Journal of Electronic Imaging 6(4), 406–414 (October 1997).

Maximum area triangle operator for edge detection

Jean-Pierre Reveillès
Université d'Auvergne
LLAICI
Computer Science Institute of Technology
BP 86, Les Cézeaux
63720 Aubière, France
E-mail: reveil@llaic.univ-bpclermont.fr

J. Yaacoub
Université Louis Pasteur
LSIIT, Computer Science Department
7, rue R. Descartes
67084 Strasbourg, France

Abstract. *Currently images are more often noisy than regular, and universal contour extraction operators are few. The purely discrete operators are rather crude and inaccurate, while the more precise operators, which relying on the continuous approach, are complicated and rather slow. Using the restriction of gray level images to* 3×3 *masks interpreted as 3D digital surfaces, we present a new way of computing exactly the normal vector, or gradient, at regular points. Despite the fact that regular points are seldom on a general image, we deduce from this theoretical result a new discrete contour extraction operator, based on max area triangles contained in* 3×3 *masks (the MAT operator), which yields high quality results, comparable with those of continuous operators.* © *1997 SPIE and IS&T.* [S1017-9909(97)00204-3]

1 Introduction

If gray level digital images are considered as 3D digital surfaces, the most regular images are, obviously, digital planes and those having, locally, such a structure. Regular images can be met in some specialized fields.[1,2] The appearance of digital planes in images was noted by many researchers who derived results and algorithms from this guiding line (cf. Refs. 3 to 10). The work of Veelaert[9,10] went far in this direction and was especially influential.

On the other side, in many fields, various processings are applied to images (digitalization, half-tone screening, dithering, etc.), leading to very *irregular* surfaces. But regularity and irregularity are not that far apart in discrete matters, and the study of *regular* images can shed light on irregular ones. This paper extends and makes results of Ref. 11 more precise.

Paper IST-02 received Jan. 10, 1997; revised manuscript received May 15, 1997; accepted for publication May 29, 1997. This paper is a revision of a paper presented at the SPIE conference on Vision Geometry V, Aug. 1996, Denver, CO. The paper presented there appears (unrefereed) in SPIE Proceedings Vol. 2826.
1017-9909/97/$10.00 © 1997 SPIE and IS&T.

2 Regularity of Digital Images and Debled Tricubes

Digital images are treated as 3D digital surfaces in the sequel. We are interested in their *local* structure, that is, their restriction on 3×3 masks. This amounts to considering a fixed voxel, which can be supposed to be at the origin of coordinates, and its 26 neighboring voxels. Even if images are functional, requiring the consideration *a priori* of only 8 neighbors, the whole collection of images must nevertheless be introduced in the following.

Graphical representations of these local configurations are built by replacing voxels by unit cubes. Thus the origin corresponds to the cube $[0,1]^3$ of $\mathbb{R}^3$, while the voxel located at $[-1,-1,-1]$ is replaced by the unit cube $[-1,0]^3$. More generally, the voxel located at integer point $[x,y,z]$ is imaged by the unit cube $[x,x+1]\times[y,y+1]\times[z,z+1]$.

We shall denote by $\mathcal{T}_3$ the subset of $\mathbb{Z}^3$ defined by the product $[-1,1]\times[-1,1]\times[-2,2]$. It consists of 45 voxels, which give a partition of the Euclidean parallelepiped $[-1,2]\times[-1,2]\times[-2,3]$.

The simplest digital surfaces, that is, digital planes, are conveniently defined as subsets of $\mathbb{Z}^3$ satisfying the following Diophantine inequalities:

$$\gamma \leqslant ax+by+cz < \gamma+\omega, \quad (a,b,c)\in\mathbb{Z}^3, \quad (a,b,c)=1. \tag{1}$$

Such digital planes where $\omega=\max(|a|,|b|,|c|)$, are called *naive digital planes*.

Restrictions on the digital plane normal vector (a,b,c) can be assigned using the fundamental domain of the group of symmetries of the cube. This domain, also known as the *standard simplex*, is defined by inequalities

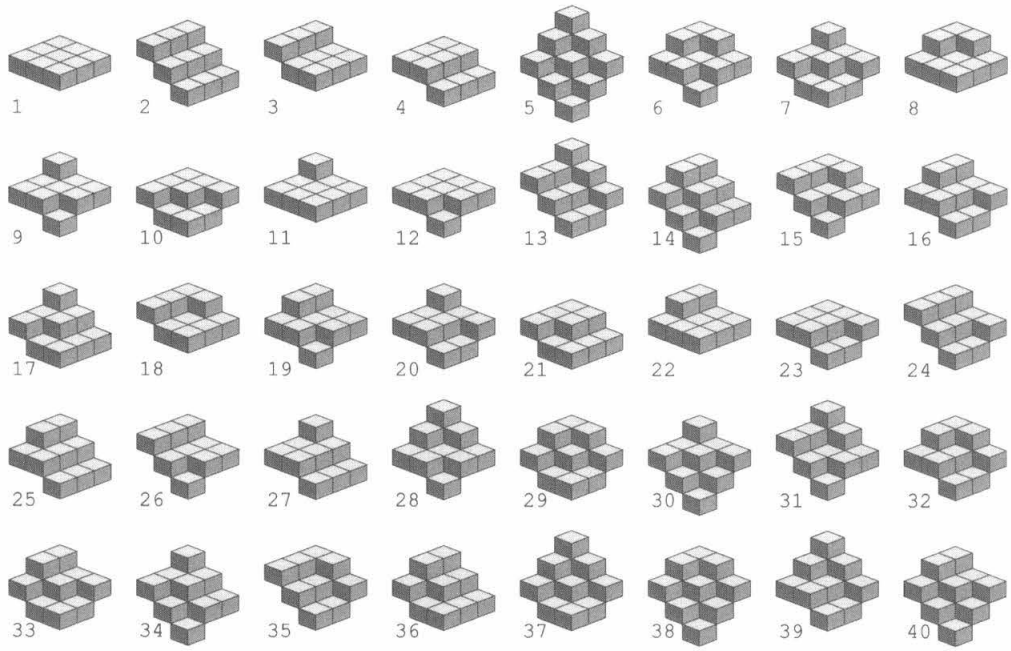


Fig. 1 Forty Debled-Rennesson tricubes.

$$0 \leq a \leq b \leq c. \tag{2}$$

Using cube symmetries, the structure of digital planes defined by any normal vector can be deduced from those where the normal vector belongs to the standard simplex, which supports restricting our study of digital planes to those satisfying the inequalities of Eq. (2). This is assumed in the following.

It would be interesting to know all possible local configurations of digital planes around a given voxel. Because they correspond to subsets of $\mathcal{T}_3$, there are a finite number of them. They are given by intersections of all digital planes containing 0, with $\mathcal{T}_3$.

Debled-Rennesson[4] found 40 combinatorially distinct local configurations of naive digital planes satisfying the conditions of Inequality (2). Each is a little piece made of 9 voxels that projects on a 3×3 mask in (x,y) space. These pieces, which appear in Fig. 1, are called *tricubes*. They are, in a certain sense, the building blocks of 3D *discrete differentiability*.

If normal vectors of *any* direction are allowed, that is if the condition of Eq. (2) is relaxed, the number of intersections between $\mathcal{T}_3$ and all naive digital planes going through 0 increases. Debled-Rennesson found 709 such tricubes, showing that they provide a fairly good discretization of space.

Matrix representations and normal vectors (a,b,c) of tricubes are shown later in Figs. 3, 4 and 5, completing information contained in Fig. 1.

In Ref. 4 tricubes were generated by an exhaustive search of the parameters a, b, c, and γ. They are built one by one and new elements are added to the set of those already discovered. Fig. 1 was drawn with this algorithm.

Tricubes whose normal vector satisfies the condition of Eq. (2) can be conveniently described in the following way. If (x,y,z) denotes voxels coordinates, the z component depends functionally on x and y. Such a tricube can be considered as the graph of the associated mapping $(x,y) \mapsto z(x,y)$, where (x,y) belongs to the set $[-1,0,1] \times [-1,0,1]$. Of course, such a graph can also be represented by an integer matrix of dimension 3 whose entries are given by the third component and where line and row numbers (between 1 and 3) replace the x and y coordinates. Tricubes definition, saying that associated naive digital planes contain 0, is equivalent requiring the central value of these matrices being equal to 0.

Our graphical representation choices and the condition of Eq. (2) determine how coordinates can be recovered from these matrices:

- The x axis is oriented from right to left.
- The y axis is oriented downward.
- The z components are matrix entries.

Let us be more specific about the way these matrices are related to the pictures of Fig. 1. All of them appear later in Figs. 3, 4 and 5, though in an order different from that in Fig. 1. Matrix of tricube number 8 shows up in Fig. 5; its first row corresponds to the farthest row of voxels of its 3D representation in Fig. 1 and so on (remember central voxel has 0 value).

It is rather surprising that tricubes can be intrinsically characterized with the help of *local* conditions, which are simple inequalities relating matrix entries.

If tricube matrices are denoted $(z_{i,j})$, where $1 \leq i \leq 3$ and $1 \leq j \leq 3$, these inequalities are:

$$\max(z_{i-1,j}, z_{i,j-1}) \leq z_{i,j} \leq 1 + \min(z_{i-1,j}, z_{i,j-1})$$

$$\text{for } 2 \leq i \leq 3 \quad \text{and } 2 \leq j \leq 3, \tag{3a}$$

$$z_{2,1} \leq z_{1,2} \quad \text{and} \quad z_{3,2} \leq z_{2,3}, \tag{3b}$$

$$z_{3,1} \leq z_{2,2} \leq z_{1,3}, \tag{3c}$$

$$(z_{1,1} \neq z_{2,2} \quad \text{and} \quad z_{2,2} \neq z_{3,3}), \tag{3d}$$

or

$$\max(z_{1,1}, z_{2,2}, z_{3,3}) - \min(z_{1,1}, z_{2,2}, z_{3,3}) < 2.$$

The last restriction concerns values located on the first diagonal. It says that if the *oscillation* along the diagonal is greater or equal to 2 then all three values are distinct (no horizontal step in this case).

Conjecture 1. There are 40 integer matrices of dimension 3 satisfying Eqs. (3a) to (3d). These matrices correspond bijectively to the 40 Debled-Rennesson tricubes.

This result has been discovered experimentally; some partial results, which appear in Sec. 4, should provide a way of proving (fastidiously) this conjecture. A more synthetical solution using Fourier-Motzkin theory[5] is under development. This characterization of tricubes gives a simpler way of constructing them using an obvious nested loop.

Thus there are two independent generations of tricubes; the first follows their definition and uses parameters a,b,c (and γ), that is, the normal vector, while the second is a local characterization that relies on voxels with z values, ignoring normal vector (a,b,c). But it is of the greatest interest to recover these parameters directly from tricubes geometry. This is the problem of computing the gradient directly from the voxels. It is treated now.

Obviously this question can be solved if naive digital planes can be *recognized*, in other words, if the four parameters a, b, c, and γ can be recovered from the knowledge of a large enough number of voxel coordinates. This problem is solved in Ref. 4, where an incremental algorithm is given. But it would be wasteful, and not very instructive, to apply such a deep algorithm to such simple objects. It is rather clear that these numbers a, b, and c are present but hidden somewhere in the nine-point tricube geometry.

3 Finding Tricube Gradient

Results given in Ref. 4 and intuition suggest that parameters a, b, and c of the tricube normal vectors should be recoverable from the *Euclidean convex hull* of their nine points. But the complexity of 3D convex hull algorithms is well known; they are even more demanding than digital planes recognition algorithm. After tricube definition, it is easy to observe that the origin, or central point, is seldom a vertex of their convex hull. Taking this into account means that their convex hull is almost the same if we restrict to the eight border points. Adding the central point at the end is not very costly. We thus obtain a specialized 3D convex hull algorithm (not included in this paper) well adapted to that kind of wristband point sets.

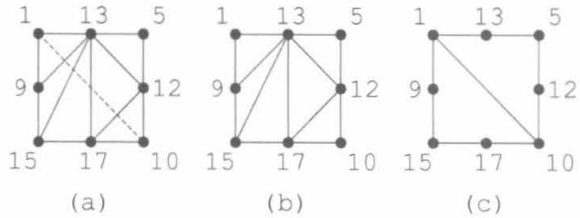


Fig. 2 Convex hull of a 3×3 mask.

When applied to all tricubes, this algorithm gives their 3D convex hull, unvealing how their normal vector (a,b,c) can be recovered. The following classification and a new gradient algorithm result.

For any tricube T we denote by Γ_3 the 3D Euclidean convex hull of its nine integer points. The projection of this small polyhedron on the (x,y) coordinate plane is contained in the square $S=[-1,1]^2$. The part of Γ_3 that projects on the S boundary is obviously made of four vertical triangles (which can degenerate into line segments); it is denoted by B (for *boundary*).

The complementary part $\Gamma_3 \backslash B$ is homeomorphic to the union of two open disks, whose closures are denoted Δ^+ and Δ^-. They are, respectively, the *upper* and *lower* parts of Γ_3. Each projection of Δ^+ and Δ^- to the (x,y) coordinate plane gives a triangulation of the square S. These projections are, respectively, denoted by D^+ and D^-. It is clear that the Γ_3 structure can be easily recovered by lifting triangulations D^+ and D^- back to space.

The most interesting part of the Γ_3 structure, leading to the classification of tricubes, is the set of edges of triangulations D^+ and D^-, which are not contained in the S boundary. We respectively denote these sets of inner edges by E^+ and E^-, while their union is denoted by $\mathscr{E}$.

Let us explain, using one example, how the convex hull of a wristband can be visualized with the help of its two associated triangulations D^+ and D^-. Vertices of this example are $(-1,-1,15)$, $(0,-1,17)$, $(1,-1,10)$, $(1,0,12)$, $(1,1,5)$, $(0,1,13)$, $(-1,1,1)$ and $(-1,0,9)$. These triangulations are given in Fig. 2. Fig. 2(b) represents D^+; it indicates that if we look at the convex hull Γ_3 from above, its apparent contour is the square $S=[-1,1]^2$ divided into six triangular faces. Fig. 2(c) represents the view from below, D^-, made of two triangular faces. All faces are represented in Fig. 2(a). Notice that four triangular faces of Γ_3 are not visible in Fig. 2 as they project on the edges of the square S. But elements of E^+ and E^- (there are respectively five and one of them in our example) enable us to reconstruct these missing faces.

Returning to the general case, we easily see that Γ_3 is planar (in the Euclidean sense) if and only if $\mathscr{E}$ is void.

In the first case B is a parallelogram and we recover the normal vector, or gradient, by computing the vector product of two vectors of B projecting on two consecutive sides of square S.

This way of recovering the normal vector also applies when E^+ or E^- is void, meaning that one face of Γ_3 is a parallelogram, which projects on the filled square S. There

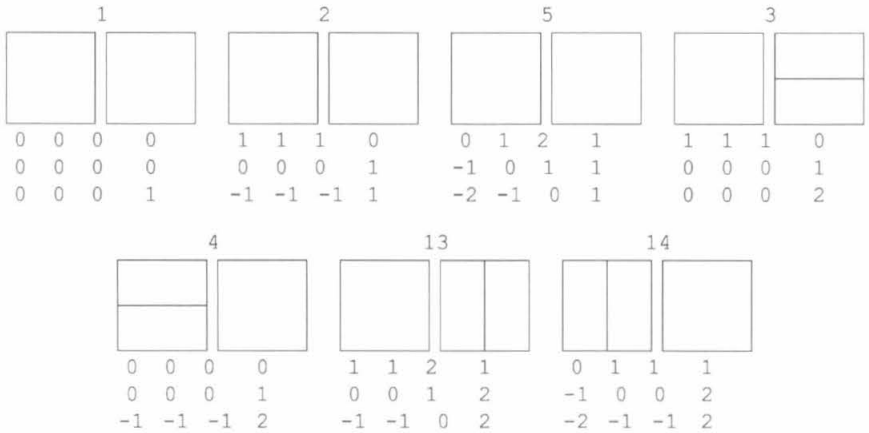


Fig. 3 Seven flat tricubes.

are, altogether, seven tricubes having such convex hulls. We call them *flat* tricubes. Their corresponding 3D convex hull Γ_3, described through triangulations D^+ and D^-, is shown in Fig. 3, with their matrices and normal vectors beneath. They are also numbered in accordance with Fig. 1.

The remaining cases split into two subfamilies.

In the second case, both E^+ and E^- contain exactly one edge; in this case (called *double roof*) the vector product of their inverse image is the sought for gradient (a,b,c). This occurs in 14 cases, shown in Fig. 4. Note that the vector product is computed with the 3D vectors that project on the two vectors of E^+ and E^-. Example 1 illustrates this gradient computation.

In the third case, Γ_3 contains a triangular face whose projection on S contains 0 and that has at least two sides nonparallel to the x and y axes; that is, two sides whose projection belongs to E^+ or E^-. There are 19 such cases where the normal vector is given by the vector product of the two mentioned edges. Relevant convex hulls have a

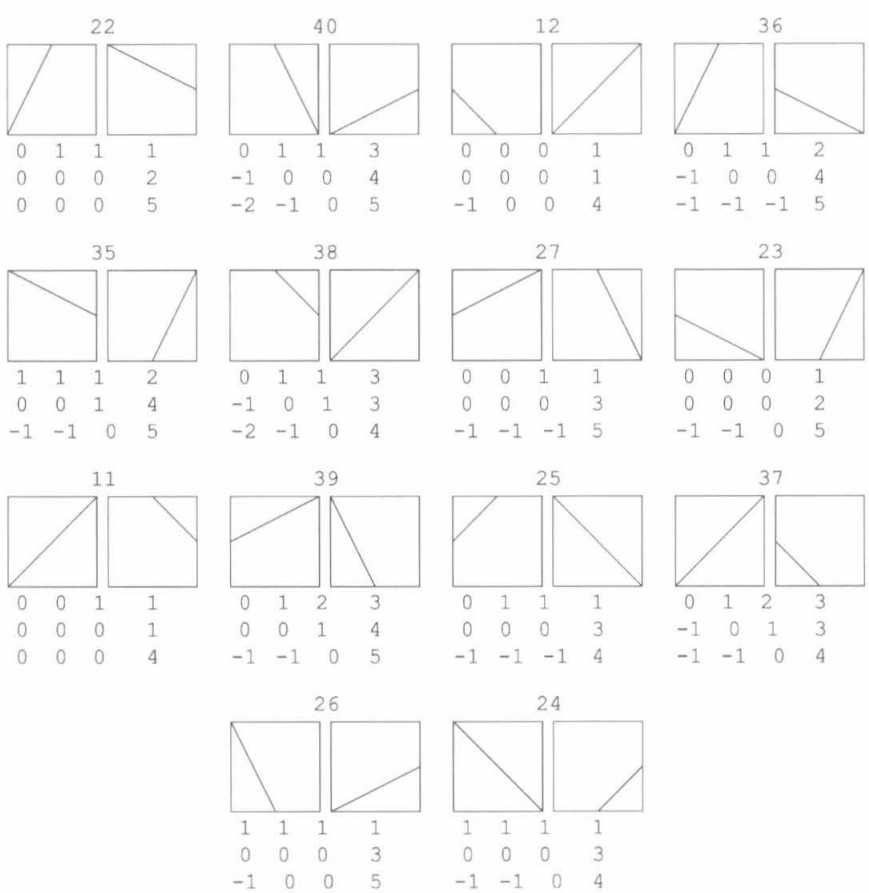


Fig. 4 Fourteen double roof tricubes.

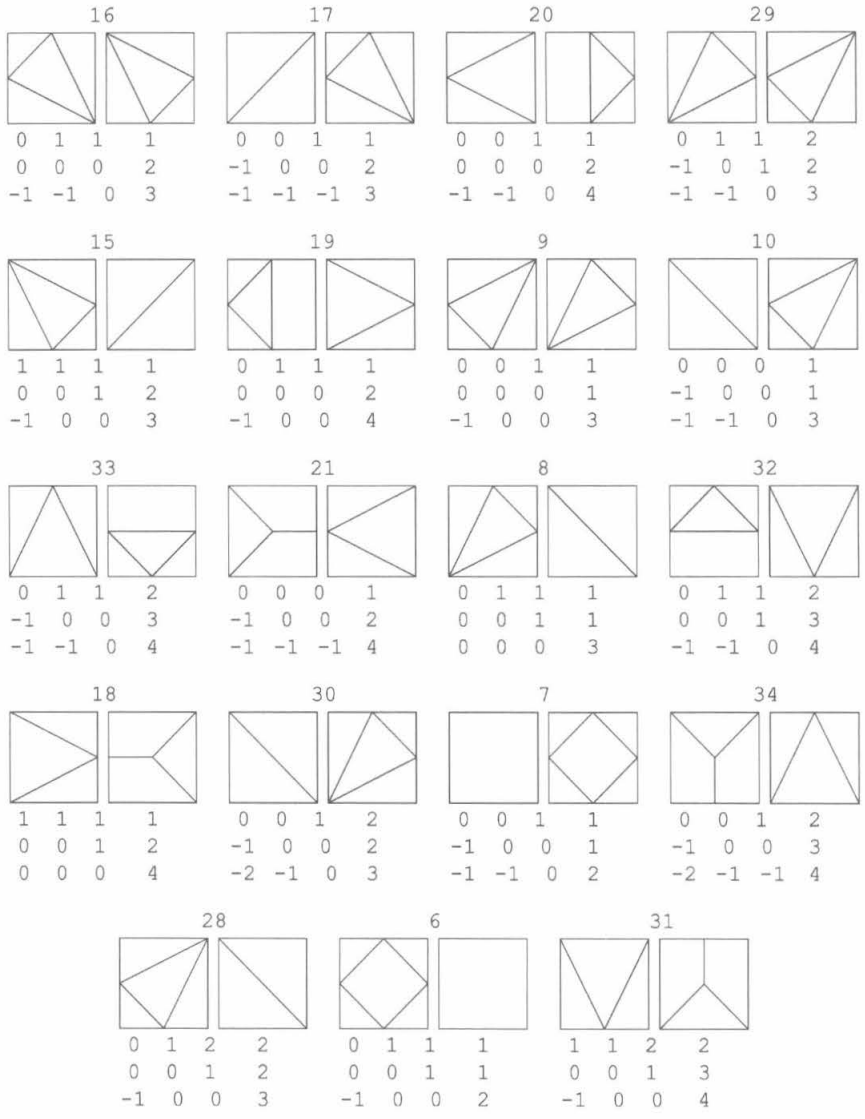


Fig. 5 Nineteen terrace tricubes.

little *terrace*. These are shown in Fig. 5. Example 2 explains this last case of our gradient algorithm.

Example 1. Let us find gradient of tricube T_{27}, the seventh double roof in Fig. 4(c). Fig. 6(a) reproduces this tricube and shows the coordinate axes. The first vector is upper roof ridge, whose components are $v_1=(-2,1,-1)$ (recall S side length is equal to 2). The second vector is $v_2=(-1,2,-1)$. Vector product $v_1 \wedge v_2$ is (1,3,5), the normal vector right to the matrix of Fig. 4.

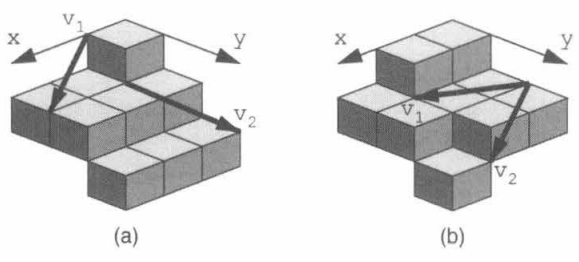


Fig. 6 Gradient computations: (a) double roof T_{27} and (b) terrace tricube T_{19}.

Example 2. This example computes gradient of terrace tricube T_{19}, the sixth tricube of Fig. 5; a copy of it appears on Fig. 6(b). We must choose the triangle for which 0 is an inner vertex; this eliminates the upper triangle. Vectors components are: $v_1=(2,-1,0)$, $v_2=(2,1,-1)$. Again the resulting vector product gives the normal vector (1,2,4). Similar computations can be done for all other tricubes.

4 Recovering Coordinates from Triangulations

It is rather interesting to observe that knowing the two triangulations D^+ and D^- associated to a given tricube T, enables us to recover its matrix, that is, the z components of its nine points.

As *extremal points* of the 3D convex hull Γ_3 is used, we insist that not every voxel of this set is an extremal point.

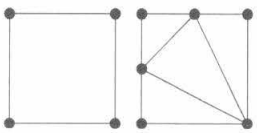


Fig. 7 Extremal points of tricube T_{17}.

Fig. 7 illustrates extremal points of $\Gamma_3(T_8)$. Seen from above, four of them appear, from below two more show up.

In all cases the four vertices of S are always projection of extremal points of Γ_3. Other voxels of Γ_3 may project as midpoints of S edges. A point of S is a vertex of a triangulation if and only if it lifts to an extremal point of Γ_3. To be precise, if three voxels $\tilde{A}$, $\tilde{B}$, and $\tilde{C}$ of a tricube T are lined up in space, the edge they define on polyhedron Γ_3 has only two extremal points; in this case, the midpoint is not extremal and is not a vertex of the triangulations. This occurs for the lowest and rightmost sides on Fig. 7.

The following proposition is a general property of tricube matrices resulting directly from the inequalities of Eq. (3).

Proposition 1. Let AB be one edge of square S where $x_A < x_B$ or $y_A < y_B$. Let us denote by z_A and z_B the z values taken by a tricube T at A and B. Let M denote the midpoint of AB.

- If M is not a vertex of D^+ then we have $z_M = z_A$ and $z_B - z_A = 1$.
- If M is a vertex of D^+ then we have $z_M - z_A = 1$ and $z_B = z_M$.

A similar result is true for vertices of D^-.

We easily deduce from Proposition 1 that if midpoint M of edge AB is neither a vertex of D^+ nor a vertex of D^-, then we have

$$z_M - z_A = 1 \quad \text{and} \quad z_B - z_M = 1,$$

or

$$z_A = z_M = z_B,$$

i.e., points A, M, and B are collinear.

Let us explain how these results can be used to eliminate configurations or to recover z values. If we examine Fig. 8(a), which describes the possible convex hull of a tricube, we see that AB and AD contain no vertex of D^+ or D^-. Thus z is linear on these segments. But if $z_A = 0$ and z_B and z_D values were supposed equal to 2, then we would have $z_M = 2$, $z_N = 2$ and $z_C = 3$. But this would imply that A, B, D and M are coplanar (after an obvious determinant computation) contradicting existence of edge AM.

On the other hand, if we suppose $z_A = z_D = 0$ and let $z_B = 2$, then the former proposition implies that $z_M = 1$ and $z_N = 0$. But this would eliminate edge MD. Because Fig. 8(a) is symmetric, this structure cannot fit any tricube.

Figure 8(b) shows another possible convex hull similar to the example of Fig. 8(a). This time AB and CD contain no vertex, thus z is linear on these segments. We can begin by fixing the value $z_A = -1$ and suppose that z is constant on AB, giving $z_B = -1$. Thus $z_N = -1$ and $z_M = 0$, $z_C = z_D = 0$; all other values can be deduced from these at once; we recognize tricube number 8. All other choices lead to incompatible situations.

A systematic study of all tricubes is certainly possible using this kind of reasoning. But this tedious task has not yet been done; a complete treatment of all the cases by a formal computer system is currently being developed.

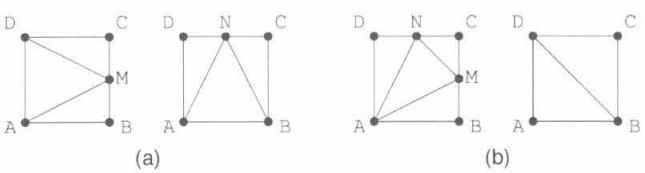


Fig. 8 Tricube reconstructed from projection.

5 New Edge Extractor for General Images

To understand what tricubes and their convex hulls can bring to image processing, we observe that their associated Euclidean plane is given by two edges of their convex hull Γ_3. These two edges generally define a *large triangle* of Γ_3, which is a *face* of this polyhedron in case of flat or terrace tricubes (cf. Figs. 3 and 5) or not, as in case of double-roof tricubes (cf. Fig. 4). If we define *regular* points of an image as those where a 3×3 piece of digital plane exists, these points, and their associated gradient, can be found using knowledge of all 40 tricubes. The global knowledge can be organized in several ways to build efficient look-up tables. Observe, for example, that only four combinatorially distinct tricubes rows (and columns) appear in the whole family.

However, except in specialized domains, common images generally do not have many regular points. As 3×3 masks are frequently used, the convex hull Γ_3 of the eight neighbors of the current point can always be considered. In this case, our former study of Γ_3 and the very little computation in the following, suggest approximating the Euclidean tangent plane by the *max area triangle* whose vertices belong to the eight neighbors.

If ABC is a triangle whose vertices belong to Γ_3 vertices and if D is the *farthest*, among Γ_3 vertices, from the Euclidean plane defined by ABC, denoting Γ_3 volume by $\mathscr{V}$, the ABC area by α and the distance of D to the plane by δ, we immediately see that

$$\delta \leq \frac{3\mathscr{V}}{\alpha},$$

so that the upper bound is the smallest when α is the largest, i.e., when ABC has maximum area. Of course, this does not prevent another Euclidean plane with a smaller δ from existing, but this is a rather good approximation of a Euclidean tangent plane and, moreover, the properties of this simply defined operator can be easily studied. This *max area triangle* operator, or MAT operator, is a new kind of gradient operator, having some nice properties for bound-

−1	0	1
−2	0	2
−1	0	1

−1	−2	−1
0	0	0
1	2	1

Fig. 9 Sobel masks.

ary extraction in image processing. It is defined as follows: For each image point M compute MAT leaning on the eight-neighborhood of M (there are 56 such triangles) and replace M gray levels by the area value adjusted to fit the gray level scale.

All areas can be formally precomputed, which makes this algorithm especially well adapted to parallel machines.

6 Comparison of MAT and Sobel Operators

The simplicity of the MAT operator puts it in the category of the most elementary discrete operators, such as the Sobel operator, which attempts to provide the gray level variation gradient at one point of a digital image using two discrete directional derivatives. Several improvements made by other authors[3,12] also follow this line of thinking.

A comparison of Sobel and MAT operators must briefly recall the definition of the Sobel operator. Its value at any point is obtained using two 3×3 convolution masks, for the x and y directions; they are respectively defined as follows (see Fig. 9).

These masks detect gradients in the x and y directions, giving two vectors that can be, respectively, denoted by Δ_x and Δ_y. The modulus of their vector sum

$$(\|\Delta_x\|^2+\|\Delta_y\|^2)^{1/2},$$

is the value given to the pixel of the new image (after scaling to the interval of actual gray level values and rounding to the closest integer). This operator recognizes rather large local variations, but is generally unable to treat small ones. And, as large variations are concerned, it does not catch geometrical properties very accurately.

Synthetical images, produced by discretization of continuous geometrical objects, can be useful to analyze the advantages or drawbacks of image processing algorithms. As digital lines and segments are very common, let us first analyze how Sobel operator acts on them.

Let us suppose that $\mathscr{D}(a,b,\gamma,\omega)$ denotes the digital line made of the integer points (x,y) satisfying the following diophantine inequalities

$$\gamma\leqslant ax+by<\gamma+\omega, \quad \omega=\max(|a|,|b|),$$

where, a, b, γ, and ω are also integers. Let us suppose that the gray level of this *image* is equal to 1 and the background pixels have a gray level of 0 ($\mathscr{D}$ is grayed in Fig. 10). Applying the Sobel operator on it gives another image whose values (we use the square of modulus $\|\Delta_x\|^2+\|\Delta_y\|^2$) are given in Fig. 10. The following result, which can be proved easily, gives the theoretical content of Fig. 10.

- On points of $\mathscr{D}$ the values given by Sobel operator are 2 if the point is located at the beginning or the end of a horizontal step and 0 if the point is a middle point of $\mathscr{D}$.
- On the two bordering digital lines $\mathscr{D}(a,b,\gamma+\omega,\omega)$ and $\mathscr{D}(a,b,\gamma-\omega,\omega)$ the values are 10 on extremal points and 16 on middle points.
- At all points belonging to the digital (nonconnected) lines $\mathscr{D}(a,b,\gamma+2\omega,|a|)$ and $\mathscr{D}(a,b,\gamma-\omega-|a|,|a|)$ the Sobel algorithm takes the value 2.

This shows that if we use a threshold value of 2, the resulting image is made of the two digital lines $\mathscr{D}(a,b,\gamma+\omega,\omega)$ and $\mathscr{D}(a,b,\gamma-\omega,\omega)$.

Thus, if the threshold is equal to 2, Sobel operator splits a digital line into *two* identical parallel digital lines. This is true for all thresholds greater or equal to 2 and lower than 10. But, if the threshold increases over 10 the resulting image loses almost all its information and the digital lines is broken into little pieces. In all cases, the 0 value assigned to some pixels of the line causes unrecoverable damages.

MAT operator works differently. When applied to a vertical or horizontal digital line made of level h pixels, it gives the value $1/2(8h^2+4)^{1/2}$ to the line pixels and the value $(1/2)(5h^2+16)^{1/2}$ to the neighboring pixels. Thus when $h>2$, MAT operator takes its maximum value *on the line* and if a threshold value of $(1/2)(5h^2+16)^{1/2}$ is applied, the line itself can be recovered.

This shows that a grid of parallel (vertical or horizontal) lines regularly spaced by 3 pixels can be preserved by the MAT operator, which is not the case with the Sobel operator. For a general digital line, the MAT operator is a little

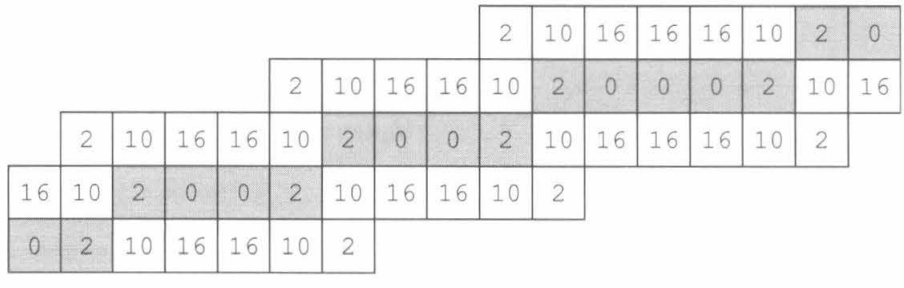


Fig. 10 Action of Sobel operator on a digital line.

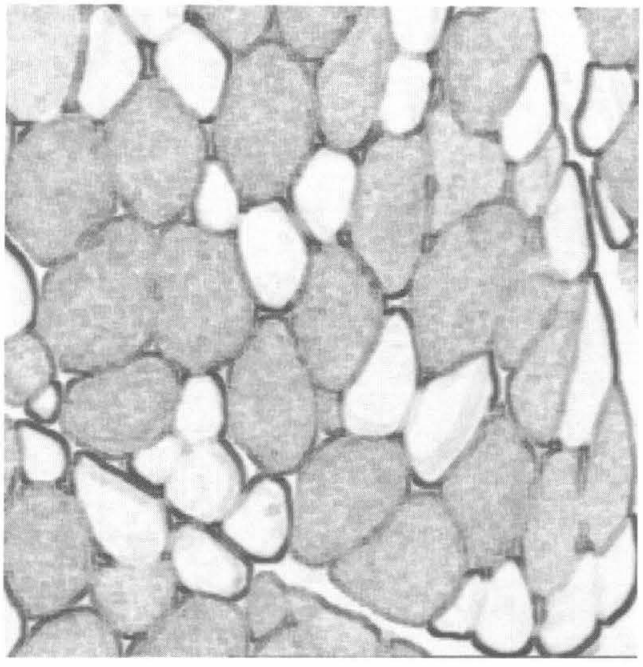


Fig. 11 MAT algorithm on muscle image.

Fig. 12 MAT algorithm on "Lena."

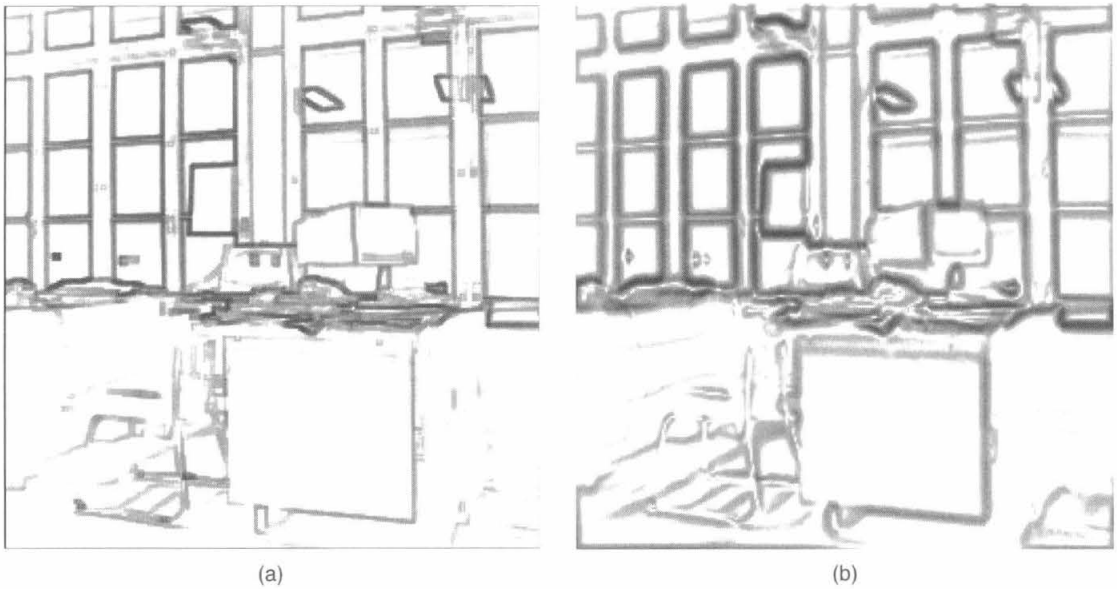
bit coarser than for vertical or horizontal lines, but it does not introduce 0 values on some of the line pixels, as is the case with the Sobel operator.

It should not be surprising then that the MAT edge extraction abilities should be better. In Fig. 11 specialists will recognize the famous *butterfly* appearing on the lower right part of the image (at least on screen!).

In the "Lena" picture (Fig. 12), one should observe the soft outlines and the smoothly treated details. In many cases, the MAT operator is comparable to the Canny-Deriche operator[13] (derived through a continuous approach). Due to its *smoothing processing* the Canny-Deriche operator is slightly superior in the average, but the MAT operator comes very close to it, and even goes beyond it on some images like the well known "office" of Fig. 13, where window frames are better treated by the MAT operator.

7 Conclusion

Two apparently nonrelated kinds of results have been presented here. First was a discrete approach to regular digital surfaces using tricubes, giving all local configurations and a gradient algorithm to recover their normal vector. Second was a general gradient operator for gray level images that almost reaches the precision of continuous edge extraction operators.



(a) (b)

Fig. 13 Comparing (a) MAT and (b) Canny-Deriche operators.

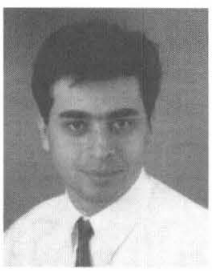
The connection between these subjects is realized by 3D convex hull geometry. This shows that research on *discrete* differentiability, for example, the way a tangent tricube evolves on a digital surface, may have practical and useful by-products.

As this subject is still new, many questions remain to be settled and could be basic for the future, in particular, languages built by tricubes (or smallest pieces) on surfaces as well as topology relating neighboring tricubes. Some refinements of the MAT operator itself have been undertaken to achieve thin contours and image smoothing (see Ref. 14).

References

1. W. E. L. Grimson, *From Images to Surfaces*, M.I.T. Press (1986).
2. R. Haralick and L. Shapiro, *Computer and Robot Vision*, Vol. 1, Addison Wesley (1992).
3. J. M. Chassery, A. Montanvert, *Analyse d'images: filtration et segmentation*, Masson, Paris (1995).
4. I. Debled-Rennesson, "Etude et reconnaissance des droites et plans discrets," Thesis, Université Louis Pasteur, Strasbourg, France (Dec. 1995).
5. J. Françon, J.-M. Schramm, and M. Tajine, "Recognizing arithmetic straight lines and planes," in *Discrete Geometry for Computer Imagery, Proc. DGCI'96*, Lyon, France, Springer Lecture Notes in Computer Science, Vol. 1176, pp. 141–149 (1996).
6. C. E. Kim, "Three-dimensional digital planes," *IEEE Trans. Pattern Anal. Mach. Intell.* **6**, 639–645 (1984).
7. C. E. Kim and I. Stomenović, "On the recognition of digital planes in three-dimensional space," *Pattern Recog. Lett.* **12**, 665–669 (1991).
8. I. Stojmenović and R. Tošić, "Digitization schemes and the recognition of digital straight lines, hyperplanes and flats in arbitrary dimensions," *Vis. Geomet. Contemp. Math. Ser.* **119**, 197–212 (1991).
9. P. Veelaert, "On the flatness of digital hyperplanes," *J. Math. Imag. Vis.* **3**, 205–221 (1993).
10. P. Veelaert, "Digital planarity of rectangular surface segments," *IEEE Trans. Pattern Anal. Mech. Intell.* **16**, 647–652 (1994).
11. J.-P. Reveillès and J. Yaacoub, "Smooth image processing," in *Vision Geometry V, Proc. SPIE* **2826**, 206–215 (1996).
12. R. Klette and P. Zamperoni, *Handbook of Image Processing Operators*, Wiley and Sons, New York (1996).
13. R. Deriche, "Fast algorithms for low-level vision," *IEEE Trans. Pattern Anal. Mach. Intell.* **12**, 78–87 (1990).
14. J. Yaacoub, "Enveloppes convexes de réseaux et applications au traitement d'images," Thesis, Université Louis Pasteur, Strasbourg, France (1997).

Jean-Pierre Reveillès graduated in mathematics from Paris VII University where he obtained a Third Cycle Thesis in algebraic topology. He defended a State Thesis on discrete geometry at Strasbourg University and is now with Auvergne University, Clermont-Ferrand. His main interest is the study of discrete analogues of topology and geometry for a better use of these tools in computers.



J. Yaacoub received a PhD from the Louis Pasteur University, Strasbourg, France, in 1997. He is currently with the Laboratoire des Sciences de l'Image, de l'Informatique et de la Télédétection (LSIIT), Strasbourg. His research interests include digital geometry, arithmetic, computational geometry and image processing.