Segment- and Arc-Based Vectorizations by Multi-scale/Irregular Tangential Covering

Antoine Vacavant$^{1(✉)}$, Bertrand Kerautret$^{2}$, and Fabien Feschet$^{3}$

$^{1}$ Université Clermont Auvergne, CNRS, SIGMA Clermont, Institut Pascal, 63000 Clermont-Ferrand, France 
antoine.vacavant@uca.fr

$^{2}$ Univ Lyon, Lyon 2, LIRIS, 69676 Lyon, France 
bertrand.kerautret@univ-lyon2.fr

$^{3}$ Université Clermont Auvergne, CNRS, ENSMSE, LIMOS, 63000 Clermont-Ferrand, France 
fabien.feschet@u-auvergne.fr

Abstract. In this paper, we propose an original manner to employ a tangential cover algorithm - minDSS - in order to vectorize noisy digital contours. To do so, we exploit the representation of graphical objects by maximal primitives we have introduced in previous work. By calculating multi-scale and irregular isothetic representations of the contour, we obtained 1-D (one-dimensional) intervals, and achieved afterwards a decomposition into maximal line segments or circular arcs. By adapting minDSS to this sparse and irregular data of 1-D intervals supporting the maximal primitives, we are now able to reconstruct the input noisy objects into cyclic contours made of lines or arcs with a minimal number of primitives. We explain our novel complete pipeline in this work, and present its experimental evaluation by considering both synthetic and real image data.

Keywords: Vectorization · Noisy contours · Multi-scale analysis · Irregular grids · Tangential cover

1 Introduction

Image vectorization (or raster-to-vector) process has been a scientific and technical challenge since the 60’s [14] and still stimulates the development of modern approaches with focus on different points like drawing vectorization [1,6], Deep learning based method [4,12], perception based clip-art vectorization [3], from geometric analysis [11], or with real time constraint [27]. Image noise is a central problem in this complex task. We can employ denoising algorithms to reduce this alteration as a pre-processing step [13]. Furthermore, segmentation methods can integrate smoothing terms, to control the behavior of active contours for instance [8,26]. Even if we can reach to smooth contours by such approaches, a
lot of parameters must be tuned finely, depending on the input noise, and execution times can be very long. Modern deep learning approaches requires large datasets to be tuned finely, for a specific application (sometimes, like for pixel art vectorization, the reference does not exist).

As a consequence, another strategy consists in calculating geometrical representations of noisy image objects, even if simple pre-processing and segmentation methods are employed first. Researches employing digital geometry notions and algorithms have mainly dealt with the reconstruction of graphical objects into line segments or circular arcs, originally by considering noise-free contours [2,19], later by processing noisy data [15,17,18].

Following this literature of digital geometrical-based approaches, in this article, we propose a complete pipeline based on our previous contribution [22] in order to vectorize noisy digital contours into line segments or circular arcs. This related work aimed at representing such contours with maximal primitives, by processing 1-D (one-dimensional) intervals obtained by a multi-scale and irregular approach, summarized in Fig. 1.

From this representation, we now propose to use the tangential cover algorithm minDSS introduced in [7] to build a representative vectorization composed of either lines or arcs. Originally, minDSS problem is the determination of the minimal length polygonalization (in term of input intervals) of a maximal primitive representation of a given regular eight-connected curve [7]. In this article, we adapt this approach by using sparse irregular 1-D intervals as inputs.

The article is organized as follows. Section 2 relates synthetically on our previous contribution devoted to represent noisy digital contours into maximal primitives (lines or arcs) [22]. Then, Sect. 3 is dedicated to our novel approach that consists in adapting tangential cover algorithm for our specific contour representation, in order to vectorize input objects into cyclic geometrical structures. Section 4 presents the experimental results we have obtained for synthetic and real images, and a comparison with related works by considering robustness against contour noise. Finally, Sect. 5 concludes the paper with different axes of progress.
2 Reconstructions into Maximal Primitives

2.1 Multi-scale Noise Detection

We automatically detect the amount of noise present on a digital structure by the algorithm detailed in [9]. From such a multi-scale analysis, the proposed algorithm consists in constructing, for each contour point, a multi-scale profile defined by the segment length of all segments covering the point for larger and larger grid sizes. From each profile, the noise level is determined by the first scale for which the slope of the profile is decreasing. This noise level can be represented as boxes and as exposed in Fig. 2, a high noise in the contour will lead to a large box, and vice-versa. The algorithm can be tested on-line from any digital contour given by a netizen [10].

![Fig. 2](image)

**Fig. 2.** Examples of multi-scale noise detection upon a synthetic binary image (a) and a segmentation obtained from a real image (b).

Also, Fig. 2 illustrates the possible noise altering image objects. In (a), a synthetic circle is generating for testing our approach, while in (b) the segmentation by a thresholding process leads to a noisy contour.

2.2 Irregular Isothetic Cyclic Representation

The multi-scale contour representation calculated earlier is converted into irregular isothetic objects, defined on the following generalized grid model [21] restricted to the 2-D case:

**Definition 1 (d-D \( I \)-grid).** Let \( P \) be a hyper-rectangular subset of \( \mathbb{R}^d \). A d-D \( I \)-grid denoted by \( G \) is a tiling of \( P \) with rectangular hyper-parallelepipeds (or
cells), which do not overlap, and whose faces of dimension $d - 1$ are parallel to successive axes of the chosen space. A cell $R$ of $G$ is defined by a central point \( p_R = (p^1_R, p^2_R, \ldots, p^d_R) \in \mathbb{R}^d \) and a size along each axis \( l^d_R = (l^1_R, l^2_R, \ldots, l^d_R) \in \mathbb{R}^+ \).

As illustrated in Fig. 3, the input meaningful boxes are converted into $k$-curves, formalized as:

**Definition 2 (k-curve).** Let $A = (R_i)_{1 \leq i \leq n}$ be a path from $R_1$ to $R_n$. $A$ is a $k$-curve iff each cell $R_i$ of $A$ has exactly two $k$-adjacent cells in $A$.

In this definition, $k$-adjacency in 2-D is given by:

**Definition 3 (k-adjacency in 2-D).** Let $R_1$ and $R_2$ be two cells of a 2-D $I$-grid $G$. $R_1$ and $R_2$ are ve-adjacent ("vertex and edge" adjacent) if:

$$
\begin{align*}
\text{or} \quad & \left| x_{R_1} - x_{R_2} \right| = \frac{l^x_{R_1} + l^x_{R_2}}{2} \quad \text{and} \quad \left| y_{R_1} - y_{R_2} \right| \leq \frac{l^y_{R_1} + l^y_{R_2}}{2} \\
& \left| y_{R_1} - y_{R_2} \right| = \frac{l^y_{R_1} + l^y_{R_2}}{2} \quad \text{and} \quad \left| x_{R_1} - x_{R_2} \right| \leq \frac{l^x_{R_1} + l^x_{R_2}}{2}
\end{align*}
$$

(1)

$R_1$ and $R_2$ are e-adjacent ("edge" adjacent) if we consider an exclusive "or" and strict inequalities in this definition.

In this article, we consider $k = ve$ wlog. Adjacency is defined as the spatial relationship between edges and vertices of two cells and may be compared to classic 4- and 8-adjacencies of regular square grids [21].

![Fig. 3](image)

**Fig. 3.** Encoding of irregular isothetic cells into $k$-curves, following 2 order relations ($\preceq^L$ and $\preceq^T$), with interfaces stored inside the lists $S_X$ and $S_Y$. We then obtain a set of horizontal and vertical segments representing the input shape. In $S_{XY}$, internal vertices are white, external ones are black.

Moreover, as presented in Fig. 3, we have proposed to calculate two $k$-curves, one for each axis, following the order relations $\preceq^L (X \text{ axis})$ and $\preceq^T (Y \text{ axis})$ [21].
Then, both curves are combined to obtain 1-D $X$- and $Y$- aligned intervals stored inside the $S_X$ and $S_Y$ lists respectively. Finally, a single set $S_{XY}$ is constructed by merging them, with a special care to respect the curvilinear abscissa of the input contour when ordering the intervals in this list (see Fig. 3 (b)).

2.3 Recognition of Line Segments and Circular Arcs

From this 1-D interval representation, we then obtain maximal segments or arcs by employing a GLP (Generalized Linear Programming) approach. In this step, we consider the internal and external points of the segments into two different sets $P^o$ and $P^\bullet$. This algorithm has been adapted from the works [20, 25], which aim to solve the problem of minimal enclosing circle. In our case, its purpose is to obtain primitives enclosing a set of points (e.g. $P^o$) but not the other ($P^\bullet$). Figure 4 depicts the results obtained for two noisy digital contours, using maximal segments or arcs. More details can be found in [22].

![Maximal segments](image1)

(a) Maximal segments

![Maximal arcs](image2)

(b) Maximal arcs

Fig. 4. Reconstruction of digital shapes presented in Fig. 2 by maximal segments (a) and arcs (b). Each primitive is presented with a random color.

3 Adapted Tangential Covering

Using the results from the previous sections, at this step we have a collection of geometric primitives, either segments or circular arcs. This collection can be ordered according to the initial ordering of the constraints in $S_{XY}$. The ordered set of primitives is induced by the constraints in $S_{XY}$ such that its density is dependent on the density of the constraints. Hence, there is no reason to have a uniform density. So the density is not linked to the underlying curve of the constraints. This is different from the usual tangential cover. However,
the tangential cover can be viewed as a proper circular arcs graph. In other
word, the main structure of the tangential cover is an ordered set of overlapping
primitives corresponding to different portions of the underlying digital curve.
This is what we got from previous sections. More precisely, for two primitives,
we can build the overlapping predicate from the underlying intervals bounding
each recognized primitive by considering that the predicate is true when the
two primitives share constraints. The predicate is false otherwise. Hence for
two primitives, they overlap if and only if they share a set of constraints. As a
result, our structure can be embedded as a proper circular arcs graph. With this
structure, we can adapt the minDSS algorithm [7] to obtain the minimal length
cycle of the set of primitives since it is solely based on the graph structure of
the tangential cover. Length must be understood as the number of primitives
in a cycle. We now recall the minDSS algorithm in Sect. 3.1 and provides its
adaptation in Sect. 3.2.

3.1 The minDSS Algorithm

Let us recall the initial purpose of the minDSS algorithm by exposing its
input/output, following a question of A. Rosenfeld:

Input: An ordered list of geometric primitives, DSS (digital straight segments)
in [7] and an underlying curve.

Output: Shortest length cycle, in number of primitives, with worst case time
complexity $O(n)$.

The shortest length cycle problem could be trivially solved in quadratic time
complexity but remained a challenge in linear time complexity. The original
minDSS algorithm has two phases. In the first phase, for a DSS $D$, it defines the
residue of $D$ as the set of points of the underlying curve that only belongs to
$D$. The main property proved in this phase is the following [7]. If there exists a
primitive $D$ with non empty residue then the shortest cycle can be constructed
from any point inside the residue. This result is of practical importance since
digital curves having not all empty residues are easily encountered in practice.
The first phase of minDSS is totally parallel. However, there exists digital curves
whose residues are all empty. Hence, the first phase is not sufficient to solve the
problem.

In a second phase, a propagation of marks is done in the tangential cover to
detect a decision case. The goal is to detect locally two cycles having different
lengths. Since the length of any cycle is either the minimum length or the mini-
mum length plus one, detecting such cycles is sufficient to solve minDSS. The
propagation of marks is done in the connectivity graph of the primitives using
only an overlapping predicate. Let us denote the primitives by $\{T_k\}_{k=0,n-1}$ with
an arbitrary first element $T_0$. A cyclic ordering is used that is the next primi-
tives following $T_k$ is $T_{(k+1) \mod n}$ and the previous primitive is $T_{(k-1) \mod n}$. By
iterations using next and previous, we can move along the set of primitives by
using iterates as $next^{(k)} = next(next^{(k-1)})$ and $next^{(0)} = Id$ and obviously the
same for previous.
For a primitive $T_k$, we can define the forward function $f(T_k)$ such that $f(T_k) = \text{next}^{(j)}(T_k)$ with $T_k$ and $f(T_k)$ overlapping and $j$ maximal. It is a farthest forward move with overlapping. Since $T_k$ overlaps itself, this is well defined. By convention, we define $f^{-1}()$ similarly but using previous instead of next.

By construction $f^{-1}(f(T_k))$ is a primitive that overlaps $f(T_k)$. In a wide sense, it is a primitive before $T_k$ in the cyclic ordering, that is using previous starting at $f(T_k)$. We can remark that the interval $[f^{-1}(f(T_k)); f(T_k)]$ is a separator of all cycles in the graph. Indeed, each cycle must contain a primitive in this interval. As a consequence, propagation of marks in minDSS can be done only for primitives inside a separator so to decide locally the shortest length cycle problem. The complexity of the marking is proved to be $O(n)$ [7].

3.2 Adaptation of minDSS

In our approach, there is no underlying curve, that is, even if $S_{XY}$ is a support set, it is not a curve. For instance, its density is not constant and is different from one as in the regular grid. Furthermore, this density can vary on different parts of the curve. However, it has a global ordering obtained from the isothetic cyclic representation. The indices of the constraints in $S_{XY}$ permit to define any primitive as a succession of intervals in $S_{XY}$. Hence, we can define a connectivity graph on the set of recognized primitives using the index of any element in $S_{XY}$. Connectivity is well defined and so we can calculate $f()$ and $f^{-1}()$. Moreover, the propagations calculated in minDSS can be done on the new connectivity graph. To do so, it is sufficient to recreate the connectivity graph by using our new overlapping predicate. Two successive primitives in $S_{XY}$ that overlap with respect to the vertical or horizontal intervals are connected in the graph. The forward function only uses this connectivity and next and previous to compute the separator. Moreover, the propagation of marks is done according to the connectivity graph only and can be thus applied with no modifications. We refer to the original paper [7] for details about propagation of marks.

Of course, minimizing the number of primitives, while having variable density, should be analyzed carefully. Indeed, minimizing the number of primitives led to different length steps in the underlying curve because length is directly linked with primitives’ density. Hence, at the portions of the underlying curve with high density, using the primitives induced small movements while in the portions with low density changing from one primitive to the next one, can produce large movements in the curve. The fact that minDSS is actually a graph-based algorithm makes it independent of the embedding of the graph onto the underlying curve.

The output of the minDSS algorithm is a cycle in the set of intervals using the link between intervals made by the existence of a geometric primitive intersecting all of them. Hence, the output of minDSS is a list of intervals and not a set of primitives. As seen in Fig. 5, for segments as well as for arcs, from a list of intervals, there exists an infinite number of primitives corresponding to the given intervals. So, after the computation of the solution of minDSS, a geometric
solution must be reconstructed by choosing primitives for all edges in the graph solution given by minDSS. There are several ways to do that and a fast and simple solution is obtained by using the middle points of the intervals (see Fig. 5). Of course, the consequence of such a choice is that some intervals might not be intersected by the primitives and this happens when the extremal points of an interval is a must use points for defining the original solution of compatibility between intervals (see for instance the red intervals of Fig. 5). Since there are several solutions for minDSS, each graph solution leads to a different geometric solution.

4 Experimental Results

4.1 Global Overview of the Method

We first propose to expose the different steps of our pipeline for a low resolution fish binary image leading to a noisy shape (Fig. 6-a), from the binary shape data sets\(^1\). We depict the meaningful scales (b), the set of intervals \(S_{XY}\) (c), the collections of maximal segments and arcs and the associated tangential cover outcomes (d,e).

Thanks to our system, we can obtain faithful vectorizations of such digital contours, with a low number of primitives.

\(^1\) http://vision.lem.s.brown.edu/content/available-software-and-databases.
More precisely, when we consider segments, we can observe that straight parts of the shape are represented with line segments (*e.g.* see the bottom part), since meaningful scales have a very fine resolution, and few intervals are thus required. More complex parts have been simplified faithfully (*as triangular shapes for fish fins*).

When we select arcs instead, roundy parts of the shape are individually represented by circular primitives. Even if some regions are more smoothed (*e.g.* see fish fins at top-left), these are relevant approximations for such a low resolution image (with no visual aberrant inverted arcs). Furthermore, it should be recalled that tangential cover computes one solution, and other options could be of better interest, depending on the final application.

### 4.2 Visual Inspection of Results with Synthetic Images

In Fig. 7, two synthetic noisy objects, with polygonal and curved shapes, are respectively vectorized with segments and arcs.

In (a), a polygon image has been corrupted with different noises, to study the impact of variable perturbations into a single sample. We can notice that the final segment-based reconstruction is very stable and represents faithfully the underlying object.

In (b), a curved shape is corrupted with local noises. Our algorithm can produce a relevant arc-based reconstruction, with a low number of primitives. As a whole, even in the presence of severe (possibly only local) perturbation, we can calculate reconstructions geometrically close to the underlying contours.
4.3 Visual Inspection of Results with a Real Image

Finally, in Fig. 8, we have selected a large image composed of two contours, processed independently by our approach. The external elliptical contour is vectorized with arcs, while the internal one is reconstructed with segments.

We can see from the meaningful scales computed (a) that some parts of the contours are very finely represented (e.g. see top-left part of the external part), which leads to small maximal primitives in (b) and in the tangential cover (c). On the contrary, noisy regions are reconstructed with longer primitives (e.g. see straight lines of the internal part).

Even if the input contours are long (in term of number of pixels) tangential cover leads to reconstruction with few primitives.

![Segment-based reconstruction of a polygonal shape](image1)

![Arc-based reconstruction of a curved shape](image2)

Fig. 7. The reconstructions obtained for 2 synthetic images

5 Conclusion and Future Works

We have proposed a novel framework to vectorize possibly noisy digital contours by line segments or circular arcs, by combining our previous work devoted to reconstruct maximal primitives [22] and the tangential cover algorithm.
minDSS [7]. By means of the experiments we have exposed, that this method achieves faithful vectorization according to the underlying object contour. Moreover, by employing minDSS, we ensure that the number of primitives is low.

Our first research lead is to compare our method with other classic and modern vectorization approaches, by considering various quality measures as we did in [23] (Hausdorff distance, error of tangents’ orientation, Euclidean distance between points). We would like to show that our pipeline permits to obtain robust reconstructions (according to the input contour noise), with respect to the state-of-the-art.

![Visual evaluation of our pipeline with a large image](image.png)

**Fig. 8.** Visual evaluation of our pipeline with a large image

Another important concern is to calculate a single reconstruction combining segments and arcs. To do so, we would like to propose a combinatorial optimization process as developed in [5]. Such scheme would select the best primitive to be integrated in the final solution, depending on a given predicate, e.g. the geometrical length of the primitive. An alternative option is to first compute a set of maximal primitives containing both types, by considering the straight and curved part of the input noisy shape, as we employed already in [23].
Finally, we plan to compare the future reconstructions with other approaches able to choose the primitive automatically like the method proposed by Nguyen and Debled [16] and to explore other applications like the one related to 3-D printing [24].

References

1. Bessmeltsev, M., Solomon, J.: Vectorization of line drawings via polyvector fields. ACM Trans. Graph. 38(1), 12 p. (2019). https://doi.org/10.1145/3202661
2. Damaschke, P.: The linear time recognition of digital arcs. Pattern Recogn. Lett. 16, 543–548 (1995)
3. Dominici, E.A., Schertler, N., Griffin, J., Hoshyari, S., Sigal, L., Sheffer, A.: Poly-Fit: perception-aligned vectorization of raster clip-art via intermediate polygonal fitting. ACM Trans. Graph. (TOG) 39(4), 77-1 (2020)
4. Egiazarian, V., et al.: Deep vectorization of technical drawings. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12358, pp. 582–598. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58601-0_35
5. Faure, A., Feschet, F.: Multi-primitive analysis of digital curves. In: Wiederhold, P., Barneva, R.P. (eds.) IWCA 2009. LNCS, vol. 5852, pp. 30–42. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-10210-3_3
6. Favreau, J.D., Lafarge, F., Bousseau, A.: Fidelity vs. simplicity: a global approach to line drawing vectorization. ACM Trans. Graph. 35(4), 120:1–120:10 (2016)
7. Feschet, F., Tougue, L.: On the min DSS problem of closed discrete curves. Discret. Appl. Math. 151(1), 138–153 (2005)
8. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: active contour models. Int. J. Comput. Vision 1(4), 321–331 (1988). https://doi.org/10.1007/BF00133570
9. Kerautret, B., Lachaud, J.: Meaningful scales detection along digital contours for unsupervised local noise estimation. IEEE Trans. Pattern Anal. Mach. Intell. 34(12), 2379–2392 (2012)
10. Kerautret, B., Lachaud, J.: Meaningful scales detection: an unsupervised noise detection algorithm for digital contours. Image Process. On Line 4, 98–115 (2014)
11. Kerautret, B., Ngo, P., Kenmochi, Y., Vacavant, A.: Greyscale image vectorization from geometric digital contour representations. In: Kropatsch, W.G., Artner, N.M., Janusch, I. (eds.) DGCI 2017. LNCS, vol. 10502, pp. 319–331. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66272-5_26
12. Kim, B., Wang, O., Öztireli, A., Gross, M.: Semantic segmentation for line drawing vectorization using neural networks. In: Computer Graphics Forum, vol. 37, no. 2, pp. 329–338. Wiley Online Library (2018)
13. Lebrun, M., Colom, M., Buades, A., Morel, J.: Secrets of image denoising cuisine. Acta Numerica 21, 475–576 (2012)
14. Montanari, U.: Continuous skeletons from digitized images. J. ACM 16(4), 534–549 (1969)
15. Nguyen, T.P., Debled-Rennesson, I.: Arc segmentation in linear time. In: Real, P., Diaz-Pernil, D., Molina-Abril, H., Berciano, A., Kropatsch, W. (eds.) CAIP 2011. LNCS, vol. 6854, pp. 84–92. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-23672-3_11
16. Nguyen, T.P., Debled-Rennesson, I.: Decomposition of a curve into arcs and line segments based on dominant point detection. In: Heyden, A., Kahl, F. (eds.) SCIA 2011. LNCS, vol. 6688, pp. 794–805. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21227-7_74
17. Nguyen, T.P., Kerautret, B., Debled-Rennesson, I., Lachaud, J.-O.: Unsupervised, fast and precise recognition of digital arcs in noisy images. In: Bolc, L., Tadeusiewicz, R., Chmielewski, L.J., Wojciechowski, K. (eds.) ICCVG 2010. LNCS, vol. 6374, pp. 59–68. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15910-7_7

18. Rodríguez, M., Largeteau-Skapin, G., Andres, E.: Adaptive pixel resizing for multiscale recognition and reconstruction. In: Wiederhold, P., Barneva, R.P. (eds.) IWCLA 2009. LNCS, vol. 5852, pp. 252–265. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-10210-3_20

19. Rosin, P., West, G.: Nonparametric segmentation of curves into various representations. IEEE Trans. Pattern Anal. Mach. Intell. 17(12), 1140–1153 (1995)

20. Sharir, M., Welzl, E.: A combinatorial bound for linear programming and related problems. In: Finkel, A., Jantzen, M. (eds.) STACS 1992. LNCS, vol. 577, pp. 567–579. Springer, Heidelberg (1992). https://doi.org/10.1007/3-540-55210-3_213

21. Vacavant, A.: Robust image processing: definition, algorithms and evaluation. Habilitation, Université Clermont Auvergne (2018)

22. Vacavant, A., Kerautret, B., Roussillon, T., Feschet, F.: Reconstructions of noisy digital contours with maximal primitives based on multi-scale/irregular geometric representation and generalized linear programming. In: Kropatsch, W.G., Artner, N.M., Janusch, I. (eds.) DGCI 2017. LNCS, vol. 10502, pp. 291–303. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66272-5_24

23. Vacavant, A., Roussillon, T., Kerautret, B., Lachaud, J.: A combined multi-scale/irregular algorithm for the vectorization of noisy digital contours. Comput. Vis. Image Underst. 117(4), 438–450 (2013)

24. Valdivieso, H.: Polyline defined NC trajectories parametrization. A compact analysis and solution focused on 3D printing. arXiv preprint arXiv:1808.01831 (2018)

25. Welzl, E.: Smallest enclosing disks (balls and ellipsoids). In: Maurer, H. (ed.) New Results and New Trends in Computer Science. LNCS, vol. 555, pp. 359–370. Springer, Heidelberg (1991). https://doi.org/10.1007/BFb0038202

26. Wirjadi, O.: Survey of 3D image segmentation methods. Technical report, Berichte des Fraunhofer ITWM (2007)

27. Xiong, X., Feng, J., Zhou, B.: Real-time contour image vectorization on GPU. In: Braz, J., et al. (eds.) VISIGRAPP 2016. CCIS, vol. 693, pp. 35–50. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-64870-5_2