ELSEVIER

Image and Vision Computing 14 (1996) 21–34

image AND VISION COMPUTING

Efficient evaluations of edge connectivity and width uniformity

Qiuming Zhu

Computer Vision Laboratory, Department of Computer Science, University of Nebraska at Omaha, Omaha, NE 68182, USA

Received 21 November 1994; revised 28 February 1995

Abstract

We present three simple and efficient methods for the evaluation of edge connectivity and width uniformity in the images processed by certain edge detection and/or edge linking operations. Quantitative measurements of the edge merits are derived first from an analysis of the edge and pixel patterns in the images, and then extended to two more simplified forms. The main purpose of the evaluation methods is to provide an objective assessment of the qualities of the edge images in terms of their easiness to the successive image processing and vision tasks. The evaluation methods are moderate in terms of the accuracy and sensitivity of the measurements as well as the computational complexity of the related algorithms. No prior knowledge about the true edge positions is needed for deriving the measurement values. The evaluation methods are therefore applicable to the real edge images in practical applications, not only to the synthetic testing images.

Keywords: Edge connectivity; Width uniformity; Edge quality

1. Introduction

Edges are considered as one of the most important features in an image. They provide a concise and accurate representation of the object boundaries. From the detected edges, more complicated geometrical shapes and structures of the objects can be identified. Because of this, edge detection and/or edge linking are always the fundamental operations performed at the lower-level image processing and computer vision processes. Many researchers have devoted their efforts to the design of good edge operators and the evaluation of these operators [1–14]. However, little has been done to quantitatively evaluate the edge images for an assessment of how good they are to the successive image processing and vision tasks.

It is known that the appearance of the edges in an image has a great deal of diversity. Edges may be crowded or sparse. One edge image may contain many isolated edge points and broken segments. Edges also vary with respect to the types of objects and the object shapes present in the images. There is no single edge operator that guarantees the production of good quality edges for the images in the varieties of applications. Therefore, it is desirable to have an evaluation scheme that quantitatively measures the edge quality factors on the individual edge images after they have processed by the edge detection and/or linking operations.

The measurements of the edge qualities were mostly done on synthetic images created for testing purpose only [6,15–21]. These synthetic images usually have a known set of true edge positions, so a quantitative measurement of the edge quality can be derived from a comparison of the detected edge pixels and the true edge points. However, these methods are not applicable to images processed in real world applications of computer vision. When an evaluation of edge quality on images in practical applications was needed, manual visual inspection remained the primary method. A major problem with the manual inspection is that the method is subjective to the individual person's perception intuitions. Moreover, no quantitative measurements of the edge qualities can be made, only a qualitative judgment of the image qualities can be reached.

Another important reason for requiring an objective and quantitative evaluation of the edge quality is the possibility of using this evaluation to direct the processes of improving the edge quality in images. For example, the objective and quantitative measurement can be used as an online feedback signal supplied to the iterative processes of edge detection and/or edge linking. This

email: zhuq@unomaha.edu

SSDI 0262-8856(95)01036-X

signal can be processed to provide a guideline for the selection of a better edge operator or a set of better operator parameters. Neither the synthetic-image testing method nor the manual inspection method can perform such a function for the edge operations in real world applications of computer vision.

It is true that there are diverse requirements for the edge qualities in real applications. However, for most computer vision tasks, the edges are required to be (1) correctly identified, (2) at right positions, (3) forming continual lines, and (4) with a uniform width (thinness). While the correctness of the edge occurrence and positions are essential, the connectivity and width uniformity are equally important to the vision processes such as the boundary modelling and object recognition. Most of the synthetic-image testing methods only measure the correctness of the edge occurrence and positions, but not on the quality of connectivity and width uniformity. It is true that the correctness of occurrence and positions usually leads to good connectivity and width uniformity. However, they are by no means the equivalent. In one aspect, the successfulness of the higher level image processes is more closely related to the qualities of edge connectivity and width uniformity. In other aspect, the correctness of the edge occurrence and positions is often not measurable on the images in real applications.

In this paper, we concentrate on quantitative measurements of the edge continuity and width uniformity. The methods are based on a statistical analysis of the edge pixel patterns identified in the images after an edge detection and/or linking operation. The edge connectivity measures how edge pixels are contiguous and edge segments are connected. The edge width uniformity measures how the edges are formed as thin lines, not to be confused with image regions. These measurements are to be conducted on the images in real applications. We do not intend to compare edge operators. Rather, the evaluation is focused on the measurement of the readiness of the edge images to the successive image processing and vision tasks. Application of the evaluation can be found in many image processing tasks, such as the threshold/enhancement of edge images [2,22], the edge thinning and edge linking processes, and the object boundary modelling [23]. The evaluation can also be used to guide the search for a better edge detection and/or linking operator, or to adjust the edge operator parameters for generating a better edge image [24].

The next section reviews some of the present edge evaluation methods, which mostly came out as a by-product of the evaluations of the edge operators. We then introduce three methods for measuring the edge connectivity and width uniformity on images in real world applications. Corresponding to the methods, three algorithms are presented. The algorithms are tested with both synthetic (for simple comparison and analysis) and real images. The results of these tests are used to compare and analyse the algorithms. The final section contains a discussion and concluding remarks.

2. Edge evaluations

Early work of quantitative edge evaluation was reported by Fram and Deutsch [17]. They modeled an edge image as a binary plane in which a signal 1 at the corresponding point denotes an edge point and a 0 otherwise. They had evaluated two parameters: (1) the ratio of the total number of signal 1's divided by the sum of the number of noise 1's plus the number of signal 1's; and (2) the fraction of remaining edge rows 'covered' by the signal 1's. Pratt [20] later gave a single edge measurement which was better formulated than the Fram and Deutsch's. The edge merit F Pratt suggested was defined as:

$$F = \frac{1}{\max\{I_I, I_A\}} \sum_{i=1}^{I_A} \frac{1}{1 + \alpha d^2(i)}$$

where I_I is the number of ideal edge points, and I_A is the number of actually detected edge points. The $d(i)$ is the pixel-miss distance of the ith edge detected, that is, the distance between an edge point and the true edge point. The α is a scaling constant.

Abdou and Pratt [15] evaluated a number of edge operators in terms of the following statistical measurements: (1) edge gradient amplitude response as a function of actual edge orientation; (2) detected edge orientation as a function of actual edge orientation; (3) edge gradient amplitude response as a function of edge displacement; (4) probability of detection P_D *versus* probability of false detection P_F; (5) Pratt's figure of merit. Some of these measures (such as 1–3) can be directly derived from the mathematical representations of the operators. The others are from the resulting edges of testing images. At almost the same time as Abdou and Pratt did their work, Shaw [21] examined some edge detectors by comparison of the resulting edge images using a measurement of the signal-to-noise ratio. The asymmetry feature of Pratt's figure of merit was also discussed by Baddeley [16] and an alternative symmetric measurement was developed by him later.

Based on the previous work of Abdou and Pratt, Peli and Malah [19] proposed their seven quantitative measures: (1) percentage of edge points detected on the ideal (desired) edge; (2) number of detected edge points which do not coincide with the ideal edge (normalized by the number of points on the edge); (3) noise-to-signal ratio, defined as the ratio of the number of detected edge points, which do not coincide with the ideal edge, to the number of detected edge points which coincide with the ideal edge; (4) mean width of a detected edge, defined as the ratio of the total number of detected edge points to

the number of ideal edge points; (3) Pratt's figure of merit measurement; (6) average squared deviation of a detected edge point form the ideal edge; and (7) mean absolute value of the deviation. Haralick [6] also developed his own edge evaluation scheme. He employed three evaluation criteria for edge operator evaluation. They are: (1) conditional probabilities of assigned edges given that they are true edges; (2) conditional probabilities of true edges given the assigned edges; (3) error distance, defined as the average distance to the closest true edge pixels of pixels which are assigned non edge labels but which are true edge pixels.

All the methods discussed above made use of synthetic images which have known location of the actual edge points. The evaluations were also based on the measurements of discrepancies between the detected edges and the actual positions of the edges. None of the above methods evaluated the edge connectivity and the width uniformity directly.

A quantitative evaluation of edge quality on non-synthetic image was reported by Kitchen and Rosenfeld [22]. Since not all edge features are measurable on non-synthetic images, they proposed combining two desirable qualities: (1) good continuation; and (2) thinness, to measure the edges. The method was further generalized to arbitrary window size by Haralick and Lee [18]. However, their method relied on the direct use of the edge strength and gradient values of the edge pixels. It was assumed that for continuation two edge pixels should have an almost identical direction. The method would miss the continuity at the edge turning and the intersecting points. Moreover, the method required a subjective selection of the parameters for the linear equation used to combine the measures of edge continuation and thinness.

Our evaluation methods differ from the Kitchen and Rosenfeld's in the following two main aspects: (1) rather than the direct use of edge strength and gradient values of the edge pixels, our measurements are based essentially on the statistical analyses of the edge pixel patterns. The only information we need for the pixels in the images is the binary values which indicate a pixel is an 'edge' or a 'non-edge'. Because of this, the methods are applicable to edge images processed by a much larger variety of edge operators; (2) rather than using a single scalar value to represent the overall edge continuation and thinness, a pair of measurement values are derived in our evaluations. This pair is: (i) mean strength of edge connectivity and width uniformity; and (ii) deviation of the edge connectivity and width uniformity from the mean value. The methods also differ from the Kitchen and Rosenfeld's in that (1) they deal with the turning and intersecting edge pixels equally well as with the straight edges, and (2) they do not require any subjective parameter setting. The methods are applicable to the edge processes conducted in real world applications.

3. Evaluating pixel patterns

We denote an edge image as a two dimensional function $f(x, y)$ and let $E[x, y]$ be an edge pixel at the position $[x, y]$ of the image. The edge pixel $E[x, y]$ has a value $f[x, y] = 1$. A non-edge pixel has a value $f[x, y] = 0$. Let $W_d(E[x, y])$ be a window covering $d \times d$ (usually $d \geq 3$) image pixels at the neighbourhood region of $E[x, y]$, with $E[x, y]$ sitting at the centre of the window. We call this $W_d(E[x, y])$ an edge evaluation window and express it as a set of image pixels:

$$W_d(E[x,y]) = \left\{ [x \pm i, y \pm j];\ i,j = 0, 1, \ldots, \left\lfloor \frac{d}{2} \right\rfloor - 1, \left\lfloor \frac{d}{2} \right\rfloor \right\}$$

where $\lfloor d/2 \rfloor$ is an integral truncation of $d/2$. The pixels having either an $i = \lfloor d/2 \rfloor$ or a $j = \lfloor d/2 \rfloor$ as a component in their coordinates are called the border pixels of the window.

Edges are local features of an image. Many characteristics of the edge pixels can be described by the pixels in a $W_d(E[x, y])$. In this paper we will focus on the analysis of the pixel patterns relevant to the formations of edge segments within this evaluation window. Let $P_d(n)$ denote the pixel patterns that are formed by n edge pixels in a $W_d(E[x, y])$, where $n \leq d^2$. That is, a $P_d(n)$ pattern has n edge pixels in a square window consisting of $d \times d$ image pixels, with the evaluating edge pixel at the centre. Note that though a $P_d(n)$ is based on the definition of $W_d(E[x, y])$, it is not related to the positions of edge pixel $E[x, y]$ in the image. To avoid confusion, we simply express a $P_d(n)$ pattern as a set:

$$P_d(n) = \{ p_i;\ i = 1, 2, \ldots, d \times d \}$$

where each element p_i represents a pixel value such that $p_i \in \{0, 1\}$ and $\sum_{i=1}^{d^2} p_i = n$. For example, $P_3(4) = \{0, 0, 1, 0, 1, 1, 0, 0, 1\}$ stands for an edge pattern with 4 edge pixels in a 3×3 evaluation window. Since the n edge pixels can be distributed in many different ways in the $d \times d$ window, a $P_d(n)$ actually represents a group of pixel patterns. Figures in later sections show some of these edge patterns in pattern group $P_3(n)$. We use $P_d^i(n)$ to denote the individual edge patterns in the $P_d(n)$ group. The choice of the parameter d is related to the edge width definition. Let w be the edge width, a $d = 2w + 1$ is generally required for a proper evaluation of the w-width edges. In most part of this paper, the cases of $w = 1$ and $d = 3$ will be discussed.

The continuity of the edge pixels can be defined in either a 4-connection or an 8-connection mode. In a 4-connection mode, only the upper, bottom, right and left neighbours of the edge pixel are considered to be adjacent. In an 8-connection mode, the neighbours in all eight geographical directions are considered to be

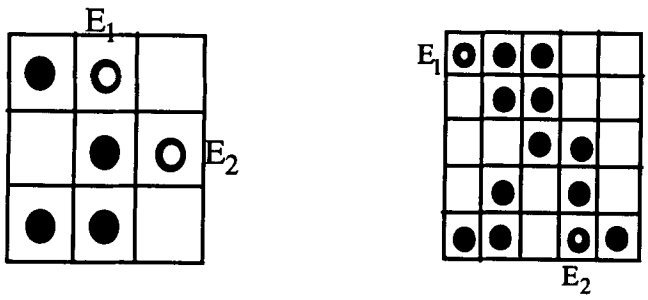


Fig. 1. Examples of non-adjoining border pixel.

adjacent. An 8-connection mode is implied in most part of this paper when no explicit indication is made. To properly describe the edge connectivity and width uniformity in these patterns, we have the following definitions:

Definition 1 *Two edge pixels E_1 and E_2 are **non-adjoining border pixels** if:*

1. E_1 and E_2 are two border pixels of a pattern $P_d^i(n)$.
2. E_1 and E_2 are NOT *adjacent* in terms of 4-connection mode.
3. No *4-connected edge path* between the E_1 and E_2 is formed on the border of the $P_d^i(n)$.*

Examples of some *non-adjoining border pixels* are shown in Fig. 1. Here we need a further explanation for the definition of the border pixels and their adjacencies. A border pixel of a window is the pixel that has at least one adjacent pixel outside the window. Note that when we count the adjacencies among the border pixels, only 4-connectivity is considered. Moreover, the existence of 4-connected edge path is checked only on the set of border pixels of the pattern. Thus the pixels E_1 and E_2 in the 3×3 window at the left part of the Fig. 1. are non-adjoining border pixels.

Definition 2 *Let E_0 be the edge pixel located at the centre of a pattern $P_d^i(n)$. The pixel E_0 is **fully connected** if:*

1. *There exist two **non-adjoining border pixels** E_1 and E_2.*
2. *There exist two **8-connected edge paths**† P_1 and P_2, where the P_1 connects the E_0 to E_1 and the P_2 connects the E_0 to E_2.*
3. *No edge pixel, except E_0, appears on both the paths P_1 and P_2.*

Condition (3) of this definition means that the two paths P_1 and P_2 cannot intersect anywhere else except at the E_0 point. This condition makes E_0 the only pixel that is connected to both P_1 and P_2.

* A 4-connected edge path is a sequence of edge pixels in 4-connection mode.

† An 8-connected edge path is a sequence of edge pixels in 8-connection mode.

Definition 3 *Let E_0 be the edge pixel located at the centre of a pattern $P_d^i(n)$. The pixel E_0 is **partially connected** if:*

1. *There exists one edge pixel E_1 on the border of a pattern $P_d^i(n)$.*
2. *There exists an **8-connected edge path** P_1 that connects the E_0 to E_1.*

For examples of the fully connected and the partially connected edge patterns, see Fig. 2. The definitions of edge connectivity and width uniformity strengths for the edge pattern $P_d^i(n)$ are as follows.

Definition 4 *Let $\mathbf{s}_c(P_d^i(n))$ be the connectivity strength of an edge pattern $P_d^i(n)$, and let E_0 be the edge pixel at the centre of the $P_d^i(n)$:*

1. $\mathbf{s}_c(P_d^i(n)) = 1.0$ *if E_0 is **fully connected**.*
2. $\mathbf{s}_c(P_d^i(n)) = 0.5$ *if E_0 is **partially connected**.*
3. $\mathbf{s}_c(P_d^i(n)) = 0.0$ *otherwise.*

Definition 5 *Let $\mathbf{s}_w(P_d^i(n))$ be the width uniformity strength value of an edge pattern $P_d^i(n)$. Let E_0 be the edge pixel at the centre of $P_d^i(n)$. Let $R_{w+1}(P_d^i(n))$ denote a squared region formed by $(w+1) \times (w+1)$ edge pixels which include the edge pixel E_0:*

1. $\mathbf{s}_w(P_d^i(n)) = 0.0$ *if there exists a $R_{w+1}(P_d^i(n))$ in the pattern.*
2. $\mathbf{s}_w(P_d^i(n)) = 1.0$ *otherwise.*

Fig. 3 shows the edge patterns containing the $R_{w+1}(P_d^i(n))$ regions.

It is not difficult to find out that an edge image with good connectivity is not necessary to have good width uniformity. An edge image with good width uniformity is also not necessary to have good connectivity. For a well-formed edge image, the two measurements must be combined together. We thus define the following:

Definition 6 *Let $\mathbf{s}(P_d^i(n))$ be the strength value of the edge connectivity and width uniformity for an edge pattern $P_d^i(n)$):*

1. $\mathbf{s}(P_d^i(n)) = 1.0$ *if E_0 is **fully connected** and NO $R_{w+1}(P_d^i(n))$ exists in the pattern.*
2. $\mathbf{s}(P_d^i(n)) = 0.5$ *if E_0 is **partially connected** and NO $R_{w+1}(P_d^i(n))$ exists in the pattern.*
3. $\mathbf{s}(P_d^i(n)) = 0.0$ *otherwise.*

The above definition can be equivalently expressed as:

$$\mathbf{s}(P_d^i(n)) = \min(\mathbf{s}_c(P_d^i(n)),\ \mathbf{s}_w(P_d^i(n)))$$

The $\mathbf{s}(P_d^i(n))$s can be pre-computed for the patterns with a given window size d, therefore saving a large amount of computation time.

Let $[\mu_{\mathbf{s}}(f(x,y)),\ \sigma_{\mathbf{s}}(f(x,y))]$ be an overall measurement of the edge connectivity and width uniformity strength for an image $f(w,y)$. The $\mu_{\mathbf{s}}(f(x,y))$ represents the mean value of the strengths of the total edge pixels in

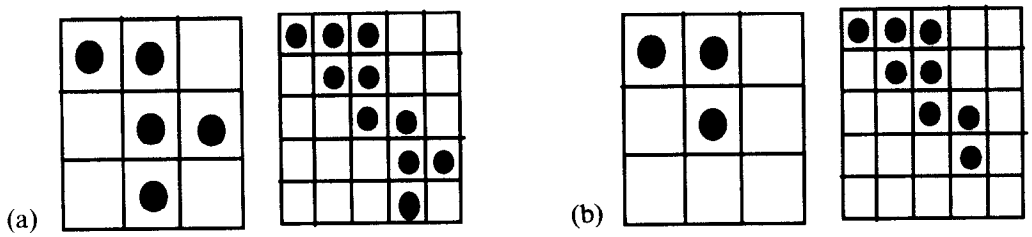


Fig. 2. Examples of (a) fully connected and (b) partially connected edge patterns.

the image. The $\sigma_s(f(x,y))$ represents the deviation of the strength measurements away from the mean value. Note that both $\mu_s(f(x,y))$ and $\sigma_s(f(x,y))$ values will have a range from 0 to 1.

Let $|P_d(n)|$ be the number of individual edge patterns in the $P_d(n)$ pattern group, and $\#P_d^i(n)$ be the total number of occurrences of a $P_d^i(N)$ edge pattern in an image. Let N be the total number of edge pixels in an image $f(x,y)$. We have, for an image $f(x,y)$, the $[\mu_s(f(x,y)),\ \sigma_s(f(x,y))]$ values calculated by:

Algorithm 1

1. For every edge pixel $E[x,y]$, identify its belonging pattern $P_d^i(n)$.
2. $\mu_s(f(x,y)) = 1/N \sum_{n=1}^{d^2} \sum_{i=1}^{|P_d(n)|} \#P_d^i(n)\mathbf{s}(P_d^i(n))$.
3. $\sigma_s(f(x,y)) = 1/N \sum_{n=1}^{d^2} \sum_{i=1}^{|P_d(n)|} \#P_d^i(n) \,|\, \mathbf{s}(P_d^i(n)) - \mu_s(f(x,y))|$.

The time complexity of the algorithm can be easily computed. Let M_d be the total number of pixel patterns in a $d \times d$ window, i.e.:

$$M_d = \sum_{n=1}^{d^2} |P_d(n)|$$

It takes, in the worst case, M_d computations in step 1 to identify the belonging pattern for an edge $E[x,y]$. For N edge pixels, it is M_dN. Since the $\mathbf{s}(P_d^i(n))$ values can be pre-computed, it takes N unit time in steps 2 and 3 to accumulate the $\mu_s(f(x,y))$ and $\sigma_s(f(x,y))$ values for the total edge pixels. The overall time complexity for the entire algorithm then is $O(M_dN)$.

Examples of edge images tested with the above measurements are shown in Fig. 4, where the images are ordered according to the increment of the $\mu_s(f(x,y))$ values computed on these images. Notice that the images in (a), (b), and (c) of Fig. 4 contain over-crowded and thick edges. The images in (d), (e) and (f) of Fig. 4 show sparse edge pixels and disjoint edge segments. The better formed edges are shown in images (g), (h) and (i). Table 1 shows the resulting $[\mu_s(f(x,y)),\ \sigma_s(f(x,y))]$ values of the test images, using $d = 3$ patterns.

The edge strength values of the above tests are also graphically plotted in Fig. 5. From the plot it is clear that the images having either the crowded or the sparse edges, as shown in (a) to (f) of Fig. 4, exhibit lower $\mu_s(f(x,y))$ values and higher $\sigma_s(f(x,y))$ values. The images having better formed edges, as shown in (g) to (i) of Fig. 4, exhibit higher $\mu_s(f(x,y))$ values and lower $\sigma_s(f(x,y))$ values.

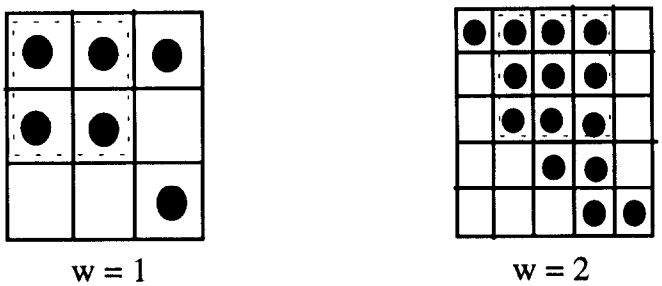


Fig. 3. Edge patterns containing $R_{w+1}(P_d^i(n))$ regions (shown by dotted lines).

4. Evaluating pattern groups

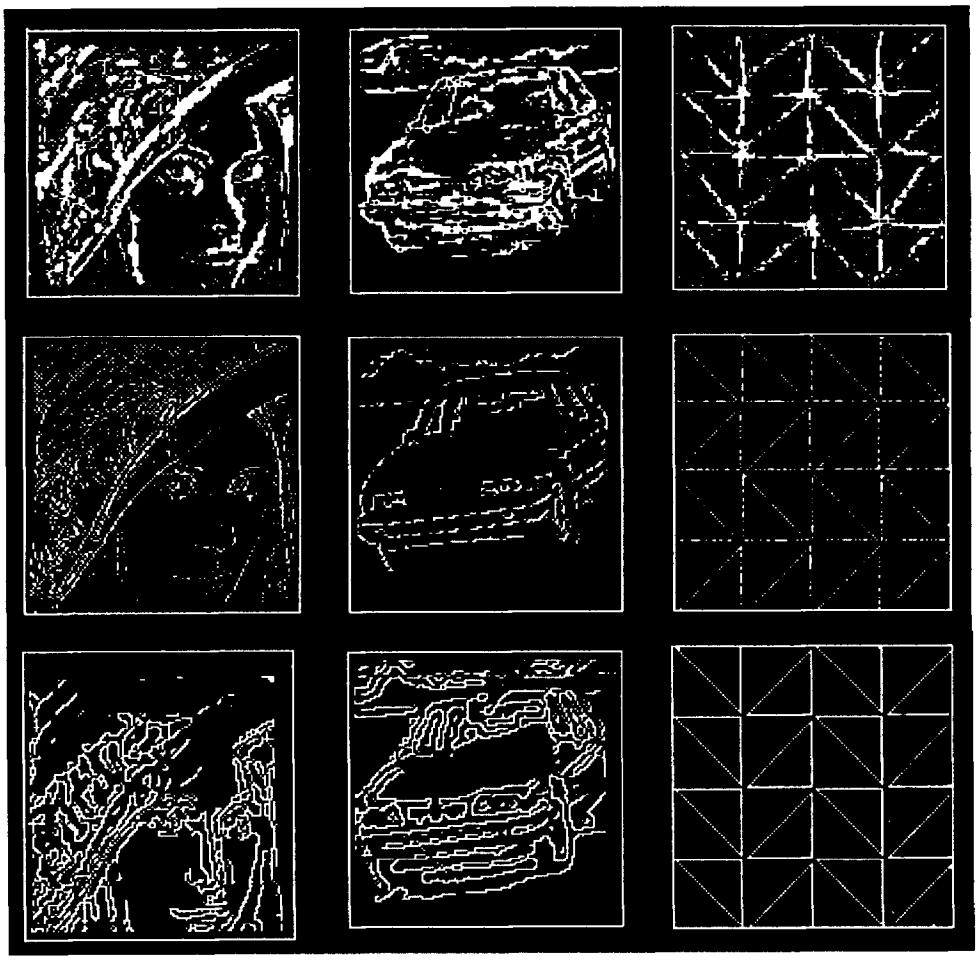
The computation of the $[\mu_s(f(x,y)),\ \sigma_s(f(x,y))]$ values in the above algorithm could be computation intensive and time consuming. In general, we have $|P_d(n)|$, the number of patterns in a pattern group $P_d(n)$, equal to $C(d^2-1, n-1)$, where $C(d^2-1, n-1)$ stands for the number of combinations of selecting d^2-1 elements from a set of $n-1$ elements. The total number of edge patterns in a window size d, is then equal to:

$$M_d = \sum_{n=1}^{d^2} |P_d(n)| = \sum_{n=1}^{d^2} C(d^2-1, n-1)$$

For the case of $d = 3$, that is the patterns in 3×3 window, we have the following:

$$|P_3(1)| = C(8,0) = 1, \quad |P_3(2)| = C(8,1) = 8,$$
$$|P_3(4)| = C(8,3) = 56, \quad |P_3(5)| = C(8,4) = 70,$$
$$|P_3(7)| = C(8,6) = 28, \quad |P_3(8)| = C(8,7) = 8,$$
$$|P_3(3)| = C(8,2) = 28,$$
$$|P_3(6)| = C(8,5) = 56,$$
$$|P_3(9)| = C(8,8) = 1$$

The total number of edge patterns is $M_3 = 256$. That is, for $d = 3$, the algorithm 1 needs to classify each edge pixel $E[x,y]$ to one of these 256 patterns before the strength values can be accumulated. In general,



(a)	(b)	(c)
(d)	(e)	(f)
(g)	(h)	(i)

Fig. 4. Sample edge images (size 128 × 128) used for testing the algorithms.

we have:

$$M_d = \sum_{n=1}^{d^2} C(d^2 - 1, n - 1) = \frac{(d^2 - 1)^d}{2}$$

The M_d value grows exponentially along with the increase of the pattern size d.

An alternative way to compute the overall edge strengths of an image $f(x, y)$ is to do the evaluation on the edge pattern group $P_d(n)$s, rather than on the individual pixel pattern $P_d^i(n)$s. We will see that the strength value $\mathbf{s}(P_d^i(n))$defined on the edge pattern groups $P_d(n)$ would be sufficient for a moderate

Table 1
Resulting values of test images

$f(x,y)$	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)
$\mu_s(f(x,y))$	0.374	0.485	0.561	0.646	0.701	0.736	0.784	0.876	0.995
$\sigma_s(f(x,y))$	0.215	0.244	0.227	0.219	0.226	0.161	0.186	0.212	0.008

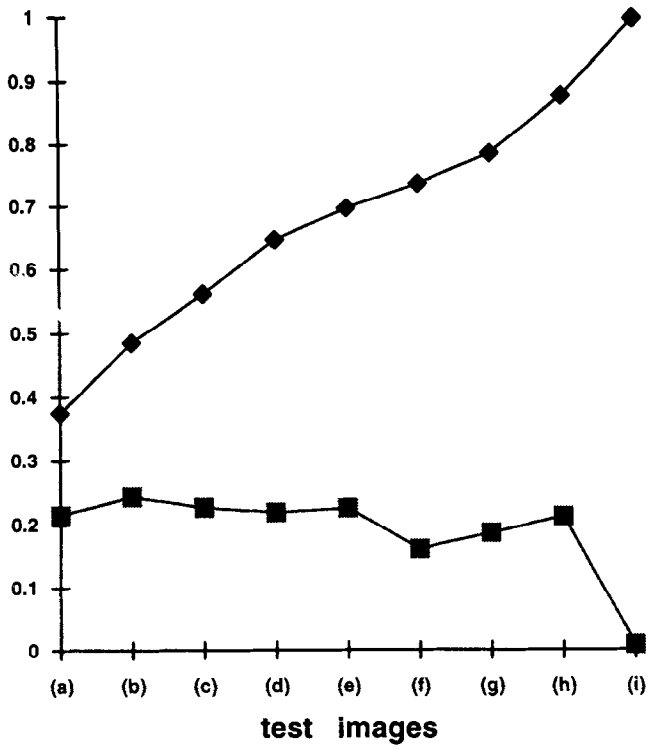


Fig. 5. Plot of edge strength values on test images applying algorithm 1. ◆$\mu_s(\ )$; ■$\sigma_s(\ )$.

evaluation of the edge quality in the connectivity and widht uniformity.

From the analysis of the $\mathbf{s}(P_d^i(n))$ patterns, we have the following:

Definition 7 *The strength value* $\mathbf{s}(P_d(n))$ *for a pattern group* $P_d(n)$ *is:*

$$\mathbf{s}(P_d(n)) = \frac{\sum_{i=1}^{|P_d(n)|} s(p_d^i(n))}{|P_d(n)|}$$

The $\mathbf{s}(P_3(n))$ values, with $w = 1$ as an example, are derived as follows:

1. $n = 1$. We have $|P_3(1)| = 1$. The $P_3^i(1)$ pattern is shown in Fig. 6. It is obvious that the $P_3^i(1)$ pattern represents isolated edge points. Since the $\mathbf{s}(P_3^i(1))$ value is 0, the edge strength for this pattern group is, $\mathbf{s}(P_3^i(1)) = 0$.
2. $n = 2$. We have $|P_3(2)| = 8$. The patterns are shown in Fig. 7. The $\mathbf{s}(P_3^i(2))$ value for these patterns is 0.5. The overall strength for this pattern group is:
   $\mathbf{s}(P_3(2)) = (8 * 0.5)/8 = 0.5$

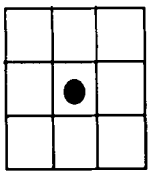


Fig. 6. Edge pixel pattern of $p_3(1)$.

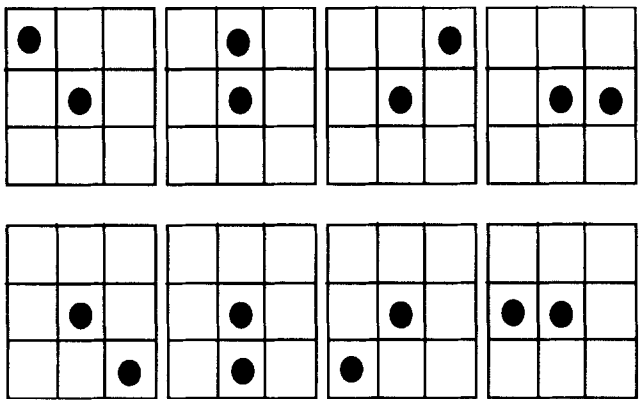


Fig. 7. Edge pixel patterns of $p_3(2)$.

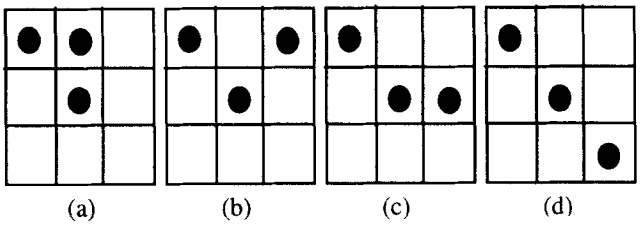


Fig. 8. Edge pixel patterns of $p_3(3)$.

3. $n = 3$. We have $|P_3(3)| = 28$. Some of these patterns are shown in Fig. 8, where each example in (a), (b) and (c) represents eight individual patterns. The (d) represents four individual patterns. These individual patterns can be obtained by rotating every 45 degrees around the centre pixel of the above example patterns. The $\mathbf{s}(P_3^i(3))$ value is equal to 0.5 for the patterns in (a), and is equal to 1.0 for the patterns in (b), (c) and (d). The overall strength of this pattern group is:

   $\mathbf{s}(P_3(3)) = (8 * 0.5 + 20 * 1.0)/28 = 0.86$

4. $n = 4$. We have $|P_3(4)| = 56$. The patterns are shown in Fig. 9, where each represents eight possible patterns by applying the 45 degree rotations. Each of

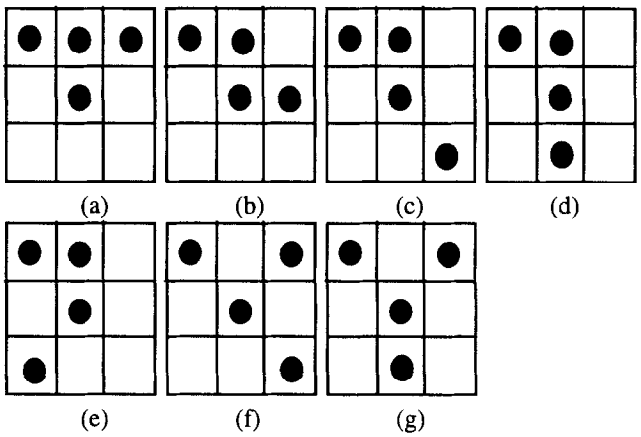


Fig. 9. Edge pixel patterns of $p_3(4)$.

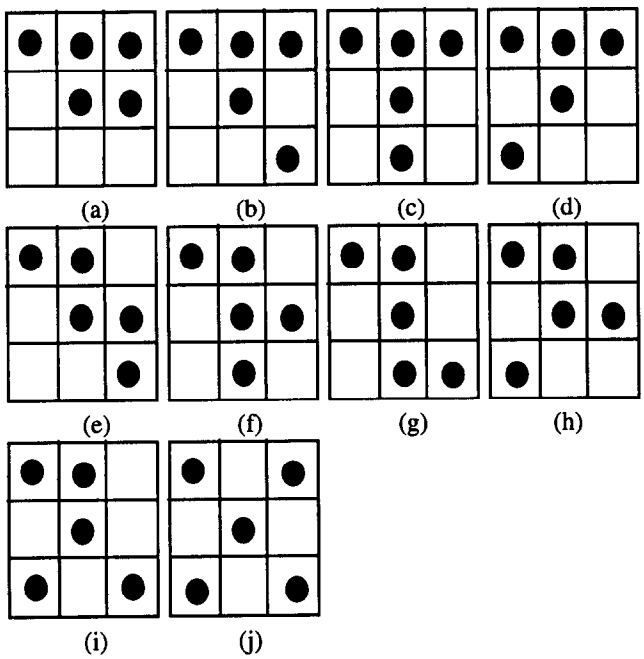


Fig. 10. Edge pixel patterns of $p_3(5)$.

these patterns has a $\mathbf{s}(P_3^i(4)) = 1$ except the patterns in (a) where four of them have a $\mathbf{s}(P_3^i(4)) = 0.5$ and four have a $\mathbf{s}(P_3^i(4)) = 0$. The overall strength of this pattern group then is:

$$\mathbf{s}(P_3(4)) = (4 * 0.0 + 4 * 0.5 + 48 * 1.0)/56 = 0.89$$

5. $n = 5$. We have $|P_3(5)| = 70$. The patterns are shown in Fig. 10, where the (g) represents four possible patterns and the (j) represents two patterns. Each of the others represents eight patterns. The $\mathbf{s}(P_3^i(5))$ values for the individual patterns are listed in Table 2.

   Summarizing the above, we have:

$$\mathbf{s}(P_3(5)) = (20 * 0.0 + 50 * 1.0)/70 = 0.71$$

6. $n = 6$. We have $|P_3(6)| = 56$. The patterns are shown in Fig. 11, where each represents eight patterns. The $\mathbf{s}(P_3^i(6))$ values for the individual patterns are listed in Table 3. The overall edge strength for $P_3(6)$ patterns then is:

$$\mathbf{s}(P_3(6)) = (36 * 0.0 + 20 * 1.0)/56 = 0.36$$

7. $n = 7$. We have $|P_3(7)| = 28$. The patterns are shown in Fig. 12, where each of the (a), (b) and (c) represents eight patterns, and the (d) represents four patterns. Each of these patterns has a $\mathbf{s}(P_3^i(7)) = 1.0$

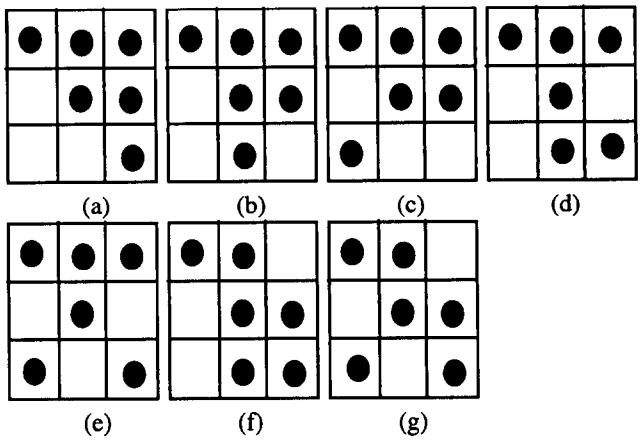


Fig. 11. Edge pixel patterns of $p_3(6)$.

Table 3
Individual pattern values

	(a)	(b)	(c)	(d)	(e)	(f)	(g)
# patterns	8	8	8	8	8	8	8
$\mathbf{s}(P_3^i(6)) = 1$	0	0	0	4	4	4	8
$\mathbf{s}(P_3^i(6)) = 0$	8	8	8	4	4	4	0

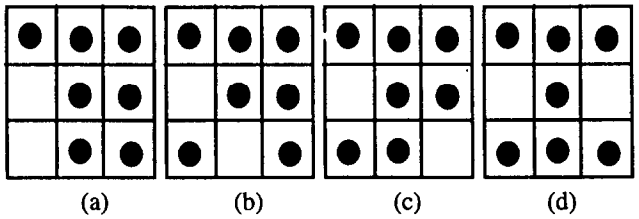


Fig. 12. Edge pixel patterns of $p_3(7)$.

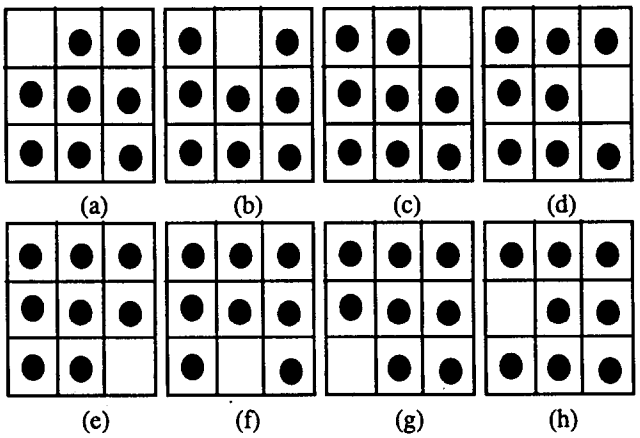


Fig. 13. Edge pixel patterns of $p_3(8)$.

Table 2
Individual pattern values

	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)
# patterns	8	8	8	8	8	8	4	8	8	2
$\mathbf{s}(P_3^i(5)) = 1$	0	4	4	4	8	8	4	8	8	2
$\mathbf{s}(P_3^i(5)) = 0$	8	4	4	4	0	0	0	0	0	0

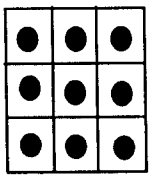


Fig. 14. Edge pixel patterns of $p_3(9)$.

Table 4

n	1	2	3	4	5	6	7	8	9
$\mathbf{s}(P_3(n))$	0.0	0.5	0.86	0.89	0.71	0.36	0.07	0.0	0.0

except the patterns in (d) where two of them have a $\mathbf{s}(P_3^i(7)) = 1.0$ and two have a $\mathbf{s}(P_3^i(7)) = 0$. We thus have:

$$\mathbf{s}(P_3(7)) = (26 * 0.0 + 2 * 1.0)/28 = 0.07$$

8. $n = 8$. We have $|P_3(8)| = 8$. The patterns are shown in Fig. 13. It is not difficult to see that all these patterns have $\mathbf{s}(P_3^i(8)) = 0$. Thus $\mathbf{s}(P_3(8)) = 0.0$.
9. $n = 9$. We have $|P_3(9)| = 1$. The pattern is shown in Fig. 14. Obviously, we have $\mathbf{s}(P_3(9)) = 0$.

The above quantities are summarized in Table 4.

Based on the above discussion, we have the second algorithm for computing the strength of edge connectivity and width uniformity for an image $f(x, y)$. To distinguish from the measurements made in algorithm 1, we denote the measurement computed here as $[\mu'_{\mathbf{s}}(f(x,y)), \sigma'_{\mathbf{s}}(f(x,y))]$. Let $\#P_d(n)$ be the total number of edge pixels identified in pattern group $P_d(n)$, and again let N be the total number of edge pixels in image $f(x, y)$; we have:

Algorithm 2

1. For every edge pixel $E[x, y]$, identify its belonging pattern group $P_d(n)$.
2. $\mu'_{\mathbf{s}}(f(x,y)) = 1/N \sum_{n=1}^{d^2} \#P_d(n)\, \mathbf{s}(P_d(n))$.
3. $\sigma'_{\mathbf{s}}(f(x,y)) = 1/N \sum_{n=1}^{d^2} \#P_d(n) |\mathbf{s}(P_d(n)) - \mu'_{\mathbf{s}}(f(x,y))|$.

The time complexity of this algorithm can be computed in the following. For an evaluation done in $d \times d$ patterns, we have d^2 as the total number of pattern groups. This is the number of classes to which an edge pixel is to be assigned to in step 1 of the algorithm. In fact, the classification of edge pixels to the pattern groups can be done by simply counting the number of edge pixels in a $d \times d$ window. For $d = 3$, only nine groups need to be considered. For N edge pixels, step 1 of the algorithm takes d^2N computations. Since the $\mathbf{s}(P_d(n))$ values can be pre-computed, it takes N unit time to accumulate the $\mu'_{\mathbf{s}}(f(x,y))$ and $\sigma'_{\mathbf{s}}(f(x,y))$ values in step 2 to 3 of the algorithm for processing the total edge pixels in the image. The overall time complexity for the algorithm then is $O(d^2N)$. Comparing to the $O(M_dN)$ complexity of algorithm 1, and noticing that $M_d = (d^2 - 1)^d/2$, algorithm 2 is computationally advantageous to algorithm 1.

Table 5
Modified values

n	1	2	3	4	5	6	7	8	9
$\mathbf{s}(P_3(n))$	0.0	0.56	0.97	1.0	0.80	0.40	0.08	0.0	0.0

Notice that the $\mathbf{s}(P_d(n))$ values derived in the above has a maximum value less than 1.0 (in $d = 3$ case, it is 0.89). The $[\mu'_{\mathbf{s}}(f(x,y)), \sigma'_{\mathbf{s}}(f(x,y))]$ values calculated from them then would also has a maximum value less than 1.0. To make the values range uniformly from 0 to 1.0, we need to adjust the $\mathbf{s}(P_d(n))$ by applying the following normalization:

$$\mathbf{s}(P_d(n)) = \frac{s(P_d(n))}{\max_n\{s(P_d(n))\}}$$

where $\max_n$ is a function that returns the maximum value of the $\mathbf{s}(P_3(n))s$, $n = 1, 2, \ldots, d^2$. We thus have the $\mathbf{s}(P_3(n))$s in Table 4 modified using this scheme. The new values are listed in Table 5.

By using the $\mathbf{s}(P_3(n))$ values in Table 5, the $\mu'_{\mathbf{s}}(f(x,y))$ and $\sigma'_{\mathbf{s}}(f(x,y))$ values computed from algorithm 2 would theoretically in the ranges of from 0 to 1. Table 6 shows the resulting $[\mu'_{\mathbf{s}}(f(x,y)), \sigma'_{\mathbf{s}}(f(x,y))]$ values on the test images shown in Fig. 4, using $d = 3$ patterns.

Fig. 15 plots the $\mu'_{\mathbf{s}}(f(x,y))$ and $\sigma'_{\mathbf{s}}(f(x,y))$ values for the test images. Similar conclusions as we made for the testing results of algorithm 1 can be made here for the data shown in Fig. 15.

5. Evaluating edge indices

From the studies of the relations between the $\mathbf{s}(P_d(n))$ values and the n values, which is the number of edge pixels in each pattern, we have a hint that the number of edge pixels in a given $d \times d$ window could also be used

Table 6
Resulting values

$f(x,y)$	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)
$\mu'_{\mathbf{s}}(f(x,y))$	0.488	0.591	0.637	0.696	0.733	0.764	0.799	0.868	0.954
$\sigma'_{\mathbf{s}}(f(x,y))$	0.243	0.271	0.234	0.189	0.196	0.183	0.165	0.178	0.033

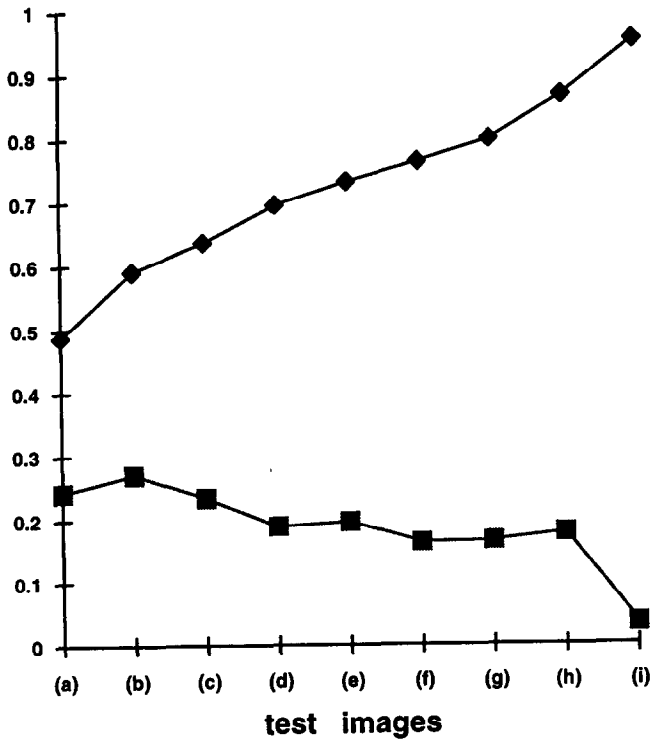


Fig. 15. Plot of edge strength values on test images applying algorithm 2. ◆μ_s'( ); ■σ_s' ( ).

as an indicator to the strength of edge connectivity and uniformity. Look at Table 5, which lists the $\mathbf{s}(P_d(n))$ values with respect to the n; we found that the edge pixels in the patterns of $n = 1, 7, 8$ and 9 make almost no contribution to the edge connectivity and uniformity. The edge pixels in the patterns of $n = 3, 4$ and 5 have the most significant contribution to the edge connectivity and uniformity in an image. The statistics of the occurrences of the edge pattern themselves in an image thus would also provide us with a useful information about the strength of edge connectivity and uniformity. That is, we may have a measurement of edge connectivity and uniformity simply based on the counting of $\#P_d(n)$, the number of edge pixels in each pattern group, without resorting to the $\mathbf{s}(P_d^i(n))$ or $\mathbf{s}(P_d(n))$ values. This observation leads to the following definitions and the third edge evaluation algorithm.

Let $\mathbf{I}_e[x, y]$ be called the edge index for an edge pixel $E[x, y]$; we have:

Definition 8 *The edge index of an edge $E[x, y]$ in a given evaluation window with a size d is:*

$$\mathbf{I}_e[x, y] = n \text{ if } E[x, y] \in P_d(n)$$

In fact, the $\mathbf{I}_e[x, y]$ value can be easily obtained by counting the number of edge pixels in a $d \times d$ window centred at $E[x, y]$. Moreover, an ideal $\mathbf{I}_e[x, y]$ value could be derived from the $\mathbf{s}(P_d(n))$ values for an image with well-formed edges in terms of the edge connectivity and width uniformity. Denote this ideal edge index as $\mathbf{I}_e^*$; we have:

Definition 9 *The ideal edge index $\mathbf{I}_e^*$ for an edge image evaluated by $d \times d$ edge patterns is:*

$$\mathbf{I}_e^* = \frac{\sum_{n=1}^{d^2} s(P_d(n))n}{\sum_{n=1}^{d^2} s(P_d(n))}$$

From the $\mathbf{s}(P_d(n))$ values in Table 5, we have, for $w = 1$ and $d = 3$, the following $\mathbf{I}_e^*$ value:

$$\mathbf{I}_e^* = \frac{\sum_{n=1}^{9} s(P_3(n))n}{\sum_{n=1}^{9} s(P_3(n))} = \frac{14.99}{3.81} = 3.93$$

The ideal edge index means that for a well-formed edge image (with good edge connectivity and width uniformity), most edge pixels in the image should have an $\mathbf{I}_e[x, y]$ value very close to $\mathbf{I}_e^*$. Let $\mathbf{s}(E[x, y])$ denote the strength of connectivity and width uniformity for an edge pixel $E[x, y]$. We have:

Definition 10 *The $\mathbf{s}(E[x, y])$ value defined on the edge index $\mathbf{I}_e[x, y]$ is:*

$$\mathbf{s}(E[x, y]) = 1 - \frac{|I_e[x, y] - I_e^*|}{d^2}$$

Let $\{E\}$ denote the set of edge pixels in an image $f(x, y)$, i.e. $\{E\}$ contains all the edge pixels in $f(x, y)$. The cardinal of $\{E\}$, $\|\{E\}\|$, is equal to N, where N is the total number of edge pixels in image $f(x, y)$. Based on the above definitions, we have our third algorithm for measuring the edge connectivity and uniformity. To distinguish from the measurements made in algorithms 1 and 2, we denote them as $[\mu_s''(f(x, y)), \sigma_s(f(x, y))]$:

Algorithm 3

1. For every edge pixel $E[x, y]$, calculate its $\mathbf{s}(E[x, y]$ value.
2. $\mu_s''(f(x, y)) = 1/N \sum_{E[x,y] \in \{E\}} \mathbf{s}(E[x, y])$.
3. $\sigma_s''(f(x, y)) = 1/N \sum_{E[x,y] \in \{E\}} \times |\mathbf{s}(E[x, y]) - \mu_s''(f(x, y))|$.

The time complexity of this algorithm is derived as follows. Since the computation of the $\mathbf{I}_e[x, y]$ value requires counting the $d \times d$ pixels within the evaluation window, the computation of $\mathbf{s}(E[x, y])$ value for each edge pixel in step 1 takes d^2 times. Both the computation of the $\mathbf{s}(E[x, y])$ and the $[\mu_s''(f(x, y)), \sigma_s''(f(x, y))]$ on N edge pixels takes N unit time. The overall time complexity of this algorithm for processing N edge pixels in an image is then $O(d^2 N)$, which is the same as algorithm 2.

Table 7 shows the resulting $[\mu_s''(f(x, y)), \sigma_s''(f(x, y))]$

Table 7
Test image values

$f(x,y)$	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)
$\mu_s''(f(x,y))$	0.588	0.660	0.689	0.721	0.732	0.738	0.745	0.793	0.869
$\sigma_s''(f(x,y))$	0.183	0.181	0.157	0.148	0.152	0.093	0.108	0.142	0.022

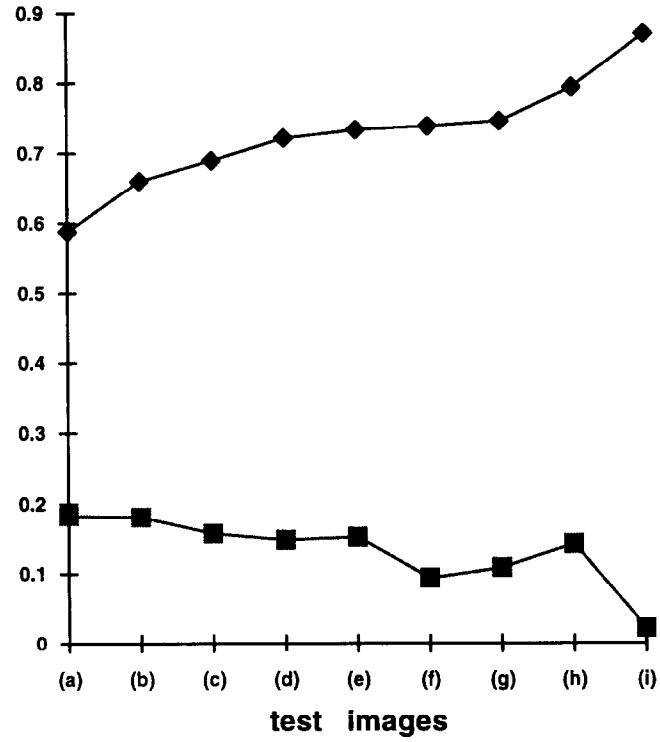


Fig. 16. Plot of edge strength values on test images applying algorithm 3. ◆μ_s''( ); ■σ_s'' ( ).

values on the test images shown in Fig. 4, using $d = 3$ patterns.

Fig. 16 plots the $\mu_s''(f(x,y))$ and $\sigma_s''(f(x,y))$ values for the test images. Theoretically, the $\mu_s''(f(x,y))$ and the $\sigma_s''(f(x,y))$ values are in the ranges from 0 to 1.

6. Discussion

Edges often correspond to the bounding contours of objects as they appear in the images. The contour lines provide important 3D structural information for the objects. In most cases, edges do not appear as coherent features in the images initially. They must be reconstructed from fragments obtained in low level edge detection processes. Good edge detection requires that the operators be designed to fit the nature of the specific images. Conventionally, an edge detection is performed in an open-loop process, where no feedback from the outcome of the detection is supplied to the operator. This system leads to a problem that the quality of the resulting edge images is not controllable. A system with feedback from the outcome of the operation provides an alternative approach for improving the edge detection. However, this requires the use of an objective edge evaluation.

In this paper, we have presented an edge evaluation scheme which is applicable to provide feedback information to a closed-loop edge detection process and many

Fig. 17. Test images for the comparison of edge strength measurements.

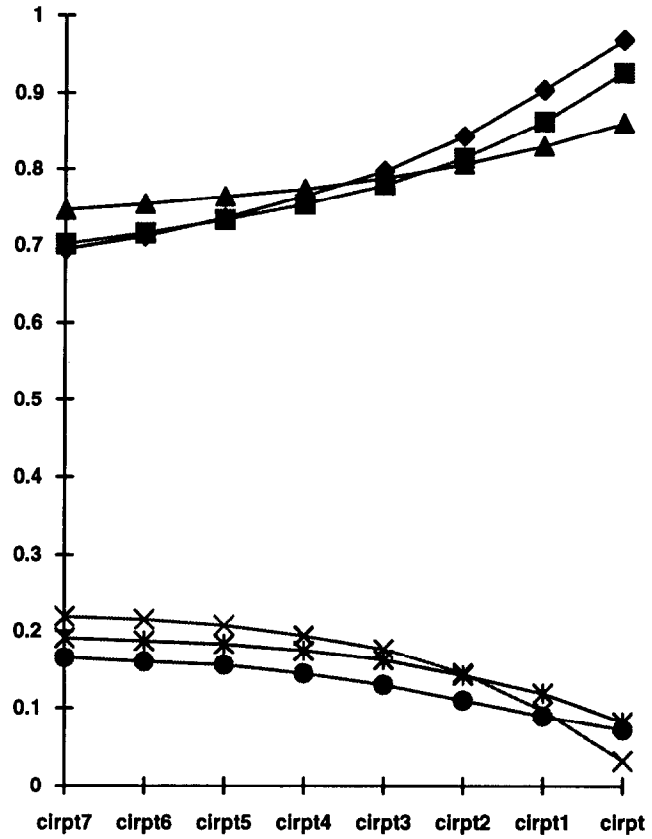


Fig. 18. Edge strength values computed on the test images of Fig. 17. ◆μ_s( ); ■μ_s' ( ); ▲μ_s'' ( ); ×σ_s ( ); ∗σ_s'( ); ●σ_s''( ).

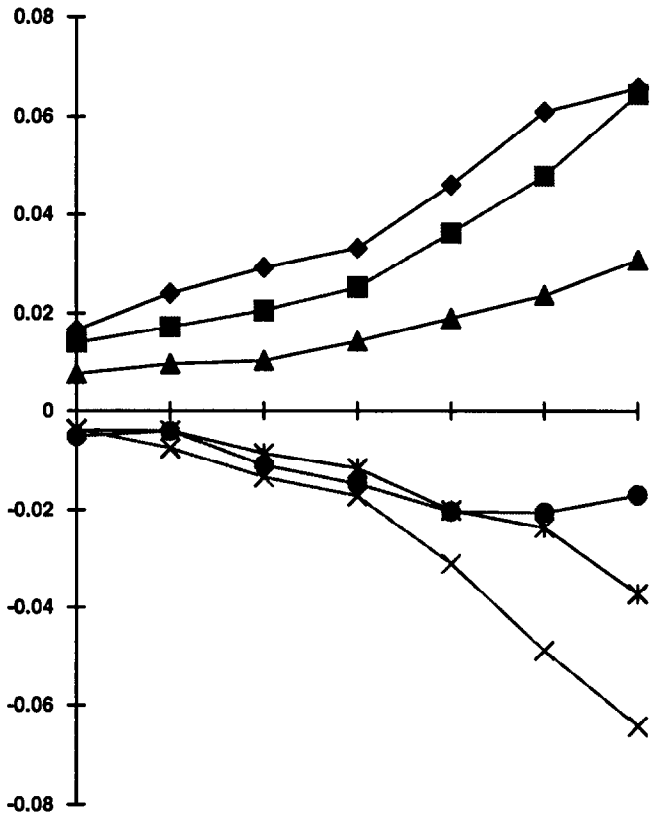


Fig. 19. Differences of edge strength values between images of Fig. 17. ◆μ_s( ); ■μ_s' ( ); ▲μ_s'' ( ); ×σ_s ( ); ∗σ_s'( ); ●σ_s''( ).

other edge-related image operations. The evaluation scheme measures the edge connectivity and width uniformity on an image after it has been processed by an edge detection operation. We presented three algorithms. The first algorithm is based on the measurement of individual edge pattern strengths, $\mathbf{s}(P_d^i(n))$. The second algorithm is based on the measurement of edge strengths of pattern groups, $\mathbf{s}(P_d(n))$. The third algorithm is based on the measurement of edge index, $\mathbf{I}_e[x, y]$, which is defined on the number of edge pixels in a given pattern. The difference of the measurements from these three algorithms are experimentally studied. We used a group of synthetic images with known edge noise-to-signal ratios. Fig. 17 shows these images with a noise-to-signal ratio in decreasing order from 0.7 to 0.0. Fig. 18 plots the measurement values as the three algorithms were applied to these images, respectively. The plot shows that both the mean values (μ_s, μ_s', μ_s'') and the deviation values (σ_s, σ_s', σ_s'') change proportionally with respect to the changes of the edge noise-to-signal ratios in the images. Fig. 19 shows the changes of the (μ_s, μ_s', μ_s'') and (σ_s, σ_s', σ_s'') values between every two images in the sequence shown in Fig. 17. The plot provides partly a comparison of the performances of these algorithms, in terms of their sensitivity to the variations of the edge qualities.

The sensitivity of the three measurements on real edge images is also studied. We computed the mean and variance values of the (μ_s, μ_s', μ_s'') and (σ_s, σ_s', σ_s'') measurements on a large number of different testing images. The test results are listed in Table 8. While the mean values are used as indications to the average magnitude of the measurement, the variance values represent the sensitivity of the measurements for each algorithm. A comparison of the variance values of the measurements (μ_s, μ_s', μ_s'') shows that the μ_s measurement has the best sensitivity among the three, while the μ_s'' has the least sensitivity. The σ_s measurement also has a higher variance value than the other two, while the σ_s'' has the lowest variance value.

To better compare the results, Fig. 20 shows a plot of them. Regarding to the time complexity, the three algorithms exhibit an $O(M_dN)$, $O(d^2N)$ and $O(d^2N)$ complexity, respectively.

We must point out that the measurements presented here are not a complete evaluation of edge qualities in an image. They do not measure the edge locations and the ratio of true edges and non-edges. The evaluation also has limitations in applying to certain edge detectors like Marr–Hildreth's and Canny's. However, since the true edge locations and detection ratios are not measurable on edge images in real applications, this evaluation is still valuable in providing a quantitative measurement of the

Table 8
Mean and variance values of the edge measurements from experiments

	(μ_s)	(σ_s)	(μ'_s)	(σ'_s)	(μ''_s)	(σ''_s)
Mean (·)	0.6092	0.1908	0.6868	0.2061	0.7046	0.1502
Variance (·)	0.0397	0.0035	0.0164	0.0033	0.0084	0.0027

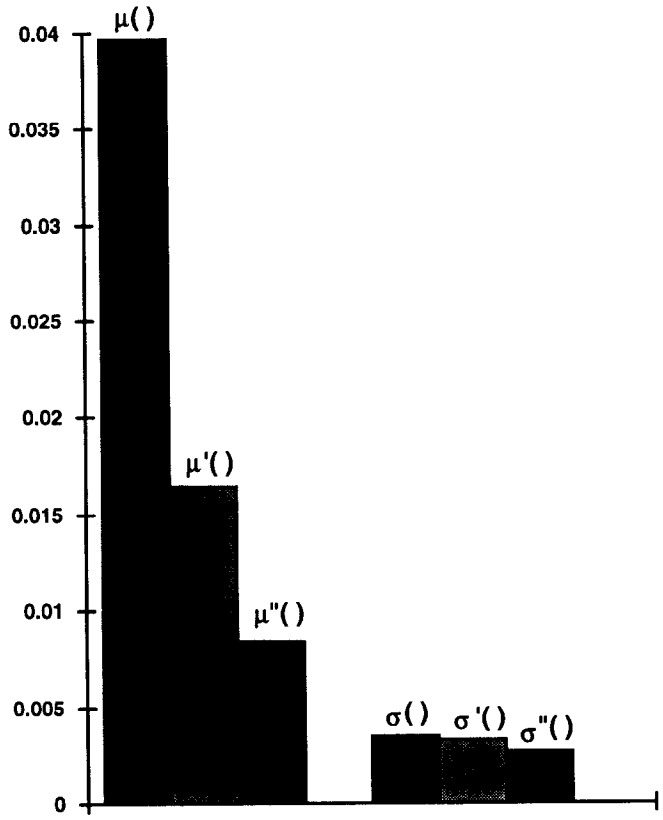


Fig. 20. Comparison of the sensitivities of the measurements.

edge qualities for image in many real applications. One of its main application would be using these measurement to determine automatically the edge detection and linking operator parameters and the threshold values [22,25]. For example, the evaluations can made when a range of edge threshold parameters are applied to the images, and the one maximizing the edge measurements could be chosen. For the focusing of the discussion, these problems are not dealt with in detail in this paper. The readers are referred to Kitchen [22] and Venkatesh [25].

Acknowledgements

The author is extremely thankful to the reviewers, who provided very detailed and useful comments on an earlier version of this paper. The support of this research from the University Committee on Research, University of Nebraska at Omaha, is acknowledged.

References

[1] M.J. Brook, Rationalizing edge detectors, Comput. Graph., Image Process., 8 (1978) 277–285.
[2] F. Bergholm, Edge focusing, IEEE Trans. PAMI, 9 (1987) 726–741.
[3] J. Canny, A computational approach to edge detection, IEEE Trans. PAMI, 8 (1986) 679–698.
[4] J.S. Chen and G. Medioni, Detection, localization, and estimation of edges, IEEE Trans. PAMI, 11, No 2 (February 1989) 191–199.
[5] L.S. Davis, A survey of edge detection techniques', Comput. Graph., Image Process., 4, No 3 (1975) 248–270.
[6] R.M. Haralick, Digital step edges from zero crossing of second directional derivatives, IEEE Trans. PAMI, 6 (1984) 58–68.
[7] A. Kundu and S.K. Mitra, A new algorithm for image edge extraction using a statistical classifier approach, IEEE Trans. PAMI, 9, No 4 (July 1987) 569–577.
[8] Y. Lu and R.C. Jain, Behaviour of edges in scale spcae, IEEE Trans. PAMI, 11, No 4 (April 1989) 337–356.
[9] D.C. Marr and E.C. Hildreth, Theory of edge detection, Proc. Roy. Soc. Lond., B207 (1980) 187–217.
[10] E. Mitchell, B. Caprile, P. Ottonello and V. Torre, Localization and noise in edge detection, IEEE Trans. PAMI, 11 (1989) 1106–1116.
[11] V.S. Nalwa and T.O. Binford, On detecting edges, IEEE Trans. PAMI, 8 (1986) 699–714.
[12] J.M. Prager, Extracting and labeling boundary segments in natural scenes, IEEE Trans. PAMI, 2, No 1 (January 1980) 16–27.
[13] V. Toree and T.A. Poggio, On edge-detection, IEEE Trans. PAMI, 8, No 2 (March 1986) 147–163.
[14] Q. Zhu, Y.Y. Hung, S.H. Tang and D.H. Shi, Edge extraction by active defocusing, J. Spatial Vision, 5, No 4 (1991) 253–268.
[15] I.E. Abdou and W.K. Pratt, Quantitative design and evaluation of enhancement/thresholding edge detectors, Proc. IEEE, 67, No 5 (May 1979) 753–763.
[16] A.J. Baddeley, An error metric for binary images, in Robust Computer Vision, Forstner and Ruwiedel, eds., Wichmann (1992) 59–80.
[17] J.R. Fram and E.W. Deutsch, On the quantitative evaluation of edge detection schemes and their comparison with human performance, IEEE Trans. Comput., 24 (June 1975) 616–628.
[18] R.M. Haralick and J.S.J. Lee, Context dependent edge detection and evaluation, Patt. Recogn., 23 (1990) 1–19.
[19] T. Peli and D. Malah, A study of edge detection algorithms, Comput. Graph., Image Process., 20 (1982) 1–21.
[20] W.K. Pratt, Digital Image Processing, Wiley, New York (1978).
[21] G.B. Shaw, Local and regional edge detectors: some comparisons, Comput. Graph., Image Process., 9 (1979) 135–149.

[22] L. Kitchen and A. Rosenfeld, Edge evaluation using local edge coherence, IEEE Trans. Syst., Man Cybern., 11, No 9 (September 1981) 597–605.

[23] V. Riordan and Q. Zhu, Informed edge linking using a directional potential function, Proc. SPIE Int. Symposium on Optical Tools for Manuf. and Advanced Automation, Boston, MA (September 1993).

[24] Q. Zhu, Improving edge detection by an objective edge evaluation, Proc. ACM/SIGAPP Symposium on Applied Computing, Kansas City, MO (March 1992) 459–468.

[25] S. Venkatesh and P.L. Rosin, Dynamic thresholding determination by local and global edge evaluation, SPIE Conf. Applic. Artif. Intell.: Machine Vision and Robotics, 1964 (1993) 40–50.