Computers & Graphics 33 (2009) 2–10

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Technical Section

Normals estimation for digital surfaces based on convolutions ☆

Sébastien Fourey [a,*], Rémy Malgouyres [b]

[a] GREYC, UMR6072, ENSICAEN, 6 bd maréchal Juin, 14050 Caen Cedex, France
[b] LAIC, Univ Clermont 1, EA2146, BP 86, 63172 Aubiére Cedex, France

ARTICLE INFO

Article history:
Received 24 June 2008
Received in revised form
12 November 2008
Accepted 13 November 2008

Keywords:
Digital surfaces
Normal estimation
Convolution

ABSTRACT

In this paper, we present a method that we call *on-surface convolution* which extends the classical notion of a 2D digital filter to the case of digital surfaces (following the *cuberille* model). We also define an averaging mask with local support which, when applied with the iterated convolution operator, behaves like an averaging with large support. The interesting property of the latter averaging is the way the resulting weights are distributed: given a digital surface obtained by discretization of a differentiable surface of $\mathbb{R}^3$, the masks isocurves are close to the Riemannian isodistance curves from the center of the mask. We eventually use the iterated averaging followed by convolutions with differentiation masks to estimate partial derivatives and then normal vectors over a surface. The number of iterations required to achieve a good estimate is determined experimentally on digitized spheres and tori. The precision of the normal estimation is also investigated according to the digitization step.

1. Introduction

Estimation of geometrical and differential properties and quantities of objects known through their digitizations is an important goal of discrete geometry. One of the classical problems is simply to measure the length of a curve (or a perimeter) in the digital plane [1,2]. One may also quote the estimation of tangents or normals to a curve [3], normal vectors over a surface [4], or area of a digital surface [5–7].

In 2D, a whole set of methods rely on the *digital straight segments* recognition algorithm [8] used to find maximal line segments in a curve, which may in turn be used to estimate the curve's length or its tangent vectors [3]. These methods have been extended to the 3D case with digital plane recognition to estimate the area of the surface of a 3D digital object [9]. Directional tangent estimation based on straight segments recognition was used in [10] to compute normal vectors on a digital surface and later in [11] for the nD case. A first remark about this set of methods is that they are sensitive to noise.

In the case of digital surfaces, another method was introduced by Papier and Françon [12,13] to estimate the normal vector field. It is based on a weighted averaging of the canonical normals in a neighborhood of each surfel. Their method generalizes to large neighborhoods the approach proposed by Chen et al. [14] and is very close to the one we propose here, although it differs in at least two points: umbrellas in Papier's method grow following a breadth-first traversal of the surfels v-adjacency graph, whereas our method may be seen as the result of an averaging process using masks which grow in a more *geodesic* and isotropic way (see Section 4.2). Also, their averaging process applies on canonical normal vectors, whereas our method relies on the averaging of the surfel centers. Furthermore, very few tests have been conducted by Papier to determine the optimal size of the neighborhood taken into account by the averaging process.

The normal estimation method introduced here (Section 4.1) is based on the notion of *on-surface convolution* (Section 3) which extends to digital surfaces the classical 2D filters used in image processing. Using an averaging mask defined locally, we apply an iterated convolution operation on the centers of the surfels. Then, we use two orthogonal differentiation operators on the resulting centers to estimate partial derivatives, and by a cross product we obtain normal vectors. We will study in Section 4.3 the optimal number of convolution iterations for the normal estimation on two basic shapes: a sphere and a torus.

Some conclusions and perspectives are presented, including the problem of higher order derivatives and curvature estimation.

2. Digital voxel objects and digital surfaces

In this paper, we simply call a *digital object* a subset of $\mathbb{Z}^3$, the classical 3D grid. Such an object is seen as a set of unit cubes called *object voxels* centered at points with integer

☆ This work was supported by the French National Agency of Research under contract GEODIB ANR-06-BLAN-0225.

* Corresponding author. Tel.: +33 231 45 29 25; fax: +33 231 45 26 98.
E-mail addresses: Sebastien.Fourey@greyc.ensicaen.fr (S. Fourey), Remy.Malgouyres@laic.u-clermont1.fr (R. Malgouyres).

0097-8493/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cag.2008.11.003

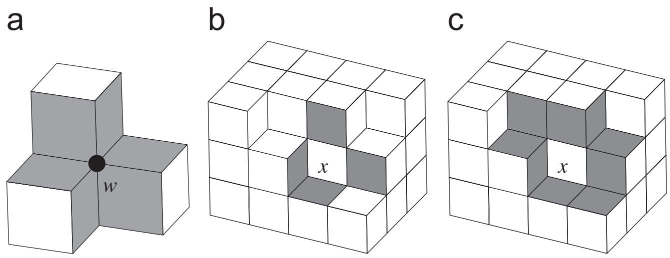

Fig. 1. Loops and neighborhoods on a digital surface. (a) A loop of surfels (in gray). (b) The e-neighborhood of x. (c) The v-neighborhood of x.

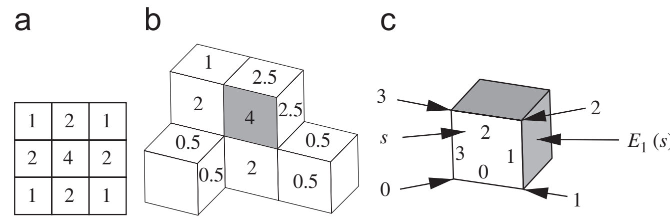

Fig. 2. Illustrations of the masks definition. (a) A 2D mask. (b) A surfel x (in gray) and the values of $16 \cdot W_{\text{avg}}(x, y)$ for the surfels y of $N_v(x) \cup \{x\}$. (c) Ordering of vertices and edges for a given surfel s.

coordinates. *Background voxels* are voxels that do not belong to the object.

The surface of a digital object can be defined as a set of *surfels*, provided with relevant adjacency graphs. Surfels are unit squares that are shared by two 6-adjacent voxels. There are exactly six types of surfels according to the direction of their normal vectors. Thus, a surfel can be uniquely defined by the data of its center's coordinates and its orientation. In the sequel, a surfel is a pair $(p, \vec{n})$ where $p \in \mathbb{R}^3$ (the center) and $\vec{n} \in \{(\pm 1, 0, 0), (0, \pm 1, 0), (0, 0, \pm 1)\}$ (the normal vector). A *digital surface* is a set of surfels which is the set of all the surfels of a digital object.

We will use in the sequel the two functions σ and v which associate to a surfel $s = (p, \vec{n})$, respectively, its center $\sigma(s) = p$ and its normal vector $v(s) = \vec{n}$.

We can define two adjacency relations between surfels: the e-adjacency and the v-adjacency relations. See [15] for further details.

A relation of e-adjacency (see Fig. 1(b)) and v-adjacency (Fig. 1(c)) can be defined between some surfels that share an edge or a vertex. Note that the considered adjacency relation (6, 18) on the set of voxels must be taken into account when defining the e-adjacency and v-adjacency relations [16]. In this way, a surfel has exactly four e-neighbors, but has a variable number of v-neighbors.

Next, we define a *loop* in a digital surface Σ as an e-connected component of the set of the surfels of Σ which share a given vertex w. For example, if Σ is the surface of the object depicted in Fig. 1(a) (which is made of three voxels), then the vertex w defines two loops: one that contains the six gray surfels, and another one in the back with three surfels. Two surfels are *v-adjacent* iff they belong to a common loop of Σ.

3. On-surface convolution

The work presented in the next sections illustrates the use of *on-surface convolution*, which we introduce here. In the sequel of the paper, Σ is a digital surface and S is a vector space over $\mathbb{R}$. We define the space of *digital surface filters over* Σ as the set of functions from $\Sigma \times \Sigma$ to $\mathbb{R}$.

Definition 1 (*Generalized convolution operator*). For $f : \Sigma \longrightarrow S$ and $F : \Sigma \times \Sigma \longrightarrow \mathbb{R}$, we define the operator Ψ as follows:

$$\begin{aligned} \Psi_{f,F} &: \Sigma \longrightarrow S \\ & x \mapsto \sum_{y \in \Sigma} F(x, y) \cdot f(y). \end{aligned}$$

Intuitively, Ψ acts like a convolution of the values of f on the surface with a convolution kernel whose values should depend on the relative positions of two surfels. We also define the iterated operator $\Psi^{(n)}$.

Definition 2 (*Iterated convolution operator*). The *iterated convolution operator* is defined for $n \in \mathbb{N}$ by

$$\begin{cases} \Psi^{(0)}_{f,F} = f, \\ \Psi^{(n)}_{f,F} = \Psi_{\Psi^{(n-1)}_{f,F}, F} & \text{if } n > 0. \end{cases}$$

Next, we define an averaging and two derivative filters which we will use in Section 4.1 to estimate the normal field on a digital surface.

3.1. The averaging filter

In order to define convolution filters on an arbitrary digital surface, we define some local masks, and then obtain larger masks by iteration. We define a local averaging mask $W_{\text{avg}} : \Sigma \times \Sigma \mapsto \mathbb{R}$. This mask should be seen as a wrapping of the 2D classical mask (Fig. 2(a)) which follows the *local* shape of the digital surface. The choice of this mask is a heuristic. We tried several masks but this one appears to give particularly good results relating to the Riemannian metrics (see Section 4.2). Intuitively, we define this mask as a generalization of the 2D local mask (and indeed they coincide on a planar surface). The weights are the same as in 2D for the e-neighbors, but for the strict v-neighbors (i.e., v-neighbors which are not e-neighbors) the weight of which would be a unique pixel in 2D is split and distributed over the several strict v-neighbors of the loop. The global mass of the mask remains unchanged.

More precisely, let x and y be two surfels of Σ such that $y \in N_v(x)$. If y is v-adjacent but not e-adjacent to x then there is a single loop L of Σ that contains both x and y. In this case, we define $\delta_x(y) = \text{card}(L) - 3$. The number $\delta_x(y)$ is used to take into account the number of surfels in a loop containing x which are not e-adjacent or equal to x. Within a loop, all these surfels will end up with a total contribution of $\frac{1}{16}$.

If there are no such surfels, the weight $\frac{1}{16}$ is spread among the two e-neighbors of x in the loop. Thus, if y is e-adjacent to x, then y has exactly two e-neighbors in $N_v(x)$, say s and t. We define $\gamma_x(y)$ as the number of surfels in $\{s, t\}$ which are e-adjacent to x.

Now, let x be a surfel of Σ. For any surfel $y \in \Sigma$ we define the *weight* $W_{\text{avg}}(x, y)$ as follows:

$$W_{\text{avg}}(x, y) = \begin{cases} \dfrac{1}{4} & \text{if } y = x, \\ \dfrac{1}{8} + \dfrac{\gamma_x(y)}{32} & \text{if } y \in N_e(x), \\ \dfrac{1}{16 \cdot \delta_x(y)} & \text{if } y \in N_v(x) \backslash N_e(x), \\ 0 & \text{if } y \notin N_v(x). \end{cases}$$

We may say that W_{avg} defines an averaging mask because of the following property, which we prove in the Appendix.