immediate area is the shape from texture processing where the boundaries obtained from the texture segmentation may be used in localizing the process of surface geometry extraction. A feedback from the shape from texture algorithms may in turn provide useful information about possible borders between textured regions.

ACKNOWLEDGMENT

The implementation of the algorithm described in Sections III-A and B was partly done by Y. Lee Hoffman. S. Howden has also helped in parts of the implementation.

REFERENCES

[1] J. Beck, K. Prazdny, and A. Rosenfeld, "A theory of textural segmentation," *Human and Machine Vision*. J. Beck, B. Hope, and R. Rosenfeld, Eds. New York: Academic, 1983.
[2] J. Beck, A. Sutter, and R. Ivry, "Spatial frequency channels and perceptual grouping in texture segregation," *Comput. Vision, Graphics, Image Processing*, vol. 37, pp. 299-325, 1987.
[3] D. Blostein and N. Ahuja, "Representation and three-dimensional interpretation of image texture: An integrated approach," in *Proc. First Int. Conf. Computer Vision*, 1987, pp. 444-449.
[4] P. Brodatz, *Textures: A Photographic Album for Artists and Designers*. New York: Dover, 1966.
[5] M. Clark and A. C. Bovik," Texture segmentation using Gabor modulation/demodulation," *Pattern Recognition Lett.*, vol. 6, pp. 261-267, Sept. 1987.
[6] J. M. Coggins and A. K. Jain, "A spatial filtering approach to texture analysis," *Pattern Recognition Lett.*, vol. 3, pp. 195-203, 1985.
[7] G. C. Cross and A. K. Jain, "Markov random field texture models," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-5, pp. 25-39, 1983.
[8] L. S. Davis, M. Clearman, and J. K. Aggarwal, "An empirical evaluation of generalized cooccurrence matrices," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-3, pp. 214-221, 1981.
[9] A. Gagalowicz, "Blind texture segmentation," in *Proc. 9th Int. Conf. Pattern Recognition*, Rome, Italy, Nov. 1988, pp. 46-50.
[10] R. M. Haralick, "Statistical and structural approaches to texture," *Proc. IEEE*, vol. 67, pp. 786-804, 1979.
[11] M. K. Hu, "Visual pattern recognition by moment invariants," *IRE Trans. Inform. Theory*, vol. IT-8, pp. 179-187, 1962.
[12] B. Julesz, "Texton gradients: The texton theory revisited," *Biol. Cybern.*, vol. 54, pp. 245-251, 1986.
[13] ——, "Textons, the elements of texture perception, and their interactions," *Nature*, vol. 290, pp. 91-97, 1981.
[14] ——, "Visual pattern discrimination," *IRE Trans. Inform. Theory*, vol. IT-8, pp. 84-92, 1962.
[15] B. Julesz, E. N. Gilbert, L. A. Shepp, and H. L. Frisch, "Inability of humans to discriminate between visual textures that agree in second-order statistics—Revisited," *Perception*, vol. 2, pp. 391-405, 1973.
[16] R. L. Kashyap and R. Chellappa, "Estimation and choice of neighbors in spatial interaction models of images," *IEEE Trans. Inform. Theory*, vol. 29, pp. 60-72, 1983.
[17] R. L. Kashyap, R. Chellappa, and N. Ahuja, "Decision rules for the choice of neighbors in random field models of images," *Comput. Graphics Image Processing*, vol. 15, pp. 301-318, 1981.
[18] D. Marr, *Vision*. San Francisco, CA: Freeman, 1982.
[19] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C*. New York: Cambridge University Press, 1988.
[20] T. C. Rearick, "A texture analysis algorithm inspired by a theory of preattentive vision," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, San Francisco, CA, 1985, pp. 312-317.
[21] A. Rosenfeld, R. Hummel, and S. Zucker, "Scene labeling by relaxation operations," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-6, pp. 420-433, 1976.
[22] M. I. Shamos and D. Hoey, "Closest-point problems," in *Proc. 16th Annu. Symp. Foundations of Computer Science*, 1975, pp. 131-162.
[23] M. Tuceryan and N. Ahuja, "Extraction of early perceptual structure in dot patterns: Integrating region, boundary, and component gestalt," *Comput. Vision, Graphics, Image Processing*, Dec. 1989.
[24] M. R. Turner, "Texture discrimination by Gabor functions," *Biol. Cybern.*, vol. 55, pp. 71-82, 1986.
[25] H. Voorhees and T. Poggio, "Detecting textons and texture boundaries in natural images," in *Proc. First Int. Conf. Computer Vision*, London, 1987, pp. 250-258.
[26] G. Voronoi, "Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Deuxième mémoire: Recherches sur les parallélloèdres primitifs," *J. Reine Angew. Math.*, vol. 134, pp. 198-287, 1908.
[27] H. B. Wilson, Jr. and D. S. Farrior, "Computation of geometrical and inertial properties for general areas and volumes of revolution," *Comput. Aided Design*, vol. 8, no. 4, pp. 257-263, 1976.
[28] S. W. Zucker and R. A. Hummel, "On the foundations of relaxation labeling processes," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-5, pp. 267-287, 1983.

A Fast Parallel Algorithm for Blind Estimation of Noise Variance

PETER MEER, JEAN-MICHEL JOLION, AND AZRIEL ROSENFELD

***Abstract*—The only information available to a blind noise variance estimation algorithm is the corrupted image and the white nature of the zero mean Gaussian noise. The proposed algorithm recovers the variance of the noise in two steps. First, the sample variances are computed for square cells tessellating the noisy image. Several tessellations are applied with the size of the cells increasing fourfold for consecutive tessellations. The four smallest sample variance values (the outcomes of the first four order statistics) are retained for each tessellation and combined through an outlier analysis into one estimate. The different tessellations thus yield a variance estimate sequence. In the second part of the algorithm, the value of the noise variance is determined from this variance estimate sequence. We have applied the blind noise variance algorithm to 500 noisy 256 × 256 images derived from seven prototypes of classes often employed in computer vision and image processing. In 98 percent of the cases the relative estimation error was less than 0.2 with an average error of 0.06. Application of the algorithm to differently sized images is also discussed. All the operations in the algorithm are parallel and if they are implemented on an image pyramid, the variance of the noise is recovered in $O[\log(\text{image\_size})]$ processing time.**

***Index Terms*—Image pyramids, noise estimation, order statistics.**

I. INTRODUCTION

Let f be an image defined on an $N \times N$ square lattice. The pixels have values between 0 and (say) 255. Let w be samples of a stationary, zero mean, normal, white noise source independent of f, with variance σ_w^2. When the image f is additively corrupted by w the noisy image g is obtained. The following *blind noise variance estimation* problem is of importance in computer vision and image processing:

> Estimate σ_w^2, the variance of the noise, from the noisy image $g(m, n)$ without having access to *a priori* information about the original image $f(m, n)$.

In this correspondence we present a fast parallel algorithm for the blind noise variance estimation problem. In most of our tests of

Manuscript received September 1, 1988; revised May 15, 1989. Recommended for acceptance by A. K. Jain.

P. Meer and A. Rosenfeld are with the Center for Automation Research, University of Maryland, College Park, MD 20742.

J.-M. Jolion is with the Laboratoire d'Informatique Graphique et d'Intelligence Artificielle, Université Claude Bernard, Lyon I, 69622 Villeurbanne Cedex, France.

IEEE Log Number 8932043.

this algorithm the estimation error was below 10 percent. In a parallel implementation the estimate is obtained after a number of processing steps on the order of log[image_size].

The precise value of the noise variance is required in many applications in which algorithms are employed that have the variance as a parameter. Image restoration by least squares (Wiener filtering) [14, vol. 1], smoothing with Kalman filters [1], and optimal threshold selection [15] are among such applications.

The noise is independent of the original image $f(m, n)$ and for any region in the noisy image $g(m, n)$ we have

$$s_g = s_f + s_w \tag{1}$$

where the different s are the sample variances computed for that region. In blind noise variance estimation we are given a set of values of s_g (or of some function of it) and must find σ_w^2 without knowing the corresponding s_f. The discussion of the relation between s_w and σ_w^2 can be postponed for the moment. In order to estimate the variance of the noise, the ensemble of s_g values must be separated into signal, s_f and noise, s_w components. The dichotomy is achieved by noticing that uniform regions in the image contribute only to the s_w term in (1), while local changes in gray level (e.g., edges) influence both terms.

To estimate the variance of the noise, Canny [5] separated the signal and noise components at the output of his edge detector. The lower 80th percentile in the amplitude histogram of the edge detector responses was taken as the contribution from noise and employed in the estimation. Voorhees and Poggio [16] fitted a Rayleigh distribution to the histogram of the image gradient magnitude. The mode of the distribution is then the noise variance estimate. Meer *et al.* [9] made use of the orthogonality properties of their template edge detection masks to obtain the noise variance estimate from regions declared as uniform. Besl and Jain [2] discriminated uniform patches in an image by local planar fits and computed the average variance over these regions.

The accuracy of the estimates is not available for most of the above mentioned methods. Their dependence on discriminating features (edges and/or uniform patches) in the noisy image, prior to the estimation, introduces limitations. When a small noise variance has to be estimated from an image with severe fluctuations, usually a residual component from the image is incorporated into the value of the estimate. At low signal-to-noise ratios (the noise is comparable with the image fluctuation), edge detection and surface fitting are no longer robust operations, and the regions discriminated as uniform, or the histogram of the edge detector magnitudes, may not be reliable. Whenever the estimation procedure requires the storage of a global histogram, parallel and therefore fast implementations are not feasible.

In the new method we overcome these difficulties. Only local statistics are employed and there is no need for a large memory. All the computations are recursive and can be implemented in parallel on an image pyramid, allowing the estimation procedure to be accomplished in $O[\log(\text{image\_size})]$ processing steps. We present estimation results for hundreds of noisy images derived from several pictures representing frequently met classes in computer vision and image processing. The estimates are obtained for a large range of noise variance values, from no noise to signal-to-noise ratios below one.

In Section II the new noise variance estimator is introduced and its behavior is analyzed. In Section III the estimation algorithm is described. The experimental results are presented in Section IV. In Section V the computational complexity of the image pyramid based implementation of the algorithm and its limitations are discussed.

II. The Noise Variance Estimator

Without loss of generality we can take the size of the image $N \times N$ to be a power of 2, i.e., $N = 2^n$. The image is tessellated into square cells of size $c_l \times c_l$. It is convenient to have the cells' side length increase twofold for consecutive tessellations: $c_l = 2^l$, $l = 1, 2, \cdots, n$. The hierarchy of such tessellations defines an *image pyramid*, a frequently employed tool in multiresolution image understanding [13]. We describe the image pyramid in detail in Section V-A when the computational complexity of the algorithm is assessed. The tessellation of the image with $2^l \times 2^l$ cells will be referred to as level l.

The cells belonging to level l are located within the image with index $k_l = 1, 2, \cdots, K_l$ where $K_l = 4^{n-l}$. The unbiased estimate of the variance, the sample variance $s_g(k_l)$, computed for the kth cell is

$$s_g(k_l) = \frac{1}{4^l - 1} \sum_{i=1}^{2^l} \sum_{j=1}^{2^l} \left[g_{k_l}(i, j) - \bar{g}_{k_l} \right]^2 \tag{2}$$

where the subscript k_l means that the computation is restricted to the k_lth cell. Similarly to (1) we have

$$s_g(k_l) = s_f(k_l) + s_w(k_l). \tag{3}$$

As the noise variance estimator for the tessellation of the noisy image representing level l, we employ the smallest value of the sample variance $s_g(k_l)$:

$$q(l, 1) = \min_{k_l} s_g(k_l) \tag{4}$$

where the second parameter, set to 1, emphasizes that the first order statistic is the one computed. The significance of this parameter will become clear later in this section. The contribution of the signal component $s_f(k_l)$ is the minimized term in (4) because the noise component $s_w(k_l)$ is a quasi-constant function of the noise variance σ_w^2. The cell providing $q(l, 1)$ thus is one of the cells with the most uniform gray level distribution in the original image.

An important property of the noise variance estimate $q(l, 1)$ is that its value increases monotonically with the level l for the tessellation sizes we are concerned with. This monotonic behavior of the first order statistic is a consequence of the well known fact that the sample variance is a consistent estimator, i.e., its spread (confidence interval) decreases with the increase of the degrees of freedom of the sample. For a uniform field, the larger the data set from which the sample variance is computed, the closer to the real variance of the image must the obtained value of the first order statistic be. Thus, for a sufficiently large increase in the cell size, the outcomes of the first order statistic must increase too with probability one. The behavior of the estimator (4) in the case of a uniform field is analyzed in detail in [9]. In the case of a real image, a larger cell incorporates more changes, further increasing $q(l + 1, 1)$ relative to $q(l, 1)$.

For a given level, the first order statistic of the sample variance alone cannot be a robust estimator of the noise variance. Its outcome may be an *outlier*, a value far away from the mode of the distribution, having a very small, but nonzero probability. The outliers will strongly underestimate the noise variance and should be discarded. The effect is especially important for the finer tessellations where the number of cells K_l is large.

We improve the robustness of the estimation procedure by retaining for a level not only the first but the first four order statistics, that is, the smallest four values of the sample variance. The higher the order of a statistic the lower is the probability that its value is still an outlier and therefore cannot contribute to a reliable estimate. We have found experimentally that employing more than the first four order statistics does not yield further improvement on the accuracy of the variance estimate $v(l)$. The amount of computation and the data flow in the algorithm is kept minimal by fewer order statistics.

Let the obtained outcomes be $q(l, 1), q(l, 2), q(l, 3), q(l, 4)$. We can now perform a set of *slippage tests* to investigate whether or not $q(l, i)$, $i = 1, 2, 3$ are outliers. The outliers are discarded and the remaining values averaged. The final value of the estimate is denoted $v(l)$.

We employed Dixon's r statistics for the slippage tests [6]. These statistics are ratios of differences of order statistics and are com-

pared to threshold values:

$$r_0(l, 1) = \frac{q(l, 2) - q(l, 1)}{q(l, 4) - q(l, 1)} \qquad r_0(l, 2) = \frac{q(l, 3) - q(l, 2)}{q(l, 4) - q(l, 2)}$$

$$r_0(l, 3) = \frac{q(l, 4) - q(l, 3)}{q(l, 4) - q(l, 2)} = 1 - r_0(l, 2). \tag{5}$$

The following tests were employed:

$$\text{if } r_0(l, 1) \le 0.5 \text{ then } v(l) = \frac{1}{4}\sum_{i=1}^{4} q(l, i) \tag{6}$$

else

$$\text{if } r_0(l, 2) \le 0.7 \text{ then } v(l) = \frac{1}{3}\sum_{i=2}^{4} q(l, i) \tag{7}$$

else

$$\text{if } r_0(l, 3) \le 0.7 \text{ then } v(l) = \frac{1}{2}\sum_{i=3}^{4} q(l, i) \tag{8}$$

else

$$v(l) = q(l, 4). \tag{9}$$

The thresholds correspond to a 0.25 error probability of not detecting an outlier [3]. Their values, as long as they do not yield a too low or too high error probability, have only weak influence on the overall performance of the algorithm.

For the very coarse tessellations $l = n - 1$, n the slippage tests cannot be applied. In these cases we have only four samples and one sample, respectively. The value of $v(l)$ is taken as the average of the four order statistic outcomes ($l = n - 1$), or equal to the global sample variance of the noisy image ($l = n$).

Thus the following operations are performed on the noisy image to obtain the *sequence of variance estimates.*

- The $2^n \times 2^n$ noisy image is tessellated with cells of size $2^l \times 2^l$, $l = 2, 3, \cdots, n$.
- For each level l:
  1) the sample variance is computed at every cell, (2);
  2) the four smallest values (except for $l = n$) are retained;
  3) a set of slippage tests (except for $l = n - 1, n$) are performed, and the final value $v(l)$ of the variance estimate is computed, (5)-(9).

The obtained sequence of variance estimates, $v(l)$, $l = 2, 3, \cdots, n$, is the input into the blind noise variance estimation algorithm. It preserves the monotonicity property of the first order statistics discussed at the beginning of the section.

The variance estimate retained for a given level is the result of nonlinear operations on the first four order statistics of the sample variances. In the noise variance estimation algorithm we will employ the ratio of consecutive variance estimates and will make use of a lower bound on the values of these ratios. In Table I we present the experimentally obtained smallest values of $v(l - 1)/v(l)$ for different image sizes: 64×64, 128×128, 256×256. The uncorrupted uniform field has 128 as gray level. Fourteen instances of noise variance ranging from 3^2 to 58^2, with five different random fields per variance value (a total of 70 cases), were employed. The average values of the ratios are also given.

The experimental ratios are slightly larger than the theoretically found bounds for the ratios of first order statistics [9]. In the blind noise variance estimation algorithm we employ the bounds $\beta(l)$ generated by the expression:

$$\beta(l) = 1 - 0.1 \cdot 2^{-l+6}. \tag{10}$$

The values of $\beta(l)$ are given in Table II.

TABLE I
EXPERIMENTAL BOUNDS ON THE RATIO $v(l-1)/v(l)$ FOR THE UNIFORM FIELD IMAGE

		Level l					
Size	Value	3	4	5	6	7	8
64×64	Minimum	0.323	0.660	0.833	0.998		
	Average	0.489	0.780	0.902	1.000		
128×128	Minimum	0.332	0.634	0.831	0.921	0.999	
	Average	0.424	0.739	0.892	0.954	1.000	
256×256	Minimum	0.263	0.639	0.827	0.922	0.961	1.000
	Average	0.370	0.710	0.875	0.944	0.976	1.000

TABLE II
THE EMPLOYED BOUNDS $\beta(l)$

l	3	4	5	6	7	8	9
$\beta(l)$	0.2	0.6	0.8	0.9	0.95	0.975	0.9875

III. The Blind Noise Variance Estimation Algorithm

An efficient blind noise variance estimation algorithm should return the correct noise variance for a large range of values. This includes the case of an uncorrupted image when we expect to obtain zero as the estimated variance of the noise. To obtain the sequence of estimates, at every level the algorithm automatically selects the sample variances obtained in the cells with most uniform pixel values in the original image. For a noiseless image, the variations seen by the selected cells are the smallest ones at that tessellation. Thus the noise variance estimate cannot be less than these residual signal components. The following rule detects that an image is not corrupted by noise.

Rule 1:

$$\text{if } \exists l \ge 2 \text{ such that } q(l, 4) < 1 \text{ then } \hat{\sigma}_w^2 = 0 \tag{11}$$

where $\hat{\sigma}_w^2$ is the estimated noise variance value, the output of the algorithm. Erroneous decisions for small noise variances are avoided by comparing the fourth order statistic to a minimum variation threshold.

In our blind noise variance estimation algorithm the dichotomy between the contribution of the signal and that of the noise is achieved by analyzing the shape of the *deviation sequence* computed as:

$$\alpha(l) = \frac{v(l-1)}{v(l)} - \beta(l) \qquad l = 3, 4, \cdots, n \tag{12}$$

where $\beta(l)$ are the lower bounds obtained for the *uniform-field* (Table II). Since $v(l-1) < v(l)$ the values of $\alpha(l)$ are always between -1 and 1. Tessellations at lower levels employ small cells and at least a few of them contain quasi-uniform regions. For quasi-uniform regions, $\alpha(l)$ is positive or has very small negative values since $\beta(l)$ is a lower bound for uniform regions.

As the cell size increases (l increases) the sample variances are computed for larger and larger regions of the image. For arbitrary images, the larger the region the less constant its gray level distribution should be. Depending on the original image, at a given level l_o, cells with almost constant pixel distribution no longer can be found. All the sample variance values increase steeply, yielding a much larger estimate $v(l_o)$ and $-1 < \alpha(l_o) \ll 0$. The noise-signal dichotomy thus can be achieved by detecting in the deviation sequence the level where the value of $\alpha(l)$ becomes significantly negative.

To avoid false alarms for images with a not well defined noise-signal separation (to be discussed below) a cumulative thresholding

technique is employed:

$$\text{for } l = 3, 4, \cdots, n \qquad l_u = \min_l \arg\left[\alpha(l) < 0\right]$$

$$l_o = \min_l \arg\left[\sum_{i=l_u}^{l} \alpha(l) < T\right]$$

$$\text{for } l_0 = 3, 4, \cdots, n-2 \qquad \sum_{i=l_u}^{l_o+1} \alpha(l) < T. \tag{13}$$

The threshold T is equal to -0.1, a small negative number. Its precise value does not effect the performance of the algorithm. The estimates at levels $l < l_0$ belong to the *noise domain*, and the ones at levels $l \geq l_o$ to the *signal domain*.

In Fig. 1 examples of deviation sequences are shown. The images are assumed to have size 256×256 and thus $n = 8$. In the case of a *uniform field* [Fig. 1(a)] the $\alpha(l)$ values decrease with the level but remaining always positive. The noise domain spans all the levels. The estimation of the variance of a normally distributed stationary random field thus can be accomplished by applying the following rule.

Rule 2:

$$\text{if } l_o = n \text{ then } \hat{\sigma}_w^2 = v(n). \tag{14}$$

Note that the variance of the *uniform field* is recovered without the knowledge that there is no variation present in the original image.

For an image with a good noise–signal separation [Fig. 1(b)] the noise domain is more extended than the signal domain. The transition between the two domains is sharp, $l_o = l_u$, and occurs at a high level yielding reduced spread of the sample variances. The outcomes of the order statistics are then close to the correct noise variance value and an accurate $\hat{\sigma}_w^2$ can be obtained.

In the moderate noise–signal separation case [Fig. 1(c)] the signal domain is more extended than the noise domain. The transition between the domains is not well defined ($l_o = l_u + 1 = 5$ in the example) and occurs at intermediate levels. The estimates in the noise domain are obtained from ensembles with larger spreads.

An example of poor noise–signal separation is shown in Fig. 1(d). The noise domain is reduced to level 3, the first one for which α value is available. At every level all the cells see regions in the original image with significant variations. The algorithm should recognize the case of poor noise–signal separation ($l_o \leq 4$) and warn about its incompetence to return an accurate estimate. This is achieved by the following two rules.

Rules 3 and 4:

$$\text{if } l_o = 3 \text{ or } 4 \text{ then 'WARNING: CANNOT ESTIMATE!'} \tag{15}$$

The busyness of the signal, i.e., how much gray level variation is present per unit area [14, vol. 2], is an important parameter of our blind noise variance estimation algorithm. The more busy the signal, the less good is the separation between the two domains. Information about the busyness of the original image is obtained from the signal domain. If the signal domain spans several levels and its sequence of variance estimates increase steeply, the image is probably a busy one. The assumption appears to hold for real scenes, as will be proved by our experimental results. In the rules applied for $5 \leq l_o < n$ we employ a parameter derived from the signal domain to characterize the busyness of the original image.

While Rules 1 to 4 are independent of the image size, the interpolation rules employed when $5 \leq l_o < n$ do depend on it. In what follows we describe these rules for 256×256 images, $n = 8$. The cases of 128×128 and 512×512 images are discussed in Section V-B.

A. Transition at Level $l_o = 5$

The busyness of the signal is characterized by the value of the parameter

$$\rho = \alpha(5) + \alpha(6) \qquad -2 < \rho \leq T = -0.1. \tag{16}$$

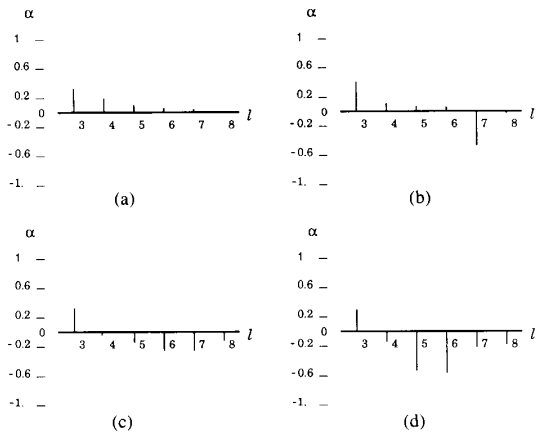

Fig. 1. Examples of deviation sequences. (a) Uniform field. (b) Good noise–signal separation. (c) Moderate noise–signal separation. (d) Poor noise–signal separation.

More negative values of ρ indicate more variation in the original image and the estimates belonging to the noise domain are then more contaminated by residual signal variations. The noise variance is underestimated at the lower levels; however, the residual signal components compensate for the underestimation and interpolation between two such levels yields the correct estimate.

The $(-2, T\,]$ range of ρ is partitioned into four regions marked by the index $i = 1, 2, 3, 4$ with ρ_{i-1} and ρ_i being the bounds of the ith region. The first region, $i = 1$, corresponds to the busiest signal, ρ having the smallest negative values. The noise and the signal components of the variance (3) are strongly combined. The output of the algorithm is as follows.

Rule 51:

$$\text{for } \rho_0 = -2 < \rho \leq \rho_1 = -1.5 \quad \hat{\sigma}_w^2 = v(3). \tag{17}$$

The contamination by the signal is the weakest at level 3 and thus the value allocated to $\hat{\sigma}_w^2$ in (17) is the most "reliable" one. Such busy images are seldom met in the classes to which the algorithm is applicable. In fact in our experiments Rule 51 was never employed.

In the other three regions linear interpolation is employed. First the interpolation variable is computed:

$$\text{for } \rho_{i-1} \leq \rho \leq \rho_i \qquad \delta = \frac{\rho - \rho_i}{\rho_{i-1} - \rho_i}. \tag{18}$$

The explicit interpolation rules are given below.

Rule 52:

$$\text{for } \rho_1 = -1.5 \leq \rho \leq \rho_2 = -1$$

$$\hat{\sigma}_w^2 = \delta \cdot v(3) + (1 - \delta) \cdot v(4). \tag{19}$$

Rule 53:

$$\text{for } \rho_2 = -1 \leq \rho \leq \rho_3 = -0.5$$

$$\hat{\sigma}_w^2 = \delta \cdot v(4) + (1 - \delta) \cdot v(5). \tag{20}$$

Rule 54:

$$\text{for } \rho_3 = -0.5 \leq \rho \leq \rho_4 = T$$

$$\hat{\sigma}_w^2 = \delta \cdot v(5) + (1 - \delta) \cdot v(6). \tag{21}$$

These interpolation rules are identical and the interpolation is continuous across the whole range of ρ.

B. Transition at Level $l_o = 6$ or 7

When the transition occurs at level 6 or 7, the variance estimates at the higher levels (but still in the noise domain) are close to the

correct value of the variance, being computed from larger samples. To characterize the busyness of the signal the two α's at the transition between the noise and signal domains are employed. Their ranges, $T < \alpha(l_o - 1) < 1$ and $-1 < \alpha(l_o) \leq T$, are partitioned into two regions. The threshold T has the same -0.1 value.

The parameter from the noise domain, $\alpha(l_o - 1)$, is taken into consideration first. In its first region $\alpha(l_o - 1)$ takes only negative values corresponding to a "blurred" transition between the noise and signal domains. As in the case of $l_o = 5$, we again interpolate based on the busyness of the signal between two levels in the noise domain, leaving the residual signal component to compensate for the underestimation. As interpolation variable $\delta = |\alpha(l_o)|$ is employed.

Rule 61:

$$\text{for} \quad T < \alpha(l_o - 1) \leq 0$$

$$\hat{\sigma}_w^2 = \delta \cdot v(l_o - 2) + (1 - \delta) \cdot v(l_o - 1). \quad (22)$$

The more busy is the signal, i.e., the more negative $\alpha(l_o)$ is, the closer to $v(l_o - 2)$ is $\hat{\sigma}_w^2$.

In the second region, $\alpha(l_o - 1)$ takes positive values. This region corresponds to a well-defined transition between the noise and signal domains. Let the two regions of the parameter $\alpha(l_o)$ be bounded by α_{i-1} and α_i, $i = 1, 2$. Similarly to (18) the interpolation variable δ is

$$\text{for} \quad 0 \leq \alpha(l_o - 1) < 1 \quad \text{and}$$

$$\alpha_{i-1} \leq \alpha(l_o) \leq \alpha_i \qquad \delta = \frac{\alpha(l_o) - \alpha_i}{\alpha_{i-1} - \alpha_i}. \quad (23)$$

The interpolation is achieved by the following rules.

Rule 62:

$$\text{for} \quad \alpha_0 = -1 < \alpha(l_o) \leq \alpha_1 = -0.5$$

$$\hat{\sigma}_w^2 = \delta \cdot v(l_o - 2) + (1 - \delta) \cdot v(l_o - 1). \quad (24)$$

Rule 63:

$$\text{for} \quad \alpha_1 = -0.5 \leq \alpha(l_o) \leq \alpha_2 = T$$

$$\hat{\sigma}_w^2 = 0.5\left[(1 + \delta) \cdot v(l_o + 1) + (1 - \delta) \cdot v(l_o)\right]. \quad (25)$$

The interpolation is continuous across the regions of $\alpha(l_o)$. The largest $\hat{\sigma}_w^2$ value cannot exceed $0.5[v(l_o - 1) + v(l_o)]$, still in the noise domains. At the higher levels there is no need to go forward into the signal domain; the estimate values at the transition are already reliable.

Rules 3 and 4 (15) warn about the incompetence of the algorithm to return an accurate output. While this warning is necessary, sometimes the correct noise variance can be obtained if the algorithm is applied a second time with a smaller threshold value T_1. For the few 256×256 images in which $l_o = 3$ or 4, changing the threshold to $T_1 = -0.2$ was successful in our experiments. A rule employed in this second iteration has its number supplemented with 1 on the left. Thus in the experimental results we can have Rules 4 and 14, or 52 and 152. If in the second iteration the transition is still at level 3 or 4, the value $v(3)$ is the output of the algorithm.

To conclude this section, we summarize the steps of the blind noise estimation algorithm, *after* the variance estimate sequence $v(l)$ has been obtained:

- Compute the deviation sequence $\alpha(l)$ (12).
- Find the level of transition l_o between the noise and signal domains by cumulative thresholding, (13).
- Apply the correct rule (14) to (25).
- Optional: If $l_o = 3$ or 4, lower the threshold and repeat the above procedure.

The rules employed in the blind noise variance estimation algorithm take into account all the available information: the level at which the transition between the noise and signal domains occurs, and the cue about the amount of variation present in the signal. In the following section we demonstrate the effectiveness of the algorithm. On a set of 500 noisy images derived from seven classes, the algorithm returned a variance within a 20 percent relative error in more than 98 percent of the cases. The average relative error was 6 percent.

IV. Experimental Results

To assess the efficiency and robustness of the blind noise variance estimation algorithm we ran it on several 256×256 noisy images derived from seven prototype images belonging to different classes often employed in image processing and computer vision: *uniform field*, *binary*, *parts*, *portrait*, *outdoor*, *aerial*, and *texture*. The images are ranked on a scale of increasing structural complexity, i.e., decreasing separation between the noise and signal domains in their noise corrupted versions. All the images were defined on a 256 gray level scale and were corrupted additively with zero mean, normal noise having one of fourteen variance values between 3^2 and 58^2. For each noise variance value five noisy images were generated. The noise variance value was also estimated in the noiseless case, $\sigma_w^2 = 0$. Thus in total 71 different instances were measured for each prototype image. Two parameters were used to assess the accuracy of the algorithm's output the *relative error* ϵ and the *signal-to-noise ratio* λ:

$$\epsilon = \frac{\left|\sigma_w^2 - \hat{\sigma}_w^2\right|}{\sigma_w^2} \qquad \lambda = \frac{\sigma_f^2}{\sigma_w^2} \quad (26)$$

where σ_w^2 is the variance of the corrupting noise, $\hat{\sigma}_w^2$ the variance estimated by the algorithm, and σ_f^2 is the variance of the original image. In the noiseless case $\epsilon = \sigma_w^2$. For the largest σ_w^2 values employed the signal-to-noise ratio was less than 1 for all the images.

We present now the performance of the algorithm for each of the seven prototype images. A more detailed analysis can be found in [9].

All the pixels in the uncorrupted *uniform field* image have their gray level value equal to 128. The obtained relative error was in 97.18 percent of the cases less than 0.05, and no error above 0.1 was obtained.

In the *binary* image [Fig. 2(a)] only two gray level values are present, the object having 75 and the background 175. The mean gray level value is $\mu_f = 162.6$ and the variance $\sigma_f^2 = 32.9^2$. The uncorrupted *binary* image has large uniform regions and the sharp transition between the noise and signal domains always occurs at level 6 or 7. The Rule 61 is never employed. No ϵ value larger than 0.1 was obtained.

The *parts* (industrial-type) image [Fig. 2(b)] is a gray level image with mean $\mu_f = 132.93$ and variance $\sigma_f^2 = 35.87^2$. The background is no longer uniform but has a slowly changing illumination pattern and also contains shadows of the objects. The noise and signal domain are well separated, with the transition between them appearing in most of the cases at level 6. The transition, however, is less sharp than in the *binary* image and Rule 61 was also employed. The relative error for the *parts* image was always less than 0.2.

The *portrait* image [Fig. 2(c)] is often employed in image processing. The image has mean gray level value $\mu_f = 99.18$ and variance $\sigma_f^2 = 52.31^2$. The image contains much more detail than the *parts* image, but the scale of the details is relative large and regions with quasi-constant gray level values can still be found. The transition between the noise and signal domains occurs at level 5 in most of the cases. Only one image out of 71 had ϵ between 0.2 and 0.25.

The *outdoor* image [Fig. 2(d)] is a representative of the class of natural scenes with significant detail at every scale. Its mean gray level value is $\mu_f = 154.78$ and its variance $\sigma_f^2 = 41.24^2$. Uniform regions are more difficult to find, most of the background having a fine texture (grass, bushes). The noise–signal domain transition is almost always detected at level 6. Only two images had ϵ between 0.2 and 0.25.

The *aerial* image [Fig. 2(e)] is a representative of the class of images with significant detail at the smallest scale. The contrast of the image is less than that of the other images. The *aerial* image has mean gray level value $\mu_f = 88.82$ and variance $\sigma_f^2 = 31.37^2$.

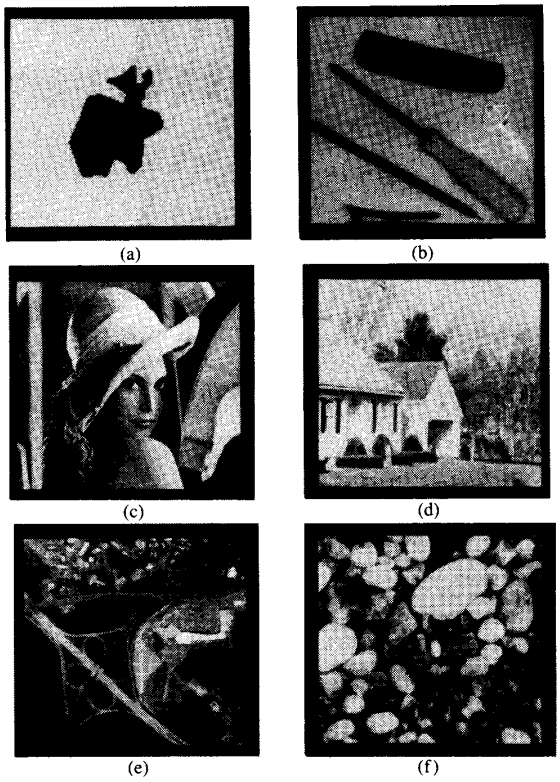

Fig. 2. The 256 × 256 images employed in the experimental results. (a) Binary. (b) Parts. (c) Portrait. (d) Outdoor. (e) Aerial. (f) Texture.

The transition between the noise and signal domains always occurs at level 6. Only three images out of 71 had ϵ between 0.2 and 0.3.

The *texture* image [Fig. 2(f)] is an example of a highly structured natural scene taken from [4]. The mean gray level is μ_f = 102.18 and the variance $\sigma_f^2 = 57.89^2$. The image has no uniform regions, even the largest pebbles having ramp-like gray level profiles and the noise and signal domains are strongly intertwined. For 21 outcomes Rule 4 (warning about incompetence to estimate) had to be applied. The transition detection threshold was then lowered from $T = -0.1$ to $T = -0.2$ and the algorithm was run again for the rejected cases. For all the 71 images, and both iterations, only two images had ϵ between 0.2 and 0.25.

The experimental results are shown centralized in Table III. The upper part of Table III shows the distribution of the relative error ϵ by its density and cumulative value. The list of rules employed, the number of times they were used, and the distribution of the relative error obtained are given in the lower part. Of the total of 497 images derived from the 7 prototypes, the variance of the noise was estimated in the first iteration for 476 images with an average relative error 0.0597 and standard deviation 0.0543. The average error for all the images (after two iterations) was 0.0605 and the standard deviation 0.0539.

The range of corrupting noise was very large, from noiseless images to signal-to-noise ratios well below 1. Nevertheless, our blind noise variance estimation algorithm returned an estimate with less than 20 percent error in more than 98 percent of the cases. The algorithm did identify correctly all the noiseless images (Rule 1) and uniform fields (Rule 2). It was also successful for very noise images where methods based on edge detection (see Section I) would be less reliable.

V. Discussion

The noise variance estimation algorithm has two parts. The first part, summarized at the end of Section II, yields the variance estimate sequence $v(l)$. For every level (tessellation) the four smallest sample variance values are selected. This can be done in parallel and recursively, if the algorithm is implemented on the hierarchical structure of an image pyramid, as will be shown below. The second part, summarized at the end of Section III, returns the value of the corrupting noise variance by applying the corresponding rule. Only a few numbers are involved in the second part of the algorithm, and the computations do not require access to the original image.

TABLE III
Experimental Results for All the Images

	Relative Error Distribution (%)						
ϵ	0-0.05	0.05-0.1	0.1-0.15	0.15-0.2	0.2-0.25	0.25-0.3	> 0.3
Bin	54.62	24.58	11.98	7.14	1.26	0.42	0
Total	54.62	79.20	91.18	98.32	99.58	100	100

Employed rules		Distribution Against Relative Error		
Number	Times Used	0-0.1	0.1-0.2	> 0.2
Rule 1	7	7	0	0
Rule 2	70	70	0	0
Rule 52	38	29	8	1
Rule 53	42	19	21	2
Rule 54	30	18	12	0
Rule 61	74	65	7	2
Rule 62	93	88	2	3
Rule 63	129	81	41	0
Rule 14	5	3	2	0
Rule 152	10	6	4	0
Rule 153	6	5	1	0

A. Parallel Implementation on an Image Pyramid

An *image pyramid* is a stack of decreasing resolution representations derived recursively from the input image. Usually, the resolution decreases twofold between consecutive levels of the image pyramid. Thus, if the size of the input image is $2^n \times 2^n$, the representation at level l is of size $2^{n-l} \times 2^{n-l}$. The image pyramid has n = log(image_size) levels, the "highest" level, the *apex*, having only one pixel. For the implementation of the blind noise variance estimation algorithm the simplest architecture suffices. A cell at level l, the *parent*, is connected with four cells at level $l-1$, its *children*.

The *receptive field* of a cell at level l can be delineated at the input by mapping downward the connections between the parents and their children. The receptive field of the parent is the concatenation of the four fields of its children and has the size $2^l \times 2^l$. The ensemble of receptive fields of the cells at level l generates the level l tessellation of the blind noise variance estimation algorithm.

Assume that the sample variances have already been computed for level $l-1$, i.e., for cells of size $2^{l-1} \times 2^{l-1}$. The sample variances $s_g(k_l)$ at level l must be computed for cells of size $2^l \times 2^l$. If in (2) the average gray level within the cell, $\bar{g}_{k_l}$ is made explicit, the well known computational expression of the sample variance is obtained:

$$s_g(k_l) = \frac{\sum_{i=1}^{2^l}\sum_{j=1}^{2^l}\left[g_{k_l}(i,j)\right]^2 - 4^l\left[\sum_{i=1}^{2^l}\sum_{j=1}^{2^l} g_{k_l}\right]^2}{4^l - 1} \tag{27}$$

where the subscript k_l means that the computation is restricted to the k_lth cell on level l. The sums in (27) are computed by concatenating the corresponding sums of the children. The procedure is parallel because a parent at level l needs access only to its four children at level $l-1$. It is fast, the pyramid being "loaded" with all the sample variance values after $O[\log(\text{image\_size})]$ =

$O[\log n]$ processing steps. Round-off errors [12] are also reduced by computing recursively the sums in (27) since all the quantities are of the same order of magnitude.

The outcomes of the first four order statistics for level l can also be selected in parallel if a parent retains only the four smallest values among the 16 available from its children. At the apex of the pyramid the four smallest sample variance values from the whole ensemble at level l are obtained. The four order statistic values derived from level l are available at the apex of the pyramid in $O[\log(n - l)]$ steps. The specific architecture of the machine on which the algorithm is implemented determines the convenient data communication pattern.

The Connection Machine [7] is a fine-grained parallel machine with increasing popularity in computer vision applications. The parallel noise variance estimation algorithm based on the Canny edge detector (see Section I), was implemented on the Connection Machine by employing local histograms [11]. The authors complain about "costly processing required to accumulate a histogram" and necessary improvements "where a more precise estimate of noise is needed" [11, p. 183]. Image pyramids can be implemented on the Connection Machine [8] and thus the blind noise estimation algorithm described in this correspondence avoids the above problems.

B. Limitations of the Algorithm

To obtain accurate estimates with the blind noise variance estimation algorithm, the noise and signal components should be separable. That is, there should exist at least a few cells at lower levels for which the sample variance is given by the noise alone. These cells should be corrupted by the same noise process, and thus homogeneity of the distorsion across the image is of importance. Highly textured images like the one shown in Fig. 3 do not satisfy this condition. The residual signal component of this image (the variance returned by the algorithm for $\sigma_w^2 = 0$) is 8.25^2. For large noise variance values the residual signal component becomes insignificant; however, the algorithm is not effective because in highly textured images the noise–signal domain transition is not well defined.

Clipping, pixel values driven into saturation (0 or 256) by the noise, may influence the output of the algorithm. For our examples the effect of clipping was negligible since most of the pixels in the regions providing the order statistics outcomes were close to the average gray level value (minimum 88.82 for the *aerial* image). However, if the quasi-uniform regions have gray levels close to the gray scale boundaries, severe distortions are introduced and the algorithm strongly underestimates the variance of the noise.

The presence of clipping can be recognized by the algorithm. The average gray level value is available to every cell for the computation of its sample variance. Thus, after the noise variance is determined, the average gray level values of the eight cells involved (supplying the order statistics outcomes for the rule being employed) could be compared to $\hat{\sigma}_w$. If most of these values are less than (say) twice $\hat{\sigma}_w$, a warning about underestimation should be issued. It is possible to correct the variance estimate when the amount of clipping is known; however, the necessary computations do not justify such a procedure.

The application of the algorithm to 128 × 128 images is expected to be less successful because the uncertainty associated with the variance estimate sequence becomes much larger. The experiments described in Section IV were repeated for seven 128 × 128 prototype images derived by either reduction or windowing from the 256 × 256 prototypes. The same number, 71, of noisy images was derived and the algorithm was first applied without any modification. Rules 3 and 4 (warning about incompetence to estimate) were employed for 82 noisy images. The average error for the remaining 415 images was 0.2695 with a standard deviation of 0.344. The performance of the algorithm was improved if the threshold was increased to $T = 0$, and the rules for $l_o = 5$ modified. Rules 52 and 53 both yielded interpolation between levels 3 and 4, while Rule 54 interpolated between levels 4 and 5. The number of rejected images increased to 132; however, the average relative error for the remaining 365 images decreased to 0.0911 with a standard deviation of 0.0796. The distribution of the rejected images (Rules 3 or 4) was as follows: *parts* 10, *portrait* 30, *outdoor* 7, *aerial* 20, and *texture* 65. The 128 × 128 images will tend to have more details carried by fewer pixels, increasing the busyness of the image and decreasing the chances of finding uniform regions. This can explain why the noise-signal domain transition appeared frequently at levels 3 or 4 in 128 × 128 images.

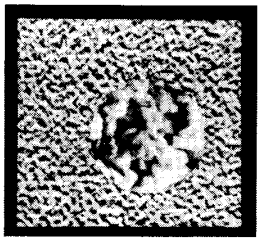

Fig. 3. An example of an image for which the algorithm cannot be applied successfully.

The algorithm was also applied unchanged to several 512 × 512 images whose prototypes were enlarged from the 256 × 256 ones, unless the latter were part of larger images. Of 23 images, the algorithm warned about 8 (6 of them *texture*). The average relative error for the remaining 15 images was 0.0671 with a standard deviation of 0.0881.

In conclusion, application of the blind noise variance estimation algorithm to a 128 × 128 image should be assessed for the specific image to be processed. For relatively simple images the algorithm will perform well; for more complicated ones it may fail. For 512 × 512 images the algorithm is valid.

VI. Conclusion

We have presented a blind noise variance estimation algorithm for recovering the variance of additively corrupting zero mean Gaussian noise without any information about the characteristics of the original image. While the algorithm is most effective if implemented on the parallel, hierarchical structure of an image pyramid, it can be equally successful (although not as fast) when employed on a serial machine. In the former case, the algorithm returns the value of the noise variance after a few tens of processing cycles and may be used on-line in control loops which require updating of the noise variance value continuously.

References

[1] A. V. Balakrishnan, *Kalman Filtering Theory*. New York: Springer, 1984.
[2] P. J. Besl and R. C. Jain, "Segmentation through variable-order surface fitting," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 10, pp. 167–192, 1988.
[3] W. H. Beyer, Ed., *CRC Handbook of Tables for Probability and Statistics*, 2nd ed. Boca Raton, FL: CRC Press, 1968.
[4] P. Brodatz, *Textures*. New York: Dover, 1966.
[5] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, pp. 679–698, 1986.
[6] H. A. David, *Order Statistics*. New York: Wiley, 1970.
[7] W. D. Hillis, *The Connection Machine*. Cambridge, MA: MIT Press, 1985.
[8] H. A. H. Ibrahim, "Pyramid algorithms on the Connection Machine," in *Proc. Image Understanding Workshop*, Cambridge, MA, Apr. 6–8, 1988. Morgan Kaufmann, pp. 634–639.
[9] P. Meer, J. M. Jolion, and A. Rosenfeld, "A fast parallel algorithm for blind estimation of noise variance," Comput. Vision Lab., Univ. Maryland, College Park, Rep. CAR-TR-373, June 1988.
[10] P. Meer, S. Wang, and H. Wechsler, "Edge detection by associative mapping," *Pattern Recogn.*, vol. 22, pp. 491–503, 1989.

[11] T. Poggio, J. Little, E. Gamble, W. Gillett, D. Geiger, D. Weinshall, M. Villalba, N. Larson, T. Cass, H. Bulthoff, M. Drumheller, P. Oppenheimer, W. Yang, and A. Hurlbert, "The MIT vision machine," in *Proc. Image Understanding Workshop*, Cambridge, MA, Apr. 6-8, 1988. Morgan Kaufmann, pp. 177-198.
[12] A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1975.
[13] A. Rosenfeld, Ed., *Multiresolution Image Processing and Analysis*. Berlin: Springer, 1984.
[14] A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, 2nd ed. Orlando, FL: Academic, 1982.
[15] P. K. Sahoo, S. Soltani, and A. K. C. Wong, "A survey of thresholding techniques," *Comput. Vision, Graphics, Image Processing*, vol. 41, pp. 233-260, 1988.
[16] H. Voorhees and T. Poggio, "Detecting blobs as textons in natural images," in *Proc. Image Understanding Workshop*, Los Angeles, CA, Feb. 23-25, 1987. Morgan Kaufmann, pp. 892-899.

Comments on "A Three-Module Strategy for Edge Detection"

RAE-HONG PARK AND WOO YOUNG CHOI

***Abstract*—The purpose of this correspondence is to indicate that in the above paper,[1] the proposed new orthogonal bases are the same as the edge subspace bases in Frei and Chen's approach and the ratio for detecting edge is not different from that proposed by Frei and Chen for special cases. We also propose the fast computational method of Frei and Chen's approach by using the property of normalized orthogonal masks.**

***Index Terms*—Edge detection, edge subspace, orthogonal basis.**

In the above paper,[1] Lacroix pointed out a mistake in Frei and Chen's approach [1], in which (1) will not provide the desired ratio unless some normalization factors are introduced.

$$r = \left(\sum_{i=1}^{e} (\boldsymbol{W} \cdot \boldsymbol{T}_i)^2\right)^{1/2} \Big/ \left(\sum_{i=1}^{9} (\boldsymbol{W} \cdot \boldsymbol{T}_i)^2\right)^{1/2} \tag{1}$$

Lacroix used the natural canonical basis and built new orthogonal bases $\boldsymbol{B}_1, \cdots, \boldsymbol{B}_4$. In fact, the new orthogonal bases with $a = \sqrt{2}$ are the same as the edge subspace bases proposed by Frei and Chen [1]. By using the canonical basis in order to compute the norm of the pattern vector (1) becomes

$$r \propto \left(\sum_{i=1}^{e} (\boldsymbol{W} \cdot \boldsymbol{B}_i)^2\right)^{1/2} \Big/ \left(\sum_{k=1}^{9} w_k^2\right)^{1/2} \tag{2}$$

and (2) has similar structure to the criterion proposed by Frei and Chen (see [1, eq. (2)]) for orthogonal bases having same norms.

Instead of using canonical basis, Gonzalez and Wintz used the normalized orthogonal basis (see [2, p. 344]). When the orthogonal masks are normalized, we can extract the property of orthogonal masks which can be used for fast computation of Frei and Chen's approach. To solve the complex computational requirement of Frei and Chen's approach, we propose the fast computation method by making use of the property of the normalized orthogonal masks.

Manuscript received June 20, 1989.

The authors are with the Department of Electronic Engineering, Sogang University, C.P.O. Box 1142, Seoul 100-611, Korea.

IEEE Log Number 8932063.

[1] V. Lacroix, *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-10, no. 6, pp. 803-810, Nov. 1988.

t_{i0}	t_{i1}	t_{i2}
t_{i7}	t_{i8}	t_{i3}
t_{i6}	t_{i5}	t_{i4}

Fig. 1. Weights of a mask.

The masks proposed by Frei and Chen are classified as follows [1]: the first four masks are suitable for detecting edges; the second set of four masks represents templates suitable for line detection; and the last one is proportional to the average of the pixels in a 3 × 3 mask. If the weights of each mask are defined as t_{ij}'s as shown in Fig. 1, each mask can be represented in the vector form $\boldsymbol{T}_i = (t_{i0}, t_{i1}, \cdots, t_{i8})'$, $i = 1, 2, \cdots, 9$, where $(*)'$ denotes the transpose operation of a vector $*$. With pixels corresponding to the 3 × 3 mask represented by $\boldsymbol{W} = (w_0, w_1, \cdots, w_8)'$, assuming that the vectors $\boldsymbol{T}_i$, $i = 1, 2, \cdots, 9$, are normalized to $\boldsymbol{N}_i$, we can find the magnitudes, $\boldsymbol{P}_e$, $\boldsymbol{P}_l$, and $\boldsymbol{P}_a$, of the projection of $\boldsymbol{W}$ onto the edge, line, and average subspaces, respectively.

$$\begin{aligned} \boldsymbol{P}_e &= \left(\sum_{i=1}^{4} (\boldsymbol{W} \cdot \boldsymbol{N}_i)^2\right)^{1/2}, \\ \boldsymbol{P}_l &= \left(\sum_{i=5}^{8} (\boldsymbol{W} \cdot \boldsymbol{N}_i)^2\right)^{1/2}, \\ \boldsymbol{P}_a &= (\boldsymbol{W} \cdot \boldsymbol{N}_9) \end{aligned} \tag{3}$$

where $\boldsymbol{N}_i$ is $\boldsymbol{T}_i / \| \boldsymbol{T}_i \|$ and $\| \boldsymbol{T}_i \| = (t_{i0}^2 + t_{i1}^2 + \cdots + t_{i8}^2)^{1/2}$. Similarly, θ_e, θ_l, and θ_a denote the angles between $\boldsymbol{W}$ and its projections onto the edge, line, and average subspaces, respectively [1]. If θ_e is less than the threshold, the point is assumed to be an edge. In our proposed method for reducing the computation time, we use the relation $\boldsymbol{P}_e^2 + \boldsymbol{P}_l^2 + \boldsymbol{P}_a^2 = \| \boldsymbol{W} \|^2$ and the relation between vectors of edge subspace. First, $\boldsymbol{P}_e^2$ can be calculated as follows:

$$\boldsymbol{P}_e^2 = \sum_{i=1}^{4} (\boldsymbol{W} \cdot \boldsymbol{N}_i)^2 = \frac{1}{8} \sum_{i=1}^{4} (\boldsymbol{W} \cdot \boldsymbol{T}_i)^2. \tag{4}$$

Since $t_{i8} = 0$ for $i = 1, 2, 3, 4$, we can expand each squared term as follows.

$$\boldsymbol{P}_e^2 = \frac{1}{8} \left(\sum_{j=0}^{7} \sum_{i=1}^{4} t_{ij}^2 w_j^2 + \sum_{j=0}^{7} \sum_{k \neq j}^{7} \left(\sum_{i=1}^{4} t_{ij} t_{ik} \right) w_j w_k \right) \tag{5}$$

Here we use the relation of the masks of the edge subspace, i.e.,

$$\begin{aligned} \sum_{i=1}^{4} t_{ij} t_{ik} &= -4\delta(n) \\ &= \begin{cases} -4, & \text{if } n = (k - j - 4) \bmod 8 = 0 \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \tag{6}$$

$$\sum_{i=1}^{4} t_{ik}^2 = 4, \qquad \text{for } k = 0, 1, \cdots, 7. \tag{7}$$

We can write

$$\begin{aligned} \boldsymbol{P}_e^2 &= \frac{1}{8} \left(4 \left(\sum_{j=0}^{7} w_j^2 \right) - 4 \sum_{j=0}^{7} \sum_{k \neq j}^{7} \delta(n) w_j w_k \right) \\ &= \frac{1}{2} \left(\sum_{i=0}^{3} (w_i - w_{i+4})^2 \right). \end{aligned} \tag{8}$$