Online Camera Pose Optimization for the Surround-view System

Xiao Liu
School of Software Engineering,
Tongji University
Shanghai, China
1532787@tongji.edu.cn

Lin Zhang*
School of Software Engineering,
Tongji University
Shanghai, China
cslinzhang@tongji.edu.cn

Ying Shen*
School of Software Engineering,
Tongji University
Shanghai, China
yingshen@tongji.edu.cn

Shaoming Zhang
College of Surveying and
Geo-informatics, Tongji University
Shanghai, China
08053@tongji.edu.cn

Shengjie Zhao
School of Software Engineering,
Tongji University
Shanghai, China
shengjiezhao@tongji.edu.cn

ABSTRACT

Surround-view system is an important information medium for drivers to monitor the driving environment. A typical surround-view system consists of four to six fish-eye cameras arranged around the vehicle. From these camera inputs, a top-down image of the ground around the vehicle, namely the surround-view image can be generated with well calibrated camera poses. Although existing surround-view system solutions can estimate camera poses accurately in offline environment, how to correct the camera poses' change in online environment is still an open issue. In this paper, we propose a camera pose optimization method for surround-view system in online environment. Our method consists of two models: Ground Model and Ground-Camera Model, both of which correct the camera poses by minimizing photometric errors between ground projections of adjacent cameras. Experiments show that our method can effectively correct the geometric misalignment of the surround-view image caused by camera poses' change. Since our method is highly automated with low requirement of calibration site and manual operation, it has a wide range of applications and is convenient for the end-users. To make the results reproducible, the source code is publicly available at https://cslinzhang.github.io/CamPoseOpt/.

CCS CONCEPTS

• **Computing methodologies → Camera calibration**.

KEYWORDS

Surround-view system, ADAS, camera pose optimization, photometric error minimization

ACM Reference Format:
Xiao Liu, Lin Zhang, Ying Shen, Shaoming Zhang, and Shengjie Zhao. 2019. Online Camera Pose Optimization for the Surround-view System. In *Proceedings of the 27th ACM International Conference on Multimedia (MM '19), October 21–25, 2019, Nice, France.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3343031.3350885

1 INTRODUCTION

Surround-view system is an emerging ADAS (Advanced Driver Assistance System) technology and an important information media for drivers to monitor the driving environment [12, 14, 16, 17]. It can generate a top-down image of ground around the vehicle, namely the surround-view image, which can provide information of the road conditions for drivers. In recent years, surround-view images have also been widely used in traffic sign recognition, parking-slot detection and other computer vision tasks in autonomous driving [2, 7, 11, 21]. A typical surround-view system consists of four to six fish-eye cameras arranged around the vehicle. By using the existing calibration methods for the surround-view system, the camera poses of the system can be estimated quite accurately [9, 10, 15, 19]. Then, the images captured by the surround-view system can be projected onto the ground with the estimated camera poses. Finally, a seamless surround-view image can be generated.

However, most of the existing methods are designed for the offline environment, such as an automobile factory. The camera poses calibrated with these methods may change after the automobiles leave the factory for daily-use. In daily-use scenarios, namely the online environment, the camera poses may change due to bumps, collisions and tire pressure changes, which will cause obvious geometric misalignment at stitching boundaries in the surround-view images. In such a situation, most commercial surround-view solutions need to take the automobile back to the factory with calibration sites and technical supports to re-calibrate the system, which is very troublesome for both the automobile enterprises and the clients.

In this paper, we propose a new camera pose optimization method for the surround-view system. The method is oriented to the online environment and can optimize the camera poses on a flat ground with natural texture, which is very common in daily use. In the meanwhile, the calibration process is highly automated and does

[*] Corresponding Author: Lin Zhang and Ying Shen

MM '19, October 21–25, 2019, Nice, France
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6889-6/19/10…$15.00
https://doi.org/10.1145/3343031.3350885

not require any manual operation. After being corrected by our proposed method with only one frame, the new camera poses can still be valid for a period.

1.1 Related Work

Most existing extrinsic calibration methods for the surround-view system estimated the camera poses through geometric alignment with specific kinds of image features. According to what kind of features are used, existing methods can be categorized into two classes: pattern-based approaches and feature-based approaches.

The pattern-based approaches estimated the camera poses with special patterns like chessboard or point array, which are precisely drawn on a calibration board. In [18, 20, 22], the authors adopted a factorization based method to calibrate the multi-camera system by placing a calibration pattern between adjacent cameras. Gao *et al.* [6] proposed a method that adopted a colinear constraint to estimate coordinate transformation with a special designed checkerboard. However, since these methods do not take the loop closure relation of the surround-view system into consideration, the calibration results usually have accumulative error. To guarantee precision of the calibration result, Natroshvili and Scholl [15] jointly optimized the camera poses after the orientation of each camera has been estimated with a fixed pattern placed on the ground. Analogously, Liu *et al.* [13] and Zhang *et al.* [19] proposed a surround-view camera solution for embedded system by using bundle adjustment to minimize geometric misalignment. In [8], Hedi and Loncaric estimated homograph matrix by mapping 3D coordinates of chessboard corners to their 2D coordinates on the images, then minimized stitching error within the overlapping areas of adjacent cameras. Although the camera poses can be estimated highly accurately by pattern-based methods, most of them require large calibration sites, which are inconvenient for end-users. Once the camera poses change, the users must seek for professional assistance to re-calibrate the surround-view system.

The feature-based approaches estimated the camera poses with natural features such as corners, lines and photometric information in the images. Heng *et al.* [9, 10] proposed infrastructure-based calibration methods which leveraged on multi-sensor SLAM to build a highly accurate map of a calibration area using an already calibrated robotic system. Given camera intrinsic parameters, the vehicle was required to be driven in the calibration area for a period of time to complete the calibration process. By specifying corresponding feature points on the ground in adjacent images, Liu *et al.* [14] calculated the homography matrix of the camera to the ground to generate the surround-view image. In [1], Choi *et al.* calibrated multiple cameras by automatically finding corresponding lane markings across images of adjacent cameras. Zhao *et al.* [24] utilized multiple vanishing points of lane markings for camera orientation calibration with the weighted least squares method, followed by a tracking process with Kalman Filter for better consistency and robustness. Although feature-based approaches are more flexible than pattern-based approaches, they also introduce new limitations like a prepared map of the calibration area or particular pattern of lane markings. In addition, it is difficult for feature-based approaches to obtain accurate camera poses, which influences the stitching effect of the surround-view image.

To our knowledge, most of existing extrinsic calibration methods for the surround-view system are designed for the off-line environment. The pattern-based approaches are more accurate but need calibration sites, while the feature-based approaches are easy to use but not accurate enough. That is to say, these methods can not meet both the convenience and accuracy requirements of camera pose correction in the online environment.

1.2 The Motivations and Contributions

Through the literature survey, we find that existing methods of surround-view system calibration have limitations in the following aspects:

(1) Existing methods are not designed for the online environment. Although existing methods work well in the off-line environment, once the camera poses change, these methods need to re-calibrate the surround-view system, which is very cumbersome. In fact, there are initial camera poses available under the online condition. However, to our knowledge, no existing method takes advantage of this important information.

(2) As mentioned in Sect. 1.1, existing methods are not able to balance the convenience of calibration process and the accuracy of calibration results, which are key demands of the end-users. The pattern-based methods are accurate, but they require precisely drawn calibration sites and manual operations to assist the calibration process, which is troublesome. On the contrary, the feature-based methods are relatively flexible and have lower dependence on the calibration sites, but their accuracy is not satisfactory.

Therefore in this paper, we attempt to fill the above-mentioned research gaps to some extent. Our major motivations and contributions are summarized as follows:

(1) Aiming at the characteristics of the online environment, we propose a camera pose optimization method which fully exploits the initial camera poses of the calibrated surround-view system. The method consists of two models: Ground Model and Ground-Camera Model. Considering the lack of reliable feature points in the online environment, both of the two models correct the camera poses by minimizing the photometric error between ground projections of adjacent cameras. The optimization objective of the Ground Model is consistent with the objective of surround-view image stitching. This model theoretically establishes the relationship between the camera poses and the photometric error of the surround-view image. It can correct particular types of camera poses' change. On the other hand, the Ground-Camera Model steps further on the basis of the Ground Model and solves the problem of DOF (degree of freedom) loss. Since Ground-Camera Model is more general and complete in theory, it has a wider application range.

(2) In order to meet the convenience and accuracy requirements of the end-users, we design and implement a highly automated algorithm based on our proposed models. It only needs a flat ground with rich texture where no fixed pattern or regular shape is necessary. In the meanwhile, the calibration process does not require any manual operation. Experiments show that our algorithm can effectively correct the geometric misalignment in the surround-view image caused by camera poses' change. After the camera pose correction, the stitching error of the surround-view image is

significantly reduced and the visual effect is greatly improved as well.

2 OVERVIEW OF SURROUND-VIEW SYSTEM

This section describes how to generate a surround-view image from the images captured by the surround-view system.

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

$$\boldsymbol{p}_{C_i} = \frac{1}{Z_{C_i}} \boldsymbol{K}_{C_i} \boldsymbol{T}_{C_iG} \boldsymbol{P}_G \tag{1}$$

where $\boldsymbol{K}_{C_i}$ is the intrinsic matrix [23] of the camera C_i, and Z_{C_i} is the depth of the point in the camera's coordinate system.

Besides, assume that the world is wide and flat, 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. For a point $\boldsymbol{p}_G = [u_G, v_G, 1]^T$ in the bird's-eye-view image, the relationship between $\boldsymbol{p}_G$ and $\boldsymbol{P}_G$ can be presented 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, 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 by $\boldsymbol{K}_G$, the Eq. 2 can be written as,

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

By combining Eq. 1 and Eq. 3, we can get,

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

Using Eq. 4, we can project the 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 image captured by camera C_i, $\boldsymbol{I}_{GC_i}$ is the ground projection of $\boldsymbol{I}_{C_i}$, namely the bird's-eye-view image. By projecting the images of the four cameras onto the ground and choosing the appropriate stitching seam, the surround-view image can be generated.

3 CAMERA POSE OPTIMIZATION

Using correct camera poses, we can generate an almost seamless surround-view image with the model in the previous section. However, in online environment, the camera poses are often disturbed. For example, vehicle bumps can cause the cameras shaking, which may change the camera poses. In addition, the change of tire pressure will make the whole camera system move closer or farther to the ground. All these factors will lead to misalignment on the generated surround-view image.

In order to solve the problem of camera poses' change during vehicle driving, we propose a new camera pose optimization approach. Our approach draws lessons from direct methods [3–5] in SLAM field. It estimates optimal camera poses by optimizing the photometric error between specific images' projections. In the following subsections, we will introduce two models in our approach, namely Ground Model and Ground Camera Model. The Ground Model theoretically establishes the relationship between the camera poses and the photometric error of the surround-view image, while the Ground-Camera Model steps further on the basis of the Ground Model and solves the problem of DOF (degree of freedom) loss.

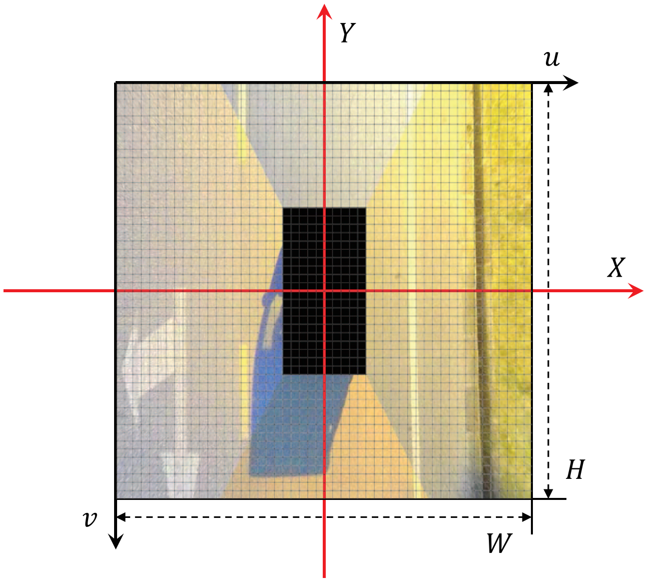


Figure 1: Relationship between the ground coordinate system and the surround-view image coordinate system. ***W*** **and** ***H*** **are the width and height of the scope covered by the bird's-eye-view image.**

3.1 Ground Model

Suppose C_i and C_j are two adjacent cameras in a surround-view system. Their projections on the ground are $\boldsymbol{I}_{GC_i}$ and $\boldsymbol{I}_{GC_j}$, respectively. The photometric error of a point $\boldsymbol{p}_G$ on the ground in $\boldsymbol{I}_{GC_i}$ and $\boldsymbol{I}_{GC_j}$ 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,

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

where ξ_{GC_i} is the Lie algebra representation of the transformation matrix from the camera C_i to the ground $\boldsymbol{T}_{GC_i}$. Put Eq. 7 into Eq. 6, we can get,

$$\begin{aligned} \varepsilon_{\boldsymbol{p}_G} = \| & \boldsymbol{I}_{GC_i}(\boldsymbol{K}_G \exp(\xi_{GC_i}) \boldsymbol{P}_{C_i}) \\ & - \boldsymbol{I}_{GC_j}(\boldsymbol{K}_G \exp(\xi_{GC_j}) \boldsymbol{P}_{C_j}) \|_2 \end{aligned} \tag{8}$$

Then the optimization objective of Ground Model can be defined as,

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

where $\mathcal{N}_{ij}$ is the overlapping region of $\boldsymbol{I}_{GC_i}$ and $\boldsymbol{I}_{GC_j}$.

To optimize this objective function, we need to analyze the derivative relationship between $\varepsilon_{\boldsymbol{p}_G}$ and ξ_{GC_i}. The Jacobian of $\varepsilon_{\boldsymbol{p}_G}$ to ξ_{GC_i} can be expressed as,

$$\boldsymbol{J}_i = \frac{\partial \varepsilon_{\boldsymbol{p}_G}}{\partial \xi_{GC_i}} \tag{10}$$

With $\boldsymbol{P}_G = \exp(\xi_{GC_i})\boldsymbol{P}_{C_i}$, it can be decomposed as,

$$\boldsymbol{J}_i = \frac{\partial \varepsilon_{\boldsymbol{p}_G}}{\partial \boldsymbol{I}_{GC_i}} \frac{\partial \boldsymbol{I}_{GC_i}}{\partial \boldsymbol{p}_G} \frac{\partial \boldsymbol{p}_G}{\partial \boldsymbol{P}_G} \frac{\partial \boldsymbol{P}_G}{\partial \xi_{GC_i}} \tag{11}$$

Next, we will discuss these four parts separately:

(1) $\partial \varepsilon_{\boldsymbol{p}_G}/\partial \boldsymbol{I}_{GC_i}$ is the derivative of photometric error to image pixel intensity, denoted by δ. Suppose that $\boldsymbol{I}_{GC_i}$ is a gray scale image,

$$\delta = \frac{\partial \varepsilon_{\boldsymbol{p}_G}}{\partial \boldsymbol{I}_{GC_i}} = \boldsymbol{I}_{GC_i}(\boldsymbol{p}_G) - \boldsymbol{I}_{GC_j}(\boldsymbol{p}_G) \tag{12}$$

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

$$\frac{\partial \boldsymbol{I}_{GC_i}}{\partial \boldsymbol{p}_G} = \begin{bmatrix} \frac{\partial \boldsymbol{I}_{GC_i}}{\partial u_G} & \frac{\partial \boldsymbol{I}_{GC_i}}{\partial v_G} \end{bmatrix} = \begin{bmatrix} \nabla u_G & \nabla v_G \end{bmatrix} \tag{13}$$

(3) $\partial \boldsymbol{p}_G/\partial \boldsymbol{P}_G$ is the derivative of a pixel's 2D coordinate to its 3D coordinate,

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

(4) $\partial \boldsymbol{P}_G/\partial \xi_{GC_i}$ is the derivative of 3D coodinate $\boldsymbol{P}_G$ to Lie algebra ξ_{GC_i},

$$\begin{aligned} \frac{\partial \boldsymbol{P}_G}{\partial \xi_{GC_i}} &= \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{P}_G^{\wedge} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 & Z_G & -Y_G \\ 0 & 1 & 0 & -Z_G & 0 & X_G \\ 0 & 0 & 1 & Y_G & -X_G & 0 \end{bmatrix} \end{aligned} \tag{15}$$

where $\boldsymbol{P}_G^{\wedge}$ is the anti-symmetric matrix of $\boldsymbol{P}_G$.

By combining the above four parts, noticing that $\boldsymbol{p}_G$ is a point on the ground plane $Z_G = 0$, the final form of $\boldsymbol{J}_i$ can be expressed as,

$$\boldsymbol{J}_i = \delta \begin{bmatrix} \frac{\nabla u_G}{d_{X_G}} & -\frac{\nabla v_G}{d_{Y_G}} & 0 & 0 & 0 & -\frac{\nabla u_G Y_G}{d_{X_G}} - \frac{\nabla v_G X_G}{d_{Y_G}} \end{bmatrix} \tag{16}$$

Once $\boldsymbol{J}_i$ is obtained, Eq. 9 can be iteratively optimized with conventional optimization methods, such as Gradient Descent, Gauss-Newton method and Levenberg-Marquardt algorithm. Take the Gradient Descent method as an example, for the n_{th} iteration, the camera pose $\xi_{GC_i}^n$ can be updated by,

$$\xi_{GC_i}^n \leftarrow \xi_{GC_i}^{n-1} - \alpha \boldsymbol{J}_i^n \tag{17}$$

where α is the rate factor.

For the surround-view system, we can optimize all camera poses together by minimizing the photometric error of each camera and its adjacent cameras. The optimization objective of the surround-view system can be expressed as,

$$\xi_{GC_i}^* = \underset{\xi_{GC_i}^*}{\arg\min} \sum_{i=1}^{4} \sum_{j \in \Omega(i)} \sum_{\boldsymbol{p}_G \in \mathcal{N}_{ij}} \varepsilon_{\boldsymbol{p}_G} \tag{18}$$

where j is the index of C_i's adjacent cameras.

Although the Ground Model can solve the problem of camera pose optimization to some extent, it also has an obvious shortcoming. The camera pose ξ_{GC_i} has 6 DOF (degree of freedom), but the Jacobian matrix $\boldsymbol{J}_i$ calculated by the Ground Model has only 3 DOF, which means 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. To solve the problem of DOF loss, we propose a more universal method called Ground-Camera Model.

3.2 Ground-Camera Model

Unlike Ground Model, Ground-Camera Model uses a different projection plane to calculate photometric error. To correct the pose of camera C_i, we firstly project $\boldsymbol{I}_{GC_j}$ to camera C_i. Suppose that the projection of bird's-eye-view image $\boldsymbol{I}_{GC_j}$ on camera C_i is $\boldsymbol{I}_{GC_j}^{C_i}$, the photometric error of camera C_i in Ground-Camera Model can be defined as,

$$\varepsilon_{\boldsymbol{p}} =\| \boldsymbol{I}_{C_i}(\boldsymbol{p}) - \boldsymbol{I}_{GC_j}^{C_i}(\boldsymbol{p}) \|_2 \tag{19}$$

where $\boldsymbol{p}$ is a point on the imaging plane of the camera C_i. Similar to Ground Model, $\boldsymbol{p}$ is firstly expanded into a form with the camera pose,

$$\boldsymbol{p} = \frac{1}{Z_{C_i}} \boldsymbol{K}_{C_i} \exp(\xi_{C_i G}) \boldsymbol{P}_G \tag{20}$$

where $\boldsymbol{K}_{C_i}$ is the intrinsic matrix of C_i,

$$\boldsymbol{K}_{C_i} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{21}$$

and,

$$\boldsymbol{P}_{C_i} = \exp(\xi_{C_i G}) \boldsymbol{P}_G = \begin{bmatrix} X_{C_i} & Y_{C_i} & Z_{C_i} \end{bmatrix}^T \tag{22}$$

Then the photometric error can be written as,

$$\begin{aligned} \varepsilon_{\boldsymbol{p}} =& \| \boldsymbol{I}_{C_i}(\frac{1}{Z_{C_i}} \boldsymbol{K}_{C_i} \exp(\xi_{C_i G}) \boldsymbol{P}_G) \\ &- \boldsymbol{I}_{GC_j}^{C_i}(\frac{1}{Z_{C_i}} \boldsymbol{K}_{C_i} \exp(\xi_{C_i G}) \boldsymbol{P}_G) \|_2 \end{aligned} \tag{23}$$

By analyzing the derivative relationship between $\varepsilon_{\boldsymbol{p}}$ and $\xi_{C_i G}$, the Jacobian of $\varepsilon_{\boldsymbol{p}}$ to $\xi_{C_i G}$ can be decomposed into,

$$\begin{aligned} \boldsymbol{J} = \frac{\partial \varepsilon_{\boldsymbol{p}}}{\partial \xi_{C_i G}} =& \frac{\partial \varepsilon_{\boldsymbol{p}}}{\partial \boldsymbol{I}_{C_i}} \frac{\partial \boldsymbol{I}_{C_i}}{\partial \boldsymbol{p}} \frac{\partial \boldsymbol{p}}{\partial \boldsymbol{P}_{C_i}} \frac{\partial \boldsymbol{P}_{C_i}}{\partial \xi_{C_i G}} \\ &+ \frac{\partial \varepsilon_{\boldsymbol{p}}}{\partial \boldsymbol{I}_{GC_j}^{C_i}} \frac{\partial \boldsymbol{I}_{GC_j}^{C_i}}{\partial \boldsymbol{p}} \frac{\partial \boldsymbol{p}}{\partial \boldsymbol{P}_{C_i}} \frac{\partial \boldsymbol{P}_{C_i}}{\partial \xi_{C_i G}} \end{aligned} \tag{24}$$

Obviously, this formula can be divided into two symmetric main parts with four small parts each. The four small parts are:

(1) $\partial \varepsilon_{\boldsymbol{p}}/\partial \boldsymbol{I}_{C_i}$ and $\partial \varepsilon_{\boldsymbol{p}}/\partial \boldsymbol{I}_{GC_j}^{C_i}$ are the derivative of photometric error to the two images. Suppose that

$$\delta = \boldsymbol{I}_{C_i}(\boldsymbol{p}) - \boldsymbol{I}_{GC_j}^{C_i}(\boldsymbol{p}) \tag{25}$$

then

$$\frac{\partial \varepsilon_{\boldsymbol{p}}}{\partial \boldsymbol{I}_{C_i}} = \delta, \frac{\partial \varepsilon_{\boldsymbol{p}}}{\partial \boldsymbol{I}_{GC_j}^{C_i}} = -\delta \tag{26}$$

(2) $\partial \boldsymbol{I}_{C_i}/\partial \boldsymbol{p}$ and $\partial \boldsymbol{I}_{GC_j}^{C_i}/\partial \boldsymbol{p}$ are the gradient of $\boldsymbol{I}_{C_i}$ and $\boldsymbol{I}_{GC_j}^{C_i}$ at $\boldsymbol{p}$. Suppose that $\boldsymbol{p} = \begin{bmatrix} u & v \end{bmatrix}$, then,

$$\frac{\partial \boldsymbol{I}_{C_i}}{\partial \boldsymbol{p}} = \begin{bmatrix} \nabla_i u & \nabla_i v \end{bmatrix} \tag{27}$$

$$\frac{\partial \boldsymbol{I}_{GC_j}^{C_i}}{\partial \boldsymbol{p}} = \begin{bmatrix} \nabla_j u & \nabla_j v \end{bmatrix} \tag{28}$$

(3) $\partial \boldsymbol{p}/\partial \boldsymbol{P}_{C_i}$ is the derivative of 2D coordinate to 3D coordinate,

$$\frac{\partial \boldsymbol{p}}{\partial \boldsymbol{P}_{C_i}} = \begin{bmatrix} \frac{f_x}{Z_{C_i}} & 0 & -\frac{f_x X_{C_i}}{Z_{C_i}^2} \\ 0 & \frac{f_y}{Z_{C_i}} & -\frac{f_y Y_{C_i}}{Z_{C_i}^2} \end{bmatrix} \tag{29}$$

(4) $\partial \boldsymbol{P}_{C_i}/\partial \boldsymbol{\xi}_{C_i G}$ is the derivative of 3D coordinate $\boldsymbol{P}_{C_i}$ to Lie algebra $\boldsymbol{\xi}_{C_i G}$,

$$\begin{aligned} \frac{\partial \boldsymbol{P}_{C_i}}{\partial \boldsymbol{\xi}_{C_i G}} &= \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{P}_{C_i}^{\wedge} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 & Z_{C_i} & -Y_{C_i} \\ 0 & 1 & 0 & -Z_{C_i} & 0 & X_{C_i} \\ 0 & 0 & 1 & Y_{C_i} & -X_{C_i} & 0 \end{bmatrix} \end{aligned} \tag{30}$$

By merging the decomposed parts of Eq. 24, the Jacobian $\boldsymbol{J}$ can be expressed as,

$$\begin{gathered} \boldsymbol{J} = \delta \begin{bmatrix} \nabla_i u - \nabla_j u & \nabla_i v - \nabla_j v \end{bmatrix} \\ \begin{bmatrix} \frac{f_x}{Z_{C_i}} & 0 & -\frac{f_x X_{C_i}}{Z_{C_i}^2} & -\frac{f_x X_{C_i} Y_{C_i}}{Z_{C_i}^2} & f_x + \frac{f_x X_{C_i}^2}{Z_{C_i}^2} & -\frac{f_x Y_{C_i}}{Z_{C_i}} \\ 0 & \frac{f_y}{Z_{C_i}} & -\frac{f_y Y_{C_i}}{Z_{C_i}^2} & -f_y - \frac{f_y Y_{C_i}^2}{Z_{C_i}^2} & \frac{f_y X_{C_i} Y_{C_i}}{Z_{C_i}^2} & \frac{f_y X_{C_i}}{Z_{C_i}} \end{bmatrix} \end{gathered} \tag{31}$$

After $\boldsymbol{J}$ is obtained, we can minimize the photometric error $\varepsilon_{\boldsymbol{p}}$ by updating $\xi_{C_i G}$ iteratively with,

$$\xi_{C_i G}^n \leftarrow \xi_{C_i G}^{n-1} - \alpha \boldsymbol{J}^n \tag{32}$$

For the surround-view system, the optimization objective of all cameras can be expressed as,

$$\xi_{C_i G}^* = \underset{\xi_{C_i G}^*}{\arg\min} \sum_{i=1}^{4} \sum_{j \in \Omega(i)} \sum_{\boldsymbol{p} \in \mathcal{N}_{ij}^{C_i}} \varepsilon_{\boldsymbol{p}} \tag{33}$$

where j is the index of C_i's adjacent cameras, and $\mathcal{N}_{ij}^{C_i}$ is $\mathcal{N}_{ij}$'s projection on C_i. We can optimize all camera poses together by minimizing the photometric error of the whole surround-view system.

Obviously, no dimension of the Jacobian $\boldsymbol{J}$ in Ground-Camera Model is constantly equal to zero. This means that the Ground-Camera Model can theoretically correct the rotation and translation shift of the camera poses in any directions. In other words, the Ground-Camera Model is superior to the Ground Model in performance. Therefore, we adopt the Ground-Camera Model to solve the camera pose optimization problem and the surround-view image generation.

4 EXPERIMENTAL RESULTS

4.1 Implementation Details

We evaluate our approach on a calibrated surround-view system. The system consists of four cameras mounted in the four directions of the vehicle, namely F, L, B, R (short for Front, Left, Back and Right). The initial camera poses of the system are ξ_F, ξ_L, ξ_B, ξ_R, respectively. Based on the models described in the previous section, we design and implement an automated algorithm, whose pseudo-code is as following.

Algorithm 1 Camera Pose Correction

```
Input: $\xi_C, \boldsymbol{I}_C \quad C \in \{F, L, B, R\}$
Output: $\xi_C^*$
Function
    $\boldsymbol{I}_{GC} \leftarrow P(\boldsymbol{I}_C, \xi_C, \boldsymbol{K}_C)$
    $\boldsymbol{J} \leftarrow G(\boldsymbol{I}_1, \boldsymbol{I}_2, \xi_1, \xi_2)$

Initialize $\boldsymbol{I}_C, \xi_C, \boldsymbol{K}_C$
for camera $C_i$ in $\{F, L, B, R\}$ do
    $\boldsymbol{I}_{GC_i} \leftarrow P(\boldsymbol{I}_{C_i}, \xi_{C_i}, \boldsymbol{K}_{C_i})$
end for
while $iter < iter\_max$ and $\delta > threshold$ do
    for camera $C_i$ in $\{F, L, B, R\}$ do
        for $C_j$ adjacent to $C_i$ do
            $J_{C_i} \leftarrow G(\boldsymbol{I}_{GC_i}, \boldsymbol{I}_{GC_j}, \xi_{C_i}, \xi_{C_j})$
            $\xi_{C_i} \leftarrow \xi_{C_i} - \alpha J_{C_i}$
            $\alpha \leftarrow decay\_rate * \alpha$
        end for
    end for
    $iter \leftarrow iter + 1$
end while
$\xi_C^* \leftarrow \xi_C$
return $\xi_C^*$
```

P is the function of projecting a camera image onto the ground and G represents our proposed models, namely the Ground Model or the Ground-Camera Model. The loop will quit after $iter\_max$ iterations or when the normalized average photometric error of all the overlap regions in the surround-view image δ is less than the $threshold$. The rate factor α varies in different implementations and different parameter settings. The $decay\_rate$ is used to reduce vibration in optimization.

δ here is the average photometric error of the ROIs in the surround-view image. In order to reduce the influence of different exposure level of different cameras, the photometric error is calculated as following:

$$\delta = \frac{1}{N} \sum_{i=1}^{4} \sum_{j \in \Omega(i)} \sum_{\boldsymbol{p}_G \in \mathcal{N}_{ij}} (\boldsymbol{I}_{GC_i}(\boldsymbol{p}_G) - \gamma_{ij} \boldsymbol{I}_{GC_j}(\boldsymbol{p}_G)) \tag{34}$$

where N is the number of the pixels in ROI, and

$$\gamma_{ij} = \frac{\sum_{\boldsymbol{p}_G \in \mathcal{N}_{ij}} \boldsymbol{I}_{GC_i}(\boldsymbol{p}_G)}{\sum_{\boldsymbol{p}_G \in \mathcal{N}_{ij}} \boldsymbol{I}_{GC_j}(\boldsymbol{p}_G)} \tag{35}$$

In our implementation, $threshold$ is set to 0.10, $iter\_max$ is 30, α is $5.0E - 10$ and $decay\_rate$ is 0.95.

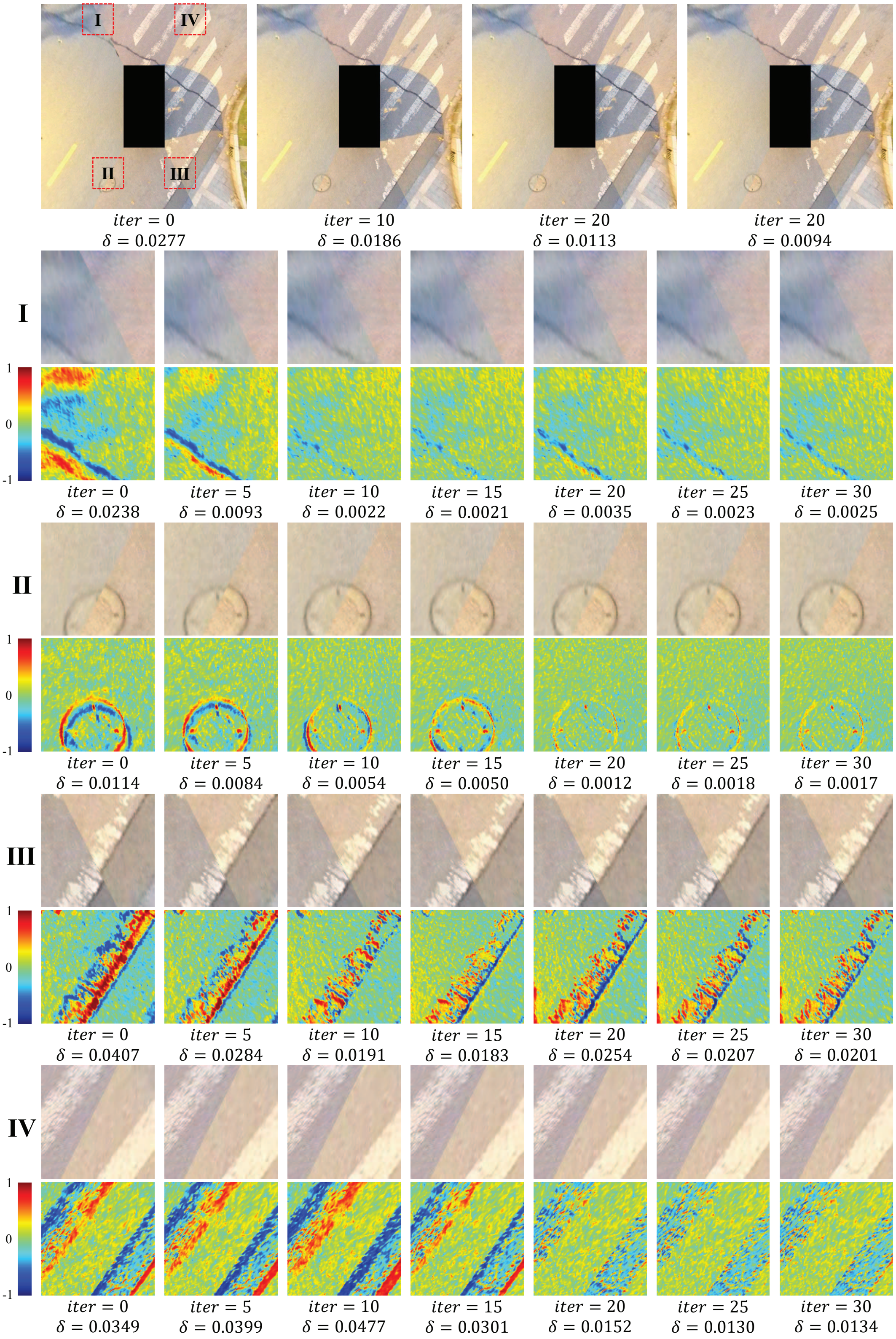


Figure 2: The optimization process of our algorithm. The first row is the overall effect of the surround-view image after 0, 10, 20, and 30 iterations. The following eight rows are the magnified images and the corresponding photometric errors maps of four ROIs (region of interest) marked in the first surround-view image as the Roman numerals I, II, III, IV. δ under each image is the average photometric error of the whole image, which quantitatively illustrates the reduction of stitching misalignment in the surround-view image.

4.2 optimization procedure

To evaluate our algorithm, we firstly apply random disturbance to the initial camera poses to simulate camera poses' change in the real world. Then we use our proposed algorithm to iteratively optimize the camera poses, and generate the surround-view image of different iterative stages, as shown in Fig. 2.

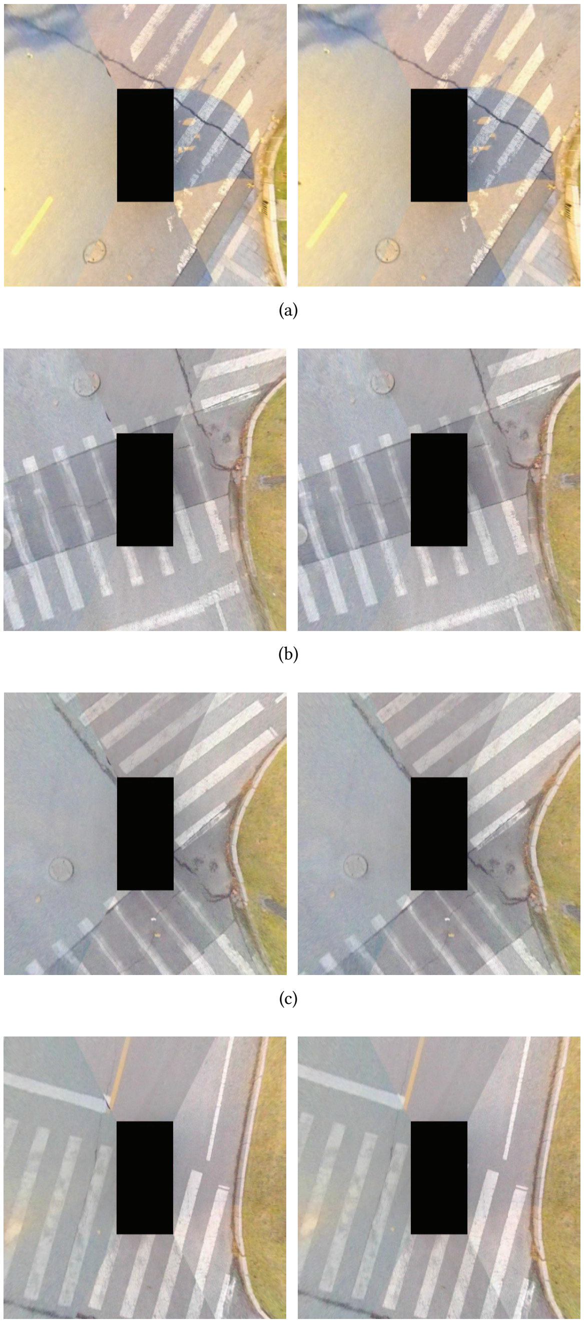
The first row of Fig. 2 shows the overall effect of the surround-view images after 0, 10, 20, and 30 iterations. It can be seen that, within the iteration process, the geometric misalignment of the surround-view image is decreased, which reflects that the camera poses are gradually corrected. In addition, δ under each image is the average photometric error of the whole image, which quantitatively illustrates the reduction of stitching misalignment in the surround-view image. The following eight rows are the magnified images and the corresponding photometric error maps of four ROIs marked in the first surround-view image as the Roman numerals I, II, III, IV.

For each ROI, the first row are the local magnified images, which shows the gradually closing up of the texture around the stitching seam during the iteration process in detail. Correspondingly, the second row are the normalized photometric error maps of the adjacent cameras' images in the same region. As shown in the color bar on the left, the red tendency represents that the intensity of a pixel in one image is greater than the intensity of its corresponding pixel in another image, while the blue tendency is the opposite. It is obvious that even the stitching is misaligned at the beginning of the iteration, the photometric errors of weak texture areas like the pavement are not very large. On the contrary, the photometric errors are greater in the areas with strong texture, such as the road lanes and the zebra crossing. Thus these regions make major contributions to the camera pose correction. Although there are still some photometric errors in the final results, especially in ROI III, the stitching misalignment are essentially eliminated by our algorithm. Actually, these photometric errors are mainly caused by local texture inhomogeneity and ground surface reflection. The text under the images shows the average photometric error of each ROI at different iterations. Although some of them increase slightly in later iterations, the overall photometric errors of all the four ROIs decrease because of the global optimization. Take ROI II as an example to analyze, the manhole cover on the ground is obviously divided into two parts due to wrong initial camera poses at the beginning. At the same time, the red and blue circles in the photometric error map also reflect the stitching misalignment of the ground images. As the iteration proceeds, the red and blue circles gradually approach to each other and disappear, while the misalignment of the ground image is also reduced, which indicates that the camera poses have been effectively corrected.

To sum up, after optimization with our algorithm, the geometric misalignment and the photometric error of the surround-view image are significantly reduced.

4.3 Quantitative Evaluation

We also validate our algorithm on other test samples, as shown in Fig. 3. Each row of the figure is a specific test sample, in which the image on the left is the surround-view image before the camera pose optimization, and the one on the right is the result after optimization. It can be seen that after being processed by our algorithm,



(a)

(b)

(c)

(d)

Figure 3: Comparisons of surround-view images before and after camera pose optimization. The images in the left column are the surround-view images before the camera pose optimization, and the ones in the right column are the results after optimization.

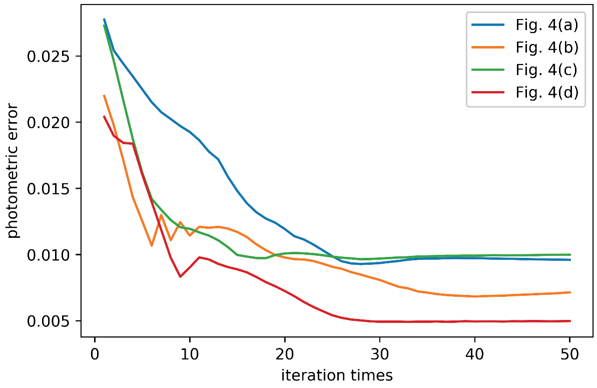


Figure 4: Photometric errors of the samples in Fig. 3 in different iteration times.

Table 1: Normalized photometric error of the samples in Fig. 3

Image	Iter 0	Iter 5	Iter 10	Iter 20	Iter 30
Fig. 3(a)	0.0277	0.0215	0.0192	0.0111	**0.0094**
Fig. 3(b)	0.0219	0.0106	0.0120	0.0096	**0.0078**
Fig. 3(c)	0.0272	0.0142	0.0116	0.0101	**0.0097**
Fig. 3(d)	0.0203	0.0139	0.0097	0.0068	**0.0049**

the misalignment of the surround-view image is greatly reduced, and the visual effect is improved. We also calculate objective indicators to measure the performance of our algorithm. Table 1 shows the normalized photometric errors of the four test samples in Fig. 3 under different iterations, which indicates that our method can optimize the photometric error to a low level, under 0.01. Fig. 4 is the line chart corresponding to Table 3. It shows that our algorithm could converge around 30 iterations.

It is worth mentioning that all the test samples include roads with the zebra crossing. That is because during the optimization process, each ROI needs a clear texture for calculating the image gradient in our algorithm . The texture of the common lane lines on the road is not sufficient enough, while the zebra crossing areas can meet our requirements well. Therefore, it is recommended to use our algorithm in similar scenarios. However, on the premise of ensuring sufficient ground texture, the orientation relationship between the vehicle and the ground texture is not important thus can be arbitrarily specified, which reduces the requirement of manual operation and guarantees the convenience of our method.

4.4 Long-term Performance

In online environment, the change of camera poses occur accidentally in common. That is to say, it is not necessary to optimize the camera poses all the time and the corrected camera poses by our method should be effective for a period of time. Therefore, we test the long-term performance of our approach on a video stream in this subsection. We firstly collected raw videos of the fish-eye cameras with our surround-view system. Then random disturbance was applied to the original camera poses to simulate camera poses' change in online environment. With the disturbed camera poses, we generated a surround-view video and calculated the photometric error of each frame in the video. After that, we used the proposed approach to correct the camera poses with the first frame of the video. Utilizing the corrected camera pose, we generated a new surround-view video and calculated its the photometric error as well. Fig. 5 shows the comparison of photometric errors of the videos before and after camera pose correction. As we can see in Fig. 5, the photometric errors of most frames after camera pose correction are significantly lower than the ones before, which illustrates that the camera poses corrected with the first frame are still valid in the following period.

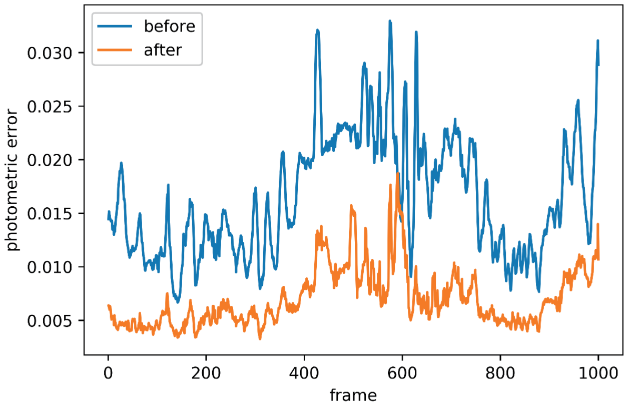


Figure 5: Photometric errors of a videos before and after camera pose optimization with the first frame.

5 CONCLUSION

In this paper, we propose a novel camera pose optimization method for the surround-view system. Our method consists of two models, Ground Model and Ground-Camera Model, both of which can correct the camera poses by minimizing the photometric error of the surround-view image. The Ground Model establishes the relationship between the photometric error and the camera pose, while the Ground-Camera model solves the DOF loss problem of Ground Model and improves the universality of our method. We also design and implement a highly automated algorithm based on the propsed model to meet the convenience and accuracy requirements in online environment. The experiments show that our method can effectively reduce the geometric misalignment and the photometric error of the surround-view image caused by camera poses' change. It is also improved that our method has a long-term effectiveness.

ACKNOWLEDGMENTS

This work was supported in part by the Natural Science Foundation of China under grant no. 61672380, in part by the Natural Science Foundation of Shanghai under grant no. 19ZR1461300, and in part by the Science and Technology Commission of Shanghai under grant no. 17DZ1100202.

REFERENCES

[1] Kyoungtaek Choi, Ho Jung, and Jae Suhr. 2018. Automatic calibration of an around view monitor system exploiting lane markings. *Sensors* 18, 9, Article 2956 (Sept. 2018). https://doi.org/10.3390/s18092956

[2] Learning driving models with a surround-view camera system and a route planner. 2018. Hecker, Simon and Dai, Dengxin and Van Gool, Luc. (2018). arXiv:arXiv:1803.10158

[3] Jakob Engel, Vladlen Koltun, and Daniel Cremers. 2018. Direct sparse odometry. *IEEE Trans. Pattern Anal. Mach. Intell.* 40, 3 (March 2018), 611–625. https://doi.org/10.1109/TPAMI.2017.2658577

[4] Jakob Engel, Thomas Schöps, and Daniel Cremers. 2014. LSD-SLAM: Large-scale direct monocular SLAM. In *13th European conference on computer vision (ECCV '14)*. Springer, Zurich, Switzerland, 834–849. https://doi.org/10.1007/978-3-319-10605-2_54

[5] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. 2014. SVO: Fast semi-direct monocular visual odometry. In *2014 IEEE international conference on robotics and automation (ICRA '14)*. IEEE, Hong Kong, China, 15–22. https://doi.org/10.1109/ICRA.2014.6906584

[6] Yi Gao, Chunyu Lin, Yao Zhao, Xin Wang, Shikui Wei, and Qi Huang. 2018. 3-D surround view for advanced driver assistance systems. *IEEE Trans. Intelligent Transportation Systems* 19, 1 (Jan. 2018), 320–328. https://doi.org/10.1109/TITS.2017.2750087

[7] Simon Hecker, Dengxin Dai, and Luc Van Gool. 2018. End-to-end learning of driving models with surround-view cameras and route planners. In *15th European Conference on Computer Vision (ECCV '18)*. Springer, Munich, Germany, 435–453. https://doi.org/10.1007/978-3-030-01234-2_27

[8] Adam Hedi and Sven Loncaric. 2012. A system for vehicle surround view. *IFAC Proceedings Volumes* 45, 22 (Sept. 2012), 120–125. https://doi.org/10.3182/20120905-3-HR-2030.00193

[9] Lionel Heng, Mathias Bürki, Gim Hee Lee, Paul Furgale, Roland Siegwart, and Marc Pollefeys. 2014. Infrastructure-based calibration of a multi-camera rig. In *2014 IEEE International Conference on Robotics and Automation (ICRA '14)*. IEEE, Hong Kong, China, 4912–4919. https://doi.org/10.1109/ICRA.2014.6907579

[10] Lionel Heng, Bo Li, and Marc Pollefeys. 2013. CamOdoCal: Automatic intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '13)*. IEEE, Tokyo, Japan, 1793–1800. https://doi.org/10.1109/IROS.2013.6696592

[11] Chang-Hoon Kum, Dong-Chan Cho, Moon-Soo Ra, and Whoi-Yul Kim. 2013. Lane detection system with around view monitoring for intelligent vehicle. In *2013 International SoC Design Conference (ISOCC '13)*. IEEE, Busan, South Korea, 215–218. https://doi.org/10.1109/ISOCC.2013.6864011

[12] Chien-Chuan Lin and Ming-Shi Wang. 2012. A vision based top-view transformation model for a vehicle parking assistant. *Sensors* 12, 4 (March 2012), 4431–4446. https://doi.org/10.3390/s120404431

[13] Yucheng Liu and Buyue Zhang. 2014. Photometric alignment for surround view camera system. In *2014 IEEE International Conference on Image Processing (ICIP '14)*. IEEE, Paris, France, 1827–1831. https://doi.org/10.1109/ICIP.2014.7025366

[14] Yu-Chih Liu, Kai-Ying Lin, and Yong-Sheng Chen. 2008. BirdÂąÂŕs-eye view vision system for vehicle surrounding monitoring. In *International Workshop on Robot Vision (RobVis '08)*. Springer, Auckland, New Zealand, 207–218. https://doi.org/10.1007/978-3-540-78157-8_16

[15] Koba Natroshvili and Kay-Ulrich Scholl. 2017. Automatic extrinsic calibration methods for surround view systems. In *2017 IEEE Intelligent Vehicles Symposium (IVS '17)*. IEEE, Los Angeles, CA, USA, 82–88. https://doi.org/10.1109/IVS.2017.7995702

[16] Frank Nielsen. 2005. Surround video: A multihead camera approach. *The Visual Computer* 21, 1-2 (Feb. 2005), 92–103. https://doi.org/10.1007/s00371-004-0273-z

[17] Kapje Sung, Joongryoul Lee, Junsik An, and Eugene Chang. 2012. Development of image synthesis algorithm with multi-camera. In *2012 IEEE 75th Vehicular Technology Conference (VTC '12)*. IEEE, Yokohama, Japan, 1–5. https://doi.org/10.1109/VETECS.2012.6240323

[18] Toshio Ueshiba and Fumiaki Tomita. 2003. Plane-based calibration algorithm for multi-camera systems via factorization of homography matrices. In *2003 IEEE 9th International Conference on Computer Vision (ICCV '03)*. IEEE, Nice, France, France, 966. https://doi.org/10.1109/ICCV.2003.1238453

[19] Buyue Zhang, Vikram Appia, Ibrahim Pekkucuksen, Yucheng Liu, Aziz Umit Batur, Pavan Shastry, Stanley Liu, Shiju Sivasankaran, and Kedar Chitnis. 2014. A surround view camera solution for embedded systems. In *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW '14)*. IEEE, Columbus, OH, USA, 662–667. https://doi.org/10.1109/CVPRW.2014.103

[20] Chao Zhang, Dong-xiao Li, and Ming Zhang. 2009. Multi-camera calibration based on iterative factorization of measurement matrix. In *2009 3rd International Conference on Multimedia and Ubiquitous Engineering (MUE '09)*. IEEE, Qingdao, China, 3–8. https://doi.org/10.1109/MUE.2009.11

[21] Lin Zhang, Junhao Huang, Xiyuan Li, and Lu Xiong. 2018. Vision-based parking-slot detection: A DCNN-based approach and a large-scale benchmark dataset. *IEEE Trans. Image Processing* 27, 11 (Nov. 2018), 5350–5364. https://doi.org/10.1109/TIP.2018.2857407

[22] Liuxin Zhang, Bin Li, and Yunde Jia. 2007. A practical calibration method for multiple cameras. In *4th International Conference on Image and Graphics (ICIG '07)*. IEEE, Sichuan, China, 45–50. https://doi.org/10.1109/ICIG.2007.59

[23] Zhengyou Zhang et al. 1999. Flexible camera calibration by viewing a plane from unknown orientations.. In *7th IEEE International Conference on Computer Vision (ICCV '99)*. IEEE, Kerkyra, Greece, 666–673. https://doi.org/10.1109/ICCV.1999.791289

[24] Kun Zhao, Uri Iurgel, Mirko Meuter, and Josef Pauli. 2014. An automatic online camera calibration system for vehicular applications. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC '14)*. IEEE, Qingdao, China, 1490–1492. https://doi.org/10.1109/ITSC.2014.6957643