MVA '92 IAPR Workshop on Machine Vision Applications Dec. 7-9,1992, Tokyo

Computation of Surface Curvature from Range Images Using Geometrically Intrinsic Weights

T. Kurita[1) and P. Boulanger[2)

[1)Electrotechnical Laboratory
1-1-4 Umezono, Tsukuba, Japan 305

[2)National Research Council Canada
Ottawa, Canada K1A 0R6

Abstract

This paper presents a method to compute Mean and Gaussian curvatures from range images. The first and second partial derivatives are estimated by the weighted least squares fit of a biquadratic polynomial within a local moving window. The weights are determined on the basis of surface distances and normal angular distances from the center points of the window.

1. Introduction

In recent years, range image (depth map) processing has become one of the most important topics in computer vision research. The main reason is that the quality of the digitized range data has been improved by the developments of active and passive range sensing techniques (for example, [8]). Range data provide explicit geometrical information about the shape of visible surfaces. Using this basic information, some problems in 3-D object description and recognition are easier to solve. Especially, one can obtain dense range maps very effectively using active range finders.

Besl and Jain [1] have proposed an attractive idea for surface characterization from the point of view of differential geometry. Mean and Gaussian curvatures (surface curvatures) are invariant under rigid transformation. Smooth surfaces are locally characterized by them and are classified into one of eight surface types using a combination of their signs. Thus, if we could compute local surface curvatures accurately, one would be able to segment range images into several surface regions with similar surface types called the topographic primal sketch [6].

Since differential geometry is a theory for smooth differentiable surfaces, one must take into account the fact that real range images have discontinuities (depth and orientation) which will influence the computation of the curvatures. In order to prevent this problem Fan [5] proposed to detect discontinuities first and then do a local curvature measurement. Yokoya [9] proposed a method which employs a selective surface fit. The best local window which provides a minimum fitting error among the covering windows is selected and is used for curvature estimation.

Boulanger [3] proposed a new smoothing filter for range images which is invariant to viewpoint and is capable of preserving depth and orientation discontinuities. The filter employs two new surface distance, one is the length of the minimum trajectory joining two points on the surface, and the second one is the normal angular distance, which is defined as the average angular variation of the normal vector along the minimum trajectory joining two points on the surface.

In this paper, the first and second partial derivatives are determined by the weighted least squares fit of a biquadratic polynomial within a local moving window based on these distances. The present weighting method assigns small weights for the points which are not geometrically compatible with the center point of the window. Thus, the derivatives are estimated mostly based on the points which are compatible with the center point. Since the distances across a discontinuity are large, small weights are assigned to points which are on the other side of that discontinuity. Then, surface curvatures are computed from these derivatives according to the definitions.

In section 2, we review the concept of intrinsic surface distance and normal angular distance. Then an algorithm to compute these distances within a window is presented. In section 3, we show how to use these distances to do a weighted least squares fit. In section 4, we show experimental results of Mean and Gaussian curvature computation using our method and then compare it with the results produced by a non-weighted least squares method.

2. Surface and Normal Angular Distance

In this section, we will briefly review concepts of differential geometry [4], and then define the notion of intrinsic surface distance and normal angular distance[3]. We will also present an algorithm to compute these distances within a window.

2.1. Differential Geometry of Range Images

Usually, range data is presented in the form of a real matrix $z(x, y)$. Consider a graph surface, namely the graph of a differentiable function $z = h(x, y)$, where (x, y) belong to an open set $U \subset R^2$. Let us parametrize the surface by

$$\eta(x, y) = (x, y, h(x, y)),\ (x, y) \in U. \tag{1}$$

Then the first and second partial derivatives are

$$\eta_x = (1, 0, h_x),\ \eta_y = (0, 1, h_y),$$

$$\eta_{xx} = (0, 0, h_{xx}),\ \eta_{xy} = (0, 0, h_{xy}),\ \eta_{yy} = (0, 0, h_{yy}).$$

Thus the surface normal at the point (x, y) is given by

$$N(x, y) = \frac{\eta_x \wedge \eta_y}{|\eta_x \wedge \eta_y|} = \frac{(-h_x, -h_y, 1)}{(1 + h_x^2 + h_y^2)^{1/2}}, \tag{2}$$

where $\wedge$ denotes the vector product.

The Gaussian and mean curvatures are

$$K = \frac{h_{xx}h_{yy} - h_{xy}^2}{(1 + h_x 2 + h_y^2)^2}, \tag{3}$$

$$H = \frac{(1 + h_x^2)h_{yy} + (1 + h_y^2)h_{xx} - 2h_x h_y h_{xy}}{2(1 + h_x^2 + h_y^2)^{3/2}} \tag{4}$$

2.2. Surface and Normal Angular Distances

Consider a curve $\alpha(t)$ on a surface. The arc length s_α between two points $p = \alpha(t_p)$ and $q = \alpha(t_q)$ along the curve is given by

$$s_\alpha(p, q) = \int_{t_p}^{t_q} |\alpha'(t)| dt. \tag{5}$$

Since only the z-values are available on discrete points of the (x, y) coordinate for a digitized range image, we must have a discrete form of equation (5) to compute the arc length between two points along the curve.

Let us consider a partition of a curve $\alpha(t)$ defined as $\alpha(t_i), t_p = t_0 < t_1 < \ldots < t_k < t_{k+1} = t_q$. If the steps of the piecewise approximation are sufficiently small, one can approximate the curve by a linear equation of the form

$$\alpha(t) = (x(t_i) + a_i(t - t_i), y(t_i) + b_i(t - t_i), z(t_i) + c_i(t - t_i)), \tag{6}$$

where $a_i = \frac{x(t_{i+1}) - x(t_i)}{t_{i+1} - t_i}$ and b_i and c_i have similar expressions based on y and z. Then the derivative $\alpha'(t)$ of α with respect to t is given by

$$\alpha'(t) = (a_i, b_i, c_i) \text{ for } t_i < t < t_{i+1}.$$

Thus a discrete approximation of the arc length between p and q is given by

$$s_\alpha(p, q) = \sum_{i=0}^{k} \sqrt{a_i^2 + b_i^2 + c_i^2}. \tag{7}$$

This is equivalent to a polygonal approximation of the surface.

Then the surface distance d_S, namely the minimum distance among all trajectories joining the two points, is defined by

$$d_S(p, q) = \min_\alpha s_\alpha(p, q). \tag{8}$$

Note that the value of d_S will be large if the minimum trajectory goes across a depth discontinuity.

However, this distance is not very sensitive to orientation discontinuity. Therefore, it is necessary to consider another distance which is sensitive to the change of surface orientation. The normal angular distance is defined as the average angular variation of the normal vector along the trajectory joining the points p and q by

$$d_A(p, q) = \frac{1}{(t_q - t_p)} \int_{t_p}^{t_q} \cos^{-1}(< N(t_p), N(t) >) dt, \tag{9}$$

where $N(t_p)$ is the normal vector at t_p.

From the definition (9), it is obvious that $d_A(p, q)$ is equal to zero if the normal vector along the trajectory is constant. If, however, the trajectory goes across an orientation discontinuity, the value of d_A will be large. Moreover, this measure is also independent to viewpoint. The discrete form of equation (9) is given by

$$d_A(p, q) = \frac{1}{k} \sum_{i=1}^{k} cos^{-1}(< N(t_0), N(t_i) >). \tag{10}$$

2.3. Algorithm to Find Minimum Trajectory

To obtain surface distances from the center point to the other points in a moving window, we need to find minimum trajectories from the center point to all other points in the window. From the definition (8), we can design an efficient algorithm by using Single-Source Shortest Paths Algorithm for weighted graphs (for example, see [7]).

The vertices of the graph correspond to the points in the window and the edges represent neighboring connections of points. From the equation (7), the arc length of the edge between points $(x(t_i), y(t_i))$ and $(x(t_{i+1}), y(t_{i+1}))$ is given by

$$\sqrt{a_i^2 + b_i^2 + c_i^2}.$$

The running time of this algorithm is $O((|E| + |V|)\log|V|)$, where $|E|$ denotes the number of edges.

In order to compute the normal angular distance, we need to estimate the surface normal at each point. In the following experiments, we used the estimates obtained by a least squares fitting of a plane within local 3 by 3 windows (for example [3]).

Once we have estimates of the surface normal, the normal angular distance is easily computed by tracing the minimum trajectory obtained by the previous algorithm.

3. Mean and Gaussian Curvatures Computation

In this section, we will describe how to compute weights from these distances and how to apply this weight function to curvature computation.

3.1. Geometrical Weights

To assign large weights for points which are geometrically compatible with the center point of the window and small weights for points which are not compatible, we use Gaussian weights based on the surface and normal angular distances. For surface distance $d_S(p,q)$, the weight $w_S(q)$ is defined by

$$w_S(q) = exp(-\frac{d_S^2(p,q)}{2\sigma_S^2}),$$

where p is the center point of the window and σ_S is a scaling parameter. For normal angular distance $d_A(p,q)$, the weight $w_A(q)$ is defined by

$$w_A(q) = exp(-\frac{d_A^2(p,q)}{2\sigma_A^2}),$$

where σ_A is also a scaling parameter.

By combining these two weights, we have

$$w(q) = w_S(q)w_A(q) = exp(-\frac{d_S^2(p,q) + \beta d_A^2(p,q)}{2\sigma^2}), \quad (11)$$

where β is a parameter to account for the relative importance of the normal angular distance with respect to the surface distance.

The present weights have the following attractive properties:

- The weights are independent of the viewpoint. In particular, they preserve local curvature information.
- Larger weights are assigned for the points which are geometrically more compatible with the center point.
- The weights are sensitive to both depth and orientation discontinuities. Thus, small weights are assigned to points which are in the opposite side of a discontinuity.

3.2. Surface Fitting

Once we have the weights, the computation of derivatives is straightforward. It is common to fit a second-degree surface to an L by L window centered at each point of the range surface, where L is usually odd. To fit a surface, we use the weighted least squares method. The partial derivatives of the fitting surface at the center of the window are taken as the estimates of the partial derivatives of the range data at that point.

Let the range image depth values inside the window be denoted by h_k, $k = 1, \ldots, N$ ($N = L^2$) and let the weights of corresponding points be w_k, $k = 1, \ldots, N$. Let the fitting second-degree surface be

$$\hat{h}(x,y) = a_1 + a_2 x + a_3 y + a_4 x^2 + a_5 xy + a_6 y^2. \quad (12)$$

The coefficients of the fitting surface are determined such that the weighted squares error

$$\varepsilon^2(\mathbf{a}) = \sum_{k=1}^{N} w_k |h_k - \hat{h}_k|^2 \quad (13)$$

is minimized.

The optimum coefficients vector $\mathbf{a}$ is given by

$$\mathbf{a} = (X^T W X)^{-1} X^T W \mathbf{h}, \quad (14)$$

where $\mathbf{a} = (a_i)$, $\mathbf{h} = (h_k)$, $W = \mathrm{diag}(w_k)$, and

$$X = \begin{bmatrix} 1 & x_1 & y_1 & x_1^2 & x_1 y_1 & y_1^2 \\ & & & \cdots & & \\ 1 & x_N & y_N & x_N^2 & x_N y_N & y_N^2 \end{bmatrix}.$$

3.3. Mean and Gaussian Curvatures

The partial derivatives of the fitting surface at the center of the window are given by

$$h_x = a_2,\ h_y = a_3,\ h_{xx} = 2a_4,\ h_{xy} = a_5,\ h_{yy} = 2a_6.$$

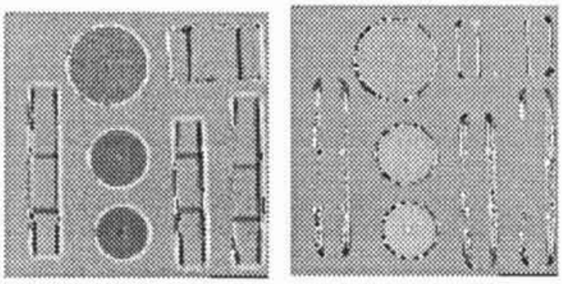

Figure 1. Mean and Gaussian curvatures estimated by the present algorithm. (a) Mean curvatures. (b) Gaussian curvatures.

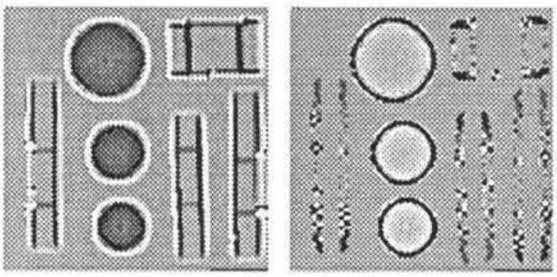

Figure 2. Mean and Gaussian curvatures estimated by the non-weighted least squares fitting. (a) Mean curvatures. (b) Gaussian curvatures.

These are considered as the estimates of the partial derivatives at the object's surface point.

By using these derivatives, the surface normal at the point is computed from equation (2). The Gaussian and mean curvatures are also easily obtained from equations (3) and (4).

4. Experimental Results

In this section, we present experimental results of Mean and Gaussian curvature computation and compare them with those obtained by the usual method based on non-weighted least squares.

Figure1 (a) and (b) are the Mean and Gaussian curvature maps estimated by the present algorithm. In this computation, we used 5 by 5 windows and set the parameters as $\sigma = 1.5$ and $\beta = 20.0$.

Figure2 (a) and (b) show the Mean and Gaussian curvature maps computed by the usual non-weighted least squares fitting of a biquadratic polynomial. The size of the moving window was also 5 by 5.

One can see the improvements of the curvature computation by using the geometrical weights, especially near discontinuities.

Acknowledgments

The authors would like to thank M. Rioux, L. Cournoyer, and J. Domey of the National Research Council of Canada who kindly provided access to the range image database.

References

[1] P.J. Besl and R.C. Jain, "Invariant Surface Characteristics for 3D Object Recognition in Range Images", Comp. Vision, Graphics, and Image Processing 33, 33–80 (1986).

[2] P. Boulanger and P. Cohen, "Stable Estimation of a Topographic Primal Sketch for Range Image Interpretation," Proc. of IAPR Workshop on CV–Special Hardware and Industrial Applications Oct. 12–14, Tokyo (1988).

[3] P. Boulanger and P. Cohen, "Adaptive smoothing of range images based on intrinsic surface properties," Proc. of the SPIE Technical Symposium on Optical Engineering and Photonics in Aerospace Sensing, Orlando, FL April 16–20 (1990).

[4] M.P. do Carmo, "Differential Geometry of Curves and Surfaces," Prentice-Hall (1976).

[5] T.J. Fan. "Describing and recognizing 3-D objects using surface properties", PhD thesis, Institute for Robotics and Intelligent Systems, School of Engineering, University of Southern California, Los Angeles, California, August 1988.

[6] R.M. Haralick, L.T. Watson, and T.J. Laffey. "The topographic primal sketch", Int. J. Robotics Res. 2(1): 50–72; 1983.

[7] U. Manber, "Introduction to Algorithms, A Creative Approach," Addison-Wesley (1989).

[8] M. Rioux, F. Blais, J.A. Beraldin and P. Boulanger, "Range image sensors development at NRC laboratories", Proc. of the Workshop on Interpretation of 3D scenes, Austin, TX, November 27–29 (1989).

[9] N. Yokoya and M.D. Levine, "Range Image Segmentation Based on Differential Geometry: A Hybrid Approach," IEEE Trans. on Pattern Anal. Machine Intell., vol. PAMI-11, no. 6, pp. 643–649 (1989).