Journal of Computer-Assisted Microscopy, Vol. 10, No. 3, 1998

Three-Dimensional Thinning Algorithm that Peels the Outmost Layer with Application to Neuron Tracing

Xingbai He,[1,2] Eric Kischell,[1] Marc Rioult,[3] and Timothy J. Holmes[1,2]

A novel thinning algorithm for three-dimensional (3D) binary images is presented, with applications in the study of neuronal micro-anatomy by light microscopy. This algorithm satisfies properties important to many biological applications, including (a) connectivity preservation, (b) thinness, and (c) geometry preservation. It is fast to execute (a few minutes for typical 3D data sizes) on a personal computer. The algorithm addresses many challenges that are presented by 3D data. Algorithm improvements, over precursory algorithms, include the following: (1) Stricter and more exhaustive constraints on identifying *outmost-layer border* points are applied. (2) Border points are deleted by a novel *ascending order of weighted neighbor count* approach. The algorithm is robust in that it retains the above three properties (a–c) in the presence of relatively severe noise, uneven dye uptake, and nonuniform background.

KEY WORDS: Three-dimensional thinning; neuron tracing; skeleton; light microscopy.

INTRODUCTION

Thinning is a layer-by-layer erosion of a binary object until only a "skeleton" of the object remains. In recent years the collection of three-dimensional (3D) data sets in light microscopy has become commonplace (Pawley, 1995). Thinning on 3D data sets is much more challenging than thinning on the more conventional two-dimensional 2D data sets. The three major challenges are listed as follows:

1. *Connectivity preservation* (Latecki and Ma, 1996; Ma and Sonka, 1996; Lee and Kashyap, 1981; and Mulherjee *et al.*, 1990): Connectivity preservation is also called *topology preservation* (Ma, 1994; Saha *et al.*, 1994). It implies that any possible input image and its output image should have the same connections among components in the skeleton. For example, the resulting skeleton has the same number of foreground objects, background objects, and holes as the original image. No connected object in an image can be split or deleted, and no hole in an object can be spuriously created or eliminated.
2. *Thinness* (Pudney, 1998): The unit width of the thinned result should be guaranteed for any possible input image except at points where a thick skeleton is necessary to preserve connectivity.
3. *Geometry preservation* (Ma and Sonka, 1996; Lee and Kashyap, 1981): There is no precise definition of geometry preservation. Roughly speaking, the resulting skeleton of the algorithm should be located in the center of the object and "appear similar to" the original object except that it will be thinned. For example, an object in the shape of the letter "b" should not be converted into an object in the shape of the letter "o", and an object in the shape of the character "▌" should not be converted into an object in the shape of the character "〈" or "ǂ."

[1] AutoQuant Imaging, Inc., 877 25th St., Watervliet, New York 12189.
[2] Rensselaer Polytechnic Institute, Troy, New York 12180.
[3] Brown University, Providence, Rhode Island.

Table I. The Definitions of Terms

Term	Definition
Point	Coordinate or location in the 3D image marking the center of a voxel
p	Notation of a point
q	Notation of a point
P	Vector (or array) denoting the 3D image. The elements in this vector are the locations (points) of the voxels and their binary values
6-adjacent	See Eq. (1)
18-adjacent	See Eq. (2)
26-adjacent	See Eq. (3)
$N_k(p)$	The set of all the points that are k-adjacent to p and include p
X	A set of points in the 3D image P
k-path	A sequence of distinct points in X in which every two adjacent points are k-adjacent
k-connected	Two points in X are k-connected if there exists a k-path in X joining them
k-adjacent to a set of points	A point is said to be k-adjacent to a given set of points if it is k-adjacent to at least one of the points in the given set of points
object point	A point with value 1
Background point	A point with value 0
k-connected subset in X	A set of points in X in which all points are k-connected to each other
An object O within X	An object O, contained within X, is a 26-connected subset for which: (1) All points within O are object point. (2) For all object points contained within X, there are no such points that are both 26-adjacent to O and not contained within O. In other words, every object point that is 26-adjacent to any point in O must be also be contained within O
A background component B within X	A background component B, contained within X, is a 6-connected subset for which: (1) All points within B are background point. (2) For all background points contained within X, there are no such points that are both 6-adjacent to B and not contained within B. In other words every background point that is 6-adjacent to any point in B must be also be contained within B
Simple point (Malandian *et al.*, 1992)	A object point p in a 3D image that: (1) p is 26-adjacent to only one object in $N_{26}(p) - \{p\}$. (2) p is 6-adjacent to only one background component in $N_{18}(p)$
$s(p)$: South neighbor of p	If $p = (p_x, p_y, p_z)$, then the point $s(p) = (p_x - 1, p_y, p_z)$ is called the south neighbor of p
$n(p)$: North neighbor of p	If $p = (p_x, p_y, p_z)$, then the point $n(p) = (p_x + 1, p_y, p_z)$ is called the north neighbor of p
$w(p)$: West neighbor of p	If $p = (p_x, p_y, p_z)$, then the point $w(p) = (p_x, p_y - 1, p_z)$ is called the west neighbor of p
$e(p)$: East neighbor of p	If $p = (p_x, p_y, p_z)$, then the point $e(p) = (p_x, p_y + 1, p_z)$ is called the east neighbor of p
$u(p)$: Up neighbor of p	If $p = (p_x, p_y, p_z)$, then the point $u(p) = (p_x, p_y, p_z - 1)$ is called the up neighbor of p
$d(p)$: Down neighbor of p	If $p = (p_x, p_y, p_z)$, then the point $d(p) = (p_x, p_y, p_z + 1)$ is called the down neighbor of p
East border	An object point is called an east border if its east neighbor is a background point
West border	An object point is called a west border if its west neighbor is a background point
South border	An object point is called a south border if its south neighbor is a background point
North border	An object point is called a north border if its north neighbor is a background point
Up border	An object point is called an up border if its up neighbor is a background point
Down border	An object point is called a down border if its down neighbor is a background point
Border	An object point is called a border if it is either an east border, west border, south border, north border, up border or down border
Border pair	There are 12 types of border pairs, which are north–west, north–east, north–up, north–down, south–up, south–down, south–east, south–west, west–up, west–down, east–up, and east–down border pair, respectively. An object point is called a north–west border pair if it is both north and west border. The other types of border pair can be defined similarly
Interior point	An object point that is not adjacent to any background point
Exterior point	An object point that is adjacent to one or more background points