A Subiteration-Based Surface-Thinning Algorithm with a Period of Three

Kálmán Palágyi

Department of Image Processing and Computer Graphics,
University of Szeged, Hungary
palagyi@inf.u-szeged.hu

Abstract. Thinning on binary images is an iterative layer by layer erosion until only the "skeletons" of the objects are left. This paper presents an efficient parallel 3D surface–thinning algorithm. A three–subiteration strategy is proposed: the thinning operation is changed from iteration to iteration with a period of three according to the three deletion directions.

1 Introduction

Skeleton is a region–based shape feature that is extracted from binary image data. A very illustrative definition of the skeleton is given using the prairie–fire analogy: the object boundary is set on fire and the skeleton is formed by the loci where the fire fronts meet and quench each other [4]. In discrete spaces, the thinning process is a frequently used method for producing an approximation to the skeleton in a topology–preserving way [7]. It is based on digital simulation of the fire front propagation: border points of a binary object that satisfy certain topological and geometric constraints are deleted in iteration steps. The entire process is repeated until only the "skeleton" is left.

A simple point is a point whose deletion (or addition) does not alter the topology of the picture [10]. Sequential thinning algorithms delete simple points which are not end–points, since preserving end–points provides important information relative to the shape of the objects. Curve thinning (i.e. a thinning process for extracting medial line) preserves line end–points while surface thinning (i.e. a thinning process for extracting medial surface) does not delete surface end–points.

Parallel thinning algorithms delete a set of simple points simultaneously. A possible approach to preserve topology is to use subiteration–based approach [6]: the thinning operation is changed from iteration to iteration with a period of n ($n \geq 2$); each iteration of a period is then called a subiteration where only border points of certain kind can be deleted. Since there are six kinds of major directions in 3D pictures, 6–subiteration thinning algorithms were generally proposed [3,5,8,9,12,13,18,19]. Note, that 3–, 8–, and 12–subiteration algorithms were also developed [14,15,16].

In this paper, a non–conventional 3–subiteration surface thinning algorithm is proposed. Some experiments are made on synthetic objects and it is demonstrated that the new algorithm is computationally efficient.

F.A. Hamprecht, C. Schnörr, and B. Jähne (Eds.): DAGM 2007, LNCS 4713, pp. 294–303, 2007.

2 Basic Notions

Let p be a point in the 3D digital space $\mathbb{Z}^3$. Let us denote $N_j(p)$ (for $j = 6, 18, 26$) the set of points *j–adjacent* to point p (see Fig. 1a). The sequence of distinct points $\langle x_0, x_1, \ldots, x_n\rangle$ is a *j–path* of length $n \geq 0$ from point x_0 to point x_n in a non–empty set of points X if each point of the sequence is in X and x_i is j–adjacent to x_{i-1} for each $1 \leq i \leq n$. (Note that a single point is a j–path of length 0.) Two points are *j–connected* in the set X if there is a j–path in X between them. A set of points X is *j–connected* in the set of points $Y \supseteq X$ if any two points in X are j–connected in Y.

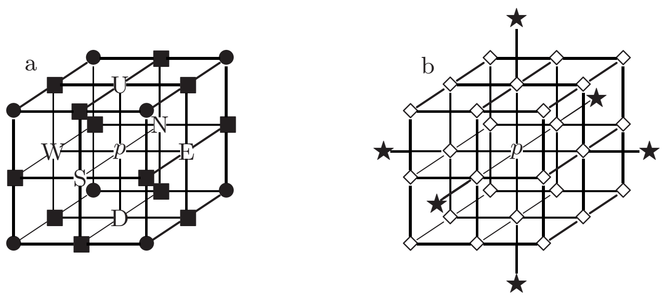


Fig. 1. The frequently used adjacencies in $\mathbb{Z}^3$ (a). The set $N_6(p)$ of the central point $p \in \mathbb{Z}^3$ contains the central point p and the 6 points marked U=$u(p)$, N=$n(p)$, E=$e(p)$, S=$s(p)$, W=$w(p)$, and D=$d(p)$. The set $N_{18}(p)$ contains the set $N_6(p)$ and the 12 points marked "■". The set $N_{26}(p)$ contains the set $N_{18}(p)$ and the 8 points marked "●". The special local neighbourhood of the proposed algorithm (b). The new value of a black point p depends on $N_{26}(p)$ (marked "◇") and six additional points (marked "★").

The *3D binary (m,n) digital picture* P is a quadruple $P = (\mathbb{Z}^3, m, n, B)$ [7]. Each element of $\mathbb{Z}^3$ is called a *point* of P. Each point in $B \subseteq \mathbb{Z}^3$ is called a *black point* and value 1 is assigned to it. Each point in $\mathbb{Z}^3 \backslash B$ is called a *white point* and value 0 is assigned to it. Adjacency m belongs to the black points and adjacency n belongs to the white points. A *black component* (or *object*) is a maximal m–connected set of points in B. A *white component* is a maximal n–connected set of points in $B \subseteq \mathbb{Z}^3$.

We are dealing with $(26, 6)$ pictures. It is assumed that any picture contains finitely many black points.

A black point is called *border point* in $(26, 6)$ pictures if it is 6–adjacent to at least one white point. A border point p is called **U**–*border point* if the point marked by **U**=$u(p)$ in Fig. 1a is white. We can define **N**–, **E**–, **S**–, **W**–, and **D**–border points in the same way. A black point p is called *interior point* if it is not border point (i.e. $u(p)$, $n(p)$, $e(p)$, $s(p)$, $w(p)$, and $d(p)$ are all black points). A black point is called *simple point* if its deletion does not alter the topology of the picture [7].

We propose a new surface thinning algorithm for extracting medial surfaces from 3D $(26, 6)$ pictures. The deletable points of the algorithm are border points of certain types and not *surface end–point*s (i.e. which are not extremities of surfaces). The proposed algorithm uses the following characterization of the surface end–points: a black point is *surface end–point* in a picture if it is border point and it is not 6-adjacent to any interior point. Note, that the same characterization has been used by other authors [1,11].

3 The New Thinning Algorithm

Each conventional 6–subiteration 3D thinning algorithm uses the six deletion directions that can delete certain **U**–, **D**–, **N**–, **E**–, **S**–, and **W**–border points, respectively [3,5,8,9,12,13,18,19]. In our 3–subiteration approach, two kinds of border points can be deleted in each subiteration. The three deletion directions correspond to the three kinds of opposite pairs of points, and are denoted by **UD**, **NS**, and **EW**. The first subiteration assigned to the deletion direction **UD** can delete certain **U**– or **D**–border points; the second subiteration associated with the deletion direction **NS** attempt to delete **N**– or **S**–border points, and some **E**– or **W**–border points can be deleted by the third subiteration corresponding to the deletion direction **EW**. The proposed algorithm is given as follows:

Input: picture $P = (\mathbb{Z}^3, 26, 6, B)$
Output: picture $P' = (\mathbb{Z}^3, 26, 6, B')$

`3-subiteration_thinning(`B`,`B'`)`
begin
  $B' = B$;
  repeat
    $B' =$ `deletion_from_UD(`B'`)`; /* *1st subiteration* */
    $B' =$ `deletion_from_NS(`B'`)`; /* *2nd subiteration* */
    $B' =$ `deletion_from_EW(`B'`)`; /* *3rd subiteration* */
  until *no points are deleted*;
end.

The new value of a black point depends on the values of $26{+}6 = 32$ additional points. The considered special neighbourhood is presented in Fig. 1b.

Deletable points in a subiteration are given by a set of matching templates. A black point is deletable if at least one template in the corresponding set of templates matches it.

The deletion rule corresponding to the first subiteration is given by the set of templates $T_{\mathbf{UD}}$ (see Fig. 2). Note that Fig. 2 shows only the ten base templates **U1**–**U5**, **D1**–**D5**. Additionally, all their rotations around the vertical axis belong to $T_{\mathbf{UD}}$, where the rotation angles are 90°, 180°, and 270°.

It is easy to see that the complete $T_{\mathbf{UD}}$ contains $2 \cdot (1 + 4 + 4 + 2 + 4) = 30$ templates. This set of templates was constructed for deleting some simple points which are neither surface end–points nor extremities of surfaces. The deletable

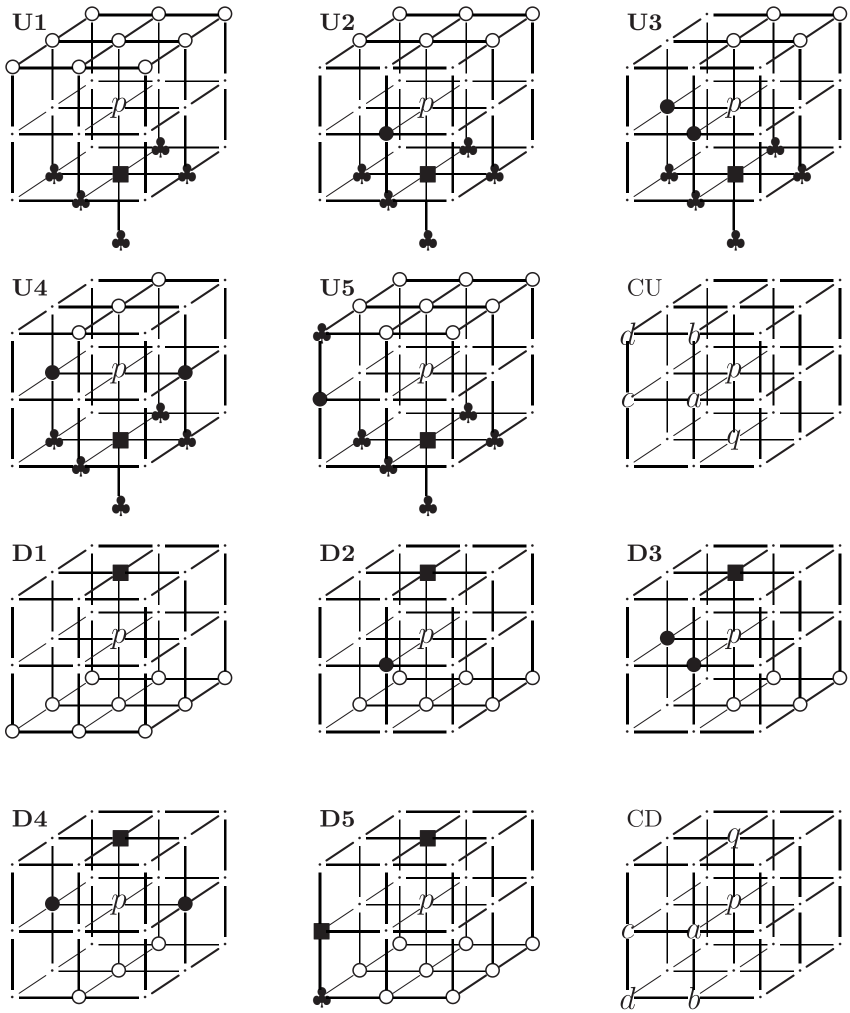


Fig. 2. Base templates **U1**–**U5**, **D1**–**D5** and their rotations around the vertical axis form the set of templates $T_{\mathbf{UD}}$ assigned to the deletion direction **UD**. Note, that a point p deleted by templates **D1**–**D5** and their rotated version must be 6–adjacent to at least one interior point.

Notations: each position marked "p", "●", "■", and "♣" matches a black point; each position marked "○" matches a white point; each "·" ("don't care") matches either a black or a white point.

Configurations "CU" and "CD", and using different symbols for black template positions help us to prove the topological correctness of the algorithm.

points of the other two subiterations (corresponding to deletion directions **NS** and **EW**) can be obtained by proper rotations of the templates in $T_{\mathbf{UD}}$. Note that choosing another order of the deletion directions yields another algorithm. The proposed algorithm terminates when there are no more black points to be deleted. Since all considered input pictures are finite, it will terminate.

Although the proposed algorithm may seem complicated, in fact it can be simply implemented and it runs efficiently. We can state that a border point is to be deleted from deletion direction **UD** if:

(($d(p)$ is interior point *and* $u(p)$ is white) *or*
($u(p)$ is black *and* p is 6–adjacent to interior point *and* $d(p)$ is white)
) *and* $f(x_0, x_1, \ldots, x_{24}) = 1$,

where f is a Boolean–function of 25 variables derived from the set of templates $T_{\mathbf{UD}}$. It is easy to see, that function f can be given by a pre-calculated 2^{25} bit $\equiv 4$ Mbyte (unit time access) look-up-table. The considered 25 variables correspond to 25 points in $N_{26}(p)$ (see Fig. 3). More details concerning the implementation of 3D thinning algorithms are presented in [17].

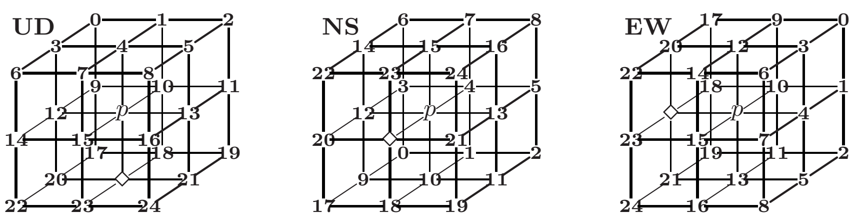


Fig. 3. Indices of the 25 Boolean variables (i.e. the considered points in $N_{26}(p)$). Note, that investigating the point marked "◇" is not needed. Since the deletion rule of a subiteration can be derived from the deletion rule of the reference subiteration **UD** by the proper rotation, the indexing scheme of a subiteration corresponds to the proper permutation of positions assigned to the reference subiteration.

4 Discussion

Thinning algorithms have to take care of the following four aspects:

1. forcing the "skeleton" to retain the topology of the original object (i.e. topology has to be preserved);
2. providing "shape preservation" (i.e. significant features of the original object are to be produced);
3. forcing the "skeleton" to be in its geometrically correct position (i.e. in the "middle" of the object);
4. producing "maximal" thinning (i.e. the desired "width" of the "skeleton" is one point).

The topological correctness (the 1st requirement) of the proposed algorithm is shown in Section 5.

Shape preservation (the 2nd requirement) is a fairly important requirement, too. For example, an object having same shape as letter "b" cannot be thinned to a circular shape. The aim of the thinning is not to produce the topological kernel [2] of an object: the thinning differs from shrinking. That is the reason why end–point criteria are used in thinning. It is easy to see that surface–end points are removed by none of our templates (see Fig. 2).

Geometrical correctness (the 3rd requirement) of the extracted skeleton is mostly achieved by the subiteration (multi–directional) thinning approach. An object is to be shrunk uniformly from each direction.

It is rather difficult to prove that the 4th requirement about maximal thinning is satisfied. Due to the used surface end–point criterion, the produced skeleton may contain 2–point thick surface patches [1,11]. It is easy to overcome this problem (e.g., by applying the final thinning step proposed by Arcelli et al. [1]).

Our algorithm has been tested on objects of different shapes. Here we present five examples (see Figs. 4–5).

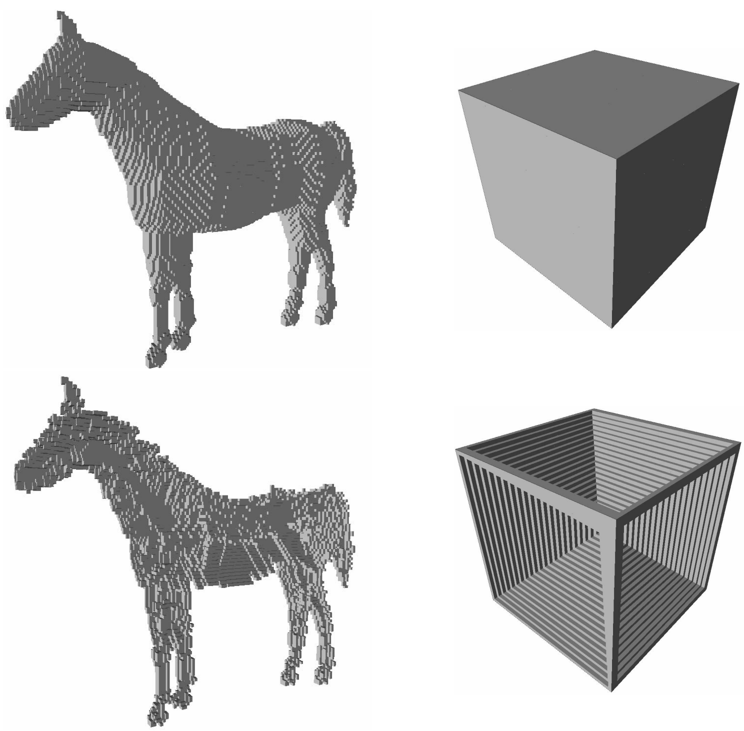


Fig. 4. Two synthetic pictures containing a $140 \times 140 \times 50$ horse and a $45 \times 45 \times 45$ cube (top); and their skeletons produced by the proposed surface–thinning algorithm (bottom)

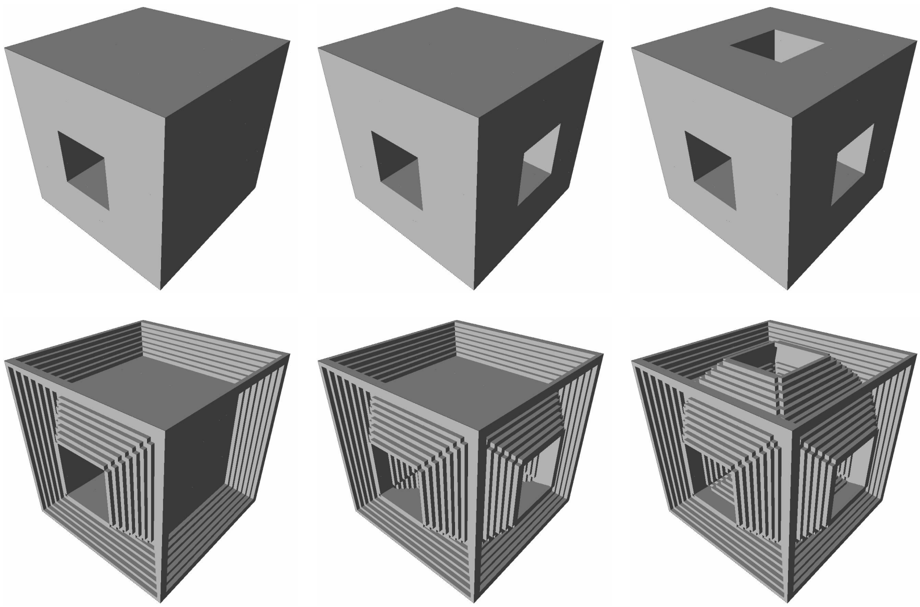
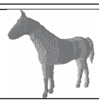


Fig. 5. Three synthetic pictures containing a $45 \times 45 \times 45$ cube with one, two, and three hole(s), respectively (top); and their skeletons produced by the proposed surface–thinning algorithm (bottom)

The computation time of a thinning process depends on the complexity of an iteration step and the required number of iteration steps. The 3–subiteration 3D thinning strategy has been compared with other subiteration–based approaches with periods of 6, 8, or 12. It has been shown that the 3–subiteration approach requires the least number of iterations [16]. If we use unit time access look-up-tables (corresponding the deletion rules of the considered algorithms) and our efficient implemetation method [17] is applied, then 3–subiteration algorithms are the fastest subiteration–based ones. The efficiency of the proposed method is illustrated in Table 1.

Note, that the new algorithm differs greatly from the existing 3–subiteration surface–thinning algorithm [16] in its surface end-point characterization and deletion rule. While in the earlier work a black point p is a surface end-point if $u(p) = d(p) = 0$ or $n(p) = s(p) = 0$ or $e(p) = w(p) = 0$ (see Fig. 1a), in the new algorithm a black point is surface end–point if it is border point and it is not 6-adjacent to any interior point. In addition, the set of matching templates corresponding to a deletion rule of the earlier 3–subiteration surface–thinning algorithm contains only 26 templates [16] in contrast to the 30, used in this work. Consequently, the new and the earlier algorithms produce significantly different medial surfaces.

Table 1. Computation times for the considered five kinds of test pictures. The implemented surface–thinning algorithm was run under Linux on an Intel Pentium 4 CPU 2.80 GHz PC. (Note, that only the thinning itself was considered; reading the input volume and the 4 MB look-up-table, and writing the output image were not taken into account.).

test picture	size	number of object points	running time (sec.)
	$140 \times 140 \times 50$	92 534	0.146
	$45 \times 45 \times 45$	91 125	0.074
	$93 \times 93 \times 93$	804 357	0.377
	$141 \times 141 \times 141$	2 803 221	1.465
	$45 \times 45 \times 45$	81 000	0.033
	$93 \times 93 \times 93$	714 984	0.405
	$141 \times 141 \times 141$	2 491 752	1.493
	$45 \times 45 \times 45$	74 250	0.028
	$93 \times 93 \times 93$	655 402	0.343
	$141 \times 141 \times 141$	2 284 106	1.389
	$45 \times 45 \times 45$	67 500	0.029
	$93 \times 93 \times 93$	595 820	0.393
	$141 \times 141 \times 141$	2 076 460	1.271

5 Verification

The proposed 3–subiteration thinning algorithm is topology preserving for $(26, 6)$ pictures [7]. It is sufficient to prove that reduction operation given by the set of templates T_{UD} is topology preserving. If the first subiteration of the algorithm is topology preserving, then the other two ones are topology preserving as well, since rotation of the deletion templates do not alter their topological properties. Therefore, the proposed algorithm is topology preserving, since it is composed of topology preserving reductions.

We make use of the following result for $(26, 6)$ pictures:

Theorem 1. [10] *Black point p is simple in picture $(\mathbb{Z}^3, 26, 6, B)$ if and only if all of the following conditions hold:*

1. *the set $(B \backslash \{p\}) \cap N_{26}(p)$ contains exactly one 26–component; and*
2. *the set $(\mathbb{Z}^3 \backslash B) \cap N_6(p)$ is not empty and it is 6–connected in the set $(\mathbb{Z}^3 \backslash B) \cap N_{18}(p)$.*

Theorem 1 shows that the simplicity in $(26, 6)$ pictures is a local property; it can be decided in view of the $3 \times 3 \times 3$ neighbourhood of a given point.

We need to consider what is meant by topology preservation when a number of black points are deleted simultaneously. We use the following sufficient conditions for parallel reduction operations:

Theorem 2. [14] *Let F be a parallel reduction operation on $(26, 6)$ pictures. Then F is topology preserving, if for all pictures $P = (\mathbb{Z}^3, 26, 6, B)$, all of the following conditions hold:*

1. *for all points $p \in B$ that are deleted by F and for all sets $Q \subseteq (N_{18}(p)\backslash\{p\}) \cap B$ that are deleted by F, p is simple in the picture $(\mathbb{Z}^3, 26, 6, B\backslash Q)$; and*
2. *no black component contained entirely in a $2 \times 2 \times 2$ configuration in $\mathbb{Z}^3$ can be deleted completely by F.*

Unfortunately, there is no room to present the detailed proof concerning the topological correctness. Our proof is based on the following properties of the deletion rule of the first subiteration given by the set of templates $T_{\mathbf{UD}}$ (see Fig. 2):

1. Each template in $T_{\mathbf{UD}}$ deletes only simple points.
2. The simplicity of a deletable point p does not depend on the points that coincide with a template position marked "♣" and "·".
3. Black points that coincide with template positions marked "■" cannot be deleted by any template in $T_{\mathbf{UD}}$.
4. Let us investigate the configuration "CU" (and its rotations around the vertical axis) and assume that central point p is black and it can be deleted by a template in **U1**–**U5** (or their rotations). Then the followings hold:
   - If point q is black, then it cannot be deleted by any template in $T_{\mathbf{UD}}$.
   - If point a is black and it can be deleted by a template in $T_{\mathbf{UD}}$, then point b is white.
   - If points a and c are black and they can be deleted by a template in $T_{\mathbf{UD}}$, then point d is white.
5. Let us investigate the configuration "CD" (and its rotations around the vertical axis) and assume that central point p is black and it can be deleted by a template in **D1**–**D5** (or their rotations). Then the followings hold:
   - If point q is black, then it cannot be deleted by any template in $T_{\mathbf{UD}}$.
   - If point a is black and it can be deleted by a template in $T_{\mathbf{UD}}$, then point b is white.
   - If points a and c are black and they can be deleted by a template in $T_{\mathbf{UD}}$, then point d is white.
6. If a black point p can be deleted by a template in $T_{\mathbf{UD}}$, then p must be 6–adjacent at least one interior point. Hence p cannot be in a small object contained entirely in a $2 \times 2 \times 2$ configuration in $\mathbb{Z}^3$.

Condition 1 of Theorem 2 can be seen with the help of properties 1–5. Condition 2 of Theorem 2 is obvious by property 6. Therefore, the proposed algorithm is topology preserving for $(26, 6)$ pictures.

Acknowledgements

The author is grateful to Stina Svensson (Centre for Image Analysis, Swedish University of Agricultural Sciences, Uppsala, Sweden) for supplying the horse image data (see Fig. 4).

References

1. Arcelli, C., Sanniti di Baja, G., Serino, L.: New removal operators for surface skeletonization. In: Kuba, A., Nyúl, L.G., Palágyi, K. (eds.) DGCI 2006. LNCS, vol. 4245, pp. 555–566. Springer, Heidelberg (2006)
2. Bertrand, G., Aktouf, Z.: A 3D thinning algorithms using subfields. In: Proc. SPIE Conf. on Vision Geometry III, vol. 2356, pp. 113–124 (1994)
3. Bertrand, G.: A parallel thinning algorithm for medial surfaces. Pattern Recognition Letters 16, 979–986 (1995)
4. Blum, H.: A transformation for extracting new descriptors of shape. In: Models for the Perception of Speech and Visual Form, pp. 362–380. MIT Press, Cambridge (1967)
5. Gong, W.X., Bertrand, G.: A simple parallel 3D thinning algorithm. In: Proc. 10th Int. Conf. on Pattern Recognition, pp. 188–190 (1990)
6. Hall, R.W.: Parallel connectivity–preserving thinning algorithms. In: Kong, T.Y., Rosenfeld, A. (eds.) Topological algorithms for digital image processing, pp. 145–179. Elsevier Science, Amsterdam (1996)
7. Kong, T.Y., Rosenfeld, A.: Digital topology: Introduction and survey. Computer Vision, Graphics, and Image Processing 48, 357–393 (1989)
8. Lee, T., Kashyap, R.L., Chu, C.: Building skeleton models via 3–D medial surface/axis thinning algorithms. CVGIP: Graphical Models and Image Processing 56, 462–478 (1994)
9. Lohou, C., Bertrand, G.: A 3D 6-subiteration curve thinning algorithm based on P-simple points. Discrete Applied Mathematics 151, 198–228 (2005)
10. Malandain, G., Bertrand, G.: Fast characterization of 3D simple points. In: Proc. 11th IEEE Internat. Conf. on Pattern Recognition, pp. 232–235 (1992)
11. Manzanera, A., Bernard, T.M., Pretêux, F., Longuet, B.: Medial faces from a concise 3D thinning algorithm. In: Proc. 7th IEEE Internat. Conf. Computer Vision, ICCV'99, pp. 337–343 (1999)
12. Mukherjee, J., Das, P.P., Chatterjee, B.N.: On connectivity issues of ESPTA. Pattern Recognition Letters 11, 643–648 (1990)
13. Palágyi, K., Kuba, A.: A 3D 6–subiteration thinning algorithm for extracting medial lines. Pattern Recognition Letters 19, 613–627 (1998)
14. Palágyi, K., Kuba, A.: Directional 3D thinning using 8 subiterations. In: Bertrand, G., Couprie, M., Perroton, L. (eds.) DGCI 1999. LNCS, vol. 1568, pp. 325–336. Springer, Heidelberg (1999)
15. Palágyi, K., Kuba, A.: A parallel 3D 12–subiteration thinning algorithm. Graphical Models and Image Processing 61, 199–221 (1999)
16. Palágyi, K.: A 3-subiteration 3D thinning algorithm for extracting medial surfaces. Pattern Recognition Letters 23, 663–675 (2002)
17. Palágyi, K.: Efficient implementation of 3D thinning algorithms. In: Proc. 6th Conf. Hungarian Association for Image Processing and Pattern Recognition, pp. 266–274 (2007)
18. Tsao, Y.F., Fu, K.S.: A parallel thinning algorithm for 3–D pictures. Computer Graphics and Image Processing 17, 315–331 (1981)
19. Xie, W., Thompson, R.P., Perucchio, R.: A topology-preserving parallel 3D thinning algorithm for extracting the curve skeleton. Pattern Recognition 36, 1529–1544 (2003)