Online Correction of Camera Poses for the Surround-view System: A Sparse Direct Approach

TIANJUN ZHANG, HAO DENG, LIN ZHANG, and SHENGJIE ZHAO, School of Software Engineering, Tongji University, China
XIAO LIU, College of Information and Computer Sciences, University of Massachusetts Amherst, USA
YICONG ZHOU, Department of Computer and Information Science, University of Macau, China

The surround-view module is an indispensable component of a modern advanced driving assistance system. By calibrating the intrinsics and extrinsics of the surround-view cameras accurately, a top-down surround-view can be generated from raw fisheye images. However, poses of these cameras sometimes may change. At present, how to correct poses of cameras in a surround-view system online without re-calibration is still an open issue. To settle this problem, we introduce the sparse direct framework and propose a novel optimization scheme of a cascade structure. This scheme is actually composed of two levels of optimization and two corresponding photometric error based models are proposed. The model for the first-level optimization is called the ground model, as its photometric errors are measured on the ground plane. For the second level of the optimization, it's based on the so-called ground-camera model, in which photometric errors are computed on the imaging planes. With these models, the pose correction task is formulated as a nonlinear least-squares problem to minimize photometric errors in overlapping regions of adjacent bird's-eye-view images. With a cascade structure of these two levels of optimization, an appropriate balance between the speed and the accuracy can be achieved. Experiments show that our method can effectively eliminate the misalignment caused by cameras' moderate pose changes in the surround-view system. Source code and test cases are available online at https://cslinzhang.github.io/CamPoseCorrection/.

CCS Concepts: • **Computing methodologies → Camera calibration**; *Scene understanding;*

Additional Key Words and Phrases: Surround-view system, direct method, cascade structure, photometric error minimization

This work was supported in part by the National Natural Science Foundation of China under Grants 61973235 and 61936014, in part by the Natural Science Foundation of Shanghai under Grant 19ZR1461300, in part by the Shanghai Science and Technology Innovation Plan under Grant 20510760400, in part by the Dawn Program of Shanghai Municipal Education Commission under Grant 21SG23, in part by the Shanghai Municipal Science and Technology Major Project under Grant 2021SHZDZX0100, and in part by the Fundamental Research Funds for the Central Universities.
Authors' addresses: T. Zhang, H. Deng (corresponding author), L. Zhang (corresponding author), and S. Zhao, School of Software Engineering, Tongji University, No. 4800, Caoan RD., Shanghai, Shanghai 200092, China; emails: {1911036, denghao1984, cslinzhang, shengjiezhao}@tongji.edu.cn; X. Liu, College of Information and Computer Sciences, University of Massachusetts Amherst, 141 Commonwealth Ave, Amherst, MA 01002, USA; email: xiaoliu1990@cs.umass.edu; Y. Zhou, Department of Computer and Information Science, University of Macau, Avenida da Universidade, Taipa, Macau, Macau 999078, China; email: yicongzhou@um.edu.mo.

1551-6857/2022/02-ART106 $15.00
https://doi.org/10.1145/3505252

ACM Reference format:
Tianjun Zhang, Hao Deng, Lin Zhang, Shengjie Zhao, Xiao Liu, and Yicong Zhou. 2022. Online Correction of Camera Poses for the Surround-view System: A Sparse Direct Approach. *ACM Trans. Multimedia Comput. Commun. Appl.* 18, 4, Article 106 (February 2022), 24 pages.
https://doi.org/10.1145/3505252

1 INTRODUCTION

Typically, a surround-view system consists of four to six fisheye cameras. By calibrating its intrinsics and extrinsics accurately, we can synthesize high-quality surround-view images at runtime from fisheye images by multiview geometry knowledge [13]. A surround-view system can help reduce a driver's blind spots and make driving safer and more convenient. In addition, in recent years, surround-views have been widely used in driving assistance tasks, such as parking-slot detection [23, 42], autonomous parking [24, 41] and pedestrian detection [11, 18]. These tasks usually play important roles in driving assistance and their performance can be affected much by the quality of surround-view images.

When cameras in a surround-view system are offline calibrated [22, 34], they are supposed to be fixed to keep their relative poses unchanged. However, due to collisions, bumps, tire pressure changes and other factors, camera poses may alter indeed afterwards. If we do not adjust cameras' extrinsics accordingly, there will be observable geometric misalignment in the generated surround-view. In most commercial solutions, to eliminate such geometric misalignment, drivers have to drive to the auto service stores and to re-calibrate the vehicles by professionals. This is undoubtedly quite troublesome for both customers and automobile manufacturers. Thus, many automobile manufacturers now are looking for online methods to correct a surround-view system's extrinsics. Unfortunately, at present, relevant studies are quite few in this field. In this paper, we attempt to tackle this problem to some extent by proposing an online scheme for correcting camera poses of a surround-view system without resorting to re-calibration. More importantly, our online camera pose correction scheme relies totally on minimizing photometric errors without the need for additional physical equipment or calibration sites. Therefore, it can be easily integrated into pipelines of existing surround-view systems to improve their robustness and stability.

The remainder of this paper is organized as follows. Section 2 introduces the related work and our contributions. Section 3 makes an overview of the surround-view calibration pipeline. Section 4 presents our proposed approach in details. Experimental results are reported in Section 5. Finally, Section 6 concludes the paper.

2 RELATED WORK AND OUR CONTRIBUTIONS

2.1 Online Pose Correction for the Multi-camera System

The surround-view system is a special kind of multi-camera system. A multi-camera system consists of at least two cameras. Before using such a system, we often need to calibrate the intrinsics of its cameras. In addition, most of the multi-camera systems require extrinsics calibration to estimate the relative poses among cameras [20]. When one or more cameras move after calibration, the extrinsics of the multi-camera system will definitely change and we need to correct them. Existing online camera pose correction schemes for multi-camera systems roughly fall into three categories, odometry based ones, lane-line based ones and bundle adjustment based ones.[1]

Odometry based methods. The odometry based methods resort to a visual odometry or a complete SLAM system to correct the poses of the camera system. In [38], Schneider et al. proposed a

[1]The traits of the methods reviewed in this subsection along with the one proposed in this article are given in Table 2.

method of resolving cameras' extrinsics based on localization results of a visual odometry. It has a merit that it can be applicable to not only cameras but also lidars. However, it takes about 500 frames for the system to converge. Heng et al. [14, 15] proposed an infrastructure-based calibration pipeline. With their scheme, a vehicle equipped with a surround-view system needs to travel in the calibration area for a while to establish the map.

In fact, the online camera pose correction problem can be regarded as a variant of the SLAM problem, so these methods are theoretically feasible. However, since the operation of the SLAM system will occupy a significant amount of computation resources, these methods somewhat do not have the necessary portability. In addition, in the SLAM system, the construction of stable maps usually takes much time. Therefore, in real application scenarios, such methods are usually unlikely to meet the industrial portability requirements.

Lane-line based methods. The lane-line based methods rely on a strong assumption, that is two parallel lane-lines on the ground can be captured by the cameras. One of the earliest works in this field is Collado et al.'s in [4]. Collado et al. used the Sobel operator and the Hough transform to extract the calibration pattern from two parallel ground lanes, and then with the pattern, they estimated the extrinsics of the stereo-vision cameras. The solution proposed in [32] first estimates the vanishing point based on two lanes parallel to each other on the flat ground, and then with the estimated vanishing point, the pose of the multi-camera system relative to the world coordinate system is solved. In Hold et al.'s work [17], a method of online extrinsics calibration also using ground lanes was proposed. To begin with, they detected the lane and obtained a series of feature points by sampling the lane with the scanning line. Then, fast Fourier transform was adopted to measure the distance of lane points, and finally, they made use of lane points to solve the cameras' extrinsics. In [44], Zhao et al. proposed to utilize multiple vanishing points of lane markings for calibrating cameras' orientations. Their approach performs better in accuracy compared with the previous competitors, but just as the aforementioned lane-line based solutions, it's still not applicable to the surround-view system. In [3], Choi et al. designed an online extrinsics calibration pipeline for the surround-view case, in which the surround-view system was calibrated by aligning lane markings across images of adjacent cameras.

As we mentioned earlier, this type of lane-line based solutions make an assumption for the working environment, that is, there must be two parallel lane-lines clearly observed in the field of view. However, this is an assumption that cannot usually be satisfied. For example, when a car is running on a road without lane-lines or in an underground parking lot, this assumption will be broken. Therefore, the application scope of these lane-line based frameworks is quite limited.

Bundle adjustment based methods. The existing bundle adjustment based methods all follow a similar basic pipeline. First, feature extraction and matching are performed on images collected by different cameras. Then 3D positions of the points are determined by triangulating the paired 2D features. Finally, by bundle adjustment, the reprojection error is minimized, thereby optimizing the camera poses. It is worth mentioning that the bundle adjustment is not unique to this kind of scheme, and is also often used in the odometry based ones. The bundle adjustment based methods discussed here refer to those ones that do not include the complete front-end and back-end but instead only use the bundle adjustment technology to achieve pose correction.

Dang et al. [5] presented an approach for continuous self-calibration of the stereo-vision cameras. Three different categories of constraint equations were formulated as a Gauss-Helmert model for the self-recalibration task, bundle adjustment with reduced parameter vector, the epipolar constraint, and the trilinear constraints. In [12], Hansen et al. proposed an online extrinsics calibration method based on sparse feature matching using a sequence of frames. They first acquired the initial estimation of extrinsics by minimizing the epipolar error in a single

frame. Then, they optimized the states of the following multi-frames by the Extended Kalman Filter. Knorr et al. [21] established a recursive optimization algorithm, in which relative camera poses were corrected by the Extended Kalman Filter, and the relationship between the camera system and the ground was determined via homography estimation. Both Hansen et al.'s and Knorr et al.'s methods resort to the Extended Kalman Filter, so a sequence of frames are required for them to converge. In Ling and Shen's approach [25], the initial calibration result was taken as the starting point, the epipolar error was minimized by non-linear optimization, and the accuracy of calibration was evaluated by the minimum eigenvalue of the covariance matrix. It is worth mentioning that their method takes cameras in the system as a whole and supposes that relative poses among the cameras are fixed and will not change. Consequently, it actually does not consider the relative pose optimization between cameras.

Aforementioned bundle adjustment based schemes are all online approaches to re-calibrate or to optimize the extrinsics of a multi-camera system via "bundle adjustment". However, they are all designed for common multi-camera systems. Although the surround-view system is also a multi-camera system, unfortunately, these schemes are usually not directly applicable to it due to its following characteristics compared with conventional multi-camera systems:

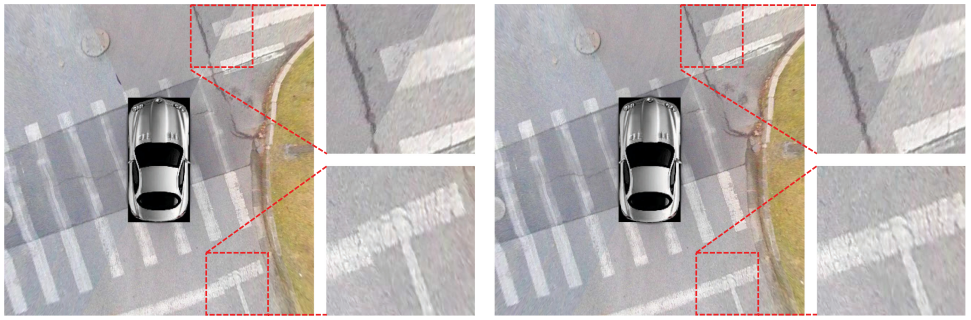
(1) The wide-angle fisheye camera is often used in the surround-view system. Compared with ordinary cameras, the distortion of fisheye cameras is much more serious and more difficult to be eliminated completely. This phenomenon is particularly noticeable at the boundary of the image.
(2) In vehicle-mounted surround-view systems, cameras are usually mounted facing four different directions around the vehicle, while for an ordinary binocular camera system, the base-line length is only tens of centimeters at most. Therefore, poses of cameras in a surround-view system differ greatly and the common-view area between adjacent cameras is smaller.

2.2 Direct Method

Our proposed cameras' extrinsics optimization approach for the surround-view system follows a sparse direct framework. Therefore, here we make a brief review of the studies of direct methods.

The direct methods, which are evolved from optical flow [29] approaches, are modern techniques for camera pose estimation. In direct methods, the local intensity gradient is utilized to determine the step of the optimization [10]. All image pixels can be utilized by direct methods and hence they usually demonstrate better robustness in scenes with sparse textures.

Nowadays, more and more researchers are willing to adopt direct methods rather than feature-point based ones to recover camera poses from images, especially in the field of SLAM. The direct method was first proposed by Irani and Anandan [19]. They explained the brightness constancy constraint and properties of direct methods in detail. In [33], Newcombe et al. built the **DTAM (Dense Tracking And Mapping)** system, in which the direct method was applied to generate the dense map. Engel et al. proposed **LSD-SLAM (Large-Scale Direct monocular SLAM)** in 2014 [9] while another influential SLAM system called **SVO (Semi-direct monocular Visual Odometry)** was proposed in the same year by Forster et al. [10]. LSD-SLAM was one of the most advanced monocular SLAM systems at that time and SVO exhibits distinguished processing speed. Both LSD and SVO are based on the semi-dense direct framework. In 2017, Engel et al. [8] proposed **DSO (Direct Sparse Odometry)**, which is one of the most advanced SLAM systems nowadays. The sparse direct method makes DSO much more efficient than its counterparts. Engel et al. claimed that DSO is five times faster than ORB-SLAM [31, 36], a representative feature-point based SLAM system.



(a) (b)

Fig. 1. The surround-view images before (a) and after (b) performing our camera pose correction algorithm.

2.3 Our Motivations and Contributions

Through literature review, we find that the existing online extrinsics correction methods for the surround-view systems have the following limitations.

(1) To the best of our knowledge, there is currently no effective online extrinsics correction scheme specifically designed for the vehicle-mounted surround-view system. On one hand, most bundle adjustment based online extrinsics correction solutions are designed for common multi-camera systems and they are not suitable for the vehicle-mounted surround-view systems. On the other hand, existing online extrinsics correction approaches specially designed for the surround-view camera system mostly belong to odometry based or lane-line based ones. Since odometry based methods are usually more or less cumbersome and the lane-line based methods have strict requirements on the environmental conditions, they often have obvious deficiencies in usability.
(2) The majority of existing methods in this area are based on matching interest-points or line features extracted from common-view areas of adjacent cameras. The interest-point based ones require a large number of feature points while the line based ones usually require two parallel lanes, implying that these methods have special requirements for the environment. Besides, features of common-view areas are highly probable near boundaries of fisheye images and thus they are prone to mismatch due to large distortion.

In this paper, we aim to fill the aforementioned research gap and have proposed an online cameras' extrinsics correction scheme for the surround-view system. Using our scheme, if camera poses of a calibrated surround-view system change moderately, the associated extrinsics can be corrected online. Figure 1 shows the results of the surround-view images before and after camera poses' correction with our algorithm. The characteristics of the proposed scheme are summarized as follows.

(1) The optimization objective of the proposed scheme is to minimize the photometric errors of the common-view areas between adjacent bird's-eye-views. In order to make it work, the user only needs to park the vehicle in a normal flat field with relatively rich textures. Except for this requirement, it does not require any other additional physical tools or special calibration sites. Hence, it can be seen that the proposed scheme has the advantage of being easy to use and having fewer requirements on the conditions of the operating site. Therefore, it is suitable for ordinary non-professional end-users.
(2) Our scheme follows a sparse direct framework, implying that it does not depend on visual feature points and thus requires less on its working conditions. Within the sparse direct

framework, a novel pixel selection strategy is proposed, with which noise and mismatched objects between images captured by adjacent cameras can be eliminated effectively. Photometric errors are then only computed on the selected positions. Such a pixel selection strategy can effectively improve the whole pipeline's speed and robustness.

(3) The proposed scheme actually is of a cascade structure, comprising two different models, the "ground model" and the "ground-camera model". The ground model is simpler and more efficient than the ground-camera model, but it suffers from the loss of degree-of-freedom, while the ground-camera model does not have such a shortcoming. In actual use, our scheme first tries to use the ground model. If it does not work satisfactorily, the scheme switches to the ground-camera model, which is theoretically more sophisticated and effective.

A preliminary version of this manuscript has been accepted by ACM MM 2019 [26]. The following improvements are made in this version. (1) The whole pipeline is formulated in a sparse direct framework with a new pixel selection strategy introduced. (2) The ground model and the ground-camera model are organized in a cascade structure; in addition, more details and illustrations for the derivation of the two models are provided. (3) More experimental results and discussions, including the robustness analysis, the additional failure case analysis and the comparison with the other existing counterparts, are provided. (4) A more thorough survey of related studies is given.

3 OVERVIEW OF THE SURROUND-VIEW SYSTEM

This section describes the pipeline about how to generate a surround-view from images captured by the cameras mounted around the vehicle.

Given the ground coordinate system O_G and a surround-view system consisting of four cameras C_1, C_2, C_3 and C_4, the poses of cameras in O_G are denoted by $\boldsymbol{T}_{C_1G}, \boldsymbol{T}_{C_2G}, \boldsymbol{T}_{C_3G}$ and $\boldsymbol{T}_{C_4G}$, respectively. For a point $\boldsymbol{P}_G = [X_G, Y_G, Z_G, 1]^T$ in O_G, its corresponding pixel coordinate $\boldsymbol{p}_{C_i}$ in the camera coordinate system of C_i is given by,

$$\boldsymbol{p}_{C_i} = \frac{1}{Z_{C_i}} \boldsymbol{K}_{C_i} \boldsymbol{T}_{C_iG} \boldsymbol{P}_G, i = 1, 2, 3, 4 \tag{1}$$

where Z_{C_i} is the depth of $\boldsymbol{P}_G$ in camera C_i's coordinate system, and $\boldsymbol{K}_{C_i}$ is the 3×3 intrinsic matrix of camera C_i, which can be estimated by Zhang's salient work [43] and some subsequent work of others [7, 45]. Poses of the four cameras with respect to O_G can be calibrated offline by Shao et al.'s method [39]. It's worth mentioning that $\boldsymbol{p}_{C_i}$ is an undistorted point.

The bird's-eye-view image can be generated by projecting a camera image to the ground, namely the plane $Z_G = 0$ in O_G. Consider a point $\boldsymbol{p}_G = [u_G, v_G, 1]^T$ in the bird's-eye-view image, where u_G and v_G are the coordinate values of $\boldsymbol{p}_G$ in the bird's-eye-view coordinate system, respectively. Its corresponding point on the ground plane is $\boldsymbol{P}_G = [X_G, Y_G, Z_G = 0, 1]^T$ with respect to the ground coordinate system, where X_G, Y_G and Z_G are the coordinate values of $\boldsymbol{P}_G$. The relationship between $\boldsymbol{p}_G$ and $\boldsymbol{P}_G$ can be represented as,

$$\begin{bmatrix} u_G \\ v_G \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{d_{X_G}} & 0 & \frac{W}{2d_{X_G}} \\ 0 & -\frac{1}{d_{Y_G}} & \frac{H}{2d_{Y_G}} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_G \\ Y_G \\ 1 \end{bmatrix} \tag{2}$$

where d_{X_G} and d_{Y_G} are the size of each pixel[2], and W and H are the width and height of the scope covered by the surround-view image. It is worth mentioning that because $Z_G = 0$, Z_G is ignored implicitly here. Denote the transformation matrix from $\boldsymbol{P}_G$ to $\boldsymbol{p}_G$ by $\boldsymbol{K}_G$, and Equation (2) can be

[2]More accurately, each pixel in the surround-view image corresponds to a $d_{X_G} \times d_{Y_G}$ physical area on the ground plane.

accordingly simplified as,

$$\boldsymbol{p}_G = \boldsymbol{K}_G \boldsymbol{P}_G. \tag{3}$$

By combining Equation (1) and Equation (3), we can get,

$$\boldsymbol{p}_{C_i} = \frac{1}{Z_{C_i}} \boldsymbol{K}_{C_i} \boldsymbol{T}_{C_i G} \boldsymbol{K}_G^{-1} \boldsymbol{p}_G. \tag{4}$$

Equation (4) actually depicts the relationship of a point $\boldsymbol{p}_{C_i}$ on the image plane of camera C_i and its projection $\boldsymbol{p}_G$ on the surround-view. Using Equation (4), we can project the undistorted image of camera C_i onto the ground to generate a bird's-eye-view image by,

$$\boldsymbol{I}_{GC_i}(\boldsymbol{p}_G) = \boldsymbol{I}_{C_i}(\boldsymbol{p}_{C_i}) \tag{5}$$

where $\boldsymbol{I}_{C_i}$ is the undistorted image captured by camera C_i, and $\boldsymbol{I}_{GC_i}$ is the ground projection of $\boldsymbol{I}_{C_i}$, namely the bird's-eye-view image. By projecting the undistorted images of the four cameras onto the ground and choosing appropriate stitching seams, the surround-view image can be generated.

4 ONLINE CAMERA POSE OPTIMIZATION: A CASCADE STRUCTURE

With accurate camera poses, seamless surround-view images can be synthesized at run-time. However, in online environment, camera poses sometimes may change due to some reasons like bumps and collisions, which will inevitably lead to observable misalignment in adjacent bird's-eye-views of the surround-view image.

To correct the inaccurate extrinsics without re-calibration, this paper proposes an online optimization scheme. Such a scheme takes initial camera poses inherited from offline calibration as input and outputs their optimal estimations that can make the current surround-view seamless. The proposed scheme is of a cascade structure, consisting of two levels of optimization. Each level of optimization is based on a model designed by us. The first level is based on the ground model and the second level is based on the ground-camera model. Both models estimate optimal camera poses by minimizing the photometric errors between adjacent cameras. The overall structure of the proposed online camera poses optimization pipeline is shown in Figure 2.

4.1 Ground Model

Suppose that C_i and C_j are two adjacent cameras in a surround-view system. Denote by $\boldsymbol{I}_{GC_i}$ and $\boldsymbol{I}_{GC_j}$ the bird's-eye-view images generated from C_i and C_j, respectively. Given a point $\boldsymbol{p}_G$ on the common-view area of $\boldsymbol{I}_{GC_i}$ and $\boldsymbol{I}_{GC_j}$, the photometric error $\varepsilon_{\boldsymbol{p}_G}$ between $\boldsymbol{I}_{GC_i}(\boldsymbol{p}_G)$ and $\boldsymbol{I}_{GC_j}(\boldsymbol{p}_G)$ can be defined as,

$$\varepsilon_{\boldsymbol{p}_G} = \|\boldsymbol{I}_{GC_i}(\boldsymbol{p}_G) - \boldsymbol{I}_{GC_j}(\boldsymbol{p}_G)\|_2. \tag{6}$$

By expanding $\boldsymbol{p}_G$ into a form that includes the camera's pose, $\boldsymbol{p}_G$ can be written as,

$$\boldsymbol{p}_G = \boldsymbol{K}_G \exp(\boldsymbol{\xi}_{GC_i}^{\wedge}) \boldsymbol{P}_{C_i} \tag{7}$$

where $\boldsymbol{P}_{C_i}$ represents the spatial point on the ground corresponding to $\boldsymbol{p}_G$ in the camera coordinate system of C_i and $\boldsymbol{\xi}_{GC_i}$ is the Lie algebra representation [16] of the transformation from camera C_i to the ground. $\boldsymbol{\xi}_{GC_i}^{\wedge}$ is just the anti-symmetric matrix induced by $\boldsymbol{\xi}_{GC_i}$. By substituting $\boldsymbol{p}_G$ in Equation (6) with Equation (7), we can get,

$$\varepsilon_{\boldsymbol{p}_G} = \|\boldsymbol{I}_{GC_i}(\boldsymbol{K}_G \exp(\boldsymbol{\xi}_{GC_i}^{\wedge}) \boldsymbol{P}_{C_i}) - \boldsymbol{I}_{GC_j}(\boldsymbol{K}_G \exp(\boldsymbol{\xi}_{GC_j}^{\wedge}) \boldsymbol{P}_{C_j})\|_2. \tag{8}$$

Then the optimization objective of the ground model can be defined as,

$$\boldsymbol{\xi}_{GC_i}^*, \boldsymbol{\xi}_{GC_j}^* = \underset{\boldsymbol{\xi}_{GC_i}, \boldsymbol{\xi}_{GC_j}}{\arg\min} \sum_{\boldsymbol{p}_G \in \mathcal{N}_{ij}} \varepsilon_{\boldsymbol{p}_G}^2 \tag{9}$$

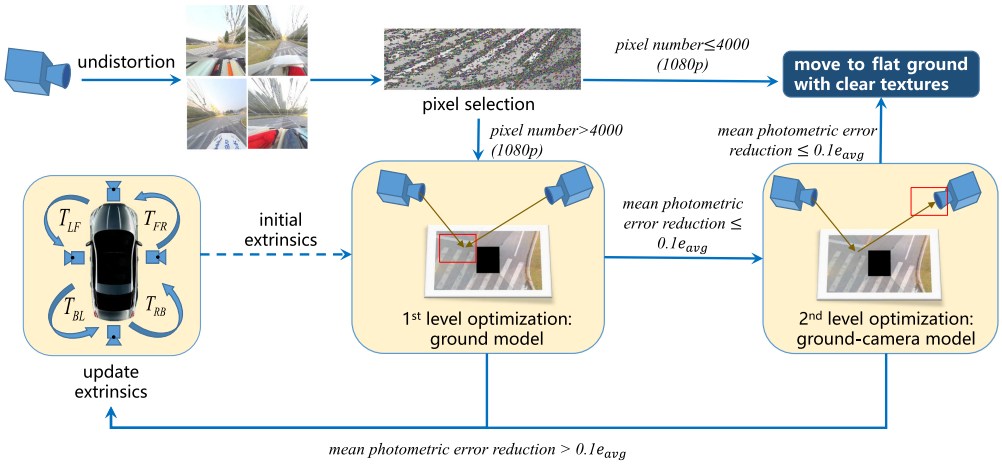


Fig. 2. Structural sketch of our algorithm for online cameras' poses correction of a surround-view system. There are two levels of optimization. The first level of optimization is based on the ground model while the second level is based on the ground-camera model. e_{avg} in the figure stands for the average photometric error over the common-view regions of the used bird's-eye-view images.

where $\mathcal{N}_{ij}$ is the set of qualified points in the overlapping region of $\boldsymbol{I}_{GC_i}$ and $\boldsymbol{I}_{GC_j}$ chosen by a pixel selection strategy, whose details will be discussed in Section 4.5.

To optimize the objective function Equation (9), the derivative relationship between $\varepsilon^2_{\boldsymbol{p}_G}$ and $\boldsymbol{\xi}_{GC_i}$ needs to be determined. The Jacobian of $\varepsilon^2_{\boldsymbol{p}_G}$ to $\boldsymbol{\xi}_{GC_i}$ can be expressed as,

$$\boldsymbol{J}_i = \frac{\partial \varepsilon^2_{\boldsymbol{p}_G}}{\partial \boldsymbol{\xi}^T_{GC_i}}. \tag{10}$$

Equation (10) can be decomposed to four parts with the chain rule,

$$\boldsymbol{J}_i = \frac{\partial \varepsilon^2_{\boldsymbol{p}_G}}{\partial \boldsymbol{I}_{GC_i}(\boldsymbol{p}_G)} \cdot \frac{\partial \boldsymbol{I}_{GC_i}(\boldsymbol{p}_G)}{\partial \boldsymbol{p}^T_G} \cdot \frac{\partial \boldsymbol{p}_G}{\partial \boldsymbol{P}^T_G} \cdot \frac{\partial \boldsymbol{P}_G}{\partial \boldsymbol{\xi}^T_{GC_i}}. \tag{11}$$

Next, we will discuss these four parts one by one:

(1) $\partial \varepsilon^2_{\boldsymbol{p}_G} / \partial \boldsymbol{I}_{GC_i}(\boldsymbol{p}_G)$ is the derivative of squared error $\varepsilon^2_{\boldsymbol{p}_G}$ to the pixel intensity $\boldsymbol{I}_{GC_i}(\boldsymbol{p}_G)$. We denote it by δ. Supposing that $\boldsymbol{I}_{GC_i}$ is a grayscale image, then δ is,

$$\delta = \frac{\partial \varepsilon^2_{\boldsymbol{p}_G}}{\partial \boldsymbol{I}_{GC_i}(\boldsymbol{p}_G)} = 2\left(\boldsymbol{I}_{GC_i}(\boldsymbol{p}_G) - \boldsymbol{I}_{GC_j}(\boldsymbol{p}_G)\right). \tag{12}$$

(2) $\partial \boldsymbol{I}_{GC_i}(\boldsymbol{p}_G) / \partial \boldsymbol{p}^T_G$ is the intensity gradient of $\boldsymbol{I}_{GC_i}$ at the pixel $\boldsymbol{p}_G$,

$$\frac{\partial \boldsymbol{I}_{GC_i}(\boldsymbol{p}_G)}{\partial \boldsymbol{p}^T_G} \triangleq \begin{bmatrix} \nabla \boldsymbol{I}^{u_G}_{GC_i} & \nabla \boldsymbol{I}^{v_G}_{GC_i} \end{bmatrix}. \tag{13}$$

(3) $\partial \boldsymbol{p}_G / \partial \boldsymbol{P}^T_G$ is the derivative of $\boldsymbol{p}_G$ to its corresponding spatial point $\boldsymbol{P}_G$. From Equation (2), we can have,

$$\frac{\partial \boldsymbol{p}_G}{\partial \boldsymbol{P}^T_G} = \begin{bmatrix} \frac{\partial u_G}{\partial X_G} & \frac{\partial u_G}{\partial Y_G} & \frac{\partial u_G}{\partial Z_G} \\ \frac{\partial v_G}{\partial X_G} & \frac{\partial v_G}{\partial Y_G} & \frac{\partial v_G}{\partial Z_G} \end{bmatrix} = \begin{bmatrix} \frac{1}{d_{X_G}} & 0 & 0 \\ 0 & -\frac{1}{d_{Y_G}} & 0 \end{bmatrix}. \tag{14}$$

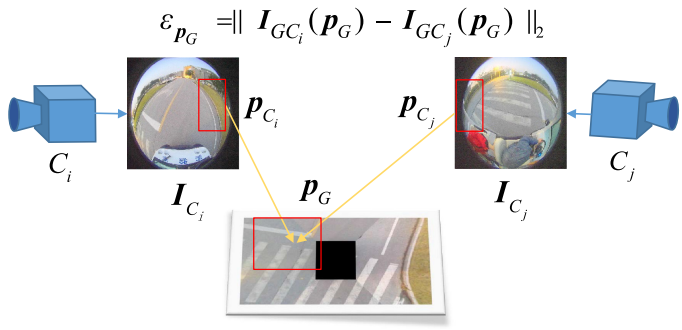


Fig. 3. Illustration of the ground model. The error is constructed on the surround-view image plane.

(4) $\partial P_G / \partial \xi_{GC_i}^T$ is the derivative of the 3D point P_G to the camera pose ξ_{GC_i},

$$\frac{\partial P_G}{\partial \xi_{GC_i}^T} = \begin{bmatrix} I_{3\times3} & -P_G^\wedge \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & Z_G & -Y_G \\ 0 & 1 & 0 & -Z_G & 0 & X_G \\ 0 & 0 & 1 & Y_G & -X_G & 0 \end{bmatrix} \tag{15}$$

where $I_{3\times3}$ is a 3×3 identity matrix and $P_G^\wedge$ is the 3×3 anti-symmetric matrix generated from P_G. It needs to be noted that P_G is a point on the ground plane and accordingly $Z_G = 0$.

By combining the above four parts together, the final form of J_i can be expressed as,

$$J_i = \delta \begin{bmatrix} \frac{\nabla I_{GC_i}^{u_G}}{d_{X_G}} & \frac{\nabla I_{GC_i}^{v_G}}{d_{Y_G}} & 0 & 0 & 0 & -\frac{\nabla I_{GC_i}^{u_G} Y_G}{d_{X_G}} - \frac{\nabla I_{GC_i}^{v_G} X_G}{d_{Y_G}} \end{bmatrix}. \tag{16}$$

Following a similar derivation, we can get the Jacobian of $\varepsilon_{p_G}^2$ to ξ_{GC_j} and denote it by J_j. Once J_i and J_j are available, Equation (9) can be iteratively optimized with any non-linear optimization methods, such as the gradient descent method, the Gauss-Newton method and the Levenberg-Marquardt method [1, 6, 27, 30, 35, 40].

For a surround-view system, we can jointly optimize all camera poses by minimizing the overall photometric error of the whole system. Thus, the optimization objective of the surround-view system can be expressed as,

$$\xi_{GC_i}^* = \arg\min_{\xi_{GC_i}} \sum_{i=1}^{4} \sum_{j \in \Omega(i)} \sum_{p_G \in \mathcal{N}_{ij}} \varepsilon_{p_G}^2 \tag{17}$$

where $\Omega(i)$ contains all the indexes of C_i's adjacent cameras.The sketch of the ground model is shown in Figure 3.

Although the ground model can solve the problem of camera pose optimization to some extent, it has an obvious shortcoming. The camera pose ξ_{GC_i} has six degrees-of-freedom, but the Jacobian matrix J_i derived by the ground model has only three degree-of-freedoms, implying that only three dimensions of ξ_{GC_i} can be updated. The first two dimensions represent translations parallel to the ground plane, while the last one represents rotation around the Z-axis of the ground coordinate system. That is to say, the ground model can only correct particular types of camera poses' change, which limits its application scope. To solve the issue of degree-of-freedom loss, we propose a more universal optimization model, namely the "ground-camera" model.

4.2 Ground-Camera Model

Unlike the ground model, the ground-camera model resorts to a different projection plane to compute the photometric error. We also suppose that C_i and C_j are two adjacent cameras. To correct

the pose of camera C_i, we firstly project $\boldsymbol{I}_{GC_j}$ to camera C_i using Equation (4). Suppose that the projection of the bird's-eye-view image $\boldsymbol{I}_{GC_j}$ on camera C_i is $\boldsymbol{I}_{GC_j}^{C_i}$. In the ground-camera model, the photometric error $\varepsilon_{\boldsymbol{p}}$ at $\boldsymbol{p}$ of camera C_i is defined as,

$$\varepsilon_{\boldsymbol{p}} = \|\boldsymbol{I}_{C_i}(\boldsymbol{p}) - \boldsymbol{I}_{GC_j}^{C_i}(\boldsymbol{p})\|_2 \tag{18}$$

where $\boldsymbol{p}$ is a point on the imaging plane of the camera C_i and should also be a qualified point on $\boldsymbol{I}_{GC_j}^{C_i}$. Similar to the ground model, $\boldsymbol{p}$ is firstly expanded into a form which includes the camera pose,

$$\boldsymbol{p} = \frac{1}{Z_{C_i}} \boldsymbol{K}_{C_i} \exp(\boldsymbol{\xi}_{C_iG}^{\wedge}) \boldsymbol{P}_G. \tag{19}$$

Define $\boldsymbol{P}_{C_i} \triangleq \exp(\boldsymbol{\xi}_{C_iG}^{\wedge})\boldsymbol{P}_G = [X_{C_i} \quad Y_{C_i} \quad Z_{C_i}]^T$. Then the photometric error at $\boldsymbol{p}$ can be written as,

$$\varepsilon_{\boldsymbol{p}} = \left\| \boldsymbol{I}_{C_i}\left(\frac{1}{Z_{C_i}} \boldsymbol{K}_{C_i} \exp(\boldsymbol{\xi}_{C_iG}^{\wedge}) \boldsymbol{P}_G\right) - \boldsymbol{I}_{GC_j}^{C_i}\left(\frac{1}{Z_{C_i}} \boldsymbol{K}_{C_i} \exp(\boldsymbol{\xi}_{C_iG}^{\wedge}) \boldsymbol{P}_G\right) \right\|_2 \tag{20}$$

and for camera C_i, its optimal camera pose is given by,

$$\boldsymbol{\xi}_{C_iG}^{*} = \arg\min_{\boldsymbol{\xi}_{C_iG}} \sum_{i=1}^{4} \sum_{j \in \Omega(i)} \sum_{\boldsymbol{p} \in \mathcal{N}_{ij}^{C_i}} \varepsilon_{\boldsymbol{p}}^2 \tag{21}$$

where $\Omega(i)$ contains all the indexes of C_i's adjacent cameras and $\mathcal{N}_{ij}^{C_i}$ is a set containing the corresponding projection points on the imaging plane of C_i of all points in $\mathcal{N}_{ij}$.

To optimize the objective function Equation (21), the derivative relationship between $\varepsilon_{\boldsymbol{p}}^2$ and $\boldsymbol{\xi}_{C_iG}$ needs to be determined. The Jacobian of $\varepsilon_{\boldsymbol{p}}^2$ to $\boldsymbol{\xi}_{C_iG}$ can be decomposed to,

$$\boldsymbol{J} = \frac{\partial \varepsilon_{\boldsymbol{p}}^2}{\partial \boldsymbol{\xi}_{C_iG}} = \frac{\partial \varepsilon_{\boldsymbol{p}}^2}{\partial \boldsymbol{I}_{C_i}} \frac{\partial \boldsymbol{I}_{C_i}}{\partial \boldsymbol{p}^T} \frac{\partial \boldsymbol{p}}{\partial \boldsymbol{P}_{C_i}^T} \frac{\partial \boldsymbol{P}_{C_i}}{\partial \boldsymbol{\xi}_{C_iG}^T} + \frac{\partial \varepsilon_{\boldsymbol{p}}^2}{\partial \boldsymbol{I}_{GC_j}^{C_i}} \frac{\partial \boldsymbol{I}_{GC_j}^{C_i}}{\partial \boldsymbol{p}^T} \frac{\partial \boldsymbol{p}}{\partial \boldsymbol{P}_{C_i}^T} \frac{\partial \boldsymbol{P}_{C_i}}{\partial \boldsymbol{\xi}_{C_iG}^T}. \tag{22}$$

Obviously, this formula contains two terms and each of them can be decomposed to four simpler parts using the chain rule. The four simpler parts are discussed below one by one:

(1) $\partial \varepsilon_{\boldsymbol{p}}^2 / \partial \boldsymbol{I}_{C_i}$ and $\partial \varepsilon_{\boldsymbol{p}}^2 / \partial \boldsymbol{I}_{GC_j}^{C_i}$ are the derivatives of the squared photometric error $\varepsilon_{\boldsymbol{p}}^2$ to the intensities at $\boldsymbol{p}$ of images $\boldsymbol{I}_{C_i}$ and $\boldsymbol{I}_{GC_j}^{C_i}$, respectively. Define $\delta = 2(\boldsymbol{I}_{C_i}(\boldsymbol{p}) - \boldsymbol{I}_{GC_j}^{C_i}(\boldsymbol{p}))$, and accordingly we have,

$$\frac{\partial \varepsilon_{\boldsymbol{p}}^2}{\partial \boldsymbol{I}_{C_i}} = \delta, \frac{\partial \varepsilon_{\boldsymbol{p}}^2}{\partial \boldsymbol{I}_{GC_j}^{C_i}} = -\delta. \tag{23}$$

(2) $\partial \boldsymbol{I}_{C_i} / \partial \boldsymbol{p}^T$ and $\partial \boldsymbol{I}_{GC_j}^{C_i} / \partial \boldsymbol{p}^T$ are the intensity gradients of $\boldsymbol{I}_{C_i}$ and $\boldsymbol{I}_{GC_j}^{C_i}$ at $\boldsymbol{p}$, respectively. Suppose that $\boldsymbol{p} = [u \quad v]^T$, and then these two parts can be expressed as,

$$\frac{\partial \boldsymbol{I}_{C_i}}{\partial \boldsymbol{p}^T} = \nabla \boldsymbol{I}_{C_i}(\boldsymbol{p}) \triangleq [\nabla_i u \ \nabla_i v], \frac{\partial \boldsymbol{I}_{GC_j}^{C_i}}{\partial \boldsymbol{p}^T} = \nabla \boldsymbol{I}_{GC_j}^{C_i}(\boldsymbol{p}) \triangleq \left[\nabla_j u \ \nabla_j v\right]. \tag{24}$$

(3) $\partial \boldsymbol{p} / \partial \boldsymbol{P}_{C_i}^T$ is the derivative of $\boldsymbol{p}$ to its corresponding spatial point $\boldsymbol{P}_{C_i}$. From the pin-hole camera model, we have,

$$\frac{\partial \boldsymbol{p}}{\partial \boldsymbol{P}_{C_i}^T} = \begin{bmatrix} \frac{f_x}{Z_{C_i}} & 0 & -\frac{f_x X_{C_i}}{Z_{C_i}^2} \\ 0 & \frac{f_y}{Z_{C_i}} & -\frac{f_y Y_{C_i}}{Z_{C_i}^2} \end{bmatrix}. \tag{25}$$

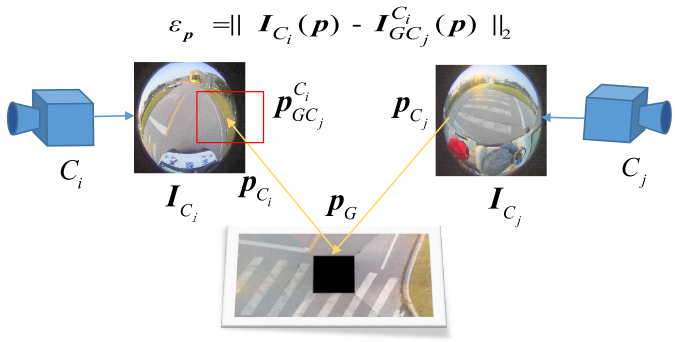


Fig. 4. Illustration of the ground-camera model. The error in the ground-camera model is constructed on the image plane of C_i.

(4) $\partial P_{C_i}/\partial \xi^T_{C_iG}$ is the derivative of the 3D point P_{C_i} to the camera pose ξ_{C_iG},

$$\frac{\partial P_{C_i}}{\partial \xi^T_{C_iG}} = \begin{bmatrix} I_{3\times3} & -P^{\wedge}_{C_i} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & Z_{C_i} & -Y_{C_i} \\ 0 & 1 & 0 & -Z_{C_i} & 0 & X_{C_i} \\ 0 & 0 & 1 & Y_{C_i} & -X_{C_i} & 0 \end{bmatrix}. \tag{26}$$

By merging the decomposed terms in Equations (22)~(26), the Jacobian J of ε_p^2 to ξ_{C_iG} can be expressed as,

$$J = \delta \begin{bmatrix} \nabla_i u - \nabla_j u & \nabla_i v - \nabla_j v \end{bmatrix} \begin{bmatrix} \frac{f_x}{Z_{C_i}} & 0 & -\frac{f_x X_{C_i}}{Z^2_{C_i}} & -\frac{f_x X_{C_i} Y_{C_i}}{Z^2_{C_i}} & f_x + \frac{f_x X^2_{C_i}}{Z^2_{C_i}} & -\frac{f_x Y_{C_i}}{Z_{C_i}} \\ 0 & \frac{f_y}{Z_{C_i}} & -\frac{f_y Y_{C_i}}{Z^2_{C_i}} & -f_y - \frac{f_y Y^2_{C_i}}{Z^2_{C_i}} & \frac{f_y X_{C_i} Y_{C_i}}{Z^2_{C_i}} & \frac{f_y X_{C_i}}{Z_{C_i}} \end{bmatrix}. \tag{27}$$

Once J is available, Equation (21) can be optimized iteratively to find the optimal camera pose $\xi^*_{C_iG}$ with proper optimization methods. The sketch of the ground-camera model is shown in Figure 4.

4.3 Cascade Structure

Obviously, no dimension of the Jacobian J in the ground-camera model is constantly equal to zero. Thus, it does not suffer from the problem of degree-of-freedom loss as the ground model. That is to say, the ground-camera model has a wider application scope than the ground model. However, this does not mean that the ground model is useless. Although the ground model can only deal with limited types of camera pose changes, it has extremely low computational complexity compared to the ground-camera model. For these reasons, our proposed online camera pose correction scheme is naturally designed as a cascade structure comprising two levels of optimizations. Specifically, the first level of optimization is based on the ground model while the second one is based on the ground-camera model. Only when the ground model fails to reduce the photometric errors to a satisfactory value, the ground-camera model is activated. In our implementation, if the decrease of the mean photometric error of each pixel is less than a dynamic threshold by using the ground model, the results are considered as to be unsatisfactory and then the ground-camera model should be activated. The dynamic threshold is set to $0.1e_{avg}$, where e_{avg} is the average photometric error over the common-view regions of the bird's-eye-view images utilized in the correction.

4.4 Exposure Correction

Because of the differences on lighting conditions, environmental reflections, cameras' internal constructions, etc., for a same point p_G on the ground, corresponding pixel values $I_{C_i}(p_{C_i})$ and

$\boldsymbol{I}_{C_j}(\boldsymbol{p}_{C_j})$ won't be completely the same, even if the camera poses are absolutely accurate. Actually, for an image of a physical object, besides the properties of the object itself, it will also be affected by the exposure time, the vignette and the non-linear response function of the camera [8]. Based on our experience, exposure time is the most important factor. We define the exposure factor γ_{ij} as,

$$\gamma_{ij} = \frac{t_i}{t_j} \tag{28}$$

where t_i is C_i's exposure time and t_j is that of C_j's. Even though the exposure time can't be obtained directly in general, the factor γ_{ij} can be fitted as,

$$\gamma_{ij} = \frac{\sum_{\boldsymbol{p}_G \in \boldsymbol{O}_{ij}} \boldsymbol{I}_{GC_i}(\boldsymbol{p}_G)}{\sum_{\boldsymbol{p}_G \in \boldsymbol{O}_{ij}} \boldsymbol{I}_{GC_j}(\boldsymbol{p}_G)} \tag{29}$$

where $\boldsymbol{I}_{GC_i}$ and $\boldsymbol{I}_{GC_j}$ are bird's-eye-view images of camera C_i and C_j, respectively, and $\boldsymbol{O}_{ij}$ is the set of all pixels in the common-view region of C_i and C_j on bird's-eye-view images. Then, the error term $\varepsilon_{\boldsymbol{p}_G}$ of the ground model in Equation (6) can be reformulated as,

$$\varepsilon_{\boldsymbol{p}_G} = \|\boldsymbol{I}_{GC_i}(\boldsymbol{p}_G) - \gamma_{ij}\boldsymbol{I}_{GC_j}(\boldsymbol{p}_G)\|_2 \tag{30}$$

and for the ground-camera model, a similar reformulation is also required. Besides, corresponding derivatives should be adjusted accordingly. With the exposure correction, the negative influence of the intensity discrepancies aroused by different lighting conditions or environmental reflections can be weakened effectively.

4.5 Pixel Selection Strategy

For the consideration of robustness and computational speed, our online camera pose optimization scheme follows a sparse direct framework. Actually, pixels with tiny gradient moduli can't provide "confident" guidance information to the updating step. What's worse, such pixels affect the optimization mainly by noise and thus can even do harm to the final correction accuracy. Hence, it is necessary to figure out and discard such pixels.

Besides, in Section 3 it's mentioned that all points $\boldsymbol{P}_G$s are assumed to be on the ground plane and their Z coordinates are considered as zero. Such an assumption is also the precondition for the establishment of both the ground model and the ground-camera model. However, in the field-of-view of the surround-view system, there are usually some objects whose heights are non-negligible. Such objects will cause obvious parallax between adjacent bird's-eye views, so in short, we call them "mismatched objects". Lawns, curbs, pedestrians, and other vehicles can all be regarded as mismatched objects. In the pixel selection process, pixels from mismatched objects should also be removed.

Take two adjacent cameras C_i and C_j as an example. Primarily, the pixels we select should be in the common-view region of C_i and C_j, which can be represented as $\boldsymbol{O}_{ij}$. A set of pixels $\boldsymbol{N}_{ij}$ will be selected out by the selection strategy and involved in optimization. For every pixel in $\boldsymbol{N}_{ij}$, its corresponding pixel coordinate $\boldsymbol{p}$ must satisfy the following three criteria:

- $\boldsymbol{p}$ must lie in the common-view region $\boldsymbol{O}_{ij}$,

$$\boldsymbol{p} \in \boldsymbol{O}_{ij}. \tag{31}$$

- The color discrepancy between $\boldsymbol{I}_{GC_i}(\boldsymbol{p})$ and $\boldsymbol{I}_{GC_j}(\boldsymbol{p})$ is not allowed to be too large. With this rule the effect of mismatched objects captured by adjacent cameras can be eliminated effectively. Let $\boldsymbol{I}^c_{GC_i}$ and $\boldsymbol{I}^c_{GC_j}$ be the channel map of $\boldsymbol{I}_{GC_i}$ and $\boldsymbol{I}_{GC_j}$ of channel c, respectively.

The color ratio $r_c(\boldsymbol{p})$ is defined as,

$$r_c(\boldsymbol{p}) = \frac{\boldsymbol{I}^c_{GC_i}(\boldsymbol{p})}{\boldsymbol{I}^c_{GC_j}(\boldsymbol{p})}. \tag{32}$$

We use the standard deviation of $\boldsymbol{p}$'s color ratios in different channels as the measurement of its color discrepancy,

$$D_{color}(\boldsymbol{p}) = \sqrt{\frac{\sum_{c=1}^{n_c}\left(r_c(\boldsymbol{p}) - r_\mu(\boldsymbol{p})\right)^2}{n_c}} \tag{33}$$

where n_c is the number of channels (normally 3) and r_μ is the average of all $\boldsymbol{p}$'s color ratios. For any $p \in \boldsymbol{\mathcal{N}}_{ij}$, it must satisfy,

$$D_{color}(\boldsymbol{p}) < D_{mean} - 2\sigma_d \tag{34}$$

where D_{mean} is the average color discrepancy of all the points in $\boldsymbol{O}_{ij}$ and σ_d is the associated standard deviation.

- $\boldsymbol{p}$'s intensity gradient modulus $G_i(\boldsymbol{p})$ should be large enough,

$$G_i(\boldsymbol{p}) > G_{mean} + 2\sigma_g \tag{35}$$

where G_{mean} is the mean intensity gradient modulus over $\boldsymbol{O}_{ij}$ and σ_g is the associated standard deviation.

It is worth mentioning that images lacking textures should not be used to optimize camera poses. In our implementation, if the total number of qualified points in a frame is fewer than 4,000 (for 1080p images), then the frame cannot be used for optimization and should be substituted by another frame with richer textures.

5 EXPERIMENTAL RESULTS

5.1 Experiment Setup

To validate the performance of our online camera pose optimization approach, experiments were performed on an electric car equipped with a surround-view system. We tested our approach on flat fields with six typical kinds of textures and collected the associated surround-view data (fish-eye images and the associated offline calibration parameters). These data were divided into six groups (Groups A, B, C, D, E, and F) according to the sites where they were collected, samples of which are shown in Figure 5. For each group, there were 100 surround-views and altogether 600 surround-views were collected. It should be noted that for all groups, cameras' poses were changed moderately from the state of initial offline calibration. Hence, if un-updated extrinsics were used, the synthesized surround-views would exhibit observable geometric misalignment between adjacent bird's-eye-views as shown in Figure 5 (the left one of each image pair).

The resolution, the field-of-view, and the acquisition frequency of the cameras are 1920×1080 (1080p), 190 degrees and 30 FPS, respectively. Images of other common resolutions can also be obtained by resizing the captured images. Our approach was implemented with standard C++ and all the experiments were conducted on a laptop with an Intel(R) Core(TM) i5-7300HQ CPU.

More results in the form of images or videos are available online at https://cslinzhang.github.io/CamPoseCorrection/.

5.2 Typical Samples

In order to qualitatively demonstrate the superiority of the proposed online camera pose optimization scheme in terms of generalization, we selected a typical sample from each of the six

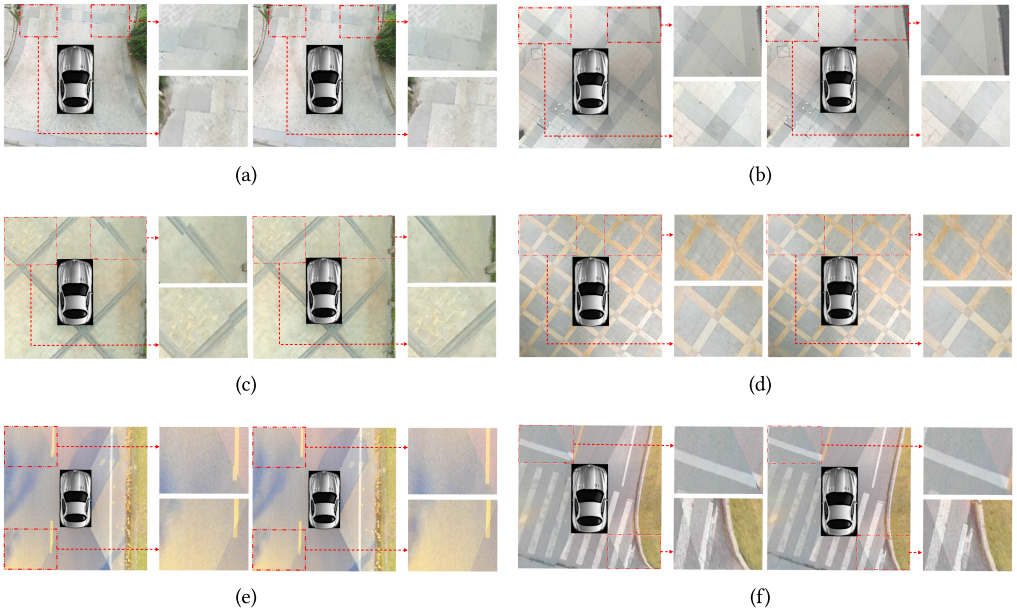


Fig. 5. Comparisons of surround-views before and after cameras' poses correction in various environments. From (a) to (f), the six image pairs are typical ones selected from groups A∼F mentioned in Section 5.1, respectively. For each pair, the left image is generated with disturbed extrinsics and the right one is synthesized with corrected cameras' poses obtained by our approach. It can be observed that for all the examined cases, after applying our camera pose optimization approach, the geometric misalignments between adjacent bird's-eye-views are all greatly reduced, corroborating the superior efficacy of the proposed approach.

groups of data for processing, and the results are shown in Figure 5. Figure 5(a)∼5(f) correspond to samples from groups A∼F, respectively. For each sample, the original surround-view and the one generated using the optimized camera poses are displayed side by side. It can be clearly seen from the comparison that for each case the geometric misalignment in the surround-view generated by the optimized camera poses is significantly ameliorated. It implies that our scheme has loose requirements for external working environments, and thus has a good usability and a strong generalization ability.

5.3 Quantitative Evaluation

Minimizing photometric errors. As mentioned in Section 5.1, six groups of data were collected and for all groups, camera poses were changed moderately from the state of initial offline calibration. In this experiment, we optimized the camera poses of each group online and used the resulting camera poses to regenerate the surround-views. The photometric errors over surround-views of each group were used to measure the effectiveness of our camera pose optimization approach. As the accuracy of offline calibration methods is generally satisfactory now, we take the photometric errors over surround-views synthesized with offline calibrated camera poses as the baseline (in short it can be called as "offline baseline"), and use the relative values of photometric errors to measure the effectiveness of the correction. It's worth mentioning that, without special declaration, to validate the effectiveness of our scheme on most of our collected data, the proposed method was tested on every frame, while in practice, only one shot is enough for the correction.

Specifically, we denote the photometric error over a surround-view synthesized with the camera poses to be measured as P_1, the photometric error over the same surround-view synthesized with

the offline calibrated poses as P_2. Then the relative photometric error of the poses to be measured is actually $P_r = P_1 - P_2$. Thus, if the relative photometric errors are lower than 0, the photometric errors of the camera poses to be measured are lower than the offline baseline and we can say that the camera poses in the surround-view system are accurate (more accurate than the offline calibrated camera poses). Results in terms of relative photometric errors of each group along with the optimization evolvement are illustrated in Figure 6. The results corroborate that using our proposed approach, camera poses of a surround-view system could be effectively corrected online under various environmental conditions.

Pixel selection effect. Actually, without pixel selection the optimization in our method becomes a dense direct approach rather than a sparse one. In short, we call the optimization approach of our method with and without pixel selection as the "sparse approach" and the "dense approach", respectively. Two factors are considered in the evaluation of the pixel selection strategy, the speed and the accuracy. On one hand, as shown in Table 1, under the same experimental conditions, the optimizations in the "sparse approaches" are all obviously faster than those in the "dense approaches" regardless of the resolution. On the other hand, the accuracy of the method is evaluated by the relative photometric errors calculated over all surround-views. The relative photometric errors along with the optimization evolvement of both the "sparse approach" and the "dense approach" are illustrated in Figure 7. It can be seen that for the "dense approach", after 30 iterations of the optimization, the final relative photometric error is 226.00, while for the "sparse approach" the value is −289.73. To sum up, the experimental results corroborate that both the speed and the accuracy can be enhanced effectively by our proposed pixel selection strategy.

Ablation study of the cascade structure. As aforementioned, our algorithm is of a cascade structure. The first level of the optimization is based on the ground model and the second level is based on the ground-camera model. In this part, we evaluate the performance of these two models with respect to the speed and the accuracy.

On one hand, we recorded the single-iteration time costs of the ground model and the ground-camera model under various experimental conditions in Table 1. From Table 1, it can be found that under the same experimental conditions, the speed of the ground model is always much faster than that of the ground-camera model.

On the other hand, the ground-camera model performs much better than the ground model in accuracy, since it doesn't suffer from the problem of degree-of-freedom loss. To quantitatively verify this claim, related experimental results are offered in this part. We define the proportion of the number of iterations of the first-level optimization to the total number of iterations as "p". For example, in the case that the optimization approach lasts for 30 iterations totally, the first level of optimization will last for $30p$ iterations and other $30\,(1-p)$ iterations are all conducted using the second-level model (the ground-camera model). It's worth mentioning that when p is set to 0, the optimization is thoroughly based on the ground-camera model, and when p is 1, the ground model will be the only model used. Figure 8 shows the relationship between the relative photometric errors and p's settings. From the experimental results, it can be seen that the ground-camera model performs much better than the ground model in accuracy. Besides, from Figure 8 it can also be found that as long as p is lower than 0.5, the accuracy of the cascade structure is always outstanding. And since the ground model is much faster than the ground-camera model, with a combination of both models in cascade, a good balance between the speed and the accuracy can be achieved.

Robustness to initial poses. To evaluate the robustness of our scheme to camera poses' changes, we firstly quantify the changes. We define the "basis disturbance" on camera poses as a 6-dimension

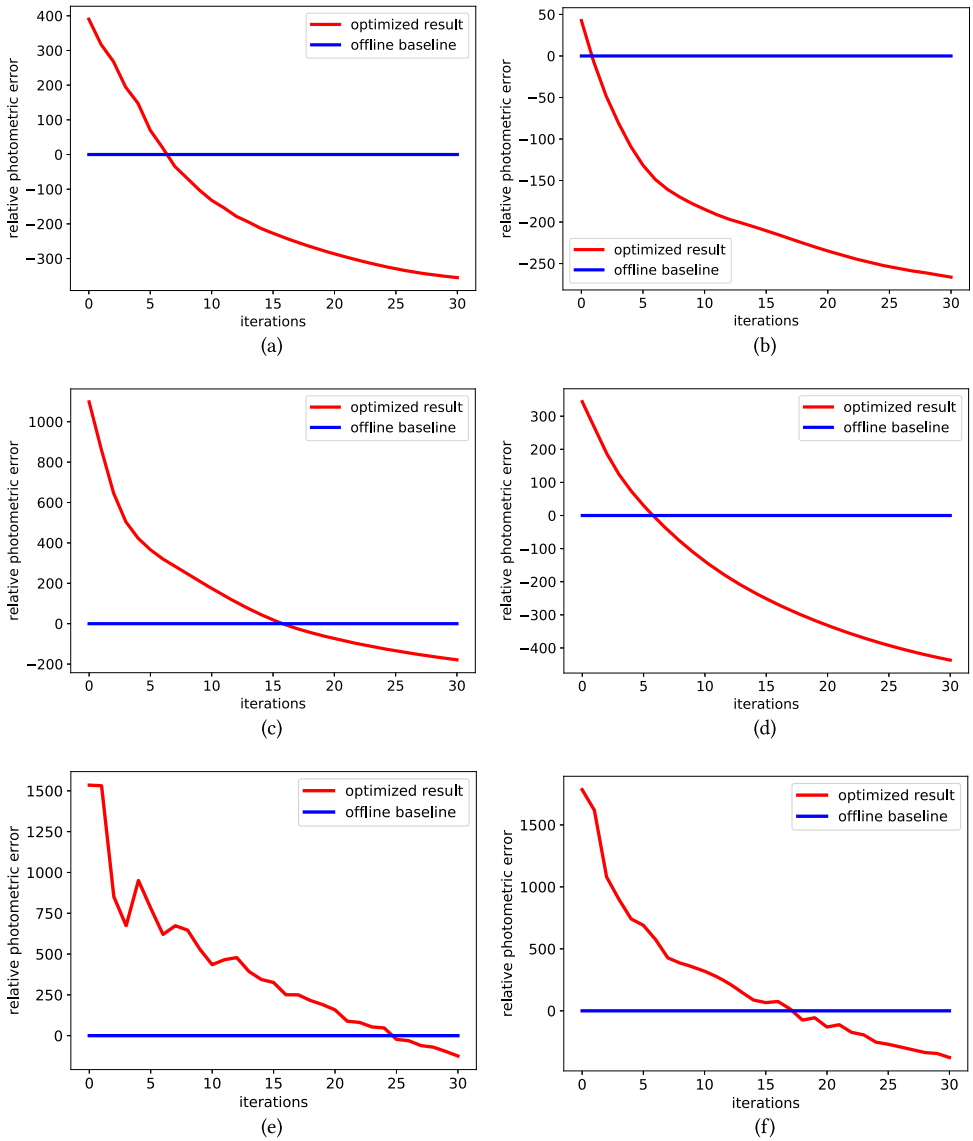


Fig. 6. (a)~(f) are relative photometric errors over surround-views of groups A~F, respectively, along with the optimization evolvement of our approach. The red curve plots relative photometric errors while the blue line shows the offline baseline. Since the photometric errors shown are relative values, the offline baseline is always equal to zero.

vector $\boldsymbol{V}_6$, which is given as,

$$\boldsymbol{V}_6 = [0.01, -0.01, 0.01, -0.01, 0.01, -0.01]^T. \tag{36}$$

Then the disturbed pose $\boldsymbol{\xi}^d_{C_iG}$ of camera C_i can be expressed as,

$$\boldsymbol{\xi}^d_{C_iG} = \boldsymbol{\xi}_{C_iG} + \alpha \boldsymbol{V}_6 \tag{37}$$

Table 1. Time Cost Analysis of the Proposed Two Models

Model	Sparsity	Resolution	Time cost	Pixel number
Ground	Dense	1080p	0.1071s/iter	20000
	Sparse	1080p	0.0572s/iter	8103.03
	Dense	900p	0.1008s/iter	13889
	Sparse	900p	0.0467s/iter	4907.67
	Dense	720p	0.0936s/iter	8889
	Sparse	720p	0.0301s/iter	3329.63
	Dense	640p	0.0758s/iter	5000
	Sparse	640p	0.0220s/iter	1992.20
Ground-Camera	Dense	1080p	0.8074s/iter	20000
	Sparse	1080p	0.5004s/iter	8103.03
	Dense	900p	0.5339s/iter	13889
	Sparse	900p	0.3609s/iter	5391
	Dense	720p	0.3417s/iter	8889
	Sparse	720p	0.2343s/iter	4907.67
	Dense	640p	0.2026s/iter	5000
	Sparse	640p	0.1363s/iter	3329.63

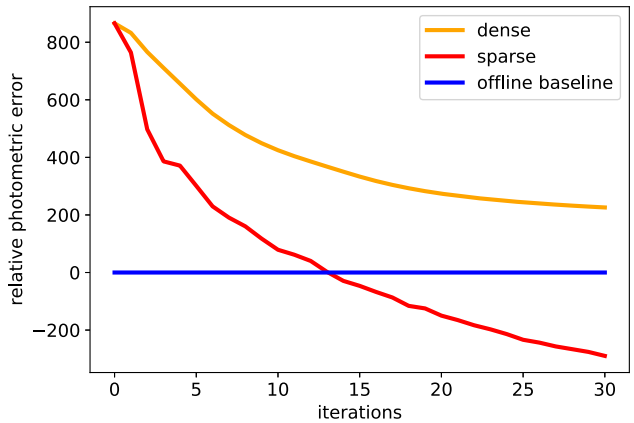


Fig. 7. The relative photometric errors along with the optimization evolvement of approaches based on the sparse framework and the dense framework.

where α is the disturbance factor. Suffering from a "basis disturbance" is actually equivalent to translating the camera for about a centimeter in three orthogonal directions and then rotating the camera for about 1°.

We corrected poses of the surround-view system and recorded the corresponding relative photometric errors under different α's settings. The experimental results are illustrated in Figure 9. It can be seen that when the disturbance on poses is less than three "basis disturbance", or more concretely the translation in each orthogonal direction and the rotation are within three centimeters and three degrees, respectively, the correction results of our scheme are always more accurate than the offline calibrated ones. Since cameras in the surround-view system are fixed on the vehicle, the natural move of cameras due to collisions or bumps won't be violent and it's most probably

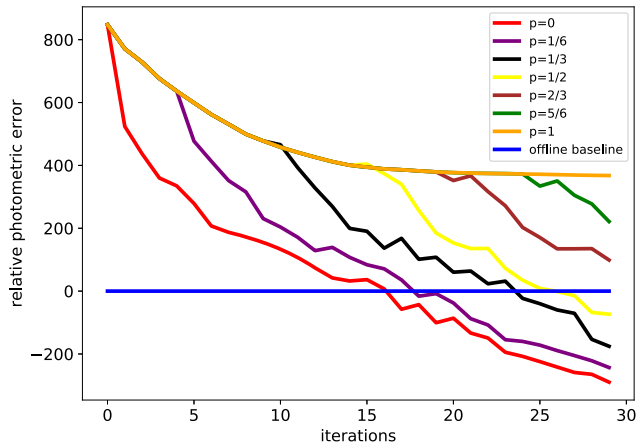


Fig. 8. The relative photometric errors along with the optimization evolvement under different settings of the "first level proportion" p.

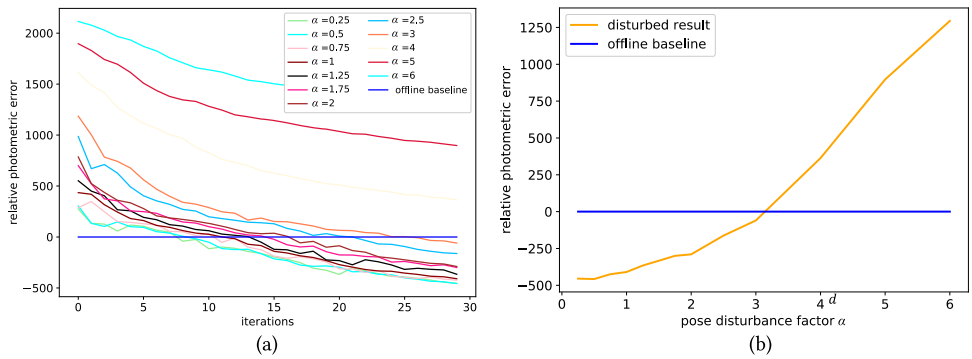


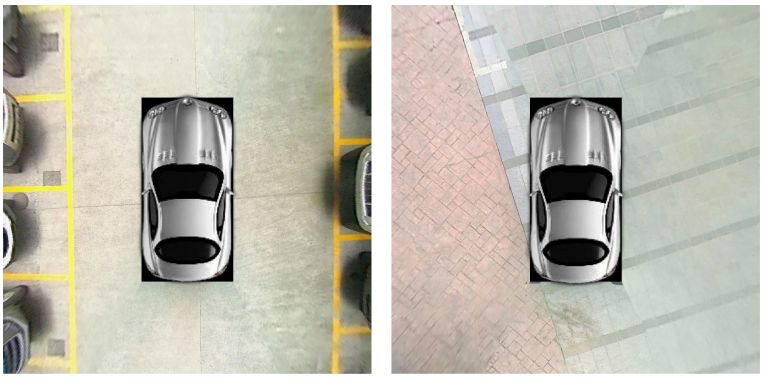
Fig. 9. The relative photometric errors of correction results under different α settings (mentioned in Equation 37). In (a), those curves plot the photometric errors along with the correction evolvement under different α settings. The relationship between the relative photometric errors and different settings of α is plotted as the orange curve in (b), while the blue line is the offline baseline.

in the millimeter scale unless the fixed structures are broken severely. Therefore, in most practical situations, our method is quite robust to the variations of initial poses.

5.4 Failure Case Analysis

Although the online camera poses optimization approach for the surround-view system proposed in this paper can work stably in most cases, it may fail in some cases. Failure cases can be roughly divided into two categories:

Few pixels with large enough gradient moduli. As mentioned in Section 4, the "contribution" of one pixel to the optimization is proportional to its local intensity gradient modulus. If there are no clearly observable textures on the ground around the vehicle, the gradients of most pixels will be close to 0. In such a case, the optimization is mainly guided by noise and thus fails. Figure 10(a) shows a typical example of this case. In the surround-view image shown in Figure 10(a), there is geometric misalignment in the common-view areas between adjacent bird's-eye-views.



(a) (b)

Fig. 10. Two typical failure cases of our approach. (a) was taken from an underground parking lot, and it can be seen that the ground surface covered by the overlapping regions of adjacent bird's-eye-views is lacking clearly observable textures. In (b), camera poses of the system deviate too much from the states of initial offline calibration and consequently, there is severe geometric misalignment between adjacent bird's-eye-views.

Unfortunately, there are no clearly visible textures in these areas and thus the pixels participating in the optimization cannot provide effective gradient information. That will eventually lead to the failure of the camera pose optimization operation. Therefore, it should be emphasized that in order to make our method work successfully, the vehicle needs to be parked on flat ground with clearly observable textures. Besides, in our implementation, we offer a quantitative threshold. That is if the number of qualified pixels is smaller than 4,000 when the fisheye resolution is 1080p, this frame should be abandoned and the user needs to try another operation site.

Excessive changes of camera poses. As our approach follows a sparse direct framework, during the optimization the gradient information of pixels will greatly affect the optimization step. However, the intensity function of the image is with strong non-convexity and discontinuity. Therefore, our method could perform quite well in "fine-tuning" tasks (i.e., the degrees of camera poses' changes are not very high). Otherwise, if the camera poses change too much, our algorithm may fall into a local optimum instead of a global one.

For example, in Figure 10(b), it can't be determined whether the line on the ground in the front view should be aligned with the line above or below in the left view. Actually, it should be matched with the line below, while our algorithm mistakenly aligns it with the line above. That is to say, our approach falls into a local optimum for this case. Generally speaking, if geometric misalignments in the surround-view are extremely serious, it is difficult to successfully correct the camera poses using an online method based on optimization. Quantitatively, according to Figure 9, we found that as the pose change exceeds three "basis disturbance", the correction performance of our method won't be satisfactory. Instead, the best solution in such a case is still re-calibration.

5.5 Comparison with Other Methods

In this subsection, we compare our scheme with the existing relevant methods mentioned in Section 2 qualitatively and quantitatively to show its superiority.

Traits of methods. As we have reviewed in Section 2, there are several studies in the literature that are relevant to our work in this paper. In order to understand the different characteristics of these methods more clearly, in Table 2 we compare them in three aspects: (1) Does it reuse the

Table 2. Qualitative Comparison with Related Methods

method	prior	applicable to surround-view	feature type
Collado et al. [4]	×	×	ground lane
Hold et al. [17]	×	×	ground lane
Hansen et al. [12]	×	×	feature point
Schneider et al. [38]	×	×	odometry
Dang et al. [5]	√	×	feature point
Nedevschi et al. [32]	√	×	ground lane
Knorr et al. [21]	√	×	feature point
Ling and Shen [25]	√	×	feature point
Heng et al. [15]	×	√	odometry
Heng et al. [14]	×	√	odometry
Zhao et al. [44]	×	√	ground lane
Choi et al. [3]	×	√	ground lane
Ours	√	√	sparse pixels

Table 3. Numbers of Frames Required and the Time Costs for Compared Schemes

Scheme	Number of Frames	Time Cost
Schneider et al.'s [38]	500	Not mentioned
Heng et al.'s [15]	2000	About 90 minutes
Heng et al.'s [14]	500	About 10 minutes
Ours	1	10–20s

prior information from the offline calibration? (2) Can it be readily used for the surround-view system? and (3) What kind of features does it rely on? It can be seen that Dang et al.'s method, Ling and Shen's method, Knorr et al.'s method, Nedevschi et al.'s method, and ours can reuse the offline calibration information as a prior. But only our method is applicable to the surround-view system. Besides, since our approach follows a sparse direct framework, it has no dependence on explicit visual features, implying that it has the potential to be more robust and more efficient.

Comparison with odometry based methods. The methods proposed in [38], [15], and [14] are all odometry based. Such methods will perform a series of tasks to construct a stable map, solve the trajectory of the vehicle, and finally re-estimate the camera poses by joint optimization. As mentioned in Section 2.1, such methods are usually more or less cumbersome so that they are unlikely to satisfy the industrial portability requirements. In Table 3, we summarize the required numbers of frames and time cost for [38], [15], [14], and ours to complete the correction. It can be seen that existing odometry based solutions usually take hundreds of frames and spend tens of minutes to yield useful outputs. By contrast, ours only needs one frame and its correction process can be thoroughly completed within ten to twenty seconds, demonstrating that our method is more lightweight and easier to be integrated than its rivals.

Comparison with lane-line based methods. Researches in [4], [32], [17], [44], and [3] all belong to the lane-line based category. As aforementioned, these methods all rely on a strong assumption, "two parallel lane lines on the ground can be captured by the cameras", which is not usually established. Thus, the application scopes of such methods are limited to some extent. For the six groups of data we collected, we counted the ratio of surround-views with lane-lines and surround-views with sufficient textures to all surround-views, respectively. Whether textures are sufficient

Table 4. The Proportion of Data with Enough Features

Feature Type \ Group	A	B	C	D	E	F	Total
Lane-lines	0	0	0	0	1	0.34	0.22
Textures	1	0.84	1	1	0.65	1	0.92

Table 5. Relative Photometric Errors Under Different Settings of Bundle Adjustment

Robust Strategy	Feature Type \ Feature Number	30	50	70	120	150
No Kernel	ORB	1112.52	1965.82	1124.87	1134.37	1383.42
	SURF	1083.56	1187.75	1348.01	928.09	1335.41
	SIFT	986.24	1251.46	1292.31	1204.49	1260.40
Huber Kernel	ORB	1666.56	1707.35	1704.45	991.22	1123.48
	SURF	938.74	1016.63	1100.40	923.27	923.38
	SIFT	959.79	1059.77	1017.47	957.58	972.29
Cauchy Kernel	ORB	879.66	873.11	916.27	906.39	883.17
	SURF	919.57	916.91	914.91	923.77	853.04
	SIFT	925.07	904.82	886.43	874.78	876.81
Tukey Kernel	ORB	936.64	937.95	932.93	925.02	933.36
	SURF	923.61	928.46	937.33	923.67	932.40
	SIFT	924.85	928.25	930.23	931.19	932.37
Before Correction				865.76		
Ours				**−289.73**		

depends on whether the number of qualified pixels selected by our pixel selection strategy is sufficient. The results are summarized in Table 4. From Table 4, it can be seen that for normal flat grounds, the cases having rich textures are much more than those having clear and regular lane-lines (at least on the dataset we collected). It implies that the potential application scope of the camera pose correction model proposed in this article is much wider than the ones that rely on lane-lines.

Comparison with bundle adjustment based methods. Typical existing bundle adjustment based solutions for camera poses correction include [12], [5], [21], and [25], and it can be noticed that they are all designed for common multi-camera systems. Actually, bundle adjustment based methods have a notable defect when they are extended to adapt to the surround-view case.

The most fatal shortcoming of bundle adjustment based solutions for the surround-view case is that high-quality paired features are hard to obtain. As mentioned in Section 2.1, the common-view regions between adjacent cameras in the surround-view system are usually narrow and of large distortion. When extracting and matching features on such small and severely distorted image regions, lots of mismatches will be foreseeable. To make our claim more intuitive and convincing, related experiments were conducted. We implemented a bundle adjustment based pose correction pipeline and used it to correct camera poses under different experimental settings, including different types of features (SIFT [28], SURF [2], and ORB [37]), different numbers of features, different types of kernel functions (Huber, Cauchy, and Tukey), etc. The flow of the pipeline is as follows. We firstly performed feature extraction in the common-view regions of adjacent cameras, and used the brute-force matcher for feature matching. During matching, the "ratio test" was also utilized. Then, the matched features were triangulated, and the camera poses were optimized via bundle adjustment. Finally, we recorded the average relative photometric error over all surround-views. The relative photometric errors under different settings are summarized in Table 5.

The experimental results demonstrate that the results of bundle adjustment based methods are unlikely to yield usable calibration outputs, since the corresponding photometric errors are even larger than errors before the correction. Therefore, it can be concluded that for the surround-view case, compared with bundle adjustment based ones, our scheme has obvious superiority in both the accuracy and the usability.

6 CONCLUSION AND FUTURE WORK

In this paper, we studied a practical problem, online correction of cameras' extrinsics for the surround-view system, emerging from the field of advanced driving assistance system, and proposed a solution. Our method is of a two-level cascade structure and each level is based on one model designed by us, the ground model and the ground-camera model. Both of the models follow a sparse direct framework and can correct the camera poses by minimizing the system's overall photometric error without feature matching. The ground model establishes the relationship between the photometric error and cameras' poses, while the ground-camera model solves the degree-of-freedom loss problem in further. With a cascade structure, an appropriate balance between speed and accuracy can be achieved. A novel pixel selection strategy was also proposed, by which pixels from "mismatched objects" can be eliminated effectively and the performance of our scheme can be consequently enhanced in both the speed and the accuracy. An outstanding merit of our solution is that it only relies on one frame and has little requirements on environmental conditions. As long as the vehicle is driving on a normal flat road with relatively rich textures, our scheme will work well. In addition, it does not require additional apparatuses or calibration sites. Experimental results show that our method can effectively eliminate the geometric misalignment in the surround-view images and thus reduce the photometric errors caused by changes of camera poses. However, up to now, the performance of our approach in the environment having low texture or strong texture repeatability is still not satisfactory and thus we will continue to devote efforts in this area.

REFERENCES

[1] R. Battiti. 1992. First- and second-order methods for learning: Between steepest descent and Newton's method. *Neural Computation* 4, 2 (1992), 141–166.

[2] H. Bay, T. Tuytelaars, and L. V. Gool. 2006. SURF: Speeded up robust features. In *Proc. European Conf. Comput. Vis.* 404–417.

[3] K. Choi, H. Jung, and J. Suhr. 2018. Automatic calibration of an around view monitor system exploiting lane markings. *Sensors* 18, 9 (2018), 2956:1–26.

[4] J. Collado, C. Hilario, A. Escalera, and J. Armingol. 2006. Self-calibration of an on-board stereo-vision system for driver assistance systems. In *Proc. Int'l IEEE Conf. Intell. Vehicles Symposium.* 156–162.

[5] T. Dang and C. Hoffmann. 2006. Tracking camera parameters of an active stereo rig. In *Joint DAGM Symposium.* 627-636.

[6] J. E. Dennis and R. B. Schnabel. 1983. Numerical methods for unconstrained optimization and nonlinear equations. *Prentice Hall, Inc.* 28, 3 (1983), 417–419.

[7] F. Du and M. Brady. 1993. Self-calibration of the intrinsic parameters of cameras for active vision systems. In *Proc. IEEE Int'l Conf. Comput. Vis. Pattern Recognit.* 477–482.

[8] J. Engel, V. Koltun, and D. Cremers. 2018. Direct sparse odometry. *IEEE Trans. Pattern Analysis and Machine Intell.* 40, 3 (2018), 611–625.

[9] J. Engel, T. Schöps, and D. Cremers. 2014. LSD-SLAM: Large-scale direct monocular SLAM. In *Proc. European Conf. Comput. Vis.* 834–849.

[10] C. Forster, M. Pizzoli, and D. Scaramuzza. 2014. SVO: Fast semi-direct monocular visual odometry. In *Proc. IEEE Int'l Conf. Robotics and Automation.* 15–22.

[11] M. Gressmann, G. Palm, and O. Löhlein. 2011. Surround view pedestrian detection using heterogeneous classifier cascades. In *Proc. Int'l IEEE Conf. Intell. Transportation Systems.* 1317–1324.

[12] P. Hansen, H. Alismail, P. Rander, and B. Browning. 2012. Online continuous stereo extrinsic parameter estimation. In *Proc. IEEE Int'l Conf. Comput. Vis. Pattern Recognit.* 1059–1066.
[13] R. Hartley and A. Zisserman. 2003. *Multiple View Geometry in Computer Vision* (2 ed.). Cambridge University Press, USA.
[14] L. Heng, M. Bürki, G. H. Lee, P. Furgale, R. Siegwart, and M. Pollefeys. 2014. Infrastructure-based calibration of a multi-camera rig. In *Proc. IEEE Int'l Conf. Robotics and Automation.* 4912–4919.
[15] L. Heng, B. Li, and M. Pollefeys. 2013. CamOdoCal: Automatic intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry. In *Proc. IEEE/RSJ Int'l Conf. Intell. Robots and Systems.* 1793–1800.
[16] W. C. Hoffman. 1966. The Lie algebra of visual perception. *J. Mathematical Psychology* 3, 1 (1966), 65–98.
[17] S. Hold, S. Görmer, A. Kummert, M. Meuter, and S. Muller-Schneiders. 2009. A novel approach for the online initial calibration of extrinsic parameters for a car-mounted camera. In *Proc. Int'l IEEE Conf. Intell. Transportation Systems.* 420–425.
[18] C. Hou, H. Ai, and S. Lao. 2007. Multiview pedestrian detection based on vector boosting. In *Proc. Asian Conf. Comput. Vis.* 18–22.
[19] M. Irani and P. Anandan. 1999. About direct methods. In *Proc. Int'l Workshop on Vis. Algorithms.* 267–277.
[20] R. Klette, A. Koschan, and K. Schluns. 1998. *Computer Vision: Three-dimensional Data from Images.* Springer, Singapore.
[21] M. Knorr, W. Niehsen, and C. Stiller. 2013. Online extrinsic multi-camera calibration using ground plane induced homographies. In *Proc. IEEE Intell. Vehicles Symposium.* 236–241.
[22] Pierre Lébraly, Eric Royer, Omar Ait-Aider, Clément Deymier, and Michel Dhome. 2011. Fast calibration of embedded non-overlapping cameras. In *Proc. IEEE Int'l Conf. Robotics and Automation.* 221–227.
[23] L. Li, L. Zhang, X. Li, X. Liu, Y. Shen, and L. Xiong. 2017. Vision-based parking-slot detection: A benchmark and a learning-based approach. In *Proc. IEEE Int'l Conf. Multimedia and Expo.* 649–654.
[24] C. Lin and M. Wang. 2012. A vision based top-view transformation model for a vehicle parking assistant. *Sensors* 12, 4 (2012), 4431–4446.
[25] Y. Ling and S. Shen. 2016. High-precision online markerless stereo extrinsic calibration. In *Proc. IEEE/RSJ Int'l Conf. Intell. Robots and Systems.* 1771–1778.
[26] X. Liu, L. Zhang, Y. Shen, S. Zhang, and S. Zhao. 2019. Online camera pose optimization for the surround-view system. In *Proc. ACM Int'l Conf. Multimedia.* 383–391.
[27] M. I. A. Lourakis. 2019. Sparse non-linear least squares optimization for geometric vision. In *Proc. European Conf. Comput. Vis.* 43–56.
[28] D. G. Lowe. 2004. Distinctive image features from scale-invariant keypoints. *Int'l J. Comput. Vis.* 60, 2 (2004), 91–110.
[29] B. D. Lucas and T. Kanade. 1981. An iterative image registration technique with an application to stereo vision. In *Proc. Int'l Joint Conf. Artificial Intell.* 674–679.
[30] J. J. Moré. 1978. The Levenberg-Marquardt algorithm: Implementation and theory. In *Numerical Analysis.*
[31] R. Mur-Artal and J. D. Tardós. 2017. ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras. *IEEE Trans. Robotics* 33, 5 (2017), 1255–1262.
[32] S. Nedevschi, C. Vancea, T. Marita, and T. Graf. 2007. Online extrinsic parameters calibration for stereo vision systems used in far-range detection vehicle applications. *IEEE Trans. Intell. Transportation Systems* 8, 4 (2007).
[33] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. 2011. DTAM: Dense tracking and mapping in real-time. In *Proc. IEEE Int'l Conf. Comput. Vis.* 2320–2327.
[34] F. Nielsen. 2005. Surround video: A multihead camera approach. *The Visual Computer* 21, 1-2 (2005), 92–103.
[35] J. Nocedal. 1992. Theory of algorithms for unconstrained optimization. *Acta Numerica* 1, 8 (1992), 199–242.
[36] J. M. M. Montiel R. Mur-Artal and J. D. Tardós. 2015. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Trans. Robotics* 31, 5 (2015), 1147–1163.
[37] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. 2011. ORB: An efficient alternative to SIFT or SURF. *Proc. IEEE Int'l Conf. Comput. Vis.* (2011), 2564–2571.
[38] S. Schneider, T. Luettel, and H. Wuensche. 2013. Odometry-based online extrinsic sensor calibration. In *Proc. IEEE/RSJ Int'l Conf. Intell. Robots and Systems.* 1287–1292.
[39] X. Shao, X. Liu, L. Zhang, S. Zhao, Y. Shen, and Y. Yang. 2019. Revisit surround-view camera system calibration. In *Proc. IEEE Int'l Conf. Multimedia and Expo.* 1486–1491.
[40] R. W. M. Wedderburn. 1974. Quasi-likelihood functions, generalized linear models, and the Gauss-Newton method. *Biometrika* 61, 3 (1974), 439–447.
[41] J. Xu, G. Chen, and M. Xie. 2000. Vision-guided automatic parking for smart car. In *Proc. IEEE Intell. Vehicles Symposium.* 725–730.
[42] L. Zhang, J. Huang, X. Li, and L. Xiong. 2018. Vision-based parking-slot detection: A DCNN-based approach and a large-scale benchmark dataset. *IEEE Trans. Image Processing* 27, 11 (2018), 5350–5364.

[43] Z. Zhang. 1999. Flexible camera calibration by viewing a plane from unknown orientations. In *Proc. IEEE Int'l Conf. Comput. Vis.* 666–673.
[44] K. Zhao, U. Iurgel, M. Meuter, and J. Pauli. 2014. An automatic online camera calibration system for vehicular applications. In *Proc. Int'l IEEE Conf. Intell. Transportation Systems.* 1490–1492.
[45] H. Zhu, J. Yang, and Z. Liu. 2009. Fisheye camera calibration with two pairs of vanishing points. In *Proc. Int'l Conf. Inf. Tech. Comput. Sci.* 321–324.

Received May 2021; revised October 2021; accepted December 2021