IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 25, 2021, accepted October 12, 2021, date of publication October 18, 2021,
date of current version October 25, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3121276

Accurate Uncalibrated Rectification Using Feature Matching Re-Selection

TAI HUU-PHUONG TRAN, HUY-HUNG NGUYEN, AND JAE WOOK JEON, (Senior Member, IEEE)
Department of Electrical and Computer Engineering, Sungkyunkwan University, Suwon 440-746, South Korea

Corresponding author: Jae Wook Jeon (jwjeon@skku.edu)

This work was supported by the National Research Foundation of Korea (NRF) Grant by the Korean Government through Ministry of Science and ICT (MSIT) under Grant 2020R1A2C3011286.

ABSTRACT Rectification is a core programming process that involves multiple views. In this study, we focus on uncalibrated cases that neglect intrinsic and extrinsic camera information. Existing uncalibrated rectification methods use feature-matching techniques to form relationships between different views and use those features to estimate optimized homography matrices. However, outliers are inevitable in feature matching. Using these outliers in a rectification process produces vertical disparity errors and unwanted geometric distortion. To tackle the problem, we propose a novel method that can learn from the rectification results, re-select the matching pairs, and find superior solutions. The proposed method introduces a novel workflow for uncalibrated rectification that incorporates three cores: field of view (FoV) neutralization, rectification, and feature matching re-selection (FMR). While the FoV neutralization module handles FoV differences among views, a combination of the rectification module and FMR module results in the optimal homography matrices. The rectification module takes neutral correspondences and estimates the optimized rectified matrices. Applying the results from the rectification module, the FMR module optimizes the correspondences and return them to the rectification module. For the rectification module, we apply adaptive geometric constraints and our updated optimization strategy to secure satisfied vertical disparity errors while maintaining low distortion levels. Multidisciplinary experiments are performed to analyze the capabilities of the proposed method. We combine existing datasets with extra samples, including various outdoor environments, to gauge the performance better. Compared with existing methods, the proposed method produces fewer rectification errors and keeps the rectified images under low distortion rates.

INDEX TERMS Different fields of view, feature matching re-selection, low distortion, multiple views, uncalibrated rectification.

I. INTRODUCTION

Applications with multiple views have become popular in recent years for both consumer products and industrial activities. Multiple cameras provide information that cannot be generated by a single camera, such as depth information that can be computed through stereoscopic vision. By replicating human vision, stereo systems can recreate depth information from the real world using two cameras. Compared with light detection and ranging (LiDAR) sensors, which also provide depth information, stereo vision is more compact, less expensive, capable of working in hostile conditions such as rain or snow, and can operate over long periods due to low energy needs.

Stereo vision is commonly applied in several industrial contexts [1], [2] including stereo matching [4], robotics [3], and autonomous vehicles [5]–[7]. Rectification plays an important role in stereo vision, reducing the computational cost of the correspondence process, which is the first and crucial step. While image registration method [8] finds the most suitable stationary reference images to fit the target images, rectification involves a projective transformation of input images to parallel their epipole lines. The image rectification process is illustrated in Fig. 1. After being rectified, matching processes on the images are simplified from a two-dimensional window searching to one-dimensional line scanning.

The associate editor coordinating the review of this manuscript and approving it for publication was Hossein Rahmani.

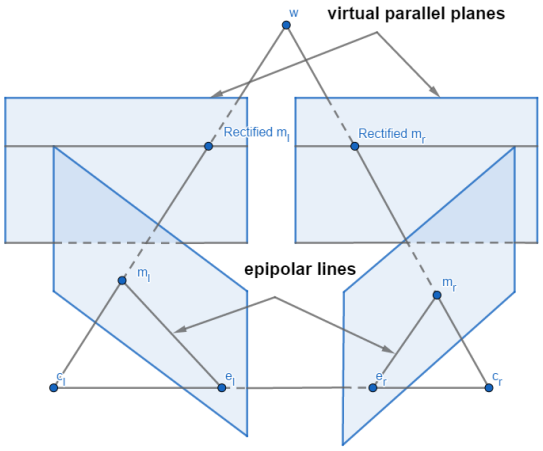

FIGURE 1. Image rectification.

In the calibration of cameras, computation of the homographies of rectification is straightforward. However, calibration can be is tedious. The pioneers of rectification research, Ayache and Francis [9] and Fusiello *et al.* [10], considered both the internal and external information of cameras. To avoid depending on intrinsic and extrinsic parameters, researchers use various restraints to control the number of possible solutions. Most such methods are based on epipolar geometry theory [11]. Hartley [12], [13] are among the first to explore uncalibrated rectification. Their method involves keeping one of two homographies close to a rigid transformation while optimizing the other. Loop and Zhang [14] approached from a different perspective, separating homographies into an affine transformation and a projective transformation. Unlike those who work with planar surfaces, Pollefeys *et al.* [15] reprojected images onto a cylindrical surface. This method is suitable for images in which epipoles are within or close to each other.

The results of rectification are homography matrices. By using homographies, multi-view applications perform warping procedures to transform an initial captured images into a rectified input. The only way to prevent unwanted distortion due to warping is to optimize the homography matrices. Several research teams used deep learning to decrease distortion rates of a single image [16], [17]. For stereo vision, the epipolar constraints need to be satisfied and low distortion values on rectified images are crucial. Mallon and Whelen [18] applied singular vector decomposition to a first-order approximation of an orthogonal-like transformation to reduce the effects of distortion. Without using explicit computation of fundamental matrices, Isgro and Trucco [20] proposed calculating homographies by minimizing disparities [12]. However, the disparity constraint can cause extreme distortion on rectified images. Wu and Yu [19] combined a previously proposed idea [20] with a shearing transform to reduce distortion. A rectification algorithm based on Quasi-Euclidean epipolar was proposed by Fusiello and Irsara [21], who approximated the Euclidean (calibrated) case by enforcing the rectifying transformation with collineation induced by the plane at infinity. Using the same Quasi-Euclidean constraints, Ko *et al.* [22] enhanced the algorithm by taking into account the impact of multiple types of geometric distortions in their computations. In the case of handheld devices such as smartphones, Xiao *et al.* [23] proposed a low distortion rectification method under uncalibrated scenarios.

All these methods assumes that cameras use the same focal lengths. This means that images captured by those cameras share a similar field of view (FoV). However, in real-world applications such as industrial robots or autonomous vehicles, camera systems often use a variety of focal lengths. A lens with a wide FoV is designed to capture close objects in larger contexts, while those with a narrow FoV can emphasize distant objects. Differences in FoV and large baselines often cause errors in rectification. Few studies have been published on this field. Kumar *et al.* [24], [25] explored how to balance object sizes in dissimilar FoV scenarios. However, their approach needs focal length information [24] or a calibration process [25]. Dinh *et al.* [26] presented the first algorithm that estimated a re-scaling ratio using correspondence information by including a FoV compensation module. Their method is inefficient because it requires repeating scale-invariant feature transform (SIFT) procedure [27]. Moreover, due to the lack of feature matching enhancement, their method is robustly affected by outliers. Finally, because of the lack of module to control for geometric errors, rectified images are generally over-distorted when produced under dissimilar FoVs. Recently, Tran and Nguyen [29] proposed including a FoV neutralization module and rectification process with adaptive geometric constraints (R-wAGC). This method can keep rectified images under geometric thresholds while minimizing vertical disparity errors.

All uncalibrated rectification algorithms require matching input pairs. Matching pairs are the results from feature matching techniques such as SIFT [27] or oriented fast and rotated BRIEF [28]. However, these matching pairs are imperfect and include outliers that, in some cases, can cause crucial miscalculations in the optimization process, particularly when rectification is used to satisfy multiple purposes. We proposed a method that can learn from its own rectification results and enhance the set of correspondences. The proposed method works great in both normal and dissimilar FoV conditions. We also enhanced the optimization strategy described in Tran and Nguyen [29]. The updated strategy requires fewer empirical parameters and provide superior stability between rectification and geometric errors. Our proposed method addresses each of the problems discussed previously. The experimental results show that the proposed algorithm outperforms other benchmark methods in terms of both vertical disparity errors and geometric errors. The main contributions of our paper are summarized below.

- We proposed a new pipeline inherited from Tran and Nguyen [29] for uncalibrated rectification in both normal conditions and with different FoV circumstances.

The inclusion of an FMR module means our approach can learn from its own results to enhance correspondence inputs. The resulting rectified images exhibit significant improvement in quality.
- We also introduced a new optimization strategy. The upgraded version produces lower rectification errors and remove unwanted artifacts associated with distortion.
- We increased the SKKU-dFoVs dataset by adding lens combinations. We also developed an extra dataset, SKKU-dFoVs+, which includes dissimilar FoVs. However, for the new dataset, we focused on hostile conditions, such as blurring, overexposure, or dim lighting.

The rest of the paper is organized as follows. In Section II, we provide a brief summary of uncalibrated rectification and the Tran and Nguyen [29] method. In Section III, we describe a workflow for uncalibrated rectification and the role and operation of feature matching re-selection (FMR). We then demonstrate our enhanced rectification strategy. Section IV supplies an analysis of our experiments using four different datasets. Last, we discuss possible improvements and conclusions in Section V.

II. PRELIMINARY

A. UNCALIBRATED RECTIFICATION

The corresponding pairs in two images are on the same horizontal axis when they are rectified. The cameras used for capturing must fulfill the pinhole model assumption [12]. The views from those cameras needs to be under the epipolar constraint to perform rectification. The epipolar constraint is defined as:

$$X_l^T \mathbf{F} X_r = 0 \tag{1}$$

where X_l and X_r are the homogeneous coordinates of matching points $(x, y, 1)$. The subscripts l for left and r for right define which image the matrix belong to. $\mathbf{F}$ is the fundamental 3×3 matrix. For rectification, $\mathbf{F}$ is defined by the following equation:

$$\mathbf{F} = \mathbf{F}_\infty = [e_1]_\times = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} \tag{2}$$

The fundamental matrix $\mathbf{F}$ is:

$$\mathbf{F} = \mathbf{H}_l^T [\mathbf{e}_1]_\times \mathbf{H}_r \tag{3}$$

where $\mathbf{H}_l$ and $\mathbf{H}_r$ are the invertible 3×3 homography matrices for the left and right cameras, respectively. The homography matrix $\mathbf{H}$ can be derived as the combination of rotation matrix $\mathbf{R}$ and the internal matrix $\mathbf{K}$.

$$\mathbf{H} = \mathbf{K}_n \mathbf{R} \mathbf{K}_o^{-1} \tag{4}$$

The $\mathbf{K}$ matrix contains the unresolved focal length f and the principal point $P(x_P, y_P)$. Generally, with the pinhole model assumption [12], the center point of the images is the principal point. The following equation describes the $\mathbf{K}$ matrix format:

$$\mathbf{K} = \begin{pmatrix} f & 0 & x_P \\ 0 & f & y_P \\ 0 & 1 & 1 \end{pmatrix} \tag{5}$$

Three rotation coordinates x, y, and z apply to a camera in affine space when it comes to rotation matrices. $\mathbf{R}_x$, $\mathbf{R}_y$, and $\mathbf{R}_z$ are the three rotation matrices for each camera. The rotating angle for each rotation matrix is represented by θ value.

From Eq.(1), the relationship between $\mathbf{K}$, $\mathbf{R}$, and $\mathbf{H}$ matrices can be described as:

$$\mathbf{X}_l^T \mathbf{K}^{-1^T} \mathbf{R}_l^T \mathbf{K}^T [\mathbf{e}_1]_\times \mathbf{K} \mathbf{R}_r \mathbf{K}^{-1} \mathbf{X}_r = 0 \tag{6}$$

The Sampson error defined in [21] is commonly used for the optimization process.

$$E_{Sampson}(\mathbf{X}_l, \mathbf{X}_r)^2 = \frac{E(\mathbf{X}_l, \mathbf{X}_r)^2}{\left\| [\mathbf{e}_3]_\times \mathbf{F}^T \mathbf{X}_l \right\|^2 + \left\| [\mathbf{e}_3]_\times \mathbf{F} \mathbf{X}_r \right\|^2} \tag{7}$$

where $[\mathbf{e}_3]_\times = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$.

The x-axis of the left camera is treated as an anchor and set to 0. The optimization process is looking for six optimized parameters (θ_{ly} θ_{lz} θ_{rx} θ_{ry} θ_{rz} g) which minimizes Sampson errors, where $f = 3^g(w + h)$. The parameter g, with a range of$[-1, 1]$, controls the focal length f in a $\mathbf{K}$ matrix. The width and the height of the input images are defined as w and h, respectively.

B. UNCALIBRATED RECTIFICATION WITH LOW GEOMETRIC DISTORTION

In Tran and Nguyen [29], the proposed method is relatively accurate and comes with a low distortion rate. The algorithm work great in both normal and dissimilar FoV scenarios. Fig. 2 demonstrates the workflow of [29] algorithm:

The workflow contains three main modules: matching pair generation (MPG), field of view neutralization (FoV neutralization), and rectification. The outputs from each module affect the next module's outputs and final results. In MPG, the correspondences $\{m_l, m_r\}$ are generated for left and right images, respectively, by combining the existing SIFT [27]

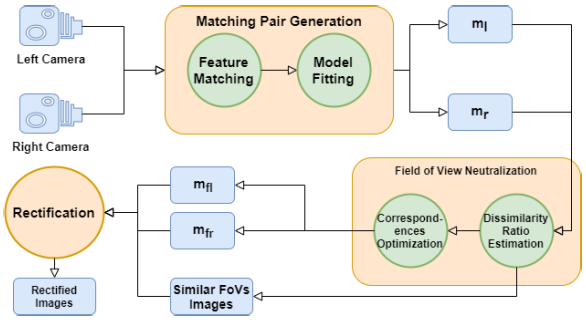

FIGURE 2. Rectification workflow for dissimilar FoV scenarios.

method with a neural-guided RANSAC (NG-RANSAC) [30]. The number of correspondences $\{m_l, m_r\}$ are marked as N.

Using the correspondences $\{m_l, m_r\}$ as inputs, the FoV neutralization module estimates the approximate dissimilar ratio γ between left and right images. Due to the cameras' focal lengths, objects in wide views (small focal lengths) are smaller than those in narrow views (large focal lengths). The method automatically recognizes the narrower views. To simplify notation, we assume that narrower FoV images are on the left side. Based on this assumption, the dissimilarity ratio γ is in the (0, 1] range, which is calculated by:

$$\gamma = \frac{\sum_{i=1}^{N} d_l^i / d_r^i}{N} \tag{8}$$

where $\mathbf{D1}_l : \{d_l^i, \ldots, d_l^N\}$ and $\mathbf{D1}_r : \{d_r^i, \ldots, d_r^N\}$ are the Manhattan distances for m_l and m_r, respectively. Using dissimilarity ratio γ, the method applies a transformation matrix $\mathbf{H}_{balance}$ onto narrower FoV images. $\mathbf{H}_{balance}$ can be expressed as

$$\mathbf{H}_{balance} = \begin{pmatrix} \gamma & 0 & w\frac{1-\gamma}{2} \\ 0 & \gamma & h\frac{1-\gamma}{2} \\ 0 & 1 & 1 \end{pmatrix} \tag{9}$$

Matching points m_l are converted to m_{bl}, using the similar transformation matrix $\mathbf{H}_{balance}$. To maintain uniformity in notation, the balanced correspondences are defined as m_{bl} and m_{br} for the left and right images, respectively.

Next, the FoV neutralization module uses the dissimilarity ratio γ to remove correspondences in blind areas. Due to the assumption that right cameras have wider FoVs, blind areas are the regions that the left cameras cannot see. A new set of correspondences belonging to viewable areas, $\{m_{blo}; m_{bro}\}$ are selected from the m_{bl} and m_{br} pool. The new correspondences must satisfy the following conditions:

$$w(\frac{1+\gamma}{2}) > x > w(1-\gamma), \quad h(\frac{1+\gamma}{2}) > y > h(1-\gamma) \tag{10}$$

The FoV neutralization module then calculates the sequences $\mathbf{V} : \{v_i, \ldots, v_{M_1}\}$ containing M_1 elements of L_1 distances between m_{blo} and m_{bro}. The system now estimates the variance $\sigma_{\mathbf{V}}^2$ between $v_i \in \mathbf{V}$.

$$\sigma_{\mathbf{V}}^2 = \frac{1}{M_1} \sum_{i=1}^{M_1} (v_i - \mu_r)^2 \tag{11}$$

The $\{m_{blo}^i; m_{blo}^i\}$ pairs for which $(\sigma_{\mathbf{V}_i} > \sigma_{\mathbf{V}})$ are then removed. The remaining correspondences are denoted as $(m_{fl}; m_{fr})$ with M_2 elements.

Unlike the original rectification method [21], [26], the algorithms are extended to look for the nine optimized parameters [22], [29]: (θ_{ly} θ_{lz} θ_{rx} θ_{ry} θ_{rz} g_l g_r t_l t_r). Instead of using the same focal length f for both left and right images, the authors used different values of f for each images, resulting in different values for $\mathbf{K}_l$ and $\mathbf{K}_r$.

$$\mathbf{K}_l = \begin{pmatrix} f_l & 0 & x_C \\ 0 & f_l & y_C \\ 0 & 1 & 1 \end{pmatrix} \quad \text{and } \mathbf{K}_r = \begin{pmatrix} f_r & 0 & x_C \\ 0 & f_r & y_C \\ 0 & 1 & 1 \end{pmatrix}$$

Other extra parameters are the translation parameters: t_l and t_r, which stand for left and right translation values, respectively. Eq.(6) can be described as:

$$(\mathbf{K}_l\mathbf{T}_l\mathbf{R}_l\mathbf{K}_l^{-1})^T[\mathbf{e}_1]_\times(\mathbf{K}_r\mathbf{T}_r\mathbf{R}_r\mathbf{K}_r^{-1}) \tag{12}$$

where,

$$\mathbf{T}_l = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & t_l \\ 0 & 0 & 1 \end{pmatrix} \quad \text{and } \mathbf{T}_r = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & t_r \\ 0 & 0 & 1 \end{pmatrix}$$

To lower the distortion level, Ko *et al.* [22] and Tran and Nguyen [29] both use four geometric constraints: the modified aspect ratio E_{AR}, skewness E_{SK}, rotation angle E_R, and size ratio E_{SR}. They defined four corner points (a_1, a_2, a_3, a_4), four middle points (b_1, b_2, b_3, b_4) and the center point of input images o.

$$E_{AR} = \frac{1}{2}(\frac{\overline{a_1'o'}}{\overline{a_3'o'}} + \frac{\overline{a_2'o'}}{\overline{a_4'o'}}), \quad E_{SK} = \frac{1}{4}\sum_{i=1}^{4}(|90^o - \angle CA_i|)$$
$$E_R = \cos^{-1}(\frac{\overline{ob_2}.\overline{o'b_2'}}{|\overline{ob_2}|\,|\overline{o'b_2'}|}), \quad E_{SR} = \frac{Area_{rec}}{Area_{orginal}} \tag{13}$$

The cost function defined in [29] is the combination of a Sampson error $E_{Sampson}$ and four geometric constraints:

$$C(\varphi) = \frac{1}{2}(E_{Sampson} + \sum_{\hat{X}} \frac{\acute{\rho}_{\hat{X}}}{N_{\hat{X}}} E_{\hat{X}}), \quad \text{where } \acute{\rho}_{\hat{X}} = 1/n_{\hat{X}} \tag{14}$$

where X can be $AR, SK, R,$ or SR. N_X is a normalization terms that considers their value ranges.

$$N_{AR} = 1.5, \quad N_{SK} = 6.5, \; N_R = 18.5, \; N_{SR} = 2.5 \tag{15}$$

The sum of $\rho_{\hat{X}}$ is equal to 1. $E_{Sampson}$ and $E_{G_{\hat{X}}}$ therefore are weighted evenly to the cost function:

$$\sum_{\hat{X}} \rho_{\hat{X}} = 1, \text{where } \hat{X} = \{\hat{x} | \hat{x} \text{ is active term, and } \hat{x} \in X\} \tag{16}$$

where $n_{\hat{X}}$ as the number of effective geometric terms. In both USR-CGD [22] and Tran and Nguyen [29], the thresholds for geometric parameters are predefined as follows:

$$0.8 \le E_{AR} \le 1.2, \quad E_{SK} \le 5^\circ,$$
$$0.8 \le E_{SR} \le 1.2, \quad E_R \le 30^\circ \tag{17}$$

According to [29], to find optimal solutions for the cost function in Eq.(14), the system uses Levenberg-Marquardt

algorithms [31] and follows a strategy represented in Algorithm 1. The reduction ratio ψ between φ_i and φ_{i+1} is calculated as:

$$\psi = \frac{C(\varphi_{i+1})}{\Delta(C(\varphi_i), C(\varphi_{i+1}))} \tag{18}$$

where $C(\varphi_i)$ is the errors of φ_i parameters, and $C(\varphi_{i+1})$ represents the errors of the predicting parameters. The damping parameter λ is a non-negative component controlling the reduction rates of the cost function.

Algorithm 1 Optimization Algorithm

Require: , $(m_{fl}; m_{fr})$
Ensure: Optimized φ
  $\varphi_0 = 0$
  $\lambda = 10^{-3}$
  $iteration_{max} = 300$
  $T_\psi = 1e^3$, $C_{max} = 10^{-2}M$
  while $C(\varphi_i) > C_{max}$ and $iteration < iteration_{max}$ **do**
    if $\Delta(C(\varphi_{i+1}), C(\varphi_i)) < T_{\Delta_C}$ **then**
      accept φ_i, stop the optimization
    else if $C(\varphi_{i+1}) > C(\varphi_i)$ **then**
      deny φ_{i+1}, increase λ
    else
      Calculate $\psi_i = \frac{C(\varphi_{i+1})}{\Delta(C(\varphi_i), C(\varphi_{i+1}))}$
      if $\psi_i > T_\psi$ and $\sum_{\hat{X}} E_{G_{\hat{X}}} > 0$ **then**
        deny φ_{i+1}, increase λ
      else
        accept φ_{i+1}
        $\lambda = \lambda/\psi_i$
      end if
    end if
  end while

III. PROPOSED METHOD

A. FEATURE MATCHING RE-SELECTION MODULE AND NOVEL PIPELINE

Despite the previously reported outstanding performance of the FoV neutralization module [29], misleading matching pairs remain in the final optimization process and inaccurate correspondences are inevitable. To lower the impact of incorrect inputs, we proposed our new workflow with a FMR module. As shown in Fig. 3, the FMR module analyzes the rectification results and selects the optimal matching pairs. The idea was inspired by machine learning technologies in which inaccurate correspondences are gradually removed from the pool by observing the results of rectification.

The standard measurement to remove the outliers from feature matching process is vertical disparity error. The vertical disparity error of (m_l^i, m_r^i) matching pair is calculated as:

$$(E_v)_i = \left| \left(\mathbf{H}_l m_l^i\right)_2 - \left(\mathbf{H}_r m_r^i\right)_2 \right| \tag{19}$$

where $\mathbf{H}_l$ and $\mathbf{H}_r$ are outputs from the rectification module. We also removed the correspondence optimization sub-module from the FoV neutralization module because FMR module removes outliers. Our new FMR module covers the function of correspondence optimization sub-module. From the correspondences $\{m_{bl}, m_{br}\}$ having M_1 elements which is the output of the dissimilar ratio estimation module, we estimate $\mathbf{J} : \{E_{v_i}, \ldots, E_{v_{M_1}}\}$, the set of vertical disparity errors. Those pairs for which $E_{v_i} > T_{ver\_disp\_err}$ were removed from the pool. In this work, we set $T_{ver\_disp\_err} = 0.5\ pixel$. The remaining pairs were fed back to the rectification module. $\mathbf{H}_l$ and $\mathbf{H}_r$ from the rectification module were accepted if those conditions were satisfied:

$$E_v(\mathbf{J}) = 0, \quad or\ M_1 < 10 \tag{20}$$

where

$$E_v(\mathbf{J}) = \frac{1}{M_1} \sum_{i=1}^{M_1} (E_v)_i \tag{21}$$

While the first condition stops FMR module when there is no outlier, the second condition works as a safe trigger to keep enough input correspondences for the rectification stage.

Fig. 4 provides an illustration of our FMR module. In this example, the system proceeded through three iterations and the wrong matching pairs were gradually removed from the correspondence pool.

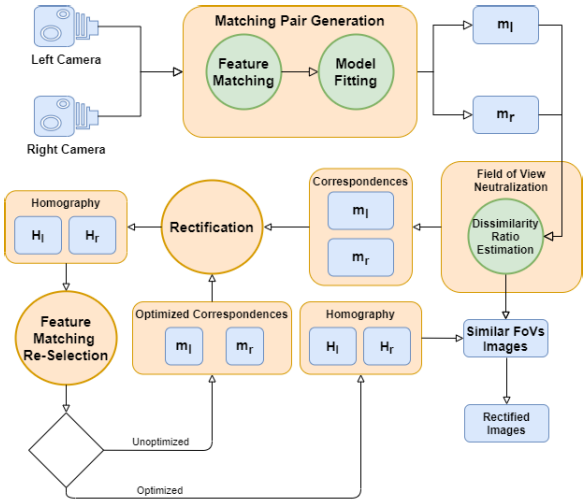

FIGURE 3. Proposed workflow for uncalibrated rectification with a FMR module.

B. IMPROVED OPTIMIZATION STRATEGY

In Algorithm 1, the damping parameter λ plays a critical role in the optimization process, controlling the direction of the optimizing process. For rectification processes that do not involve geometric errors such as [21], [26], λ typically only works with the rectification error itself. However, in [22], [29], geometrical terms are parts of the process. The λ adjustment now needs to consider the geometric errors generated at each step of the procedure. Ko *et al.* [22], they use a cost function expressed as a combination of the Sampson error

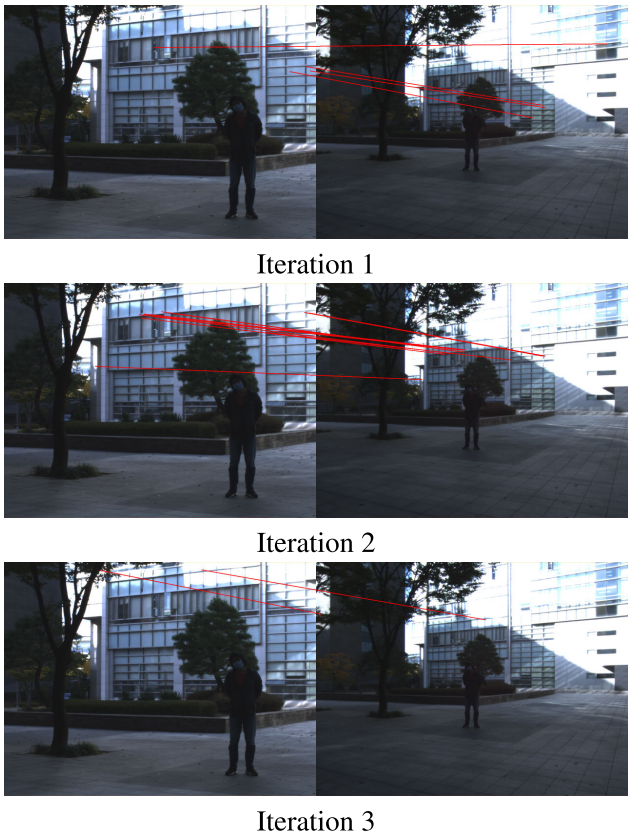

FIGURE 4. A demonstration of the feature matching re-selection module. The sample is 0804_900mm from SKKU-dFoVs dataset.

and geometric errors. By evaluating the combined cost function, the authors change λ accordingly to the Levenberg-Marquardt [31] method. Tran and Nguyen [29] used the same technique with their optimization strategy shown in the Algorithm 1. In this work, we followed Tran *et al.*'s optimization strategy with two modifications.

First, we removed $T_\psi = 1e^3$, the threshold value for the reduction ratio ψ. Generally, if ψ is too high, the convergences are too fast and the optimizing process can bypass the optimal solution. The purpose of T_ψ is to prevent careless convergences. However, as T_ψ is an empirical value, it will not work in every case. By removing T_ψ, we improved the generalization of the entire process.

Second, after removing T_ψ, the process needs to be modified to handle the speed of convergences. Without T_ψ, we proposed Algorithm 2, an updated version of Algorithm 1. In the updated strategy, when the $C(\varphi_{i+1}) \leq C(\varphi_i)$, we only considered one condition, specifically $\sum_{\hat{X}} E_{G_{\hat{X}}} > 0$. If the above scheme appears, we calculated two outputs with two different λ values and compared them. The two trial λ_{t1} and λ_{t2} values were calculated as:

$$\lambda_{t1} = \lambda_i * \lambda_{base\_fact} \quad and \ \lambda_{t2} = \lambda_i / \psi_i \tag{22}$$

where λ_i and ψ_i are the λ and reduction rate at the i^{th} iteration. λ_{base_fact} is the parameter used for increasing damping value,

TABLE 4. Run time of the compared methods.

	FoV Neutralization (s)	Rectification (s)	Total (s)	Average (s)
Fusiello	0.00	37.13	37.13	0.65
USR-CGD	0.00	11.85	11.85	0.21
Dinh	77.97	35.36	113.33	1.99
Tran	28.63	13.07	41.70	0.73
Proposed Method	28.41	24.35	52.76	0.93

which is generally set to 10 in most rectification methods using Levenberg-Marquardt's method such as [21], [22], [26], [29]. By comparing $C(\varphi_{\lambda_{t1}})$ and $C(\varphi_{\lambda_{t2}})$, which are computed using Eq.(14), we selected the λ value if its $C(\varphi)$ was smaller than that of another scenario. Because the cost function $C(\varphi_\lambda)$ is based on Eq.(14) in which both $E_{Sampson}$ and geometric errors E_G are taken into account evenly, a smaller value between $C(\varphi_{\lambda_{t1}})$ and $C(\varphi_{\lambda_{t2}})$ represents a superior route for the optimization process. Details of the improved optimization strategy are shown in Algorithm 2.

Algorithm 2 Updated Optimization Algorithm

Require: , $(m_{fl}; m_{fr})$
Ensure: Optimized φ
$\varphi_0 = 0$
$\lambda = 10^{-3}$, $\lambda_{base\_fact} = 10$
$iteration_{max} = 300$
~~$T_\psi = 1e^3$~~, $C_{max} = 10^{-2}M$
while $C(\varphi_i) > C_{max}$ and $iteration < iteration_{max}$ **do**
  if $\Delta(C(\varphi_{i+1}), C(\varphi_i)) < T_{\Delta_C}$ **then**
    accept φ_i, stop the optimization
  else if $C(\varphi_{i+1}) > C(\varphi_i)$ **then**
    deny φ_{i+1}, $\lambda_{t1} = \lambda_i * \lambda_{base\_fact}$
  else
    Calculate $\psi_i = \frac{C(\varphi_{i+1})}{\Delta(C(\varphi_i), C(\varphi_{i+1}))}$
    if ~~$\psi_i > T_\psi$~~ and $\sum_{\hat{X}} E_{G_{\hat{X}}} > 0$ **then**
      ~~deny φ_{i+1}, increase λ~~
      Calculate $\lambda_{t1} = \lambda_i * \lambda_{base\_fact}$, $\lambda_{t2} = \lambda_i / \psi_i$
      if $C(\varphi_{\lambda_{t1}}) < C(\varphi_{\lambda_{t2}})$ **then**
        deny φ_{i+1}, $\lambda_{t1} = \lambda_i * \lambda_{base\_fact}$
      else
        accept φ_{i+1}, $\lambda = \lambda / \psi_i$
      end if
    else
      accept φ_{i+1}, $\lambda = \lambda / \psi_i$
    end if
  end if
end while

IV. EXPERIMENTS

A. EXPERIMENT DATABASES

We used total four different datasets to evaluate algorithms. These are Syntim [32], MCL-RS [33], SKKU-dFoVs [29],

TABLE 1. Dataset information.

	Syntim	MCL-RS	SKKU-dFoVs	SKKU-dFoVs+
# of Test Images	12	10	20	15
Resolution	512x512 or 768x576	1920x1080	1280x960	2448x2048
Additional Information	- 12 indoor scenes	- 3 indoor scenes - 7 outdoor scenes	- Outdoor dataset, baselines: 300mm - 900mm - Focal length combination: 8mmx6mm, 8mmx4mm, 8mmx8mm, 12mmx8mm, 16mmx8mm.	- Outdoor dataset - Hostile conditions: overexposed, low-light or blurring.

FIGURE 5. Comparison of vertical disparity errors and geometric distortion for Serfaty0_OG2 in the Syntim database.

and SKKU-dFoVs+. While Syntim and MCL-RS are for normal cases and most are captured indoors, SKKU-dFoVs focuses on dissimilar FoV scenarios in outside environments. The images in the SKKU-dFoVs dataset were captured using Sony XCL-C130C cameras. Left and right cameras are linked to a MATROX frame grabber to capture images simultaneously. The baselines between cameras vary from 300mm to 900mm. In [29], the SKKU-dFoVs dataset contains five lens combinations: 8mm_4mm, 8mm_6mm, 8mm_8mm, 12mm_8mm, and 16mm_12mm. In this work, we add one new combination 16mm_8mm. We also add more cases to the existing pairs and an extra dataset containing images of random lens combinations with multiple environmental conditions such as overexposure, low-light or blurring. We named these extra datasets SKKU-dFoVs+. The details of dataset information is provided in Table 1.

B. PERFORMANCE EVALUATION

We compared our proposed method to four rectification algorithms: Fusiello and Irsara [21], USR-CGD [22], Vinh [26], Tran *et al.* [29]. We used the vertical disparity error $E_v(Z)$ in Eq.(22) as the objective error measurement, with Z as the final correspondence set generated by our proposed algorithm. In addition to the standard measurement, we used the modified aspect ratio E_{AR}, skewness E_{SK}, rotation angle

TABLE 2. Performance comparison of five rectification methods in terms of vertical disparity error and geometric errors.

Image Name	Fusiello		USR-CGD		Vinh		Tran		Proposed Method	
Syntim (E_V / E_G)										
Aout0_OG0	0.407	1.157	0.356	1.156	2.101	0.593	0.258	0.000	**0.232**	0.000
Baballe0_OG0	1.201	0.000	0.274	0.740	0.334	0.638	0.221	0.000	**0.216**	0.000
BalMouss0_OG0	8.907	0.000	**0.253**	0.000	0.411	0.504	0.272	0.000	**0.253**	0.000
BatInria0_OG0	0.219	0.000	**0.210**	0.000	0.722	0.930	0.269	0.000	**0.210**	0.000
Color0_OG0	1.325	0.495	**0.323**	0.726	1.537	0.000	0.394	0.000	0.357	0.000
GrRub0_OG0	0.828	4.424	**0.216**	0.600	3.238	0.516	1.385	0.617	0.484	0.000
NBuste0_OG0	0.171	0.000	**0.162**	0.000	0.272	0.000	0.176	0.000	0.163	0.000
Rubik0_OG0	0.268	0.000	**0.257**	0.000	1.009	0.596	0.291	0.000	0.258	0.000
Sabine0_OG0	0.229	0.000	0.230	0.000	2.403	19.611	0.230	0.000	**0.228**	0.000
Serfaty0_OG2	0.285	1.045	0.315	0.927	0.379	1.278	0.331	0.964	**0.281**	0.000
Sport0_OG0	0.337	0.000	0.189	0.000	0.429	0.000	0.192	0.000	**0.187**	0.000
Tot0_OG0	0.242	0.000	0.226	0.000	0.602	1.171	0.243	0.000	**0.220**	0.000
Mean	1.202	0.593	**0.251**	0.346	1.120	2.153	0.355	0.132	0.258	0.000
MCL-RS (E_V / E_G)										
DOG	47.722	4.396	0.640	1.984	0.505	0.000	0.613	0.000	**0.339**	0.000
DOHENY	0.172	0.000	0.169	0.000	0.203	0.000	0.172	0.000	**0.168**	0.000
DOLLS	0.340	0.000	0.351	0.000	0.416	0.000	0.365	0.000	**0.337**	0.000
DRAWER	0.312	0.808	0.290	1.289	0.201	0.000	**0.204**	0.000	**0.204**	0.000
KEYCHAIN	0.327	0.000	**0.285**	0.000	0.423	0.000	0.331	0.000	0.286	0.000
Leavy	0.245	0.000	**0.242**	0.000	2.915	0.000	0.312	0.000	**0.242**	0.000
MuddHall	0.205	0.000	**0.196**	0.000	0.371	0.000	0.267	0.000	**0.196**	0.000
RTCC	1.174	0.000	**0.357**	0.640	1.375	0.000	0.377	0.620	0.482	0.000
STEP	0.738	0.000	**0.495**	3.166	1.150	0.000	1.036	0.000	0.543	0.000
ViterbiSchool	0.408	0.510	0.304	0.566	0.368	0.518	**0.298**	0.553	0.323	0.000
Mean	5.164	0.571	0.333	0.764	0.793	0.052	0.397	0.117	**0.312**	0.000
SKKU_dFoVs (E_V / E_G)										
0806_300mm	2.057	1.311	0.294	0.736	0.564	0.000	**0.206**	0.000	0.207	0.000
0806_300mm_0	2.059	2.047	0.324	1.002	0.436	0.906	**0.232**	0.000	**0.232**	0.000
0806_300mm_1	44.714	0.000	0.349	1.046	0.233	0.000	0.250	0.976	**0.208**	0.000
0806_300mm_2	1.679	24.170	0.272	0.812	0.362	0.484	0.233	0.000	**0.197**	0.000
0806_600mm_0	1.179	1.162	0.327	0.801	0.730	0.000	0.311	0.000	**0.239**	0.000
0806_600mm_1	0.722	1.191	0.314	0.810	0.458	0.000	**0.263**	0.000	0.344	0.000
0806_600mm_2	3.976	1.140	0.348	0.857	0.961	0.000	0.879	0.000	**0.256**	0.000
0806_900mm	2.188	37.374	0.331	0.792	1.237	0.000	0.334	0.000	**0.246**	0.000
0804_300mm	19.048	2.604	53.202	1.759	0.752	7.427	0.587	0.000	**0.461**	0.000
0804_600mm	1.909	1.974	0.480	2.353	0.350	1.530	0.229	0.000	**0.170**	0.000
0804_900mm	15.025	2.287	19.932	2.035	30.183	1.351	1.673	2.239	**0.565**	0.000
0808_300mm	0.211	0.000	0.202	0.000	0.415	0.000	0.208	0.000	**0.200**	0.000
0808_600mm	0.231	0.000	0.229	0.000	1.349	1.435	0.241	0.000	**0.228**	0.000
0808_900mm	0.791	0.000	0.288	0.776	1.171	0.000	**0.276**	0.752	0.278	0.000
1208_300mm	1.068	2.842	0.329	0.858	0.616	0.000	0.249	0.000	**0.228**	0.000
1208_600mm	3.968	96.563	0.291	0.831	0.956	0.923	0.244	0.000	**0.209**	0.000
1208_900mm	1.023	1.979	0.309	0.843	0.929	0.754	0.230	0.000	**0.209**	0.000
1608_300mm	8.040	985.047	0.310	1.399	0.553	2.120	0.228	0.000	**0.164**	0.000
1608_600mm	0.895	2.197	0.351	1.393	2.271	2.019	0.274	0.000	**0.184**	0.000
1608_900mm	1.866	56.332	0.697	3.465	0.968	1.677	0.449	1.385	**0.380**	0.000
Mean	5.821	64.153	4.152	1.149	2.365	1.086	0.389	0.282	**0.263**	0.000
SKKU_dFoVs+ (E_V / E_G)										
Back_Yard	8.610	460.596	0.467	3.597	1.084	36.478	0.414	4.746	**0.292**	0.000
Building	1.053	0.864	0.441	2.067	0.267	0.000	0.310	0.000	**0.220**	0.000
Building2	32.733	19.533	0.339	0.602	0.310	0.701	**0.235**	0.000	0.244	0.000
Building3	2.372	41.800	0.287	0.712	0.633	0.000	0.240	0.000	**0.220**	0.000
Building4	0.507	0.896	0.321	0.808	0.732	0.000	0.327	0.000	**0.261**	0.000
Building5	0.602	0.883	0.294	0.685	0.328	0.000	0.242	0.000	**0.226**	0.000
Building_Entrance	0.481	1.010	0.352	0.719	0.477	0.000	0.292	0.000	**0.271**	0.000
FlowerGarden	0.592	0.790	0.220	0.703	0.504	0.000	0.178	0.000	**0.168**	0.000
Graffiti	0.639	0.964	0.252	0.942	0.199	0.000	0.203	0.951	**0.178**	0.000
Graffiti2	0.991	0.794	0.234	0.694	0.250	0.000	0.268	0.000	**0.180**	0.000
Library_Front	3.832	363.917	0.368	1.002	1.639	1.860	0.305	0.000	**0.238**	0.000
Library_Side	2.636	0.969	0.282	0.959	0.612	2.421	0.246	0.000	**0.201**	0.000
Mirror	8.626	1.423	0.297	0.614	1.476	0.000	**0.290**	0.000	0.292	0.000
Parking	8.688	0.870	0.300	0.686	0.328	0.000	0.252	0.000	**0.231**	0.000
Parking2	0.491	1.322	0.257	0.609	0.234	0.000	0.251	0.000	**0.213**	0.000
Mean	4.857	59.775	0.314	1.027	0.605	2.764	0.270	0.380	**0.229**	0.000

TABLE 3. The comparison between tran and the proposed method in terms of geometric distortion errors.

Ideal Values	$E_V = 0$		$E_{AR} = 1$		$E_{SK} = 0$		$E_R = 0$		$E_{SR} = 1$	
Dataset	Tran	Proposed Method	Tran	Proposed Method	Tran	Proposed Method	Tran	Proposed Method	Tran	Proposed Method
Syntim	0.355	**0.258**	1.021	**1.014**	2.686	**2.513**	2.787	**2.705**	1.149	**1.078**
MCL-RS	0.397	**0.312**	**1.005**	1.006	**1.762**	1.827	6.877	**6.679**	1.114	**1.094**
SKKU_dFoVs	0.380	**0.260**	**1.006**	0.987	3.499	**2.136**	3.104	**1.652**	1.423	**1.079**
SKKU_dFoVs+	0.270	**0.229**	1.079	**0.989**	3.173	**2.497**	**1.692**	1.775	2.700	**1.095**

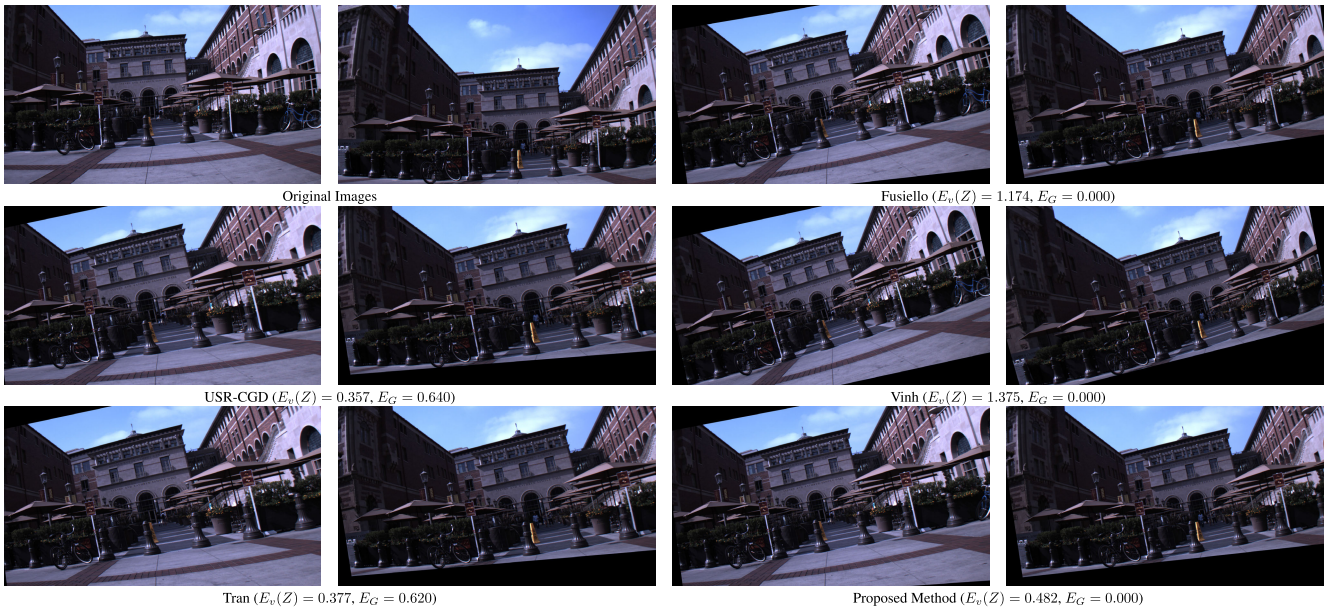

FIGURE 6. Comparison of vertical disparity errors and geometric distortion for RTCC in the MCL-RS database.

E_R, and size ratio E_{SR}, as subjective measures. By using the geometric thresholds in Eq.(13) as references, we calculated the geometric error E_G as:

$$E_G = \sum_{\hat{X}} \frac{\acute{\rho}_{\hat{X}}}{N_{\hat{X}}} E_{\hat{X}}, \quad \text{where } \acute{\rho}_{\hat{X}} = 1/n_{\hat{X}} \tag{23}$$

where X can be AR, SK, R, or SR. N_X is the normalization term that considers the value ranges, mentioned in Eq.(17). Table 2 provides results for the vertical disparity error $E_v(Z)$ and the geometric error E_G for five benchmark algorithms. For each dataset, we calculated the additional mean values of $E_v(Z)$ and E_G, along with the individual result for each image pair. Optimal results are in bold. In Table 3, we directly compare the new proposed method to the Tran *et al.* [29] method for both rectification error and each geometric term. In Table 4, we compare the running times between benchmark methods.

In the experiments, we applied SIFT [27] and then NG-RANSAC [30] to derive the initial correspondences between two input images. In case of NG-RANSAC, a pre-trained model was used. The model was trained on KITTI datasets and the matching pairs are generated by using SIFT method. During the experiments, the number of correspondences varied between image pairs. However, for a particular image pair, the number of input matching pairs was the same for every benchmark algorithm to maintain the consistency. The PC used for the experiments had a core i7-8700 3.20Ghz processor and 32GB of RAM. All programs were run on Windows 10.

In the first comparison, as shown in Table 2, our results outperformed those of other algorithms. Fusiello and Vinh's method achieved extremely high values in both vertical disparity error $E_v(Z)$ and geometric error E_G. The reason is that their method does not include modules that can process matching input pairs and geometric distortion. Vinh's method, because of its FoV neutralization module, can process some relatively simple cases in the SKKU-dFoVs and SKKU-dFoVs+ datasets. Unfortunately, when the differences between focal length are large or the distances between two cameras are great, their method begins to introduce errors. The USR-CGD method encountered the same problem, although it performed well in normal cases without the benefit of a FoV neutralization module. In some experiments, the $E_v(Z)$ results of USR-CGD or Tran were smaller than ours. However, their geometric error E_G rose up. In most cases, our rectification method achieved the lowest vertical disparity error $E_v(Z)$, resulting in the lowest mean values for

FIGURE 7. Comparison of vertical disparity errors and geometric distortion for 0804_900mm in the SKKU_dFoVs database.

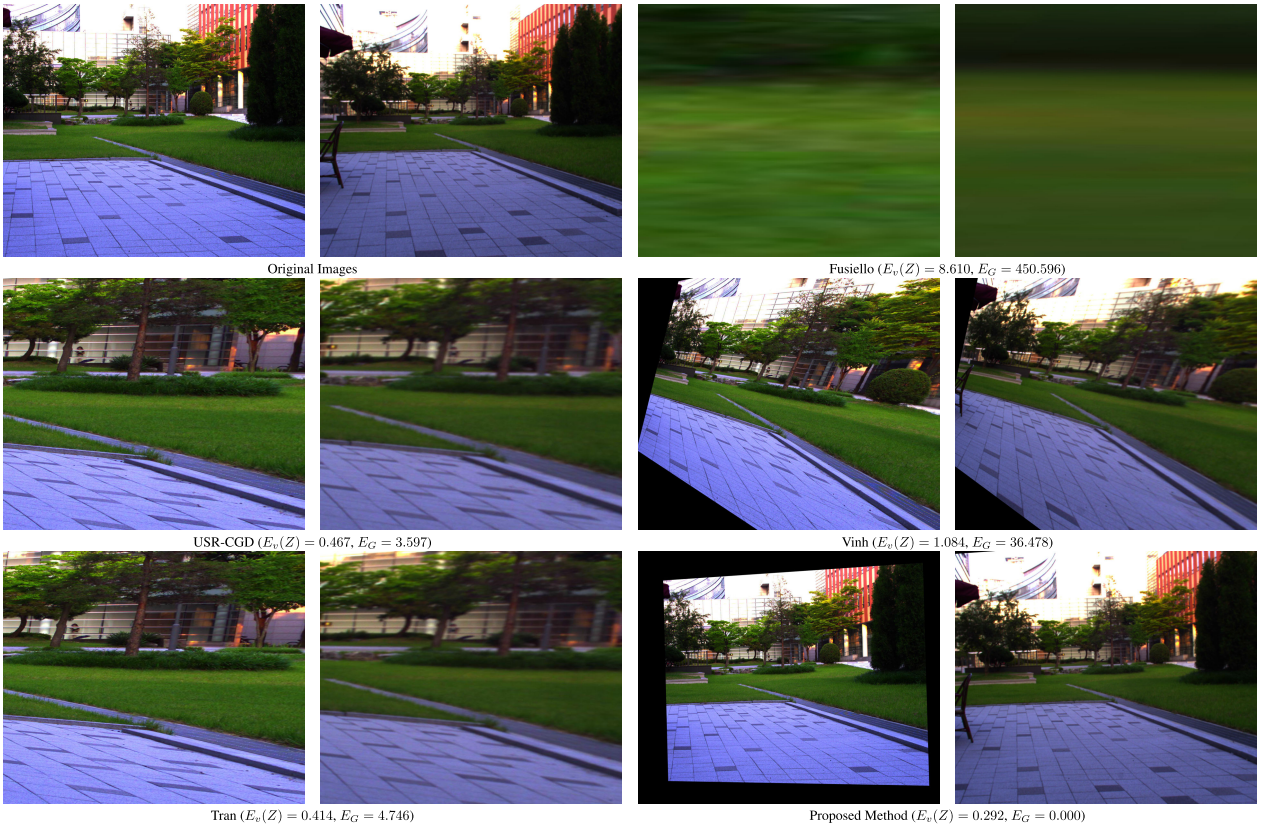

FIGURE 8. Comparison of vertical disparity errors and geometric distortion for Back_Yard in the SKKU_dFoVs+ database.

the MCL-RS, SKKU-dFoVs, and SKKU-dFoVs+ datasets. For the Syntim dataset, although the USR-CGD method produced a lower vertical disparity error $E_v(Z)$ than our method, it also contained geometric error. For all samples in four

datasets, our method consistently kept the $E_G = 0$, meaning that all of our geometric errors were under the defined geometric thresholds in Eq.(17). For each dataset, we selected one example and presented them in Fig. 5, Fig. 6, Fig. 7, and Fig. 8.

Because of our proposed method and Tran's share many similarities, they can be directly compared. In Table 3, we compared the vertical disparity error $E_v(Z)$ and also each of the geometric terms including the modified aspect ratio E_{AR}, skewness E_{SK}, rotation angle E_R, and size ratio E_{SR}. In terms of vertical disparity error, our proposed method outperformed Tran's due to our FMR module, which acquired superior correspondences. Compared to our optimized correspondences, Tran's rectification module needs to handle inputs with many outliers. In terms of geometric errors, results were comparable between our method and Tran'. However, in terms of size ratio E_{SR}, Tran's method lost control in the SKKU-dFoVs and SKKU-dFoVs datasets, producing values that are exceeded the geometric thresholds in Eq.(17). Our new and updated optimization strategy play a crucial part in geometrical results. We retained the balance between vertical disparity error $E_v(Z)$ and geometric distortion to identify the optimal solution based on that definition.

Table 4 displays the running times for five benchmark methods using all 57 samples from four datasets. The FoV neutralization running times and rectification running times are shown separately. The executing times of initial SIFT and NG-RANSAC are excluded because they are the same for every benchmark method. According to the results, Dinh's method has the longest running time in the FoV neutralization step because it requires an extra SIFT operation. Because of the removal of correspondence optimization sub-module in our FoV neutralization module, our running times were slightly shorter than those produced by Tran's method. In the rectification phase, Fusiello and Dinh's method required multiple iterations to find the optimal solutions. The USR-CGD method converged extremely fast, and it is the fastest algorithm in this phase. However, Tran's method was not notably slower than USR-CGD, because they used a strategy to prevent a local optimal that cost them fewer iterations compared with USR-CGD. Our method requires longer times for rectification than other methods due to the nature of our FMR module. However, the average running time of our method was still under 1 *s*, which is tolerable for a pre-processing task that does not need to run continuously as rectification.

V. CONCLUSION

We presented an uncalibrated rectification method that can balance the trade-off between vertical disparity error and geometric errors. The built-in FMR module improved the overall quality of the input correspondences. It helped remove noisy outliers from the matching pairs to identify the most suitable set of parameters. Our proposed optimization strategy also made significant impacts on the final results. While our vertical disparity errors were superior to those of other benchmark methods, we kept the geometric errors under predefined thresholds even when the matching pairs were complex. The experiments demonstrate the advantages of our method in both common cases of uncalibrated rectification and the dissimilar FoV cases. In the future, we intend to improve our process running time while maintaining the same degree of accuracy. We also hope to work with more complex sensor setups such as autonomous vehicles and industrial robots.

REFERENCES

[1] J.-K. Oh and C.-H. Lee, "Development of a stereo vision system for industrial robots," in *Proc. Int. Conf. Control, Autom. Syst.*, 2007, pp. 659–663.
[2] J. Zhang, Z. Liu, Y. Gao, and G. Zhang, "Robust method for measuring the position and orientation of drogue based on stereo vision," *IEEE Trans. Ind. Electron.*, vol. 68, no. 5, pp. 4298–4308, May 2021.
[3] Y.-Z. Hsieh and S.-S. Lin, "Robotic arm assistance system based on simple stereo matching and Q-learning optimization," *IEEE Sensors J.*, vol. 20, no. 18, pp. 10945–10954, Sep. 2020.
[4] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong, "A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry," *Artif. Intell.*, vol. 78, nos. 1–2, pp. 87–119, 1995.
[5] C. C. Pham and J. W. Jeon, "Robust object proposals re-ranking for object detection in autonomous driving using convolutional neural networks," *Signal Process., Image Commun.*, vol. 53, pp. 110–122, Apr. 2017.
[6] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3061–3070.
[7] T. P. Nguyen, T. H.-P. Tran, and J. W. Jeon, "MultiLevel feature pooling network for uncalibrated stereo rectification in autonomous vehicles," *IEEE Trans. Ind. Electron.*, vol. 68, no. 10, pp. 10281–10290, Oct. 2021.
[8] Q. Wang, C. Zou, Y. Yuan, H. Lu, and P. Yan, "Image registration by normalized mapping," *Neurocomputing*, vol. 101, pp. 181–189, Feb. 2013.
[9] N. Ayache and F. Lustman, "Trinocular stereo vision for robotics," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 1, pp. 73–85, Jan. 1991.
[10] A. Fusiello, E. Trucco, and A. Verri, "A compact algorithm for rectification of stereo pairs," *Mach. Vis. Appl.*, vol. 12, no. 1, pp. 16–22, 2000.
[11] Q.-T. Luong and O. D. Faugeras, "The fundamental matrix: Theory, algorithms, and stability analysis," *Int. J. Comput. Vis.*, vol. 17, no. 1, pp. 43–75, 1996.
[12] R. I. Hartley, "Theory and practice of projective rectification," *Int. J. Comput. Vis.*, vol. 35, no. 2, pp. 115–127, 1999.
[13] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. New York, NY, USA: Cambridge Univ. Press, 2003.
[14] C. Loop and Z. Zhang, "Computing rectifying homographies for stereo vision," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 1999, pp. 1125–1131.
[15] M. Pollefeys, R. Koch, and L. Van Gool, "A simple and efficient rectification method for general motion," in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, vol. 1, 1999, pp. 496–501.
[16] X. Li, B. Zhang, P. V. Sander, and J. Liao, "Blind geometric distortion correction on images through deep learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4855–4864.
[17] Z. Xue, N. Xue, G.-S. Xia, and W. Shen, "Learning to calibrate straight lines for fisheye image rectification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1643–1651.
[18] J. Mallon and P. F. Whelan, "Projective rectification from the fundamental matrix," *Image Vis. Comput.*, vol. 23, no. 7, pp. 643–650, Mar. 2005.
[19] H.-H. P. Wu and Y. Yu-Hua, "Projective rectification with reduced geometric distortion for stereo vision and stereoscopic video," *J. Intell. Robotic Syst.*, vol. 42, no. 1, pp. 71–94, Jan. 2005.
[20] F. Isgro and E. Trucco, "On robust rectification for uncalibrated images," in *Proc. ICIAP*, Sep. 1999, pp. 297–302.
[21] A. Fusiello and L. Irsara, "Quasi-Euclidean epipolar rectification of uncalibrated images," *Mach. Vis. Appl.*, vol. 22, no. 4, pp. 663–670, Jul. 2011.
[22] H. Ko, H. S. Shim, O. Choi, and C.-C. J. Kuo, "Robust uncalibrated stereo rectification with constrained geometric distortions (USR-CGD)," *Image Vis. Comput.*, vol. 60, pp. 98–114, Apr. 2017.
[23] R. Xiao, W. Sun, J. Pang, Q. Yan, and J. Ren, "DSR: Direct self-rectification for uncalibrated dual-lens cameras," in *Proc. Int. Conf. 3D Vis. (3DV)*, Sep. 2018, pp. 561–569.

[24] S. Kumar, C. Micheloni, C. Piciarelli, and G. L. Foresti, "Stereo rectification of uncalibrated and heterogeneous images," *Pattern Recognit. Lett.*, vol. 31, pp. 1445–1452, Aug. 2010.
[25] M. S. S. Kumar and N. Avinash, "Stereo image rectification using focal length adjustment," in *Proc. 5th Int. Conf. Signal Image Process.*, Jan. 2014, pp. 241–249.
[26] V. Q. Dinh, T. P. Nguyen, and J. W. Jeon, "Rectification using different types of cameras attached to a vehicle," *IEEE Trans. Image Process.*, vol. 28, no. 2, pp. 815–826, Feb. 2019.
[27] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
[28] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2564–2571.
[29] T. H.-P. Tran, T. P. Nguyen, H. H. Nguyen, and J. W. Jeon, "Robust uncalibrated rectification with low geometric distortion under unbalanced field of view circumstances," *IEEE Trans. Ind. Electron.*, early access, Feb. 25, 2021, doi: 10.1109/TIE.2021.3060673.
[30] E. Brachmann and C. Rother, "Neural-guided RANSAC: Learning where to sample model hypotheses," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4322–4331.
[31] J. Moré, "The Levenberg-Marquardt algorithm: Implementation and theory," in *Numerical analysis* (Lecture Notes in Mathematics), vol. 630. Berlin, Germany: Springer, 1978, pp. 105–116.
[32] *Syntim*. Accessed: Mar. 18, 2021. [Online]. Available: http://www.iro.umontreal.ca/~mignotte/TestImages.html
[33] *MCL-RS Dataset*. Accessed: Jan. 25, 2019. [Online]. Available: http://mcl.usc.edu/mcl-rs-database

TAI HUU-PHUONG TRAN received the B.S. degree in computer science and engineering from International University—Vietnam National University, Vietnam, in 2015. He is currently pursuing the Ph.D. degree in electrical and computer engineering with Sungkyunkwan University, Suwon, South Korea.

His current research interests include image processing, computer vision, and deep learning.

HUY-HUNG NGUYEN received the B.S. degree in computer science and the M.E. degree in information technology management from International University—Vietnam National University, Vietnam, in 2014 and 2017, respectively. He is currently pursuing the Ph.D. degree in electrical and computer engineering with Sungkyunkwan University, Suwon, South Korea.

His research interests include computer vision, image processing, and deep learning.

JAE WOOK JEON (Senior Member, IEEE) received the B.S. and M.S. degrees in electronics engineering from Seoul National University, Seoul, South Korea, in 1984 and 1986, respectively, and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, USA, in 1990.

From 1990 to 1994, he was a Senior Researcher with Samsung Electronics, Suwon, South Korea. Since 1994, he has been an Assistant Professor with the School of Electrical and Computer Engineering, Sungkyunkwan University, Suwon, where he is currently a Professor with the School of Information and Communication Engineering. His current research interests include robotics, embedded systems, and factory automation.