Pattern Recognition Letters 76 (2016) 3–12

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

A survey on skeletonization algorithms and their applications ☆

Punam K. Saha [a,b,*], Gunilla Borgefors [c], Gabriella Sanniti di Baja [d,e]

[a] Department of Electrical and Computer Engineering, University of Iowa, Iowa City, IA, USA
[b] Department of Radiology, University of Iowa, Iowa City, IA, USA
[c] Centre for Image Analysis, Uppsala University, Uppsala, Sweden
[d] Institute of Cybernetics "E.Caianiello", CNR, 80078 Pozzuoli, Naples, Italy
[e] Institute for High Performance Computing and Networking, CNR, 80131 Naples, Italy

ARTICLE INFO

Article history:
Available online 28 April 2015

Keywords:
Skeletonization
Centers of maximal balls
Distance transform
Topology preservation
Parallel algorithms
Applications

ABSTRACT

Skeletonization provides an effective and compact representation of objects, which is useful for object description, retrieval, manipulation, matching, registration, tracking, recognition, and compression. It also facilitates efficient assessment of local object properties, e.g., scale, orientation, topology, etc. Several computational approaches are available in literature toward extracting the skeleton of an object, some of which are widely different in terms of their principles. In this paper, we present a comprehensive and concise survey of different skeletonization algorithms and discuss their principles, challenges, and benefits. Topology preservation, parallelization, and multi-scale skeletonization approaches are discussed. Finally, various applications of skeletonization are reviewed and the fundamental challenges of assessing the performance of different skeletonization algorithms are discussed.

1. Introduction

Skeletonization provides an effective and compact representation of an image object by reducing its dimensionality to a "medial axis" or "skeleton" while preserving the topologic and geometric properties of the object. In two dimensions (2-D), an object is reduced to a curve skeleton consisting of one-dimensional (1-D) structures. In three dimensions (3-D), an object may be converted to a surface skeleton, i.e., a union of 1-D and 2-D structures, or, it may be reduced to a curve skeleton consisting of only 1-D structures. Blum [24] established the foundation of skeletonization in the form of medial loci of an object in R^n that forms the skeleton of the object. This skeleton consists of planes/axes of symmetry with lower dimensionality. The skeleton is useful for object description, retrieval, manipulation, matching, registration, tracking, recognition, compression, and it also facilitates efficient assessment of local object properties, e.g., scale, orientation, topology etc. Analytically, Blum's skeleton, or medial axis, is defined using a grassfire transform process [25] where the object is assumed to be a field of dry grass and a fire is simultaneously lit at all boundary points. The fire propagates inside the object at a uniform velocity. The skeleton is the set of quench points, where two independent fire-fronts meet [64,84,85,96,165,173,175].

Blum's grassfire transform was later generalized and adopted by the image processing, computer vision, and graphics community in the form of the medial axis, or skeleton, of objects. Commonly, when the *continuous* approach is taken, the boundary of the object is approximated by a polygon or a curve, the grassfire propagation process is realized by curve evolution or constrained mathematical morphological erosion and the skeleton is formed at quench points where the curve evolution process is interrupted [84,85,96,173]. Several researchers have sought geometric features, e.g., the Voronoi diagram, to identify symmetry structures in an object [36,125,128,129]. *Digital* approaches can simulate the grassfire propagation using an iterative constrained erosion process [92,95,126,135,158,190]. Another group of digital skeletonization algorithms [4,7,23,33,145,165] locate maximal balls [165] on digital distance transform field [27].

Blum inspired the representation of an object by the loci of the centers of its maximal inscribed balls (MIBs), together with their radii, which allows exact reconstruction of the object from its medial loci. Computationally, a skeleton may be perceived in three ways: the Blum's quench points by opposing fire-fronts, the centers of MIBs, or the centers of the enclosed balls that touch the object boundary at two or more disjoint locations. For objects in R^2 or R^3, these three skeletonization approaches are roughly equivalent. However, the same is not true for a digital object and these three definitions may produce different skeletons [175]. The inherent discrete nature of digital objects further complicates the skeletonization task by

☆ This paper has been recommended for acceptance by Gabriella Sanniti di Baja.

* Corresponding author at: Department of Electrical and Computer Engineering, University of Iowa, Iowa City, IA, USA. Tel.: +1 319 335 5959; fax: +1 319 335 6028.
E-mail address: punam-saha@uiowa.edu, pksaha@engineering.uiowa.edu, pksaha@healthcare.uiowa.edu (P.K. Saha).

http://dx.doi.org/10.1016/j.patrec.2015.04.006
0167-8655/

posing major hurdles, e.g., high sensitivity to small details on the object boundary, homotopy, non-standard definitions of digital balls, etc. Note that high sensitivity to small details on the object boundaries is also featured by non-discrete or continuous approaches to skeletonization [10,118]. In most applications, it is desired that the computed skeleton of a digital object is robust under different conditions of digitization and imaging artifacts; consist of one-voxel thin curves and surfaces to enable tracking; and allow acceptable reconstruction of the original object. This makes the evaluation of the performance of digital skeletonization algorithms challenging [176,177]. Note that taking the continuous route does not remove these problems–the digital object in an image then has to be converted to continuous data and the resulting skeleton may have to be digitized. Neither is trivial.

As discussed above, several computational approaches have been reported in literature toward extracting the skeleton of an object, some of which are widely different in terms of their principles. Several researchers have used continuous methods while others have used purely digital approaches to compute the skeleton of an object. Discussion on different principles of skeletonization algorithms has been reported [175]. In this paper, we present a concise survey of different skeletonization approaches and algorithms and discuss their principles, challenges, and advantages. Also, topology preservation, parallelization, and multi-scale skeletonization approaches are discussed. Finally, various applications of skeletonization are reviewed and the fundamental challenges of assessing the performance of different skeletonization algorithms are discussed.

2. Different approaches of skeletonization

Skeletonization algorithms may be grouped into three major categories based on their principles and the underlying object representation:

(1) Algorithms based on Voronoi diagram or continuous geometric approaches of point clouds, polygonal, or polyhedral representations of object boundaries. Such algorithms use Voronoi edges or planes to construct the symmetry structures or the skeleton.
(2) Algorithms based on the principle of continuous evolution of object boundary-curves where the skeleton is formed at the locations of singularities, e.g., at collision points of opposing boundaries.
(3) Algorithms based on the principle of digital morphological erosion or location of singularities on a digital distance transform (DT) field, e.g., maximal included balls.

Besides the above three categories, Pizer and coworkers [59,122,139] presented algorithms to extract zoom-invariant cores from intensity images. The medial cores are defined as generalized maxima in scale-space produced by a medial filter that is invariant to translation, rotation, and, in particular, zoom. Lantuejoul [94] introduced a mathematical morphological approach for skeletonization using influence zones. Maragos and Schafer [116] used mathematical morphological set operations to transform a discrete binary image using parts of its skeleton containing complete information about its shape and size. See [117,118,170] for early works on mathematical morphological approaches to homotopic thinning in continuous and discrete spaces. Skeletal subsets produced by such methods are dependent on structure elements used for mathematical morphology operations. Also, the resulting skeletons may not preserve connectedness.

It may be noted that most digital approaches to skeletonization use the object representations in pixel (in 2-D) or voxel (in 3-D) grids. A major drawback with such skeletonization algorithms is that these methods may not guarantee single-pixel (or voxel) thin skeletons for all objects, especially, at busy junctions; see Fig. 1. In other words, if the object of Fig. 1 or its 3-D version is used as an input, pixel- or voxel-based methods may not remove any pixel or voxel and thus fails to produce a one-pixel thin skeleton. Note that the example of Fig. 1 may be modified to increase the size of the central blobby region. A more complex, but also richer, approach to the object representation is using simplicial or cubical complex frameworks [40,47,50,54,88,98]. The above problem disappears when skeletonizing objects in these frameworks [50,54,98]. In the rest of this section, a brief survey of skeletonization algorithms for each of the above three categories is presented.

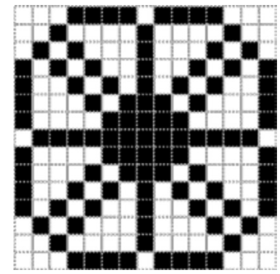

Fig. 1. An example of a busy junction of digital lines forming a diamond-shaped region in 2-D, which may not be thinned any further as the removal of any pixel from the object alters its topology. Note that it is possible to generalize a similar example in 3-D where the junction forms a volume that cannot be thinned.

2.1. Continuous geometric approaches

Several algorithms [36,128,129] focus on geometric properties of Blum's medial symmetry axis to locate the skeleton of an object. These methods are generally applied on a mesh representation of the object, or on a point-cloud generated by sampling the object boundary. One popular approach under this category is based on the principle of the Voronoi diagram [3,36,125,128,129,185]. The Voronoi skeleton of a polygonal shape is obtained by computing the Voronoi diagram of its boundary vertices and then taking its intersection with the polygonal shape. It may be noted that each additional vertex on the polygon adds a new skeletal branch. Thus, a proper polygonal approximation of a shape is crucial to generate the desired complexity of the skeleton. On the other hand, an accurate polygonal representation of a shape requires a large number of vertices. Thus, in general, Voronoi skeletonization produces a large number of spurious skeletal branches that are not essential for overall representation of the shape. Ogniewicz and Ilg [128] observed that *the skeletal segments, which lie deeply inside the polygonal shape are less sensitive to small changes on the boundary*. Such segments are essential for the description of the global topology and geometry of an object. Based on this observation, they derived different *residual functions* and used those to differentiate spurious branches from those essential to represent the object topology and geometry. Schmitt [167] proved that, as the number of generating boundary points increases, the *Voronoi diagram converges in the limit to the continuous medial locus*, with the exception of the edges generated by neighboring pairs of boundary points. Later, Voronoi skeletonization was generalized for 3-D polyhedral solids [3,11,52,172]. Amenta et al. [3] characterized *inner* and *outer Voronoi balls* for a set of boundary sample points to reconstruct its power crust, an approximation of a polyhedral boundary, and to compute its Voronoi skeleton. Jalba et al. [80] developed a *GPU-based efficient framework* for extracting surface and curve skeletons from large meshes. Bucksch and Lindenbergh [38] presented a *graph-based approach* to extract the skeletal tree from point clouds using collapsing and merging procedures in octree-graphs. This approach offers a computationally efficient solution for computing skeletons from point clouds that is robust to varying point density and the complexity of the skeleton may be adjusted by varying the size of the octree cell.