

3D Discrete Normal Vectors

Pierre Tellier and Isabelle Debled-Rennesson

LSIIT UPRES-A ULP-CNRS 7005

(Laboratoire des Sciences de l'Image, d'Informatique et de Télédétection),

Pôle API, Bd. Sébastien Brandt, F67400 ILLKIRCH

Tel : +33 (0)3 88 65 5500, Fax : +33 (0)3 88 65 5501

{tellier, debled}@lsiit.u-strasbg.fr

Abstract. Precise knowledge of normal vectors to discrete objects is mandatory in rendering algorithms. This article introduces a new method for the calculation of normal vectors to a digital object. This technique relies on discrete geometry theories : the recognition of discrete straight lines and tangential lines in dimension 2. Results obtained with synthetic and real objects from medical imagery are presented and commented.

1 Introduction

Well estimating the normal vectors of a digital object is primordial for the simulation of lighting and special effects in relation with light on digital objects. But these objects are in general digitized and the normal vectors to volumes are unknown.

Numerous authors have been interested in the calculation problem of normal vectors to a digital object by using miscellaneous approaches. A synthesis of the works undertaken on this subject until 1992 is presented in the article [15]. Since this, other approaches considering the 3D neighborhood of the points has appeared, one of them [13] uses the volume around the point where the normal vector is calculated. Another one [10] is based on a generalization of the umbrella notion [9,4,5,6] at the vertex of a polygonal volume.

The method proposed in this article takes his origin in a DEA practice [8] and considers that an integer point is represented by a voxel; a cube whose side is one unit and whose faces are surfels. The neighborhood of each point is considered according to its regularity, thus plane parts and holes are integrally rendered without any threshold.

This work introduces a new estimation of the 3D discrete normal vectors based on the definitions of discrete straight lines and tangential lines in dimension 2 given in [3,2,14,12,11]; to calculate the normal vector at one point to the edge of a digital object, we use the slices of this object along the three canonical planes through this point (see Fig. 1). On the outline of each slice, we calculate the tangential line through the considered point [14] then by combining the 2D tangential lines obtained (1, 2 or 3) and by applying a vectorial product, we obtain the 3D normal vector to the point of the voxel object. A surfel approach is also proposed with a similar method by calculating the normal vector at each

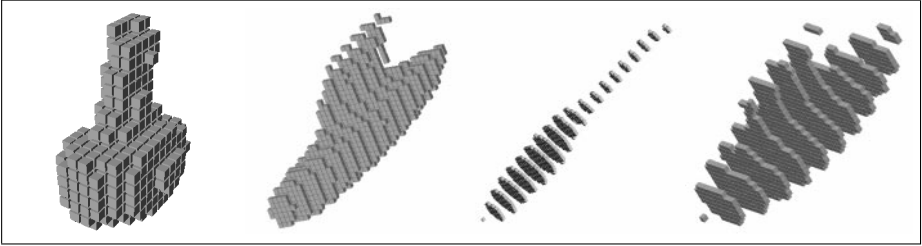


Fig. 1. Digitized mushroom and its slices

surfel to the edge of the voxel object. Firstly, we recall the results of 2D discrete geometry necessary for the calculation of the 3D normal vectors. Then, we define and explicitly give the calculation of normal vectors to any voxel object according to two methods in 8-connexity for the calculation of a normal vector to a voxel of the edge then in 4-connexity for a surfel of the voxel object edge. Finally, we comment the results obtained with synthetic objects and real objects.

2 2D Discrete Straight Lines and Tangential Lines

We present in this section the basic notions of 2D discrete geometry useful for the determination of 3D normal vectors.

2.1 Discrete Straight Lines

The arithmetical definition of a discrete straight line has been introduced by JP. Réveillès [12,11] as follows : a **discrete straight line** whose slope is $\frac{a}{b}$ with $b \neq 0$ and $a \wedge b = 1$, with a lower bound μ , an arithmetical thickness ω , is the set of points (x, y) of \mathbb{Z}^2 which satisfies the double diophantine inequation $\mu \leq ax - by < \mu + \omega$, where all parameters are integer.

We call the preceding discrete straight line $\mathcal{D}(a, b, \mu, \omega)$. In the following, we shall be interested in the **naïve straight lines** which verify $\omega = \sup(|a|, |b|)$ and which are 8-connex and, in the **standard straight lines** which verify $\omega = |a| + |b|$ and are 4-connex (see Fig. 2). In order to simplify the writing, we shall suppose in the following that the slope coefficients verify $0 \leq a \leq b \neq 0$.

2.2 Recognition of Discrete Straight Line Segment

Let us consider Σ a segment of \mathcal{D} , naïve or standard straight line whose characteristics are a, b, μ, ω with $0 \leq a < b \neq 0$ and l is the maximal value of the segment points on the x axis. Let us suppose that the point $M(x_M, y_M)$ (with $x_M = l + 1$ or $x_M = l$) is added to Σ , is $\Sigma' = \Sigma \cup \{M\}$ a segment of the straight line and, in this case, what are the characteristics a', b', μ' ? This problem is solved in [3,2] and relies on some particular points of a discrete straight line defined hereafter.

The real straight lines $ax - by = \mu$ and $ax - by = \mu + \omega - 1$ are called **leaning straight lines** of the naive discrete straight line $\mathcal{D}(a, b, \mu, \omega)$. An integer point of these straight lines is called **leaning point**.

The leaning straight line located above (resp. under) \mathcal{D} has the following equation $ax - by = \mu$ (resp. $ax - by = \mu + \omega - 1$), it is called **upper leaning straight line** (resp. **lower leaning straight line**) of \mathcal{D} . Some leaning points are particularly important along the recognition of a straight line segment, they are the ones located at the extremities of the segment whose recognition is in progress. We note U_B (resp. L_B) the upper leaning point (resp. lower) whose value on the x axis is minimal. In the same way, we note U_F (resp. L_F) the upper leaning point (resp. lower) whose value on the x axis is maximal (see Fig. 2).

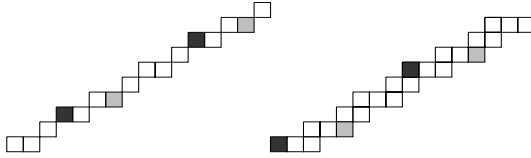


Fig. 2. On the left hand side, the naive straight line $\mathcal{D}(5, 8, -1, 8)$ for $x \in [0, 15]$ and on the right hand side, the standard straight line $\mathcal{D}(5, 8, 0, 13)$ for $x \in [0, 15]$, the upper leaning points of these straight lines are in dark gray and the lower leaning points in light gray.

Theorem 1 ([3,2]). Let us consider $r(M)$, the **remainder at the point** $M(x, y)$ as a function of \mathcal{D} defined by :

$$r(M) = ax - by$$

- (i) If $\mu \leq r(M) < \mu + \omega$, then $M \in \mathcal{D}(a, b, \mu)$, $\Sigma \cup \{M\}$ is a segment of the straight line \mathcal{D} .
- (ii) If $r(M) = \mu - 1$, then $\Sigma \cup \{M\}$ is a segment of the straight line whose slope is given by the vector $U_B M$ with U_B the upper leaning point of \mathcal{D} whose value on the x axis is minimal.
- (iii) If $r(M) = \mu + \omega$, then $\Sigma \cup \{M\}$ is a segment of the straight line whose slope is given by the vector $L_B M$ with L_B the lower leaning point of \mathcal{D} whose value on the x axis is minimal.
- (iv) If $r(M) < \mu - 1$ or $r(M) > \mu + \omega$, then $\Sigma \cup \{M\}$ is not a segment of a discrete straight line.

This theorem allows to obtain an incremental algorithm of discrete straight line segment recognition [3,2] by scanning series of pixels 8 or 4-connex called **discrete way**. The algorithm in 8-connexity has been adapted by A.Vialard [14] in order to determine a notion of **discrete tangential line**.

2.3 Discrete Tangential Line at a Point to a 2D Curve Line

The **discrete tangential line** at a point M to a discrete way \mathcal{C} is defined [14] as the longest part of \mathcal{C} centered to M which is a segment of a discrete straight line.

The algorithm of discrete straight line segment recognition centered to a point is used in order to determine the longest discrete segment centered to M by adding at each step a point to each extremity. If the couple of added points does not permit to enlarge the segment, none of them is added.

Addition of a point at the front extremity of a segment of the naive or standard straight line $\mathcal{D}(a, b, \mu, \omega)$ of the first octant

```

 $r = ax_M - by_M$  ;
If  $\mu \leq r < \mu + \omega$ 
    If  $r = \mu$  then  $U_F = M$  Eif
    If  $r = \mu + \omega - 1$  then  $L_F = M$  Eif
else
    If  $r = \mu - 1$  then
         $L_B = L_F$  ;  $U_F = M$  ;
         $a = |y_M - y_{U_B}|$  ;  $b = |x_M - x_{U_B}|$  ;
         $\mu = ax_M - by_M$  ;
    else
        If  $r = \mu + \omega$  then
             $U_B = U_F$  ;  $L_F = M$  ;
             $a = |y_M - y_{L_B}|$  ;  $b = |x_M - x_{L_B}|$  ;
             $\mu = ax_M - by_M - \omega + 1$  ;
        else
            the new point cannot be added to the segment
    Eif
Eif
Eif

```

To add a point at the rear extremity of the segment, the preceding algorithm must be used by exchanging L_F and L_B as well as U_F and U_B .

We define the **right tangential line** (resp. **left**) at a point M to a discrete way \mathcal{C} as the longest part of \mathcal{C} following (resp. preceding) the point M which is a segment of a discrete straight line. The calculation of the right (resp. left) tangential lines uses the algorithm to add a point at the front (resp. rear) extremity of a segment of a naive or standard straight line given above.

3 Calculation of the 3D Normal Vectors at each Point of a Voxel Object Edge

The calculation of the normal vectors at all points of a voxel object relies on the study of outlines associated with each point of the object located on the object slices through the considered point. We call **slice in x (resp. y, z) at the**

point M , with $M(x_M, y_M, z_M)$, the set of points of the object having the same co-ordinate x_M (resp. y_M, z_M) (see Fig. 1).

In the following, the slice at a point according to one co-ordinate may be viewed as an object in two dimensions. The **outlines** associated with an object slice are composed of the closed 8-connex series of pixels of the slice edges (see Fig. 5).

We shall call **associated pixel to M** the 2D points corresponding to the projected points of M along the co-ordinate planes. Normal vectors at each point of the object are calculated as a function of the outlines which these points belong to.

3.1 Voxel Position on the Edge of the Object

We have voluntarily adopted a realistic viewpoint by wanting to calculate the best normal vector estimation to obtain an efficient rendering. Thus the voxel approach, which consist in considering an integer point as a cube with 6 faces, leads to problems due to the point position on the edge of the object. In the general case called **regular** cases, voxels possess 1, 2, or 3 faces called **surfels**, not opposite 2 to 2 others, on the edge of the object. In the other cases called **non regular**, we find two types of voxel configuration :

- voxels having at least two opposite faces adjacent to the edge, ie. belonging to the "fine" locations of the object with a 1-voxel thickness, (Fig. 3 right)
- corner or side voxels (Fig. 3 left) corresponding to the object discontinuities.

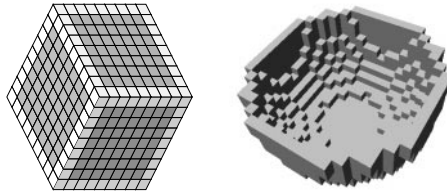


Fig. 3. On the left hand side, in gray regular voxels and in white non regular voxels. On the right hand side, an example of a fine object with a 1-voxel thickness.

The different configurations are simply detectable in dimension 2 on the outlines calculated at the M considered point. Let us suppose that each outline is coded through a Freeman code [7]. For each outline through M , we note d_B (resp d_F) the direction which represents the move to the following (resp. preceding) pixel associated to M on the outline according to the trigonometric way.

The general configuration of the voxel M is if, in **each slice** through M , there is **at the most one outline through a pixel associated** to M and if :

$$d_B \equiv d_F + 3[8] \text{ or } d_B \equiv d_F + 4[8] \text{ or } d_B \equiv d_F + 5[8]$$

In the other case,

- if, in a same slice, there are several outlines through a pixel associated to M , the point M is located at a "fine" location of the object,
- if, on an outline through M , the directions d_B and d_F are not compliant, the voxel object shows a discontinuity at this location.

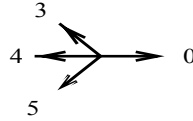


Fig. 4. On this figure, $d_F = 0$, the only possible back directions to be in the general configuration are $d_B = 3, 4$ or 5 .

Examples of these three point types are shown on one of the outlines associated to a slice of the digitized mushroom of Fig. 5. For instance, at the pixel R , $d_F = 0$ and $d_B = 7$ are two directions which are not compliant, thus there is a discontinuity at the related point. It is clear that no unique normal vector may be assigned to a voxel in a non regular configuration, the decomposition in surfels of this voxel must then be considered. Normal vector calculation for these different configurations are described in the following paragraph.

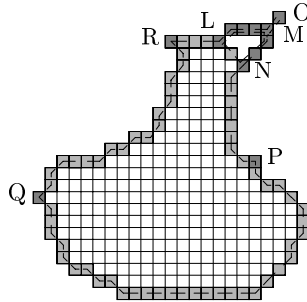


Fig. 5. The outlines of that mushroom slice are in dotted lines. Pixels L to M and M to N are non regular pixels because there are two outlines through these points; it is the same for pixels O , P , Q and R because directions d_F and d_B associated to these pixels are not compliant. The other pixels of the outlines (in gray color) are regular.

3.2 Definitions

In the *general case*, we define *normal vector* at the point M of the voxel object edge as the sum of (one, two or three) vectors obtained through the following steps :

- seek in each slice associated to the point M , on the outline through M , the 2D tangent vector to the pixel associated to M ,
- if 2 or 3 tangent vectors are obtained, compute the vectorial product of these vectors, if not take the perpendicular vector to the only tangent vector obtained.

The research of the 2D tangent vectors needs several phases :

- (i) Determine the 3 slices which the point M belongs to.
- (ii) Seek, in the list of outlines associated to each of these 3 slices, the outlines through M .
- (iii) For each outline through M , consider this outline as a discrete way and calculate the tangent line to the pixel associated to M by using the algorithm in 8-connexity described in Sect. 2.

The normal vector obtained is then associated to each surfel of the voxel belonging to the edge of the object.

In the **non regular cases**, the research of tangent vectors and the allocation of the normal vectors to the surfels are different; surfels of the edge of a same voxel will not have the same normal vector. Thus three tangent lines (one by slice) may be associated to each surfel of a voxel, the research algorithm of the normal vectors to the surfels is the following one :

In the three slices which the point M belongs to,

- If the pixel associated to M is isolated, the standard tangent lines are assigned to each of these sides then forwarded to the corresponding surfels (see the study of point B in the plane Oxz of Fig. 6).
- If not, for each outline through a pixel associated to M ,
 - If the outline has a discontinuity at this point (d_F and d_B not compliant), calculate the left and right tangent lines to the considered pixel by using the algorithms in 8-connexity described in Sect. 2. The allocation of the tangent lines to the pixel sides then to the corresponding surfels is undertaken in the following way :
 - * The surfel which corresponds to the side adjacent to the direction (d_F) in the trigonometric way inherits the half-tangent line which corresponds to the front half-outline.
 - * Symmetrically, the surfel which corresponds to the side adjacent to the direction (d_B) in the inverse trigonometric way shall get the value of the tangent line associated to the rear half-outline.
 - * Finally if, by scanning the set of directions in the trigonometric way from d_F to d_B , one of the pixel sides is on the edge of the object and does not possess any tangent line in this slice, assign to it the sum of both half-tangent lines oriented in the same way.
 - If not, the outline has no discontinuity at this point, calculate the tangent line to the pixel associated to M by using the algorithm described in Sect. 2. This tangent line is associated to each surfel (represented by a side of the pixel) included in the part of the plane associated to this outline.

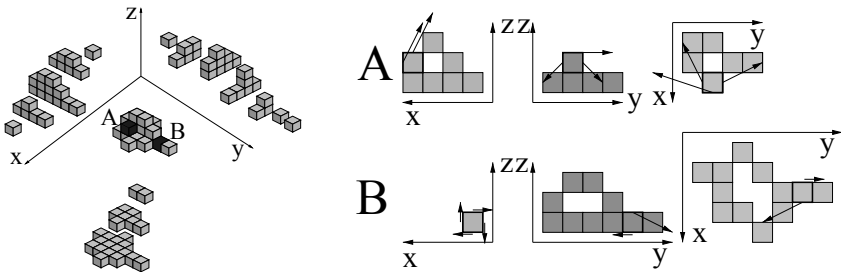


Fig. 6. Study of two voxels A and B non regular on a voxel object. On the left hand side, the voxel representation of this object with its slices. On the right hand side, the tangent lines obtained on each outline for the points A and B.

As a function of the tangent lines assigned to the surfels, the normal vectors to the surfels are calculated by vectorial products.

Remarks

1. The non regular cases only represent a very small part of the synthetic or real voxel objects; results are given in section 5. But taking into account these particular cases allows to obtain a very realistic rendering on the extremities of the objects even in case of irregularities (see Fig. 8).
2. Trials have been done by using variants [14] of the discrete tangent line definition given in paragraph 2 :
 - Reduced tangent line : points of the segment extremities are deleted.
 - Asymmetrical tangent line : prolongation of the tangent line at one extremity.
 - Asymmetrical reduced tangent line : combination of the two preceding definitions.

These different definitions have not given satisfactory results and do not bring any improvements to the first definition.

4 Calculation of the 3D Normal Vectors at each Surfel on the Edge of a Voxel Object

After the definition of the preceding chapter, it seems to be natural to be interested in a surfel approach, by calculating the normal vector at each surfel on the edge of a voxel object. The general idea remains identical but a few supplementary definitions are necessary : The pixel sides are called *lignels* and its vertices are called *pointels*. The *pointel outlines* associated to a slice of the object are composed of the 4-connex series of pointels of pixels on the edge of the slice (see Fig. 7). The *normal vector at a surfel* on the edge of a voxel object is the vector obtained after the following steps :

- Seek the two slices it belongs to. For instance if the surfel is parallel to the plane Oxy , both slices shall be parallel to the planes Oxz and Oyz .
- Determine for each slice the pointel outline through the lignel associated to the surfel, then calculate the discrete 4-connex tangent lines to each pointel which delimits the lignel associated to the considered surfel. The tangent line to the lignel is obtained by calculating the average tangent line to the pointels.
- The vectorial product which is the result of the tangent lines calculated in each slice is the normal vector to the surfel.

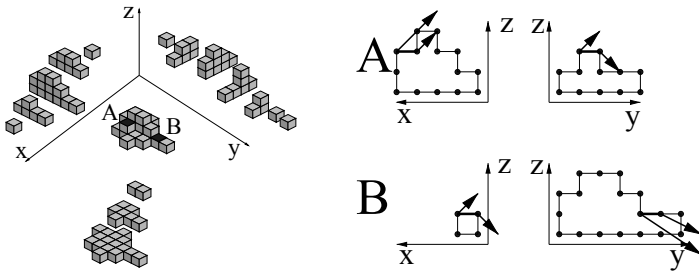


Fig. 7. Study of two surfels A and B on a voxel object. On the left hand side, the voxel representation of this object with its slices. On the right hand side, the tangent lines to the pointels obtained on each pointel outline for the surfels A and B.

Remarks

1. For each surfel on the edge, four calculations of the discrete tangent lines are necessary (only one is necessary in the voxel approach in the regular cases).
2. The proposed method may always be applied, there is no particular cases.
3. Two tangent lines are assigned for each surfel ; in the preceding method, up to three tangent lines might be assigned.

5 Results

The evaluation of the normal vectors is realized in two ways ; on synthetic objects by comparing the obtained normal vectors to the theoretical normal vectors and by visualizing them on digitized objects. The Figs. 9 to 15 show in the left hand side the voxel object and in the right hand side the rendering obtained thanks to the normal vectors calculated at each voxel of the object.

We have compared the normal vectors obtained to the theoretical normal vectors calculated as follows:

- A plane $P(a, b, c, \mu, \omega)$ is defined by the set of integer points (x, y, z) which verifies $\mu \leq ax + by + cz < \mu + \omega$ where all parameters are integer. The theoretical normal vector is the vector (a, b, c) .

- A sphere with an integer radius r is defined by the integer points (x, y, z) which verifies $x^2 + y^2 + z^2 < r(r + 1)$ [1]. The theoretical normal vector is the vector starting from the center of the sphere through the voxel (resp. surfel) centers in the case of the voxel (resp. surfel) approach.
- The theoretical normal vector to a voxel (resp. surfel) of a cube is the normal vector to the face it belongs to.

The angle between the theoretical normal vector and the estimated one is calculated at each point of the tested objects. The average value and standard deviation of these values are presented in the following table according to the methods presented in Sects. 3 and 4. We observe more important errors with the second method on objects as the cube (see Fig. 8); corners are smoothed.

Table 1. Discrete normal vectors vs theoretical normal vectors

	8-connectivity method (Sect. 3)		4-connectivity method (Sect. 4)	
Object	Avg. error	Std. deviation	Avg. error	Std. deviation
Sphere radius = 5	4.83	3.63	6.95	5.23
Sphere radius = 15	2.87	2.20	3.01	1.86
Sphere radius = 25	2.84	2.24	3.22	1.92
Cube side = 15	0.00	0.00	8.18	4.82
Plane (1,2,-5,-10)	1.78	0.00	1.78	0.00
Plane (3,4,-5,-5)	1.02	0.00	1.02	0.00

In order to estimate the cost of both methods, the following table gives, for each presented object, the resolution, the number of voxels and surfels of the edge, the number of non regular voxels as well as the computing time of Freeman’s codes of the object slices and computing time of the normal vectors. Computing time has been obtained on a Silicon Graphics R10000 195 Mhz. Moreover the pre-processing step for the calculation of object slices outlines has not been optimized yet.

Table 2. Computation times

Object	Resolution	Edge voxels	Surfels	Non regular	Outlines	Normals
Blood veins	266x285x241	40912	89238	11232	438 s	40 s
Brain	128x128x128	32259	60052	16907	47 s	96 s
Head	128x128x128	36108	62006	1910	15 s	53 s
Mushroom	86x100x56	15253	26392	714	2 s	14 s
Knight	249x157x156	121540	246714	69026	47 s	251 s
Sphere	50x50x50	6342	11646	0	0.6 s	4 s
Sphere	100x100x100	25746	46950	72	5 s	28 s
Sphere	200x200x200	103734	188382	0	50 s	198 s

6 Conclusion

We have presented an original, simple and efficient method for the calculation of the normal vectors to a voxel object. It relies on strong theoretical bases, the arithmetic definitions of discrete straight lines. Numerous examples have been tested by using real data coming from medical imagery; the algorithms used have proven their consistency and quickness. They are currently used by voxel object handlers. Nevertheless an accurate comparative study between the different methods existing in the literature [10,13,15] still needs to be done to estimate which one shall be the most efficient.

Acknowledgements: The authors thank J. Françon for the ideas he raised and for the numerous advices he gave us. We also thank the CIRAD (Centre de Coopération Internationale en Recherche Agronomique), the IRCAD (Institut de Recherche sur les Cancers de l'Appareil Digestif) and the IPB (Institut de Physique Biologique) for the data they supplied.

References

1. E. ANDRES. Cercles discrets et rotations discrètes. Thèse. Université Louis Pasteur, Strasbourg, 1992.
2. I. DEBLED-RENNESON. Etude et reconnaissance des droites et plans discrets. Thèse. Université Louis Pasteur, Strasbourg, 1995.
3. I. DEBLED-RENNESON and J.P. REVEILLÈS. A linear algorithm for segmentation of digital curves. In *International Journal of Pattern Recognition and Artificial Intelligence*, volume 9, pages 635–662, 1995.
4. J. FRANÇON. Arithmetic planes and combinatorial manifolds. In *Proceedings of the 5th International Workshop Discrete Geometry for computer Imagery*, pages 209–217, Clermont-Ferrand (France), September 1995.
5. J. FRANÇON. Discrete combinatorial surfaces. In *CVGIP*, pages 20–26, 1995.
6. J. FRANÇON. Sur la topologie d'un plan arithmétique. In *Theoretical Computer Science*, volume 156, pages 159–176, 1996.
7. H. FREEMAN. Boundary encoding and processing. In *Picture Processing and Psychopictorics*, pages 241–266. New York Academic, 1970.
8. Y. KIRSCHHOFFER. Estimation de normale dans un espace discret 3d. analyse comparative de méthodes, Juin 1997. Rapport de DEA, U.L.P., Strasbourg.
9. S. LEFSCHETZ. Applications of algebraic topology. In *Springer Berlin*, 1975.
10. L. PAPIER and J. FRANÇON. Evaluation de la normale au bord d'un objet discret 3d. In , *Revue internationale de CFAO et d'Informatique graphique*, à paraître.
11. J.P. REVEILLÈS. Géométrie discrète, calculs en nombre entiers et algorithmique. Thèse d'état. Université Louis Pasteur, Strasbourg, 1991.
12. J.P. REVEILLÈS. Structure des droites discrètes. In *Journées mathématique et informatique*, Marseille-Luminy, Octobre 1989.
13. G. THURMER and A. WUTHRICH. Normal computation for discrete surface in 3d space. In *Eurographics*, 1997.
14. A. VIALARD. Chemins euclidiens : un modèle de représentation des contours discrets. Thèse. Université Bordeaux I, Bordeaux, 1996.
15. R. YAGEL, D. COHEN, and A. KAUFMAN. Normal estimation in 3d discrete space. In *The visual computer*, volume 8, pages 278–291, 1992.

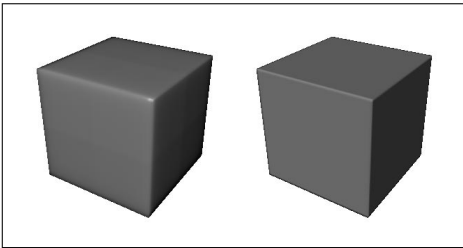


Fig. 8. Rendered in 4 and 8 connexity

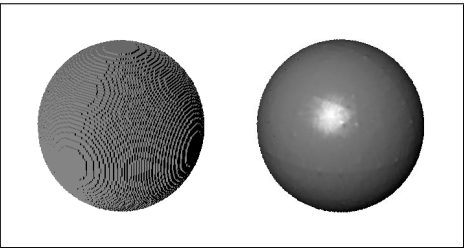


Fig. 9. Sphere 50x50x50

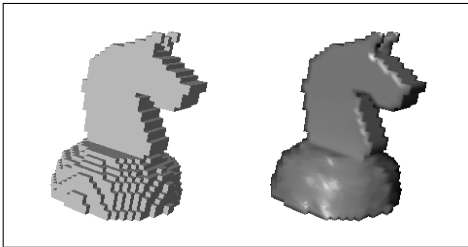


Fig. 10. Knight 39x25x25

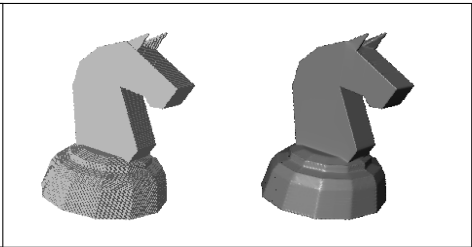


Fig. 11. Knight 249x157x156

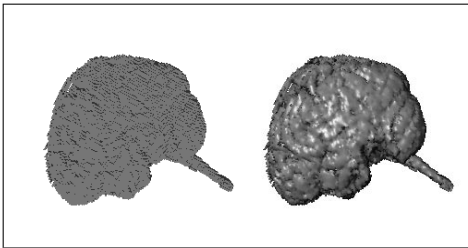


Fig. 12. Brain 128x128x128

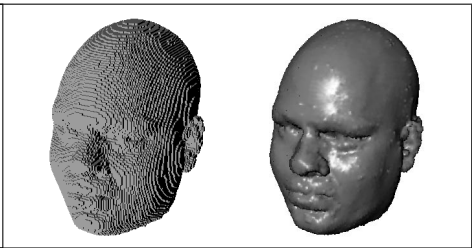


Fig. 13. Head 128x128x128

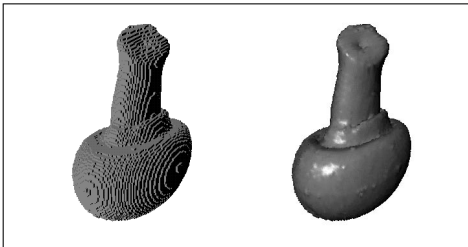


Fig. 14. Mushroom 86x100x56

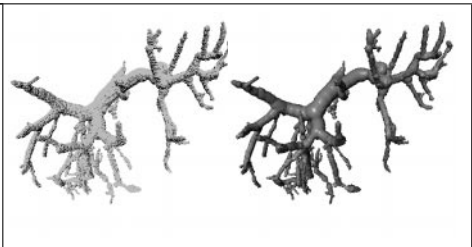


Fig. 15. Vein of human liver
266x285x241