensoRF [16]		0.590	0.357	0.335	0.467	0.482	0.375	0.536	0.626	0.471
iNGP [116]		4.24	3.38	3.41	4.26	4.13	5.15	4.60	4.67	4.23
DVGO [156]	SN-RMSE↓	4.27	3.48	3.20	4.19	4.24	5.04	4.67	4.71	4.22
TensoRF [16]		3.96	3.24	3.10	4.00	3.91	4.91	4.41	4.48	**4.00**

- **DVGO [156].** In this work, the training process consists of two phases: a coarse training phase of 5,000 iterations, followed by a fine training phase of 20,000 iterations that aims to improve the model's ability to learn intricate details of the scene. In our experiments, we applied a batch size of 8192 while maintaining the default scene size.

5.2.2 Metrics

The effectiveness of the chosen methods has been assessed thanks to the typical perceptual metrics used in NeRF-based reconstruction tasks, namely PSNR, SSIM [178], and LPIPS [197].

However, the appearance-based metrics are strongly related to the emitted radiance besides the learned volume density. We suggest two supplementary depth-based metrics for the sole purpose of assessing the volume density. Since it is not feasible to obtain ground truth 3D models of the vehicles in real-world scenarios, we utilize the depth map as our knowledge of the 3D surface of the objects. Specifically, we define a depth map as a matrix

$$D = \{d_{ij}\}, d_{ij} \in [0, R] \tag{5.1}$$

Table 5.3: Quantitative results on the *CarPatch* test set for each vehicle component averaged over the vehicle models.

Method	Component	PSNR↑	SSIM↑	LPIPS↓	D-RMSE↓	SN-RMSE↓
iNGP [116]		33.05	0.986	0.019	0.281	0.79
DVGO [156]	*bumper*	34.41	0.989	0.011	**0.236**	0.72
TensoRF [16]		**35.49**	**0.991**	**0.010**	0.311	**0.68**
iNGP [116]		28.71	0.993	0.009	0.421	0.48
DVGO [156]	*light*	29.10	0.995	**0.006**	**0.384**	0.43
TensoRF [16]		**29.68**	**0.996**	**0.006**	0.438	**0.38**
iNGP [116]		29.60	0.994	0.011	0.427	0.43
DVGO [156]	*mirror*	31.16	**0.996**	**0.007**	**0.345**	**0.38**
TensoRF [16]		**31.68**	**0.996**	0.008	0.372	0.39
iNGP [116]		32.28	0.977	0.052	0.260	1.33
DVGO [156]	*hood/trunk*	32.68	0.981	**0.038**	**0.259**	1.35
TensoRF [16]		**33.75**	**0.983**	0.040	0.302	**1.24**
iNGP [116]		32.44	0.990	0.021	0.253	0.87
DVGO [156]	*fender*	33.55	**0.993**	**0.013**	**0.223**	0.85
TensoRF [16]		**34.36**	**0.993**	0.015	0.267	**0.77**
iNGP [116]		34.19	0.969	0.079	0.182	0.67
DVGO [156]	*door*	35.48	0.977	**0.042**	**0.173**	0.74
TensoRF [16]		**36.25**	**0.979**	0.051	0.191	**0.62**
iNGP [116]		33.12	0.995	0.008	0.391	0.87
DVGO [156]	*wheel*	33.65	0.995	0.006	**0.267**	**0.79**
TensoRF [16]		**34.55**	**0.996**	**0.005**	0.334	**0.79**
iNGP [116]		26.44	0.897	0.166	0.879	2.52
DVGO [156]	*window*	26.54	**0.899**	**0.147**	**0.779**	2.57
TensoRF [16]		**26.74**	0.896	0.160	0.834	**2.38**

in which each value d_{ij} ranges from 0 to the maximum depth value R. Furthermore, we estimate the surface normals from the depth maps [132]. Initially, we establish the orientation of a surface normal as:

$$\mathbf{d} = \langle d_x, d_y, d_z \rangle = \left(-\frac{\partial d_{ij}}{\partial i}, -\frac{\partial d_{ij}}{\partial j}, 1 \right) \approx \left(d_{(i+1)j} - d_{ij}, d_{i(j+1)} - d_{ij}, 1 \right) \tag{5.2}$$

where the first two elements represent the depth gradients in the i and j directions, respectively. Afterward, we normalize the normal vector to obtain a unit-length

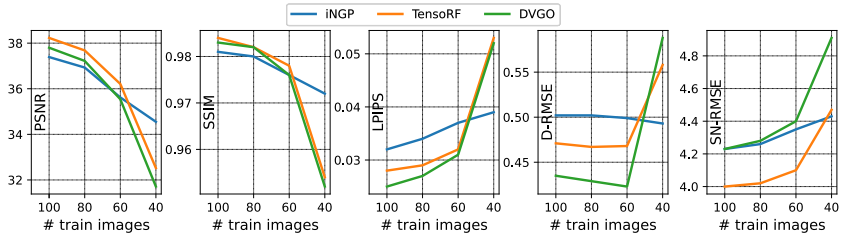


Figure 5.2: Performance by varying the number of training images, in terms of PSNR, SSIM, LPIPS, D-RMSE, and SN-RMSE. Despite its lower overall performance, Instant-NGP [116] exhibits low variance with respect to the amount of training data.

vector $\mathbf{n}(d_{ij}) = \frac{\mathbf{d}}{\|\mathbf{d}\|}$.

We assess the 3D reconstruction's quality through the following metrics:

- **Depth Root Mean Squared Error (D-RMSE).** This metric measures the average difference in meters between the ground truth and predicted depth maps.

$$\text{D-RMSE} = \sqrt{\frac{\sum_{i=0}^{M}\sum_{j=0}^{N}(\hat{d}_{ij} - d_{ij})^2}{M \cdot N}} \tag{5.3}$$

- **Surface Normal Root Mean Squared Error (SN-RMSE).** This metric measures the average angular error in degrees between the angle direction of the ground truth and predicted surface normals.

$$\text{SN-RMSE} = \sqrt{\frac{\sum_{i=0}^{M}\sum_{j=0}^{N}(\arccos\left(\mathbf{n}(\hat{d}_{ij})\right) - \arccos(\mathbf{n}(d_{ij})))^2}{M \cdot N}} \tag{5.4}$$

D-RMSE and SN-RMSE are computed only for those pixels with a positive depth value in both GT and predicted depth maps. This avoids computing depth estimation errors on background pixels (which have a fixed depth value of 0).

5.2.3 Results

The following section presents both quantitative and qualitative results obtained from the selected NeRF baselines. We will discuss their performance on the

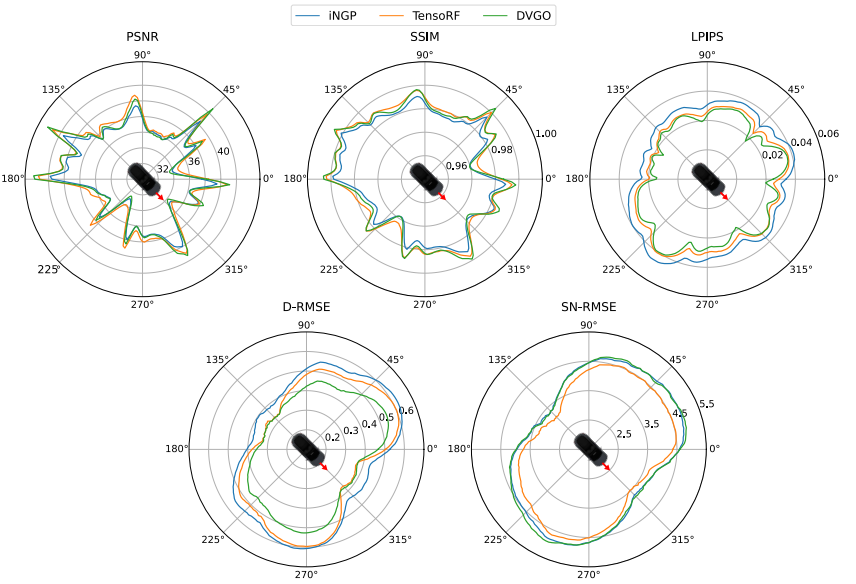


Figure 5.3: Performance by camera viewing angle, in terms of PSNR, SSIM, LPIPS, D-RMSE, and SN-RMSE. Depending on the training camera distribution, all the methods struggle wherever the viewpoints are more sparse (*e.g.* between $225°$ and $270°$). The red arrow represents where the front of the vehicle is facing.

CarPatch dataset, by analyzing the impact of viewing camera angle and the number of training images.

According to Table 5.2, all the selected NeRF approaches obtain satisfying results. Although the baselines demonstrate similar performances in terms of appearance scores (PSNR, SSIM, and LPIPS), our evaluation using depth-based metrics (D-RMSE and SN-RMSE) reveals significant differences in the 3D reconstruction of the vehicles. DVGO outperforms its competitors by achieving better depth estimation, resulting in a $+13.5\%$ improvement compared to iNGP and a $+7.6\%$ improvement compared to TensoRF. In contrast, TensoRF predicts a more accurate 3D surface with the lowest angular error on the surface normals.

Since our use case is related to vehicle inspection, in Table 5.3 we report results computed on each car component. For this purpose, we mask both GT and predictions using a specific component mask before computing the metrics. How-

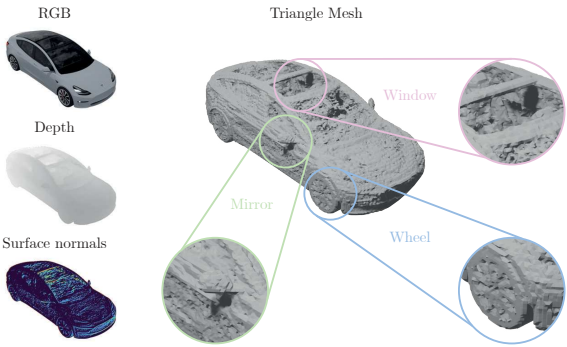


Figure 5.4: Sample of 3D reconstruction of the TESLA: (left) the reconstructed RGB, depth, and surface normals, (right) the reconstructed surfaces on the triangle mesh.

ever, this would lead to an unbalanced ratio between background and foreground pixels, due to the limited components' area, and finally to a biased metric value. By computing D-RMSE and SN-RMSE only on foreground pixels (see Sec. 5.2.2), depth-based metrics are not affected by this issue. For PSNR, SSIM, and LPIPS, instead, we compute component-level metrics over the image crop delimited by the bounding boxes around each mask. As expected, it is worth noting that NeRF struggles to reconstruct transparent objects (*e.g.* mirrors, lights, and windows) obtaining the highest errors in terms of depth and normal estimation. However, over the single components, TensoRF outperforms the competitors in most of the metrics and in particular on the surface normal estimation. The errors in the reconstruction of specific components' surfaces can also be appreciated in the qualitative results of Fig. 5.4.

Fig. 5.2 shows baseline results on the *CarPatch* dataset. While reducing training images degrades all metrics across methods, Instant-NGP proves more robust, exhibiting a smoother decline in LPIPS, D-RMSE, and SN-RMSE. Furthermore, uneven viewpoint distribution impacts performance (Fig. 5.3). We observe considerable metric variations between $180°$–$270°$ and $0°$–$45°$, where the dataset suffers from viewpoint sparsity, negatively affecting all methods.

Chapter 6

3D representations of human pose & gaze estimation

6.1 Introduction

The importance of 3D gaze estimation lies in its ability to unlock deeper insights into human attention [135], and cognition [39], which are central to a wide range of applications such as human-computer interaction [147], behavioral analysis [81], and extended reality systems [152], surveillance [172], autonomous driving [126], and robotics [127].

Computer vision researchers have traditionally approached automated gaze analysis by dividing it into two main tasks [162]: *gaze estimation* and *gaze target detection*, also referred to as *gaze following*. Specifically, gaze estimation aims to predict the direction of a person's gaze, while gaze target detection aims to pinpoint the exact location a person is looking at within the scene.

Methods for 3D gaze estimation are often based on the availability or extraction of detailed information about the human face or upper body [22, 47, 55, 78], ranging from the positions of the pupils to the exact location of the eyes. Instead, only a few methods take advantage of the context, and even fewer works attempt to combine it with the human pose [160]. These elements are used more often for gaze target detection [162, 53], as they are required to relate the target of the gaze to the elements in the scene.

However, we believe that the scene context and the human pose contain

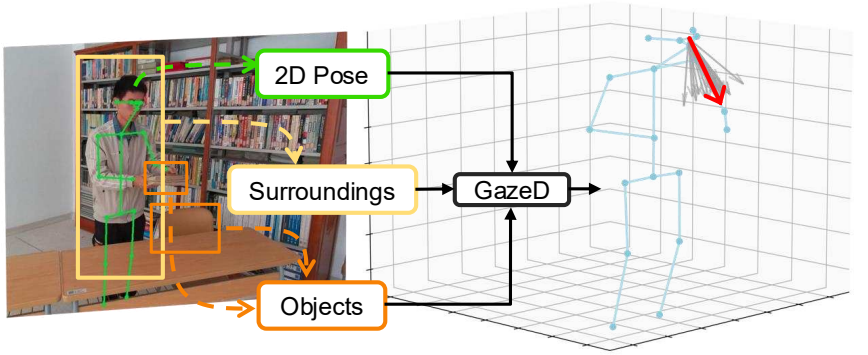


Figure 6.1: GazeD method jointly predicts 3D gaze and body pose analyzing the 2D pose, the surrounding of the subject and the context, in terms of objects in the scene.

knowledge useful also for 3D gaze estimation: the context influences the gaze and the gaze itself strictly depends on the body pose. Indeed, previous studies have shown that gaze direction and body pose are closely interrelated [68]. Some works [160, 121] utilize the 2D pose or the head or body orientation to estimate the 3D gaze. However, these methods lack a mechanism to directly correlate the pose with the final gaze output.

Therefore, in this chapter , we introduce GazeD, that efficiently combines different elements from the scene, *i.e.*the 2D body pose, the subject's surroundings, and the global context with objects, to output the 3D gaze direction (see Fig. 6.1). A key idea of GazeD is to model the gaze as a virtual protrusion from the forehead of the person, between the eyes, where we placed **an additional joint** here referred to as **gaze joint** (see Fig. 6.2). The gaze joint has a variable direction (the gaze angle), while its distance from the head is fixed. Therefore, the gaze joint is not positioned in correspondence with the target object, as required to solve the gaze target detection task, but it acts as a proxy to compute the gaze direction.

Having modeled the gaze direction using an additional joint enables the resolution of the problem as an extension of 3D human pose estimation. GazeD is thus based on a regression head, which outputs both pose and gaze working on a common embedding.

Because of the intrinsic ambiguity of lifting 2D information to the 3D world, as well as the multiple possible gaze directions given the body posture and the

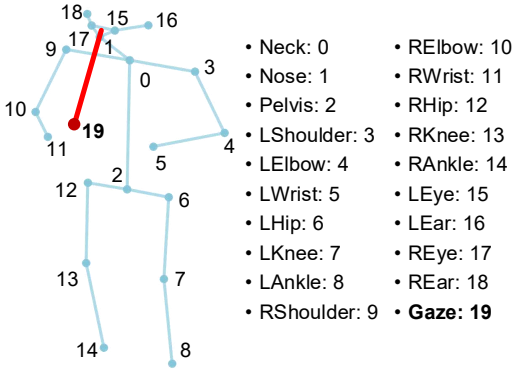


Figure 6.2: Human body skeleton with our additional gaze joint.

context of the scene, GazeD regresses the 3D gaze and pose using a diffusion model. By conditioning the denoising process using 2D pose, surroundings, and context features, GazeD models the uncertainty in the data and generates multiple plausible output hypotheses. To our knowledge, we are the first to adopt a diffusion model to regress 3D gaze direction.

The embeddings used as conditioning for the diffusion model are generated by GazeD as two consecutive steps. Starting from the 2D pose estimated by an off-the-shelf method, the first step recovers information from the context close to the subject. The second step extracts additional cues from the objects in the scene, *i.e.*it captures the context of the scene far from the subject.

As an additional advantage, GazeD works on a single RGB image, avoiding the computational complexity of processing video sequences and the need for specific hardware. This streamlines its adoption in real-world applications, simplifies the training procedure, and facilitates the acquisition of new datasets. In contrast, 3D estimators based on sequences of frames [121, 54, 76] or specific data modalities, such as depth maps or point clouds [160, 67, 47], have more limited applicability in real-world scenarios.

In summary, our contributions encompass the following:

- We introduce GazeD, a method for 3D gaze estimation that combines surrounding context, object-level cues, and 2D human pose features to condition a diffusion model, whose denoising process yields multiple plausible hypotheses for 3D gaze and human pose.

- We propose a novel gaze representation as an additional joint of the human skeleton, enabling the method to jointly output both the 3D gaze direction and the 3D human pose in a unified framework.

- Extensive experiments on multiple datasets demonstrate that GazeD achieves state-of-the-art performance in 3D gaze estimation, even surpassing approaches relying on multiple input modalities, while also maintaining high accuracy in 3D human pose prediction.

6.2 Related Work

6.2.1 3D Gaze Estimation

Recently, research in 3D gaze estimation has evolved significantly. Approaches are broadly categorized into two categories [121]: geometry-based and appearance-based ones.

Geometry-based methods. These methods [107, 58, 91, 211] rely on constructing a 3D model of the eye using optical or geometric properties. These techniques are accurate in controlled environments (*e.g.* good light conditions [118]) with consistent subject characteristics (*e.g.* head position [169]). Unfortunately, they often require specialized and expensive hardware – such as infrared cameras or eye-tracking devices – extensive calibration, limiting their applicability in real-world settings.

Appearance-based methods. These methods [43, 198, 199, 22] have gained popularity due to their reliance on standard RGB cameras to estimate the gaze from eye and face images directly. Early appearance-based approaches used hand-crafted features, such as pixel intensity or eye shape, but suffered from limited robustness in unconstrained environments. The advent of deep learning has significantly improved the performance of these methods, enabling more robust gaze estimation across varying lighting conditions, head poses, and subjects [23].

Some methods [160, 67, 47, 66] use RGB and depth data to recover scene depth, but this requires specialized hardware and is less suitable for outdoor use due to limitations like sunlight interference [141]. Alternatively, temporal information modeled with RNNs or LSTMs has improved gaze estimation by capturing movement patterns [121, 206, 129], though such approaches demand significant computational resources to handle long video sequences.

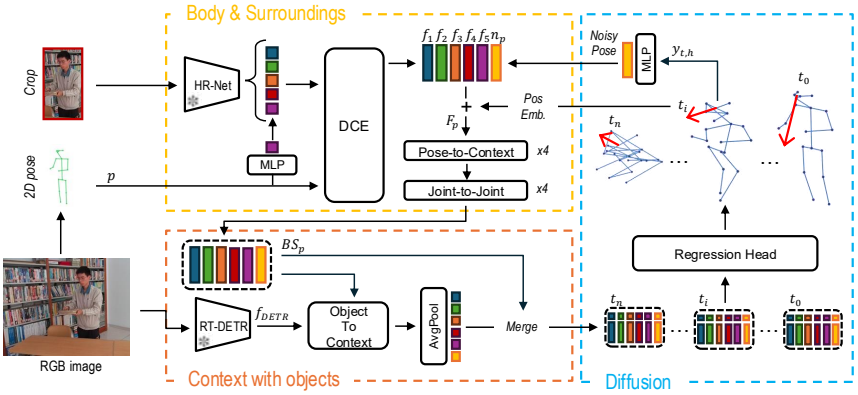


Figure 6.3: Overview of the proposed GazeD method that predicts the 3D gaze and human pose starting from a single input RGB image, combining information from the 2D body pose, surroundings, and context with objects.

6.2.2 3D Human Pose Estimation

3D Human Pose Estimation (HPE) typically involves estimating 2D poses and then lifting them to 3D, a step that remains challenging due to the ambiguity of inferring 3D from 2D [14, 17]. To address this, some methods use temporal information [94, 205], although this adds latency. Given the under-constrained nature of the problem [151], multihypothesis approaches generate multiple plausible 3D poses instead of a single estimate, using techniques such as mixture density networks [122] or conditional variational autoencoders [148].

Diffusion Models. More recently, Denoising Diffusion Probabilistic Models [60] have been applied to 3D HPE. These models treat 3D pose estimation as a reverse diffusion process, where a highly uncertain 3D pose distribution is progressively refined toward a more accurate pose. Methods like DiffPose [49] leverage spatial-temporal context from 2D pose sequences to guide this diffusion process. A key advantage of diffusion models in this context is their ability to generate multiple hypotheses, providing a probabilistic framework naturally, and this allows for improved performance by aggregating multiple outputs, effectively reducing the impact of outliers. Furthermore, graph convolutional neural networks have been integrated with diffusion models [24] to explicitly capture the correlations between joints, enhancing pose estimation accuracy.

6.3 Method

Given a single RGB image $I \in \mathbb{R}^{H \times W \times 3}$ as input, our goal is to predict the 3D gaze direction together with the 3D pose of the person in the scene. To this end, we define an additional gaze joint and we concatenate it to the list of body joints to provide the output $\mathbf{y} \in \mathbb{R}^{J \times 3}$, where J is the number of skeleton's joints, including the gaze joint (see Fig. 6.2). The gaze unit vector v is defined as the direction from the midpoint between the eyes and the gaze joint:

$$v = \left\| \mathbf{y}^{Gaze} - \left(\frac{\mathbf{y}^{LEye} + \mathbf{y}^{REye}}{2} \right) \right\| \tag{6.1}$$

As shown in Figure 6.3, GazeD is composed of three modules: **Body & Surroundings**, responsible for the feature extraction from the human pose and surroundings, **Context with objects** to integrate the general context, including the location of objects, and a **Diffusion** module that contains a regression head for the 3D gaze and pose prediction as well as the diffusion scheduler.

6.3.1 Body & Surroundings

This module takes as input a cropped image of the person, along with its 2D pose $\mathbf{p} \in \mathbb{R}^{J \times 2}$. The pose $\mathbf{p}$ is obtained by concatenating the output of a 2D pose estimator with an additional joint representing the 2D gaze. However, as the gaze joint is virtual and lacks a correspondence in the image, we set the 2D gaze point as the midpoint between the eyes. Although it is not the 2D projection of the corresponding 3D gaze joint, such point has proven to be a good and consistent initialization. For this joint, it will not only be necessary to perform a third-dimensional lifting, but all three coordinates must be correctly estimated by the method.

We use a HR-Net[157] backbone to extract intermediate hierarchical features $\mathcal{H} = \{\mathbf{H}_l \in \mathbb{R}^{H_l \times W_l \times C_l}\}_{l=1}^{L}$, where L is the number of feature maps ($L = 4$ in our experiments).

Deformable Surroundings Extraction. As shown in [108, 202], it is possible to encode fine-grained visual cues – *i.e.*, the joint locations – and extract high-level semantics – *i.e.*, the spatial configuration of the joints – via the high- and low-level features of a stacked network based on down-sampling operations [157, 194]. Therefore, following [202], we leverage a Deformable Context Extraction (DCE) module, based on the deformable attention mechanism [210]. DCE extracts spatial

Method	Office	Living Room	Kitchen	Library	Courtyard	All
Fixed bias	88.0/76.0	85.5/76.7	86.0/82.4	89.0/85.1	89.7/88.7	88.1/79.7
Frontal gaze	22.6/21.9	36.6/35.4	17.9/19.6	27.1/25.8	30.5/33.8	28.8/28.8
Dias *et al.* [36]	−/27.2	−/25.2	−/19.8	−/24.9	−/36.1	−/27.1
XGaze [198]	24.2/23.0	42.0/40.9	23.3/22.9	24.6/22.3	30.2/31.9	29.2/28.4
Nonaka *et al.* [121]	20.0/18.1	25.6/25.5	21.5/18.6	21.9/20.1	28.4/30.5	24.1/23.3
Gaze360 [78]†	24.0/19.2	41.1/31.3	32.4/21.2	27.5/20.7	28.2/28.3	30.4/24.5
Nonaka *et al.* [121]†	**14.4**/**14.3**	25.1/22.6	20.4/19.6	19.8/18.4	25.4/**26.9**	21.7/20.9
Ours$_{(H=20,\ A=AVG)}$	15.8/16.3	**19.3**/**20.6**	**18.2**/**19.5**	**17.6**/**16.9**	**25.3**/29.1	**19.5**/**20.5**
Ours$_{(H=20,\ A=ORC_G)}$	11.6/11.6	13.2/13.8	14.6/13.7	14.2/12.9	23.9/27.9	15.9/16.3

Table 6.1: Experimental results on GAFA dataset expressed as MAE_{3D}/MAE_{2D}. † indicates methods leveraging temporal information.

contextual cues from the computed intermediate feature maps using the initial 2D pose joints as reference points. Linear projections of the 2D input poses are concatenated to the hierarchical features $\mathcal{H}$ as an additional channel.

The output of the DCE module $F'_p \in \mathbb{R}^{(L+1)\times J\times d}$ is an embedding containing near context (*i.e.*, surroundings) and body pose features. We fixed $d = 128$ in our experiments. A linear projection of the noisy 3D poses at the current timestep coming from the diffusion scheduler (see Sect. 6.3.3) is then concatenated to F'_p. Moreover, a positional encoding of the diffusion timestep is added in order to generate $F_p \in \mathbb{R}^{(L+2)\times J\times d}$ and to make the model aware of the current diffusion step.

Pose-to-Context and Joint-to-Joint Modules. Drawing inspiration from multi-modality models [82, 3] that employ a unified transformer encoder, we utilize a similar architecture to learn a joint representation. F_p is a multichannel descriptor, which contains two channels for the pose and L channels for the context. The Pose-to-Context Attention Module performs a self-attention among the $L + 2$ descriptors (tokens) of size d for each joint independently. The Joint-to-Joint Attention Module considers J tokens of size $d' = d \cdot (L + 2)$ and computes self-attention among them. Concretely, the Pose-to-Context module enriches the embeddings of each joint with contextual information, while the Joint-to-Joint module enables data sharing between the different joints. $BS_p \in \mathbb{R}^{d'\times J}$ is the final output of the Body & Surroundings module, with a descriptor of size d' for each joint.

6.3.2 Context with Objects

The goal of this module is extracting information from the whole image, particularly focusing on elements that can affect or guide the person's gaze – *i.e.*, the objects. To this aim, we use a DETR-like object detector, which is able to provide a descriptor of the objects in the image, with knowledge of both their location and their class. Let $F_{DETR} \in \mathbb{R}^{Q \times d'}$ be the last hidden states (removing the localization and classification heads and projecting to the common size d') of the detector obtained with Q input queries. The Object-To-Context block performs a cross-attention between F_{DETR} and BS_p. An average pooling operation is applied along the query dimension to merge all the important information related to the scene objects. The obtained embedding $CO \in \mathbb{R}^{d'}$ is merged with BS_p^{Gaze} to generate the final Pose&Gaze embedding $PG \in \mathbb{R}^{d' \times J}$.

6.3.3 Diffusion-based Multi-hypothesis Generation

Estimating the 3D gaze and pose of people from RGB is inherently challenging. Major issues are the partial or complete occlusion of the eyes, and the lack of depth information. Therefore, in this context, we propose the use of diffusion models to estimate the gaze direction, as their ability to generate multiple plausible hypotheses based on the person's pose and contextual information becomes highly valuable. By modeling various potential gaze directions, the diffusion process accommodates the inherent uncertainties and ambiguities in the data.

Representing pose and gaze direction using a single skeleton with an additional joint brings two advantages. First, it enables the formalization of the global inference process as a denoising task, starting from a completely random pose sampled from a unique Gaussian distribution. Second, it simplifies the optimization function: we adopted a single MSE loss between the predicted and the ground-truth joint coordinates, implicitly incorporating and standardizing the contributions of the pose and the gaze.

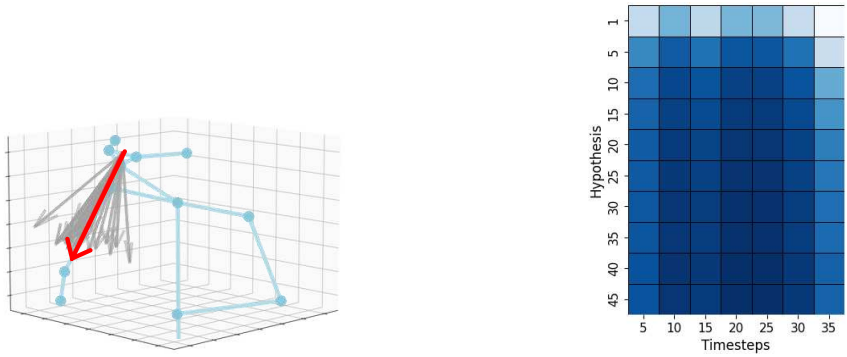
The iterative denoising procedure can be applied in parallel to H initial hypotheses $\hat{y}_{N,h} \sim \mathcal{N}(0; 1)$ in order to generate H final predictions $\hat{y}_{0,h} \in \mathbb{R}^{J \times 3}$ after N denoising iterations. For efficient inference, we employ the optimized DDIM [154] denoising scheduler. A regression head is included in the diffusion module and it is trained to perform the denoising task.

Gaze and Pose aggregation. GazeD generates H hypotheses of the gaze and the pose, each one representing a plausible 3D solution. The distribution itself contains additional information about the real gaze (and posture). Therefore, a

correct aggregation of the generated hypotheses allows to reduce the prediction error of a single hypothesis.

As aggregation function A, we adopt the average operation (**AVG**) at joint level. Despite its simplicity, this aggregation has proven to be effective and accurate, as reported in experiments. For the sake of completeness, we also compute the "Supervision from an Oracle" [148] aggregation ($\mathbf{ORC}_{\mathrm{G}}$) that selects the closest hypothesis with respect to the ground truth annotation. This aggregation is useful to highlight the upper-bound performance of the proposed method, but it is limited in its applicability when ground truth annotations are not available. Additional aggregation functions based on oracle are investigated in Section 6.4.6. We avoided using additional aggregation techniques that required ground-truth information or camera calibration parameters [146], which are not always available or predictable in single-frame methods.

6.4 Experimental Evaluation



(a) (b)

Figure 6.4: (a) GazeD predicts multiple hypotheses on which aggregation functions are applied (see Sect. 6.3.3). (b) Investigation on the number of different hypotheses vs number of timesteps. Darker color represents a lower $\mathrm{MAE}_{3\mathrm{D}}$ value.

6.4.1 Datasets

As GazeD is based on the rich information extracted from the surroundings and global context, we focus on datasets containing 3D gaze annotations and full images, thus excluding those only containing crops around the faces, eyes or body of the subject [78, 43, 198]. Additional details about datasets are reported in the Supplementary.

GAFA [121] (Gaze from Afar) dataset is designed for 3D gaze estimation in surveillance scenarios, capturing freely moving people in natural settings. It includes more than 850k video frames from 5 different daily environments. It features a wide range of head poses, including back views and high-pitch angles, reflecting realistic conditions. GAFA is annotated with 3D gaze directions and body orientations, using wearable cameras and AR marker-based positioning systems for ground truth.

GFIE [67] is a dataset introduced for 2D and 3D gaze-following tasks, created using a system that combines a laser rangefinder and an RGB-D camera to record and annotate gaze behaviors in natural indoor environments. The system guides the subject's gaze target using a laser spot, which is then detected in the RGB images to generate precise annotations, and then removed using image inpainting [166]. The 3D gaze target is reconstructed using the distance measured by the laser rangefinder and the camera's intrinsic parameters. The dataset includes about 71k frames of 61 subjects, performing a wide range of activities.

Ego-Gaze. We create this dataset starting from the multimodal Ego-Exo4D dataset [52]. Specifically, we select frames from the Ego-Pose subset in which the 3D annotation of the human pose is available. Then, we compute 3D gaze annotations from the data acquired with the Aria glasses[1]. The dataset includes a wide range of skilled activities— such as sports, music, dance—performed in natural settings. Because the Ego-Pose dataset is still used for competitions, the official test set has not been made available. Therefore, we use the official validation split as test set and we sample validation instances from the training set. We will release the detailed lists of frames for each split, enabling future fair comparisons on Ego-Gaze.

6.4.2 Implementation Details and Training

As backbones, we use different pre-trained models. For 2D human pose estimation, we use HRNet [157], capable to maintains high-resolution representations through

[1] www.projectaria.com

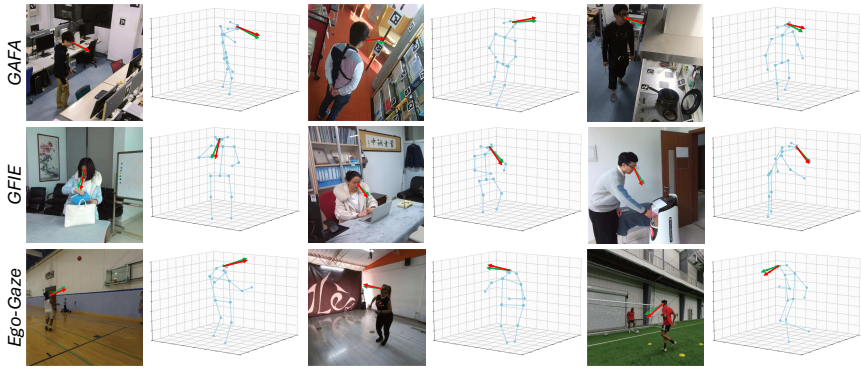


Figure 6.5: Qualitative Results on the three datasets. For the gaze, green is for the ground truth while red for the prediction

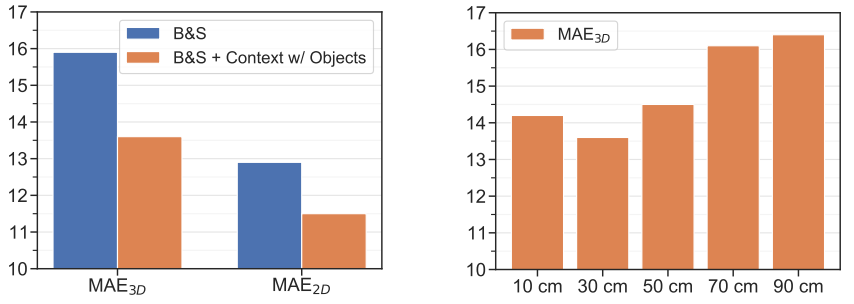
Method	**RGB**	**Crops**	**Depth**	**MAE$_{3D}$**
Random				84.4
Center				87.2
GazeFollow [136]	✓	✓		41.5
Lian *et al.* [97]	✓	✓		26.7
Rt-Gene [43]	✓	✓		21.0
Hu *et al.* [67]	✓	✓	✓	17.7
Toaiari *et al.* [160]			✓	15.9
Chong *et al.* [25]†	✓	✓		20.8
Gaze360 [78]†	✓			19.8
Ours$_{(H=20,\ A=AVG)}$	✓			**13.6**
Ours$_{(H=20,\ A=ORC_g)}$	✓			9.9

Table 6.2: Quantitative results on GFIE dataset. For each method, the input data is reported: **RGB** for color images, **Crops** for head, face, or eye crops, and **Depth** for depth maps. † indicates methods leveraging temporal information.

the whole architecture and to achieve great accuracy. As an object detector, we use RT-DETR [204], a recent end-to-end architecture with good accuracy and real-time performance for the object detection task from single RGB images. Both models are frozen during the training phase and are used with their original weights and

Method	Basket	Dance	Various	All
XGaze	21.6/20.6	23.8/27.1	21.6/20.5	23.1/24.9
Gaze360	21.8/17.0	21.0/21.8	22.1/17.6	21.3/20.4
Ours	**15.3/14.5**	**18.7/19.1**	**15.5/14.4**	**17.7/17.5**

Table 6.3: Quantitative results on the Ego-Gaze dataset. GazeD is tested with H=20, A=AVG.



(a) (b)

Figure 6.6: Ablation study on GFIE dataset: (a) the contribution of the "Context with objects" module in addition to the "Body&Surroundings". (b) Performance of GazeD in terms of MAE_{3D} by varying the distance of the gaze joint from the head.

parameters.

We train GazeD with a batch size of 64 for 100 epochs on all datasets. We use Adam optimizer [83] with a starting learning rate of $6e^{-4}$ using a linear decay with factor 0.993. No data augmentation is applied on input images.

6.4.3 Baselines and Competitors

We compare GazeD with several 3D gaze estimation baselines and competitors.

For the GAFA dataset [121], we compute two baselines. The first is *fixed bias*, *i.e.*the mean gaze direction is obtained from the training set, and the error metric is computed using this mean value over the test set [121]. This baseline is intended to show the lower bound accuracy on this dataset. The second baseline

#	Diffusion	Objects	Near Context	MAE_{3D}	MAE_{2D}
1	✗	✓	✓	16.1	14.2
2	✓	✗	✓	15.9	12.9
3	✓	✓	✗	15.1	12.0
4	✓	✓	✓	13.6	11.5

Table 6.4: Ablation analysis of each component.

Aggregation method	GAFA [121]		GFIE [67]	
	MAE_{3D}	MAE_{2D}	MAE_{3D}	MAE_{2D}
AVG	19.5	20.5	13.6	11.5
ORC_P	19.7	20.4	13.4	10.8
ORC_G	15.9	16.3	9.9	7.9
ORC_J	12.6	13.3	8.7	7.5

Table 6.5: Impact of aggregation methods.

is *frontal gaze*, where we compute the angular error assuming that the predicted gaze direction is always orthogonal to the line between the two eyes. This is a useful reference for understanding the precision that a method based solely on the pose of the head would achieve. As competitors, we use a variety of recent and state-of-art methods from the literature. The approach proposed by Dias *et al.* [36] estimates 2D gaze on the image plane using facial keypoints detected by OpenPose [13]. Gaze360 [78] takes a sequence of full-head images as input and provides the 3D gaze direction. XGaze [198] uses facial images as input and assumes high-resolution facial images.

The GFIE dataset was originally proposed for gaze target detection, and thus provides additional information on the scene – *i.e.*depth maps. For this reason, we evaluate on this dataset with other baselines and competitors. The baseline methods include the *random* approach, *i.e.*the 2D and 3D gaze directions are randomly selected within the image and point cloud, respectively. In addition, the *center* baseline localizes the gaze always at the center of the point cloud of the 3D space. As competitors, we also use existing 2D gaze-following methods, *i.e.*GazeFollow [136], Lian [97], and Chong [25]. To retrieve their 3D gaze angle, we first back-projected the 2D gaze target into the 3D space using the available registered depth maps. The results of Gaze360 [78] and Rt-Gene [43] are collected from [66]. Finally, we report the results of the recent work by Toaiari *et al.* [160], which utilizes upper-body skeleton data and the depth map of the scene to predict the 3D gaze.

6.4.4 Comparison with state-of-the-art

We report the result using the Mean Angular Error (MAE), the standard metric for the evaluation of gaze estimation methods. MAE is expressed in degrees, and it is calculated as the average of the angular difference between the predicted and ground-truth gaze directions over all the testing samples. In addition to the 3D

Aggregation method	GAFA [121]		GFIE [67]	
	MAE_{3D}	MAE_{2D}	MAE_{3D}	MAE_{2D}
AVG	19.5	20.5	13.6	11.5
ORC_P	19.7	20.4	13.4	10.8
ORC_G	15.9	16.3	9.9	7.9
ORC_J	12.6	13.3	8.7	7.5

Table 6.6: Impact of different aggregation methods on gaze.

errors (MAE_{3D}), we report the metric using the directions on the image plane (MAE_{2D}).

Table 6.1 reports the performance of our GazeD and other approaches. On the GAFA dataset, as shown, GazeD achieves the best performance on both metrics, even outperforming methods that leverage additional temporal information, *i.e.* [78, 121]. Both MAE_{3D} and MAE_{2D} achieved by our method are always well below the *frontal gaze* baseline, indicating that the idea to model the gaze as additional joint is effective in estimating the 3D gaze direction and that the output of our method is not the mere head pose.

Table 6.2 reports similar results on the GFIE dataset. Also in this case, our method largely outperforms the competitors. In particular, GazeD outperforms even methods that are based on fine details, such as the face or body crops, or additional input data as depth maps, whose contribution is significant in 3D estimation tasks.

Methods	Mart. [110]	Zhao [200]	Sun [158]	Yang [188]	Hoss. [63]	Liu [102]	Xu [187]	Zhao [203]	Zhao [201]	Diffu. [24]	Diffp. [49]	**Ours** (H=20, A=AVG)	**Ours** (H=20, A=ORC)
MPJPE ↓	62.9	60.8	59.1	58.6	58.3	52.4	51.9	51.8	**43.4**	<u>49.4</u>	49.7	49.7	41.1

Table 6.7: Results on Human3.6M dataset for the 3D Human Pose Estimation task. The best result is in **bold**, the second one is <u>underlined</u>.

Table 6.8: Results on MPI-INF-3DHP dataset for the 3D Human Pose Estimation task. The best result is in **bold**, the second one is <u>underlined</u>.

Methods	Pavllo [130]	Zheng [205]	Wang [173]	Li [94]	Zheng [205]	Zhang [195]	Zhao [201]	**Ours** (H=20, A=AVG)	**Ours** (H=20, A=ORC)
MPJPE ↓	84.0	77.1	68.1	58.0	57.7	54.9	**44.7**	<u>46.6</u>	33.8

In both datasets, we also report the results obtained using the ORC_G aggregation function. As expected, these are the best results. However, the ground truth is not normally available in the inference phase, and it is not completely correct to use it to select the best hypothesis. We report these results to highlight that the diffusion process is able to generate hypotheses very close to the ground truth and that future work may focus on more sophisticated aggregation strategies to improve performance.

The newly introduced Ego-Gaze dataset imposes re-training the Gaze360 and XGaze methods. Unfortunately, it was not possible to implement more recent methods, such as the ones developed in [121, 67], due to a lack of depth maps and body or head orientation, respectively. Results are reported in Table 6.3, organized in three main scenes, *i.e.*, basket, dance and various. The latter includes the less represented classes, such as cooking, soccer and bike repair. As shown, GazeD achieves the best results in all the scenes. These results demonstrate the robustness of the proposed approach on a challenging dataset with complex scenes.

6.4.5 Qualitative results

Some qualitative results are reported in Figure 6.5, where the input image and the predicted 3D gaze and pose are shown. For the gaze, the ground truth vector is drawn in green, the prediction in red. These results confirm the ability of GazeD to predict gaze direction in wide angle ranges, also when the face is not visible or partially occluded. Additional qualitative results are reported in the Supplementary material.

6.4.6 Ablation Studies

Ablation studies are mainly computed on the GFIE dataset, using GazeD in the configuration described in Section 6.4.2.

Module contributions We investigate the contribution of each module (see Table 6.4). In experiment #1, we use the transformer-based model without the diffusion process, training the network to predict directly pose and gaze. In #2, we remove the module "Context with Objects" for context analysis. In #3, we remove the part of the method responsible to extract and process the surroundings of the person. Each module is a key part of the method.

Context with Objects module To highlight the performance improvement provided by the proposed "context with object" module, we tested the results of GazeD directly using BS_p in input to the diffusion step (see Fig. 6.3). The results are

reported in Figure 6.6a, Adding the object embeddings clearly improves the model ability to solve the 3D gaze estimation task.

Distances of the Gaze Keypoint The gaze joint is an auxiliary point used to solve the task, but it is not physically present. Its distance from the eyes was chosen to be close enough to the body to be modeled as a joint and, at the same time, far enough to reduce the dependence on noise in the final conversion into angles. The measurement used, equal to 30 centimeters, is supported by an experimental analysis. MAE_{3D} errors vs distances are plotted in Figure 6.6b.

Number of hypothesis and timesteps A key advantage of diffusion models is their ability to generate multiple hypotheses. In Figure 6.4a, a real multiple hypotheses prediction is depicted. Then, we analyze how the number of hypotheses H and denoising iterations N affect the final MAE_{3D}. Figure 6.4b shows the matrix from which we selected the final values of H and N, where darker colors denote lower errors. Based on this analysis, we selected $H = 20$ and $N = 20$ for our evaluation, as a favorable trade-off between accuracy and computational load.

Multiple Hypothesis Aggregation Strategies Having multiple generated hypotheses allows us to explore various aggregation strategies. In Table 6.6, we compare the different aggregations described in Section 6.3.3. We also investigate different oracle selections [148] obtained in three different ways. ORC_{G} chooses the hypothesis with the lowest error specifically at the gaze joint. ORC_{P} selects the hypothesis with the lowest Mean Per-Joint Position Error (MPJPE) relative to the ground truth; however, our results indicate that minimizing MPJPE at the pose level does not necessarily produce the most accurate gaze estimation. Finally, ORC_{J} employs a per-joint selection strategy in which, for each joint, the coordinates with the lowest error are independently selected, resulting in a more accurate estimation of gaze direction. Since ORC_p, ORC_g, and ORC_j rely on ground truth data, they are not applicable in real-world scenarios. Therefore, we consider **AVG** as the most appropriate baseline for fair comparison.

6.4.7 Additional Evaluation

GazeD predicts not only the 3D gaze, but also the 3D body pose: then, we analyze the performance of this task.

Dataset The Human3.6M dataset [72] is a well-known dataset of 3.6 million images with 3D human pose annotations. It contains 17-joint skeleton annotations for 11 subjects performing 15 activities, captured by 4 cameras in an indoor environment. For evaluation, we follow the standard protocol of training on subjects S1, S5, S6, S7, and S8, and testing on subjects S9 and S11. The MPI-INF-3DHP

dataset consists of over 1.3 million frames captured from 14 cameras and it is widely used for training and evaluating 3D human pose estimation models. It contains 8 actors performing activities such as walking, sitting, sports. The frames are annotated using a skeleton model with 17 joints.
3D Pose Evaluation For the training, we use a batch of 128 for 50 epochs. Other training settings are the same used for the gaze evaluation. Performance is evaluated using the Mean Per Joint Position Error (MPJPE) [77], which calculates the average Euclidean distance (in millimeters) between predicted and ground truth 3D joint coordinates. In Tables 6.7 and 6.8, we report the comparison for the 3D pose estimation task between our model and literature competitors, on Human3.6M and MPI-INF datasets, respectively. Among the others, Diffupose [24], Diffpose [49] are the most similar methods since based on a diffusion architecture. As shown, results obtained are better than a large portion of the literature, and comparable with the most recent one. These experimental results suggest that, although our method was not specifically developed for the HPE task, it still achieves competitive results with a good level of accuracy.

6.5 Conclusion

We introduced GazeD, a novel method for 3D gaze and pose estimation from single RGB images. By modeling 3D gaze through a diffusion process, GazeD effectively integrates 2D pose, surrounding context, and global scene cues. Morever, the use of a diffusion model allows GazeD to address the inherent ambiguity of 3D gaze estimation, generating multiple plausible hypotheses. Experimental results demonstrate the efficacy of GazeD, highlighting its potential for accurate 3D gaze and pose estimation.

6.6 Additional qualitative results

We report qualitative results for the three datasets (GFIE [67], GAFA [121] and Ego-Gaze) in different scenarios, with different camera angles and subject distances from the camera. ”Context with objects” helps the model to get a more accurate result. In Figure 6.7, 6.8, 6.9, the first column shows the original image, the second one shows the predicted and GT gaze direction, the third compares the GT pose with the predicted one. Every prediction is the average of the 20 hypotheses output of the diffusion model. In Figure 6.10 a visual comparison

between GazeD and Gaze360 [78] and Hu *et al.*[67] is reported.

6.7 Dataset preprocessing

6.7.1 GFIE

GFIE dataset does not provide 2D/3D human poses. Every dataset sample is composed by a pair of rectified RGB and depth images and gaze annotations. Since the depth maps are noisy with lots of missing values, computing only the 2D pose and then using depth values to obtain the 3D pose was not a good solution. We then decided to use metrabs [140] to estimate both the 2D and 3D poses.

6.7.2 GAFA

GAFA dataset [121] is provided with only 2D poses computed with an old version of OpenPose [12]. As GAFA dataset is a multiview dataset, the optimal solution to get 3D poses was to triangulate multi-view keypoints using intrinsics and extrinsics provided by the dataset, applying a RANSAC triangulation method [43] and then transform the resulting pose from the world reference frame to each camera frame. Despite the presence of these poses, GAFA dataset contains several complex scenarios, with the subjects often occluded; openpose annotations often included missing keypoints, resulting in non-accurate triangulated human poses. For this reason, 2D poses were recomputed with metrabs, which can better handle these cases. If some 3D poses were not computable due to limited views, those poses were discarded from the training set, while in the test set, body and head locations present in the annotations of the dataset were used to substitute the fundamental joints to run our method (pelvis and eye midpoint) and gaze joint were added to ensure that all test samples were included.

6.7.3 Ego-Gaze

Ego-Gaze dataset is realized starting from Ego-Exo [52] dataset, which includes Aria Glasses[2] eye-tracking data. However, despite the presence of these annotations, the dataset does not include a dedicated subset specifically designed for the gaze estimation task. Therefore, we created a subset starting by collecting exocentric frames from the Ego-Pose subset, which comprises automatic and

[2] www.projectaria.com

manual annotations of 2D Poses and 3D triangulated poses. We picked frames with manual annotations to ensure the best possible poses. Since also manual annotations have missing keypoints, only poses with all the joints were included in our subset. Gaze was inserted as a joint in all the poses just by converting the rotation angles reported in the aria files into normalized cartesian coordinates and rotating them in the camera reference frame using the camera poses. Since the pose annotation is not provided with a center pelvis, it was manually added as the mean point between the left and the right hip just to have the root joint. To enable a comparison with competitors, also head crops were needed. Since the dataset doesn't provide them, and using face joints to crop the images was not an efficient solution in many cases, we trained a YOLOv8[3] with the Hollowood Head dataset [171] to detect heads in the scene. The Yolov8 was run in inference with all the videos of Ego-Pose to ensure that sequential frames were produced for a temporal method such as Gaze360. After obtaining the head crops, only samples with the available complete 3D pose were kept. The final Ego-Gaze subset comprises about 157k frames for the training set, 20k for validation, 45k for testing.

[3] https://github.com/ultralytics/ultralytics

Figure 6.7: Qualitative results of GazeD on the GFIE dataset. Ground truth data is shown in green, predictions from GazeD in red.

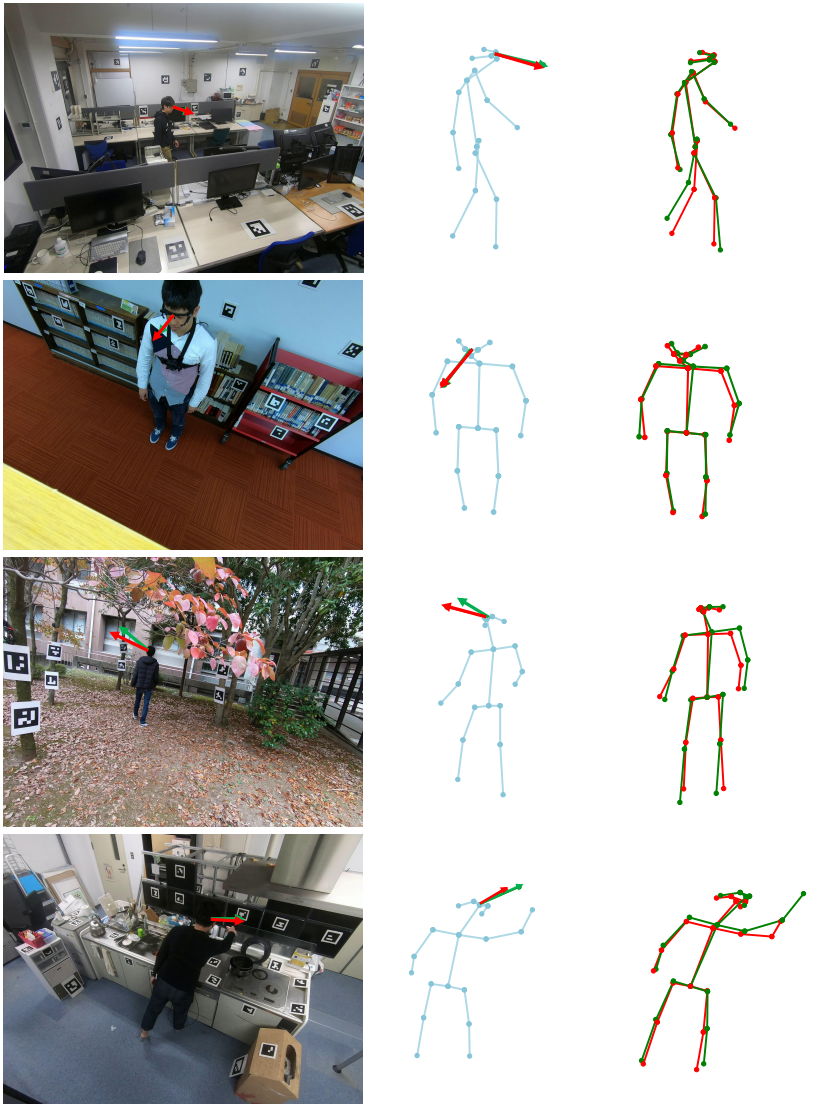


Figure 6.8: Qualitative results of GazeD on the GAFA dataset. Ground truth data is shown in green, predictions from GazeD in red.

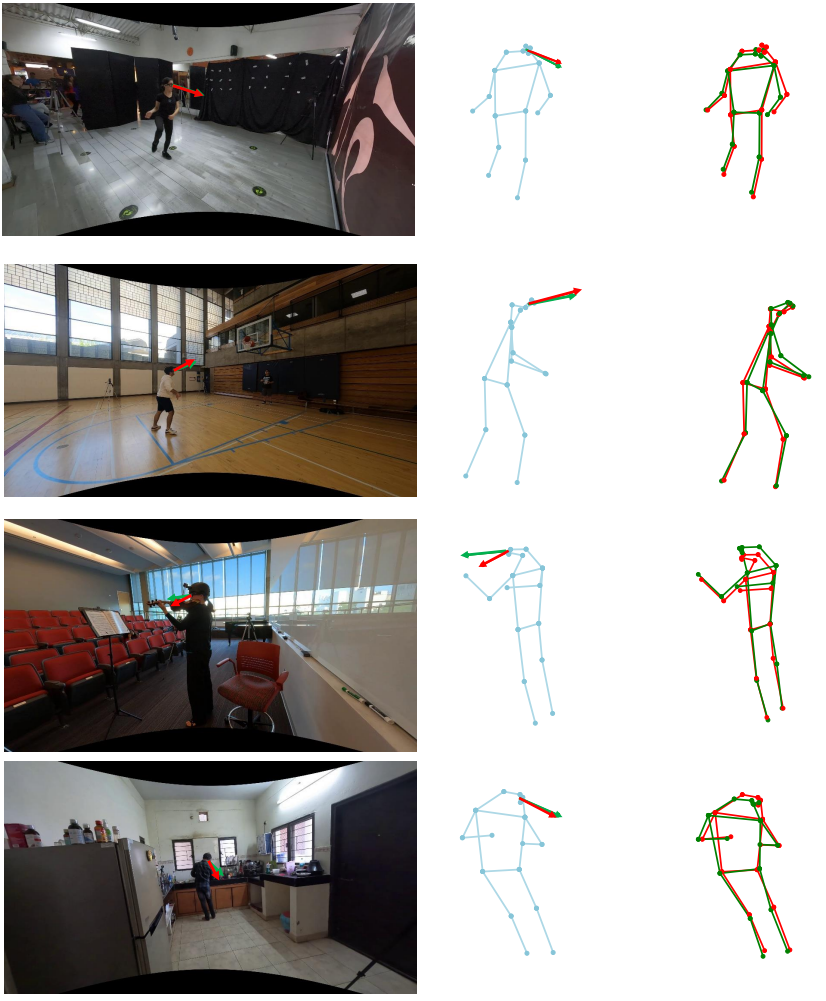


Figure 6.9: Qualitative results of GazeD on the Ego-Gaze dataset. Ground truth data is shown in green, predictions from GazeD in red.

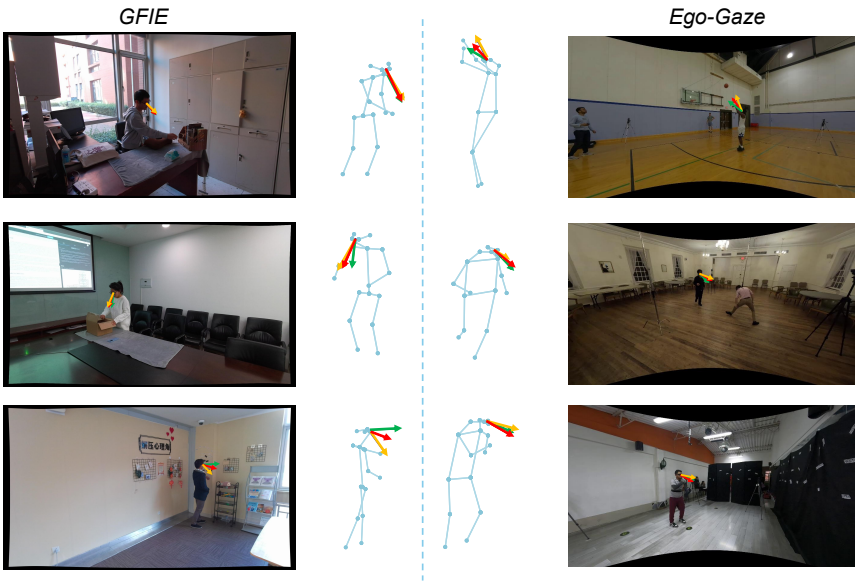


Figure 6.10: Qualitative comparison with competitors. Ground truth data is shown in green, predictions from GazeD in red, and in yellow are reported the predictions of Gaze360 [78] (right) and [67] (left).

Chapter 7

Fake Detection

7.1 Introduction

With the widespread adoption of Generative AI, 2D deep fake detection has become a well-established task in computer vision, with numerous datasets and benchmarks available [96, 177, 4, 30]. However, generative methods are rapidly evolving beyond the two-dimensional domain. Recent developments integrate models such as Variational Autoencoders (VAE) [84], GANs [50], and Diffusion Models [138] to improve 3D reconstruction fidelity and reduce inference time. For instance, modern text-to-3D pipelines rely on text-to-image generation followed by 3D synthesis to ensure both semantic consistency and geometric accuracy [186, 105]. Other approaches enable complex edits on real 3D scenes [57, 181].

Unlike 2D manipulations, edits performed directly in the 3D scene representation can produce an unlimited number of photorealistic, view-consistent images, making traditional artifact-based detection strategies limited in efficacy. This shift introduces a new class of deep fakes where geometric consistency becomes an advantage for the attacker rather than a vulnerability. The ability to edit 3D scenes is increasingly deployed in Augmented Reality or Virtual Reality environments, digital twins, and simulation pipelines for autonomous systems [75, 207, 209]. These elements make the authenticity of 3D content not only a scientific challenge but a security concern. Indeed, despite these progresses and risks, 3D Deep Fake Detection remains largely unexplored in the literature. Also, while 2D deep fake detection benefits from mature benchmarks and shared evaluation protocols, the 3D domain lacks definitions, datasets, and standardized procedures for assessing

scene authenticity.

Motivated by these reasons, we formalize 3D Deep Fake Detection as the task of determining whether a 3D scene representation has been altered in its geometry, spatial layout, or appearance parameters, regardless of how realistic its rendered images appear. We establish the first benchmark for authenticity assessment in 3D Gaussian Splatting, laying the foundations for future research on multi-view consistency and 3D aware forensics. To sum up, our contributions encompass the following:

- We formalize 3D Deep Fake Detection for 3D scene representations focusing on manipulations of geometry, spatial layout, and appearance parameters rather than 2D rendering artifacts and we establish the first benchmark for authenticity assessment in 3D Gaussian Splatting.

- We introduce Fake3DGS, a large-scale and balanced dataset of real and edited 3DGS scenes , where fake samples are generated with two complementary editors (GaussCtrl [181] and Instruct-GS2GS [167]) and evaluated under both mixed and cross-edit protocols to measure generalization to unseen manipulation pipelines.

- We propose a 3D detector operating directly on Gaussian primitives by adapting PointTransformerV3 to ingest Gaussian attributes and aggregate them into a scene-level embedding, and we demonstrate strong performance and robustness compared to 2D deepfake baselines especially under cross-edit generalization supported by a detailed ablation of Gaussian feature groups.

7.2 Fake3DGS Dataset

To evaluate 3D fake detection in a controlled yet realistic setting, we introduce Fake3DGS dataset, a large-scale benchmark of original and edited 3D Gaussian Splatting scenes. The dataset comprises more than 41k reconstructed scenes, evenly balanced between real and manipulated samples.

We export all reconstructions as `nerfstudio`[1] checkpoints, which provides a widely adopted and well-documented format for neural rendering research, making the dataset easier to use and reproduce with standard tooling. We then post-process each 3DGS model using the Gaussian splatting compression strategy

[1] `https://github.com/nerfstudio-project/nerfstudio`

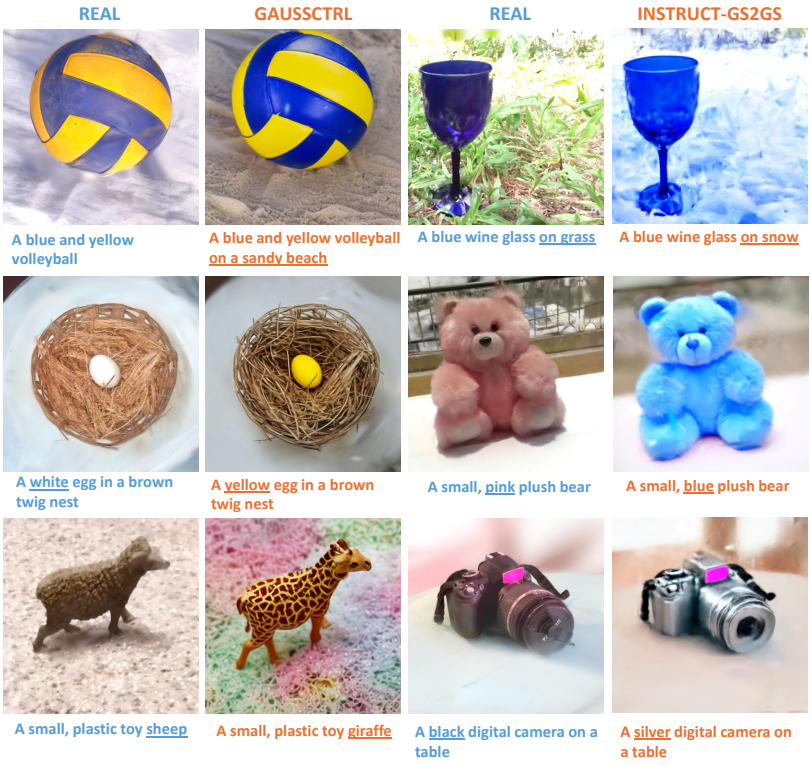


Figure 7.1: Some sample renderings of the data in our dataset. Below each view of the scene, the corresponding prompt used to generate edited samples is reported.

of Self-Organizing Gaussian Grids (SOGS) [115], leveraging the `gsplat`[2] implementation [190] that is integrated in the `nerfstudio` ecosystem. Concretely, we independently quantize Gaussian parameters with uniform scalar quantization (8-bit per parameter in our implementation). The quantized tensors are then reorganized into locally coherent 2D grids (via sorting) and encoded with lossless PNG compression, enabling exact recovery of the quantized values at load time. This post-processing substantially reduces storage and I/O overhead, shrinking

[2] `https://github.com/nerfstudio-project/gsplat`

the dataset footprint from approximately 7 TB to about 200 GB, which makes large-scale training and evaluation feasible on standard research hardware while preserving faithful reconstruction of the compressed representation.

Figure 7.1 shows representative examples from Fake3DGS dataset. For each underlying scene we report a real rendering and its edited counterpart generated with either GaussCtrl [181] or Instruct-GS2GS [168] (see Sect. 7.2.2). The edits cover the three instruction families used in our benchmark background/surface changes (*e.g.*, grass $\rightarrow$ snow), object appearance changes (*e.g.*, color), and object type/material substitutions (*e.g.*, sheep $\rightarrow$ giraffe) while keeping the overall scene visually plausible. Captions under each image indicate the text prompt used to produce the corresponding edited sample, highlighting that the resulting manipulations are often photorealistic and view consistent, making detection based solely on 2D artifacts challenging.

7.2.1 Real 3D Scenes

Building a real-world benchmark for our task requires processing a large number of scenes, each with multi-view imagery, accurate camera poses, and a stable 3D scene representation for evaluation. Collecting and reconstructing such data from scratch would be prohibitively time-consuming and would introduce additional variability due to the reconstruction pipeline itself (*e.g.*, pose estimation failures, incomplete coverage, or inconsistent scene quality). For this reason, we started from UCO3D [103], an existing large-scale dataset that already provided these prerequisites in a consistent format.

UCO3D contains a wide variety of objects, spanning more than 1000 categories, captured in diverse indoor and outdoor environments, and additionally provides pretrained 3D Gaussian Splatting reconstructions. Leveraging these ready-to-use reconstructions allows scaling the dataset construction to many scenes while keeping reconstruction quality consistent across categories, so that our evaluation primarily reflects the behavior of the editing method rather than artifacts of the reconstruction process.

To avoid biasing the benchmark toward categories with many instances, we balance the subset across categories. Specifically, for each category we select

$$n = \min_{c} \texttt{num\_objects}(c),$$

i.e., the minimum number of available instances across categories, and sample n objects per category.

7.2.2 Fake 3D Scenes Generation

As mentioned, we employed two different methods to edit real scenes.

The first one is GaussCtrl [181], which edits rendered images from 3DGS and recreates the edited scene through depth-conditioned editing based on ControlNet [196] for geometry consistency and attention-based latent code alignment for improving consistency during editing.

The second one is Instruct-GS2GS [168] which uses InstructPix2Pix [10] to iteratively edit the input images while optimizing the underlying scene, resulting in an optimized 3D scene edited accordingly to the instruction.

To construct our dataset, edited captions were automatically generated using an LLM, specifically the Meta-Llama-3-8B-Instruct [2]. We chose this model because it is open-source and can be deployed locally, allowing full control over generation parameters, reproducibility of the editing process, and large-scale caption generation without reliance on external APIs. For each original caption, we create one edited caption by the following procedure: (i) we randomly sample one prompt template from the three editing strategies below, and (ii) we append a fixed suffix that provides the input caption and enforces the output format (single caption only). The resulting full prompt is then fed to the LLM, and its output is used as the edited caption.

For clarity, we report the prompts, categories and suffix.

Prompt templates (randomly sample one):

1. *Modify the following sentence by changing only the background or supporting surface on which the main object stands, but do not change the color, the shape, or any other attribute of the main object.*
2. *Modify the following sentence by changing only the material or the type of the main object, but do not change the color, the background, or the shape.*
3. *Modify the following sentence by changing only the color of the main object, but do not change the shape or any other attribute of the main object, and do not change the background.*

Fixed suffix (always appended):

> *Sentence: caption. The output should be a single caption without any additional explanation or text.*

Three edited captions were generated for each scene, corresponding to the three editing attributes. For each scene, one caption was randomly selected and assigned, ensuring a balanced distribution of edit types across the dataset.

7.2.3 Benchmark

We adopt different dataset splitting strategies depending on the evaluation setting. In all cases, splits are performed at the edit level rather than enforcing a strict scene-level separation, allowing different edited versions of the same underlying 3D scene to appear across partitions.

1. **Mixed split.** In the mixed setting we perform an 80–20% train–test split, resulting in approximately 33k scenes for training and 8k scenes for testing. This setting evaluates overall fake detection performance in a classic deep learning setup.

2. **Cross-editing split.** To evaluate generalization across editing methods, we define two additional splits where the model is trained on fake scenes generated with one editing approach (GaussCtrl or Instruct-GS2GS) and tested on the other. This setup ensures that the manipulation method used at test time is unseen during training, reflecting a more realistic scenario in which newly developed editing techniques are deployed after the detection model has already been trained.

7.3 Experimental Results

7.3.1 Baselines

Since, to the best of our knowledge, no method specifically addresses fake detection for 3D Gaussian scenes, we adopt state-of-the-art 2D deepfake detectors as baselines and evaluate them on renderings of the corresponding 3DGS scenes. In addition to our method, we compare against established detectors designed to recognize synthetic images produced by modern generators. All baselines are fine-tuned on our training split (starting from the authors' released weights) to ensure a fair comparison under the same data distribution. Specifically, we include the models of Wang *et al.* [175], which build on a ResNet-50 (RN50) backbone [149] and investigate training strategies based on different image transformations to improve robustness and cross-generator generalization. Furthermore, we evaluate the reconstruction-based approach of Wang *et al.* [177], which detects synthetic content by comparing an input image to its reconstruction and analyzing the resulting residual, and we also include CoDE (Contrastive Deepfake Embeddings) [4], which learns a deepfake oriented embedding space via contrastive learning while

enforcing global/local agreement between full-image and cropped-view representations to capture both global cues and fine-grained artifacts. Finally, we include CLIP based baselines following Ojha *et al.* [123]: we load pretrained CLIP encoders [133] and train a linear classifier on top of the CLIP image embedding. In this setting, we report results for two CLIP variants (ViT-B/16 and ViT-L/14), finetuned on our training split under the same protocol as the other baselines. Additionally, we evaluate self-supervised visual backbones based on DINOv2 [125], by fine-tuning the corresponding pretrained representations with an identical linear classification head.

For all experiments involving 2D detectors, we render images from each 3D scene and fine-tune/evaluate the baselines on these renderings using the same split protocol described above.

7.3.2 Proposed method

We propose a 3D fake detector that operates directly on the Gaussian splatting representation. Our backbone is PointTransformerV3 [182], a transformer based architechture for point clouds that treats a point set as an unordered collection of 3D samples and learns features by *local self-attention*: for each point, the model aggregates information from nearby points in space. This design is well suited to our setting because a 3DGS scene can be naturally viewed as a set of primitives distributed in 3D, where the authenticity cues may depend on spatial context rather than on individual Gaussians in isolation. Inspired by the work of Wu et al.[182] and Li et al.[95] we applied some modifications to the PointTransformerV3 to enable the processing of Gaussian splats as input primitives instead of standard point clouds. These changes mainly concern the input feature representation, which is extended to incorporate the attributes associated with each Gaussian. Beyond these adaptations, we benefit from the inherent flexibility of PointTransformerV3, which naturally supports batching scenes with varying numbers of Gaussian splats. This property allows us to avoid padding or resampling strategies and enables efficient training and inference across heterogeneous scenes. Each gaussian is represented with the 3D coordinates of its mean point, and instead of the traditional color feature used for point clouds, we provide opacity, scale, quaternions, spherical harmonics s_h with the zeroth-order term s_0 representing the view-independent color component. We employ both the encoder and decoder stages of PointTransformerV3 to extract contextualized Gaussian-level features. Given that each 3D scene is represented by a variable number of Gaussian splats, we aggregate these gaussian-level features into a single scene-level representation using a global mean

Table 7.1: Fine-tuning results on Fake3DGS dataset under different train/test split protocols. We report overall accuracy together with class-wise accuracy on fake and real samples. For each backbone, we evaluate (i) cross-edit generalization by training on one editing method (GaussCtrl or Instruct-GS2GS) and testing on the other, and (ii) the mixed setting where training and testing use the combined data.

	Train		Test		Accuracy (%)		
Backbone	GaussCtrl	Instruct-GS2GS	GaussCtrl	Instruct-GS2GS	Overall	Fake	Real
CLIP ViT-B	✓			✓	80.7	70.6	95.9
CLIP ViT-B		✓	✓		74.3	41.5	98.5
CLIP ViT-B	✓	✓	✓	✓	84.0	84.2	93.8
CLIP ViT-L	✓			✓	81.5	67.1	96.6
CLIP ViT-L		✓	✓		75.4	44.5	98.2
CLIP ViT-L	✓	✓	✓	✓	84.0	83.7	94.3
DINOv2-B	✓			✓	76.8	72.7	89.2
DINOv2-B		✓	✓		73.2	45.9	93.3
DINOv2-B	✓	✓	✓	✓	87.8	89.7	85.8
CoDE	✓			✓	80.2	60.5	92.4
CoDE		✓	✓		79.3	54.1	95.8
CoDE	✓	✓	✓	✓	92.2	91.4	92.9
DM	✓			✓	83.8	74.8	95.8
DM		✓	✓		82.4	80.3	96.9
DM	✓	✓	✓	✓	90.4	89.3	97.2
UFD	✓			✓	79.5	69.4	91.9
UFD		✓	✓		78.2	59.5	95.4
UFD	✓	✓	✓	✓	89.9	90.6	89.2
Fake3DGS		✓	✓		98.3	98.0	98.5
Fake3DGS	✓			✓	98.7	99.1	98.4
Fake3DGS	✓	✓	✓	✓	**98.9**	**98.4**	**99.3**

pooling operation. Specifically, features belonging to the same scene are averaged based on their batch indices, producing a fixed-dimensional embedding per scene. These embeddings are then passed to a classification head for binary prediction.

7.3.3 Results

Table 7.1 summarizes the performance of 2D baselines and our 3D Gaussian-based detector under mixed and cross-edit protocols. In the mixed setting (both editors in train and test), most 2D detectors achieve reasonably high overall accuracy, with the strongest baseline reaching 92.2% (CoDE). However, when evaluated under cross-edit generalization, all 2D methods exhibit a pronounced drop. Importantly, this degradation is largely driven by failures on the *fake* class: e.g., CLIP and DINO variants can maintain high *Real* accuracy (often above 93–98%) while their

Fake accuracy collapses (down to 41.5-45.9% in several cases).

Notably, the strongest cross-edit baseline is DM trained on GaussCtrl and tested on Instruct-GS2GS ($G \rightarrow I$), achieving 83.8% overall accuracy. However, the class-wise results indicate that this performance is still driven by a much higher accuracy on real samples (95.8%) than on fake samples (74.8%). This gap suggests that, under an unseen editor, the detector remains conservative and tends to misclassify a substantial portion of edited scenes as real.

In contrast, our method reaches 98.7% on the same $G \rightarrow I$ protocol and remains well balanced across classes (99.1% on fake and 98.4% on real). Compared to the best cross-edit baseline, this corresponds to a +14.9 percentage points gain in overall accuracy, largely explained by a +24.3 pp improvement on the fake class (74.8% $\rightarrow$ 99.1%), while also improving real accuracy (+2.6 pp). These results support the hypothesis that 2D detectors tend to exploit editor-specific cues that do not transfer across manipulation pipelines, whereas operating directly on the 3DGS representation provides more editor-agnostic evidence for authenticity. However, it is important to note that despite the high accuracy achieved by the PointTransformer, these results do not imply that the detection of fake 3D scenes is a fully solved problem.

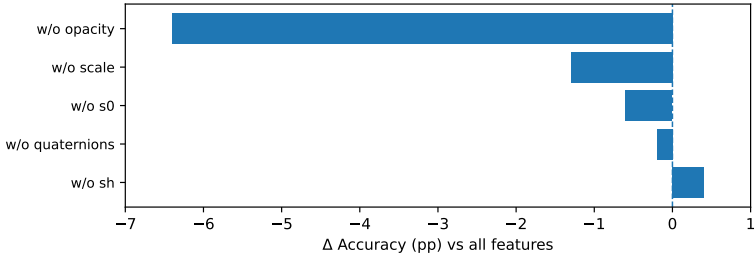


Figure 7.2: Ablation over Gaussian feature groups. We report the change in accuracy (in percentage points) obtained by removing each feature group from the input, relative to the full-feature model.

7.3.4 Ablation Study

As shown in Fig. 7.2, we conducted an ablation study to assess the contribution of each Gaussian attribute to the detection performance. We remove one feature group at a time from the input and report the resulting change in accuracy with respect

to the full model. Removing opacity caused the largest drop in accuracy (92.5%), indicating that transparency information is critical. Scale and the zeroth-order spherical harmonics coefficient s_0 provide a moderate contribution to detection performance. Scale captures geometric properties related to the spatial extent and distribution of Gaussian primitives, which can be affected by scene editing. The s_0 term encodes view-independent color information; however, its impact is limited due to redundancy with other attributes and the scene-level mean pooling operation. Quaternion-based orientation features exhibit minimal influence on performance, suggesting that Gaussian orientation remains largely consistent between real and fake scenes and is therefore less informative for this task and carry limited discriminative information. Higher-order SH coefficients had minimal impact suggesting that view-dependent appearance cues are largely irrelevant for this binary classification task. Despite removing s_h seems to improve the overall accuracy of 0.4, this was not the case in the cross setting, resulting in a lower accuracy w.r.t using s_h. We therefore retain s_h in all main experiments to favor robustness and generalization across editing methods.

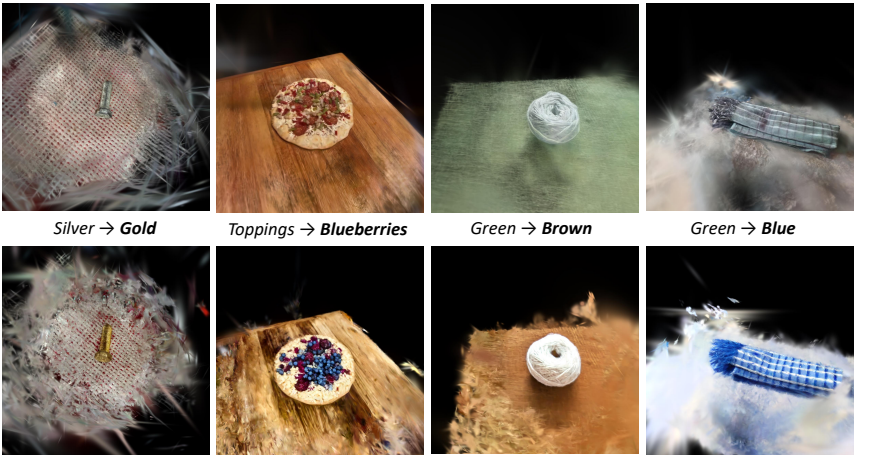


Figure 7.3: Some sample renderings of the data in our dataset. Below each image, there is the correspondent prompt used to generate edited samples.

7.4 Conclusions and Future Work

In this work, we introduced a novel 3D fake detection framework based on Gaussian splatting representations. We also presented Fake3DGS, a large-scale benchmark composed of over 40k real and edited 3D scenes generated using different editing methods. By leveraging Gaussian-level attributes and a modified PointTransformerV3 architecture, our approach effectively captures scene-level inconsistencies introduced by editing operations. Experimental results demonstrate that while 2D-based fake detection methods perform well when trained and tested on similar editing distributions, they struggle to generalize across unseen editing methods. In contrast, our 3D approach consistently outperforms 2D baselines, particularly in cross-edit evaluation settings, highlighting its robustness to editor-specific artefacts and its ability to generalize to novel manipulations. An extensive ablation study further shows that opacity and geometric attributes play a crucial role in detection, whereas view-dependent appearance cues contribute marginally to performance.

Future work will focus on evaluating the proposed framework on a broader range of 3D editing methods to further assess its generalization capabilities. Additionally, we plan to expand the dataset by explicitly annotating the spatial location and extent of edits within each scene, enabling more fine-grained tasks such as edit localization and region-level fake detection. We believe these directions will help advance the study of reliable and robust fake detection in 3D content generation pipelines.

Chapter 8

Conclusion

This thesis reports the research activity I carried out during my PhD, from problem formulation to method design and experimental validation, with the overarching goal of advancing 3D digitization and interpretation under real-world constraints. In particular, the thesis focuses on the reconstruction, representation, and analysis of 3D content in challenging conditions such as sparse viewpoints, reflective materials, imperfect camera poses, and limited ground-truth geometry and proposes solutions grounded in neural rendering and explicit 3D representations such as 3D Gaussian Splatting.

A recurring theme across the thesis is that, while neural rendering has made 3D reconstruction from images dramatically more accessible, practical deployment still faces critical bottlenecks. Pose accuracy remains a dominant factor affecting reconstruction quality; vehicle-centric scenarios introduce strong specularities and thin structures; and evaluation is often hindered by the lack of targeted benchmarks and geometry-aware metrics. In addition, the thesis addresses two complementary directions that are increasingly relevant to modern 3D vision systems: uncertainty-aware human-centric inference (through multi-hypothesis 3D gaze estimation) and trustworthiness of 3D content (through fake detection for edited 3D representations).

In the following, I summarize the main contributions presented in this thesis, outline promising future research directions, and highlight how these components jointly support more robust, efficient, and trustworthy 3D systems.

8.0.1 Summary of contributions

Keypoint-based camera pose refinement for vehicle reconstruction (KRONC). In Chapter 3, I introduced KRONC, a lightweight and effective camera pose refinement strategy tailored to vehicle reconstruction. KRONC leverages 2D semantic keypoints to optimize camera extrinsics (and auxiliary depth variables for keypoint back-projection), reducing reliance on expensive or fragile SfM refinements and improving downstream novel view synthesis quality. The method is designed to be plug-and-play, and its impact is demonstrated across multiple reconstruction backends, including Gaussian Splatting-based pipelines.

Sparse-view vehicle reconstruction with improved geometry recovery (BRUM and pipeline). Chapter 4 addressed the difficulties of reconstructing vehicles from limited and challenging captures, especially in outdoor settings. I proposed a reconstruction pipeline that augments the available views and replaces fragile components of classical pose estimation with more robust geometric priors, notably adopting DUSt3R in place of standard SfM and incorporating masking strategies to handle low-confidence regions and background clutter. To support evaluation in a domain where data is scarce, I also introduced the BRUM-dataset, including both synthetic and real public transportation vehicles, enabling systematic analysis of reconstruction quality under realistic conditions.

Diffusion-based multi-hypothesis 3D gaze estimation (GazeD). In Chapter 6, I tackled 3D gaze estimation as an inherently ambiguous problem and proposed GazeD, a diffusion-based approach that models uncertainty by generating multiple plausible gaze hypotheses and then aggregating them. The method leverages contextual cues including body pose and surrounding objects to improve robustness, and achieves strong performance across multiple benchmarks, highlighting the benefits of a sampling-based, multi-modal formulation for human-centric 3D interpretation.

CarPatch: a synthetic benchmark for radiance fields on vehicle components. Chapter 5 contributed to the evaluation ecosystem by presenting CarPatch, a synthetic dataset and benchmarking protocol designed to test neural rendering methods specifically in vehicle inspection settings. CarPatch provides RGB, depth, and pixel-wise semantic masks for multiple vehicle components, enabling analysis beyond global appearance metrics and supporting component-level, geometry aware

evaluation. The benchmark includes multiple training-set sizes to probe sparse-view behavior and introduces depth-based metrics (including surface-normal error) to better isolate geometric fidelity.

Fake3DGS and 3D-aware fake detection for edited Gaussian scenes. Finally, Chapter 7 addressed the emerging problem of 3D authenticity. I introduced Fake3DGS, a large-scale benchmark of real and edited 3D scenes built from consistent multi-view reconstructions and generated using modern editing pipelines (e.g., GaussCtrl and Instruct-GS2GS). Building on this dataset, I proposed a fake detection framework that operates directly on Gaussian attributes with a modified PointTransformerV3 backbone, showing strong robustness especially in cross-edit generalization compared to purely 2D-based detectors.

8.1 Future directions

The results in this thesis suggest several promising directions for future work.

- **Tighter integration between pose refinement and reconstruction.** KRONC shows that lightweight semantic structure can substantially improve pose quality. A natural next step is to integrate keypoint-driven constraints more deeply into Gaussian Splatting / NeRF training (e.g., joint optimization with stronger geometric priors and robustness to missing or noisy keypoints).

- **Geometry-first evaluation and training objectives for vehicles.** CarPatch highlights that similar image metrics can mask meaningful differences in 3D surface quality, especially on challenging components (mirrors, lights, windows). Extending these ideas to real captures, through stronger depth supervision, better normal estimation, or hybrid mesh/field evaluation could improve reliability in safety-critical inspection tasks.

- **Bridging synthetic-to-real in vehicle-centric neural rendering.** Both BRUM and CarPatch enable controlled testing, but real deployments require robustness to domain shifts (illumination, weather, sensor artifacts). Future work could explore domain adaptation strategies, better material modeling, and data generation pipelines that preserve the failure modes observed in real-world captures.

- **Richer uncertainty modeling for human-centric 3D understanding.** GazeD demonstrates the value of multi-hypothesis prediction. Future research could incorporate temporal reasoning, interaction-aware cues, and richer 3D scene semantics (e.g., explicit gaze targets and affordances) to move from direction estimation toward grounded attention understanding in complex environments.
- **Fine-grained 3D forensics: from detection to localization and provenance.** The fake detection results indicate that 3D representations expose inconsistencies that generalize across editors. A key next step is enabling edit localization and region-level attribution by enriching datasets with spatial annotations of manipulations and expanding coverage to a broader range of editing methods and attack settings.

Publications

1. D. Di Nucci, A. Simoni, M. Tomei, L. Ciuffreda, R. Vezzani, R. Cucchiara, *CarPatch: A Synthetic Benchmark for Radiance Field Evaluation on Vehicle Components*, Proceedings of the 22nd International Conference on Image Analysis and Processing (ICIAP), Udine, Italy, 2023.
2. D. Di Nucci, A. Simoni, M. Tomei, L. Ciuffreda, R. Vezzani, R. Cucchiara, *KRONC: Keypoint-based Robust Camera Optimization for 3D Car Reconstruction*, Proceedings of the European Conference on Computer Vision Workshops (ECCVW), Milan, Italy, 2024.
3. D. Di Nucci, M. Tomei, G. Borghi, L. Ciuffreda, R. Vezzani, R. Cucchiara, *BRUM: Robust 3D Vehicle Reconstruction from 360° Sparse Images*, Proceedings of the IEEE Intelligent Vehicles Symposium (IV), Cluj-Napoca, Romania, 2025.
4. R. Catalini, D. Di Nucci, G. Borghi, D. Davoli, L. Garattoni, G. Francesca, Y. Kawana, R. Vezzani, *GazeD: Context-Aware Diffusion for Accurate 3D Gaze Estimation*, International Conference on 3D Vision (3DV), Vancouver, Canada, 2026.

Bibliography

[1] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building rome in a day. *Communications of the ACM*, 54(10), 2011. 5

[2] AI@Meta. Llama 3 model card, 2024. 87

[3] Hangbo Bao, Wenhui Wang, Li Dong, Qiang Liu, Owais Khan Mohammed, Kriti Aggarwal, Subhojit Som, Songhao Piao, and Furu Wei. Vlmo: Unified vision-language pre-training with mixture-of-modality-experts. *Advances in Neural Information Processing Systems*, 35, 2022. 65

[4] Lorenzo Baraldi, Federico Cocchi, Marcella Cornia, Lorenzo Baraldi, Alessandro Nicolosi, and Rita Cucchiara. Contrasting Deepfakes Diffusion via Contrastive Learning and Global-Local Similarities. In *Proceedings of the European Conference on Computer Vision*, 2024. 8, 83, 88

[5] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proc. of the IEEE/CVF ICCV*, 2021. 6

[6] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 6, 14

[7] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. *arXiv:2304.06706*, 2023. 6

[8] Wenjing Bian, Zirui Wang, Kejie Li, Jia-Wang Bian, and Victor Adrian Prisacariu. Nope-nerf: Optimising neural radiance field with no pose prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 6

[9] Christopher M Bishop. Mixture density networks. 1994. 8

[10] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18392–18402, 2023. 2, 9, 87

[11] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual-inertial, and multimap slam. *IEEE Transactions on Robotics*, 2021. 5

[12] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 76

[13] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE transactions on pattern analysis and machine intelligence*, 43(1), 2019. 71

[14] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7291–7299, 2017. 63

[15] Jiazhong Cen, Zanwei Zhou, Jiemin Fang, Wei Shen, Lingxi Xie, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, et al. Segment anything in 3d with nerfs. In *NeurIPS*, 2023. 11

[16] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *Proceedings of the European Conference on Computer Vision*. Springer, 2022. 6, 20, 21, 52, 53, 54

[17] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation.

In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7103–7112, 2018. 63

[18] Yiwen Chen, Zilong Chen, Chi Zhang, Feng Wang, Xiaofeng Yang, Yikai Wang, Zhongang Cai, Lei Yang, Huaping Liu, and Guosheng Lin. Gaussianeditor: Swift and controllable 3d editing with gaussian splatting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2024. 9

[19] Yue Chen, Xingyu Chen, Xuan Wang, Qi Zhang, Yu Guo, Ying Shan, and Fei Wang. Local-to-global registration for bundle-adjusting neural radiance fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 6, 13, 19, 20, 23, 27, 28

[20] Zixuan Chen, Guangcong Wang, Jiahao Zhu, Jianhuang Lai, and Xiaohua Xie. Guardsplat: Efficient and robust watermarking for 3d gaussian splatting. In *CVPR*, 2025. 9

[21] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 15

[22] Yihua Cheng, Feng Lu, and Xucong Zhang. Appearance-based gaze estimation via evaluation-guided asymmetric regression. In *Proceedings of the European Conference on Computer Vision*, 2018. 59, 62

[23] Yihua Cheng, Haofei Wang, Yiwei Bao, and Feng Lu. Appearance-based gaze estimation with deep learning: A review and benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 8, 62

[24] Jeongjun Choi, Dongseok Shim, and H. Jin Kim. Diffupose: Monocular 3d human pose estimation via denoising diffusion probabilistic model. pages 3773–3780, 2023. 63, 72, 75

[25] Eunji Chong, Yongxin Wang, Nataniel Ruiz, and James M Rehg. Detecting attended visual targets in video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 69, 71

[26] Abril Corona-Figueroa, Jonathan Frawley, Sam Bond-Taylor, Sarath Bethapudi, Hubert PH Shum, and Chris G Willcocks. Mednerf: Medical neural radiance fields for reconstructing 3d-aware ct-projections from a

single x-ray. In *Annual International Conference of the IEEE Engineering in Medicine & Biology Society*. IEEE, 2022. 11

[27] Riccardo Corvi, Davide Cozzolino, Giovanni Poggi, Koki Nagano, and Luisa Verdoliva. Intriguing properties of synthetic images: from generative adversarial networks to diffusion models. In *CVPR*, 2023. 8

[28] Riccardo Corvi, Davide Cozzolino, Giada Zingarini, Giovanni Poggi, Koki Nagano, and Luisa Verdoliva. On the detection of synthetic images generated by diffusion models. In *ICASSP*, 2023. 8

[29] Davide Cozzolino, Justus Thies, Andreas Rössler, Christian Riess, Matthias Nießner, and Luisa Verdoliva. Forensictransfer: Weakly-supervised domain adaptation for forgery detection. *arXiv preprint arXiv:1812.02510*, 2018. 8

[30] Hao Dang, Feng Liu, Joel Stehouwer, Xiaoming Liu, and Anil K Jain. On the detection of digital face manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern recognition*, pages 5781–5790, 2020. 83

[31] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *Advances in Neural Information Processing Systems*, volume 34, pages 8780–8794, 2021. 8

[32] Davide Di Nucci, Alessandro Simoni, Matteo Tomei, Luca Ciuffreda, Roberto Vezzani, and Rita Cucchiara. Carpatch: A synthetic benchmark for radiance field evaluation on vehicle components. In *Proceedings of the International Conference on Image Analysis and Processing*. Springer, 2023. 3

[33] Davide Di Nucci, Alessandro Simoni, Matteo Tomei, Luca Ciuffreda, Roberto Vezzani, and Rita Cucchiara. Carpatch: A synthetic benchmark for radiance field evaluation on vehicle components. In *International Conference on Image Analysis and Processing*. Springer, 2023. 7, 11, 12, 19, 27, 33, 34, 42

[34] Davide Di Nucci, Alessandro Simoni, Matteo Tomei, Luca Ciuffreda, Roberto Vezzani, and Rita Cucchiara. Kronc: Keypoint-based robust camera optimization for 3d car reconstruction. In *Proceedings of the European Conference on Computer Vision Workshops*, 2024. 3

[35] Davide Di Nucci, Alessandro Simoni, Matteo Tomei, Luca Ciuffreda, Roberto Vezzani, and Rita Cucchiara. Kronc: Keypoint-based robust camera optimization for 3d car reconstruction. In *Proceedings of the European Conference on Computer Vision Workshops*, 2024. 6, 33, 34, 44

[36] Philipe Ambrozio Dias, Damiano Malafronte, Henry Medeiros, and Francesca Odone. Gaze estimation for assisted living environments. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2020. 65, 71

[37] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv:2010.11929*, 2020. 40

[38] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*. PMLR, 2017. 7, 12

[39] Maria K Eckstein, Belén Guerra-Carrillo, Alison T Miller Singley, and Silvia A Bunge. Beyond eye gaze: What else can eyetracking reveal about cognition and cognitive development? *Developmental cognitive neuroscience*, 25, 2017. 59

[40] Zhiwen Fan, Kevin Wang, Kairun Wen, Zehao Zhu, Dejia Xu, and Zhangyang Wang. Lightgaussian: Unbounded 3d gaussian compression with 15x reduction and 200+ fps. In *Advances in Neural Information Processing Systems*, 2024. 7

[41] Hany Farid. Lighting (in) consistency of paint by text. *arXiv preprint arXiv:2207.13744*, 2022. 9

[42] Hany Farid. Perspective (in) consistency of paint by text. *arXiv preprint arXiv:2206.14617*, 2022. 9

[43] Tobias Fischer, Hyung Jin Chang, and Yiannis Demiris. Rt-gene: Real-time eye gaze estimation in natural environments. In *Proceedings of the European Conference on Computer Vision*, 2018. 8, 62, 68, 69, 71, 76

[44] MA FISCHLER AND. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 1981. 5

[45] Joel Frank, Thorsten Eisenhofer, Lea Schönherr, Asja Fischer, Dorothea Kolossa, and Thorsten Holz. Leveraging frequency analysis for deep fake image recognition. In *ICML*, 2020. 8

[46] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proc. of the IEEE/CVF Conf. on CVPR*, 2022. 6

[47] Kenneth Alberto Funes Mora, Florent Monay, and Jean-Marc Odobez. Eyediap: A database for the development and evaluation of gaze estimation algorithms from rgb and rgb-d cameras. In *Proceedings of the symposium on eye tracking research and applications*, 2014. 59, 61, 62

[48] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012. 7, 12

[49] Jia Gong, Lin Geng Foo, Zhipeng Fan, Qiuhong Ke, Hossein Rahmani, and Jun Liu. Diffpose: Toward more reliable 3d pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 13041–13051, 2023. 8, 63, 72, 75

[50] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 83

[51] Diego Gragnaniello, Davide Cozzolino, Francesco Marra, Giovanni Poggi, and Luisa Verdoliva. Are gan generated images easy to detect? a critical analysis of the state-of-the-art. *ICME*, 2021. 8

[52] Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, et al. Ego-exo4d: Understanding skilled human activity from first-and third-person perspectives. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2024. 68, 76

[53] Jian Guan, Liming Yin, Jianguo Sun, Shuhan Qi, Xuan Wang, and Qing Liao. Enhanced gaze following via object detection and human pose estimation. In *MultiMedia Modeling: 26th International Conference, MMM*

2020, Daejeon, South Korea, January 5–8, 2020, Proceedings, Part II 26. Springer, 2020. 59

[54] Yiran Guan, Zhuoguang Chen, Wenzheng Zeng, Zhiguo Cao, and Yang Xiao. End-to-end video gaze estimation via capturing head-face-eye spatial-temporal interaction context. *IEEE Signal Processing Letters*, 30, 2023. 61

[55] Zidong Guo, Zejian Yuan, Chong Zhang, Wanchao Chi, Yonggen Ling, and Shenghao Zhang. Domain adaptation gaze estimation by embedding with prediction consistency. In *Proceedings of the Asian Conference on Computer Vision*, 2020. 59

[56] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 5

[57] Runze He, Shaofei Huang, Xuecheng Nie, Tianrui Hui, Luoqi Liu, Jiao Dai, Jizhong Han, Guanbin Li, and Si Liu. Customize your nerf: Adaptive source driven 3d scene editing via local-global iterative training. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6966–6975, 2024. 9, 83

[58] Craig Hennessey, Borna Noureddin, and Peter Lawrence. A single camera eye-gaze tracking system with free head motion. In *Proceedings of the 2006 symposium on Eye tracking research & applications*, 2006. 62

[59] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 2, 3, 8

[60] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 63

[61] Karl Holmquist and Bastian Wandt. Diffpose: Multi-hypothesis human pose estimation using diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15977–15987, October 2023. 8

[62] Kunlong Hong, Hongguang Wang, and Bingbing Yuan. Inspection-nerf: Rendering multi-type local images for dam surface inspection task using climbing robot and neural radiance field. *Buildings*, 13(1), 2023. 11

[63] Mir Rayat Imtiaz Hossain and James J Little. Exploiting temporal information for 3d human pose estimation. In *Proceedings of the European Conference on Computer Vision*, pages 68–84, 2018. 72

[64] Benran Hu, Junkai Huang, Yichen Liu, Yu-Wing Tai, and Chi-Keung Tang. Nerf-rpn: A general framework for object detection in nerfs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 11

[65] Xu Hu, Yuxi Wang, Lue Fan, Junsong Fan, Junran Peng, Zhen Lei, Qing Li, and Zhaoxiang Zhang. Semantic anything in 3d gaussians. *arXiv:2401.17857*, 2024. 7

[66] Zhengxi Hu, Dingye Yang, Shilei Cheng, Lei Zhou, Shichao Wu, and Jingtai Liu. We know where they are looking at from the rgb-d camera: Gaze following in 3d. *IEEE Transactions on Instrumentation and Measurement*, 71, 2022. 8, 62, 71

[67] Zhengxi Hu, Yuxue Yang, Xiaolin Zhai, Dingye Yang, Bohan Zhou, and Jingtai Liu. Gfie: A dataset and baseline for gaze-following from 2d to 3d in indoor environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 8, 61, 62, 68, 69, 71, 72, 73, 75, 76, 81

[68] Zhiming Hu, Jiahui Xu, Syn Schmitt, and Andreas Bulling. Pose2gaze: Eye-body coordination during daily activities for gaze prediction from full-body poses. *IEEE Transactions on Visualization and Computer Graphics*, 2024. 8, 60

[69] Xiufeng Huang, Ziyuan Luo, Qi Song, Ruofei Wang, and Renjie Wan. Marksplatter: Generalizable watermarking for 3d gaussian splatting model via splatter image structure. 2025. 9

[70] Matthew Hull, Haoyang Yang, Pratham Mehta, Mansi Phute, Aeree Cho, Haoran Wang, Matthew Lau, Wenke Lee, Willian T. Lunardi, Martin Andreoni, and Polo Chau. 3d gaussian splat vulnerabilities, 2025. 9

[71] Sumin In, Youngdong Jang, Utae Jeong, MinHyuk Jang, Hyeongcheol Park, Eunbyung Park, and Sangpil Kim. Compmarkgs: Robust watermarking for compressed 3d gaussian splatting, 2025. 9

[72] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, 2013. 74

[73] Youngdong Jang, Hyunje Park, Feng Yang, Heeju Ko, Euijin Choo, and Sangpil Kim. 3d-gsw: 3d gaussian splatting for robust watermarking. In *CVPR*, 2025. 9

[74] Yoonwoo Jeong, Seokjun Ahn, Christopher Choy, Anima Anandkumar, Minsu Cho, and Jaesik Park. Self-calibrating neural radiance fields. In *Proceedings of the IEEE International Conference on Computer Vision*, 2021. 6, 19

[75] Ying Jiang, Chang Yu, Tianyi Xie, Xuan Li, Yutao Feng, Huamin Wang, Minchen Li, Henry Y. K. Lau, Feng Gao, Yin Yang, and Chenfanfu Jiang. Vr-gs: A physical dynamics-aware interactive gaussian splatting system in virtual reality. In *ACM TOG*, 2024. 83

[76] Swati Jindal, Mohit Yadav, and Roberto Manduchi. Spatio-temporal attention and gaussian processes for personalized video gaze estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2024. 61

[77] Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social motion capture. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 75

[78] Petr Kellnhofer, Adria Recasens, Simon Stent, Wojciech Matusik, and Antonio Torralba. Gaze360: Physically unconstrained gaze estimation in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019. 8, 59, 65, 68, 69, 71, 72, 76, 81

[79] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuhler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 2023. 1, 7

[80] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM TOG*, 2023. 11, 20, 22, 23, 28, 33, 41, 42

[81] Jess Kerr-Gaffney, Amy Harrison, and Kate Tchanturia. Eye-tracking research in eating disorders: A systematic review. *International Journal of Eating Disorders*, 52(1), 2019. 59

[82] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *International conference on machine learning*. PMLR, 2021. 65

[83] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 70

[84] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 83

[85] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 15

[86] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 41

[87] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 2017. 5, 14

[88] Sven Kreiss, Lorenzo Bertoni, and Alexandre Alahi. OpenPifPaf: Composite Fields for Semantic Keypoint Detection and Spatio-Temporal Association. *IEEE Transactions on Intelligent Transportation Systems*, 2021. 15, 22, 24

[89] Yixing Lao, Xiaogang Xu, Xihui Liu, Hengshuang Zhao, et al. Corresnerf: Image correspondence priors for neural radiance fields. In *Advances in Neural Information Processing Systems*, 2023. 6

[90] Yixing Lao, Xiaogang Xu, Xihui Liu, Hengshuang Zhao, et al. Corresnerf: Image correspondence priors for neural radiance fields. In *Advances in Neural Information Processing Systems*, 2023. 13, 16

[91] Ji Woo Lee, Chul Woo Cho, Kwang Yong Shin, Eui Chul Lee, and Kang Ryoung Park. 3d gaze tracking method using purkinje images on eye optical model and pupil. *Optics and Lasers in Engineering*, 50(5), 2012. 62

[92] Chen Li and Gim Hee Lee. Generating multiple hypotheses for 3d human pose estimation with mixture density network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9887–9895, 2019. 8

[93] Jiahe Li, Jiawei Zhang, Xiao Bai, Jin Zheng, Xin Ning, Jun Zhou, and Lin Gu. Dngaussian: Optimizing sparse-view 3d gaussian radiance fields with global-local depth normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2024. 7, 41, 42

[94] Wenhao Li, Hong Liu, Hao Tang, Pichao Wang, and Luc Van Gool. Mhformer: Multi-hypothesis transformer for 3d human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 13147–13156, 2022. 8, 63, 72

[95] Yue Li, Qi Ma, Runyi Yang, Huapeng Li, Mengjiao Ma, Bin Ren, Nikola Popovic, Nicu Sebe, Ender Konukoglu, Theo Gevers, et al. Scenesplat: Gaussian splatting-based scene understanding with vision-language pretraining. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025. 89

[96] Yuezun Li, Xin Yang, Pu Sun, Honggang Qi, and Siwei Lyu. Celeb-df: A large-scale challenging dataset for deepfake forensics. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 8, 83

[97] Dongze Lian, Zehao Yu, and Shenghua Gao. Believe it or not, we know what you are looking at! In *Asian Conference on Computer Vision*. Springer, 2018. 69, 71

[98] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *Proceedings of the IEEE International Conference on Computer Vision*, 2021. 6, 13, 19, 20, 27, 28

[99] Chen-Hsuan Lin, Oliver Wang, Bryan C Russell, Eli Shechtman, Vladimir G Kim, Matthew Fisher, and Simon Lucey. Photometric mesh optimization

for video-aligned 3d object reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 20

[100] Jeffrey Yunfan Liu, Yun Chen, Ze Yang, Jingkang Wang, Sivabalan Manivasagam, and Raquel Urtasun. Real-time neural rasterization for large scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 11

[101] Jia-Wei Liu, Yan-Pei Cao, Tianyuan Yang, Zhongcong Xu, Jussi Keppo, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Hosnerf: Dynamic human-object-scene neural radiance fields from a single video. In *Proceedings of the IEEE International Conference on Computer Vision*, 2023. 11

[102] Kenkun Liu, Rongqi Ding, Zhiming Zou, Le Wang, and Wei Tang. A comprehensive study of weight sharing in graph networks for 3d human pose estimation. In *Proceedings of the European Conference on Computer Vision*, pages 318–334, 2020. 72

[103] Xingchen Liu, Piyush Tayal, Jianyuan Wang, Jesus Zarzar, Tom Monnier, Konstantinos Tertikas, Jiali Duan, Antoine Toisoul, Jason Y Zhang, Natalia Neverova, et al. Uncommon objects in 3d. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 14102–14113, 2025. 86

[104] Yang Liu, Chuanchen Luo, Lue Fan, Naiyan Wang, Junran Peng, and Zhaoxiang Zhang. Citygaussian: Real-time high-quality large-scale scene rendering with gaussians. In *Proceedings of the European Conference on Computer Vision*, 2024. 34

[105] Ying-Tian Liu, Yuan-Chen Guo, Guan Luo, Heyi Sun, Wei Yin, and Song-Hai Zhang. Pi3d: Efficient text-to-3d generation with pseudo-image diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19915–19924, 2024. 83

[106] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE International Conference on Computer Vision*, 2021. 15

[107] Feng Lu, Yue Gao, and Xiaowu Chen. Estimating 3d gaze directions using unlabeled eye images via synthetic iris appearance fitting. *IEEE Transactions on Multimedia*, 18(9), 2016. 62

[108] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. *Advances in neural information processing systems*, 29, 2016. 64

[109] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European Conference on Computer Vision*, 2018. 15

[110] Julieta Martinez, Rayat Hossain, Javier Romero, and James J Little. A simple yet effective baseline for 3d human pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2640–2649, 2017. 8, 72

[111] Falko Matern, Christian Riess, and Marc Stamminger. Exploiting visual artifacts to expose deepfakes and face manipulations. In *2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*. IEEE, 2019. 8

[112] Marko Mihajlovic, Sergey Prokudin, Siyu Tang, Robert Maier, Federica Bogo, Tony Tung, and Edmond Boyer. Splatfields: Neural gaussian splats for sparse 3d and 4d reconstruction. In *Proceedings of the European Conference on Computer Vision*, 2024. 7, 41, 42

[113] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics*, 2019. 6, 14, 23

[114] 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. *Communications of the ACM*, 65(1), 2021. 1, 5, 11, 15, 33, 36, 49

[115] Wieland Morgenstern, Florian Barthel, Anna Hilsmann, and Peter Eisert. Compact 3d scene representation via self-organizing gaussian grids. In *ECCV*, 2024. 7, 85

[116] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics*, 2022. 6, 12, 21, 52, 53, 54, 55

[117] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5), 2015. 5

[118] Atsushi Nakazawa and Christian Nitschke. Point of gaze estimation through corneal surface reflection in an active illumination environment. In *Proceedings of the European Conference on Computer Vision*. Springer, 2012. 62

[119] Simon Niedermayr, Josef Stumpfegger, and Rüdiger Westermann. Compressed 3d gaussian splatting for accelerated novel view synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2024. 7

[120] Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 42

[121] Soma Nonaka, Shohei Nobuhara, and Ko Nishino. Dynamic 3d gaze from afar: Deep gaze estimation from temporal eye-head-body coordination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 8, 60, 61, 62, 65, 68, 70, 71, 72, 73, 75, 76

[122] Tuomas Oikarinen, Daniel Hannah, and Sohrob Kazerounian. Graphmdn: Leveraging graph structure and deep learning to solve inverse problems. In *IJCNN*, pages 1–9, 2021. 63

[123] Utkarsh Ojha, Yuheng Li, and Yong Jae Lee. Towards universal fake image detectors that generalize across generative models. In *CVPR*, 2023. 8, 89

[124] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 24

[125] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal,

Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research (TMLR)*, 2024. 89

[126] Anwesan Pal, Sayan Mondal, and Henrik I Christensen. "looking at the right stuff"-guided semantic-gaze for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 59

[127] Oskar Palinko, Francesco Rea, Giulio Sandini, and Alessandra Sciutti. Eye gaze tracking for a humanoid robot. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2015. 59

[128] Avinash Paliwal, Wei Ye, Jinhui Xiong, Dmytro Kotovenko, Rakesh Ranjan, Vikas Chandra, and Nima Khademi Kalantari. Coherentgs: Sparse novel view synthesis with coherent 3d gaussians. In *Proceedings of the European Conference on Computer Vision*, 2024. 7

[129] Cristina Palmero, Javier Selva, Mohammad Ali Bagheri, and Sergio Escalera. Recurrent cnn for 3d gaze estimation using appearance and shape cues. *arXiv preprint arXiv:1805.03064*, 2018. 62

[130] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7753–7762, 2019. 8, 72

[131] Rui Peng, Wangze Xu, Luyang Tang, Liwei Liao, Jianbo Jiao, and Ronggang Wang. Structure consistent gaussian splatting with matching prior for few-shot novel view synthesis. In *Advances in Neural Information Processing Systems*, 2024. 7

[132] Stefano Pini, Guido Borghi, Roberto Vezzani, Davide Maltoni, and Rita Cucchiara. A systematic comparison of depth map representations for face recognition. *Sensors*, 21(3), 2021. 54

[133] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the*

38th International Conference on Machine Learning (ICML), volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763, 2021. 89

[134] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, 2021. 43

[135] George E Raptis, Christina Katsini, Marios Belk, Christos Fidas, George Samaras, and Nikolaos Avouris. Using eye gaze data and visual activities to infer human cognitive styles: method and feasibility studies. In *proceedings of the 25th conference on user modeling, Adaptation and Personalization*, 2017. 59

[136] Adria Recasens, Aditya Khosla, Carl Vondrick, and Antonio Torralba. Where are they looking? *Advances in neural information processing systems*, 28, 2015. 8, 69, 71

[137] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bjorn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 2, 3

[138] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 83

[139] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to detect manipulated facial images. In *CVPR*, 2019. 8

[140] István Sárándi, Alexander Hermans, and Bastian Leibe. Learning 3D human pose estimation from dozens of datasets using a geometry-aware autoencoder to bridge between skeleton formats. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2023. 76

[141] Hamed Sarbolandi, Damien Lefloch, and Andreas Kolb. Kinect range sensing: Structured-light versus time-of-flight kinect. *Computer vision and image understanding*, 139, 2015. 62

[142] Johannes L. Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 1, 3, 5

[143] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 12, 16, 34, 40

[144] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *Proceedings of the European Conference on Computer Vision*, 2016. 5, 15

[145] Zeyang Sha, Zheng Li, Ning Yu, and Yang Zhang. De-fake: Detection and attribution of fake images generated by text-to-image generation models. In *ACM*, 2023. 8

[146] Wenkang Shan, Zhenhua Liu, Xinfeng Zhang, Zhao Wang, Kai Han, Shanshe Wang, Siwei Ma, and Wen Gao. Diffusion-based 3d human pose estimation with multi-hypothesis aggregation. In *Proceedings of the IEEE International Conference on Computer Vision*, 2023. 8, 67

[147] Anjana Sharma and Pawanesh Abrol. Eye gaze techniques for human computer interaction: A research survey. *International Journal of Computer Applications*, 71(9), 2013. 59

[148] Saurabh Sharma, Pavan Teja Varigonda, Prashast Bindal, Abhishek Sharma, and Arjun Jain. Monocular 3d human pose estimation by generation and ordinal ranking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2325–2334, 2019. 63, 67, 74

[149] Zhenning Shi, haoshuai zheng, Chen Xu, Changsheng Dong, Bin Pan, Xie xueshuo, Along He, Tao Li, and Huazhu Fu. Resfusion: Denoising diffusion probabilistic models for image restoration based on prior residual noise. In *NeurIPS*, 2024. 88

[150] Ken Shoemake. Animating rotation with quaternion curves. In *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pages 245–254, 1985. 37

[151] Edgar Simo-Serra, Arnau Ramisa, Guillem Alenya, Carme Torras, and Francesc Moreno-Noguer. Single image 3d human pose estimation from noisy observations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2673–2680, 2012. 63

[152] Vincent Sitzmann, Ana Serrano, Amy Pavel, Maneesh Agrawala, Diego Gutierrez, Belen Masia, and Gordon Wetzstein. Saliency in vr: How do people explore virtual environments? *IEEE transactions on visualization and computer graphics*, 24(4), 2018. 59

[153] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. *ACM Transactions on Graphics*, 2006. 5

[154] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*. 66

[155] Xibin Song, Peng Wang, Dingfu Zhou, Rui Zhu, Chenye Guan, Yuchao Dai, Hao Su, Hongdong Li, and Ruigang Yang. Apollocar3d: A large 3d car instance understanding benchmark for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 7, 12, 15, 16, 20

[156] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 6, 12, 20, 21, 52, 53, 54

[157] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5693–5703, 2019. 64, 68

[158] Xiao Sun, Jiaxiang Shang, Shuang Liang, and Yichen Wei. Compositional human pose regression. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2602–2611, 2017. 72

[159] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Justin Kerr, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David McAllister, and Angjoo Kanazawa. Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH 2023 Conference Proceedings*, 2023. 13, 21

[160] Andrea Toaiari, Vittorio Murino, Marco Cristani, and Cigdem Beyan. Upper-body pose-based gaze estimation for privacy-preserving 3d gaze

target detection. *arXiv preprint arXiv:2409.17886*, 2024. 59, 60, 61, 62, 69, 71

[161] Adam Tonderski, Carl Lindström, Georg Hess, William Ljungbergh, Lennart Svensson, and Christoffer Petersson. Neurad: Neural rendering for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2024. 7, 33

[162] Francesco Tonini, Nicola Dall'Asen, Cigdem Beyan, and Elisa Ricci. Object-aware gaze target detection. In *Proceedings of the IEEE International Conference on Computer Vision*, 2023. 8, 59

[163] Prune Truong, Marie-Julie Rakotosaona, Fabian Manhardt, and Federico Tombari. Sparf: Neural radiance fields from sparse and noisy poses. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 6, 13, 16, 19, 25, 34

[164] Danyang Tu, Xiongkuo Min, Huiyu Duan, Guodong Guo, Guangtao Zhai, and Wei Shen. End-to-end human-gaze-target detection with transformers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2022. 8

[165] Haithem Turki, Deva Ramanan, and Mahadev Satyanarayanan. Mega-nerf: Scalable construction of large-scale nerfs for virtual fly-throughs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 6, 34

[166] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 68

[167] Cyrus Vachha and Ayaan Haque. Instruct-gs2gs: Editing 3d gaussian splats with instructions, 2024. 2, 3, 9, 84

[168] Cyrus Vachha and Ayaan Haque. Instruct-gs2gs: Editing 3d gaussian splats with instructions, 2024. 86, 87

[169] Roberto Valenti, Nicu Sebe, and Theo Gevers. Combining head pose and eye location information for gaze estimation. *IEEE Transactions on Image Processing*, 21(2), 2011. 62

[170] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017. 40

[171] Tuan-Hung Vu, Anton Osokin, and Ivan Laptev. Context-aware cnns for person head detection. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015. 77

[172] Ulas Vural and Yusuf Sinan Akgul. Eye-gaze based real-time surveillance video synopsis. *Pattern Recognition Letters*, 30(12), 2009. 59

[173] Jingbo Wang, Sijie Yan, Yuanjun Xiong, and Dahua Lin. Motion guided 3d pose estimation from videos. In *Proceedings of the European Conference on Computer Vision*, pages 764–780, 2020. 8, 72

[174] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 6

[175] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. Cnn-generated images are surprisingly easy to spot... for now. In *CVPR*, 2020. 8, 88

[176] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2024. 3, 34, 40

[177] Zhendong Wang, Jianmin Bao, Wengang Zhou, Weilun Wang, Hezhen Hu, Hong Chen, and Houqiang Li. Dire for diffusion-generated image detection. In *ICCV*, 2023. 8, 83, 88

[178] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4), 2004. 19, 39, 42, 53

[179] 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. 6

[180] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. Synsin: End-to-end view synthesis from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 38

[181] Jing Wu, Jia-Wang Bian, Xinghui Li, Guangrun Wang, Ian Reid, Philip Torr, and Victor Adrian Prisacariu. Gaussctrl: Multi-view consistent text-driven 3d gaussian splatting editing. In *European Conference on Computer Vision*, pages 55–71. Springer, 2024. 2, 3, 9, 83, 84, 86, 87

[182] Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler, faster, stronger. In *CVPR*, 2024. 89

[183] Yitong Xia, Hao Tang, Radu Timofte, and Luc Van Gool. Sinerf: Sinusoidal neural radiance fields for joint pose estimation and scene reconstruction. In *British Machine Vision Conference*, 2022. 6

[184] Haozhe Xie, Zhaoxi Chen, Fangzhou Hong, and Ziwei Liu. Citydreamer: Compositional generative model of unbounded 3d cities. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2024. 34

[185] Ziyang Xie, Junge Zhang, Wenye Li, Feihu Zhang, and Li Zhang. S-nerf: Neural radiance fields for street views. In *Proceedings of the International Conference on Learning Representations*, 2023. 7, 11

[186] Jiale Xu, Xintao Wang, Weihao Cheng, Yan-Pei Cao, Ying Shan, Xiaohu Qie, and Shenghua Gao. Dream3d: Zero-shot text-to-3d synthesis using 3d shape prior and text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20908–20918, 2023. 83

[187] Tianhan Xu and Wataru Takano. Graph stacked hourglass networks for 3d human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 16105–16114, 2021. 72

[188] Wei Yang, Wanli Ouyang, Xiaolong Wang, Jimmy Ren, Hongsheng Li, and Xiaogang Wang. 3d human pose estimation in the wild by adversarial learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5255–5264, 2018. 72

[189] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In *Proc. of the IEEE/CVF Conf. on CVPR*, 2020. 5

[190] Vickie Ye, Ruilong Li, Justin Kerr, Matias Turkulainen, Brent Yi, Zhuoyang Pan, Otto Seiskari, Jianbo Ye, Jeffrey Hu, Matthew Tancik, and Angjoo Kanazawa. gsplat: An open-source library for gaussian splatting. *Journal of Machine Learning Research*, 2025. 7, 85

[191] 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. In *RSJ International Conference on Intelligent Robots and Systems*, 2021. 6, 20

[192] Ruihong Yin, Vladimir Yugay, Yue Li, Sezer Karaoglu, and Theo Gevers. Fewviewgs: Gaussian splatting with few view matching and multi-stage training. In *Advances in Neural Information Processing Systems*, 2024. 7

[193] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 6

[194] Yuhui Yuan, Rao Fu, Lang Huang, Weihong Lin, Chao Zhang, Xilin Chen, and Jingdong Wang. Hrformer: High-resolution vision transformer for dense predict. *Advances in neural information processing systems*, 34, 2021. 64

[195] Jinlu Zhang, Zhigang Tu, Jianyu Yang, Yujin Chen, and Junsong Yuan. Mixste: Seq2seq mixed spatio-temporal encoder for 3d human pose estimation in video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 13232–13242, 2022. 8, 72

[196] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023. 87

[197] 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*, 2018. 19, 42, 53

[198] Xucong Zhang, Seonwook Park, Thabo Beeler, Derek Bradley, Siyu Tang, and Otmar Hilliges. Eth-xgaze: A large scale dataset for gaze estimation under extreme head pose and gaze variation. In *Proceedings of the European Conference on Computer Vision*. Springer, 2020. 8, 62, 65, 68, 71

[199] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. Mpiigaze: Real-world dataset and deep appearance-based gaze estimation. *IEEE transactions on pattern analysis and machine intelligence*, 41(1), 2017. 8, 62

[200] Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris N. Metaxas. Semantic graph convolutional networks for 3d human pose regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3425–3435, 2019. 72

[201] Qitao Zhao, Ce Zheng, Mengyuan Liu, and Chen Chen. A single 2d pose with context is worth hundreds for 3d human pose estimation. In *Advances in Neural Information Processing Systems*, 2023. 72

[202] Qitao Zhao, Ce Zheng, Mengyuan Liu, and Chen Chen. A single 2d pose with context is worth hundreds for 3d human pose estimation. *Advances in Neural Information Processing Systems*, 36, 2024. 64

[203] Weixi Zhao, Weiqiang Wang, and Yunjie Tian. Graformer: Graph-oriented transformer for 3d pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 20438–20447, 2022. 72

[204] Yian Zhao, Wenyu Lv, Shangliang Xu, Jinman Wei, Guanzhong Wang, Qingqing Dang, Yi Liu, and Jie Chen. Detrs beat yolos on real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2024. 69

[205] Ce Zheng, Sijie Zhu, Matias Mendieta, Taojiannan Yang, Chen Chen, and Zhengming Ding. 3d human pose estimation with spatial and temporal transformers. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 11656–11665, 2021. 8, 63, 72

[206] Xiaolong Zhou, Jianing Lin, Jiaqi Jiang, and Shengyong Chen. Learning a 3d gaze estimator with improved itracker combined with bidirectional lstm. In *Int. Conf. Multimedia and Expo*. IEEE, 2019. 62

[207] Xiaoyu Zhou, Zhiwei Lin, Xiaojun Shan, Yongtao Wang, Deqing Sun, and Ming-Hsuan Yang. Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2024. 7, 33, 83

[208] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 19

[209] Yunsong Zhou, Michael Simon, Zhenghao Peng, Sicheng Mo, Hongzi Zhu, Minyi Guo, and Bolei Zhou. Simgen: Simulator-conditioned driving scene generation. *arXiv preprint arXiv:2406.09386*, 2024. 83

[210] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *Proceedings of the International Conference on Learning Representations*, 2020. 64

[211] Zhiwei Zhu and Qiang Ji. Eye gaze tracking under natural head movements. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1. IEEE, 2005. 62