Article

SNR Analysis for Quantitative Comparison of Line Detection Methods

Suyoung Seo

Department of Civil Engineering, Kyungpook National University, Daegu 41566, Korea; syseo@knu.ac.kr

Abstract: The need for line detection in images is growing rapidly due to its importance in many image processing applications. The selection of an appropriate line detection method is essential for accurate detection of line pixels, but few studies provide an analytical basis for selecting a specific line detection method. In this study, to solve the problem, a method to analytically determine the signal-to-noise ratio (SNR) of line detection methods is proposed. Three line detection methods were selected for comparison: edge-detection (ED)-based, second derivative (SD)-based, and the sum of gradient angle differences (SGAD)-based line detection methods. Then, this study quantifies the SNR of the three line detectors through error propagation and signal noise coupling. In addition, the derived SNRs are graphically visualized to explicitly compare the performance of line detectors. Then, the quantified SNRs were validated by showing that they are highly correlated with the completeness and correctness observed in the experiment with a set of natural images. The experimental results show that the proposed SNR analysis can be used to select or design a suitable line detector.

Keywords: error propagation; line detection; quantitative comparison; signal-to-noise ratio

Citation: Seo, S. SNR Analysis for Quantitative Comparison of Line Detection Methods. *Appl. Sci.* **2021**, *11*, 10088. <https://doi.org/10.3390/app112110088>

Academic Editor: Zhengjun Liu

Received: 6 September 2021

Accepted: 25 October 2021

Published: 28 October 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In the field of image processing, edges and lines are important features used to detect object shapes in a scene. In the literature, lines are classified as “ridge” or “valley” depending on their relative intensity compared to their neighbor intensities [1–6]. Lines are the primary features observed in various types of images and are used to detect and recognize the appearance of objects in an image. Recently, by developing various applications, the demand for efficient line extraction methods is rapidly increasing [7–13].

Frequently, edge detection (ED) is used to indirectly extract line features [7–12], and different operators have been proposed for ED [14–18]. However, ED-based methods are often inefficient for detecting lines because they require the detection of edge pixels on both sides of the lines.

Quality line extraction requires line detection with high signal-to-noise (SNR) values and high quality subpixel line localization and linking processes [5]. Moreover, although the criteria for establishing a method for detecting linear features in noisy images are always necessary, not many studies have compared the performances of line detection methods to help establish such criteria. Therefore, in this study, a quantitative comparison of the effectiveness of line detection methods, and their merits and demerits in detecting line features under varying conditions, has been described based on SNR analysis.

In this study, the influence of image smoothing on both the signal and noise strengths was first investigated. For the comparative study of line feature detectors, a second-order derivative (SD)-based method [19] was first selected and compared with an ED-based indirect method.

Moreover, the SD-based line detection method was shown to result in dislocalized line pixels under relatively large line widths [6]. A line detection method based on the addition of gradient angle differences (SGAD) was proposed to overcome the problems [6]. Evaluating line detectors based on experiments with a limited number of test images

does not comprehensively demonstrate the performance of line detectors under varying conditions. Thus, the analytical quantification of the SNR of line detectors is required to determine their efficiency. A SNR analysis to identify an optimal edge detector was used in [18]. However, a few studies have considered evaluating line detectors based on analytical quantification of their SNR. This primarily motivated the quantitative derivation of the SNRs of line detectors in this study.

A set of line detectors were compared based on their experimental characteristic curves in [20]. Moreover, multiple line detection methods were evaluated using multiple images in [21]. However, evaluating line detectors with a set of images does not comprehensively demonstrate the performance of them under varying conditions. In this study, quantitative derivation of the SNRs of line detectors overcomes the limitations of the existing evaluation methods. The contributions of this study are listed below:

- Streamlined derivation of the SNRs of line detectors including the derivation of correlations among noises after smoothing; derivations of signal and noise strengths of elementary operators; combination of SNRs for line detectors; and application of a penalty function for considering the influence of blurring, smoothing, and line width on line detectors.
- Analytical quantification of the SNRs of line detectors using error propagation and visualization of the quantity of the derived values including signal strengths, noise strengths, and SNRs.
- Verification of the validity of the SNRs derived for the SD- and SGAD-based line detectors by investigating the relationship of the SNRs with completeness and correctness of their line detection results.

The rest of this study is organized as follows. Section 2 describes the related studies for line detection. Section 3 compares the performance of the ED-based and SD-based line detections based on SNR analysis. Section 4 describes the SNR of the SGAD-based line detection method. The results obtained using real images are described in Section 5 and are followed by the conclusion in Section 6.

2. Related Work

Pixels in an image were classified into edges, ridges, valleys, and other geometric features based on the analysis of the coefficients resulting from fitting a Legendre polynomial of 2 degrees or less with local intensity in [22]. However, this approach has a nontrivial problem with line detection because it requires multiple steps to detect line pixels and a set of well-tuned thresholds. In another study, ridges and valleys were detected by finding zero crossings of the first-order directional derivatives in [1]. The method was based on 10 coefficients resulting from fitting a bivariate cubic polynomial to local intensities. One of the limitations of this method is that it may produce nontrivial ringing artifacts, and the approach requires a large window size for the 10 coefficients of the bivariate cubic polynomial to be calculated. A Zernike moment-based edge and line detection method was proposed in [23], which was a parameter-based edge and line detection method. Differential geometry was used to detect the line pixels in [2,3]. One of the limitations with this approach is that it requires a high processing time to calculate the first-, second-, and third-order derivatives. A line detection approach based on second-order derivatives was proposed in [19,24]. In a similar study, this SD-based approach was used to detect line pixels in aerial images and synthetic aperture radar (SAR) images [25,26]. However, as it will be shown in the following sections, the SD-based line detection methods have limitations under relatively large line widths. A multi-scale-based line detection approach was proposed in [24]. This approach may, however, face limitations when line features are located close to other non-homogeneous features. These limitations are attributed to the blending of neighboring features at a larger scale.

A simple line detection method using relational operations was proposed in [4]. Although its computational cost is low, one of the limitations of this approach is that selecting an appropriate window size depends on the width of a line to be detected.

However, multi-scale anisotropic Gaussian kernels were used for line detection in [27]. This approach has a problem of high computational cost. Moreover, a multi-step-based line detection approach was proposed in [28], and its computation is cost-intensive because of the convolution of a set of filters and a sequence of steps. Recently, a convolutional neural network-based approach was proposed to detect the boundaries among walls, floors, and ceilings in [29]. However, this approach requires a well-trained network to implement boundary detection and detects only predefined types of boundary pixels such as wall–wall, wall–ceiling, and wall–floor.

Furthermore, a creaseness measure was used to detect line pixels in [21]. One of the limitations of this approach is that a simple sum of gradient differences may lose the direction information in detecting line pixels. The SGAD-based line detection approach resolved this limitation using the addition of absolute values of gradient angle differences [6].

3. Method—SNR of ED and SD for Line Detection

It is necessary to select between edge and line detectors when extracting linear features from an image. To make the right decision, it is necessary to know their performance under various conditions such as line width, noise level, and smoothing factor for noise suppression. Although this information is important for making the decision, presumably, very few studies have focused on this issue. To compensate for these limitations, in this section, the performances of ED and SD are compared by investigating their respective SNRs under various conditions.

3.1. Line Model and Derivation of Its Derivatives

In camera-captured images, there is always a certain amount of blur and noise. An image $\mathbf{I}$ can be modeled by signals $\mathbf{F}$ convoluted with a certain amount of blur $\mathbf{b}$ and noise $\mathbf{n}$, which is described in various approaches [30–38] as follows:

$$\mathbf{I} = \mathbf{F} * \mathbf{b} + \mathbf{n}, \quad (1)$$

where $*$ indicates the convolution operator.

Variations of blurring patterns around edges because of varying contrast have been analyzed by modeling edge profiles with two blur parameters in [39]. For simplicity, however, the imaging model with one blur parameter in Equation (1) is used in the following derivation. In this study, the line profile has been modeled by two factors, namely, width, w , and contrast, k , as shown in Figure 1. Then, according to Figure 1, the 1D line signal can be mathematically modeled as follows:

$$F(x) = \begin{cases} h + k, & \text{if } |x - L| < \frac{w}{2}; \\ h, & \text{otherwise} \end{cases} \quad (2)$$

where h , L , and x are the background intensity, coordinate of the center of the line, and coordinate of an arbitrary location, respectively. L and x are the coordinates with reference to the line normal direction.

Figure 1. Line profile model used in this study.

To consider the blurring effect in the image formation process, a 1D Gaussian blur function is introduced as follows:

$$\mathbf{b}(t) = \frac{1}{\sqrt{2\pi}\sigma_b} e^{-\frac{t^2}{2\sigma_b^2}} \quad (3)$$

where σ_b and t are the blurring factor and distance from the center of the Gaussian function, respectively. The dashed line in Figure 2 shows a scaled version of the blur function, $k \cdot \exp(-\frac{(L-x)^2}{2\sigma_b^2})$ with variation of the location, x . Thus, from Figure 2, the intensity value at x captured by a camera can be derived as follows:

$$\mathbf{F}_b(x) = (\mathbf{F} * \mathbf{b})(x) = \int_{x-\frac{w}{2}-L}^{x+\frac{w}{2}-L} \frac{k}{\sigma_b \sqrt{2\pi}} e^{-\frac{t^2}{2\sigma_b^2}} dt. \quad (4)$$

Figure 2. Gaussian blur for a line model.

For noise removal, image smoothing is usually applied by convoluting an image with a 2D Gaussian function, which can be described as follows:

$$\mathbf{s}(u, v) = \frac{1}{2\pi\sigma_s^2} e^{-\frac{u^2+v^2}{2\sigma_s^2}}, \quad (5)$$

where σ_s is the smoothing factor, and u and v are the distances from the center of the Gaussian function in the column and row directions, respectively. Accordingly, its 1D function can be written as follows:

$$\mathbf{s}(x) = \frac{1}{\sqrt{2\pi}\sigma_s} e^{-\frac{x^2}{2\sigma_s^2}} \quad (6)$$

Then, the application of the smoothing function $\mathbf{s}$ generates the image $\mathbf{I}_s$, which can be written as follows:

$$\mathbf{I}_s = \mathbf{I} * \mathbf{s} = (\mathbf{F} * \mathbf{b} + \mathbf{n}) * \mathbf{s} = \mathbf{F} * \mathbf{b} * \mathbf{s} + \mathbf{n} * \mathbf{s}. \quad (7)$$

Thus, after blurring and smoothing, the line signal in Equation (2) is reformed as per [40] as follows:

$$\mathbf{F}_{b*s}(x) = (\mathbf{F} * \mathbf{b} * \mathbf{s})(x) = (\mathbf{F}_b * \mathbf{s})(x) = \int_{x-\frac{w}{2}-L}^{x+\frac{w}{2}-L} \frac{k}{\sqrt{2\pi(\sigma_b^2 + \sigma_s^2)}} e^{-\frac{t^2}{2(\sigma_b^2 + \sigma_s^2)}} dt. \quad (8)$$

Thus, the first derivative of the cross section profile of the smoothed line with respect to x is derived from Equation (8) as follows:

$$\mathbf{F}'_{b*s}(x) = \frac{d\mathbf{F}_{b*s}}{dx} = \frac{k}{\sqrt{2\pi(\sigma_b^2 + \sigma_s^2)}} \left[e^{-\frac{(x+\frac{w}{2}-L)^2}{2(\sigma_b^2 + \sigma_s^2)}} - e^{-\frac{(x-\frac{w}{2}-L)^2}{2(\sigma_b^2 + \sigma_s^2)}} \right]. \quad (9)$$

Additionally, the second derivative of the cross section profile of the smoothed line with respect to x is derived from Equation (9) as follows:

$$\begin{aligned} F''_{b*s}(x) &= \frac{d^2 F_{b*s}}{dx^2} \\ &= -\frac{k}{\sqrt{2\pi}(\sigma_b^2 + \sigma_s^2)^{\frac{3}{2}}} \left[e^{-\frac{(x+\frac{w}{2}-L)^2}{2(\sigma_b^2 + \sigma_s^2)}} (x + \frac{w}{2} - L) - e^{-\frac{(x-\frac{w}{2}-L)^2}{2(\sigma_b^2 + \sigma_s^2)}} (x - \frac{w}{2} - L) \right]. \end{aligned} \quad (10)$$

3.2. Measure of Signal Strengths

Edge strengths are measured at the boundaries of smoothed line model in terms of the values of first and second derivatives. The first derivative indicates the gradient of the smoothed edge profile and is derived by substituting $x = L - \frac{w}{2}$ in Equation (9) as follows:

$$F'_{b*s}|_{x=L-\frac{w}{2}} = \frac{k}{\sqrt{2\pi}(\sigma_b^2 + \sigma_s^2)} \left[1 - e^{-\frac{w^2}{2(\sigma_b^2 + \sigma_s^2)}} \right]. \quad (11)$$

To distinctly determine an edge pixel in the local areas in an image in a non-maxima suppression process, the absolute values of the gradients at its neighboring pixels should be sufficiently lower than the absolute values at the edge location, $x = L - \frac{w}{2}$. Thus, the absolute value of the second derivatives of the neighborhood pixels should be high. Therefore, the neighborhood pixel, which is one pixel away from the edge location, is selected. Moreover, the second derivative at the pixel location $x = L - \frac{w}{2} - 1$ is derived as another measure of edge strength from Equation (10) as follows:

$$F''_{b*s}|_{x=L-\frac{w}{2}-1} = \frac{k}{\sqrt{2\pi}(\sigma_b^2 + \sigma_s^2)^{\frac{3}{2}}} \left[e^{-\frac{1}{2(\sigma_b^2 + \sigma_s^2)}} - e^{-\frac{(1+w)^2}{2(\sigma_b^2 + \sigma_s^2)}} (1 + w) \right]. \quad (12)$$

Alternatively, to distinctly determine a line pixel in a non-maxima suppression process, the absolute values of the first derivatives or the gradients of the neighborhood pixel of the line pixel should be high. Therefore, as one of the neighborhood pixels, a pixel located at $x = L - 1$ is selected. The first derivative of this pixel, as a measure of line strength, is derived from Equation (9) as follows:

$$F'_{b*s}|_{x=L-1} = \frac{k}{\sqrt{2\pi}(\sigma_b^2 + \sigma_s^2)} \left[e^{-\frac{(1-\frac{w}{2})^2}{2(\sigma_b^2 + \sigma_s^2)}} - e^{-\frac{(1+\frac{w}{2})^2}{2(\sigma_b^2 + \sigma_s^2)}} \right]. \quad (13)$$

Moreover, for detecting a line pixel, its second derivative at the line location $x = L$ must be high, which can be derived from Equation (10) as follows:

$$F''_{b*s}|_{x=L} = -\frac{k w}{\sqrt{2\pi}(\sigma_b^2 + \sigma_s^2)^{\frac{3}{2}}} e^{-\frac{w^2}{8(\sigma_b^2 + \sigma_s^2)}}. \quad (14)$$

3.3. Measure of Noise Strengths

To measure the SNR for line detection in a smoothed image, it is necessary to quantify the amount of noise after smoothing and the signal strengths. However, there have been few studies on the quantification of the amount of noise after smoothing. Thus, in the following section, the residual noise after the smoothing is first quantified, and the correlation between the pixels in an image of smoothed noises is derived using an error propagation scheme.

3.3.1. Correlation of Noises in Smoothed Images

First, the amount of noise at an arbitrary location (r, c) in a noise image is denoted by n as follows:

$$n = \mathbf{n}(r, c). \quad (15)$$

Noise n is assumed to be symmetrically distributed around zero, and its expectation $E\{n\}$ satisfies the following equation:

$$E\{n\} = 0. \quad (16)$$

Then, the dispersion or variance of the noise at a single pixel σ_n^2 is defined as follows:

$$\sigma_n^2 = D\{n\} = E\{[n - E\{n\}]^2\} = E\{n^2\}. \quad (17)$$

By extrapolating the expectation and dispersion of the noise to the noises of all the pixels in an image, the expectation and dispersion of the vector containing all noises can be written as follows:

$$\begin{aligned} E\{\text{vec}(\mathbf{n})\} &= \mathbf{0}, \\ D\{\text{vec}(\mathbf{n})\} &= \sigma_n^2 \mathcal{I}_c. \end{aligned} \quad (18)$$

where $\text{vec}(\cdot)$ is an operation converting a matrix into a column vector and $\mathcal{I}_c$ is the continuous version of the identity matrix defined as $\mathcal{I}_c(a, b) = 1$ if $a = b$, else $\mathcal{I}_c(a, b) = 0$.

The noise amount remaining after smoothing at an arbitrary location (r, c) can be modeled as follows:

$$(\mathbf{n} * \mathbf{s})(r, c) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathbf{n}(r - u, c - v) \mathbf{s}(u, v) du dv. \quad (19)$$

Furthermore, the expectation of the vectors of the smoothed noises is derived as follows:

$$E\{\text{vec}(\mathbf{n} * \mathbf{s})\} = \text{vec}(E\{\mathbf{n}\} * \mathbf{s}) = \mathbf{0}. \quad (20)$$

Moreover, the dispersion of the vectors of the smoothed noises is derived as follows:

$$\begin{aligned} D\{\text{vec}(\mathbf{n} * \mathbf{s})\} &= E\{[\text{vec}(\mathbf{n} * \mathbf{s}) - E\{\text{vec}(\mathbf{n} * \mathbf{s})\}][\text{vec}(\mathbf{n} * \mathbf{s}) - E\{\text{vec}(\mathbf{n} * \mathbf{s})\}]^T\} \\ &= E\{[\text{vec}(\mathbf{n} * \mathbf{s})][\text{vec}(\mathbf{n} * \mathbf{s})]^T\} \end{aligned} \quad (21)$$

To quantify the correlation between the noises remaining after the smoothing, the covariance between them is first derived for two arbitrary pixels at (r, c) and $(r - \alpha, c - \beta)$ as follows:

$$\begin{aligned} \sigma_{\mathbf{n} * \mathbf{s}}|_{(\alpha, \beta)} &= \text{cov}\{(\mathbf{n} * \mathbf{s})|_{(r, c)}, (\mathbf{n} * \mathbf{s})|_{(r - \alpha, c - \beta)}\} \\ &= E\{(\mathbf{n} * \mathbf{s})|_{(r, c)} (\mathbf{n} * \mathbf{s})|_{(r - \alpha, c - \beta)}\} \\ &= \sigma_n^2 \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{1}{2\pi\sigma_s^2} e^{-\frac{\xi^2 + \eta^2}{2\sigma_s^2}} \frac{1}{2\pi\sigma_s^2} e^{-\frac{(\xi - \alpha)^2 + (\eta - \beta)^2}{2\sigma_s^2}} d\xi d\eta \\ &= \frac{\sigma_n^2}{4\pi^2\sigma_s^4} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-\frac{\xi^2 + \eta^2 + (\xi - \alpha)^2 + (\eta - \beta)^2}{2\sigma_s^2}} d\xi d\eta \\ &= \frac{\sigma_n^2}{4\pi^2\sigma_s^4} \cdot e^{-\frac{\alpha^2 + \beta^2}{4\sigma_s^2}} \cdot \pi\sigma_s^2 \frac{1}{2\pi(\frac{\sigma_s}{\sqrt{2}})^2} \cdot \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-\frac{(\xi - \frac{\alpha}{\sqrt{2}})^2 + (\eta - \frac{\beta}{\sqrt{2}})^2}{2(\frac{\sigma_s}{\sqrt{2}})^2}} d\xi d\eta \\ &= \frac{\sigma_n^2}{4\pi\sigma_s^2} e^{-\frac{\alpha^2 + \beta^2}{4\sigma_s^2}}. \end{aligned} \quad (22)$$

If we let d be the distance between the two positions (r, c) and $(r - \alpha, c - \beta)$, then d is calculated as follows:

$$d = (\alpha^2 + \beta^2)^{\frac{1}{2}}. \quad (23)$$

Now, according to Equation (22), the covariance between the two pixels can be calculated as follows:

$$\sigma_{\mathbf{n}*\mathbf{s}}|_d = \frac{\sigma_n^2}{4\pi\sigma_s^2} e^{-\frac{d^2}{4\sigma_s^2}}. \quad (24)$$

Then, the dispersion $D\{\text{vec}(\mathbf{n} * \mathbf{s})\}$, which is the auto-covariance of the remaining noise at (r, c) , is derived from Equation (24) as follows:

$$\sigma_{\mathbf{n}*\mathbf{s}}^2 = \sigma_{\mathbf{n}*\mathbf{s}}|_{d=0} = \frac{\sigma_n^2}{4\pi\sigma_s^2}. \quad (25)$$

Therefore, the correlation among noises remaining after the convolution with a smoothing function $\mathbf{s}$ for two arbitrary pixels separated by a distance d is derived from Equations (24) and (25) as follows:

$$\rho_d = \frac{\sigma_{\mathbf{n}*\mathbf{s}}|_d}{\sigma_{\mathbf{n}*\mathbf{s}}^2} = e^{-\frac{d^2}{4\sigma_s^2}}. \quad (26)$$

3.3.2. Measure of Noise in Derivatives

To extract edges or lines, the calculation of the first and second derivatives is typically performed after smoothing. While calculating the first and second derivatives, the post-smoothing residual noise propagates, which should be quantified for computing the SNR of ED-based and SD-based line detections. Calculating the first and second derivatives in an image is implemented by applying a convolution of certain kernels to the image. In this study, the kernels with size 3×3 are used for the implementation. Accordingly, to investigate the propagation of noise, while calculating the first and second derivatives, it is necessary to consider the smoothed noises in the 3×3 neighborhood at each pixel as follows:

$$(\mathbf{n} * \mathbf{s})_{3 \times 3} = \begin{bmatrix} (\mathbf{n} * \mathbf{s})_1 & (\mathbf{n} * \mathbf{s})_2 & (\mathbf{n} * \mathbf{s})_3 \\ (\mathbf{n} * \mathbf{s})_4 & (\mathbf{n} * \mathbf{s})_5 & (\mathbf{n} * \mathbf{s})_6 \\ (\mathbf{n} * \mathbf{s})_7 & (\mathbf{n} * \mathbf{s})_8 & (\mathbf{n} * \mathbf{s})_9 \end{bmatrix}, \quad (27)$$

where i in $(\mathbf{n} * \mathbf{s})_i$ is the unique sequential number assigned to each pixel within a 3×3 kernel. After rearranging the pixels in a vector ordered by their sequential numbers, their correlations are calculated by Equation (26) by considering their distance, and transformed into a correlation matrix, $\mathbf{R}_{(\mathbf{n}*\mathbf{s})_{3 \times 3}}$ as follows:

$$\mathbf{R}_{(\mathbf{n}*\mathbf{s})_{3 \times 3}} = \begin{bmatrix} 1 & \rho_1 & \rho_2 & \rho_1 & \rho_{\sqrt{2}} & \rho_{\sqrt{5}} & \rho_2 & \rho_{\sqrt{5}} & \rho_{2\sqrt{2}} \\ & 1 & \rho_1 & \rho_{\sqrt{2}} & \rho_1 & \rho_{\sqrt{2}} & \rho_{\sqrt{5}} & \rho_2 & \rho_{\sqrt{5}} \\ & & 1 & \rho_{\sqrt{5}} & \rho_{\sqrt{2}} & \rho_1 & \rho_{2\sqrt{2}} & \rho_{\sqrt{5}} & \rho_2 \\ & & & 1 & \rho_1 & \rho_2 & \rho_1 & \rho_{\sqrt{2}} & \rho_{\sqrt{5}} \\ & & & & 1 & \rho_{\sqrt{2}} & \rho_1 & \rho_{\sqrt{2}} \\ & & & & & 1 & \rho_{\sqrt{5}} & \rho_{\sqrt{2}} & \rho_1 \\ & & & & & & 1 & \rho_1 & \rho_2 \\ & & & & & & & 1 & \rho_1 \\ & & & & & & & & 1 \end{bmatrix}. \quad (28)$$

To calculate the first derivative in the column direction, this study used a scaled Sobel operator such that it should produce the gradient for one pixel unit as follows:

$$\mathbf{D}_c = \frac{1}{8} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}. \quad (29)$$

The kernel in Equation (29) can be rewritten in its vector form as follows:

$$\mathbf{V}_{D_c} = \text{vec}(\mathbf{D}_c^T). \quad (30)$$

Then, the dispersion $\sigma_{\mathbf{n}*\mathbf{s}*\mathbf{D}_c}^2$ of the noise resulting from the convolution of the smoothed noise with the kernel in Equation (29) can be quantitatively derived as follows:

$$\sigma_{\mathbf{n}*\mathbf{s}*\mathbf{D}_c}^2 = \sigma_{\mathbf{n}*\mathbf{s}}^2 \mathbf{V}_{D_c}^T \mathbf{R}_{(\mathbf{n}*\mathbf{s})_{3 \times 3}} \mathbf{V}_{D_c} = \frac{\sigma_n^2}{64\pi\sigma_s^2} (3 + 4\rho_1 - 2\rho_2 - 4\rho_{\sqrt{5}} - \rho_{2\sqrt{2}}). \quad (31)$$

Moreover, to calculate the second derivative in the column direction, this study used the following kernel such that it should produce the second derivative for one pixel unit, defined as follows:

$$\mathbf{D}_{cc} = \frac{1}{4} \begin{bmatrix} 1 & -2 & 1 \\ 2 & -4 & 2 \\ 1 & -2 & 1 \end{bmatrix}. \quad (32)$$

The kernel in Equation (32) can be represented in the vector form as follows:

$$\mathbf{V}_{D_{cc}} = \text{vec}(\mathbf{D}_{cc}^T). \quad (33)$$

Then, the dispersion $\sigma_{\mathbf{n}*\mathbf{s}*\mathbf{D}_{cc}}^2$ of the noise resulting from the convolution of the smoothed noise with the kernel in Equation (32) can be quantitatively derived as follows:

$$\sigma_{\mathbf{n}*\mathbf{s}*\mathbf{D}_{cc}}^2 = \sigma_{\mathbf{n}*\mathbf{s}}^2 \mathbf{V}_{D_{cc}}^T \mathbf{R}_{(\mathbf{n}*\mathbf{s})_{3 \times 3}} \mathbf{V}_{D_{cc}} = \frac{\sigma_n^2}{16\pi\sigma_s^2} (9 - 16\rho_{\sqrt{2}} + 6\rho_2 + \rho_{2\sqrt{2}}). \quad (34)$$

3.4. Derivation of SNRs

In the following section, the SNRs of ED-based and SD-based line detections are derived using the signal strengths quantified in Section 3.2 and the amount of noise resulting from the smoothing and convolution of specific kernels described in Section 3.3. To make the calculation simple, the normal direction of lines to be detected is assumed to be aligned in the column direction in the following derivations.

To detect edge pixels at the boundaries of a line, its SNR for the first derivative in the normal direction is derived from Equations (11) and (31) as follows:

$$\text{SNR}(\text{edge}_{D_c}) = \frac{\mathbf{F}'_{\mathbf{b}*\mathbf{s}}|_{x=L-\frac{w}{2}}}{\sigma_{\mathbf{n}*\mathbf{s}*\mathbf{D}_c}}. \quad (35)$$

Similarly, the edge SNR for the second derivative in the normal direction can be derived from Equations (12) and (34). However, in this study, the second derivative of the edge profile model is assumed to be positive at $x = L - \frac{w}{2} - 1$. Thus, the $\max(\cdot)$ operation is applied to suppress the negative signals and the edge SNR for the second derivative in the normal direction is derived as follows:

$$\text{SNR}(\text{edge}_{D_{cc}}) = \frac{\max(\mathbf{F}''_{\mathbf{b}*\mathbf{s}}|_{x=L-\frac{w}{2}-1}, 0)}{\sigma_{\mathbf{n}*\mathbf{s}*\mathbf{D}_{cc}}}. \quad (36)$$

Given that the edge detection is performed based on combining the first and second derivatives in this study, the SNRs for both the derivatives are combined into the measure $\text{SNR}_C(\text{edge})$ as follows:

$$\text{SNR}_C(\text{edge}) = \sqrt{\text{SNR}(\text{edge}_{D_c}) \cdot \text{SNR}(\text{edge}_{D_{cc}})}. \quad (37)$$

The SNR of the SD-based line detection for the first derivative in the normal direction can be derived from Equations (13) and (31) as follows:

$$\text{SNR}(\text{line}_{D_c}) = \frac{\mathbf{F}'_{b*s}|_{x=L-1}}{\sigma_{n*s*D_c}}. \quad (38)$$

The SNR of SD-based line detection for the second derivative in the normal direction can be derived from Equations (14) and (34) as follows:

$$\text{SNR}(\text{line}_{D_{cc}}) = -\frac{\mathbf{F}''_{b*s}|_{x=L}}{\sigma_{n*s*D_{cc}}}. \quad (39)$$

Then, because the SD-based line detection is considered to be performed based on combining the first and second derivatives in this study, the SNRs of both the derivatives are combined into the measure $\text{SNR}_C(\text{line}_{SD})$ as follows:

$$\text{SNR}_C(\text{line}_{SD}) = \sqrt{\text{SNR}(\text{line}_{D_c}) \cdot \text{SNR}(\text{line}_{D_{cc}})}, \quad (40)$$

where the subscript SD in $\text{SNR}_C(\text{line}_{SD})$ represents the SNR measure for the SD-based line detection so that it can be distinguished from that of the SGAD-based line detection.

With the increase in sizes of blurring and smoothing, the signals tend to mix with their neighborhood signals and degenerate. Based on this rationale, in this study, a penalty function is introduced to measure the degeneration of signals with the increase in the sizes of blurring and smoothing. However, as line width increases, the degeneration of signal owing to blurring and smoothing decreases. Accordingly, the penalty function is modeled as follows:

$$p(\sigma_b, \sigma_s, w) = w^\lambda (\sigma_b^2 + \sigma_s^2)^{-1}, \quad (41)$$

where λ is the power factor applied to line width, w , so that it can reflect the decrease of degeneration effect of signal strength because of blurring and smoothing. From a set of brief experiments, the value of λ was set to 0.3 in this study. Thus, the SNR of edge detection with the penalty for blurring and smoothing, i.e., $\text{SNR}_{PS}(\text{edge})$, is calculated as follows:

$$\text{SNR}_{PS}(\text{edge}) = p(\sigma_b, \sigma_s, w) \cdot \text{SNR}_C(\text{edge}). \quad (42)$$

Moreover, the SNR of the SD-based line detection with the penalty for blurring and smoothing, i.e., $\text{SNR}_{PS}(\text{line}_{SD})$, is calculated as follows:

$$\text{SNR}_{PS}(\text{line}_{SD}) = p(\sigma_b, \sigma_s, w) \cdot \text{SNR}_C(\text{line}_{SD}). \quad (43)$$

Because the detection of edge pixels is required at either sides of the line to find the line pixel, the SNR_{PS} of edge detection for the purpose of line detection is measured by halving $\text{SNR}_{PS}(\text{edge})$ in Equation (42) as follows:

$$\text{SNR}_{PS}(\text{line}_{ED}) = \frac{\text{SNR}_{PS}(\text{edge})}{2}. \quad (44)$$

To compare the performance of the ED-based and SD-based line detections, a set of graphical investigations was used. The tests were performed with various values of the smoothing factor σ_s and the line width w for investigation. The smoothing factor was set

to vary from 0.4 to 10.0 pixels with an interval of 0.1 pixels, and the line width was set to vary from 0.5 to 20.0 pixels with an interval of 0.1 pixels. The blurring factor σ_b was set to 1.0, which is a reasonable value to represent the amount of blurs observed in many camera-captured images [39]. The standard deviation of noise and contrast were set to $\sigma_n = 0.05$ and $k = 1.0$, respectively, in the following graphical investigations.

Figure 3a shows the SNR_C of edge detection. In Figure 3a, the SNR becomes very high, when the smoothing factor is set to be large and distinct, particularly when the line width is large. However, it is not realistic because the signals become interspersed with other signals with increase in the smoothing factor. Thus, according to Equation (42), a realistic measure of SNR is obtained by applying the penalty function to $\text{SNR}_C(\text{edge})$. The resulting SNR_{PS} is shown in Figure 3b.

Figure 3. SNR_C and SNR_{PS} of edge detection. (a) $\text{SNR}_C(\text{edge})$ and (b) $\text{SNR}_{PS}(\text{edge})$.

Figure 4a shows the $\text{SNR}_C(\text{line}_{SD})$. Similarly, in Figure 4a, the SNR becomes extremely high when the smoothing factor is set to be large, and a realistic measure of SNR is obtained by applying the penalty function to $\text{SNR}_C(\text{line}_{SD})$ according to Equation (43). The resulting $\text{SNR}_{PS}(\text{line}_{SD})$ is shown in Figure 4b.

Figure 4. SNR_C and SNR_{PS} of the SD-based line detection. (a) $\text{SNR}_C(\text{line}_{SD})$; (b) $\text{SNR}_{PS}(\text{line}_{SD})$.

The $\text{SNR}_{PS}(\text{line}_{ED})$ is shown in Figure 5a, which is calculated using Equation (44). Then, Figure 5b shows the difference in $\text{SNR}_{PS}(\text{line}_{SD}) - \text{SNR}_{PS}(\text{line}_{ED})$. According to Figure 5b, when the line width is relatively small, the SD-based line detection in terms of

SNR is shown to be more effective than ED-based line detection. For example, the SD-based line detection is more effective than ED-based line detection when the line width w is less than 5 pixels with a smoothing factor of 1.0, and less than 11 pixels when a smoothing factor of 3.0 is applied to an image. Moreover, the highest values in $\text{SNR}_{\text{PS}}(\text{line}_{\text{SD}}) - \text{SNR}_{\text{PS}}(\text{line}_{\text{ED}})$ against the varying line widths are observed when a smoothing factor within the range from 1.0 to 2.0 is applied to an image with the line width within the range of 4 pixels.

Figure 5. Comparison of SNRs of the ED-based and the SD-based line detections. (a) $\text{SNR}_{\text{PS}}(\text{line}_{\text{ED}})$ and (b) $\text{SNR}_{\text{PS}}(\text{line}_{\text{SD}}) - \text{SNR}_{\text{PS}}(\text{line}_{\text{ED}})$. In panel (b), areas with negative values are denoted with white grid lines.

4. Method—SNR of SGAD for Line Detection

In this section, a line detection method, based on the sum of gradient angle differences [6], is introduced. Then, its performance is investigated based on its SNR derivations. Subsequently, the SGAD-based line detection can be compared with the SD-based line detection in terms of their SNRs.

4.1. Definition of SGAD

According to the work in [6], the gradient angle at each pixel can be derived as follows:

$$\theta_i = \tan^{-1} \left(\frac{g_{r_i}}{g_{c_i}} \right), \quad (45)$$

where g_{r_i} and g_{c_i} are the gradients at pixel i in the row and column directions, respectively.

Then, the gradient angle difference (GAD) of two pixels is defined as the minimum positive angle between their gradient vectors as follows:

$$\delta_{i,j} = \min(\text{abs}(\theta_i - \theta_j), 2\pi - \text{abs}(\theta_i - \theta_j)) \quad (46)$$

The angle difference is calculated for the pairs of gradient vectors (Figure 6). Then, the measure, SGAD is derived by adding the gradient angle differences as follows:

$$\text{SGAD} = \delta_{1,8} + \delta_{2,7} + \delta_{3,6} + \delta_{4,5}. \quad (47)$$

Figure 6. Pairings for calculating gradient angle differences.

4.2. Dispersion of Gradient Angle Differences

The dispersion of the single-angle difference between the i th and the j th pixels, $D\{\theta_i - \theta_j\}$ is calculated as follows:

$$D\{\theta_i - \theta_j\} = \begin{bmatrix} 1 & -1 \end{bmatrix} D \left\{ \begin{bmatrix} \theta_i \\ \theta_j \end{bmatrix} \right\} \begin{bmatrix} 1 \\ -1 \end{bmatrix}. \quad (48)$$

The dispersion of the vector of the angles in Equation (48) is derived as follows:

$$D \left\{ \begin{bmatrix} \theta_i \\ \theta_j \end{bmatrix} \right\} = \mathbf{J}_{\theta(i,j)} D \left\{ \begin{bmatrix} g_{c_i} \\ g_{r_i} \\ g_{c_j} \\ g_{r_j} \end{bmatrix} \right\} \mathbf{J}_{\theta(i,j)}^T, \quad (49)$$

where

$$\mathbf{J}_{\theta(i,j)} = \begin{bmatrix} \frac{\partial \theta_i}{\partial g_{c_i}} & \frac{\partial \theta_i}{\partial g_{r_i}} & 0 & 0 \\ 0 & 0 & \frac{\partial \theta_j}{\partial g_{c_j}} & \frac{\partial \theta_j}{\partial g_{r_j}} \end{bmatrix},$$

$$\frac{\partial \theta}{\partial g_{c_i}} = -\frac{g_{r_i}}{g_{c_i}^2 + g_{r_i}^2},$$

$$\frac{\partial \theta}{\partial g_{r_i}} = \frac{g_{c_i}}{g_{c_i}^2 + g_{r_i}^2}.$$

Then, the dispersion of the vector of the gradients in Equation (49) is derived as follows:

$$D \left\{ \begin{bmatrix} g_{c_i} \\ g_{r_i} \\ g_{c_j} \\ g_{r_j} \end{bmatrix} \right\} = \mathbf{J}_{g(i,j)} \sigma_{\mathbf{n} * \mathbf{s}}^2 \mathbf{R}_{i,j} \mathbf{J}_{g(i,j)}^T. \quad (50)$$

The value of the gradient in the column direction at the i th pixel is calculated by convoluting the smoothed image with the kernel $\mathbf{D}_c$ as follows:

$$g_{c_i} = (\mathbf{I} * \mathbf{s} * \mathbf{D}_c)_i = (\mathbf{F} * \mathbf{b} * \mathbf{s} * \mathbf{D}_c)_i + (\mathbf{n} * \mathbf{s} * \mathbf{D}_c)_i \quad (51)$$

To calculate the gradients in the row direction, a kernel is defined as follows:

$$\mathbf{D}_r = \frac{1}{8} \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}. \quad (52)$$

Then, the value of the gradient in the row direction at the i th pixel is calculated using the convolution of the smoothed image with the kernel $\mathbf{D}_r$ as follows:

$$g_{r_i} = (\mathbf{I} * \mathbf{s} * \mathbf{D}_r)_i = (\mathbf{F} * \mathbf{b} * \mathbf{s} * \mathbf{D}_r)_i + (\mathbf{n} * \mathbf{s} * \mathbf{D}_r)_i. \quad (53)$$

Figure 7 shows the windows for calculating the gradients, where the numbers within the pixels represent the pixel indices used to indicate the pixel locations in the Jacobian matrices below. In Figure 7a, the center pixel is located at 8 and the gradient angle difference is calculated for the pixels located at 5 and 11. In Figure 7b, the center pixel is located at 9 and the gradient angle difference is calculated for the pixels located at 5 and 13.

Figure 7. Neighborhood pixels for calculating gradient angle difference. (a) Column direction; (b) Diagonal direction.

Then, the Jacobian matrix $J_{g(i,j)}$ in Equation (50) when $i = 4$ and $j = 5$ is derived using D_c and D_r . Moreover, the Jacobian matrix, $J_{g(i,j)}$ in Equation (50) when $i = 1$ and $j = 8$ is derived using D_c and D_r . The correlation matrix $R_{i,j}$ in Equation (50) when $i = 4$ and $j = 5$ is derived as per Figure 7a and Equation (26). The correlation matrix $R_{i,j}$ in Equation (50) when $i = 1$ and $j = 8$ is derived as per Figure 7b and Equation (26).

Thus, the dispersion of the vector of the gradients in Equation (50) when $i = 4$ and $j = 5$ is derived as follows:

$$D \left\{ \begin{bmatrix} g_{c_4} \\ g_{r_4} \\ g_{c_5} \\ g_{r_5} \end{bmatrix} \right\} = \frac{\sigma_{n*s}^2}{64} \begin{bmatrix} P_1 & 0 & P_2 & 0 \\ 0 & P_1 & 0 & P_3 \\ P_2 & 0 & P_1 & 0 \\ 0 & P_3 & 0 & P_1 \end{bmatrix}, \quad (54)$$

where

$$\begin{aligned} P_1 &= 12 + 16\rho_1 - 8\rho_2 - 16\rho_{\sqrt{5}} - 4\rho_{2\sqrt{2}}, \\ P_2 &= -6 - 8\rho_1 + 10\rho_2 + 16\rho_{\sqrt{5}} + 4\rho_{2\sqrt{2}} - 6\rho_4 - 8\rho_{\sqrt{17}} - 2\rho_{2\sqrt{5}}, \\ P_3 &= 2 + 8\rho_1 + 10\rho_2 - 8\rho_{\sqrt{5}} - 12\rho_{2\sqrt{2}} + 8\rho_3 - 8\rho_{\sqrt{13}} + 2\rho_4 - 2\rho_{2\sqrt{5}}. \end{aligned}$$

Furthermore, the dispersion of the vector of the gradients in Equation (50) when $i = 1$ and $j = 8$ is derived as follows:

$$D \left\{ \begin{bmatrix} g_{c_1} \\ g_{r_1} \\ g_{c_8} \\ g_{r_8} \end{bmatrix} \right\} = \frac{\sigma_{n*s}^2}{64} \begin{bmatrix} Q_1 & 0 & Q_2 & Q_3 \\ 0 & Q_1 & Q_3 & Q_2 \\ Q_2 & Q_3 & Q_1 & 0 \\ Q_3 & Q_2 & 0 & Q_1 \end{bmatrix}, \quad (55)$$

where

$$\begin{aligned} Q_1 &= 12 + 16\rho_1 - 8\rho_2 - 16\rho_{\sqrt{5}} - 4\rho_{2\sqrt{2}}, \\ Q_2 &= -1 - 4\rho_1 - 4\rho_2 + 8\rho_{\sqrt{5}} + 12\rho_{2\sqrt{2}} - 4\rho_3 + 8\rho_{\sqrt{13}} - 2\rho_4 - 4\rho_{\sqrt{17}} \\ &\quad - 4\rho_{2\sqrt{5}} - 4\rho_5 - \rho_{4\sqrt{2}}, \\ Q_3 &= -1 - 4\rho_1 - 4\rho_{\sqrt{2}} + 4\rho_3 + 8\rho_{\sqrt{10}} + 2\rho_4 + 4\rho_{\sqrt{17}} - 4\rho_{3\sqrt{2}} - 4\rho_5 - \rho_{4\sqrt{2}}. \end{aligned}$$

Thus, the dispersion of the angle difference in Equation (48), $D\{\theta_i - \theta_j\}$, when $i = 4$ and $j = 5$ is derived as follows:

$$\sigma_{\theta_4 - \theta_5}^2 = D\{\theta_4 - \theta_5\} = \frac{\sigma_n^2}{256\pi\sigma_s^2} \left[\frac{P_1}{g_{c_4}^2 + g_{r_4}^2} + \frac{P_1}{g_{c_5}^2 + g_{r_5}^2} - 2 \cdot \frac{g_{r_4}g_{r_5}P_2 + g_{c_4}g_{c_5}P_3}{(g_{c_4}^2 + g_{r_4}^2)(g_{c_5}^2 + g_{r_5}^2)} \right]. \quad (56)$$

Moreover, the dispersion of the angle difference in Equation (48), $D\{\theta_i - \theta_j\}$, when $i = 1$ and $j = 8$ is derived as follows:

$$\begin{aligned}\sigma_{\theta_1 - \theta_8}^2 &= D\{\theta_1 - \theta_8\} \\ &= \frac{\sigma_n^2}{256\pi\sigma_s^2} \left[\frac{Q_1}{g_{c_1}^2 + g_{r_1}^2} + \frac{Q_1}{g_{c_8}^2 + g_{r_8}^2} - 2Q_2 \cdot \frac{g_{r_1}g_{r_8} + g_{c_4}g_{c_5}}{(g_{c_1}^2 + g_{r_1}^2)(g_{c_8}^2 + g_{r_8}^2)} \right. \\ &\quad \left. + 2Q_3 \cdot \frac{g_{r_1}g_{c_8} + g_{c_1}g_{r_8}}{(g_{c_1}^2 + g_{r_1}^2)(g_{c_8}^2 + g_{r_8}^2)} \right].\end{aligned}\quad (57)$$

4.3. Derivation of SNRs

To effectively compare the performance of SGAD-based line detection with that of the SD-based line detection, a line with width w and aligned along the direction of row was used. Then, the line was assumed to have the following gradient values at pixels i and j , which were located one pixel to the left and right, respectively, of the center of the line and can be expressed as follows:

$$g_{c_i} = g, g_{r_i} = 0, g_{c_j} = -g, \text{ and } g_{r_j} = 0, \quad (58)$$

where

$$g = \mathbf{F}'_{\mathbf{b}*\mathbf{s}}|_{x=L-1}.$$

Then, at the center of the line, the standard deviation of the gradient angle difference in the column direction, $\sigma_{\theta_4 - \theta_5}$, was derived from Equations (56) and (58) as follows:

$$\sigma_{\theta_4 - \theta_5} = \frac{\sigma_n \sqrt{P_1 + P_3}}{8\sqrt{2\pi}\sigma_s g}. \quad (59)$$

Moreover, at the center of the line, the standard deviation of the gradient angle difference in the lower-right diagonal direction, $\sigma_{\theta_1 - \theta_8}$, was derived from Equations (57) and (58) as follows:

$$\sigma_{\theta_1 - \theta_8} = \frac{\sigma_n \sqrt{Q_1 + Q_2}}{8\sqrt{2\pi}\sigma_s g}. \quad (60)$$

For the line model assumed in the first paragraph of this section, the strength of the line signal based on the angle difference was π . Thus, the SNR of the angle difference in the column direction can be derived as follows:

$$\text{SNR}_{\theta_4 - \theta_5} = \frac{\pi}{\sigma_{\theta_4 - \theta_5}} = \frac{8\pi\sqrt{2\pi}\sigma_s}{\sigma_n \sqrt{P_1 + P_3}} g. \quad (61)$$

Moreover, the SNR of the angle difference in the lower-right diagonal direction can be derived as follows:

$$\text{SNR}_{\theta_1 - \theta_8} = \frac{\pi}{\sigma_{\theta_1 - \theta_8}} = \frac{8\pi\sqrt{2\pi}\sigma_s}{\sigma_n \sqrt{Q_1 + Q_2}} g \quad (62)$$

For the SGAD-based line detection, the performance was measured by combining both the gradient angle difference and SNR of the first derivative. Thus, the SNR combined for the line detection using the gradients at the fourth and fifth pixels in Figure 6 can be modeled as follows:

$$\text{SNR}_C(\text{line}_{\theta_4 - \theta_5}) = \sqrt{\text{SNR}(\text{line}_{\mathbf{D}_c}) \cdot \text{SNR}_{\theta_4 - \theta_5}}. \quad (63)$$

Moreover, the SNR combined for the line detection using the gradients at the first and eighth pixels in Figure 6 can be modeled as follows:

$$\text{SNR}_C(\text{line}_{\theta_1 - \theta_8}) = \sqrt{\text{SNR}(\text{line}_{\mathbf{D}_c}) \cdot \text{SNR}_{\theta_1 - \theta_8}}. \quad (64)$$

Considering the penalty for blurring, smoothing, and line width in Equation (41), the $\text{SNR}_C(\text{line}_{\theta_4-\theta_5})$ in Equation (63) can be derived into $\text{SNR}_{PS}(\text{line}_{\theta_4-\theta_5})$ as follows:

$$\text{SNR}_{PS}(\text{line}_{\theta_4-\theta_5}) = p(\sigma_b, \sigma_s, w) \cdot \text{SNR}_C(\text{line}_{\theta_4-\theta_5}). \quad (65)$$

Similarly, the $\text{SNR}_C(\text{line}_{\theta_1-\theta_8})$ in Equation (64) can be derived into $\text{SNR}_{PS}(\text{line}_{\theta_1-\theta_8})$ as follows:

$$\text{SNR}_{PS}(\text{line}_{\theta_1-\theta_8}) = p(\sigma_b, \sigma_s, w) \cdot \text{SNR}_C(\text{line}_{\theta_1-\theta_8}). \quad (66)$$

Then, the SNR_{PS} of the sum of the gradient angle differences for all of the four pairs in Figure 6 can be modeled as follows:

$$\begin{aligned} \text{SNR}_{PS}(\text{line}_{SGAD}) &= \text{SNR}_{PS}(\text{line}_{\theta_1-\theta_8}) + \text{SNR}_{PS}(\text{line}_{\theta_2-\theta_7}) \\ &+ \text{SNR}_{PS}(\text{line}_{\theta_3-\theta_6}) + \text{SNR}_{PS}(\text{line}_{\theta_4-\theta_5}). \end{aligned} \quad (67)$$

Because of the symmetry between the pair of gradients at 1 and 8, and the pair of gradients at 3 and 6 in Figure 6, $\text{SNR}_{PS}(\text{line}_{\theta_3-\theta_6})$ can be written as follows:

$$\text{SNR}_{PS}(\text{line}_{\theta_3-\theta_6}) = \text{SNR}_{PS}(\text{line}_{\theta_1-\theta_8}). \quad (68)$$

Because the gradients at 2 and 7 in Figure 6 are zero for the assumed line model, $\text{SNR}_{PS}(\text{line}_{\theta_2-\theta_7})$ can be written as follows:

$$\text{SNR}_{PS}(\text{line}_{\theta_2-\theta_7}) = 0. \quad (69)$$

Furthermore, considering the SNR_{PS} of the four directions, the SNR_{PS} of SGAD for detecting the assumed line model can be derived as follows:

$$\text{SNR}_{PS}(\text{line}_{SGAD}) = \text{SNR}_{PS}(\text{line}_{\theta_4-\theta_5}) + 2 \cdot \text{SNR}_{PS}(\text{line}_{\theta_1-\theta_8}). \quad (70)$$

The ratio of $\text{SNR}_{PS}(\text{line}_{SGAD})$ to $\text{SNR}_{PS}(\text{line}_{SD})$ can be derived from Equations (43) and (70) as follows:

$$\text{ratio}(\text{SGAD}, \text{SD}) = \frac{\text{SNR}_{PS}(\text{line}_{\theta_4-\theta_5}) + 2 \cdot \text{SNR}_{PS}(\text{line}_{\theta_1-\theta_8})}{\text{SNR}_{PS}(\text{line}_{SD})}. \quad (71)$$

Furthermore, Equation (71) can be reduced from Equations (43), (65) and (66) as follows:

$$\text{ratio}(\text{SGAD}, \text{SD}) = \frac{\sqrt{\text{SNR}_{\theta_4-\theta_5}} + 2\sqrt{\text{SNR}_{\theta_1-\theta_8}}}{\sqrt{\text{SNR}(\text{line}_{D_{cc}})}}. \quad (72)$$

Let

$$f = -\mathbf{F}_{b*s}''|_{x=L} \quad (73)$$

and

$$R = 9 - 16\rho\sqrt{2} + 6\rho_2 + \rho_2\sqrt{2}. \quad (74)$$

Then, $\text{ratio}(\text{SGAD}, \text{SD})$ can be derived from Equations (14), (39), (61), (62), (72), (73) and (74) as follows:

$$\text{ratio}(\text{SGAD}, \text{SD}) = [2\sqrt{2}\pi\frac{g}{f}]^{\frac{1}{2}} R^{\frac{1}{4}} [(P_1 + P_3)^{\frac{-1}{4}} + 2(Q_1 + Q_2)^{\frac{-1}{4}}]. \quad (75)$$

In Equation (75), $\frac{g}{f}$ is derived from Equations (13) and (14) as follows:

$$\frac{g}{f} = \frac{\sigma_b^2 + \sigma_s^2}{w} [e^{\frac{w-1}{2(\sigma_b^2 + \sigma_s^2)}} - e^{-\frac{w-1}{2(\sigma_b^2 + \sigma_s^2)}}]. \quad (76)$$

As shown in Equation (76), $\text{ratio}(\text{SGAD}, \text{SD})$ changes with the variations of σ_b , σ_s , and w , but does not change with the variation of σ_n .

To investigate the performance of the SGAD-based line detection, the graphical plots of SNR are used as shown below. The values of SNR used are generated under the same conditions as applied in Section 3.4.

Figure 8 shows the difference, $\text{SNR}_{\text{PS}}(\text{line}_{\text{SGAD}}) - \text{SNR}_{\text{PS}}(\text{line}_{\text{SD}})$. According to Figure 8, the SGAD-based line detection has higher SNR than the SD-based line detection under varying conditions of the line width and the smoothing factor. When the smoothing factor of 1.0 is applied, the advantage of the SGAD-based line detection becomes distinct for a line width less than 8 pixels.

Figure 8. $\text{SNR}_{\text{PS}}(\text{line}_{\text{SGAD}}) - \text{SNR}_{\text{PS}}(\text{line}_{\text{SD}})$. (a) 3-D view and (b) contours.

5. Experimental Results and Discussion

To validate the derived SNRs in natural images, the relationships of the SNRs with the completeness and correctness were also used in this section. For the experiments with natural images, the Lena image with the size of 512×512 pixels in gray color $[0, 255]$ was first selected to describe the procedure of the experiments for natural images. Figure 9 shows the Lena image with three annotated regions for further investigation.

Figure 9. Lena image.

Figure 10 shows the images resulting from each process for the annotated regions in Figure 9. To measure the SNRs, the completeness and correctness, ridge, and valley pixels were detected in the smoothed version of the original image with $\sigma_s = 1.0$ by the SGAD-based line detection and considered as the ground truth pixels of ridge and valley pixels. Then, the line width and contrast of each ground truth pixel were calculated as follows. At each ground truth pixel, line width was searched for two directions: the normal and its opposite directions. Regarding ridge pixels, for each direction, line width in one direction increased by one pixel at every extension of the line width before the intensity of the current pixel is greater than that of its previous pixel + a specified tolerance. In this

study, the tolerance was set to 0.05. Then, the direction among the two search directions, whose last intensity has less contrast with the ground truth line pixel, was selected and the line width in that direction was recorded as the line width of the ground truth line pixels. The contrast in the selected direction was recorded as the contrast of the ground truth line pixels. The method to find the line width and contrast of valley pixels was the same to that of the ridge pixels. SNRs of the SD-based and SGAD-based line detections were calculated using the given line width, contrast, and noise strength. As shown in Figure 10, SNRs of the SGAD-based line detection were much greater than those of the SD-based line detection. Accordingly, as shown in the last two columns in Figure 10, the results by SGAD show less noisy and more accurate line detection results than those by SD when they are compared with the ground truths shown in the third column in Figure 10.

Figure 10. Line detection results for the Lena subset images. First to last columns show the original images, smoothed images, ground truth line pixels, line width w , contrast k , $\log_{10}(\text{SNR})$ of the SD- and SGAD-based line detections when $\sigma_n = 8.0$, and line pixels detected by the SD- and SGAD-based line detections under $\sigma_n = 8.0$. First, third, and fifth rows show the results for ridges and second, fourth, and sixth row show the results for valleys.

The completeness and correctness of line detection results were measured as follows. First, ground truth images were generated using the line detection results by applying the SGAD-based line detection to the smoothed version of the original image with $\sigma_s = 1.0$, as shown in the third column in Figure 10. Then, a set of noisy images were generated by adding varying strengths of Gaussian random noises ranging from $\sigma_n = 4.0$ to 20.0 with the interval of 4.0 , respectively, to the original images. Next, all the noisy images were denoised by applying a smoothing convolution with $\sigma_s = 1.0$, and the SD-based and SGAD-based line detections were applied to the noisy images. When the ground truth image as reference is denoted by $\mathbf{R}$ and a line pixel image resulting from the application of a line detection method to a noisy image are denoted by $\mathbf{T}$, then the image $\mathbf{C}$ which contains line pixels correctly detected by a line detector can be written as

$$\mathbf{C} = \mathbf{R} \wedge \mathbf{T}, \quad (77)$$

where $\wedge$ denotes the logical AND operator.

To find the incorrectly detected line pixels within a specified distance from the reference pixels, the reference line pixel image was dilated with a certain radius r as

$$\mathbf{R}_d = \mathbf{R} \oplus \mathbf{S}_r, \quad (78)$$

where $\oplus$ denotes the dilation operator and $\mathbf{S}_r$ the structural element with radius r . In this study, r was set to 3 pixels because it equals $3\sigma_s$ when $\sigma_s = 1.0$.

Then, the image containing incorrectly detected pixels $\mathbf{Q}$ was generated as

$$\mathbf{Q} = \mathbf{R}_d \wedge (\mathbf{T} \wedge \neg \mathbf{C}), \quad (79)$$

where $\neg$ denotes the logical NOT operator.

Next, the number of incorrectly detected pixels were counted at each ground truth line pixel within a certain distance d_t in its line normal and opposite to the directions using the images $\mathbf{R}$ and $\mathbf{Q}$ and recorded into an image $\mathbf{V}$. In this study, d_t was set to 3 pixels, which is the same as the value of r used for dilation of a reference image.

Subsequently, the SNRs at ground truth line pixels were summarized into a histogram with a certain bin size. The bin size was determined by dividing the maximum of SNRs in each SNR image by a specified number n . In this study, the number n was set to 100. Then, new bins were generated in a descending order so that each new bin has about a certain percentage p_b of the total ground truth line pixels. The value of p_b was set to 10 percent in this study. Figure 11 shows the results for the SNRs of the ridge pixels in the SD-based line detection, when $\sigma_n = 8.0$.

Figure 11. Example of binning results. The blue line shows the histogram under a given bin size and the red lines the boundaries of the new bins.

Next, pixels having SNRs within the boundaries of each SNR bin were selected and the mean of their SNRs was recorded. Moreover, among the selected pixels, the total number of correctly detected pixels N_c was counted using the image $\mathbf{C}$ derived from Equation (77). Then, the completeness corresponding to the mean SNR was measured by dividing the number of correctly detected line pixels by the total number of the ground truth line pixels N_{gt} as shown here:

$$\text{Completeness} = \frac{N_c}{N_{gt}} \times 100. \quad (80)$$

Additionally, among the selected pixels, the total number of incorrectly detected pixels N_{ic} was counted using the image $\mathbf{Q}$ derived from Equation (79). Then, the correctness corresponding to the mean SNR was measured as

$$\text{Correctness} = \frac{N_c}{N_{gt} + N_{ic}} \times 100. \quad (81)$$

Figure 12 shows the observed relationships of SNRs with completeness and correctness for the Lena image under varying noise strengths. As shown in the figure, the trend of their relationships is manifest under varying noise strengths. More so, the SGAD-based line detection showed higher SNRs than the SD-based line detection and higher completeness and correctness under all the tested noise strengths.

Figure 12. Completeness and correctness against SNR for the Lena image under varying σ_n . Top row shows completeness and bottom row correctness.

Figure 13 shows all the plots together which are shown in Figure 12. As shown in Figure 13, the validity of the SNRs derived for the SD-based and the SGAD-based line detections was proven by the overall strong and consistent relationships of SNRs with completeness and correctness under varying noise strength.

Figure 13. Completeness and correctness against SNR for the Lena image. The second and fourth plots show the zoom-in versions of the first and third plots with SNR range [0, 10], respectively.

The experiments were also applied to other eight natural images as shown in Figure 14. Before implementation of line detections, the natural images were converted into grayscale images using the MATLAB's *rgb2gray* function. As shown in Figure 15, experiments with the images show that there exist distinct and strong relationships of SNRs with completeness and correctness. The pattern of variations of completeness against SNR was observed similar to that of the correctness. Although the content differs depending on the test images, it was found that the line detection completeness and correctness against SNRs showed similar patterns throughout the test images. This observation indicates the consistency of SNR measures with respect to completeness and correctness under various line conditions. However, as shown in Figure 15, the SD-based line detection was observed to produce low completeness and correctness as compared to its SNR when $\text{SNR} > 3.0$. This is caused by frequent bifurcations of the SD-based line detection. When the line width was not relatively large, but the contrast was relatively large, the calculated value of SNR was relatively large. Meanwhile, in the experiments, the shape of many line profiles with such conditions deviated from that of the ideal line profile and the SD-based line detection could not overcome this problem and produced low completeness

and correctness. However, the SGAD-based line detection was observed to overcome this problem upto some extent and produce high completeness and correctness, when the value of SNR was high.

Figure 14. Natural images. (Top) From left to right: Baboon, Barbara, Boats, Cablecar; (Bottom) From left to right: Flowers, Goldhill, Monarch, and Yacht.

Figure 15. Cont.

Figure 15. Completeness and correctness against SNR for the natural images. The second and fourth columns show the zoom-in versions of the plots in the first and third columns with SNR range [0, 10], respectively.

6. Conclusions

In this study, the performances of line detectors were evaluated by the analytical quantifications of their SNRs and completeness and correctness. The correlations arising among pixels when a Gaussian smoothing filter was applied were first identified. Then, the amount of noise remaining in the derived values, such as gradients, SD, and SGAD, was derived based on the error propagation. Furthermore, the SNRs of line detectors were analytically derived based on the derived signal and noise strengths. In addition, the penalty function was proposed to consider the influence of the blur, smoothing, and the line-width on line detectors. Verification of the validity of the derived SNRs based on investigation of the relationships of the SNRs with completeness and correctness indicates that the derivations of the SNRs of line detectors proposed in this study were effective in quantifying their performance. The validation test of SNRs was performed using nine color images and will be extended to more image sets for further test completion in the future.

Regarding feature extraction, it was observed that the edge features could be accurately extracted in the vector form using the methods proposed in [18,41]. In contrast, the line features were considered to be accurately extracted in the vector form using the SGAD method followed by the subpixel line localization and location-linking methods as described in [5].

Therefore, a set of methods for line detection, non-maxima suppression, subpixel line localization, and linking can produce high-quality ridge and valley features for various applications. Moreover, the error propagation scheme used in this study to derive the relevant theoretical SNRs can be used to develop high-performance operators for extracting features.

Funding: This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(2016R1D1A1B02011625).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Haralick, R.M. Ridges and valleys on digital images. *Comput. Vis. Graph. Image Process.* **1983**, *22*, 28–38. [\[CrossRef\]](#)
2. Gauch, J.M.; Pizer, S.M. Multiresolution analysis of ridges and valleys in grey-scale images. *IEEE Trans. Pattern Anal. Mach. Intell.* **1993**, *15*, 635–646. [\[CrossRef\]](#)
3. Serrat, J.; Lopez, A.; Lloret, D. On ridges and valleys. In Proceedings of the 15th International Conference on Pattern Recognition, Barcelona, Spain, 3–7 September 2000.
4. Rashe, C. Rapid contour detection for image classification. *IET Image Process.* **2018**, *12*, 532–538. [\[CrossRef\]](#)
5. Seo, S. Subpixel line localization with normalized sums of gradients and location linking with straightness and omnidirectionality. *IEEE Access* **2019**, *7*, 180155–180167. [\[CrossRef\]](#)
6. Seo, S. Line-detection based on the sum of gradient angle differences. *Appl. Sci.* **2020**, *10*, 254. [\[CrossRef\]](#)
7. Jung, C.; Kelber, C. Lane following and lane departure using a linear-parabolic model. *Image Vis. Comput.* **2005**, *23*, 1192–1202. [\[CrossRef\]](#)
8. An, X.; Shang, E.; Song, J.; Li, J.; He, H. Real-time lane departure warning system based on a single FPGA. *EURASIP J. Image Video Process.* **2013**, *38*, 1–18. [\[CrossRef\]](#)
9. Wu, P.; Chang, C.; Lin, C. Lane mark extraction for automobiles under complex conditions. *Pattern Recognit.* **2014**, *47*, 2756–2767. [\[CrossRef\]](#)
10. Son, J.; Yoo, H.; Kim, S.; Sohn, K. Real-time illumination invariant lane detection for lane departure warning system. *Expert Syst. Appl.* **2015**, *42*, 1816–1824. [\[CrossRef\]](#)
11. Karasulu, B. Automatic extraction of retinal blood vessels: A software implementation. *Eur. Sci. J.* **2012**, *8*, 47–57.
12. Majumdar, J.; Kundu, D.; Tewary, S.; Ghosh, S.; Chakraborty, S.; Gupta, S. An automated graphical user interface based system for the extraction of retinal blood vessels using Kirsch's template. *Int. J. Adv. Comput. Sci. Appl.* **2015**, *6*, 86–93. [\[CrossRef\]](#)
13. Wang, W.; Yang, N.; Zhang, Y.; Wang, F.; Cao, T.; Eklund, P. A review of road extraction from remote sensing images. *J. Traffic Transp. Eng.* **2016**, *3*, 271–282. [\[CrossRef\]](#)
14. Roberts, L. Machine perception of three-dimensional solids. In *Optical and Electro-Optical Information Processing*; Tippet, J., Berkowitz, D., Clapp, L.C., Koester, C.J., Alexander Vanderburgh, J., Eds.; MIT Press: Cambridge, MA, USA, 1965; pp. 159–197.
15. Pingle, K. Visual perception by computer. In *Automatic Interpretation and Classification of Images*; Grasselli, A., Ed.; Academic Press: New York, NY, USA, 1969; pp. 277–284.
16. Prewitt, J. Object enhancement and extraction. In *Picture Processing and Psychophysics*; Rosenfeld, A., Lipkin, B., Eds.; Academic Press: New York, NY, USA, 1970; pp. 75–149.
17. Marr, D.; Hildreth, E. Theory of edge detection. *Proc. R. Soc. Lond.* **1980**, *207*, 187–217. [\[PubMed\]](#)
18. Canny, J. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1986**, *PAMI-8*, 679–698. [\[CrossRef\]](#)
19. Steger, C. An unbiased detector of curvilinear structures. *IEEE Trans. Pattern Anal. Mach. Intell.* **1998**, *20*, 113–125. [\[CrossRef\]](#)
20. Zwiggelaar, R.; Astely, S.M.; Boggis, C.R.M.; Taylor, C.J. Linear structures in mammographic images: Detection and classification. *IEEE Trans. Med Imaging* **2004**, *23*, 1077–1086. [\[CrossRef\]](#)
21. Lopez, A.M.; Lumbreras, F.; Serrat, J.; Villanueva, J.J. Evaluation of methods for ridge and valley detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1999**, *21*, 327–335. [\[CrossRef\]](#)
22. Paton, K. Picture description using Legendre polynomials. *Comput. Graph. Image Process.* **1975**, *4*, 40–54. [\[CrossRef\]](#)
23. Ghosal, S.; Mehrotra, R. Detection of composite edges. *IEEE Trans. Image Process.* **1994**, *3*, 14–25. [\[CrossRef\]](#) [\[PubMed\]](#)
24. Lindberg, T. Edge detection and ridge detection with automatic scale selection. *Int. J. Comput. Vis.* **1998**, *30*, 117–154. [\[CrossRef\]](#)
25. Laptev, I.; Mayer, H.; Lindeberg, T.; Eckstein, W.; Steger, C.; Baumgartner, A. Automatic extraction of roads from aerial images based on scale space and snakes. *Mach. Vis. Appl.* **2000**, *12*, 23–31. [\[CrossRef\]](#)
26. Jeon, B.K.; Jang, J.H.; Hong, K.S. Road detection in spaceborne SAR images using a genetic algorithm. *IEEE Trans. Geosci. Remote Sens.* **2002**, *40*, 22–29. [\[CrossRef\]](#)
27. Lopez-Molina, C.; de Ulzurrun, G.V.D.; Baetens, J.; den Bulcke, J.V.; Baets, B.D. Unsupervised ridge detection using second order anisotropic Gaussian kernels. *Signal Process.* **2015**, *116*, 55–67. [\[CrossRef\]](#)
28. Cornelis, B.; Ruzic, T.; Gezels, E.; Doomes, A.; Pizurica, A.; Platasa, L.; Cornelis, J.; Martens, M.; Mey, M.D.; Daubechies, I. Crack detection and inpainting for virtual restoration of paintings: The case of the Ghent Altarpiece. *Signal Process.* **2013**, *93*, 605–619. [\[CrossRef\]](#)
29. Yan, C.; Shao, B.; Zhao, H.; Ning, R.; Zhang, Y. 3D room layout estimation from a single RGB image. *IEEE Trans. Multimed.* **2020**, *22*, 3014–3024. [\[CrossRef\]](#)

30. Legendijk, R.L.; Tekalp, A.M.; Biemond, J. Maximum likelihood image and blur identification: A unifying approach. *Opt. Eng.* **1990**, *29*, 422–435.
31. Reeves, S.J.; Mersereau, R.M. Blur identification by the method of generalized cross-validation. *IEEE Trans. Image Process.* **1992**, *1*, 301–311. [[CrossRef](#)]
32. Savakis, A.E.; Trussell, H.J. Blur identification by residual spectral matching. *IEEE Trans. Image Process.* **1993**, *2*, 141–151. [[CrossRef](#)]
33. Kundur, D.; Hatzinakos, D. Blind image deconvolution. *IEEE Signal Process. Mag.* **1996**, *13*, 43–64. [[CrossRef](#)]
34. Yitzhaky, Y.; Kopeika, N.S. Identification of blur parameters from motion blurred images. *Graph. Model. Image Process.* **1997**, *59*, 310–320. [[CrossRef](#)]
35. Chen, L.; Yap, K.H. Efficient discrete spatial techniques for blur support identification in blind image deconvolution. *IEEE Trans. Signal Process.* **2006**, *54*, 1557–1562. [[CrossRef](#)]
36. Wu, S.; Lin, W.; Xie, S.; Lu, Z.; Ong, E.P.; Yao, S. Blind blur assessment for vision-based applications. *J. Vis. Commun. Image Represent.* **2009**, *20*, 231–241. [[CrossRef](#)]
37. Hu, W.; Xue, J.; Zheng, N. PSF estimation via gradient domain correlation. *IEEE Trans. Image Process.* **2012**, *21*, 386–392. [[CrossRef](#)] [[PubMed](#)]
38. Liu, S.; Wang, H.; Wang, J.; Cho, S.; Pan, C. Automatic blur-kernel-size estimation for motion deblurring. *Vis. Comput.* **2015**, *31*, 733–746. [[CrossRef](#)]
39. Seo, S. Edge modeling by two blur parameters in varying contrast. *IEEE Trans. Image Process.* **2018**, *27*, 2701–2714. [[CrossRef](#)] [[PubMed](#)]
40. Bromiley, P.A. *Products and Convolutions of Gaussian Probability Density Functions*; TINA: Manchester, UK, 2018.
41. Seo, S. Subpixel edge localization based on adaptive weighting of gradients. *IEEE Trans. Image Process.* **2018**, *27*, 5501–5513. [[CrossRef](#)]