Efficient Computation of the Area and Length of a Planar Contour Using Digital Elevation Model Data

Hiroyuki Goto
Department of Industrial & System Engineering
Hosei University
Koganei, Tokyo 184-8584, Japan

Yoichi Shimakawa
Department of Computer Science and Technology
Salesian Polytechnic
Machida, Tokyo 194-0215, Japan

Abstract

We develop an efficient computation algorithm for calculating the area and length of a closed planar contour. For a given continuous elevation function z=f(x, y), we focus on the circuit formed by z=0. Considering the Taylor expansion to second order, we approximate f(x, y) using three types of curvatures. The circuit is partitioned into multiple portions, by which each resultant becomes exactly or approximately round. The roundness is evaluated with the three curvatures. The area is computed by cumulating the area elements in a one-dimensional line integration using the Green's theorem. An illustrative numerical experiment is performed to validate the proposed method's effectiveness. Using actual digital elevation model data of an island in Japan, we draw the contours, and calculate the area and length of them. Compared with the existing method, the proposed algorithm is beneficial in reproducing smoothed contours.

Keywords
Modeling and simulation, terrain analysis, digital elevation model, contour tracing, Green's theorem

1. Introduction
We develop an efficient method for computing the area and length of a closed planar contour, the shape of which is determined by a continuous elevation function. The primary application of the algorithm is terrain analysis using digital elevation model (DEM) (Isaaks and Srivastava 1990; Jia et al. 2013) data in a class of geological information systems (GIS) (Longley et al. 2005; Lioyd 2010). Our recent research (Goto and Shimakawa 2013) proposed a new approach referred to as "Inner Skin Algorithm" (ISA). By introducing the notion of oriented shortcut along the circuit, the calculation of the area and length can be performed by one-dimensional line integration using the Green's theorem. The remarkable feature of this approach is to use a continuous elevation function in contour tracing, the coefficients of which are determined by interpolation or regression using actual DEM data. On the other hand, the well-known existing methods for contour tracing, McLain (1971) and Dobkin et al. (1990) for example, are based on a repetition of finding and connecting interceptions on portioned edges. As a consequence, the roughness of the shape and accuracy of the area of contours depend on the mesh's size such as grids and triangles. By contrast, in the approach of using a continuous elevation function, the roughness of the boundary and accuracy of the area can be adjusted irrespective of the resolution of original data.

However, the precursory version of the ISA developed in Goto and Shimakawa (2013) bases on a broken line approximation in contour tracing, and a first order approximation is used in calculating the angle between the directional and shortcut vectors, referred to as shortcut angle. Thus, the step size must be shortened significantly wherever there is a curve, and thus the computation efficiency would decrease. In the light of this, we improve the ISA, for which the step size can be stridden as long as the roundness of a local boundary does not change significantly. In particular, we focus attention on curvature, and interpolate the control points along the boundary with arcs. This approach would be beneficial in cases where there are many roundish curves on the circuit.

2. Preliminaries

2.1 Elevation function and target contour

We first give a continuous elevation function which is usually determined by interpolation. A well-known and straightforward method is to use a bicubic spline function (Li et al. 2004) below:

$$f(\boldsymbol{x}) = \sum_{i=0}^{3}\sum_{j=0}^{3} a_{ij}[(x-x_0)/h]^i[(y-y_0)/k]^j \ , \tag{1}$$

where $\boldsymbol{x}=(x\ \ y)^T$. The coefficients (x_0, y_0) , h, and k are the base coordinate, horizontal size, and vertical size of a local patch, respectively. The collection of patches forms a global surface. In addition to these, there are also other various models and techniques in creating an elevation function. Interested readers should refer to Henderson (1998).

Our main target is; given a continuous elevation function $z = f(\boldsymbol{x})$, to consider calculating the area and length of a circuit determined by

$$f(\boldsymbol{x})=0\ . \tag{2}$$

2.2 Symbols and notations

We use the following symbols and notations hereafter.

- D : target region surrounded by $f(\boldsymbol{x})=0$,
- ∂D : oriented circuit by moving counter clockwise along the boundary of D ,
- S : signed area of D ,
- L : length of ∂D ,
- $\boldsymbol{x}\times\boldsymbol{v}=x\cdot v-y\cdot u$: outer product of two vectors $\boldsymbol{x}=(x\ y)^T$ and $\boldsymbol{u}=(u\ v)^T$,
- $d\boldsymbol{x}=(dx\ \ dy)^T$: infinitesimal displacement,
- $\nabla=(\partial/\partial x\ \ \partial/\partial y)^T$: differential operator,
- $\nabla f(\boldsymbol{x})=(f_x\ \ f_y)^T$: gradient of $f(\boldsymbol{x})$,
- $\boldsymbol{e}_n=(f_x\ \ f_y)/|\nabla f|$: unit normal vector,
- $\boldsymbol{e}_d=(-f_y\ \ f_x)^T/|\nabla f|$: unit directional vector made by rotating $\boldsymbol{e}_n$ for $\pi/2$ in counter clockwise direction,
- $H(\boldsymbol{x})=\nabla\nabla^T f(\boldsymbol{x})$: Hessian matrix,
- $J(\theta)$: rotation matrix for counter clockwise direction.

We assume that there are not stationery points on ∂D; i.e. $\nabla f(\boldsymbol{x})\neq\boldsymbol{0}$. For use in later discussions, we confirm the following relations hold with respect to $\boldsymbol{x}$, $\boldsymbol{e}_n$, $\boldsymbol{e}_d$, and $J(\theta)$:

$$\boldsymbol{e}_d^T\boldsymbol{e}_n=\boldsymbol{e}_n^T\boldsymbol{e}_d=0\ ,\ \ \boldsymbol{e}_n^T\boldsymbol{e}_n=\boldsymbol{e}_d^T\boldsymbol{e}_d=1\ ,\ \ \boldsymbol{x}\times\boldsymbol{e}_n=-\boldsymbol{e}_n\times\boldsymbol{x}=-\boldsymbol{x}^T\boldsymbol{e}_d\ ,\ \ \boldsymbol{x}\times\boldsymbol{e}_d=-\boldsymbol{e}_d\times\boldsymbol{x}=\boldsymbol{x}^T\boldsymbol{e}_n\ ,$$

$$J(\theta)\boldsymbol{e}_n=\cos\theta\boldsymbol{e}_n+\sin\theta\boldsymbol{e}_d\ ,\quad J(\theta)\boldsymbol{e}_d=-\sin\theta\boldsymbol{e}_n+\cos\theta\boldsymbol{e}_d\ , \tag{3}$$

$$[J(\theta)\boldsymbol{e}_d]^T[J(2\theta)\boldsymbol{e}_n]=\boldsymbol{e}_d^T J(-\theta)J(2\theta)\boldsymbol{e}_n=\boldsymbol{e}_d^T(\cos\theta\boldsymbol{e}_n+\sin\theta\boldsymbol{e}_d)=\sin\theta\ ,$$

$$\int_0^\theta \boldsymbol{x}^T J(2\theta)\boldsymbol{e}_n d\theta=\int_0^\theta \boldsymbol{x}^T[\cos 2\theta\boldsymbol{e}_n+\sin 2\theta\boldsymbol{e}_d]d\theta=\frac{\sin 2\theta}{2}\boldsymbol{x}^T\boldsymbol{e}_n+\frac{1-\cos 2\theta}{2}\boldsymbol{x}^T\boldsymbol{e}_d=\sin\theta\boldsymbol{x}^T J(\theta)\boldsymbol{e}_n\ . \tag{4}$$

2.3 Theorem for calculating the area

For an arbitrary class C^1 vector $\boldsymbol{u}$, the following Green's theorem (Stroud and Booth 2005) holds:

$$\iint_D \nabla\times\boldsymbol{u}\,dxdy=\oint_{\partial D}\boldsymbol{u}^T d\boldsymbol{x}\ .$$

By putting $\boldsymbol{u}=(-y\ \ x)^T$, this can be expressed as

$$2\iint_D dxdy=\oint_{\partial D}\boldsymbol{x}\times d\boldsymbol{x}=2S\ . \tag{5}$$

Let an $\boldsymbol{x}$ be a solution of equation (2). The change of f for infinitesimal directional vector $d\boldsymbol{x}$ is then $df=d\boldsymbol{x}^T\nabla f(\boldsymbol{x})$. If we move along the circuit ∂D, $df=0$ follows and $\nabla f(\boldsymbol{x})$ is perpendicular to $d\boldsymbol{x}$, it is thus a normal vector on point $\boldsymbol{x}$.

If we create $\boldsymbol{dx}$ by rotating $\nabla f(\boldsymbol{x})$ in counter clockwise, denoting its length by dl, we obtain $\boldsymbol{dx} = dl \cdot J(\pi/2)\boldsymbol{e}_n = dl \cdot \boldsymbol{e}_d$. This indicates $\boldsymbol{x} \times d\boldsymbol{x} = dl \cdot \boldsymbol{x} \times \boldsymbol{e}_d = dl \cdot \boldsymbol{x}^T \boldsymbol{e}_n$. Thus, the area S in equation (5) is reduced to

$$S = \oint_{\partial D} (\boldsymbol{x}^T \boldsymbol{e}_n)/2 \cdot dl \,, \tag{6}$$

where the sign of S is positive if we move along ∂D in counter-clockwise direction.

Moreover, the length of ∂D can be calculated by the following line integration:

$$L = \oint_{\partial D} dl \,.$$

3. Proposed Algorithm

3.1 Shortcut along the boundary

To trace the circuit ∂D, we first discretize dl to a finite value δl. We then discretize $\boldsymbol{dx}$ to $\delta \boldsymbol{x}$, which we shall refer to this as shortcut vector such that $f(\boldsymbol{x}+\delta\boldsymbol{x})=0$ is followed. We make this vector by rotating θ in counter clockwise direction as $\delta\boldsymbol{x} = \delta l \cdot J(\theta)\boldsymbol{e}_d = \delta l \cdot (-\sin\theta \boldsymbol{e}_n + \cos\theta \boldsymbol{e}_d)$. We shall refer to θ as shortcut angle. We depict in Fig. 1 the relationship between the boundary, relevant vectors, and angle.

By making a Taylor expansion of $f(\boldsymbol{x}+\delta\boldsymbol{x})$ to second order, we obtain

$$f(\boldsymbol{x}+\delta\boldsymbol{x}) = f(\boldsymbol{x}) + \delta\boldsymbol{x}^T \nabla f + \delta\boldsymbol{x}^T H \delta\boldsymbol{x}/2 \quad , \tag{7}$$

where

$$\delta\boldsymbol{x}^T \nabla f = \delta l \cdot [J(\theta)\cdot \boldsymbol{e}_d]^T |\nabla f| \boldsymbol{e}_n = \delta l |\nabla f| (-\sin\theta \boldsymbol{e}_n + \cos\theta \boldsymbol{e}_d)^T \boldsymbol{e}_n = -\delta l |\nabla f| \sin\theta \quad ,\tag{8}$$

$$\delta\boldsymbol{x}^T H \delta\boldsymbol{x} = \delta l^2 \left(-\sin\theta \boldsymbol{e}_n + \cos\theta \boldsymbol{e}_d\right)^T H \left(-\sin\theta \boldsymbol{e}_n + \cos\theta \boldsymbol{e}_d\right) = \delta l^2 |\nabla f| (k \cos^2\theta + p \sin^2\theta - 2q \sin\theta \cos\theta) \quad ,$$

$$k = (\boldsymbol{e}_d^T H \boldsymbol{e}_d)/|\nabla f|, \; p = (\boldsymbol{e}_n^T H \boldsymbol{e}_n)/|\nabla f|, \; q = (\boldsymbol{e}_n^T H \boldsymbol{e}_d)/|\nabla f|. \tag{9}$$

Matrix H is the Hessian matrix at point $\boldsymbol{x}+\eta\delta\boldsymbol{x}$, where η is a constant that satisfies $(0<\eta<1)$. The quantity k is known as curvature, and p and q are the curvatures for the other directions (Lindberg 1994). In particular, p is referred to as Zuniga-Haralick operator (Zuniga and Haralick 1983).

By dividing both hand-sides of equation (7) with $\delta l |\nabla f|$, we obtain

$$g(\boldsymbol{x}+\delta\boldsymbol{x}) = g(\boldsymbol{x}) - \sin\theta + (\delta l/2)(k\cos^2\theta + p\sin^2\theta - q\sin 2\theta) \quad , \tag{10}$$

where we put $g(\boldsymbol{x}) = f(\boldsymbol{x})/(\delta l |\nabla f|)$. Furthermore, by introducing the following quantities

$$\omega_k = (k+p)\cdot \delta l/4 \;, \quad \omega_p = (k-p)\cdot \delta l/4 \;, \quad \omega_q = q \cdot \delta l/2 \;, \tag{11}$$

and noting $\cos^2\theta = (1+\cos 2\theta)/2$, $\sin^2\theta = (1-\cos 2\theta)/2$, equation (10) is transformed into

$$g(\boldsymbol{x}+\delta\boldsymbol{x}) = g(\boldsymbol{x}) - \sin\theta + \omega_k + \omega_p \cos 2\theta - \omega_q \sin 2\theta \quad . \tag{12}$$

If $\delta\boldsymbol{x}$ is a shortcut vector, $f(\boldsymbol{x}+\delta\boldsymbol{x})=0$ and $g(\boldsymbol{x}+\delta\boldsymbol{x})=0$ follow and the corresponding shortcut angle θ is thus obtained by solving an equation

$$0 = g(\boldsymbol{x}) - \sin\theta + \omega_k + \omega_p \cos 2\theta - \omega_q \sin 2\theta \quad , \tag{13}$$

the solving method of which is discussed in section 3.5.

3.2 Calculation of three curvatures

In view of equation (9), it appears that calculating the Hessian matrix H is essential for obtaining the curvatures k, p, and q. However, we can obtain approximate ω_k, ω_p, and ω_q using equation (12) and without H. Let $\delta\boldsymbol{x}_i$ (i =1, 2, 3) be the reference point which is rotated for θ_i in counter clockwise from the directional vector $\boldsymbol{e}_d$. Considering three angles $-\alpha$, 0, and α for θ_i, the following simultaneous linear equations are obtained:

$$\delta\boldsymbol{x}_1 = \delta l \cdot (-\sin\alpha \boldsymbol{e}_n + \cos\alpha \boldsymbol{e}_d) \;, \quad \delta\boldsymbol{x}_2 = \delta l \cdot \boldsymbol{e}_d \;, \quad \delta\boldsymbol{x}_3 = \delta l \cdot (\sin\alpha \boldsymbol{e}_n + \cos\alpha \boldsymbol{e}_d) \;.$$

Fig.2 depicts the positional relationships of these vectors.

We then construct simultaneous linear equations with respect to equation (12) and obtain $\boldsymbol{A}\cdot\boldsymbol{\omega} = \boldsymbol{g}$, where

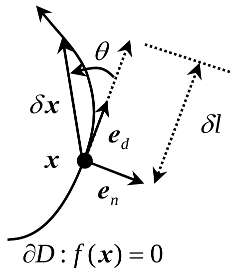

Figure 1: Relationship between the boundary, relevant vectors, and angle

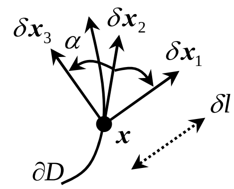

Figure 2: Three reference points for calculating the curvatures

$$\boldsymbol{A}=\begin{pmatrix}1 & \cos 2\alpha & \sin 2\alpha \\ 1 & 1 & 0 \\ 1 & \cos 2\alpha & -\sin 2\alpha\end{pmatrix}, \quad \boldsymbol{\omega}=\begin{pmatrix}\omega_k \\ \omega_p \\ \omega_q\end{pmatrix}, \quad \boldsymbol{g}=\begin{pmatrix}g(\boldsymbol{x}+\delta\boldsymbol{x}_1)-g(\boldsymbol{x}) \\ g(\boldsymbol{x}+\delta\boldsymbol{x}_2)-g(\boldsymbol{x}) \\ g(\boldsymbol{x}+\delta\boldsymbol{x}_3)-g(\boldsymbol{x})\end{pmatrix}+\begin{pmatrix}\sin\theta_1 \\ \sin\theta_2 \\ \sin\theta_3\end{pmatrix}.$$

Since $\boldsymbol{A}$ is a constant matrix, vector $\boldsymbol{\omega}$ can be calculated by multiplying $\boldsymbol{A}^{-1}$ to $\boldsymbol{g}$ from the left side.

If ∂D is a portion of an exact arc, the curvatures k, p, and q are constant along ∂D. We thus partition ∂D to multiple portions, each of which is sufficiently close to or exact arc. We then trace ∂D by stepping the portions and perform the line integration in equation (6).

3.3 Closeness to arc

Let the current portion be denoted by δD. We first confirm that δD is an exact arc if k, p, and q in equation (9) follow $k = p$ and $q = 0$. In view of equations (11) and (12),

$$\omega_k = k\delta l/2, \quad \omega_p = \omega_q = 0, \quad g(\boldsymbol{x}+\delta\boldsymbol{x}) = g(\boldsymbol{x}) - \sin\theta + k\delta l/2, \tag{14}$$

follow. If both $\boldsymbol{x}$ and $\boldsymbol{x}+\delta\boldsymbol{x}$ are on ∂D, $g(\boldsymbol{x}+\delta\boldsymbol{x}) = g(\boldsymbol{x}) = 0$ holds. Thus, letting $r(=1/k)$ be the radius of curvature, the last equality in equation (14) yields

$$\delta l = 2r\sin\theta. \tag{15}$$

This relation between δl and θ implies that portion δD is an exact arc, which can be understood in view of Fig. 3.

Next, an evaluation method for closeness to arc is developed. Let (λ_1, λ_2) and $(\boldsymbol{u}_1, \boldsymbol{u}_2)$ be the eigenvalues and unit eigenvectors of the Hessian matrix H, respectively. Since H is a symmetric matrix, the two eigenvectors are perpendicular each other and we take $\boldsymbol{u}_2$ as $\boldsymbol{u}_2 = J(\pi/2)\cdot\boldsymbol{u}_1$. These hold the following properties:

$$\boldsymbol{u}_1^T H \boldsymbol{u}_1 = \lambda_1, \quad \boldsymbol{u}_2^T H \boldsymbol{u}_2 = \lambda_2, \quad \boldsymbol{u}_1^T H \boldsymbol{u}_2 = \boldsymbol{u}_2^T H \boldsymbol{u}_1 = 0.$$

Let β be the angle from $\boldsymbol{u}_1$ to $\boldsymbol{e}_n$ in counter clockwise, which is equal to the one from $\boldsymbol{u}_2$ to $\boldsymbol{e}_d$. Recalling equation (3), and replacing $\boldsymbol{e}_n$ and $\boldsymbol{e}_d$ with $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$, respectively, we obtain $J(\beta)\boldsymbol{u}_1 = \boldsymbol{e}_n = \cos\beta\boldsymbol{u}_1 + \sin\beta\boldsymbol{u}_2$ and $J(\beta)\boldsymbol{u}_2 = \boldsymbol{e}_d = -\sin\beta\boldsymbol{u}_1 + \cos\beta\boldsymbol{u}_2$. Then, $\boldsymbol{e}_n^T H \boldsymbol{e}_n$ in equation (9) can be calculated as

$$\boldsymbol{e}_n^T H \boldsymbol{e}_n = (\cos\beta\boldsymbol{u}_1 + \sin\beta\boldsymbol{u}_2)^T H (\cos\beta\boldsymbol{u}_1 + \sin\beta\boldsymbol{u}_2) = \cos^2\beta\lambda_1 + \sin^2\beta\lambda_2. \tag{16}$$

Analogously, $\boldsymbol{e}_d^T H \boldsymbol{e}_d$ and $2\boldsymbol{e}_n^T H \boldsymbol{e}_d$ are calculated as follows:

$$\boldsymbol{e}_d^T H \boldsymbol{e}_d = \sin^2\beta\lambda_1 + \cos^2\beta\lambda_2, \quad 2\boldsymbol{e}_n^T H \boldsymbol{e}_d = -\sin 2\beta(\lambda_1 - \lambda_2). \tag{17}$$

The addition and subtraction of equations (16) and (17) yield

$$\boldsymbol{e}_n^T H \boldsymbol{e}_n + \boldsymbol{e}_d^T H \boldsymbol{e}_d = \lambda_1 + \lambda_2, \quad \boldsymbol{e}_n^T H \boldsymbol{e}_n - \boldsymbol{e}_d^T H \boldsymbol{e}_d = \cos 2\beta(\lambda_1 - \lambda_2). \tag{18}$$

By eliminating β from equations (17) and (18), we obtain

$$(\boldsymbol{e}_d^T H \boldsymbol{e}_d - \boldsymbol{e}_n^T H \boldsymbol{e}_n)^2 + (2\boldsymbol{e}_n^T H \boldsymbol{e}_d)^2 = (\lambda_1 - \lambda_2)^2. \tag{19}$$

Using k, p, and q, equations (18) and (19) can be restated as

$$(k+p)^2 |\nabla f|^2 = (\lambda_1 + \lambda_2)^2, \quad [(k-p)^2 + (2q)^2] |\nabla f|^2 = (\lambda_1 - \lambda_2)^2. \tag{20}$$

These are understood as invariants with respect to rotation.

Thus, if $\lambda_1 = \lambda_2$ holds, $k = p$ and $q = 0$ are followed and portion δD is an exact arc. Moreover, using ω_p and ω_q defined in equation (11), equation (20) is rewritten as

$$\sqrt{\omega_p^2 + \omega_q^2} = \delta l \,|\, \lambda_1 - \lambda_2 \,|\, /(4\,|\,\nabla f\,|) \, . \tag{21}$$

The left hand-side is used in determining the step size δl, the concept of which is closeness to arc.

3.4 Adjustment of the step size

We adjust the step size δl considering the following three criteria: 1. closeness to arc, 2. maximum shortcut angle, and 3. minimum and maximum limits of the step size.

For the first criterion, we adjust δl for which the closeness to arc of δD is permissible. We shorten δl if δD is dissimilar to arc whereas increase it if δD is close to arc. Denoting the left hand side of equation (21) by

$$\omega_u \equiv \sqrt{\omega_p^2 + \omega_q^2} \, ,$$

we suppose δD is sufficiently close to arc if

$$\omega_u \le \mu \, , \tag{22}$$

follows, where μ is a positive small constant. By contrast, if $\omega_u > \mu$ holds, we shorten δl for which δD becomes close to arc. In view of the right hand-side of equation (21), ω_u is approximately proportional to δl. We thus adjust δl based on

$$\delta l \Leftarrow c_1 \delta l \cdot (\mu / \omega_u) \quad , \tag{23}$$

for which equation (22) would satisfy where c_1 $(0 < c_1 < 1)$ is a conservation coefficient to avoid excess iteration. We then recalculate $\boldsymbol{\omega}$ and evaluate again if equation (22) follows. Should δl is too small compared with the required accuracy, specifically if

$$\omega_u \le c_2 \mu \quad (c_2 < c_1) \, ,$$

follows, we use the current δl in the current step for contour tracing and increase this afterward.

Next, we introduce a maximum shortcut angle for the second criterion. As considered in section 3.2, the three curvatures k, p, and q in equation (9) are calculated using the values on the three reference points, the angles of which from the directional vector $\boldsymbol{e}_d$ are $-\alpha$, 0, and α. For a case where there is a sharp curve, it would be safe to restrict the shortcut angle θ in the range

$$|\,\theta\,| \le \beta \, ,$$

where β is a positive constant which satisfies $\beta < \alpha$. Recalling equations (14) and (15), if δD is an exact arc,

$$\omega_k = k \delta l / 2 = \delta l / (2r) = \sin\theta \, ,$$

follows and we thus adjust δl for which

$$|\,\omega_k\,| \le \sin\beta \, , \tag{24}$$

holds. Recalling the proportionality of ω_k to δl, if equation (24) is not satisfied, we shorten δl based on

$$\delta l \Leftarrow c_1 \delta l \cdot (\sin\beta / |\,\omega_k\,|) \quad ,$$

where c_1 is the conservation coefficient used in equation (23). In respect to the third criteria, we restrict the step size δl in the range

$$\delta l_m \le \delta l \le \delta l_M \quad , \tag{25}$$

where δl_m and δl_M are the minimum and maximum step sizes. The former is to avoid stalling when there is a sharp curve or inflection point in front of the current point $\boldsymbol{x}$ or when $\boldsymbol{x}$ is deviated from the circuit ∂D. By contrast, the latter is to avoid overstepping when δD is exactly or approximately straight.

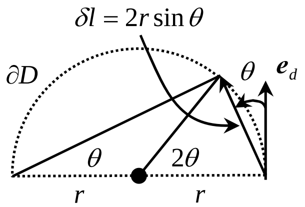

Figure 3: Relationship between the step size and shortcut angle for an arc portion

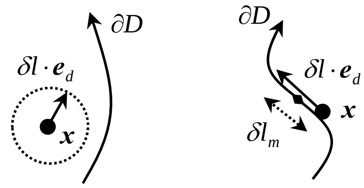

Figure 4: Two cases in which the step size and shortcut angle cannot be determined

For the current δl, if equations (22), (24), and (25) all hold, we proceed to the next step to determine the shortcut angle θ. If otherwise, we adjust δl in accordance with

$$\delta l \Leftarrow (c_1\delta l) \Big/ \max\left(\frac{\omega_u}{\mu}, \frac{|\omega_k|}{\sin\beta}, \frac{c_1\delta l}{\delta l_M}\right) \quad . \tag{26}$$

We then evaluate if equations (22) and (24) hold. The reason of using the maximum operator in the denominator is to avoid zero division, noting that $\omega_u = \omega_k = 0$ holds if δD is exactly straight. We repeat this adjustment and evaluation until equations (22) and (24) hold. Should the number of iterations reaches to a certain count, we give up adjusting δl and use a first order approximation, which is considered in section 3.6.

Furthermore, if $\delta l < \delta l_m$ holds due to equation (26), we forcibly set $\delta l \Leftarrow \delta l_m$, give up adjusting δl, and use the first order approximation.

3.5 Calculation of the shortcut angle

Once the step size δl is determined, we next determine the shortcut angle θ that satisfies equation (13). By putting the right hand-side of equation (13) to $z(\theta)$, that is,

$$z(\theta) = g(\boldsymbol{x}) - \sin\theta + \omega_k + \omega_p\cos 2\theta - \omega_q \sin 2\theta \quad ,$$

we search a root of $z(\theta) = 0$. We note that $g(\boldsymbol{x})$ is a constant and reflects the deviation of $\boldsymbol{x}$ from the circuit ∂D.

The first and second derivatives of $z(\theta)$ for θ are

$$dz(\theta)/d\theta = -\cos\theta - 2(\omega_p \sin 2\theta + \omega_q \cos 2\theta) \equiv z'(\theta) \quad ,$$

$$d^2 z(\theta)/d\theta^2 = \sin\theta - 4(\omega_p \cos 2\theta - \omega_q \sin 2\theta) \equiv z''(\theta) \quad ,$$

respectively. To find a root, we use an efficient iterative method called the Halley's method (Nocedal and Wright 2006), the update algorithm of which is as follows:

$$\theta \Leftarrow \theta + \delta\theta \ , \quad \delta\theta = -2z(\theta)\cdot z'(\theta) / \{2[z'(\theta)]^2 - z(\theta)\cdot z''(\theta)\} \quad . \tag{27}$$

We use θ in the last step as an initial guess, and repeat the update equation (27) until we find a root that satisfies

$$|z(\theta)| < \varepsilon \ ,$$

where ε is a very small positive constant.

Should $|z'(\theta)| < \varepsilon$ or $|\delta\theta| > 2\beta$ follows during the iterative update, or the number of updates reaches to a certain count, we give up obtaining θ and use the first order approximation.

3.6 First order approximation

In several cases, it becomes impossible to find a suited pair of δl and θ using the above methods, two examples of which are illustrated in Fig. 4. The left example depicts a case in which $\boldsymbol{x}$ is deviated from ∂D and the algorithm aims to shorten δl to observe equation (22). However, if we shorten δl excessively, there will not exist a corresponding θ.

The right example shows another case where there is an inflection point in front of $\boldsymbol{x}$ whilst $\boldsymbol{x}$ is on or close to ∂D. In this case, the algorithm aims to find a partitioned arc by shortening δl. On the other hand, constrained by the minimum limit of δl, i.e. δl_m, δl that satisfies equations (22), (24) and (25) cannot be found.

For cases where a suited pair of δl and θ cannot be determined, we use a method based on first order approximation. Let us recall equations (7) and (8) here. Considering the expansion of $f(\boldsymbol{x}+\delta\boldsymbol{x})$ to first order and putting $f(\boldsymbol{x}+\delta\boldsymbol{x})=0$, we obtain $f(\boldsymbol{x})\cong\delta l\,|\nabla f|\sin\theta$. We note here that θ and $f(\boldsymbol{x})$ have the same sign if we replace the approximation symbol with equality.

To restrict the maximum of $|\theta|$ to γ $(0<\gamma\le\pi/2)$, we first determine the step size δl as $\delta l =| f(\boldsymbol{x})|/(|\nabla f|\sin\gamma)$. If $\delta l<\delta l_m$ follows, which may occur when there is an inflection point just in front of $\boldsymbol{x}$ for example, we set $\delta l \Leftarrow \delta l_m$ forcibly and orient the next point by $\theta=\sin^{-1}[f(\boldsymbol{x})/(\delta l\,|\nabla f|)]$, noting again that the sign of θ and $f(\boldsymbol{x})$ is the same. On the other hand, if $\delta l>\delta l_M$ follows, which may occur when $\boldsymbol{x}$ is far deviated from ∂D for example, we set $\delta l\Leftarrow\delta l_M$ forcibly and orient the next point by $\theta=\mathrm{sgn}[f(\boldsymbol{x})]\cdot\gamma$, where $\mathrm{sgn}x=$ {1: if $x>0$, 0: if $x=0$, -1: if $x<0$ }. The parameter γ is a constant to be tuned. If γ is large, the current point $\boldsymbol{x}$ would be able to return to ∂D quickly even when $\boldsymbol{x}$ deviates from ∂D significantly but this would result in an unsmooth curve. In particular, this method corresponds to the steepest descent method if $\gamma=\pi/2$.

3.7 Line integration

A numerical integration method for calculating the area based on equation (6) is considered. We assume here that ∂D is already partitioned for which each portion δD is sufficiently close to arc. That is, the curvature k would be approximately or exactly constant on each δD.

Denoting the length of the step size δl by $2\sin\theta/k$ $(k\neq 0)$, we perform a line integration along

$$\delta D:\ \boldsymbol{x}+\delta l\cdot J(\theta)\boldsymbol{e}_d = \boldsymbol{x}+(2\sin\theta/k)\,J(\theta)\boldsymbol{e}_d\,,$$

and we regard this as new $\boldsymbol{x}$ for equation (6). Moreover, in view of Fig. 5, the unit normal vector $\boldsymbol{e}_n$ is replaced by $J(2\theta)\boldsymbol{e}_n$, and the infinitesimal arc length dl is constantly $(2d\theta)/k$ during the circulation.

We thus formulate the line integration for equation (6) by replacing $\boldsymbol{x}$, $\boldsymbol{e}_n$, and dl with $\boldsymbol{x}+\delta l\cdot J(\theta)\boldsymbol{e}_d$, $J(2\theta)\boldsymbol{e}_n$, and $(2d\theta)/k$, respectively. Letting δS be the area element, and using equations (3) and (4), we obtain

$$\begin{aligned}\delta S &= \int_{\delta D}\frac{[\boldsymbol{x}+\delta l\cdot J(\theta)\boldsymbol{e}_d]^T[J(2\theta)\boldsymbol{e}_n]}{2}dl=\int_0^\theta\frac{[\boldsymbol{x}+(2\sin\theta/k)\,J(\theta)\boldsymbol{e}_d]^T[J(2\theta)\boldsymbol{e}_n]}{2}\cdot\frac{2}{k}d\theta\\ &=\int_0^\theta\frac{\boldsymbol{x}^T J(2\theta)\boldsymbol{e}_n}{k}d\theta+\int_0^\theta\frac{2\sin^2\theta}{k^2}d\theta=\frac{\sin\theta\,\boldsymbol{x}^T J(\theta)\boldsymbol{e}_n}{k}+\frac{1}{k^2}\left(\theta-\frac{1}{2}\sin 2\theta\right)\end{aligned}.$$

By replacing back k with $2\sin\theta/\delta l$, this is rewritten as

$$\delta S=\frac{\boldsymbol{x}^T J(\theta)\boldsymbol{e}_n}{2}\delta l+\frac{\delta l^2}{4}\frac{\theta-\sin\theta\cos\theta}{\sin^2\theta}=\frac{\cos\theta\,\boldsymbol{x}^T\boldsymbol{e}_n+\sin\theta\,\boldsymbol{x}^T\boldsymbol{e}_d}{2}\delta l+\frac{\delta l^2}{4}\frac{\theta-\sin\theta\cos\theta}{\sin^2\theta}. \qquad (28)$$

In particular, if $|\theta|<\varepsilon$,

$$(\theta-\sin\theta\cos\theta)/\sin^2\theta=2\theta/3+4\theta^3/25+O(\theta^5)\,,$$

holds. If we ignore the terms higher than the second order of θ, we obtain

$$\delta S\cong\frac{(1-\theta^2/2)\,\boldsymbol{x}^T\boldsymbol{e}_n+\theta\,\boldsymbol{x}^T\boldsymbol{e}_d}{2}\delta l+\frac{\theta}{6}\delta l^2\,.$$

If we further ignore the terms of orders θ^2 or δl^2, the result is the same as the one derived in Goto and Shimakawa (2013).

The significance of equation (28) is that δS gives an exact solution for arbitrary θ $(0<|\theta|\le\pi/2)$ as long as portion δD is an exact arc. Thus, we can perform the line integration with one step even if $|\theta|$ is not very small.

Furthermore, a method for calculating the length of the circuit ∂D is considered. Recalling Fig. 3, the step size and length of the arc portion are $2r\sin\theta$ and $r(2\theta)$, respectively. Thus, once δl is determined, the length of portion δD, which we will denote this by δL, is approximated by

$$\delta L=\theta/\sin\theta\cdot\delta l\,. \qquad (29)$$

If $|\theta|<\varepsilon$, we use the approximation: $\theta/\sin\theta=1+\theta^2/6+O(\theta^4)$. We then calculate the length of ∂D using $L=\oint_{\partial D}dl\cong\sum_{\delta D}\delta L$.

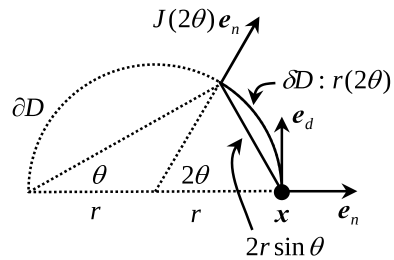

Figure 5: Line integration along an arc

Table 1: Required parameters and constants

Symbol	Role	Range	Value used
δl_m	Minimum step size	>0	Scale dependent
δl_M	Maximum step size	>0, $\delta l_m < \delta l_M$	Scale dependent
μ	Accuracy parameter representing the closeness to arc	(0, 1)	0.005–0.2
α	Direction for the reference points	(0, $\pi/2$]	$\pi/3$
β	Maximum shortcut angle	(0, $\pi/2$], $\beta<\alpha$	$\pi/4$
γ	Recovery angle for the first order approximation	(0, $\pi/2$]	$\pi/4$
c_1	Conservation coefficient	(0, 1]	0.95
c_2	Magnification threshold	(0, 1]	0.8

3.8 Circulation and termination

After the addition of the area and length elements δS and δL in equations (28) and (29), we shift the current control point $\boldsymbol{x}$ based on

$$\boldsymbol{x} \Leftarrow \boldsymbol{x} + \delta l \cdot J(\theta)\boldsymbol{e}_d \ .$$

By repeating this shift along the circuit ∂D, the control point $\boldsymbol{x}$ will be back to the initial point.

For a termination condition, we use

$$|\delta\boldsymbol{x}| \le \delta l \ , \ |\delta\boldsymbol{x}| \le \delta x_{\mathrm{pre}} \ , \tag{30}$$

where $\boldsymbol{x}_0$, $\delta\boldsymbol{x} \equiv \boldsymbol{x}_0 - \boldsymbol{x}$, and δx_{pre} represent the initial control point, residue vector, and the length of the residue vector in the previous step, respectively. The first condition of equation (30) indicates that the current point is at the vicinity of the initial point, and the second one is the current point is closer to the initial point than the one in the previous step. Once equation (30) holds, we forcibly set $\cos\theta \Leftarrow \delta\boldsymbol{x}^T\boldsymbol{e}_d / |\delta\boldsymbol{x}|$, $\sin\theta \Leftarrow -\delta\boldsymbol{x}^T\boldsymbol{e}_n / |\delta\boldsymbol{x}|$, $\delta l \Leftarrow |\delta\boldsymbol{x}|$, for which the next control point is forwarded to $\boldsymbol{x}_0$. As a summary, we list in Table 1 the relevant parameters and constants to be set, where the last five are usually fixed.

4. Numerical Experiment

An experiment using actual DEM data is performed. We choose an isolated island called Yakushima Island located in the southern part of Japan, which is registered as a UNESCO World Natural Heritage. Fig. 6 shows a geographical overview of Yakushima Island. Close to the center of the island, there is a highest peak of the island, Mt. Miyanoura, 1936(m) in altitude, the approximate location of which is 30°20’9”N, 130°30’17”E. We use gird-based DEM data, the finest mesh size of which is approximately 60.1 x 46.2 (m). We set the elevation of the sea level to -1m, the main objective of which is to handle the coastline in the same manner as other elevations. The bicubic spline function equation (1) is used for the elevation function. We now carry out computations of the lengths and areas of circuits for varying ten elevations z =0, 200, ..., 1800(m) with 200(m) interval. We use parameters $(\delta l_m, \delta l_M, \mu)$ = (0.1, 10, 0.1) and the remaining five parameters are the same as shown in Table 1. The starting guess for each elevation is determined by finding a point with the minimum difference between the target elevation and actual altitude. If there are multiple points with the same minimum value, we choose a one with the medium index. This means that only one circuit can be extracted even if there are multiple circuits inside the region.

Fig. 7 presents the results of the contour tracing for elevations from z=0(m) to 1800(m). The control points are connected with broken lines, and the outermost circuit represents z=0(m), i.e. the coastline. We can confirm that the shapes of the most circuits are fairly complex. Table 2 shows the computation results for the area and length of the circuits for the ten elevations. The area decreases as the elevation z increases. On the other hand, the length increases from z=0(m) to 400(m), and remains within 180–200(km) from z=400(m) to 800(m). This would imply that the shape of the circuit would become more complex as the elevation increases. For z=1800, the algorithm detected a small circuit close to the center of the island. Focusing on this circuit extensively, we compare the proposed algorithm and the first order approximation by Goto and Shimakawa (2013). Fig. 8 compares the set of control points by the two algorithms for z=1800(m). For the first order approximation, we used parameters $(\delta l_m, \delta l_M, \theta_M)$=(0.1, 10, 0.2), where θ_M is referred to as the maximum shortcut angle. The area and length of the circuit are 0.034634(km^2) and 0.91169(m), respectively, and the absolute differences from the proposed method are approximately 0.148(%) and 0.069(%). In respect to the stride of the control points, the number of points for circulation is 122, which is approximately three fold compared with the proposed algorithm. In view of the control points based on the proposed algorithm, the step size is shortened in the vicinity of the sharp curves and inflection points, whereas stridden around constantly roundish and rectilinear locations.

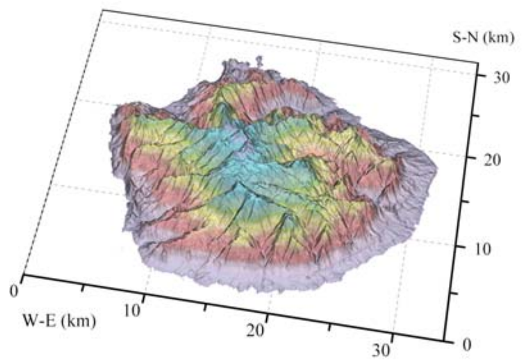

Figure 6: Overview of Yakushima Island, Japan

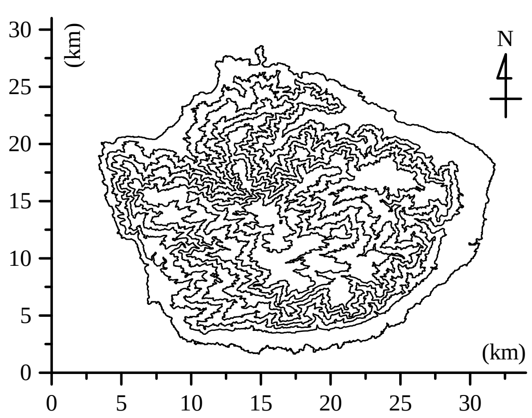

Figure 7: Extracted circuits for varying elevations from z=0(m) to 1800(m)

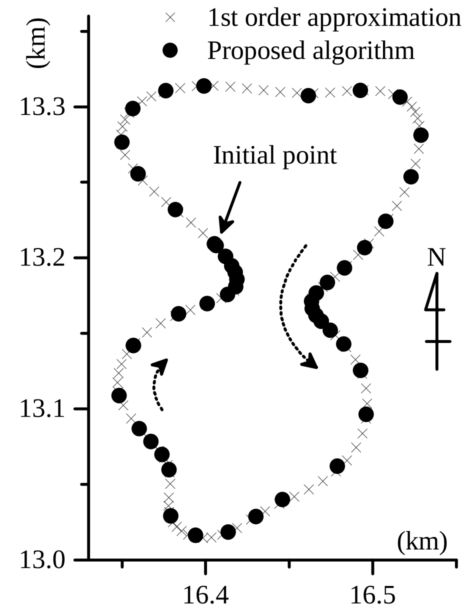

Figure 8: Comparison of the control points of the proposed algorithm and the first order approximation for z=1800(m)

Table 2: Areas, lengths, and number of control points for varying elevations

z=	0	200	400	600	800	1,000	1,200	1,400	1,600	1,800
Area (km^2)	505.754	381.452	312.699	248.086	185.086	122.127	69.780	34.838	9.016	0.034583
Length (km)	126.304	176.640	197.125	187.820	183.046	148.007	105.358	74.009	33.021	0.91232
Num. points	8,492	6,974	7,141	6,638	6,642	5,333	3,658	2,485	1,322	46

5. Conclusion

We have developed a computation algorithm for calculating the area and length of a closed planar contour, the shape of which is determined by a continuous elevation function. The framework is based on the inner skin algorithm. The elevation function is represented with three types of curvatures, which are used for evaluating the roundness of a local boundary. The circuit is partitioned into multiple portions, each of which becomes exactly or approximately arc. The computation is executed by one-dimensional line integration along the circuit. We made an experiment for an isolated island in Japan using actual DEM data. Even though the shapes of the contours are complex, the proposed algorithm worked well. In this article, we have set the accuracy parameter empirically, and the computation error is a resulted value. Improving this method to which accuracy assurance is realized remains our future work.

References

Dobkin, D. P., Levy, S. V. F., and Thurstonc, W. P., Contour tracing by piecewise linear approximations, *ACM Transactions on Graphics*, vol. 9, no. 4, pp. 389–423, 1990.

Goto, H., and Shimakawa, Y., Lightweight calculation of the area of a closed contour by an inner skin algorithm, *Journal of Computations and Modelling*, vol. 3, no. 1, pp. 57–77, 2013.

Henderson, D. W., *Differential Geometry*, Prentice-Hall, London, 1998.

Isaaks, E. H., and Srivastava. R. M., *An Introduction to Applied Geostatistics*, Oxford University Press, Oxford, 1990.

Jia, G., Wang, X., and We, H., An effective approach for selection of terrain modeling methods, *IEEE Geoscience and Remote Sensing Letters*, vol. 10, no. 4, pp. 875–879, 2013

Li, Z., Zhu, Q., and Gold, C., *Digital Terrain Modeling*, CRC Press, New York, 2004.

Lindberg, T. Scale-space theory: a basic tool for analyzing structures at different scales, *Journal of Applied Statistics*, vol. 21, no. 2, pp. 225–270, 1994.

Lioyd, C. D., *Spatial Data Analysis: An Introduction for GIS Users*, Oxford University Press, Oxford, 2010.

Longley, P. A., Goodchild, M. F., Maguire, D. J., and Rhind, D. W., *Geographic Information Systems and Science*, Wiley, New York, 2005.

McLain, D. H., Drawing contours from arbitrary data points, *The Computer Journal*, vol. 17, pp. 318–324, 1971.

Nocedal, J., and Wright. S. J., *Numerical Optimization*, Springer, New York, 2006.

Stroud, K., and Booth, D., *Vector Analysis*, Industrial Press Inc, New York, 2005.

Zuniga, O. A., and Haralick, R. M., Corner detection using the facet model, *Proceedings of the IEEE Conference on Computing Vision and Pattern Recognition*, pp. 30–37, 1983.

Biography

Hiroyuki Goto is a professor in the department of Industrial & System Engineering, Hosei University, Japan. He received his B.S. and M.S. degrees from The University of Tokyo in 1995 and 1997. He received his D.E. degree from Tokyo Metropolitan Institute of Technology in 2004. His research interests include operations research and high-performance computing.

Yoichi Shimakawa is a Professor and Director of the Department of Computer Science and Technology, Salesian Polytechnic Japan. He received his B.S. and M.Sc. degrees from Chuo University in 1990 and 1996. In 1998, he joined the staff as a research assistant on the research project “Integrated Geographic Information Systems” at Chuo University. He received his D.E. degree from Chuo University. He received a paper award from the Operations Research Society of Japan in 2002.