where f_i is associated with object α and $f_{i'}$ is associated with object α'. The above definitions are an extension of (4) for dealing with multiple objects: In (4), features due to other objects (not belonging to object α) are all labeled as NULL; (8) simply takes this into consideration.

The likelihood energy is

$$U(d|f, \mathcal{O}^{(\mathrm{all})}) = \sum_{i \in \mathcal{S},\, f_i \neq 0} V_1(d_1(i)|f_i, \mathcal{O}^{(\mathrm{all})}) + \sum_{i \in \mathcal{S},\, f_i \neq 0} \sum_{i' \in \mathcal{S}_{\setminus i},\, f_{i'} \neq 0} \cdot V_2(d_2(i, i')|f_i, f_{i'}, \mathcal{O}^{(\mathrm{all})}). \quad (9)$$

The single-site likelihood potentials are $V_1(d_1(i)|f_i, \mathcal{O}^{(\mathrm{all})}) = V_1(d_1(i)|f_i^{(\alpha)})$ defined in the same way as for (5). The pair-site likelihood potentials are $V_2(d_2(i, i')|f_i, f_{i'}, \mathcal{O}^{(\mathrm{all})}) = V_2(d_2(i, i')|f_i^{(\alpha)}, f_{i'}^{(\alpha')}, \mathcal{O}^{(\alpha)}, \mathcal{O}^{(\alpha')})$ where

$$V_2(d_2(i, i')|f_i^{(\alpha)}, f_{i'}^{(\alpha')}, \mathcal{O}^{(\alpha)}, \mathcal{O}^{(\alpha')}) = \begin{cases} V_2(d_2(i, i')|f_i^{(\alpha)}, f_{i'}^{(\alpha)}), & \text{if } \alpha = \alpha' \\ 0, & \text{otherwise} \end{cases} \quad (10)$$

where $V_2(d_2(i, i')|f_i^{(\alpha)}, f_{i'}^{(\alpha)})$ are defined in the same way as for (5) when for matching to object α.

Some parameters have to be determined, such as v_{10} and v_{20} in the MRF prior, in order to define the MAP solutions completely. They may be estimated by using a supervised learning algorithm [3].

The optimization in MAP matching and recognition is combinatorial. While an optimum is sought in a global sense, many optimization algorithms are based on local optimization. The Hummel–Zucker relaxation labeling algorithm [9] is preferable in terms of the minimized energy value and computational costs [3] and is used in our implementation. It converges after dozens of iterations. The computational time is dominated by relaxation labeling in the first stage and is roughly the complexity of the whole system.

IV. Summary

This work has presented an MAP formulation for the recognition of multiple overlapping objects. In contrast to the past work, where matching between a scene containing multiple objects is performed by considering one model object at a time, the MAP solution here is formulated with respect to all the model objects and is therefore overall optimal. Moreover, it is overall consistent by itself, with no need for *ad hoc* post-processing. The computational cost may be lowered by using a two-stage MAP estimation approach.

References

[1] R. Chellappa and A. Jain, Eds., *Markov Random Fields: Theory and Applications*. New York: Academic, 1993.

[2] K. V. Mardia and G. K. Kanji, Eds., *Statistics and Images: 1 and 2*: Carfax, 1993–1994.

[3] S. Z. Li, Ed., *Markov Random Field Modeling in Computer Vision*. New York: Springer-Verlag, 1995.

[4] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 721–741, Nov. 1984.

[5] J. W. Modestino and J. Zhang, "A Markov random field model-based approach to image interpretation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, pp. 606–615, June 1992.

[6] P. R. Cooper, "Parallel structure recognition with uncertainty: Coupled segmentation and matching," in *Proc. IEEE Int. Conf. Computer Vision*, 1990, pp. 287–290.

[7] S. Z. Li, "Bayesian object matching," *J. Appl. Stat.*, vol. 25, no. 3, pp. 425–443, 1998.

[8] ——, "Invariant representation, recognition and pose estimation of 3-D space curved under similarity transformations," *Pattern Recognit.*, vol. 30, no. 3, pp. 447–458, 1997.

[9] R. A. Hummel and S. W. Zucker, "On the foundations of relaxation labeling process," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-5, pp. 267–286, May 1983.

A Two-Subcycle Thinning Algorithm and Its Parallel Implementation on SIMD Machines

Alfredo Petrosino and Giuseppe Salvi

***Abstract*—A new parallel thinning algorithm with two subcycles is proposed and compared with other parallel thinning algorithms in terms of 8-connectedness degree, erosion, stability under pattern rotation, and boundary noise sensitivity. Computational issues are also reported based on the implementation of the thinning algorithm on the SIMD machines CM-200 and MasPar MPP-12000.**

***Index Terms*—Parallel implementation, parallel thinning, SIMD machines, skeletonization, thinning accuracy.**

I. Introduction

Parallel thinning is becoming a valid and fundamental approach to thinning and skeletonization, due to the ever increasing growth of availability of parallel architectures and the real possibility of realizing machine vision systems. Recently, its interest has also grown both in design and objective standpoint; this has led to parallel algorithms based on artificial neural networks [9] and three–dimensional (3–D) parallel thinning able to produce a medial surface representation [12]. Commonly, the parallel thinning algorithms are characterized by the repetition of a simultaneous deletion process of all the "deletable" contour points; to be efficient, the number of repetitions or iterations should be ideally equal to half of the maximum width of the figures present in a picture. Of course, the less time-units and iterations the algorithm spends the more it is efficient under the constraint of remaining the 8-curve as perfect as possible and preventing excessive erosion. To deal with the latter requirements, each iteration of a parallel thinning algorithm is subdivided in n subcycles $(n = 1, 2, 3, \ldots)$. Indeed, as Stefanelli and Rosenfeld indicated [14], parallel thinning algorithms with a 1-subcycle/iteration have the disadvantage that they may yield nonconnected or even empty medial lines for connected figures; the case where a two-pixels-wide straight stroke is entirely eliminated by a single iteration is widely known. Based on these considerations and computational requirements, we propose a thinning algorithm with 2-subcycles/iteration which is able to

1) reduce the number of iterations and the time complexity of each iteration;
2) produce a perfect 8-connected thin line;
3) prevent excessive erosion.

Manuscript received April 6, 1998; revised February 16, 1999. This work was supported by Consiglio Nazionale delle Ricerche, Progetto Finalizzato "Robotica" and the Istituto Nazionale Fisica della Materia (INFM), Research Unity of University of Salerno. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Dmitry B. Goldgof.

A. Petrosino is with INFM, University of Salerno, 84081 Baronissi, Salerno, Italy (e-mail: alfredo@synapse.irsip.na.cnr.it).

G. Salvi is with IRSIP, National Research Council (CNR), 80131 Napoli, Italy.

Publisher Item Identifier S 1057-7149(00)01160-X.

The meaning of these objectives will be clarified in the next sections. According to the algorithm, described in Section III, a pixel is deletable by analyzing the values of its neighboring pixels within the areas (3×4) and (4×3) of pixels in the first subcycle and an area (3×3) in the second. The algorithm is characterized by the fact that the first subcycle is used to remove boundary elements of an object in all directions, while the second subcycle is used to remove corners and diagonal boundary elements in the NorthEast, NorthWest, SouthWest, and SouthEast directions. This isotropic behavior allows the generation of more regular skeletons and better qualitative results. Experimental results in terms of degree of 8-connectedness, degree of erosion, stability under rotation, boundary noise sensitivity, and timings on SIMD machines are reported in Section IV.

II. Basic Notations

In the discussion below, it is understood that a pixel p examined for deletion is a black pixel, and the pixels in its (3×3) neighborhood are labeled as $x_i, i=1,\ldots,8$, as shown in Fig. 1.

Definition 1: The pixels $x_1, x_2, \ldots, x_8$ are the 8-neighbors of p and are collectively denoted by $N(p)$. They are said to be 8-adjacent to p. The number of black pixels in $N(p)$ is denoted by $b(p)$, i.e., $b(p)=\sum_{i\in N(p)} x_i$.

Definition 2: The pixels x_1, x_3, x_5, x_7 are the 4-neighbors of p and are said 4-adjacent to p.

Definition 3: A sequence of points $z_1, z_2, \ldots, z_n$ is called an 8-(or 4-) path if z_{i+1} is an 8-(or 4-) neighbor of z_i, for $i=1,2,\ldots,n-1$.

Definition 4: A subset C of a picture P is 8-(or 4-) connected if for every pair of points (x,y) in C there exists an 8-(or 4-) path from x to y consisting of points in C.

Definition 5: The number of transitions from a white point to a black point and vice-versa (when the points in $N(p)$ are crossed, for example, in a counterclockwise order) is called the *Rutowitz number* and is defined as $X_R(p)=\sum_{i=1}^{8}|x_{i+1}-x_i|$, where $x_9=x_1$.

Based on the previous common definitions, a lot of algorithms were stated. In the following sections we shall refer to all 2-subcycle parallel algorithms reported in the review paper of Lam *et al.* [10], i.e., the algorithms in [2], [4], [5], [14], [15], [20], with the addition of the 2-subcycle algorithm of Wang and Zhang [19] and the 1-subcycle algorithm of Chin *et al.* [1]. All these algorithms are widely considered to be a well assested set for benchmarking parallel thinning [11].

III. The Two-Subcycle Thinning Algorithm

A parallel thinning algorithm with two subcycles usually requires two sets of pixels deletable in parallel whose union contains all the deletable border pixels, where "deletability in parallel" means that all black pixels p in the set can be simultaneously deleted without causing excessive erosion and retaining the topology of any binary pattern. We recall that *excessive erosion* commonly means that a border point is deleted and its deletion causes the loss of the object's shape. Some definitions are to be joined to the previously reported as they pertain to our algorithm.

Definition 6: A black point p that has at least one white 4-neighbor is called an *edge-point* (i.e., it lies along the edge of the pattern).

Definition 7: A black point p that has at most one black 8-neighbor is called an *end-point*.

Definition 8: A black point p, the deletion of which would break the connectedness of the original pattern, is called a *break-point*.

In essence, the proposed thinning algorithm consists of executing several iterations over the pattern, where many black points are deleted in each iteration. If no point is deleted at the end of an iteration, the thinning procedure stops. In any subcycle, the set of black pixels p to be deleted from the pattern must satisfy the following intuitive criterion.

x_4	x_3	x_2	y_4
x_5	p	x_1	y_5
x_6	x_7	x_8	y_6
y_1	y_2	y_3	

Fig. 1. Notation for the labels of the pixels in the surrounding neighborhood.

A set S of black pixels p is deletable in parallel if every point p in this set is an edge-point, but not an end-point, or a break-point. Also, its possible deletion must not cause excessive erosion in the pattern.

To hold the above criterion, some lemmas should be stated. They regard the deletability of two sets of pixels, I_1 for the first subcycle and I_2 for the second.

Lemma 1: The set I_1 of black pixels p satisfying the following conditions:

i) $X_R(p)=2$; *ii*) $2\le b(p)\le 6$; *iii*) $R_0=0$ where $R_0=x_1x_7x_8((\overline{y_5}x_2x_3\overline{x_5})\vee(\overline{y_2}\,\overline{x_3}x_5x_6))$, are deletable in parallel from any binary picture.

Proof:: According to the condition *ii*), there exist from two up to six black pixels in the neighborhood of the pixel p candidate to be deleted. On the contrary, the condition *i*) means that there are only two 0–1 (or equivalently 1–0) transitions in the same neighborhood; this corresponds to having two $1s$ among $x_1,\ldots,x_8$, while the remaining six pixels can be all 0 or four of them are 1. Based on these considerations, the set $N(p)$ contains between two and six black pixels which are 4-connected among themselves, i.e., all the possible pairs of these pixels are connected by 4-paths; thus, p is not a *break-point*. Moreover, $b(p)\le 6$ and $b(p)\ge 2$ ensure, respectively, that p has at least a white 4-neighbor and that p is not an isolated or *end-point*. Since the entire set of black pixels p that satisfy the conditions *i*) and *ii*) is deletable in parallel, objects that are two pixels wide completely disappear. This difficulty cannot be solved using the usual (3 × 3) window operator. To overcome this problem, (3 × 4) and (4 × 3) restoring templates are used [see templates depicted in Figs. 2(a1) and (a2)]. The $\vee$ of the depicted templates, representing the boolean expression R_0, restore two pixels wide vertical and horizontal lines. That is, for any pattern such that R_0 is true, its central pixel is saved or, equivalently, all black pixels p that satisfy the condition of deletion *iii*) can be deleted in parallel. □

Let us now illustrate the second subcycle. In this case, as before mentioned, we deal with the direct thinning of the corner points and diagonal lines present in a digital pattern.

Lemma 2: The set I_2 of black pixels p satisfying the following conditions:

iv) $\overline{S_0}\wedge S_1\wedge(b(p)\ge 2)$; *v*) $(R_1=0)\wedge(b(p)\ge 3)$ where $S_0=x_3x_7\vee x_5x_1$, $S_1=x_1\overline{x_6}(\overline{x_4}\vee x_3)\vee x_3\overline{x_8}(\overline{x_6}\vee x_5)\vee x_7\overline{x_4}(\overline{x_2}\vee x_1)\vee x_5\overline{x_2}(\overline{x_8}\vee x_7)$ and $R_1=\overline{x_3}(x_1\overline{x_8}\vee x_5\overline{x_6})\vee x_7(\overline{x_5x_8}\vee\overline{x_1x_6})$, are deletable in parallel from any binary picture.

Proof:: S_0 is the $\vee$ of the templates shown in Fig. 2(b1) and (b2) and S_1 is the $\vee$ of the templates shown in Figs. 2(c1)–(c2) and their 90°, 180°, and 360° rotated versions. Since these templates contain between two and five 4-connected black pixels, p has at least a white 4-neighbor and is not a *break-point*. Moreover $b(p)\ge 2$ ensures that p is not an isolated point or *end-point*. Since the entire set of black pixels p satisfying condition *v*) is deletable in parallel, the two pixels wide diagonal lines completely disappear. To face this problem, it is necessary to save pixels which do not satisfy condition *v*). Indeed, the Boolean expression R_1 can be seen as the $\vee$ of the restoring templates depicted as Fig. 2(d1) and (d2) and their reflected versions with respect to the

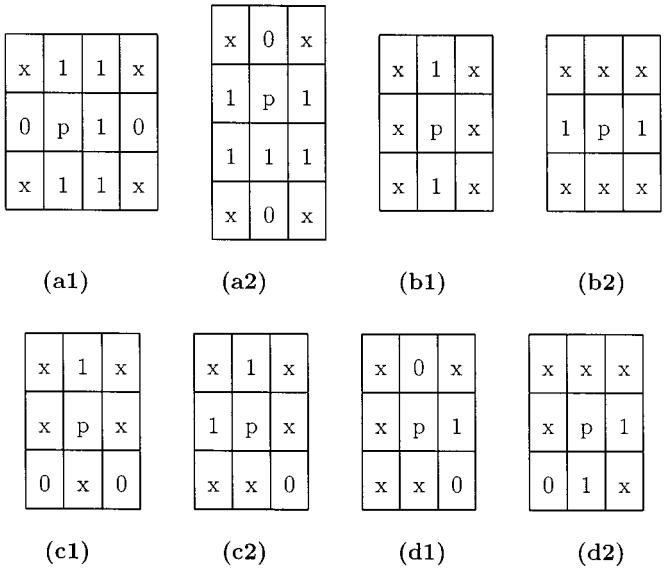

Fig. 2. Templates corresponding to the deletion conditions (x's are *don't care's*).

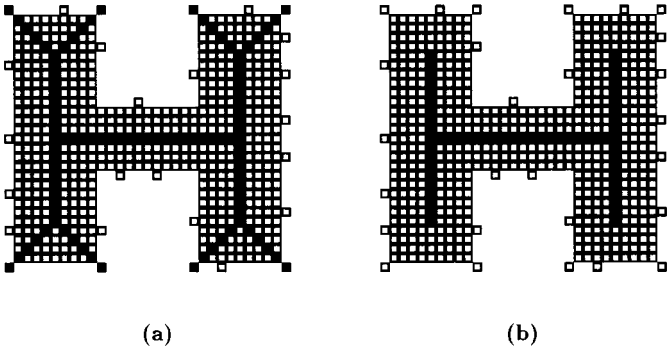

Fig. 3. Skeletons generated by the proposed algorithm (a) without noise removal condition and (b) with noise removal condition.

x	y	x	x	x
y	1	0	0	x
x	0	p	0	x
x	0	0	0	x
x	x	x	x	x

Fig. 4. Pixel configuration corresponding to the noise removal conditions. At least one pixel of those marked y must be 1, while x's are don't care's and p is a black pixel.

vertical axis. In this way all black pixels p that satisfy the conditions *iv*) and *v*) can be deleted in parallel. □

A problem associated with most of the existing algorithms is that they produce a noisy skeleton if the original pattern contains noise on the boundary. The effects of boundary noise on thinning digital patterns have been studied by several researchers [6]. Fundamentally, the growth of spurious segments inducted by noisy points on the boundary can be avoided by

1) smoothing the boundary of the original object before applying the thinning process;
2) thinning gray-scale images obtained by filtering the original binary shape [3];
3) applying a set of noise cleaning conditions to the binary image at each iteration [1], [13].

In the first two cases, the preprocessing step excessively slows down the thinning process, resulting inadequate in real-time applications. Instead, the use of cleaning conditions, while increasing the total implementation cost, does not affect the speed of the operation excessively, since they can be applied to the image in parallel with thinning conditions. Our approach follows this last method by applying a cleaning condition which avoids spurious skeleton legs [see the example reported in Fig. 3(a)]. The idea underlying the approach we propose is to delete *noisy boundary points*, defined as end-points whose only one black 8-neighbor is 4-adjacent to an edge-point. In order to evaluate this condition the 3 × 3 neighborhood is not sufficient; therefore, some neighbors in a 5 × 5 window are necessarely checked. This is consistent with existing literature [6], [13], where the description of the noise is performed by using 5 × 5 windows. In practice, the cleaning condition corresponds to the OR of the pixel configuration depicted in the template of Fig. 4 and its reflections with respect to the central row, the central column and the secondary diagonal. At least one of the pixels marked y in the configuration must be nonzero, while x's can have any value. In Fig. 3(b), the improvement obtained by applying the noise removal condition to the noisy picture representing the letter "H" is depicted. The next section reports a set of experiments for the quantitative assesment of the performance of the proposed algorithm under noise condition.

IV. Results and Comparisons with Other Thinning Algorithms

In this section, the results obtained by the proposed algorithm and algorithms due to Deutsch [4], Zhang and Suen [20], Chen and Hsu [2], Wang and Zhang [19], Suzuki and Abe [15], Guo and Hall [5], Stefanelli and Rosenfeld [14], and Chin *et al.* [1] are compared by using as comparative terms

1) degree of 8-connectedness;
2) degree of erosion;
3) stability under pattern rotation;
4) boundary noise sensitivity.

Comparisons concerning the time-units spent by the algorithm to converge are also reported against those spent by the parallel algorithms of Zhang and Suen, Wang and Zhang, Chin *et al.*, and Holt and Stewart [7] which turned out to be comparable in terms of computational complexity to our algorithm.

A. Degree of 8-Connectedness and Erosion

First, three artificial patterns of dimension 64 × 64 representing the letter "A" (as used in Tamura's study [16]), the letter "X" and the Chinese "hideogram" shown in Figs. 5–7 are designed to test the degree of 8-connectedness and erosion. We recall that a thin line, except for its junctions, is *perfectly 8-connected* if there does not exist any 8-deletable point on the line. All the thinning results shown in Figs. 5(i), 6(i), and 7(i) obtained with the proposed algorithm are perfectly 8-connected and do not have an excessive erosion. As already known, Deutsch's and Zhang and Suen's algorithms have the common deficiency of excessive erosion in thinning the diagonal lines. For example, applying the algorithms of Deutsch and Zhang and Suen for thinning the digital patterns "X" and "hideogram" shown in Figs. 6(a)–(b) and 7(a)–(b) turns out to loose the structure of the original pattern. In general, diagonal segments with thickness equal

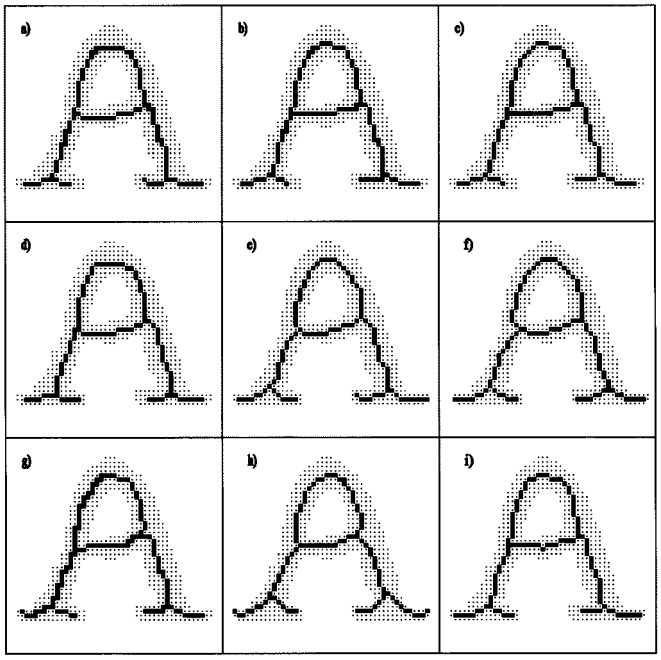

Fig. 5. Thinning results for the letter "A" by using (a) Deutsch's algorithm; (b) Zhang and Suen's algorithm; (c) Chen and Hsu's algorithm; (d) Wang and Zhang's algorithm; (e) Suzuki and Abe's algorithm; (f) Guo and Hall's algorithm; (g) Stefanelli and Rosenfeld's algorithm; (h) Chin *et al.*'s algorithm; and (i) the proposed algorithm.

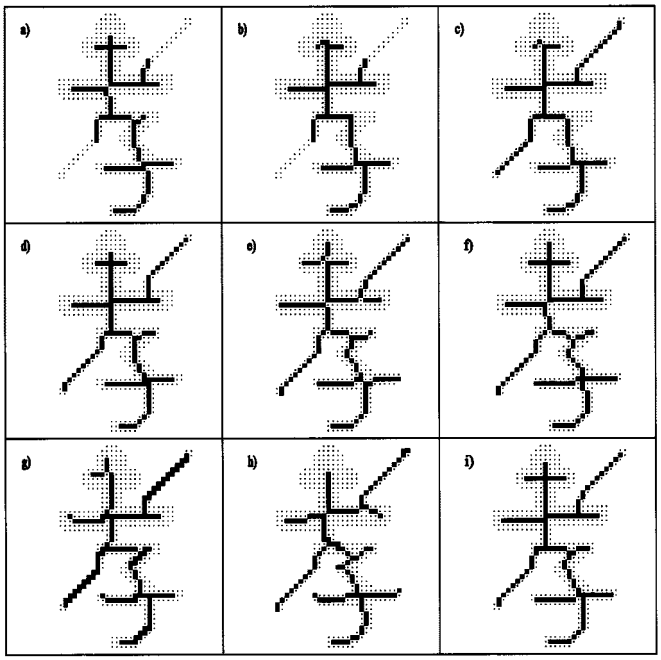

Fig. 7. Thinning results for the "hideogram" by using (a) Deutsch's algorithm; (b) Zhang and Suen's algorithm; (c) Chen and Hsu's algorithm; (d) Wang and Zhang's algorithm; (e) Suzuki and Abe's algorithm; (f) Guo and Hall's algorithm; (g) Stefanelli and Rosenfeld's algorithm; (h) Chin *et al.*'s algorithm; and (i) the proposed algorithm.

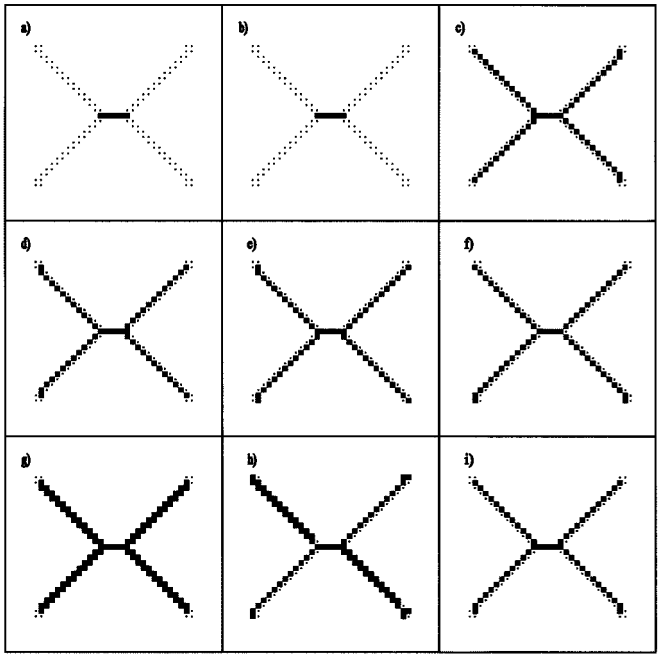

Fig. 6. Thinning results for the letter "X" by using (a) Deutsch's algorithm; (b) Zhang and Suen's algorithm; (c) Chen and Hsu's algorithm; (d) Wang and Zhang's algorithm; (e) Suzuki and Abe's algorithm; (f) Guo and Hall's algorithm; (g) Stefanelli and Rosenfeld's algorithm; (h) Chin *et al.*'s algorithm; and (i) the proposed algorithm.

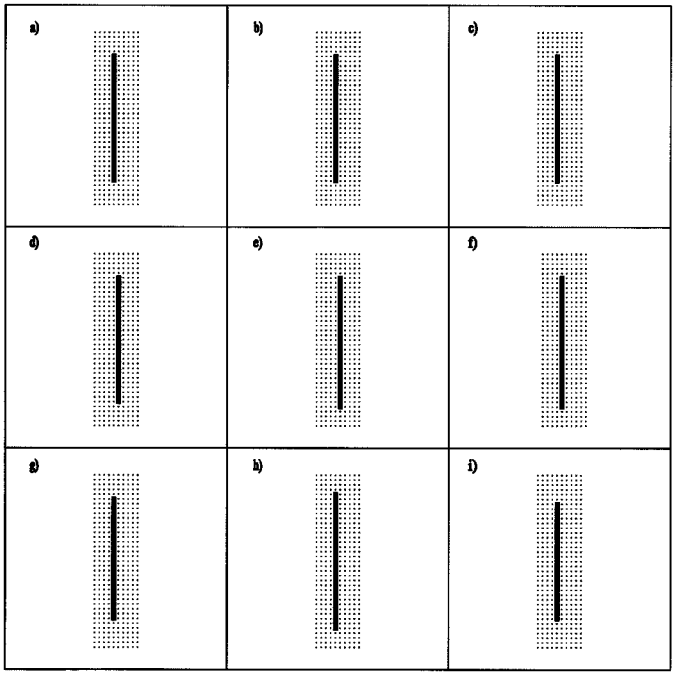

Fig. 8. Skeletons of an elongated shape obtained applying (a) Deutsch's algorithm; (b) Zhang and Suen's algorithm; (c) Chen and Hsu's algorithm; (d) Wang and Zhang's algorithm; (e) Suzuki and Abe's algorithm; (f) Guo and Hall's algorithm; (g) Stefanelli and Rosenfeld's algorithm; (h) Chin *et al.*'s algorithm; and (i) the proposed algorithm.

to two eventually reduce to only one or at most two points. The algorithm described in [1] can keep the shape of the original pattern, but sometimes it cannot exactly keep the basic structure as shown in Fig. 7(h) for the Chinese "hideogram" and does not thin the diagonal lines with thickness equal to two as shown in Fig. 6(h) for the letter "X". The same is valid for the Suzuki and Abe's algorithm [see for instance Fig. 7(e)]. The structure is lost too by applying the algorithms of Chen and Hsu and Stefanelli and Rosenfeld, as can be seen in Fig. 7(c) and (g). In addition, the algorithms of Zhang and Suen, Wang and Zhang, Stefanelli and Rosenfeld, and Chin *et al.* are imperfectly 8-connected (i.e., there exists at least a pair of points in the skeleton

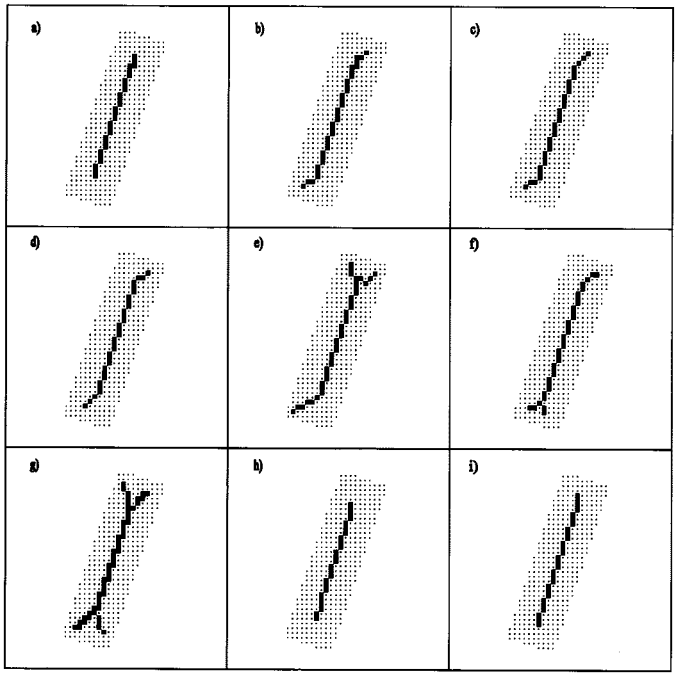

Fig. 9. Skeletons of the elongated shape of Fig. 8 rotated through 20°.

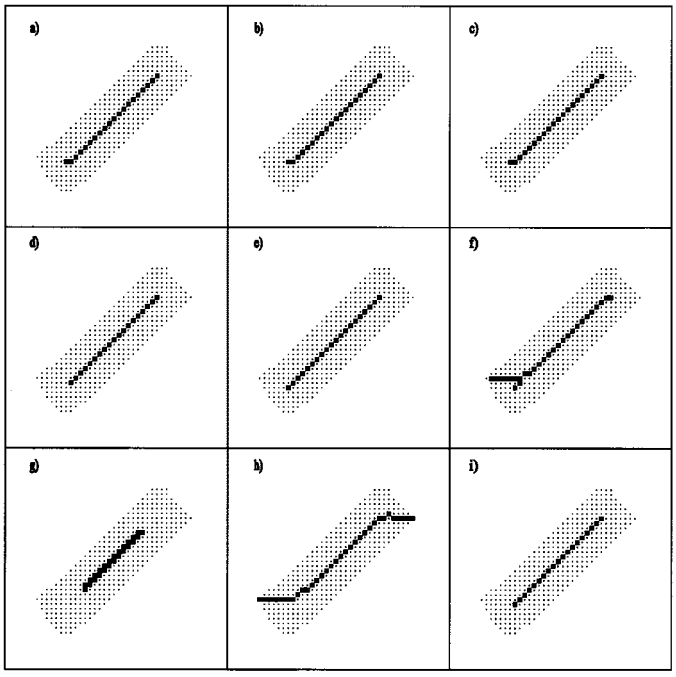

Fig. 10. Skeletons of the elongated shape of Fig. 8 rotated through 60°.

which are 8-deletable) as shown in Fig. 5(b) and (d) for the letter "A", Fig. 6(d), (g), and (h) for the letter "X" and Fig. 7 for the Chinese "hideogram." Furthermore, as already known, the algorithm of Guo and Hall suffers of the Y shape problem in presence of junctions, although it achieves good results in terms of 8-connectedness.

B. Stability Under Pattern Rotation

Figs. 8–13 show an elongated pattern and the pattern "aircraft" in three different orientations and the thinning results obtained by applying all the thinning algorithms. In all cases the proposed algorithm produces the same number of branches, which lie in the medial region

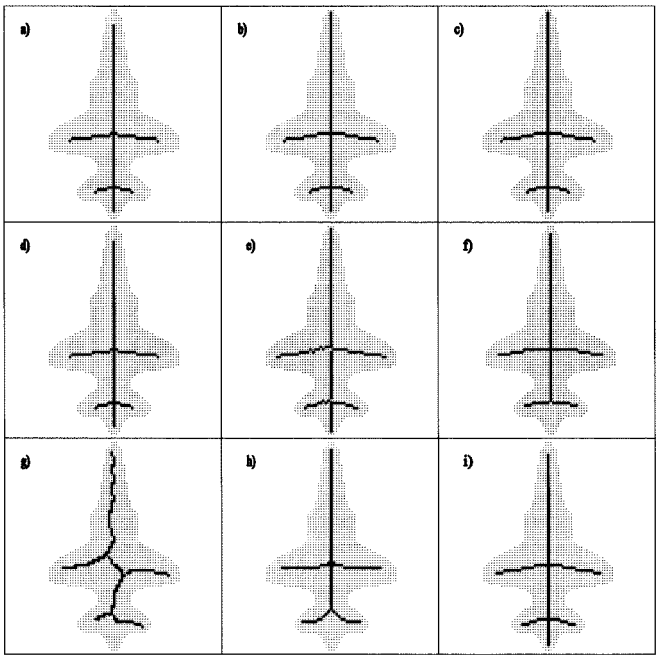

Fig. 11. Skeletons of the shape "aircraft" obtained applying (a) Deutsch's algorithm; (b) Zhang and Suen's algorithm; (c) Chen and Hsu's algorithm; (d) Wang and Zhang's algorithm; (e) Suzuki and Abe's algorithm; (f) Guo and Hall's algorithm; (g) Stefanelli and Rosenfeld's algorithm; (h) Chin *et al.*'s algorithm; and (i) the proposed algorithm.

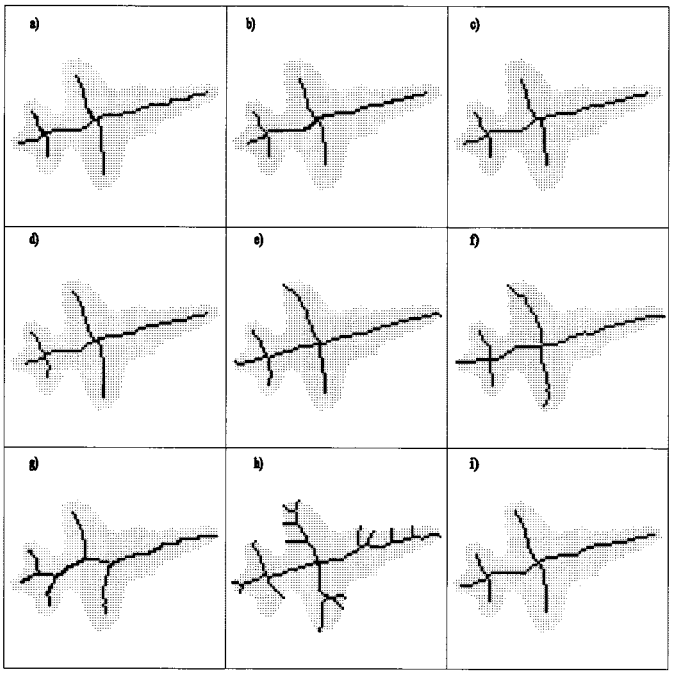

Fig. 12. Skeletons of the shape "aircraft" of Fig. 11 rotated through 80°.

of the pattern. The stability under pattern rotation is due to its noise immunity and to the symmetrical cancellation of each pixel in each direction. Also, the Wang and Zhang algorithm compares well with the proposed thinning technique.

C. Boundary Noise Sensitivity

To evaluate the sensitivity of the algorithm to boundary noise, we use the measure introduced in [8]. Let I be the noise free binary image

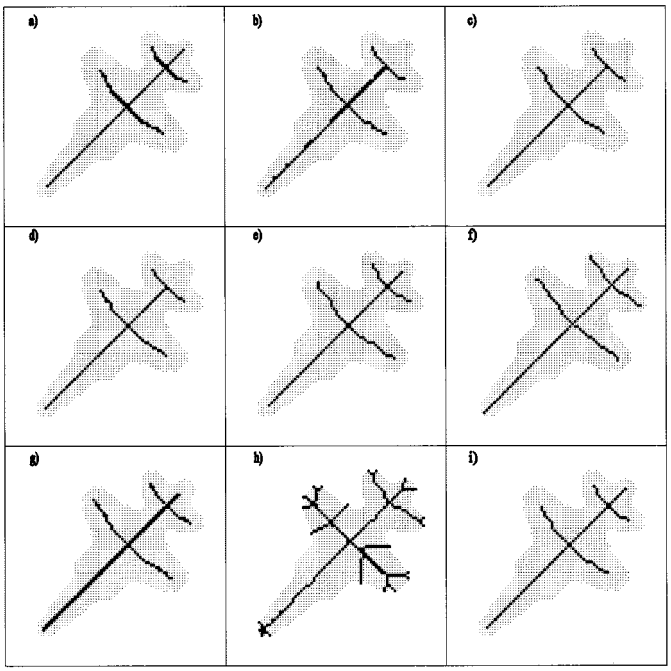

Fig. 13. Skeletons of the shape "aircraft" of Fig. 11 rotated through 230°.

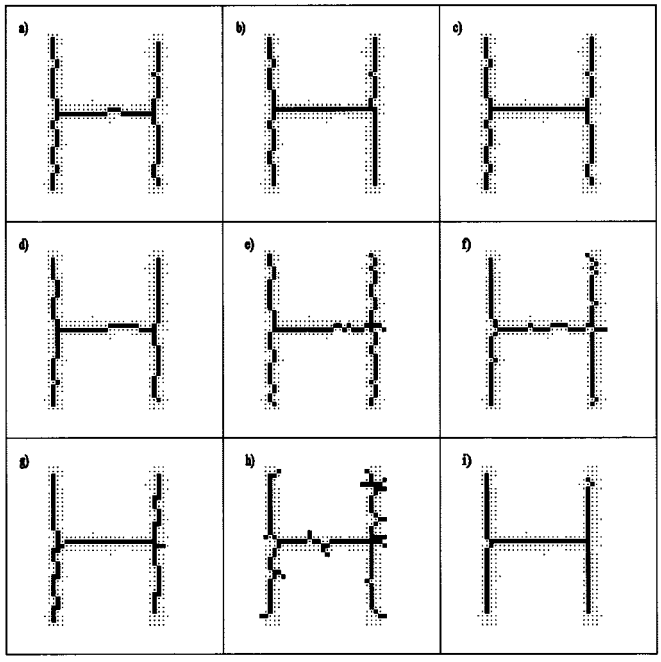

Fig. 14. Skeletons of a noisy shape and the normalized error: (a) Deutsch's algorithm, $m_e = 0.44886$; (b) Zhang and Suen's algorithm, $m_e = 0.55682$; (c) Chen and Hsu's algorithm, $m_e = 0.42045$; (d) Wang and Zhang's algorithm, $m_e = 0.40909$; (e) Suzuki and Abe's algorithm, $m_e = 0.43182$; (f) Guo and Hall's algorithm, $m_e = 0.14943$; (g) Stefanelli and Rosenfeld's algorithm, $m_e = 0.31034$; (h) Chin *et al.*'s algorithm, $m_e = 0.38068$; and (i) the proposed algorithm, $m_e = 0.01807$.

and N its noisy version generated by randomly adding and subtracting points along the boundary of I for a percentage p of boundary lenght. Thus, the error introduced by boundary noise at a particular noise percentage p can be measured by the normalized quantity

$$m_e(p) = \min\left[1, \frac{Area[I_S/N_S] + Area[N_S/I_S]}{Area[N_S]}\right]$$

TABLE I
BOUNDARY NOISE SENSITIVITY INDICATED BY THE AVERAGE m_e

Thinning algorithms	Noise percentage					
	5	10	15	20	25	30
Deutsch	0.0967	0.1740	0.2379	0.2876	0.3265	0.3598
Zhang & Suen	0.0921	0.1786	0.2501	0.3145	0.3757	0.4194
Chen & Hsu	0.0949	0.1730	0.2371	0.2876	0.3284	0.3628
Wang & Zhang	0.0929	0.1756	0.2541	0.3176	0.3765	0.4291
Suzuki & Abe	0.1043	0.2005	0.2897	0.3703	0.4405	0.5048
Guo & Hall	0.0835	0.1625	0.2380	0.3062	0.3708	0.4288
Stefanelli & Rosenfeld	0.0909	0.1706	0.2326	0.2859	0.3351	0.3684
Chin *et al.*	0.1559	0.2860	0.4016	0.4939	0.5780	0.6510
Proposed	0.0538	0.1023	0.1464	0.1879	0.2324	0.2717

TABLE II
EXPERIMENTAL TIMES (IN SECONDS) FOR THINNING ALGORITHMS IMPLEMENTED ON CM-200 WITH 4096 SINGLE BIT PROCESSORS AND MasPar WITH 4096 4-BIT PROCESSORS. WITH AT WE DENOTE THE AVERAGE COMPUTATION TIME

Pattern / Algorithm	"X"	"hideogram"	"A"	"H"	AT
CM-200					
Chin et al. [1]	0.0124160	0.0372455	0.0372391	0.0372440	0.0310362
Wang & Zhang [14]	0.0159606	0.0319251	0.0399074	0.0399084	0.0319254
Zhang & Suen [15]	0.0706206	0.0543255	0.0217367	0.0271687	0.0434629
Holt & Stewart [5]	0.1001026	0.0720738	0.0240341	0.0280393	0.0560625
Proposed	0.0110109	0.0220107	0.0220107	0.0275127	0.0206363
MasPar					
Chin et al. [1]	0.0058016	0.0176674	0.0176130	0.0176341	0.0146702
Wang & Zhang [14]	0.0040618	0.0084262	0.0105058	0.0100022	0.0082450
Zhang & Suen [15]	0.0238888	0.0184789	0.0073381	0.0091970	0.0147257
Holt & Stewart [5]	0.0446822	0.0324010	0.0109563	0.0128272	0.0252167
Proposed	0.0035123	0.0079787	0.0078192	0.0099082	0.0073046

where I_S and N_S are the resulting skeletons of I and N, respectively, while the operation / represents the set difference. The measure is normalized between 0 and 1 such that a highly noise-sensitive algorithm will yield an m_e close to one. The adopted data set consists of 36 patterns, alphabets "A" to "Z", numerals "0" to "9", in the 48-point Helvetica font. For each pattern, a set of 100 noisy images has been generated by adding noise to a certain percentage of boundary pixels. In particular, we have applied noise percentages equal to 5, 10, 15, 20, 25, and 30. All the 21 600 data set patterns were thinned by the nine algorithms and their results compared in terms of m_e. This experiment is summarized in Table I, which lists the average values of m_e against different noise percentages. Both m_e and visual inspection indicate that the proposed algorithm outperforms the others and is therefore least sensitive to boundary noise. A sample of the resulting skeletons from the data set is shown in Fig. 14.

D. Computational Issues

To compare the time performance, the algorithm has been implemented and tested on the SIMD parallel machines Connection Machine CM-200 with 4096 one-bit processing elements (PE's) [17] and MasPar MPP 12 000 with 4 096 four-bit PE's [18], the implementation being as intuitively efficient as possible. The comparisons were made against the best parallel implementations of the Zhang and Suen, Wang and Zhang, Chin *et al.*, and Holt and Stewart [7] which were available to us at the moment we wrote this paper. The thinning algorithms were tested on the 64×64 digital patterns "X," "hideogram," "A," and "H." The times, expressed in seconds, are shown in Table II. The analysis of the tables makes evident that the proposed two-subcycle algorithm and the Wang algorithm outperform the remaining. Nevertheless, the average computation time (AT) of the proposed algorithm is lower than the Wang algorithm (≈60% on CM-200 and ≈85% on MasPar). In particular, we observe in our experiments that the number of iterations required by the proposed algorithm is equal to about the half of the maximum width of the input picture; a characteristic of the proposed parallel algorithm which makes it near optimal.

V. Concluding Remarks

In this paper, we have reported a new parallel thinning algorithm with two subcycles, characterized by templates of dimension 3×4 and 4×3 for the first subcycle, while a 3×3 template is used in the second. The algorithm has been tested on different patterns and the results compared with those obtained by applying other algorithms of analogous nature. We achieve better results according to the degree of 8-connectedness (perfect skeleton), accuracy, degree of erosion, stability under pattern rotation, and boundary noise sensitivity. Timings taken on the CM-200 and MasPar MPP-12 000 also show the time complexity to be very low for the proposed algorithm.

References

[1] R. T. Chin, H.-K. Wan, D. L. Stover, and R. D. Iverson, "A one-pass thinning algorithm and its parallel implementation," *Comput., Vis., Graph., Image Process.*, vol. 40, pp. 30–40, 1987.

[2] Y.-S. Chen and W.-H. Hsu, "A modified fast parallel algorithm for thinning digital patterns," *Pattern Recognit. Lett.*, vol. 7, no. 2, pp. 99–106, 1988.

[3] Y.-S. Chen and Y.-T. Yu, "Thinning approach for noisy digital patterns," *Pattern Recognit.*, vol. 29, no. 11, pp. 1847–1862, 1996.

[4] E. S. Deutsch, "Thinning algorithms on rectangular, hexagonal and triangular arrays," *Commun. ACM*, vol. 15, pp. 827–837, 1972.

[5] Z. Guo and R. W. Hall, "Parallel thinning with two-subiteration algorithms," *J. ACM*, vol. 32, no. 3, pp. 359–373, 1989.

[6] W. L. M. Hoeks, "Performance evaluation for thinning algorithms with respect to boundary noise," in *Proc. Int. Conf. Image Processing Applications*, Maastricht, NE, 1992, pp. 250–253.

[7] C. Holt and A. Stewart, "A parallel thinning algorithm with fine grain subtasking," *Parallel Comput.*, vol. 10, pp. 329–334, 1989.

[8] B. K. Jang and R. T. Chin, "One-pass parallel thinning: Analysis, properties, and qualitative evaluation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, no. 11, 1992.

[9] R. Krishnapuram and L.-F. Chen, "Implementation of parallel thinning algorithms using recurrent neural networks," *IEEE Trans. Neural Networks*, vol. 4, no. 1, 1993.

[10] L. Lam, S.-W. Lee, and C. Y. Suen, "Thinning methodologies—A comprehensive survey," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, pp. 869–885, Sept. 1992.

[11] L. Lam and C. Y. Suen, "An evaluation of parallel thinning algorithms for character recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, Sept. 1995.

[12] P. K. Saha, B. B. Chaudhuri, and D. D. Majumder, "A new shape preserving parallel thinning algorithm for 3D digital images," *Pattern Recognit.*, vol. 30, no. 12, pp. 1939–1955, 1997.

[13] F. Y. Shih and W.-T. Wong, "Fully parallel thinning with tolerance to boundary noise," *Pattern Recognit.*, vol. 27, no. 12, pp. 1677–1695, 1994.

[14] R. Stefanelli and A. Rosenfeld, "Some parallel thinning algorithms for digital pictures," *J. ACM*, vol. 18, no. 2, pp. 255–264, 1971.

[15] S. Suzuki and K. Abe, "Binary picture thinning by an iterative parallel two-subcycle operation," *Pattern Recognit.*, vol. 10, no. 3, pp. 297–307, 1987.

[16] H. Tamura, "A comparison of line thinning algorithms from a digital geometry viewpoint," in *Proc. 4th Int. Conf. Pattern Recognition*, 1978, pp. 715–719.

[17] "Technical Summary, Connection Machine Model CM-200," Thinking Machines Corporation, Cambridge, MA, Ver. 6.1, Oct. 1991.

[18] "Technical Summary, MasPar Machine Model MPP-12 000," Digital Equipment Corporation, Maynard, MA, Ver. 1.0, Jan. 1992.

[19] P. S. P. Wang and Y. Y. Zhang, "A fast and flexible thinning algorithm," *IEEE Trans. Comput.*, vol. 38, pp. 741–745, May 1989.

[20] T. Y. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns," *Commun. ACM*, vol. 27, no. 3, pp. 236–239, 1984.

A Fast Implementation of 3-D Binary Morphological Transformations

Nikos Nikopoulos and Ioannis Pitas

***Abstract*—This paper proposes a fast algorithm for implementing the basic operation of Minkowski addition for the special case of binary three-dimensional (3-D) images, using 3-D structuring elements of arbitrary size and shape. The application of the proposed algorithm for all the other morphological transformations is straightforward, as they can all be expressed in terms of Minkowski addition. The efficiency of the algorithm is analyzed and some experimental results of its application are presented. As shown, the efficiency of the algorithm increases with the size of the structuring element.**

***Index Terms*—Fast algorithm, image processing, mathematical morphology, 3-D binary morphological transformations.**

I. Introduction

Over the last two decades, mathematical morphology (MM) has proven itself as a powerful image processing and analysis tool [1], [2]. A plethora of successful applications of MM in different fields have been presented in the bibliography. A problem arised from the early stages of MM was the high computational complexity of the basic morphological transformations *dilation* and *erosion* [1]. This problem has hindered the application of MM in three-dimensional (3-D) image processing and analysis, which is very promising, since the basic theory of MM is based on set theory and can be easily extended to three and higher dimensions.

Many different techniques have been proposed for implementing the basic morphological operations more efficiently than by using their definition. Many of them involve the use of parallel computers or specialized hardware. Other techniques are restricted to two-dimensional (2-D) images [4], [5]. Also, most techniques are applicable only for

Manuscript received October 23, 1997; revised March 10, 1998. The associate editor coordinating the review of this paper and approving it for publication was Prof. Scott T. Acton.

The authors are with the Department of Informatics, University of Thessaloniki, 54006 Thessaloniki, Greece.

Publisher Item Identifier S 1057-7149(00)01154-4.