photonics

MDPI

Article

YOLO-OBB and Two-Stage Geometric Correction for RGB-LED Array Optical Camera Communication

Jiaqi Ju [1], Pan Qiu [1], Yipeng Tan [1] and Zhengguang Shi [1,2,*]

[1] School of Science, Shanghai Institute of Technology, Shanghai 201418, China; jujiaqi@sit.edu.cn (J.J.); 246182120@mail.sit.edu.cn (P.Q.); 246182125@mail.sit.edu.cn (Y.T.)
[2] School of Art and Design, Shanghai Institute of Technology, Shanghai 201418, China
* Correspondence: shizhengguang@sit.edu.cn

Abstract

In Optical Camera Communication (OCC), precise localization of LED arrays under complex tilt conditions is a core challenge for reliable decoding. This paper proposes an OCC reception scheme for RGB-LED arrays that integrates YOLO-OBB rotated object detection with two-stage geometric correction. The system first employs a YOLOv8n-OBB model to extract a quadrilateral region of interest that tightly encloses the LED array boundary. This effectively suppresses background interference caused by superimposed perspective tilt and in-plane rotation. A coarse-to-fine two-stage correction framework is then applied. The first stage rapidly eliminates the dominant perspective distortion based on the detected bounding-box corners. The second stage performs a refined correction using the actual LED center positions. Two homography matrices are cascaded into a combined transformation, achieving two-stage correction accuracy through a single coordinate mapping. In the corrected image, K-Means clustering constructs a 16 × 16 LED topological grid. A locking strategy is adopted so that subsequent frames skip repeated LED detection and clustering. The steady-state per-frame processing time is reduced to approximately 78.9 ms. Experiments covered 16 cross-combinations of vertical tilt from 0° to 45° (0°, 15°, 30°, 45°) and in-plane rotation from 0° to 40° (0°, 15°, 30°, 40°). The uncorrected scheme and the horizontal-box scheme experienced severe bit errors or complete failure under complicated distortion. The proposed scheme maintained error-free transmission under all 16 tested conditions. The ratios of opposite sides of the corrected LED grid remained stable between 0.997 and 1.004. The system simultaneously achieves high reliability and low-latency real-time processing under complex geometric distortions.

Keywords: optical camera communication; LED array; YOLO-OBB; K-Means clustering

Check for updates

Received: 8 May 2026
Revised: 7 June 2026
Accepted: 18 June 2026
Published: 20 June 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Optical Camera Communication (OCC) is an important branch of Optical Wireless Communication (OWC) [1,2]. It employs Light-Emitting Diode (LED) arrays as transmitters and cameras as receivers. Compared with radio-frequency communication, OCC offers abundant spectrum resources, immunity to electromagnetic interference, and the ability to serve illumination simultaneously. In recent years, OCC has attracted wide attention in indoor positioning [3], vehicular networks [4], and Internet of Things sensing [5,6]. In 2018, the IEEE 802.15.7-2018 standard incorporated OCC into the short-range optical wireless communication specification system [7]. Visible Light Communication (VLC) systems use photodiodes as receivers and demodulate continuous photoelectric signals directly. In

contrast, OCC receivers must first accomplish spatial localization of LED elements from two-dimensional images and then recover data through color or brightness identification [8]. For OCC systems based on LED arrays, the element density is high and each LED element occupies a small image area. Existing research indicates that spatial localization error is a significant factor affecting OCC system reliability [9], and the limited spatial resolution is considered one of the primary challenges [10].

In practical deployment, it is difficult to maintain strict perpendicular alignment between the camera optical axis and the LED array plane. Two common types of geometric distortion pose serious threats to localization accuracy. The first is perspective tilt around the vertical axis (Yaw), which causes the originally rectangular LED panel to exhibit trapezoidal distortion in the image, where the far end appears smaller. Under this distortion, the row and column spacing of LED elements is no longer uniform, and the imaging size and brightness of elements at the far end are noticeably smaller than those at the near end [11]. The second is in-plane rotation of the LED panel (Roll), which introduces an angle between the LED row direction and the horizontal reference line of the image [12]. These two distortions often occur simultaneously. The spatial distribution of LED elements is then jointly affected by perspective distortion and rotational transformation, making it impossible for a fixed uniform grid to align with the actual element positions. Consequently, conventional schemes based on fixed-grid sampling produce systematic center deviations, leading to large-area color misclassification and, in severe cases, complete failure of communication synchronization [13].

To address the two major challenges of LED array localization and geometric correction, researchers have explored the problem from multiple perspectives. Early methods relied on techniques such as grayscale thresholding and circularity filtering to extract LED contours. These methods are effective under normal alignment conditions, but when distortion increases or background illumination becomes complex, the spot morphology undergoes severe deformation, and the accuracy of contour extraction degrades significantly [14]. With breakthroughs in deep learning for object detection, models such as YOLO and SSD have been introduced to replace manual ROI selection [15,16]. Such detectors output axis-aligned horizontal bounding boxes (HBB). When the LED panel undergoes in-plane rotation, the upper and lower edges of the horizontal box will include stray pixels from the desktop or walls [17]. These additional background pixels generate dense noise in the subsequent thresholding and contour detection steps, severely reducing the signal-to-noise ratio of valid LED elements. OBB predicts quadrilaterals with rotation angles, whose four edges tightly enclose the physical boundaries of the target. Therefore, OBB holds a principled advantage in scenarios involving both rotation and perspective distortion [18]. However, research on OCC reception systems that deeply integrate OBB detection with geometric correction and real-time decoding remains scarce.

In the geometric correction stage, the fundamental tool for removing perspective distortion is the planar homography transformation [19]. Traditional approaches compute the perspective matrix in a single step using the four corners of the detection box or the LED panel boundary, mapping the distorted quadrilateral to a standard rectangle [20,21]. This single-stage strategy performs reasonably well under small distortions. When the tilt angle increases or multiple distortions superimpose, however, the coarse corner information struggles to achieve sufficient alignment accuracy, and the positional deviation continuously degrades color recognition accuracy [22]. K-Means clustering, as a lightweight unsupervised learning method, has demonstrated promising real-time potential in LED array topology construction [23,24]. Nevertheless, if the entire pipeline from LED detection to cluster-based localization is executed in every frame, the cumulative time severely limits the maximum achievable communication frame rate [25]. Some studies have attempted

to reduce the computational load by tracking LED positions using template matching or Kalman filtering [26,27]. These methods require independent modeling for each LED element, leading to high system complexity and tracking drift under severe distortion.

Various strategies have been explored to enhance the geometric robustness of OCC systems. In 3D modeling, Liu et al. [28] remained confined to simulations, while Palitharathna et al. [29] relied on hardware-intensive liquid lenses. Other receiver-side detection or tracking schemes (Sitanggang et al. [30]; Shin and Jang [31]; Song et al. [32]) typically overlook composite perspective and in-plane rotation distortions. Meanwhile, transmitter-side pre-correction (Zhang et al. [33]) requires signal modifications, limiting its generalizability across unmodified LEDs. Consequently, a practical, software-only receiver architecture that actively rectifies composite Yaw-Roll distortions is still lacking. To address this gap, we propose a robust software-defined OCC receiver integrating YOLOv8n-OBB detection, cascaded two-stage homography correction, and K-Means grid locking. The main contributions of the system are as follows:

First, a YOLOv8n-OBB rotated object detector is introduced to extract a clean ROI that contains almost no background pixels in composite distortion scenarios. Experiments show that under extreme conditions of 45° tilt combined with in-plane rotation, the OBB scheme operates stably, whereas the horizontal-box scheme fails in detection due to excessive background noise.

Second, a cascaded framework of coarse-to-fine correction is designed. Two homography matrices are multiplied to obtain a unified combined transformation, achieving correction accuracy through a single mapping. After correction, the LED grid achieves sub-pixel alignment.

Third, K-Means clustering is adopted to construct a 16 × 16 LED topological grid, and a locking mechanism is introduced. After locking, subsequent frames skip grayscale segmentation and repeated clustering, significantly reducing the computational time and lowering the per-frame processing latency to the millisecond level.

Fourth, the proposed scheme is benchmarked against mainstream deep-learning-based OCC architectures under identical channel conditions and compared quantitatively with recent SOTA works, validating its superior robustness and scientific advancement.

The rest of this paper is organized as follows: Section 2 describes the transmitter frame structure and the receiver processing pipeline. Section 3 presents the core algorithm design. Section 4 reports the experimental setup and comparative result analysis. Section 5 concludes the paper.

2. System Model and Working Principle

2.1. Transmitter Design and Frame Structure

The transmitter employs a 16 × 16 RGB LED array (WS2812, WorldSemi Co., Ltd., Shenzhen, China) with a total of 256 independently controllable light-emitting units. Each data LED can be individually set to one of four colors: red (R), green (G), blue (B), or white (W). These four colors correspond to 2-bit information, as follows: 00 → R, 01 → G, 10 → B, and 11 → W. Calibration LEDs use green (G) and blue (B), each representing 1 bit (G → 0, B → 1). Page-number LEDs use red (R) and green (G), each representing 1 bit (R → 0, G → 1). The array is divided into 15 pure data rows and one hybrid function row (the 16th row), as illustrated in Figure 1.

Each row carries four ASCII characters and is divided into two coding groups. Each group uses seven consecutive LEDs to encode two characters. The most significant bit of each standard ASCII character is truncated because it always represents 0, and then seven bits are left. The two resulting 7-bit sequences together form a 14-bit data stream. Starting from the most significant bit, every two adjacent bits are mapped to the color of one LED

in the group. This interleaved coding scheme allows seven LEDs to carry two characters simultaneously. At the end of each group (columns 8 and 16), a check LED is placed. If the total number of logic "1" bits among the 14 data bits is even, the check LED displays green (G); otherwise, it displays blue (B).

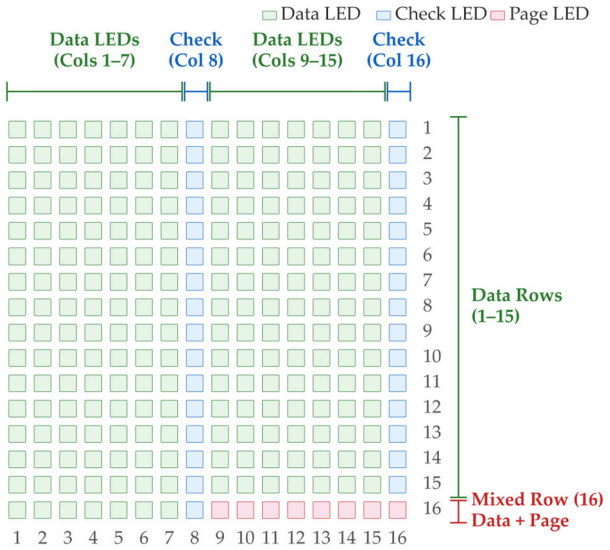


Figure 1. Logical frame structure of the 16 × 16 LED array.

The 16th row is a hybrid function row. The first seven LEDs carry the last two characters using the same coding scheme. The eighth LED serves as the check bit. LEDs in columns 9 through 16 represent an 8-bit binary page number, using only red (R) and green (G).

The transmitter refreshes at a frame rate of 5 fps with a frame dwell time of 0.2 s. Each frame transmits a total of 473 bits: data LEDs account for 434 bits (217 LEDs × 2 bits each); parity-check LEDs account for 31 bits (31 LEDs × 1 bit each); and page-number LEDs account for 8 bits (8 LEDs × 1 bit each). The total throughput (physical layer) is 2365 bps, calculated as 473 bits/frame × 5 frames/s. During BER evaluation, the transmitter cyclically sends a fixed, known test sequence. The receiver pre-calculates the expected color of every LED according to the coding rules and constructs a ground-truth matrix. The cumulative BER is then obtained by comparing each frame against this ground truth on an element-by-element basis. This approach prevents the BER evaluation from being coupled to multi-page message decoding logic, so that the results purely reflect the performance of the physical channel and the receiver algorithm.

2.2. Receiver Processing Pipeline

The receiver uses a Hikrobot MV-CU120-10UC industrial camera (Hikrobot Co., Ltd., Hangzhou, China). The image acquisition resolution is set to 640 × 640 pixels, and the camera is connected to a computer via a USB 3.0 interface. The software environment is based on Python 3.9. Image processing relies on the OpenCV 4.11.0 and scikit-learn libraries; deep learning inference is based on the Ultralytics YOLOv8 framework.

The receiver processing pipeline is shown in Figure 2. To reduce the computational overhead, the system adopts a two-stage execution model. The initial calibration stage builds a stable LED grid. Once the "locked" stage is entered, the grid and transformation matrices are reused, and several modules are bypassed, thereby substantially shortening the per-frame processing time.

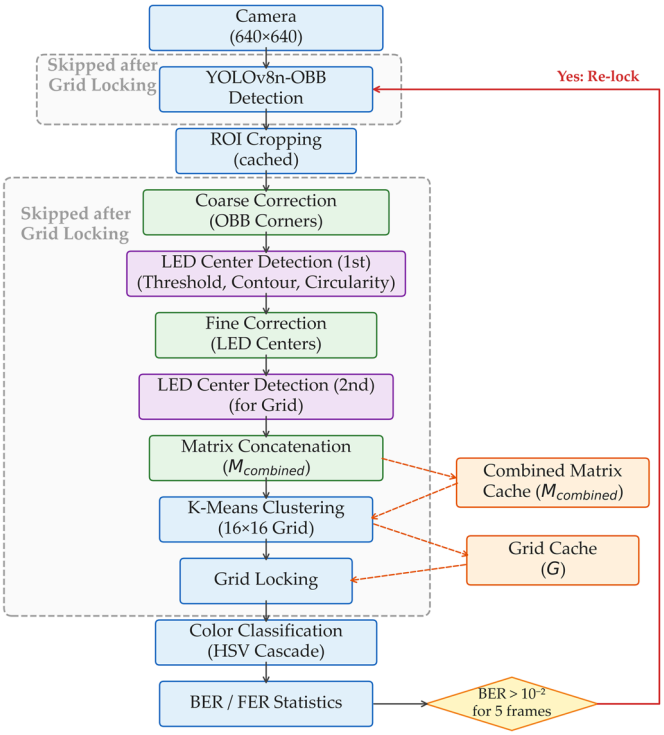


Figure 2. Proposed OCC receiver pipeline. Solid arrows indicate the main data flow, while dashed arrows indicate control flow. Different colors distinguish module types: blue blocks represent main processing modules; green blocks represent correction-related modules; purple blocks represent LED center detection modules; orange blocks represent cached data/state. The two dashed boxes enclose modules that are bypassed once the grid is locked. The yellow diamond with an orange border, together with the red arrow, indicates the re-locking mechanism, which is triggered when the bit error rate (BER) exceeds 10^{-2} for five consecutive frames.

During the calibration stage, each raw image is first fed into the YOLOv8n-OBB model, which outputs the four corner coordinates of the oriented bounding box of the LED array. The system then crops the ROI accordingly. A coarse perspective correction is subsequently performed on the ROI using the OBB corner points to re-align the tilted LED panel. After the coarse correction, the first LED center detection is executed. Sub-pixel center coordinates are obtained through grayscale thresholding, contour extraction, and circularity filtering. The fine-correction module then uses these centers to compute a second homography matrix. This matrix is multiplied by the coarse-correction matrix to obtain the combined transformation matrix M_{combined}. A second LED center detection is then performed on the fine-corrected image, yielding a more accurate set of center coordinates. These coordinates are fed into K-Means clustering to build a 16 $\times$ 16 LED topological grid. Once the grid passes quality validation, it is locked. From this point onward, all of the modules enclosed by the two dashed boxes in Figure 2 are skipped. These modules include YOLO inference, coarse correction, the two LED center detection passes, fine correction, matrix concatenation, and K-Means clustering. ROI cropping continues to execute, but uses the previously cached OBB coordinates and no longer relies on new YOLO detections. The combined matrix M_{combined} and the grid G are read directly from the cache modules on the right side and applied to every subsequent frame.

The color classification module samples HSV values at each grid intersection after locking. A cascaded decision tree based on the HSV color space is adopted. The first level eliminates unlit LEDs according to their brightness. The second level separates white LEDs from colored LEDs according to saturation. The third level classifies colored LEDs as red, green, or blue according to their hue. A 16 $\times$ 16 color matrix, i.e., the decoded data frame,

is thus obtained. The recognition result of each frame is compared element-by-element with the ground-truth matrix, and the cumulative BER and frame error rate are updated in real time.

The processing paths before and after locking differ substantially. Before locking, every frame requires YOLO inference, two perspective transformations, grayscale segmentation, contour filtering, two-dimensional clustering, and color classification. After locking, only ROI cropping (using cached coordinates), a single combined-matrix mapping, fixed-grid sampling, and color classification remain, greatly reducing the computational overhead.

3. Core Algorithm Design

3.1. YOLO-OBB-Based Rotated Detection and ROI Extraction

The front-end of the system employs a YOLOv8n-OBB rotated object detector. Compared with axis-aligned horizontal bounding boxes, the rotated quadrilateral predicted by OBB tightly encloses the boundary of the LED array, even under composite distortion. This effectively prevents background pixels from entering the subsequent grayscale segmentation and contour detection steps, thereby eliminating the root cause of the clustering failure that occurs in horizontal-box schemes due to the inclusion of background noise.

The YOLOv8n-OBB detector used in this work was originally designed for oriented object detection in aerial and remote sensing imagery, where targets appear at arbitrary angles. In this paper, we adapt it to OCC by training the model to locate the four corner points of the LED array, producing an oriented bounding box that tightly encloses the array boundary even under in-plane rotation and perspective tilt. This yields a clean region of interest (ROI) without background interference, a key advantage over horizontal bounding boxes. However, the detector alone only provides a clean ROI; it does not correct geometric distortion. The novelty of our work is that we then feed this clean ROI into a cascaded two-stage homography correction module, which actively corrects the distorted LED array (i.e., restores it to a frontal, regularly spaced grid) under composite Yaw-Roll distortion. After correction, the LED grid becomes aligned to sub-pixel accuracy. A grid locking mechanism further eliminates repeated detection and correction in subsequent frames, achieving low steady-state latency and error-free transmission under the tested distortion conditions.

The order of the four corner points output by the OBB detector is not determined. The subsequent coarse correction stage requires the corners to be arranged in the order top-left, top-right, bottom-right, bottom-left so that source and destination points are correctly indexed. An extremum-based sorting strategy is adopted. The corner that minimizes $x + y$ is assigned as top-left; the one that maximizes $x + y$ as bottom-right; the one that maximizes $x - y$ as top-right; and the one that minimizes $x - y$ as bottom-left. This rule forces the corners into the sequence top-left $\rightarrow$ top-right $\rightarrow$ bottom-right $\rightarrow$ bottom-left, which establishes a deterministic correspondence with the four vertices of the destination rectangle used in coarse correction and ensures the correctness of the homography solution.

After sorting, the axis-aligned bounding rectangle of the rotated quadrilateral is cropped as the ROI. The sorted corner coordinates are transformed into the ROI-local coordinate system and passed to the coarse correction module described in Section 3.2.

The detection model is trained on a self-built dataset. The dataset consists of 446 images of the 16 $\times$ 16 LED array, split into training (90%) and validation (10%) sets. Each image was manually annotated with oriented bounding boxes (OBB), and the labels were stored as a class_id followed by four normalized corner coordinates ($x1$, $y1$, $x2$, $y2$, $x3$, $y3$, $x4$, $y4$). Data augmentation followed the training configuration: HSV color space augmentation (hue shift $\pm$1.5%, saturation shift $\pm$70%, value shift $\pm$40%), translation ($\pm$10%), scaling ($\pm$50%), horizontal flipping (50% probability), and mosaic (100% probability, disabled in

the last 10 epochs). No rotation, shear, perspective, or mixup augmentations were applied. The dataset covers the full range of Yaw and Roll angles used in our experiments (Yaw 0–45°, Roll 0–40°). The dataset is not publicly available but can be obtained from the corresponding author upon reasonable request. The model achieves an mAP@0.5 of 0.995 on the validation set, with both precision and recall at approximately 1.00. This detection accuracy satisfies the requirements of the subsequent correction and clustering tasks.

3.2. Cascaded Two-Stage Perspective Correction

The angular distortion between the LED array and the camera can be described by the perspective projection model $p = K[R \mid t]P$, where K is the camera intrinsic matrix, R represents Yaw and Roll rotations, and t is a translation vector. The planar homography H used in our cascaded correction directly follows from this 3D geometry.

Conventional geometric correction for OCC mostly adopts a single-stage strategy, which performs the perspective transformation in one step using either the bounding-box corners or the LED corners directly. The single-stage approach is acceptable when the distortion is mild. Under large-angle composite distortion, however, it faces a fundamental trade-off: detection-box corners are easy to obtain but lack sufficient spatial accuracy, whereas true LED corners offer higher precision but are difficult to extract reliably from a distorted image.

The proposed framework resolves this dilemma by cascading a coarse correction stage and a fine correction stage. The coarse stage uses OBB corners to rapidly remove the dominant perspective distortion, thereby providing a favorable initial condition for LED corner detection. The fine stage further improves the accuracy of extracted LED centers. The two stages are complementary, addressing the problems of "undetectable" and "imprecise," respectively.

Coarse correction. The sorted OBB corners are taken as the source control points, denoted c_1, c_2, c_3, c_4, corresponding to the top-left, top-right, bottom-right, and bottom-left corners. The destination rectangle vertices are determined from the average edge lengths, with a margin of 30 pixels kept on all four sides. The coarse homography matrix H_{coarse} is solved from the four point correspondences:

$$d_i = H_{\text{coarse}} \cdot c_i, i = 1, 2, 3, 4 \tag{1}$$

Applying H_{coarse} to the ROI image I_{ROI} yields the coarsely corrected image I_{coarse}:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = H_{\text{coarse}} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \; I_{\text{coarse}}(x', y') = I_{\text{ROI}}(x, y). \tag{2}$$

Fine correction. A set of LED centers P is extracted from I_{coarse}. The same method is used to obtain the four corner points, denoted c'_1, c'_2, c'_3, c'_4. To enhance boundary robustness, each edge of the source quadrilateral is extended outward by 1.2 times the average LED spacing. The destination rectangle is set to 1.1 times the dimensions of I_{coarse}. The fine homography matrix H_{fine} is then solved and applied to generate the fine-corrected image I_{fine}.

Matrix concatenation. Two successive resampling operations would introduce accumulated interpolation errors and increase the computational cost. To avoid this, the two homography matrices are combined according to their application order:

$$H_{\text{combined}} = H_{\text{fine}} \cdot H_{\text{coarse}} \tag{3}$$

After this concatenation, any source point x can be mapped with a single matrix multiplication:

$$x' = H_{\text{combined}}x, \ where \ x = \left[x, y, 1\right]^{\top}, \ x' = \left[x', y', 1\right]^{\top}. \tag{4}$$

Once the grid is locked, H_{combined} is cached and reused. Every subsequent frame only needs to apply this single mapping to achieve the two-stage correction accuracy.

The framework also provides geometric calibration metrics. The four edges of the corrected LED grid quadrilateral are extracted: the top edge L_t, the bottom edge L_b, the left edge L_l, and the right edge L_r. The ratios of opposite sides are defined as

$$R_h = \frac{L_t}{L_b}, R_v = \frac{L_l}{L_r} \tag{5}$$

The center offset is defined as $d = \left\| c_{\text{grid}} - c_{\text{image}} \right\|_2$, i.e., the Euclidean distance between the grid center and the image center. Under ideal correction, $R_h \approx R_v \approx 1$ and $d \approx 0$.

3.3. LED Center Detection and Grid Locking

LED center detection is performed on the fine-corrected image. The grayscale image is binarized with a fixed threshold $T_g = 20$. This threshold is determined from the statistical distribution of the background grayscale in the ROI when the LEDs are off, and it effectively separates the dark background from the lit LED elements. Connected components are filtered by area and circularity. The area threshold is $A > 2$ pixels, and the circularity threshold is $C > 0.5$. After filtering, sub-pixel center coordinates are computed using spatial image moments. Based on the area distribution, outliers are removed by retaining only those points whose areas fall within $\mu A \pm 4\sigma A$, yielding the valid center set P_{valid}.

P_{valid} is then grouped into a 16×16 regular grid by K-Means clustering. K-Means partitions the point set into K clusters by minimizing the within-cluster sum of squared errors:

$$J = \sum_{k=1}^{K} \sum_{p_i \in S_k} \|p_i - \mu_k\|^2 \tag{6}$$

where S_k is the set of points assigned to the k-th cluster and μ_k is the cluster center. In this system, $K = 16$. Clustering is first performed along the vertical axis to obtain 16 rows, and then a second clustering is performed along the horizontal axis within each row to obtain 16 columns. The number of K-Means iterations is fixed at 10. Missing rows are filled by interpolation using the spacing of adjacent rows. Grid construction is considered to have failed if the number of valid points is less than half of the total, or if more than two rows are missing. After clustering, the grid intersections closely coincide with the actual center positions of the LED elements.

Once the grid passes quality validation, the system enters the locked state. The grid is denoted $G = \{g_{ij}\}$, where $g_{ij} = (x_{ij}, y_{ij})$ represents the expected image coordinate of the LED at logical position (i, j) in the fine-corrected image. This grid is stored in the cache. Thereafter, the per-frame processing is simplified to four operations: cropping the ROI (using cached coordinates), applying H_{combined}, sampling HSV values at the fixed grid positions, and performing color classification followed by a BER update. All LED detection and clustering operations are bypassed. If the BER exceeds a preset threshold over several consecutive frames, the system automatically triggers grid re-locking.

Re-locking mechanism. After grid locking, the system monitors the bit error rate (BER) on a per-frame basis. If the BER exceeds 10^{-2} for five consecutive frames, the lock is

considered invalid. The system then automatically exits the locked state and restarts the full initialization pipeline (YOLO detection, LED center detection, K-Means clustering, and color classification). The measured average recovery time for this complete re-initialization is 453.01 ms on the hardware platform described in Section 4.4. During the recovery process, a few frames may be lost, but subsequent decoding resumes correctly. Under static geometric distortion, long-term operation (over 4 h) showed no observable grid drift. The stability under dynamic perturbations (e.g., slight camera movement) has not been systematically evaluated in this study and remains a direction for future investigation.

3.4. Color Classification

The color classification module samples HSV values at the 256 fixed grid intersections. Classification employs a three-level cascaded decision tree, whose flow is illustrated in Figure 3.

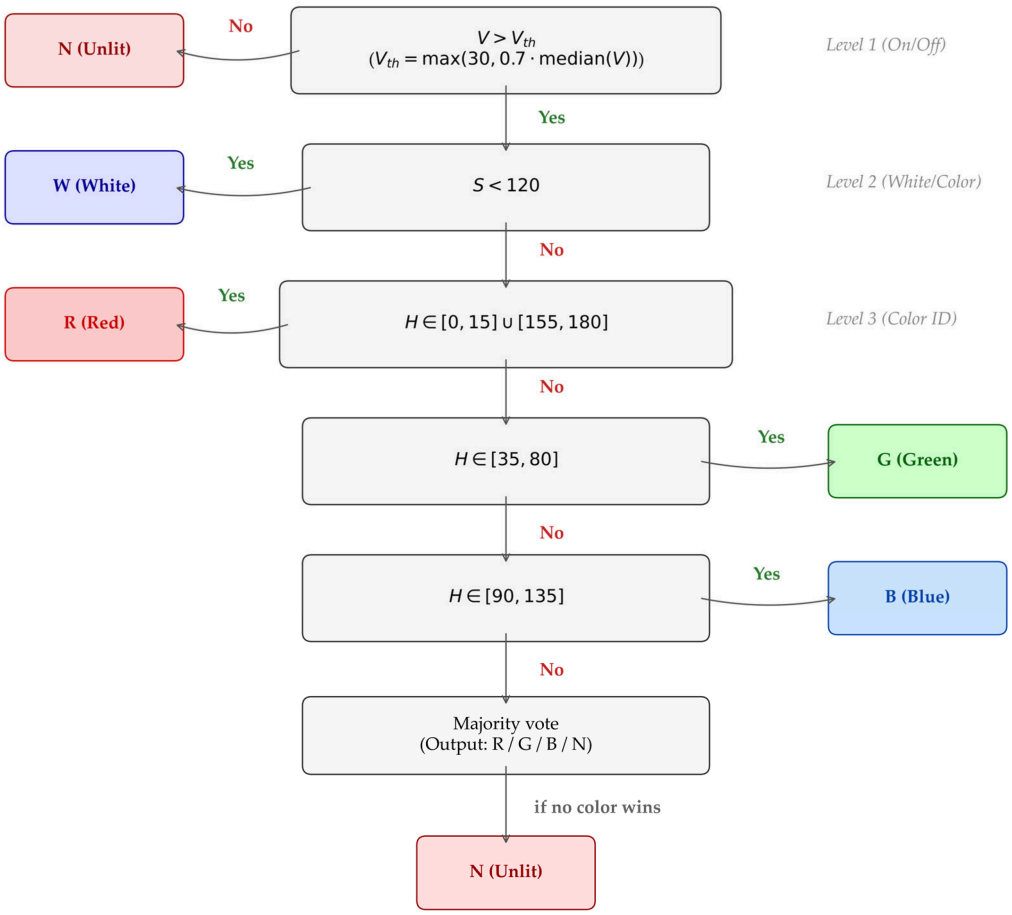


Figure 3. Cascaded decision tree for HSV-based color classification.

The first level is lit/unlit detection. A dynamic threshold is computed from the median of the V channel within the sampling region as $V_{\text{th}} = \max\left(30, 0.7 \cdot \widetilde{V}\right)$. LEDs for which the proportion of bright pixels is less than 8% are classified as unlit (N). The remaining LEDs proceed to the second level.

The second level separates white from colored LEDs. The mean saturation S of the lit LED is calculated. White LEDs, whose three channels emit simultaneously, usually have S in the 40–100 range. Colored LEDs have S concentrated in the 160–240 range. With 120 as the boundary, $S < 120$ is classified as white (W); otherwise, the LED proceeds to the third level.

The third level performs color identification. The mean hue H is evaluated sequentially. If $H \in [0, 15] \cup [155, 180]$, the LED is classified as red (R). Otherwise, if $H \in [35, 80]$, it is classified as green (G). Otherwise, if $H \in [90, 135]$, it is classified as blue (B). If H satisfies none of the above, a majority vote is performed by counting the proportions of pixels belonging to each color category. The majority vote may output red, green, or blue. Only when all three color pixel counts are zero is the LED classified as N.

This cascaded scheme requires no training. The dynamic brightness threshold adapts to changes in ambient lighting. The classification result is a 16 × 16 color matrix, i.e., the decoded data frame. This matrix is then compared element-by-element with the ground-truth matrix to update the cumulative BER and FER.

4. Experiments and Results Analysis

4.1. Experimental Platform and Scheme Configuration

The transmitter consists of a 16 × 16 WS2812 flexible LED array and an ESP32-WROOM-32 development board (Espressif Systems Co., Ltd., Shanghai, China), powered by a 5 V DC supply. The receiver employs a Hikrobot MV-CU120-10UC industrial camera (12-megapixel CMOS sensor, 12 mm fixed-focus lens, Hikrobot Co., Ltd., Hangzhou, China) mounted on a tripod. The image output resolution is 640 × 640 pixels. Experiments were conducted under typical indoor office lighting conditions. The ambient illuminance in front of the camera, with the LED array completely turned off, was approximately 130–160 lux. The communication distance was fixed at 1 m. The experimental setup is shown in Figure 4.

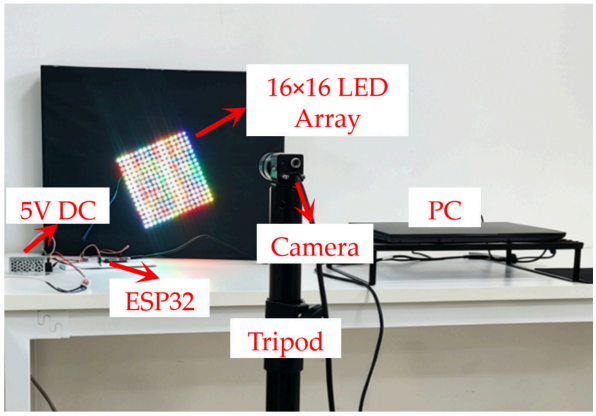


Figure 4. Experimental setup. The 16 × 16 LED array is driven by an ESP32 module and powered by a 5 V DC supply. The camera (Hikrobot MV-CU120-10UC, 12 mm lens) is mounted on a tripod at a distance of 1 m. The LED array is tilted and rotated to introduce composite geometric distortion.

The key system parameters are summarized in Table 1.

Table 1. System parameters of the experimental setup.

Parameter	Value/Description
LED array	16 × 16 WS2812 RGB LED
Modulation	4-Color (2 bits/LED)
Frame interval	0.2 s (5 fps)
Effective data rate	≈2365 bps
Camera model	Hikrobot MV-CU120-10UC
Sensor	CMOS, 12 MP
Output resolution	640 × 640 pixels
Software	Python 3.9, OpenCV 4.11.0, scikit-learn, Ultralytics YOLOv8
Distance	1 m

The tilt angle (Yaw) and in-plane rotation angle (Roll) of the LED panel were preset using a protractor. After manually rotating the panel to the marked positions, the angles were verified with a smartphone inclinometer. The actual angular deviation from the preset values did not exceed $\pm 2^{\circ}$. By combining different Yaw and Roll preset values, a total of 16 cross-test conditions were formed.

To clearly define the comparison schemes, the three configurations are defined as follows:

- Scheme A (Baseline): A manually selected fixed rectangle is used as the ROI. No geometric correction is performed, and color sampling is carried out on a uniformly spaced fixed grid.
- Scheme B (HBB + single-stage correction): To establish a representative baseline, Scheme B follows the deep learning processing paradigm commonly adopted in recent OCC literature (e.g., Nguyen et al. [15], Sitanggang et al. [30]). This scheme uses a conventional YOLOv8n horizontal bounding box (HBB) for ROI extraction, followed by a standard single-stage perspective homography transformation. Moreover, LED center detection and clustering are performed independently on every frame without any frame-to-frame acceleration strategy such as locking. This scheme belongs to the traditional uncorrected, frame-by-frame processing type.
- Scheme C (Proposed): YOLOv8n-OBB rotated bounding box detection is adopted. A cascaded coarse-to-fine two-stage correction is applied. After the first frame successfully completes clustering, the grid is locked. Subsequent frames reuse the cached grid and the combined transformation matrix.

The evaluation metrics include the bit error rate (BER), frame error rate (FER), 95% confidence interval (CI), and the average per-frame processing latency. All BER measurements are based on no fewer than 1500 valid frames. With a maximum of 473 bits per frame, the cumulative number of transmitted bits over 1500 frames is approximately 7.1×10^5. When the observed BER = 0, the half-width of the 95% confidence upper bound is $1.96/\sqrt{N} \approx 2.3 \times 10^{-3}$, i.e., the true BER does not exceed this upper bound with 95% confidence. Therefore, BER entries marked as 0 in Table 2 correspond to a true BER lower than 2.3×10^{-3}.

Table 2. BER and FER of the three schemes under different tilt and rotation conditions.

Yaw	Roll	Scheme A (BER/FER)	Scheme B (BER/FER)	Scheme C (BER/FER)
0°	0°	6.58×10^{-4}/16.84%	0/0%	0/0%
0°	15°	Failed	0/0%	0/0%
0°	30°	Failed	Failed	0/0%
0°	40°	Failed	Failed	0/0%
15°	0°	0.2321/100%	0/0%	0/0%
15°	15°	0.1829/100%	0/0%	0/0%
15°	30°	0.3102/100%	Failed	0/0%
15°	40°	0.4510/100%	Failed	0/0%
30°	0°	0.2117/100%	0/0%	0/0%
30°	15°	0.1508/100%	0/0%	0/0%
30°	30°	Failed	0/0%	0/0%
30°	40°	Failed	Failed	0/0%
45°	0°	0.386/100%	0/0%	0/0%
45°	15°	0.4987/100%	Failed	0/0%
45°	30°	Failed	Failed	0/0%
45°	40°	Failed	Failed	0/0%

"Failed" indicates that the system cannot produce a valid decoded frame. For Scheme A (baseline), no geometric correction is applied; the fixed uniform grid becomes misaligned with the actual LED centers under Yaw-Roll distortion, leading to color misclassification (most LEDs are identified as 'N' or incorrect colors). For Scheme B (HBB + single-stage correction), the horizontal bounding box includes background pixels and the single-stage correction is insufficient, which causes K-Means clustering to fail, thus preventing the construction of a complete 16×16 grid.

4.2. BER Performance Comparison

Table 2 summarizes the BER and FER data of the three schemes under 16 distortion combinations. The 95% confidence interval of the BER is calculated as follows:

$$\mathrm{CI}_{95\%} = \hat{p} \pm 1.96\sqrt{\frac{\hat{p}(1-\hat{p})}{N}} \tag{7}$$

where $\hat{p}$ is the observed BER and N is the cumulative number of transmitted bits. When the observed BER = 0, the 95% confidence upper bound is $1.96/\sqrt{N} \approx 2.3 \times 10^{-3}$, i.e., the true BER does not exceed 2.3×10^{-3} at the 95% confidence level.

Three progressive conclusions can be drawn from Table 2. First, Scheme A exhibits severe bit errors or complete failure under any non-zero angle, with a residual BER of 6.58×10^{-4} even at 0°. BER generally degrades with distortion, but not strictly monotonically. Second, Scheme B (representing the mainstream HBB paradigm) achieves error-free decoding under pure perspective tilt (Yaw-only) but fails completely when joint in-plane rotation (Roll) is introduced. This failure exposes the inherent vulnerability of standard axis-aligned object detectors in unconstrained OCC channels. When the LED array undergoes Roll rotation, the conventional HBB inevitably engulfs extensive out-of-domain background pixels, such as ambient reflections. This severe spatial noise directly corrupts the subsequent binarization and clustering processes, leading to synchronization and decoding failure. The vulnerability of Scheme B under joint Yaw-Roll distortions fully demonstrates that standard deep-learning toolchains are inadequate for complex geometric channels, thereby justifying the necessity of our proposed OBB-driven framework. Third, Scheme C achieves error-free transmission under all 16 tested conditions.

To intuitively illustrate the performance difference between Scheme B and Scheme C, Figure 5 presents the BER comparison of the two schemes in the form of a heatmap across all 16 angle combinations.

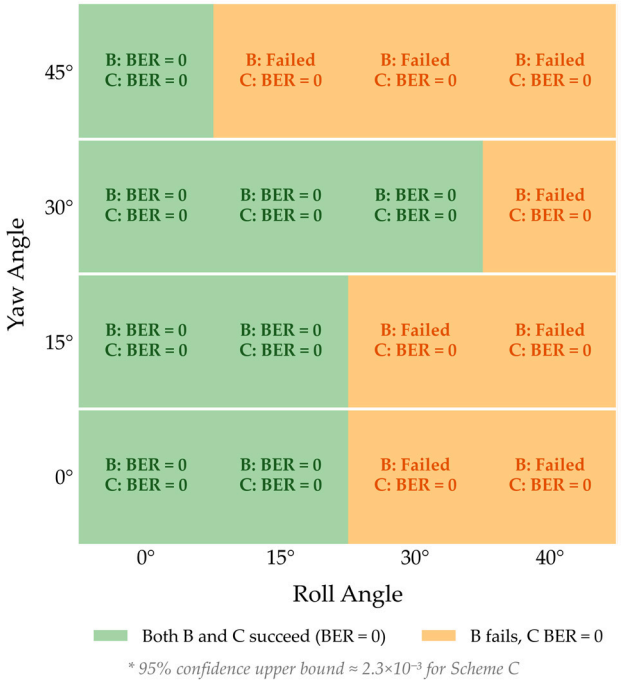


Figure 5. BER performance comparison between Scheme B and Scheme C under all 16 Yaw–Roll combinations. Green cells indicate both schemes achieve BER = 0. Orange cells represent conditions where Scheme B fails but Scheme C remains error-free.

4.3. Ablation Study and Correction Accuracy Analysis

To verify the necessity of each stage in the cascaded framework, an ablation study was conducted using the Yaw 30°, Roll 30° condition as an example. Each ablation configuration was measured three times, and the average values are reported in Table 3. "Coarse only" refers to performing only the OBB-corner-based coarse correction while skipping the fine correction; "Fine only" refers to skipping the coarse correction and directly detecting LED corners on the original distorted ROI for fine correction. With coarse correction alone, although the edge-length ratios approach 1.03, residual sub-pixel deviations still result in a BER on the order of 10^{-3}. With fine correction alone, grid construction fails completely because the LED corners cannot be reliably extracted under composite distortion. The cascaded strategy optimizes the ratios of opposite sides to 0.998 and achieves error-free transmission.

Table 3. Ablation study of the cascaded correction (Yaw 30°, Roll 30°).

Configuration	R_h	R_v	BER	FER
Coarse only	1.032	0.978	3.2×10^{-3}	45%
Fine only	Detection failed	Detection failed	—	—
Coarse + Fine	0.998	1.001	0	0%

Figure 6 presents the results of the ablation study in a dual-axis chart.

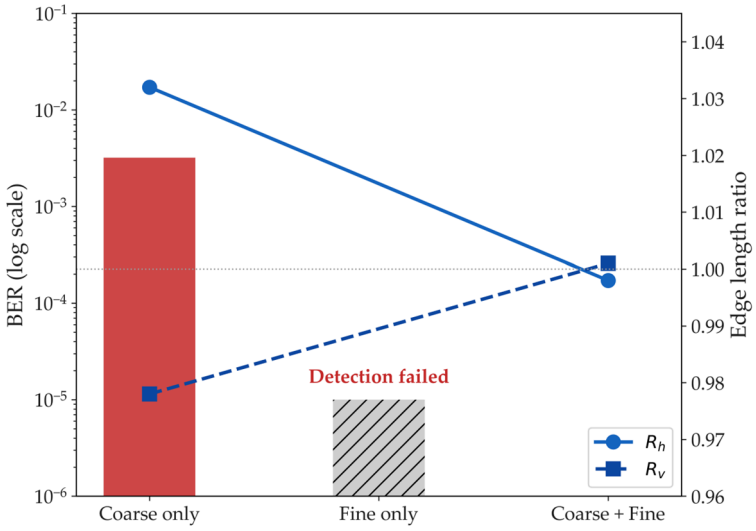


Figure 6. Ablation study results under Yaw 30° and Roll 30°. Bars represent BER (left axis, log scale). The fine-only configuration fails because LED corners cannot be reliably extracted without prior coarse correction. The cascaded approach reduces BER to zero while optimizing both edge length ratios R_h and R_v to within 0.002 of unity. The solid blue line with circle markers indicates R_h; the dashed blue line with square markers indicates R_v; the gray dotted line indicates the ideal reference at 1.0 (right axis).

Table 4 lists the geometric calibration metrics of Scheme C under three representative conditions. Even under composite distortion, R_h and R_v remain stable between 0.997 and 1.004, and the center offset is less than 2 pixels, confirming the high-precision alignment capability of the cascaded correction.

Table 4. Geometric calibration metrics of Scheme C under representative conditions.

Condition	R_h	R_v	d (Pixels)
Yaw 0° + Roll 0°	1.002	1.001	0.4
Yaw 30° + Roll 0°	1.008	0.995	1.2
Yaw 45° + Roll 30°	0.997	1.004	1.8

4.4. Processing Latency and Real-Time Performance

Hardware and inference configuration. All latency measurements were performed on the following platform:

- CPU: AMD Ryzen 7 4800U (8 cores, 16 threads, base frequency 1.80 GHz)
- RAM: 16 GB
- GPU: None (only integrated AMD Radeon Graphics, not used for computation)
- CUDA acceleration: Not used (inference was executed on CPU only)
- YOLO backend: Ultralytics YOLOv8 framework with PyTorch 2.7.0 backend
- Image resolution: 640 × 640 pixels

All compared schemes (Schemes A, B, and C) were evaluated on the same platform to ensure a fair comparison.

The average per-frame processing latency was measured under the same Yaw 30°, Roll 30° condition used in the ablation study. Scheme A, which had already failed under this condition, was excluded from the latency comparison. The results are shown in Figure 7.

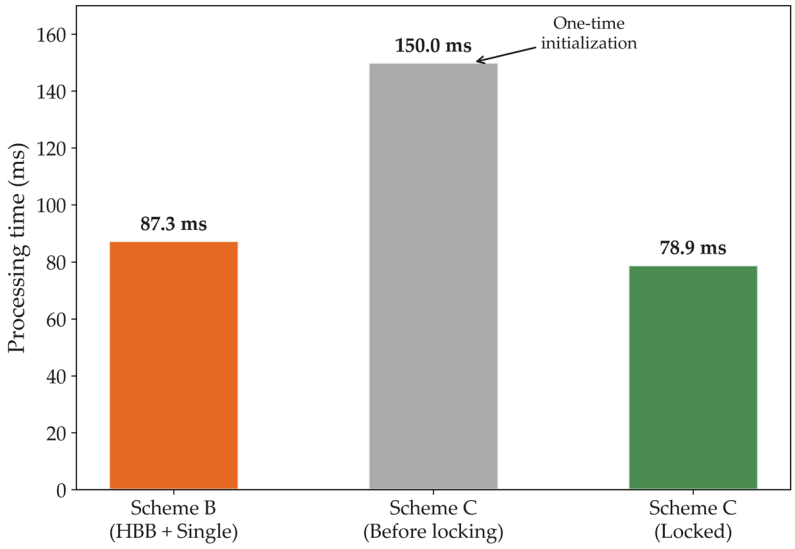


Figure 7. Per-frame processing latency of Scheme B and Scheme C under Yaw 30° and Roll 30°. For Scheme C, both the one-time initialization cost before grid locking (150 ms) and the steady-state latency after locking (78.9 ms) are shown.

Scheme B requires a complete run of YOLOv8n horizontal-box inference, single-stage perspective correction, LED center detection, K-Means clustering, and color classification for every frame, consuming 87.3 ms.

The latency of Scheme C exhibits a distinct two-stage characteristic. Before locking, the first frame completes the entire initialization pipeline, taking approximately 150 ms. After locking, each frame retains only ROI cropping, combined matrix mapping, and fixed-grid color sampling, reducing the latency to 78.9 ms—a reduction of approximately 47% compared with the pre-locking stage.

The post-locking latency remains stable in the 76–83 ms range across all 16 distortion conditions, independent of the degree of distortion. The four operations after locking all

have fixed computational complexity: ROI cropping is an image array slicing operation, combined matrix mapping is a fixed-resolution perspective transformation, grid sampling computes the neighborhood mean for 256 positions, and color classification traverses a three-level decision tree. None of these involves iterative convergence or object detection, satisfying the real-time requirement.

At a frame rate of 5 fps, the available processing window per frame is 200 ms. The post-locking latency of 78.9 ms for Scheme C occupies only about 39% of the frame period, leaving ample computational margin for higher-order modulation or more complex decoding in the future.

Computational complexity and parameter count comparison. Our YOLOv8n-OBB detector has approximately 3.1 million parameters and costs about 9.1 billion FLOPs for a 640×640 input (estimated by scaling the official benchmark value of 23.3 billion FLOPs at 1024×1024 resolution by the area ratio $(640/1024)^2$). After grid locking, the core decoding per frame (perspective warp + color classification) takes 45.85 ms on average, consisting of 10.46 ms for warping and 35.37 ms for color classification. Even when visualization and GUI overhead are included, the total per-frame latency is 78.91 ms, which is significantly lower than the 200 ms budget allowed for 5 fps. All heavy operations (YOLO inference, coarse/fine correction, LED center detection, K-Means clustering) are executed only once during initialization.

Compared with horizontal-bounding-box (HBB) based deep learning detectors (e.g., Nguyen and Jang [15], Jocher et al. [16]), which run detection on every frame and do not handle geometric distortion, our method offers fundamental advantages. Traditional grayscale thresholding [14] has low complexity but fails under composite distortion. Other related works, such as the derivative-based demodulation method [23] and the gapless sampling method [24], are designed for symbol detection or localization and do not include geometric correction or frame locking. In contrast, our OBB tightly encloses the LED array, and the cascaded two-stage homography correction enables BER = 0 under composite 45° Yaw and 40° Roll distortion. Moreover, the grid locking mechanism executes YOLO inference only once; after that, each frame only requires a fixed set of simple operations (warping and color classification), resulting in a constant per-frame workload. Hence, with a parameter count similar to that of HBB detectors, our scheme achieves error-free transmission under composite distortion and extremely low steady-state latency. Based on our literature review, this combination of features has not been reported in previous OCC works.

4.5. Quantitative Comparison with State-of-the-Art (SOTA) Schemes

To evaluate system advancement, this section compares the proposed system with recent representative SOTA OCC and MIMO-VLC schemes. To eliminate variations in diverse testing environments and hardware performance, Scheme B is introduced as a benchmark baseline. Data for other studies [28–33] are directly cited from their original publications, as detailed in Table 5.

By comparing the quantitative data in Table 5, two core technical boundaries can be defined:

Trade-off between distortion tolerance and hardware cost: Literature [29] relies on a high-cost adaptive liquid lens hardware route to mitigate multi-angle tilts, achieving a BER of 1.4×10^{-3}. In contrast, the proposed scheme achieves distortion correction purely through receiver-side algorithmic innovation, utilizing the oriented rigid boundary of YOLOv8n-OBB to precisely demarcate the region of the tilted LED array and suppress spatial edge noise. Compared with the baseline schemes, traditional horizontal bounding box (HBB) architectures (e.g., Scheme B and Literature [30]) suffer from a complete collapse of topological synchronization under in-plane rotation (Roll) distortion. Conversely,

without requiring specialized optical components or transmitter-side modifications [32], the proposed scheme achieves error-free decoding (BER = 0) under extreme composite distortions of 45° Yaw and 40° Roll.

Table 5. System-level comparison between the proposed scheme and representative OCC studies.

Scheme Source	Light Source & Scale	Core Architecture & Features	Distortion Boundary	BER Performance	Delay & Hardware Platform
Proposed Scheme	16 × 16 RGB-LED	YOLOv8n-OBB, Two-stage Homography, Grid Locking	Yaw $\leq 45°$, Roll $\leq 40°$	0 (Error-free)	78.9 ms (Pure CPU)
Scheme B (Baseline)	16 × 16 RGB-LED	YOLOv8n (HBB), Single Homography	Single-axis Yaw tilt only	Fails under Roll	87.3 ms (Pure CPU)
Sitanggang et al. [30]	8 × 8 LED Matrix	YOLOv8 (HBB), 2D MIMO Demod.	Slight perspective (No Roll)	$\approx 10^{-2}$ (Dynamic)	1.25 ms (High-end GPU)
Song et al. [32]	4 × 4 Infrared LED	ROI Extr., PnP Pose Solver	Single-axis (Yaw $<$ 30°)	N/R (Positioning focus)	≈45 ms (Single frame)
Liu et al. [28]	Tail-light LED Array	3D Geo. & Channel Modeling	3-axis (Yaw/Pitch/Roll)	Theoretical simulation only	Theoretical (No runtime network)
Shin & Jang [31]	Multi-LED Vehicular	YOLO Det., DeepSORT Tracking	Pure mobility (No angles)	N/R	≈15 ms (Edge GPU)
Palitharathna et al. [29]	Indoor Multi-LED	Liquid Lens, CNN + LSTM	Dynamic random 3-axis	1.4×10^{-3} (@30 dB)	N/R (Hardware driver + GPU)
Zhang et al. [33]	Screen-Camera Link	Tx-side Spatial Pre-correction	Single-axis (Yaw $<$ 20°)	$<10^{-3}$ (Single-axis tilt)	N/R (Intrusive at Tx)

N/R: Not reported.

Trade-off between computational efficiency and hardware overhead: Under the strict constraints of a 16 × 16 high-density topology (256 nodes) and a pure CPU platform, the proposed scheme utilizes a spatio-temporally decoupled pipeline. This optimizes the single-frame steady-state delay to 78.9 ms, significantly outperforming Scheme B (87.3 ms) which requires frame-by-frame full boundary resolution. This demonstrates an excellent balance between geometric robustness and steady-state efficiency on low-cost edge platforms.

5. Discussion

The proposed scheme successfully balances geometric robustness and computational efficiency. In the spatial domain, YOLOv8n-OBB provides a background-free ROI, overcoming the limitation of HBB methods under rotation. In the mathematical domain, two-stage homography matrices are cascaded into a single transformation, enabling sub-pixel alignment with only one resampling per frame after locking. This pure software receiver architecture requires no hardware modification.

The grid locking strategy is key to real-time performance. All heavy operations (YOLO inference, homography recomputation, K-Means clustering) are executed only once during initialization, reducing the steady-state per-frame latency to 78.9 ms on a pure CPU platform, well below the 200 ms budget for 5 fps. Static tests over 4 h showed no grid drift. A BER-monitored re-locking mechanism provides autonomous recovery capability.

The current experiments were conducted under fixed conditions (1 m distance, indoor lighting, 5 fps, static distortion). While these are sufficient to validate the core innovations, practical OCC scenarios involve varying distances, illumination, dynamic motion, and camera settings. Systematic evaluation under such diverse conditions is left for future work. Nevertheless, this work bridges the gap between theoretical 3D geometric analyses and a deployable, software-only receiver-side correction for composite distortion.

6. Conclusions

This study proposes and validates a robust software-defined OCC receiver architecture integrating YOLOv8n-OBB oriented object detection, cascaded two-stage perspective

correction, and K-Means grid locking, specifically designed to achieve precise LED array localization under composite geometric distortions. Experimental results across 16 extreme Yaw–Roll cross-conditions indicate that while the uncorrected and traditional horizontal bounding box (HBB) schemes fail under non-axial and severe composite distortions, the proposed architecture maintains error-free transmission (BER = 0) under all test conditions, with the opposite-side ratios of the corrected LED grid remaining highly stable between 0.997 and 1.004. The ablation study further confirms the distinct and indispensable functionality of each correction stage. Benefiting from the K-Means grid locking mechanism, the steady-state single-frame processing latency is reduced by approximately 47%, remaining stable within the 76–83 ms range (averaging 78.9 ms) on a resource-constrained, pure CPU platform, thereby satisfying real-time communication requirements. The system's robustness has been verified for Yaw angles up to 45° in the current setup. Future work will focus on incorporating lightweight state predictors (e.g., Kalman filtering) to enhance adaptive grid re-locking capabilities and systematic temporal continuous robustness under complex dynamic perturbations and UAV flight profiles.

Author Contributions: Conceptualization, J.J. and P.Q.; methodology, P.Q.; software, P.Q.; validation, P.Q. and J.J.; formal analysis, P.Q.; investigation, P.Q. and Y.T.; resources, Z.S.; data curation, P.Q.; writing—original draft preparation, P.Q.; writing—review and editing, P.Q., J.J. and Z.S.; visualization, P.Q.; supervision, J.J. and Z.S.; project administration, J.J. and Z.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

BER	Bit Error Rate
CI	Confidence Interval
CMOS	Complementary Metal-Oxide-Semiconductor
FER	Frame Error Rate
fps	frames per second
HBB	Horizontal Bounding Box
HSV	Hue, Saturation, Value
LED	Light-Emitting Diode
mAP	mean Average Precision
OBB	Oriented Bounding Box
OCC	Optical Camera Communication
OWC	Optical Wireless Communication
ROI	Region of Interest
VLC	Visible Light Communication
YOLO	You Only Look Once

References

1. Saeed, N.; Guo, S.; Park, K.-H.; Al-Naffouri, T.Y.; Alouini, M.-S. Optical Camera Communications: Survey, Use Cases, Challenges, and Future Trends. *Phys. Commun.* **2019**, *37*, 100900. [CrossRef]

2. Chowdhury, M.Z.; Hossan, M.T.; Islam, A.; Jang, Y.M. A Comparative Survey of Optical Wireless Technologies: Architectures and Applications. *IEEE Access* **2018**, *6*, 9819–9840. [CrossRef]
3. Hassan, N.U.; Naeem, A.; Pasha, M.A.; Jadoon, T.; Yuen, C. Indoor Positioning Using Visible LED Lights: A Survey. *ACM Comput. Surv.* **2015**, *48*, 20. [CrossRef] [PubMed]
4. Hasan, M.K.; Ali, M.O.; Rahman, M.H.; Chowdhury, M.Z.; Jang, Y.M. Optical Camera Communication in Vehicular Applications: A Review. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 6260–6281. [CrossRef]
5. Teli, S.R.; Matus, V.; Zvanovec, S.; Perez-Jimenez, R.; Vitek, S.; Ghassemlooy, Z. Optical Camera Communications for IoT–Rolling-Shutter Based MIMO Scheme with Grouped LED Array Transmitter. *Sensors* **2020**, *20*, 3361. [CrossRef] [PubMed]
6. Zhang, P.; Liu, Z.; Hu, X.; Sun, Y.; Deng, X.; Zhu, B.; Yang, Y. Constraints and Recent Solutions of Optical Camera Communication for Practical Applications. *Photonics* **2023**, *10*, 608. [CrossRef]
7. *IEEE Std 802.15.7-2018*; IEEE Standard for Local and Metropolitan Area Networks—Part 15.7: Short-Range Optical Wireless Communications (Revision of IEEE Std 802.15.7-2011). IEEE: Piscataway, NJ, USA, 2019; pp. 1–407.
8. Matheus, L.E.M.; Vieira, A.B.; Vieira, L.F.M.; Vieira, M.A.M.; Gnawali, O. Visible Light Communication: Concepts, Applications and Challenges. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 3204–3237. [CrossRef]
9. Danakis, C.; Afgani, M.; Povey, G.; Underwood, I.; Haas, H. Using a CMOS Camera Sensor for Visible Light Communication. In Proceedings of the IEEE Globecom Workshops, Anaheim, CA, USA, 3–7 December 2012; pp. 1244–1248.
10. Luo, P.; Zhang, M.; Ghassemlooy, Z.; Le Minh, H.; Tsai, H.-M.; Tang, X.; Png, L.C.; Han, D. Experimental Demonstration of RGB LED-Based Optical Camera Communications. *IEEE Photonics J.* **2015**, *7*, 7904212. [CrossRef]
11. Roberts, R.D. Undersampled Frequency Shift ON-OFF Keying (UFSOOK) for Camera Communications (CamCom). In Proceedings of the Wireless and Optical Communication Conference, Chongqing, China, 16–18 May 2013; pp. 645–648.
12. Kuo, Y.-S.; Pannuto, P.; Hsiao, K.-J.; Dutta, P. Luxapose: Indoor Positioning with Mobile Phones and Visible Light. In Proceedings of the 20th Annual International Conference on Mobile Computing and Networking, Maui, HI, USA, 7–11 September 2014; pp. 447–458.
13. Hu, X.; Zhang, P.; Sun, Y.; Deng, X.; Yang, Y.; Chen, L. High-Speed Extraction of Regions of Interest in Optical Camera Communication Enabled by Grid Virtual Division. *Sensors* **2022**, *22*, 8375. [CrossRef] [PubMed]
14. Gonzalez, R.C.; Woods, R.E. *Digital Image Processing*, 4th ed.; Global ed.; Pearson: New York, NY, USA, 2017.
15. Nguyen, H.; Jang, Y.M. Design and Implementation of Deep Learning-Based MIMO C-OOK Scheme for Optical Camera Communication Considering Mobility Environment. In Proceedings of the International Conference on Artificial Intelligence in Information and Communication (ICAIIC), Osaka, Japan, 19–22 February 2024; pp. 860–863.
16. Jocher, G.; Chaurasia, A.; Stoken, A.; Borovec, J.; NanoCode012; Kwon, Y.; Michael, K.; Xie, T.; Fang, J.; Imyhxy; et al. *Ultralytics/Yolov5: V7.0—YOLOv5 SOTA Realtime Instance Segmentation*; Zenodo: Geneva, Switzerland, 2022.
17. Yang, X.; Yan, J.; Feng, Z.; He, T. R3Det: Refined Single-Stage Detector with Feature Refinement for Rotating Object. *Proc. AAAI Conf. Artif. Intell.* **2021**, *35*, 3163–3171. [CrossRef]
18. Liu, Y.; Chow, C.-W.; Liang, K.; Chen, H.-Y.; Hsu, C.-W.; Chen, C.-Y.; Chen, S.-H. Comparison of Thresholding Schemes for Visible Light Communication Using Mobile-Phone Image Sensor. *Opt. Express* **2016**, *24*, 1973. [CrossRef] [PubMed]
19. Hartley, R.; Zisserman, A. *Multiple View Geometry in Computer Vision*, 2nd ed.; Cambridge University Press: Cambridge, UK, 2004.
20. Zhang, Z.; Zhang, T.; Zhou, J.; Lu, Y.; Qiao, Y. Thresholding Scheme Based on Boundary Pixels of Stripes for Visible Light Communication with Mobile-Phone Camera. *IEEE Access* **2018**, *6*, 53053–53061. [CrossRef]
21. Chen, Q.; Wen, H.; Deng, R.; Chen, M.; Xu, Q.; Zong, T.; Geng, K. Spaced Color Shift Keying Modulation for Camera-Based Visible Light Communication System Using Rolling Shutter Effect. *Opt. Commun.* **2019**, *449*, 19–23. [CrossRef]
22. He, J.; Yu, K.; Huang, Z.; Chen, Z. Multi-Column Matrices Selection Combined with k-Means Scheme for Mobile OCC System with Multi-LEDs. *IEEE Photon. Technol. Lett.* **2021**, *33*, 623–626. [CrossRef]
23. De Murcia, M.; Boeglen, H.; Julien-Vergonjanne, A. ZEROES: Robust Derivative-Based Demodulation Method for Optical Camera Communication. *Photonics* **2024**, *11*, 949. [CrossRef]
24. Hong, Y.; Xie, X.; Shen, X. TickRS: A High-Speed Gapless Signal Sampling Method for Rolling-Shutter Optical Camera Communication. *Photonics* **2025**, *12*, 720. [CrossRef]
25. Nguyen, D.T.; Park, Y. Data Rate Enhancement of Optical Camera Communications by Compensating Inter-Frame Gaps. *Opt. Commun.* **2017**, *394*, 56–61. [CrossRef]
26. Nguyen, V.H.; Thieu, M.D.; Nguyen, H.; Jang, Y.M. Design and Implementation of the MIMO–COOK Scheme Using an Image Sensor for Long-Range Communication. *Sensors* **2020**, *20*, 2258. [CrossRef] [PubMed]
27. Lee, H.-Y.; Lin, H.-M.; Wei, Y.-L.; Wu, H.-I.; Tsai, H.-M.; Lin, K.C.-J. RollingLight: Enabling Line-of-Sight Light-to-Camera Communications. In Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services, Florence, Italy, 18–22 May 2015; pp. 167–180.
28. Liu, A.; Shi, W.; Safari, M.; Liu, W. Investigating the Angular Distortion Impact on Vehicular Optical Camera Communication (OCC) Systems. *Opt. Express* **2024**, *32*, 19697. [CrossRef] [PubMed]

29. Palitharathna, K.W.S.; Skouroumounis, C.; Krikidis, I. Liquid Lens-Based Imaging Receiver for MIMO VLC Systems. *arXiv* **2025**, arXiv:2503.10316.
30. Sitanggang, O.S.; Nguyen, V.L.; Nguyen, H.; Pamungkas, R.F.; Faridh, M.M.; Jang, Y.M. Design and Implementation of a 2D MIMO OCC System Based on Deep Learning. *Sensors* **2023**, *23*, 7637. [CrossRef] [PubMed]
31. Shin, E.; Jang, Y.M. Implementation of Multiple Transmitter Tracking Mechanism Based on DeepSORT for 2D MIMO Optical Camera Communications. *J. Korean Inst. Commun. Inf. Sci.* **2025**, *50*, 803–811. [CrossRef]
32. Song, S.; Wu, P.; Liu, Y.; Zhao, L.; Wu, T.; Liu, X.; Guo, L. Implementation of an Indoor Optical Camera Communication and Localization Fusion System Using Infrared LED Markers. *Opt. Express* **2024**, *32*, 41361. [CrossRef] [PubMed]
33. Zhang, H.; Zhu, B.; Liu, Y.; Peng, J.; Yu, X.; Zhang, Z. PreSCC: Robust Screen-Camera Communication via Pre-Correction Against Perspective Distortion and Illumination Interference. *TechRxiv* **2026**. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.