IPSJ Transactions on Computer Vision and Applications Vol.5 70–74 (July 2013)

[DOI: 10.2197/ipsjtcva.5.70]

Express Paper

A Weighted Integral Method for Parametrically Describing Local Image Appearance

Hidetoshi Goto[1,a)] Hidekata Hontani[1,b)]

Received: March 11, 2013, Accepted: April 24, 2013, Released: July 29, 2013

Abstract: We propose a new method that efficiently and accurately estimates the parameters of the Gaussian function that describes the given local image profiles. The Gaussian function is non-linear with respect to the parameters to be estimated, and this non-linearity makes their efficient and accurate estimation difficult. In our proposed method, the weighted integral method is introduced to linearize the parameter estimation problem: A system of differential equations is firstly derived that is satisfied by the Gaussian function and that is linear with respect to the parameters. The system is then converted to that of integral equations. Given a local sub-window of the image, one can obtain the system of integral equations and estimate the parameters of the Gaussian that describe the appearance in the sub-window by solving the linear system of the parameters. Experimental results showed that our proposed method estimates the parameters more efficiently and accurately than existing state-of-the-art methods.

Keywords: feature description, parameter estimation problem, scale analysis, weighted integral method

1. Introduction

1.1 Background

Edges are classical but still important primitives for describing local image appearance [4], [9]. In the framework proposed by Zhu et al. [12], local image appearances are divided into two classes: primitives and textures. The former ones are defined as the local appearances that can be represented by explicit model functions, e.g., the Gabor functions or the Gaussian ones. An explicit model function has a set of parameters, and the given local appearances can be described by estimating the values of these parameters. The objective of this study was to develop an accurate and efficient method for describing local image appearances with the Gaussian model function.

The Gaussian function is widely used as the explicit model function. It has a scale parameter, of which the value can be estimated to describe the blurring scale at each location in given images. Such descriptions of the blurring scales are required for many applications, e.g., image matting [11], moving object detection [5], image enhancement [3], and 3D-shape reconstruction from defocus [6], [10]. Many methods, hence, have been proposed for estimating the blurring scale at each location in the given images. One of the main difficulties of the estimation comes from the non-linearity of the Gaussian: It is a non-linear function of the parameters.

1.2 Parameter Estimation Problem

Let $G(\boldsymbol{m})$ denote a model function, where $\boldsymbol{m}$ is a vector, of which the components are the parameters to be estimated. Let $\boldsymbol{d}$ denote measurements. The model function represents the observation as follows [2]:

$$G(\boldsymbol{m}) + \boldsymbol{\epsilon} = \boldsymbol{d}, \tag{1}$$

where $\boldsymbol{\epsilon}$ denotes the measurement noises. When $\boldsymbol{\epsilon}$ obeys the i.i.d. Gaussian, the ML estimate, $\hat{\boldsymbol{m}}$, can be obtained by minimizing the squared errors:

$$\hat{\boldsymbol{m}} = \arg\min_{\boldsymbol{m}} E(\boldsymbol{m}), \tag{2}$$

where $E(\boldsymbol{m}) = \|G(\boldsymbol{m}) - \boldsymbol{d}\|^2$. It is not easy to accurately and efficiently solve the above problem, when $G(\boldsymbol{m})$ is non-linear: The iterative minimization process is required for computing $\hat{\boldsymbol{m}}$. Many parameter estimation methods, including the Fourier analysis for frequency estimation, prepare a set of paramters, $\mathcal{M} = \{\boldsymbol{m}_i | i = 1, 2, \ldots, M\}$, computes $E(\boldsymbol{m}_i)$ for each $\boldsymbol{m}_i \in \mathcal{M}$, and selects the minimizer, $\hat{\boldsymbol{m}} = \arg\min_{\boldsymbol{m}_i \in \mathcal{M}} E(\boldsymbol{m}_i)$. It is easier, on the other hand, to solve the problem, when $G(\boldsymbol{m})$ is linear: $G(\boldsymbol{m}) = A\boldsymbol{m}$. The minimizer can then be computed as $\hat{\boldsymbol{m}} = A^{-}\boldsymbol{d}$.

In this study, in order to linearize the parameter estimation problem, we introduce a weighted integral method proposed by Ando et al. [1], which was developed for accurately and efficiently estimating the frequencies of temporal signals. Let m and ϕ denote the angular frequency and the phase of a signal. The model function of time, $f(t|m) = A\cos(mt + \phi)$, is non-linear with respect to m, which is the parameter to be estimated. The Fourier analysis or the wavelet one estimates m as described above: The frequency, $m_i \in \mathcal{M}$, that minimizes the squared errors, $E(m_i)$, is searched. In the weighted integral method, on the other hand, a linear equation of m is derived from a differential equation that is satisfied by $f(t|m)$, and the frequency is efficiently and accurately estimated by solving the equation. The details of the framework will be shown in the following sections. In this paper, the

[1] Nagoya Institute of Technology, Nagoya, Aichi 466–8555, Japan
[a)] goto@iu.nitech.ac.jp
[b)] hontani@nitech.ac.jp

weighted integral method is expanded for 2D image signals, and is applied for the parametric edge description.

2. Weighted Integral Method

Let $I(x, y)$ denote a given image, and let $f(x, y) = \|\nabla I(x, y)\|$. Let $\Omega(x, y)$ denote a small $L \times L$ window, of which the center is located at (x, y) in the image, and let (u, v) be the local coordinates in $\Omega(x, y)$. Given $f(u, v)$ where $(u, v) \in \Omega(x, y)$, the proposed method estimates the values of the parameters in the following Gaussian function (see also **Fig. 1**):

$$N(u, v|\boldsymbol{m}) = A \exp\{-(u \cos\theta + v \sin\theta - \mu)^2/2\sigma^2\}, \qquad (3)$$

where the parameters to be estimated are $\boldsymbol{m} = [\theta, \sigma^2, \mu]^T$. The function, $N(u, v|\boldsymbol{m})$, is non-linear with respect to $\boldsymbol{m}$.

Using the weighted integral method, one can linearize the problem. If $f(u, v)$ can be described by the model function shown in Eq. (3), the following differential equations are satisfied at all locations in Ω:

$$\frac{\partial f(u, v)}{\partial u} = (q_{10}u + q_{01}v + q_{00}) f(u, v), \qquad (4)$$

$$\frac{\partial f(u, v)}{\partial v} = (r_{10}u + r_{01}v + r_{00}) f(u, v), \qquad (5)$$

where $q_{10} = -\cos^2\theta/\sigma^2$, $q_{01} = r_{10} = -\cos\theta\sin\theta/\sigma^2$, $r_{01} = -\sin^2\theta/\sigma^2$, $q_{00} = \mu\cos\theta/\sigma^2$, and $r_{00} = \mu\sin\theta/\sigma^2$. It should be noted that the above equations are linear with respect to q_{st} and r_{st}, which are the functions of the parameters. The values of u, v, $\partial f/\partial u$, $\partial f/\partial v$, and $f(u, v)$, are directly determined at each location in Ω from the input signal, $f(u, v)$. The objective here is to estimate the values of the five unknowns, q_{st} and r_{st} (it should be noted that $q_{01} = r_{10}$). Once those values are estimated, then, one can easily compute the values of $\boldsymbol{m}$.

One can obtain two equations at each pixel, $(u, v) \in \Omega$. Solving the system of $2 \times L \times L$ equations obtained from all of the pixels in Ω, one would be able to obtain the estimates of q_{st} and r_{st}. The resultant estimates, though, are not accurate, because the values of the coefficients, $\partial f/\partial u$, $\partial f/\partial v$, and $f(u, v)$, are perturbed by measurement noises. The weighted integral method converts those differential equations into integral ones, in order to improve the accuracy. From Eqs. (4) and (5), one can obtain the following equations:

$$\int_\Omega \left\{\frac{\partial f}{\partial u} - (q_{10}u + q_{01}v + q_{00}) f(u, v)\right\} w(u, v) du dv = 0, \qquad (6)$$

$$\int_\Omega \left\{\frac{\partial f}{\partial v} - (r_{10}u + r_{01}v + r_{00}) f(u, v)\right\} w(u, v) du dv = 0, \qquad (7)$$

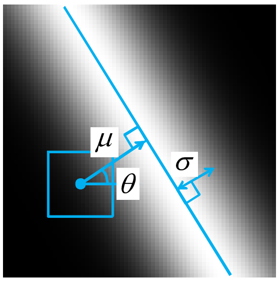


Fig. 1 A window, Ω, indicated by a blue rectangle, and the parameters to be estimated.

where $w(u, v)$ is a weight function defined on Ω. The above equations are satisfied for any weight function, $w(u, v)$, if and only if Eqs. (4) and (5) are satisfied at all locations in Ω.

Let

$$w(u, v) = p(u)p(v)e^{-j(\omega_u u + \omega_v v)}, \qquad (8)$$

where $p(\cdot)$ is a smooth window function that satisfies $p(\cdot) = 0$ at the boundary of Ω (e.g., you can use the Hann window function for $p(\cdot)$), and $\omega_u = 2\pi N_u/L$ and $\omega_v = 2\pi N_v/L$ (N_u and N_v are integers). It is necessary that the equations in Eqs. (6) and (7) are satisfied when the weight function in Eq. (8) is substituted, if Eqs. (4) and (5) are satisfied at all locations in Ω. Integrating each term in Eqs. (6) and (7) by parts, one can obtain the following linear equations:

$$g^{10}q_{10} + g^{01}q_{01} + g^{00}q_{00} = h^u - j\omega_u g^{00}, \qquad (9)$$

$$g^{10}q_{01} + g^{01}r_{01} + g^{00}r_{00} = h^v - j\omega_v g^{00}, \qquad (10)$$

where

$$g^{st} \equiv \int_\Omega u^s v^t f(u, v) w(u, v) du dv, \qquad (11)$$

$$h^u \equiv \int_\Omega f(u, v) \frac{\partial p(u)}{\partial u} p(v) e^{-j(\omega_u u + \omega_v v)} du dv, \qquad (12)$$

$$h^v \equiv \int_\Omega f(u, v) p(u) \frac{\partial p(v)}{\partial v} e^{-j(\omega_u u + \omega_v v)} du dv. \qquad (13)$$

The coefficients, g^{st}, h^u, and h^v, are complex numbers, and the system of the linear equations shown in Eqs. (9) and (10) consists of four equations with real number coefficients. Let the real part and the imaginary part of a variable y be denoted by $y_\mathcal{R}$ and by $y_\mathcal{I}$, respectively. Then, the equations shown in Eqs. (9) and (10) can be represented as follows:

$$A\boldsymbol{q} = \boldsymbol{b}, \qquad (14)$$

where $\boldsymbol{q} = [q_{10}, q_{01}, r_{01}, q_{00}, r_{00}]^T$,

$$A = \begin{bmatrix} g^{10}_\mathcal{R} & g^{01}_\mathcal{R} & 0 & g^{00}_\mathcal{R} & 0 \\ g^{10}_\mathcal{I} & g^{01}_\mathcal{I} & 0 & g^{00}_\mathcal{I} & 0 \\ 0 & g^{10}_\mathcal{R} & g^{01}_\mathcal{R} & 0 & g^{00}_\mathcal{R} \\ 0 & g^{10}_\mathcal{I} & g^{01}_\mathcal{I} & 0 & g^{00}_\mathcal{I} \end{bmatrix}, \text{ and } \boldsymbol{b} = \begin{bmatrix} h^u_\mathcal{R} + \omega_u g^{00}_\mathcal{I} \\ h^u_\mathcal{I} - \omega_u g^{00}_\mathcal{R} \\ h^v_\mathcal{R} + \omega_v g^{00}_\mathcal{I} \\ h^v_\mathcal{I} - \omega_v g^{00}_\mathcal{R} \end{bmatrix}. \qquad (15)$$

The accurate values of these coefficients, A and $\boldsymbol{b}$, can be obtained stably, because the weighted integration suppresses the effects of the measurement noises. One can vary the weight function, $w(u, v)$, by changing the values of ω_u and ω_v, and can obtain the system of four linear equations shown in Eq. (14) for each weight function. We need more than one weight function for determining the values of the five unknowns, $\boldsymbol{q}$, uniquely. Let A_i and $\boldsymbol{b}_i$ denote the matrix and the vector obtained by using the i-th weight function, $w_i(u, v)$ $(i = 1, 2)$. Let

$$A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} \text{ and } \boldsymbol{b} = \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \end{bmatrix}. \qquad (16)$$

Then, the values of $\boldsymbol{q}$ can be computed by solving the linear system, $A\boldsymbol{q} = \boldsymbol{b}$ for each window, $\Omega(x, y)$.

3. Experimental Results

3.1 Experiments with Artificial Images

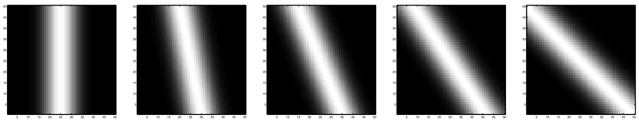
The performance of the proposed method was evaluated by using artificial images. Fixing the scale as $\sigma = 5$, we generated artificial Gaussian input signals, $f(x, y)$, with various directions, θ. Some examples of the generated images are shown in **Fig. 2**.

The window size for the proposed method was fixed at $L = 10$. Two weight functions, $w_1(u, v)$ and $w_2(u, v)$, were used for obtaining the values of the parameters for each window. We set $(N_u, N_v) = (1, 1)$ for $w_1(u, v)$ and $(N_u, N_v) = (1, -1)$ for $w_2(u, v)$. **Figure 3** shows the appearances of the weight functions. As shown in the figures, the directions of the waves are orthogonal to each other. We computed the estimates as $\hat{\boldsymbol{q}} = A^{-}\boldsymbol{b}$, where A^{-} is the generalized inverse matrix of A.

The graph shown in **Fig. 4** shows the result. In the graph, the x-axis indicates the true values of θ, and the y-axis indicates the estimated values, $\hat{\theta}$. The colors, red, green, and blue, correspond to the SN ratios, ∞ dB, 60 dB, and 40 dB, respectively. As can be seen in the graph, the estimates were almost unbiased, though the variances of the estimates increased as the SN ratio decreased.

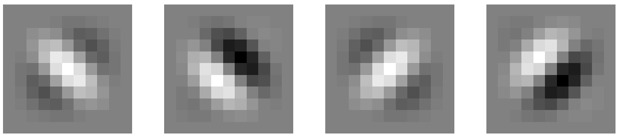
The performance of the estimation of σ^2 was then evaluated. Fixing the direction as $\theta = 0$ and the window size as $L = 10$, we generated a set of test images as shown in **Fig. 5** and estimated the scale, σ, for each of the test images. The performance was compared with a multi-scale analysis [8], which is not new but is one of the most reliable methods for the blurring scale estimation. In the multi-scale analysis, the input signal is iteratively blurred by the Gaussian function with a small scale for blurring the input signal with various scales, t. Then, the scale-normalized curvature, $\kappa(x, y; t)$, of the image profile is computed at each pixel at each blurring scale, and the scale, $\hat{t}(x, y)$, at which $|\kappa(x, y; t)|$ is local maximum with respect to the scale change is selected for estimating the blurring scale, $\hat{\sigma}^2(x, y) = \hat{t}(x, y)$. For computing the scale-space in the experiments, we employed the algorithm described in Ref. [8]: The solution of the heat equation for a given image is computed by convolving a 3 × 3 isotropic filter, iteratively. It is known that this computation method is time consuming but generates no false critical points [8].

Table 1 presents the results. As shown in Table 1, the performances of two methods were comparable. Though, as shown in **Table 2**, the computation times of the proposed method were much faster than those of the multi-scale analysis, which is very time-consuming because of the iterative convolution of the filter.



$\theta = 0$ $\theta = \pi/16$ $\theta = \pi/8$ $\theta = 3\pi/16$ $\theta = \pi/4$

Fig. 2 Examples of artificial images for evaluating the performance of the estimation of θ.



Real part Imaginary part Real part Imaginary part

$w_1(u, v)$ $w_2(u, v)$

Fig. 3 Two complex weight functions used in the experiments reported in this article.

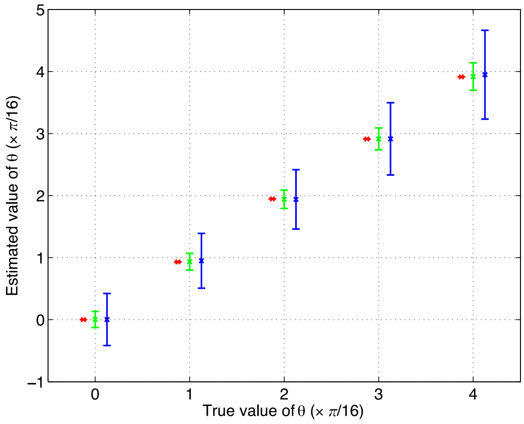
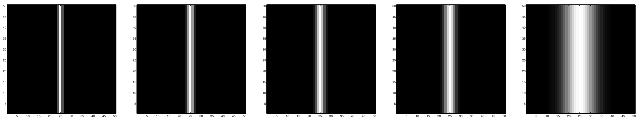


Fig. 4 Estimation accuracy of θ with respect to SN ratio of input images. Red: ∞ dB, Green: 60 dB, and Blue: 40 dB.



$\sigma = 2/3$ $\sigma = 1$ $\sigma = 4/3$ $\sigma = 5/3$ $\sigma = 5$

Fig. 5 Examples of artificial images for evaluating the performance of the estimation of σ.

Table 2 Comparision of computation time for processing 1,000 images.

	Computation Time
Multi-scale analysis [8]	137 min. 34.0 sec.
Proposed method	8 min. 43.2 sec.

Table 1 Performance comparison between the proposed method and the multi-scale analysis [8]. In the experiments, SNRs of the input images were ∞ dB and 40 dB, and the window size was $L = 10$. In each cell, Percent Bias (PB) [%] and Relative Standard Deviation (RSD) [%] are indicated.

		$L/\bar{\sigma} = 15$		$L/\bar{\sigma} = 10$		$L/\bar{\sigma} = 15/2$		$L/\bar{\sigma} = 6$		$L/\bar{\sigma} = 5$		$L/\bar{\sigma} = 30/7$	
		PB	RSD	PB	RSD	PB	RSD	PB	RSD	PB	RSD	PB	RSD
Multi-scale analysis [8]	∞ dB	6.07	0.00	0.00	0.00	1.55	0.00	1.00	0.00	0.00	0.00	0.51	0.00
	40 dB	3.68	10.80	1.65	6.35	1.03	5.05	0.52	4.39	0.37	3.97	0.25	3.59
Proposed method	∞ dB	−14.35	0.00	−3.52	0.00	−0.78	0.00	−0.36	0.00	−0.11	0.00	−0.32	0.00
	40 dB	−8.85	2.74	−2.10	1.39	−0.35	1.10	−0.24	1.06	−0.06	1.14	−0.32	1.31

	$L/\bar{\sigma} = 15/4$		$L/\bar{\sigma} = 10/3$		$L/\bar{\sigma} = 3$		$L/\bar{\sigma} = 30/11$		$L/\bar{\sigma} = 5/2$		$L/\bar{\sigma} = 30/13$		$L/\bar{\sigma} = 15/7$		$L/\bar{\sigma} = 2$	
	PB	RSD	PB	RSD	PB	RSD	PB	RSD	PB	RSD	PB	RSD	PB	RSD	PB	RSD
∞ dB	0.39	0.00	0.00	0.00	0.25	0.00	0.21	0.00	0.00	0.00	0.15	0.00	0.13	0.00	0.00	0.00
40 dB	0.22	3.41	0.21	3.24	0.23	2.99	0.18	2.81	0.08	2.71	0.12	2.66	0.15	2.48	0.07	2.34
∞ dB	−0.54	0.00	−0.27	0.00	−0.48	0.00	−0.48	0.00	−0.66	0.00	−0.68	0.00	−1.08	0.00	−0.80	0.00
40 dB	−0.51	1.54	−0.23	1.80	−0.43	2.14	−0.39	2.55	−0.57	2.30	−0.52	3.49	−0.87	3.98	−0.42	4.66

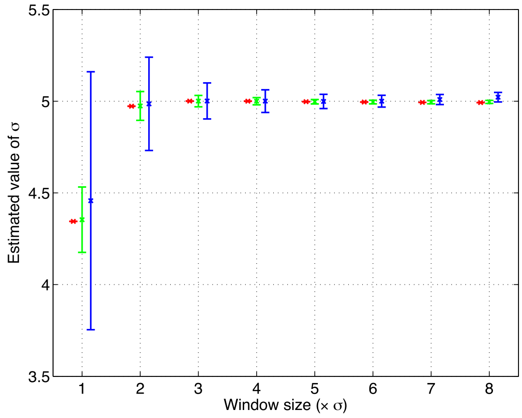


Fig. 6 Relationship between the estimaes of σ and the ratio, $L/\bar{\sigma}$ ($\bar{\sigma} = 5.0$ in the experiments). Red: ∞ dB, Green: 60 dB, and Blue: 40 dB.

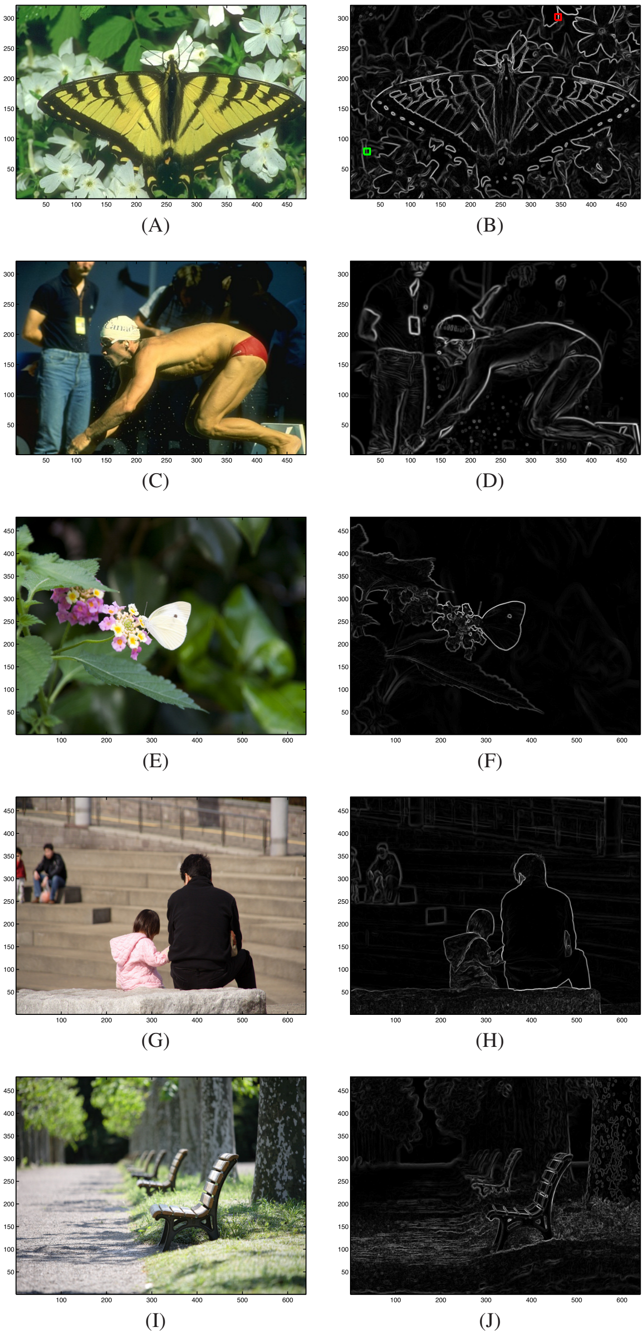

(A) (B) (C) (D) (E) (F) (G) (H) (I) (J)

Fig. 7 Examples of $I(x, y)$ and the input signal, $f(x, y)$.

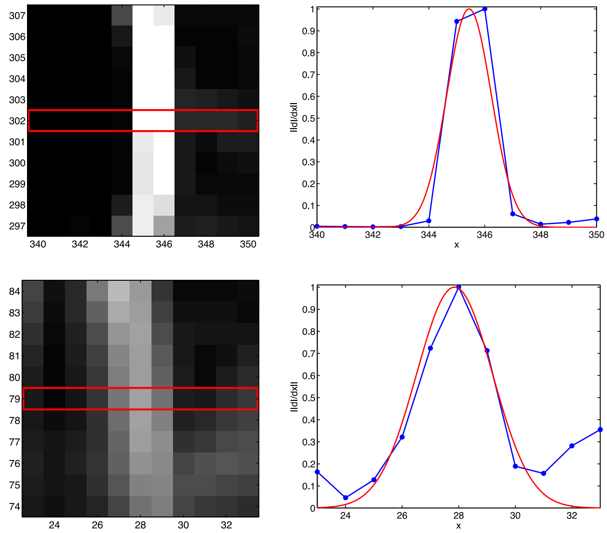


Fig. 8 Examples of $f(u, v)$ and the reconstructed signals. Top and bottom are the results obtained at the red and the green rectangles shown in Fig. 7 (B), respectively.

When the SN ratio of a given image is low, one can improve the accuracy of the estimates by applying the Gaussian smoothing filter with some scale $\tilde{\sigma}^2$ before the estimation. One can correct the resulting estimate, $\hat{\sigma}$, as $\hat{\sigma} \leftarrow (\hat{\sigma}^2 - \tilde{\sigma}^2)^{1/2}$.

The proposed method estimates the values of the parameters for $f(u, v)$ in the $L \times L$ window, Ω. We evaluated the performance of the scale estimation with respect to the window size, L. Fixing the scale of the input signals as $\bar{\sigma} = 5.0$, we evaluated the accuracies of the estimated scale, $\hat{\sigma}$, for each value of L. The graph shown in **Fig. 6** shows the results. The x-axis indicates the ratio, $L/\bar{\sigma}$, and the y-axis shows the estimated value of σ. As shown in the graph, the estimated scales were biased, when the window width, L, was comparable with the scale of the signal. The biases of the estimates decreased with respect to the increase of the window width. The variances of the estimates improved when larger size windows were used.

3.2 Experiments with Natural Images

The proposed method was applied to some natural images. Some examples of them are shown in **Fig. 7**. Given $f(x, y)$, the method estimates the set of parameters. Once you obtain the values of those parameters, you can reconstruct the profile of $f(x, y)$ at each location by substituting the estimated values to the model function. **Figure 8** shows examples of the input images, $f(x, y)$, and the reconstructed ones. The red curves accurately approximate the blue profiles, even though they are not the Gaussian curves, exactly. These results demonstrate the robustness of the proposed method. **Figure 9** shows the experimental results of the scale estimation. The estimated sclaes were larger in the backgrounds, in which the images were defocused.

4. Conclusion

A new method was proposed that accurately and efficiently estimates the parameters of the Gaussian model for describing local image profiles. The Gaussian function is non-linear with respect to the parameters to be estimated, and this fact makes it difficult to estimate them efficiently and accurately. The pro-

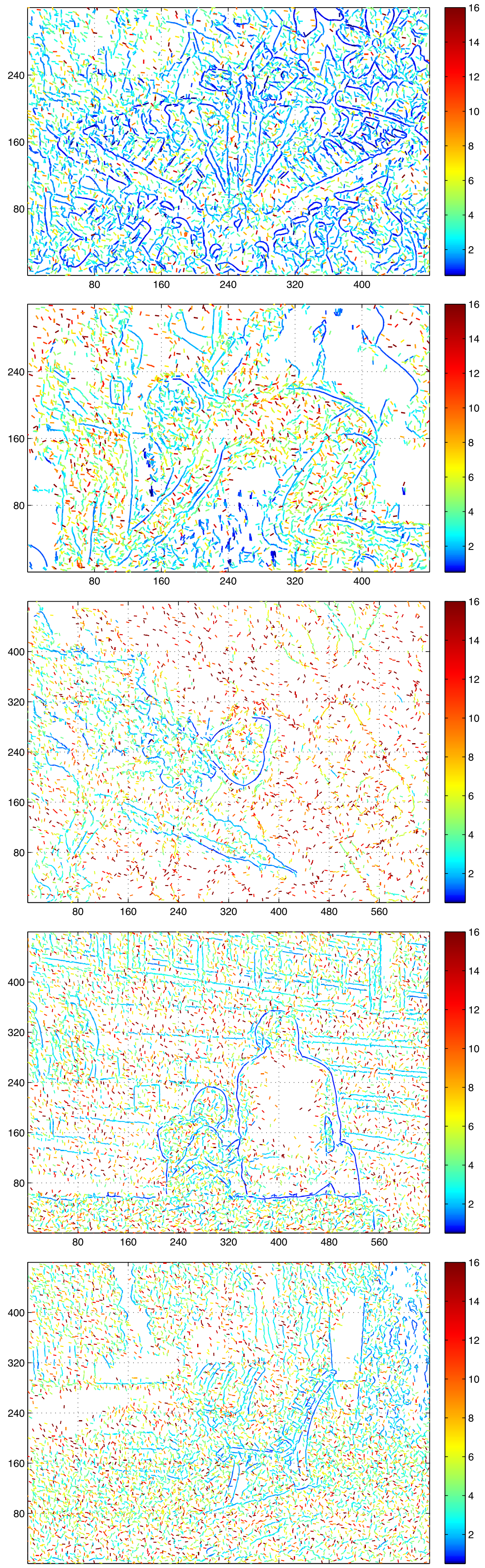


Fig. 9 The estimated scales at each location.

posed method employs the framework of the weighted integral method [1], which linearizes the parameter estimation problem based on the differential equations satisfied by the Gaussian function. Experimental results demonstrated that the estimation was as accurate as the state-of-the-art method and was much more efficient than the existing method. Future work includes fusing the estimated results for describing the entire image. As mentioned in this paper, the values of the parameters are estimated by computing $\hat{\boldsymbol{q}} = A^{-}\boldsymbol{b}$. It should be noted that this computation scheme has room for improvement: The components of A include errors generated by image noises, and the statistical distributions of the errors are anisotropic and inhomogeneous. The authors believe one can improve the estimation accuracy by applying, e.g., the renormalization method [7]. This improvement is also part of our future work.

References

[1] Ando, S. and Nara, T.: An exact direct method of sinusoidal parameter estimation derived from finite Fourier integral of differential equation, *IEEE Trans. Signal Processing*, Vol.57, No.9, pp.3317–3329 (2009).

[2] Aster, R., Borchers, B. and Thurber, C.: *Parameter Estimation and Inverse Problems*, Academic Press (2012).

[3] Bae, S. and Durand, F.: Defocus magnification, *Computer Graphics Forum*, Vol.26, No.3, pp.571–579 (2007).

[4] Canny, J.: A computational approach to edge detection, *PAMI*, Vol.PAMI-8, No.6, pp.679–698 (1986).

[5] Chakrabarti, A., Zickler, T. and Freeman, W.: Analyzing spatially-varying blur, *CVPR*, pp.2512–2519 (2010).

[6] Favaro, P. and Soatto, S.: A geometric approach to shape from defocus, *PAMI*, Vol.27, No.3, pp.406–417 (2005).

[7] Kanatani, K.: Renormalization for unbiased estimation, *ICCV*, pp.599–606 (1993).

[8] Lindeberg, T.: *Scale-Space Theory in Computer Vision*, Kluwer Academic Publishers (1994).

[9] Perona, P. and Malik, J.: Scale-space and edge detection using anisotropic diffusion, *PAMI*, Vol.12, No.7, pp.629–639 (1990).

[10] Rajagopalan, A., Chaudhuri, S. and Mudenagudi, U.: Depth estimation and image restoration using defocused stereo pairs, *PAMI*, Vol.26, No.11, pp.1521–1525 (2004).

[11] Rhemann, C., Rother, C., Kohli, P. and Gelautz, M.: A spatially varying PSF-based prior for alpha matting, *CVPR*, pp.2149–2156 (2010).

[12] Zhu, S., Shi, K. and Si, Z.: Learning explicit and implicit visual manifolds by information projection, *Pattern Recogn. Lett.*, Vol.31, No.8, pp.667–685 (2010).

(Communicated by *Mitsuru Ambai*)