Computer Science Department of The University of Auckland
CITR at Tamaki Campus (http://www.citr.auckland.ac.nz/)

CITR-TR-124 January 2003

A Comparative Discussion of Distance Transforms and Simple Deformations in Image Processing

Gisela Klette[1]

Abstract

Algorithms for transformations of digital images into reconstructible subsets of the original image, and algorithms for deformations of digital images into topologically equivalent images are subjects of hundreds of publications. Two images are topologically equivalent if their adjacency trees are isomorphic. Skeletonization is a transformation of components of a digital image into a subset of the original component. There are different categories of skeletonization methods: one category is based on distance transforms, and a specified subset of the transformed image is called distance skeleton. The original component can be reconstructed from the distance skeleton. But the result is not a topologically equivalent image. A different category is defined by thinning approaches, and the result of skeletonization using thinning algorithms should be a topologically equivalent image. Thinning algorithms are one-way simple deformations, which mean object points are transferred into background points without destroying the topology of the image. Two-way simple deformations transfer object points into background points and vice versa without destroying the topology of the image. This report reviews contributions in this area with respect to properties of algorithms and characterizations of simple points, and informs about a few new results.

[1] Center for Image Technology and Robotics Tamaki Campus, The University of Auckland, Auckland, New Zealand.

You are granted permission for the non-commercial reproduction, distribution, display, and performance of this technical report in any format, BUT this permission is only for a period of 45 (forty-five) days from the most recent time that you verified that this technical report is still available from the CITR Tamaki web site under terms that include this permission. All other rights are reserved by the author(s).

A Comparative Discussion of Distance Transforms and Simple Deformations in Digital Image Processing

Gisela Klette
CITR Tamaki, University of Auckland
Tamaki Campus, Building 731, Auckland, New Zealand

Abstract

Algorithms for transformations of digital images into reconstructible subsets of the original image, and algorithms for deformations of digital images into topologically equivalent images are subjects of hundreds of publications. Two images are topologically equivalent if their adjacency trees are isomorphic. Skeletonization is a transformation of components of a digital image into a subset of the original component. There are different categories of skeletonization methods: one category is based on distance transforms, and a specified subset of the transformed image is called distance skeleton. *The original component can be reconstructed from the distance skeleton. But the result is not a topologically equivalent image. A different category is defined by thinning approaches, and the result of skeletonization using thinning algorithms should be a topologically equivalent image. Thinning algorithms are one-way simple deformations, which mean object points are transferred into background points without destroying the topology of the image. Two-way simple deformations transfer object points into background points and vice versa without destroying the topology of the image. This report reviews contributions in this area with respect to properties of algorithms and characterizations of simple points, and informs about a few new results.*

Keywords: shape simplification, skeletonization, thinning, distance transform.

1 Introduction

Motivations for the interest in skeletonization algorithms are either requirements to reduce the amount of data, or the interest in simplifying the shape of an object in order to find features for recognition algorithms and classifications. Additionally the transformation of a component into an image showing essential characteristics can eliminate local noise at object frontiers. Thinning algorithms deliver topologically equivalent images with a reduced or equal number of object points and magnification algorithms deliver topologically equivalent images with a reduced or equal number of non-object points.

Many operators have been proposed for presenting connected components in a digital image by simplified shapes. In general we have to state that the development, choice and modification of such algorithms in practical applications are domain and task dependent, and there is no “best method”. However, it is interesting to note that there are several equivalences between published methods and notions, and characterizing such equivalences or differences is useful to categorize the broad diversity of published methods for skeletonization. A main intention of this report is the discussion of experimental results and theoretical equivalences.

1.1 Categories of Methods

One class of shape reduction operators is based on distance transforms. A *distance skeleton* is a subset of points of a given component such that every point of this subset represents the center of a maximal disc (labeled with the radius of this disc) contained in the given component. This report discusses methods for calculating a distance skeleton using three different distance functions.

A second class of operators produces median or center lines of the digital object in a non-iterative way. Normally such operators locate critical points first, and calculate a specified path through the object by connecting these points.

The third class of operators is characterized by iterative thinning. Historically, Listing [13] used already in 1862 the term *linear skeleton* for the result of a continuous deformation of the frontier of a connected subset of a Euclidean space without changing the connectivity of the original set, until only a set of lines and points remains. Many algorithms in image analysis are based on this *general concept of thinning*. The goal is a calculation of characteristic properties of digital objects which are not related to size or quantity. Methods should be independent from the position of a set in the plane or space, grid resolution (for digitizing this set) or the shape complexity of the given set. In the literature the term "thinning" is not used in a unique interpretation besides that it always denotes a connectivity preserving reduction operation applied to digital images, involving iterations of transformations of specified contour points into background points. A subset $Q \subseteq I$ of object points is reduced by a defined set D in one iteration, and the result $Q' = Q \setminus D$ becomes Q for the next iteration.

Topology-preserving skeletonization is a special case of thinning resulting in a connected set of digital arcs or curves. A *digital curve* is a path $p = p_0, p_1, p_2, ..., p_n = q$ such that p_i is a neighbor of p_{i-1}, $1 \leq i \leq n$, and $p = q$. A digital curve is called *simple* if each point p_i has exactly two neighbors in this curve. A *digital arc* is a subset of a digital curve such that $p \neq q$. A point of a digital arc which has exactly one neighbor is called an *end point* of this arc.

Within this third class of operators (thinning algorithms) we may classify with respect to algorithmic strategies: individual pixels are either removed in a sequential order or in parallel. A local operation on a pixel p of an image I is called sequential if the arguments of the new value $I^*(p)$ are the new values of the already processed neighbors and the original values of p and the succeeding neighbors in a defined raster sequence. For example, the often cited algorithm by Hilditch [6] is an iterative process of testing and deleting contour pixels sequentially in standard raster scan order. Another sequential algorithm [15] uses the definition of *multiple points* and proceeds by contour following. The result is a connected set of a reduced number of object points. The algorithm is an example for topology-preserving thinning but it does not deliver a skeleton in general.

In parallel operations the arguments are the original pixel values in a defined neighbourhood and the results $I^*(p)$ are stored but they are not used until the operation has been performed for all pixels of the image. Differences between parallel algorithms are typically defined by tests implemented to ensure connectedness in a local neighbourhood and by different numbers of sub-cycles.

The notion of a *simple pixel* is of basic importance for simple deformations and it will be shown in this report that different definitions of simple pixels are actually equivalent. It is not enough to characterize single pixels in order to preserve adjacency relations because during one iteration a whole set of pixels change the values. It is necessary to examine specified sets of pixels. Several publications characterize properties of a set D of pixels (to be turned from object points to background points or vice versa) to ensure that adjacency relations of object and background remain unchanged. The report discusses some of these properties in order to justify parallel thinning algorithms.

1.2 Basics

The used notation follows [23]. A *digital image* I is a function defined on a discrete set C, which is called the carrier of the image. The elements of C are grid points or grid cells, and the elements $(p, I(p))$ of an image are pixels (2D case) or voxels (3D case). The range of a (scalar) image is $\{0, ...G_{max}\}$ with $G_{max} \geq 1$. The range of a binary image is $\{0, 1\}$. We only use binary images I in this report. Let $\langle I \rangle$ be the set of all pixel locations with value 1, i.e. $\langle I \rangle = I^{-1}(1)$.

The image carrier is defined on an orthogonal grid in 2D or 3D space. There are two options: using the grid cell model a 2D pixel location p is a closed square (2-cell) in the Euclidean plane and a 3D pixel location is a closed cube (3-cell) in the Euclidean space, where edges are of length 1 and parallel to the coordinate axes, and centers have integer coordinates. As a second option, using the grid point model a 2D or 3D pixel location is a grid point.

Two pixel locations p and q in the grid cell model are called *0-adjacent* iff $p \neq q$ and they share at least one vertex (which is a 0-cell). Note that this specifies 8-adjacency in 2D or 26-adjacency in 3D if the grid point model is used. Two pixel locations p and q in the grid cell model are called *1-adjacent* iff $p \neq q$ and they share at least one edge (which is a 1-cell). Note that this specifies 4-adjacency in 2D or 18-adjacency in 3D if the grid point model is used. Finally, two 3D pixel locations p and q in the grid cell model are called *2-adjacent* iff $p \neq q$ and they share at least one face (which is a 2-cell). Note that this specifies 6-adjacency if the grid point model is used. Any of these adjacency relations A_α, $\alpha \in \{0, 1, 2, 4, 6, 18, 26\}$, is irreflexive and symmetric on an image carrier C. The *α-neighborhood* $N_\alpha(p)$ of a pixel location p includes p and its α-adjacent pixel locations.

Coordinates of 2D grid points are denoted by (i, j), with $1 \leq i \leq n$ and $1 \leq j \leq m$; i, j are integers and n, m are the numbers of rows and columns of C. In 3D we use integer coordinates (i, j, k).

Based on neighborhood relations we define connectedness as usual: two points $p, q \in C$ are *α-connected* with respect to $M \subseteq C$ and neighborhood relation N_α iff there is a sequence of points $p = p_0, p_1, p_2, ..., p_n = q$ such that p_i is an α-neighbor of p_{i-1}, for $1 \leq i \leq n$, and all points on this sequence are either in M or all in the complement of M ($\overline{M}$). A subset $M \subseteq C$ of an image carrier is called *α-connected* iff M is not empty and all points in M are pairwise α-connected with respect to set M. An *α-component* of a subset S of C is a maximal α-connected subset of S. The study of connectivity in digital images has been introduced in [18]. It is standard practice to use different types of connectness for pixels $p \in \langle I \rangle$ and for pixels $p \in \langle \overline{I} \rangle$. For brevity, we call them 1's and 0's. We use α-connectivity for the 1's and α'-connectivity for the 0's where $(\alpha, \alpha') = (4, 8)$ or $(8, 4)$. The α-components of 1's and the α'-components of 0's are the *regions*. Then the region adjacency graph is a tree [20].

It follows that any set $\langle I \rangle$ consists of a number of α-components. In case of the grid cell model, a component is the union of closed squares (2D case) or closed cubes (3D case). The *frontier* of a 2-cell is the union of its four edges and the frontier of a 3-cell is the union of its six faces.

Let U, V, W be pairwise disjoint sets of pixels. We say that V 4-(8)separates U from W if any 4-(8-)path from a pixel of U to a pixel of W must intersect V (i.e. must contain a pixel of V). The *border* of a set M of 1's (0's) is the set of pixels of M that are α',(α)-adjacent to the complement of M.

For practical purposes it is easy to use neighborhood operations (called *local operations*) on a digital image I which define a value at $p \in C$ in the transformed image based on pixel values in I and coordinates at $p \in C$ and its immediate neighbors in $N_\alpha(p)$.

2 Examples of algorithms for 2D images and results of practical experiments

In this section we only use the grid point model.

2.1 "Distance Skeleton" algorithms

Blum [3] suggested a skeleton representation by a set of symmetric points. In a closed subset M of the Euclidean plane a point $p \in M$ is called *symmetric* iff at least two points exist on the border of $\overline{M}$ with equal distances to p. For every symmetric point, the associated maximal disc is the largest disc in this set. The set of symmetric points, each labeled with the radius of the associated maximal disc, constitutes the skeleton of the set.

This idea of presenting a component of a digital image as a "distance skeleton" is based on the calculation of a specified distance from each point in a component $M \subseteq C$ to $\overline{M}$. The local maxima of the subset represent a "distance skeleton". In [18] the distance is specified as follows.

Definition 1 *The distance $d(p, q)$ from point p to point q, $p \neq q$, is the smallest positive integer n such that there exists a sequence of distinct grid points $p = p_0$,$p_1, p_2, ..., p_n = q$with p_i is a α-neighbor of p_{i-1}, $1 \leq i \leq n$. If $p = q$ the distance between them is defined to be zero.*

For $\alpha = 4$ the distance $d(p, q)$ is called d_4-distance or Manhattan distance. Let i_p, j_p and i_q, j_q the coordinates of p and q. Then the d_4-distance is defined as follows:

$$d_4(p, q) = |i_p - i_q| + |j_p - j_q|$$

For $\alpha = 8$ the distance $d(p,q)$ is called d_8-distance or chessboard distance and we have:

$$d_8(p,q) = \max\{|i_p - i_q|, |j_p - j_q|\}$$

Both distances have all properties of a metric. The following algorithm uses the d_4-distance. Given is a binary digital image. We transform this image into a new one which represents at each point $p \in \langle I \rangle$ the $d_4(p,q)$-distance to pixels having value zero. The transformation includes two steps. We apply functions f_1 to the image I in standard scan order, producing $I^*(i,j) = f_1(i,j,I(i,j))$, and f_2 in reverse standard scan order, producing $T(i,j) = f_2(i,j,I^*(i,j))$, as follows:

$$\begin{aligned} f_1(i,j,I(i,j)) &= \begin{cases} 0 & \text{if } I(i,j) = 0 \\ min\{I^*(i-1,j)+1, I^*(i,j-1)+1\} & \\ \qquad \text{if } I(i,j) = 1 \text{ and } i \neq 1 \text{ or } j \neq 1 & \\ m+n & \text{otherwise} \end{cases} \\ f_2(i,j,I^*(i,j)) &= min\{I^*(i,j), T(i+1,j)+1, T(i,j+1)+1\} \end{aligned}$$

The resulting image T is the *distance transform image* of I. Note that T is a set $\{[(i,j),T(i,j)] : 1 \leq i \leq n \wedge 1 \leq j \leq m\}$, and let $T^* \subset T$ such that $[(i,j),T(i,j)] \in T^*$ iff none of the four points in $A_4((i,j))$ has a value in T equal to $T(i,j)+1$. For all remaining points (i,j) let $T^*(i,j) = 0$. This image T^* is called *distance skeleton.*

Now we apply functions g_1 to the distance skeleton T^* in standard scan order, producing $T^{**}(i,j) = g_1(i,j,T^*(i,j))$, and g_2 to the result of g_1 in reverse standard scan order, producing $T^{***}(i,j) = g_2(i,j,T^{**}(i,j))$, as follows:

$$\begin{aligned} g_1(i,j,T^*(i,j)) &= \max\{T^*(i,j), T^{**}(i-1,j)-1, T^{**}(i,j-1)-1\} \\ g_2(i,j,T^{**}(i,j)) &= \max\{T^{**}(i,j), T^{***}(i+1,j)-1, T^{***}(i,j+1)-1\} \end{aligned}$$

The result T^{***} is equal to the distance transform image T. Both functions g_1 and g_2 define an operator G, with $G(T^*) = g_2(g_1(T^*)) = T^{***}$, and we have [18]:

Theorem 1 *$G(T^*) = T$, and if T' is any subset of image T (extended to an image by having value 0 in all remaining positions) such that $G(T') = T$, then $T'(i,j) = T^*(i,j)$ at all positions of T^* with non-zero values.*

Informally, the theorem says that the original image is reconstructible from the distance skeleton, and it is the smallest data set needed for such a reconstruction. Obviously, the used distance d_4 differs from the Euclidean metric and the d_4-distance skeleton is not invariant under rotation.

In case we use the d_8-distance function the algorithm is similar, applying functions f_1, f_2, g_1, g_2 as follows:

$$\begin{aligned} f_1(i,j,I(i,j)) &= \begin{cases} 0 & \text{if } I(i,j) = 0 \\ min\{I^*(i-1,j)+1, I^*(i,j-1)+1, I^*(i-1,j-1)+1, I^*(i-1,j+1)+1\} & \\ \qquad \text{if } I(i,j) = 1 \text{ and } i \neq 1 \text{ or } j \neq 1 & \\ m+n & \text{otherwise} \end{cases} \\ f_2(i,j,I^*(i,j)) &= min\{I^*(i,j), T(i+1,j)+1, T(i,j+1)+1, T(i+1,j-1)+1, T(i+1,j+1)+1\} \end{aligned}$$

$$\begin{aligned} g_1(i,j,T^*(i,j)) &= \max\{T^*(i,j), T^{**}(i-1,j)-1, T^{**}(i,j-1)-1, T^{**}(i-1,j-1)-1, T^{**}(i-1,j+1)-1\} \\ g_2(i,j,T^{**}(i,j)) &= \max\{T^{**}(i,j), T^{***}(i+1,j)-1, T^{***}(i,j+1)-1, T^{***}(i+1,j+1)-1, T^{***}(i+1,j-1)-1\} \end{aligned}$$

The Euclidean distance is defined as usual:

$$d_E(p,q) = \sqrt{(i_p - i_q)^2 + (j_p - j_q)^2}$$

The algorithm for the Euclidean distance transform used in the following examples determines in the first step all border pixels of the background and saves the coordinates in a 2-dimensional array. The algorithm calculates the Euclidean distance of each pixel of the component and each border pixel of the background rounded by the ceiling

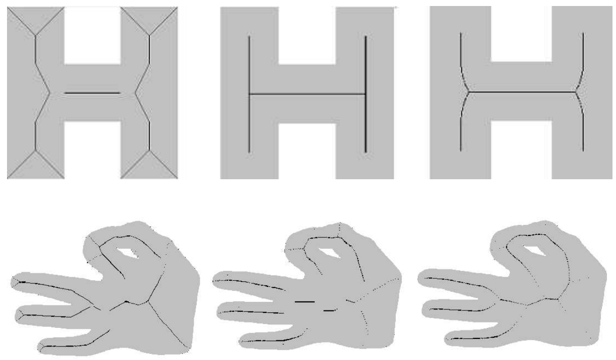


Figure 1. Distance skeletons using d_4-, d_8- and Euclidean-distance

function and the minimum of these rounded Euclidean distances for each pixel. The calculation of the distance skeleton is analogue to the calculations based on the d_4- and d_8-distances.

Examples for calculated distance skeletons are shown in figure 1:The resulting distance skeletons are unconnected subsets of pixels $(p, T(p))$ of the transformed image. They approximate the medial axis of the object. The ratio between the number of identified distance skeleton pixels and number of object pixels for one objects converges to a constant value close to zero with increasing grid resolution in our experiments.

For different approximations of the Euclidean distance, some authors suggested the use of different weights for grid point neighborhoods [4]. Montanari [14] introduced a quasi-Euclidean distance.

Distance skeletons can also represented by morphological operations. Let $A \subseteq I$. Then the distance skeleton can be defined in terms of erosions and openings. It can be shown [25] that

$$S(A) = \bigcup_{k=0}^{K} S_k(A)$$

with

$$S_k(A) = (A \ominus kB) - ((A \ominus kB) \circ B)$$

where B is a structural element and $(A \ominus kB)$ means k successive erosions of A by B. K is the number of the last iterative step before A erodes to an empty set. The structural elements are

$$\begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix} \qquad \begin{matrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{matrix}$$

to determine the coordinates of the D_8- or D_4-distance skeletons. During this process we assign the iteration number k to determined pixel coordinates in order to calculate distance skeletons that are reconstructible.

2.2 Iterative Thinning Algorithms

Thinning is an operation applied to binary images I which preserves the topology of the original image. We use the term "topology preservation" in this paper to specify deformations of images that change the value of simple pixels while preserving the adjacency/surroundedness relations between the connected components of 1's and 0's. For this type of deformations, a bijective mapping exists between the components of the original image and the components of the deformed image and the adjacency relations are the same. Thinning is a one-way simple deformation that reduces the number of object points (in this paper 1's). It is normally implemented by an iterative process of transforming specified contour points into background points. The result of a thinning algorithm might be defined to be *ideally thin* if no point which is not an end point can be deleted without violating connectivity properties. It can happen that a thinning procedure results in digital arcs which intersect each other, generating *branching points*. An ideally thin subset of $\langle I \rangle$ may contain branching points as in the following example:

$$\begin{array}{ccccccc}
1 & & & 1 & & & 1 \\
 & 1 & & 1 & & 1 & \\
 & & 1 & 1 & 1 & & \\
1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 & & 1 & 1 & 1 & & \\
 & 1 & & 1 & & 1 & \\
1 & & & 1 & & & 1
\end{array}$$

Note that this is not a set of digital arcs or curves as defined above. However, these calculated connected components can be split into a set of disjoint digital arcs. The analysis of brunching points is not subject of this report. A thinning algorithm should satisfy the following conditions in order to achieve skeletonization within feasible computation time:

1. The resulting skeleton should be a set of digital arcs or curves, which may have brunching points
2. The resulting subset must approximate the medial axis.
3. End points must be preserved.
4. The topology of the image must be preserved.
5. The algorithm should be computationally efficient.
6. The algorithm should be robust against noise.

The location of the resulting skeleton (condition 2) critically depends upon the sequence of applications of conditions or masks. Algorithms in the literature differ in tests of conditions 3 and 4 and types of applied local operators. The choice of an operator (sequential or parallel, with a different number of sub-cycles) and the choice of connectivity tests have influence on conditions 1 and 2. Connectivity preservation for single points can be precisely defined in different ways and will be discussed separately in this report. Some of the connectivity preserving criterions are equivalent and algorithms differ in ways of implementation and additional tests which are designed to improve the algorithm based on above conditions. These goals are used for comparisons and evaluations of thinning algorithms apart from the fact that they are not exactly defined.

This report presents a few examples of often cited algorithms and lists some properties without extensive comparisons and evaluations.

Definition 2 *A* simple pixel *is a single element p of a digital image with value $I(p)$ that can change this value $I(p)$ without destroying the topology of the image.*

This definition is general. For practical applications it is necessary to determine simple pixels. The following definitions of the *crossing number* of point p will be used for testing a point p to be simple. Let $A_8(p)$ be the 8-adjacency set of p and the elements x_i of this adjacency set are numbered in the following way:

$$\begin{array}{ccc}
x_4 & x_3 & x_2 \\
x_5 & p & x_1 \\
x_6 & x_7 & x_8
\end{array}$$

Definition 3 *[24] The number of transitions from a background point to an object point or vice versa when the points of $A_8(p)$ are traversed in counterclockwise order is called* R-crossing number *$X_R(p)$:*

$$X_R(p) = \sum_{i=1}^{8} |x_{i+1} - x_i| \quad \textit{where } x_9 = x_1 \ .$$

Let $X_A(p)$ be the number of distinct 4-components of object points (1's) in $A_8(p)$. Then $X_A(p) = X_R(p)/2$ if at least one element in $A_8(p)$ is a non-object point (a point with value 0). Hilditch [6] defined the crossing number as follows:

Definition 4 *The number of times of crossing over from a background point to an object point when the points in $A_8(p)$ are traversed in order, cutting the corner between 8-adjacent 1's, is called* H-crossing number $X_H(p)$*:*

$$X_H(p) = \sum_{i=1}^{4} b_i$$

where

$$b_i = \begin{cases} 1 & \text{if } x_{2i-1} = 0 \text{ and } (x_{2i} = 1 \text{ or } x_{2i+1} = 1) \\ 0 & \text{otherwise .} \end{cases}$$

The H-crossing number is equivalent to the number of distinct 8-components of 1's in $A_8(p)$ in case there is at least one 0 in $A_4(p)$ and the H-crossing number is always equal to the number of distinct 4-adjacent 4-components of 0's in $A_8(p)$.

Yokoi et al [26] introduced the notion of connectivity number.

Definition 5 *The number of distinct 4-adjacent 4-components of 1's (0's) is called* connectivity number $X_B(p)$ *($\overline{X_B}(p)$) with:*

$$X_B(p) = \sum_{i=1}^{4} a_i$$

where

$$a_i = x_{2i-1} - x_{2i-1} * x_{2i} * x_{2i+1}, x_9 = x_1$$

and

$$\overline{X_B}(p) = \sum_{i=1}^{4} b_i$$

where

$$b_i = \overline{x}_{2i-1} - \overline{x}_{2i-1} * \overline{x}_{2i} * \overline{x}_{2i+1}$$

and

$$\overline{x}_i = 1 - x_i$$

in $A_8(p)$.

For the grid cell model, Kong [9] defined the I-attachment set of a pixel p.

Definition 6 *The set of all points on the frontier of p that also lie on the frontier of at least one other pixel q with $I(p) = I(q), p \neq q$ is the I-attachment set of p in I.*

Figure 2 shows examples for I-attachment sets, crossing numbers and connectivity numbers.

There are straightforward ways to compute crossing numbers $X_R(p)$ and $X_H(p)$. These definitions are frequently used for the design of thinning algorithms in order to find a decision whether $I(p) = 1$ can be changed to $I'(p) = 0$ in the transformed image $\langle I' \rangle = \langle I \rangle \setminus \{p\}$ without changing the adjacency relations.

Non-endpoints of ideally thin curves do not satisfy pixel deletion criterions in the sense that they are non-simple and a thinning algorithm stops if all remaining points are non-simple. This is different for endpoints. Let $B(p)$ denotes the number of object points in $A_8(p)$. There are the following definitions of an endpoint in the literature.

1. $B(p) = 1$,
2. $B(p) \in \{1, 2\}$,
3. $B(p) = 1$, or $B(p) = 2$ and $X_R(p) = 2$.

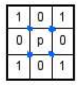
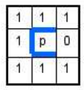
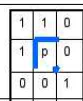
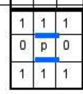
I-attachment sets	$X_R(p)$	$X_H(p)$	$X_B(p)$
	$X_R(p)=8$	$X_H(p)=4$	$X_B(p)=0$
	$X_R(p)=2$	$X_H(p)=1$	$X_B(p)=1$
	$X_R(p)=4$	$X_H(p)=2$	$X_B(p)=1$
	$X_R(p)=4$	$X_H(p)=2$	$X_B(p)=2$

Figure 2. Examples: I-attachment sets, crossing numbers, connectivity number, 2D.

Condition 1 is not very strong. Algorithms using this condition can delete "important parts" of the skeleton.

One example for sequential algorithms is given in [6] (Algorithm A). The following criterions are tested for all points $p \in C$ and pixels are marked in standard scan order from the top to the bottom, and from the left to the right.

A1: p is an object pixel.

A2: p is a border pixel, that means at least one 4-neighbor is a non-object pixel.

A3: p is not isolated or an end pixel, that means $B(p) > 1$.

A4: At least one object pixel in $A_8(p)$ is unmarked.

A5: $X_H(p) = 1$.

A6: If x_3 is marked, setting $I(x_3) = 0$ does not change $X_H(p) = 1$.

A7: If x_5 is marked, setting $I(x_5) = 0$ does not change $X_H(p) = 1$.

At the beginning let Q be the set of all object pixels (1's) and B the set of all background pixels (0's). After one iteration all marked pixels (set D) are changed to background pixels. The result $Q' = Q \setminus D$ becomes Q and $B' = B \cup D$ becomes B for the next iteration and so on until no simple pixel is left. Condition 5 implies that p is object pixel, at least one 4-neighbor is a 0 and p is not isolated. Conditions 1, 2 and the end point condition are checked at the beginning to be computationally efficient. Condition 5 determines whether a single pixel p is simple. Conditions 4, 6 and 7 are necessary to preserve the topology of the image because the algorithm determines and changes the values of a set D of simple pixels in one iteration. Figure 3 shows two results of our experiments:

These examples show that the algorithm is sensitive regarding small holes or sharp peaks. We pre-processed the images using the morphological operations *closing* and *opening* as usually defined for example in [5] and the results (see figure) show improvements regarding robustness against noise.

After smoothing the border the algorithm delivers a set of connected digital arcs that is a topologically equivalent image to the smoothed image. The resulting subset approximates the medial axis. End points are preserved.

One standard example of using subcycles or subiterations is the four- subiteration algorithm by Rosenfeld [21] (Algorithm B). A pixel p is deleted if

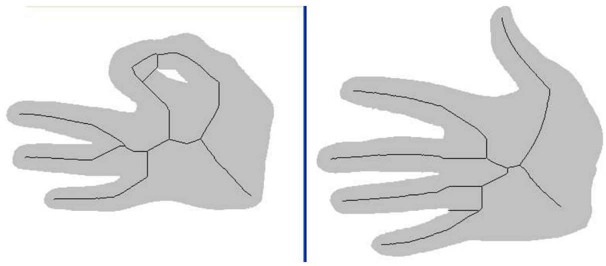


Figure 3. Examples: Results of algorithm A without morphological pre-processing

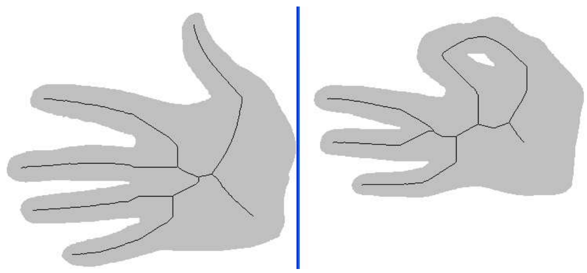


Figure 4. Examples: Results of algorithm A with morphological pre-processing

B1: p is an object pixel.

B2: p is not isolated or an end pixel, that means $B(p) > 1$.

B3: $X_H(p) = 1$.

B4: $x_{2i+1} = 0$ where $i = 1, ..., 4, 1, ...$ at successive iterations.

The first of these four subiterations deletes in parallel only contour pixels where $x_3 = 0$, the second subiteration deletes only contour pixels where $x_5 = 0$ and so on. The algorithm terminates when no deletions occur during four successive iterations. Rosenfeld has shown that these four iterations preserve connectivity. It turned out that the end point condition $B(p)$ is not strong enough. Too many pixels are deleted in some cases. Therefore condition D2 should be replaced by: $B(p) = 1$, or $B(p) = 2$ and $X_R(p) = 2$.

Computationally more efficient are parallel algorithms with two or only one subcycle. A typical example for a 2-subcycle algorithm [27] (Algorithm C) is the following: The first subcycle deletes only the following pixels:

C1: p is an object pixel.

C2: $1 < B(p) < 6$.

C3: $X_R(p) = 2$.

C4: $x_1 x_3 x_7 = 0$.

C5: $x_1 x_7 x_5 = 0$.

Equations E4 and E5 are equal to 0 if $x_1 = 0$ or $x_7 = 0$ or $x_3 = 0$ and $x_5 = 0$. In the second subcycle the last two conditions are replaced by equations $x_3 x_5 x_7 = 0$ and $x_1 x_3 x_5 = 0$. The algorithm terminates when no deletions occur at two successive subiterations. The algorithm delivers connected digital arcs similar to the first algorithm. The locations of the skeletons (see figures) are different. The subcycles differ only in conditions C4 and C5. The algorithm does not identify an object pixel if $x_2 = x_3 = x_4 = 0$ or $x_4 = x_5 = x_6 = 0$ in the first subcycle.

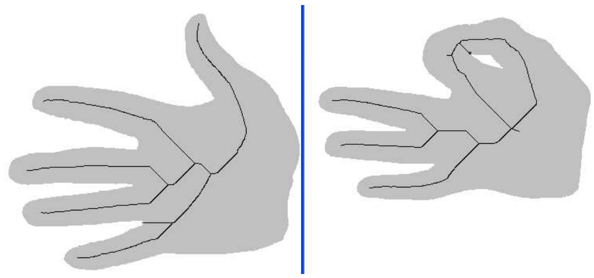


Figure 5. Results of algorithm C

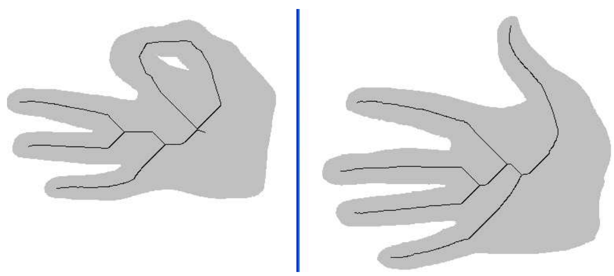


Figure 6. Results of algorithm C with morphological pre-processing

Another group of thinning operators are window matching algorithms. For example, the free software package *ImageJ* contains an implementation of the above described 2-subcycle algorithm using windows. Conditions C1 to C5 can be equivalently represented by 30 3×3 windows. In case $N_8(p)$ matches one of the stored windows then p can be marked for deletion. Note that for the second subcycle the windows are only different for $B(p) = 5$.

Algorithm D [15] belongs to the group of sequential algorithms. Border pixels are traced and labeled. Only multiple pixels are retained. Pixel p is *multiple* if at least one of the following conditions is true:

D1: p is traversed more than once during tracing $(X_H(p) > 1)$

D2: p has no 4-neighbors in the interior.

D3: p has at least one 4-neighbor that belongs to the contour but it is not traced immediately before or after p.

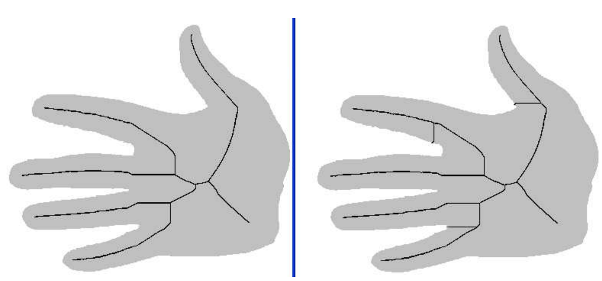


Figure 7. Result of algorithm D with and without pre-processing

Thinning algorithms A and D produce 8-connected sets of the original subsets and algorithm C delivers 4-connected subsets. All three algorithms preserve the topology of the original images. The skeletons produced by algorithms A and C are ideally thin. Algorithm D delivers a reduced amount of pixels approximating the medial axis, but it does not deliver skeletons. The resulting component is not ideally thin because it is easy to determine pixels that can be deleted without changing the adjacency/surroundedness relations. All simple deformation algorithms include special tests to preserve the topology of the image. Characterizations of single simple pixels are subject of many publications. This report reviews a few of them.

2.3 Equivalent characterizations of simple points

Simple pixels have in common that changes from 1 to 0 or vice versa, preserve the adjacency/surroundedness relations between the connected components of 1's and 0's of the image. There are different ways to define this important property. Some of the characterizations in the literature are abstract such as Kong's definition [9] using homotopy equivalence in order to include pixels of 3- and higher dimensional images. However for the design of algorithms it is necessary to determine whether a given pixel of an image I is simple in I. We review a few characterizations which are easy to compute or easy to visualize for 2D digital images. We used two of them in previous described algorithms. We use the grid point model and we define a simple 1 [22] as follows:

Definition 7 *A* 1 *of an image I is called α-simple if it is α-adjacent to exactly one α-component of 1's in $A_8(p)$ and it is α'-adjacent to exactly one α'-component of 0's in $A_8(p)$.*

In case that p is a border pixel we can simplify above definition: Let $(p, I(p))$ be a border pixel of an image I. Then $(p, I(p))$ is α-simple in I iff p is α-adjacent to exactly one α-component of $I(p)$ in $A_8(p)$.

Note that a 4-simple pixel p can be 4-adjacent to exactly one 4-component of 1's in $A_8(p)$ and 8-adjacent to distinct 4-components of 1's in $A_8(p)$. In the example below p is a 4-simple 1 or an 8-simple 0.

$$\begin{matrix} 1 & 1 & 1 \\ 0 & p & 0 \\ 1 & 0 & 1 \end{matrix}$$

We consider both changing 1 to 0 and vice versa. The result of changing an α-simple point p of an α-component U is a non-empty α-component $U \setminus \{p\}$ and a non-empty α'-component $V \cup \{p\}$, and the adjacency relations to all other components remain the same. The region adjacency trees of the original image and the resulting image are isomorph. We extend Definition 7: A pixel $(p, I(p))$ is called α-simple if it is α-adjacent to exactly one α-component in $A_8(p)$ and it is α'-adjacent to exactly one distinct α'-component in $A_8(p)$. The result of changing the value of a single α-simple pixel p is an α'-simple pixel.

The change of the value of a simple pixel delivers a topologically equivalent image [22]. Two images differ by *simple deformation* if one can be obtained from the other one by repeatedly changing simple pixels from 1 to 0 or vice versa. Thinning or shrinking procedures are one-way simple deformations (from 1 to 0).

In the literature different characterizations of "simple pixel" are based on different assumptions (i.e. 8-connectivity is used for the 1's and 4-connectivity for the 0's). Simple pixels are often restricted to simple 1's. This paper reviews a few earlier characterizations of simple 1's but we consider general properties of simple pixels in order to use them for two-way simple deformations.

Characterization 1 *A* 1 *of an image I is 4-simple in I iff $X_R(p) = 2$.*

Changing 1 to 0 of an image I preserves 4-connectivity of I if there is exactly one change from 0 to 1 and exactly one change from 1 to 0 in $A_8(p)$. In $A_8(p)$ is exactly one 4-component of 1's and exactly one 4-component of 0's. This is a restriction compared to 4-simple pixels based on Definition 7.

Hilditch [6] defined an 8-simple 1 as follows:

Characterization 2 *A* 1 *of an image I is 8-simple in I iff $X_H(p) = 1$.*

This characterization is equivalent to Hall's characterization [8] where a 1 is 8-simple iff there is exactly one distinct 8-component of 1's in $A_8(p)$ and p is a border 1.

The following well-known characterization of simple pixels is equivalent to Characterization 2 [19].

Characterization 3 *A 1 of an image I is 8-simple in I iff both of the following conditions hold: The union of all pixels in $I \setminus \{p\}$ that is 8-adjacent to p is nonempty and connected. p is 4-adjacent to a 0.*

Kong [9] used the concept of an I-attachment set of a pixel $(p, I(p))$ (grid cell model). This characterization is easy to visualize.

Characterization 4 *A 1 at p of an image I is 0-simple in I iff the I-attachment set of p, and the complement of that set in the frontier of p, are non-empty and connected.*

We can simplify this preposition for 2D images in the following way:

Characterization 5 *A 1 at p of an image I is 0-simple in I iff the I-attachment set of p is non-empty and connected, and it is not the entire frontier of p.*

Kong has shown in [9], that the last two characterizations are equivalent for 2D images.

In order to determine whether a pixel is simple for two-way simple processes we show the following theorem [**?**]:

Theorem 2 *A 1(0) of an image I is 4-simple iff $X_B(p) = 1$ ($\overline{X}_B(p) = 1$). A 1(0) of an image I is 8-simple iff $\overline{X}_B(p) = 1$ ($X_B(p) = 1$).*

Proof. Let p be a 4-simple 1 (Definition 7). First we assume $X_B(p) = 0$; i.e. $a_i = 0$, for all $1 \leq i \leq 4$. Per Definition 3 we have $a_i = x_{2i-1} - x_{2i-1} * x_{2i} * x_{2i+1}$. It follows that $a_i = 0$ iff $x_{2*i-1} = 0$ or $x_{2*i-1} = x_{2*i} = x_{2*i+1} = 1$. In case $x_{2*i-1} = 0$ for all $1 \leq i \leq 4$ there is a contradiction to the property of having exactly one 4-adjacent 4-component of 1's. In case $x_{2*i-1} = x_{2*i} = x_{2*i+1} = 1$, for all $1 \leq i \leq 4$, there is a contradiction to the property of having exactly one 8-component of 0's in $A_8(p)$.

Now assume $X_B(p) > 1$. Then there exist at least two a_i's with $a_i = a_j = 1, i \neq j, 1 \leq i, j \leq 4$. It follows that there are at least two 4-adjacent 4-components of 1's and two 8-adjacent 8-components of 0's which is a contradiction to the assumption that p is a 4-simple 1. It follows that $X_B(p) = 1$.

Let $X_B(p) = 1$; i.e. one $a_i = 1$, say $a_1 = 1$ and $a_j = 0$, for all $2 \leq j \leq 4$. $a_1 = 1$ iff $x_1 = 1$ and $x_2 = 0$ or $x_3 = 0$. It follows that at least one 4-adjacent 4-component of 1's and at least one 8-adjacent 8-component of 0's exist. For all other terms is $a_j = 0$, for all $2 \leq j \leq 4$, that means $x_{2*j-1} = 0$ or $x_{2*j-1} = x_{2*j} = x_{2*j+1} = 1$.

$x_{2*j-1} = 0$, for all $2 \leq j \leq 4$, it follows that there is exactly one 4-adjacent 4-component of 1's, and that there is exactly one 8-adjacent 8-component of 0's. In case $x_{2*j-1} = x_{2*j} = x_{2*j+1} = 1$, for all $2 \leq j \leq 4$, it follows that $x_2 = 0$ is the only 8-adjacent 8-component of 0's. Now we consider $x_{2*j-1} = 0$ and $x_{2*j+1} = 1$, for all $j = 2, 3$ then x_{2*j+1} is always 4-connected to $x_1 = 1$. The value of x_{2*j} can be 0 or 1, it doesn't change the number of 4-adjacent 4-components of 1's and the number of 8-adjacent 8-component of 0's. In each case there is exactly one 4-adjacent 4-component of 1's and exactly one 8-adjacent 8-component of 0's. All other cases follow by symmetry.

Let p be an 8-simple 1. There is exactly one 4-adjacent 4-component of 0's and exactly one 8-adjacent 8-component of 1's. Based on the definition of $\overline{X_B(p)}$ the proof is analog. The proof for 4-(8)-simple 0's is analog to the given proof for simple 1's. □

It follows that a pixel $(p, 1)$ is 4-simple in I iff $(p, 0)$ is 8-simple in I and a pixel $(p, 1)$ is 8-simple in I iff $(p, 0)$ is 4-simple in I. It follows as well that $(p, 0)$ is 8-simple in I iff $(p, 1)$ is 8-simple in $\overline{I}$ where $\overline{I}$ is the negative image of I (all 1's in I are 0's in $\overline{I}$ and vice versa). The good pair concept for binary images (use of 8-connectivity for object components and 4-connectivity for non-object components or vice versa) provides this property of duality for simple pixels. For simple deformations the use of binary images is verified by applying the characteristic function after segmentation of grey value input images. The application of algorithm A on $\overline{I}$ for example delivers a magnification of the original components in I that is topologically equivalent to the original image.

Note that for a 1 at (p) of an image I the crossing number $X_H(p)$ is always equal to $\overline{X}_B(p)$. The characterization of an 8-simple 1 in above theorem is equivalent to Characterization 2. It follows that in case $X_H(p) = 1$ then p is an 8-simple 1 or a 4-simple 0 depending on the value $I(p) = 1(0)$.

The following theorem shows that characterizations for 8-simple 1's based on I-attachment sets are equivalent to Characterization 2.

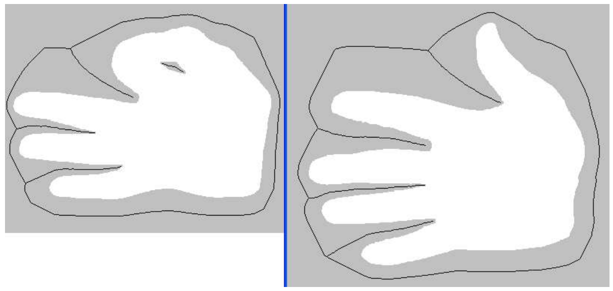


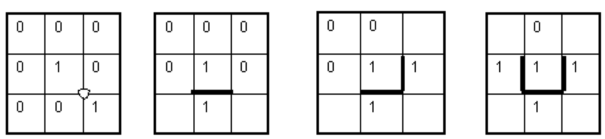
Figure 8. Result of algorithm A applied on $\overline{I}$

Theorem 3 *Let p be a* 1 *of an image I. Then the I-attachment set of p is non-empty and connected and not the entire frontier of p iff* $X_H(p) = 1$.

Proof. Let p be 8-simple. Then we have only 4 ways in which the I-attachment set is non-empty and connected and not the entire frontier, in each case $X_H(p) = 1$. Let $X_H(p) = 1$. At least one 8-neighbor is a 1 per definition. It follows the I-attachment set is non-empty. At least one 4-neighbor is a 0 per definition, it follows the I-attachment set is not the entire frontier. It remains to show that the I-attachment set is connected. Assume the I- attachment set is not connected then $X_H(p) > 1$ what is a contradiction. □

The following theorem justifies the choice of masks for window matching algorithms.

Theorem 4 *Let p be a* 1 *of an image I. Then p is per characterization 5 8-simple in I iff the neighborhood of p matches one of the following masks (simple point masks 1 to 4, from left to right, empty squares can be either* 0 *or* 1*)*



or one of their 90° *rotations.*

Proof. These masks represent the only 4 ways in which the I-attachment set is non-empty and connected and not the entire frontier. □

Previously published thinning algorithms have used Characterization 1 in order to preserve 4-connected subsets of the original image. The following example shows two 4-components of 1's that are 8-adjacent.

$$\begin{array}{ccc} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{array}$$

Based on Characterization 1 p would be identified as a 4-simple 1 because $X_R(p) = 2$. If we would use only this criterion then a 4-component of a single 1 that is 8-adjacent to a component of 1's would completely disappear. An algorithm based on this condition determines pixels as simple which have exactly one 4-component of 1's and exactly one 4-component of 0's in $A_8(p)$. The 4-component of 1's can be 4-adjacent or 8-adjacent to p. For $(\alpha, \alpha') = (8, 4)$ and $(\alpha, \alpha') = (4, 8)$ a thinning algorithm using Characterization 1 for connectivity preservation results in a 4-connected subset of the original 4-component. A 4-component consisting of a single pixel and 8-adjacent to a disjoint 4-component disappears.

Now let us consider the good pair $(\alpha, \alpha') = (4, 8)$. In case two disjoint 4-components of 1's are 8-adjacent to each other then there exists a 1 which is 4-simple and $X_R(p) \neq 2$. Evidently this 1 is not 4-simple based on

Characterization 1. To avoid this conflict Latecki et al [11] introduced the concept of *well-composedness*. The following Theorem includes an equivalent condition to well-composedness that ensures that the described critical configurations cannot occur.

Theorem 5 *Let all disjoint 4-components of 1's of an image I be pairwise 8-separated by 0's to each other. Then a 1 is 4-simple in I iff $X_R(p) = 2$.*

Proof. Let $X_B(p) = 1$; and any 8-path from a 1 of a 4-component to a disjoint 4-component of 1's must intersect a 0. It follows that we have exactly one 4-adjacent 4-component of 1's and exactly one 4-component of 0's in $A_8(p)$ and $X_R(p) = 2$. Now we consider $X_R(p) = 2$, and $X_B(p) = 0$. The only possible configurations would be $x_2 = 1$ and all other pixel values in $A_8(p)$ are 0 and all symmetric cases. This is a contradiction to our precondition. Let us consider $X_R(p) = 2$, and $X_B(p) > 1$. A configuration for this condition does not exist. It follows that $X_B(p) = 1$. □

All simple deformation algorithms have in common that a set of pixels is satisfying a number of tests and these pixels change the values $I(p)$ in one iteration. This is the reason that tests of single points to be simple are insufficient to preserve the topology of an image.

2.4 Characterizations of simple sets

Similar to tests of single points to be simple we are interested in characterizing sets of points to be simple in order to show that specified algorithms for simple deformations preserve the topology of the image. Analog to Ronse [17] we define:

Definition 8 *Let D be the set of pixels in the original image I that has changed the value of the image I^* after deformation, $D = \{p : p \in I \wedge I(p) \neq I^*(p)\}$, $D \subseteq I$. Let S be the set of α-components of 1's and B' the set α'-components of 0's after deformation. Then $S = I \setminus D$ (thinning) or $S = I \cup D$ (magnification) and $B' = B \cup D$ (thinning) or $B' = B \setminus D$ (magnification). $C_\alpha(I)$ is the number of α-connected components of I, α is equal to 8 or 4. D is* α-simple *iff $C_\alpha(I) = C_\alpha(S)$, $C_{\alpha'}(B) = C_{\alpha'}(B')$. D is* strongly α-simple *iff a) for each α-connected component exists exactly one α-connected component after deformation and vice versa b) for each α'- connected background component exists exactly one α'-connected background component after deformation and vice versa.*

Analogue we say an image I is α-deformable by set D iff $C_\alpha(I) = C_\alpha(S)$, $C_{\alpha'}(B) = C_{\alpha'}(B')$ and an image I is strongly α-deformable by set D iff a) for each α-connected component exists exactly one α-connected component in the resulting image and vice versa b) for each α'- connected component exists exactly one α'-connected component after deformation and vice versa.

In other words a subset D of a digital image I is strongly α-simple iff two bijective maps exist: one between α-connected components of I and α-connected components after deformation of I by D, and the other one between the α'-connected components of I and the α'-connected components after deformation of I by D.

We extend Kongs definition of a *simple sequence* [9]:

Definition 9 *Let $I = I_0$ the original image and I_i the result of deformation i of a single α-simple pixel ($\alpha = \{8, 4\}$). A sequence $q_1, q_2, ...q_n$ of distinct pixels of an α-connected component in a digital image I is called an* α-simple sequence *of I if q_1 is α-simple in I, and q_i is α-simple in $I_{i-1}, 2 \leq i \leq n$. A set D of pixels in a digital image is called* simple *in I if D is empty or if D is finite and the elements of D can be ordered as an α-simple sequence of I.*

We show:

Theorem 6 *An image I is strongly α-deformable by D iff D is simple in I.*

Proof. First we assume that I is strongly α-deformable. Per definition no α-connected component and no α'-connected component can vanish. For each α-connected component P' exists a proper subset D' with $D' \subset P'$, $S' = P' \setminus D'$ or $S' = P' \cup D'$, and D is the union of all these subsets D'. I is a digital image, it follows that D is finite or empty. If D is empty then no pixel value has been changed and the α-connected components in the original image are unchanged. In case D is finite then all subsets D' are finite or empty. For each α-connected component in the original image exists exactly one α-connected component after deformation. For each α-connected component exists a set D' which changed the value during deformation. This set is empty or finite. Each non-empty D' has at least one element what must be simple otherwise the change of the value would split the α-connected component in I or the α'-connected component in $\overline{I}$. In case D' has more than one element all these elements can be ordered in an α-simple sequence. All these sequences can be ordered in one sequence one after another and this new sequence is simple. That means D is simple.

Now we assume D is simple. Per definition D must be empty or finite. In case D is empty then the α-connected components of I stay exactly the same and all properties of strongly α-deformable are valid. In case D is finite the elements of D can be ordered as a simple sequence of I. Now we consider all elements of D' of the same α-connected component with $D' \subseteq D$. That means no α-connected component can be split or vanish and no α'-connected component can be split or vanish. It follows that two bijective maps exist between α-connected components before and after simple deformation, and between α'-connected components before and after simple deformation. It follows that D is strongly α-deformable. □

This theorem shows only the equivalence of two characterizations of simple sets. We are interested in using local operations to show that the implemented algorithms preserve the topology of the image.

2.5 Topology Preserving Algorithms

An algorithm preserves the topology of an image if it satisfies the following properties:

a: It must not split an α-connected component of I into two or more α-connected components of I.

b: It must not completely delete an α-connected component of I.

c: It must not split an α'-connected component of $\overline{I}$.

d: It must not completely delete an α'-connected component of $\overline{I}$.

Obviously an algorithm preserves the topology if it changes the values of a set D of an image I which is simple in I. For practical reasons it is more interesting to find test criterions based on local neighborhoods. In order to report about conditions for testing algorithms we introduce two more definitions:

Definition 10 *A set of pixels is* small *if every 2 elements of the set are adjacent to each other. An* 8-deformable set *is a set which can be deformed while preserving 8-connectivity.*

Obviously, every small set of pixels is connected. A pair of 8-simple points p, q is 8-deformable iff q is 8-simple after p is deformed. In case p and q are 4-neighbors the pair p, q is 8-deformable iff the number of 8-connected components in the adjacency set of p, q is one. The following examples show configurations where p and q are 4-adjacent and the number of 8-connected components is larger than one.

$$\begin{matrix} 0 & 1 & 0 & 0 \\ 0 & p & q & 0 \\ 0 & 0 & 1 & 0 \end{matrix} \qquad \begin{matrix} 1 & 1 & 0 & 0 \\ 1 & p & q & 0 \\ 0 & 0 & 1 & 1 \end{matrix} \qquad \begin{matrix} 0 & 1 & 1 & 0 \\ 0 & p & q & 0 \\ 0 & 1 & 1 & 0 \end{matrix}$$

Let A be a thinning algorithm. The following conditions are sufficient to show that A preserves topology (topology preservation test):

a: If an object point is deleted by A then it must be simple.

b: If two 4-neighbors in I are deleted by A then they must constitute a simple set.

c: No small set of object points vanished by A.

For instance we like to prove that algorithm A preserves connectivity. Condition (a) is valid because of pixel deletion criteria A5. If x_3 or x_5 are simple and marked then criterions A6 and A7 guarantee that p can only be marked if p is simple in $I \setminus \{x_3\}$ and p is simple in $I \setminus \{x_5\}$. A small set cannot be completely deleted because of A3 and A5.

3 Summary and Conclusions

The determination of distance skeletons on binary digital images is a non-iterative method based on the calculation of distance transforms using different metrics. The notions of simple pixels and simple sets are important for topology preserving deformations of binary images. Topology preserving thinning is a one-way simple deformation. There are two types of operators used in thinning. One type works sequentially in standard scan order or following the contour [6], [15]. The other type of operators works parallel. Two-way simple deformations include both ways, that means thinning (changing 1 to 0) and magnification (0 to 1). We showed that the duality of simple points in binary images can be used to magnify a connected component without destroying the topology of the image.

References

[1] C. Arcelli, L. Cordella, S. Levialdi: Parallel thinning of binary pictures. *Electron. Lett.* **11**:148–149, 1975.

[2] C. Arcelli, G. Sanniti di Baja: Skeletons of planar patterns. in: *Topological Algorithms for Digital Image Processing* (T. Y. Kong, A. Rosenfeld, eds.), North-Holland, 99–143, 1996.

[3] H. Blum: A transformation for extracting new descriptors of shape. in: *Models for the Perception of Speech and Visual Form* (W. Wathen- Dunn, ed.), MIT Press, Cambridge, Mass., 362–380, 1967.

[4] G. Sanniti di Baja: Well-shaped, stable and reversible skeletons from the (3,4)-distance transform. in *J. Visual Comm. Image Repres.* **5:** 107–115, 1994.

[5] R. C. Gonzalez, R. E. Woods: Digital Image Processing. *Prentice-Hall, Inc.*, Upper Saddle River, New Jersey (2002), 519–566

[6] C. J. Hilditch: Linear skeletons from square cupboards. in: *Machine Intelligence 4* (B. Meltzer, D. Mitchie, eds.), Edinburgh University Press 403–420, 1969.

[7] R.W. Hall: Fast parallel thinning algorithms: parallel speed and connectivity preservation. *Comm. ACM* **32:** 124–131, 1989.

[8] R.W. Hall: Parallel Connectivity-Preserving Thinning Algorithms. in: *Topological Algorithms for Digital Image Processing* (T. Y. Kong, A. Rosenfeld, eds.), Elsevier Science B. V. 145–179, 1996

[9] G. Klette: Characterizations of Simple Pixels in Binary Images. in Proceedings: *Image and Vision Computing New Zealand 2002*, 227–232, 2002

[10] T. Y. Kong: On topology preservation in 2-D and 3-D thinning. in: *International Journal for Pattern recognition and artificial intelligence* vol. 9 No.5 (1995) 813–844.

[11] V. Kovalevsky: Algorithms and data structures for computer topology. in: *Digital and Image Geometry* (G. Bertrand, A. Imiya, R. Klette, eds.), LNCS 2243, Springer, Berlin Heidelberg (2001) 38–58.

[12] L. Latecki, U. Eckhardt, A. Rosenfeld: Well-composed sets. in *Computer Vision Image Understanding* **61**, 1995, 70–83.

[13] L. Lam, S. Lee, C. Suen: Thinning methodologies - a comprehensive survey. *IEEE PAMI* **14** (1992) 869–885.

[14] J. B. Listing: Der Census räumlicher Complexe oder Verallgemeinerungen des Euler'schen Satzes von den Polyëdern. Abhandlungen der Mathematischen Classe der Königlichen Gesellschaft der Wissenschaften zu Göttingen **10** (1862 and 1862) 97–182.

[15] U. Montanari: A method of obtaining skeletons using a quasi-Euclidean distance. *Comm. ACM* **15** (1968) 600–624.

[16] T. Pavlidis: A thinning algorithm for discrete binary images. In: *Computer Graphics Image Processing* **13**(1980) 142–157.

[17] J. L. Pfaltz, A. Rosenfeld: Computer representation of planar regions by their skeletons. *Comm. ACM* **10** (1967) 119–122.

[18] C. Ronse: A topological characterization of thinning. in: *Theoretical Computer Science*, **43**, (1986) 31–41.

[19] A. Rosenfeld, J. L. Pfaltz: Sequential operations in digital picture processing. *Comm. ACM* **13** (1966) 471–494.

[20] A. Rosenfeld: Connectivity in digital pictures. *Comm. ACM* **17** (1970) 146–160.

[21] A. Rosenfeld: Adjacency in Digital Pictures. In: *Information and Control* **26** (1974) 24-33.

[22] A. Rosenfeld: A Characterization of parallel thinning algorithms. *Information and Control* **29** (1975) 286–291.

[23] A. Rosenfeld, T. Y. Kong, A. Nakamura: Topology- Preserving Deformations of Two-Valued Digital Pictures. *Graphical Models and Image Processing* **60** (1998) 24-34.

[24] A. Rosenfeld, R. Klette: Topologies for binary or multi- level images - A review. Keynote, 6th Joint Conf. Inform. Sciences, March 8-13, 2002, Research Triangle Park, North Carolina, 2002 (to appear).

[25] D. Rutovitz: Pattern recognition. *J. Royal Statist. Soc.* **129** (1966) 504–530.

[26] J. Serra: Image Analysis and Mathematical Morphology, vol.2, Academic Press, New York (1982)

[27] S. Yokoi, J. I. Toriwaki, T. Fukumura: An analysis of topological properties of digitized binary pictures using local features. in: *Computer Graphics and Image Processing* 4, (1975), 63-73.

[28] T. Y. Zhang, C. Y. Suen: A fast parallel algorithm for thinning digital patterns. *Comm. ACM* **27** (1984) 236–239.