

# COMPUTATION OF NORMAL VECTORS OF DISCRETE 3D OBJECTS: APPLICATION TO NATURAL SNOW IMAGES FROM X-RAY TOMOGRAPHY

FREDERIC FLIN<sup>1</sup>, JEAN-BRUNO BRZOSKA<sup>1</sup>, BERNARD LESAFFRE<sup>1</sup>, CECILE COLEOU<sup>1</sup> AND  
PASCAL LAMBOLEY<sup>2</sup>

<sup>1</sup>Météo-France, Centre d'Etudes de la Neige, 1441 rue de la Piscine, F38406 St Martin d'Hères cedex, <sup>2</sup>Météo-France, SCEM/Prévi/Compas, 42 av. G. Coriolis, F31057 Toulouse cedex  
e-mail: First\_name.second\_name@meteo.fr  
(Accepted October 18, 2001)

## ABSTRACT

Estimating the normal vector field on the boundary of discrete 3D objects is essential for rendering and image measurement problems. Most of the existing algorithms do not provide an accurate determination of the normal vector field for shapes that present edges. We propose here a new and simple computational method to obtain accurate results on all types of shapes whatever their degree of local convexity. The presented method is based on the analysis of the gradient vector field of the distance map of the object. Some results on simulated data and snow images from X-ray tomography are presented and discussed.

Keywords: discrete geometry, distance map, normal vectors, snow.

## INTRODUCTION

With the development of modern imaging techniques like microtomography or MRI, scientists are faced with the problem of the interpretation of 3D numerical images.

The calculation of the normal vector field on the boundary of discrete 3D objects is essential for a good visualization. Moreover, the knowledge of the normal vector allows the determination of many interesting parameters to analyze the object.

For example, algorithms for the determination of local curvature (Brzoska *et al.*, 1999a) and surface area (Brzoska *et al.*, 2001) were recently developed at the Centre d'Etudes de la Neige. These two algorithms offer interesting outlooks for modeling snow metamorphism. Nevertheless, they require an accurate computation of the normal vector field on the surface boundary to be relevant.

The normal vector field can commonly be determined by triangulation (Lorenzen and Cline, 1987). This method is complex regarding to the accuracy of the obtained vector field. Therefore many imaging scientists developed fully discrete algorithms. These techniques provide good results on smooth surfaces such as sphere, torus and infinite plane surface (Lenoir *et al.*, 1996; Thürmer and Würthrich, 1997). However, they often do not provide an accurate

determination of the normal vector field for shapes that present edges (Papier and Françon, 1998). An interesting method (Tellier and Debled-Renesson, 1999) based on the tangential lines computation allows to obtain a realistic rendering on shapes that present edges. However, it does not seem very accurate when applied on small spheres.

We propose here a new and simple computational method to obtain accurate results on all types of shapes whatever their degree of local convexity. A discrete background distance map of the original 3D image is first constructed. For each voxel of the object, an elementary gradient vector of the distance map is then computed. The normal vector of a surface voxel is obtained by summation of each elementary gradient vector in an appropriate neighborhood. This neighborhood is determined for each surface voxel so that it does not contain any singularities of the gradient field. Some results obtained on real and simulated data are shown and commented.

## METHOD

Normal vectors are computed at the center of each voxel  $V_P$  of the surface  $S$  of an object  $O$ .

Definitions: in a binary image constituted of "0" and "1":

- $V_P \in O \Leftrightarrow V_P = 1$ .
- $V_P$  is a voxel belonging to the background  $\Leftrightarrow V_P \notin O$ .
- $V_P \in S \Leftrightarrow V_P$  has a 6-connected neighbor belonging to the background.

Our main idea for the computation of the normal vectors is to use the gradient vector field of the background distance map of the object.

## DISTANCE MAP

Let  $P$  and  $Q$  be the centers of the voxels  $V_P$  and  $V_Q$  and  $d(P, Q)$  the distance from  $P$  to  $Q$ .

In a numerical object  $O$ , the background distance  $d_b$  on a point  $P$  is defined as follows:

$$\forall (Q \notin O), d_b(P) = \min [d(P, Q)]. \quad (1)$$

For instance, the background distance map can be built using a two-run iterative chamfer algorithm (Borgefors, 1984; Chassery and Montanvert, 1991). In summary, this algorithm increments at each run the discrete distance to the closest already processed voxel of the object. Depending on the size of the used chamfer kernel, some distortion with respect to the Euclidean distance map (Danielsson, 1980; Yamada, 1984; Ragnemalm, 1989) may occur in deeper parts of the map; however, it is negligible near the surface (Rolland, 1991), *i.e.* on the first layers of voxels.

For our method, we used a second-rank chamfer distance  $d_{5.7.9.11}$  ( $d_5$  for short), which seems a good compromise between the fastness of the first-rank chamfer distance  $d_{3.4.5}$  and the accuracy of the Euclidean distance (Verwer, 1991; Borgefors, 2001).

## GRADIENT MAP

Let  $N(P)$  be the 3D first-rank neighborhood of  $P$  and  $M_i$  a point of  $N(P)$ .

Let be  $M_{j_i} \in N(P) / j(M_i) = j(P) - 1$  and  $M'_{j_i} \in N(P) / j(M_i) = j(P) + 1$ , where  $j = x, y, z$  are canonical coordinates. For each current voxel  $V_P$  of  $O$ , the components of the gradient vector in  $P$  are computed as follows:

$$\text{grad}_j(P) = \sum_{i=1}^9 (d_b(M_{j_i}) - d_b(M'_{j_i})) \quad j = x, y, z, \quad (2)$$

$\text{grad}(P)$  is then normalized.

The gradient map obtained is very sensitive to each detail of the distance map, and to its digitization effects, too. Thus, to compute an accurate normal vector field on the surface  $S$  of  $O$ , we have to sum

these elementary gradient vectors on an appropriate neighborhood.

## SUMMATION

A summation of the elementary gradient vectors in a fixed spherical neighborhood can induce significant errors for shapes that present singularities (like the edges or the vertices of cubes).

To overcome this problem, for each voxel  $V_P$  of the surface  $S$ , the construction of the neighborhood is driven by the detection of the singularities of the gradient field.

The problem of the edge detection on a surface is replaced by the search of irregularities inside the gradient map (Fig. 1).

Let us define a critical angle  $a_0$  so that, for an angle  $a < a_0$ , the surface is assumed as having an edge. Given  $P \in S$  and  $Q \in O$ , we name  $b$  the angle ( $\text{grad}(d_b(Q)), \text{grad}(d_b(P))$ ).

If  $b > (\pi - a_0)/2$ , then the segment  $[PQ]$  intersects a singularity of the gradient map (such a singularity corresponds to a local extremum of the distance map). Thus, the local gradient whose angle  $b$  is wider than  $b_0 = (\pi - a_0)/2$  has to be ignored in the summation.

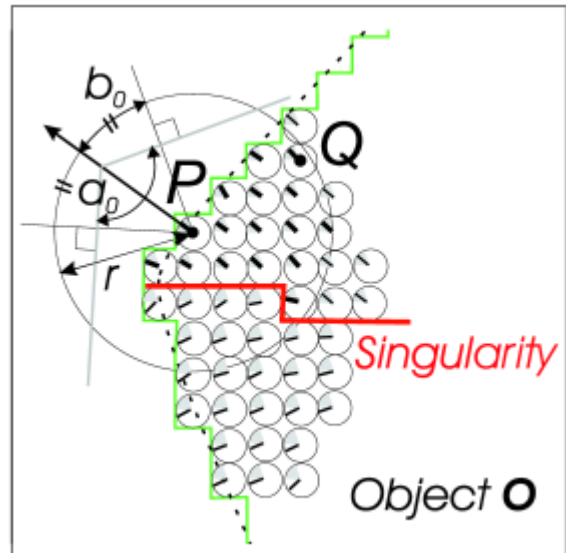


Fig. 1. Summation of gradient vectors (in 2D). For each voxel, the orientation of the gradient vector is represented by a bar in a circle. If  $(\text{grad}(d_b(Q)), \text{grad}(d_b(P))) > b_0$ , then  $[PQ]$  intersects a singularity of the gradient map. In this case, the contribution of  $Q$  has to be ignored.

Let  $r$  be the radius of the spherical neighborhood around  $P$  where local gradient contributions must be

tested. For each point  $P \in S$ , the summation is calculated as follows:

-  $\mathbf{grad}(P)$  is assumed to be the first estimation of the normal vector  $V(P)$

While ( $r \leq r_{max}$ )

{  
 $V(P)$  is updated by summing each contribution of the gradient map on the spherical cap of  $r$  radius. To be taken into account, each contribution from the current point  $Q$  should fulfill:

$$\frac{\mathbf{grad}(d_b(Q)) \cdot V(P)}{\|\mathbf{grad}(d_b(Q))\| \cdot \|V(P)\|} < \cos b_0, \quad (3)$$

$r$  is incremented  
 }

$V(P)$  is normalized.

Note that  $a_0$  must be acute enough to obtain a smooth vector field and wide enough to detect discontinuities. In our case, we choose  $a_0 = 120^\circ$ . This angle verifies the above conditions and corresponds to the angle of hexagonal ice crystals.

## RESULTS

Our algorithm was tested on simulated data and natural snow images processed by X-ray microtomography (Brzoska *et al.*, 1999b). The evaluation of the normal vectors was realized in two ways:

- by comparing the obtained normal vectors to the theoretical normal vectors for synthetic objects whose surface is differentiable.

- by visualizing them for others objects. For visualization,  $\mathbf{u}$  being the illumination vector, the gray level in  $P$  was simply computed as follows:  $\text{Gray\_level} = -V(P) \cdot \mathbf{u}$

Except in the last test where the effect of  $r_{max}$  is analyzed,  $r_{max}$  was set to 5 voxels.

## SPHERES

Tests were carried out on a set of digitized spheres from 1 to 60 voxels of radius. Spheres were obtained according to the definition proposed by Papier and Françon (1998) and Tellier and Debled-Rensson (1999):

A sphere with an integer radius  $r$  is defined by  $(x, y, z) \in \mathbb{Z}^3$  so that:

$$x^2 + y^2 + z^2 < r(r+1). \quad (4)$$

The theoretical normal vector is the vector starting from the center of the sphere through the centers of the voxels.

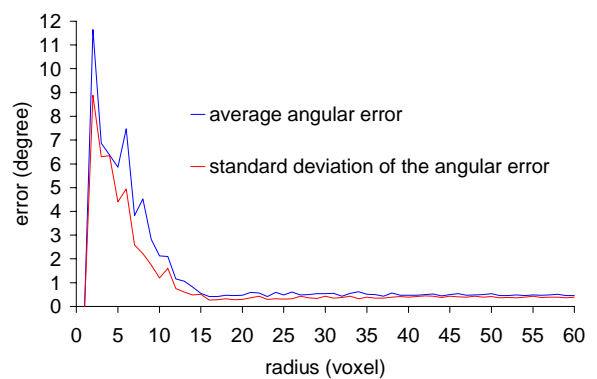


Fig. 2. Angular error for spheres.

## OTHER GEOMETRICAL SHAPES

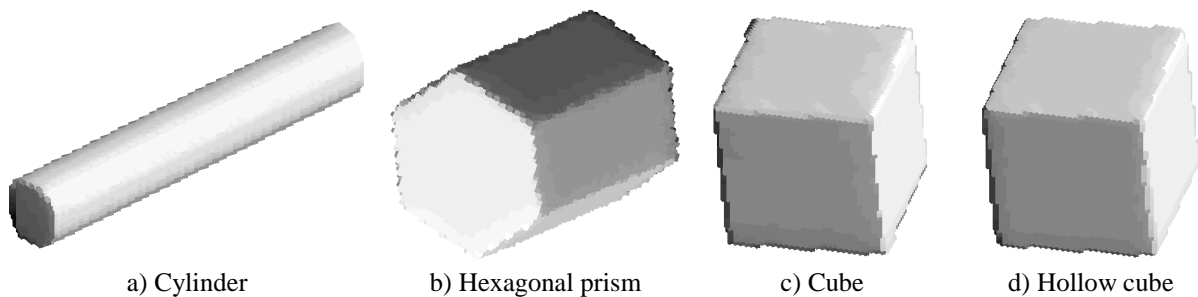


Fig. 3. Image rendering for simulated data. The Fig. 3d shows the normal rendering of the same cube as Fig. 3c. However, the calculation was processed on the complementary of this cube. Illuminating vector  $\mathbf{u}$  was set to  $-\mathbf{u}$  to compare the two images.

SNOW IMAGE

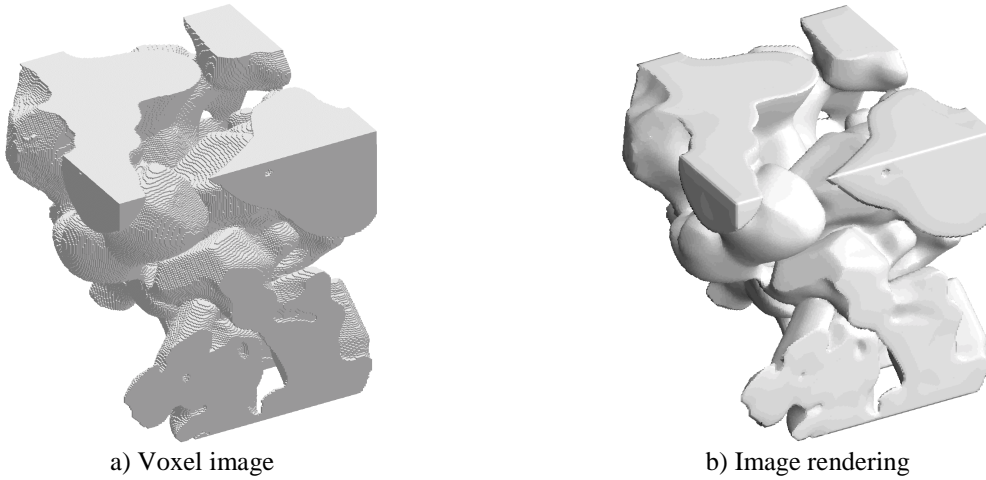


Fig. 4. *Sample of a melt-freeze crust taken at the Col de Porte (France).*

EFFECTS OF THE NEIGHBORHOOD SIZE

For this test, an object whose normal vector field is well known but relatively complex was chosen. The surface equation of this object:  $x^6 + y^6 + z^6 = 35^6$  allows to define the theoretical normal vector field on each point of its surface as follows:

$$\mathbf{V}_{th}(x, y, z) = 6x^5 \cdot \mathbf{i} + 6y^5 \cdot \mathbf{j} + 6z^5 \cdot \mathbf{k} \quad (5)$$

( $\mathbf{i}, \mathbf{j}, \mathbf{k}$ ) being the canonical base.

To obtain a tiled cube, a rotation of the object and of his theoretical vector field was done. To evidence the effect of the choice of  $r_{max}$ , we computed the normal vectors of this object for  $r_{max} = 0$  to 24 voxels. Here are the average errors and standard

deviation of the errors obtained:

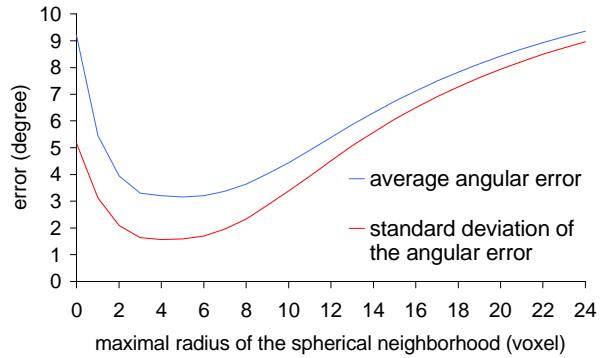


Fig. 5. *Variation of the angular error with the size of the neighborhood for a surface of equation  $x^6 + y^6 + z^6 = 35^6$ .*

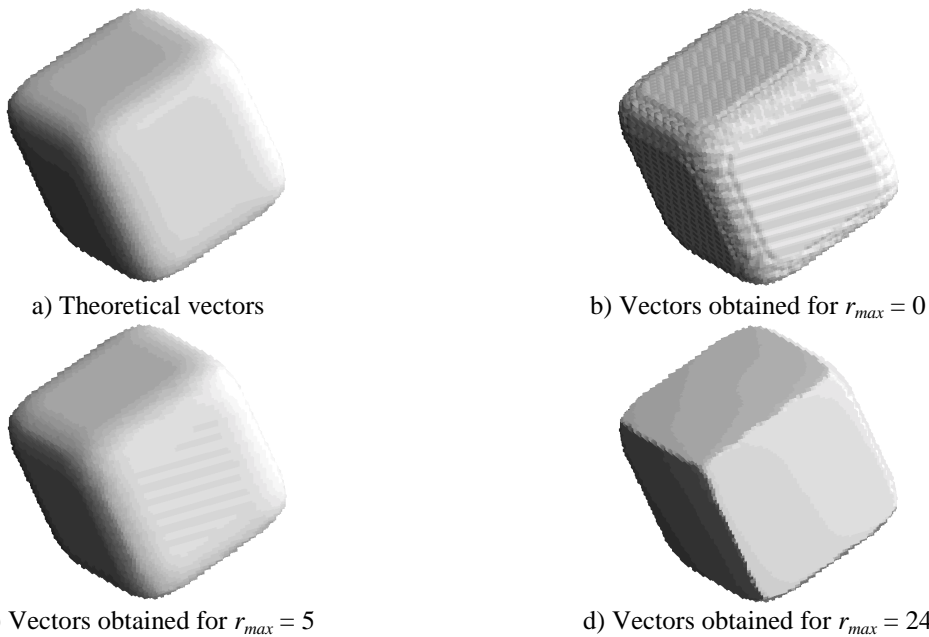


Fig. 6. *Image rendering of the surface of equation:  $x^6 + y^6 + z^6 = 35^6$ .*

## DISCUSSION

The results obtained for spheres (Fig. 2) are close to commonly obtained results in others works (Lenoir *et al.*, 1996; Papier and Françon, 1998). We can note that for spheres of radius  $< 10$ , errors are significant. This can easily be explained by the fact that a sphere of small dimensions is not a sphere but a polyhedron.

Rendering images point out the validity of the method on both convex and concave sharp edges (Fig. 3). The relevance of the computed vector field on edges allows to display faceted snow grains with accuracy (Fig. 4).

The choice of the maximal radius of the neighborhood depends on two constraints (Fig. 5):

- the normal vector needs to be computed from a sufficient number of gradient vectors. Otherwise, it is corrupted by digitization effects (Fig. 6b).
- the maximal size of the neighborhood has to be close to the size of the smallest detail we want to process correctly. Otherwise, the vector orientation may be altered by irrelevant contributions (Fig. 6d).

Besides, retained surface voxel of the search neighborhood should remain connected together; this is mostly the case for snow at a resolution of 10 microns.

The choice of a maximal radius close to 4-5 voxels, used for all images, seems a good compromise because of the two above constraints.

We have presented a new and simple computational method to obtain accurate normal vectors on the surface of a numerical object. This method allows to distinguish a surface that contains edges from a smooth one. Thus, relevant vectors can be obtained in both cases.

A way to improve this algorithm is to make the calculation as independent as possible of  $r_{max}$ . This could be achieved by summing gradient contributions on the largest "well-balanced" neighborhood. A method for the determination of such a neighborhood is now in progress.

## ACKNOWLEDGEMENT

The authors are grateful to David Coeurjolly for fruitful discussions and valuable references.

## REFERENCES

- Borgefors G (1984). Distance transformations in arbitrary dimensions. *Comp Vis Graph Image Proc* 27:321.
- Borgefors G (2001). Optimal local distances for distance transforms in 3D using an extended neighbourhood. *Lecture Notes in Computer Science* 2059:113-22.
- Brzoska JB, Lesaffre B, Coléou C, Xu K, Pieritz RA (1999a). Computation of 3D curvatures on a wet snow sample. *Eur Phys J AP* 7:45-57.
- Brzoska JB, Coléou C, Lesaffre B, Borel S, Brissaud O, Ludwig W, Boller E, Baruchel J (1999b). 3D visualization of snow samples by microtomography at low temperature. *ESRF Newsletter*.
- Brzoska JB, Flin F, Lesaffre B, Coleou C, Lamboley P, Delesse JF, Le Saëc B, Vignoles G (2001). Computation of the surface area of natural snow 3D images from x-ray tomography: two approaches. *Image Anal Stereol* 20 (Suppl. 1):306-12.
- Chassery J-M, Montanvert A (1991). *Géométrie discrète en analyse d'images*. Hermès. Paris.
- Danielsson PE (1980). Euclidean distance mapping. *CGIP* 14:227-48.
- Lenoir A, Malgouyres R, Revenu M (1996). Fast computation of the normal vector field of the surface of a 3-D discrete object. *Proceedings of DGCI'96, LNCS 1176, Springer, 101-12*.
- Lorensen WE, Cline HE (1987). *Comp Graphics* 21(4):163-9.
- Papier L, Françon J (1998). Evaluation de la normale au bord d'un objet discret 3D. *Revue de CFAO et d'informatique graphique* 13(2):205-26.
- Ragnemalm I (1989). Contour processing distance transforms. *5th Int. Conf. On Image Analysis and Processing, Positano (Italy), 204-12*.
- Rolland F (1991). *Représentation tridimensionnelle et reconstruction 3D à partir de coupes 2D*. Thèse de doctorat. Université Joseph Fourier, Grenoble.
- Tellier P, Debled-Renesson I (1999). 3D discrete normal vectors. *Proceedings of DGCI'99, LNCS 1568, Springer, 447-58*.
- Thürmer G, Würthrich CA (1997). Normal computation for discrete surfaces in 3D space. *Computer Graphics Forum* 16(3):15-26.
- Verwer BJH (1991). Local distances for distance transformations in two and three dimensions. *Pattern Recognition Letters* 12:671-82.
- Yamada H (1984). Complete Euclidean distance transformation by parallel operation. *7<sup>th</sup> ICPR, IEEE Comp Soc Press, Montreal, 69-71*.