KRONC: Keypoint-based Robust Camera Optimization for 3D Car Reconstruction

Davide Di Nucci[1], Alessandro Simoni[1], Matteo Tomei[2]
Luca Ciuffreda[2], Roberto Vezzani[1], and Rita Cucchiara[1]

[1] University of Modena and Reggio Emilia `{davide.dinucci,alessandro.simoni,`
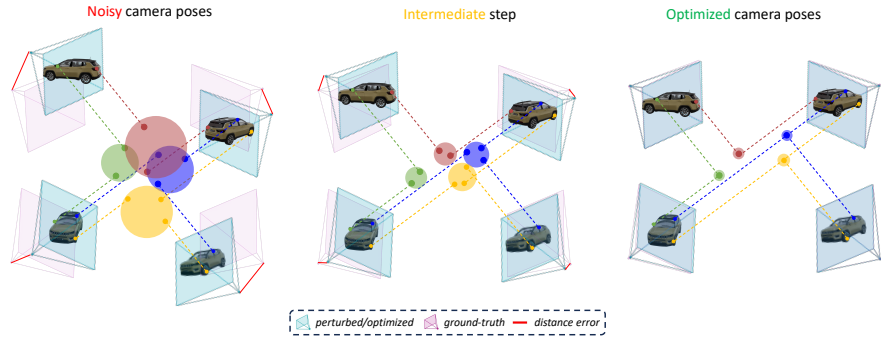`roberto.vezzani,rita.cucchiara}@unimore.it`
[2] Prometeia `{matteo.tomei,luca.ciuffreda}@prometeia.com`

Abstract. The three-dimensional representation of objects or scenes starting from a set of images has been a widely discussed topic for years and has gained additional attention after the diffusion of NeRF-based approaches. However, an underestimated prerequisite is the knowledge of camera poses or, more specifically, the estimation of the extrinsic calibration parameters. Although excellent general-purpose Structure-from-Motion methods are available as a pre-processing step, their computational load is high and they require a lot of frames to guarantee sufficient overlapping among the views. This paper introduces KRONC, a novel approach aimed at inferring view poses by leveraging prior knowledge about the object to reconstruct and its representation through semantic keypoints. With a focus on vehicle scenes, KRONC is able to estimate the position of the views as a solution to a light optimization problem targeting the convergence of keypoints' back-projections to a singular point. To validate the method, a specific dataset of real-world car scenes has been collected. Experiments confirm KRONC's ability to generate excellent estimates of camera poses starting from very coarse initialization. Results are comparable with Structure-from-Motion methods with huge savings in computation. Code and data will be made publicly available.

Keywords: Bundle adjustment · 3D reconstruction

1 Introduction

Recent view synthesis techniques, such as Neural Radiance Fields [34] and 3D Gaussian Splatting [20], 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 [18] or segmentation [7], paving the way for a variety of applications [11, 27, 28, 52]. For instance, owning a NeRF model and directly applying downstream recognition to it allows for easier inspection and assessment [17], compared to conducting the same analysis across individual pictures. The *vehicle inspection* task [12] 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

Fig. 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.

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 Gaussian Splatting methods still rely on classical Structure-from-Motion pipelines (*e.g.* COLMAP [40]) for camera parameters estimation, sometimes even exceeding the time and resource requirements of the actual downstream optimization [35,45]; (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 [13, 15, 44]), leading to a pool of established works and priors to be leveraged for *vehicle inspection*, too.

Motivated by these observations, in this paper 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 SfM, recent works proposed Bundle-Adjusting NeRF [9, 25] 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 SfM packages, making it suitable for every downstream novel view synthesis

technique not limited to NeRFs. As shown in Fig. 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 SfM and methods relying on pairwise image correspondences [24, 47], 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 [46]. 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 making cameras follow a circular trajectory, recovering a plausible pose configuration with a speedup reaching one order of magnitude w.r.t. COLMAP.

2   Related work

In this section, we provide an overview of methodologies centered on camera pose estimation and 3D reconstruction techniques.

Novel view synthesis. The success of NeRF [34] resulted in follow-up strategies aiming to improve both quality and speed. Mip-NeRF [2, 3] along with Zip-NeRF [4] reach state-of-the-art novel-view generation quality by introducing anti-aliasing procedures. A widely recognized issue in 3D reconstruction concerns computation. Several efforts have demonstrated the viability of achieving high-fidelity reconstructions while also shortening overall training time. Instant-NGP [35] employs a multi-resolution hash table alongside a streamlined MLP architecture, enhancing the efficiency of the training process. Alternative techniques like DVGO [45] focus on optimizing voxel grids containing features to facilitate rapid reconstruction of radiance fields. TensoRF [8] combines the traditional CP decomposition with a novel vector-matrix decomposition technique [6]

resulting in accelerated training and improved reconstruction quality. Apart from neural radiance fields optimization, other approaches like Gaussian Splatting [20] have demonstrated impressive outcomes by characterizing the 3D space as a collection of Gaussians. To meet the real-time requirements of *vehicle inspection*, we choose baselines based on training time *vs.* reconstruction quality trade-offs.

Bundle-adjustment and pose refinement. Estimating or refining camera poses represents a critical challenge in both NeRFs and vehicle inspection domains. Structure-from-Motion (S*f*M) techniques [1, 16, 38, 39, 41–43] are widely established methods for acquiring precise geometry and camera poses from video or image data through an offline per-scene optimization process. In contrast, simultaneous localization and mapping (SLAM) methods [5, 14, 36], typically operate online. However, they are known to exhibit unreliability in scenarios with heavily rotating trajectories or scenes containing sparse visual features. Works such as GNeRF [31], NeRF++ [50], and SinERF [51] made efforts towards enhancing the camera poses within NeRF architectures. Approaches such as Barf [25] and L2G-NeRF [9] employ a joint optimization strategy to refine both the radiance field and camera parameters starting from noisy poses. These methods rely solely on the photometric loss as the training signal during optimization. Newer techniques, such as Sparf [47], KeypointNeRF [32] and Corres-NeRF [24] aim to enhance camera pose estimation in few-images scenarios by leveraging multi-view correspondences derived from matches between training views. However, they depend on pairwise image correspondences. In contrast, we introduce a novel method for refining poses based on keypoint information shared among multiple views.

Datasets. The NeRF Synthetic Blender dataset [34] is one of the most extensively used benchmarks for assessing NeRFs performance. This dataset consists of scenes created with Blender[3]. Other synthetic datasets include the Blend DMVS [53], which provides scenes at different scales, and the Shiny Blender dataset [48], which mostly contains objects with simple geometries. Regarding vehicle inspection, the only real-world resource holding captures from a single high-quality vehicle has been introduced in Ref-NeRF [48]. While datasets like Tanks and Temples [22] and LLFF [33] serve as valuable benchmarks for evaluating novel view synthesis in various real-world scenarios, their scope might not be comprehensive enough for in-depth studies focused on vehicles. The CarPatch [12] dataset, despite its detailed annotations and scene diversity, provides synthetic cars only. Our proposed KRONC-dataset aims to address these limitations by facilitating the evaluation of cars from real-world settings, by filling a crucial gap in evaluating and improving vehicle inspection.

3   The KRONC-dataset

In this section, we discuss the source data and the methodology employed to create our KRONC-dataset. Specifically, motivated by the lack of data pertinent

---

[3] http://www.blender.org

Table 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

to vehicle inspection within real-world settings, we detail the steps carried out to manually gather car scenes.

3.1   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 One-Plus 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[4].

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 SfM 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.

3.2   Dataset metadata

To leverage car keypoints for camera extrinsic optimization (as will be detailed in Sec. 4), we automatically annotated semantic keypoints on each single frame of the KRONC-dataset, by adopting the OpenPifPaf [23] framework. Specifically, we used the ShufflenetV2K16 model [30] trained to predict the 66 distinct keypoints defined in ApolloCar3D [44]. 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 [10] with a Swin Large [29] backbone trained on the COCO panoptic dataset [21]. Masks isolate the vehicle from complex backgrounds, allowing to focus on vehicle reconstruction in presence of challenging

---

[4] https://lumalabs.ai/

environmental conditions, which however is not the case for the KRONC-dataset. Finally, each dataset underwent rigorous COLMAP [41] 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 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.

4   Keypoint-based camera optimization

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 [34]).

4.1   Exploiting keypoint projections

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 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 SfM algorithms (*e.g.* COLMAP [40]). KRONC optimizes a noisy/coarse initial approximation of the extrinsic camera parameters. Differently from recent methods exploiting visual pairwise image correspondences [24, 47], 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 [44] 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:

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

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. 5 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.

4.2   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. 1, before computing their 3D centroid C^j as follows:

$$C^j = \begin{bmatrix} X_C^j \\ Y_C^j \\ Z_C^j \end{bmatrix} = \frac{1}{M^j} \sum_i \left(m_i^j \cdot \begin{bmatrix} X_i^j \\ Y_i^j \\ Z_i^j \end{bmatrix} \right). \tag{2}$$

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:

$$\begin{bmatrix} u_{C,i}^j \\ v_{C,i}^j \\ 1 \end{bmatrix} \propto \begin{bmatrix} K & \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} \end{bmatrix} \begin{bmatrix} R_i & \mathbf{t}_i \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} X_C^j \\ Y_C^j \\ Z_C^j \\ 1 \end{bmatrix}. \tag{3}$$

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 $R_i \in \mathbb{R}^{3\times 3}$ 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**

  $\quad R_i = f^{-1}(\mathbf{r}_i)$;

  $\quad \mathcal{L} = 0$;

  $\quad$**foreach** $p^j \in \mathcal{P}$, $j \in \{1, ..., J\}$ **do**

  $\quad\quad C^j = \frac{1}{\sum_{i=1}^N m_i^j} \sum_{i=1}^N m_i^j \pi_i(p_i^j)$;

  $\quad\quad$**foreach** $I_i \in \mathcal{I}$, $i \in \{1, ..., N\}$ **do**

  $\quad\quad\quad \mathcal{L} = \mathcal{L} + m_i^j \left(\|\pi_i(p_i^j) - C^j\|_2 + \lambda\|p_i^j - \pi_i^{-1}(C^j)\|_2 \right)$;

  $\quad\quad$**end**

  $\quad$**end**

  $\quad \mathbf{r}_i = \mathbf{r}_i - \eta \frac{\partial \mathcal{L}}{\partial \mathbf{r}_i}$, $i \in \{1, ..., N\}$;

  $\quad \mathbf{t}_i = \mathbf{t}_i - \eta \frac{\partial \mathcal{L}}{\partial \mathbf{t}_i}$, $i \in \{1, ..., N\}$;

  $\quad z_i^j = z_i^j - \eta \frac{\partial \mathcal{L}}{\partial z_i^j}$, $i \in \{1, ..., N\}$, $j \in \{1, ..., J\}$;

end

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

$$\mathcal{L}_i^j(R_i, \mathbf{t}_i, z_i^j) = \|(X_i^j, Y_i^j, Z_i^j) - (X_C^j, Y_C^j, Z_C^j)\|_2 + \lambda\|(u_i^j, v_i^j) - (u_{C,i}^j, v_{C,i}^j)\|_2 \quad (4)$$

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

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 $\mathcal{P}$:

$$\min_{R_i, \mathbf{t}_i, z_i^j} \frac{1}{J} \sum_{j=1}^J \sum_{i=1}^N m_i^j \mathcal{L}_i^j(R_i, \mathbf{t}_i, z_i^j). \quad (5)$$

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 [19, 47], we adopt the 6D representation of [56], 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. 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 [9, 25], allowing for seamless integration with every downstream method requiring accurate camera poses. KRONC is detailed in Algorithm 1.

5   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 [25], 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 [49], and LPIPS [55].

5.1   Synthetic vehicle scenes

We use the CarPatch dataset [12] 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. Since KRONC requires the annotation of keypoints, we added them in the original CarPatch 3D Blender models, following the semantic convention defined in [44]. 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 [9], 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 $\sigma_R = 4°$ and $\sigma_{\mathbf{t}} = 0.5$ m, respectively.
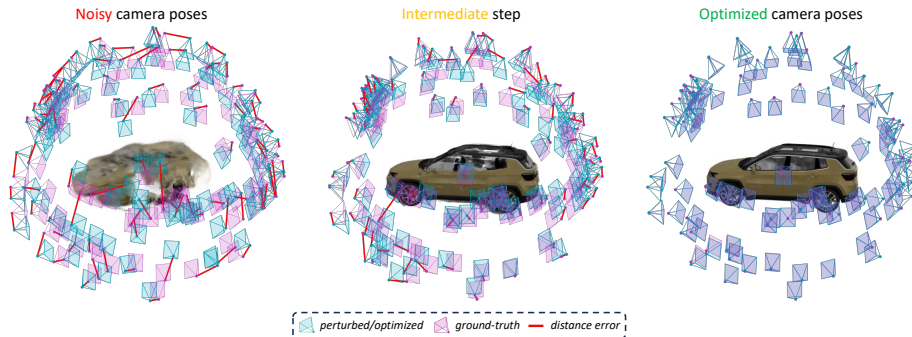
Table 2: Quantitative results averaged over the CarPatch scenes. We assign gold, silver, and bronze medals to the best three methods.

Method	Poses	$\epsilon_R(°)\downarrow$	ϵ_t (cm) $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	Runtime
TensoRF [8]	GT	-	-	34.74	0.973	0.043	35 min
DVGO [45]	GT	-	-	36.09	0.979	0.024	10 min
GaussianSplatting [20]	GT	-	-	34.86	0.982	0.014	5 min
Barf [25]	Noisy	7.67 ●	49.38 ●	17.46	0.870	0.142	12 h
L2G-NeRF [9]	Noisy	0.50 ●	5.26 ●	31.91	0.966	0.060	6 h
KRONC + TensoRF [8]	Noisy	0.65 ●	3.06 ●	33.80 ●	0.971 ●	0.042 ●	35.5 min
KRONC + DVGO [45]	Noisy	0.65 ●	3.06 ●	34.03 ●	0.975 ●	0.029 ●	10.5 min
KRONC + GaussianSplatting [20]	Noisy	0.65 ●	3.06 ●	34.38 ●	0.982 ●	0.014 ●	5.5 min

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 [26,54] 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. 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. 5.2. 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 to 400×400, as in L2G and Barf experimental settings. After a comprehensive assessment of various methods, we select the following baselines:

- **Gaussian Splatting [35]**: the experiments are conducted without altering the original settings. We train for 10k iterations before rendering test images.
- **TensoRF [8]**: we choose to employ the Nerfstudio [46] 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 [45]**: 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 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

Fig. 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.

Table 3: KRONC results on CarPatch by varying rotation and translation noise.

Table 4: KRONC performance on CarPatch using 2D loss, 3D loss, or both.

$\sigma_R(°)$	$\sigma_{\mathbf{t}}$ (cm)	$\epsilon_R(°) \downarrow$	$\epsilon_{\mathbf{t}}$ (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

Loss	$\epsilon_R(°) \downarrow$	$\epsilon_{\mathbf{t}}$ (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**

quality obtained by training on ground truth poses. In terms of camera registration quality, relative rotation error increases by ∼30%, while relative translation error decreases by ∼42% compared to L2G. Both KRONC and L2G demonstrate superior performance compared to Barf. The overall alignment achieved by KRONC closely approximates the ground truth camera poses, as visually depicted in Fig. 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, results do not show significant deterioration even with a 7°, 3 m noise magnitude. Moreover, as explained in section 4.2, 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 4 shows that their combination further improves performance compared to using them individually. All experiments adopt Gaussian Splatting [20].

5.2 Real-world vehicle scenes

To assess the performance of our method in the real domain, we use the proposed KRONC-dataset as our benchmark. As described in Sec. 3, semantic keypoints information come from [23].
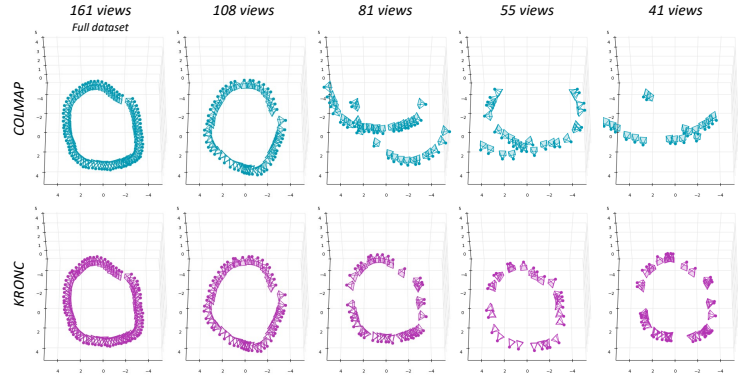
Table 5: Quantitative results on the KRONC dataset. The GaussianSplatting [20] baseline trained with COLMAP poses and an optimized standard trajectory. The results in (·) are computed after masking out the background.

| Env | Vehicle | Init Pose | # Opt. Cameras | Full scene(Masked vehicle) | | |
				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)

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 as our initial coarse camera configuration (the same for all real scenes), which we optimize using KRONC. We follow the LLFF dataset [33] train/test split protocol sampling one test image every 8 frames for each recording. We select Gaussian Splatting [20] as the 3D reconstruction baseline based on the results obtained on the synthetic scenario. Experiments are conducted with the same configuration described in Sec. 5.1, with an image resolution of 480×270.

Results. In Table 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 [9] 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

Fig. 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.

due to the keypoint detector recall, which may leave some viewpoints without 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. 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 *vs.* 8 minutes on a single GPU for the same number of images. It is worth

Table 6: KRONC results with varying depth initialization on KRONC-dataset for masked cars (* indicates unoptimized depth).

Depth type	PSNR ↑	SSIM ↑	LPIPS ↓
DinoV2 [37]*	17.54	0.703	0.223
DinoV2 [37]	20.53	0.827	0.149
Random	**21.89**	**0.852**	**0.114**

noting that this comparison does not take into account the inference time needed for OpenPifPaf [23] keypoint extraction, which is 33 seconds on average over the KRONC-dataset scenes. This leads to an effective $\sim 8\times$ speedup.

Additional analysis. For real scenes, the z_i^j depth values are randomly initialized following the same approach described in Sec. 5.1 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 [37]. 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 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, COLMAP camera pose estimation capability rapidly degrades when reducing the number of images (*i.e.* decreasing image overlap), as already noted in [47]. In contrast, KRONC results demonstrate that by replacing pairwise matches with global reasoning via shared

Fig. 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.

semantic keypoints and by coarsely initializing camera poses using some prior knowledge, robust registration can be achieved even with limited data.

5.3 Qualitative results

In Fig. 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.

6   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. SfM 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. 5.2. 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.

References

1. Agarwal, S., Furukawa, Y., Snavely, N., Simon, I., Curless, B., Seitz, S.M., Szeliski, R.: Building rome in a day. Communications of the ACM **54**(10) (2011) 4

2. Barron, J.T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., Srinivasan, P.P.: Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In: Int. Conf. Comput. Vis. (2021) 3

3. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: IEEE Conf. Comput. Vis. Pattern Recog. (2022) 3

4. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Zip-nerf: Anti-aliased grid-based neural radiance fields. Int. Conf. Comput. Vis. (2023) 3

5. Campos, C., Elvira, R., Rodríguez, J.J.G., Montiel, J.M., Tardós, J.D.: Orb-slam3: An accurate open-source library for visual, visual-inertial, and multimap slam. IEEE Transactions on Robotics (2021) 4

6. Carroll, J.D., Chang, J.J.: Analysis of individual differences in multidimensional scaling via an n-way generalization of "eckart-young" decomposition. Psychometrika **35**(3) (1970) 3

7. Cen, J., Zhou, Z., Fang, J., Yang, C., Shen, W., Xie, L., Zhang, X., Tian, Q.: Segment anything in 3d with nerfs. In: Adv. Neural Inform. Process. Syst. (2023) 1

8. Chen, A., Xu, Z., Geiger, A., Yu, J., Su, H.: Tensorf: Tensorial radiance fields. In: Eur. Conf. Comput. Vis. Springer (2022) 3, 10

9. Chen, Y., Chen, X., Wang, X., Zhang, Q., Guo, Y., Shan, Y., Wang, F.: Local-to-global registration for bundle-adjusting neural radiance fields. In: IEEE Conf. Comput. Vis. Pattern Recog. (2023) 2, 4, 9, 10, 12, 19, 20

10. Cheng, B., Misra, I., Schwing, A.G., Kirillov, A., Girdhar, R.: Masked-attention mask transformer for universal image segmentation. In: IEEE Conf. Comput. Vis. Pattern Recog. (2022) 5

11. Corona-Figueroa, A., Frawley, J., Bond-Taylor, S., Bethapudi, S., Shum, H.P., Willcocks, C.G.: 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) 1

12. Di Nucci, D., Simoni, A., Tomei, M., Ciuffreda, L., Vezzani, R., Cucchiara, R.: Carpatch: A synthetic benchmark for radiance field evaluation on vehicle components. In: International Conference on Image Analysis and Processing. Springer (2023) 1, 4, 9, 19

13. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: Carla: An open urban driving simulator. In: Conference on robot learning. PMLR (2017) 2

14. Engel, J., Schöps, T., Cremers, D.: Lsd-slam: Large-scale direct monocular slam. In: Eur. Conf. Comput. Vis. Springer (2014) 4

15. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: IEEE Conf. Comput. Vis. Pattern Recog. IEEE (2012) 2

16. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge university press (2003) 4

17. Hong, K., Wang, H., Yuan, B.: Inspection-nerf: Rendering multi-type local images for dam surface inspection task using climbing robot and neural radiance field. Buildings **13**(1) (2023) 1

18. Hu, B., Huang, J., Liu, Y., Tai, Y.W., Tang, C.K.: Nerf-rpn: A general framework for object detection in nerfs. In: IEEE Conf. Comput. Vis. Pattern Recog. (2023) 1

19. Jeong, Y., Ahn, S., Choy, C., Anandkumar, A., Cho, M., Park, J.: Self-calibrating neural radiance fields. In: Int. Conf. Comput. Vis. (2021) 8

20. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Trans. Graph. **42**(4) (2023) 1, 4, 10, 11, 12, 20

21. Kirillov, A., He, K., Girshick, R., Rother, C., Dollár, P.: Panoptic segmentation. In: IEEE Conf. Comput. Vis. Pattern Recog. (2019) 5

22. Knapitsch, A., Park, J., Zhou, Q.Y., Koltun, V.: Tanks and temples: Benchmarking large-scale scene reconstruction. ACM Transactions on Graphics (2017) 4

23. Kreiss, S., Bertoni, L., Alahi, A.: OpenPifPaf: Composite Fields for Semantic Keypoint Detection and Spatio-Temporal Association. IEEE Transactions on Intelligent Transportation Systems (2021) 5, 11, 13

24. Lao, Y., Xu, X., Liu, X., Zhao, H., et al.: Corresnerf: Image correspondence priors for neural radiance fields. In: Adv. Neural Inform. Process. Syst. (2023) 3, 4, 6

25. Lin, C.H., Ma, W.C., Torralba, A., Lucey, S.: Barf: Bundle-adjusting neural radiance fields. In: Int. Conf. Comput. Vis. (2021) 2, 4, 9, 10, 19, 20

26. Lin, C.H., Wang, O., Russell, B.C., Shechtman, E., Kim, V.G., Fisher, M., Lucey, S.: Photometric mesh optimization for video-aligned 3d object reconstruction. In: IEEE Conf. Comput. Vis. Pattern Recog. (2019) 10

27. Liu, J.Y., Chen, Y., Yang, Z., Wang, J., Manivasagam, S., Urtasun, R.: Real-time neural rasterization for large scenes. In: IEEE Conf. Comput. Vis. Pattern Recog. (2023) 1

28. Liu, J.W., Cao, Y.P., Yang, T., Xu, Z., Keppo, J., Shan, Y., Qie, X., Shou, M.Z.: Hosnerf: Dynamic human-object-scene neural radiance fields from a single video. In: Int. Conf. Comput. Vis. (2023) 1

29. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Int. Conf. Comput. Vis. (2021) 5

30. Ma, N., Zhang, X., Zheng, H.T., Sun, J.: Shufflenet v2: Practical guidelines for efficient cnn architecture design. In: Eur. Conf. Comput. Vis. (2018) 5

31. Meng, Q., Chen, A., Luo, H., Wu, M., Su, H., Xu, L., He, X., Yu, J.: Gnerf: Gan-based neural radiance field without posed camera. In: IEEE Conf. Comput. Vis. Pattern Recog. (2021) 4

32. Mihajlovic, M., Bansal, A., Zollhoefer, M., Tang, S., Saito, S.: Keypointnerf: Generalizing image-based volumetric avatars using relative spatial encoding of keypoints. In: Eur. Conf. Comput. Vis. Springer (2022) 4

33. Mildenhall, B., Srinivasan, P.P., Ortiz-Cayon, R., Kalantari, N.K., Ramamoorthi, R., Ng, R., Kar, A.: Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. ACM Trans. Graph. (2019) 4, 12

34. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1) (2021) 1, 3, 4, 6

35. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM Trans. Graph. **41**(4) (2022) 2, 3, 10

36. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: Orb-slam: a versatile and accurate monocular slam system. IEEE Transactions on Robotics **31**(5) (2015) 4

37. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 (2023) 13

38. Pollefeys, M., Koch, R., Gool, L.V.: Self-calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameters. Int. J. Comput. Vis. **32**(1) (1999) 4

39. Pollefeys, M., Van Gool, L., Vergauwen, M., Verbiest, F., Cornelis, K., Tops, J., Koch, R.: Visual modeling with a hand-held camera. Int. J. Comput. Vis. **59** (2004) 4

40. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: IEEE Conf. Comput. Vis. Pattern Recog. (2016) 2, 6

41. Schönberger, J.L., Zheng, E., Pollefeys, M., Frahm, J.M.: Pixelwise view selection for unstructured multi-view stereo. In: Eur. Conf. Comput. Vis. (2016) 4, 6

42. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: exploring photo collections in 3d. ACM Trans. Graph. (2006) 4

43. Snavely, N., Seitz, S.M., Szeliski, R.: Modeling the world from internet photo collections. Int. J. Comput. Vis. **80** (2008) 4

44. Song, X., Wang, P., Zhou, D., Zhu, R., Guan, C., Dai, Y., Su, H., Li, H., Yang, R.: Apollocar3d: A large 3d car instance understanding benchmark for autonomous driving. In: IEEE Conf. Comput. Vis. Pattern Recog. (2019) 2, 5, 6, 9

45. Sun, C., Sun, M., Chen, H.T.: Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In: IEEE Conf. Comput. Vis. Pattern Recog. (2022) 2, 3, 10

46. Tancik, M., Weber, E., Ng, E., Li, R., Yi, B., Kerr, J., Wang, T., Kristoffersen, A., Austin, J., Salahi, K., Ahuja, A., McAllister, D., Kanazawa, A.: Nerfstudio: A modular framework for neural radiance field development. In: ACM SIGGRAPH 2023 Conference Proceedings (2023) 3, 10

47. Truong, P., Rakotosaona, M.J., Manhardt, F., Tombari, F.: Sparf: Neural radiance fields from sparse and noisy poses. In: IEEE Conf. Comput. Vis. Pattern Recog. (2023) 3, 4, 6, 8, 13

48. Verbin, D., Hedman, P., Mildenhall, B., Zickler, T., Barron, J.T., Srinivasan, P.P.: Ref-nerf: Structured view-dependent appearance for neural radiance fields. In: IEEE Conf. Comput. Vis. Pattern Recog. (2022) 4

49. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4) (2004) 9

50. Wang, Z., Wu, S., Xie, W., Chen, M., Prisacariu, V.A.: Nerf--: Neural radiance fields without known camera parameters. arXiv preprint arXiv:2102.07064 (2021) 4

51. Xia, Y., Tang, H., Timofte, R., Gool, L.V.: Sinerf: Sinusoidal neural radiance fields for joint pose estimation and scene reconstruction. In: British Machine Vision Conference (2022) 4

52. Xie, Z., Zhang, J., Li, W., Zhang, F., Zhang, L.: S-nerf: Neural radiance fields for street views. In: Int. Conf. Learn. Represent. (2023) 1

53. Yao, Y., Luo, Z., Li, S., Zhang, J., Ren, Y., Zhou, L., Fang, T., Quan, L.: Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In: IEEE Conf. Comput. Vis. Pattern Recog. (2020) 4

54. Yen-Chen, L., Florence, P., Barron, J.T., Rodriguez, A., Isola, P., Lin, T.Y.: inerf: Inverting neural radiance fields for pose estimation. in 2021 ieee. In: RSJ International Conference on Intelligent Robots and Systems (2021) 10

55. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: IEEE Conf. Comput. Vis. Pattern Recog. (2018) 9
56. Zhou, Y., Barnes, C., Lu, J., Yang, J., Li, H.: On the continuity of rotation representations in neural networks. In: IEEE Conf. Comput. Vis. Pattern Recog. (2019) 8

A   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.

B   Additional details

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 on both the main paper and this supplementary material 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 our strategy for adding noise differs from Barf [25], 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 [9], 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 of the main paper, our dataset captures a diverse set of 7 vehicles across 3 distinct environments. Figure 8 showcases example captures from each environment, along with keypoint and mask annotations.

C   Additional quantitative results

The CarPatch [12] dataset provides ground-truth camera pose annotations, which can be thought of as an upper bound for KRONC optimization, as already done

Table 7: 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
ϵ_t (cm) →	Barf [25]	53.87 ●	73.68 ●	37.08 ●	76.05 ●	56.70 ●	24.37 ●	13.41 ●	59.89 ●
	L2G-NeRF [9]	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 ●
ϵ_R (°) →	Barf [25]	15.38 ●	7.08 ●	5.15 ●	13.60 ●	7.69 ●	2.99 ●	2.27 ●	7.27 ●
	L2G-NeRF [9]	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 [25]	17.88 ●	13.43 ●	17.51 ●	12.63 ●	14.83 ●	21.08 ●	27.16 ●	15.19 ●
	L2G-NeRF [9]	33.19 ●	33.22 ●	31.55 ●	31.88 ●	32.44 ●	30.19 ●	31.24 ●	31.59 ●
	KRONC + GaussianSplatting [20]	36.31 ●	36.59 ●	36.77 ●	33.05 ●	34.67 ●	31.32 ●	32.57 ●	33.74 ●
SSIM ↑	Barf [25]	0.879 ●	0.827 ●	0.912 ●	0.827 ●	0.844 ●	0.868 ●	0.942 ●	0.858 ●
	L2G-NeRF [9]	0.972 ●	0.976 ●	0.972 ●	0.971 ●	0.927 ●	0.937 ●	0.965 ●	0.966 ●
	KRONC + GaussianSplatting [20]	0.986 ●	0.987 ●	0.988 ●	0.983 ●	0.985 ●	0.961 ●	0.980 ●	0.981 ●
LPIPS →	Barf [25]	0.139 ●	0.190 ●	0.092 ●	0.198	0.157 ●	0.130 ●	0.084 ●	0.146 ●
	L2G-NeRF [9]	0.052 ●	0.056 ●	0.043 ●	0.054 ●	0.048 ●	0.098 ●	0.069 ●	0.057 ●
	KRONC + GaussianSplatting [20]	0.012 ●	0.010 ●	0.009 ●	0.011 ●	0.012 ●	0.027 ●	0.015 ●	0.014 ●

Table 8: KRONC performances by sampling a different number of poses from the CarPatch dataset.

# poses	ϵ_R (°) ↓	ϵ_t (cm) ↓	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

in Sec. 5.1 of the main paper. In this section, we show additional ablation studies performed on the synthetic data.

KRONC vs state-of-the-art. Table 7 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 8, we show KRONC's robustness by 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

Table 9: Performance comparison of KRONC + GaussianSplatting and L2G-NeRF on CarPatch dataset with different noise levels.

$\sigma_R(^\circ)$	σ_t (cm)	KRONC+GaussianSplatting					L2G-NeRF				
		$\epsilon_R(^\circ)\downarrow$	ϵ_t (cm)$\downarrow$	PSNR$\uparrow$	SSIM$\uparrow$	LPIPS$\downarrow$	$\epsilon_R(^\circ)\downarrow$	ϵ_t (cm)$\downarrow$	PSNR$\uparrow$	SSIM$\uparrow$	LPIPS$\downarrow$
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

shown in Table 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.

D   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.
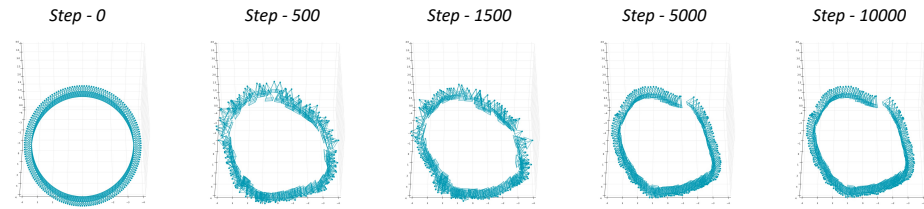
KRONC vs state-of-the-art. Figure 5 presents a qualitative comparison among various methods utilized for reconstructing vehicles in the CarPatch dataset from noisy camera poses. Barf encounters challenges in accurately reconstructing vehicles, while L2G-NeRF demonstrates greater consistency in this task. Notably, leveraging KRONC alongside Gaussian Splatting (GS) leads to a more precise vehicle reconstruction, effectively capturing intricate details.

Trajectory optimization. Figure 6 illustrates the trajectory optimization process for real-world scenarios, as detailed in Section 5.2 of the main paper. 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).
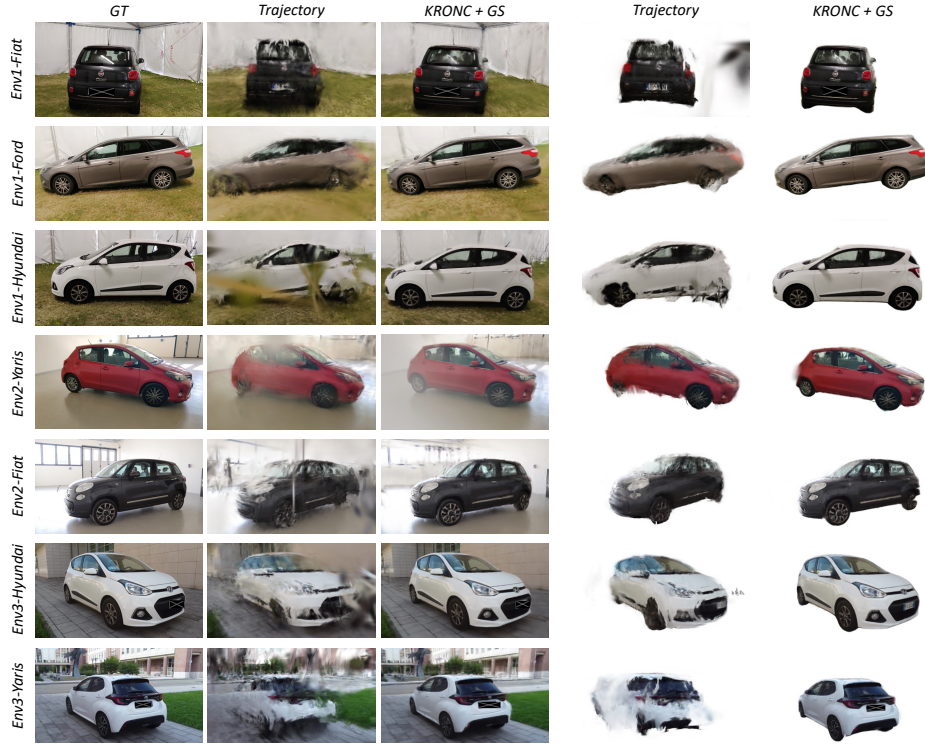
Coarse vs optimized trajectory. In Figure 7, we present qualitative results illustrating KRONC's capability to reconstruct vehicles in a real-case scenario. Starting from the initialization of cameras, as detailed in 5.2 of the main paper, our method successfully achieves an enhanced vehicle reconstruction. This improvement is evident in both environments, with or without background.

Fig. 5: Comparison of qualitative results across all scenes in the CarPatch dataset, showcasing vehicle reconstructions from Barf, L2G-NeRF, and KRONC + Gaussian Splatting.
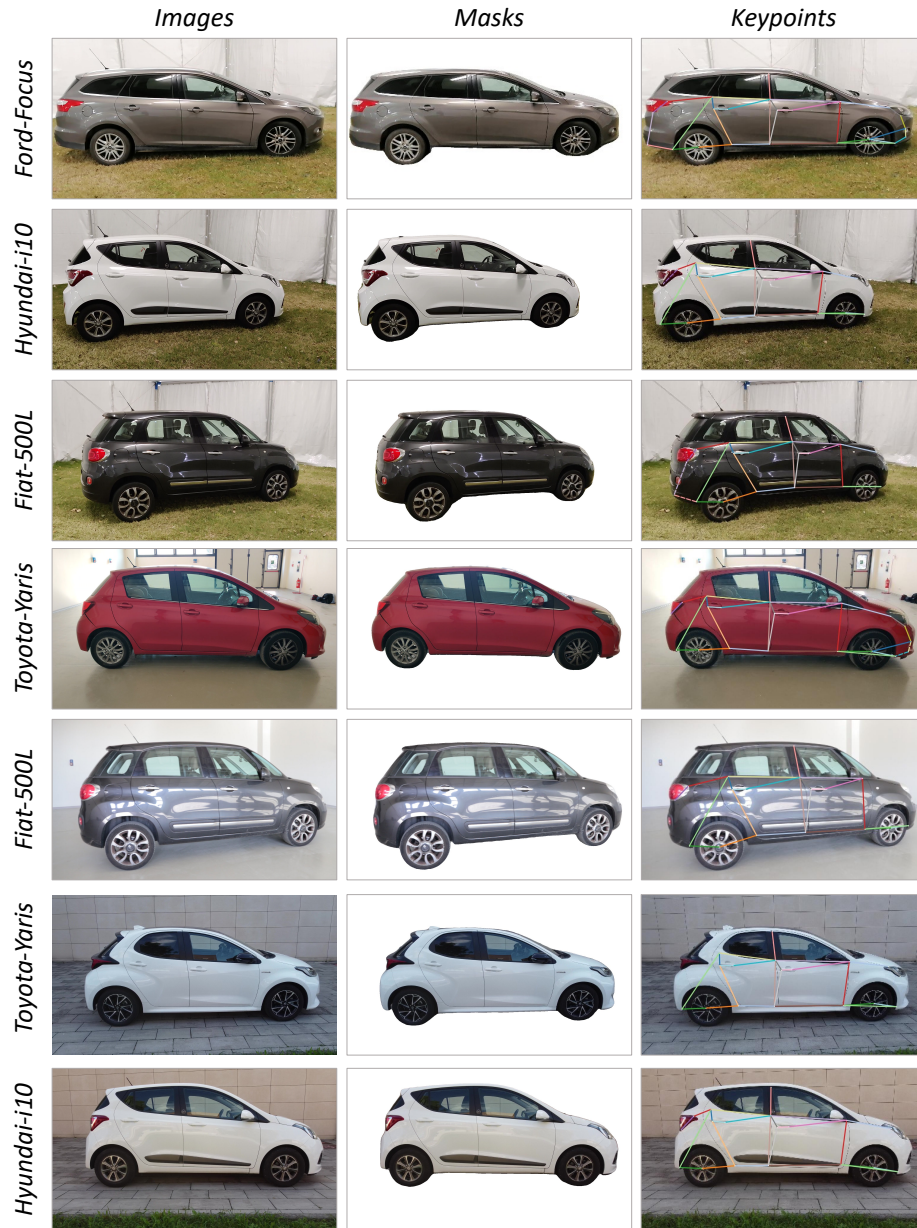

Fig. 6: Optimization steps in a real scenario. On the left, a visualization of the initial circular trajectory. On the right, the optimized trajectory at each intermediate step.

Fig. 7: 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.

Fig. 8: Overview of the KRONC-dataset showing the full-scene images, the segmented vehicles, and the predicted keypoints.