Structural-Information-based Robust Corner Point Extraction for Camera Calibration under Lens Distortions and Compression Artifacts

SEONGHYEON KANG1, SEONG DAE KIM1, (LIFE SENIOR MEMBER, IEEE), AND MUNCHURL KIM1, (Senior Member, IEEE)
1School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon 34141, Republic of Korea
Corresponding author: Munchurl Kim (e-mail: mkimee@kaist.ac.kr).

ABSTRACT
Previous camera calibration methods often use a checkerboard to capture images and estimate the camera parameters from the correspondences between images and the checkerboard. The corner points in the checkerboard images are used as useful features for correspondence matching. Therefore, it is essential to precisely find the corner points in the checkerboard images. In many previous works, the corner points are extracted assuming that the checkerboard images are not distorted by its lens. Instead, image blurring and Gaussian noise on the images are usually considered, but other cases are not dealt with. However, the captured checkerboard images are often corrupted by lens distortions and compression artifacts, which leads to performance degradation of corner point extraction. Moreover, the corner points are extracted individually in the previous methods without considering their geometric relations. To better handle the corner point extraction problem under lens distortions, in our corner point extraction optimization, the distorted locations of the pixels on checkerboard images are corrected with the camera parameters, and the structural constraints for checkerboard image grids are then applied under the line-to-line mapping. Also, to robustly find the blurred edges between corner points due to JPEG compression, an edge surface model is newly proposed that models the transitions with over- and under-shoots around the blurred edges. Extensive experimental results show that our method significantly outperforms the state-of-the-art method with average 88.3% and 54.3% reduction in RMSE for corner point reprojection and camera parameter estimation, respectively under compression and lens distortions for synthetic and real data.

INDEX TERMS Corner point extraction, structural information, camera calibration, lens distortion, compression distortion.

I. INTRODUCTION
Calibration is the task of estimating the projection parameters of a camera. Accurate camera parameter estimation is essential in the fields where 3D information is estimated from images, such as 3D reconstruction, virtual reality, and autonomous driving. For accurate results in the fields, precise camera calibration is required. So, many studies have been made for the camera calibration [11] - [4].

The camera calibration techniques generally relate known target information to the camera parameters to find 2D-to-3D mapping which is a common issue in the fields of computer vision. Two-dimensional boards are conventional tools for the camera calibration. The 2D boards are used not only to calibrate optical cameras but also to calibrate Lidar cameras which obtain depth information [5], [6]. The pattern on the boards usually consists only of black and white colors. Feature points are easily extracted on boundaries of black and white areas from the board images. By matching the feature points on the images to the board in the real world, the camera parameters are estimated [9]. There are various patterns for the camera calibration such as a circular pattern, a checkerboard pattern, a deltite [24] pattern and a time-coded pattern [12]. Among them, the checkerboard pattern is the most widely used. The corner points of the checkerboard
images are usually extracted for the feature points. There are various methods of extracting the corner points for the camera calibration [19] - [32].

For an accurate camera calibration result, lens distortion of the camera should be taken into account. The lens distortion is one of the optical aberrations by the camera lens [14], [15]. As a result of this distortion phenomenon, the projection by the camera is not a line-to-line mapping, but a line-to-curve mapping. The complex non-linear mapping does not occur in a distortion-free camera. There are difficulties in producing a completely distortion-free lens. To reduce the lens distortion, a compound lens may be designed or an aspherical lens (whose surface profile is not a portions of a sphere) [16] are required. Placing those lenses inside a camera often requires a large space to increase the volume of the camera. Therefore, the lens distortion is inevitable for small-sized cameras without such compound lenses or aspherical lenses. Also, since the degree of distortion depends on the focal length changes [17], it is very difficult to design a distortion-free lens for a wide range of the changes in focal length. Therefore, most cameras often suffer from the lens distortion.

Despite many efforts, the studies on finding the corner points have mostly been made for images without considering compression distortion. JPEG [11] is widely used for digital cameras, which relies on 8 × 8 Discrete Cosine Transform (DCT) and quantization on the DCT coefficients. The quantization causes distortion which leads to over- and under-shoot (called Gibbs phenomenon) along the edges of the checkerboard image. This makes it difficult to find the corner points of the checkerboard image precisely, thus leading to unreliable camera calibration.

In this paper, we propose a novel and elaborate corner point extraction method that incorporates the structural information under both lens distortion and compression artifacts, which leads to outperforming the previous corner point extraction methods with significant margins.

II. RELATED WORK

Harris corner detector method [19] is the most widely known method where the gradient values of the pixels are used to determine whether the pixels belong to edges, corners, or flat areas. However, it did not provide subpixel-level coordinates of the corner points. So, some methods extended the Harris corner detector method to estimate the subpixel-level corner points [20], [21]. Harris detector finds the corner points integer-pixel levels where the patches with the corner points tend to have large eigenvalues of Hessian matrices of their pixels. Han et al. [21] extended Harris detector to find subpixel level corner points by fitting Gaussian functions centered on the estimated integer-level corner points. Sánchez et al. [20] used a secondary polynomial function instead of Gaussian functions to find sub-pixel level corner points.

Geiger’s method [22] [23] has been widely adopted in corner point extraction tools such as those in OpenCV [7] and MATLAB [8]. By using the fact that the difference vectors of two pixels on the edges are perpendicular to the gradients of the edges, the corner points by Geiger’s method are determined as the points that minimize the square sum of the inner products of the difference vectors between their neighbor pixels and corner point candidates and the gradients at the neighbor pixels. Although Geiger’s method is simple and fast for corner point extraction, the gradient values are sensitive to noise.

Saddle points in checkerboard images can be extracted as the corner points [24], [31], [32]. For the saddle points, the pixels near the corner points are fitted to a polynomial function. The polynomial surface fitting is suitable only for very small areas near the corner points, not in the region where the brightness values are steady. That is, the polynomial fitting error becomes dramatically larger as the size of the fitting surface area increases. However, with smaller fitting surface areas, the surface fitting becomes sensitive to the noise of the pixels in the areas.

Zhao et al. [26] utilized the property that the distributions of brightness values around the corner points are symmetrical. In Zhao’s method, the weights of pixels are set based on how much symmetrical the distribution of their neighboring pixels is. The weighted averages of the pixels are the corner points in Zhao’s method. The method works well when the distributions of the weights are of very sharp shapes centered at the corner points, so the weights of pixels near the corner points often have larger values, which makes it vulnerable to blurring near the corner points.

Duda et al. [28] utilized the property of lines through the corner points. In the method, the line segments of an equal length centered at a pixel location in a checkerboard image are set, and the maximum and the minimum among the sums of pixel values along the line segments are found. The difference between the two extreme values is maximized when the pixel passed by the line segments is a corner point because the line segment through the only bright areas and the line segment through the only dark areas meet at the corner point. A difference map of the two extreme values at neighboring pixel locations is near the corner point. The local maximum of the map is calculated as a corner point in subpixel-level via Gaussian peak fitting. When generating the difference map near the corner point, the pixels in the flat area away from the corner point are mostly used, rather than those near the corner point. So, the accuracy of the result may be degraded by using the pixels away from the corner point. It can also be a disadvantage that the map is composed of the maximum difference values of two extreme values which are sensitive to disturbances in pixel values.

Yang et al. [25] fitted the edges of the checkerboard images to a surface model and extracted the intersections of the edge surfaces as the corner points. The edge fitting was performed under the assumption that the edges were straight. The performance of the method might be degraded when the edges are curved by lens distortions.

If the corner points are obtained using structural information from the edges, they can be more robustly estimated. There are some methods to utilize the edges for the camera
calibration. Liu et al. [30] proposed a corner point extraction algorithm that utilized the curvature of curved edges by extending Geiger’s method. In Chen’s method [27], the curved edges in checkerboard images distorted by the lens are extracted, and the resulting corner points are determined as the intersection points of the extracted curved edges. Zhou et al. [29] proposed a method that extracted feature points on the edges, rather than the corner points, for camera calibration. In Zhou’s method, the feature points on the edges are extracted from local regions near the edges. However, the methods [30], [27], [29] handle the edges independently and do not consider the structural information of the grid formed by the edges, especially for corner point extraction.

The previous methods have extracted the corner points on the checkerboard images individually or have utilized the relations between the corner points and the edges partially. However, for better calibration, the corner points in the checkerboard images should be extracted more accurately. So, the structural property of the corner points should be considered. Unlike the previous methods, we consider the structural information such as the grid structure formed by the edges to robustly estimate corner points even under the lens distortions as well as compress distortions.

III. SUMMARY

In this paper, our contributions for corner point extraction are summarized as follows:

- We firstly propose a corner point extraction method that utilizes the structural information with the geometric property of the edges formed by the structures of the projected grids in the checkerboard images. The corner points on the checkerboard images should be intersection points of the edges, and the edges should meet at vanishing points. The cross-ratios between each checkerboard image and the checkerboard should be preserved after camera projection [10]. When extracting the corner points and the edges in the checkerboard images, these conditions must be satisfied. So, all the corner points and edges in a single checkerboard image should be expressed by parameters of the camera projection, rather than being treated independently.

- For corner point extraction, the lens distortions are taken into account in conjunction with the structural constraints. Camera projection under the lens distortions results in a line-to-curve mapping, which causes the corner point extraction to be more complicated. With estimated parameters for the lens distortions, we correct the distorted checkerboard images and apply the structural constraints under the line-to-line mapping. We estimate the camera projection parameters which determine the edges and the corner points, and the distortion parameters to make the line-to-line mapping work well.

- We propose a novel edge surface model to robustly find the corner points of the edges corrupted by JPEG compression. Since, by the JPEG compression, high-frequency components of the checkerboard images are discarded due to quantization of their DCT coefficients, the corner points and the edges of the checkerboard images are damaged due to the blurriness (transition) with over/under-shoot along the edges, called Gibbs phenomenon. Our edge surface model takes into account Gibbs phenomenon in precisely finding the edges, thus leading to accurate estimation of corner points.

IV. CAMERA MODEL

We used a camera model with the lens distortion. The pin hole model [10] is described as follows:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = K[R|T]\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

$$D = \{k_1, k_2, t_1, t_2\}$$ where $K$ is an internal camera calibration matrix with a focal length $f$, the number of pixels per unit distance in image coordinate $(s_x, s_y)$, a principal point $(p_x, p_y)$, and a skew term $\alpha$. $R$ and $T$ are external parameters which are a rotation matrix and a translation vector, respectively. $(X, Y, Z)$ is the location of a 3D point, and $(x, y)$ is the location of a 2D point projected from the 3D point to an image plane.

Lens distortion is a kind of optical aberration, and a straight object is bent in an image due to the distortion. There are two main distortions: radial and tangential distortions [14], [15]. The radial distortion is caused by the refraction of the lens. The tangential distortion occurs when the image plane and the lens are not in parallel. A camera model with the lens distortions [33] is given as:

$$\hat{x}_d = \hat{x}(1 + k_1r^2 + k_2r^4) + 2t_1\hat{x}\hat{y} + t_2(r^2 + 2\hat{x}^2)$$

$$\hat{y}_d = \hat{y}(1 + k_1r^2 + k_2r^4) + 2t_2\hat{x}\hat{y} + t_1(r^2 + 2\hat{y}^2)$$

where $r^2 = \hat{x}^2 + \hat{y}^2$ and $(\hat{x}, \hat{y})$ is a coordinate on a 2D plane in the 3D real world, which is projected from a 3D point via $[\hat{x} \, \hat{y} \, 1]^T = [R|T]\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$. $(\hat{x}_d, \hat{y}_d)$ is a coordinate on a distorted 2D plane in the 3D real world by the lens distortion with a set of distortion coefficients $D = \{k_1, k_2, t_1, t_2\}$ where $k_1, k_2$ indicate the coefficients of radial distortion, and $t_1, t_2$ are the coefficients of tangential distortion. Finally the coordinate $(x_d, y_d)$ in a distorted image is defined by transforming $(\hat{x}_d, \hat{y}_d)$ as:

$$[x_d \, y_d \, 1]^T = K[\hat{x}_d \, \hat{y}_d \, 1]^T$$

We also consider a distortion model for fish-eye lens cameras which are designed to create ultra-wide-angle views [18]. The fish-eye lens distortion model [7] is described as:

$$\theta = \tan^{-1}(r)$$

$$\theta_d = \theta(1 + \rho_1\theta^2 + \rho_2\theta^4 + \rho_3\theta^6 + \rho_4\theta^8)$$

$$\hat{x}_d = (\theta_d/r)\hat{x}, \quad \hat{y}_d = (\theta_d/r)\hat{y}$$
where \( r^2 = \hat{x}^2 + \hat{y}^2 \) and \((\hat{x}, \hat{y})\) is a coordinate on a 2D plane in the 3D real world, which is projected from a 3D point via \( [\hat{x} \hat{y} 1]^T = [R|T][X Y Z 1]^T \). \((x_d, y_d)\) is a coordinate on a distorted 2D plane in the 3D real world by the fish-eye lens distortion with a set of distortion coefficients \( D_{fe} = \{\rho_1, \rho_2, \rho_3, \rho_4\}\).

V. PROPOSED METHOD FOR PRECISE CORNER-POINT ESTIMATION

Fig. 1 shows the block diagram of our corner point extraction-based camera calibration. For each checkerboard image, its corner points are extracted via the corner point extraction block that utilizes the structural information of the checkerboard grid under the lens and compression distortions. Based on the estimated corner points of all the checkerboard images, the camera calibration is performed via the Camera Parameter Estimation block using Zhang’s method [9]. In Zhang’s method, the corner points on a checkerboard are projected to the checkerboard images by the estimated camera parameters. The camera parameters are found in the sense that a cost function is minimized. The cost function is defined as the square sum of the distances between the estimated corner points and the projected points. Next, we explain our proposed edge surface model under the JPEG compression distortion, the structural information, and the lens distortion for precise corner point extraction.

A. AN EDGE SURFACE MODEL

The corner points are the most basic features of checkerboard images for camera calibration. Many studies have performed camera calibration based on corner points [9], [22], [24], [30], [31], [32]. However, image blur makes it difficult to find corner points accurately. To remedy this, Yang et al. [25] used an error function model [13] to handle the Gaussian blurred surfaces along the edges of the checkerboard images. Chen et al. [27] applied a morphological filter and extracted subpixel-level coordinates of edges for polynomial fitting, thus finding the intersections of the edges as corner points. However, the checkerboard images are corrupted from JPEG compression by which the surfaces around the edges tend to have transitions with over- and under-shoots thus leading to imprecise findings of edges and corner points. This compression distortion is mostly originated from the quantization of DCT coefficients. Fig. 2 shows an example of over- and under-shoot of pixel values across an edge of the

![FIGURE 1. A scheme of our corner point extraction for camera calibration. The CB stands for a checkerboard. We first estimate \( K_i, D_i, H_i \) of each image and extract its corner points of a checkerboard image via the projection. The parameters, \( K_i, D_i \) are not for the camera calibration, but only corner-point extraction. \( H_i \) is a homography and \( A \) is a set of steepness factors of the edge surface model of (6).](image_url)

![FIGURE 2. Gibbs phenomenon along an edge (blue line \( \gamma \) in (a)) in a checkerboard image due to JPEG compression: (a) A cropped region with JPEG compression artifacts; (b) A 3D mesh plot of the pixels in the cropped region; (c) A 1-D profile (cross-section) of the 3-D mesh plot in (b) with over- and under-shoots for which a blue solid curve fits the pixel values across the edge. The pixels between the two dashed red lines in (a) are the pixels for the edge surface model fitting. Note that the pixels values in the bright areas and the dark areas are normalized to 1 and -1 respectively. So, the normalized pixel values of over- and under-shoot pixels are above 1 and below -1, respectively.](image_url)
checkboard image. Such pixel variations around the edge in Fig. 2 make it difficult to find the edge and its corner points accurately. In this paper, we first consider the JPEG compression distortion as well as lens distortions for edges in the corner point extraction problem.

In order to robustly find the corner points under JPEG distortion, we design a novel edge surface model to be applied for the vicinity zone \((Z_v)\) of each edge as shown in Fig. 2. We first perform normalization of the pixel values in \(Z_v\) to estimate the parameter values of our edge surface model. The normalization procedure is as follows: (i) As shown in Fig. 3, the pixels with small fluctuations of the values in each bright area (a blue-dotted rounded box), which are located in a distance from the edges, are modeled as a surface by a second-order polynomial, as denoted by \(F_W(x, y)\). The distance is defined to get pixels on flat areas whose values are little damaged by the compression. Since the edge widths in our checkerboard images are often smaller than 5 pixels, we set the distance value to 5. Similarly, the pixels with small fluctuations in a dark area (a red-dotted rounded box) are modeled also as a surface using another second-order polynomial, as denoted by \(F_B(x, y)\); (ii) Once we get the two surface models \((F_W(x, y), F_B(x, y))\), we normalize the pixel values of the pixels in \(Z_v\) by expanding \(F_W(x, y)\) and \(F_B(x, y)\) into \(Z_v\) where \((x, y)\) \(\in Z_v\) and applying them as follows:

\[
G(\mathbf{p} = [x, y]^T) = \frac{2V(\mathbf{p}) - F_W(\mathbf{p}) - F_B(\mathbf{p})}{F_W(\mathbf{p}) - F_B(\mathbf{p})} \tag{5}
\]

where \(\mathbf{p} = [x, y]^T\) is a pixel location in \(Z_v\), and \(V(\mathbf{p})\) is the pixel value at \(\mathbf{p}\). It can be noted in Eq. (5) that, if \(V(\mathbf{p}) = F_W(\mathbf{p})\), \(V(\mathbf{p})\) is normalized to 1 while it is normalized to -1 if \(V(\mathbf{p}) = F_B(\mathbf{p})\). Then, the normalized over- and under-shoot pixel values are larger than 1 and smaller than -1, respectively, which are the red dots in an 1-D profile of the normalized pixels as shown in Fig. 2c.

Our edge surface model under the JPEG compression distortion is defined in \(Z_v\) as:

\[
\phi(a, d = [p^T 1]d) = S(ad + \sin(ad))/\pi \tag{6}
\]

where \(S\) is an amplitude of over/under-shoot, \(a\) is a steepness factor of the edge surface on the edge surface model, \(d\) is a distance in pixel unit between a coordinate of pixel \(\mathbf{p} = [x, y]^T\) and line \(\gamma = [\gamma_x, \gamma_y, \gamma]^T\) where \(\gamma_x x + \gamma_y y + \gamma = 0\) and \(\gamma_x^2 + \gamma_y^2 = 1\). As shown in Fig. 2, the over- and under-shoot values are located at \(d = \pi/a\) and \(-\pi/a\), respectively, as in (6) and edge surface model \(\phi\) is defined between \(d = -\pi/a\) and \(+\pi/a\). The pixels (outside \(Z_v\)) from line \(\gamma\) in distances of \(|d| > \pi/a\) are discarded for the parameter estimation of edge surface model. The estimation of edge surface model parameters \((a, \gamma)\) can be found within our corner point extraction optimization according to (9) in the following subsection. The values of \(S\) are the mean values of absolute over-/under-shoot values. Note that the gradient values of the edge surface model near \(d = 0\) are almost constant along the edge. As the \(|d|\) increases, the magnitudes of the gradient values of the edge surface model decrease so as to fit the model to the over/under-shoot. In our experiments, we also compare the edge surface model in Eq. (6) to a simple version with a plane which is given by

\[
\phi(a, d = [p^T 1]d) = ad \tag{7}
\]

where \(Z_v\) is defined as \(|d| < \pi/a\).

For an \(N_X \times N_Y\) checkerboard grid, we have total \(N_{esm} (= N_X (N_Y - 1) + N_Y (N_X - 1))\) edge surface models since there is one edge surface model for the edge between two corner points. For a vertical line of the checkerboard, there are \(N_v (= N_Y - 1)\) edge surface models on \(N_v\) edges corresponding to the vertical line. If there is no lens distortion, the edge surface models corresponding to a vertical line share a single \(\gamma\) because the edges are generated by projections of the single vertical line in checkerboard. The edge surface models with the other vertical lines are the same. This constraint is also applied equally to the edge surface models corresponding to the horizontal lines of the checkerboard.

B. STRUCTURAL INFORMATION FOR EDGE SURFACE MODEL PARAMETER ESTIMATION

In our method, corner points are found not independently but jointly by utilizing structural information of the checkerboard grid. The independent findings of corner points are vulnerable to local damages due to compression distortions, which causes degradation of camera calibration performance. Instead, the structural information can be effectively used to robustly find the corner points where the entire corner points and edges are jointly found over a whole checkerboard image, which can overcome the local distortions.

The structural information we are dealing with is originated from the grid of the checkerboard. Actually, with
the lens distortion, we have to deal with complex nonlinear
relations for the structural constraints: (i) The corner points
on the checkerboard images should be the intersection points
of the edges; (ii) the edges should meet at vanishing points;
and (iii) The cross-ratios between each checkerboard image
and the checkerboard should be preserved after camera pro-
jection. We first consider a distortion-free camera to simplify
the corner point extraction problem and expand it for the
camera with the lens distortions in the next subsection.

The corner points in the checkerboard images should be
intersection points of the $\gamma$’s in (6). So, rather than finding
the corner points directly from the checkerboard images,
the $\gamma$’s are obtained from the checkerboard images, and
the intersections of the $\gamma$’s are set as the corner points.
The structural constraints of the $\gamma$’s include that the $\gamma$’s
meet at vanishing points, and the cross-ratios between each
checkerboard image and the checkerboard are preserved.
These constraints on $\gamma$’s are from the fact that the $\gamma$’s on
each checkerboard image are projected from the grid lines
of the checkerboard by a single homography. So, to satisfy
the constraints, all the $\gamma$’s in each checkerboard image are
expressed as

$$\gamma_n = H^{-T}l_n \quad (8)$$

where $l_n$ is the $n$-th line in the checkerboard, $n = 1, \ldots, N_X+N_Y$, and $H \in \mathbb{R}^{3 \times 3}$ is a homography [10] with
8 parameters. From the edge surface model in (6) and the
$\gamma$’s in (8), the corner points can be found via the following
optimization of the corner point extraction as

$$\begin{align*}
(H_i, A_i = \{a_{n,m}^i\}) &= \arg \min_{H \in \mathcal{H}, A} \sum_{n=1}^{N_L} \sum_{m=1}^{M_n} \sum_{p \in J_{n,m}} \left\| \phi_{n,m}^i(a_{n,m}^i, pT^{-1}1)[\gamma_n^i] - G^i(p) \right\|_2^2 \\
\text{subject to } \gamma_n^i &= H^{-T}l_n \quad (9)
\end{align*}$$

where $n$ is an index of the lines on the checkerboard including
vertical and horizontal lines, $m$ is an index of the edges
along the $n$-th line where an edge is defined as a line segment
between two corner points on a same line, $N_L (=N_X+N_Y)$ is
the total number of lines on a checkerboard, and $M_n (=N_X-1
or N_Y-1)$ is the number of the edge surface models along the
$n$-th line. $\phi_{n,m}^i(\cdot)$ and $a_{n,m}^i$ indicate the $m$-th edge surface
model and its steepness factor, respectively, along the $n$-th line
in the $i$-th checkerboard image. $J_{n,m}$ is a set of pixels in $\mathbb{Z}_0$
(See Fig. 3) and $G^i(p)$ is the normalized pixel value by
Eq. (5) at pixel location $p$ in the $i$-th checkerboard image. It
should be noted that the $S$ value of each edge surface model
is computed after the pixels within a predefined distance ($\pm 1.5$
pixels) around an edge are assigned to $J_{n,m}$ before solving
the corner point extraction optimization problem in (9). The
$S$ value for each edge surface model is determined as the
mean value of absolute over- and under-shoot pixel values
for $J_{n,m}$ (See Fig. 2c).

In conclusion, from (9), the corner point extraction op-
timization forces all the $\gamma$’s and the corner points to be
precisely estimated by obeying the structural constraints.

C. CONSIDERATION OF LENS DISTORTIONS

Without any lens distortion, the line-to-line projection can
be done by a homography in (8) at camera projection.
However, under lens distortion, (8) is no longer valid. So,
the distorted locations of the pixels by the lens distortions
need to be corrected before applying the homography in (8).
When estimating $H$ and $A = \{a_{n,m}\}$, we have to find
the camera parameters $K$ and the distortion coefficients $D$, both of which are related to the lens distortions [33]. After
the distorted pixel locations are corrected, the curved edges
become straight lines, and the straight lines are projected
from the checkerboard grid by $H$. The desired parameters
for corner points can be estimated via the optimization based
on Levenberg–Marquardt (LM) method [34], [35] as

$$\begin{align*}
(K_i, D_i, H_i, A_i = \{a_{n,m}^i\}) &= \arg \min_{K, D} \sum_{n=1}^{N_L} \sum_{m=1}^{M_n} \sum_{p \in J_{n,m}} \left\| \phi_{n,m}^i(a_{n,m}^i, [pT^{-1}1][\gamma_n^i] - G^i(p_c) \right\|_2^2 \\
\text{subject to } \gamma_n^i &= H^{-T}l_n \quad (10)
\end{align*}$$

where $p_c$ is a corrected location of $p$ by $K_i$ and $D_i$ for the
$i$-th checkerboard image. $G^i_c(p_c)$ is a normalized pixel value at
$p_c$ of the corrected $i$-th checkerboard image. The others are
the same as those in (9). Through the optimization in (10),
$K_i$, $D_i$, $H_i$ and $A_i$ are estimated. The cost function of our
method converges well in the optimization. So the damping
factor used in the LM method is empirically set to be small,
$\lambda = 10^{-20}$, with its associated constant, $\nu = 20$.

For the optimization, we need to estimate the internal pa-
rameter $K$, a set of distortion coefficients $D$, a homography
$H$, and a set of steepness factors $A$. The initial values of
the homography are estimated by matching the edge pixels
on the checkerboard images with the known lines of the
checkerboard. Please note that the distortion is ignored in this

![FIGURE 4. Our proposed process of camera parameter estimation with $K$, $D$, and $H$ for corner-point extraction. $L$ is a set of lines on the checkerboard. $E$ refers to the cost function (10).](image-url)
case. The estimated homography is not accurate, but good enough for the initial guess. The principle point of $K$ is set to be the center of images. The skew term $\alpha$ in (1) is set to zero. Then the remaining parameters in (1) are $f_x$, $f_y$, $s_x$, and $s_y$ which are the products of the focal lengths and the number per unit distance. Also, we assume $s_x = s_y = s$. In order to find $f$, we use the following relation among the internal/external parameters and the homography as:

$$H = \eta K [r_1 \ r_2 \ T]$$  \hspace{1cm} (11)

For the equation above, there are 8 unknowns (a scale factor $\eta$, $f$ and $s$ in $K$, $\theta_x$, $\theta_y$, $\theta_z$ in $R = [r_1 \ r_2 \ r_3]$, and $t_x$, $t_y$, $t_z$ in $T$), but we have 9 equations. Note that $r_1$ is a $3 \times 1$ vector whose elements are functions of $\theta_x$ and $\theta_y$, and $r_2$ is a $3 \times 1$ vector whose elements are functions of $\theta_x$, $\theta_y$, and $\theta_z$ [36]. So, we can determine the internal parameter $f \cdot s$ which is the only needed variable to compute the initial values. The initial distortion coefficients for $D = \{k_1, k_2, t_1, t_2\}$, in (2) and $D_{fe} = \{\rho_1, \rho_2, \rho_3, \rho_4\}$ in (4) are all set to zeros. Each of the steepness factors $A = \{a_{i,m,n}\}$ in (10) initially tries with 0.5 and -0.5 for which the one that results in a lower cost value is used as its initial value.

Based on these estimated camera parameters from (10), we can find the corner points on the $i$-th checkerboard image by projecting the corner points on the checkerboard (See Fig. 4). With the corner points of all checkerboard images, we finally estimate the camera internal parameters $K$, the lens distortion coefficients $D$, and the external parameters, $\{R_i\}$, $\{T_i\}$ using Zhang’s method [9] as shown in Fig. 1.

### VI. EXPERIMENTAL RESULTS

Experiments are conducted with synthetic and real data. The eight previous methods are compared with ours, which include Geiger’s [22], Ma’s [23], Ha’s [24], Yang’s [25], Zhao’s [26], Chen’s [27], Duda’s [28], and Zhou’s [29] methods. In Zhou’s [29] method, the points along the edges are extracted for camera calibration. We used a built-in function in MATLAB for Geiger’s method, and an open source code of Ha’s method. The codes for the other methods are implemented by ourselves.

A. EXPERIMENTS WITH SYNTHETIC DATA

We conducted two experiments with the synthetic data. One is an ablation test to see the effectiveness of each contribution in our method for corner point extraction performance and the camera calibration. The other one is the comparison with the previous methods. For the comparison test, cameras with and without a fish-eye lens are utilized.

The dimension of a square of the checkerboard is 40mm × 40mm, and the number of the corner points is 56 ($= 7 \times 8$). The checkerboard is located approximately 800mm away from the camera. The ground truth parameters of the non-fish-eye lens camera are $s_x = 1,280$, $s_y = 1,260$, $\alpha = 1$, $p_x = 510$, $p_y = 355$, $k_1 = -0.15$, $k_2 = -0.01$, $t_1 = -0.015$, and $t_2 = 0.01$. For the fish-eye lens camera, $\rho_1 = -2.4$, $\rho_2 = -0.18$, $\rho_3 = -0.12$, and $\rho_4 = -0.036$. The internal parameters of the fish-eye lens camera are the same as the internal parameters of the non-fish-eye lens camera. The image size is 1,000×700. A total of 100 checkerboard images was used where 20 checkerboard images were randomly selected for the cameras calibration by repeating the camera calibration 40 times.

To evaluate the accuracy of the estimated corner points, the root mean square errors (RMSE) are measured in pixel units as

$$E_C = \sqrt{\frac{1}{N_C} \sum_{i=1}^{N_C} ||c_i - c_i^{gt}||^2_2}$$  \hspace{1cm} (12)

where $N_C$ is the total number of the corner points of all checkerboard images, $c_i$ and $c_i^{gt}$ are the $i$-th estimated and ground-truth corner points, respectively.

By the camera calibration, we estimate $K$, $D$, $\{R_i\}$, and $\{T_i\}$. To evaluate the accuracy of the parameters, the error of the reprojected corner points are measured in pixel unit as

$$E_P = \sqrt{\frac{1}{N_P} \sum_{i=1}^{N_P} \sum_{j=1}^{N_i} ||f_P(P_j, K, D, R_i, T_i) - P_{ij}^{gt}||^2_2}$$  \hspace{1cm} (13)

where $P_j$ is the $j$-th corner points on the checkerboard in the real world, $P_{ij}^{gt}$ is ground truth coordinate of the $j$-th corner points on the $i$-th checkerboard image, $N_I$ is the number of images for the camera calibration, $N_P$ is the number of corner points on the checkerboard, and $f_P$ returns coordinate of projections of $P_j$ by $K$, $D$, $R_i$, and $T_i$ according to Eq. (2).

The $K$, and $D$ are the results of the camera calibration which are the camera’s unique characteristic values. So, to see the performance of each camera calibration method, these parameters should be evaluated. The errors of the estimated $K$’s, and $D$’s are measured in pixel unit as

$$E_K = \sqrt{\frac{1}{N_X} \sum_{i=1}^{N_X} ||f_K(K_i, K, D) - f_K(X_i, K^{gt}, D^{gt})||^2_2}$$  \hspace{1cm} (14)

where $f_K(X, K, D)$ returns 2D coordinates of a projection of 3D points, $\{X\}$, in a region of interest (ROI) by $K$, and $D$ according to Eq. (2) with $R = I$ and $T = 0$. When we evaluate Zhou’s method with the others, only these evaluation method is utilized because the corner points are not extracted in Zhou’s method.

1) Ablation Study on Proposed Method

Our contributions include 1) the edge surface model to handle the JPEG compression distortion for corner point extraction and 2) the utilization of structural information to deal with the lens distortion. To see the effectiveness of the two contributions, we inspect the four variants (A, B, C and D) of our methods as shown in Table 1. The two variants (A and C) use the plane model in Eq. (7) instead of the edge surface model in Eq. (6). The other two variants (A and B) do not rely on the structural information. They assume that the edges are
straight, and each edge surface is individually estimated from the pixels around each edge. So, the intersection lines of the surfaces are searched and the intersections of the lines are extracted as the corner points. Note that the variant D is our full model with the two contributions that are optimized in Eq. (10). The quality factor (Q) of JPEG compression is set to 40 for experiments. The camera for the experiments is the non fish-eye lens camera.

**Table 1.** Four Model Variants for Effectiveness Tests with edge surface model and Structural Information

| Model variants | A   | B   | C   | D   |
|----------------|-----|-----|-----|-----|
| Edge surface model for JPEG | x   | O   | O   | O   |
| Structural information | x   | x   | O   | O   |

Table 2 shows the performances of the four variants. As shown in Table 2, the variants C and D show significantly lower $E_C$, $E_P$, and $E_K$ values compared to the other two variants A and B that do not incorporate the structural information for corner point extraction. This implies that our proposed usage of structural information plays a significant role in precise corner point extraction. In comparison between two variants A (C) and B (D), the variant B (D) has lower $E_C$, $E_P$, and $E_K$ values than the variant A (C). From this, our edge surface model works positively to further improve the corner point extraction performance. Among those variants, the variant D is our final proposed method (Proposed-ESM). ESM stands for the edge surface model. We will compare it with the previous methods in the next subsection. The variant C (Proposed-Plane) is also used for the comparison test. The model parameters of ‘Proposed-ESM’ and ‘Proposed-Plane’ are found according to our optimization in Eq. (10).

**Table 2.** Performance Comparisons in RMSE of Four Model Variants for Estimated Corner Points, Reprojection of checkerboard Corner Points and Estimated Camera Parameters

| Model variants | A   | B   | C   | D   |
|----------------|-----|-----|-----|-----|
| $E_C$          | 0.2709 | 0.2582 | 0.0181 | 0.0165 |
| $E_P$          | 0.0706 | 0.0702 | 0.0118 | 0.0107 |
| $E_K$          | 1.9021 | 1.7372 | 0.3765 | 0.3241 |

2) Performance Comparison with Previous Methods

In comparison with the previous methods, to see the robustness of our method to the compression distortion, we used various Q values of JPEG compression and various lens distortion coefficients. Note that the lower Q values result in more compression distortions with smaller visual quality. The camera for the experiments is the non fish-eye lens camera.

Table 3 compares the performances of corner point extraction methods for various Q values. In Table 3, our methods, ‘Proposed-Plane’ and ‘Proposed-ESM’ indicate a simple model with the plane in Eq. (7) and the edge surface model in Eq. (6), respectively. As shown in Table 3, our proposed methods yield significantly lower RMSE values for all Q values with at least one order in magnitude, compared to the other methods. Table 4 shows errors of reprojection of checkerboard corner points. Since the reprojected corner points of all the methods satisfy the structural information, less errors tend to occur than those in Table 3. The proposed methods also appear less than the other methods. Fig. 5 shows the scatter plots of reprojection errors for the estimated corner points by the eight corner point extraction methods for Q = 40. As shown, our proposed method with edge surface model clearly outperforms all the other methods with the smallest error. Also, a simplified version of our method, which uses a plane model instead of edge surface model, also significantly outperforms all the previous methods. The dispersions of the reprojected corner points in Fig. 5 are measured as the standard deviation values in Table 4. Table 5 shows the performance of camera calibration for various methods. As shown, both of our methods, ‘Proposed-plane’ and ‘Proposed-ESM’ outperform all the previous methods with large margins for all Q values. The superiority of our ‘Proposed-ESM’ method comes from the utilization of structural information for corner point extraction and the edge surface model to cope with the compression distortion. For smaller Q values, the ‘Proposed-ESM’ more significantly outperforms the ‘Proposed-Plane’ because the ‘Proposed-Plane’ suffers from over- and under-shoot problems of pixels in $Z_0$ due to more severe JPEG distortions. Fig. 6 shows the accuracy maps of camera calibration based on the estimated $K$’s and $D$’s and Table 6 shows the numerical RMSE errors of the estimated camera parameters for Q = 40. As shown, our methods clearly outperform all the previous methods for camera calibration with help of more precise corner point extraction performance.

Tables 7, 8 and 9 show the performances of the corner point estimation methods for Q = 40 under various lens distortions with $k_1 = -0.15n$, $k_2 = -0.01n$, $t_1 = -0.015n$ and $t_2 = 0.01n$ for $n = 1, 2, 3$ and 4. As shown, our methods show the superiority in corner point extraction, thus leading to more accurate camera calibration performances compared to the other methods in all cases. It is worthwhile to note that Yang’s method exhibited the highest RMSE although it incorporates an error friction surface model into the corner point extraction problem. This is because it does not take into the lens distortions for the resulting curved edges.

Table 10 shows the performances of the corner point estimation for the synthetic images obtained for the fish-eye lens camera. The Q value is set to 40 for the test. The fish-eye lens often causes to generate large lens distortions with wide angle views. Even for this situation, it can be seen in Table 10 that our method outperforms all the other methods in RMSE. As shown, Chen’s method shows the worst performance because their method is based on polynomial curve fitting to the edges of checkerboard images and is not sufficient to model the curved edges caused by the fish-eye lens.

Our method shows fast convergence with about 25 iterations, and takes To process each checkerboard image of 1,000x700 resolution, our method takes about 3.05 seconds in MATLAB7.31 running on a PC platform with Intel
TABLE 3. RMSE Performance for corner points estimation ($E_C$)

| Q  | 20   | 40   | 60   | 80   |
|----|------|------|------|------|
| Geiger’s [22] | 0.1985 | 0.1364 | 0.1368 | 0.1422 |
| Ma’s [23] | 0.1957 | 0.1445 | 0.1238 | 0.1138 |
| Ha’s [24] | 0.1512 | 0.1118 | 0.1049 | 0.1011 |
| Yang’s [25] | 0.2666 | 0.2624 | 0.2610 | 0.2260 |
| Zhao’s [26] | 0.3563 | 0.2988 | 0.2921 | 0.2852 |
| Chen’s [27] | 0.2702 | 0.1896 | 0.1563 | 0.1294 |
| Duda’s [28] | 0.1835 | 0.1545 | 0.1532 | 0.1456 |
| Proposed-Plane | 0.0225 | 0.0181 | 0.0174 | 0.0157 |
| Proposed-ESM | 0.0204 | 0.0165 | 0.0156 | 0.0140 |

TABLE 4. RMSE and standard deviation (std) for reprojections of checkerboard corner points ($E_P$)

| Q  | 20   | 40   | 60   | 80   |
|----|------|------|------|------|
| Geiger’s [22] | 0.0553 | 0.0520 | 0.0335 | 0.0326 |
| Ma’s [23] | 0.0499 | 0.0290 | 0.0356 | 0.0320 |
| Ha’s [24] | 0.0413 | 0.0403 | 0.0305 | 0.0283 |
| Yang’s [25] | 0.0638 | 0.0580 | 0.0574 | 0.0405 |
| Zhao’s [26] | 0.0823 | 0.0801 | 0.0642 | 0.0586 |
| Chen’s [27] | 0.0568 | 0.0520 | 0.0423 | 0.0391 |
| Duda’s [28] | 0.0666 | 0.0549 | 0.0522 | 0.0400 |
| Proposed-Plane | 0.0145 | 0.0131 | 0.0118 | 0.0101 |
| Proposed-ESM | 0.0124 | 0.0120 | 0.0107 | 0.0105 |

TABLE 5. RMSE Performance for projected points by $K$’s, and $D$’s ($E_K$)

| Q  | 20   | 40   | 60   | 80   |
|----|------|------|------|------|
| Geiger’s [22] | 0.9499 | 1.1488 | 0.7664 | 0.6466 |
| Ma’s [23] | 1.0417 | 0.8644 | 1.0762 | 0.4423 |
| Ha’s [24] | 0.9593 | 0.8916 | 0.7118 | 0.4776 |
| Yang’s [25] | 1.4119 | 1.3228 | 0.9458 | 0.8112 |
| Zhao’s [26] | 1.5924 | 1.6867 | 1.6314 | 1.0632 |
| Chen’s [27] | 1.1122 | 1.0055 | 0.7466 | 0.5256 |
| Duda’s [28] | 2.0391 | 1.7004 | 1.3984 | 1.3805 |
| Zhou’s [29] | 1.4561 | 0.9617 | 1.0088 | 0.6465 |
| Proposed-Plane | 0.4359 | 0.3765 | 0.4108 | 0.3234 |
| Proposed-ESM | 0.3534 | 0.3241 | 0.4108 | 0.3234 |

TABLE 6. RMSE accuracy for the estimated camera parameters

| $f_x$ | $f_y$ | $p_x$ | $p_y$ | $\alpha$ | $k_1$ | $k_2$ | $t_1$ | $t_2$ |
|-------|-------|-------|-------|----------|-------|-------|-------|-------|
| Geiger’s [22] | 0.3615 | 0.4263 | 0.5614 | 0.5762 | 0.3989 | 0.0038 | 0.0542 | 7.90e-05 |
| Ma’s [23] | 0.3620 | 0.3591 | 0.6055 | 0.6033 | 0.2535 | 0.0029 | 0.0353 | 4.76e-05 |
| Ha’s [24] | 0.3387 | 0.4532 | 0.5981 | 0.4806 | 0.3395 | 0.0036 | 0.0547 | 4.05e-05 |
| Yang’s [25] | 0.6408 | 0.6845 | 0.9169 | 1.1044 | 0.5771 | 0.0072 | 0.0834 | 1.35e-04 |
| Zhao’s [26] | 0.5017 | 0.5243 | 1.1785 | 1.1891 | 0.4728 | 0.0090 | 0.1168 | 2.04e-04 |
| Chen’s [27] | 0.4181 | 0.4308 | 0.8098 | 0.6074 | 0.3399 | 0.0049 | 0.0622 | 7.44e-05 |
| Duda’s [28] | 0.4732 | 0.4563 | 0.8500 | 1.1370 | 0.3929 | 0.0055 | 0.0704 | 2.57e-04 |
| Zhou’s [29] | 0.3288 | 0.3945 | 0.8119 | 0.8796 | 0.3225 | 0.5985 | 0.0751 | 0.0453 |
| Proposed - plane | 0.1041 | 0.0815 | 0.2585 | 0.2268 | 0.1151 | 0.0010 | 0.0149 | 2.19e-05 |
| Proposed - ESM | 0.0859 | 0.0691 | 0.2109 | 0.2252 | 0.1037 | 0.0009 | 0.0142 | 2.45e-05 |
FIGURE 5. Scatter plots of reprojection errors for the estimated corner points.

FIGURE 6. Accuracy maps for camera calibration based on estimated $K$’s and $D$’s by the methods under comparison. The 2D points in the range of $x=250\sim750$ and $y=175\sim525$ in a synthesis image are back-projected to 3D space by ground truth $K$ and $D$, and the 3D points are projected to the synthesis image plane by the estimated $K$’s and $D$’s. After that, the absolute position errors are calculated as an accuracy plane.
TABLE 7. RMSE Performance for corner points estimation ($E_C$)

| $k_1$, $k_2$, $t_1$, $t_2$ | n=1 | n=2 | n=3 | n=4 |
|-----------------------------|-----|-----|-----|-----|
| Geiger’s [22]              | 0.1364 | 0.1660 | 0.1688 | 0.1638 |
| Ma’s [23]                  | 0.1445 | 0.1778 | 0.1638 | 0.1453 |
| Ha’s [24]                  | 0.1118 | 0.1447 | 0.1245 | 0.1358 |
| Yang’s [25]                | 0.2624 | 0.2168 | 0.2831 | 0.3267 |
| Zhao’s [26]                | 0.2988 | 0.3174 | 0.3210 | 0.3337 |
| Chen’s [27]                | 0.1896 | 0.2074 | 0.1957 | 0.2028 |
| Duda’s [28]                | 0.1545 | 0.1808 | 0.1621 | 0.1866 |
| Proposed-Plane             | 0.0181 | 0.0207 | 0.0210 | 0.0212 |
| Proposed-ESM              | 0.0165 | 0.0163 | 0.0188 | 0.0206 |

TABLE 8. RMSE Performance for reprojection of checkerboard corner points ($E_R$)

| $k_1$, $k_2$, $t_1$, $t_2$ | n=1 | n=2 | n=3 | n=4 |
|-----------------------------|-----|-----|-----|-----|
| Geiger’s [22]              | 0.0335 | 0.0297 | 0.0393 | 0.0411 |
| Ma’s [23]                  | 0.0356 | 0.0317 | 0.0278 | 0.0374 |
| Ha’s [24]                  | 0.0305 | 0.0245 | 0.0340 | 0.0353 |
| Yang’s [25]                | 0.0574 | 0.0339 | 0.0667 | 0.0767 |
| Zhao’s [26]                | 0.0642 | 0.0669 | 0.0638 | 0.0644 |
| Chen’s [27]                | 0.0423 | 0.0390 | 0.0475 | 0.0498 |
| Duda’s [28]                | 0.0522 | 0.0629 | 0.0574 | 0.0674 |
| Proposed-Plane             | 0.0118 | 0.0129 | 0.0128 | 0.0132 |
| Proposed-ESM              | 0.0107 | 0.0113 | 0.0121 | 0.0138 |

TABLE 9. RMSE Performance for projected points by $K$’s, and $D$’s ($E_K$)

| $k_1$, $k_2$, $t_1$, $t_2$ | n=1 | n=2 | n=3 | n=4 |
|-----------------------------|-----|-----|-----|-----|
| Geiger’s [22]              | 1.1488 | 1.2017 | 1.0522 | 1.2673 |
| Ma’s [23]                  | 0.8644 | 0.9289 | 0.6208 | 1.1171 |
| Ha’s [24]                  | 0.8916 | 0.5361 | 1.0179 | 1.2016 |
| Yang’s [25]                | 1.3228 | 1.3712 | 1.6093 | 1.9180 |
| Zhao’s [26]                | 1.6867 | 1.5406 | 1.6911 | 1.7180 |
| Chen’s [27]                | 1.0055 | 0.9354 | 1.2260 | 1.3140 |
| Duda’s [28]                | 1.7004 | 1.6789 | 1.6786 | 1.5091 |
| Zhou’s [29]                | 0.8441 | 0.8153 | 0.8243 | 0.8416 |
| Proposed-Plane             | 0.3765 | 0.5776 | 0.3677 | 0.4814 |
| Proposed-ESM              | 0.3241 | 0.5350 | 0.3394 | 0.4371 |

TABLE 10. Performance Comparisons in RMSE of the fish-eye lens camera.

| $E_C$ | $E_R$ | $E_K$ |
|-------|-------|-------|
| Geiger’s [22] | 0.0885 | 0.0217 | 4.4374 |
| Ma’s [23] | 0.0782 | 0.0248 | 4.4465 |
| Ha’s [24] | 0.1560 | 0.0465 | 4.6448 |
| Yang’s [25] | 0.2579 | 0.0624 | 4.3644 |
| Zhao’s [26] | 0.1153 | 0.0374 | 3.9279 |
| Chen’s [27] | 0.8341 | 0.4254 | 8.8653 |
| Duda’s [28] | 0.1931 | 0.0505 | 5.4406 |
| Zhou’s [29] | 3.4234 |
| Proposed-Plane | 0.0321 | 0.0158 | 2.9473 |
| Proposed-ESM | 0.0247 | 0.0146 | 2.8473 |

Core i7 CPU of 3.50 GHz and 16.0GB RAM. It is about 5 times longer compared to Ma’s method. This complexity overhead comes from the fact that, in each iteration of the optimization, our method updates $K$ and $D$, and corrects the locations of edge pixels with the parameters.

TABLE 11 shows the reconstruction performances in terms of RMSE in millimeter units. Our methods outperform all the other methods by yielding more accurate camera parameter estimations that lead to more accurate cuboid reconstruction.

B. EXPERIMENTS WITH REAL DATA

Experiments with real data are often difficult to evaluate estimated parameters because the true parameter values are not available. To overcome this difficulty, we evaluate the camera parameters by reconstructing a cuboid with four checkerboards on its sides (Fig. 7b). The number of corner points in the checkerboard for calibration (Fig. 7a) is $110 (= 10 \times 11)$, and the dimension of a checkerboard square is $38mm \times 38mm$. From a single cuboid image with two checkerboards (Fig. 7b), we can reconstruct two facets of the cuboid image (Fig. 7c) based on the estimated camera parameters. For the cuboid images, we have taken four images for the cuboid in approximately 1.2m distance away from the camera. The angles and rotations of the cuboid are somewhat restricted to have similar areas for the two faces of the cuboid, as shown in Fig. 7b. So, angles and rotations were determined such that each of the four cuboid images includes two facets of the cuboid and the boundary between the two facets, as shown in Fig. 7b.

Evaluation can be performed in terms of distances in RMSE between corner points after reconstruction. A CamelTM’s WC100 camera is used. The size of the image is $1,920 \times 1,080$. A total of 50 checkerboard images was used where 20 checkerboard images were randomly selected for each camera calibration by repeating the camera calibrations 40 times.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
values along the edges of checkerboard images due to compression distortion. Furthermore, the structural information of the corner points and the edges from the checkerboard grid is utilized to precisely find the corner points. Extensive experimental results have verified that our method outperformed all previous methods with significantly large margins, especially for corner point estimation on synthetic data. This has led to more accurate camera calibration.

As future work, it might be worthwhile to explore our edge surface model for different image data such as multispectral color images having blurred corners from compression.

REFERENCES

[1] C. Zhang, T. Huang, and Q. Zhao, “A new model of RGB-D camera calibration based on 3D control field,” Sensors, vol. 19, pp. 5082-1 - 5082-19, Nov. 2019.
[2] Z. Gao, Y. Gao, Y. Su, Y. Liu, Z. Fang, Y. Wang, and Q. Zhang, “Stereo camera calibration for large field of view digital image correlation using zoom lens,” Measurement, vol. 185, pp. 100999-1 - 100999-8, Aug. 2021.
[3] G. H. An, S. Lee, M. W. Seo, K. Yun, W. S. Cheong, and S. J. Kang, “Charuco board-Based omnidirectional camera calibration method,” Electronics, vol. 7, pp. 421-1 - 421-15, Dec. 2018.
[4] B. Fu, Y. Wang, X. Ding, Y. Jiao, L. Tang, and R. Xiong, “LiDAR-camera calibration under arbitrary configurations: observability and methods,” IEEE Transactions on Instrumentation and Measurement, vol. 69, issue 6, June 2020.
[5] Q. Liao, Z. Chen, Y. Liu, Z. Wang, and M. Liu, “Extrinsic calibration of lidar and camera with polygon,” 2018 IEEE International Conference on Robotics and Biomimetics, pp. 200-205, Dec. 2018.
[6] J. K. Huang, and J. W. Grizzle, “Improvements to target-based 3d lidar to camera calibration,” IEEE Access, vol. 8, pp. 134101-134110, July 2020.
[7] OpenCV Library. Available online: https://opencv.org/ (accessed on Sep. 2021).
[8] J.Y. Bouguet. Camera calibration toolbox for Matlab, available from: https://www.vision.caltech.edu/bouguet/calib_doc/ (accessed on Sep. 2021).
[9] Z. Zhang, “A flexible new technique for camera calibration,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, pp. 1330-1334, Nov. 2000.
[10] R. Hartley, and A. Zisserman, “Camera geometry and single view geometry,” in Multiple view geometry in computer vision, Cambridge university press, 2004, Part 1, pp. 151-236.
[11] G. K. Wallace, “The JPEG still picture compression standard,” IEEE Transactions on Consumer Electronics, vol. 38, pp. xviii-xxiv, Feb. 1992.
[12] M. Große, M. Schaffer, B. Harendt, and, R. M. Kowarschik, “Camera calibration using time-coded planar patterns,” Optical Engineering, vol. 51, no. 8, pp. 083604-1 - 083604-9, Aug. 2012.
[13] G. Boeze, “An a priori estimate for the truncation error of a continued fraction expansion to the Gaussian error function” Computing, vol. 29, pp. 135-152, Mar. 1982.
[14] C. B. Duane, “Close-range camera calibration,” Photogrammetric Engineering, vol. 37, no. 8, pp. 855-866, Jan. 1971.

**TABLE 11. RMS reconstruction errors of a 3D cuboid (in millimeter)**

| Method       | Reconstruction errors |
|--------------|-----------------------|
| Geiger’s [22]| 1.6237                |
| Ma’s [23]    | 1.8190                |
| Ha’s [24]    | 1.5099                |
| Yang’s [25]  | 1.7429                |
| Zhao’s [26]  | 1.5233                |
| Chen’s [27]  | 1.7376                |
| Duda’s [28]  | 1.8919                |
| Zhou’s [29]  | 1.8456                |
| Proposed-Plane| 1.2011               |
| Proposed-ESM | **1.1752**            |

[15] J. Sasián, “The wave aberration function,” in Introduction to Aberrations in Optical Imaging System, Cambridge, Cambridge University Press, 2012, Chapter 5, pp. 67-75.
[16] Z. Zhuang, X. Dailaur, J. Parent, P. Roullet, and S. Thibault, “Geometrical-based quasi-aspher surface description and design method for miniature, low-distortion, wide-angle camera lens,” Applied Optics, vol. 59, no. 27, pp. 8408-8417, Sept. 2020.
[17] S. S. Jeong, S. K. Woo, and J. Heo, “Empirical modeling of lens distortion in change of focal length,” Journal of the Korean Society of Surveying, Geodesy, Photogrammetry and Cartography, vol. 26, issue 1, pp. 93-100, Feb. 2008.
[18] C. Hughes, M. Glinv, E. Jones, and P. Denny, “Review of geometric distortion compensation in fish-eye cameras,” IET Irish Signals and Systems Conference, pp. 162-167, June 2008.
[19] C. Harris and M. Stephens, “A combined corner and edge detector,” Proceedings of the 4th Alvey Vision Conference, vol. 15, no. 50, pp. 147-151, Aug. 1988.
[20] J. Sánchez, N. Monzon, and A. Salgado, “An analysis and implementation of the Harris corner detector,” Image Processing On Line, vol. 8, pp. 305-328, Oct. 2018.
[21] Y. Han, P. Chen, and T. Meng, “Harris corner detection algorithm at sub-pixel level and its application,” International Conference on Computational Science and Engineering, pp. 133-137, July 2015.
[22] A. Geiger, F. Moosmann, O. Car, and B. Schuster, “Automatic camera and range sensor calibration using a single shot,” International Conference on Robotics and Automation, pp. 3936-3943, May, 2012.
[23] C. Ma, F. Meng, Y. Guo, Li. Y, and Z. Zhan, “An improved method of corner point extraction used in grating projection phase measurement,” International Society for Optics and Photonics, vol. 11567, pp. 115673-1 - 115673-8, Nov. 2020.
[24] H. Ma, M. Perdoch, H. Alismail, I. S. Kweon, and Y. Sheikh, “Deltile grids for geometric camera calibration,” Proceedings of the IEEE International Conference on Computer Vision, pp. 5344-5352, Oct. 2017.
[25] T. Yang, Q. Zhao, X. Wang, and Q. Zhou, “Sub-pixel chessboard corner localization for camera calibration and pose estimation,” Applied Sciences, vol. 8, no. 11, pp. 2118-1 - 2118-19, Nov. 2018.
[26] Q. Zhao, Z. Chen, T. Yang, and Y. Zhao, “Detection of sub-pixel chessboard corners based on gray symmetry factor,” Ninth International Symposium on Precision Engineering Measurement and Instrumentation, vol. 9446, pp. 94464S-1 - 94464S-7, Mar. 2015.
[27] Y. Chen, F. Huang, F. Shi, B. Liu, and H. Yu, “Plane chessboard-based calibration method for a LWIR ultra-wide-angle camera,” Applied optics, vol. 58, pp. 744-751, Feb. 2019.
[28] A. Duda, and U. Frese, “Accurate detection and localization of checkerboard corners for calibration,” British Machine Vision Conference, pp. 126.1-126.10, Newcastle, UK, Sept. 2018.
[29] F. Zhou, Y. Cui, H. Gao, and Y. Wang, “Line-based camera calibration with lens distortion correction from a single image,” Optics and Lasers in Engineering, vol. 51, pp. 1332-1343, Dec. 2013.
[30] J. Liu, Z. Yang, H. Huo, and T. Fang, “Camera calibration method with checkerboard pattern under complicated illumination,” Journal of Electronic Imaging, vol. 27, issue 4, pp. 0430380-1 - 0430380-11, July 2018.
[31] S. Placht, P. Fursattel, A. E. Mengue, H. Hofmann, C. Schaller, M. Balda, and E. Angelopoulou, “ROCHADE: Robust checkerboard advanced detection for camera calibration,” European Conference on Computer Vision, vol. 8692, pp. 766-779, Sept. 2014.
[32] L. Luccchese, and S. K. Mitra. “Using saddle points for sub-pixel feature detection in camera calibration targets,” In Proceedings of the Asia-Pacific Conference on Circuits and Systems, vol. 2, pp. 191–195, Oct. 2002.
[33] J. Heikkila, and O. Silvén, “A four-step camera calibration procedure with implicit image correction,” Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1106-1112, June 1997.
[34] K. Levenberg, “A method for the solution of certain non-linear problems in least squares,” Quarterly of Applied Mathematics, vol. 2, no. 1, pp. 164-185, July 1944.
[35] D. W. Marquardt, “An algorithm for least-squares estimation of nonlinear parameters,” Journal of the Society for Industrial and Applied Mathematics, vol. 11, no. 2, pp. 431-441, June 1963.
[36] J. Diebel, “Representing attitude: Euler angles, unit quaternions, and rotation vectors,” Matrix, vol. 58, pp. 1-35, Oct. 2006.