HAL open science

Distance Transformation, Reverse Distance Transformation and Discrete Medial Axis on Toric Spaces

David Coeurjolly

▶ **To cite this version:**

David Coeurjolly. Distance Transformation, Reverse Distance Transformation and Discrete Medial Axis on Toric Spaces. International Conference on Pattern Recognition, Dec 2008, Tampa, United States. pp.3541. hal-00350154

HAL Id: hal-00350154

https://hal.science/hal-00350154v1

Submitted on 6 Jan 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distance Transformation, Reverse Distance Transformation and Discrete Medial Axis on Toric Spaces

David Coeurjolly
LIRIS, UMR CNRS 5205, Université Claude Bernard Lyon 1,
F-69622 Villeurbanne, France

Abstract

In this paper, we present optimal in time algorithms to compute the distance transform, the reverse distance transform and the discrete medial axis on digital objects embedded on $n-$dimensional toric spaces.

1. Introduction

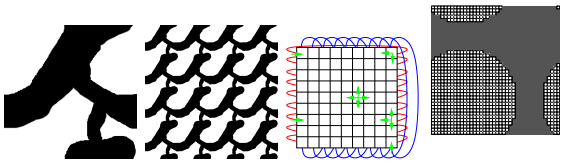
In binary images, the distance transformation (DT) and the geometrical skeleton extraction are classic tools for shape analysis [4]. The distance transformation consists in labeling each pixel of an object with the distance to the closest pixel of its complement (also called the *background*). If we consider a set of pixels labeled with a distance, the reverse distance transformation problem (RDT) consists in reconstructing the binary shape obtained as the union of all discs centered in the pixels and with the distances as radii. If the underlying distance is the Euclidean one, we can consider the SEDT (Squared Euclidean Distance Transformation) and the REDT (Reverse Euclidean Distance Transformation) see [3] for a survey. The medial axis is a usual and convenient representation for shape description or recognition purposes [1]. In the following we focus on the discrete medial axis (DMA) with Euclidean metric. In the digital space, we have many efficient algorithms to compute such transformations [3]. The aim of this paper is to generalize such techniques to toric spaces which is a widely used model in the material analysis field [2].The main idea is to perform measurements on a material sample under the hypothesis that the overall material is composed of a regular tiling of the sample (see Fig.1-(a,b)). Thus, in order to make the measurements consistent through tiling, we have to consider that the sample in embedded in a toric space. The main contributions are n-dimensional, error-free and optimal in time algorithms for the SEDT, REDT and DMA on toric spaces.

2. Discrete Toric Spaces

In this section, we consider notations proposed in [2]: let $\mathbb{Z}_d$ (with $d \in \mathbb{N}$) denotes the set $\{0, 1, \ldots, d-1\}$ and $\oplus_d$ be the sum operator on integers modulo d: $a \oplus_d b = (a+b)$ modulo d. Note that $(\mathbb{Z}_d, \oplus_d)$ forms a cyclic group and can be considered as a 1-D toric space. In dimension n, given $(d_1, \ldots, d_n) \in \mathbb{N}^n$, the direct product $\mathbb{T}_n = \mathbb{Z}_{d_1} \times \ldots \times \mathbb{Z}_{d_n}$ is an *n-dimensional discrete toric space* [2]. On this space, we can define the $\oplus$ operator as the composition of the dimensional $\oplus_{d_i}$ operators. A *toric image* is a mapping which associate a value to each discrete point on a toric space. We can define several n-dimensional adjacency relations [2], in the following, we only consider a simple $1-$adjacency: two point $p, q \in \mathbb{T}_n$ are $1-$adjacent iff there is a vector $s = (0, \ldots, s_i, \ldots, 0)$ on length n with $s_i = \pm 1$ such that $p \oplus s = q$. The 1-adjacency is equivalent to the 4-connectivity in 2-D and to the 6-connectivity in 3-D. In the following, a discrete object on a toric space $\mathbb{T}_n$ is a set of 1-adjacent grid points of $\mathbb{T}_n$. Illustrations of a toric space and a toric object in dimension 2 are provided in Fig. 1.

3. Separable Algorithms for the SEDT, REDT and DMA

In this section, we first overview the separable algorithms to compute the SEDT, the REDT and the DMA of a discrete object in the classical $\mathbb{Z}^n$ grid. [3]. Let us first consider the SEDT algorithm in the 2D case: given a two-dimensional binary object P in a $d_1 \times d_2$ image, $\bar{P}$ denotes the complementary of



(a) (b) (c) (d)

Figure 1. (a) **a sample on a toric image,** (b) **sample tiling,** (c) **1-adjacency illustration in 2-D, and** (d) **example of a disc on a toric image.**

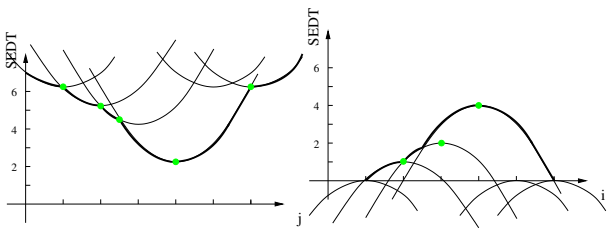


Figure 2. Lower and upper envelope computations in SEDT and the REDT problems.

P, *i.e.* the set of background pixels. The output of the algorithm is a 2D image H storing the squared distance transform. The SEDT algorithm consists of the following steps: first, build from the source image P, a one-dimensional SEDT according to the first dimension ($x-$axis) denoted by $G = \{g(i,j)\}$, where, for a given row j:

$$g(i,j) = \min_x \{(i-x)^2;\ 0 \leq x < d_1 \text{ and } (x,j) \in \bar{P}\}\,. \tag{1}$$

Then, construct the image $H = \{h(i,j)\}$ with a $y-$axis process:

$$h(i,j) = \min_y \{g(i,y) + (j-y)^2;\ 0 \leq y < d_2\}\,. \tag{2}$$

To compute the first step of the SEDT, we perform a two-scan of each image row independently and obtain process in $O(d_1 \cdot d_2)$. To solve the second step, we can first observe that Eq. (2) corresponds to a lower envelope computation of the set of parabolas $F_y^i(j) = g(i,y)^2 + (j-y)^2$, independently column by column. To compute such a set, a stack based technique has been proposed leading to a computational cost in $O(d_1 \cdot d_2)$ (see [3] for a complete bibliography and Fig. 2$-left$). Note that Eq. (1) can also be interpreted in terms of parabolas: given the set of parabolas $F_x^j(i) = (i-x)^2$ for $(x,j) \in \bar{P}$, $g(i,j)$ can be computed as the lower envelope of this set. If no background pixels exist in the ith row, the distance values of $g(i,j)$ are set to $+\infty$ (and thus propagated through the dimensions with adapted arithmetical operators).

REDT and the DMA can also be decomposed into separable upper envelope computations, similarly to Eq. (2) [3]. Indeed, given a set of discs $L = \{x_k, y_k, r_k\}$ with centers (x_k, y_k) and radii r_k, the REDT consists of extracting the set of grid points P such that

$$P = \{(i,j) \mid (i-x)^2 + (j-y)^2 < r_k^2,\ (x_k, y_k, r_k) \in L\}\,. \tag{3}$$

Let $F = \{f(i,j)\}$ be a picture of size $d_1 \times d_2$ such that $f(i,j)$ is set to $r(i,j)^2$ if (i,j) belongs to L and 0 otherwise. Hence, if we compute the map $H = \{h(i,j)\}$ such that

$$\begin{aligned} h(i,j) = \max\{&f(x,y) - (i-x)^2 - (j-y)^2; \\ &0 \leq x < d_1, 0 \leq y < d_2 \text{ and } (x,y) \in F\}\,, \end{aligned} \tag{4}$$

we obtain P by extracting from H all pixels of strictly positive values. So, to build H from F, we can decompose the computation into two one-dimensional steps: first, build from the image F the picture $G = \{g(i,j)\}$ such that

$$g(i,j) = \max_x \{f(x,j) - (i-x)^2, 0 \leq x < d_1\}\,. \tag{5}$$

Then define from G the picture H such that

$$h(i,j) = \max_y \{g(i,y) - (j-y)^2, 0 \leq y < d_2\}\,. \tag{6}$$

Once again, both processes require a computation of the upper envelope of a set of d parabolas that can be done in $O(d)$ per line or column (see Fig. 2$-right$) [3]. Hence, in dimension 2, the overall computational cost is in $O(d_1 \cdot d_2)$, which is optimal.

Concerning the Discrete Medial Axis computation, we have proven in [3] that the DMA can be extracted during the REDT process with exactly the same computational cost.

We do not go further into details but just focus on the advantages of such algorithms: first the computations are exact since we use an error free computation with the Euclidean metric. Then, all these algorithms are optimal in time $O(d_1 \cdot \ldots \cdot d_n)$ for a d-dimensional image and can be extended to higher dimensions since the internal processes are separable. Finally, all tools are based on the same 1-D algorithmic tool: the upper/lower envelope computation of a set of parabolas. In the following sections, we extend all these algorithms to toric spaces

The main problem we have to consider in the toric spaces is the information propagation between the domain borders, otherwise, values may be incorrect (see Fig. 5). Due to the separability of the previous algorithms, the problem can be simplified since, at each step, we have a 1-D process and thus a 1-D propagation. For SEDT algorithm in dimension 2, during the first step, we just have to take into consideration propagations along the $x-$axis (Eq. (1)). Similarly, only propagations along the $y-$axis may occur during the second step. If we consider now the 1-D processes, for each algorithm, we have to perform a upper (resp. lower) envelope computation of a given set of parabolas. over 1-D cyclic group $(\mathbb{Z}_d, \oplus_d)$.

4. Lower/Upper Parabola Envelope Computation on 1-D Toric Image

For the sake of clarity, we focus in the following on the lower envelope computation problem. Results for the upper envelope computation are similar. As illustrated in Fig. 3, let us consider a set $\mathcal{F}$ of d parabolas $\{(a_l, h_l)\}_{l=0\ldots d-1}$ such that $F_l : x \rightarrow h_l + (a_l - x)^2$ and $a_i \neq a_j$ for all i and j. The point (a_l, h_l) is thus the apex of the parabola F_l (dots in Fig. 3). To compute the lower envelope on a cyclic group, the main idea is to find the index $k \in \{0, \ldots, d-1\}$, so-called *break index* in the following, which allows us to break the cycle and thus use the classical lower envelope computation algorithm to obtain the result.

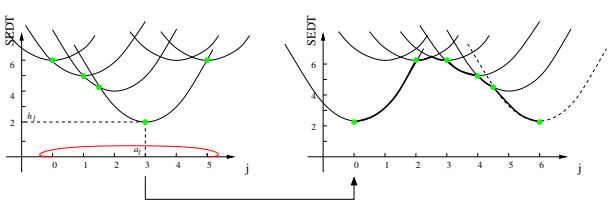


Figure 3. The cycle unfolding along the break index (3).

Lemma 1 *A break index for the toric lower (resp. upper) parabola envelope computation is obtained by taking the abscissa a_l of a parabola F_l with minimum (resp. maximum) height h_l.*

PROOF: first of all, let us consider two parabolas $F_1(x) = h_1 + (a_1 - x)^2$ and $F_2(x) = h_2 + (a_2 - x)^2$ such that $a_1 < a_2$ and such that $h_2 < h_1$. In other words, F_2 is the parabola with minimum height of $\{F_1, F_2\}$. Since the intersection between any two parabolas of $\mathcal{F}$ is reduced to a point, we denote $P_{ij}(x_{ij}, y_{ij})$ the intersection point between F_i and F_j. Since $h_2 < h_1$ and $a_1 < a_2$ then $x_{12} < a_2$. Hence, for $x > a_2$ we have $F_2(x) < F_1(x)$. In other words, the parabola F_1 does not interfere in the lower envelope computation of the right part of the F_2 parabola. Let us consider now the set $\mathcal{F}$ on a non-cyclic index group and suppose that F_l is the parabola with minimum height. We denote $\mathcal{F}_<$ (resp. $\mathcal{F}_>$) the parabolas F_i such that $a_i < a_l$ (resp. $a_i > a_l$). We can compute the set of parabolas belonging to the lower envelope of $\mathcal{F}$ as the union of the lower envelope of $\mathcal{F}_<$, the parabola F_l and the lower envelope of $\mathcal{F}_>$ (since F_l is the parabola with minimum height, it necessarily belongs to the lower envelope). With the above argument used on each pair F_i and F_l, we can conclude that the envelope computation on $\mathcal{F}_<$ and $\mathcal{F}_>$ are independent. If we consider now a cyclic index group for $\mathcal{F}$, the parabola F_l allows us to control the propagation on the envelope computation. Hence, if we consider the sequence of $d+1$ parabolas $\mathcal{F}' = \{F_l, F_{l\oplus 1}, \ldots, F_{l\oplus d-1}, F_l\}$ (the parabola F_l is duplicated in some sense). The cyclic lower envelope computation on $\mathcal{F}'$ is exactly the same as the non-cyclic lower envelope computation. Hence, l is the break index of $\mathcal{F}$ and allows us to use the classical lower envelope computation on the unfolded cycle.□

Hence, to compute the lower envelope of $\mathcal{F}$, the overall algorithm can be sketched as follows: first we identify the parabola with minimum height, we construct the set $\mathcal{F}'$ and then we use the classical algorithm. We conclude with the fact that the parabolas in the lower envelope computation of $\mathcal{F}'$ are also in the lower envelope of the cyclic group $\mathcal{F}$.

5. Application to SEDT, REDT and DMA problems

Considering the SEDT problem, Fig. 4 illustrates the overall 1-D process for the first step (Eq. (1)): in this case, the break index is the first background pixel found in during a forward scan. We reconstruct a $d+1$ vector, duplicating the first break index value at the end. We apply the classical 1-D SEDT algorithm on this vector and finally copy the values into the cyclic vector. Since all rows can be processed independently, the above procedure can be applied of every rows leading to a correct result thanks to Lemma 1. Concerning the sec-

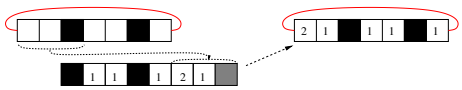


Figure 4. Illustration of the cycle unfolding for the first step of the SEDT computation.

ond step (Eq. (2)), we us the same principle with a break index defined as the index of the parabola with minimum height (obtained in linear time). Finally, thanks to Lemma 1 and to the separability if the problem, we obtain a correct SEDT algorithm on toric space that is linear in the number of grid points (see Fig. 5). Concerning the REDT and

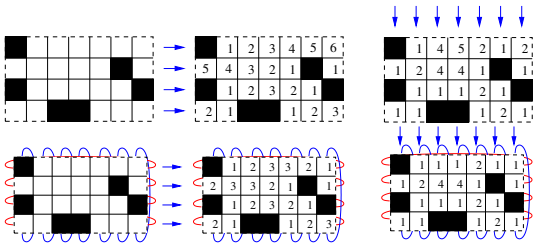
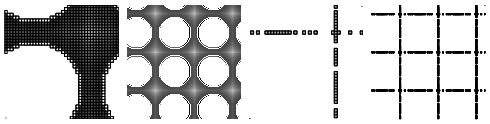


Figure 5. Overall SEDT computation on a classic space (top) and on a toric space (bottom).

the DMA problems, we have the same results using the parabola index with maximum height as break index. Finally, we also have optimal in time algorithms for the REDT and DMA problems in arbitrary dimension for toric spaces.

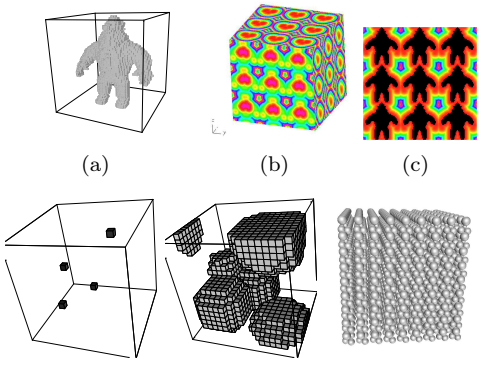
To illustrate these algorithms on toric spaces in dimension 2 and 3, Fig. 6 gives results of the toric SEDT and DMA in dimension 2 on the input object presented in Fig. 1-(d). Note that tiling illustrations allow the reader to make sure that the toric behaviors of the algorithms are visually correct. Finally, Fig. 7 presents results in dimension



(a) (b) (c) (d)

Figure 6. (a) **SEDT of the toric sample,** (b) **tiling of the sample,** (c) **DMA of the toric sample, and** (d) **tiling of the DMA.**

3 (SEDT and REDT).



(a) (b) (c)

(d) (e) (f)

Figure 7. Toric SEDT in dimension 3: (a) **input sample,** (c) **volumetric SEDT transformation computed on the sample and tiled,** (d) **represents a slice of the volumetric SEDT. REDT example:** (a) **an input set of discs,** (b) **a toric reconstruction, and** (c) **a tiling of the reconstruction.**

6. Conclusion

In this paper, we have presented a generalization to toric spaces of several tools widely used in shape analysis: the SEDT, the REDT and the DMA. More precisely, we have first identified the core procedure in all the optimal in time algorithms for these problems: the 1-D lower/upper envelope computation of a set of parabolas. We have solved this problem on toric spaces and thus obtained $n-$dimensional separable, optimal in time, algorithms for the SEDT, the REDT and the DMA problems on toric domains.

References

[1] H. Blum. A transformation for extracting descriptors of shape. In *Models for the Perception of Speech and Visual Forms*, pages 362–380. MIT Press, 1967.

[2] J. Chaussard, G. Bertrand, and M. Couprie. Characterizing and detecting toric loops in n-dimensional discrete toric spaces. In *14th DGCI*, volume 4992 of *LNCS*. Springer, 2008.

[3] D. Coeurjolly and A. Montanvert. Optimal separable algorithms to compute the reverse euclidean distance transformation and discrete medial axis in arbitrary dimension. *IEEE Trans. on PAMI*, 29(3):437, MAR 2007.

[4] A. Rosenfeld and J. L. Pfaltz. Sequential operations in digital picture processing. *Journal of the ACM*, 13(4):471–494, OCT 1966.