features. By directly using part or all of the run-length matrix as a feature vector, much of the texture information is preserved. This approach is made possible by the utilization of the multilevel dominant eigenvector estimation method, which reduces the computation complexity of KLT by several orders of magnitude. Combined with the Bhattacharyya measure, they form an efficient feature selection algorithm.

The advantage of this approach is demonstrated experimentally by the classification of two independent texture data sets. Experimentally, we also observe that most texture information is stored in the first few columns of the run-length matrix, especially in the first column. This observation justifies development of a new, fast, parallel run-length matrix computation scheme.

Comparisons of this new approach with the co-occurrence and wavelet features demonstrate that the run-length matrices possess as much discriminatory information as these successful conventional texture features and that a good method of extracting such information is key to the success of the classification. We are currently investigating the application of the new feature extraction approach on other texture matrices. We hope our work here will renew interest in run-length texture features and promote future applications.

References

[1] P. Brodatz, *Textures: A Photographic Album for Artists and Designers.* New York: Dover, 1966.

[2] T. Chang and C.-C. J. Kuo, "Texture analysis and classification with tree-structured wavelet transform," *IEEE Trans. Image Processing,* vol. 2, pp. 429–441, Oct. 1993.

[3] A. Chu, C. M. Sehgal, and J. F. Greenleaf, "Use of gray value distribution of run lengths for texture analysis," *Pattern Recognit. Lett.,* vol. 11, pp. 415–420. June 1990.

[4] R. W. Conners and C. A. Harlow, "A theoretical comparison of texture algorithms," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. 2, pp. 204–222, May 1980.

[5] B. R. Dasarathy and E. B. Holder, "Image characterizations based on joint gray-level run-length distributions," *Pattern Recognit. Lett.,* vol. 12, pp. 497–502, 1991.

[6] K. Fukunaga, *Introduction to Statistical Pattern Recognition.* New York: Academic, 1972.

[7] M. M. Galloway, "Texture analysis using gray level run lengths," *Comput. Graphics Image Process.,* vol. 4, pp. 172–179, June 1975.

[8] R. M. Haralick, K. S. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Trans. Syst., Man, Cybern.,* vol. SMC-3, pp. 610–621, 1973.

[9] M. E. Jernigan and F. D'Astous, "Entropy-based texture analysis in the spatial frequency domain," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. PAMI-6, pp. 237–243, Mar. 1984.

[10] A. Laine and J. Fan, "Texture classification by wavelet packet signatures," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. 15, pp. 1186–1191, Nov. 1993.

[11] S. Liu and M. E. Jernigan, "Texture analysis and discrimination in additive noise," *Comput. Vis., Graph., Image Process.,* vol. 49, pp. 52–67, 1990.

[12] X. Tang, "Dominant run-length method for image classification," Woods Hole Oceanog. Inst., Tech. Rep. WHOI-97–07, June 1997.

[13] C. W. Therrien, *Decision Estimation and Classification, An Introduction to Pattern Recognition and Related Topics.* New York: Wiley, 1989.

[14] M. Unser and M. Eden, "Multiresolution feature extraction and selection for texture segmentation," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. 11, pp. 717–728, July 1989.

[15] J. S. Weszka, C. R. Dyer, and A. Rosenfeld, "A comparative study of texture measures for terrain classification," *IEEE Trans. Syst., Man, Cybern.,* vol. SMC-6, pp. 269–285, 1976.

Real-Time Computation of Two-Dimensional Moments on Binary Images Using Image Block Representation

Iraklis M. Spiliotis and Basil G. Mertzios

***Abstract*—This work presents a new approach and an algorithm for binary image representation, which is applied for the fast and efficient computation of moments on binary images. This binary image representation scheme is called *image block representation*, since it represents the image as a set of nonoverlapping rectangular areas. The main purpose of the image block representation process is to provide an efficient binary image representation rather than the compression of the image. The block represented binary image is well suited for fast implementation of various processing and analysis algorithms in a digital computing machine. The two-dimensional (2-D) statistical moments of the image may be used for image processing and analysis applications. A number of powerful shape analysis methods based on statistical moments have been presented, but they suffer from the drawback of high computational cost. The real-time computation of moments in block represented images is achieved by exploiting the rectangular structure of the blocks.**

***Index Terms*—Binary image, image analysis, image block representation, moments.**

I. Introduction

The most common image representation format is a two-dimensional (2-D) array, each element of which has the brightness value of the corresponding pixel. For a binary image these values are zero or one. In a serial machine, only one pixel is to be processed at a time, by using the 2-D array representation. Many research efforts have considered the problem of selecting an image representation suitable for concurrent processing in a serial machine. The need for such approaches arises from the fact that an image contains a great amount of information, thus rendering the processing a difficult and slow task. Existing approaches to image representation aim to provide machine perception of images in pieces larger than a pixel and are separated in two categories: 1) boundary-based methods and 2) region-based methods. Such approaches include quadtree representations [1], chain code representations [2], contour control point models [3], autoregressive models [4], the interval coding representation [5], and block implementation techniques [6]–[8]. One common objective of the above methods is the representation of an image in a more suitable form for a specific operation.

This correspondence presents a new advantageous representation for binary images called *image block representation* (IBR) and constitutes an efficient tool for image processing and analysis techniques [9], [10]. Using the block represented binary images, real-time computation of 2-D statistical moments is achieved through analytical formulae. The computational complexity of the proposed technique is $O(L^2)$, where $(L-1, L-1)$ is the order of the 2-D moments to be computed.

Various sets of 2-D statistical moments constitute a well-known image analysis and pattern recognition tool [11]–[20]. In pattern recognition applications, a small set of the lower order moments is

Manuscript received June 4, 1994; revised September 18, 1995. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. John Juyang Weng.

The authors are with the Automatic Control Systems Laboratory, Department of Electrical and Computer Engineering, Democritus University of Thrace, 67100 Xanthi, Greece (e-mail: spiliot@demokritos.cc.duth.gr; mertzios@demokritos.cc.duth.gr).

Publisher Item Identifier S 1057-7149(98)07752-5.

used to discriminate among different patterns. The most common moments are geometrical moments, central moments, normalized central moments, and moments invariants [17], [18]. Other sets of moments are Zernike moments and Legendre moments (which are based on the theory of orthogonal polynomials) [19], [21], and complex moments [20]. One main difficulty concerning the use of moments as features in image analysis applications is the implied high computational time. A number of approaches that reduce the computational time concerning calculation of moments have appeared [3], [23]–[25]. In [23]–[25], the problem has been reduced from 2-D to a one-dimensional (1-D) one, using Green's theorem; this approach reduces the complexity from $O(N^2)$ to $O(N)$, since only the boundary pixels are considered and the length P of the boundary is linearly related to $\sqrt{A}$, where A is the object area. In [3] control point models based on the least-square normalized B-splines are used for the representation of the object boundary, where the complexity of the moments computation is analogous to the shape model order and independent of the scale. The computational cost of this method is comparable with the cost of the proposed method, but mainly due to the deviations of the boundary representation model, the moment values are significantly affected. In [26], the computation formula of each one central moment has been considered as an impulse response of a filter, which is then transformed to the z-domain and the transfer function of the corresponding digital filter is obtained. This latter approach is also inferior to block-based computation, since it is dependent on the image size and its computational complexity for the calculation of the 16 central moments up to the order (4, 4) of an image with $N \times N$ points, is $4N^2 + 16N + 80$ additions and only 32 multiplications or power calculations.

II. Image Block Representation

A bilevel digital image is represented by a binary 2-D array. Without loss of generality, we suppose that the object pixels are assigned to level one and the background pixels to level zero. Due to this kind of representation, there are rectangular areas of object value one, in each image. These rectangulars, which are called *blocks* in the terminology of this work, have their edges parallel to the image axes and contain an integer number of image pixels. At the extreme case, the minimum rectangular area of the image is one pixel.

Consider a set that contains as members all the nonoverlapping blocks of a specific binary image, in such a way that no other block can be extracted from the image (or equivalently, each pixel with object level belongs to only one block). It is always feasible to represent a binary image with a set of all the nonoverlapping blocks with object level, and this representation is IBR. According to the above discussion, two useful definitions concerning IBR are formulated.

Definition 1: Block is called a rectangular area of the image, with edges parallel to the image axes, that contains pixels of the same value. ■

Definition 2: A binary image is called *block represented,* if it is represented as a set of blocks with object level, and if each pixel of the image with object value belongs to one and only one block. ■

According to Definitions 1 and 2, it is concluded that the IBR is an information lossless representation. Given a specific binary image, different sets of different blocks can be formed. Actually, the nonunique block representation does not have any implications on the implementation of any operation on a block represented image.

The IBR concept leads to a simple and fast algorithm, which requires just one pass of the image and simple bookkeeping process. In fact, considering a $N_1 \times N_2$ binary image $f(x, y)$, $x = 0, 1, \cdots, N_1 - 1$, $y = 0, 1, \cdots, N_2 - 1$, the block extraction process requires a pass from each line y of the image. In this pass, all object level intervals are extracted and compared with the previous extracted blocks. In the following, an IBR algorithm is given.

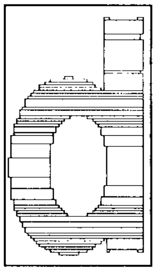


Fig. 1. Image of the character d and the blocks.

Algorithm 1—Image Block Representation:

```
Step 1: Consider each line $y$ of the image $f$ and find the object
        level intervals in line $y$.
Step 2: Compare intervals and blocks that have pixels in line $y-1$.
Step 3: If an interval does not match with any block, this is the
        beginning of a new block.
Step 4: If a block matches with an interval, the end of the block
        is in the line $y$. ■
```

As a result of the application of the above algorithm, we obtain a set of all the rectangular areas with level one that form the object. A block represented image is denoted as

$$f(x, y) = \{b_i : i = 0, 1, \cdots, k-1\} \tag{1}$$

where k is the number of the blocks. Each block is described by four integers, the coordinates of the upper left and down right corner in vertical and horizontal axes. The block extraction process is implemented easily with low computational complexity, since it is a pixel checking process without numerical operations. Fig. 1, illustrates the blocks that represent an image of the character d.

The optimum representation is characterized by the minimum possible number of blocks. Different IBR algorithms, which may result to smaller number of blocks at the cost of the increased required time, may be implemented. Specifically, the algorithm for finding the maximal empty rectangle [27]–[29] in an area may be applied recursively: at each stage the points of this rectangle are labeled as background points, in order to apply the algorithm recursively to the remaining points. The time complexity of this method is $O(n \log n)$, where n is the number of the points of the area at each recursion. Therefore, the selection of the optimum representation implies an additional computational cost, which may compensate the achieved savings due to the minimum number of blocks.

III. Computation of Moments

A. Geometrical Moments

Consider a binary digital image $f(x, y)$, with N_1 pixels in horizontal axis and N_2 pixels in vertical axis. The 2-D geometrical moments of order (p, q) of the image are defined by the relation

$$m_{pq} = \sum_{x=0}^{N_1-1} \sum_{y=0}^{N_2-1} x^p y^q f(x, y), \qquad p, q = 0, 1, 2, \cdots. \tag{2}$$

Since the background level is zero, only the pixels with level one are taken into account for the computation of the moments. Thus, the 2-D geometrical moments of order (p, q) of the image $f(x, y)$ are defined by the relation

$$m_{pq} = \sum_x \sum_y x^p y^q \qquad \forall\, x, y\colon f(x, y) = 1. \tag{3}$$

Specifically, if the image $f(x, y)$ is represented by k blocks, as it is described in (1), all the image pixels with level 1 belong to the k image blocks and therefore (3) may be rewritten as

$$m_{pq} = \sum_{i=0}^{k-1} m_{pq}^{b_i} = \sum_{i=0}^{k-1} \sum_{x=x_{1,b_i}}^{x_{2,b_i}} \sum_{y=y_{1,b_i}}^{y_{2,b_i}} x^p y^q \tag{4}$$

where x_{1,b_i}, x_{2,b_i} and y_{1,b_i}, y_{2,b_i} are the coordinates of the block b_1 with respect to the horizontal axis and to the vertical axis, respectively. In (4), if the rectangular form appeared within the blocks is taken into account, then the geometrical moments of one block b, with coordinates $x_{1b}, x_{2b}, y_{1b}, y_{2b}$, are given by

$$\begin{aligned} m_{pq}^b &= \sum_{x=x_{1b}}^{x_{2b}} \sum_{y=y_{1b}}^{y_{2b}} x^p y^q = x_{1b}^p \sum_{y=y_{1b}}^{y_{2b}} y^q + (x_{1b}+1)^p \sum_{y=y_{1b}}^{y_{2b}} y^q \\ &\quad + \cdots + x_{2b}^p \sum_{y=y_{1b}}^{y_{2b}} y^q = \left(\sum_{x=x_{1b}}^{x_{2b}} x^p \right) \left(\sum_{y=y_{1b}}^{y_{2b}} y^q \right). \end{aligned} \tag{5}$$

Using the rectangular form appeared within the block, the computational effort, which is characterized by the complexity $O(N^2)$ for the calculation of moments using (2), is reduced to $O(N)$ for the calculation of moments using (5). For the computation of (5), it is adequate to calculate the following summations of the powers of x and y:

$$S_{x_{1b},\, x_{2b}}^p = \sum_{x=x_{1b}}^{x_{2b}} x^p, \; S_{y_{1b},\, y_{2b}}^q = \sum_{y=y_{1b}}^{y_{2b}} y^q, \qquad x,\, y,\, p,\, q \in Z. \tag{6}$$

Moreover, taking into account the known formulae

$$\begin{aligned} S_{1,\,n}^1 &= \frac{n(n+1)}{2}, \qquad S_{1,\,n}^2 = \frac{n(n+1)(2n+1)}{6}, \\ S_{1,\,n}^3 &= \frac{n^2(n+1)^2}{4}, \qquad S_{1,\,n}^4 = \frac{n(n+1)(2n+1)(3n^2+3n+1)}{30} \end{aligned} \tag{7}$$

and, in the general case for sums of powers greater than four, the formula

$$\begin{aligned} &\binom{m+1}{1} S_{1,\,n}^1 + \binom{m+1}{2} S_{1,\,n}^2 + \cdots + \binom{m+1}{m} S_{1,\,n}^m \\ &\quad = (n+1)^{m+1} - (n+1) \end{aligned} \tag{8}$$

where $m,\, n \in Z$ and $\binom{i}{j} = (i!)/[j!(i-j)!]$ are with combinations of i objects, taken j each time, it is concluded that the summation $S_{x_1,\, x_2}^p$ can be directly calculated by the analytical formulae (9), shown at the bottom of the page.

The summation $S_{y_{1b},\, y_{2b}}^q$ is computed in a similar manner. Fast computation of the 2-D geometrical meoments of one block, according to (5), is achieved using the above simple and analytical formulae.

According to (4), the 2-D geometrical moments of the whole image are computed as the summation of the 2-D geometrical moments of all the individual blocks of the binary image.

B. Central Moments

The 2-D central moments of an image $f(x,\, y)$ are invariant under image translation and they are defined as

$$\mu_{pq} = \sum_{x=0}^{N_1-1} \sum_{y=0}^{N_2-1} (x-\overline{x})^p (y-\overline{y})^q f(x,\, y) \tag{10}$$

where $\overline{x} = m_{10}/m_{00}$, $\overline{y} = m_{01}/m_{00}$ are the coordinates of the centroid.

Since all the image pixels with level one belong to the k image blocks, (10) may be rewritten as

$$\mu_{pq} = \sum_{i=0}^{k-1} \mu_{pq}^{b_i} = \sum_{i=0}^{k-1} \sum_{x=x_{1,\,b_i}}^{x_{2,\,b_i}} \sum_{y=y_{1,\,b_i}}^{y_{2,\,b_i}} (x-\overline{x})^p (y-\overline{y})^q \tag{11}$$

where $x_{1,\,b_i}$, $x_{2,\,b_i}$, $y_{1,\,b_i}$, $y_{2,\,b_i}$ are with coordinates of the block. The coordinates in (11) refer to the center of gravity of the whole image and not to the centroid of each block. The computation of the geometrical moments $m_{00},\, m_{10},\, m_{01}$ using (5) and (9) ensures the fast computation of the centroid of the image. In (11), if the rectangular form appeared within the blocks is taken into account, then the moments of one block b, with coordinates $x_{1b},\, x_{2b},\, y_{1b},\, y_{2b}$, are given by

$$\begin{aligned} \mu_{pq}^b &= \sum_{x=x_{1b}}^{x_{2b}} \sum_{x=y_{1b}}^{y_{2b}} (x-\overline{x})^p (y-\overline{y})^q \\ &= \left(\sum_{x=x_{1b}}^{x_{2b}} (x-\overline{x})^p \right) \left(\sum_{x=y_{1b}}^{y_{2b}} (y-\overline{y})^q \right). \end{aligned} \tag{12}$$

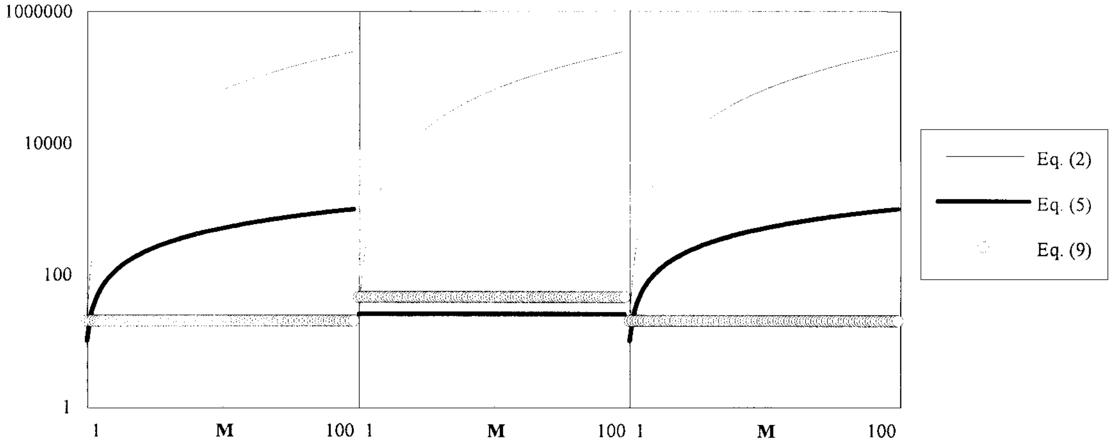
The complexity is reduced from $O(N^2)$ to $O(N)$. For the computation of (12), it is adequate to calculate the two summations of the product. Using the mathematical identities

$$\begin{aligned} (c-d)^2 &= c^2 - 2cd + d^2 \\ (c-d)^3 &= c^3 - 3c^2 d + 3cd^2 - d^3 \\ (c-d)^4 &= c^4 - 4c^3 d + 6c^2 d^2 - 4cd^3 + d^4 \\ (c-d)^m &= c^m - \binom{m}{1} c^{m-1} d + \binom{m}{2} c^{m-2} d \\ &\quad - \cdots + (-1)^m\, d^m \end{aligned} \tag{13}$$

$$\begin{aligned} S_{x_{1b},\, x_{2b}}^1 &= S_{1,\, x_{2b}}^1 - S_{1,\, x_{1b}-1}^1 = \frac{x_{2b}(x_{2b}+1) - x_{1b}(x_{1b}-1)}{2} \\ S_{x_{1b},\, x_{2b}}^2 &= S_{1,\, x_{2b}}^2 - S_{1,\, x_{1b}-1}^2 = \frac{x_{2b}(x_{2b}+1)(2x_{2b}+1) - x_{1b}(x_{1b}-1)(2x_{1b}-1)}{6} \\ S_{x_{1b},\, x_{2b}}^3 &= S_{1,\, x_{2b}}^3 - S_{1,\, x_{1b}-1}^3 = \frac{x_{2b}^2(x_{2b}+1)^2 - x_{1b}^2(x_{1b}-1)^2}{4} \\ S_{x_{1b},\, x_{2b}}^4 &= S_{1,\, x_{2b}}^4 - S_{1,\, x_{1b}-1}^4 = \frac{x_{2b}(x_{2b}+1)(2x_{2b}+1)(3x_{2b}^2+3x_{2b}-1) - x_{1b}(x_{1b}-1)(2x_{1b}-1)(3x_{1b}^2+3x_{1b}-1)}{30} \\ S_{x_{1b},\, x_{2b}}^p &= \frac{(x_{2b}+1)^{p+1} - x_{1b}^{p+1} - (x_{2b}-x_{1b}+1) - \binom{p+1}{1} S_{x_{1b},\, x_{2b}}^1 - \binom{p+1}{2} S_{x_{1b},\, x_{2b}}^2 - \cdots - \binom{p+1}{p-1} S_{x_{1b},\, x_{2b}}^{p-1}}{p+1}, \qquad \forall\, p \in Z^+ \end{aligned} \tag{9}$$

TABLE I
REQUIRED NUMBER OF OPERATIONS FOR THE COMPUTATION OF GEOMETRICAL MOMENTS UP TO THE ORDER $(L-1,\ L-1)$, OF ONE BLOCK WITH $M \times M$ PIXELS

Operations Number	Direct computation from equation (2)	Computation from equation (5)	Computation from equation (9)
power calculations	L^2M^2	$2LM$	$4L$
multiplications	L^2M^2	L^2	$2L^2$-L
additions	L^2M^2	$2LM$	L^2-L



(a) (b) (c)

Fig. 2. Number of operations for the geometrical moments computation up to the order $(L-1,\ L-1)$, of a $M \times M$ block, from (2), (5) and (9) with $L = 5$ and $M = 1, 2, \cdots, 100$: (a) number of power calculations; (b) number of multiplications; (c) number of additions.

it is concluded that

$$\begin{aligned}
\sum_{x=1}^{x_{2b}} (x-\overline{x}) &= S^1_{1,\,x_{2b}} - x_{2b}\overline{x},\\
\sum_{x=1}^{x_{2b}} (x-\overline{x})^2 &= S^2_{1,\,x_{2b}} - 2\overline{x}S^1_{1,\,x_{2b}} + x_{2b}\overline{x}^2\\
\sum_{x=1}^{x_{2b}} (x-\overline{x})^3 &= S^3_{1,\,x_{2b}} + 3\overline{x}^2 S^1_{1,\,x_{2b}} - x_{2b}\overline{x}^3,\\
\sum_{x=1}^{x_{2b}} (x-\overline{x})^4 &= S^4_{1,\,x_{2b}} - 4\overline{x}S^2_{1,\,x_{2b}} + 6\overline{x}^2 S^2_{1,\,x_{2b}}\\
\sum_{i=1}^{x_{2b}} (x-\overline{x})^p &= S^p_{1,\,x_{2b}} - \overline{x}\binom{p}{1}S^{p-1}_{1,\,x_{2b}} + \cdots + (-1)^p x_{2b}\overline{x}^p,\\
&\forall p \in Z^+ \qquad (14)
\end{aligned}$$

and

$$\begin{aligned}
\sum_{x=x_{1b}}^{x_{2b}} (x-\overline{x}) &= \sum_{x=1}^{x_{2b}} (x-\overline{x}) - \sum_{x=1}^{x_{1b}-1} (x-\overline{x})\\
&= S^1_{x_{1b},\,x_{2b}} - (x_{2b} - x_{1b} + 1)\overline{x}\\
\sum_{x=x_{1b}}^{x_{2b}} (x-\overline{x})^2 &= \sum_{x=1}^{x_{2b}} (x-\overline{x})^2 - \sum_{x=1}^{x_{1b}-1} (x-\overline{x})^2\\
&= S^2_{x_{1b},\,x_{2b}} - 2\overline{x}S^1_{x_{1b},\,x_{2b}} + (x_{2b} - x_{1b} + 1)\overline{x}^2\\
\sum_{x=x_{1b}}^{x_{2b}} (x-\overline{x})^3 &= \sum_{x=1}^{x_{2b}} (x-\overline{x})^3 - \sum_{x=1}^{x_{1b}-1} (x-\overline{x})^3\\
&= S^3_{x_{1b},\,x_{2b}} - 3\overline{x}S^2_{x_{1b},\,x_{2b}} + 3\overline{x}^2 S^1_{x_{1b},\,x_{2b}}\\
&\quad - (x_{2b} - x_{1b} + 1)\overline{x}^3\\
\sum_{x=x_{1b}}^{x_{2b}} (x-\overline{x})^4 &= \sum_{x=1}^{x_{2b}} (x-\overline{x})^4 - \sum_{x=1}^{x_{1b}-1} (x-\overline{x})^4\\
&= S^4_{x_{1b},\,x_{2b}} - 4\overline{x}S^3_{x_{1b},\,x_{2b}} + 6\overline{x}^2 S^2_{x_{1b},\,x_{2b}}\\
&\quad - 4\overline{x}^3 S^1_{x_{1b},\,x_{2b}} + (x_{2b} - x_{1b} + 1)\overline{x}\\
\sum_{x=x_{1b}}^{x_{2b}} (x-\overline{x})^p &= S^p_{x_{1b},\,x_{2b}} - \overline{x}\binom{p}{1}S^{p-1}_{x_{1b},\,x_{2b}} + \cdots\\
&\quad + (-1)^p (x_{2b} - x_{1b} + 1)\overline{x}^p, \qquad \forall p \in Z^+
\end{aligned} \qquad (15)$$

and where $S^p_{x_{1b},\,x_{2b}}$ have been calculated from (9).

The above analytical formulae (15) are used for the fast computation of the factor $\sum_{x=x_{1b}}^{x_{2b}}$ of the central moments (12) of the block b. The factor $\sum_{y=y_{1b}}^{y_{2b}}(y-\overline{y})^q$, appeared in (12) is calculated in a similar manner. The fast computation of the central moments of each block according to the proposed method ensures the fast computation of the central moments of the whole image, according to (11).

C. Normalized Central Moments and Moments Invariants

The 2-D normalized central moments of an image $f(x,\,y)$ are defined as $\eta_{pq} = \mu_{pq}/\mu_{00}^{\gamma}$ where $\gamma = [(p+q)/2] + 1,\ p+q = 2, 3, \cdots$ and μ_{pq} are the corresponding central moments of the image. The central moments are required for the computation of the normalized central moments.

A set of seven moments, which are invariant to translation, rotation, and scaling factors, called *moments invariants* [17], [18] is derived from the normalized central moments. Therefore, the fast computation of the central moments ensures the fast computation of the normalized central moments and the moments invariants.

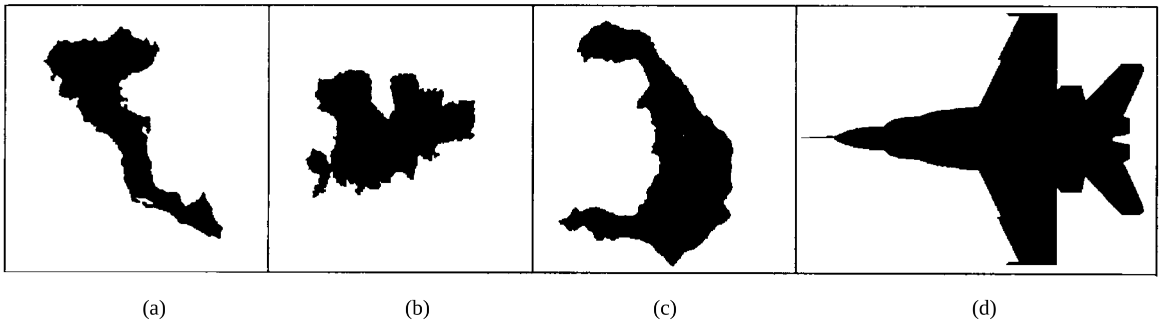


Fig. 3. Set of test images. (a) Image of the island of Corfu of 512 × 512 pixels. (b) Image of the island of Mikonos of 512 × 512 pixels. (c) Image of the island of Santorini of 512 × 512 pixels. (d) Aircraft image of 512 × 697 pixels.

IV. Computational Complexity

It is clear from the IBR algorithm that block extraction is a pixel checking process, without involving any numerical operations and requires only one pass from each point of the image. Therefore, IBR is fast implemented and the required time is proportional to the image size. However, in pattern recognition applications, the IBR is applied to the images of the separated objects rather than to the whole image. In a number of such applications, it resulted that the computation time for block extraction is much less than the time for image file reading and image segmentation.

Consider that a binary image contains one rectangular block with level one. For simplicity and without loss of generality, suppose a square block with $M \times M$ points. In the sequel we estimate the computational complexity required for the computation of the geometrical moments of order up to $(L-1, L-1)$. The analysis concerning the computational complexity of other sets of moments may be given in a similar manner.

It is seen from (2) that the direct computation of one geometrical moment requires M^2 power computations, M^2 multiplications and M^2 additions. For the computation of L^2 moments, L^2M^2 power computations, L^2M^2 multiplications L^2M^2 and additions are required.

Consider (5), which exploits the rectangular form appearing within the block. For the computation of the factor $S^p_{x_{1b}, x_{2b}}$, LM power calculations and LM additions are required. The same number of operations are required for the computation of the factor $S^q_{y_{1b}, y_{2b}}$. Therefore, for the computation of L^2 geometrical moments using (5), $2LM$ power calculations, L^2 multiplications and $2LM$ additions, are required for one block.

Now, consider the analytical formula (9), where the same binomial coefficients appear for any specific geometrical moment of each block of the image; therefore, the corresponding computational effort is reduced by the number of the blocks. Moreover, the factorials for the determination of the binomial coefficients require the least effort, i.e., one multiplication for the calculation of each one of the factorial $m!$ in terms of $(m-1)!$.

There are two alternative approaches for the execution of the power calculations in (9). In the first one, the two power calculations are considered. Alternatively, the calculation of x^p_{1b} and $(x_{2b}+1)^{p+1}$ may be seen as two multiplications in terms of x^p_{1b} and $(x_{2b}+1)^{p+1}$, respectively. In the following analysis, the first approach is used. The factors $S^i_{x_{1b}, x_{2b}}$, $i = 1, 2, \cdots, p-1$, that have been computed previously and their values are stored, are used for the computation of the $S^p_{x_{1b}, x_{2b}}$. Thus, the computation of the $S^p_{x_{1b}, x_{2b}}$ from (9) requires two power calculations, p multiplications, and p additions. The computation of the sum of $S^q_{y_{1b}, y_{2b}}$ requires two power calculations, q multiplications, and q additions. The calculation of all the L^2 moments requires $4L$ power calculations, $2L^2 - L$ multiplications, and $L^2 - L$ additions.

Table I demonstrates the above results. The complexity is reduced from 2-D form from (2) to 1-D using image block representation and (5). Moreover, it results that the complexity is independent of the size when the analytical formula (9) is used. The required number of power calculations, multiplications and additions for the computation of the geometrical moments up to the order (4, 4) of a block with $M \times M$ pixels, where M varies from 1–100, is shown in Fig. 2.

Lemma 1: Assuming that the complexity of raising a number to a power is the same as one multiplication, the computation of (5) and (9) requires $L^2 + 2LM$ and $3L^2 + 2L$ multiplications, respectively. Comparing the above number of multiplications, it is concluded that (5) has less computational complexity than (9) when

$$L^2 + 2LM \le 2L^2 + 3L \Rightarrow M \le \frac{L+3}{2}. \qquad (16)$$

■

However, in typical pattern recognition applications the moments usually are calculated up to the order (4, 4). The higher order moments are not used, since they are very sensitive to noise. From (16), it becomes clear that for $L = 5$, if an edge of a block contains less than four pixels, it is faster to compute the sum of powers of the variable that corresponds to that edge, directly using (5) instead of (9).

Consider the worst case of an $N \times N$ chessboard image with $N^2/2$ blocks. Since now $M = 1$, according to the criterion of Lemma 1, the computation using (5) requires LN^2 power calculations, $L^2N^2/2$ multiplications, and LN^2 additions. Using (2), $L^2N^2/2$ power calculations, $L^2N^2/2$ multiplications, and $L^2N^2/2$ additions are required. Thus, it is concluded that the IBR for the computation of moments is still computationally attractive in comparison with the use of (2).

V. Examples

Consider the test images of Fig. 3. Table II shows the number of the pixels with object level, the number of the rows with object pixels, the number of the blocks extracted from these images using the Algorithm 1, the required storage for the 2-D images, and the required storage for the block represented images are shown. It can been seen that the storage of the blocks requires less space in comparison with the required storage for the 2-D images.

In images with a high entropy value, like images of text where a significant number of small blocks appears, the required time for the computation of the moments is reduced, using image block representation by a factor between 10 and 50. In images with large areas of object level, like images of industrial parts, aircrafts, ships, etc., the factor of time reduction is much greater. The computation time of moments up to the order (4, 4), for the set of the test images of Fig. 3, using the four different methods described earlier, is given in this section. These methods are

TABLE II
NUMBER OF PIXELS WITH OBJECT LEVEL, NUMBER OF ROWS WITH OBJECT PIXELS, NUMBER OF BLOCKS, REQUIRED STORAGE FOR 2-D IMAGES, AND REQUIRED STORAGE FOR BLOCK REPRESENTED IMAGES FOR THE SET OF TEST IMAGES OF FIG. 3

Image	Pixels with object level	Rows with object pixels	Number of blocks	Storage for the 2-D image	Storage for blocks
Image of the island Corfu	41605	411	250	32768 bytes	2000 bytes
Image of the island Mikonos	47368	249	232	32768 bytes	1856 bytes
Image of the island Santorini	63203	474	257	32768 bytes	2056 bytes
Image of the aircraft	118831	494	397	44608 bytes	3176 bytes

TABLE III
COMPUTATION TIME OF THE GEOMETRICAL MOMENTS UP TO ORDER (4, 4) OF THE TEST IMAGES OF FIG. 3 USING DIFFERENT METHODS

Computation of the geometrical moments	image of the island Corfu		image of the island Mikonos		image of the island Santorini		image of the aircraft	
	time (sec)	reduction factor	time (sec)	reduction factor	time (sec)	reduction factor	time (sec)	reduction factor
from equation (2)	7.850		8.840		11.210		20.150	
from equation (5)	0.380	20	0.330	27	0.490	23	1.430	14
from equation (9)	0.060	131	0.060	147	0.065	172	0.060	336
using the criterion of Lemma 1	0.020	392	0.015	589	0.025	448	0.015	1343

1) the regular computation from (2);
2) the IBR and the use of (5);
3) the IBR with the analytical formula (9);
4) the IBR with the criterion provided by Lemma 1.

The geometrical moments of the test images of Fig. 3 have been computed up to the order (4, 4) and the results are summarized in Table III. It is shown that the use of IBR and (5) results to a reduction of the computation time by a factor of ≈20. The use of the analytical formula (9) decreases the computation time by a factor of ≈200. Using the criterion provided by Lemma 1, the computation time is decreased by a factor of hundreds or thousands, since in most of the extracted blocks one edge has width less than four points.

VI. CONCLUSIONS

In this work, the image block representation idea and the associated algorithm are presented. Owing to the nature of the digital image, only rectangular areas with the same level are present. The IBR uses these rectangular similarities and offers advantages in image handling and computational cost. The IBR provides also a perception about rectangular image regions larger than a pixel. Two-dimensional moments is a classical image analysis tool, and the use of block represented binary images dramatically decreases the computation effort. The complexity of the algorithm for the computation of moments in block represented images, is independent of the image size. Using the IBR scheme for the computation of moments a rate of 35–50 frames/s with 512 × 512 images is achieved. The real-time moments computation in block represented binary images is useful in motion detection, moving object recognition, target identification, and tracking and robot vision applications. Other image processing and analysis tasks can be also performed on block represented images, but this is a topic for future research.

The extension of the proposed method to gray-level images is straightforward. Each block is represented by five integers: the coordinates of the upper left and lower right corners and its gray-level value. For the moments computation, it is adequate to calculate the moments of the corresponding binary block and to multiply them by the gray-level value of the block, since all the pixels of the block have the same gray-level value.

REFERENCES

[1] H. Samet, “The quadtree and related hierarchical data structures,” *Comput. Surv.*, vol. 16, pp. 187–260, 1984.

[2] H. Freeman, “Computer processing of line drawings,” *ACM Comput. Surv.*, vol. 6, pp. 57–97, 1974.

[3] D. W. Paglieroni and A. K. Jain, “Control point transforms for shape representation and measurement,” *Comput. Vis., Graphics, Image Process.*, vol. 42, pp. 87–111, 1988.

[4] R. L. Kashyap and R. Chellappa, “Stochastic models for closed boundary analysis: Representation and reconstruction,” *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 627–637, Sept. 1981.

[5] J. Piper, “Efficient implementation of skeletonization using interval coding,” *Pattern Recognit. Lett.*, vol. 3, pp. 389–397, 1985.

[6] B. G. Mertzios, “Block realization of 2-D IIR digital filters,” *Signal Process.*, vol. 7, pp. 135–149, Oct. 1984.

[7] ——, “Fast block implementation of two-dimensional recursive digital filters via VLSI array processors,” *Archiv für Elektron. Übertragungstech. (AEU)*, vol. 43, pp. 55–58, 1990.

[8] X. Liu and A. Fettweis, “Multidimensional digital filtering by using parallel algorithms based on diagonal processing,” *Multidimen. Syst. Signal Process.*, vol. 1, pp. 51–66, 1990.

[9] I. M. Spiliotis and B. G. Mertzios, “Real-time computation of statistical moments on binary images using block representation,” in *Proc. 4th Int. Workshop on Time-Varying Image Processing and Moving Object Recognition*, Florence, Italy, June 10–11, 1993, pp. 27–34.

[10] I. M. Spiliotis, D. A. Mitzias, and B. G. Mertzios, “A skeleton-based hierarchical system for learning and recognition,” in *Proc. Int. Symp. Mathematical Theory of Networks and Systems*, Regensburg, Germany, Aug. 2–6, 1993, pp. 873–878.

[11] S. A. Dudani, K. J. Breeding, and R. B. McGhee, “Aircraft identification by moment invariants,” *IEEE Trans. Comput.*, vol. C-26, pp. 39–45, Jan. 1977.

[12] G. L. Cash and M. Hatamian, “Optical character recognition by the method of moments,” *Comput. Vis., Graph., Image Process.*, vol. 39, pp. 291–310, 1987.

[13] K. Tsirikolias and B. G. Mertzios, “Statistical pattern recognition using efficient 2-D moments with applications to character recognition,” *Pattern Recognit.*, vol. 26, pp. 877–882, 1993.

[14] F. A. Sadjaji and E. L. Hall, “Three-dimensional moments invariants,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-2, pp. 127–136, 1980.

[15] A. P. Reeves, M. L. Aleey, and O. R. Mitchell, “A moment based two-dimensional edge operator,” in *Proc. IEEE Conf. Computer Vision Pattern Recognition*, 1983, pp. 114–120.

[16] A. Khotanzad and Y. H. Hong, “Invariant image recognition by Zernike moments,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 489–497, May 1990.

[17] M.-K. Hu, “Pattern recognition by moments invariants,” in *Proc. IRE,* Sept. 1961, vol. 49.
[18] ——, “Visual pattern recognition by moment invariants,” *IRE Trans. Inform. Theory,* vol. IT-8, pp. 179–187, Feb. 1962.
[19] M. R. Teague, “Image analysis via the general theory of moments,” *J. Opt. Soc. Amer.,* vol. 70, pp. 920–930, Aug. 1980.
[20] Y. S. Abu-Mostafa and D. Psaltis, “Image normalization by complex moments,” *IEEE Trans. Pattern Anal. Machine Intell.,* vol. PAMI-7, pp. 46–55, Jan. 1985.
[21] C.-H. Teh and R. T. Chin, “On image analysis by the methods of moments,” *IEEE Trans. Pattern Anal. Machine Intell.,* vol. 10, pp. 496–513, 1988.
[22] A. Papoulis, *Propability, Random Variables, and Stochastic Processes.* New York: McGraw-Hill, 1965.
[23] G. Y. Tang, “A discrete version of Green’s theorem,” *IEEE Trans. Pattern Anal. Machine Intell.,* vol. PAMI-4, May 1982.
[24] X. Y. Jiang and H. Bunke, “Simple and fast computation of moments,” *Pattern Recognit.,* vol. 24, pp. 801–806, 1991.
[25] B. C. Li and J. Shen, “Fast computation of moments invariants,” *Pattern Recognit.,* vol. 24, pp. 807–813, 1991.
[26] M. Hatamian, “A real-time two-dimensional moment generating algorithm and its single chip implementation,” *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-34, pp. 546–553, June 1986.
[27] H. S. Baird, S. E. Jones, and S. J. Fortune, “Image segmentation by shape-directed covers,” in *Proc. 10th Int. Conf. Pattern Recognition,* Atlantic City, NJ, June 16–21, 1990, pp. 820–825.
[28] M. McKenna, J. O’Rourje, and S. Suri, “Finding the largest empty rectangle in an orthogonal polygon,” in *Proc. 23rd Ann. Allerton Conf. on Communication, Control, and Computing,* Urbana-Champaign, IL, Oct. 1985, pp. 486–495.
[29] M. Orlowski, “A new algorithm for the largest empty rectangle problem,” *Algorithmica,* vol. 5, pp. 65–73, 1990.

Texture Synthesis-by-Analysis with Hard-Limited Gaussian Processes

Giovanni Jacovitti, Alessandro Neri, and Gaetano Scarano

***Abstract*— A twin stage texture synthesis-by-analysis method is presented. It aims to approximate first- and second-order distributions of the texture, accordingly to the Julesz conjecture. In the first stage, the binary textural behavior of a given prototype is represented by means of a hard-limited Gaussian process. In the second stage, the texture is synthesized by passing the binary hard-limited Gaussian process through a linear filter followed by a zero memory histogram equalizer.**

***Index Terms*—Bussgang deconvolution, hard limiter, image synthesis.**

I. Introduction

Although there is not a unique notion of textures, they are loosely thought of as surfaces characterized by more or less regular aggregates of similar patterns. Among different approaches (morphological, syntactic, fractal, mosaic, etc.), textures are often modeled as realizations of stationary random fields (RF’s).

Generally, in analysis and automatic classification problems, the success of techniques based on RF models relies on the actual detectability of some distinctive features in the decision space, and largely depends upon the specific application. For instance, fine discrimination can be achieved by decomposing the image into multiple narrow spatial frequency and orientation channels [1], [2].

In human user oriented applications, such as assisted browsing of image data bases, selection criteria must comply with visual perception requirements. This implies that textures have to be modeled in terms of visual features, defined on the basis of the available knowledge about the vision phenomena. The same problem is present in synthesis problems, where the goal is generation of textures possessing some characteristics in terms of “perceptive closeness” to a given prototype. However, artificial reproduction of actual textures is a very challenging task, due to the unlimited variety of combinations of possible surfaces and illumination conditions, and to the strong capacity of the human visual system to discriminate even slightly different patterns. From a pragmatic viewpoint, we consider artificial textures to be visually similar to their natural counterparts if they can not be distinguished in a “preattentive” stage of vision (before detailed examination). In this respect, experiments conducted with stochastic models have shown that the human visual system preattentively distinguishes textures with different first- and second-order probability distributions, but is generally less sensitive to differences in higher order distributions (Julesz conjecture [3]).

As a matter of fact, many approaches to texture modeling aim to copy the histogram and the empirical autocorrelation function (acf) of a given sample (see, for instance, [4], [5]). In particular, recent techniques consist of decomposing a texture as a convolution of a non-Gaussian i.i.d. excitation and a filter, identified by means of higher order statistical analysis [6]–[8]. In order to deal more directly with some fundamental characteristics of textures, namely “repetitiveness,” “directionality,” and “granularity and complexity” [19]–[21], the texture is preliminary decomposed into its “deterministic” (periodic and directional) and indeterministic (unpredictable) components, according to a two-dimensional (2-D) extension of Wold decomposition paradigm.

These techniques are often satisfactory for classification purposes, and some visually good examples of synthesis have been also presented in literature. However, they are limited by the assumption of linear stochastic interaction between pixels. In fact, in [8] a deterministic *ad hoc* excitation has to be added to the independent, identically distributed (i.i.d.) input random field. In other approaches [9], [22], this problem is circumvent by using more flexible non-Gaussian Markov RF’s (MRF’s). The main problems associated with the latter techniques are related to the difficulty of determining the MRF parameters (choice of the neighborhood set and identification of probabilities or potentials) with simple methods.

In this contribution, we propose a new automatic texture synthesis-by-analysis technique based on a model constituted by a cascade of linear filters and zero-memory nonlinearities, excited by a white Gaussian source. A preliminary description of this technique has been presented in [10].

In essence, the method replaces usual i.i.d. sources employed in linear texture models with non-i.i.d. binary sources. This choice is motivated by the simplicity of controlling the second-order distribution of binary random fields. In fact, under weak symmetry constraints, the second-order distribution of a binary field is completely described by its acf. Now, regarding a binary random field

Manuscript received June 5, 1996; revised March 6, 1998. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. John Goutsias.

G. Jacovitti and G. Scarano are with the Dipartimento INFOCOM, Universitá di Roma “La Sapienza,” I-00184 Rome, Italy (e-mail: gjacov.@infocom.ing.uniroma1.it; gaetano@infocom.ing.uniroma1.it).

A. Neri is with the Dipartimento di Elettronica, Università degli Studi di Roma Tre, I-00146, Rome, Italy (e-mail: neri@ele.uniroma3.it).

Publisher Item Identifier S 1057-7149(98)07751-3.