OECS: TOWARDS ONLINE EXTRINSICS CORRECTION FOR THE SURROUND-VIEW SYSTEM

Tianjun Zhang[1], *Lin Zhang*[1,*], *Ying Shen*[1,*], *Yong Ma*[2], *Shengjie Zhao*[1], *Yicong Zhou*[3]

[1]School of Software Engineering, Tongji University, Shanghai, China
[2]School of Computer Information Engineering, Jiangxi Normal University, China
[3]Department of Computer and Information Science, University of Macau, Macau

ABSTRACT

A typical surround-view system consists of four fisheye cameras. By performing an offline calibration that determines both the intrinsics and extrinsics of the system, surround-view images can be synthesized at runtime. However, poses of calibrated cameras sometimes may change. In such a case, if cameras' extrinsics are not updated accordingly, observable geometric misalignment will appear in surround-views. Most existing solutions to this problem resort to re-calibration, which is quite cumbersome. Thus, how to correct cameras' extrinsics in an online manner without using re-calibration is still an open issue. In this paper, we attempt to propose a novel solution to this problem and the proposed solution is referred to as "*Online Extrinsics Correction for the Surround-view system*", OECS for short. We first design a Bi-Camera error model, measuring the photometric discrepancy between two corresponding pixels on images captured by two adjacent cameras. Then, by minimizing the system's overall Bi-Camera error, cameras' extrinsics can be optimized and the optimization is conducted within a sparse direct framework. The efficacy and efficiency of OECS are validated by experiments. Data and source code used in this work are publicly available at `https://z619850002.github.io/OECS_HomePage/`.

***Index Terms*—** Online extrinsics correction, sparse direct method, surround-view system, bird's-eye view

1. INTRODUCTION

The surround-view system (SVS), which provides a top-down view to drivers, is currently an integral part of modern vehicles. In addition to provide a broader view to the driver, the surround-view image generated by the system is also the basis of multiple computer vision tasks in autonomous driving, such as parking slot detection [1] and pedestrian detection [2]. With the current offline calibration methods, accurate extrinsics can be obtained, which can guarantee that the generated surround-view images are seamless.

*Corresponding authors: {cslinzhang, yingshen}@tongji.edu.cn.

After the surround-view system is calibrated, cameras are supposed to be fixed to keep their relative poses unchanged. However, due to some reasons (collisions, bumps, or tire pressure changes), cameras' poses may actually change afterwards. If we do not update the representation of extrinsics accordingly, there will be observable misalignment in the synthesized surround-views. Many automobile manufacturers now are looking for online methods to correct the system's extrinsics. Unfortunately, existing studies on cameras' extrinsics online correction are primarily designed for common multi-camera systems (such as the binocular system) and they cannot be readily adapted to the surround-view case. In this paper, we attempt to solve this problem and proposed a highly effective and efficient solution, namely "OECS".

2. RELATED WORK AND OUR CONTRIBUTIONS

The SVS is a special kind of multi-camera system. A multi-camera system is composed of several cameras and provides a wider view than the monocular one. For most multi-camera systems, apart from their intrinsics, their extrinsics are also need to be calibrated to get the relative poses among member cameras.

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 extrinsics correction schemes for the multi-camera systems roughly fall into two categories, online re-calibration and online optimization.

Online Re-calibration. In online re-calibration approaches, original offline calibration information will be abandoned and extrinsics are re-calibrated based on natural scene features without using auxiliary tools or special calibration sites. In [3], Hold *et al*. proposed a method of online extrinsics calibration for the binocular system. They adopted a conventional detector to detect the lane and sampled a series of feature points with a scanning line. By fast fourier transform (FFT) they measured the distance between lane points and finally, they solved the cameras' extrinsics based on lane points. In [4], Hansen *et al*.'s approach was based on a sequence of frames. They resorted to sparse feature matching to improve

978-1-7281-1331-9/20/$31.00

the algorithm's efficiency and used a sequence of frames to weaken the effect of noise. The method proposed in [5] relies on the localization results from a visual odometry, which is actually a complicated task itself.

Online re-calibration methods are simple and straightforward. However, prior information inherited from offline calibration will be totally abandoned in such methods and thus generally they cannot achieve high calibration accuracy.

Online Optimization. The key idea of online optimization approaches is to take the offline calibration results as prior knowledge, and then fine-tunes the cameras' extrinsics to obtain preciser parameters when cameras move. Ling and Shen [6] detect feature points and matched them between adjacent cameras. During optimization, the starting point is the offline calibration result and the epipolar error is minimized by nonlinear optimization. Knorr *et al.* [7] established a recursive optimization algorithm. Relative camera poses are corrected by the Extended Kalman Filter and the relationship between the multi-camera system and the ground is re-calculated by homography estimation. The solution proposed in [8] relies on two parallel lanes on the flat ground. The relative pose between the camera system's coordinate system and the world coordinate system is obtained through vanishing point derivation. It's worth mentioning that in [8] the authors considered the cameras system as a whole and thus poses among cameras were not optimized. In [9], Liu *et al.* studied the online extrinsics optimization problem for the surround-view system and their work is quite relevant to this paper. They proposed two models, the Ground model and the Ground-Camera model, and both of them can correct extrinsics by minimizing the system's photometric errors of overlapping areas.

Our motivations and contributions. How to correct cameras' extrinsics of a surround-view system in an online manner is an emerging problem in ADAS. Unfortunately, relevant studies in this area are quite rare. Most of the existing studies on online extrinsics correction focus on common multi-camera systems and they cannot be straightforwardly adapted to the surround-view case. To our knowledge, Liu *et al.*'s work is the only solution specially designed to cope with such a problem. But its correction accuracy, robustness, and computational complexity are still not satisfactory. To this end, our paper investigates this practical problem and the contributions are summarized as follows:

- We propose a new error model namely "Bi-Camera error", which can measure the photometric discrepancy between two corresponding pixels $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$ on two images, where $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$ are the projections from the same physical point $\boldsymbol{p}_G$ on the ground plane. Cameras' extrinsics are embedded in the projection relationships between $\boldsymbol{p}_G$ and $\boldsymbol{p}_i$ (or $\boldsymbol{p}_j$). Thus, by minimizing the Bi-Camera errors, optimal extrinsics can be worked out.

- Based on Bi-Camera error model, an online extrinsics correction algorithm for the surround-view system, namely (***O**nline **E**xtrinsic **C**orrection for the **S**urround-view system*) (OECS for short), is proposed. In OECS, cameras' optimal extrinsics are figured out by iteratively minimizing the system's overall Bi-Camera error. It needs to be noted that OECS follows a sparse direct framework, implying that it does not depend on visual feature points. Hence, OECS requires less on its working conditions and is quite robust.

- Within the sparse direct framework, a novel pixel selection strategy is proposed. Using such a selection strategy based on color matching and gradient screening, noise and unmatched objects between images captured by adjacent cameras can be eliminated effectively. Bi-Camera errors are then only computed on the selected positions. Such a pixel selection scheme can effectively improve OECS's speed and robustness.

3. METHOD

In this section, details of OECS are presented. The definition of Bi-Camera error model will be given first. Then, details of minimizing the system's Bi-Camera error are presented. Finally, we will introduce the pixel selection strategy adopted by OECS.

3.1. Bi-Camera error

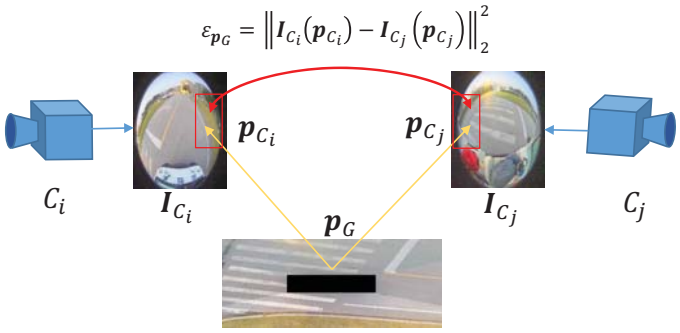

Fig. 1. Illustration of the Bi-Camera error.

Suppose that an SVS is composed of four fisheye cameras, C_1, C_2, C_3 and C_4. For a camera C_i, the mapping relationship between a point $\boldsymbol{p}_G$ on the surround-view image and a corresponding point $\boldsymbol{p}_{C_i}$ on the undistorted image is given by,

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

where $\boldsymbol{K}_{C_i}$ is the intrinsic matrix of C_i, $\boldsymbol{T}_{C_i G}$ is the pose of camera C_i with respect to the ground coordinate system, and Z_{C_i} is the depth of $\boldsymbol{p}_G$ in C_i's coordinate system. $\boldsymbol{K}_G$ is the

transformation matrix from the ground coordinate system to the surround-view coordinate system, which is given by,

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

where (d_{X_G}, d_{Y_G}) stands for the size of the physical area on the ground plane corresponding to each pixel of the surround-view image, and W and H are the width and height of the surround-view, respectively. Actually, $\boldsymbol{p}_G$ is the projection of a 3D point $\boldsymbol{P}_G = [X_G, Y_G, Z_G, 1]^T$ on the ground and they can be linked via $\boldsymbol{K}_G$ as,

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

So Eq. 1 can be reformulated as,

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

If $\boldsymbol{p}_G$ can be seen by both C_i and C_j, its two projections $\boldsymbol{p}_{C_i}$ and $\boldsymbol{p}_{C_j}$ on the undistorted images $\boldsymbol{I}_{C_i}$ and $\boldsymbol{I}_{C_j}$ captured by C_i and C_j can be obtained using Eq. 1. For $\boldsymbol{p}_G$, we define its Bi-Camera error term as,

$$\varepsilon_{\boldsymbol{p}_G} = \frac{1}{2} \| \boldsymbol{I}_{C_i}\left(\boldsymbol{p}_{C_i}\right) - \boldsymbol{I}_{C_j}\left(\boldsymbol{p}_{C_j}\right) \|_2^2 \tag{5}$$

Fig. 1 illustrates the Bi-Camera error term for a single point $\boldsymbol{p}_G$. For each of the qualified points chosen by the pixel selection strategy, which will be discussed in Sect. 3.3, such an error term can be built. By summing all the error terms together, we get the system's overall Bi-Camera error ε_B,

$$\varepsilon_B = \sum_{(i,j)\in\boldsymbol{\mathcal{A}}} \sum_{\boldsymbol{p}_G \in \boldsymbol{\mathcal{N}}_{ij}} \varepsilon_{p_G} \tag{6}$$

where $\boldsymbol{\mathcal{A}}$ is the set of all adjacent camera pairs and $\boldsymbol{\mathcal{N}}_{ij}$ is the set of qualified points in the common-view region of C_i and C_j. The four common-view regions are shown in Fig. 2.

Since $\boldsymbol{p}_{C_i}$ and $\boldsymbol{p}_{C_j}$ are imaging points of the same physical object, $\varepsilon_{\boldsymbol{p}_G}$ should be equal to zero ideally. Similarly, if the system's extrinsics are estimated precisely (and all the other conditions are ideal), ε_B should be zero. Thus, in OECS, ε_B is taken as the objective to be minimized to find the optimal extrinsics.

3.2. Optimization

For optimization, the camera pose $\boldsymbol{T}_{C_i G}$ is expressed in its Lie algebra form $\boldsymbol{\xi}^{\wedge}_{C_i G}$ [10] and thus Eq. 5 can be reformulated as,

$$\begin{aligned} \varepsilon_{\boldsymbol{p}_G} = \frac{1}{2} \| \boldsymbol{I}_{C_i} & \left(\frac{1}{Z_{C_i}} \boldsymbol{K}_{C_i} \exp\left(\boldsymbol{\xi}^{\wedge}_{C_i G}\right) \boldsymbol{K}_G^{-1} \boldsymbol{p}_G \right) \\ & - \boldsymbol{I}_{C_j} \left(\frac{1}{Z_{C_j}} \boldsymbol{K}_{C_j} \exp\left(\boldsymbol{\xi}^{\wedge}_{C_j G}\right) \boldsymbol{K}_G^{-1} \boldsymbol{p}_G \right) \|_2^2 \end{aligned} \tag{7}$$

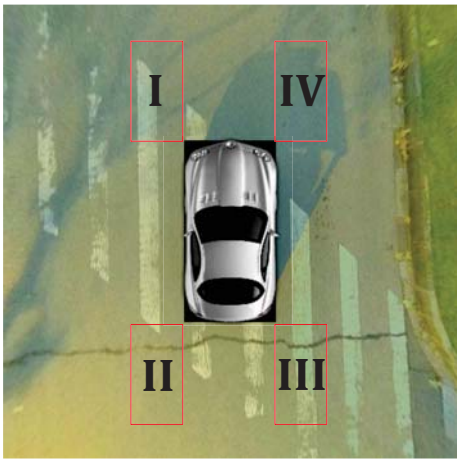

Fig. 2. The surround-view image and common-view regions. There are 4 common-view regions marked on the figure as the Roman numericals I, II, III and IV.

Thus, for camera C_i, its optimal pose $\boldsymbol{\xi}^*_{C_i G}$ is given by,

$$\boldsymbol{\xi}^*_{C_i G} = \arg\min_{\boldsymbol{\xi}_{C_i G}} \sum_{(i,j)\in\mathcal{A}} \sum_{\boldsymbol{p}_G \in \boldsymbol{\mathcal{N}_{ij}}} \varepsilon_{\boldsymbol{p}_G} \tag{8}$$

To optimize the objective function Eq. 8, the derivative relationship between $\varepsilon_{\boldsymbol{p}_G}$ and $\boldsymbol{\xi}_{C_i G}$ needs to be determined. The Jacobian of $\varepsilon_{\boldsymbol{p}_G}$ to $\boldsymbol{\xi}_{C_i G}$ can be expressed as,

$$\boldsymbol{J}_i = \frac{\partial \varepsilon_{\boldsymbol{p}_G}}{\partial \boldsymbol{\xi}^T_{GC_i}} \tag{9}$$

Eq. 9 can be decomposed to four parts with the chain rule,

$$\boldsymbol{J}_i = \frac{\partial \varepsilon_{\boldsymbol{p}_G}}{\partial \boldsymbol{I}_{C_i}} \cdot \frac{\partial \boldsymbol{I}_{C_i}}{\partial \boldsymbol{p}^T_{C_i}} \cdot \frac{\partial \boldsymbol{p}_{C_i}}{\partial \boldsymbol{P}^T_{C_i}} \cdot \frac{\partial \boldsymbol{P}_{C_i}}{\partial \boldsymbol{\xi}^T_{C_i G}} \tag{10}$$

Now we will discuss these four parts one by one:

(1) $\partial \varepsilon_{\boldsymbol{p}_G} / \partial \boldsymbol{I}_{C_i}$ is the derivative of the error $\varepsilon_{\boldsymbol{p}_G}$ to pixel intensities of image I_{C_i}. We use δ to denote it,

$$\delta = \frac{\partial \varepsilon_{\boldsymbol{p}_G}}{\partial \boldsymbol{I}_{C_i}} = \boldsymbol{I}_{C_i}(\boldsymbol{p}_{C_i}) - \boldsymbol{I}_{C_j}(\boldsymbol{p}_{C_j}) \tag{11}$$

Obviously, this term is the intensity value difference between two points on the undistorted fisheye images corresponding to $\boldsymbol{p}_G$. It's of large amount of computation for projecting $\boldsymbol{p}_G$ and calculating this difference one by one. Therefore, we use the difference of corresponding pixels on bird's-eye view images to substitute it,

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

where $\boldsymbol{I}_{GC_i}$ and $\boldsymbol{I}_{GC_j}$ are bird's-eye view images generated from $\boldsymbol{I}_{C_i}$ and $\boldsymbol{I}_{C_j}$, respectively.

(2) $\partial \boldsymbol{I}_{C_i} / \partial \boldsymbol{p}^T_{C_i}$ is the intensity gradient of image $\boldsymbol{I}_{C_i}$ at the pixel $\boldsymbol{p}_{C_i}$,

$$\frac{\partial \boldsymbol{I}_{C_i}}{\partial \boldsymbol{p}^T_{C_i}} = \begin{bmatrix} \frac{\partial \boldsymbol{I}_{C_i}}{\partial u_{C_i}} & \frac{\partial \boldsymbol{I}_{C_i}}{\partial v_{C_i}} \end{bmatrix} \triangleq \begin{bmatrix} \nabla \boldsymbol{I}^{u_G}_{C_i} & \nabla \boldsymbol{I}^{v_G}_{C_i} \end{bmatrix} \tag{13}$$

(3) $\partial \boldsymbol{p}_{C_i}/\partial \boldsymbol{P}_{C_i}^T$ is the derivative of a pixel's 2D coordinate to its 3D position in the camera coordinate. From the pin-hole camera model, we have

$$\frac{\partial \boldsymbol{p}_{C_i}}{\partial \boldsymbol{P}_{C_i}^T} = \begin{bmatrix} \frac{f_x^i}{Z_{C_i}} & 0 & -\frac{f_x^i X_{C_i}}{Z_{C_i}^2} \\ 0 & \frac{f_y^i}{Z_{C_i}} & -\frac{f_y^i Y_{C_i}}{Z_{C_i}^2} \end{bmatrix} \tag{14}$$

where f_x^i and f_y^i are focal lengths of C_i. X_{C_i}, Y_{C_i} and Z_{C_i} are the coordinate values of $\boldsymbol{P}_{C_i}$ in C_i's coordinate system, which can be obtained by,

$$\boldsymbol{P}_{C_i} = \boldsymbol{T}_{C_iG} \boldsymbol{K}_G^{-1} \boldsymbol{p}_G \tag{15}$$

(4) $\partial \boldsymbol{P}_{C_i}/\partial \boldsymbol{\xi}_{C_iG}^T$ is the derivative of the 3D point $\boldsymbol{P}_{C_i}$ to the camera pose $\boldsymbol{\xi}_{C_iG}$,

$$\frac{\partial \boldsymbol{P}_{C_i}}{\partial \boldsymbol{\xi}_{C_iG}^T} = \begin{bmatrix} \boldsymbol{I}_{3\times 3} & -\boldsymbol{P}_{C_i}^{\wedge} \end{bmatrix} \tag{16}$$

where $\boldsymbol{I}$ is a 3×3 identity matrix and $\boldsymbol{P}_{C_i}^{\wedge}$ is the 3×3 anti-symmetric matrix generated from $\boldsymbol{P}_{C_i}$.

By merging the four terms in Eqs. 13∼16, we get the final form of Jacobian $\boldsymbol{J}_i$,

$$\boldsymbol{J}_i = \delta \begin{bmatrix} \nabla \boldsymbol{I}_{C_i}^{uG} & \nabla \boldsymbol{I}_{C_i}^{vG} \end{bmatrix} \begin{bmatrix} \frac{f_x^i}{Z_{C_i}} & 0 \\ 0 & \frac{f_y^i}{Z_{C_i}} \\ -\frac{f_x^i X_{C_i}}{Z_{C_i}^2} & -\frac{f_y^i Y_{C_i}}{Z_{C_i}^2} \\ -\frac{f_x^i X_{C_i} Y_{C_i}}{Z_{C_i}^2} & -f_y^i - \frac{f_y^i Y_{C_i}^2}{Z_{C_i}^2} \\ f_x^i + \frac{f_x^i X_{C_i}^2}{Z_{C_i}^2} & \frac{f_y^i X_{C_i} Y_{C_i}}{Z_{C_i}^2} \\ -\frac{f_x^i Y_{C_i}}{Z_{C_i}} & \frac{f_y^i X_{C_i}}{Z_{C_i}} \end{bmatrix}^T \tag{17}$$

Once $\boldsymbol{J}_i$ is available, Eq. 8 can be iteratively optimized with proper optimization methods [11, 12, 13, 14].

3.3. Pixel selection

In consideration of the robustness and the computational speed, OECS follows a sparse direct framework [15]. In other words, only points that meet certain conditions are selected and involved in the Bi-Camera error computation.

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 $\boldsymbol{\mathcal{O}}_{ij}$. A set of pixels $\boldsymbol{\mathcal{N}}_{ij}$ will be selected out by the selection strategy and involved in optimization. Every pixel $\boldsymbol{p}$ in $\boldsymbol{\mathcal{N}}_{ij}$ must satisfy the following three criteria:

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

$$\boldsymbol{p} \in \boldsymbol{\mathcal{O}}_{ij} \tag{18}$$

- The color discrepancy between $\boldsymbol{I}_{GC_i}(\boldsymbol{p})$ and $\boldsymbol{I}_{GC_j}(\boldsymbol{p})$ is not allowed to be too large. Let $\boldsymbol{I}_{GC_i}^c$ and $\boldsymbol{I}_{GC_j}^c$ 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,

$$r_c(\boldsymbol{p}) = \frac{\boldsymbol{I}_{GC_i}^c(\boldsymbol{p})}{\boldsymbol{I}_{GC_j}^c(\boldsymbol{p})} \tag{19}$$

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

$$D_{color}(\boldsymbol{p}) = \sqrt{\frac{\sum_{c=1}^{n_c} \left(r_c(\boldsymbol{p}) - r_\mu(\boldsymbol{p})\right)^2}{n_c}} \tag{20}$$

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

$$D_{color}(\boldsymbol{p}) < D_{mean} - 2\sigma_d \tag{21}$$

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

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

$$G_i(\boldsymbol{p}) > G_{mean} + 2\sigma_g \tag{22}$$

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

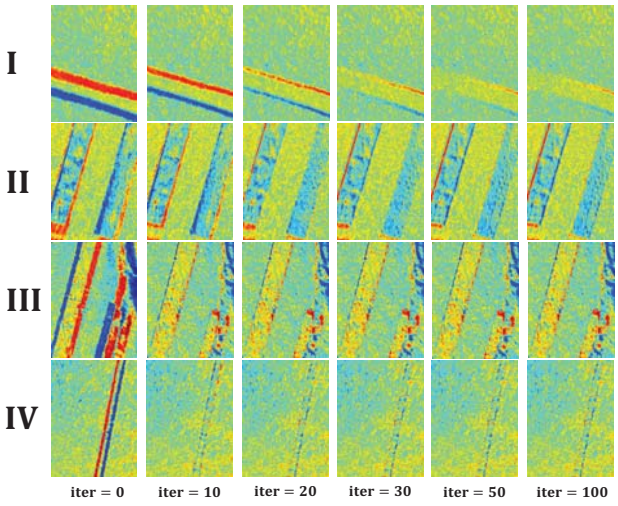

Fig. 3. The photometric error maps of the surround-view image in different ROIs at various iterations. From the first row to the last row, they are photometric error maps of the ROI I, II, III, and IV marked in Fig. 2, respectively. For each column, it shows error maps obtained after a particular number of optimization iterations.

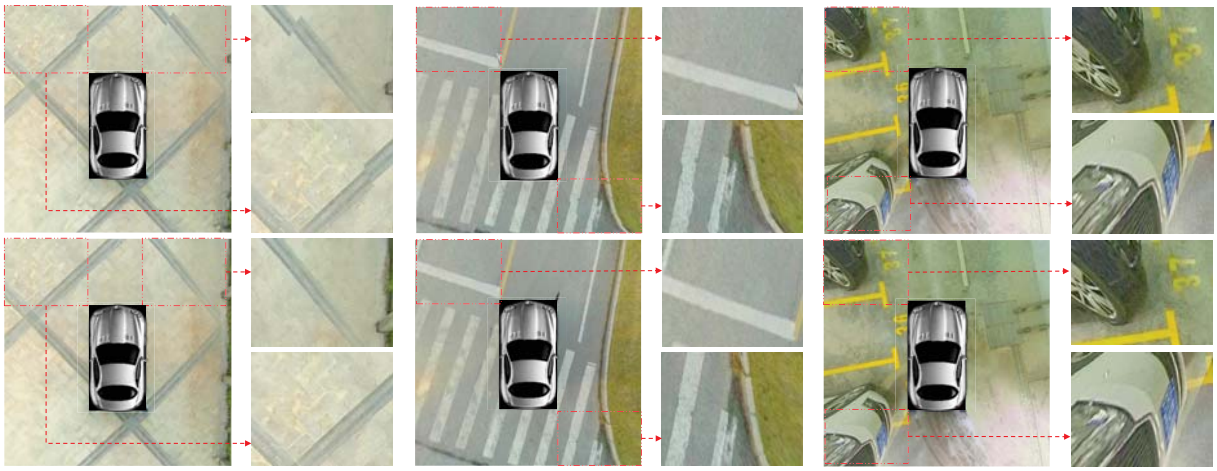

Fig. 4. Comparison of the surround-view images before and after extrinsics correction by OECS in various environments. For each pair, the upper left image is generated with disturbed extrinsics while the lower left one is the result after optimization. Enlarged local areas are shown on the right.

4. EXPERIMENTAL RESULTS

4.1. Experiment setup

To validate the performance of OECS, we performed experiments on an electric car equipped with an SVS. OECS was implemented with standard C++ and tested on a general laptop with an Intel (R) Core (TM) i5-7300HQ CPU.

4.2. Qualitative evaluation

Traits of Methods. As we have reviewed in Sect. 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 1 we compare them from three aspects: 1) Does it reuse the 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 only Liu *et al*.'s method and OECS are applicable to the surround-view system. One of the significant differences between Liu *et al*.'s method and OECS is that the former depends on dense pixels while the latter relies on sparse pixels, implying that the latter one will have the potential to be more robust and more efficient.

Table 1. Qualitative comparison with related methods

method	prior	SVS	feature type
Hold *et al*. [3]	×	×	ground lane
Hansen *et al*. [4]	×	×	feature point
Schneider *et al*. [5]	×	×	odometry
Ling and Shen [6]	√	×	feature point
Knorr *et al*. [7]	√	×	feature point
Edevschi *et al*. [8]	√	×	feature point
Liu *et al*. [9]	√	√	dense pixels
OECS	√	√	sparse pixels

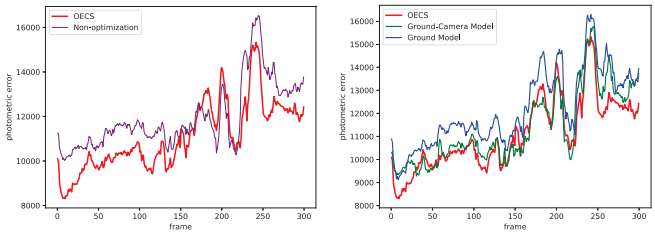

Fig. 5. Photometric errors of frames in SVS videos synthesized from the same four fisheye videos using different extrinsics.

Correction on A Single Image. To qualitatively validate OECS, we chose a sample and observed how the optimization process evolved. Fig. 3 shows the evolutions of Bi-Camera error maps in the common-view regions of adjacent cameras during optimization. It can be seen that the geometric misalignment existing in neighboring bird's-eye-view images was gradually eliminated, which qualitatively demonstrates the effectiveness of OECS.

Robustness. We tested OECS under a variety of different environmental conditions. The results show that, in most cases, OECS can accurately correct the camera system's extrinsics. It implies that OCES has lower requirements for external environments, and thus has good usability and strong robustness. Three typical examples are shown in Fig. 4.

4.3. Quantitative evaluation

To the best of our knowledge, Liu et al.'s work is the only publicly available solution to solve the problem of online extrinsics correction for the surround-view systems. Thus, for quantitative evaluation, we compared OECS with the two models, the Ground Model and the Ground-Camera Model,

proposed in Liu et al.'s work.

Minimizing Photometric Error. In this experiment, we first collected 100 groups of images from the SVS. For all groups, cameras' poses were changed slightly from the state of initial offline calibration. For each group, we then tried to optimize the system's extrinsics using compared methods. For each examined approach, its average photometric errors over 100 groups at sampled iterations are summarized in Table 2. Obviously, OECS performs best.

Table 2. Comparison of photometric errors by examined approaches

Method	iter = 0	iter = 10	iter = 20	iter = 50
Ground Model	10762.06	10748.62	10748.78	10748.49
Ground-Camera Model	10762.06	10731.35	10730.60	10671.34
OECS	10762.06	10552.65	10466.46	10345.96

Table 3. Average photometric errors of a video with different extrinsics

Ground Model	G-C Model	OECS	Non-optimization
12171.24	11382.36	11115.61	11976.49

Long-term Performance. In this experiment, we used different methods to correct the extrinsics by one frame and then recorded the photometric errors in the following 350 frames. The results are summarized in Fig. 5 and Table 3. From the results, it can be found that both the Ground-Camera Model and OECS can effectively correct the system's extrinsics, making the photometric errors greatly reduced, and OECS can achieve more accurate correction results. By contrast, the Ground Model performs much poorer.

Time Cost. Both OECS and the Ground Model takes only about 2s to finish the correction, and their speed is tenfold to the Ground-Camera Model. However, it needs to be noted that for the task of online extrinsics correction, the Ground Model is much inferior to OECS.

5. CONCLUSIONS

In this paper, we studied a practical problem, online correction of cameras' extrinsics for the surround-view system, emerging from the field of ADAS, and proposed a solution namely OECS. With OECS, by minimizing the system's overall Bi-Camera error, cameras' extrinsics can be optimized. One eminent feature of the proposed solution is that it fully exploits the prior information inherited from the offline calibration. Experimental results have corroborated OECS's superiority over the state-of-the-art competitors in this area.

6. ACKNOWLEDGMENT SECTION

This research was funded in part by the National Natural Science Foundation of China under grants 61672380, 61973235, 61972285, and 61936014 and in part by the Natural Science Foundation of Shanghai under grant 19ZR1461300.

7. REFERENCES

[1] L. Zhang, J. Huang, X. Li, and L. Xiong, "Vision-based parking-slot detection: A dcnn-based approach and a large-scale benchmark dataset," *IEEE Trans. IP*, vol. 27, no. 11, pp. 5350–5364, 2018.

[2] M. Gressmann, G. Palm, and O. Löhlein, "Surround view pedestrian detection using heterogeneous classifier cascades," in *ITSC*, 2011, pp. 1317–1324.

[3] S. Hold, S. Gormer, A. Kummert, M. Meuter, and S. Muller-Schneiders, "A novel approach for the online initial calibration of extrinsic parameters for a car-mounted camera," in *ITSC*, 2009, pp. 1–6.

[4] P. Hansen, H. Alismail, P. Rander, and B. Browning, "Online continuous stereo extrinsic parameter estimation," in *CVPR*, 2012, pp. 1059–1066.

[5] S. Schneider, T. Luettel, and H. Wuensche, "Odometry-based online extrinsic sensor calibration," in *IROS*, 2013, pp. 1287–1292.

[6] Y. Ling and S. Shen, "High-precision online markerless stereo extrinsic calibration," *CoRR*, vol. abs/1903.10705, 2019.

[7] M. Knorr, W. Niehsen, and C. Stiller, "Online extrinsic multi-camera calibration using ground plane induced homographies," in *IV*, 2013, pp. 236–241.

[8] S. Nedevschi, C. Vancea, T. Marita, and T. Graf, "Online extrinsic parameters calibration for stereovision systems used in far-range detection vehicle applications," *ITSC*, vol. 8, no. 4, pp. 651–660, 2007.

[9] X. Liu, L. Zhang, Y. Shen, S. Zhang, and S. Zhao, "Online camera pose optimization for the surround-view system," in *ACM MM*, 2019, pp. 383–391.

[10] J. E. Dennis and R. B. Schnabel, "The lie algebra of visual perception," *Journal of Mathematical Psychology*, vol. 3, no. 1, pp. 65–98, 1966.

[11] R. Battiti, "First- and second-order methods for learning: Between steepest descent and newton's method," *Neural Computation*, vol. 4, no. 2, pp. 141–166, 1992.

[12] R. W. M. Wedderburn, "Quasi-likelihood functions, generalized linear models, and the gauss-newton method," *Biometrika*, vol. 61, no. 3, pp. 439–447, 1974.

[13] J. Jorge, "The levenberg-marquardt algorithm: Implementation and theory," *Springer Berlin Heidelberg*, vol. 630, pp. 105–116, 1978.

[14] J. E. Dennis and R. B. Schnabel, "Numerical methods for unconstrained optimization and nonlinear equations," *Prentice Hall, Inc.*, 1983.

[15] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Trans. PAMI*, vol. 40, no. 3, pp. 611–625, 2019.