Available online at www.sciencedirect.com

ScienceDirect

Pattern Recognition Letters 28 (2007) 501–506

Pattern Recognition Letters

www.elsevier.com/locate/patrec

A note on 'A fully parallel 3D thinning algorithm and its applications'

Tao Wang, Anup Basu *

Department of Computing Science, University of Alberta, Edmonton, Canada AB T6G 2E8

Received 7 October 2005; received in revised form 8 July 2006
Available online 13 November 2006

Communicated by G. Borgefors

Abstract

A 3D thinning algorithm erodes a 3D binary image layer by layer to extract the skeletons. This paper presents a correction to Ma and Sonka's thinning algorithm, 'A fully parallel 3D thinning algorithm and its applications', which fails to preserve connectivity of 3D objects. We start with Ma and Sonka's algorithm and examine its verification of connectivity preservation. Our analysis leads to a group of different deleting templates, which can preserve connectivity of 3D objects.
© 2006 Elsevier B.V. All rights reserved.

Keywords: Thinning; Skeletonization; 3D; Parallel

1. Introduction

Thinning is a useful technique having potential applications in a wide variety of problems. It creates a compact representation (*skeleton*) of the models that may be used for further processing. 3D skeletons can be used in many applications (Besl and Jain, 1985; Funkhouser et al., 2003) such as 3D pattern matching, 3D recognition and 3D database retrieval.

A *3D binary image* is a mapping that assigns the value of 0 or 1 to each point in the 3D space. Points having the value of 1 are called *black* (*object*) points, while 0's are called *white* (*background*) ones. Black points form objects of the binary image. The thinning operation iteratively *deletes* or *removes* some object points (that is, changes some black points to white) until only some restrictions prevent further operation. Note that the white points will never be changed to black ones in any circumstances. Most of the existing thinning algorithms are parallel, since the *medial axis transform* (MAT) can be defined as *fire front propagation*, which is by nature parallel (Blum, 1967). There are three categories of parallel thinning algorithms in literature, *sub-iteration parallel* thinning algorithm (Palagyi and Kuba, 1998; Lohou and Bertrand, 2004; Palagyi and Kuba, 1999), *sub-field parallel* thinning algorithm (Saha et al., 1997; Bertrand et al., 1994) and *fully parallel* thinning algorithm (Ma and Sonka, 1996). Brief surveys of algorithms in each category can be found in the literature (Palagyi and Kuba, 1999; Ma and Sonka, 1996).

The rest of this paper is organized as follows. In Section 2, some basic concepts will be presented. Section 3 will briefly discuss Ma and Sonka's algorithm (Ma and Sonka, 1996). The problematic part in this algorithm is analyzed and the modification is presented in Section 4, before the work is concluded in Section 5.

2. Basic concepts

We first describe some terms and notation:

Let p and q be two different points with coordinates (px, py, pz) and (qx, qy, qz), respectively, in a 3D binary image P. The Euclidean distance between p and q is defined as

$$\text{dis} = \sqrt{(px-qx)^2+(py-qy)^2+(pz-qz)^2}$$

* Corresponding author. Tel.: +1 780 492 3330; fax: +1 780 492 1071.
E-mail addresses: taowang@cs.ualberta.ca (T. Wang), anup@cs.ualberta.ca (A. Basu).

0167-8655/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2006.09.004

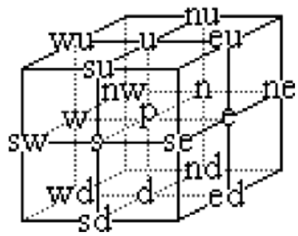

Fig. 1. The adjacencies in a 3D binary image. Points in $N_6(p)$ are marked *u*, *n*, *e*, *s*, *w*, and *d*. Points in $N_{18}(p)$ but not in $N_6(p)$ are marked *nu*, *nd*, *ne*, *nw*, *su*, *sd*, *se*, *sw*, *wu*, *wd*, *eu*, and *ed*. The unmarked points are in $N_{26}(p)$ but not in $N_{18}(p)$.

Then p and q are 6-*adjacent* if $dis \leqslant 1$. 18-*Adjacent* if $dis \leqslant \sqrt{2}$. 26-*Adjacent* if $dis \leqslant \sqrt{3}$. Let us denote the set of points k-*adjacent* to point p by $N_k(p)$, where $k = 6, 18, 26$ (see Fig. 1). $N_k(p)$ is also called p's k-*neighborhood*. Let p be a point in a 3D binary image. Then, $e(p)$, $w(p)$, $n(p)$, $s(p)$, $u(p)$, and $d(p)$ are 6-*neighbors* of p, which represent six directions of *east*, *west*, *north*, *south*, *up*, and *down*, respectively. The 18-*neighbors* of p (but not in p's 6-*neighborhood*) are $nu(p)$, $nd(p)$, $ne(p)$, $nw(p)$, $su(p)$, $sd(p)$, $se(p)$, $sw(p)$, $wu(p)$, $wd(p)$, $eu(p)$, and $ed(p)$, which represent 12 directions of *north-up*, *north-down*, *north-east*, *north-west*, *south-up*, *south-down*, *south-east*, *south-west*, *west-up*, *west-down*, *east-up*, and *east-down*, respectively.

It is very important for thinning algorithms to *preserve connectivity* for 3D objects (Palagyi and Kuba, 1998; Ma and Sonka, 1996). If a thinning algorithm fails to preserve connectivity, the skeletons extracted from the object will be disconnected, which is unacceptable in many applications. A sequential thinning algorithm can preserve connectivity easily if it is only allowed to delete *simple points* (Kong, 1995; Bertrand, 1994; Saha and Chaudhuri, 1994). However, a parallel thinning algorithm may delete many black points in every iteration, even if it is only allowed to delete simple points, the algorithm may not preserve connectivity. This problem was investigated in (Kong, 1993; Ma, 1994; Bertrand, 1995).

3. Ma and Sonka's algorithm

In 1996, Ma and Sonka proposed a fully 3D thinning algorithm (Ma and Sonka, 1996), which was applied to many applications such as medical image processing

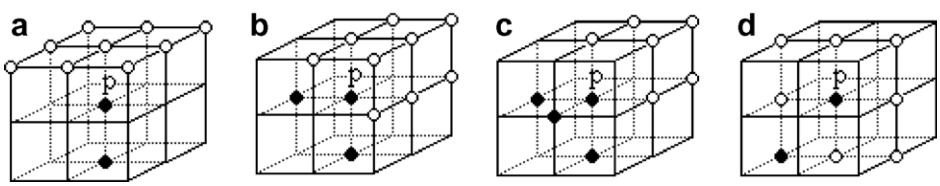

Fig. 2. Four template cores (Classes A, B, C and D) of the fully parallel thinning algorithm. A "●" is an object point. A "○" is a background point. An unmarked point is a "don't care" point that can be either an object point or a background point. For (d), there is an additional restrict that p must be a simple point.

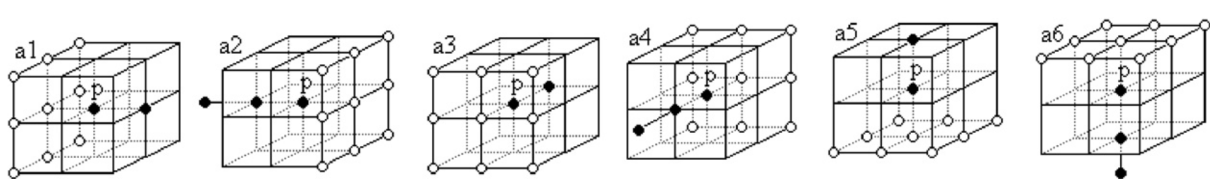

Fig. 3. Six deleting templates in Class A.

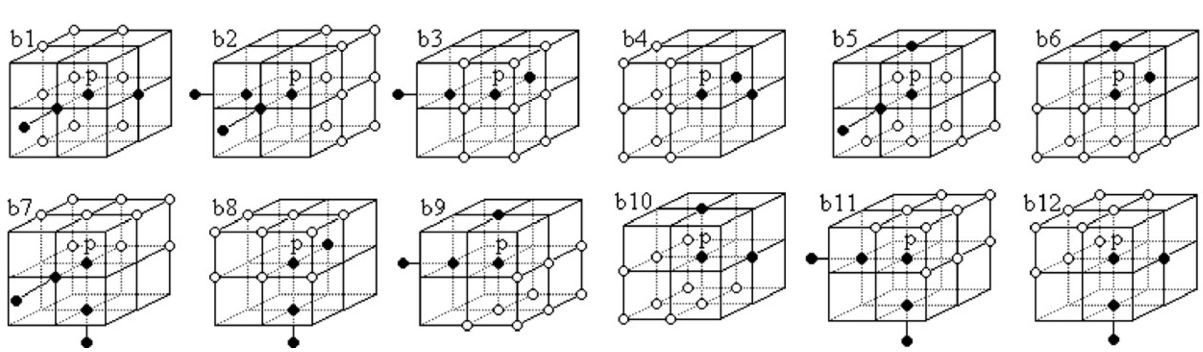

Fig. 4. Twelve deleting templates in Class B.

Download English Version:

https://daneshyari.com/en/article/535582

Download Persian Version:

https://daneshyari.com/article/535582

Daneshyari.com