Proceedings of the 6th IASTED International Conference
COMPUTER GRAPHICS AND IMAGING
August 13-15, 2003, Honolulu, Hawaii, USA

A TRIPLE-DIAGONAL GRADIENT-BASED EDGE DETECTION

Qiang Wu, Xiangjian He and Tom Hintz

Department of Computer Systems
University of Technology, Sydney
PO Box 123, Broadway 2007, Australia
E-mail: {wuq, sean, hintz}@it.uts.edu.au

Abstract

Gradient-based edge detection is a straightforward method to identify the edge points in the original grey-level image. It is consistent with the intuition that in the human vision system the edge points always appear where the change of grey-level is greatest within their neighbourhood. In this paper, triple-diagonal gradient-based edge detection is introduced. It is based on the features of Spiral Architecture and computes the gradients in three diagonal directions instead of approximating the gradient in one direction only as the traditional methods do. Essentially, it improves the accuracy for locating edge points. As a result, it does not need any supplementary processing to enhance the edge map.

Key Words

Spiral Architecture, Edge Detection, Laplacian of Gaussian Method

1. Introduction

Edge detection is an important aspect of image analysis in computer vision. It is a data reduction process. Images contain enormous amounts of data, typically on the order of hundreds of kilobytes or even megabytes. Often, much of this information is not necessary for solving a specific computer imaging problem. Edge detection can eliminate much unnecessary information while retaining the key shape information. This simplifies further processing.

Gradient-based edge detection is a straightforward method to identify the edge points from the original grey-level image. It is consistent with the intuition that in the human vision system the edge points always appear where the change of grey-level is greatest within their neighbourhood. In this approach, the edge detection operator computes the relationship between a pixel and its neighbours. The edge detection operator examines each pixel neighbourhood, and quantifies the slope and often the direction (i.e., gradient) as well, of the grey-level transition. If a pixel's change of grey-level value is similar to those of pixels in the gradient direction, it may not be identified as an edge point. In [1], edge point is defined by Lindeberg as follows.

Definition 1. *A pixel is classified as an edge point if and only if the amplitude of the gradient of brightness function at this pixel is a local maximum compared with the gradient magnitudes at all its neighbouring points in the gradient direction.*

Gradient-based edge detection has been developed from a concept within continuous domains. Theoretically, gradient based edge detection on an image should create a perfect edge map. However, as the image domain for digital image processing is discrete, it is not always possible to get the exact gradient value of a pixel. Instead, we obtain only an approximation of the gradient value.

There are many operators reported [2] for approximation of gradient values. The quality of edge map detected depends on the approximation method chosen and the detection error produced. There are two problems that one will always face when dealing with edge detection. Firstly, a detection operator may inaccurately locate the neighbouring pixels in the gradient direction of a pixel. When the gradient direction of a pixel does not exactly point to (the centre of) any of the neighbouring pixels, the pixel in the gradient direction is approximated by the neighbouring pixel that is closest to the gradient direction. It is possible that some edge points are not found while some non-edge points are detected and included in the edge map. It is also possible that some edge points detected are a few pixels away from their real locations. The second problem is with computation error of an edge detection algorithm that produces unexpected gaps between edge points. Although the well-known Laplacian of Gaussian (LOG) [3] method guarantees the closed edge boundaries, it introduces many false edge points and retains many insignificant edge points. In addition, the LOG method suffers from the computational complexity as it needs the computation of second-order derivatives.

In order to improve the quality of edge map, a *triple-diagonal* gradient-based edge detection scheme will be developed in this paper. This new method is based on the

features of Spiral Architecture [4], a developing theory in image processing. We will compute the gradient components of each pixel in three diagonal directions instead of approximating the gradient in a single direction as computed by the traditional methods.

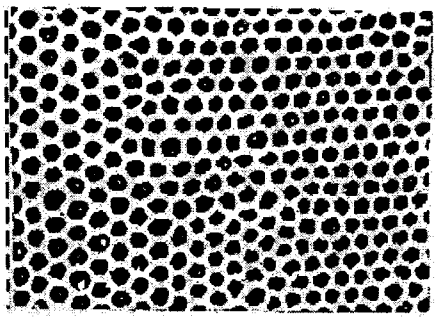
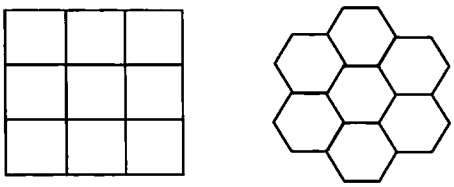


Figure 1: Distribution of Cones on the Retina



(a) Rectangular Architecture (b) Spiral Architecture

Figure 2: Unit of vision in the two image architectures

One will find that the accuracy for locating edge points will be greatly improved so that it will no longer need any supplementary processing to further enhance the edge map.

The organization of this paper is as follows. Spiral Architecture is introduced in Section 2. This is followed by the triple-diagonal gradient-based edge detection in Section 3. We show the experimental results using the new edge detection methods in Section 4. We conclude in Section 5.

2. Spiral Architecture

Spiral Architecture is inspired from anatomical considerations of the primate's vision [5]. From studies of the geometry of the cones of a primate's retina (See Figure 1), it has been concluded that the cones' distribution has inherent organization and is featured by its potential powerful computation abilities. In Spiral Architecture, an image is represented as a collection of hexagonal pixels. A hexagonal pixel relates to the image as the cones relate to the primate's retina. Spiral Architecture consists of organizational units of vision. Each unit is a set of seven hexagonal pixels in contrast with the traditional 3×3 vision unit in the rectangular image architecture as shown in Figure 2.

In Spiral Architecture, each pixel has only six neighbouring pixels that are an equal distance away. They together form a vision unit. Each pixel is identified by a number, base 7, called a *spiral address*. The numbered (or addressed) hexagons form the cluster of size 7^n, where n is a positive integer. These hexagons starting from address 0 towards address 7^n tile the plane in a recursive modular manner along a spiral-like curve. As an example, a cluster with size of 7^2 and the corresponding spiral addresses are shown in Figure 3.

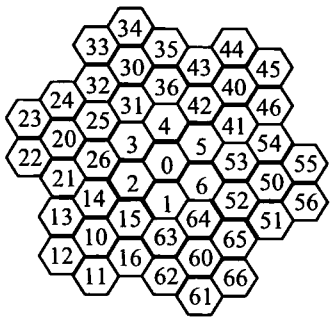


Figure 3: Cluster of size 49 including spiral addresses

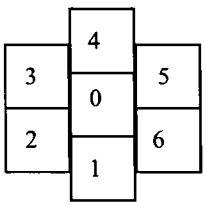


Figure 4: Example of mimic Spiral Architecture with the corresponding spiral addresses

Spiral Architecture possesses some geometric and algebraic properties, which are very useful and can be interpreted in terms of a Euclidean ring (Refer to [6] for details of this mathematical object). Two algebraic operations have been defined on Spiral Architecture based on spiral addresses. They are *Spiral Addition* and *Spiral Multiplication*. These two operations correspond to two transformations on Spiral Architecture, which are translation and rotation with a scaling.

In order to make research results based on Spiral Architecture practically workable with the existing image capture devices, a mimic Spiral Architecture [7] has been widely used. In the mimic Spiral Architecture, a set of four square pixels that are adjacent to each other and form a bigger square like those in Figure 4 is used to mimic one hexagonal pixel (See Figure 4). In this paper, we once again use the mimic Spiral Architecture.

3. Edge Detection on Spiral Architecture

In this section, we present a novel gradient-based edge detection method on Spiral Architecture. In this method, three gradient components on three diagonal directions are computed at each point (pixel).

Let $L_{\hat{v}}$ be the gradient of the brightness function L at a given reference point and $\left|L^i_{\hat{v}}\right|$ ($i \in \{1,2,3\}$) be the three gradient components in the three diagonal directions respectively for a given reference point as shown in Figure 5. We call the three gradient components *triple-diagonal gradient components*. In the real Spiral Architecture, the distance between the reference point and any of its neighbouring point is the same. Without loss of generality, we assume that the distance is 1. It is then easy to see that (refer to [8])

$$L_{\hat{v}} = \left|L^1_{\hat{v}}\right|(0,1) + \left|L^2_{\hat{v}}\right|(\sqrt{3}/2, 1/2) + \left|L^3_{\hat{v}}\right|(\sqrt{3}/2, -1/2) \quad (1)$$

where the three vectors correspond to the three diagonal directions.

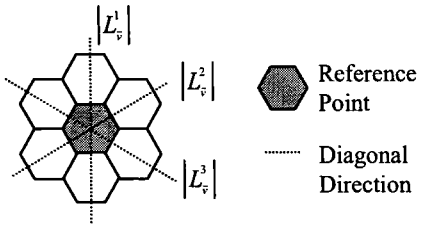


Figure 5: Triple-diagonal gradient components on Spiral Architecture

Each gradient component is dependent on the grey values at two neighbouring points of the reference point in the corresponding diagonal direction. It can be computed using the following equation (refer to [8]),

$$\left|L^i_{\hat{v}}\right| = \left|L_{i,1} - L_{i,2}\right| / 2 \quad (2)$$

where $L_{i,1}$ and $L_{i,2}$ $i \in \{1,2,3\}$ are the grey values of two neighbouring points of the reference point in the i^{th} diagonal direction. As can be seen in Equation 2, we use the difference of two grey values at two neighbouring pixels of a reference point to approximate the magnitude of the first-order directional derivative in a diagonal direction at the reference point.

Unlike the work done in [8] where a neural network was used for edge detection, our proposed edge detection scheme in this paper is based on these three gradient components. These three gradient components are distributed onto exact three diagonal directions as shown in Figure 5. This prevents edge detection from being affected by the errors due to the gradient direction offset.

For a given reference point, there are three possible cases when comparing gradient components at this reference point with the gradient components at all its six neighbouring points.

1. Any of three gradient components in its diagonal direction is larger than the corresponding gradient component for each of the two neighbouring pixels in this direction.

   In other words, as shown in Figure 5, the gradient component for the central pixel in the diagonal direction (0,1) is larger than the gradient components for the two neighbouring pixels of the reference point in the diagonal direction (0,1). These two neighbouring pixels are the one above the central pixel and the other one below the central pixel.

   The same argument applies to the other two diagonal directions.

2. At least two gradient components in two diagonal directions are both larger than their corresponding gradient components for all four neighbouring points of the reference point in the two diagonal directions.

3. At least one gradient component in one direction is larger than its corresponding gradient components for the neighbouring points of the reference point in that direction.

Definition 2. *A pixel is classified as an edge point of type 1 if and only if case 1 above is met when the given pixel acts as a reference point.*

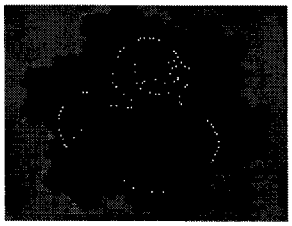
Definition 3. *A pixel is classified as an edge point of type 2 if and only if case 2 above is met when the given pixel acts as a reference point.*

Definition 4. *A pixel is classified as an edge point of type 3 if and only if case 3 above is met when the given pixel acts as a reference point.*

It is easy to see that Case 3 implies Case 2, and Case 2 implies Case 1. In another word, an edge point of type 1 is an edge point of type 2, and an edge point of type 2 is an edge point of type 3.

Figure 6 shows three binary edge maps of a toy duck (Figure 7a). Figure 6a is the edge map consisting of all edge points of type 1, Figure 6b is the edge map consisting of all edge points of type 2, and Figure 6c is the edge map consisting of all edge points of type 3. The flowing theorems are obvious.

Theorem 1. *Almost any edge point of type 1 is an edge point as defined in Definition 1.*



(a) Type 1

(b) Type 2

(c) Type 3

Figure 6: Edge maps of three types

Proof. Without loss of generality, let us test the pixel with spiral address 0 as shown in Figure 3. Note that the gradient magnitude at each pixel is the absolute value of $L_{\bar{v}}$ as defined in Equation 1. Hence, by the definition of type 1 edge point, if the pixel with spiral address 0 is an edge point of type 1, then $|L^1_{\bar{v}}|$ value at this pixel is lager than the $|L^1_{\bar{v}}|$ values in the pixels with spiral addresses 1 and 4 as shown in Figure 3. Similarly, $|L^2_{\bar{v}}|$ value at the central pixel is lager than the $|L^2_{\bar{v}}|$ values in the pixels with spiral addresses 2 and 5, and $|L^3_{\bar{v}}|$ value at this pixel is lager than the $|L^3_{\bar{v}}|$ values in the pixels with spiral addresses 3 and 6. Let denote the $|L^1_{\bar{v}}|$ values at spiral addresses 1 and 4 by $|L^{1-}_{\bar{v}}|$ and $|L^{1+}_{\bar{v}}|$ respectively. Similarly, we denote the $|L^2_{\bar{v}}|$ values at spiral addresses 2 and 5 by $|L^{2-}_{\bar{v}}|$ and $|L^{2+}_{\bar{v}}|$ respectively, and denote the $|L^3_{\bar{v}}|$ values at spiral addresses 3 and 6 by $|L^{3-}_{\bar{v}}|$ and $|L^{3+}_{\bar{v}}|$ respectively.

Without loss of generality, let assume that the gradient $L_{\bar{v}}$ is closest to (0,1) among the three diagonal directions, then the gradient magnitude $|L_{\bar{v}}|$ is close to $|L^1_{\bar{v}}|$. Furthermore, at the gradient direction, the two neighbouring pixels of the central pixel are the pixels with spiral addresses 1 and 4. Note that the gradient magnitudes at these two pixels are also close to $|L^{1-}_{\bar{v}}|$ and $|L^{1+}_{\bar{v}}|$. Since $|L^1_{\bar{v}}| > |L^{1+}_{\bar{v}}|$ and $|L^1_{\bar{v}}| > |L^{1+}_{\bar{v}}|$, the gradient magnitude at the central pixel assumes a local maximum at the gradient direction. □

Theorem 2. Almost *any edge point as defined in Definition 1 is an edge point of type 3.*

Proof. If the central pixel with spiral address 0 in Figure 3 is not an edge point of type 3, then the following three cases are all met.

1. Either $|L^1_{\bar{v}}| \le |L^{1+}_{\bar{v}}|$ or $|L^1_{\bar{v}}| \le |L^{1+}_{\bar{v}}|$.
2. Either $|L^2_{\bar{v}}| \le |L^{2+}_{\bar{v}}|$ or $|L^2_{\bar{v}}| \le |L^{2+}_{\bar{v}}|$.
3. Either $|L^3_{\bar{v}}| \le |L^{3+}_{\bar{v}}|$ or $|L^3_{\bar{v}}| \le |L^{3+}_{\bar{v}}|$.

Hence, similar to the proof of Theorem 1, we can claim that the central pixel is not a edge point as defined in Definition 1. □

Based on Theorems 1 & 2, we can claim that an edge map of type 1 (Figure 6a) contains minimum number of edge points, and the edge map of type 3 (Figure 6c) contains maximum edge points.

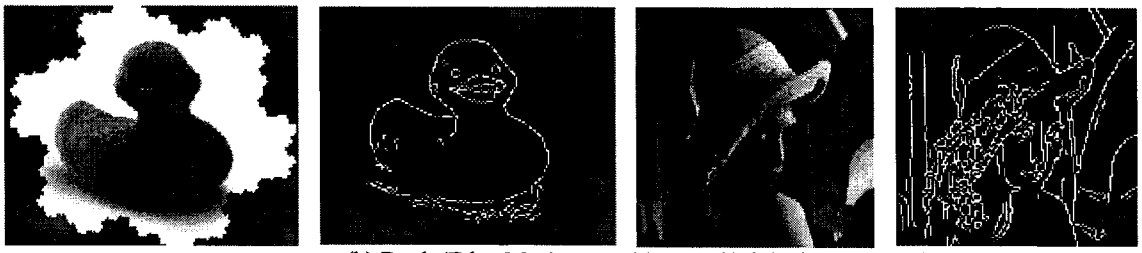
The conditions for a type 1 edge map are too strict to cover all the edge points. There are too many edge points filtered and only a small amount of edge points added to the edge map. On the other hand, the edge map of type 3 may contain some points that are less significant or are false (and should not be classified as) edge points.

In our work in this paper, a method to generate contours from edge points called *linking edge* [9] is adopted. It works together with edge maps of all three types. Our edge detection scheme tends to find all significant edge points while removing as many false points as possible. The procedure for linking edge is presented as follows.

1. We start from the edge map of type 1. We look at all neighbouring points of any edge point of type 1.

2. If any neighbouring point of a type 1 edge point is found in the edge map of type 2, it is added to the edge map obtained in Step 1 above. This rocess continues until no more neighbouring points of any point in the most updated edge map can be found in the edge map of type 2, and hence can be added to the existing edge map.

3. A similar process to Step 2 above is performed. This time, we add the edge points of type 3 to the edge map obtained in Step 2. If any neighbouring point of an edge point in the existing edge map is found in the edge map of type 3, it is added to the existing edge map. This process continues until no more neighbouring points of any point in the



(a) Duck (Original Image) (b) Duck (Edge Map) (c) Lena (Original Image) (d) Lena (Edge Map)

Figure 7: Triple-diagonal gradient-based edge detection

most updated edge map can be found in the edge map of type 3.

Shaving edge is the step immediate after the linking edge procedure described above. This step further improves the quality of edge map. It consists of two operations: *eliminating single points* and *thinning edges*. There may still be some disassociated points that are not connected to any other edge points after linking edge procedure. They can be regarded as noise points and hence are removed immediately from the edge map obtained from Step 3 above. In addition, thinning edge narrows any edge segment (a set of contiguous edge points) to one pixel wide. Thinning edge is important because object shape analysis [9] benefits from single-pixel-wide object boundaries.

4. Experimental Results

In our experiment, the proposed algorithm is tested using two representative images, toy duck and Lena, of which both are 256 grey level images.

We first mapped these two images to Spiral Architecture as shown in Figure 7a and Figure 7c. They were then blurred using a Gaussian Multi-scale process [7] to suppress noise and insignificant edge points. This was followed by the triple-diagonal gradient-based edge detection process including a shaving edge step.

Experimental results show that our new edge detection algorithm has successfully extracted high-quality object edges as shown in Figure 7b and Figure 7d.

5. Conclusion

In this paper, a triple-diagonal gradient-based edge detection scheme has been presented. This novel algorithm is based on the features of Spiral Architecture. It also required the properties of the hexagonal (mimic) array [6]. The gradient components in three diagonal directions were used to define three types of edge points. Based on the edge maps of three types and the techniques used for suppressing image noise, linking edge points, eliminating edge points and thinning edge segments, high-quality single-pixel-wide object edge maps have been obtained.

Our new edge detection scheme has greatly improved the accuracy for locating edge points. It does not need any supplementary processing to enhance the edge map.

Furthermore, our new method can eliminate most of the noise points and other trivial edge points while retaining key points as well as fine shape information of objects in an image. For example, the folds of feathers on the tail of the duck can still be seen clearly in the edge map (Figure 7b).

If one wants to further eliminate the false edge points caused by noise or computation errors, a threshold on gradient magnitude can be applied. Any edge points whose gradient amplitudes are below a threshold such as 10% of maximum gradient magnitude can be eliminated or removed from the edge map. The threshold can be estimated using different methods including histogram analysis or other statistical methods.

References

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

[2] L. S. Davis, A Survey of Edge Detection Techniques, *Computer Graphics and Image processing*, vol. 4, pp. 248-270, 1975.

[3] A. Huertas and G. Medioni, Detection of Intensity Changes with Subpixel Accuracy Using Laplacial-Gaussian Masks, *IEEE-T-PAMI*, Vol. 8(5), pp. 651-664, 1986.

[4] P. Sheridan, T. Hintz, and W. Moore, Spiral Architecture in Machine Vision, *Proceedings of Australian Occam and Transputer Conference*, 1991.

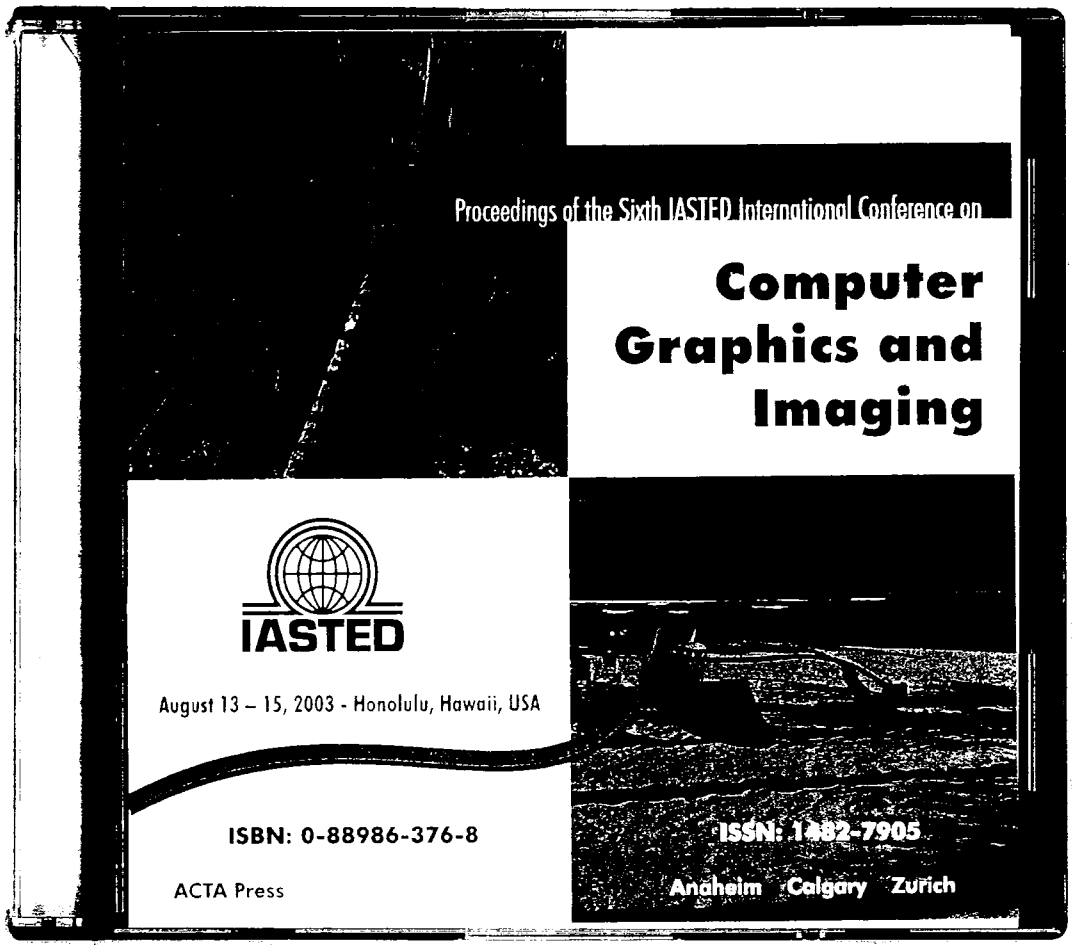
[5] E. Schwartz, Computational Anatomy and Functional Architecture of Striate Cortex: A Spatial Mapping Approach to Perceptual Coding, *Vision Research 20*, pp. 645-669, 1980.

[6] P. Sheridan, T. Hintz, and D. Alexander, Pseudo-invariant Image Transformations on a Hexagonal Lattice, *Image and Vision Computing*, vol. 18, pp. 907-917, 2000.

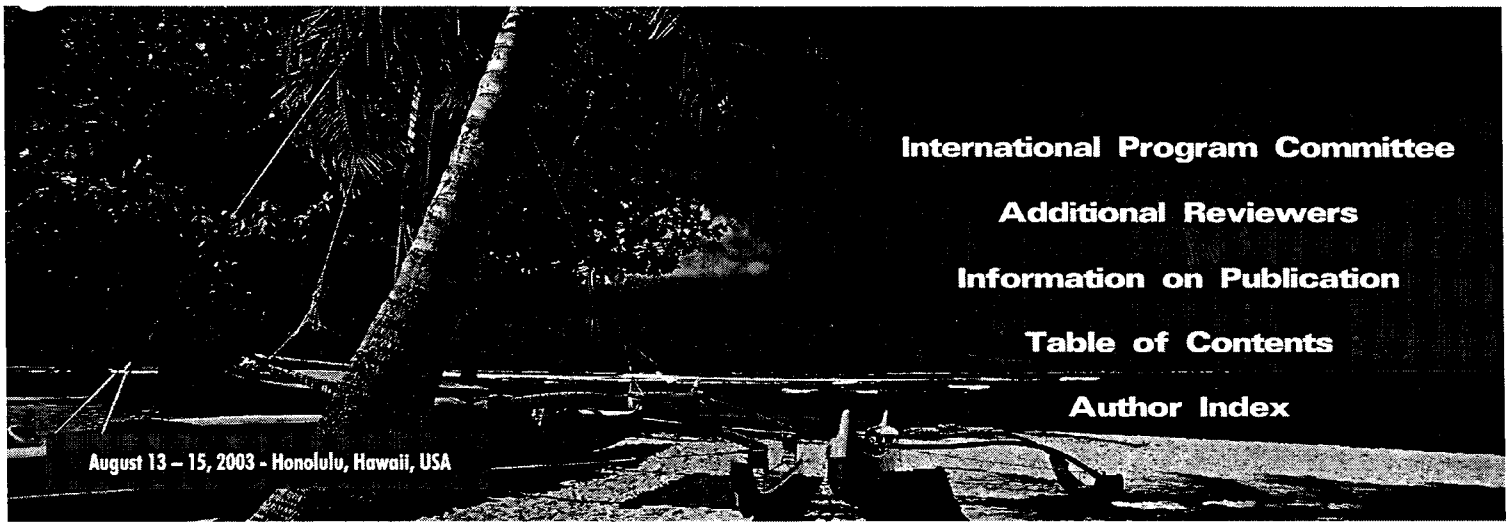
[7] X. He, *2D-Object Recognition with Spiral Architecture* (PhD Thesis, University of Technology, Sydney, 1998).

[8] Xiangjian He, Tom Hintz and Qiang Wu, Neural Network Based Image Edge Detection within Spiral Architecture, *Proc. International Conference on Parallel and Distributed Processing Techniques and Applications*, Las Vegas, 2002, pp.21-27.

[9] Q. Wu, T. Hintz, and X. He, Number Recognition Using Integral Invariants on Spiral Architecture, *Proceedings of CISST'02*, Las Vegas, US, pp. 7-13, 2002.


Proceedings of the Sixth IASTED International Conference on
Computer Graphics and Imaging
IASTED
August 13 – 15, 2003 - Honolulu, Hawaii, USA
ISBN: 0-88986-376-8
ACTA Press
ISSN: 1482-7905
Anaheim Calgary Zurich

A Publication of The International Association of Science and Technology for Development — IASTED



COMPUTER GRAPHICS AND IMAGING

Editor: M.H. Hamza

IASTED

ACTA Press Anaheim • Calgary • Zurich

Proceedings of the Sixth IASTED International Conference on **Computers, Graphics, and Imaging**, held August 13 – 15, 2003, Honolulu, Hawaii, USA.

Sponsors

The International Association of Science and Technology for Development (IASTED)

- Technical Committee on Graphics
- Technical Committee on Image Processing

Editor

M.H. Hamza

International Program Committee

Additional Reviewers

For each IASTED conference, the following review process is used to ensure the highest level of academic content. Each full manuscript submission is peer reviewed by a minimum of two separate reviewers on the International Program Committee / Additional Reviewers list. The review results are then compiled. If there is a conflict in reviews, the paper is sent to a third reviewer.

ACTA Press
P.O. Box 5124
Anaheim, CA 92814-5124
USA

ACTA Press
#80, 4500 – 16th Ave. NW
Calgary, AB T3B 0M6
Canada

ACTA Press
P.O. Box 354
CH-8053 Zurich
Switzerland

Publication Code: 398