IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 23, 2019, accepted December 9, 2019, date of publication December 13, 2019,
date of current version December 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2959320

Subpixel Line Localization With Normalized Sums of Gradients and Location Linking With Straightness and Omni-Directionality

SUYOUNG SEO[1], **(Member, IEEE)**
Department of Civil Engineering, Kyungpook National University, Daegu 41566, South Korea
(e-mail: syseo@knu.ac.kr)

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea, Ministry of Education, under Grant 2016R1D1A1B02011625.

ABSTRACT This paper presents a method to localize line locations with subpixel accuracy and a method to link the locations based on a linking distance. This paper first proposes a subpixel line localization method based on normalized sums of gradients (NSG) calculated by dividing pixel sum of gradients by the sum of gradient lengths within the pixel neighborhood. The proposed NSG method is compared with current state-of-the art based on a Taylor series approximation of intensity surface and the normal vector derived from the Hessian matrix. Comparative experiments for subpixel line localization methods were performed with simulated and natural images and confirmed the proposed subpixel localization method provided superior accuracy and faster localization under most combinations of varying line width and noise strengths than the state-of-the art localization method. The proposed linking method was also designed to have more straightness and omni-directionality than a current state-of-the art method. Experimental comparison of linking methods confirmed the proposed linking method provided superior linking quality than current state-of-the art.

INDEX TERMS Location linking, normalized sum, subpixel line localization.

I. INTRODUCTION

In image processing, a line is a geometrically elongated feature that has brighter or darker intensity than neighboring pixels. Some literature also refers to a line as a ridge or valley when the line intensity is greater or less than neighboring pixels, respectively [1]–[3]. Line features are one of the most important, frequently observed features for indoor, building facade, satellite, medical, road, etc. imagery, and extensively used by perceptual systems to analyze and interpret scenes in an image. Therefore, line extraction is important for image processing and computer vision fields.

The demand for efficient line feature extraction from images has grown rapidly due to advancements in various applications, including lane extraction for autonomous driving [4]–[7], blood vessel extraction for medical images [8], [9], road extraction from satellite images [10].

The associate editor coordinating the review of this manuscript and approving it for publication was Gangyi Jiang.

To extract high quality localized linear features, one must first find subpixel locations of linear features. However, this topic has received scant research attention, aside from Steger [11], who proposed a subpixel line localization method based on Taylor series approximation (TA) of surface intensity and the normal vector derived from the Hessian matrix. However, this paper shows that the proposed method suffers from unstable linear feature localization for weak line signals and high noise level. Therefore, this paper proposes a subpixel line localization method based on normalized sums of gradients (NSG) within each neighborhood of a detected line pixel to resolve this issue.

To obtain line features in vector form, one must link localized feature locations by grouping the locations into a sequence. Although the process is critical to extract linear features in a semantic form, previous studies have largely considered linking locations at the pixel level [12], [13], with few at the subpixel level [11]. Therefore, this paper

also tackles this issue by proposing a subpixel level linking method.

This remainder of this paper is organized as follows. Section II describes related studies for subpixel line localization and linking. Section III proposes subpixel line localization and linking methods. Sections IV and V present line localization experiments for simulation images and natural images, respectively. Section VI presents experiments for line linking. Section VII concludes the paper.

II. RELATED WORK

A. SUBPIXEL LINE LOCALIZATION

Although many studies have considered subpixel edge localization [14]–[18], few approaches have considered subpixel line localization. Steger [11] proposed a normal vector based method that localized line points to subpixel accuracy by finding the location where the first derivative of a TA of a line profile was zero in the direction corresponding to the maximum absolute eigenvalue of the Hessian matrix. However, this paper shows that the proposed approach is highly sensitive to weak line signal and high noise level. Therefore, this paper proposes an alternate approach based on normalized sums of gradients (NSG), which has less sensitivity to weak line signal and high noise level than the TA method.

B. LINKING

Line linking creates a sequence of coordinates from a set of isolated pixels or positions by joining them sequentially, and is similar to edge linking since both methods can be performed on either detected pixels or localized positions. Therefore both edge and line linking approaches were reviewed.

Basak *et al.* [12] proposed a method to link edge or line pixels based on updating the network model. However, the approach employed a multi-step procedure to find the pixel links, with consequentially high computational cost. Farag and Delp [19] proposed a graph search-based linking method, where pixel links were established based on optimal paths between ending pixels, determined by estimated maximum likelihood. Lu and Chen [20] proposed a method to link edge pixels with similar intensity pixels with high contrast to eight neighboring pixels based on ant-colony optimization. However, this method required carefully tuning several parameters, which could be arduous.

Steger [11] proposed a method to link line pixels based on a combination of distance and angle differences between subpixel localized locations. However, a critical limitation of the approach was that it only considered three candidate pixels closely located about the current line direction, and hence may not link the line pixels successfully, particularly for highly curved line features. Wang and Zhang [21] proposed an edge linking method based on geodesic distance, calculated using Euclidean distance between pixels, pixel intensity differences, and directional differences between past movement and the line joining the current edge pixel to candidate edge pixels.

Lin *et al.* [22] proposed an edge linking method for lane extraction that connected edge pixels by checking three pixels nearest to past movement direction within a 3×3 neighborhood for each pixel. Flores *et al.* [23] proposed a linking method that searched for a connecting direction among five directions within 90° from past movement direction. Akinlar and Topal [24] proposed selecting pixels as anchor points when their gradient magnitude exceeded gradient magnitudes of neighboring pixels by a given threshold, and checked three pixels in the immediate neighborhood of the current linking direction. Akinlar and Chome [13], [25] smoothed a traditional Canny edge detector's binary edge map (BEM), and then traced edge pixels in the smoothed BEM. Akinlar [26] proposed an edge linking method that first calculated edge directions based on the maximum number in the soft contour map containing the number of accumulated detections at each pixel for a series of images generated by varying the degree of smoothness; detected anchor pixels based on directional coherence in prespecified directions in the 3×3 neighborhood for each pixel; and then connected the anchor pixels by linking edge pixels close to the direction determined from past movement directions.

Gap closing connects edge or line segment endpoints [27]–[30]. Zhu *et al.* [27] proposed an edge gap closing method based on a directional potential function that modeled linking potentials by classifying edge pixels within a 3×3 neighborhood. Ghita [28] proposed a method to link edge segments' endpoints based on cost calculated as the distance between an endpoint and candidate edge pixels, and on the condition that the candidate edge pixel was an endpoint. Shih and Cheng [29] proposed a similar edge gap closing method based on adaptive mathematical morphology, using adaptive elliptical structuring elements for dilation and erosion, depending on local edge segment shapes. Sappa and Vintimilla [30] proposed a gap closing method using graph based linking method with Euclidean distance between edge pixels in 3D space. Pixel coordinates comprised two location and one intensity values. Although the current paper does not consider gap closing due to space considerations, gap closing should be performed after edge or line linking when gaps affect feature extraction significantly to obtain complete linear features.

A common problem for most previous, even state-of-the-art, linking methods is low precision due to the linking process being performed at the pixel level. This leads to nonsmooth (jagged) line features since line point locations are limited to pixel precision. Steger [11] proposed a method to address this problem by using subpixel localized line locations to perform linking. However, the method only considered three neighboring pixels' subpixel localized positions as candidate positions, which may not connect well with line positions when line features are highly curved. In contrast, the linking process proposed in the current paper overcomes this limitation by considering all localized positions as candidate line positions based on a proposed linking distance measure.

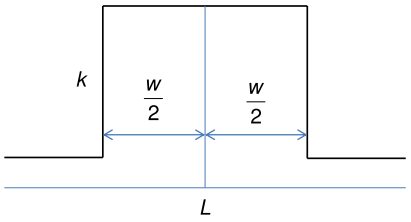


FIGURE 1. One-dimensional line profile model used in the current study.

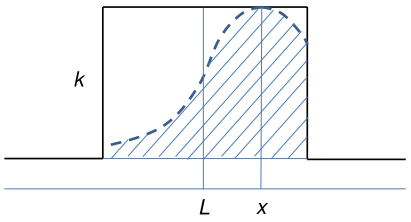


FIGURE 2. Gaussian blur for the one-dimensional line model.

III. PROPOSED SUBPIXEL LINE LOCALIZATION AND LINKING METHODS

A. LINE MODEL

Image processing must always deal with some amount of blur and noise due to limitations imposed by the camera lens system. Many previous studies have adopted the approach that any real image, $\mathbf{I}$, can be modeled as a base signal, $\mathbf{F}$ convolved with blur, $\mathbf{b}$, and noise, $\mathbf{n}$ [31]–[39],

$$\mathbf{I} = \mathbf{F} * \mathbf{b} + \mathbf{n}, \tag{1}$$

where $*$ indicates convolution.

Seo [40] proposed a more comprehensive blur model with blurring patterns in edge neighborhoods. However, this paper employed the image formation model in (1) in the following derivation because of its simplicity.

This study modeled line profile by width, w, and contrast, k, as shown in Fig. 1. Although a more general line model has been proposed previously [11], modelling a line with different contrasts on either side, the derivation here uses the simpler model in Fig. 1 because it is much simpler for subsequent tests and most line features have similar contrasts on either side.

From Fig. 1, the one-dimensional line signal is modeled as

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

where L is the coordinate of the line center.

A one-dimensional Gaussian blur is introduced to consider blur effects in the image formation,

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

where σ_b is the blurring factor.

Fig. 2 shows an exaggerated version of the blur function, $k \cdot \exp(-\frac{(t-x)^2}{2\sigma_b^2})$, centered at an arbitrary location, x. Thus, intensity at x after blurring is

$$\begin{aligned} \mathbf{F_b}(x) = (\mathbf{F} * \mathbf{b})(x) &= \int_{x-\frac{w}{2}-L}^{x+\frac{w}{2}-L} \frac{k}{\sigma_b\sqrt{2\pi}} e^{-\frac{t^2}{2\sigma_b^2}} dt, \\ &= \frac{k}{2}\Big[\mathrm{erf}\Big(\frac{x + \frac{w}{2} - L}{\sigma_b\sqrt{2}}\Big) - \mathrm{erf}\Big(\frac{x - \frac{w}{2} - L}{\sigma_b\sqrt{2}}\Big)\Big]. \end{aligned} \tag{4}$$

B. SUBPIXEL LINE LOCALIZATION BASED ON TAYLOR'S EXPANSION APPROXIMATION

This section describes the previously proposed TA approach to subpixel line localization [11]. The TA approach sets up a Hessian matrix for each line pixel using second derivatives, $\mathbf{g_{cc}}$, $\mathbf{g_{rc}}$, and $\mathbf{g_{rr}}$ at the center of the line pixel, (r, c),

$$H = \begin{bmatrix} \mathbf{g_{cc}}(r, c) & \mathbf{g_{rc}}(r, c) \\ \mathbf{g_{rc}}(r, c) & \mathbf{g_{rr}}(r, c) \end{bmatrix}. \tag{5}$$

The approach then found the eigenvector corresponding to the maximum absolute eigenvalue of the Hessian matrix, which indicated the direction in which the intensity surface around the line pixel is maximally curved. The normalized curvature direction vector of the eigenvector is denoted as $[n_r, n_c]$. Then the translation, t along the normalized maximal curvature direction vector from the center of the current line pixel to the subpixel line location is modeled by a Taylor approximation for a two-variable function, and calculated using the first and second derivatives at the center of the line pixel and the normalized maximal curvature direction vector,

$$t = -\frac{\mathbf{g_c}(r, c)n_c + \mathbf{g_r}(r, c)n_r}{\mathbf{g_{cc}}(r, c)n_c^2 + \mathbf{g_{rc}}(r, c)n_c n_r + \mathbf{g_{rr}}(r, c)n_r^2}, \tag{6}$$

where $\mathbf{g_c}$ and $\mathbf{g_r}$ are first derivatives or gradients in the column and row directions, respectively.

Translations from the center line pixel to subpixel line location can be expressed as

$$\begin{aligned} \delta c_{\mathrm{TA}} &= t n_c \\ \delta r_{\mathrm{TA}} &= t n_r. \end{aligned} \tag{7}$$

Finally, subpixel line coordinates are

$$\begin{aligned} c_{\mathrm{TA}} &= c + \delta c_{\mathrm{TA}} \\ r_{\mathrm{TA}} &= r + \delta r_{\mathrm{TA}}. \end{aligned} \tag{8}$$

C. PROPOSED SUBPIXEL LINE LOCALIZATION METHOD

This section proposes a subpixel line localization method based on NSG. One of essential elements is to find the sum of gradient magnitudes corresponding to a unit pixel length for normalization.

The method first calculates the sums of gradients within a local window with size $(2K+1)\times(2K+1)$ pixels for gradients in both row and column directions,

$$\begin{aligned} S_{g_r}(r, c) &= \sum_{m=-K}^{K} \sum_{n=-K}^{K} \mathbf{g_r}(r + m, c + n) \\ S_{g_c}(r, c) &= \sum_{m=-K}^{K} \sum_{n=-K}^{K} \mathbf{g_c}(r + m, c + n), \end{aligned} \tag{9}$$

respectively, where K is a positive integer that controls the size of a local window, and set to $K = 1$ in this paper. Units for $S_{g_r}(r, c)$ and $S_{g_c}(r, c)$ in (9) are intensity per $(2K + 1)^2$ pixels.

The method then calculates the sum of gradient magnitudes within the local window,

$$S_{g_{mag}}(r, c) = \sum_{m=-K}^{K} \sum_{n=-K}^{K} \left[\mathbf{g_r}^2(r + m, c + n) + \mathbf{g_c}^2(r + m, c + n)\right]^{\frac{1}{2}}, \quad (10)$$

where units for $S_{g_{mag}}(r, c)$ in (10) is intensity per $(2K + 1)^2$ pixels per pixel, because the gradient magnitude sum is assumed to be measured for unit pixel length.

Subsequently, the sums of gradients in (9) are normalized using the sum of gradient magnitude in (10). However, a metric is calculated for normalization, which can be used to determine the state of a line pixel, i.e., ridge or valley. Two kernels are defined to achieve this end,

$$\mathbf{N_c} = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad (11)$$

and

$$\mathbf{N_r} = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}, \quad (12)$$

and the following metric is calculated to determine the line pixel state at (r, c),

$$P(r, c) = -\left[(\mathbf{g_r} * \mathbf{N_r}) + (\mathbf{g_c} * \mathbf{N_c})\right](r, c). \quad (13)$$

A line pixel is classified as ridge when $P(r, c) > 0$, and otherwise as a valley. Although this classification into ridge or valley is performed as part of line pixel detection, it was included in this section to assist understanding the following derivations.

The normalization term in (10) is signed based on the line pixel state in (13),

$$D(r, c) = \begin{cases} S_{g_{mag}}(r, c), & \text{if } P(r, c) > 0; \\ -S_{g_{mag}}(r, c), & \text{otherwise.} \end{cases} \quad (14)$$

Translations relative to the line pixel center are calculated by normalizing the sums of gradients from (9),

$$\delta c_{\mathrm{NSG}} = \frac{S_{g_c}(r, c)}{D(r, c)}$$
$$\delta r_{\mathrm{NSG}} = \frac{S_{g_r}(r, c)}{D(r, c)}, \quad (15)$$

where units of δc_{NSG} and δr_{NSG} in (15) are pixels (from the units of $S_{g_c}(r, c)$, $S_{g_r}(r, c)$, and $D(r, c)$).

Finally, subpixel line coordinates are

$$c_{\mathrm{NSG}} = c + \delta c_{\mathrm{NSG}}$$
$$r_{\mathrm{NSG}} = r + \delta r_{\mathrm{NSG}}. \quad (16)$$

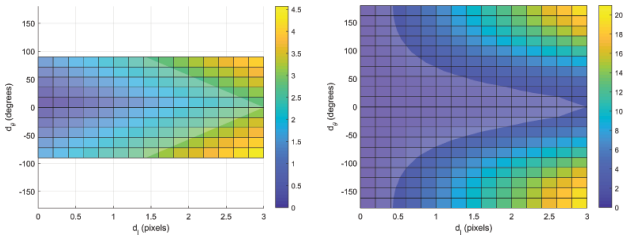


FIGURE 3. Line point linking distance by (a) Steger's and (b) proposed method. Colors represent calculated linking distance and half-transparent regions represent linking distances ≤ 3.0.

D. PROPOSED LINE LINKING METHOD

Linking generates a set of connected line points by joining line locations, which requires measuring distances from the current line to a set of candidate locations, called linking distances. Steger [11] proposed a simple addition of geometric and angular distances to calculate linking distance, whereas this paper proposes linking distance calculated as the product of geometric and angular distances.

The geometric distance between location, (r_c, c_c), and each forward candidate location, (r_f, c_f), to be linked can be expressed as

$$d_l = \left[(c_f - c_c)^2 + (r_f - r_c)^2\right]^{\frac{1}{2}}, \quad (17)$$

and the corresponding directional angle as

$$\theta_f = \tan^{-1}\left(\frac{r_f - r_c}{c_f - c_c}\right). \quad (18)$$

Hence the difference between the current line direction, θ_c, and the candidate line direction, θ_f is

$$d_\theta = \theta_f - \theta_c, \quad (19)$$

and angular distance is

$$d_a = \lambda(1 - \cos d_\theta), \quad (20)$$

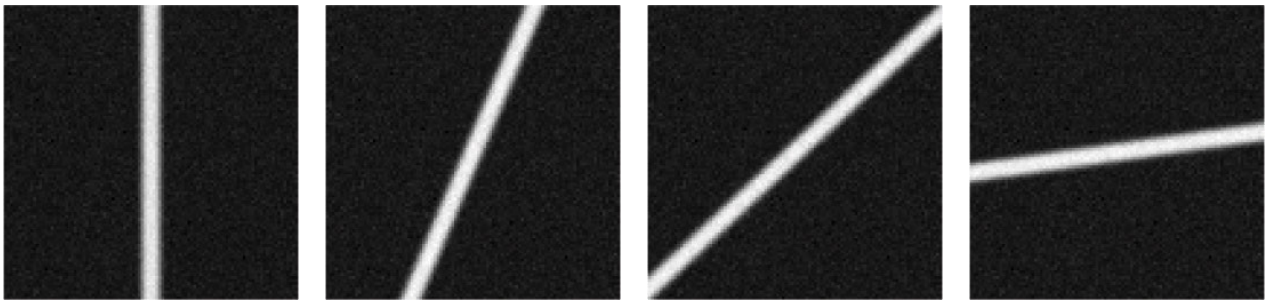
where λ is a factor to control angular distance influence on linking distance; $\lambda = 3$ in this study.

Finally, linking distance is

$$d = d_l(1 + d_a). \quad (21)$$

Given a set of candidate locations, the location with minimum linking distance is selected and connected to the current location, the current proceeding direction is updated to the selected forward direction, and the current location is updated to the selected location.

Figure 3 shows linking distances used by Steger's [11] and the proposed method for geometric and angle distances (d_l and d_θ, respectively). When d_l is relatively large, e.g. $d_l > 0.8$ pixels, the proposed linking method considers straightness more significant than geometric distance compared with Steger's method. Although Steger's method does not consider locations with $d_\theta > \frac{\pi}{2}$, the proposed method considers those locations as candidates when d_l is relatively



(a) (b) (c) (d)

FIGURE 4. Simulated 101 × 101 pixel images, generated with constant contrast ($k = 1$), blurring factor ($\sigma_b = 1$), line width ($w = 6$), and noise strength ($\sigma_n = 8/255$), for (a) $\theta = 0°$, (b) $\theta = 23°$, (c) $\theta = 47°$, and (d) $\theta = 82°$.

small, e.g. $d_l < 0.8$ pixels. In principle, this could allow locations almost opposite to the current direction to be linked to the current location. However, this rarely occurs in practical cases, because the distance between detected and subpixel localized line features tends to be greater than 0.4 pixels in most cases (experimental data not shown due to space limitations). However, a culling process is also applied prior to the linking process to remove locations with weak line signals located within a tolerance, e.g. 0.2 pixels, from locations with stronger line signals to completely remove the concern regarding linking opposite direction locations.

IV. SUBPIXEL LINE LOCALIZATION EXPERIMENTS WITH SIMULATED IMAGES

A. GENERATION OF SIMULATIVE IMAGES

To test subpixel localization performance for the TA and proposed NSG method, a series of images were generated containing line features with fixed blurring factor, $\sigma_b = 1$, varying line width, w, line normal angle, θ, and noise, σ_n. For the given image center coordinates, (r_0, c_0), and line normal direction, θ, normal distance from the origin to the center of the line, ρ was calculated as

$$\rho = c_0 \cos\theta + r_0 \sin\theta, \tag{22}$$

and two distances were calculated for each pixel location, (r, c),

$$D_1(r,c) = \frac{c\cos\theta + r\sin\theta - \rho + \frac{w}{2}}{\sigma_b\sqrt{2}}$$
$$D_2(r,c) = \frac{c\cos\theta + r\sin\theta - \rho - \frac{w}{2}}{\sigma_b\sqrt{2}}. \tag{23}$$

For the given contrast, k, described in (2), intensity at each pixel was then calculated as

$$\mathbf{F_b}(r,c) = \frac{k}{2}\big[\mathrm{erf}(D_1(r,c)) - \mathrm{erf}(D_2(r,c))\big]. \tag{24}$$

and an image was generated by adding a noise image, $\mathbf{n}$,

$$\mathbf{I} = \mathbf{F_b} + \mathbf{n}, \tag{25}$$

where each value in $\mathbf{n}(r, c)$ was assumed to follow a normal distribution, $N(0, \sigma_n^2)$.

Figure 4 shows four typical example 101 × 101 pixel simulation line images with varying θ and constant $k = 1$, $\sigma_b = 1$, $w = 6$, and $\sigma_n = 8/255$.

Denoising is commonly applied before features are extracted from noise contaminated images, to obtain good quality features. Therefore, this paper applied a smoothing convolution,

$$\mathbf{I_s} = \mathbf{I} * \mathbf{s}, \tag{26}$$

where $\mathbf{s}$ is a two-dimensional gaussian function with standard deviation σ_s.

In this paper, the gradient images, $\mathbf{g_c}$ and $\mathbf{g_c}$ were derived by the convolution of a smoothed image with the Sobel operators. In addition, the elements of the Hessian matrix, $\mathbf{g_{cc}}$, $\mathbf{g_{rc}}$, and $\mathbf{g_{rr}}$ were derived by the convolution of the gradient images with the Sobel operators.

B. SPECIAL SIMULATION TESTS

First, line pixels in the simulation images were detected by a procedure based on the sum of gradient angle differences, and the TA and proposed NSG model was subsequently applied for subpixel line localization. Fig. 5 shows typical localization results for $\sigma_n = 30/255$ with varying w and θ. The proposed NSG model produced better localizations than TA for $w = 1, \theta = 0°$; $w = 2, \theta = 15°$; $w = 3, \theta = 30°$; $w = 6, \theta = 60°$; $w = 8$, and $\theta = 75°$. However, NSG and TA localizations were comparable for $w = 4, \theta = 45°$. NSG localizations were found significantly more accurate than TA when $w = 8$ (Fig. 5(f)), i.e., relatively weak line signals at line feature centers.

Figure 6 compares TA and NSG performance using root mean square error (RMSE) of localized positions for varying θ, w, and σ_n. All the graphs have the same scale except Figs. 6(h), (k), and (l). NSG produced better localizations than TA when $\sigma_n = 10/255$ and $w = 2, 8$, i.e., relatively weak line signals at line feature centers. Thus, the proposed NSG based approach provides more accurate localization than TA under severe conditions.

Localization RMSE for TA and NSG are comparable when $\sigma_n = 10/255$ and $w = 4$ (Figure 6(b)), indicating that TA and NSG performances are comparable for relatively strong

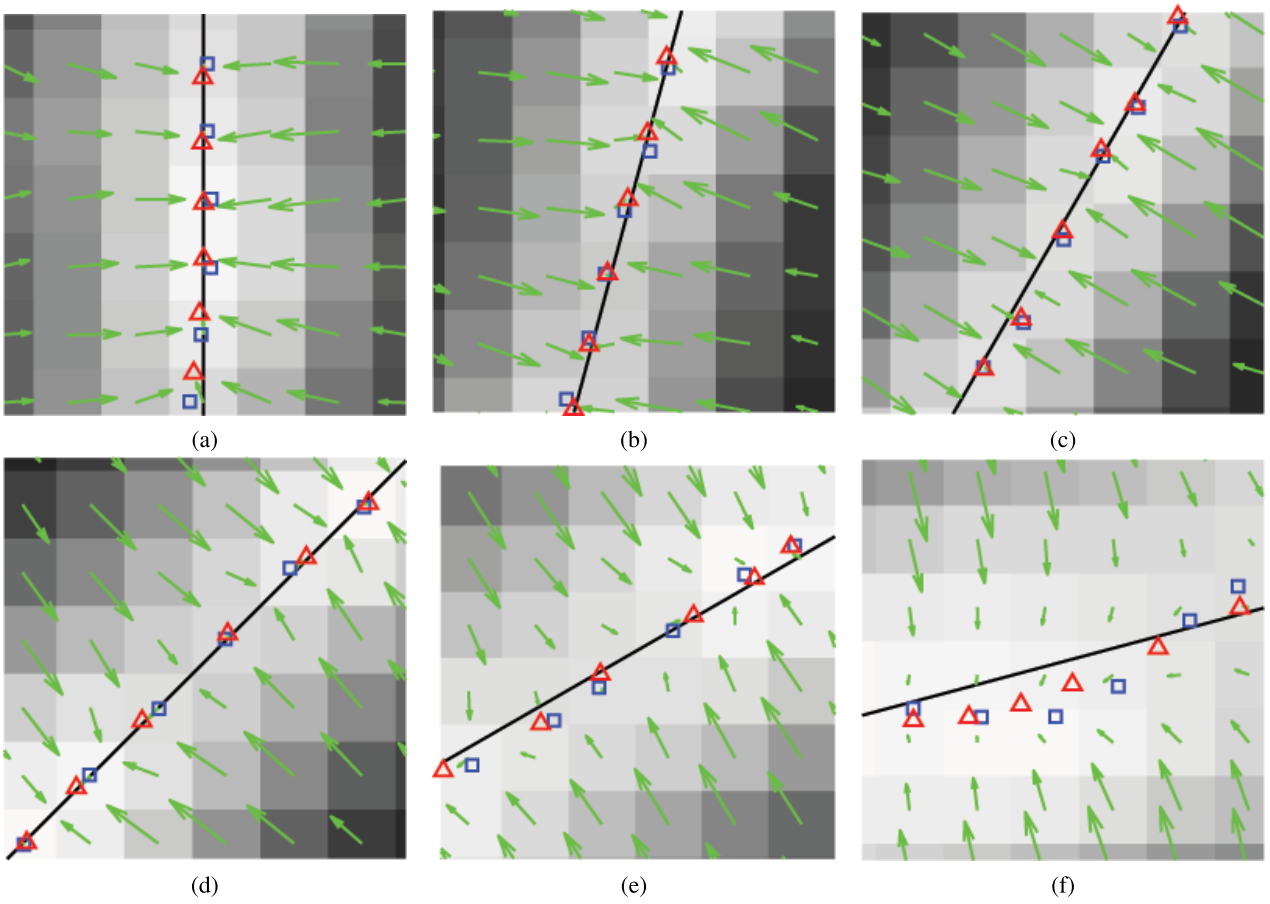


FIGURE 5. Localization results for TA and NSG methods on simulated images with noise strength $\sigma_n = 30/255$ with (a) line width $w = 1$ and line normal angle $\theta = 0°$, (b) $w = 2$ and $\theta = 15°$, (c) $w = 3$ and $\theta = 30°$, (d) $w = 4$ and $\theta = 45°$, (e) $w = 6$ and $\theta = 60°$, and (f) $w = 8$ and $\theta = 75°$. TA and NSG results are shown in blue rectangles and red triangles, respectively.

line signals. NSG produced lower quality localizations than TA only when $\sigma_n = 10/255$ and $w = 6$ (Fig. 6(c)). However, the performance difference is not significant and should not be considered a limiting case, because NSG produced superior localizations than TA for increased $\sigma_n = 30/255$ with constant $w = 6$ (Fig. 5(e)). This is discussed further in Section IV-C.

Considering noise strength, NSG produced superior localizations across all tested line widths (Figs. 6(h), (k), and (l)) and significantly outperformed TA for relatively weak or highly noise contaminated line signals at line feature centers. Thus, overall NSG provides better localization than TA aside from very limited cases.

C. GENERAL SIMULATION TESTS

Tests were performed on simulated line images generated with $w = 1, 2, \cdots, 8$ and $\sigma_n = 0, 5, \cdots, 60/255$, with $\theta = 0, 1, \cdots, 90°$ for each case. Localizations were performed using TA and NSG, and then RMSE was calculated for each case. Finally, mean RMSE was calculated for each w and σ_n combination. The effect of smoothing on localization was also investigated, setting $\sigma_s = 0.5, 0.75, \cdots, 1.25$; $\sigma_s > 1.25$ were not considered because larger σ_s tends to mix signals within a neighborhood in real images, and hence are inappropriate to extract fine features from images.

Figure 7 shows the outcomes for the various w, σ_n, θ, and σ_s experiments on simulated images. Overall, NSG produced more accurate localizations than TA under varying w and σ_n. Both methods produce large localization errors when $w < 2$ or $w > 5$ pixels compared with other conditions, but NSG remains superior to TA (Figs. 7(c), (f), and (i)). Localization errors for both TA and NSG monotonically increased with increasing σ_n, but NSG increases were significantly less than TA. Thus, NSG localizations were more accurate and less sensitive to noise and signal strength than TA.

Table 1 summarizes processing time for TA and NSG localization of all detected line pixels in the simulated images. The number of the detected line pixels varied with σ_s, line pixel detection generated multiple line responses around some true line locations for small σ_s. The proposed NSG method was approximately 20% faster and 40% more accurate than TA for $\sigma_s = 0.5$, and 25 % more accurate for $\sigma_s = 1.0$, providing significant justification to apply NSG rather than TA to real images, since images are commonly filtered with $\sigma_s = 1.0$

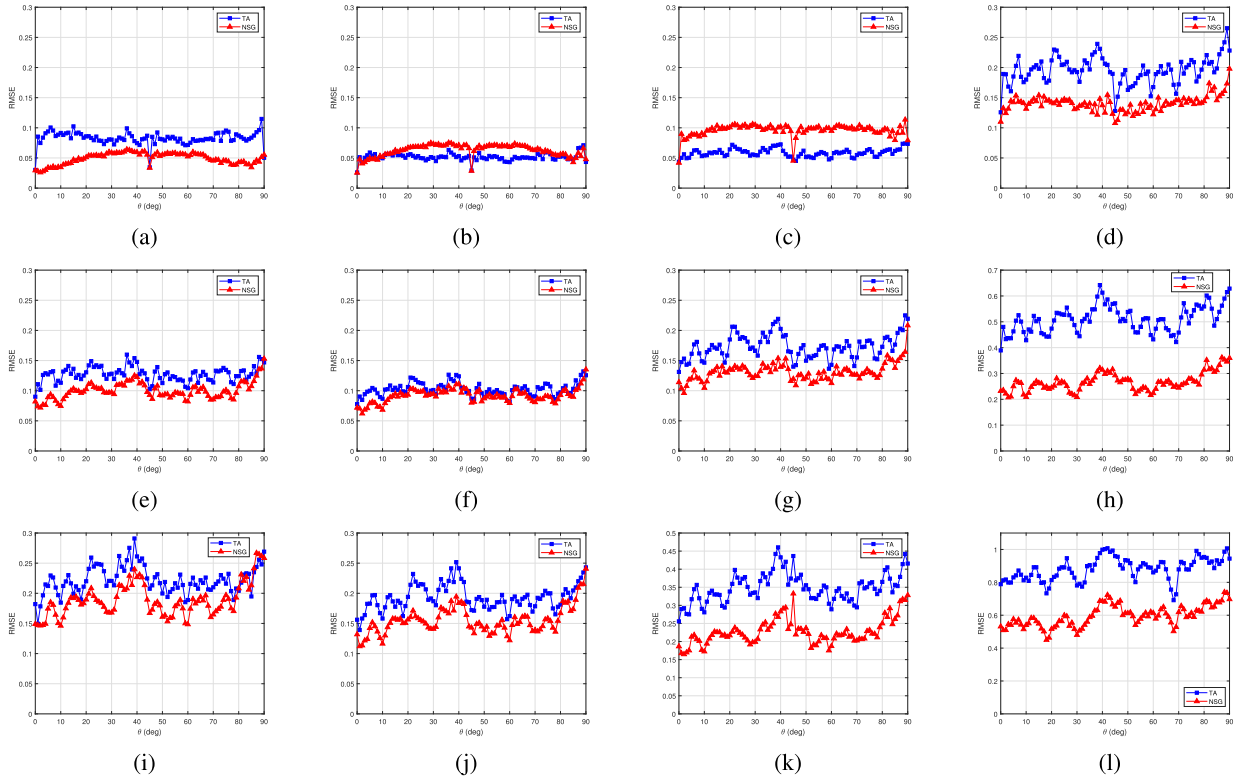


FIGURE 6. Simulation image localization against line normal angle (θ) for various noise strength (σ_n) and line width (w) with constant smoothing factor, $\sigma_s = 1.0$, and (a)–(d) $\sigma_n = 10/255$, (e)–(h) $\sigma_n = 30/255$, and (i)–(l) $\sigma_n = 60/255$. Line width (a), (e), and (i) $w = 2$; (b), (f), and (j) $w = 4$, (c), (g), and (k) $w = 6$, and (d), (h), and (l) $w = 8$ pixels.

TABLE 1. Processing times and overall root mean square error (RMSE) for Taylor's expansion approximation (TA) and proposed normalized sum of gradients (NSG) approaches on simulated images.

σ_s	Number of localized pixels	Processing time (s)		Mean RMSE	
		TA	NSG	TA	NSG
0.50	884,932	8.52	6.84	0.41	0.25
0.75	797,728	7.94	6.27	0.28	0.18
1.00	764,803	7.58	5.91	0.20	0.15
1.25	755,400	7.15	5.72	0.14	0.13

in many real world applications to better preserve fine image features while reducing noise significantly.

V. SUBPIXEL LINE LOCALIZATION EXPERIMENTS WITH NATURAL IMAGES

Figure 8 shows the twelve natural images collected for experimental tests of localization and linking methods. As shown in Fig. 8, the first nine images were color images having three bands and the rest of the images were greyscale images. Intensity of the images was [0, 255].

Column 1 in Fig. 9 shows the results of a sum of gradient angle differences-based line detection method with classification of line pixels into ridge and valley pixels, and Table 2 summarizes the number of detected line pixels for each class. Line features generated by this process are zigzagged because this typical line detection method produces line locations at pixel accuracy.

TABLE 2. Identified ridge and valley pixels in the natural images from Fig. 8.

Image	Ridge pixels (count)	Valley pixels (count)
Baboon	39,156	39,340
Barbara	40,009	41,489
Boats	34,094	33,516
Cablecar	23,778	24,548
Goldhill	46,096	47,384
Lena	16,203	16,849
Flowers	19,065	12,579
Monarch	19,157	22,695
Yacht	26,551	26,489
Fingerprint	25,960	23,587
Houses	27,161	27,759
Kiel	20,563	20,866

Columns 2 and 3 in Fig. 9 show TA and NSG localization outcomes, respectively on the natural images smoothed with $\sigma_s = 1.0$. The first derivatives or gradients were calculated by image convolution with the Sobel operators and the second derivatives by convolution of the first derivative images with the Sobel operators. Subpixel localization methods employed the first and second derivatives calculated for the previous line detection process without recalculation. TA and NSG localizations exhibit comparable accuracy for Lena subset

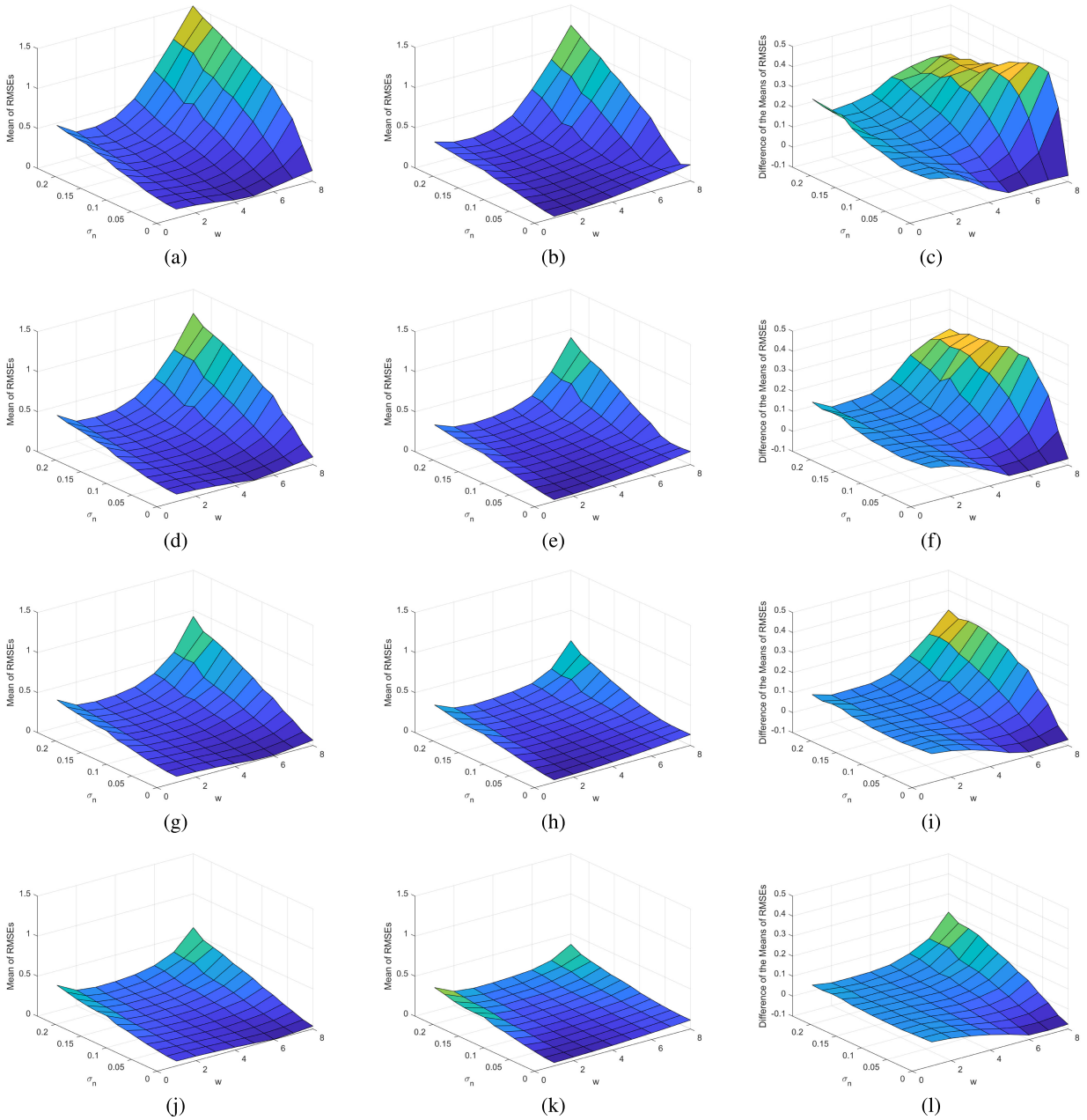


FIGURE 7. Root mean square error (RMSE) distribution effects for simulated localization under varying line width (w), noise strength (σ_n), and smoothing factor (σ_s). (column 1) TA and (column 2) NSG localization approaches, (column 3) difference between TA and NSG. (row 1) $\sigma_s = 0.5$, (row 2) $\sigma_s = 0.75$, (row 3) $\sigma_s = 1.0$, and (row 4) $\sigma_s = 1.25$.

images; whereas Barbara subset images exhibit that TA-based localizations were poor but NSG-based localizations good. As shown in Barbara subset images, TA-based localization was found to produce nonsmooth transitions during localization, especially when line is thin. The reason for the nonsmooth transition of TA-based localization is considered to be caused by unstability in calculation of the second derivatives when line is thin. This suggests NSG suffers less localization problems than TA when line width is relatively small.

To test TA and NSG performance under noisy conditions, random gaussian noise with $\sigma_n = 10$ was added to the original images, and subsequently smoothed with $\sigma_s = 1.0$. Columns 4 and 5 in Fig. 9 show the resultant TA and NSG localization performance, respectively. Localization quality of the Lena subset images was not significantly affected for both TA and NSG approaches when line width is not relatively small. However, localization quality of the Barbara subset images show that TA localizations in particular were significantly displaced from their true locations, whereas

FIGURE 8. Natural 512 × 512 pixel images: (a) Baboon, (b) Barbara, (c) Boats, (d) Cablecar, (e) Goodhill, (f) Lena, (g) Flowers, (h) Monarch, (i) Yacht, (j) Fingerprint, (k) Houses, and (l) Kiel.

NSG localizations corresponded well to visually detectable line feature locations when line signal strength at line feature centers afer smoothing was relatively low.

Quantification of the performance of the line localization methods was evaluated based on the localization results from the original images as ground truth. The reason for this strategy is due to the following reasons. Firstly, as listed in Table 2, the numbers of line pixels detected in the natural images are too large to check their correctness individually with human vision. Secondly, as shown in Fig. 10, there exist many line features in natural images that cannot be detected well by human vision. Thirdly, human vision is not good at localization of features at subpixel accuracy because a feature localization process requires calculations with all the intensities of pixels near the feature to be localized [18]. These reasons indicate that the ground truth made by human vision is not reliable to test the performance of line localization methods. Thus, in this paper, these problems were solved by using the localization result from the original images as ground truth and by evaluating the localization results from noisy images based on the ground truth.

Noisy images were generated by adding random gaussian noise with $\sigma_n = 5, 10, 15, 20, 30$. TA and NSG localization results from the original images are denoted as L^0_{TA} and L^0_{NSG}, and from images with additional noise as L^n_{TA} and L^0_{NSG}, respectively. Thus, L^0_{TA} and L^0_{NSG} were used as ground truth to evaluate the accuracy of L^n_{TA} and L^n_{NSG}, repectively by measuring the following mean distances. Mean distances among the localizations were calculated as

$$\begin{aligned} \mathrm{d1} &= \text{Mean Distance}(L^0_{\mathrm{TA}}, L^0_{\mathrm{NSG}}), \\ \mathrm{d2} &= \text{Mean Distance}(L^0_{\mathrm{TA}}, L^n_{\mathrm{TA}}), \\ \mathrm{d3} &= \text{Mean Distance}(L^o_{\mathrm{NSG}}, L^n_{\mathrm{NSG}}). \end{aligned} \tag{27}$$

Table 3 summarizes the mean distances for the natural images. Distances among ridge and valley pixels are prefixed with R and V, respectively. Mean distance, d1$\approx$ 0.5 pixels for all the test images except the Fingerprint image; and 0.40 pixels for the Fingerprint image for both ridge and valley

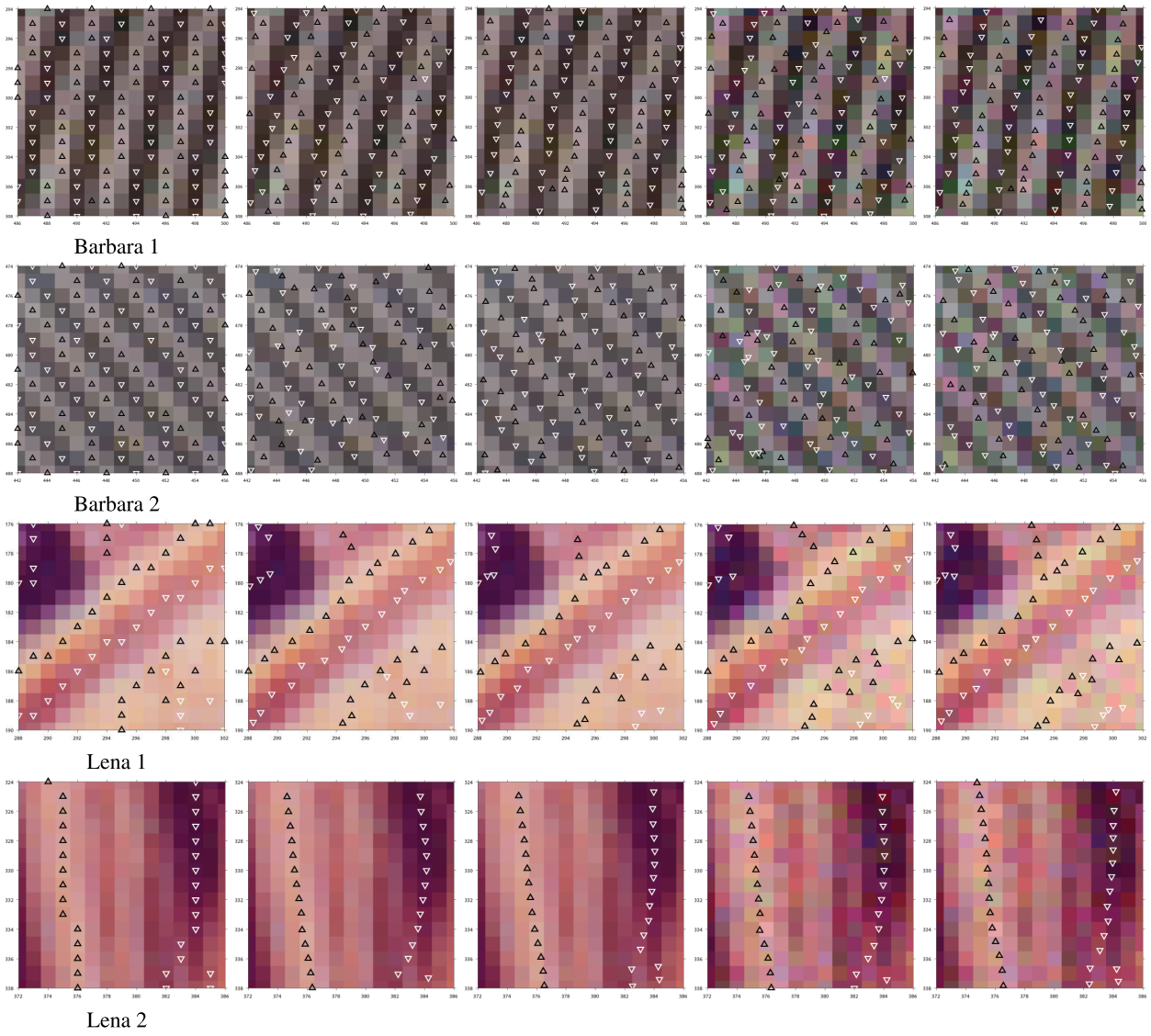


FIGURE 9. Examples of the experimental results of the line localization methods. (column 1) Line pixels detected by a line detection method. Ridge and valley pixels are denoted by upward-pointing black and downward-pointing white triangles, respectively. Background images are original images. (columns 2 and 3) Line localization results for the original images with smoothing factor $\sigma_S = 1.0$ by TA and NSG-based localizations, respectively. Background images are original images. (columns 4 and 5) Line localization results for the images with contamination of additional noise ($\sigma_n = 10$) and smoothing (smoothing factor, $\sigma_S = 1$) by TA and NSG-based localizations, respectively. Background images are the noise contaminated images.

pixels. The amount of mean distances for d1 can be considered as significant for precise engineering applications. Generally, d3 was approximately 30 to 40 % less than d2 across all additional noise levels and natural images. Thus, NSG is experimentally proved to produce more noise-insensitive and accurate localizations than TA across varying noise levels.

VI. LINE LINKING EXPERIMENT

Linking method experiments were performed for all the test images shown in Fig. 8. For implementing Steger's method [11], neighborhood size was set to 5 × 5 pixels, increasing the originally proposed 3 × 3 pixels to produce more suitable linking results. Implementations with the original size (not shown here) generated very unsatisfactory results with many fragmented line segments due to many subpixel localized positions that should have been linked having distance exceeding 1 pixel. Neighborhood size for the proposed linking method was also set to 5 × 5 pixels. Both linking methods were performed with subpixel locations produced by the proposed NSG method.

Figure 10 shows the line segments produced by the linking methods (line segments containing less than 5 points are not shown). The proposed linking method produced longer line segments than Steger's method across all image parts. In addition, Steger's method produced many missing line

TABLE 3. Mean distance between localized positions for the original and noisy images.

Baboon					
Distance	$\sigma_n = 5$	$\sigma_n = 10$	$\sigma_n = 15$	$\sigma_n = 20$	$\sigma_n = 30$
Rd1	0.52	0.52	0.52	0.52	0.52
Rd2	0.18	0.34	0.47	0.57	0.71
Rd3	0.10	0.18	0.25	0.31	0.40
Rd3/Rd2(%)	51.6	52.8	53.7	54.2	56.8
Vd1	0.52	0.52	0.52	0.52	0.52
Vd2	0.19	0.35	0.48	0.58	0.73
Vd3	0.10	0.19	0.26	0.32	0.41
Vd3/Vd2(%)	51.5	52.7	54.2	55.4	56.6

Barbara					
Distance	$\sigma_n = 5$	$\sigma_n = 10$	$\sigma_n = 15$	$\sigma_n = 20$	$\sigma_n = 30$
Rd1	0.51	0.51	0.51	0.51	0.51
Rd2	0.27	0.46	0.58	0.68	0.83
Rd3	0.14	0.25	0.33	0.39	0.48
Rd3/Rd2(%)	52.8	54.5	56.5	57.5	57.9
Vd1	0.51	0.51	0.51	0.51	0.51
Vd2	0.28	0.47	0.60	0.70	0.83
Vd3	0.15	0.26	0.34	0.41	0.49
Vd3/Vd2(%)	55.6	56.3	57.5	58.0	59.2

Boats					
Distance	$\sigma_n = 5$	$\sigma_n = 10$	$\sigma_n = 15$	$\sigma_n = 20$	$\sigma_n = 30$
Rd1	0.49	0.49	0.49	0.49	0.49
Rd2	0.26	0.44	0.58	0.68	0.83
Rd3	0.16	0.27	0.35	0.42	0.51
Rd3/Rd2(%)	59.5	61.0	61.4	62.0	62.2
Vd1	0.48	0.48	0.48	0.48	0.48
Vd2	0.30	0.46	0.59	0.69	0.84
Vd3	0.18	0.28	0.36	0.42	0.51
Vd3/Vd2(%)	60.1	61.8	61.1	61.7	60.9

Cablecar					
Distance	$\sigma_n = 5$	$\sigma_n = 10$	$\sigma_n = 15$	$\sigma_n = 20$	$\sigma_n = 30$
Rd1	0.51	0.51	0.51	0.51	0.51
Rd2	0.25	0.42	0.55	0.65	0.79
Rd3	0.14	0.24	0.32	0.38	0.47
Rd3/Rd2(%)	55.2	56.9	57.4	58.3	59.2
Vd1	0.50	0.50	0.50	0.50	0.50
Vd2	0.26	0.44	0.56	0.66	0.78
Vd3	0.14	0.24	0.32	0.39	0.47
Vd3/Vd2(%)	54.5	55.9	57.7	58.4	60.3

Goldhill					
Distance	$\sigma_n = 5$	$\sigma_n = 10$	$\sigma_n = 15$	$\sigma_n = 20$	$\sigma_n = 30$
Rd1	0.52	0.52	0.52	0.52	0.52
Rd2	0.31	0.52	0.68	0.79	0.92
Rd3	0.16	0.28	0.37	0.45	0.54
Rd3/Rd2(%)	52.1	54.9	55.1	56.6	58.6
Vd1	0.52	0.52	0.52	0.52	0.52
Vd2	0.32	0.55	0.72	0.83	0.96
Vd3	0.17	0.30	0.40	0.47	0.56
Vd3/Vd2(%)	52.9	54.1	55.1	56.6	58.6

Lena					
Distance	$\sigma_n = 5$	$\sigma_n = 10$	$\sigma_n = 15$	$\sigma_n = 20$	$\sigma_n = 30$
Rd1	0.49	0.49	0.49	0.49	0.49
Rd2	0.34	0.55	0.71	0.78	0.90
Rd3	0.17	0.30	0.40	0.46	0.55
Rd3/Rd2(%)	52.0	55.0	56.0	59.0	61.2
Vd1	0.50	0.50	0.50	0.50	0.50
Vd2	0.30	0.50	0.65	0.76	0.89
Vd3	0.16	0.29	0.38	0.45	0.54
Vd3/Vd2(%)	55.0	58.1	58.9	59.8	60.5

Flowers					
Distance	$\sigma_n = 5$	$\sigma_n = 10$	$\sigma_n = 15$	$\sigma_n = 20$	$\sigma_n = 30$
Rd1	0.53	0.53	0.53	0.53	0.53
Rd2	0.23	0.39	0.52	0.61	0.76
Rd3	0.13	0.23	0.30	0.35	0.44
Rd3/Rd2(%)	56.7	57.5	57.2	58.0	57.8
Vd1	0.53	0.53	0.53	0.53	0.53
Vd2	0.25	0.43	0.54	0.62	0.77
Vd3	0.15	0.25	0.32	0.38	0.48
Vd3/Vd2(%)	59.1	57.5	59.4	60.5	61.6

Monarch					
Distance	$\sigma_n = 5$	$\sigma_n = 10$	$\sigma_n = 15$	$\sigma_n = 20$	$\sigma_n = 30$
Rd1	0.56	0.56	0.56	0.56	0.56
Rd2	0.31	0.49	0.62	0.71	0.83
Rd3	0.16	0.27	0.36	0.42	0.51
Rd3/Rd2(%)	52.8	55.0	57.1	58.7	60.8
Vd1	0.52	0.52	0.52	0.52	0.52
Vd2	0.28	0.46	0.58	0.67	0.80
Vd3	0.16	0.27	0.35	0.41	0.50
Vd3/Vd2(%)	58.5	59.3	60.4	61.7	63.1

Yacht					
Distance	$\sigma_n = 5$	$\sigma_n = 10$	$\sigma_n = 15$	$\sigma_n = 20$	$\sigma_n = 30$
Rd1	0.45	0.45	0.45	0.45	0.45
Rd2	0.25	0.43	0.55	0.64	0.78
Rd3	0.16	0.27	0.35	0.41	0.50
Rd3/Rd2(%)	63.7	62.6	63.2	63.5	63.7
Vd1	0.45	0.45	0.45	0.45	0.45
Vd2	0.25	0.42	0.53	0.63	0.75
Vd3	0.15	0.26	0.34	0.40	0.48
Vd3/Vd2(%)	60.3	62.1	63.3	63.3	65.0

Fingerprint					
Distance	$\sigma_n = 5$	$\sigma_n = 10$	$\sigma_n = 15$	$\sigma_n = 20$	$\sigma_n = 30$
Rd1	0.37	0.37	0.37	0.37	0.37
Rd2	0.09	0.17	0.25	0.33	0.45
Rd3	0.07	0.13	0.18	0.23	0.32
Rd3/Rd2(%)	74.9	73.4	72.4	71.7	71.0
Vd1	0.44	0.44	0.44	0.44	0.44
Vd2	0.09	0.17	0.25	0.32	0.46
Vd3	0.06	0.11	0.16	0.21	0.30
Vd3/Vd2(%)	66.3	67.9	67.0	66.4	65.3

Houses					
Distance	$\sigma_n = 5$	$\sigma_n = 10$	$\sigma_n = 15$	$\sigma_n = 20$	$\sigma_n = 30$
Rd1	0.46	0.46	0.46	0.46	0.46
Rd2	0.21	0.35	0.45	0.53	0.65
Rd3	0.13	0.22	0.29	0.34	0.42
Rd3/Rd2(%)	62.2	63.1	64.1	64.1	65.0
Vd1	0.45	0.45	0.45	0.45	0.45
Vd2	0.19	0.33	0.44	0.53	0.65
Vd3	0.13	0.22	0.29	0.34	0.42
Vd3/Vd2(%)	66.1	67.0	66.6	64.8	64.3

Kiel					
Distance	$\sigma_n = 5$	$\sigma_n = 10$	$\sigma_n = 15$	$\sigma_n = 20$	$\sigma_n = 30$
Rd1	0.47	0.47	0.47	0.47	0.47
Rd2	0.24	0.40	0.53	0.62	0.76
Rd3	0.15	0.26	0.33	0.39	0.47
Rd3/Rd2(%)	63.5	63.3	62.6	62.5	61.6
Vd1	0.48	0.48	0.48	0.48	0.48
Vd2	0.23	0.38	0.50	0.60	0.74
Vd3	0.14	0.23	0.31	0.37	0.46
Vd3/Vd2(%)	60.0	61.5	61.7	61.5	61.9

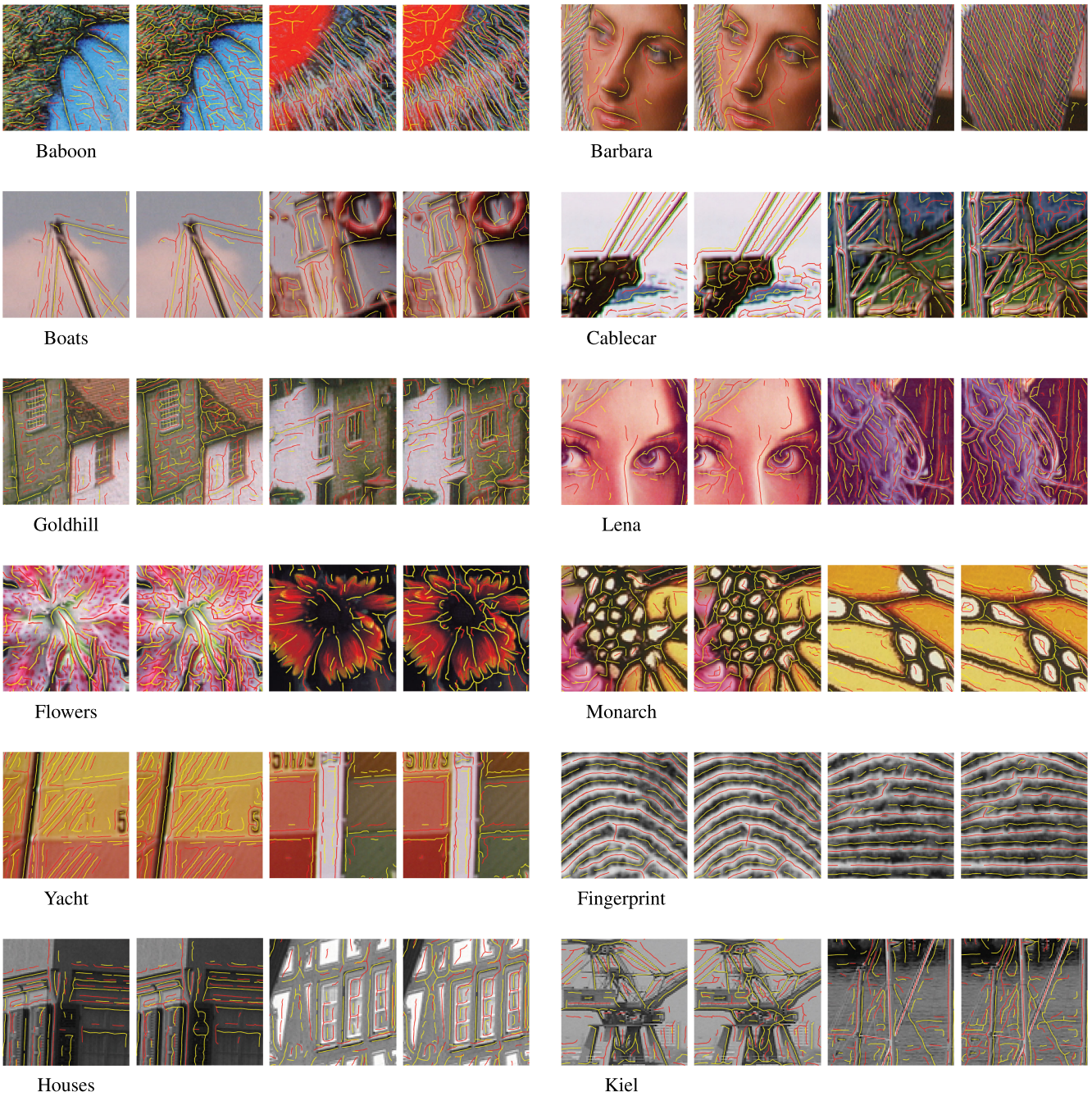


FIGURE 10. Line segments generated by the tested linking methods. (columns 1, 3, 5, 7) Results produced by Steger's method. (columns 2, 4, 6, 8) Results produced by the proposed linking methods. Ridge and valley lines are shown in red and yellow lines, respectively. Background images are subsets of the original images.

segments parts compared to the proposed method. The reason the proposed linking method produced longer line segments was due to better straightness and omni-directionality in the linking process, discussed in Section III-D. As shown in Fig. 10, many phantom lines were observed near real line and edge features. This effect is considered to be caused by nonhomogeneous blurs for the bright and dark sides of image features during imaging [40].

VII. CONCLUSION

This paper proposed a subpixel line localization method based on normalized sums of gradients. Subpixel line localization experiments with simulated and natural images showed that the proposed subpixel line localization method was faster and more accurate than a current state-of-the art method under varying combinations of line width and noise strength.

A linking method based on a linking distance measure derived from multiplicative combination of geometric and angular distances was also proposed, and shown to produce better quality line features than a current state-of-the art method.

Thus, combining the proposed subpixel line localization and linking methods is expected to be used to extract high quality line features from real images with various signal and noise levels. In this paper, the proposed line localization and linking methods were only tested for natural images. However, it is expected that the proposed methods can be readily extended to the extraction of line features from binary document images and binary contour images.

REFERENCES

[1] R. M. Haralick, "Ridges and valleys on digital images," *Comput. Vis., Graph., Image Process.*, vol. 22, no. 1, pp. 28–38, 1983.
[2] J. M. Gauch and S. M. Pizer, "Multiresolution analysis of ridges and valleys in grey-scale images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 6, pp. 635–646, Jun. 1993.
[3] J. Serrat, A. Lopez, and D. Lloret, "On ridges and valleys," in *Proc. 15th Int. Conf. Pattern Recognit.*, Sep. 2000.
[4] C. R. Jung and C. R. Kelber, "Lane following and lane departure using a linear-parabolic model," *Image Vis. Comput.*, vol. 23, no. 13 pp. 1192–1202, 2005.
[5] X. J. An, E. Shang, J. Z. Song, J. Li, and H. G. He, "Real-time lane departure warning system based on a single FPGA," *EURASIP J. Image Vide Process.*, vol. 38, pp. 1–18, Dec. 2013.
[6] P. Wu, C. Chang, and C. Lin, "Lane mark extraction for automobiles under complex conditions," *Pattern Recognit.*, vol. 47, no. 8, pp. 2756–2767, 2014.
[7] J. Son, H. Yoo, S. Kim, and K. Sohn, "Real-time illumination invariant lane detection for lane departure warning system," *Expert Syst. Appl.*, vol. 42, no. 4, pp. 1816–1824, Oct. 2014.
[8] B. Karasulu, "Automatic extraction of retinal blood vessels: A software implementation," *Eur. Sci. J.*, vol. 8, no. 30, pp. 47–57, 2012.
[9] J. Majumdar, D. Kundu, S. Tewary, S. Ghosh, S. Chakraborty, and S. Gupta, "An automated graphical user interface based system for the extraction of retinal blood vessels using kirsch's template," *Int. J. Adv. Comput. Sci. Appl.*, vol. 6, no. 6, pp. 86–93, 2015.
[10] W. Wang, N. Yang, Y. Zhang, F. Wang, T. Cao, and P. Eklund, "A review of road extraction from remote sensing images," *J. Traffic Transp. Eng.*, vol. 3, no. 3, pp. 271–282, 2016.
[11] C. Steger, "An unbiased detector of curvilinear structures," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 2, pp. 113–125, Feb. 1998.
[12] J. Basak, B. Chanda, and D. D. Majumder, "On edge and line linking with connectionist models," *IEEE Trans. Syst., Man, Cybern.*, vol. 24, no. 3, pp. 413–428, Mar. 1994.
[13] C. Akinlar and E. Chome, "PEL: A predictive edge linking algorithm," *J. Vis. Commun. Image Represent.*, vol. 36, pp. 159–171, Apr. 2016.
[14] A. Tabatabai and O. Mitchell, "Edge location to subpixel values in digital imagery," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-6, no. 2, pp. 188–201, Feb. 1984.
[15] E. P. Lyvers, O. R. Mitchell, M. L. Akey, and A. P. Reeves, "Subpixel measurements using a moment-based edge operator," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 12, pp. 1293–1309, Dec. 1989.
[16] Y. Shan and G. W. Boon, "Sub-pixel location of edges with non-uniform blurring: A finite closed-form approach," *Image Vis. Comput.*, vol. 18, no. 13, pp. 1015–1023, Oct. 2000.
[17] M. Hagara and P. Kulla, "Edge detection with sub-pixel accuracy based on approximation of edge with Erf function," *Radioengineering*, vol. 20, no. 2, pp. 516–524, Jun. 2011.
[18] S. Seo, "Subpixel edge localization based on adaptive weighting of gradients," *IEEE Trans. Image Process.*, vol. 27, no. 11, pp. 5501–5513, 2018.
[19] A. A. Farag and E. J. Delp, "Edge linking by sequential search," *Pattern Recognit.*, vol. 28, no. 5, pp. 611–633, 1995.
[20] D.-S. Lu and C.-C. Chen, "Edge detection improvement by and colony optimization," *Pattern Recognit. Lett.*, vol. 29, no. 4, pp. 416–425, 2008.
[21] Z. Wang and H. Zhang, "Edge linking using geodesic distance and neighborhood information," in *Proc. IEEE/ASME*, Jul. 2008, pp. 151–155.
[22] Q. Lin, Y. Han, and H. Hahn, "Real-time lane detection based on extended edge-linking algorithm," in *Proc. 2nd Int. Conf. Comput. Res. Develop.*, May 2010.
[23] J. L. Flores, G. A. Ayubi, J. R. Alonso, A. Fernandez, J. M. D. Martino, and J. A. Ferrari, "Edge linking and image segmentation by combining optical and digital methods," *Optik*, vol. 124, no. 18, pp. 3260–3264, 2013.
[24] C. Akinlar and C. Topal, "EDPF: A real-time parameter-free edge segment detector with a false detection control," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 26, no. 1, 2012, Art. no. 1255002.
[25] C. Akinlar and E. Chome, "CannySR: Using smart routing of edge drawing to convert canny binary edge maps to edge segments," in *Proc. INISTA*, Sep. 2015.
[26] C. Akinlar, "CEDContours: A high speed contour detector for color images," *Image Vis. Comput.*, vol. 54, pp. 60–70, 2016.
[27] Q. Zhu, M. Payne, and V. Riordan, "Edge linking by a directional potential function (DPF)," *Image Vis. Comput.*, vol. 14, no. 1, pp. 59–70, 1996.
[28] O. Ghita, "Computational approach for edge linking," *J. Electron. Imag.*, vol. 11, no. 4, pp. 479–485, 2002.
[29] F. Y. Shih and S. Cheng, "Adaptive mathematical morphology for edge linking," *Inf. Sci.*, vol. 167, nos. 1–4, pp. 9–21, Dec. 2004.
[30] A. Sappa and B. Vintimilla, "Edge point linking by means of global and local schemes," in *Signal Processing for Image Enhancement Multimedia Processing*, E. Damiani, A. Dipanda, K. Yetongnon, L. Legrand, P. Schelkens, and R. Chbeir, Eds. Boston, MA, USA: Springer, 2008, pp. 115–125.
[31] J. Biemond, A. M. Tekalp, and R. L. Lagendijk, "Maximum likelihood image and blur identification: A unifying approach," *Opt. Eng.*, vol. 29, no. 5, pp. 422–435, 1990.
[32] S. J. Reeves and R. M. Mersereau, "Blur identification by the method of generalized cross-validation," *IEEE Trans. Image Process.*, vol. 1, no. 3, pp. 301–311, Jul. 1992.
[33] A. E. Savakis and H. J. Trussell, "Blur identification by residual spectral matching," *IEEE Trans. Image Process.*, vol. 2, no. 2, pp. 141–151, Apr. 1993.
[34] D. Kundur and D. Hatzinakos, "Blind image deconvolution," *IEEE Signal Process. Mag.*, vol. 13, no. 3, pp. 43–64, May 1996.
[35] Y. Yitzhaky and N. S. Kopeika, "Identification of blur parameters from motion blurred images," *Graph. Models Image Process.*, vol. 59, no. 5, pp. 310–320, 1997.
[36] L. Chen and K.-H. Yap, "Efficient discrete spatial techniques for blur support identification in blind image deconvolution," *IEEE Trans. Signal Process.*, vol. 54, no. 4, pp. 1557–1562, Apr. 2006.
[37] S. Wu, W. Lin, Z. Lu, E. P. Ong, and S. Yao, "Blind blur assessment for vision-based applications," *J. Vis. Commun. Image Represent.*, vol. 20, no. 4, pp. 231–241, May 2009.
[38] W. Hu, J. Xue, and N. Zheng, "PSF estimation via gradient domain correlation," *IEEE Trans. Image Process.*, vol. 21, no. 1, pp. 386–392, Jan. 2012.
[39] S. Liu, H. Wang, J. Wang, S. Cho, and C. Pan, "Automatic blur-kernel-size estimation for motion deblurring," *Vis. Comput.*, vol. 31, no. 5, pp. 733–746, 2015.
[40] S. Seo, "Edge modeling by two blur parameters in varying contrasts," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 2701–2714, Feb. 2018.

SUYOUNG SEO (M'17) received the B.S. and M.S. degrees in civil engineering (major in Urban Engineering) from Seoul National University, Seoul, in 1992 and 1994, respectively, and the Ph.D. degree in geodetic science from The Ohio State University, Columbus, OH, USA, in 2003. He was a Senior Researcher with the Korea Land Spatialization Group, Inha University, Incheon, from 2007 to 2009. He is currently a Professor with the Department of Civil Engineering, Kyungpook National University, Daegu, South Korea. His research interests include feature extraction and object reconstruction.