AutoMatch: Leveraging Traffic Camera to Improve Perception and Localization of Autonomous Vehicles

Yuze He[†], Li Ma[‡], Jiahe Cui[§], Zhenyu Yan[†], Guoliang Xing[†], Sen Wang[¶], Qintao Hu[¶], Chen Pan[$]
[†]The Chinese University of Hong Kong, Hong Kong SAR, China
[‡]The Hong Kong University of Science and Technology, Hong Kong SAR, China
[§]School of Computer Science and Engineering, Beihang University, Beijing, China
[¶]2012 Lab, Huawei Technologies, Shenzhen, China
[$]Smart Car Solutions Business Unit, Huawei Technologies, Hangzhou, China
Email: hy019@ie.cuhk.edu.hk, lmaag@connect.ust.hk, cuijiahe@buaa.edu.cn, zyyan@cuhk.edu.hk, glxing@cuhk.edu.hk, wangsen31@huawei.com, huqintao2@huawei.com, panchen13@huawei.com

ABSTRACT

Traffic camera is one of the most ubiquitous traffic facilities, providing high coverage of complex, accident-prone road sections such as intersections. This work leverages traffic cameras to improve the perception and localization performance of autonomous vehicles at intersections. In particular, vehicles can expand their range of perception by matching the images captured by both the traffic cameras and on-vehicle cameras. Moreover, a traffic camera can match its images to an existing high-definition map (HD map) to derive centimeter-level location of the vehicles in its field of view. To this end, we propose *AutoMatch* - a novel system for *real-time image registration*, which is a key enabling technology for traffic camera-assisted perception and localization of autonomous vehicles. Our key idea is to leverage landmark keypoints of distinctive structures such as ground signs at intersections to facilitate image registration between traffic cameras and HD maps or vehicles. By leveraging the strong structural characteristics of ground signs, *AutoMatch* can extract very few but precise landmark keypoints for registration, which effectively reduces the communication/compute overhead. We implement *AutoMatch* on a testbed consisting of a self-built autonomous car, drones for surveying and mapping, and real traffic cameras. In addition, we collect two new multi-view traffic image datasets at intersections, which contain images from 220 real operational traffic cameras in 22 cities. Experimental results show that *AutoMatch* achieves pixel-level image registration accuracy within 88 milliseconds, and delivers an 11.7× improvement in accuracy, 1.4× speedup in compute time, and 17.1× data transmission saving over existing approaches.

CCS CONCEPTS

• **Computer systems organization → Sensor networks**.

*Corresponding author.

SenSys '22, November 6–9, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9886-2/22/11…$15.00
https://doi.org/10.1145/3560905.3568519

KEYWORDS

Image registration, Vehicle-infrastructure cooperative sensing, Infrastructure-assisted autonomous driving, Edge computing

ACM Reference Format:
Yuze He[†], Li Ma[‡], Jiahe Cui[§], Zhenyu Yan[†], Guoliang Xing[†], Sen Wang[¶], Qintao Hu[¶], Chen Pan[$]. 2022. AutoMatch: Leveraging Traffic Camera to Improve Perception and Localization of Autonomous Vehicles. In *The 20th ACM Conference on Embedded Networked Sensor Systems (SenSys '22), November 6–9, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3560905.3568519

1 INTRODUCTION

In this work, we leverage traffic cameras to assist two fundamental applications of autonomous driving (see Fig. 1): 1) Boosting vehicle perception. Vehicles will be able to see beyond obstacle occlusions and expand their range of perception by taking advantage of the traffic cameras, which are typically mounted a few meters above ground and hence provide a much wider and almost unobscured field of view. Specifically, by matching the images captured by both traffic cameras and itself, an autonomous vehicle can complement and expand its field of view and improve its situational awareness. 2) *High-precision vehicle localization.* A traffic camera can match its images to existing high-definition global maps (HD maps) to derive centimeter-level location of the vehicles in its view. This process can be implemented by the infrastructure or cloud and hence significantly lower the requirements on the vehicle's compute/localization capabilities. Such two applications provide autonomous vehicles with boosted perception and high-precision localization, which greatly improves the accuracy and reliability of vehicles' downstream tasks at complex intersection environments, including path planning, decision making, and vehicle control. In this work, we focus on leveraging traffic cameras at intersections because of the following three reasons. First, intersections are more accident-prone than other road sections. Second, intersections have highly complex structures, introducing unique challenges for autonomous driving. Third, to date, most traffic cameras are installed around intersections [16, 51].

The key technology that enables both above applications is *real-time high-precision image registration*, which refers to the process of finding the homography between two image coordinate systems. In the above two applications, images from traffic cameras are registered with those from either vehicles or HD maps. Through

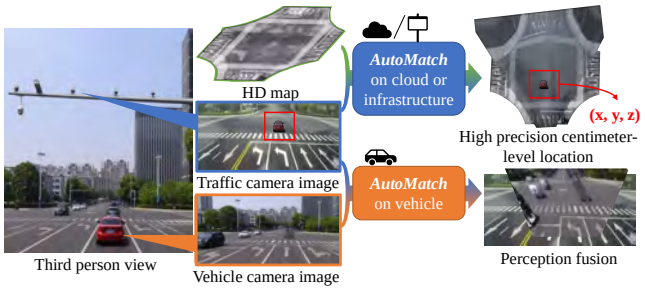


Figure 1: **Two applications of *AutoMatch*. 1) The vehicle's perception is boosted by fusing the perception information of the traffic camera and 2) The vehicle's high-precision location is derived from a traffic camera image and an HD map.**

registration, the raw data or high-level information such as detection results of one image can be transformed and merged into the coordinate system of the other image. There are three challenges in high-precision image registration involving traffic camera images. First, these images are taken in dramatically varied conditions (e.g., viewpoints, scales and view angles), which poses great challenges to high-precision registration. Second, to support autonomous driving, two images need to establish pixel-level correspondences in real-time (e.g., within tens of milliseconds), which requires the image registration method to be computationally efficient. Third, due to the significant dynamics and limited bandwidth between infrastructures and vehicles, the amount of data sharing required for registration should be as small as possible.

Although there exist methods for image registration in the computer vision literature [5, 6, 9, 17, 18, 25, 30, 38, 61, 62], they are not specifically designed for traffic scenarios and yield unsatisfactory performance in latency, robustness, and accuracy, making them ill-suited for infrastructure-assisted autonomous driving. Most current image registration techniques [5, 6, 9, 17, 18, 25] first extract a large amount of keypoints throughout two images and then match them to register two images. Other methods [30, 38, 61, 62] directly find the correspondences of two images in an end-to-end manner by leveraging deep learning techniques. The former requires the transmission of hundreds or thousands of infrastructure keypoints and features for each frame between infrastructure and vehicle, whose excessive communication overhead poses a major challenge in meeting the stringent real-time requirement of autonomous driving applications. The latter usually requires a large DNN model to achieve accurate transformation from in the wild images, which incurs excessive compute overhead on the vehicle and hence is ill-suited for real-time autonomous driving. Therefore, there still remains a major gap between the vision of traffic camera-assisted autonomous driving and the capabilities of current image registration technologies.

To tackle these challenges, we propose *AutoMatch* - a novel system that accurately registers image pairs from different views in real-time to support traffic camera-assisted autonomous driving at intersections. Our key idea is to extract landmark keypoints of salient structures at intersections to facilitate image registration. Specifically, *AutoMatch* first detects and extracts ground signs, which are the most common semantic objects at intersections and distinctive structures shared by the images from both vehicle's onboard camera and traffic camera. Then, we propose a novel *landmark keypoint extractor* to robustly and accurately locate very few landmark keypoints of ground signs. The novelty of our design lies in the integration of a landmark detector and a general keypoint detector. In this paper, we refer to the points extracted by the general keypoint detector as *keypoints*, the points extracted by the landmark detector as *landmarks*, and the points extracted by the landmark keypoint extractor as *landmark keypoints*. Motivated by the fact that most ground signs have a dominant structural pattern (e.g., arrows), we develop a new landmark detector to find structurally meaningful landmarks of ground signs and refine them using a general keypoint detector to achieve sub-pixel accuracy of the landmark keypoint location. The *landmark keypoint extractor* greatly improves the robustness of image registration by eliminating noisy and irrelevant points. At last, we design an efficient keypoint matching algorithm based on the detected ground signs and their landmark keypoints from the two images.

To summarize, fundamentally different from the current image registration methods in the computer vision literature, our system offers several key advantages: (i) *AutoMatch* is robust to environments including different types of intersections, traffic signs, roadside trees and buildings around the intersections, since our approach only focuses on distinctive structures and filter out unimportant information that may affect the accuracy of matching. (ii) *AutoMatch* is computationally efficient and memory friendly, which is crucial for practical deployment in real-world traffic scenarios. *AutoMatch* achieves this by only processing small image patches and extracting few but semantically rich landmark keypoints for registration. In contrast, existing approaches require processing the whole image or extracting massive keypoints. (iii) *AutoMatch* significantly reduces the communication overhead between infrastructure and vehicle for the registration, since it only requires the infrastructure to share with the vehicle a small number of landmark keypoints extracted from static structures independent of traffic dynamics.

We implemented *AutoMatch* on a real testbed consisting of a self-built autonomous car, a survey drone for mapping, and real traffic cameras. In addition, we collect two new multi-view traffic image datasets, which correspond to the perception and localization of traffic camera-assisted autonomous driving, respectively. The first dataset contains 1,136 image pairs from 48 traffic cameras of 19 intersections and onboard cameras of vehicles. The second dataset contains images from 172 traffic cameras of 32 intersections in 21 cities and the corresponding high-resolution maps. Experiments show that *AutoMatch* is able to extend the vehicle's field of view by 72.9% on average, with an average image registration error of 3 pixels, which delivers an 11.65× improvement in registration accuracy compared with the state-of-the-art. Besides, *AutoMatch* leverages traffic cameras to provide high-precision localization for autonomous vehicles with an error of less than 20 cm. Moreover, *AutoMatch* only requires the traffic camera to share the data with the vehicle at a rate of 72 Kbps. Lastly, *AutoMatch* achieves an end-to-end system latency within 88 ms, which enables real-time image registration for autonomous vehicles.

The rest of this paper is organized as follows: Section 2 introduces related work. Section 3 presents the background, applications, and challenges. In Section 4, we describe the design of *AutoMatch*.

We discuss the collection of two datasets and the system implementation in Sections 5 and 6, respectively. Section 7 shows the experiment results and Section 8 concludes the paper.

2 RELATED WORK

Image Registration. Image registration aims to find correspondence between two images and has many applications in autonomous driving such as camera calibration [26], Simultaneous Localization and Mapping (SLAM) [46], and Structure from Motion (SfM) [27, 55]. A typical registration pipeline consists of three stages, keypoint detection, description generation, and keypoint matching. Both classical [6, 25, 35, 39, 42] and learning-based [53, 69] methods detect points of interest throughout a whole image. However, they are not applicable to complex and diverse intersection scenarios since there is no guarantee that meaningful keypoints for registration can be extracted at the first step of image registration. The feature descriptors are extracted from a local patch centered around each keypoint to capture higher-level information and generate robust and precise representations for keypoints. However, they may suffer from ambiguity when there are repetitive contents that are common in traffic scenarios. Moreover, these descriptors are usually represented as large-sized feature vectors, which incur significant communication overhead and are ill-suited for traffic camera-assisted autonomous driving. The final step, keypoint matching, matches two keypoints in the two input images that have the most similar descriptors. Nearest neighbor [45] and fast approximation nearest neighbor [44] algorithms are two representative methods, but they perform poorly when encountering too many outlier keypoints. Tracking-based matching methods are widely adopted in visual SLAM and can achieve real-time performance. However, they work well only for two similar images, such as the neighboring frames of a video. Recent works use Graph Neural Networks (GNN) [52] and transformers [30] to boost the matching performance for challenging cases. Nevertheless, the methods based on the above three-stage pipeline require a certain similarity of scales and perspectives of two images, while the images to be matched in the traffic camera-assisted autonomous driving usually have significant scale and viewpoint differences as well as overly repeated contents.

Landmark Detection. The goal of landmark detection is to localize a group of pre-defined landmarks on objects with semantically meaningful structures. For example, facial landmark detectors [49, 59, 72] predict 5, 20 or 68 fiducial points, outlining the face boundaries, eye, nose and mouth. Body keypoint detectors [11, 15, 47, 66] detect 14 or 17 keypoints, indicating shoulders, wrists, etc. Unlike general keypoint detectors that extract keypoints in an indiscriminate manner, landmark detectors "recognize" the semantic part of the object by exploiting the shape pattern, like symmetry and spatial relationships. Surprisingly, the use of landmark detectors for image registration has not been well explored despite the following advantages: (i) Robustness to noise and outliers caused by similar low-level image appearances, as the shape and structured information provide constraints to each landmark. (ii) Unlike general keypoint descriptors where each descriptor is represented as a one-dimensional feature vector, landmarks are more interpretable and discriminative. However, a critical shortcoming of landmark detectors is that the predicted landmark location is usually less accurate compared to general keypoint detectors, and can not achieve pixel-level registration precision. In this work, we address this key issue by integrating the general keypoint detector to refine the detected landmarks to obtain landmark keypoints with precise locations.

Cooperative Infrastructure-Vehicle or Vehicle-Vehicle Perception and Localization. To improve the perception performance of autonomous vehicles, Arnold et al. [3] propose a cooperative 3D object detection scheme, where several infrastructure sensors are used for multi-view simultaneous 3D object detection. Zhang et al. [71] propose an edge-assisted multi-vehicle perception system called EMP, where connected and autonomous vehicles' (CAVs') individual point clouds are optimally partitioned and merged to form a complete point cloud with a higher resolution. In [43], cameras and LiDARs are leveraged to assist the localization of autonomous vehicles. Fascista et al. [21] propose to localize vehicles using the angle of arrival estimation of beacons from several infrastructure nodes. Different from these studies where infrastructures or CAVs have known accurate pose, i.e., position and orientation, we focus on leveraging existing traffic cameras with unknown poses to assist autonomous vehicles in both perception and localization through real-time image registration.

3 BACKGROUND, APPLICATIONS AND CHALLENGES

In this section, we first review autonomous driving perception and localization technologies available today, which motivates our approach. We then present the two applications of *AutoMatch* for assisting autonomous driving at intersections. Finally, the challenges addressed in the design of *AutoMatch* are discussed.

3.1 Perception/Localization of Autonomous Driving

Like human drivers, autonomous vehicles must know where they are on the road (localization) and which objects are in the surroundings (perception). Perception and localization are essential for autonomous vehicles to make accurate and reliable decisions for vehicle control. Due to the mission-critical nature, autonomous driving imposes stringent requirements on the accuracy and delay of perception and localization [64].

Mainstream autonomous driving platforms typically use a combination of sensors such as cameras, LiDARs, radars, GNSS/IMUs, and odometers for high-precision perception and localization [37]. Specifically, vehicles consume incoming camera images or LiDAR point clouds to detect and track obstacles such as moving vehicles and people within. Then the free navigable space is identified to ensure that the vehicle does not collide with moving objects. However, on-vehicle sensors have a limited field of view, and the perception will often be obscured by surrounding objects, which may unavoidably cause traffic accidents. To achieve high-precision localization, many commercial vehicles, such as the Google and Uber cars, use a priori mapping approach [28, 60], which consists of pre-driving specific roads, collecting detailed 3D point clouds, and generating high precision maps. Vehicles can store such maps or download them from the cloud. Localization is then performed by matching the current sensor data with HD maps. However, the

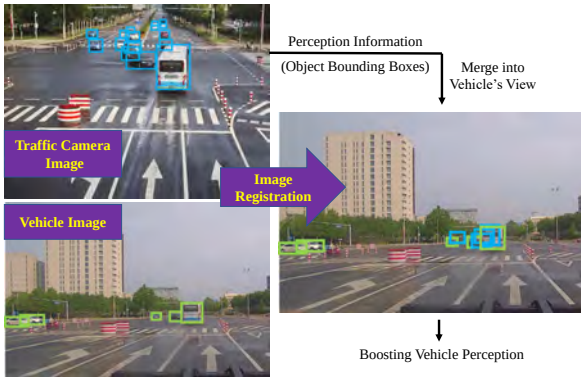


Figure 2: **Illustration of leveraging traffic camera to boost vehicle perception.**

large size of HD maps, the high latency in transmissions between the cloud and the vehicle, and the low updating frequency of HD maps pose significant barriers to wide adoption in practice [56].

3.2 Applications of *AutoMatch*

Boosting vehicle perception via image registration between traffic cameras and vehicles. The first application of *AutoMatch* is real-time image registration between the traffic camera image and vehicle image. Image registration establishes the transformation between the two image coordinate systems of the traffic camera and the vehicle, so that the vehicle can directly utilize the perception information in the traffic camera image. The scene perception information shared from the traffic camera to the vehicle can be the entire image (with all the details of the scene) or abstract semantic information (such as object bounding boxes). To actually achieve such benefits, the data transmission volume for registration needs to be small enough due to the limited communication bandwidth between the traffic camera and the vehicle. Besides, the end-to-end traffic camera-vehicle image registration delay needs to be within tens of milliseconds to meet the real-time requirements of autonomous driving.

In practice, infrastructures and vehicles need to independently extract points in their images for registration. Different from other image registration approaches [5, 17, 19, 35, 57] that would need the infrastructure to extract points in real-time, *AutoMatch* allows the infrastructure to extract points less frequently. This is because *AutoMatch* extracts static points in the scene backgrounds, which remain unchanged most of the time. Once extracted, the infrastructure then periodically broadcasts the points and the perception information (object bounding boxes) extracted on its own coordinate. When a vehicle enters the intersection, it first receives the points from infrastructure and then matches them with the points extracted from its own image to calculate the transformation. Then the vehicle could merge the bounding boxes from the infrastructure into its field of view. Experiments (Section 7.2.2) show that this process typically takes a data-sharing rate of only 72 Kbps. Fig. 2 shows two typical images from a driving vehicle and a traffic camera. Blue and green boxes show the perceived objects in the views of the traffic camera and the vehicle, respectively. Due to the occlusion, the vehicle cannot see the remaining 13 vehicles (blue boxes) while they are visible to the traffic camera. In contrast, the traffic camera has a broader field of view and is less prone to occlusion than vehicles. Therefore, autonomous vehicles can leverage the perception information from the traffic camera to achieve more comprehensive scene perception.

Note that when there are multiple traffic cameras at one intersection, *AutoMatch* can process the images from all cameras that may benefit the vehicle one by one and identify the one that is the most useful to the vehicle. This "naive" design is lightweight since Experiments (Section 7) show that the added computational overhead and communication overhead are extremely low compared to registration with one camera.

Centimeter-level localization via image registration between traffic cameras and HD maps. The second application of *AutoMatch* is the image registration between the traffic camera image and an HD map. Fig. 3(a) shows a traffic camera image and an HD map. An HD map is a highly accurate map where each pixel in it corresponds to a precise world position. HD maps are usually constructed using drones [29, 68] or map data collection cars equipped with high precision sensors (e.g. LiDARs, digital cameras and RTK GPS) [33, 65]. HD maps are able to achieve centimeter-level precision [31, 65]. Fig. 3(b) shows the image registration result between the HD map and the traffic camera image, which establishes a dense correspondence between the pixels in the traffic camera image and the points in the HD map. Given this correspondence, we could derive the 3D world position for each pixel in the traffic camera image, establishing a highly lightweight local map for the traffic camera, which is about the size of an image (around 1 MB). As a result, 1) we can easily find a vehicle's world position if the vehicle is in the traffic camera's field of view; 2) the vehicle doesn't have to match its sensor data with HD maps for localization which saves significant compute overhead. Note that the image registration between the traffic camera image and the HD map can be a one-time offline task. Once the registration is completed, the local map of the traffic camera is established. The local map is only related to the pose of the traffic camera, and hence remains unchanged as long as the traffic camera is still. To count for possible camera pose changes, the local map can be updated by periodically performing image registration between the traffic camera image and the HD map. Specifically, the traffic camera detects vehicles in view, derives, and broadcasts the world positions of these vehicles. Each vehicle obtains not only its own position but also the positions of other vehicles nearby, which is useful for downstream autonomous driving tasks such as path planning and collision avoidance. Vehicle identification is needed in this application. The infrastructure can use vehicle attributes such as color and type, or other techniques such as license plate recognition (LPR) [58] or RFID [67] to distinguish different vehicles. In order to meet the requirement of high-precision localization for autonomous driving, HD maps and traffic camera images need to be matched with pixel-level accuracy so that the localization error can be suppressed within centimeter-level [34].

3.3 Challenges

Despite the promising applications, the design of *AutoMatch* faces several major challenges in practice. First, there often exists significant scale and viewpoint gaps between the image pairs in the aforementioned two applications. The reason is that the working

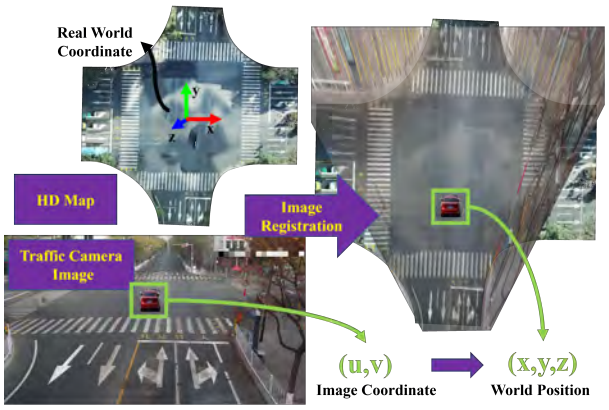


Figure 3: **A traffic camera image and an HD map generated from aerial images taken by a survey drone. The registration result can be used to localize vehicles from the traffic camera image.**

positions and orientations of the drone that constructs the HD map, the traffic camera, and the vehicle are usually different. Drones generally shoot vertically at a height of around one hundred meters from the ground. Traffic cameras are generally installed at a distance of about 10 m to shoot obliquely downward. On-vehicle cameras are usually installed at a height of about 1.5 m above the ground and are almost parallel to the ground. As a result, the resultant differences in scale, rotation, and viewpoint between two images will result in poor performance for existing image registration methods [19, 36], which is consistent with our experimental results (Section 7.4). Second, images captured in traffic scenes often contain a large number of repeated textures such as crosswalk lines, lane lines, etc., which unavoidably lead to similar keypoint patches and ambiguous keypoint features, resulting in a large number of false matches [36]. Third, existing image registration methods incur significant compute and communication overhead. They usually extract a large number of keypoints for every frame and describe them in the form of large-size feature vectors, which would need to be transmitted from infrastructure to vehicle. Moreover, existing image registration methods typically have high compute overhead, which cannot be used in autonomous driving scenarios with stringent real-time constraints such as tens of milliseconds of delay.

4 DESIGN OF AUTOMATCH

4.1 Motivation and Overview

Our design objective is to achieve pixel-level image registration in real-time with low communication overhead under challenging traffic camera-assisted autonomous driving scenarios. As a result, the image registration results of *AutoMatch* can assist the perception and localization of autonomous vehicles, which can benefit various downstream tasks for autonomous driving such as accident alarming, route planning, etc. Moreover, in practice, most traffic cameras are installed around intersections [16, 51]. Our key idea is to utilize landmark keypoints of domain-specific structures to match image pairs. Focusing on distinctive structures instead of the whole image helps to mitigate the adverse effects of large perspective variations on image registration and eliminate the ambiguity caused by the repeated contents. It also leads to high compute efficiency because less data is being processed. We select ground signs such as those shown in Fig. 5 as regions of interest (ROIs) and extract landmark keypoints inside each ROI. We then match the corresponding ROIs and landmark keypoints in the two input images to complete the registration. We focus on ground signs because they are: 1) usually required to present at intersections to show vehicle movements [63]; 2) sufficiently discriminative to serve as target structures for matching and less repetitive compared with other structures like crosswalk lines or lane lines; and 3) static structures and hence lead to a low compute overhead on infrastructure. This is because the points extracted from ground signs on the infrastructure side remain largely unchanged, which can be updated less frequently.

Accurately detecting the landmark keypoints of ground signs plays an important role in the performance of image registration. However, this is highly challenging due to a variety of imperfections in real-world settings: incompleteness caused by the limited field of view, occlusion of vehicles or other objects, stains caused by oil or water blobs, uneven lighting caused by shadows of trees or vehicles, or confusion with other objects such as speed bumps or manhole covers. These imperfections make the keypoint extraction from ground signs error-prone, as the yellow points shown in Fig. 5. On the other hand, humans can exploit prior knowledge of ground signs to robustly extract the locations of keypoints. This inspires us to apply the idea of landmark detectors to address the challenges faced by general keypoint detection. Landmark detection learns the prior shape and appearance of structured objects to localize a group of predefined points. There are numerous landmark detectors designed to locate the landmarks on human faces (e.g., eye corners, mouth corners, etc.) or bodies (e.g., shoulders, wrists, etc.).

However, despite the robustness, existing landmark detectors can not be directly used in our image registration method for the following two reasons. First, although all faces/bodies have the same landmark template, there are different categories of ground signs, and each category has a different landmark template. We thus design a novel unified landmark template applicable to all categories of ground signs. Second, landmark detectors can result in unsatisfactory landmark localization accuracy (shown as green points in Fig. 5). To address this issue, we propose a new module, i.e., the landmark keypoint extractor, to integrate the landmark detector with the general keypoint detector to benefit from both methods: robustness from the landmark detector and pixel-level localization accuracy from the general keypoint detector (see the red points in Fig. 5). One additional benefit of the landmark keypoint extractor is that the following landmark keypoint matching stage can be highly computationally efficient. Since all landmark keypoints inside two ground signs can be easily matched once the ground signs are matched (see Fig. 6), we only need to match the ground signs in the image, which reduces the search space to a large extent. Such efficiency of landmark keypoint matching lies in the fact that the descriptors of the landmark keypoints are implicitly encoded into the class of the ground sign and the index from the template point set.

The system architecture of *AutoMatch* is shown in Fig. 4. We first detect the regions of interest (ROI) in both images (Section 4.2). Then these ROIs are fed into a novel *landmark keypoint extractor* to extract landmark keypoints (Section 4.3), which contains a landmark detection branch, a general keypoint detection branch,

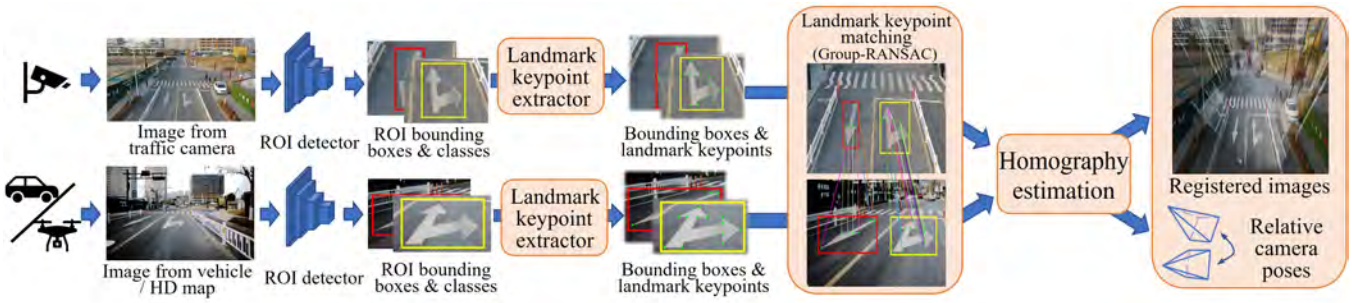


Figure 4: **Framework of our image registration approach for traffic camera-assisted autonomous driving.**

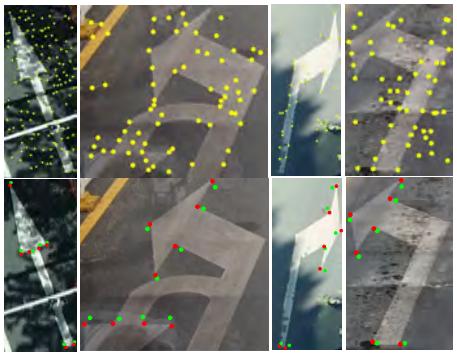


Figure 5: **Points detected by a general keypoint detector (yellow) [17], the landmark detector (green) and *AutoMatch* (red). Note the challenging conditions caused by occlusion, incompleteness, uneven lighting, and stains.**

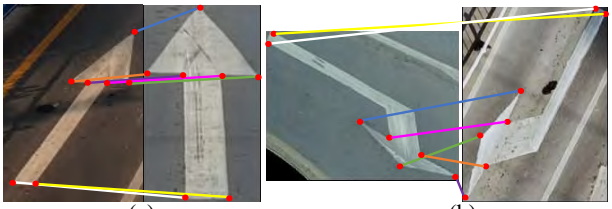


Figure 6: **Illustrations of the landmark keypoint correspondences between two matched ground signs.**

as well as a newly designed *Landmark-guided Non-Maximum Suppression (Landmark-guided NMS)* module to fuse the two detection results to obtain accurate landmark keypoints (see Fig. 7). Lastly, the landmark keypoint matching module (Section 4.4) based on a newly proposed *Group RANSAC* algorithm matches the ROIs and landmark keypoints extracted from previous steps.

4.2 Ground Sign Detector

Given input images, we first locate the region of interest and ignore the unrelated regions to improve the robustness and computational efficiency. We focus on ground signs because they are commonly present in complex traffic sections like intersections. Moreover, most traffic cameras are installed at busy intersections [16, 51]. We note that our approach can be easily extended to detect other traffic ground markers. We carefully categorize ground signs into seven classes: going straight, turning left, turning right, going straight or left, going straight or right, turning around, and turning around or left. We employ YOLOv4 [7], a real-time object detection model

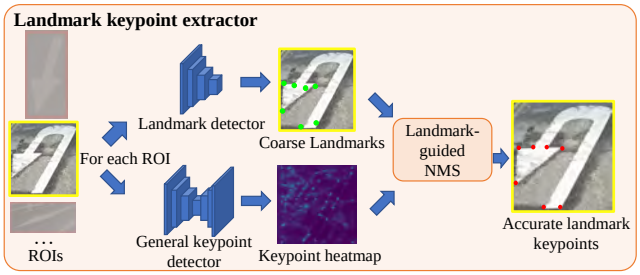


Figure 7: **Design of the landmark keypoint extractor.**

widely used in embedded sensing applications, to jointly detect the bounding boxes of all ground signs in each image and classify each sign into one of the seven categories. After detection, we crop the ground signs according to the detected bounding boxes, and each ground sign will be processed independently in the subsequent steps as shown in Fig. 7.

Our training dataset consists of two parts: 1/6 images of two self-collected datasets (Section 5) and the raw images of the "City" category in the autonomous driving dataset KITTI [23]. We annotate the bounding boxes and classes of the ground signs in images and finetune the YOLOv4 model in this dataset. Note that ground signs in different countries and regions may be slightly different, and the ground sign dataset in our method can be updated accordingly, which will not affect the generality and performance of our method.

4.3 Landmark Keypoint Extractor

The landmark keypoint extractor is designed to extract the landmark keypoints in each ground sign patch in the presence of the challenges illustrated in Fig. 5. The design of this module is motivated by the fact that general keypoint detection methods [6, 25, 35, 40, 54, 70] usually consider low-level local features, which will inevitably be affected by imperfections of ground signs and hence lead to noisy and unpredictable keypoints (see the yellow points in Fig. 5). In contrast, we propose to extract landmarks following a pre-defined landmark template. However, unlike facial landmark detection, which only has a single template, every class of ground signs has a unique shape. Therefore, We design **a unified landmark template** for all ground signs (shown in Fig. 8), which allows *AutoMatch* to reuse the existing landmark detection pipeline. Moreover, since landmark detection can only roughly localize each landmark but cannot achieve sub-pixel accuracy, we

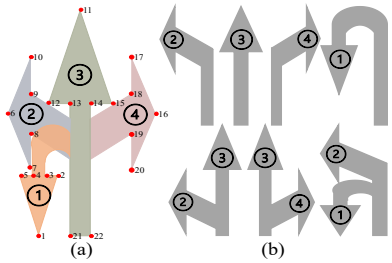


Figure 8: **Illustration of the unified landmark template (a) and some examples of ground signs that can be modeled using this template (b).**

propose to refine the result of landmark detection using a general keypoint detector. To this end, a **Landmark-guided NMS** algorithm is proposed to integrate both detectors to extract the final landmark keypoints, where the landmarks serve as guidance for picking the keypoints to achieve more accurate landmark keypoint localization. Such an approach enables both accurate and highly robust landmark keypoint extraction despite various interferences on ground sign appearances. We now discuss each component of the landmark keypoint extractor in detail.

4.3.1 Landmark Detector. We design a new landmark detector based on a real-time state-of-the-art facial landmark detector PFLD [24]. We zero-pad the ground sign patches before feeding them into the landmark detector to meet the aspect ratio requirement. To be able to generate landmarks with different templates, we design a unified landmark template as shown in Fig. 8. All categories of ground signs are stacked together with similar components merged, which results in a template with 4 components and a total number of 22 landmarks. Each landmark has its own ID number, which implicitly encodes rich semantic information. The neural network will predict the pixel locations of all 22 landmarks. The output landmarks of each ground sign class constitute a subset of these components, e.g., the turning left sign contains component 2, with a total of 7 landmarks. To achieve this, we define a binary mask M with a length of 22 for each category of ground sign to mask out unused landmarks. The mask is predefined and determined by the class of the ground sign. We then define the training loss as follows:

$$\mathcal{L} := \frac{1}{|M|N} \sum_{m=1}^{|M|} \sum_{n=1}^{N} M_m^n \left\| \mathbf{p}_m^n - \hat{\mathbf{p}}_m^n \right\|_2^2 \tag{1}$$

where $|M| = 22$ is the total number of landmarks and the subscript m indicates the m-th point. N denotes the batch size. $\mathbf{p}$ and $\hat{\mathbf{p}}$ are the ground truth and predicted locations of each landmark, respectively. This masked loss means that only the landmarks that fall into the current ground sign's category will contribute to the training loss. The same mask operation is performed in the inference stage, where only landmarks belonging to the category of the current ground sign are picked, and other landmarks are discarded.

To train the landmark detector, we crop the ground sign bounding boxes from the training dataset mentioned in Section 4.2. Then we resize and zero-pad them into patches of size 224×224 and then label the landmarks on them. During training, we also add a small random perturbation of homography transformations to each patch to augment the training examples.

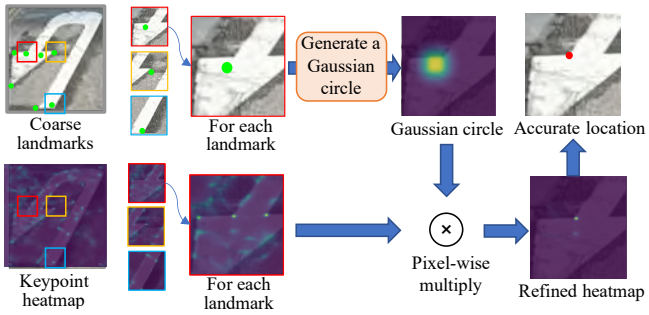


Figure 9: **Illustration of the Landmark-guided NMS method for combining the landmark detector and the general keypoint detector.**

4.3.2 Landmark-guided NMS. Despite robustness, the main limitation of the landmark detector is that the detected landmarks do not fall precisely on the corners of the ground sign (see green points in Fig. 5). To address this issue, we use the general keypoint detector to boost positioning accuracy. We adopt the widely-used general keypoint detector SuperPoint [17], a fast and lightweight model that computes accurate keypoint locations, which generates a keypoint response heatmap of the same size as the input. Each pixel of the heatmap corresponds to the probability of the pixel that is a keypoint. The training process is similar to the one in [17]. The difference is that our synthetic dataset only consists of structures with corners such as quadrilaterals, triangles, lines, and stars, which strengthens the detection of corner-like keypoints. The synthetic dataset is rendered on-the-fly, and no example is seen by the network twice.

We now have the landmarks from the landmark detector and the keypoint heatmap from the general keypoint detector. Landmarks capture the global structure and provide guidance for the positions of final landmark keypoints. By exploiting this property of landmarks, we look for the maximum response of the keypoint heatmap around each landmark, to fine-tune the position of landmarks for the final landmark keypoints. As a result, the final landmark keypoints not only inherit the landmarks' expression of the global structure but also precisely localize the corner points. Specifically, as shown in Fig. 9, we first generate a Gaussian distribution map centered at each landmark and multiply this Gaussian map with the keypoint heatmap pixel-wisely. The pixel with the maximum value in the map is selected as the final landmark keypoint $(\hat{u}, \hat{v})$. This operation filters out the keypoints far away from the landmark and allows the final landmark keypoints to have both rich semantics and accurate locations. Formally, this can be expressed as:

$$(\hat{u}, \hat{v}) = \underset{(u,v)}{\text{argmax}} G(u, v) \cdot H(u, v), \tag{2}$$

where

$$G(u, v) = \exp\left(-\left(\frac{(u - u_o)^2}{2\sigma^2} + \frac{(v - v_o)^2}{2\sigma^2}\right)\right) \tag{3}$$

is a Gaussian distribution centered on a landmark (u_o, v_o) and $H(u, v)$ represents the keypoint heatmap from the general keypoint detector.

4.4 Group RANSAC

After the previous modules of our pipeline, we now have the ground sign bounding boxes $\mathbb{A}$ and $\mathbb{B}$ in the two input images, as well as the

landmark keypoints belonging to each bounding box. To calculate the final homography $\mathbf{H}$, we need to find all the inlier correspondence between the landmark keypoints of the two images. We develop a fast landmark keypoint matching algorithm based on the traditional Random Sample Consensus (RANSAC) [22] algorithm. Unlike the classical RANSAC that randomly samples matched point pairs, we sample pairs of bounding boxes that have the same class. This is motivated by the fact that two matched signs must belong to the same class and share the same landmark template (see Fig. 6). We name our method *Group RANSAC*, where the landmark keypoints in a template are matched as a group. Specifically, we first randomly samples two bounding box pairs $(\mathbf{A}_1, \mathbf{B}_1)$ and $(\mathbf{A}_2, \mathbf{B}_2)$ from $\mathbb{A}$, and $\mathbb{B}$, respectively, so that the classes of each pair are the same, i.e., $Class(\mathbf{A}_i) = Class(\mathbf{B}_i)$ for $i = 1, 2$. We can now easily obtain the landmark keypoint correspondences from the bounding box pairs since the landmark keypoints of a bounding box are arranged in a fixed order as shown in Fig. 8. We then estimate the homography matrix $\mathbf{H}$ using all the corresponding landmark keypoint pairs obtained from the bounding box pairs. We check the correctness of the estimated $\mathbf{H}$ by counting the total number of inlier landmark keypoint pairs. Two landmark keypoints are defined as inlier point pairs if 1) they belong to bounding boxes of the same class, and 2) the reprojection error using $\mathbf{H}$ is smaller than a threshold. When the number of inlier landmark keypoint pairs is larger than a threshold, we finalize the algorithm by re-estimating $\mathbf{H}$ using all of the inlier landmark keypoint pairs. Otherwise, the current bounding box pairs are false matches, and we repeat all of the above steps to continue searching for correct bounding box pairs.

5 TESTBED AND DATASETS

We built a real-world testbed consisting of existing traffic cameras at intersections, DJI drones, and a self-built autonomous car (see Fig. 10). DJI drones are equipped with Ultra HD Lenses (Fig. 10(b)) for generating HD maps of intersections. Our self-built car (Fig. 10(a)) is equipped with a small computing unit with an Intel Core i7 CPU and multiple sensors, including two Pointgrey CM3-U3 cameras and three LiDARs (a Robosense RS32, a Robosense RS16, and a Livox AVIA). In this work, we use one camera of the car to capture images. As there is no dataset consisting of multi-view image pairs at intersections, i.e., traffic camera-HD map image pairs and traffic camera-vehicle image pairs, we collect two new multi-view intersection image datasets for traffic camera-assisted autonomous driving. One dataset is the traffic camera-vehicle dataset, which is collected for the evaluation of traffic camera-assisted vehicle perception. The other dataset is the traffic camera-HD map dataset, which is collected for the evaluation of traffic camera-assisted vehicle localization. We summarize our two datasets in Table 1. Below we describe the data acquisition process of each dataset in detail.

For the traffic camera-vehicle dataset, we manipulate our self-built autonomous car at a speed of 8 m/s to collect images of the vehicle's view around a city's intersections. Vehicle images are collected by the camera mounted on the car, which is about 1.5 m above the ground. Meanwhile, we collect the images of traffic cameras at these intersections. In total, we collected 4544 traffic camera-vehicle

(a) The self-built car equipped with sensors and computing unit. (b) The DJI drone used for constructing HD maps.

Figure 10: **Devices used in the system implementation.**

Table 1: **Summary of two datasets.**

Datasets	# intersections	# traffic cameras	# vehicle images	# HD maps	# image pairs
Traffic camera-vehicle dataset	19	48	4544	-	4544
Traffic camera-HD map dataset	32	172	-	32	172

Figure 11: **Two examples of the collected traffic camera images with different road types, road widths, and lighting conditions.**

image pairs from 48 traffic cameras at 19 intersections. For the traffic camera-HD map dataset, we use DJI drones to capture the aerial images of intersections at a speed of 9 m/s, and then generate HD maps of these intersections with centimeter-level accuracy using the drone image processing software ODM [4]. We also collect images captured by the traffic cameras at these intersections. In total, we collected traffic camera-HD map image pairs from 172 traffic cameras of 32 intersections in 21 cities. In addition, we label the corresponding points for each image pair in both two datasets manually to provide the ground-truth homography.

The collected images in the two datasets cover diverse and complex traffic scenarios with different road types (i.e., crossroads, T-junctions, highway entrances and exits), road widths (3 lanes to 12 lanes), road conditions (new or old, under construction or not), and lighting conditions (day and dusk). Some examples of traffic camera images are shown in Fig. 11. The private information such as street names, image acquisition timestamps, and license plates are removed by an independent third-party organization. This data collection is approved by the governing department of the city, and the study is approved by the ethics committee of the authors' institutes.

6 SYSTEM IMPLEMENTATION AND EXPERIMENT SETUP

This section introduces the system implementations of the two applications, i.e., traffic camera-assisted perception and traffic camera-assisted localization. In the first application, i.e., the traffic camera-assisted perception, we set up infrastructures and vehicles for perception fusion. We install an NVIDIA Jetson TX2 as the computing unit on 48 traffic cameras to collect and store the camera images at 25 fps. We implement *AutoMatch* on a laptop and use it as the computing platform at the vehicle end. The laptop is equipped with an Intel i7-9750H CPU and an NVIDIA RTX2060 Super GPU, whose computing capability lags far behind that of the mainstream computing platforms for autonomous driving, such as NVIDIA DRIVE AGX Pegasus [10, 48][1]. We collect the images from the vehicle camera at 30 fps and store them on the laptop for offline processing. Moreover, we use an 802.11ac WiFi router for wireless communication between the Jetson TX2 and the laptop to simulate the communication between the traffic camera and the vehicle. The data transfer takes place through UDP broadcasting, which transmits the infrastructure key points and perception information (object bounding boxes). The data transmission frequency is set to 2 Hz which is consistent with the frequency of decision-making on autonomous vehicles [32]. We simply discard extra frames that are not used for communication. The second application, i.e., traffic camera-assisted localization, requires image registration between traffic camera images and HD maps. This application is an offline task and can be implemented by running *AutoMatch* with traffic camera images and HD maps inputs.

We train the ground sign detector and the landmark keypoint detector with PyTorch [50] using the two datasets (Section 5) on a server equipped with Intel Xeon Silver 4210 CPU and one Nvidia RTX2080Ti GPU. The training of the two detectors takes around 20 hours in total. The implementing details can be found in Section 4. For inference, we export the trained models in ONNX format [2] using TensorRT [1] on Jetson TX2. For a brand-new region, the ground sign detector and the landmark keypoint detector need to be retrained or fine-tuned. Therefore the training overhead is roughly the same as the overhead we mentioned earlier. Considering the training is a one-time offline task, the overhead is reasonable in this setting.

7 EVALUATION

In this section, we first define evaluation metrics in Section 7.1. Then, we present an end-to-end evaluation of *AutoMatch* in Section 7.2. Next, we show application-level results in Section 7.3, which show that *AutoMatch* can not only significantly extend the vehicle's perception range but also provide vehicles with high-precision localization. In addition, we compare the performance of *AutoMatch* with other methods on two real-world multi-view intersection image datasets in Section 7.4. Finally, we conduct an ablation study to validate the effectiveness of our method in Section 7.5.

[1]NVIDIA DRIVE AGX Pegasus can achieve 320 TOPS (trillion operations per second) of computing capability, while that of NVIDIA GeForce RTX 2060 is only 14 TOPS.

7.1 Evaluation Metrics

7.1.1 Perception range gain and Field of View (FoV) gain. In order to measure how much autonomous vehicles can benefit from *AutoMatch* in perception, we define two application-level metrics, the *perception range gain* and the *FoV gain*. We project the vehicle image to the traffic camera image using the ground truth homography, and then calculate the two metrics in the traffic camera image coordinate. We quantify the two metrics in pixels instead of physical distances because 2D images cannot represent distances in the real world. Perception range gain is the increased ratio in distance before and after the image registration. It is defined as $(L_{traf}/L_{proj} - 1) \times 100\%$, where L_{traf} and L_{proj} are the lengths (in the vehicle's heading direction) of the traffic camera image and the projected vehicle image in pixels, respectively. FoV gain is the increased ratio in area, which is defined as $(N_{traf}/N_{proj} - 1) \times 100\%$, where N_{traf} and N_{proj} are the total pixel numbers of the traffic camera image and projected vehicle image, respectively.

7.1.2 RRE, RTE, and localization error. To evaluate the performance of *AutoMatch* in assisting the localization of autonomous vehicles, we first measure how accurate the traffic camera can localize vehicles in world coordinate, which is equivalent to measuring the accuracy of the dense correspondence between pixels in the traffic camera image and those in the HD map. Specifically, we measure localization error, the distance between the localized world position of the vehicle in the constructed traffic camera local map and the ground truth position of the vehicle in the HD map. This metric reflects the accuracy of the local map. Another metric to measure the performance of traffic camera-assisted vehicle localization is the accuracy of traffic camera pose estimation. Only if the pose estimation of the traffic camera itself is accurate can it accurately locate the vehicles in its field of view. The traffic camera pose can be derived based on the homography between the image and the HD map. We adopt two metrics - the relative rotation error (RRE) and the relative translational error (RTE) used in [13, 14, 20] to evaluate the errors of the estimated traffic camera poses. RRE is defined as:

$$\begin{aligned} E_R &= |\theta| + |\phi| + |\psi| \\ (\theta, \phi, \psi) &= F\left(R_T^{-1} R_E\right) \end{aligned} \tag{4}$$

where R_T and R_E are the rotation matrices decomposed from the ground-truth homography and the estimated homography, respectively. $F(\cdot)$ transforms a rotation matrix to three Euler angles (θ, ϕ, ψ). RRE is the sum of the absolute differences in three Euler angles. RTE is defined as: $E_T = \|t_T - t_E\|_2$, where t_T and t_E are the translation vectors decomposed from the ground-truth homography and the estimated homography, respectively.

7.1.3 Reprojection error and MMA. To compare the image registration performance with other algorithms, we follow the same methodology in [19, 41], which computes the reprojection error and the mean matching accuracy (MMA). Reprojection error is the Euclidean distance between the observed image point p and the image point p' reprojected from the other image. It reflects the accuracy of the estimated homography transformation. MMA is the average percentage of correct keypoint matches per image pair. A keypoint match is considered correct if its reprojection error estimated using the ground truth homography is below a given

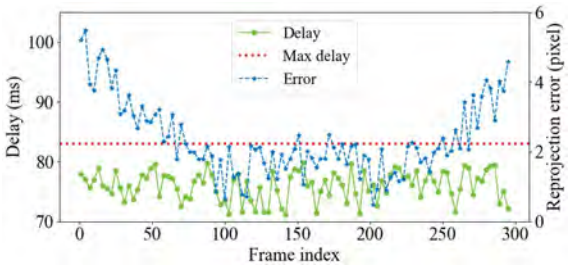


Figure 12: **The delay and reprojection error in an end-to-end evaluation experiment where our vehicle passes by roadside traffic cameras.**

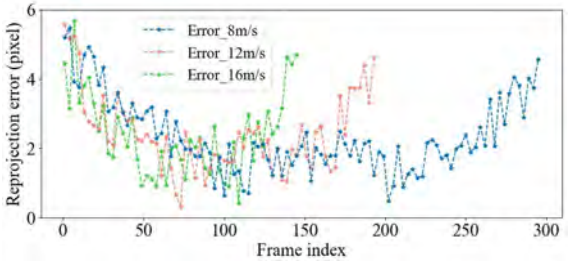


Figure 13: **The reprojection errors in end-to-end evaluation experiments where the vehicle goes at different speeds.**

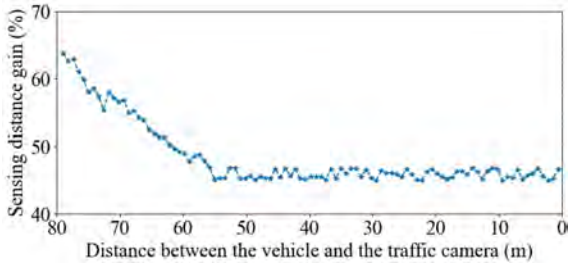


Figure 14: **Sensing distance gain in the process of the vehicle gradually approaching the traffic camera.**

threshold. This metric measures: 1) the repeatability of the keypoints: the same points in the two images need to be detected. 2) the distinguishability of the keypoints detected: two different keypoints that look similar should not be confused as one. 3) the quality of the matching algorithm.

7.2 End-to-End System Evaluation

7.2.1 Delay and error. To evaluate the end-to-end system performance of *AutoMatch*, we implement our system as described in Section 6. We take a typical process where a vehicle passes by an intersection as an example. The laptop continuously receives data from the Jetson TX2 and registers the traffic camera image with its own image. We record the results of the image registrations and then calculate the reprojection errors, the perception range gains, and the end-to-end delay. Note that the end-to-end delay includes both communication delay and processing time. The three metrics demonstrate the registration accuracy, the perception improvement, and the real-time performance of *AutoMatch*. Since delay is a major concern in autonomous driving, we also report the maximum delay among multiple experiments.

Fig. 12 shows that the maximum end-to-end delay is 82 *ms*, which is faster than the processing speed (typically 100 ms per image) of mainstream image-based visual tasks [12]. The results show that

Table 2: **The communication overhead of different methods for boosting vehicle perception.**

Methods	Data size for registration	Overall shared data size	Bandwidth needed
SIFT	73.8 KB	76.9 KB	1.2 Mbps
SuperGlue	121.2 KB	124.3 KB	2.0 Mbps
D2-Net	31.7 MB	31.7 MB	507.2 Mbps
AutoMatch	1.4 KB	4.5 KB	72 Kbps

AutoMatch is able to register the images from traffic cameras and vehicles in real time. We can also see that *AutoMatch* achieves pixel-level image registration between traffic camera images and vehicle images. The reprojection error can be a bit larger when the vehicle is far away from the traffic camera or when it drives away from the traffic camera. We also evaluate the performance of *AutoMatch* when the vehicle goes at different speeds. Fig. 13 shows that when the vehicle's speed is 16 m/s, the reprojection error is similar to that at 8 m/s. Fig. 14 shows the sensing distance gain vs. distance between the vehicle and the traffic camera. The result shows that when the vehicle has not yet entered the camera's field of view, the perception range can be increased by around 65%. This is because the perception improvement is more significant when the overlap of the two fields of view is small. The perception range gain becomes stable at around 45% when the vehicle enters the traffic camera's field of view.

7.2.2 Communication overhead. In the following evaluation, we compare *AutoMatch* with four image registration algorithms. These baselines have the same settings as *AutoMatch*: Input two images and output the homography between the two images. (i) SIFT [35], a traditional and the most widely used image registration algorithm; (ii) SuperGlue [52], an algorithm based on Graph Neural Network(GNN) proposed recently and also one of the state-of-the-art (SOTA) image registration algorithms; (iii) COTR [30], the latest registration algorithm based on transformer; (iv) D2-Net [19], a typical CNN-based algorithm. The implementation of SIFT is from OpenCV [8]. For the other three baselines, we used the codes published by the authors and adjusted the parameters to yield the best performance in our datasets.

We evaluate the communication overhead of *AutoMatch* for traffic camera-vehicle image registration by comparing it with these baselines. Communication is a simple channel from the traffic camera to the vehicle. The total data to be shared consists of two parts: one used for registration, i.e., the extracted keypoints and the keypoint descriptors, and the other is the perception information, which is implemented as the object bounding boxes. We do not compare with baseline *COTR* since it registers two images in an end-to-end manner, which requires the infrastructure to directly share raw images. We evaluate the average data volume used for registration and the overall average data volume shared between traffic cameras and vehicles. Besides, we also evaluate the communication bandwidth needed for each method. The frequency of data broadcasting from the traffic camera is set to 2 Hz as discussed in Section 5. Table 2 shows the evaluation results on the traffic camera-vehicle dataset. It can be seen that *AutoMatch* reduces the data volume for registration and the overall shared data volume by about 53× and 17× compared with SIFT baselines. *AutoMatch* only needs to transmit 4.5 KB data per frame to boost vehicle perception,

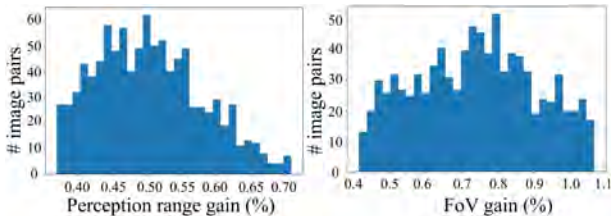


Figure 15: **Histograms of perception range gain and FoV gain before and after registration.**

among which only 31% are used for registration, compared to that of almost 100% for other baselines. The three baselines demonstrate high communication overhead since they extract massive keypoints and heavy descriptors for each keypoint. Besides, the bandwidth requirement of *AutoMatch* is as low as 72 Kbps, which can be easily supported by the current LTE network.

7.3 Application-Level Results

In this section, We first evaluate how much application-level perception and localization benefits *AutoMatch* can bring to autonomous vehicles using real traffic datasets. This evaluation supports our claims that: (i) *AutoMatch* implements and extends the vehicle's perception to areas that cannot be seen without the traffic camera-vehicle image registration; (ii) *AutoMatch* accurately constructs the local map of the traffic camera by matching the traffic camera image with an HD map, which enables the traffic camera to localize vehicles in its view. Then we discuss the robustness of *AutoMatch* to different lighting conditions and traffic conditions.

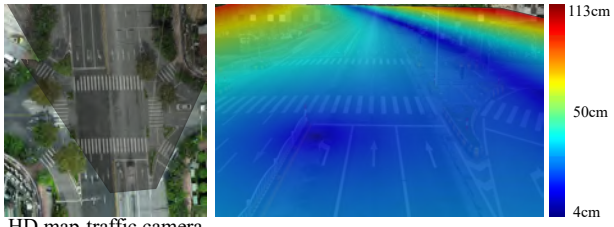
7.3.1 Boosting vehicle perception. For each traffic camera-vehicle image pair, we calculate the perception range gain and FoV gain of the vehicle after image registration. As the perception range gain and FoV gain vary under different situations (e.g., different scenes, the relative position between the vehicle and the traffic camera), we instead plot the distributions of these two metrics in Fig. 15. It can be seen that *AutoMatch* significantly improves autonomous vehicles' perception range by an average of 47.6%, and increases the vehicle's FoV by an average of 72.9%. In the best case, the FoV of the vehicle can be more than doubled.

7.3.2 High-precision vehicle localization. We present localization evaluation by comparing it with the four image registration algorithms introduced in Section 7.2.2. Table 3 shows the average RRE and RTE scores of *AutoMatch* and four baselines on the traffic camera-HD map dataset. The average RREs of all four baselines are more than 30°, while *AutoMatch* only generates 2.41° RRE. The average RTEs of baselines are larger than 42 cm while *AutoMatch* is less than 10 cm. The large RREs and RTEs from baselines introduce non-trivial challenges to localizing autonomous vehicles. In contrast, *AutoMatch* outperforms the four baselines by 7.7% and 22.73% ∼ 4.04% in average RREs and RTEs, respectively.

We then calculate the localization error of *AutoMatch* for localizing autonomous vehicles. To visualize the results intuitively, we visualize a localization error map in Fig. 16, which shows the localization error when a vehicle appears in different positions in the camera's field of view. In other words, the error map shows the accuracy of the local map. We can see that the localization error is smaller than 20 cm in 70% of the region. Note that at the top of the error map, the localization error is relatively large, because each pixel at that region typically occupies more than 15 cm in world space.

Table 3: **Traffic camera pose estimation results of baselines and *AutoMatch* on the traffic camera-HD map dataset.**

Methods	RRE	RTE
SIFT	102.71°	236.75 cm
SuperGlue	31.25°	42.11 cm
COTR	79.43°	57.43 cm
D2-Net	68.22°	52.39 cm
AutoMatch	***2.41°***	***9.57 cm***



HD map-traffic camera image registration

Localization error map

Figure 16: **A color-coded localization error map.**

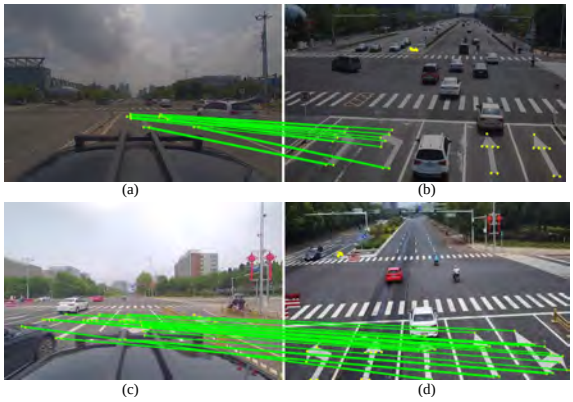


Figure 17: **Illustration of the robustness of *AutoMatch*. (a) and (b) show the result of *AutoMatch* in a dimly lit evening with a ground sign completely obscured by a white vehicle. (c) and (d) show the result in bright daytime with two ground signs partially obscured by two black vehicles.**

7.3.3 Performance under varied lighting and traffic conditions. Next, we use two typical results in the traffic camera-vehicle datasets to discuss the robustness of our system under different lighting conditions and traffic conditions. Heavy traffic may cause different degrees of occlusions of ground signs. Fig. 17 show two traffic camera-vehicle image pairs captured in the dimly lit evening and bright daytime respectively. Results show that *AutoMatch* can work well in different lighting conditions. This is because we apply data augmentation techniques such as brightness level changes, motion blur, and homography warps in the training process to improve *AutoMatch*'s robustness to lighting and viewpoint changes. For the case where the ground sign is fully or partially occluded, we also show two examples in Fig. 17(a,c). In Fig. 17(a), a ground sign is completely occluded by a white vehicle. There are also ground signs that are not visible from the traffic camera and the vehicle camera at the same time. Results show that our method successfully registers the images. This is achieved by the fact that the

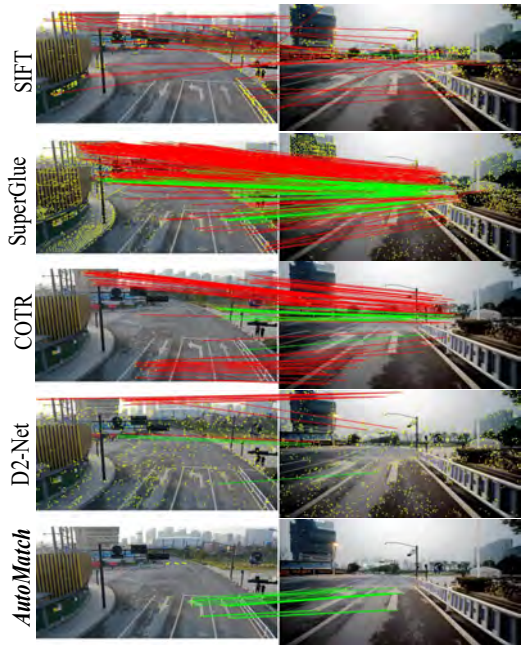


Figure 18: **Qualitative results of the four baselines and *AutoMatch* in a real traffic scene. *AutoMatch* detects fewer keypoints (yellow) while estimates all correct matches (green) without false matches (red).**

Group RANSAC algorithm maximizes the matches between the sets of ground signs in the two images without requiring them to be identical. In Fig. 17(c), two ground signs are partially obscured by vehicles. However, our method still successfully estimates the locations of occluded landmark keypoints thanks to the *landmark keypoint extractor*, which encodes the structure prior of the ground sign. In conclusion, the proposed system is robust to a certain level of occlusions or incompleteness.

7.4 Performance Comparison

We present extensive performance evaluations by comparing with the same four baselines in Section 7.3.2 on the two datasets. We visualize a typical example of the registration result in a real traffic scene in Fig. 18. We can see that all baselines produce hundreds or even thousands of keypoints but can only correctly match a few of them. The reason is that the baselines tend to extract keypoints on the roadside or distant buildings and trees, which are indistinguishable from each other or even not co-visible in both images. This not only makes the registration inefficient but also produces less accurate results due to lots of false matches. On the other hand, *AutoMatch* only focuses on landmark keypoints of ground signs and matches them accurately thanks to our landmark keypoint extractor.

Table 4 shows the numeric results of *AutoMatch* and other baselines on the two real traffic datasets, where we report the reprojection error, MMA, and run time. For reprojection error, *AutoMatch* is at most a quarter of the most accurate baseline. For MMA, all four baselines are less than 50%. In contrast, *AutoMatch* achieves more than 90% correct matches in both datasets. This is because the baselines tend to detect many irrelevant keypoints, thus lowering the distinguishability of the keypoints and increasing the difficulty of keypoint matching. While in *AutoMatch*, focusing on common

Table 4: **Results of different registration algorithms on the two real traffic datasets.**

Datasets	Methods	Reproj. error	Run time	MMA
Traffic camera-vehicle dataset	SIFT	218.256 px	7.440 s	17.58%
	SuperGlue	74.579 px	0.143 s	47.13%
	COTR	91.587 px	174.730 s	40.77%
	D2-Net	77.003 px	1.543 s	29.23%
	AutoMatch	***2.986 px***	***0.043 s***	***96.01%***
Traffic camera-HD map dataset	SIFT	143.476 px	0.629 s	12.39%
	SuperGlue	49.106 px	0.125 s	49.74%
	COTR	68.402 px	67.713 s	35.22%
	D2-Net	61.284 px	0.921 s	21.16%
	AutoMatch	***4.215 px***	***0.088 s***	***92.83%***

key structures allows us to detect landmark keypoints that have high overlap rates in both images, and the explicit semantics of the landmark keypoints allows us to match them easily. For run time, *AutoMatch* is 1.42 to 4063 times faster than other baselines. The run time of COTR is significantly longer than the other three baselines due to the use of transformer architecture. We can also see that *AutoMatch*'s performance on the traffic camera-HD map dataset is slightly worse than that on the traffic camera-vehicle dataset. This is reasonable because HD maps have much higher resolutions than vehicle images, i.e., 7900×7900 vs. 1920×1080, and cover a broader range. High resolution naturally leads to numerically larger reprojection errors as the reprojection error is evaluated on the pixel. Broader range results in more matching ground sign pairs between HD maps and traffic camera images, which further leads to a longer search time for the landmark keypoint matching module, and finally results in a longer run time.

Fig. 19 shows some qualitative image registration results on the traffic camera-HD map dataset, which shows that *AutoMatch* achieves more precise image registration results compared to other baselines (see the anastomosis of crossroads). Note that *AutoMatch* not only focuses on the ground sign structures nearby but also manages to match ground signs at distance, which further improves the registration accuracy.

7.5 Ablation study

We validate our landmark keypoint extractor with an ablation study. The ablation aims to prove the effectiveness of our design of integrating the landmark detector with the general keypoint detector. We compare our full landmark keypoint extractor (*Full*) with ablations that with only the landmark detector (*LD only*) or only the general keypoint detector (*GKD only*) to generate the final keypoints. Other modules of our pipeline are kept unchanged. Note that when we extract keypoints using only the general keypoint detector, the keypoints are unstructured and thus can not be used in the proposed *Group RANSAC*. Therefore, we adopt *SuperGlue* and Nearest Neighbor search (*NN*) [45] as the keypoint matching methods when we experiment the *GKD only*. We report reprojection error, MMA, keypoint detection run time, and keypoint matching run time on the traffic camera-vehicle dataset at Table. 5.

We can see that while being slightly slower than others in terms of keypoint detection run time, our *Full* model achieves the smallest reprojection error and highest MMA. And the proposed *Group*

Figure 19: **Registration results between an HD map and a traffic camera image in the real traffic scene.**

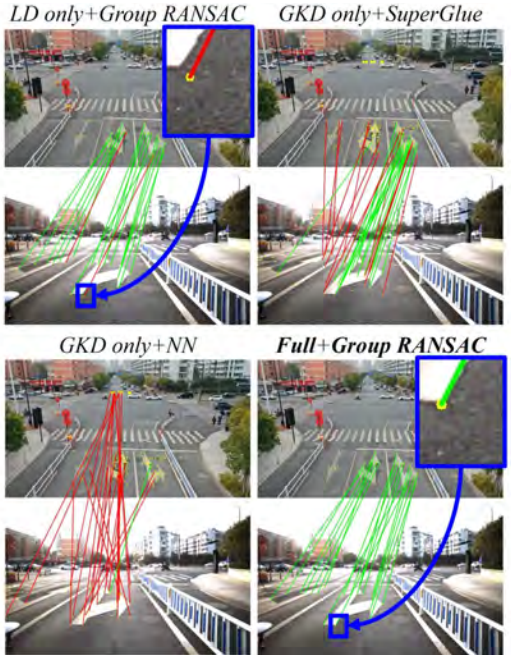


Figure 20: **Qualitative results of ablation study. Note that the *LD only + Group RANSAC* tends to detect inaccurate landmark locations as highlighted in blue.**

RANSAC achieves at least two orders of magnitude faster than the SOTA matching algorithm *SuperGlue*. We also visualize the matches in Fig. 20. We can see that without the guidance of landmarks, *GKD only + SuperGlue* and *GKD only + NN* produce many noisy and indiscriminative keypoints and further lead to numerous false matches, which are consistent with the quantitative results in Table 5. On the other hand, if we only use the landmark detector (*LD only + Group RANSAC*), although the landmarks are correctly matched, as highlighted in blue in Fig. 20, they suffer from inaccurate location, which causes performance degradation. By contrast, our *Full* model predicts accurate structured keypoint locations and matches all of them correctly by combining the benefits of the general keypoint detector and the landmark detector. Besides, it is also worth noticing that the performance of *GKD only + SuperGlue* is significantly better than the *SuperGlue* in Table 4. They share the same pipeline with the only difference being that the *GKD only + SuperGlue* works on bounding boxes instead of the whole image, which validates our core idea of focusing on key structures instead of the whole image.

Table 5: **Quantitative results of ablation study.**

Methods		Reproj. error	MMA	Detection time (ms)	Matching time (ms)
Detector	Matching				
LD only	*Group RANSAC*	6.53 px	92.60%	37.91	0.23
GKD only	*SuperGlue*	17.65 px	72.01%	***36.24***	24.61
GKD only	*NN*	53.48 px	57.89%	36.51	0.54
Full	***Group RANSAC***	***2.99 px***	***96.01%***	43.24	***0.21***

8 CONCLUSION AND FUTURE WORK

In conclusion, we present *AutoMatch*, the first system that matches traffic camera-vehicle image pairs or traffic camera-HD map image pairs at pixel-level accuracy with low communication/compute overhead in real-time, which is a key technology for leveraging traffic camera for assisting the perception and localization of autonomous driving. Extensive evaluations on two self-collected datasets show that *AutoMatch* outperforms SOTA baselines in robustness, accuracy, and efficiency. In the future, we will extend our approach to integrate the perceptions of multiple cameras which are typically installed in different directions around a road intersection. We will also study how to leverage such results to assist the perception and localization of autonomous vehicles.

REFERENCES

[1] n.d.. Nvidia TENSORRT. https://developer.nvidia.com/tensorrt.
[2] n.d.. Open Neural Network Exchange. https://onnx.ai/.
[3] Eduardo Arnold, Mehrdad Dianati, Robert de Temple, and Saber Fallah. 2020. Cooperative perception for 3D object detection in driving scenarios using infrastructure sensors. *IEEE Transactions on Intelligent Transportation Systems* (2020).
[4] OpenDroneMap Authors. 2020. ODM - A command line toolkit to generate maps, point clouds, 3D models and DEMs from drone, balloon or kite images. https://github.com/OpenDroneMap/ODM.
[5] Vassileios Balntas, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. 2016. Learning local feature descriptors with triplets and shallow convolutional neural networks.. In *Bmvc*, Vol. 1. 3.
[6] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. 2006. Surf: Speeded up robust features. In *European conference on computer vision*. Springer, 404–417.
[7] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. 2020. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934* (2020).
[8] Gary Bradski and Adrian Kaehler. 2008. *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc.".
[9] Matthew Brown, Gang Hua, and Simon Winder. 2010. Discriminative learning of local image descriptors. *IEEE transactions on pattern analysis and machine intelligence* 33, 1 (2010), 43–57.
[10] Andrew Burnes. 2019. Introducing GeForce RTX SUPER Graphics Cards: Best In Class Performance, Plus Ray Tracing. https://www.nvidia.com/en-us/geforce/news/geforce-rtx-20-series-super-gpus/.
[11] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2017. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7291–7299.
[12] Long Chen, Shaobo Lin, Xiankai Lu, Dongpu Cao, Hangbin Wu, Chi Guo, Chun Liu, and Fei-Yue Wang. 2021. Deep neural network based vehicle and pedestrian detection for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems* 22, 6 (2021), 3234–3246.

[13] Christopher Choy, Wei Dong, and Vladlen Koltun. 2020. Deep global registration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2514–2523.

[14] Christopher Choy, Jaesik Park, and Vladlen Koltun. 2019. Fully Convolutional Geometric Features. In *ICCV*.

[15] Xiao Chu, Wei Yang, Wanli Ouyang, Cheng Ma, Alan L Yuille, and Xiaogang Wang. 2017. Multi-context attention for human pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1831–1840.

[16] BRITISH COLUMBIA. 2019. Where intersection safety cameras are located. https://www2.gov.bc.ca/gov/content/transportation/driving-and-cycling/roadsafetybc/intersection-safety-cameras/where-the-cameras-are.

[17] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. 2018. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 224–236.

[18] Jingming Dong and Stefano Soatto. 2015. Domain-size pooling in local descriptors: DSP-SIFT. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5097–5106.

[19] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. 2019. D2-net: A trainable cnn for joint description and detection of local features. In *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*. 8092–8101.

[20] G. Elbaz, T. Avraham, and A. Fischer. 2017. 3D Point Cloud Registration for Localization Using a Deep Neural Network Auto-Encoder. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2472–2481. https://doi.org/10.1109/CVPR.2017.265

[21] Alessio Fascista, Giovanni Ciccarese, Angelo Coluccia, and Giuseppe Ricci. 2017. Angle of arrival-based cooperative positioning for smart vehicles. *IEEE Transactions on Intelligent Transportation Systems* 19, 9 (2017), 2880–2892.

[22] Martin A. Fischler and Robert C. Bolles. 1981. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun. ACM* 24, 6 (June 1981), 381–395. https://doi.org/10.1145/358669.358692

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

[24] Xiaojie Guo, Siyuan Li, Jinke Yu, Jiawan Zhang, Jiayi Ma, Lin Ma, Wei Liu, and Haibin Ling. 2019. PFLD: A practical facial landmark detector. *arXiv preprint arXiv:1902.10859* (2019).

[25] Chris Harris, Mike Stephens, et al. 1988. A combined corner and edge detector. In *Alvey vision conference*. Citeseer, 10–5244.

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

[27] Jared Heinly, Johannes L Schonberger, Enrique Dunn, and Jan-Michael Frahm. 2015. Reconstructing the world* in six days*(as captured by the yahoo 100 million image dataset). In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3287–3295.

[28] INSIDER. 2016. Here's why self-driving cars can't handle bridges. <http://www.businessinsider.com/autonomous-cars-bridges-2016-8.

[29] Mahdi Javanmardi, Ehsan Javanmardi, Yanlei Gu, and Shunsuke Kamijo. 2017. Towards high-definition 3D urban mapping: Road feature-based registration of mobile mapping systems and aerial imagery. *Remote Sensing* 9, 10 (2017), 975.

[30] Wei Jiang, Eduard Trulls, Jan Hosang, Andrea Tagliasacchi, and Kwang Moo Yi. 2021. Cotr: Correspondence transformer for matching across images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 6207–6217.

[31] Jialin Jiao. 2018. Machine learning assisted high-definition map creation. In *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, Vol. 1. IEEE, 367–373.

[32] Felix Kam and Henrik Mellin. 2019. Different frequencies of maneuver replanning on autonomous vehicles.

[33] I Karls and M Mueck. 2018. Networking vehicles to everything. Evolving automotive solutions.

[34] S. Kuutti, S. Fallah, K. Katsaros, M. Dianati, F. Mccullough, and A. Mouzakitis. 2018. A Survey of the State-of-the-Art Localization Techniques and Their Potentials for Autonomous Vehicle Applications. *IEEE Internet of Things Journal* 5, 2 (2018), 829–846. https://doi.org/10.1109/JIOT.2018.2812300

[35] David G Lowe. 2004. Distinctive image features from scale-invariant keypoints. *International journal of computer vision* 60, 2 (2004), 91–110.

[36] Jiayi Ma, Xingyu Jiang, Aoxiang Fan, Junjun Jiang, and Junchi Yan. 2021. Image matching from handcrafted to deep features: A survey. *International Journal of Computer Vision* 129, 1 (2021), 23–79.

[37] Juliette Marais, Cyril Meurie, Dhouha Attia, Yassine Ruichek, and Amaury Flancquart. 2014. Toward accurate localization in guided transport: Combining GNSS data and imaging information. *Transportation Research Part C: Emerging Technologies* 43 (2014), 188–197.

[38] Iaroslav Melekhov, Aleksei Tiulpin, Torsten Sattler, Marc Pollefeys, Esa Rahtu, and Juho Kannala. 2019. Dgc-net: Dense geometric correspondence network. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 1034–1042.

[39] Krystian Mikolajczyk and Cordelia Schmid. 2004. Scale & affine invariant interest point detectors. *International journal of computer vision* 60, 1 (2004), 63–86.

[40] Krystian Mikolajczyk and Cordelia Schmid. 2004. Scale & affine invariant interest point detectors. *International journal of computer vision* 60, 1 (2004), 63–86.

[41] Krystian Mikolajczyk and Cordelia Schmid. 2005. A performance evaluation of local descriptors. *IEEE transactions on pattern analysis and machine intelligence* 27, 10 (2005), 1615–1630.

[42] Krystian Mikolajczyk, Tinne Tuytelaars, Cordelia Schmid, Andrew Zisserman, Jiri Matas, Frederik Schaffalitzky, Timor Kadir, and L Van Gool. 2005. A comparison of affine region detectors. *International journal of computer vision* 65, 1 (2005), 43–72.

[43] Yanghui Mo, Peilin Zhang, Zhijun Chen, and Bin Ran. 2021. A method of vehicle-infrastructure cooperative perception based vehicle state information fusion using improved kalman filter. *Multimedia Tools and Applications* (2021), 1–18.

[44] Marius Muja and David G Lowe. 2009. Fast approximate nearest neighbors with automatic algorithm configuration. *VISAPP (1)* 2, 331-340 (2009), 2.

[45] Marius Muja and David G Lowe. 2014. Scalable nearest neighbor algorithms for high dimensional data. *IEEE transactions on pattern analysis and machine intelligence* 36, 11 (2014), 2227–2240.

[46] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. 2015. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE transactions on robotics* 31, 5 (2015), 1147–1163.

[47] Alejandro Newell, Kaiyu Yang, and Jia Deng. 2016. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*. Springer, 483–499.

[48] NVIDIA. 2022. HARDWARE FOR SELF-DRIVING CARS. https://www.nvidia.com/en-us/self-driving-cars/drive-platform/hardware/.

[49] Giuseppe Palestra, Adriana Pettinicchio, Marco Del Coco, Pierluigi Carcagnì, Marco Leo, and Cosimo Distante. 2015. Improved performance in facial expression recognition using 32 geometric features. In *International Conference on Image Analysis and Processing*. Springer, 518–528.

[50] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019).

[51] radenso. 2021. What's the difference between traffic cameras, red light cameras, and speed cameras? https://radenso.com/blogs/radar-university/what-s-the-difference-between-traffic-cameras-red-light-cameras-and-speed-cameras.

[52] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. 2020. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 4938–4947.

[53] Nikolay Savinov, Akihito Seki, Lubor Ladicky, Torsten Sattler, and Marc Pollefeys. 2017. Quad-networks: unsupervised learning to rank for interest point detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1822–1830.

[54] Nikolay Savinov, Akihito Seki, Lubor Ladicky, Torsten Sattler, and Marc Pollefeys. 2017. Quad-networks: unsupervised learning to rank for interest point detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1822–1830.

[55] Johannes L Schonberger and Jan-Michael Frahm. 2016. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4104–4113.

[56] Heiko G Seif and Xiaolong Hu. 2016. Autonomous driving in the iCity—HD maps as a key challenge of the automotive industry. *Engineering* 2, 2 (2016), 159–162.

[57] Edgar Simo-Serra, Eduard Trulls, Luis Ferraz, Iasonas Kokkinos, Pascal Fua, and Francesc Moreno-Noguer. 2015. Discriminative learning of deep convolutional feature point descriptors. In *Proceedings of the IEEE international conference on computer vision*. 118–126.

[58] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[59] Yi Sun, Xiaogang Wang, and Xiaoou Tang. 2013. Deep convolutional network cascade for facial point detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3476–3483.

[60] The N.Y. Times. 2017. Building a road map for the self-driving car. <https://www.nytimes.com/2017/03/02/automobiles/wheels/selfdriving-cars-gps-maps.html.

[61] Prune Truong, Martin Danelljan, Luc V Gool, and Radu Timofte. 2020. GOCor: Bringing globally optimized correspondence volumes into your neural network. *Advances in Neural Information Processing Systems* 33 (2020), 14278–14290.

[62] Prune Truong, Martin Danelljan, and Radu Timofte. 2020. GLU-Net: Global-local universal network for dense flow and correspondences. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 6258–6268.

[63] Federal Highway Administration U.S. Department of Transportation. 2002. United States Pavement Markings. https://mutcd.fhwa.dot.gov/services/publications/fhwaop02090/index.htm.

[64] Jessica Van Brummelen, Marie O'Brien, Dominique Gruyer, and Homayoun Najjaran. 2018. Autonomous vehicle perception: The technology of today and tomorrow. *Transportation research part C: emerging technologies* 89 (2018), 384–406.

[65] Harsha Vardhan. 2017. HD Maps: New age maps powering autonomous vehicles. *Geospatial world* 22 (2017).

[66] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. 2016. Convolutional pose machines. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 4724–4732.

[67] Ron Weinstein. 2005. RFID: a technical overview and its application to the enterprise. *IT professional* 7, 3 (2005), 27–33.

[68] Andi Zang, Runsheng Xu, Zichen Li, and David Doria. 2017. Lane boundary extraction from satellite imagery. In *Proceedings of the 1st ACM SIGSPATIAL Workshop on High-Precision Maps and Intelligent Applications for Autonomous Vehicles*. 1–8.

[69] Linguang Zhang and Szymon Rusinkiewicz. 2018. Learning to detect features in texture images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 6325–6333.

[70] Linguang Zhang and Szymon Rusinkiewicz. 2018. Learning to detect features in texture images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 6325–6333.

[71] Xumiao Zhang, Anlan Zhang, Jiachen Sun, Xiao Zhu, Y Ethan Guo, Feng Qian, and Z Morley Mao. 2021. Emp: Edge-assisted multi-vehicle perception. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*. 545–558.

[72] Erjin Zhou, Haoqiang Fan, Zhimin Cao, Yuning Jiang, and Qi Yin. 2013. Extensive facial landmark localization with coarse-to-fine convolutional network cascade. In *Proceedings of the IEEE international conference on computer vision workshops*. 386–391.