ROECS: A Robust Semi-direct Pipeline Towards Online Extrinsics Correction of the Surround-view System

Tianjun Zhang
School of Software Engineering,
Tongji University
Shanghai, China
1911036@tongji.edu.cn

Nlong Zhao
Department of Computer Science,
University of Southern California
Los Angeles, USA
briannlongzhao@gmail.com

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

Xuan Shao
School of Software Engineering,
Tongji University
Shanghai, China
1810553@tongji.edu.cn

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

Yicong Zhou
Department of Computer and
Information Science,
University of Macau, China
yicongzhou@um.edu.mo

ABSTRACT

Generally, a surround-view system (SVS), which is an indispensable component of advanced driving assistant systems (ADAS), consists of four to six wide-angle fisheye cameras. As long as both intrinsics and extrinsics of all cameras have been calibrated, a top-down surround-view with the real scale can be synthesized at runtime from fisheye images captured by these cameras. However, when the vehicle is driving on the road, relative poses between cameras in the SVS may change from the initial calibrated states due to bumps or collisions. In case that extrinsics’ representations are not adjusted accordingly, on the surround-view, obvious geometric misalignment will appear. Currently, the researches on correcting the extrinsics of the SVS in an online manner are quite sporadic, and a mature and robust pipeline is still lacking. As an attempt to fill this research gap to some extent, in this work, we present a novel extrinsics correction pipeline designed specially for the SVS, namely ROECS (Robust Online Extrinsics Correction of the Surround-view system). Specifically, a “refined bi-camera error” model is firstly designed. Then, by minimizing the overall “bi-camera error” within a sparse and semi-direct framework, the SVS’s extrinsics can be iteratively optimized and become accurate eventually. Besides, an innovative three-step pixel selection strategy is also proposed. The superior robustness and the generalization capability of ROECS are validated by both quantitative and qualitative experimental results. To make the results reproducible, the collected data and the source code have been released at https://cslinzhang.github.io/ROECS/.

CCS CONCEPTS

• **Computing methodologies → Camera calibration**.

KEYWORDS

Surround-view system, Extrinsics correction, Sparse semi-direct framework, Pixel selection strategy

ACM Reference Format:
Tianjun Zhang, Nlong Zhao, Ying Shen, Xuan Shao, Lin Zhang, and Yicong Zhou. 2021. ROECS: A Robust Semi-direct Pipeline Towards Online Extrinsics Correction of the Surround-view System. In *Proceedings of the 29th ACM International Conference on Multimedia (MM ’21), October 20–24, 2021, Virtual Event, China.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3474085.3475461

1 INTRODUCTION

A surround-view system (SVS) usually consists of four to six wide-angle fisheye cameras. These cameras are mounted on the vehicle facing different directions, so as to realize a 360° perception of the surrounding environment. By calibrating the SVS’s intrinsics and extrinsics accurately, relative poses between cameras can be determined and then high-quality surround-views can be synthesized. The surround-view cannot only broaden the driver’s view to eliminate blind areas, but also be employed in parking-slot detection [9, 17, 26], autonomous parking [11, 18, 23, 25], pedestrian detection [8, 14] and other related driving assistance tasks.

After being extrinsically calibrated, cameras in the SVS should be fixed to keep extrinsics unchanged. However, collisions or bumps may destroy the initial spatial structure of the camera system. If the initial extrinsics are still used and not properly adjusted, in generated surround-views, there will be observable geometric misalignment. In such case, prompt correction of the SVS’s extrinsics in an online manner is of great significance for the driving safety. Unfortunately, thus far the research in this area is still in its infancy. Existing schemes in this field mainly have the following limitations.

(1) Most of existing online extrinsics correction methods are designed for common multi-camera systems like binocular cameras, so that such methods usually can’t be easily extended to make them applicable to the surround-view case due to the particularity in the structure of the SVS. Concretely, most of existing online extrinsics correction schemes extract feature points in the common-view regions of different cameras, and then match them to solve accurate extrinsics. However, in

*Corresponding author: Ying Shen and Lin Zhang

MM ’21, October 20–24, 2021, Virtual Event, China
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8651-7/21/10…$15.00
https://doi.org/10.1145/3474085.3475461

the SVS, common fields of views between adjacent cameras are much narrower and more distorted than those of common binocular cameras, resulting in difficulty in resolving high-quality feature pairs from these regions.

(2) Existing solutions which are feasible to the surround-view case mostly require relatively ideal environments. For example, the approaches proposed in [2, 4, 13, 21, 28] require that, on the ground, there must be two parallel lane-lines that can be clearly detected. Thus, they usually have noticeable limitations in both the usability and the generalization capability. To the best of our knowledge, on the premise that the framework is applicable to the surround-view case, Liu *et al.*'s approach [20], Zhang *et al.*'s approach [27] and the one proposed in this paper are the only three that have quite relaxed requirements for the working conditions. Specifically, these three approaches all simply require a flat ground with relatively rich natural textures for them to work.

Currently, online extrinsics correction solutions are rarely embedded in the commercial products due to the technical immaturity. To fill such a research gap to some extent, in this paper, we propose an online extrinsics correction pipeline for the SVS, namely "ROECS" (Robust Online Extrinsics Correction of the Surround-view system). Our contributions are summarized as follows:

(1) A "refined bi-camera error" model is designed. The inaccuracy of SVS's extrinsics is mainly manifested in the geometric misalignment in bird's-eye common-view regions of adjacent cameras. Inspired by this observation, for a point $\boldsymbol{p}_G$ on the surround-view, we can construct a bi-camera error term which is actually the discrepancy of pixel values between two corresponding pixels $\boldsymbol{p}_{C_i}$ and $\boldsymbol{p}_{C_j}$ on fisheye images. The bi-camera error term can effectively measure the degree of the geometric misalignment on the surround-view at $\boldsymbol{p}_G$ without feature matching.

(2) Based on the "refined bi-camera error" model, we present the online extrinsics correction pipeline "ROECS", which follows a sparse and semi-direct framework. Each qualified point on the surround-view can be used to construct a bi-camera error term and by summing up all points' terms, the overall error of the system can be obtained. It's worth mentioning that we use multiple frames selected by our frame selection strategy and stored in a local window rather than a single frame to build the overall error, so as to improve the system's robustness. The frame selection strategy will be introduced in Sect. 4.2. By iteratively minimizing the system's overall error with any non-linear optimization scheme, the optimal camera poses can then be figured out.

(3) To further improve the speed and the accuracy of ROECS, we also propose a novel pixel selection strategy, which consists of three steps, common-view judgement, gradient screening and mismatched object elimination. Thanks to such a selection strategy, pixels with tiny gradient moduli and "mismatched" pixels, which will be demonstrated in detail in Sect. 4.1, can be effectively eliminated to reduce the computational cost and the effects of noise.

2 RELATED WORK

The SVS, which we are going to focus on, belongs to a particular type of multi-camera systems, which are sensors composed of at least two cameras. In this section, we will make a brief review on existing extrinsics correction schemes designed for multi-camera systems.

2.1 Manmade-feature based methods

Since manmade-feature based methods often strongly rely on some specific assumptions, relatively ideal environmental conditions are indispensable for them. One of the earliest work in this field is Collado *et al.*'s in [4]. To begin with, they extracted patterns from two parallel lane-lines with the Sobel operator and the Hough transform. Then camera poses could be estimated in an online manner with lane-lines' patterns. In [21], Nedevschi *et al.* proposed a solution based on vanishing point estimation. In Hold *et al.*'s work [13], a method of the online extrinsics calibration also using ground lane-lines was presented. They detected lane-lines and then sampled them with the scanning line to obtain a set of equidistant feature points, from which the extrinsics of the camera system were determined. In [28], Zhao *et al.* estimated multiple vanishing points rather than a single one to calibrate cameras' orientations. Although most of the aforementioned manmade-feature based frameworks perform satisfactorily in both the speed and the accuracy, none of them are designed for the SVS. Differently, in [2], Choi *et al.* proposed a pipeline which is specially designed for the surround-view case. They aligned lane-line markings across images captured by adjacent cameras and then the SVS can be extrinsically calibrated.

2.2 Natural-feature based methods

On account of the limited application scope of manmade-feature based approaches, more and more researchers focus on substituting manmade features in specific environments with natural features that could be extracted in common scenes. We refer to these solutions as "natural-feature based" ones. One of the earliest relevant researches along this direction can be traced back to Dang *et al.*'s work in [5]. They formulated a Gauss-Helmert model for the self-recalibration task. Their model consists of three different categories of constraint equations, the bundle-adjustment constraint, the epipolar constraint, and the trilinear constraint. Hansen *et al.*'s approach in [10] is a typical natural-feature based method. They matched feature points among different frames and then estimated extrinsics by bundle adjustment. To guarantee the efficiency of the scheme, features were sampled sparsely. Knorr *et al.* [16] established an optimization algorithm which seats on a recursive structure. In their approach, a sequence of frames were required for the pipeline to converge. In [19], taking the initial offline calibration result as the starting point, Ling and Shen minimized the epipolar error by non-linear optimization to find accurate camera poses, and the calibration accuracy was evaluated by the minimum eigenvalue of the covariance matrix. It is worth mentioning that this method takes all cameras as a whole and supposes that relative poses among them are fixed and will not change.

It needs to be noted that all of the methods reviewed above are designed for common multi-camera systems. Although the SVS also belongs to the family of multi-camera systems, unfortunately, these schemes usually are not directly applicable to it. As far as we know, the only two existing natural-feature based methods which are applicable to the SVS are Liu *et al.*'s method [20] and Zhang *et al.*'s [27]. They all studied the online extrinsics correction problem in depth and their works are quite relevant to ours in this paper. In [20], Liu *et al.* proposed two models, the "Ground Model" and the "Ground-Camera Model", and both of them can correct extrinsics by minimizing photometric errors. In [27], Zhang *et al.* designed a model to construct the least-square errors on the imaging planes of two adjacent cameras. However, the authors of these two schemes didn't take the interference of possible noise in various environments into consideration.

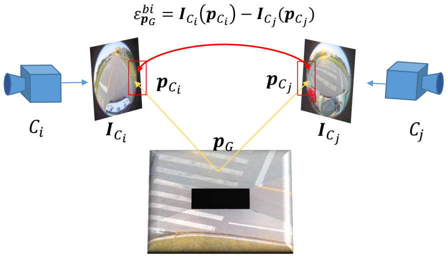

$\varepsilon^{bi}_{p_G} = I_{C_i}(p_{C_i}) - I_{C_j}(p_{C_j})$
C_i
I_{C_i}
p_{C_i}
p_{C_j}
I_{C_j}
C_j
p_G

Figure 1: Illustration of the basic structure of the bi-camera error model. For any qualified point selected by our pixel selection strategy, a corresponding bi-camera error term can be constructed.

3 REFINED BI-CAMERA ERROR MODEL

A bi-camera error term is employed to measure the photometric discrepancy of two corresponding points on original fisheye images captured by adjacent cameras. By minimizing the system's overall error, which is mainly summed up by the squares of all bi-camera error terms, accurate extrinsics of the SVS can be obtained. Since the final form of ROECS's objective function is a little bit complicated, in this section, we first analyze the basic form of the bi-camera error model, and other necessary refinements are then introduced incrementally. The basic structure of the bi-camera error model is illustrated in Fig. 1.

3.1 Basic Form

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 an observed point $\boldsymbol{p}_G$ on the surround-view coordinate system and a corresponding point $\boldsymbol{p}_{C_i}$ on the undistorted image $\boldsymbol{I}_{C_i}$ 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. The extrinsics of C_i, denoted by $\boldsymbol{T}_{C_i G}$, is the pose matrix of camera C_i with respect to the ground coordinate system. Z_{C_i} is the depth of $\boldsymbol{p}_G$ in C_i's coordinate system. $\boldsymbol{K}_G$ is the transform matrix from the ground coordinate system to the surround-view one. For more details about the imaging process of the SVS, please refer to the supplementary material.

If $\boldsymbol{p}_G$ can be seen by both C_i and C_j, its projections $\boldsymbol{p}_{C_i}$ and $\boldsymbol{p}_{C_j}$ on undistorted images $\boldsymbol{I}_{C_i}$ and $\boldsymbol{I}_{C_j}$ can then be obtained using Eq. 1. For $\boldsymbol{p}_G$, we define its corresponding bi-camera error term $\varepsilon^{bi}_{\boldsymbol{p}_G}$ as,

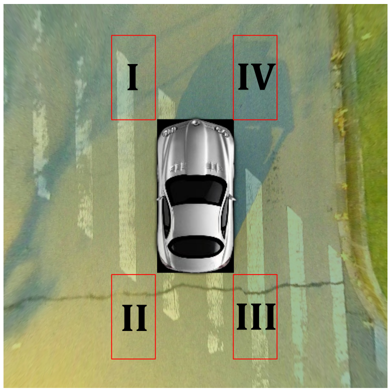


Figure 2: The surround-view image and common-view regions on the surround-view. There are four common-view regions marked on the figure as the Roman numericals I, II, III and IV.

$$\varepsilon^{bi}_{\boldsymbol{p}_G} = \boldsymbol{I}_{C_i}\left(\boldsymbol{p}_{C_i}\right) - \boldsymbol{I}_{C_j}\left(\boldsymbol{p}_{C_j}\right). \tag{2}$$

By combining Eq. 1 and Eq. 2, we have obtained the basic form of the bi-camera error term. To minimize the error under the non-linear optimization framework effectively, we introduce the "Lie algebra representation" [12] and the "inverse depth" [3] to reformulate the bi-camera error as,

$$\begin{aligned} \varepsilon^{bi}_{\boldsymbol{p}_G} = & \boldsymbol{I}_{C_i}\left(\lambda^{C_i}_{\boldsymbol{p}_G} \boldsymbol{K}_{C_i} \exp\left(\xi^{\wedge}_{C_i G}\right) \boldsymbol{K}_G^{-1} \boldsymbol{p}_G\right) \\ & - \boldsymbol{I}_{C_j}\left(\lambda^{C_j}_{\boldsymbol{p}_G} \boldsymbol{K}_{C_j} \exp\left(\xi^{\wedge}_{C_j G}\right) \boldsymbol{K}_G^{-1} \boldsymbol{p}_G\right) \end{aligned} \tag{3}$$

where $\xi_{C_i G}$ and $\xi_{C_j G}$ are Lie algebra forms of C_i's pose and C_j's, respectively. $\lambda^{C_i}_{\boldsymbol{p}_G}$ is the inverse depth of $\boldsymbol{p}_{C_i}$'s corresponding 3D point $\boldsymbol{P}_{C_i}$ in C_i's coordinate system and $\lambda^{C_j}_{\boldsymbol{p}_G}$ is that of $\boldsymbol{p}_{C_j}$'s. It's worth mentioning that $\lambda^{C_i}_{\boldsymbol{p}_G}$ and $\lambda^{C_j}_{\boldsymbol{p}_G}$ are not independent of each other and their relationship can be expressed as,

$$\lambda^{C_i}_{\boldsymbol{p}_G} = \frac{1}{[\boldsymbol{T}_{C_i C_j} (\lambda^{C_j}_{\boldsymbol{p}_G})^{-1} \boldsymbol{K}_{C_j}^{-1} \boldsymbol{p}_{C_j}]_3} \tag{4}$$

where the symbol $[*]_3$ stands for the coordinate value in the Z axis of the point, and $\boldsymbol{T}_{C_i C_j}$ is the relative pose between C_i and C_j.

3.2 Necessary Refinements

To guarantee the performance of the optimization, we made some refinements on the basic form of the bi-camera error model. Specifically, we introduce an exposure time factor and propose to compute the error on multiple pixels rather than a single one.

Exposure correction. Because of the inevitable discrepancies between different cameras' internal constructions in the SVS, for a same point $\boldsymbol{p}_G$ on the ground, corresponding imaging pixel values $\boldsymbol{I}_{C_i}\left(\boldsymbol{p}_{C_i}\right)$ and $\boldsymbol{I}_{C_j}\left(\boldsymbol{p}_{C_j}\right)$ won't be precisely the same, even if extrinsics are absolutely accurate. Actually, for an image of a physical object, except for the properties of the object itself, the corresponding pixel value is also determined by the exposure time, the vignette and the non-linear response function of the camera [6]. Based on our experience, the exposure time is the most important factor among them. To this end, we define a factor $\boldsymbol{\gamma}_{ij}$ as the ratio of exposure time of C_i and C_j,

$$\boldsymbol{\gamma}_{ij} = \frac{t_i}{t_j} \tag{5}$$

where t_i is C_i's exposure time and t_j is that of C_j's. Accordingly, the bi-camera error term (Eq. 3) can be further reformulated as,

$$\begin{aligned}\varepsilon^{bi}_{\boldsymbol{p}_G} =& \boldsymbol{I}_{C_i}\left(\lambda^{C_i}_{\boldsymbol{p}_G}\boldsymbol{K}_{C_i}\exp\left(\boldsymbol{\xi}^{\wedge}_{C_iG}\right)\boldsymbol{K}^{-1}_{G}\boldsymbol{p}_G\right)\\ &-\boldsymbol{\gamma}_{ij}\boldsymbol{I}_{C_j}\left(\lambda^{C_j}_{\boldsymbol{p}_G}\boldsymbol{K}_{C_j}\exp\left(\boldsymbol{\xi}^{\wedge}_{C_jG}\right)\boldsymbol{K}^{-1}_{G}\boldsymbol{p}_G\right).\end{aligned} \tag{6}$$

Actually, the exposure time of a camera can be obtained directly with the photometric calibration pipeline introduced in [7]. However, such a pipeline is quite cubersome. Instead, we offer a simple scheme for approximation, with which the factor $\boldsymbol{\gamma}_{ij}$ can be fitted as,

$$\gamma_{ij} = \frac{\sum_{\boldsymbol{p}_G \in \boldsymbol{O}_{ij}} \boldsymbol{I}_{GC_i}(\boldsymbol{p}_G)}{\sum_{\boldsymbol{p}_G \in \boldsymbol{O}_{ij}} \boldsymbol{I}_{GC_j}(\boldsymbol{p}_G)} \tag{7}$$

where $\boldsymbol{I}_{GC_i}$ and $\boldsymbol{I}_{GC_j}$ are bird's-eye view images of camera C_i and C_j, respectively. $\boldsymbol{O}_{ij}$ is the set of all pixels in the common-view region of C_i and C_j on bird's-eye views. In sum, there are four such regions on the surround-view as illustrated in Fig. 2.

Computing the error on multiple pixels. In most cases, the function of the pixel intensity of an image won't be absolutely smooth. Constructing the error term with a single pixel, the optimization may easily fall into the local optimum due to the non-smoothness of the image. Therefore, to improve the robustness, rather than computing the error with just one pixel pair $\boldsymbol{p}_{C_j}$ and $\boldsymbol{p}_{C_i}$, we construct the error term with $\boldsymbol{p}_{C_j}$ and nine points on I_{C_i} near $\boldsymbol{p}_{C_i}$,

$$\begin{aligned}\varepsilon^{bi}_{\boldsymbol{p}_G} =& \frac{1}{|\boldsymbol{\mathcal{P}}|}\sum_{\boldsymbol{p}_s \in \boldsymbol{\mathcal{P}}} \boldsymbol{I}_{C_i}\left(\lambda^{C_i}_{\boldsymbol{p}_G}\boldsymbol{K}_{C_i}\exp\left(\boldsymbol{\xi}^{\wedge}_{C_iG}\right)\boldsymbol{K}^{-1}_{G}\boldsymbol{p}_G + \boldsymbol{p}_s\right)\\ &-\boldsymbol{\gamma}_{ij}\boldsymbol{I}_{C_j}\left(\lambda^{C_j}_{\boldsymbol{p}_G}\boldsymbol{K}_{C_j}\exp\left(\boldsymbol{\xi}^{\wedge}_{C_jG}\right)\boldsymbol{K}^{-1}_{G}\boldsymbol{p}_G\right)\end{aligned} \tag{8}$$

where $\boldsymbol{\mathcal{P}}$ is a set that contains the relative pixel coordinates of all the utilized points to $\boldsymbol{p}_{C_i}$, and is defined as,

$$\boldsymbol{\mathcal{P}} = \{[i,j]^T \,|i,j = -2,0,2\}. \tag{9}$$

3.3 Objective Function and Jacobians

In this subsection, we mainly introduce the form of ROECS's objective function in the optimization. We consider C_i as the target camera and C_j as the reference one. During the optimization, to keep the optimal solution unique, only C_i's pose $\boldsymbol{\xi}_{C_iG}$ and $\boldsymbol{P}_{C_j}$'s inverse depth $\lambda^{C_j}_{\boldsymbol{p}_G}$ are optimized, while both $\boldsymbol{\xi}_{C_jG}$ and $\boldsymbol{p}_{C_j}$ are fixed. It's worth mentioning that $\boldsymbol{p}_G$ is not fixed, but changes with $\lambda^{C_j}_{\boldsymbol{p}_G}$.

In a single frame, for each qualified point chosen by the pixel selection strategy discussed in Sect. 4.1, a bi-camera error term can be built. To improve the robustness of the pipeline, we utilize pixels from multiple frames, which are selected by our frame selection strategy discussed in Sect. 4.2, rather than a single one during optimization. Besides, there is also a prior knowledge that most of qualified pixels should be from the ground. So for each point $\boldsymbol{p}_G$, we also introduce a prior error term $\varepsilon^{prior}_{\boldsymbol{p}_G}$ to the overall error to prevent the inverse depth $\lambda^{C_j}_{\boldsymbol{p}_G}$ from drastic changes. The prior error term $\varepsilon^{prior}_{\boldsymbol{p}_G}$ is defined as,

$$\varepsilon^{prior}_{\boldsymbol{p}_G} = \alpha(\lambda^{C_j}_{\boldsymbol{p}_G} - \lambda^{C_j*}_{\boldsymbol{p}_G}) \tag{10}$$

where α is an empirical value that controls how confident the point is on the ground. The prior inverse depth $\lambda^{C_j*}_{\boldsymbol{p}_G}$ is defined as,

$$\lambda^{C_j*}_{\boldsymbol{p}_G} = \frac{1}{\left[\boldsymbol{P}_{C_j}\right]_3} = \frac{1}{\left[\exp\left(\boldsymbol{\xi}^{\wedge}_{C_jG}\right)\boldsymbol{K}^{-1}_{G}\boldsymbol{P}_G\right]_3} \tag{11}$$

By summing up the squares of all bi-camera error terms and prior error terms, the final objective function of the system can be established and the optimal pose $\boldsymbol{\xi}^{*}_{C_iG}$ of camera C_i is given by,

$$\boldsymbol{\xi}^{*}_{C_iG} = \underset{\boldsymbol{\xi}_{C_iG},\boldsymbol{\lambda}^{C_j}}{\arg\min} \sum_{(i,j)\in\boldsymbol{\mathcal{A}}}\sum_{f\in\boldsymbol{\mathcal{F}}}\sum_{\boldsymbol{p}_G\in\boldsymbol{N}_{ij}} \rho_h((\varepsilon^{bi}_{\boldsymbol{p}_G})^2) + (\varepsilon^{prior}_{\boldsymbol{p}_G})^2 \tag{12}$$

where ρ_h is the Huber kernel function, $\boldsymbol{\mathcal{A}}$ is the set of all adjacent camera pairs, $\boldsymbol{\mathcal{F}}$ is the set of frames involved in the optimization, $\boldsymbol{N}_{ij}$ is the set of all qualified points in the common-view region of C_i and C_j, and $\boldsymbol{\lambda}^{C_j}$ is the inverse depth values to be optimized.

To minimize the objective function, the derivative relationships between the error terms and optimized variables, which consist of camera poses and the inverse depth of each point, need to be determined. Since the form of the prior error term is straightforward, we only offer the Jacobians of the bi-camera error term. The Jacobian $\boldsymbol{J}_p$ of the bi-camera error term $\varepsilon^{bi}_{\boldsymbol{p}_G}$ to C_i's pose $\boldsymbol{\xi}_{C_iG}$ is given by,

$$\boldsymbol{J}_p = \begin{bmatrix} \nabla\boldsymbol{I}^{u_{C_i}}_{C_i} & \nabla\boldsymbol{I}^{v_{C_i}}_{C_i} \end{bmatrix} \begin{bmatrix} \frac{f^i_x}{Z_{C_i}} & 0 & -\frac{f^i_x X_{C_i}}{Z^2_{C_i}} \\ 0 & \frac{f^i_y}{Z_{C_i}} & -\frac{f^i_y Y_{C_i}}{Z^2_{C_i}} \end{bmatrix} \begin{bmatrix} \boldsymbol{I}_{3\times3} & -\boldsymbol{P}^{\wedge}_{C_i} \end{bmatrix} \tag{13}$$

and the Jacobian $\boldsymbol{J}_d$ of the bi-camera error term $\varepsilon^{bi}_{\boldsymbol{p}_G}$ to point $\boldsymbol{p}_{C_j}$'s inverse depth $\lambda^{C_j}_{\boldsymbol{p}_G}$ is given by,

$$\boldsymbol{J}_d = -\frac{1}{(\lambda^{C_j}_{\boldsymbol{p}_G})} \begin{bmatrix} \nabla\boldsymbol{I}^{u_{C_i}}_{C_i} & \nabla\boldsymbol{I}^{v_{C_i}}_{C_i} \end{bmatrix} \begin{bmatrix} \frac{f^i_x}{Z_{C_i}} & 0 & -\frac{f^i_x X_{C_i}}{Z^2_{C_i}} \\ 0 & \frac{f^i_y}{Z_{C_i}} & -\frac{f^i_y Y_{C_i}}{Z^2_{C_i}} \end{bmatrix} \boldsymbol{P}_{C_i} \tag{14}$$

where $\nabla\boldsymbol{I}^{u_{C_i}}_{C_i}$ and $\nabla\boldsymbol{I}^{v_{C_i}}_{C_i}$ are intensity gradients of $\boldsymbol{I}_{C_i}$ at $\boldsymbol{p}_{C_i}$, f^i_x and f^i_y are focal lengths of C_i, X_{C_i}, Y_{C_i} and Z_{C_i} are coordinate values in three axes of $\boldsymbol{P}_{C_i}$ in C_i's coordinate system.

4 PIXEL SELECTION AND FRAME SELECTION STRATEGIES

4.1 Pixel Selection Strategy

To improve the speed and the robustness of the system, the pipeline we proposed follows a sparse semi-direct framework. That is to say, pixels which meet specific requirements rather than all pixels are chosen to establish the overall error. The selected pixels on the surround-view should meet three requirements:

a) The pixel must be in the field of view of at least two cameras.
b) The pixel should have enough intensity gradient modulus.
c) The pixel should be taken from the flat ground.

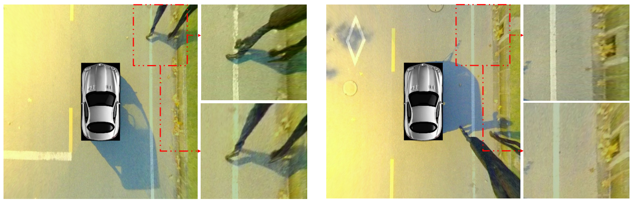


Figure 3: Typical examples of mismatched objects. The pedestrians in (a) and the curb in (b) are both typical mismatched objects, and they are marked in surround-views on the left. Enlarged areas on the right in each group from top to bottom are captured from the front-view and the right-view, respectively. It can be seen that there are obvious parallaxes between observations of mismatched objects on adjacent bird's-eye views.

With the requirements above, a novel pixel selection strategy is proposed. Take a pair of adjacent cameras C_i and C_j as an example. A set of pixels $\boldsymbol{N}_{ij}$ are selected out by the strategy. For any pixel $\boldsymbol{p}$ in $\boldsymbol{N}_{ij}$, it must pass a three-step check, involving common-view judgement, gradient screening, and mismatched object elimination.

Common-view Judgement. This is the first and simplest but most important rule. Common-view judgement implies that the pixel must be in the common-view region between adjacent cameras. To describe the criterion, the point $\boldsymbol{p}$ should be in the common-view region $\boldsymbol{O}_{ij}$ of C_i and C_j,

$$\boldsymbol{p} \in \boldsymbol{O}_{ij} \tag{15}$$

Gradient Screening. Gradient screening is an approach that selects pixels with high gradient moduli while abandons those with low ones. For the consideration of the discrepancy in both overall pixel intensities and contrast between different images, a single constant threshold won't be always appropriate. Thus, we use a dynamic threshold to select pixels. Concretely, this criterion is formulated as,

$$G_i(\boldsymbol{p}) > G_{mean} + \sigma_g \tag{16}$$

where G_{mean} is the mean intensity gradient modulus of all pixels in $\boldsymbol{O}_{ij}$ over the surround-view and σ_g is the associated standard deviation.

Mismatched Object Elimination. In reality, some objects with non-negligible heights, such as lawns, curbs or pedestrians, may appear on the surround-view, as illustrated in Fig. 3. For the convenience of statements, such objects are referred to as "mismatched objects". Pixels from mismatched objects are accordingly referred to as "mismatched pixels", and other pixels are referred to as "ground pixels". The existence of mismatched pixels has a destructive effect on the performance of the system since it breaks the premise of the establishment of the bi-camera error model. Therefore, we design such a "mismatched object elimination" approach to cull mismatched pixels. Such a step consists of two sub-steps, homography alignment and color matching.

The first sub-step is homography alignment. Since the vehicle is travelling on the flat ground, we can resolve a homography matrix to estimate the motion. For two consecutive frames $\boldsymbol{I}^t_{GC_i}$ and $\boldsymbol{I}^{t+1}_{GC_i}$, we extract ORB features [22] over them and then match these features using the Hamming distance. After that, a homography matrix $\boldsymbol{H}^{t+1}_t$ is estimated via the "4-point" method. For the consideration of the robustness, the estimation seats on a RANSAC framework. It's worth mentioning that, in this sub-step, ORB feature points are used, so our method is "semi-direct" rather than "direct".

The second sub-step is color matching. After the homography alignment, we warp $\boldsymbol{I}^{t+1}_{GC_i}$ by $\boldsymbol{H}^{t+1}_t$ to generate $\boldsymbol{I}^{t'}_{GC_i}$. For any point $\boldsymbol{p}$ on $\boldsymbol{I}^{t'}_{GC_i}$, there should be,

$$\boldsymbol{I}^{t'}_{GC_i}(\boldsymbol{p}) = \boldsymbol{I}^{t+1}_{GC_i}(\boldsymbol{H}^{t+1}_t \boldsymbol{p}). \tag{17}$$

Since color information is more discriminative than gray-scale information, the "mismatched object elimination" step is conducted based on color images rather than gray-scale ones. Ideally, for any ground pixel $\boldsymbol{p}$, it should satisfy,

$$\boldsymbol{I}^{t'}_{GC_i}(\boldsymbol{p}) = \boldsymbol{I}^{t}_{GC_i}(\boldsymbol{p}). \tag{18}$$

However, mismatched pixels are from physical objects that are not in the same plane with the ground. Thus, for a mismatched pixel $\boldsymbol{p}$, $\boldsymbol{H}^{t+1}_t$ can't provide a correct motion estimation, and there will be obvious differences between $\boldsymbol{I}^{t'}_{GC_i}(\boldsymbol{p})$ and $\boldsymbol{I}^{t}_{GC_i}(\boldsymbol{p})$. To measure this discrepancy in quantity, we first propose a coefficient, namely "color ratio", and then use the standard deviation of $\boldsymbol{p}$'s color ratios in different channels as the measurement of its corresponding color discrepancy. To simplify notations, we use $\boldsymbol{I}_t$ to represent $\boldsymbol{I}^{t}_{GC_i}$, and use $\boldsymbol{I}_{t'}$ to represent $\boldsymbol{I}^{t'}_{GC_i}$. Besides, let $\boldsymbol{I}^c_t$ and $\boldsymbol{I}^c_{t'}$ be the channel map of $\boldsymbol{I}_t$ and $\boldsymbol{I}_{t'}$ of channel c, respectively. The color ratio $r_c(\boldsymbol{p})$ of point $\boldsymbol{p}$ is defined as,

$$r_c(\boldsymbol{p}) = \frac{\boldsymbol{I}^c_{t'}(\boldsymbol{p})}{\boldsymbol{I}^c_t(\boldsymbol{p})}. \tag{19}$$

Then as aforementioned, we use the standard deviation of $\boldsymbol{p}$'s color ratios in different channels as the measurement of its color discrepancy. It's worth mentioning that to improve the robustness, the color discrepancy is computed in a local window $\boldsymbol{\mathcal{P}}_{\boldsymbol{p}}$ at $\boldsymbol{p}$,

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

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

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

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

4.2 Frame Selection Strategy

To keep the richness of textures and the uniformity of their distribution, we use multiple frames stored in a local window rather than a single one to build the overall error and then activate the optimization. The candidate frame that can be added to the local window must satisfy following three criteria:

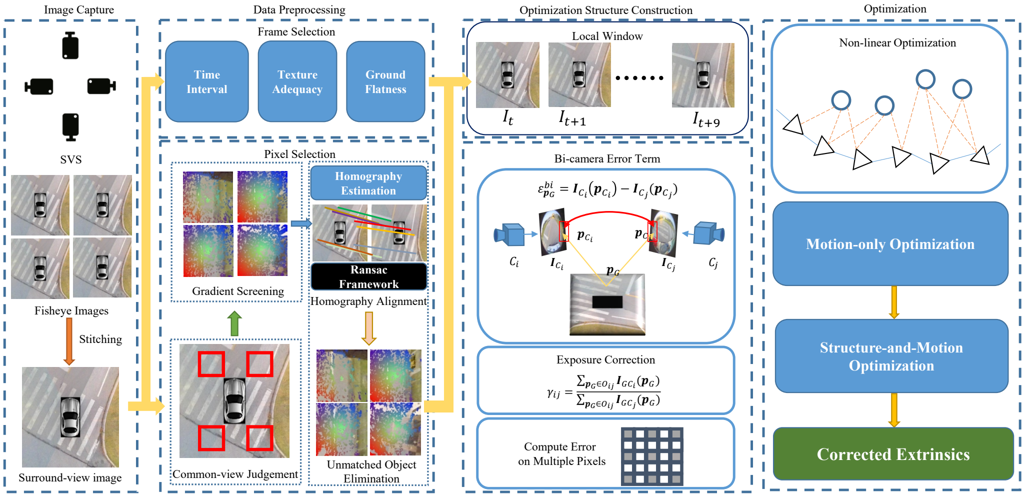


Figure 4: The overall pipeline of ROECS. Each qualified pixel p_G on the surround-view can be used to construct a bi-camera error term. By minimizing the overall error of the system, which is composed of error terms from all frames in the local window, with the non-linear optimization, extrinsics can be iteratively optimized.

a) In chronological order, there should be at least five frames between the candidate frame and the last one in the window.
b) There should be enough features in the candidate frame. In our implementations, 6000 qualified pixels are required under the 1080p resolution. For lower resolutions, such a threshold can also be turned down accordingly.
c) The ground should be relatively flat, and there cannot be too many mismatched objects in the field of the surround-view.

The first two constraints are straightforward, so here we just explain how to formulate the third constraint, which is “the ground should be relatively flat”. In fact, when the ground is flat, the vehicle is approximately moving parallel to the imaging plane of the SVS, so the estimated homography matrix should be close to the isometric matrix. Thus, we utilize a heuristic determinant based method to check the isometry of the transform matrix. The homography matrix $\boldsymbol{H}_t^{t+1}$ mentioned in Sect. 4.1 can be expressed as,

$$\boldsymbol{H}_t^{t+1} = \begin{bmatrix} \boldsymbol{A}_{2\times2} & \boldsymbol{t} \\ \boldsymbol{u} & 1 \end{bmatrix}. \tag{22}$$

If $\boldsymbol{H}_t^{t+1}$ is an isometric transform matrix, $\boldsymbol{A}_{2\times2}$ should be an orthonormal matrix. Thus, we use the determinant of $\boldsymbol{A}_{2\times2}$ to check the flatness of the ground. For a candidate frame, its corresponding matrix $\boldsymbol{H}_t^{t+1}$ should satisfy,

$$(Det\,(\boldsymbol{A}_{2\times2}) - 1)^2 < \theta \tag{23}$$

where θ is a threshold. In our implementations, θ is set to 0.2.

5 OVERALL PIPELINE OF ROECS

In Sects. 3 ∼ 4, we have presented details about our online extrinsics correction approach ROECS. To provide the reader with a clear and overall understanding of our work, more about its overall pipeline, which is illustrated in Fig. 4, are demonstrated in this section.

The pipeline of ROECS mainly consists of three phases. The first is the data preprocessing. While the vehicle is travelling on the road, the SVS will continuously capture images and synthesize surround-views. Fisheye images captured by different cameras at time t are noted as a group of images, G_t. When ROECS is activated, for each time the SVS acquires the image group G_t, the step a) and the step c) in the frame selection approach discussed in Sect. 4.2 are firstly performed. If G_t can meet these two steps’ requirements, we then select pixels on G_t by the pixel selection strategy introduced in Sect. 4.1 to find all qualified pixels. Finally, the step b) of the frame selection introduced in Sect. 4.2 is conducted to ensure that features are enough. After G_t passes the frame selection, both G_t and the related pixel selection results are stored in the local window. Once the number of image groups in the local window reaches the preset threshold n, operations in the subsequent two phases in ROECS’ pipeline will be executed.

The second phase of ROECS is the establishment of the optimization structure. Suppose I_t^i is one of the images in group G_t, which is stored in the local window. Each qualified pixel on I_t^i is projected onto adjacent cameras’ views to construct a bi-camera error term. For all frames in the local window, such bi-camera error terms and corresponding prior error ones can be built, and by summing up all terms, the overall error of the system is obtained.

Once the first two phases of ROECS have been performed, the optimization can then be conducted, which is also the last phase. As the overall error is of a least-square form, it can be minimized by any nonlinear optimization scheme, like the steepest descent [1], the Gauss-Newton method [24] and the Levenberg-Marquardt (LM) method [15]. To achieve a rather better performance, in our implementation, we adopted the LM scheme.

6 EXPERIMENTAL RESULTS

6.1 Experiment Setup

To validate the performance of our proposed pipeline ROECS, we performed experiments on an electric car equipped with an SVS, which consists of four Leopard LI-OV10640-490-GMSL cameras.

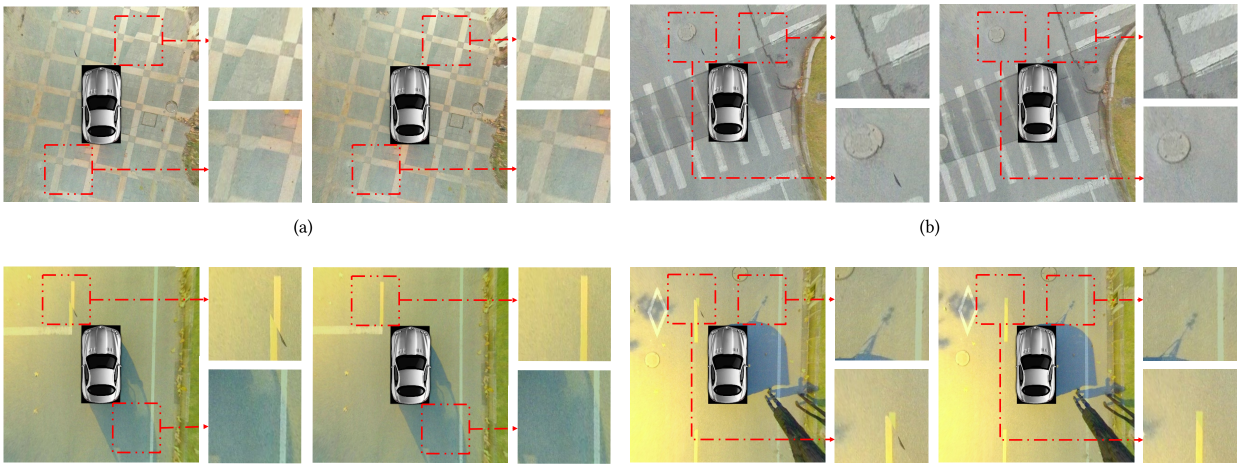

(a)
(b)
(c)
(d)

Figure 5: Comparison of surround-views before and after extrinsics correction by ROECS in various environments. From (a) to (d), four pairs of images belong to four groups of the collected data mentioned in Sect. 6.1, respectively. For each pair, the left surround-view is synthesized with inaccurate extrinsics while the right one is the result using extrinsics recovered by ROECS.

The resolution, the field-of-view, and the acquisition frequency of cameras are 1920 × 1080, 190 degrees and 30 FPS, respectively.

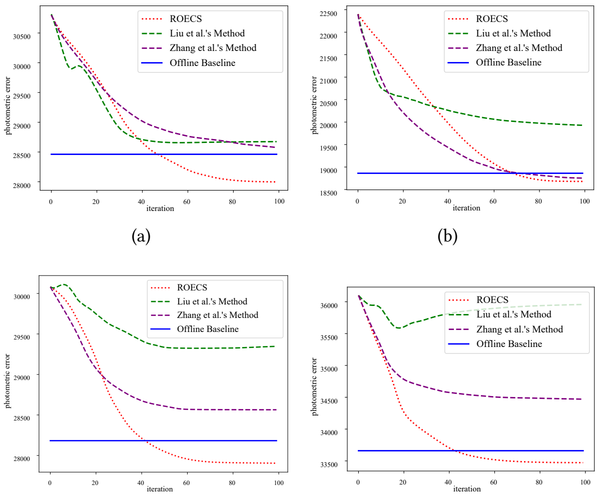
We collected four groups of surround-views (groups A, B, C and D), and for each group, there are one hundred frames. From A~D, each group of frames corresponds to a specific environmental condition, which are characterized by (A) with rich textures, (B) with relatively rich textures, (C) with sparse textures, and (D) with obvious mismatched objects, respectively. All experiments mentioned in this section were conducted based on these data. It should be noted that for all groups, cameras’ poses were changed moderately from the state of initial offline calibration.

6.2 Qualitative Experimental Results

Table 1: Qualitative comparison with related methods

Method	Method type	Prior	SVS	Feature type
Collado *et al.* [4]	Manmade-feature	×	×	Ground lanes
Nedevschi *et al.* [21]	Manmade-feature	√	×	Ground lanes
Hold *et al.* [13]	Manmade-feature	×	×	Ground lanes
Zhao *et al.* [28]	Manmade-feature	×	√	Ground lane
Choi *et al.* [2]	Manmade-feature	×	√	Ground lane
Dang *et al.* [5]	Natural-feature	√	×	Feature point
Hansen *et al.* [10]	Natural-feature	×	×	Feature point
Knorr *et al.* [16]	Natural-feature	√	×	Feature point
Ling and Shen [19]	Natural-feature	√	×	Feature point
Liu *et al.* [20]	Natural-feature	√	√	Dense pixels
Zhang *et al.* [27]	Natural-feature	√	√	Sparse pixels
ROECS	Natural-feature	√	√	Sparse pixels

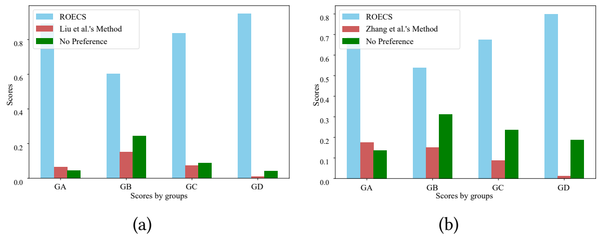
Traits of Methods. From those four aspects shown in Table 1, we compared all methods discussed in Sect. 2 and also our ROECS to demonstrate their characteristics more clearly. 1) Is this method manmade-feature based or natural-feature based? 2) Does it reuse the prior information from the offline calibration? 3) Without complex extensions, can it be applicable to the SVS? 4) What kind of features does it rely on? It can be seen that only Liu *et al.*’s method [20], Zhang *et al.*’s one [27] and ROECS can both correct extrinsics with natural features and be applicable to the SVS. Actually, compared with Liu *et al.*’s method [20] and Zhang *et al.*’s one [27], ROECS performs much better in terms of the robustness and the generalization capability.



(a)
(b)
(c)
(d)

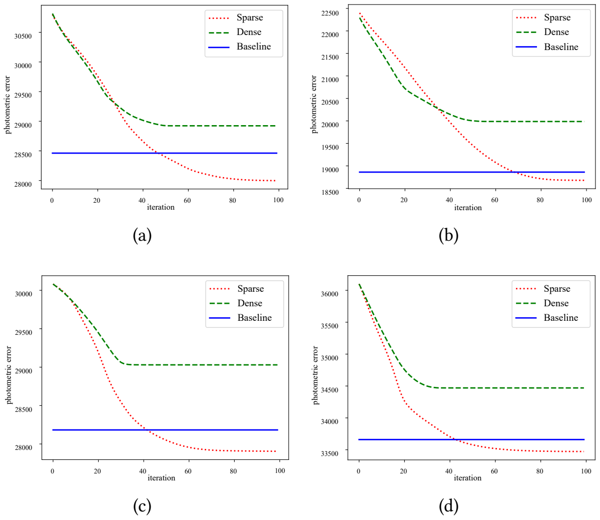
Figure 6: (a)~(d) are average photometric errors along with the correction evolvement of compared schemes over all surround-views corresponding to groups A~D, respectively.

Typical Samples. In order to qualitatively demonstrate the superiority of ROECS in terms of the correction effect, for each of the four groups of data aforementioned, we select a typical sample and show surround-views synthesized with both inaccurate extrinsics and corrected extrinsics recovered via ROECS, respectively, in Fig.



(a) (b)

Figure 7: Results of pairwise comparison user study. (a) shows the pairwise comparison result between ROECS and Liu *et al.*'s work [20], while (b) shows the result between ROECS and Zhang *et al.*'s method [27].



(a) (b)

(c) (d)

Figure 8: The photometric errors of both "dense approaches" and "sparse approaches" along with the optimization evolvement. (a) to (d) are the results achieved on groups A~D of the collected data, respectively.

5. It can be seen that the geometric misalignment in surround-views has been eliminated evidently by ROECS's correction, which qualitatively corroborates the effectiveness of ROECS.

6.3 Quantitative Experimental Results

As mentioned in Sect. 2, among all natural-feature based methods, only Liu *et al.*'s work [20], Zhang *et al.*'s work [27] and ours can be applicable to the surround-view case. Thus, in this subsection, we mainly quantitatively compare ROECS with its two rivals.

Effectiveness and Robustness. In this experiment, with each group of data we collected, we tried to optimize the system's extrinsics with Liu *et al.*'s scheme [20], Zhang *et al.*'s scheme [27] and ROECS, respectively. For each examined approach, the trends of errors along with the optimization evolvement are shown in Fig. 6. For reference, we also offered an "offline baseline", which was the average photometric error over all surround-views generated by undisturbed offline calibrated extrinsics. From the results reported, it can be found that, in most cases, ROECS performs much better than its competitors.

Table 2: Time cost analysis of ROECS

Sparsity	Resolution	Time cost	Pixel number
Dense	1080p	2.8236s/iter	216000/frame
Sparse	1080p	0.2331s/iter	13476/frame
Dense	900p	1.8638s/iter	150968/frame
Sparse	900p	0.1613s/iter	9073/frame
Dense	720p	1.1332s/iter	96480/frame
Sparse	720p	0.0942s/iter	5896/frame

Pairwise Comparison User Studies. Eight volunteers were invited to perform pairwise comparison among the correction results of Liu *et al.*'s scheme [20], Zhang *et al.*'s scheme [27] and ROECS. For each pairwise comparison, the subject had three options, "left better", "right better", or "no preference". The results of the user study are summarized in charts shown in Fig. 7. Each color bar is the average percentage of the image version favored over all eight subjects. From the results, it is obvious that the participants overwhelmingly selected the corrected results of our scheme. This user study demonstrates that in most cases, ROECS can effectively correct the geometric misalignment, and its performance is far beyond that of its counterparts.

Ablation Study of the Pixel Selection Strategy. Actually, without the pixel selection, the optimization in ROECS becomes a dense direct approach rather than the sparse one. In short, we call the optimization approach of our method with and without the pixel selection as the "sparse approach" and the "dense approach", respectively. Two factors were mainly considered in the evaluation of the pixel selection strategy, the speed and the accuracy. For the speed, we recorded time costs of both "sparse approaches" and "dense approaches" under different resolutions in Table 2. With respect to the accuracy, the photometric errors along with the optimization evolvement of both the "sparse approach" and the "dense approach" are illustrated in Fig. 8. From the experimental results, it can be corroborated that both the speed and the accuracy can be enhanced considerably by integrating our proposed pixel selection strategy.

7 CONCLUSION

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 novel solution namely ROECS. ROECS follows a sparse and semi-direct framework and fuses the prior information inherited from the offline calibration. With ROECS, by minimizing the system's overall error over multiple frames chosen by our frame selection strategy, cameras' extrinsics can be optimized effectively. One eminent feature of ROECS is that, thanks to our novel frame selection and pixel selection strategy, proper frames and pixels can be automatically selected to activate the optimization. Experimental results corroborated ROECS's superiority over the state-of-the-art competitors in this area.

8 ACKNOWLEDGEMENT

This work was supported in part by the National Natural Science Foundation of China under Grants 61973235, 61936014 and 61972285, in part by the Natural Science Foundation of Shanghai under Grant 19ZR1461300, and in part by the Shanghai Science and Technology Innovation Plan under Grant 20510760400.

REFERENCES

[1] Roberto Battiti. 1992. First- and Second-order Methods for Learning: Between Steepest Descent and Newton’s Method. *Neural Computation* 4, 2 (1992), 141–166. https://doi.org/10.1162/neco.1992.4.2.141

[2] Kyoungtaek Choi, Ho Gi Jung, and Jae Kyu Suhr. 2018. Automatic Calibration of an Around View Monitor System Exploiting Lane Markings. *Sensors* 18, 9 (2018), 2956:1–26. https://doi.org/10.3390/s18092956

[3] Javier Civera, Andrew J. Davison, and J. M. MartÍnez Montiel. 2008. Inverse Depth Parametrization for Monocular SLAM. *IEEE Trans. Robotics* 24, 5 (2008), 932–945. https://doi.org/10.1109/TRO.2008.2003276

[4] Juan M. Collado, Cristina Hilario, Arturo de la Escalera, and Jose M. Armingol. 2006. Self-calibration of an On-board Stereo-vision System for Driver Assistance Systems. In *IEEE Intelligent Vehicles Symposium (IVS’06)*. IEEE, Meguro–Ku, Japan, 156–162. https://doi.org/10.1109/IVS.2006.1689621

[5] Thao Dang and Christian Hoffmann. 2006. Tracking Camera Parameters of an Active Stereo Rig. In *Joint DAGM Symposium (DAGM’06)*. Springer, Berlin, Germany, 627–636. https://doi.org/10.1007/11861898_63

[6] Jakob Engel, Vladlen Koltun, and Daniel Cremers. 2018. Direct Sparse Odometry. *IEEE Trans. Pattern Analysis and Machine Intell.* 40, 3 (2018), 611–625. https://doi.org/10.1109/TPAMI.2017.2658577

[7] Jakob Engel, Vsenko Usenko, and Daniel Cremers. 2016. A Photometrically Calibrated Benchmark For Monocular Visual Odometry. *CoRR* abs/1607.02555 (2016). arXiv:1607.02555 http://arxiv.org/abs/1607.02555

[8] Markus Gressmann, Günther Palm, and Otto Löhlein. 2011. Surround View Pedestrian Detection Using Heterogeneous Classifier Cascades. In *International IEEE Conference on Intelligent Transportation Systems (ITSC’11)*. IEEE, Washington, DC, USA , 1317–1324. https://doi.org/10.1109/ITSC.2011.6082895

[9] Kazukuni Hamada, Zhencheng Hu, Mengyang Fan, and Hui Chen. 2015. Surround View based Parking Lot Detection and Tracking. In *IEEE Intelligent Vehicles Symposium (IVS’2015)*. IEEE, Seoul, Korea (South), 1106–1111. https://doi.org/10.1109/IVS.2015.7225832

[10] Peter Hansen, Hatem Alismail, Peter Rander, and Brett Browning. 2012. Online Continuous Stereo Extrinsic Parameter Estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR’12)*. IEEE, Providence, RI, USA, 1059–1066. https://doi.org/10.1109/CVPR.2012.6247784

[11] Simon Hecker, Dengxin Dai, and Luc Van Gool. 2018. End-to-end Learning of Driving Models with Surround-view Cameras and Route Planners. In *European Conference on Computer Vision (ECCV’18)*. Springer, Munich, Germany, 435–453. https://doi.org/10.1007/978-3-030-01234-2_27

[12] William C. Hoffman. 1966. The Lie Algebra of Visual Perception. *Journal of Mathematical Psychology* 3, 1 (1966), 65–98. https://doi.org/10.1016/0022-2496(66)90005-8

[13] Stephanie Hold, Steffen Görmer, Anton Kummert, Mirko Meuter, and Stefan Muller-Schneiders. 2009. A Novel Approach for the Online Initial Calibration of Extrinsic Parameters for a Car-mounted Camera. In *International IEEE Conference on Intelligent Transportation Systems (ITSC’09)*. IEEE, St. Louis, MO, USA, 420–425. https://doi.org/10.1109/ITSC.2009.5309853

[14] Cong Hou, Haizhou Ai, and Shihong Lao. 2007. Multiview Pedestrian Detection based on Vector Boosting. In *Asian Conference on Computer Vision (ACCV’07)*. Springer, Berlin, Heidelberg, 210–219. https://doi.org/10.1007/978-3-540-76386-4_19

[15] Moré Jorge J. 1978. The Levenberg-Marquardt algorithm: Implementation and theory. In *Numerical Analysis (Lecture Notes in Mathematics)*. Springer, Berlin, Heidelberg, 105–116. https://doi.org/10.1007/BFb0067700

[16] Moritz Knorr, Wolfgang Niehsen, and Christoph Stiller. 2013. Online Extrinsic Multi-camera Calibration Using Ground Plane Induced Homographies. In *IEEE Intelligent Vehicles Symposium (IVS’13)*. IEEE, Gold Coast, QLD, Australia, 236–241. https://doi.org/10.1109/IVS.2013.6629476

[17] Linshen Li, Lin Zhang, Xiyuan Li, Xiao Liu, Ying Shen, and Lu Xiong. 2017. Vision-based Parking-slot Detection: A Benchmark and a Learning-based Approach. In *IEEE International Conference on Multimedia and Expo (ICME’17)*. IEEE, Hong Kong, China, 649–654. https://doi.org/10.1109/ICME.2017.8019419

[18] Chien-Chuan Lin and Ming-Shi Wang. 2012. A Vision based Top-view Transformation Model for a Vehicle Parking Assistant. *Sensors* 12, 4 (2012), 4431–4446. https://doi.org/10.3390/s120404431

[19] Yonggen Ling and Shaojie Shen. 2016. High-precision Online Markerless Stereo Extrinsic Calibration. In *International Conference on Intelligent Robots and Systems (IROS’16)*. IEEE/RSJ, Daejeon, Korea (South), 1771–1778. https://doi.org/10.1109/IROS.2016.7759283

[20] Xiao Liu, Lin Zhang, Ying Shen, Shaoming Zhang, and Shengjie Zhao. 2019. Online Camera Pose Optimization for the Surround-view System. In *ACM International Conference on Multimedia (MM ’19)*. ACM, New York, the United States, 383–391. https://doi.org/10.1145/3343031.3350885

[21] Sergiu Nedevschi, Cristian Vancea, Tiberiu Marita, and Thorsten Graf. 2007. Online Extrinsic Parameters Calibration for Stereovision Systems Used in Far-range Detection Vehicle Applications. *IEEE Trans. Intell. Transportation Systems* 8, 4 (2007). https://doi.org/10.1109/TITS.2007.908576

[22] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. 2011. ORB: An Efficient Alternative to SIFT or SURF. *International Conference on Computer Vision (ICCV’11)* (2011), IEEE, Barcelona, Spain, 2564–2571. https://doi.org/10.1109/ICCV.2011.6126544

[23] Chunxiang Wang, Hengrun Zhang, Ming Yang, Xudong Wang, Lei Ye, and Chunzhao Guo. 2014. Automatic Parking based on a Bird’s Eye View Vision System. *Advances in Mechanical Engineering* 6 (2014), 847406:1–13. https://doi.org/10.1155/2014/847406

[24] R. W. M. Wedderburn. 1974. Quasi-Likelihood Functions, Generalized Linear Models, and the Gauss-Newton Method. *Biometrika* 61, 3 (1974), 439–447. https://doi.org/10.2307/2334725

[25] Jin Xu, Guang Chen, and Ming Xie. 2000. Vision-guided Automatic Parking for Smart Car. In *IEEE Intelligent Vehicles Symposium (IVS’00)*. IEEE, Dearborn, MI, USA, 725–730. https://doi.org/10.1109/IVS.2000.898435

[26] 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 (2018), 5350–5364. https://doi.org/10.1109/TIP.2018.2857407

[27] Tianjun Zhang, Lin Zhang, Ying Shen, Yong Ma, Shengjie Zhao, and Yicong Zhou. 2020. OECS: Towards Online Extrinsics Correction for The Surround-view System. In *IEEE International Conference on Multimedia and Expo (ICME’20)*. IEEE, London, UK, 1–6. https://doi.org/10.1109/ICME46284.2020.9102803

[28] Kun Zhao, Uri Iurgel, Mirko Meuter, and Josef Pauli. 2014. An Automatic Online Camera Calibration System for Vehicular Applications. In *IEEE Conference on Intelligent Transportation Systems (ITSC’14)*. IEEE, Qingdao, China, 1490–1492. https://doi.org/10.1109/ITSC.2014.6957643