Int. J. Advanced Networking and Applications
Volume: 17 Issue: 01 Pages: 6761-671 (2025) ISSN: 0975-0290


# A Method to Detect Corner Points and Feature Lines Based On Edges in Images in Visual SLAM

**Tok Gil Kang**
Institute of Information Technology, High-Tech Research and Development Center,
**Kim Il Sung** University, Pyongyang,Democratic People's Republic of Korea
Email: yonghaksin@163.com
**Jae Hyon Kim**
Institute of Information Technology, High-Tech Research and Development Center,
**Kim Il Sung** University, Pyongyang,Democratic People's Republic of Korea
Email: jonggunkim@163.com
**Jong Hyok Ri**
Institute of Information Technology, High-Tech Research and Development Center,
**Kim Il Sung** University, Pyongyang,Democratic People's Republic of Korea
Email: jonggunkim@163.com
**Jin Hyok Yom**
Institute of Information Technology, High-Tech Research and Development Center,
**Kim Il Sung** University, Pyongyang,Democratic People's Republic of Korea
Email: jonggunkim@163.com
Institute of Information Technology, High-Tech Research and Development Center,
**Kim Il Sung** University, Pyongyang, Democratic People's Republic of Korea
Email: abcdef@gmail.com

**ABSTRACT**

Points and feature lines correctly even in different brightness and noise conditions. With the purpose of making a contribution to the improvement of the performance of ORB (Oriented FAST and Rotated BRIEF) corner detection which is widely used in SLAM system, this paper proposes a method to determine corner points and feature lines based on gradient direction changes calculated at the edge points detected in the image. First, the curve of edge points is represented by a multivalued function to be used as information about feature lines. Then, a method to detect corners on edges by examining the change in the gradient direction at the edge points, not by calculating curvatures of the edge point curve, is proposed. The part of the edge curve represented by a multivalued function near corners has been emphasized to be used as information about feature lines. Finally, the proposed method was tested with EuRoC dataset images in terms of stereo camera visual SLAM and the test results show that the proposed method detected corner points better than edge-based corner detectors based on point-to-centroid distance(PCD), Simple Triangular Theory and Curvature Scale Space(CCS), and ORB.

Keywords - **Corner point, Edge, Gradient direction, multivalued function, gradient direction**

Date of Submission: May 16, 2025 Date of Acceptance: June 23, 2025

## 1. INTRODUCTION

In the field of computer vision, corner detection is a widely addressed issue for visual localization, motion tracking, object tracking, image retrieval and 3D reconstruction. Correct and repeatable detection of all possible corners is of great importance in visual localization, motion tracking, 3D reconstruction as well as SLAM. The main performance indicators of the corner detection in SLAM refer to its detection accuracy, localization ability, computational efficiency and detection repeatability under diverse conditions of image scaling and rotation, noise corruption, JPEG compression, illumination and viewpoint changes.

Corners could be detected by several methods including derivative-based method, edge-based method, template-based method and multi-scale analysis-based method [3].

First, Harris corner detector [4], Hessian matrix-based method [5], Hilbert transformation method [6], wavelet and Gabor wavelet-based method [7] as the classical detectors were proposed, and afterwards, a great deal of research works has been conducted to improve the performance of corner evaluation functions [3].

Template-based methods detect corners by using local image regions and matching parameterized templates. Typical detectors based on this method are SUSAN [8] and FAST [1, 2]. FAST algorithm with its superior performance has been widely used in many computer vision applications. However, derivative-based and template-based methods are sensitive to noise, Affine transforms and illumination variations, thus they are still not effective enough in many cases of image matching and reproduction. And the parameterized corner templates reveal shortcomings that they could not handle the corners with different directions and subtended angles. Hence the detection accuracy of these methods is insufficient to visual localization, motion estimation and 3D reconstruction [3].

Edge-based methods have emerged in order to improve the accuracy of corner detection [9-14, 26, 27]. These methods detected corners by analyzing the shape characteristics of the edges.

[19] proposed a SLAM system, ORB-SLAM2, which supports monocular, stereo and RGB-D cameras and includes map reuse, loop closing and relocalization capabilities. Afterwards, PL-SLAM (Point and Line SLAM) which utilizes both corner points and feature lines and PEL-SLAM (Point and EDlines SLAM) which utilizes EDlines (Edge Drawing lines) as an information about feature lines have been proposed to improve the performance of SLAM system [20-22].

In this paper a method to determine simultaneously the corner points and feature lines based on gradient directions at the edge points calculated in time of detecting the points is proposed. The previous methods to detect corners on the edge lines mainly depend on the calculation of curvatures.

This paper proposed a method to improve the performance ORB corner detection based on edge points. In the proposed method, firstly, the edge points are represented with multivalued functions which are then used as information about feature lines. Then, corner points are determined by evaluating the change of the curve of the directions at the edge points which are expressed with multivalued functions according to the movement of the edge point, not by calculating curvatures, thus, lowering its computational complexity as well as improving performance. Edges are extracted using Canny algorithm.

The remainder of the paper is organized as follows. In section 2, we discuss related work on corner detection, and in section 3, the detected edges are expressed with multivalued functions, and gradient direction functions are defined at the edge points. In section 4, corner points and feature lines are defined using multivalued functions and direction functions. In section 5, the performance of the proposed method is evaluated experimentally by comparing with previous edge-based corner detection methods in terms of detection performance and computing time. Finally, section 6 gives conclusions.

## 2. RELATED WORK

### 2.1. CORNER DETECTION BASED ON EDGES

Edge-based methods have emerged in order to improve the accuracy of corner detection [9-14, 26, 27]. These methods detected corners by analyzing the shape characteristics of the edges.

At first, the corners were detected by fixed size curvature calculation or by using polynomials fitted to the curves generated by edges. In [26], local operators in the neighborhoods of the gray-scaled images were utilized to calculate the values related with the curvatures of the edges passing through the neighborhoods. When the gray level in the neighborhood is $g(x,y)$, the following equation is given:

$$tan\theta = \frac{g_y}{g_x}$$

, where $\theta$ is a gradient direction at a point$(x,y)$.

And the partial derivatives of $\theta$ are as follows:

$$\theta_x = \frac{g_{xy}g_x - g_{xx}g_y}{g_x^2 + g_y^2},$$

$$\theta_y = \frac{g_{yy}g_x - g_{xy}g_y}{g_x^2 + g_y^2}$$

Multiplying the projection of the change of the gradient direction vector $(\theta_x, \theta_y)$ along the edge by local gradient magnitude gives

$$k = \frac{g_x\theta_y - g_y\theta_x}{g_x^2 + g_y^2} = \frac{g_{xx}g_y^2 + g_{yy}g_x^2 - 2g_{xy}g_xg_y}{g_x^2 + g_y^2}$$

, and this value could be taken as the curvature of the edge.

Based on this, the corners were detected through 6 steps.

In [27], they proved that planar curve recognition have two important characteristics that those curves have large values of discontinuity and curvature, and proposed a detection method based on that.

In [28], CCS was considered to be suitable to find invariant geometric features of the planar curves at multiple scales, and the curvature was defined as follows:

$$k(u,\sigma) = \frac{\dot{X}(u,\sigma)\ddot{Y}(u,\sigma) - \ddot{X}(u,\sigma)\dot{Y}(u,\sigma)}{(\dot{X}(u,\sigma)^2 - \dot{Y}(u,\sigma)^2)^{1.5}}$$

Here $\dot{X}(u,\sigma) = x(u) \otimes \dot{g}(u,\sigma)$, $\ddot{X}(u,\sigma) = x(u) \otimes \ddot{g}(u,\sigma)$, $\dot{Y}(u,\sigma) = y(u) \otimes \dot{g}(u,\sigma)$, $\ddot{Y}(u,\sigma) = y(u) \otimes \ddot{g}(u,\sigma)$, $\otimes$ and $g(u,\sigma)$ denote the convolution product and Gaussian function with derivation $\sigma$ respectively, and $\dot{g}(u,\sigma)$ and $\ddot{g}(u,\sigma)$ denote the first and second derivatives of $g(u,\sigma)$ respectively. And they introduced an adaptive threshold as follows:

$$T(u) = C \times \bar{k} = 1.5 \times \frac{1}{L_1 + L_2 + 1}\sum_{i=u-L_2}^{i=u-L_1} k(i)$$

, where $\bar{k}$ is the average curvature in a supported region, $L_1$ and $L_2$ is the size of the supported region, and the coefficient $C$ has a value of 1 to 2 (set to 1.5 in the paper). Using the curvature and the adaptive threshold above, a method to detect corners with different sizes through six steps was proposed:

The corner detection method proposed in [9] is based on the assumption that two image patches being compared along a linear edge segment would be very similar to each other, however it is not true in case the edge segment is curved in the corner, hence the corners with low similarity between image patches along the edge could be detected. Considering a linear movement in a direction $AA'$ from a point on the contour line of the same gray level, the gray level is approximated to the following quadratic function of movement $x$ by Taylor expansion:

$$f(x) = f(0) + f'(0)x + \frac{f''(0)x^2}{2}$$

, where $f(0)$ indicates the intensity at the point, and $f'(0)$ and $f''(0)$ are its derivatives with respect to $x$.

When the linear direction $AA'$ is parallel to the edge, $f'(0)$ is equal to zero, and

$$f(\mathrm{x}) = f(0) + \frac{f^{"}(0)x^2}{2} + e(x)$$

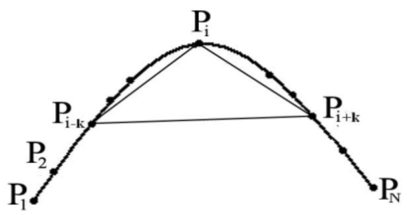
, where $e(x)$ indicates an error related with the image noise. From this, the following subtraction function for short distance $w$ is defined:

$$D(\mathrm{x}, w) = |f(x) - f(x - w)|$$
$$= \left|kxw - \frac{kw^2}{2} + e(x) - e(x + w)\right|$$

, where $k = f^{"}(0)$. Based on the fact that $k$ reflects the information about the curvature, corners were determined by processing with a threshold sequence.

Later, multi-scale local curvature evaluation methods were exploited based on variable dimensions and curvature evaluation functions [10-14]. CPDA (Chord-to-Point Distance Accumulation) technique was used to improve the robustness to noise and local variation [14].

In [29], they mentioned that CPDA technique for edge-based corner detection has good repeatability and low localization error, however it often fails to detect accurately the true corner points in some situations and, furthermore, it has high computational complexity. In order to overcome these shortcomings, they proposed a corner detection method using a simple triangular theory as follows:




In the figure above, the curvature at a point $P_i$ is calculated as follows:

$$R_L(i) = \frac{d_1}{d_2 + d_3}$$

, where

$$d_1 = \sqrt{(x_{P_{i-k}} - x_{P_{i+k}})^2 + (y_{P_{i-k}} - y_{P_{i+k}})^2}$$
$$d_2 = \sqrt{(x_{P_i} - x_{P_{i-k}})^2 + (y_{P_i} - y_{P_{i-k}})^2}$$
$$d_3 = \sqrt{(x_{P_i} - x_{P_{i+k}})^2 + (y_{P_i} - y_{P_{i+k}})^2}$$

In this way, the edge-based corner detection methods evaluate the curvatures at the edge curve points in different conditions, and based on that determine the corners. In general, such methods have higher accuracies than derivative-based methods and template-based methods, and are also capable of detecting corners at different scales, so they could be applied to visual localization, motion estimation and 3D reconstruction [3].

The performance of the edge-based methods depends largely on the qualities of the edges, and the edges are sensitive to the noise, Affine transform, viewpoint and brightness. Therefore, recent research works aim at extracting edges more correctly and safely [15-18, 30-32]. Especially, the estimation of the curvatures at all points on the edges brings difficulties to the real-time SLAM tasks.
In [30], they insisted that the current multi-scale analysis approaches based on corner detector is somewhat difficult to implement, because they do not make full use of multi-scale and multi-directional information, thus degrading their detection accuracy and capability of refining corners. Hence, they proposed an improved shearlet transform with a flexible number of directions and a reasonable support to extract accurate multi-scale and multi-directional structural information from images. Based on this shearlet transform, they constructed a multi-directional structure tensor and proposed a multi-scale corner measurement function to make full use of multi-scale and multi-directional information. In [31] they proposed a new method to measure corner sharpness termed as the point-to-centroid distance (PCD), defined corner response function (CRF) and based on this function evaluated its beneficial characteristics in distinguishing corners from non-corners. In [32] they proposed an interest points (corner or blob) detection method, in which a new blob filter i.e., second-order anisotropic Gaussian directional derivative filter with multiple scales was intorduced to smooth the input image and verified its advantages over previous methods. However, those methods presented in [30-32] have high compuational cost, and might not be applicable to practical SLAM tasks.

## 2.2. FAST CORNER DETECTION

FAST (Features from Accelerated Segment Test) corner detection considers 16 pixels in a circle around a point $p$ which is expected to be a candidate corner (Fig.1). The initial detector [1, 33] takes a point $p$ as a corner point when the circle has $n$ continously neighboring pixels which have intensities larger than $(I_p + t)$ or smaller than $I_p - t$ (here, $I_p$ is the intensity at $p$ and $t$ is a threshold). Several methods to make FAST detection faster were proposed, for example by setting $n$ to 12 and checking for 1, 5, 9, and 13 only, thus excluding many non-corner points. This detector showed high performance, however revealed some shortcomings as follows.

1) The high-speed test does not generalize well for $n <$ 12.
2) The choice and ordering of the fast test pixels contains implicit assumptions about the distribution of feature appearance.
3) Knowledge from the first 4 tests is discarded.
4) Multiple features are detected adjacent to one another.

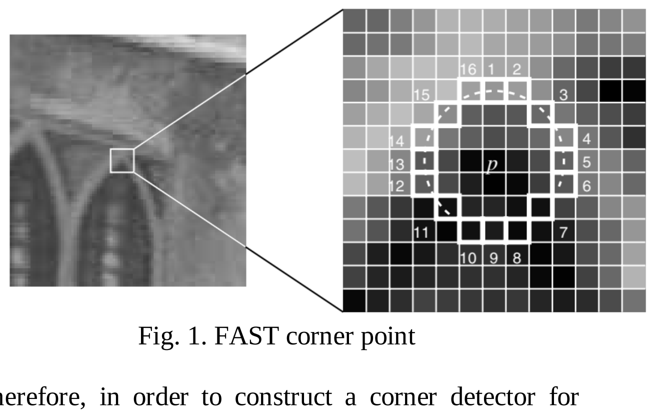



Fig. 1. FAST corner point

Therefore, in order to construct a corner detector for given $n$, corners were detected first in the set of images of the region of interest by using sub-check criteria based on $n$ and a proper threshold. This algorithm is slow, since it checks all of the 16 positions in the circle around each pixel point.

For each position $x \in \{1 \ldots 16\}$ in the circle, the pixel at a relative position could be in one of the following three cases:

$$S_{p\to x} = \begin{cases} d, & I_{p\to x} \le I_p - t \quad \text{dark} \\ s, & I_p - t < I_{p\to x} < I_p + t \quad \text{similar} \\ b, & I_p + t \le I_{p\to x} \quad \text{brighter} \end{cases} \tag{1}$$

For each $p \in P$(a set of pixels), $S_{p\to x}$ is calculated for selected $x$ ,and $P$ is decomposed into three subsets $P_d$, $P_s$, $P_b$ (denoted as $P_{S_{p\to x}}$).

In the next step of constructing corner detector, a binary variable $K_p$, which indicates $p$ is a corner or not, is introduced, and entropy of $K_p$ is calculated based on $P$ decomposition and the selection of $x$ which gives the most amount of information whether the pixel could be a corner or not. Based on $K_p$ , corner points are determined.

And, since the sub-checking does not calculate corner response function, non maximal suppression could not be directly applied to features obtained.Therfore non maximal suppression should be able to calculate score function $V$ at each corner detected and discard corners whose neighboring corners have larger values of $V$.

Here, the maximum values of $n$ and $t$ for the function $V$ when $p$ is a corner, and the sum of the absolute difference between the pixels in the contiguous arc and the center pixel are at issue [33].

The proposed method could be regarded as a solution to this problem.

## 3. EDGE DETECTION AND ITS REPRESENTATION BY A MULTIVALUED FUNCTION AND A DIRECTION FUNCTION

The 723rd left image in EuRoC stereo camera image dataset and detected edges in its partial image are shown as follows (Fig. 2).

Fig. 2. The 723rd left image in EuRoC stereo camera image dataset and detected edges in its partial image

In the following sections of this paper, the proposed corner detection method is described with this partial image and its edges. The proposed method could also be applied to other images and their partial images without modification.

The edges in the image above are composed of the following edge lines.

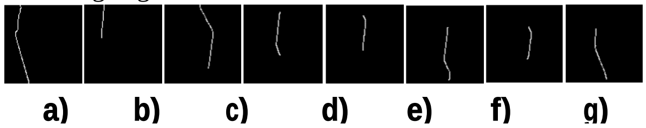



Fig. 3. Representative edge lines which comprise the edges in the image of Fig. 2

Each edge line above can be represented by a multivalued function. For example, the edge line e) is illustrated in detail as shown in Fig. 4.

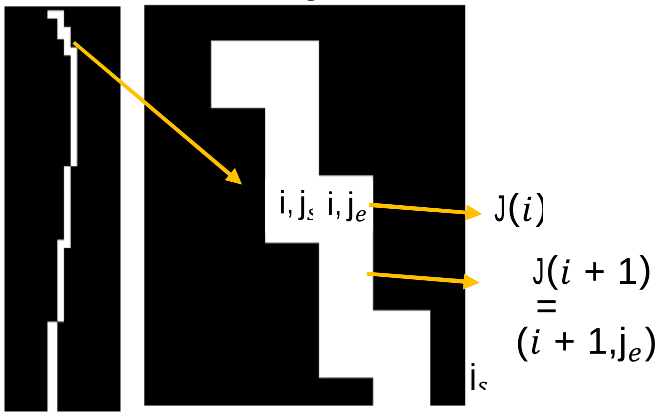



Fig. 4. Pixel points in an edge segment

In this paper, while edges are detected by Canny algorithm, pixel points $(i, j)$ on the detected edge and gradient directions are represented by multivalued function.

Each pixel points $(i, j)$ on the edge line $i$ in Fig. 4 has its corresponding pair of starting point $j_s$ and ending point $j_e$, $(j_s, j_e)$. When the edge line starts at $i_1$ and terminates at $i_m$, the multivalued function by which the pair of starting point $j_s$ and ending point $j_e$, $J(i) = (j_s, j_e)$ corresponds to each $i$ is as follows:

$$f_{edg}(n_f) = \{J(i_1), J(i_2), \ldots, J(i_m)\}$$

, where $n_f$ denotes the index number of the edge line. In this way, all the edge lines could be represented by multivalued functions.

Thus, when the number of edge lines is $N_f$( 10 in Fig. 2.), all the edges could be represented by the following set of multivalued functions

$$\{f_{edg}(n_f), n_f = 1, \ldots, N_f\} \tag{2}$$

If the following equation is satisfied for $J(i) = (j_s', j_e')$ and $J(i+1) = (j_s'', j_e'')$ in multivalued functions,

$$j_s' \le (j_e'' + 1) \text{ or } j_e' \ge (j_s'' + 1) \tag{3}$$

then $J(i)$ and $J(i+1)$ are defined to be linked (Fig. 4).

In general, a distance between $J(i_1) = (j_s', j_e')$ and $J(i_2) = (j_s'', j_e'')$, $(i_1 < i_2)$ is defined as follows:

$$d_{ij} = \max(d_i, d_j), \tag{4}$$

Here,

$$d_i = \max(i_2 - i_1 - 1, 0)$$

$$d_j = \max\left(\left|\frac{j_s' + j_e'}{2} - \frac{j_s'' + j_e''}{2}\right| - \frac{j_e' - j_s' + 1 + j_e'' - j_s'' + 1}{2}, 0\right)$$

Then the distance defined by Eq. 4 turns out to be zero for $J(i)$s which are linked to each other.

In this paper, a set of points $J(i)$ which are linked with the minimum distance not larger than 1 on the edge lines for each $i$

$$\{J(i_1), J(i_2), \dots, J(i_m)\} \quad (5)$$

is represented by a multivalued function $f_{edg}(n_f)$.

Let the intensity of the pixel point $(i, j)$ be $I(i, j)$ , then

$$\tan(\alpha(i,j)) = \frac{I(i,j+1)-I(i,j-1)}{I(i+1,j)-I(i-1,j)} \quad (6)$$

, where $\alpha(i, j)$ denotes the gradient direction of the intensity change at the point $(i, j)$ on the edge.

We denote the direction function which describes the corresponding relationship between edge points defined by multivalued functions and gradient directions $\alpha(i, j)$ as $f_\alpha(n_f)$.

An $i$ of an image could have several $J(i)$ , and its number $N_i$ can change with $i$. For $i$ of $N_i \neq N_{i+1}$, a multivalued function and a direction function are newly defined and would increase when $N_i < N_{i+1}$ or terminate when $N_i > N_{i+1}$, thus decreasing. In this case, denotion $J(i, n)$ is used to discriminate several $J(i)$s for $i$.

As can be seen in the figure, a multivalued function $f_{edg}(n_f)$ represents the set of edge points, i.e., information about the curve in 3D space, and this curve can be determined by processing left and right video frames which are input continuously from a stereo camera. This curve information can also be used to enhance the performance of the localization and mapping.

## 4. DETECTION OF CORNERS AND FEATURE LINES BASED ON MULTIVALUED FUNCTION AND DIRECTION FUNCTION

In this paper, a corner is defined by using $\alpha_{kp}$, which is the angular change in the edge segment of $kpn$ pixels where $kpn$ corresponds to $n$, the number of continuously neighboring pixels whose intensities differ from the FAST corner in Eq.1 by a value larger than the threshold. Values of $kpn$ and $\alpha_{kp}$ are set to $kpn = 9$, $\alpha_{kp} = 0.3491(\sim 20°)$.

The values of gradient directions at the edge points along the edge line c) in Fig.3 change rapidly near corner points (Fig.5).

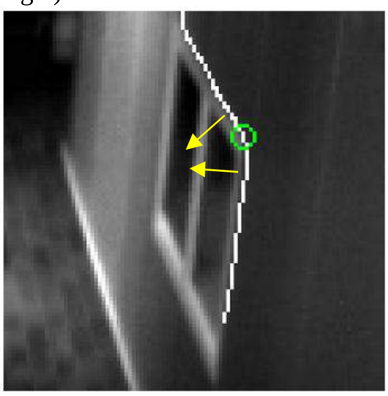


Fig. 5. Change in the direction near a corner point

Based on this property, a novel corner detection method is proposed as follows, and the edge lines represented by multivalued functions near corner points could be used as feature line informations.

### 4.1. DEFINITION OF CORNER POINT AND FEATURE LINE

First, prior to the definition of a corner point, let us consider the changes in the gradient direction at the points of multivalued function, with the edge line c) in Fig. 3 as an example.

The calculated directions at each points of the multivalued function are shown in Fig. 6 (a).

Performing convolution product to this by using the following Gaussian function

$$\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-x_0)^2}{2\sigma^2}}, \sigma = 2.1 \quad (7)$$

gives the curve in the Fig.6 (b).

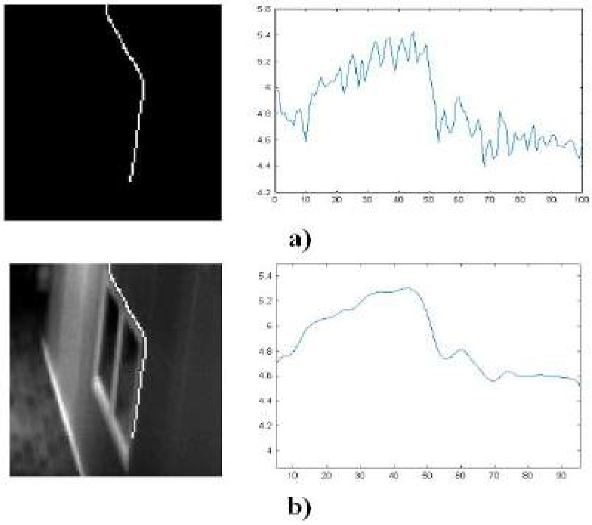



Fig. 6. A multivalued function and a gradient direction curve

Based on the above gradient direction function corresponding to the multivalued function, a corner point is defined as follows.

1). Among edge points, absolute values of the gradient of angular change rate of the convolution product direction curve

$$\left|\widehat{f}_\alpha(n_f, k+1) - \widehat{f}_\alpha(n_f, k-1)\right|, k = 1, \dots, K_\alpha \quad (8)$$

($K_\alpha$: number of edge points which belong to $f_{edg}(n_f)$) are calculated, and in the curve of this values local peaks $p_k$ and valley points $v_k$ are taken(Fig. 7). For $p_k$ and $v_k$, let $\alpha_{kp}(\approx 0.3491{\sim}20°)$ be a bending angle of edge segment corresponding to the corner point and $kpn$(=9) be the number of edge points enclosing the corner point. If the angular change near the peak satisfies the following condition

$$\left|\widehat{f}_\alpha(n_f, v_{mk}) - \widehat{f}_\alpha(n_f, v_{mk+1})\right| > \alpha_{kp} \quad (9)$$

(here, $v_{mk} = \max(p_k - hkpn, v_k)$, $v_{mk+1} = \min(p_k + hkpn, v_{k+1})$, $hkpn = floor\left(\frac{kpn}{2}\right)$),

then the edge point corresponding to the peak $p_k$ is defined to be a corner point (Fig. 7).

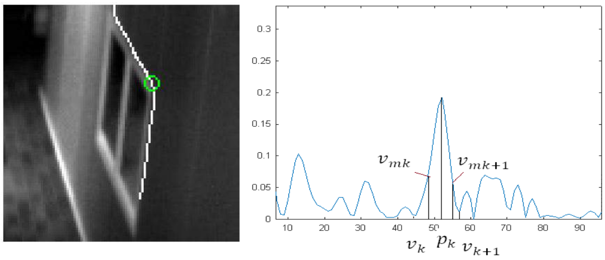



Fig. 7. A curve of the gradient direction change and the corner points

While the FAST algorithm of ORB(Oriented FAST and Rotated BRIEF) decides a corner point by thresholding the intensities of 16 pixel points bounding the candate corner point, this paper defines it based on intensity gradient direction at $kpn$(=9) edge points around it. That is, a corner point is defined when the change in the intensity gradient direction over $kpn$ edge points around it is larger than $\alpha_{kp}$.

In order to further investigate the change in the intensity gradient direction over $kpn$ (=9) edge points around a corner when it is close to other corners, gradient direction curve for the edge line a) in Fig. 3 is considered (Fig.8).

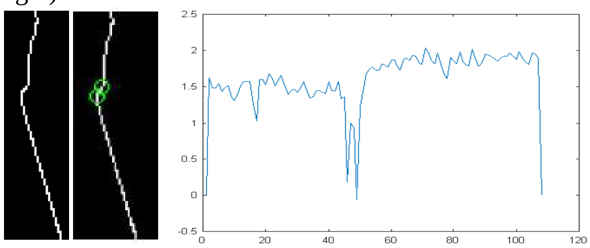


Fig. 8. The edge line a) and its gradient direction change curve

For this direction curve, the relationship of Eq.8 with the convolution product $\widehat{f_\alpha}(n_f, k)$ and its absolute value of angular change rate gradient is given in Fig. 9.

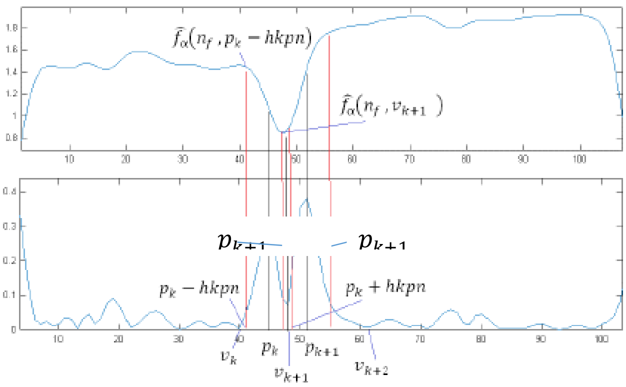



Fig. 9. Adjacent peaks in the curve of absolute value of direction change gradient

As shown in the Fig. 9, for peak $p_k$, $p_k - hkpn$ is larger than $v_k$, hence $v_{mk} = \max(p_k - hkpn, v_k)$ is the same as $p_k - hkpn$, however, since $p_k + hkpn$ is larger than $v_{k+1}$, $v_{mk+1} = \min(p_k + hkpn, v_{k+1})$ is the same as $v_{k+1}$. On the contrary, for peak $p_{k+1}$, it is $v_{mk+1} = v_{k+1}$ and $v_{mk+2} = p_{k+1} + hkpn$.

Corners near the starting point and ending point for $f_{edg}(n_f)$ in the multivalued function set $\{f_{edg}(n_f), n_f = 1, \dots, N_f\}$ described in Eq.2 will be explained separately, and so, two endpoints of the edge in the change curve are excluded here.

2). When for a $J(i, n)$ more than two $J(i+1, n')$ are configured with a distance smaller than 2 as described in Eq. 4, thus increasing number of multivalued functions, if Eq.9 is satisfied near the connection point for multivalued functions connected to the increased $J(i+1, n')$ and the correspnding direction functions, then it is defined as a corner point. Here, $J(i, n)$ is the aforementioned denotion in case that multiple $J(i)$s exist for an $i$. This is illustrated in Fig. 10-12 taking the edgelines d) and e) defined in Fig. 3 as examples. As can be seen in Fig. 10 and 11, corners defined in 1) are defined based on each multivalued function and gradient direction curve. As shown in Fig. 2, next to a $J(i, n)$ of the multivalued function d), J(i+1,$n_1$) and J(i+1,$n_2$) which belong to the multivalued functions d) and e) are configured with 1 of distance described in Eq.4. Direction functions of the edge lines d) and e) in the neighboring points (15 pixel points) near the connection point are plotted in Fig. 12 (a) and from Fig. 10(a) and Fig. 11(a). The direction curve in Fig. 10 is centered at the connection point, with 15 pixel points plotted in reverse order, i.e., the pixel point of the multivalued function d) located 15 pixel points far from the connection point is taken as the starting point, direction curve corresponding to the length to the 15 pixel points of multivalued function e) through the connection point is obtained, and based on this, if the change rate gradient of the direction curve near the connection point satisfies the condition Eq.9, the corner point is defined. Fig. 12(b) and Fig. 12(c) show the convolution product and change rate gradient for Fig. 12(a).

When more than two branches of the multivalued function pixel points spread from the connection point and hence the number of the corresponding direction curves with the connection point as a center is larger than 2, if Eq. 9 is satisfied for the largest change in the direction as in Fig. 12, then a corner point is defined.

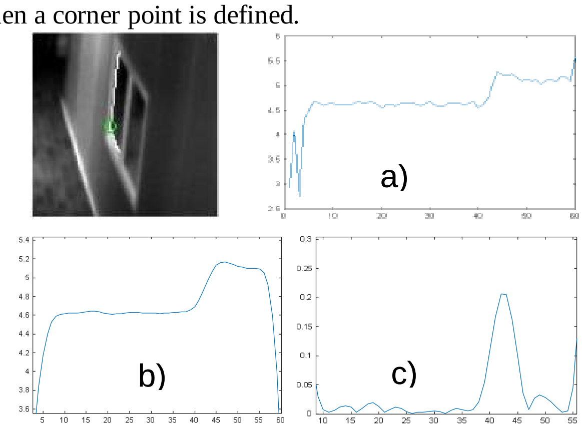



Fig. 10. Gradient direction curve for multivalued function d) and the corner point

Int. J. Advanced Networking and Applications
Volume: 17 Issue: 01 Pages: 6761-671 (2025) ISSN: 0975-0290

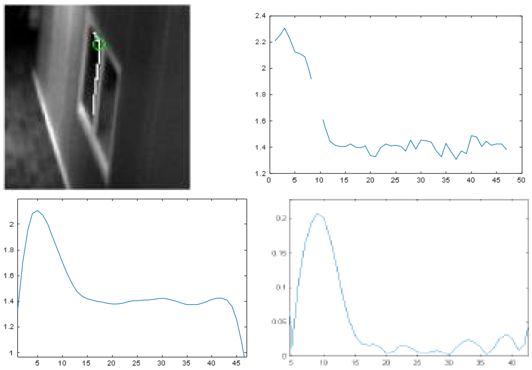



Fig. 11. Gradient direction curves for multivalued function e) and the corner point

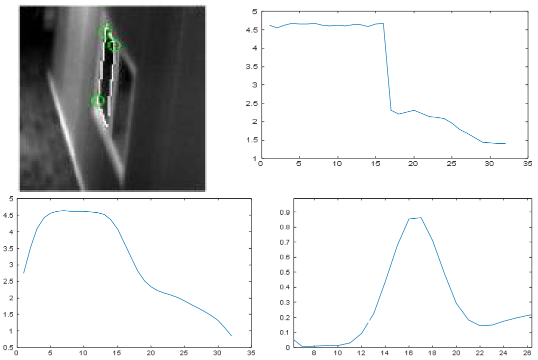


Fig. 12. Gradient direction curves for multivalued functions d) and e), and the corner points

3). When for a $J(i+1,n)$ more than two $J(i,n')$ are configured with a distance smaller than 2 and the corresponding multivalued functions terminate, hence the number of the multivalued functions decreases, if more than one direction function satisfy Eq.9 for multivalued functions connected with the connection point of $J(i,n')$ as a center to the increased $J(i+1,n')$ and the correspnding direction functions, then it is defined as a corner point.

Fig. 13 shows the definition of the corner point at the positions where the multivalued functions d) and e) defined above terminate. While the corners are being defined, the indexes of the multivalued functions associated with the corners are also saved to be used for matching of the corners.

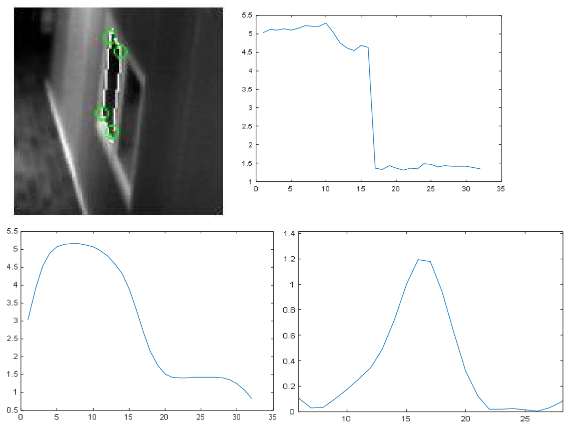


Fig. 13. Definition of the corner point at the positions where the multivalued functions d) and e) terminate

## 4.2. DETECTION OF THE CORNERS AND FEATURE LINES

Concept of the node point on the edge is explained prior to the formulation of corner detection algorithm based on the result above.

In 1) and 2) of the section 4.1 the determination of the corners in case that multivalued functions newly start or terminate were explained. When the multivalued functions start newly or terminate like this, the connection point of  is defined as a node point (node points 1 and 2 in Fig.14). While node points are defined, indexes of the multivalued functions associated with the node point are also saved to be utilized for corner detection.

It should be noted that for $J(i,n)(=(j_s,j_e))$, $J(i-1,n_1)=(j_s',j_e')$ and $J(i+1,n_2)=(j_s'',j_e'')$ of the same multivalued function, if $j_s-j_s'$ or $j_e-j_e'$ is larger than 5, then the pixel point$(i,j_s')$ or $(i,j_e')$ is defined as a node point (node point 3 in Fig. 14) and the corner determination is done using the methods 2) and 3). The same goes for $J(i+1,n_2)=(j_s'',j_e'')$.

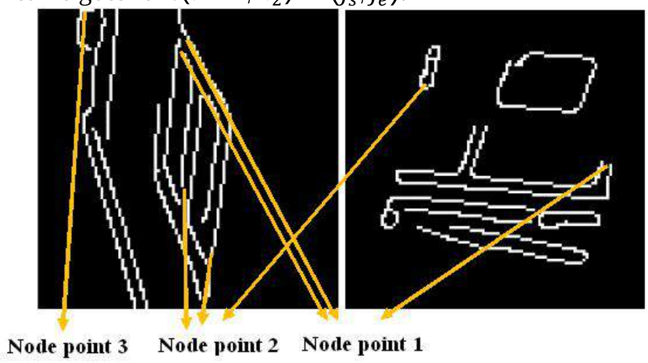



Fig. 14. Types of node points in the edges

From the above consideration, the corner point and feature line detection algorithm for images acquired from cameras in SLAM system is formulated as follows:

① Find edge points in the image and the gradient directions at the edge points, and at the same time evaluate the corresponding multivalued functions, direction functions and node points.

② Get direction change rate gradient curve based on convolution product from direction functions, and then determine corners by definition 1) above.

③ Determine corners at the node points by definition 2) and 3).
Among the adjacent corners which are located from each other within a short distance (smaller than $kpn=9$) as shown in Fig. 8 and Fig. 9, only a corner point which has the largest bending angle is selected, and the remaining corners are discarded.

④ The part of multivalued function connected with the corner point is taken as a feature line information.

The corners detected in the rectified left and right images of the 723rd image in EuRoC stereo camera dataset, using the proposed algorithm are shown in Fig. 15.

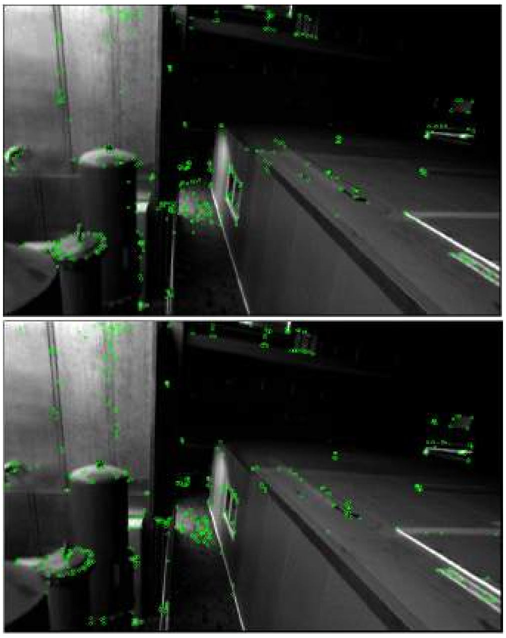


Fig. 15. Corners detected in the rectified left and right images of the EuRoC stereo camera dataset

## 5. EXPERIMENTAL RESULTS AND COMPARISON

The experiment was conducted to investigate the relationship between corners detected by the proposed method and ORB algorithm. The corners detected by the proposed method correspond to those detected by ORB, and under the condition that edges were detected, corners were detected independent of the threshold unlike in ORB. And the proposed method was compared with the existing edge-based corner detection methods, CCS and PCD, and the methods based on triangular theory in terms of calculation time.

### 5.1. RELATION WITH ORB CORNER POINTS

The proposed corner is a generalization of ORB corner used in ORB-SLAM2. Therefore, the proposed corner detection is evaluated in comparison with ORB corners.

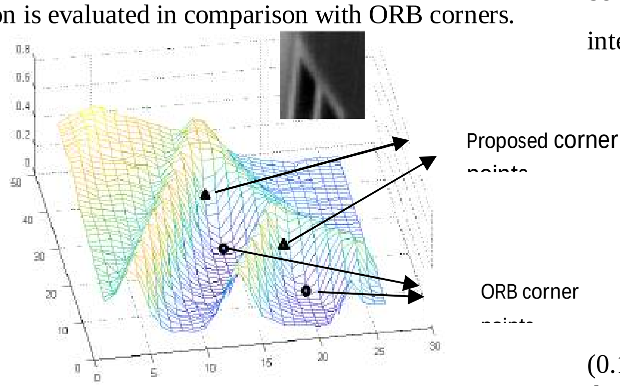



Fig. 16. ORB corner points and the proposed corner points in intensity surface

ORB corners and the proposed corners in the intensity surface (a surface which depicts the intensity values of the pixel point s according to their locations) for the partial image of Fig.1 are shown in Fig. 16.

In Fig. 1, if there exists a set of $n$ neighboring pixels which are brighter than the brightness of $p$, $I_p$, by threshold $t$ or darker than $I_p - t$, it is classified as a corner point corner, and corner points are defined based on this principle.

The corner points defined in this paper and ORB corner points are shown in intensity surface (Fig. 16). In this surface the proposed corner points are located at the position of an edge point where its intensity change gradient has a locally maximal value and the local maximum value of direction change is larger than threshold. Therefore it becomes one of the neighboring pixel points of ORB corner points, and the descriptors defined in a certain image region near the corner point and the matching techniques could still be used without change.

The intensity surface and edge points for the corner in Fig.5 are shown in Fig. 17.

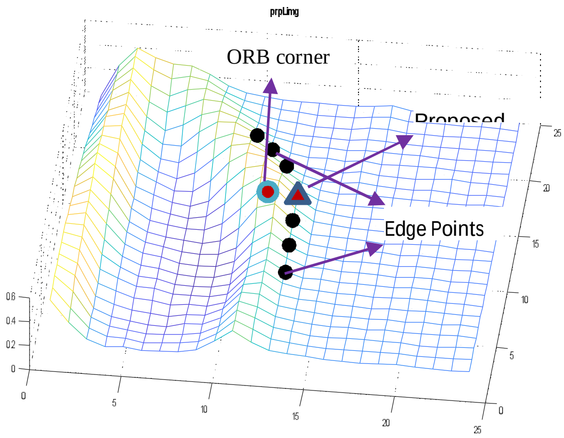



Fig. 17. The intensity surface and edge points for the corner in Fig.5

The pixel intensities and edge points near the corner in Fig.17 are shown in Table 1.

Let kpn continuous edge points from $v_{mk}$ to $v_{mk+1}(= v_{mk} + kpn - 1)$ near the corner point in Fig. 6 be denoted as $\{P_{v_{mk}}, P_{v_{mk}+1}, P_{v_{mk}+2}, \dots, P_{v_{mk+1}}\}$, and the intensities of these pixel points

$$\{I(P_{v_{mk}+k}), k = 0, \dots, kpn - 1\}$$

have a maximum value

$$I_{em} = max\{I(P_{v_{mk}+k}), k = 0, \dots, kpn - 1\}$$

, which is 0.286 in Table.1. When $P_0$ is a pixel point with the intensity value larger (or smaller) than $I_{em}$ in the direction of the corner $p_k$ in Fig. 5 b) and a threshold

$$t = I(P_0) - I_{em}$$

(0.149 in Table 1) is set for for $I(P_0)$ (0.435 in Table 1), there exist pixel points near $P_0$ which are defined by FAST algorithm, because the pixel points in the direction reverse to $P_0$ with respect to the edge points have intensities smaller than those of the edge pointers. FAST considers pixels only in the circular neighborhood, however, in general, the pixels in the ellipse neighborhood can be considered here, and from this point of view, $P_0$ is a ORB corner.

The edge points and corners in the dark part of the image in Fig. 15 are shown in Fig. 18.

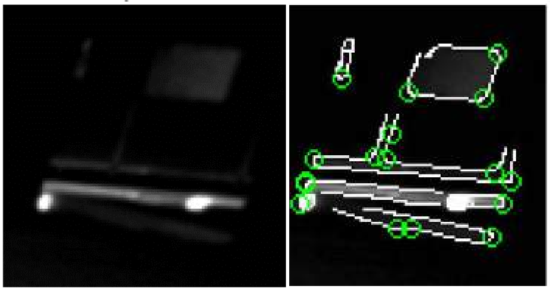


Fig. 18. The edge points and corners in the dark part of the image

Fig. 19 shows the pixel intensities and corners near the corner point detected in the topmost part of Fig. 18.

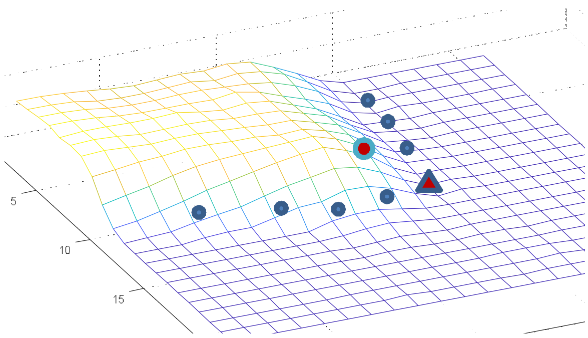



Fig. 19. Edge and corner points and FAST corners in the dark image

Table 1. The pixel intensities and edge points near the corner in Fig 5

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0.4 | 0.32 | 0.249 | 0.176 | 0.152 | 0.156 | 0.145 | 0.156 | 0.137 |
| 0.403 | 0.372 | 0.286 | 0.2 | 0.164 | 0.152 | 0.141 | 0.160 | 0.141 |
| 0.396 | 0.423 | 0.352 | 0.250 | 0.188 | 0.152 | 0.145 | 0.164 | 0.149 |
| 0.349 | 0.439 | 0.411 | 0.317 | 0.223 | 0.168 | 0.149 | 0.160 | 0.152 |
| 0.266 | 0.403 | 0.439 | 0.372 | 0.254 | 0.180 | 0.156 | [illegible] | 0.149 |
| 0.184 | 0.337 | 0.435 | 0.411 | 0.278 | 0.188 | 0.164 | 0.145 | 0.145 |
| 0.113 | 0.250 | 0.403 | 0.423 | 0.286 | 0.184 | 0.168 | 0.156 | 0.149 |
| 0.058 | 0.180 | 0.360 | 0.407 | 0.278 | 0.172 | 0.168 | 0.160 | 0.152 |
| 0.043 | 0.105 | 0.282 | 0.388 | 0.258 | 0.184 | 0.168 | 0.152 | 0.160 |

Edge Points

Proposed Corner Points

ORB(FAST) Corner Points

The pixel intensity values near the corner are shown in Table 2. In Table 2, the maximum intensity of the edge points near the corner is $I_{em} = 0.1058$, and the threshold for the pixel point $P_0$ which is in the direction of the direction curve at the corner and has the intensity $I(P_0) = 0.1529$ larger than $I_{em}$ is $t = 0.0471 (= I(P_0) - I_{em})$. And a FAST corner is defined for this threshold. Thus, the corners detected by the proposed method correspond to ORB(FAST) corners and the threshold to define ORB(FAST) corner is needless to consider. And for $v_{mk}$ and $v_{mk+1}$ in the section 4.1, $kpn' = v_{mk+1} - v_{mk} + 1$ corresponds to the number of the continuous pixel points which have intensities different from the corner by the value larger than the threshold. $kpn$ has the maximum value of 9. Like this, the corners are defined so as to correspond to ORB(FAST) corners for different thresholds based on edge and could be used for SLAM calculations.

Table 2. The pixel intensity values and edge points near the corner in Fig. 19

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0.0078 | 0.0078 | 0.0078 | 0.0078 | 0.0078 | 0.0078 | 0.0039 | 0.0039 | 0.0039 |
| 0.0156 | 0.0156 | 0.0117 | 0.0078 | 0.0039 | 0 | 0.0039 | 0.0039 | 0.0039 |
| 0.0588 | 0.0549 | 0.0509 | 0.0431 | 0.0352 | 0.0235 | 0.0078 | 0.0039 | 0 |
| 0.1019 | 0.1019 | 0.0980 | 0.0941 | 0.0745 | 0.0549 | 0.0196 | 0.0117 | 0.0039 |
| 0.1254 | 0.1294 | 0.1333 | 0.1294 | 0.1058 | 0.0745 | 0.0313 | 0.0196 | 0.0078 |
| 0.1450 | 0.1490 | 0.1568 | 0.1529 | 0.1176 | 0.0784 | 0.0313 | 0.0156 | 0.0039 |
| 0.1450 | 0.1450 | 0.1529 | 0.1450 | 0.1019 | 0.0549 | 0.0235 | 0.0078 | 0 |

### 5.2. THRESHOLD DEPENDENCY OF THE ORB(FAST) CORNERS

The selection of ORB corner point depends largely on the threshold value $t$.

Corner points and feature lines detected in the image of Fig. 2 using the proposed method are shown in Fig. 20. Edge segments connected through corner points could be exploited as feature lines to improve the accuracy of the calculations in SLAM.

Fig. 21 shows ORB corner points detected in the same image region when the threshold $t$ is increased from 0.04 to 0.12. As shown in Fig. 21, when the threshold is small, a lot of candidates for corner points are obtained and many of them which could not be corner points turn out be noise in SLAM calculation. On the contrary, when the threshold is large, some corner points were not detected.

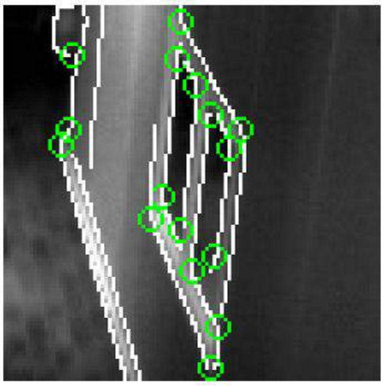


Fig. 20. Edge points and corner points in the partial image of Fig. 2

However, the proposed method detected all corner points rather correctly when edge points of the image were given. In Fig. 21.

The proposed method detected all the corner points, i.e., not only the corner points detected by ORB technique but also the corner points not detected by ORB technique.

Therefore, more corner points and feature lines connected with the corner points are detected by the proposed method and could be utilized for SLAM system.

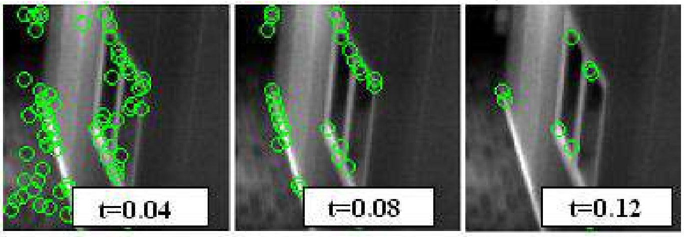



Fig. 21. ORB (FAST) corner points at several values of threshold $t$

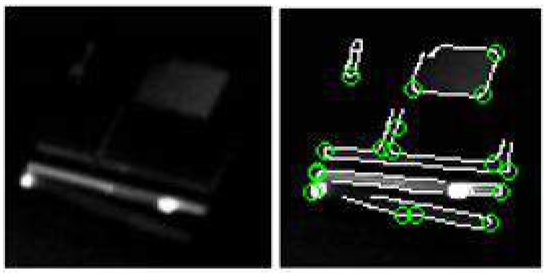


Fig. 22. Edge points and corner points in the dark region

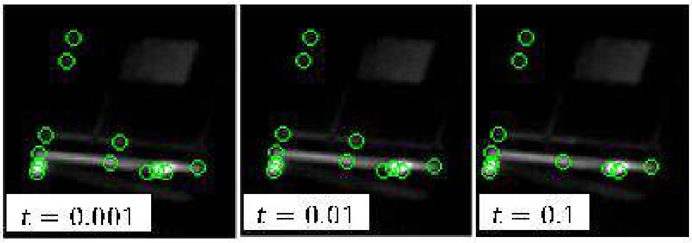



Fig. 23. ORB (FAST) corner points in the dark region

### 5.3. CALCULATION TIME COMPARISON WITH EDGE-BASED CORNER DETECTING METHODS

The calculation time of the proposed method was compared with the existing edge-based corner detection methods, namely, CCS, PCD and triangular method for given edges in the partial image of Fig.2 (See Table 3).

Table 3. Calculation time of corner detection methods

| Corner detector | Calculation time (ms) |
|---|---|
| The proposed method | 1.7146 |
| PCD | 2.2325 |
| Trianglular Method | 1.9637 |
| CCS | 2.22143 |

As shown in Table 1, the calculation time of the proposed method is shorter than those of the other edge-based methods.

The calculations were done using MATLAB_R2018a on Windows 10 platform with Intel® Core(TM) i3-3110M 2.4GHz CPU and 4GB RAM.

## 6. CONCLUSION

A method to detect corner points and feature lines for SLAM system by using edge segments was proposed. The proposed method has an advantage that it, unlike ORB(FAST) feature point detector, does not depend on threshold. This enables to detect feature lines connected with corners, thus enhancing the accuracy of the SLAM calcuation, and respond to multiple scales.

In the definition of corners, the values of the angular change in the edge segment, $\alpha_{kp}$, and the number of the edge points, $kpn$ pixel, could be set accoring to the corner matching and image noise property.

It should be noted that corner detection performance of the proposed method depends on the performance of the edge point detection.

Therefore, a proper tuning of the edge detection parameters according to the illumination state, noise property and other conditions of the input images is needed. We hope the proposed method combined with a proper scheme for tuning of the edge detection parameters according to the characteristics of the input images could be used effectively.

## REFERENCES

[1] E. Rosten and T. Drummond(2005) Fusing points and lines for high performance tracking. in 10th IEEE International Conference on Computer Vision(ICCV'05), vol. 2, Beijing, Oct:1508–1515

[2] E. Rosten, R. Porter, and T. Drummond(2010) Faster and better: A machine learning approach to corner detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 1: 105–119

[3] Mingzhe Wang(2022) Robust and Efficient Corner Detection for Computer Vision:1-198, https://doi.org/10.26190/unsworks/1635

[4] C. Harris and M. Stephens(1988) A combined corner and edge detector. in Proc. of the Fourth ALVEY Vision Conference (ACV'88), Manchester, UK: 147–151

[5] P. R. Beaudet(1978) Rotationally invariant image operators. in Proc. of the 4th International Joint Conference on Pattern Recognition (IJCPR'78), Kyoto, Japan, Nov: 579–583

[6] K. Kohlmann(1996) Corner detection in natural images based on the 2-D Hilbert transform. Signal Processing, vol. 48, no. 3:225–234

[7] C.-H. Chen and Y.-N. Sun(1995) Wavelet transformation for gray-level corner detection. Pattern Recognition, vol. 28, no. 6: 853–861

[8] S. M. Smith and J. M. Brady(1997) SUSAN–a new approach to low level image processing. International Journal of Computer Vision, vol. 23, no. 1: 45–78

[9] J. Cooper, S. Venkatesh, and L. Kitchen(1993) Early jump-out corner detectors. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 15: 823–828

[10] F. Arrebola and F. Sandoval(2005) Corner detection and curve segmentation by multiresolution chain-code linking. Pattern Recognition, vol. 38, no. 10: 1596–1614

[11] X. Gao, F. Sattar, A. Quddus, and R. Venkateswarlu(2007) Multiscale contour corner detection based on local natural scale and wavelet transform: Image and Vision Computing, vol. 25: 890–898

[12] B. K. Ray and R. Pandyan(2003) ACORD—an adaptive corner detector for planar curves. Pattern Recognition, vol. 36, no. 3: 703–708

[13] X. Zhang, M. Lei, D. Yang, Y. Wang, and L. Ma(2007) Multi-scale curvature product for robust image corner detection in curvature scale space. Pattern Recognition Letters, vol. 28, no. 5: 545–554

[14] J. H. Han and T. Poston(2001) Chord-to-point distance accumulation and planar curvature: a new approach to discrete curvature. Pattern Recognition Letters, vol. 22, no. 10: 1133–1144

[15] X. Zhang, H. Wang, A. W. Smith, X. Ling, B. C. Lovell, and D. Yang(2010) Corner detection based on gradient correlation matrices of planar curves. Pattern Recognition, vol. 43, no. 4: 1207–1223

[16] R. Elias and R. Laganiere(2012) JUDOCA: Junction detection operator based on circumferential anchors. IEEE Transactions on Image Processing, vol. 21, no. 4: 2109–2118

[17] W.-C. Zhang, F.-P. Wang, L. Zhu, and Z.-F. Zhou(2014) Corner detection using Gabor filters. IET Image Processing, vol. 8: 639–646

[18] W.-C. Zhang and P.-L. Shui(2015) Contour-based corner detection via angle difference of principal directions of anisotropic Gaussian directional derivatives. Pattern Recognition, vol. 48, no. 9: 2785–2797

[19] R. Mur-Artal and J. D. Tardos.( 2017) ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras. DOI: 10.1109/TRO.2705103, IEEE Xplore: http://ieeexplore.ieee.org/document/7946260/

[20] Hanxiao Rong, Yanbin Gao, Lianwu Guan, Alex Ramirez-Serrano, Xu Xu and Yunyu Zhu(2021) Point-Line Visual Stereo SLAM Using EDlines and PL-BoW. Remote Sens, 13, 3591. https://doi.org/10.3390/rs13183591

[21] Yanyan Li1, Nikolas Brasch, Yida Wang, Nassir Navab, Federico Tombari(2020) Structure-SLAM: Low-Drift Monocular SLAM in Indoor Environments. IEEE Robotics and Automation Letters (RAL) paper presented at the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) : 25-29

[22] Huizhong Zhou, Danping Zou, Ling Pei, Rendong Ying, Peilin Liu, Member, IEEE, and Wenxian Yu(2015) StructSLAM: Visual SLAM With Building Structure Lines. IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, VOL. 64, NO. 4

[23] Haydar Abdulameer Kadhim, Waleed Abdullah Araheemah(2020) A method to improve corner detectors (Harris, Shi-Tomasi & FAST) using adaptive contrast enhancement filter . Periodicals of Engineering and Natural Sciences ISSN 2303-4521 Vol. 8, No. 1: 508-515

[24] Mingzhe Wang, Weichuan Zhang, Changming Sun, Arcot Sowmya((2020)) Corner detection based on shearlet transform and multi-directional structure tensor. Pattern Recognition 103 : 107-299

[25] Shizheng Zhang, Luwen Huangfu, Zhifeng Zhang, Sheng Huang, Pu Li, Heng Wang(2020) Corner detection using the point-to-centroid distance technique, Corner detection using the point-to-centroid distance technique. *IET Image Process.*,, Vol. 14 Iss. 14: 3385-3392

[26] Les KITCHEN and Azriel ROSENFELD(1982) Gray-level corner detection. Pattern Recognition Letters: 195-102

[27] D.J. LANGRIDGE(1982) Curve Encoding and the Detection of Discontinuities. COMPUTER GRAPHICS AND IMAGE PROCESSING u): 58-71

[28] Mohammad Reza Alsharif , Md. Foisal Hossain and Miyahira Yoshihiko(2017) Image Corner Detection Based on Curvature Scale Space and Adaptive Thresholding. E-mail: asharif@ie.u-ryukyu.ac.jp, k068661@eve.u-ryukyu.ac.jp, e055760@ie.u-ryukyu.ac.jp, Department of Information Engineering, Ryukyu University

[29] R. M. N. Sadat, S. W. Teng and G. Lu(2011) An Effective and Efficient Contour-based Corner Detector using Simple Triangular Theory. Pacific Graphics

[30] Mingzhe Wang, Weichuan Zhang, Changming Sun, Arcot Sowmya(2020) Corner detection based on shearlet transform and multi-directional structure tensor. Pattern Recognition 103 :107-299

[31] Shizheng Zhang, Luwen Huangfu, Zhifeng Zhang, Sheng Huang, Pu Li, Heng Wang(2020) Corner detection using the point-to-centroid distance technique. IET Image Process., Vol. 14 Iss. 14: 3385-3392

[32] Jie Ren, Wenya Yu(2023) Second-order Anisotropic Gaussian Directional Derivative Filters for Blob Detection. School of Electronics and Information, Xi'an Polytechnic University, Xi'an, China renjie@xpu.edu.cn; 210411053@stu.xpu.edu.cn, arXiv:2305.00435v1 [cs.CV] 30

[33] A. Leonardis, H. Bischof, and A. Pinz (Eds. 2006): "Machine Learning for High-Speed Corner Detection" ECCV 2006, Part I, LNCS 3951, pp. 430–443