ELSEVIER

Pattern Recognition Letters 17 (1996) 197–207

Pattern Recognition Letters

A morphological point thinning algorithm

Joshua Brown, Anne Hoger *

Division of Mechanical Engineering, Department of Applied Mechanics and Engineering Sciences, University of California, San Diego, 9500 Gilman Dr. 0411, La Jolla, CA 92093-0411, USA

Received 16 August 1995; revised 2 November 1995

Abstract

Fiducial markers are used to determine object location, but in certain applications, such as biaxial strain analysis, markers may migrate. In such cases it is necessary to first establish existence and locate the approximate center of the fiducial. We have developed an algorithm (the MPT algorithm) which is an improvement of a well-known morphological thinning algorithm, and which can thin entities to a central pixel. This allows efficient location of markers. The improvement corrects a shortcoming of the previous algorithm which annihilated certain classes of small entities. The new algorithm safely thins small entities to their central points. Concave entities are acceptable since the algorithm preserves connectedness. This new algorithm will have applications in general scene analysis tasks.

Keywords: Fiducial markers; Biaxial strain measurement; Scene analysis; Mathematical morphology; Thinning algorithms; Machine vision

1. Introduction

In image analysis it is often desirable to identify and determine the positions of the distinct entities which compose an image. With respect to manufacturing of electronic parts such as printed circuit boards (PCB), it has been noted that accurate fiducial marker registration is essential for automatic placement of electronic components during manufacturing (Bose and Amir, 1990). Machine vision is used to locate the PCB on which the parts are to be placed. In this situation, the approximate locations of the fiducials are known *a priori*, so subpixel registration algorithms (Tian and Huhns, 1986) may be easily applied to find the center of the fiducial within a small neighborhood. In many other applications, however, the locations and even the number of fiducials are not known. For example, in biaxial strain analysis it can be convenient to use machine vision to track fiducial markers on a specimen in order to calculate the strain fields. (Charette et al., 1993; George and Bogen, 1991; Downs et al., 1990). The fiducials may be attached to the specimen in either a regular or random arrangement, but in any case, inhomogeneous material deformation can cause their relative locations to change. In the study of biological materials, biaxial tests at large strains are often performed to determine the mechanical properties of the materials. In these experiments, fiducial markers are often not added but, rather, are chosen from distinguishing marks on the specimen. These fiducials can migrate and change size and shape during the deformation. To use machine vision to track such irregular preexisting markers, an efficient means of

* Corresponding author. Email: ahoger@ucsd.edu

SSDI 0167-8655(95)00105-0

determining fiducial existence and approximate location is essential for providing subpixel registration algorithms with information on where to locally search for a fiducial.

We have investigated morphological transformation of binary digital images as a means to provide existence and approximate location information. The approach consists of thinning a binary digital entity down to its central point. At first glance, simple erosion may seem adequate for this task. But it is notable that in the general case of a fiducial, concavity may exist either by design or by chance. An example of this is the peanut shape, shown in Fig. 1. In this case, simple erosion may break connectivity of the thin connecting regions and indicate multiple fiducials where there is only one. For this reason, we have chosen to investigate morphological thinning algorithms rather than simple erosion techniques.

Much work has been done in the development of morphological thinning algorithms. The goal of these algorithms is the development of a medial axis transformation which, in essence, thins an entity down to a skeletal representation of its original shape without breaking connectivity. A rigorous analysis of thinning algorithms from the point of view of mathematical morphology is provided by Jang and Chin (1990), who propose a four-pass parallel thinning algorithm. They cyclically apply a set of four transformations to a 3×3 neighborhood of pixels to effect the thinning. Other parallel thinning algorithms have since been developed. A one-pass thinning algorithm (Jang and Chin, 1992) was developed which uses one pass-per-iteration instead of four, at the additional computational expense of transforming neighborhoods larger than 3×3 pixels. Another algorithm, developed in (Glazer, 1993), is a four-pass type and transforms a 3×3 neighborhood. Although similar to that proposed in (Jang and Chin, 1990), Glazer's algorithm differs in that it tends to preserve skeletal legs. The algorithm proposed in (Datta and Parui, 1994) also preserves skeletal legs. Another recent algorithm proposed in (Shih and Wong, 1994) is resistant to the boundary noise that tends to create spurious skeletal legs, again at the expense of transforming a neighborhood larger than 3×3 pixels.

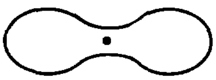
Fig. 1. Concave fiducial.

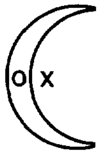

Fig. 2. Crescent showing midpoint of medial axis (o) and centroid (x).

In contrast with these algorithms, which generate digital skeletons and therefore seek to preserve skeletal legs, our goal is to thin binary digital entities down to a single central pixel as efficiently as possible. This central pixel is defined as the midpoint of the medial axes; it is the last remaining pixel when a thinning algorithm iteratively removes the end pixel from each skeletal leg of the entity being thinned. This midpoint often differs from the centroid of the object. The distinction between the two can be seen upon consideration of a crescent shape, shown in Fig. 2, in which the midpoint of the medial axis lies within the entity's boundaries. By contrast, the centroid of a crescent may in fact lie outside its boundaries (see Fig. 2). We chose to investigate Jang and Chin's algorithm, referred to as "Algorithm B" in (Jang and Chin, 1990), since it distinguished itself for our purposes by both allowing for the shortening of skeletal legs via a trimming operation and requiring a computationally efficient neighborhood of only 3×3 pixels for the transformation. In their analysis, Jang and Chin (1990) noted that "In an extreme situation Algorithm B plus the trimming operation, when applied to a one-pixel thick limb, transforms it to a single point because the trimming operation treats the two end points as noise spurs of unit length." Though Algorithm B is in other respects rigorously constructed and analyzed in (Jang and Chin, 1990), we will show here that Algorithm B plus the trimming operation will in fact completely annihilate certain classes of small entities. This paper presents an improvement of Algorithm B, which we call the Morphological Point Thinning (MPT) algorithm, that eliminates the risk of small-entity annihilation and is therefore suitable for providing the existence and approximate location data of general case fiducial markers.

The paper is organized as follows: definitions of the relevant terminology and a review of Algorithm B (Jang and Chin, 1990) are given in Section 2. In Section 3, we provide an analysis of the annihilation conditions and formulate the new algorithm. Example results are presented in Section 4, and in Section 5 we discuss the significant properties of the algorithm.

2. Definitions and background

Since the work here is essentially an improvement of Algorithm B, we adopt much of the terminology used in (Jang and Chin, 1990). We begin with definitions of some relevant terms and a description of Algorithm B.

2.1. Definitions

$\mathbb{Z}$	set of all integers
$\mathbb{Z}^2$	two-dimensional grid of discrete points
$A, B, C, \ldots$	subsets of $\mathbb{Z}^2$
$\mathbf{A}, \mathbf{B}, \mathbf{C}, \ldots$	families of subsets of $\mathbb{Z}^2$, e.g. $\{A^i\}$
$a, b, c, \ldots$	2-dimensional vector locations of points in $\mathbb{Z}^2$
$X \circledcirc Y$	hit-and-miss transform of X by Y
$X \circ Y$	thinning of X by Y
$\psi(S)$	transformation of set S by ψ

Hit-and-miss transform $(X \circledcirc Y)$. A pattern matching operation based on Minkowski set addition and subtraction. The details of the hit-and-miss transform can be found in (Jang and Chin, 1990); basically, the hit-and-miss transform picks out every point x in X whose immediate neighborhood matches exactly the neighborhood of points in Y (Fig. 3). For $X \circledcirc Y$, the set Y is a subset of $\mathbb{Z}^2$ called the template. It consists of a pattern of 1's and 0's whose center point is the middle pixel of the 3×3 neighborhood. Then the hit-and-miss transform is defined as

$$X \circledcirc Y = \{x \colon Y_x \subset X;\ Y_x^c \subset X^c\} \tag{1}$$

where Y_x denotes template Y translated in $\mathbb{Z}^2$ so that its center point corresponds to x in X.

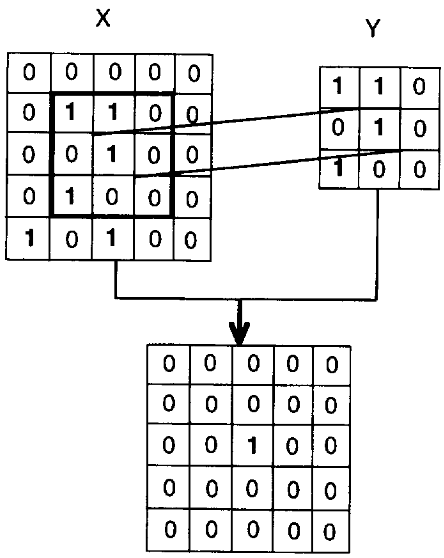

Fig. 3. Example of hit-and-miss transform.

Thinning operator $(X \circ Y)$. Removes from X all points whose neighborhoods match the neighborhoods in template Y:

$$X \circ Y = X - (X \circledcirc Y). \tag{2}$$

Entity. A single group of contiguous non-zero pixels that is isolated from other groups of non-zero pixels.

Class of entities. An entity together with all of its rotational and reflectional variants.

Annihilation. All pixels of an entity have been removed so that the entity no longer exists.

Stable. A small entity won't be annihilated by the next pass of the thinning algorithm, whichever pass it may be.

Critical point. A point which, if removed, would generate a hole or a disconnectivity.

4-connected. A point x in X has eight nearest neighbors. Four of these are non-diagonal and correspond to the north, south, east, and west. The remaining for are termed diagonal neighbors. An entity X consisting of two or more points $x_1, x_2, \ldots, x_n$ is 4-connected when there exists a unique path between x_1 and x_n such that each x_i is non-diagonally (north, south, east, or west) adjacent to its neighbor x_{i+1} (Fig. 4).

8-connected. An entity X consisting of points $x_1, x_2, \ldots, x_n$ is 8-connected when there exists a

0	0	1	1	0
0	0	0	1	0
0	0	1	1	0
1	1	1	0	0
0	0	0	0	0

Fig. 4. 4-connected example.

unique path between x_1 and x_n such that each x_i is either diagonally or non-diagonally adjacent to its neighbor x_{i+1} (Fig. 5).

Singly-connected. An entity X consisting of a set of points $x_1, x_2, \ldots, x_n$ is singly-connected if there is a unique path from x_1 to x_n such that every point x_i is either (1) uniquely 8-connected to its neighbor x_{i+1}, or (2) if it is not uniquely 8-connected to its neighbor x_{i+1}, then it is uniquely 4-connected to its neighbor x_{i+1} (Fig. 6). In addition, X must have no holes, i.e. $x_1 \neq x_n$.

2.2. *Jang and Chin's Algorithm B*

A 4-pass-per-iteration morphological thinning algorithm, called Algorithm B, has been developed in (Jang and Chin, 1990). The algorithm serves to thin a binary digital image entity down to its medial axes. Essentially, each pass of the algorithm applies a particular subset of the thinning templates **D** and **E**, shown in Fig. 7, to pick off boundary pixels of the entity. In particular, let i represent the particular pass within an iteration. Then for $1 \leqslant i \leqslant 4$, Algorithm B applies templates D^i, D^{i+1}, and E^i, where $D^5 = D^1$. All 3×3 neighborhoods which match the specified templates have their center pixel removed. Mathematically, if S is the image entity to be thinned and $\psi(S)$ is the resulting thinned image, then

$$\psi(S) = S - \left((S \circledcirc D^i) \cup (S \circledcirc D^{i+1}) \cup (S \circledcirc E^i)\right) \tag{3}$$

0	0	1	0	0
0	0	0	1	0
0	0	1	0	0
1	1	0	0	0
0	0	0	0	0

Fig. 5. 8-connected example.

0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
1	1	0	1	0
0	0	0	0	1

singly-connected

0	0	1	0	0
0	0	0	1	0
0	0	1	0	0
1	1	0	1	0
0	0	0	0	1

non-singly-connected

Fig. 6. Examples of singly- vs. non-singly-connected.

or, equivalently,

$$\psi(S) = (S \circ D^i) \cap (S \circ D^{i+1}) \cap (S \circ E^i). \tag{4}$$

Jang and Chin prove that the algorithm results in a converged, 8-connected, one-pixel-thick skeleton.

Jang and Chin note that the skeletons generated by their algorithm tended to contain spurious legs which result from random boundary noise. To correct for this, they introduce a set of trimming templates G^1–G^8 (Fig. 8), which have the effect of removing the end pixel from a skeletal leg. Each of the templates **G** removes end pixels from different directions.

3. The morphological point thinning (MPT) algorithm

Jang and Chin's Algorithm B must be modified before it can be successfully applied to thin an entity to the midpoint of its medial axes, because with their algorithm there is the possibility of completely anni-

0	0	x
0	1	1
x	1	x

D^1

x	0	0
1	1	0
x	1	x

D^2

x	1	x
1	1	0
x	0	0

D^3

x	1	x
0	1	1
0	0	x

D^4

x	0	x
1	1	1
x	1	x

E^1

x	1	x
1	1	0
x	1	x

E^2

x	1	x
1	1	1
x	0	x

E^3

x	1	x
0	1	1
x	1	x

E^4

Fig. 7. Templates **D** and **E**.

hilating small entities. Such annihilation may occur if all 8 trimming templates **G** are applied in conjunction with each pass. It is desirable for a point-thinning algorithm that it always reduces an entity to a single central point, without annihilating it. Analysis will show that this requirement can be fulfilled by the application of a certain subset of trimming templates **G** in conjunction with each pass of Algorithm B. By considering entities in order of increasing size, this analysis will establish all classes of entities which may be destroyed by simultaneous application of templates G^1–G^8 with each pass. From this information, a set of conditions will be determined and used to formulate a new algorithm, the Morphological Point Thinning (MPT) algorithm.

3.1. One-pixel entities

It is easily shown that single-pixel entities are preserved in all cases, since a single-pixel entity is not removed by any of the templates **D**, **E**, or **G**. The case of two-pixel entities, however, presents the first constraint.

3.2. Two-pixel entities

Condition 1. No more than 4 trimming templates **G** may be applied simultaneously without the risk of losing unique 2-pixel entities. Specifically,

$$\exists S \subset \mathbb{Z}^2 \colon (S \circ G^i) \cap (S \circ G^{i+4}) = \emptyset \quad \text{for } 1 \leqslant i \leqslant 8 \tag{5}$$

where $G^9 = G^1$, $G^{10} = G^2$, and so on.

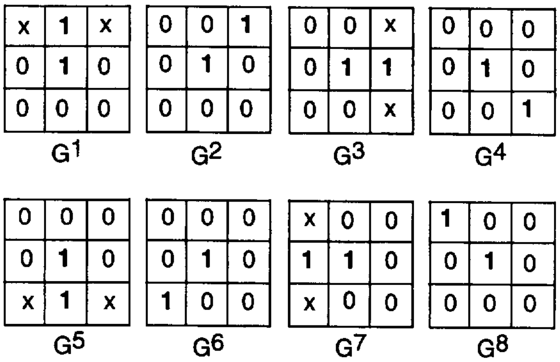

Fig. 8. Trimming templates **G**.

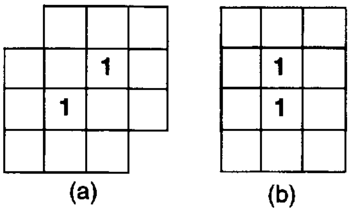

Fig. 9. Entity classes subject to annihilation by Condition 1.

To establish the condition, it is necessary to give two example sets S which are destroyed by a specific pair of G templates. The two cases correspond to even and odd choice of i. With the examples established, the other template cases follow by induction when each template pair is rotated 90°.

First, let i be an even number such as 2. Fig. 9(a) shows the unique object S which will be destroyed by the simultaneous application of the G templates specified by Condition 1 when $i = 2$. Successive rotations of the object in Fig. 9(a) by 90° yield example sets S to verify the condition for all even i. Now, let $i = 1$. Fig. 9(b) shows the unique object S which will be destroyed by the simultaneous application of these templates, just as in the case of even i. Again, successive rotations of the object by 90° yield example sets S to verify the condition for all odd i.

Now we consider 3-pixel entities, having considered all possible classes of 2-pixel entities.

3.3. Three-pixel entities

We divide the group of three-pixel entities into two categories: those that are 8-connected and those that are not. First, we examine 8-connected entities and find:

Proposition 1. *An entirely 8-connected entity of size-3 pixels or greater is stable.*

Let S be such an entity. It follows from Jang and Chin's proof of convergence and connectedness of Algorithm B that for $1 \leqslant i \leqslant 4$, $S \circledcirc D^i = \emptyset$ and $S \circledcirc E^i = \emptyset$. Thus, it is seen that all templates **D** and **E** have no effect on the entity. Furthermore, all templates **G** remove single pixels which are 8-connected to only one other pixel. An entity S has exactly two such pixels, and so at most two pixels

	1	1	
	1		

Fig. 10. Entity classes subject to annihilation by Condition 2.

will be removed from S by the application of any or all templates **G**. At least one pixel will remain, and so the entity is stable.

The case of 3-pixel entities which are not 8-connected is considered to complete the examination of 3-pixel entities. There is only one class of entities of this type, which is shown in Fig. 10. This is the class of 4-connected, 3-pixel entities, and it presents a possible annihilation condition.

Condition 2. There exist 4 cases of a unique group of trimming templates **G** which cannot be applied in conjunction with a corresponding unique group of thinning templates **D**, without risk of annihilating a corresponding unique entity. Specifically,

$$\exists S \subset \mathbb{Z}^2 \colon (S \circ G^{2i+5}) \cap (S \circ G^{2i+7}) \cap (S \circ D^i) = \emptyset \quad \text{for } 1 \leqslant i \leqslant 4 \tag{6}$$

where $G^9 = G^1$, $G^1 0 = G^2$; $D^5 = D^1$, and so on.

To see this, the case of $i = 1$ is considered. Fig. 10 shows the unique entity S which will be destroyed by the simultaneous application of the templates G^7, G^1, and D^1. The remaining cases may be proved by induction. The entity in Fig. 10 is rotated by 90° clockwise to produce the annihilated entity in the case of $i + 1$.

3.4. Four-pixel entities

The case of 4-pixel entities is more complex as there is a larger group of classes to consider. We divide this group into the singly-connected entities and the four remaining non-singly-connected classes. We consider the singly-connected classes first and find that:

Proposition 2. *A singly-connected entity of size 4 is stable.*

To see this, let S be such an entity. It can be assumed that the two end pixels may be lost by

			1	
		1		
	1	1		

One 4-connected end

		1		
	1	1		
	1			

Two 4-connected ends

Fig. 11. Singly-connected entities of four pixels.

application of templates **G**. Attention will be focused on the two middle pixels. If both end pixels are 8-connected to their neighbors, then S is entirely 8-connected, and Proposition 1 guarantees that S is stable. If either or both end pixels are 4-connected with their neighbors (Fig. 11), then Jang and Chin's proof of connectedness of Algorithm B (Jang and Chin, 1990) guarantees that not more than one of the two middle pixels will be removed. Otherwise, if both pixels were removed, a disconnectivity would result in the untrimmed skeleton. Thus, at most 3 pixels may be removed from S, so it is stable. Likewise, it may be noted that if another entity is joined to an end of S so that the singly-connected elements remain singly-connected, then the resulting larger entity is stable by the same proof of connectedness.

There are four classes of 4-pixel entities which are not singly-connected. The first, class 4-1, is shown in Fig. 12. For the entity S in Fig. 12, it is only when templates **D** are applied that the transform is non-empty ($\mathbf{S} \circledcirc \mathbf{D} \neq \emptyset$). Thus, since by definition of Algorithm B only one **D** template is applied with each pass, at most one pixel will be removed, and so the entity is stable.

The remaining three classes are intersection, in which one pixel is surrounded by 3 nearest neigh-

	1	1		
	1	1		

Fig. 12. Entity class 4-1.

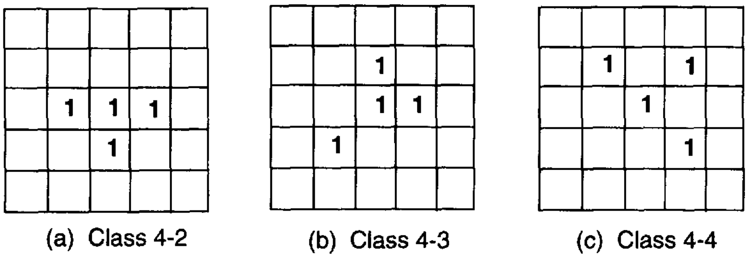

Fig. 13. Entity classes 4-2, 4-3, and 4-4.

bors. These three classes are shown in Fig. 13. It may be shown that for each class, the 3 outer pixels are eliminated by an appropriate choice of **G** templates. Attention will be focused on the center pixels. If S is in either of the classes 4-3 or 4-4 (shown in Fig. 13), it may be seen that for $1 \leqslant i \leqslant 4$, $S \circledcirc D^i = \emptyset$ and $S \circledcirc E^i = \emptyset$. Essentially, the center pixel will remain in both cases; the entity is stable. In another light, Jang and Chin's proof of connectedness guarantees that the center pixel will not be removed, because doing so would necessarily cause a disconnectivity in the entity. In contrast, examination of Fig. 13(a) yields a condition which may cause annihilation.

Condition 3. There exist 4 cases of a unique group of trimming templates **G** which cannot be applied in conjunction with a corresponding unique group of thinning templates **E**, without risk of annihilating a corresponding unique entity. Specifically,

$$\exists S \subset \mathbb{Z}^2 \colon (S \circ G^{2i+1}) \cap (S \circ G^{2i+5}) \cap (S \circ G^{2i+7}) \cap (S \circ E^i) = \emptyset \quad \text{for } 1 \leqslant i \leqslant 4 \tag{7}$$

where $G^9 = G^1$, $G^{10} = G^2$; $E^5 = E^1$, and so on.

To see this, the case of $i = 1$ is considered. Fig. 13(a) shows the unique entity which will be destroyed by the simultaneous application of templates E^1, G^1, G^3, and G^7. As before, the remaining cases are proved by induction. The entity in Fig. 13(a) is rotated by 90° clockwise to provide the example entity for the case of $i+1$.

3.5. Five-pixel entities

The case of five-pixel entities is rather complex, and no additional annihilation conditions will be found. We note that:

Proposition 3. *An entity of size 5 or greater is always stable.*

We leave the discussion of this proposition, which completes the analysis of annihilation conditions, to the Appendix.

3.6. Formulation of the MPT algorithm

All pixel classes for each entity size have been examined. Conditions 1–3 may be summarized as all unacceptable combinations of templates. By Condition 1, the following combinations of templates are unacceptable:

(a) G^1 and G^5;
(b) G^2 and G^6;

Table 1
Possible combinations of trimming templates

	Algorithm B	G Group 1	G Group 2	G Group 3	G Group 4
Pass 1	D^1, D^2, E^1	G^2 or G^6	G^3 or G^7	G^4 or G^8	G^5
Pass 2	D^2, D^3, E^2	G^1 or G^5	G^2 or G^6	G^4 or G^8	G^7
Pass 3	D^3, D^4, E^3	G^1	G^2 or G^6	G^3 or G^7	G^4 or G^8
Pass 4	D^4, D^1, E^4	G^1 or G^5	G^2 or G^6	G^3	G^4 or G^8

(c) G^3 and G^7; and
(d) G^4 with G^8.

By Condition 2, the following combinations of templates are unacceptable:
(a) D^1, G^7, and G^1;
(b) D^2, G^1, and G^3;
(c) D^3, G^3, and G^5; and
(d) D^4, G^5, and G^7.

By Condition 3, the following combinations are unacceptable:
(a) E^1, G^1, G^3, and G^7;
(b) E^2, G^1, G^3, and G^5;
(c) E^3, G^3, G^5, and G^7; and
(d) E^4, G^1, G^5, G^7.

From the conditions, it may be seen that up to 4 trimming templates **G** may be applied in conjunction with each pass. Table 1 shows the template combinations that are acceptable for each of the 4 thinning passes of Algorithm B. Each of the four **G** groups in Table 2 shows the acceptable **G** template choices for a given pass of the algorithm.

Two criteria are imposed in addition to Conditions 1–3. These are: (1) for efficiency, the number of iterations needed for convergence should be minimized, and (2) the thinning should be isotropic. The first criterion pertains to efficiency, that the number of iterations needed for convergence should be minimized. Essentially, each of the 4 thinning passes must apply the maximum possible 4 trimming templates. The second criterion is imposed to maximize accuracy. The trimming templates must be applied equally; no one template may be applied more than the others in any 4 consecutive passes. From these criteria and the data in Table 1, the Morphological Point Thinning (MPT) algorithm, a modification of Jang and Chin's Algorithm B, is proposed. Each of the 4 passes is characterized by 3 thinning templates and 4 trimming templates applied in parallel, as shown in Table 2. The algorithm can be succinctly stated as a modification of Eq. (4):

$$\psi(S) = (S \circ D^i) \cap (S \circ D^{i+1}) \cap (S \circ E^i) \cap \bigcap_{j=1}^{4} (S \circ G^{i+j}) \tag{8}$$

where $D^5 = D^1$, $E^5 = E^1$, $G^9 = G^1$, and so on.

Table 2
The MPT algorithm

	Algorithm B	Trim 1	Trim 2	Trim 3	Trim 4
Pass 1	D^1, D^2, E^1	G^2	G^3	G^4	G^5
Pass 2	D^2, D^3, E^2	G^4	G^5	G^6	G^7
Pass 3	D^3, D^4, E^3	G^6	G^7	G^8	G^1
Pass 4	D^4, D^1, E^4	G^8	G^1	G^2	G^3

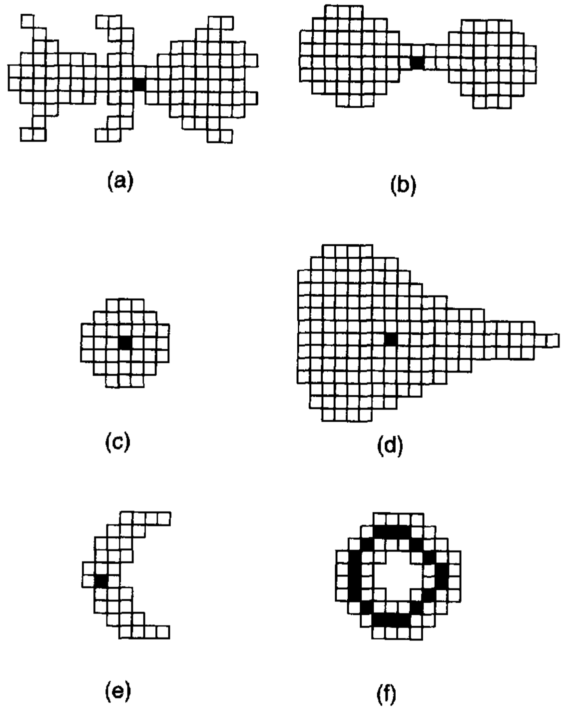

Fig. 14. Example results.

4. Example results

Fig. 14 shows the results of the MPT algorithm applied to various shapes: an insect, a peanut, a circle, a pear, a crescent, and a donut. Figs. 14(a) and 14(b) show the ability of the algorithm to uniquely identify complex shapes in which larger sections are joined by thin connecting regions. This feature is an advantage over simple erosion methods, which would break the connectivity which is preserved here. Still, the disk and pear shapes of Figs. 14(c) and 14(d), respectively, show that the algorithm is capable of isotropically eroding simpler entities. A significant feature of the algorithm is also seen in its treatment of Fig. 14(e), in which the located center point lies significantly away from the entity centroid. This serves to illustrate that the MPT algorithm produces convergence to the midpoint of the medial axes rather than the centroid. The located center point is guaranteed to lie within the actual boundaries of the original entity. Finally, Fig. 14(f) shows convergence to an 8-connected ring rather than a central point and serves to illustrate a limitation of the MPT algorithm, that it will not find the central pixel of entities containing holes.

5. Discussion and conclusion

5.1. Convergence

It may be seen that the MPT algorithm will always thin a solid entity to a single point. Jang and Chin's proof of convergence of Algorithm B guarantees that an entity will converge to a 1-pixel-thin 8-connected skeleton. It should be noted that if any holes exist, then the entity is not solid and will converge to an 8-connected ring or set of contiguous rings, not a point. Solidity of the entity is therefore a necessary condition for convergence to a point. For a solid entity, the converged skeleton will show skeletal legs at its extremities. The ends of each of these legs will be reduced one pixel in length by the corresponding **G** template. The entity will not be annihilated at any intermediate step, and the legs will be shortened with each pass until only a single pixel remains of the entity.

5.2. Complexity

The number of iterations required for convergence is related to the size of the entity. Fig. 14(a), for example, required six iterations for convergence. Figs. 14(b)–(e) required six, two, seven, and three iterations for convergence, respectively. It may be seen from these examples that the number of iterations required is approximately equal to one-half of the length, in pixels, between the center of the entity as determined by the algorithm and the point farthest from that center point, following the most direct 8-connected path through the entity. If this 8-connected path is considered as a skeletal leg, then the above observation may also be made from the fact that the algorithm applies each trimming template G^i twice in the course of an iteration, so that each skeletal leg is reduced by two pixels in length for each iteration.

There is an upper bound to the complexity of the algorithm. By the proof of convergence of Algorithm B, not less than one layer of boundary pixels will be removed from the entity for each iteration, so that the maximum number of iterations required for convergence will not be more than one-half of the length, in pixels, between the two most mutually distant points of the entity, following the most direct 8-connected path through the entity between the two points. Thus, if the number of pixels in this 8-connected path is n, the worst-case complexity of the algorithm is of $O(n)$.

5.3. Limitations

It is notable that the entity locations determined by the algorithm are limited to integral pixel values. Other methods, such as those suggested in (Tian et al., 1986) must be applied if further accuracy is required. Also, since holes in the original binary image entities will preclude convergence to a point, care must be taken to ensure that the concave features, for example the crescent shape, do not completely enclose regions of zero-value pixels, thus forming a hole.

5.4. Conclusion

We have presented an algorithm to identify and determine the approximate locations of irregularly placed fiducial markers. The MPT algorithm, as we call it, is an improvement of an existing morphological thinning algorithm in that it does not annihilate small entities. We have shown that this algorithm will find the center pixel of an arbitrarily-shaped entity provided that no holes exist within the entity. Although other methods must be used if greater precision is required, the MPT algorithm provides an efficient means of approximately locating irregularly placed fiducials. In addition to fiducial marker registration, the MPT algorithm will be useful in broader

	1	1		
	1	1		
			1	

class 5-1a

	1	1		
	1	1	1	

class 5-1b

Fig. 15. Classes 5-1a and 5-1b.

applications to scene analysis when it is necessary to locate many small and complex objects to pixel accuracy with no previous existence information.

Appendix

Proposition 3. *An entity of size 5 or greater is always stable.*

We divide the group of five-pixel-or-greater entities into five categories and consider each in turn to demonstrate the proposition.

Category 1. The first category consists of supersets of the class of entities 4-1 (Fig. 15), and it will be shown that any superset of the class is stable. Figs. 15(a) and 15(b) show the two five-pixel supersets of class 4-1. To see that both classes are stable, we simply note that in Figs. 15(a) and 15(b), removal of the lower-left and upper-right pixels in one pass would require the simultaneous application of templates D^2 and D^4, which does not happen according to the definition of Algorithm B. Similar arguments, i.e. that mutually exclusive **D** and **E** templates would have to be applied to cause annihilation, hold for all other members of classes 5-1a and 5-1b as well as in the general case of 6-or-more-pixel supersets of class 4-1.

Category 2. The second category consists of the singly-connected entities described by Proposition 2. These entities are considered in the case of a single pixel being added to increase the size to 5. By induction, more pixels may be added to the entity in each case to show that the general case of size greater than 5 is also stable. The added pixel may effect one of the following four changes in the entity:

(1) It may lengthen an end, in either a 4- or 8-connected sense, in which case the same argument as in Proposition 2 may be employed to prove stability.

(2) It may make an 8-connection into a 4-connection, in which case again the same argument as in Proposition 2 may be employed to prove stability.

(3) It may make a 4-connection into a 2×2 pixel thick entity, which is stable as discussed in the above analysis of entity classes 5-1a and 5-1b.

(4) It may add a skeletal leg spur, which will be removed by a **G** template. Nonetheless, the proof of connectedness guarantees that at least one pixel will remain to preserve connectedness as discussed in the proof of Proposition 2.

Categories 3, 4, 5. The third, fourth, and fifth categories consist of supersets of class 4-2 through 4-4 (Fig. 13). For these three classes, the addition of a pixel will create one of 4 possible situations:

(1) It may create a superset of class 4-1, which is stable as discussed above.

(2) It may fail to prevent the existence of a critical point in the resulting entity, as in the case of class 4-4. By Jang and Chin's proof of connectedness of Algorithm B, entities with critical points are stable.

(3) It may lengthen an end, in either a 4- or 8-connected sense, which is stable since it will be seen that the entity becomes identical to a superset of the singly-connected entity of length 4 discussed above, having 1 or more skeletal leg spurs.

(4) It may make an 8-connection into a 4-connection, which results in the same features described by case (3) above.

Having considered all classes of size-five pixels or greater, the discussion of Proposition 3 is complete. We conclude that Conditions 1–3 provide the complete set of annihilation conditions.

Acknowledgment

This work was supported by an SAIC Summer Fellowship (Brown), a grant from the Academic

Senate of the University of California, San Diego (Hoger), and Presidential Young Investigator Award CMS 90-57629 from NSF (Hoger).

References

Bose, C. and I. Amir. (1990). Design of fiducials for accurate registration using machine vision. *IEEE Trans. Pattern Anal. Mach. Intell.* 12, 1196–1200.

Charette, P., I.W. Hunter and P.J. Hunter (1993). Biaxial testing of pericardium using electronic speckle pattern interferometry (ESPI). *Proc. SPIE* 1756, 196–202.

Datta, A. and S. Parui (1994). A robust parallel thinning algorithm for binary images. *Pattern Recognition* 27, 1181–1192.

Downs, J., H.R. Halperin, J. Humphrey and F. Yin (1990). An improved video-based computer tracking system for soft biomaterials testing. *IEEE Trans. Biomed. Engrg.* 37, 903–907

George, D.T. and D.K. Bogen (1991). A low-cost fiber-optic strain gage system for biological applications. *IEEE Trans. Biomed. Engrg.* 38, 919–924.

Glazer, F. (1993). An alternative morphological thinning algorithm. *Proc. SPIE* 2094, 1331–1339

Jang, B.K. and R.T. Chin (1990). Analysis of thinning algorithms using mathematical morphology. *IEEE Trans. Pattern Anal. Mach. Intell.* 12, 541–551.

Jang, B.K. and R.T. Chin (1992). One-pass parallel thinning: analysis, properties, and quantitative evaluation. *IEEE Trans. Pattern Anal. Mach. Intell.* 14, 1129–1140.

Shih, F.Y. and W.T. Wong (1994). Fully parallel thinning with tolerance to boundary noise. *Pattern Recognition* 27, 1677–1695.

Tian, Q. and M. Huhns. (1986). Algorithms for subpixel registration. *Computer Vision, Graphics, and Image Processing* 35, 220–233.