Optics Communications 407 (2018) 112–126

Contents lists available at ScienceDirect

Optics Communications

journal homepage: www.elsevier.com/locate/optcom

Visible light communication based vehicle positioning using LED street light and rolling shutter CMOS sensors

Trong Hop Do, Myungsik Yoo *

School of Electronic Engineering, Soongsil University, Seoul 06978, South Korea

ARTICLE INFO

Keywords:
Visible light communication
Vehicle
Positioning
CMOS sensor
Rolling shutter

ABSTRACT

This paper proposes a vehicle positioning system using LED street lights and two rolling shutter CMOS sensor cameras. In this system, identification codes for the LED street lights are transmitted to camera-equipped vehicles through a visible light communication (VLC) channel. Given that the camera parameters are known, the positions of the vehicles are determined based on the geometric relationship between the coordinates of the LEDs in the images and their real world coordinates, which are obtained through the LED identification codes. The main contributions of the paper are twofold. First, the collinear arrangement of the LED street lights makes traditional camera-based positioning algorithms fail to determine the position of the vehicles. In this paper, an algorithm is proposed to fuse data received from the two cameras attached to the vehicles in order to solve the collinearity problem of the LEDs. Second, the rolling shutter mechanism of the CMOS sensors combined with the movement of the vehicles creates image artifacts that may severely degrade the positioning accuracy. This paper also proposes a method to compensate for the rolling shutter artifact, and a high positioning accuracy can be achieved even when the vehicle is moving at high speeds. The performance of the proposed positioning system corresponding to different system parameters is examined by conducting Matlab simulations. Small-scale experiments are also conducted to study the performance of the proposed algorithm in real applications.

1. Introduction

In the imaging industry, two types of sensors are available: CCD and CMOS. CCD has long been considered to yield superior image quality, and in the early days of the digital age, most cameras were equipped with CCD sensors. However, there has been a dramatic evolution in CMOS sensors in recent years, which has resulted in CMOS sensors performing better than CCD sensors in various aspects, including image quality, frame rate, and production costs. Consequently, CMOS sensors are now used as building blocks for most imaging devices, and their widespread adoption makes them suitable for use as receiving devices in visible light communication (VLC) [1–3] and VLC based positioning [4–6].

This paper examines the vehicle positioning problem, which currently has three popular approaches. The most popular approach is using Global Positioning System (GPS) thanks to its low cost. However, GPS might be unavailable in tunnels or streets in urban areas where the GPS signals are hindered by high building. Another approach is using 4G/LTE, which can overcome the limitation related to the availability of GPS. Nevertheless, this approach only provides tens of meters of positioning accuracy [7,8], which is insufficient for applications that require a high level of accuracy such as autonomous vehicle. The most successful approach for vehicle positioning is using Light Detection and Ranging (LIDAR), which can provide very high accuracy. However, LIDAR demands extra equipment which is generally very expensive. Because of these reasons, it is still necessary to find new techniques for vehicle positioning.

In recent years, a new kind of technique based on visible light communication has been proposed for vehicle positioning. In [9], taillights or headlights of the vehicle were used to transmit positioning signals that were received by two photodiodes installed in another vehicle. Then, a type of time difference of arrival (TDOA) algorithm was used to determine the relative position between the vehicles. This method is very difficult to achieve high accuracy in practice since with TDOA, a small error in time measurement leads to large positioning error. In [10], LED traffic lights and photodiode were used to determine the position of the vehicle. However, since photodiode cannot distinguish different light sources, this systems would be greatly vulnerable to ambient light, especially the sunlight. In [11], a neural network was used for the

* Corresponding author.
E-mail address: myoo@ssu.ac.kr (M. Yoo).

http://dx.doi.org/10.1016/j.optcom.2017.09.022
Received 28 April 2017; Received in revised form 5 August 2017; Accepted 7 September 2017
Available online 19 September 2017
0030-4018/

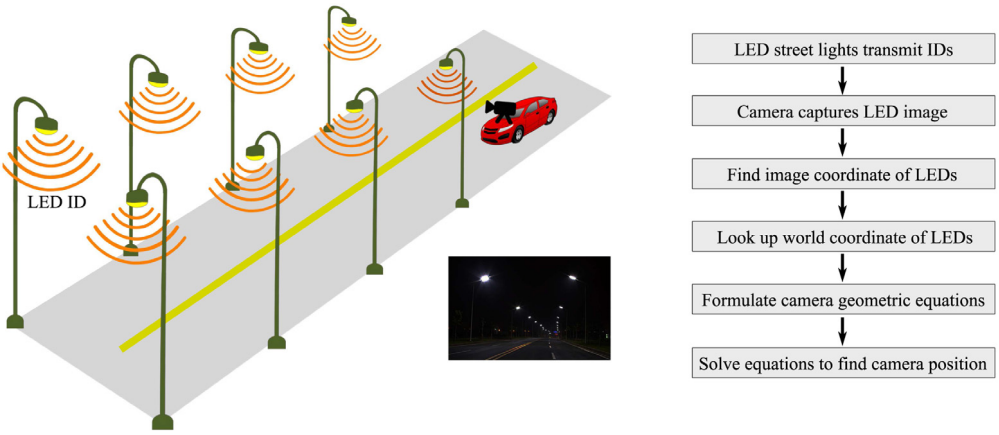


Fig. 1. System architecture.

vehicle to estimate the position of a target vehicle which sends the VLC positioning signal through its rear LED lights. The estimating vehicle used two cameras to receive the positioning signal. It is well known that in camera based positioning, the accuracy greatly depends on the distance between reference points, which are LEDs in this case. Since the rear LEDs of the target vehicle is relatively close to each other compared to the distance between the two vehicles, the positioning accuracy achieved through the simulation was not high.

This paper proposes a VLC based vehicle positioning system using LED street lights and CMOS image sensors. With many advantages including their being environmentally friendly, long lifetime, energy efficiency, etc., LED street lights have been used more and more around the world. These high power and high density illuminating systems can be utilized for communication and positioning purposes. Furthermore, in principle, positioning using street lights has a lot of potential to provide high accuracy results thanks to the dense arrangement of street lights. In the proposed system, the LED street lights are used to send identification codes from which the real world coordinates of LEDs can be obtained. CMOS image sensors are used to capture the images of LEDs in which the VLC data is embedded. The geometric relationship between the real-world coordinates of LEDs and their coordinates on the images are used to determine the position of the vehicle. The problem of using camera for positioning has been studied in the field of computer vision for a long time. However, there are two limitations of traditional positioning algorithms in solving the vehicle positioning problem considered in this paper.

First, traditional algorithms require the objects used to determine the position of the camera to not be collinear. LED street lights, however, might be collinear in many cases, and thus traditional algorithms would fail to find the position of the vehicle.

Second, regardless of the arrangement of the LED, the positioning accuracy always suffers from the inherent disadvantages of using a CMOS sensor with the rolling shutter artifact. With the rolling shutter mechanism used in CMOS sensor, each row of pixels in the sensor is exposed at a different time. Therefore, a change in the position of the objects during the exposure period results in image artifacts as the distortions of the objects in the captured images. When objects move at a high speed, these distortions might give wrong information to the positioning algorithms and thus introduce errors to the results. Neither traditional positioning algorithms, which were originally designed for use with CCD sensors that do not use the rolling shutter mechanism, nor more recent algorithms, which were intentionally designed for use with smartphone CMOS sensors [4–6] address these artifacts.

In this paper, the two difficulties mentioned above are addressed. To deal with the problem of a collinear LED arrangement, this paper proposes an algorithm that uses two cameras placed at different positions in the vehicle to capture the images of the LED street lights. The data from both images is then fused, and the position of the camera can be found even for the case with collinear LEDs. The rolling shutter artifacts are also solved using a proposed compensation method. This method uses the speed of the vehicle and the readout time of the sensor to compensate for the artifacts in the LED images. Consequently, a high positioning accuracy can be achieved even when the vehicle moves at high speeds.

To examine the performance of the proposed positioning algorithm and the rolling shutter compensation, Matlab simulations are conducted with the speed of vehicle assumed to vary from 0 to 100 km/h. The results of the simulation show that the achieved accuracy remains stable when the speed of the vehicle increases. The simulations also show the impact of the different parameters, including vehicle speed, vehicle travel distance, sensor resolution, exposure time of the sensor, and focal length of the lens on the positioning accuracy. Small-scale experiments are also conducted to give the insight into the performance of the proposed algorithm in real applications.

2. System fundamentals

2.1. System architecture

The entire system architecture is shown in Fig. 1. LED street lights installed along both sides of the street transmit a visible light signal that contains their identification (ID) codes to the camera attached on the vehicle. After receiving the IDs of the LEDs, the real world coordinates of the LEDs are obtained by looking up the database stored in the camera. This technique is normally used in VLC-based positioning systems to shorten the time required for the mobile devices to obtain the world coordinates of the LEDs [4,5]. On the other hand, the coordinates of the LEDs in the image are obtained via image processing. Given that the intrinsic parameters of the cameras are known, including sensor size, sensor resolution, and focal length of the lens, the real-world coordinates of the camera (i.e., the position of the vehicle) can be determined through the geometric relationship between the real-world coordinates and the image coordinates of the LEDs. In addition, the information regarding the speed of the vehicle is obtained through a speedometer. This information is then used by the rolling shutter compensation, as explained in the following sections.

2.2. The problem of existing positioning algorithms using camera

2.2.1. Pinhole camera model

All positioning algorithms using a camera are based on a pinhole camera model [12], which is shown in Fig. 2. There are three types of coordinate systems in this model: 3D world coordinate system, 3D camera coordinate system, and 2D image coordinate system. The 3D

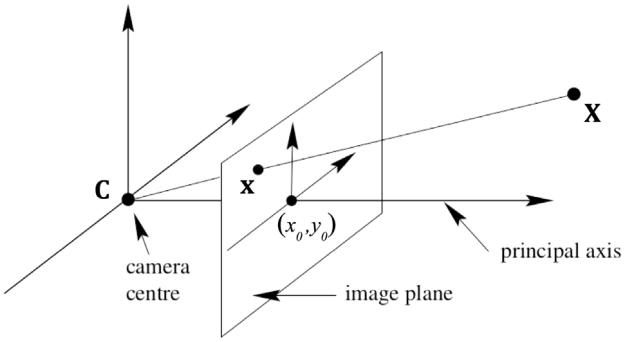


Fig. 2. Pinhole camera model.

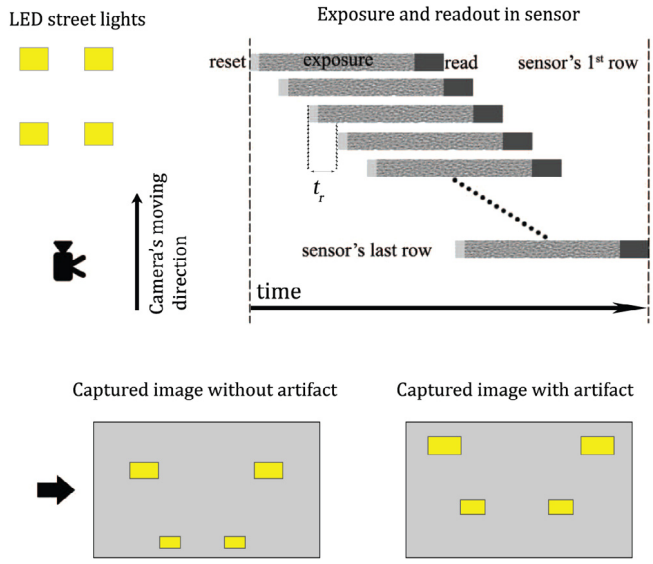


Fig. 3. Image acquisition in a CMOS sensor.

world coordinate of the camera is denoted as $\mathbf{C} = (X_0, Y_0, Z_0)^T$. The principal point, which would ideally be in the center of the sensor, has 2D image coordinates denoted as (x_0, y_0).

In general, there can be multiple LEDs in the image. Let $\mathbf{X}_i = (X_i, Y_i, Z_i, 1)^T$ and $\mathbf{x}_i = (x_i, y_i, 1)^T$ denote the world coordinate and image coordinate of the ith LED. The geometric relationship between the world coordinate and the image coordinate of each LED would formulate the equation:

$$\lambda_i \mathbf{x}_i = K \begin{bmatrix} R & t \end{bmatrix} \mathbf{X}_i, \tag{1}$$

where λ_i is a scalar, K is the 3×4 camera calibration matrix defined by the intrinsic parameters of the camera including the lens focal length f and the principal point (x_0, y_0), R is the 4×3 rotation matrix and $t = -R\mathbf{C}$ is the 4×1 translation vector.

All camera-based positioning algorithms rely on solving a set of multiple equations in the form of Eq. (1) to determine the position of the camera. Given that the intrinsic parameters of the camera are known, there are two common ways to derive the camera position from Eq. (1), depending on whether an inertial sensor is being used to obtain the information about the camera or not. These algorithms all have their own advantages and disadvantages.

2.2.2. The problem of existing positioning algorithm

The simplest algorithm to find the camera position is to use inertial sensors to get the information about the pose (i.e., orientation) of the camera [13,14]. With this information, the rotation matrix R can be obtained. Then Eq. (1) can be explicitly written as a collinearity condition from which the camera position can be found. The advantage of this method is that only two scene points are required to determine the camera position. However, the accuracy largely depends on the accuracy of the inertial sensors that are used to obtain the camera pose. In an indoor environment, the accuracy of the inertial sensors is generally sufficient. However, for the system considered in this paper, the distances from the camera to the LEDs are ten times longer than that for indoor systems, and thus the same errors in angular measurements result in much higher errors in positioning.

In the general positioning algorithm, the pose of the camera is unknown, and so is the camera rotation matrix R. Then, finding the camera position becomes the problem of solving a set of Eq. (1) to find the matrix $\begin{bmatrix} R & t \end{bmatrix}$. To solve the problem, Eq. (1) needs to be transformed into a kind of linear equation, and this can be done using a technique called Direct Linear Transformation (DLT) [15].

Let $P = \begin{bmatrix} p_1^T & p_2^T & p_3^T \end{bmatrix}^T$ denote the matrix $\begin{bmatrix} R & t \end{bmatrix}$. Each row $\begin{bmatrix} R_{m1} & R_{m2} & R_{m3} & t_{m1} \end{bmatrix}$, $m = 1, 2, 3$ in the matrix $\begin{bmatrix} R & t \end{bmatrix}$ is denoted by a 1×4 vector p_m^T. Since $\lambda_i \mathbf{x}_i = P\mathbf{X}_i$, the two vectors $\mathbf{x}_i$ and $P\mathbf{X}_i$ are parallel, and thus their cross-product is zero. By explicitly calculating the cross-product, the following equation can be formulated:

$$\begin{bmatrix} \mathbf{0} & -\mathbf{X}_i^T & y_i \mathbf{X}_i^T \\ \mathbf{X}_i^T & \mathbf{0} & -x_i \mathbf{X}_i^T \end{bmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \tag{2}$$

where $\mathbf{0} = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}$.

Each of the pair $\{\mathbf{x}_i, \mathbf{X}_i\}$ can formulate a set of two linear equations in the form of Eq. (2), and combining equations corresponding to all scene points results in the following equation set:

$$\begin{bmatrix} \mathbf{0} & -\mathbf{X}_1^T & y_1 \mathbf{X}_1^T \\ \mathbf{X}_1^T & \mathbf{0} & -x_1 \mathbf{X}_1^T \\ \mathbf{0} & -\mathbf{X}_2^T & y_2 \mathbf{X}_2^T \\ \mathbf{X}_2^T & \mathbf{0} & -x_2 \mathbf{X}_2^T \\ \vdots & \vdots & \vdots \end{bmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \end{pmatrix}. \tag{3}$$

Eq. (3) can be written as

$$M v = 0, \tag{4}$$

where M is the $2N \times 12$ matrix, v is the 12×1 vector in the left side of Eq. (3), and N is the number of scene points.

Since there are 12 unknown elements in Eq. (4), six scene points are required for the problem to be well defined. In the ideal case where there is no noise, the exact solution v can be found so that $Mv = 0$. Because of noise in the measurements, the system $Mv = 0$ will not have any exact solution. Therefore, the problem now becomes a least squares problem in which the solution v is sought to satisfy

$$\min_{\|v\|^2=1} \|Mv\|^2. \tag{5}$$

The least squares problem above is known to have the solution produced with the technique called Singular Value Decomposition (SVD) [15]. However, it is compulsory for the scene points $\mathbf{X}_i$ to not be collinear for Eq. (5) to have a unique solution. In other words, the LEDs must not be collinear so that the camera position can be found [15].

2.3. The problem of rolling shutter mechanism and image artifact

A detailed explanation of the rolling shutter mechanism can be found in [1]. See Fig. 3. Briefly, the exposures in different rows of the sensor occur one after another. The interval between the starting point of the exposures in two consecutive rows equals the readout time of one row, denoted as t_r. Since different rows in the sensor are exposed at different times, the image of LEDs captured at different rows in the sensor would correspond to the image of the LEDs at different positions relative to the camera, resulting in the image artifact as if the LEDs are closer to the camera as illustrated in the figure. More specifically, if $\mathbf{X}_i$ and $\mathbf{x}_i$ are the

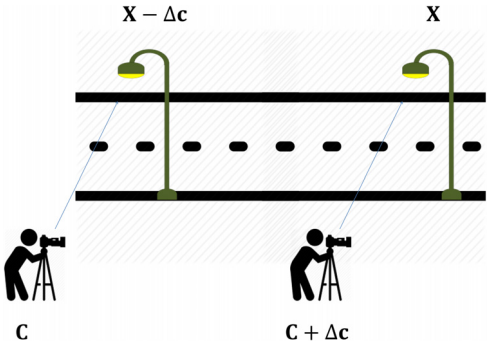


Fig. 4. Relationship between the change in camera position and LED position.

world coordinate and image coordinate of the ith LED, the relationship between these two coordinates is expressed as

$$\lambda_i \mathbf{x}_i = K \begin{bmatrix} R & t_i \end{bmatrix} \mathbf{X}_i, \tag{6}$$

where $t_i = -R\mathbf{C}_i$ is the translation vector corresponding to the position $\mathbf{C}_i$ of the camera when the ith LED is captured.

In Eq. (6), the pose of the camera is assumed to not change while capturing a single image. Therefore, the camera rotation matrix remains the same with different LEDs. Since every LED in the image corresponds to a different position of the camera, finding a single position for the camera based on these LEDs would not lead to any exact result. When the vehicles move faster, the difference between the translation vectors t_i in Eq. (6) become larger, and so does the positioning error.

3. Proposed algorithm

3.1. Relationship between camera displacement and led displacement

Of the two types of positioning algorithms explained earlier, the methods using inertial sensors are more appropriate for use in indoor environments where the distance from the camera to the LEDs is reasonably short. In the system considered in this paper, most of the inertial sensors available in the market would not be sufficiently accurate to yield a desirable positioning accuracy. Furthermore, the positioning error resulting from the error of the inertial sensor is very difficult to manage, if at all possible, using algorithms. Therefore, a type of general positioning algorithm is used in this paper to determine the position of the camera. To this end, the problem of the collinear arrangement of the LEDs must be solved first. Also, the problem of rolling shutter artifacts needs to be handled to reduce the positioning error. In this paper, the first problem is solved by using two cameras and a fusion algorithm. The second problem is then solved using a compensation method. The ideas for the solutions of these two problems are based on the conversion between the camera displacement and the LED displacement, which is stated in Lemma 1.

Lemma 1. *Let* $\mathbf{x}$ *and* $\mathbf{X}$ *denote the image coordinate and world coordinate of a scene point. Let* $\mathbf{C}$ *denote the camera position and* $\Delta\mathbf{C}$ *denote the change in the camera position. The result on the image coordinate* $\mathbf{x}$ *of the scene point* $\mathbf{X}$ *caused by the change in the camera position is equivalent to that of a change in the scene position as:* $K \begin{bmatrix} R & -R(\mathbf{C}+\Delta\mathbf{C}) \end{bmatrix} \mathbf{X} = K \begin{bmatrix} R & -R\mathbf{C} \end{bmatrix} (\mathbf{X} - \Delta\mathbf{C})$.

The meaning of Lemma 1 can be explained using Fig. 4. Suppose that at the beginning, a photographer stays at the position $\mathbf{C}$, which is also considered the camera position. Then the photographer moves to the position $\mathbf{C}+\Delta\mathbf{C}$ and captures the image of an LED placed at position $\mathbf{X}$. According to Lemma 1, this image would be identical to the image of an LED placed at position $\mathbf{X}-\Delta\mathbf{C}$ captured by the photographer when he stayed at the original position $\mathbf{C}$. To be more specific, the two image coordinates, denoted as $\mathbf{x}^1$ and $\mathbf{x}^2$, respectively, of the two LEDs in the two captured images are given as:

$$\begin{cases} \lambda^1 \mathbf{x}^1 = K \begin{bmatrix} R & -R(\mathbf{C}+\Delta\mathbf{C}) \end{bmatrix} \mathbf{X} \\ \lambda^2 \mathbf{x}^2 = K \begin{bmatrix} R & -R\mathbf{C} \end{bmatrix} (\mathbf{X} - \Delta\mathbf{C}), \end{cases} \tag{7}$$

where λ^1 and λ^2 are two arbitrary scales. According to Lemma 1, $\lambda^1\mathbf{x}^1 = \lambda^2\mathbf{x}^2$. Therefore, the image coordinates of the two LEDs in the two captured images are identical.

As seen in the following sections, the solutions for the two problems of the collinear LED arrangement and rolling shutter artifacts are all related to multiple LED images captured by the camera placed at different positions. The equations formulated from these LED images would consist of multiple camera positions, and the problem is that such a set of equations cannot be solved. By applying Lemma 1, the camera displacement can be translated into the LED displacement. Then, different images for the same LED captured at different positions of the camera can be considered as images of distinct LEDs placed at different positions captured at a single position of the camera. If the differences in the camera positions are known, then the positions of these imaginary LEDs would be known. Thus, a set of solvable equations can be formulated.

3.2. Fusing data from two cameras to deal with collinear arrangement of LEDs

3.2.1. The idea behind using two cameras

As mentioned earlier, a type of general algorithm is used in this paper. Therefore, the camera has to capture the image of at least 6 non-collinear LEDs to determine the camera position. However, in many cases of the system considered in this paper, the LEDs are arranged in straight lines along 2 sides of the street. Consequently, three or more LEDs chosen from either side of the street are always collinear. In other words, there are a maximum of 4 non-collinear LEDs from 2 sides of the street, and thus the camera position cannot be found using traditional algorithms.

To solve the collinear LED arrangement problem, one just needs to obtain images of 6 or more non-collinear LEDs to formulate the geometric equations for the camera. This would be possible using two cameras placed at different position to capture the images of the LEDs thanks to Lemma 1. Since the camera displacement can be translated into the LED displacement, the images of the same LED captured by two cameras can be considered as images of two distinct LEDs captured by a single camera. Suppose that each camera captures the image of 6 LEDs arranged in two lines, as illustrated in Fig. 5. The image captured by the second camera can be considered to be an image of 6 displaced LEDs captured by the first camera. In other words, the two images can be fused into a single image of 12 LEDs, of which at least 6 LEDs are non-collinear. More specifically, the two equation sets formulated from the two images are fused together, and each equation set includes 6 equations as in Eq. (1). Originally, the camera positions in these two equation sets are different, but after fusing, the fused equation set contains a single camera position and 12 distinct LEDs, of which 6 are non-collinear. Then, the equations corresponding to 6 non-collinear LEDs are selected to solve for the camera position.

3.2.2. Equation formulations for the 2 cameras fusion algorithm

Suppose that $\mathbf{C}$ is the world coordinate for camera 1 and $\mathbf{C}+\Delta\mathbf{C}$ is the world coordinate of camera 2. Let $\mathbf{X}_i$ denote the world coordinate of the ith LED. Also, let $\mathbf{x}_i^1$ and $\mathbf{x}_i^2$ denote the image coordinates of the ith LED on the two images captured by camera 1 and camera 2, respectively. We have

$$\begin{cases} \lambda_i^1 \mathbf{x}_i^1 = K \begin{bmatrix} R & -R\mathbf{C} \end{bmatrix} \mathbf{X}_i \\ \lambda_i^2 \mathbf{x}_i^2 = K \begin{bmatrix} R & -R(\mathbf{C}+\Delta\mathbf{C}) \end{bmatrix} \mathbf{X}_i. \end{cases} \tag{8}$$

From Lemma 1, Eq. (8) can be written as

$$\begin{cases} \lambda_i^1 \mathbf{x}_i^1 = K \begin{bmatrix} R & -R\mathbf{C} \end{bmatrix} \mathbf{X}_i \\ \lambda_i^2 \mathbf{x}_i^2 = K \begin{bmatrix} R & -R\mathbf{C} \end{bmatrix} (\mathbf{X}_i - \Delta\mathbf{C}). \end{cases} \tag{9}$$

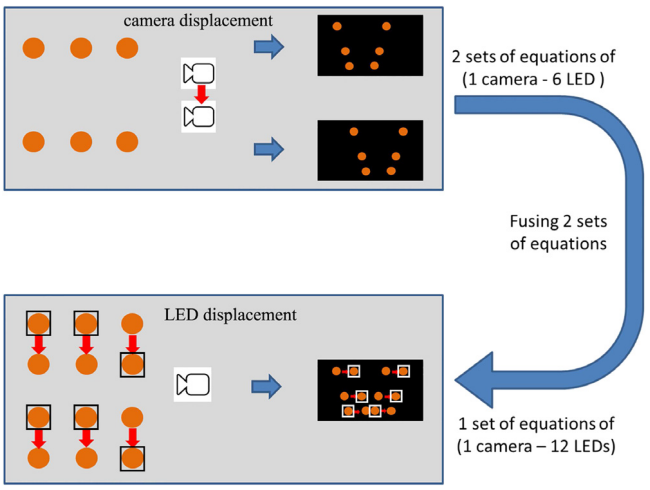


Fig. 5. Fusing equations from two cameras.

Let $P = K\begin{bmatrix} R & -R\mathbf{C} \end{bmatrix}$, with N LEDs, a set of $2N$ equations similar to Eq. (9) can be formulated. By applying DLT, a set of $2N$ linear equations is formulated:

$$\begin{bmatrix} \mathbf{0} & -\mathbf{X}_1^T & y_1^1\mathbf{X}_1^T \\ \mathbf{X}_1^T & \mathbf{0} & -x_1^1\mathbf{X}_1^T \\ \mathbf{0} & -(\mathbf{X}_1 - \Delta\mathbf{C})^T & y_1^2(\mathbf{X}_1 - \Delta\mathbf{C})^T \\ (\mathbf{X}_1 - \Delta\mathbf{C})^T & \mathbf{0} & -x_1^2(\mathbf{X}_1 - \Delta\mathbf{C})^T \\ \vdots & \vdots & \vdots \end{bmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} = 0, \tag{10}$$

where $\mathbf{x}_i^j = (x_i^j, y_i^j, 1)^T$ is the image coordinate of the ith LED in the jth camera.

Then, SVD is used to solve Eq. (10) to obtain the world coordinates for the camera.

3.3. Rolling shutter artifact compensation

3.3.1. The idea behind artifact compensation

As previously mentioned, all existing positioning algorithms do not deal with the rolling shutter artifact created when the camera moves relative to the LEDs. This section introduces a compensation method that can be applied to any positioning algorithm, including the ones using the collinearity condition, general ones, and the newly proposed one in this paper to compensate for the rolling shutter artifact.

From Eq. (6), the following equations need to be formulated to count for the rolling shutter artifact:

$$\begin{cases} \lambda_1\mathbf{x}_1 = K\begin{bmatrix} R & -R\mathbf{C}_1 \end{bmatrix}\mathbf{X}_1 \\ \lambda_2\mathbf{x}_2 = K\begin{bmatrix} R & -R\mathbf{C}_2 \end{bmatrix}\mathbf{X}_2 \\ \lambda_3\mathbf{x}_3 = K\begin{bmatrix} R & -R\mathbf{C}_3 \end{bmatrix}\mathbf{X}_3 \\ \qquad\vdots \end{cases} \tag{11}$$

where $\mathbf{C}_i$ is the world coordinate of the camera at the time the ith LED is captured.

Obviously, Eq. (11) cannot be solved since there are always more unknowns than the number of equations. However, if the differences between the camera world coordinate $\mathbf{C}_i$ are known, the camera displacement can be translated into the LED displacement thanks to Lemma 1. Then, Eq. (11) can be transformed into an equation for which the solution can be found.

More specifically, suppose that $\mathbf{C}$ is the camera position at the beginning of the exposure of the current frame and $\Delta\mathbf{c}_i$, $i = 1, 2, 3\ldots$ is the difference between $\mathbf{C}$ and the position of the camera while capturing the ith LEDs. By placing $\mathbf{C}_i = \mathbf{C} + \Delta\mathbf{c}_i$ in Eq. (11) and applying Lemma 1, Eq. (11) is transformed into a solvable equation:

$$\begin{cases} \lambda_1\mathbf{x}_1 = K\begin{bmatrix} R & -R\mathbf{C} \end{bmatrix}(\mathbf{X}_1 - \Delta\mathbf{c}_1) \\ \lambda_2\mathbf{x}_2 = K\begin{bmatrix} R & -R\mathbf{C} \end{bmatrix}(\mathbf{X}_2 - \Delta\mathbf{c}_2) \\ \lambda_3\mathbf{x}_3 = K\begin{bmatrix} R & -R\mathbf{C} \end{bmatrix}(\mathbf{X}_3 - \Delta\mathbf{c}_3) \\ \qquad\vdots\ . \end{cases} \tag{12}$$

Thus, to compensate for the rolling shutter artifact, the difference $\Delta\mathbf{c}_i$ between the camera positions needs to be determined first. Fortunately, $\Delta\mathbf{c}_i$ can be reasonably determined given that the speed of the vehicle and the readout time of the sensor are known. Let t_r denote the row readout time of the sensor, r_i denote the row at which the ith LED appears on the sensor. The elapsed time corresponding to the ith LED, denoted as Δt_i, is given as:

$$\Delta t_i = r_i t_r. \tag{13}$$

Suppose that the speed of the vehicle is $\mathbf{v} = (v_x, v_y, 0)^T$, the difference $\Delta\mathbf{c}_i$ can be determined as

$$\Delta\mathbf{c}_i = \Delta t_i \mathbf{v}. \tag{14}$$

After obtaining $\Delta\mathbf{c}_i$, Eq. (12) can be written as

$$\begin{cases} \lambda_1\mathbf{x}_1 = P(\mathbf{X}_1 - \Delta\mathbf{c}_1) \\ \lambda_2\mathbf{x}_2 = P(\mathbf{X}_2 - \Delta\mathbf{c}_2) \\ \lambda_3\mathbf{x}_3 = P(\mathbf{X}_3 - \Delta\mathbf{c}_3) \\ \qquad\vdots, \end{cases} \tag{15}$$

where $P = K\begin{bmatrix} R & -R\mathbf{C} \end{bmatrix}$.

Eq. (15) is the fundamental of the rolling shutter compensation. To apply the compensation for the general positioning algorithm, Eq. (15) is transformed into:

$$\begin{bmatrix} \mathbf{0} & -(\mathbf{X}_1 - \Delta\mathbf{c}_1)^T & y_1(\mathbf{X}_1 - \Delta\mathbf{c}_1)^T \\ (\mathbf{X}_1 - \Delta\mathbf{c}_1)^T & \mathbf{0} & -x_1(\mathbf{X}_1 - \Delta\mathbf{c}_1)^T \\ \mathbf{0} & -(\mathbf{X}_1 - \Delta\mathbf{c}_2)^T & y_2(\mathbf{X}_1 - \Delta\mathbf{c}_2)^T \\ (\mathbf{X}_1 - \Delta\mathbf{c}_2)^T & \mathbf{0} & -x_2(\mathbf{X}_1 - \Delta\mathbf{c}_2)^T \\ \vdots & \vdots & \vdots \end{bmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \end{pmatrix}. \tag{16}$$

Then, the SVD can be used to solve Eq. (16) to obtain the camera position.

3.3.2. Equation formulation for rolling shutter compensation in the proposed 2 camera fusion algorithm

Similar to Eq. (8), suppose that $\Delta\mathbf{C}$ is the difference in the position of two cameras. Let $\mathbf{x}_i^j$ denote the image coordinates of the ith LED on the image captured by the jth camera. Let $\Delta\mathbf{c}_i^j$ denote the difference between the position of the jth while capturing the image of the ith LED. The geometric relation between the world coordinate of each LED and its image coordinate on each camera is given as

$$\begin{cases} \lambda_i^1\mathbf{x}_i^1 = K\begin{bmatrix} R & R(\mathbf{C} + \Delta\mathbf{c}_i^1) \end{bmatrix}\mathbf{X}_i \\ \lambda_i^2\mathbf{x}_i^2 = K\begin{bmatrix} R & -R(\mathbf{C} + \Delta\mathbf{c}_i^2 + \Delta\mathbf{C}) \end{bmatrix}\mathbf{X}_i. \end{cases} \tag{17}$$

By applying Lemma 1 and DLT, with N LEDs, a set of $4N$ linear equations is formulated:

$$\begin{bmatrix} \mathbf{0} & -(\mathbf{X}_1 - \Delta\mathbf{c}_1^1)^T & y_1^1(\mathbf{X}_1 - \Delta\mathbf{c}_1^1)^T \\ (\mathbf{X}_1 - \Delta\mathbf{c}_1^1)^T & \mathbf{0} & -x_1^1(\mathbf{X}_1 - \Delta\mathbf{c}_1^1)^T \\ \mathbf{0} & -(\mathbf{X}_1 - \Delta\mathbf{c}_1^2 - \Delta\mathbf{C})^T & y_1^2(\mathbf{X}_1 - \Delta\mathbf{c}_1^2 - \Delta\mathbf{C})^T \\ (\mathbf{X}_1 - \Delta\mathbf{c}_1^2 - \Delta\mathbf{C})^T & \mathbf{0} & -x_1^2(\mathbf{X}_1 - \Delta\mathbf{c}_1^2 - \Delta\mathbf{C})^T \\ \vdots & \vdots & \vdots \end{bmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \end{pmatrix}, \tag{18}$$

Table 1
Simulation environment.

Parameter	Value
Sensor physical size	36 × 24 (mm^2)
Sensor resolution	3,600 × 2,400 (px)
Lens focal length	20 to 100 (mm)
Camera pose	(5,0,0) (degree)
Readout time	0.05 (s)
LED luminance	4,096 to 16,384 (cd/m^2)
ISO	100
Lens aperture	4
Exposure time	1/2,000 to 1/15 (s)
LED street light area	30 × 30 (cm^2)
Height of LEDs	7 (m)
Interdistance between LEDs	25 (m)
Lane width	10 (m)
Vehicle speed	0 to 100 (km/h)
Speedometer error	0 to 10%

where $\mathbf{x}_i^j = (x_i^j, y_i^j, 1)^T$.

Then, SVD can be used to solve Eq. (18) to obtain the camera position.

As mentioned earlier, at least 12 linear equations must be formulated in equation set (18) to determine the camera position. As can be seen in Fig. 5, each LED in the real world yields 2 mathematically effective LEDs, which are close to each other, corresponding to 2 cameras. And since each of these two effective LEDs can formulate 2 equations, each LED in the real world can end up formulating 4 equations in Eq. (18). So basically, the camera position can be determined with only 3 LEDs present in the real world. However, to increase the accuracy, the LEDs used to formulate the equations should be apart as much as possible. Therefore, even though each LED yields 2 mathematically effective LEDs, only one of them is selected to contribute 2 linear equations into Eq. (18) as illustrated in Fig. 5.

4. Simulations

4.1. Simulation environment

The simulations are conducted with the parameters listed in Table 1.

4.2. Simulation procedure

The simulation procedure is described in Fig. 6. The initial position of vehicle is assumed. Then the vehicle is assumed to move randomly along the street at a given speed. The camera is assumed to take picture after a small interval. After each interval, the LED image corresponding to current position of the vehicle is replicated. Then an LED detection algorithm is applied to this image to obtain the LED image coordinates. These LED image coordinates are tested with the proposed positioning algorithm and the performance is evaluated.

4.2.1. LED image replication

Fig. 7 describes the exposure and readout mechanism for a rolling shutter CMOS sensor. During the exposure period of the sensor, which is from T_{start} to T_{end}, the camera keeps moving along with the vehicle, and the LED projections on the image sensor keeps changing. If the LEDs appear on a row that is being exposed, they will be recorded in the image. For example, at a small period T_i, two LEDs appear on two rows of the sensor, the third and fifth, as illustrated in the figure. The image of the LEDs appearing on the third row will be recorded since this row is being exposed at T_i. The image of the LEDs appearing on the fifth row will not be recorded since this row does not start the exposure yet. Similarly, in the small period T_j, there are two LEDs appearing on the second and last row of the sensor. The LED appearing on the last row will be recorded while the one on the second row is not recorded because the second row has finished its exposure period. The simulation in this paper

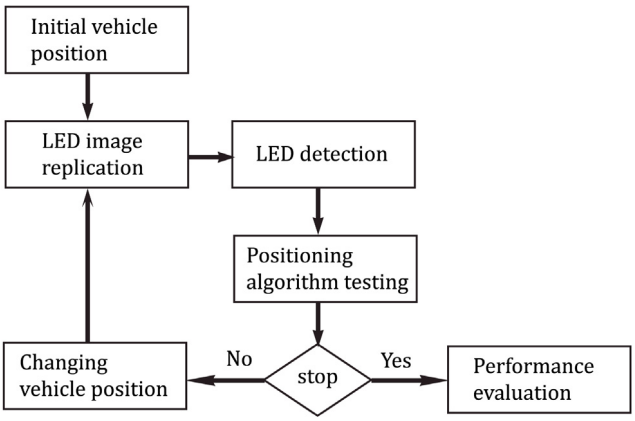


Fig. 6. Simulation procedure.

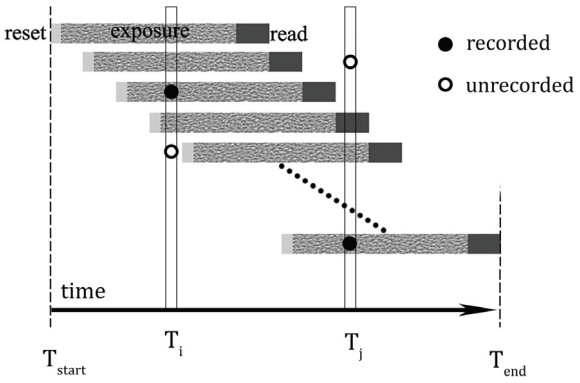


Fig. 7. Exposure and readout mechanism in CMOS sensor.

replicates this exact mechanism, and consequently the rolling shutter artifacts and motion blur will be created in the simulated image in the same way they are created in reality. The pseudo code of the simulation is given in Algorithm 1.

In Algorithm 1, $BG\_exptime$ and $LED\_exptime$ are the two matrices that contain the information of how long each pixel in the sensor are effectively exposed to background light and LED light. Since all pixels receive the background light during the exposure period, $BG\_exptime$ would be a matrix of which each element has the value equal the exposure time. However, elements in $LED\_exptime$ would have different values since some pixels receive light from LEDs in certain periods while others do not receive any LED light during their exposure period.

To calculate the value of each element in $LED\_exptime$, the exposure period of the entire sensor is divided into small period T_i. During this small period, the vehicle position is assumed to be unchanged. In each small period T_i, the current position of the camera cam_pos is calculated given the vehicle speed. The pose of the camera cam_pose is assumed to be unchanged during the exposure period. With the information about the camera position, camera pose, and the positions of all LEDs, the projections of all LEDs in the image sensor can be calculated using the function LEDS_PROJECTION. Thus, $exposed\_pxs$ is the set of all pixels in the sensor that are exposed to LED lights during the T_i period. However, some pixels in $exposed\_pxs$ are not effectively exposed in that the exposure period in these pixels have not started yet or finished already. Therefore, a set of pixels on_exp_pxs on the rows which the exposure period are happening is determined. Then the set of all pixels eff_exp_pxs which are effectively exposed to LED lights during the T_i period can be determined. Then the values of elements in $LED\_exptime$ corresponding to the pixels in eff_exp_pxs are added to T_i. After the period T_i goes from the start to the end of the exposure period of the

Algorithm 1 LED image replication

function IMAGE_INTEGRATION
  $sensor\_size \leftarrow$ physical size of sensor
  $sensor\_res \leftarrow$ resolution of sensor
  $leds\_pos \leftarrow$ position of all LEDs
  $cam\_pose \leftarrow$ pose of camera
  $exp\_time \leftarrow$ exposure time of sensor
  $row\_readtime \leftarrow$ readout time of one row in the sensor
  $BG\_exptime \leftarrow$ ONEs_matrix[$sensor\_res$] $\times$ exp_time
  $LED\_exptime \leftarrow$ ZEROs_matrix[$sensor\_res$]
  $T_i \leftarrow$ small period of time within the exposure period of sensor
  $current\_time \leftarrow T_{start}$
  while $current\_time < T_{end}$ **do**
    $cam\_pos \leftarrow$ current position of the vehicle
    $exposed\_pxs \leftarrow$ LEDS_PROJECTION($cam\_pose, cam\_pos, leds\_pos$)
    $on\_exp\_pxs \leftarrow$ set of all pixels which are on the exposure period
    $eff\_exp\_pxs \leftarrow$ set of all $exposed\_pxs$ in on_exp_pxs
    $LED\_exptime[eff\_exp\_pxs] \leftarrow LED\_exptime[eff\_exp\_pxs] + T_i$
    $current\_time \leftarrow current\_time + T_i$
  end while
  $BG\_exptime \leftarrow BG\_exptime + random\_BG\_noise$
  $LED\_exptime \leftarrow LED\_exptime + random\_LED\_noise$
  $pixels\_value \leftarrow$ GRAY_SCALE($BG\_exptime, LED\_exptime$)
  return $pixels\_value$
end function
function LEDS_PROJECTION(cam_pose,cam_pos,leds_pos)
  $sensor\_size \leftarrow$ sensor physical size
  $sensor\_res \leftarrow$ sensor resolution
  $M \leftarrow$ CAMERA_MATRIX($cam\_pose, cam\_pos, sensor\_size, sensor\_res$)
  $leds\_projection\_on\_sensor \leftarrow M \times leds\_pos$
  return $leds\_projection\_on\_sensor$
end function
function GRAY_SCALE(BG_time, LED_time)
  $N \leftarrow$ lens aperture
  $S \leftarrow$ the ISO assumed to be set in camera
  $gamma \leftarrow$ the gamma encoding value
  $LED\_lu \leftarrow$ LED luminance
  $BG\_il \leftarrow$ background light illuminance
  $ed \leftarrow$ ED($N, S, gamma, LED\_lu, BG\_il, BG\_time, LED\_time$)
  $gray\_scale \leftarrow 118 \times 2^{ed/gamma}$
  return $gray\_scale$
end function

entire sensor, the total amount of time that each pixel receives LED light during its exposure period can be determined. To replicate the random noise in the image, a small amount of random noise, which is assumed to range from 0 to 10% of the value of $BG\_exptime$ and $LED\_exptime$ depending on the ISO value is also added. Once knowing $BG\_exptime$ and $LED\_exptime$, the grayscale value of each pixel in the image can be calculated using the GRAY_SCALE function.

The pinhole camera model is assumed in the function LEDS_PROJECTION to find the projections of all LEDs in the image sensor. Given the camera extrinsic parameters including the pose and position and the camera intrinsic parameters including sensor size and resolution, the camera projection matrix M can be calculated. Then the projection of a single point in an LED on the image sensor can be determined by multiply this matrix to the coordinate of that point. Note that there is countless number of points in each LED and all these points have their own projection on the image sensor. In the simulation, the projections on the image sensor of only 4 points in the corners of each LED are determined. These 4 point projections define a region on the image sensor which is considered the projection of the entire LED on the image sensor.

To calculate the grayscale of each pixel, the shortcut method proposed in [1] is adopted. According to this method, given the lens aperture, the ISO of camera, the LED luminance, the background illuminance, and the amount of time that the pixel is exposed to LED and background light, the exposure difference ED can be calculated. From the gamma encoding value, which is usually assumed to be 2.2, and the exposure difference ED, the grayscale value in the range from 0 to 255 of the pixel can be calculated.

Simulated images with different effects are shown in Fig. 8. In Fig. 8(a), two simulated images corresponding to the vehicle speeds are assumed to be 0 km/h and 30 km/h, as shown. To demonstrate the rolling shutter artifact, the two images are combined and shown in one figure. The white rectangles are the images of the LEDs captured with a vehicle speed of 0 km/h while the gray ones are that at 30 km/h. The images of the LEDs corresponding to the vehicle speed of 30 km/h can be seen to appear closer than that at 0 km/h.

Fig. 8(b) shows two simulated images corresponding to two different positions of the camera. The first camera is placed 1 m on the right and 1 m behind the second camera. The two images are combined and are shown in one figure. The gray rectangles are the images of the LEDs captured by the first camera, while the white ones are that of the second camera.

Although motion blur is not intentionally dealt with in this paper, it is considered in the simulation for the fairness of the evaluation of the proposed algorithm. Fig. 8(c) shows a simulated image with motion blur occurring when the exposure time is set to 1/20 of a second and the vehicle speed is set to 100 km/h. Note that in most simulations in this paper, the exposure time is set to 1/2000 of a second, so there would be no motion blur in the simulated images.

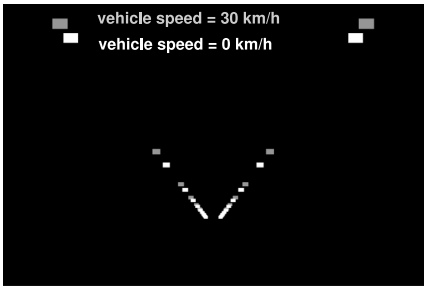
Another effect that might happen in reality of image capturing is blooming effect. This effect is caused by two reasons. The first reason is flaw of sensor with old technologies. When a pixel is overcharged, the current from that pixel will overflow to adjacent pixels and increase the charge and thus the pixel value of adjacent pixels. In modern image sensor, an anti-blooming gate is used and this phenomenon can be eliminated. The second reason is the flaws of lens. While the lens flaws are inevitable, the blooming effect can be effectively avoided by using the right exposure. In reality, the image might be overexposed due to wrong settings of the camera and the blooming effect might appear and degrade the positioning accuracy. This paper does not propose a solution to deal with the blooming effect when it appears in the image. However, to show its effect on the positioning accuracy, the LED images replicated with blooming effect are used to test the proposed algorithm. To replicate the blooming effect, the original simulated LED image is applied multiple Gaussian blur filters. Then the original simulated LED image and the blurred image are add together to create the LED image with blooming effect as shown in Fig. 8(d). In this figure, the blooming region covering each LED is three times the size of that LED.

4.2.2. LED detection

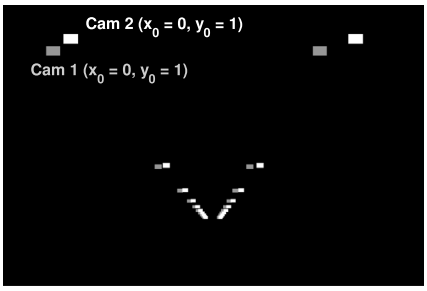
A simple algorithm, which is described in Algorithm 2 is used to detect all LEDs in the image. Firstly, a morphological open operation is used to estimate the background of the image. Then an LED image without background can be obtained. After that, this image is processed to increase the contrast and then binarized to obtain a binary image. Finally, the connected components, which are LEDs, in the binary image are found. In the simulation, all the steps described in Algorithm 2 are performed using available functions implemented in Matlab Image Processing Toolbox.

4.2.3. Algorithm evaluation

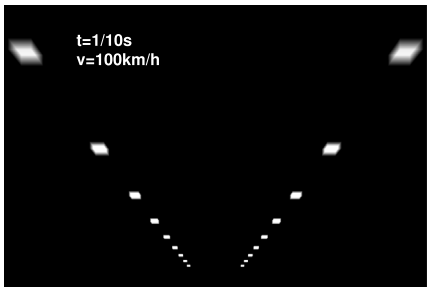
In the simulation, two versions of the proposed positioning algorithm are implemented. The first version uses two cameras without rolling shutter compensation. The second version uses two cameras with rolling shutter compensation. Then the performance of these two versions is compared to show the effectiveness of the rolling shutter compensation.



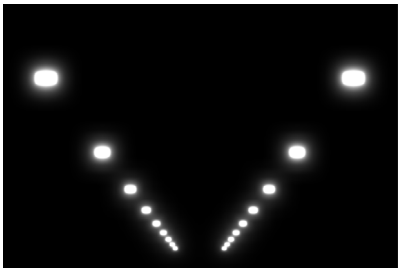
(a) Images with rolling shutter artifact.



(b) Images from two cameras.



(c) Images with motion blur.



(d) Images with blooming effect.

Fig. 8. Simulated images of LED street lights.

Algorithm 2 LED detection

```
procedure LED DETECTION
    LED_image ← input image
    background ← MORPHOLOGICAL_OPEN(LED_image)
    no_BG_image ← LED_image − background
    high_CT_image ← INCREASE_CONTRAST(noBG_image)
    BW_image ← BINARIZE(high_CT_image)          ▷ Using Otsu's thresholding method
    LED_list ← FIND_CONNECTED_COMPONENTS(BW_image)
end procedure
```

The performances of the proposed algorithm and existing ones are not entirely comparable since the simulation environments in these works are different. However, brief comparisons on the positioning accuracy of the proposed algorithm and existing ones are still being made to give an idea on the benefit of using street light beacons for positioning as mentioned before.

4.3. Simulation results

This section reveals the impact of each of the system parameters on the performance of the algorithms by showing the positioning accuracy corresponding to different settings for that parameter. The positioning accuracy is affected by multiple parameters, so when the impact of a parameter is shown, the other parameters are fixed or the average results corresponding to different values of these parameters are considered.

4.3.1. Error corresponding to different vehicle speeds

To show the impact of the vehicle speed on the accuracy of the algorithms, the simulation was conducted with the vehicle speed varying from 0 to 100 km/h. The vehicle was assumed to travel from 0 to 100 m, and its position was estimated after every 0.1 m of movement. The average and maximum positioning errors for this distance corresponding to different vehicle speeds are shown in Fig. 9.

Fig. 9 shows that the average and maximum error of the proposed algorithm without compensation increase when the vehicle speed increases. This is expected since the displacement of the camera between the moments when different LEDs are captured in the image becomes larger when the vehicle moves faster.

Regarding the average and maximum errors of the proposed algorithm with compensation, Fig. 9 shows seemingly counterintuitive results. More specifically, the average and maximum error with compensation decrease when the vehicle speed increases. This result is due to better positioning accuracy if closer LEDs are chosen to formulate the equations that are to be solved to obtain the vehicle position. This fact is described and explained in Fig. 10.

An image captured by the camera might contain various LEDs that can be used to formulate geometric equations that are to be solved for the vehicle position. However, the LED that is closer to the camera would produce more accurate information regarding the vehicle position. For example, in Fig. 10, the LED (1) is closer to the camera than LED (2) is. Therefore, the equation formulated from the LED (1) would lead to a more accurate solution. The reason for the high accuracy of information given by the close LED is that the data obtained from the image is not continuous but quantized. Since the image sensor consists of discrete pixels, different points in the real world can be mapped to a single quantization point, which is a pixel in the image sensor, and this mapping depends on the distance between the camera and the points. As illustrated in Fig. 10, the distance between points (1) and (2) equals that between points (3) and (4). However, (1) and (2) are far from the camera and are thus mapped to a single pixel, so these two points would not be distinguished in the image. On the other hand, (3) and (4) are close to the camera and are mapped to different pixels. Therefore, (3) and (4) can be distinguished in the image.

Recall the simulation shown in Fig. 9 in which the vehicle moves faster, the LEDs would be closer to the camera and thus the equations formulated from these LEDs would offer a more accurate solution. The result of the simulation is valid since the information related to the vehicle speed is assumed to be accurately known, making the error caused by the rolling shutter artifact to be completely compensated. However, in reality, the information regarding the vehicle speed always contains errors, and the scale of this error usually depends on the speed. To be more specific, this error would be higher if the speed is higher. Usually, the error of a typical speedometer in a car is 10%. A simulation was conducted with this assumption, and the results are shown in Fig. 11.

Fig. 11 shows that with 10% error in the vehicle speed, the average and maximum error of the algorithm with compensation remain stable as the vehicle speed increases.

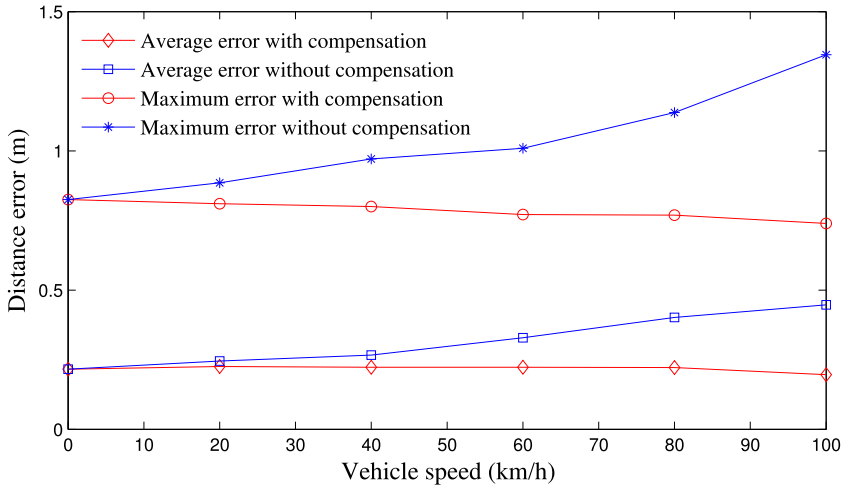


Fig. 9. Positioning error corresponding to speeds [sensor resolution = 3,600 × 2,400, focal length = 35 mm, exposure time = 1/2,000 s, vehicle traveling distance = 0 to 100 m, sampling pace = 0.1 m, speedometer error = 0%].

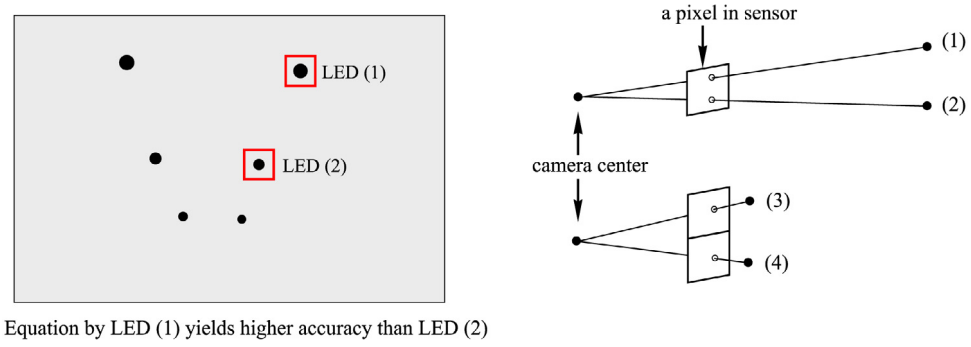


Fig. 10. Effect of the distance from camera to LED on positioning accuracy.

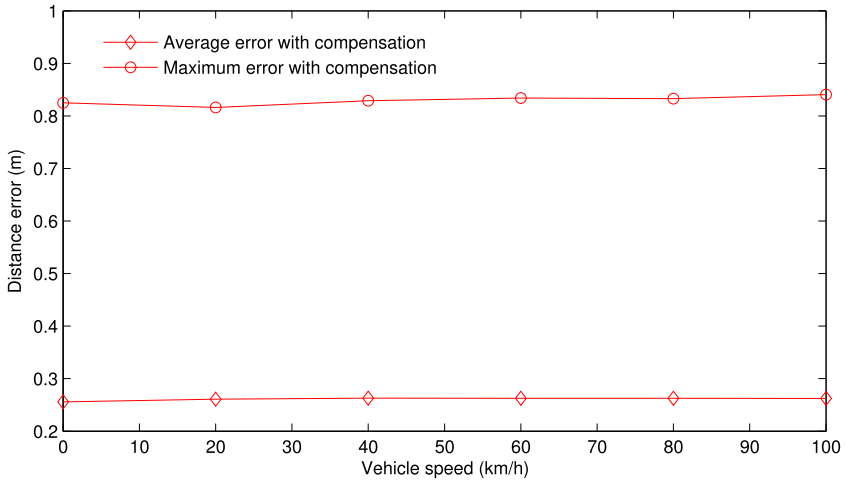
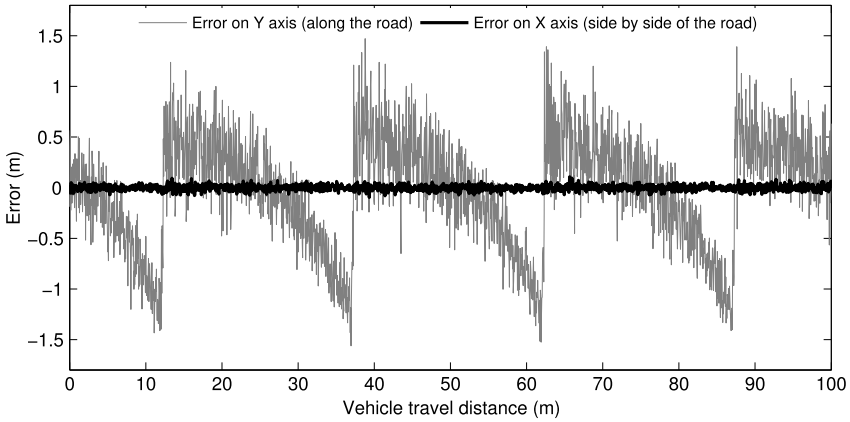


Fig. 11. Positioning error corresponding to various speeds [sensor resolution = 3,600 × 2,400, focal length = 35 mm, exposure time = 1/2,000 s, vehicle traveling distance = 0 to 100 m, sampling pace = 0.1 m, speedometer error = 10%].

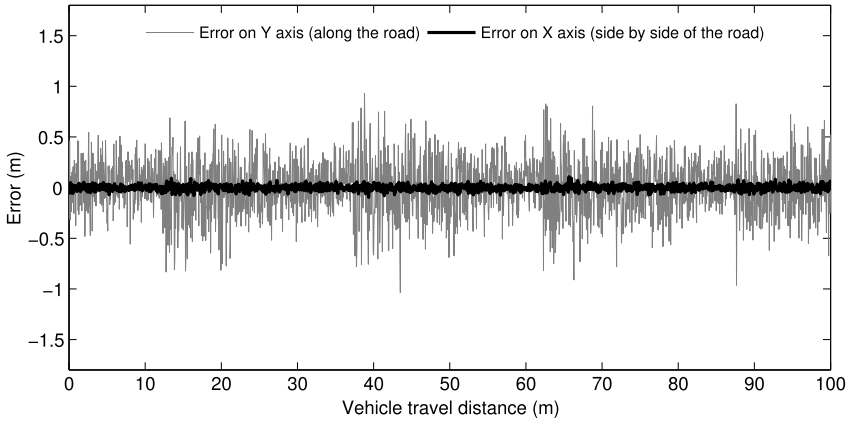
4.3.2. Error corresponding to different vehicle travel distances

In the simulations shown in Figs. 9 and 11, the error corresponding to a single value of the vehicle speed is evaluated by the average error estimated during the time the vehicle moves from 0 m to 100 m, not by the error at a specific position of the vehicle. The reason for this is that the positioning error greatly depends on the position of the vehicle relative to the LEDs. Therefore, the error would vary when the vehicle moves on the road, as shown in Fig. 12.

In the simulation shown in Fig. 12, the vehicle position is estimated after every 0.05 m of motion of the vehicle. The vehicle speed is set to 100 km/h, and the error of the speedometer is assumed to be 10%. The positioning errors on the two axes of the proposed algorithm without and with compensation are shown in Figs. 12(a) and 12(b), respectively. Both figures show that the errors on the X axis, which is the axis defining the position of vehicle from the left side to the right side of the road, are very small compared to those in the Y axis, which is the axis defining



(a) Error without compensation.



(b) Error with compensation.

Fig. 12. Error corresponding to various vehicle travel distances [sensor resolution = 3,600 × 2,400, focal length = 35 mm, exposure time = 1/2,000 s, vehicle traveling distance = 0 to 100 m, sampling pace = 0.05 m, speedometer error = 10%].

how far the vehicle has moved. Also, the errors on the X axis are almost identical with and without compensation. However, the compensation makes a big difference in the errors on the Y axis because the vehicle mainly moves along the Y axis of the road, and hence, the rolling shutter artifact causes most of its impact on the positioning error on this axis.

Note that in Fig. 12(a), there are sudden changes in the errors on the Y axis because as the vehicle moves forward, the LEDs gradually move out from the field of view of the camera. At a specific travel distance of the vehicle, this nearest LED disappears completely in the image, causing the sudden change in the error. In addition, since the distances from the vehicle to the LEDs change in a recursive manner as the vehicle moves, the errors have repetitive patterns, as shown in Fig. 12.

4.3.3. Error corresponding to resolutions of sensor

As one can expect, the positioning accuracy of the proposed algorithm is degraded when low-resolution sensors are used. Fig. 13 shows the errors corresponding to different resolutions of the sensor. The proposed algorithms, with and without the rolling shutter artifact compensation, achieve decent performance at sensor resolutions of 4 to 6 megapixels. Also, the performance does not improve much beyond a sensor resolution of 6 megapixels.

4.3.4. Error corresponding to the exposure time

The fast movement of the vehicles and long exposure time introduce motion blur in the LED in the image, which is actually a big problem for any camera-based VLC communication and positioning system since it increases the difficulty in detecting the LED in the captured image. This paper is concerned only with the positioning problem, and LED detection, which is the step prior to positioning, will not be dealt with.

Regarding positioning, motion blur causes errors in determining the image coordinates of the LED and thus in positioning. When the LED image is blurred due to motion, as illustrated in Fig. 14, the image coordinates of the LED are considered to be the average coordinates of all pixels of the blurred LED image. Obviously, this average coordinate differs from the non-blurred image coordinate of the LED.

To examine the effect of the exposure time on the positioning accuracy, a simulation is conducted with the exposure time ranging from 1/2000 s to 1/15 s, and the result is shown in Fig. 15. Note that in the simulation, the background pixels in the simulated image are almost dark, and thus, the LEDs can be detected perfectly regardless of the presence of motion blur. In other words, this simulation only shows the effect of the motion blur on positioning, and not LED detection.

The error of the proposed algorithm without compensation can be seen to slightly increase when the motion blur increases, that is, when the exposure time increases. However, the average and maximum error of the proposed positioning algorithm with compensation remain the same when the exposure time increases. This can be explained through Fig. 16. As previously mentioned, the image coordinates of the blurred LED are determined as the average coordinates of all pixels of the blurred image, and these average coordinates are actually the image coordinates of the LED captured right at the middle of the period at which the LED is exposed to the sensor, as illustrated in Fig. 16. Therefore, the motion blur basically exerts the same kind of effect on positioning as the rolling shutter artifact does, yet at a greater level. When the rolling shutter compensation is applied, the camera displacement due to the movement of vehicle from the original position to the position at the middle of the LED exposure period is compensated.

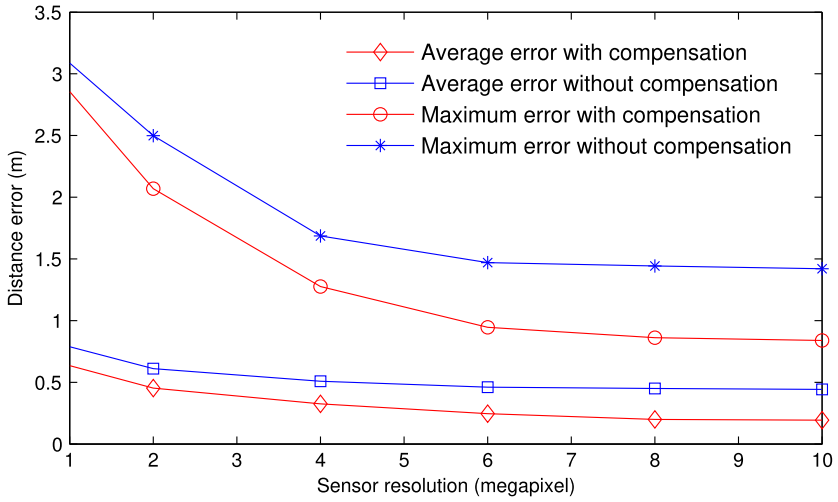


Fig. 13. Positioning error corresponding to the sensor resolution [sensor resolution = 1 to 8 mpx, focal length = 35 mm, exposure time = 1/2,000 s, vehicle traveling distance = 0 to 100 m, sampling pace = 0.1 m, speedometer error = 10%].

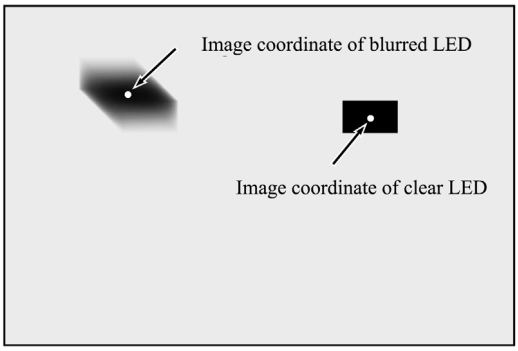


Fig. 14. Image coordinate of blurred LED image.

In other words, the proposed rolling shutter compensation actually solved the problem in positioning caused by the motion blur.

Although the proposed compensation method solves the motion blur problem, it is worthwhile to mention that the motion blur is not an issue in most cases for the system considered in this paper because the luminance of a typical LED is very high at 4,096 to 16,384 cd/m^2. In addition, the background in the image should be as dark as possible for convenience of the LED detection. Consequently, the exposure time can be set to a very high level at which motion blur does not occur. For example, there is no motion blur at an exposure time of 1/2000 s, which is assumed for most simulations in this paper.

4.3.5. *Error corresponding to the focal length of the lens*

The errors corresponding to the different focal lengths of the lens are shown in Fig. 17. Both errors with and without compensation can be seen to decrease when the focal length increases from 20 mm to 50 mm because with a longer focal length of the lens, different LEDs in the real world are separated more in the image, thus providing more accurate information for the positioning algorithm. However, the errors start to increase when the focal length exceeds 55 mm because the difference in the view of the two cameras becomes larger as the focal length increases. At some position of the vehicle, the LEDs nearest to one camera might not be seen by the other, as illustrated in Fig. 18 where the nearest LED in the top left side of the image is visible to camera 1 but not to camera 2. Consequently, the algorithm has to replace this LED with another LED farther from the camera when formulating the equations. However, the information provided by the replacement LED is less accurate for two reasons. First, the spatial information of the LED is rounded off more due to the farther distance, as explained in Fig. 10. Second, the LED is affected more by the rolling shutter effect since it appears at a lower position in the captured image. The use of the replaced LEDs at some position of the vehicle is the reason for the increase in the maximum error with and without compensation, as

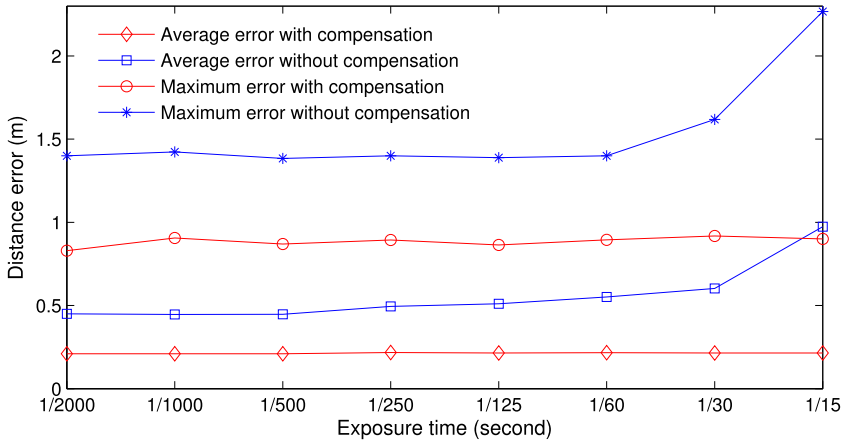


Fig. 15. Positioning error corresponding to the exposure time [sensor resolution = 3,600 × 2,400, focal length = 35 mm, exposure time = 1/2,000 s to 1/15 s, vehicle traveling distance = 0 to 100 m, sampling pace = 0.1 m, speedometer error = 10%].

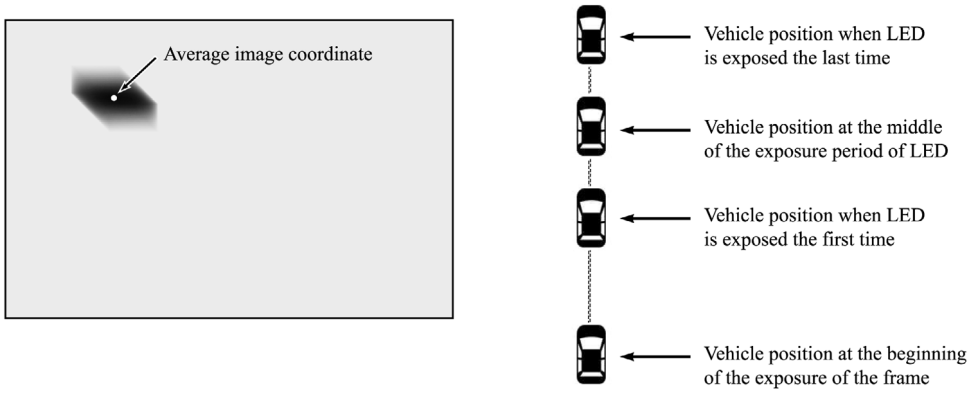


Fig. 16. Image coordinates of the blurred LED image and corresponding vehicle position.

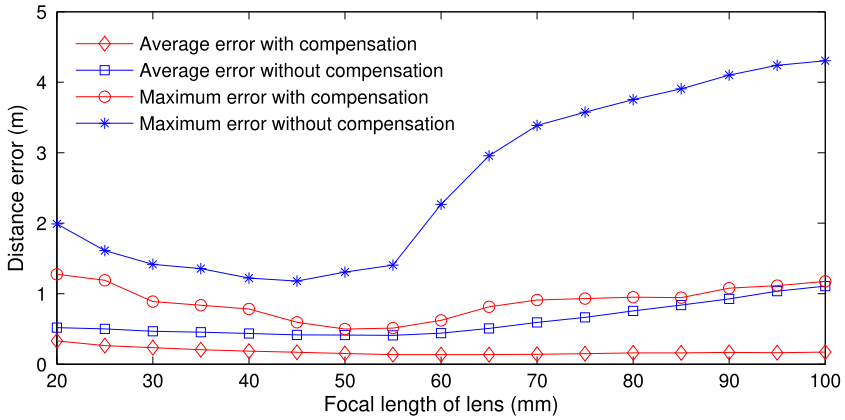


Fig. 17. Positioning error corresponding to the focal length of the lens [sensor resolution = 3,600 × 2,400, focal length = 20 to 100 mm, exposure time = 1/2,000 s, vehicle traveling distance = 0 to 100 m, sampling pace = 0.1 m, speedometer error = 10%].

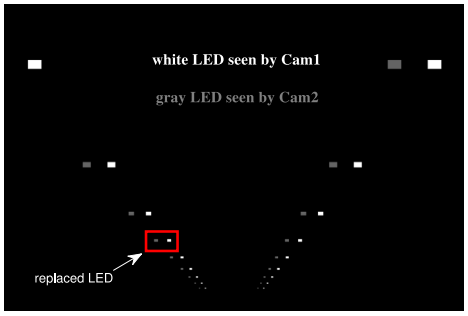


Fig. 18. LEDs captured by two cameras with a 100 mm lens.

shown in Fig. 17. When compensation is applied, the error due to the use of the replaced LED is reduced since the error due to the rolling shutter artifact is removed. As a result, the maximum error with the compensation does not increase dramatically as for the one without compensation. With compensation, the disadvantages and advantage of a long focal length of the lens cancel each other out, resulting in a steady average error corresponding to a focal length from 55 mm to 100 mm. Without compensation, the long focal length of the lens has more disadvantages than advantages, and hence the average error increase when the focal length increases.

4.3.6. Error corresponding to blooming effect level

As mentioned earlier, blooming effect can be effectively avoided by using the right exposure for the LED. In this simulation, the LED images are assumed to be overexposed and thus the blooming effect occurs at various levels. In the simulation, the level k of the blooming effect means that the blooming regions covering the LEDs are k times the size of the LEDs. The simulation result is shown in Fig. 19. It can be seen that the errors increase when the level of the blooming effect increases. This is because the blooming effect causes difficulties in detecting the pixel coordinates of the LEDs and thus degrade the positioning accuracy. However, the impact of the blooming effect is not high since this effect happens evenly around the exact position of the LEDs in the image. When the whole pixels of a LED in the image are detected, the LED pixel coordinate is calculated as the average pixel coordinate. Therefore, the final pixel coordinates that are given to the positioning algorithm do not change greatly.

5. Experiment

5.1. Experiment environment

The experiment was conducted in indoor environment with 14 LEDs placed collinearly along two lines in the ceiling of a room as shown in Fig. 20. The height of the ceiling is 2.5 m. The longitudinal and lateral distances between any 2 LEDs are 0.6 m. Each LED has a round shape with the diameter of 5 cm. The luminous flux of each LED is 800lm. The cameras used in the experiment were Sony RX100 V with the sensor size of 13.2 mm×8.8 mm, the sensor resolution of 5472×3648 pixels, and the lens focal length of 9 mm. The two cameras were placed at the height of 2.1 m under the ceilling. The distance between the two cameras was 0.3 m. For simplicity, LEDs only radiate unmodulated light. The world coordinates of the LEDs were assumed to be known by the camera. Because of the spatial limitation as well as the difficulties in measuring

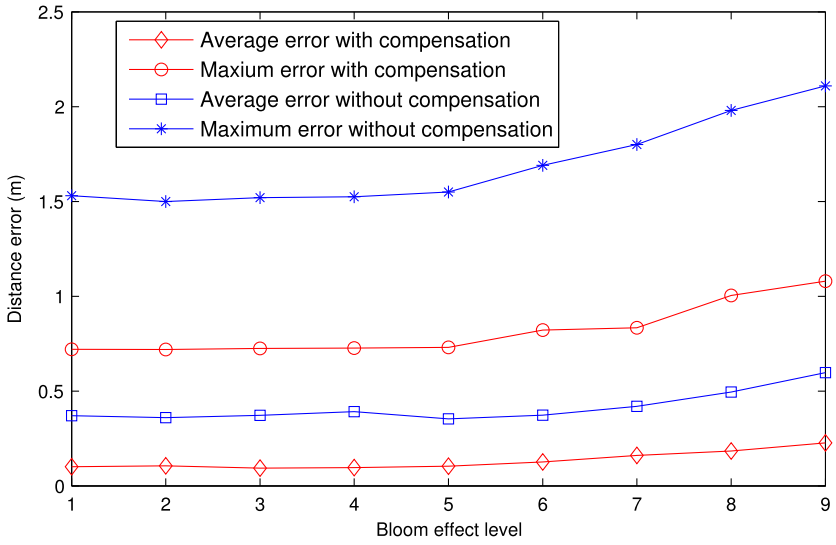


Fig. 19. Positioning error corresponding to blooming effect level [sensor resolution = 3,600 × 2,400, focal length = 35 mm, exposure time = 1/2,000 s, vehicle traveling distance = 0 to 100 m, sampling pace = 0.1 m, speedometer error = 10%].

Fig. 20. LED arrangement.

the speed of the cameras in indoor environment, the cameras were not moving in the experiment. The images of LEDs were captured by the cameras and then processed in a computer with the proposed algorithm implemented to find the position of the cameras. The positioning errors were calculated as the difference between the camera positions obtained by manual measurement and those by the proposed algorithm.

5.2. Experiment results

The purpose of the experiment was twofold: to demonstrate the validity of the proposed algorithm in solving the collinear arrangement of LEDs and to give the insight into the performance of the proposed algorithm in real applications. Regarding the collinear arrangement problem, the experiment results showed that with proper settings of the camera exposure, the proposed positioning algorithm could successfully determine the position of the camera with reasonable accuracies. Without the proposed algorithm, positioning of the camera was not possible in the sense that the errors were unacceptably high, which were tens of the size of the experimental room.

In real applications, the LED images might not be as clear as that in simulations due to the blooming effect which might lead to the positioning errors. Therefore, to have the insight into the performance of the proposed algorithm in real applications, several experiments were conducted at different levels of blooming effect, and the positioning errors were calculated.

To obtain different levels of blooming effect in the image, the LEDs were captured at different exposure settings of the camera. More specifically, the exposure time of the cameras was set to 14 levels ranging from 1/32000 s to 1/5 s, each of which roughly doubles the other as illustrated in Fig. 21.

The problem of the blooming effect is that it introduces errors in detecting the LED pixel coordinates as shown in Fig. 22. To get this experiment result, firstly the camera was kept stable at a specific position. Then at each exposure setting of the camera, multiple images of LEDs were taken. These images were processed to find the pixel coordinates of the LEDs. The standard deviations of the found pixel coordinates in images taken at the same exposures were calculated. Since these standard deviations depend on the size of the LEDs in the images, the normalized LED pixel coordinate detection errors were calculated as the percentages of the standard deviations of pixel coordinates over the size of the LEDs measured in the number of pixels.

At the lowest exposure, the blooming effect was invisible in the image and thus considered to have level 0 in Fig. 22. The blooming effect level is considered to increase 1 level when the exposure of the camera increases 1 stop, that is when the exposure time doubles. It can be seen through the figure that the errors in detecting the LED pixel coordinate increases when the blooming effect increases. These pixel coordinate errors led to the errors in determining the position of the camera, which is shown in Fig. 23.

It can be seen in Fig. 23 that at low levels of blooming effect, the proposed algorithm achieved high positioning accuracies, which were below 0.5 m. The positioning errors increased when the blooming effect increased. Especially, the positioning errors drastically increased above level 8 of blooming effect. At the level 12 and 13 of blooming effect, the camera position was unable to be determined in the sense that the errors went too high and hence the performances at these levels of blooming effect were not shown in Fig. 23.

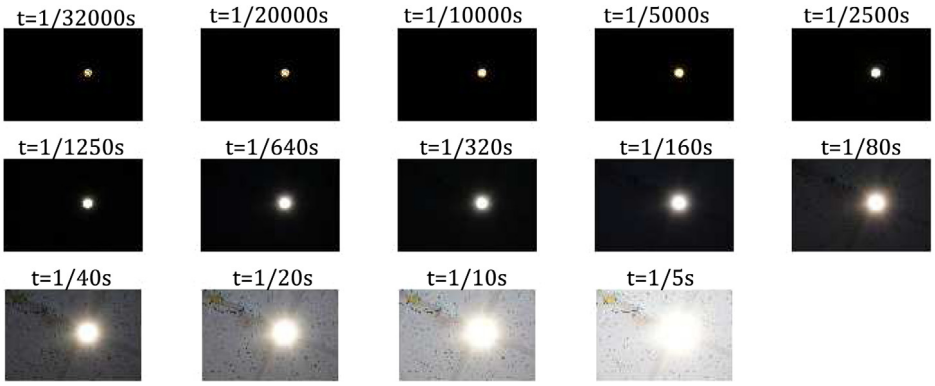


Fig. 21. LEDs captured at different levels of blooming effect.

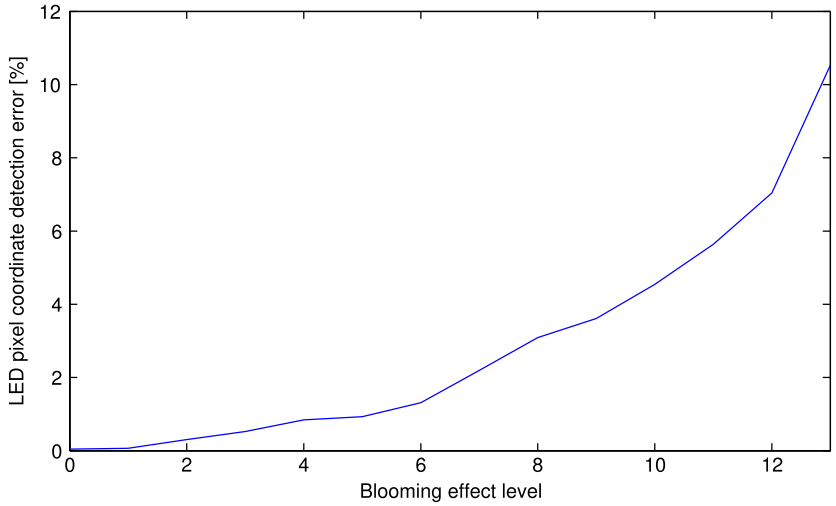


Fig. 22. Error of detected LED pixel coordinates corresponding to blooming effect.

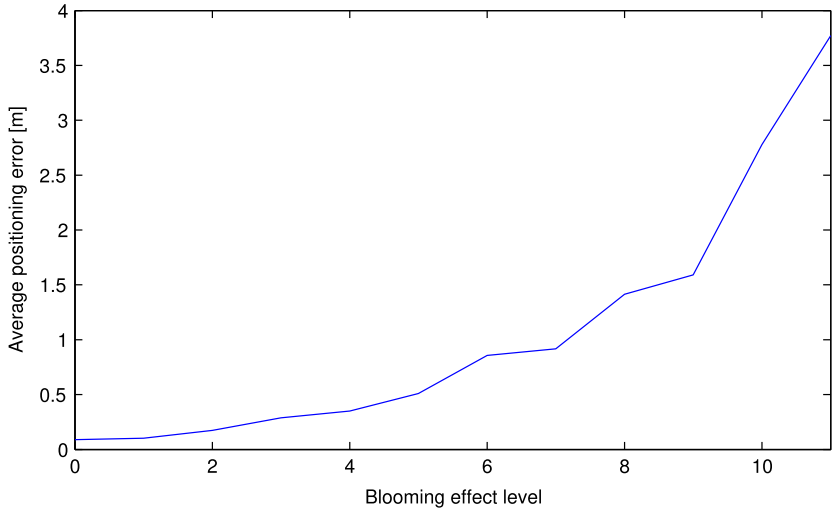


Fig. 23. Positioning error corresponding to blooming effect.

Although the experiment results show that the performance of the proposed algorithm might get worse at the high levels of blooming effect, they also show that there is a wide range of exposure settings of the cameras which allow high positioning accuracies. For example, in the experiment, the exposure time of the camera can be set from 1/32000 s to 1/2500 s to obtain the positioning accuracies of less than 0.5 m. This suggests that in real applications, it is not necessary to have an extremely precise exposure setting for the camera to mitigate the blooming effect completely to achieve high positioning accuracy. Therefore, it can be concluded that the proposed positioning algorithm will work properly in real applications without having many problems related to blooming effect.

6. Conclusion

This paper considers a VLC-based vehicle positioning in which the vehicle, attached with a CMOS camera uses the signal transmitted from LED street lights to determine its position. This paper has two major contributions. First, the collinear arrangement of LED street lights makes it impossible for traditional positioning algorithms to determine the position of the vehicle. As such, this paper proposes an algorithm that fuses the information received from two cameras to solve the problem of a collinear LED arrangement to determine the position of the vehicle. Second, the CMOS sensor combined with the movement of the vehicle creates rolling shutter artifacts that negatively impact the positioning accuracy. This paper then proposed a method to compensate for rolling shutter artifacts in order to achieve a high positioning accuracy even when the vehicle moves at a high speed.

The performance of the proposed algorithms is verified through simulations. It is found through the simulations that the positioning accuracy is affected by many system parameters. However, in all of the cases, the positioning accuracy with the proposed compensation is always higher than that without compensation. Regarding the vehicle speed, the proposed compensation keeps the error stable regardless of the speed of the vehicle. It is also found that the positioning error varies depending on the position of the vehicle relative to the LED street lights. By using the compensation, these variations of errors are almost eliminated and thus make the performance of the positioning system stable at all vehicle travel distances. When it comes to sensor resolution, the error without compensation noticeably increases while that with compensation just slightly increases when the resolution decreases. The exposure time is found to have small impact on the positioning results. Without compensation, the accuracy slightly decreases when the exposure time increase. When the compensation is applied, the accuracy remains the same at all levels of exposure time. The simulations with changing values of the lens focal length give interesting results. The simulation results show that the proposed positioning algorithm, with and without compensation, do its best at the focal length of 50 mm to 55 mm of the lens. Although these values might change from systems to systems, this result implies that there is an optimal focal length of the lens that yields the best performance for each specific system. The proposed positioning algorithm is also tested at different levels of blooming effect in the image. The simulation results show that while the positioning accuracy does decrease when the level of blooming effect increases, the impact of the blooming effect is not high.

Although the accuracy of the proposed system varies depending on system parameters, the maximum horizontal error is less than 10 cm in any cases of the simulation. And with normal setting, the average and maximum vertical errors are 15 cm and 150 cm, respectively. In the future, the vertical errors is expected to be further improved when filters such as Kalman filters, which can fine-tunes the positioning result by using knowledge on the physical model of the vehicle movement, are applied to the proposed system.

To provide the insight into the performance of the proposed system in real applications, small-scale experiments were also conducted. The results suggest that the proposed algorithm can achieve high accuracies in real applications.

Acknowledgment

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF), funded by the Ministry of Education, Science and Technology (No. 2015R1A2A2A01006431).

References

[1] T.H. Do, M. Yoo, Performance analysis of visible light communication using cmos sensorse, Sensors 16 (2016) 309.
[2] P. J, H.M. Tsai, C. Wang, F. Liu, Vehicular visible light communications with led taillight and rolling shutter camera, in: IEEE 79th Vehicular Technology Conference (VTC Spring), 18–21, 2014, pp. 1–6.
[3] I. Takai, K.Y.S. Ito, K. Kagawa, M. Andoh, S. Kawahito, Led and cmos image sensor based optical wireless communication system for automotive application, IEEE Photon. J. 5 (5) (2013) 6801418.
[4] Y.S. Kuo, P. Pannuto, K.J. Hsiao, P. Dutta, Indoor positioning with mobile phones and visible light, in: Proceedings of the 20th Annual International Conference on Mobile Computing and Networking, 2014, pp. 447–458.
[5] N. Rajagopal, P. Lazik, A. Rowe, Visual light landmarks for mobile device, in: Proceedings of the 13th International Symposium on Information Processing in Sensor Networks, 2014, pp. 249–260.
[6] M. Liu, F.C.K. Qiu, S. Li, B. Hussain, L. Wu, C.P. Yue, Towards indoor localization using visible light communication for consumer electronic devices, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2014, pp. 143–148.
[7] F.C. Commission, Wireless e911 location accuracy requirements, Ps Docket 19 (1—12) (2015) 07–114.
[8] K.Patrick McDermott, On the improvement of positioning in lte with collaboration and pressure sensors, Ph.D. dissertaiton.
[9] R. Roberts, P. Gopalakrishnan, S. Rathi, Visible light positioning: automotive use case, in: Vehicular Networking Conference (VNC), IEEE, 2010, pp. 309–314.
[10] Visible light positioning based on led traffic light and photodiode, in: Vehicular Technology Conference (VTC Fall), 2011, pp. 1–5.
[11] M.S. Ifthekhar, N. Saha, Y.M. Jang, Stereo-vision-based cooperative-vehicle positioning using occ and neural networks, Opt. Commun. 352 (2015) 166–180.
[12] T.H. Do, M. Yoo, An in-depth survey of visible light communication based positioning systems, Sensors 16 (2016) 678.
[13] Y. Nakazawa, H. Makino, K. Nishimori, D. Wakatsuki, H. Komagata, Indoor positioning using a high-speed, fish-eye lens-equipped camera in visible light communication, in: Proceedings of the Int. Conf. on Indoor Positioning and Indoor Navigation (IPIN), 2013, pp. 1–8.
[14] T. Tanaka, S. Haruyama, New position detection method using image sensor and visible light leds, in: Proceedings of the Second Int. Conf. on Machine Vision, 2009, pp. 150–153.
[15] R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, second ed. Cambridge University Press, Cambridge CB2 8RU, UK.