The Brush-Trajectory Approach to Figure Specification: Some Algebraic-Solutions

PIJUSH K. GHOSH and S. P. MUDUR
NCSDCT, Tata Institute of Fundamental Research, Bombay

The brush-trajectory method, a very natural scheme for describing two-dimensional shapes used in graphic arts and typesetting applications, has been used in only a few systems largely owing to the computational complexity involved in transforming such descriptions into raster bit maps. This paper addresses the problem. For some specific brushes and trajectories we derive algebraic solutions for describing the resulting outlines. The result of dynamic transformations on the brush as it moves along the trajectory is also studied. A special closed, smooth, convex brush defined by a fourth-order parametric equation is introduced to describe more complex shapes. An algorithmic solution to determining the outlines for an unconstrained brush is then presented. Finally, we present some ideas on a canonical brush and its use in solving the inverse problem, that is, determining the brush–trajectory description from given outlines.

Categories and Subject Descriptors: I.3.3 [**Computer Graphics**]: Picture/Image Generation—*display algorithms*; I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling—*curve, surface, solid, and object representations*

General Terms: Algorithms

Additional Key Words and Phrases: Bit-map coding, contour coding, brush–trajectory coding, supporting line, deconvolution.

1. INTRODUCTION

Currently, much high-quality computer printing is being produced on digital typesetters, laser printers, and other raster devices. All these devices consider a page of printed text as a two-dimensional matrix of tiny dots or pixels, each to be painted with the appropriate color. For monochrome output this is either black or white. Each page is composed of various two-dimensional shapes, usually letter shapes and other figures. There are a number of methods by which these basic shapes may be encoded within a computer. We shall discuss some commonly used methods below.

1.1 Bit-Map Coding

The simplest and most direct method is to encode the shape as a two-dimensional array of bits, each specifying the color of the corresponding pixel. In the context

Authors' address: NCSDCT, Tata Institute of Fundamental Research, Homi Bhabha Road, Bombay 400 005, India

of creating, maintaining, producing, and manipulating a shape, any coding scheme can itself be characterized by a number of criteria, such as production speed and quality, ease of creation, variational flexibility, and storage space requirements.

Bit-map coding allows complete freedom in the placement and intermixing of text and figures and results in high production speeds, but it is rather poor with respect to other criteria. The storage space required grows at the rate of the matrix size squared. For high-quality, high-resolution output, the information is enormous. For example, even with a typical low-resolution laser printer, a page may contain over 5 million pixels. The handling of such raster images makes heavy demands on computer memory and time. Creation and manipulation of such huge bitmaps are also problem ridden. A major drawback is that there is virtually no variational flexibility. Typographic variations, such as scaling, italicizing, lighter and bolder weights, etc., are very difficult to achieve.

1.2 Contour Coding

It can be easily verified that most of the shapes used in composing a page consist of collections of solid black regions, each covering a large number of pixels. This has been recognized in the past by computer graphics specialists. The choice of describing only the boundary, which usually is quite simple, and not the interior has been made in many graphics packages. Therefore, a shape is described by its idealized, directed contour, with the convention that its interior is filled black. The contour may be described as one or more closed loops of edges, each edge lying on a straight line, conic section, parametric cubic, or more complex curve. There are many algorithms for *raster filling* contour descriptions. Many CRT displays perform raster filling in hardware/firmware, before actually displaying the image on the screen. The contour coding method is certainly far more compact than the bit-map coding method, but it is not without its drawbacks when it comes to being used for graphic arts/typesetting.

Contour coding can hardly be said to provide flexibility in shape variation. Scaling is well implemented, and this is the only variation that can be reasonably expected from this method. For instance, italics are not just slanted versions of the upright Roman letters. It should be possible to get bolder or lighter faces from one weight [1].

1.3 Brush–Trajectory Coding

Many of the basic shapes occurring in printed texts and other graphic artwork can in fact be most naturally described as shapes generated by a brush moving along a given trajectory. This is obvious from technohistorical reasons as well. Figure 1 shows some typical examples of shapes that commonly occur and how such shapes can be easily described by some brush–trajectory combination.

The brush-trajectory method is compact and permits shape variation with greater ease. It also brings out the underlying structure in a shape. Yet the brush–trajectory method is not without its problems. For one thing it is much harder to convert into a bit map than is a contour description and hence takes much longer processing time. Also, there are some shapes that are not easily described by this method, and they are best described by the contour coding method.

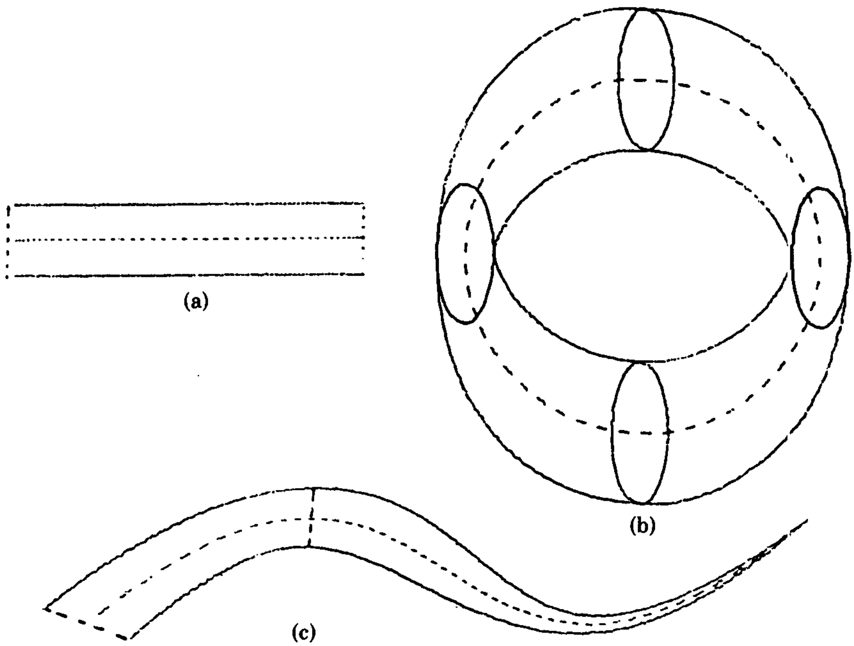


Fig. 1. (a) A horizontal or vertical bar of constant thickness. (b) Circular or elliptical shape of constant or varying thickness. (c) Tail-like shape (quite common in Indian scripts).

Experience has shown that this brush–trajectory method is quite convenient to use for shape description in typesetting and graphic arts applications. Knuth's METAFONT system [5, 6] for alphabet design is based largely on this method.

Unfortunately, the methods known for converting such specifications into bit maps are computationally quite expensive. Basically, the bit-map-encoded brush is positioned pixel by pixel along the trajectory, and for each position the pixels covered by the brush are set to black. Such an algorithm usually ends up setting the same bit a large number of times, depending on the width of the brush. Processing time increases considerably with increase in resolution. Also jaggies are difficult to avoid. Performance improvement is possible by selectively coloring only those pixels not previously painted. This, however, would require each pixel to be looked at individually. For very high resolution, there may not be adequate performance improvement.

This discussion suggests that a better approach may be to accept the shape description in the brush–trajectory form, but then convert this description into a contour-encoded form so that raster filling can be done efficiently. Some work toward achieving this goal has been done by Guibas and Stolfi [3]. Their work, however, places more emphasis on deriving a small set of mathematical tools so that many of these problems and their solutions can be specified within a computational geometric framework. Our interest today is somewhat more pragmatic. Specifically, we would like answers to questions such as:

(1) How do we analytically specify a brush and are there any restrictions that the brush shape should satisfy?

(2) Traditionally, for variations in shape, artists have varied brush angle and pressure. How can these extra degrees of freedom be computationally simulated?
(3) Given mathematical specifications for the brush and trajectory, can the contour encoding of the resulting shape be derived analytically? How much more complex does it get if one allows extra degrees of freedom such as brush angle and pressure?
(4) Is there a single canonical brush which can be used to generate any shape with some trajectory? Is dynamic transformation of the brush essential?

In the rest of this paper we shall try to provide answers to many of the above questions.

2. CONVOLUTION OF FIGURES

Let **B** and **T** be, respectively, the equations of the outlines of the brush and the trajectory. If the parametric form of the representation is chosen, then

$$\mathbf{B} = \mathbf{B}(u) = [B_x(u), B_y(u)] \tag{1}$$

and

$$\mathbf{T} = \mathbf{T}(t) = [T_x(t), T_y(t)], \tag{2}$$

where u and t are two scalar variables.

For simplicity, let us initially restrict our discussion to the cases in which **B** is a closed, smooth, convex curve and **T** a continuous smooth curve. These conditions ensure that in any given direction, **B** has two and only two parallel tangent lines. We also assume that the brush moves parallel to itself along the trajectory, which means that there is only translation of the brush along the trajectory, but no transformation of shape (rotation or scaling) as the brush moves. This brush we denote as simple brush.

Note: The above restrictions are not really essential, and we shall present a general solution toward the end of this paper. Below we present two useful theorems first stated by Guibas and Stolfi [3].

THEOREM 1 (OUTLINE THEOREM). *Interior points of the brush never lead to outline points. Therefore, OUTLINE-F (trajectory, outline of brush).*

PROOF: Let $b \in \mathbf{B}$, $t \in \mathbf{T}$, and $p = b + t \in \mathbf{B} + \mathbf{T}$. If b is an interior point of **B**, there must exist an open sphere $C_b \subseteq \mathbf{B}$ centered around b. $C_b + t \subseteq \mathbf{B} + \mathbf{T}$ is then an open sphere around the point $b + t$. Therefore, p is interior to $\mathbf{B} + \mathbf{T}$ if b is an interior point. □

This theorem is quite obvious, but it helps us to realize that, in determining the contour, we need to consider only the outline of the brush, and not its interior. Hereafter, **B** denotes the outline of the brush.

THEOREM 2 (*Slope Theorem*). Let **B** be the representation of the outline of a brush and **T** the representation of a trajectory. The boundary of the generated shape when the brush is moved along the trajectory can be expressed as a subset of the set of all points $b + t$ such that $b \in \mathbf{B}$, $t \in \mathbf{T}$, and the tangents to **B** and

T at these points are parallel. This operation is sometimes expressed as **B**∗**T** and is called the convolution of the curves **B** and **T** [3].

Our present brush, constrained to be closed, smooth, and convex, is a special case of the more general brush introduced in the Modified Slope Theorem later in Section 5.1 of this paper. We shall discuss the applicability of the proof of the Modified Slope Theorem to this special case in Section 5.1.

For the present, our interest is in obtaining analytical representations of the outline curves so that they can be computed efficiently. For this let us note the following: Both **B** and **T** are continuous and smooth. **B** is also closed. The different instances of **B**, as it is moved along the trajectory **T**, can be thought of as a family of curves. The outline generated by a family of curves is known as the *envelope* of the family, which can be defined analytically from well-known principles of differential geometry [2].

If the equation of a family of curves is represented by $F(x, y, t) = 0$, where t is the parameter of the family, the envelope is found by solving

$$F(x, y, t) = 0 \tag{3a}$$

and

$$\frac{\partial F}{\partial t}(x, y, t) = 0 \tag{3b}$$

simultaneously.

If the curves of the family $F(x, y, t) = 0$ are described parametrically by

$$x = x(u, t) \qquad \text{and} \qquad y = y(u, t), \tag{3c}$$

where u is the parameter describing the points on any given curve and t is the family parameter distinguishing the different curves of the family, then we may obtain the parametric equation of the envelope by eliminating either u or t from eq. (3c) with the aid of

$$\frac{\partial x}{\partial u}\frac{\partial y}{\partial t} - \frac{\partial x}{\partial t}\frac{\partial y}{\partial u} = 0. \tag{3d}$$

This equation is equivalent to the eq. (3b) in parametric form.

In our case,

$$x = B_x(u) + T_x(t) \qquad \text{and} \qquad y = B_y(u) + T_y(t). \tag{3e}$$

Therefore eq. (3d) transforms as follows:

$$\frac{\partial B_x}{\partial u}\frac{\partial T_y}{\partial t} - \frac{\partial T_x}{\partial t}\frac{\partial B_y}{\partial u} = 0. \tag{3f}$$

That is,

$$\left[\frac{\partial B_y(u)/du}{\partial B_x(u)/du}\right] = \left[\frac{\partial T_y(t)/dt}{\partial T_x(t)/dt}\right]_{t=t_1}. \tag{4}$$

This is essentially Theorem 2 stated in a more analytical form. We must note that this too acts as an analytical proof for Theorem 2.

In general, $u_1, u_2, u_3, \ldots, u_n$ are the solutions to eq. (4):

$$u_1 = f_1(t_1),$$
$$u_2 = f_2(t_1),$$
$$\vdots$$
$$u_n = f_n(t_1).$$

For our simple brush we obtain two solutions, that is, u_1 and u_2 corresponding to two points on **B**. These two points we label as antipodal points of **B** for some particular tangent direction. Therefore, the outline curves $\mathbf{O}_1$ and $\mathbf{O}_2$ can be expressed as

$$\mathbf{O}_1 = [O_{1x}, O_{1y}], \qquad \begin{aligned} O_{1x} &= B_x(u_1) + T_x(t_1) = h_{1x}(t_1), \\ O_{1y} &= B_y(u_1) + T_y(t_1) = h_{1y}(t_1), \end{aligned} \tag{5a}$$

$$\mathbf{O}_2 = [O_{2x}, O_{2y}], \qquad \begin{aligned} O_{2x} &= B_x(u_2) + T_x(t_1) = h_{2x}(t_1), \\ O_{2y} &= B_y(u_2) + T_y(t_1) = h_{2y}(t_1). \end{aligned} \tag{5b}$$

Note:

(i) When the trajectory is not closed, the actual contour would include at each end, some segment of the brush boundary. These segments can be easily determined and added to the contour representation.

(ii) The outlines generated using the slope theorem form the actual contour only under conditions discussed below:

Let $Cs(t)$ denote the points on the outline as determined using the slope theorem at trajectory parameter value t, where t varies from 0 to 1. $\mathrm{Br}(t)$ denotes the region covered by the brush at trajectory parameter value t.

Outlines generated using the slope theorem form the actual contour iff for all t_1, t_2, such that, $0 \le t_1 \ne t_2 \le 1$, $Cs(t_1) \notin \mathrm{Br}(t_2)$.

Violation of this condition implies that the brush-trajectory specification is such that the resulting outlines are overlapping or self-intersecting. Then a postprocessing phase which generates as its output one or more closed nonoverlapping and nonintersecting contours is needed. For this, one can make use of the winding number concept [3] or the graph manipulation approach common in geometric modeling [8].

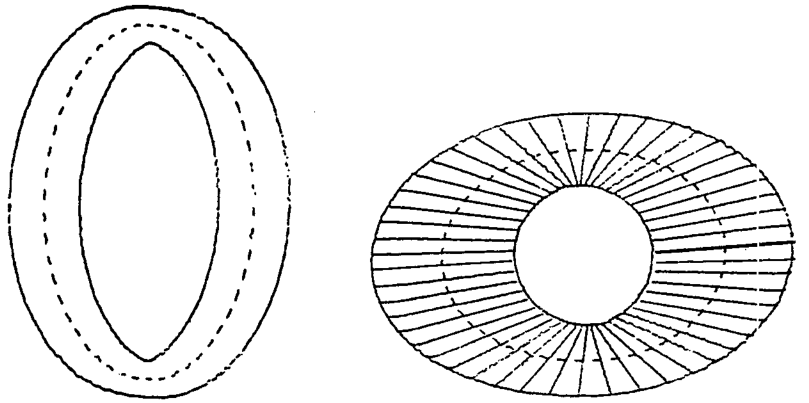
(iii) Because of the inherent difficulty involved in dealing with situations in which the brush–trajectory specifications do not satisfy the above condition, one approach that may pay off is to raster fill the brush–trajectory specification directly, without the intermediate stage of contour detection. In this paper, however, we have not investigated this approach and have only discussed detection of outlines.

3. CONTOUR EQUATIONS FOR A FEW SAMPLE CASES

3.1 Simple Elliptical Brush and Elliptical Trajectory

Let **B** be an elliptical brush, with

$$B_x = a\cos u, \qquad B_y = b\sin u, \tag{6}$$



(a) (b)

Fig. 2. (a) A simple elliptical brush and elliptical trajectory. (b) Another example of a simple elliptical brush and elliptical trajectory.

where a and b are the major and minor axes of the ellipse, respectively, and the parameter u varies from 0 to 2π radians. The trajectory equation $\mathbf{T}$, also an ellipse, can be expressed as

$$T_x = p\cos t, \qquad T_y = q\sin t,$$

where p and q are the major and minor axes and t varies from 0 to 2π radians.

Using (4) we get

$$\tan u = \frac{b\cdot p}{a\cdot q}\cdot \tan t_1 = k\cdot \tan t_1,$$

where $k = b\cdot p/a\cdot q$. Therefore,

$$u_1 = \tan^{-1}(k\cdot \tan t_1) \qquad \text{and} \qquad u_2 = \pi + \tan^{-1}(k\cdot \tan t_1). \tag{7}$$

Using (5), we get the equations for the outlines:

$$\left.\begin{aligned} O_{1x} &= a\cdot\cos u_1 + p\cdot\cos t_1 = \frac{a\cdot\cos t_1}{\sqrt{k^2\cdot\sin^2 t_1 + \cos^2 t_1}} + p\cdot\cos t_1, \\ O_{1y} &= b\cdot\sin u_1 + q\cdot\sin t_1 = \frac{b\cdot k\cdot\sin t_1}{\sqrt{k^2\cdot\sin^2 t_1 + \cos^2 t_1}} + q\cdot\sin t_1, \end{aligned}\right\} \tag{8a}$$

$$\left.\begin{aligned} O_{2x} &= -a\cdot\cos u_1 + p\cdot\cos t_1, \\ O_{2y} &= -b\cdot\sin u_1 + q\cdot\sin t_1, \end{aligned}\right\} \tag{8b}$$

where t_1 varies from 0 to 2π radians. In Figure 2 we show the outlines for typical values of a, b, p, and q.

There are a few interesting special cases.

Case 1: $a = b = r$; $p = q = R$; $r < R$. Equation (8) then becomes

$$\begin{aligned} \mathbf{O}_1 &= [(R + r)\cos t_1,\ (R + r)\sin t_1], \\ \mathbf{O}_2 &= [(R - r)\cos t_1,\ (R - r)\sin t_1]. \end{aligned} \tag{9}$$

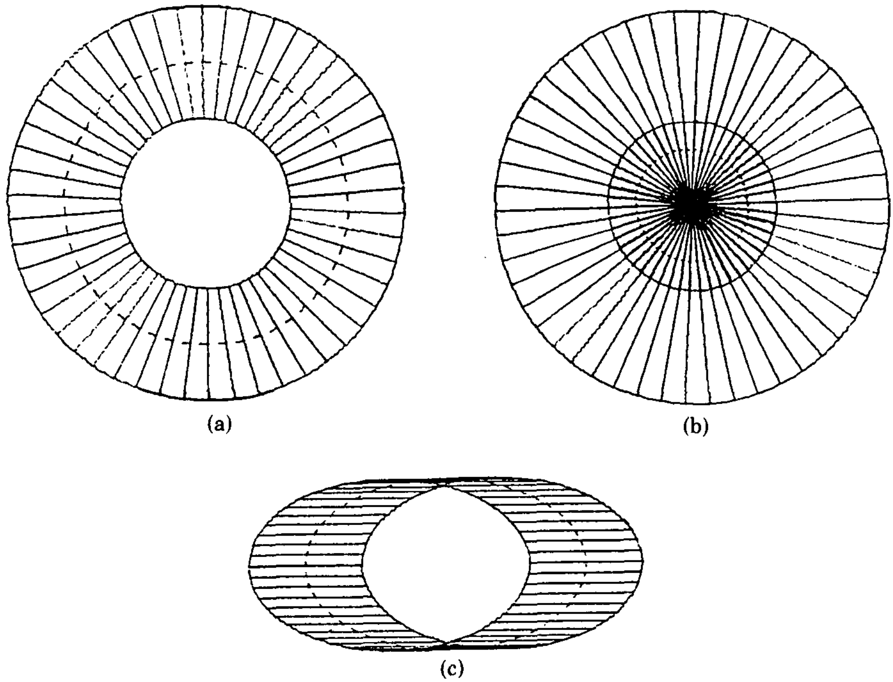


Fig. 3. (a) Circular annulus ($r < R$) (shaded portion will be black). (b) Full circular disk ($r > R$). (The inner circle overlaps with the outer one.) (c) Shape generated when minor axis of elliptical brush is made 0.

This clearly shows that if a circular brush of radius r moves along a circular trajectory of radius R, where $r < R$, the resulting shape is a circular annulus with inner outline of radius $(R - r)$ and outer outline of radius $(R + r)$ (Figure 3a).

Case 2: $a = b = r$; $p = q = R$; $r > R$. Equation (8) now becomes

$$\begin{aligned} \mathbf{O}_1 &= [(r + R)\cos t_1,\ (r + R)\sin t_1], \\ \mathbf{O}_2 &= [-(r - R)\cos t_1,\ -(r - R)\sin t_1]. \end{aligned} \tag{10}$$

Equation (10) clearly shows that the inner boundary is a circle of negative radius. In case $r = R$, $\mathbf{O}_2$ reduces to a point, and the resulting shape changes from a circular annulus to a full disk of radius $(R + r)$ with no hole inside. A negative radius means overlapping of outlines. The result is the same full disk (Figure 3b) after the postprocessing operation mentioned earlier.

Case 3: $b = 0$. This means that the brush is a horizontal line of length 2a. The outline equations are

$$\begin{aligned} \mathbf{O}_1 &= [(a + p \cdot \cos t_1),\ q \cdot \sin t_1], \\ \mathbf{O}_2 &= [(-a + p \cdot \cos t_1),\ q \cdot \sin t_1]. \end{aligned} \tag{11}$$

The outlines represented above are two ellipses whose centers are respectively, at $(a, 0)$ and $(-a, 0)$ (Figure 3c).

If the brush becomes a vertical line of length $2b$, the outlines are again two ellipses, but the centers are at $(0, b)$ and $(0, -b)$, respectively.

Case 3 shows that in certain cases it is possible to get analytical solutions for an open curve brush from the general solution of some closed, smooth, convex brush.

3.2 Elliptical Brush and Cubic Trajectory

The brush **B** is represented by eq. (6). The trajectory **T** is a general parametric cubic curve. Let **T** be expressed as

$$\begin{aligned} T_x &= p_{0x} + p_{1x} \cdot t + p_{2x} \cdot t^2 + p_{3x} \cdot t^3, \\ T_y &= p_{0y} + p_{1y} \cdot t + p_{2y} \cdot t^2 + p_{3y} \cdot t^3, \end{aligned} \tag{12}$$

where t varies from 0 to 1.

Using (4) we get,

$$\tan u = -\frac{1}{m/k_1} = -\frac{1}{m_1},$$

where

$$\begin{aligned} m_1 &= \frac{m}{k_1}, \qquad k_1 = \frac{b}{a}, \\ m &= \dot{\mathbf{T}}\ (t = t_1) = \frac{p_{1y} + 2 \cdot p_{2y} \cdot t_1 + 3 \cdot p_{3y} \cdot t_1^2}{p_{1x} + 2 \cdot p_{2x} \cdot t_1 + 3 \cdot p_{3x} \cdot t_1^2}. \end{aligned} \tag{13}$$

Therefore,

$$u_1 = \tan^{-1}\left(-\frac{1}{m_1}\right), \qquad u_2 = \pi + \tan^{-1}\left(-\frac{1}{m_1}\right).$$

Therefore, the equations of the outlines are

$$\begin{aligned} O_{1x} &= -\frac{am_1}{\sqrt{1 + m_1^2}} + (p_{0x} + p_{1x} \cdot t_1 + p_{2x} \cdot t_1^2 + p_{3x} \cdot t_1^3), \\ O_{1y} &= \frac{b}{\sqrt{1 + m_1^2}} + (p_{0y} + p_{1y} \cdot t_1 + p_{2y} \cdot t_1^2 + p_{3y} \cdot t_1^3); \end{aligned} \tag{14a}$$

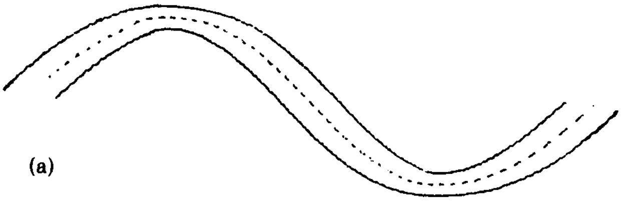
$$\begin{aligned} O_{2x} &= \frac{am_1}{\sqrt{1 + m_1^2}} + (p_{0x} + p_{1x} \cdot t_1 + p_{2x} \cdot t_1^2 + p_{3x} \cdot t_1^3), \\ O_{2y} &= -\frac{b}{\sqrt{1 + m_1^2}} + (p_{0y} + p_{1y} \cdot t_1 + p_{2y} \cdot t_1^2 + p_{3y} \cdot t_1^3). \end{aligned} \tag{14b}$$

A few typical examples are shown in Figure 4.

3.3 Dynamic Transformations of The Brush

So far we have assumed that the brush size and orientation remain fixed as the brush is moved along the trajectory. However, artists have achieved interesting

(a)



(b)

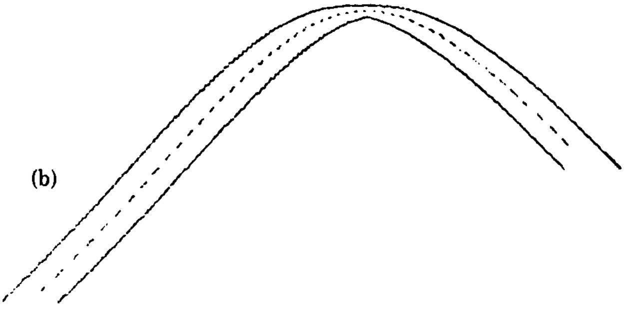


Fig. 4. (a) A simple elliptical brush and parametric cubic trajectory. (b) Another example of a simple elliptical brush and parametric cubic trajectory.

effects by continuously changing the pressure and the angle of the brush as it is moved along the path. We now provide two extra degrees of freedom to the brush; continuous scaling and rotation about a reference point as it is moved along the trajectory. These are specified as functions of the same parameter as that used for the trajectory function specification.

Thus the brush is now a function of two variable parameters, u and t, that is, $\mathbf{B}(u, t)$. If we now use this form of the brush, eq. (3e) takes the form

$$x = B_x(u, t) + T_x(t), \qquad y = B_y(u, t) + T_y(t), \tag{3g}$$

and the equation required for getting the envelope takes the form

$$\frac{\partial B_x}{\partial u}\left(\frac{\partial B_y}{\partial t} + \frac{\partial T_y}{\partial t}\right) - \frac{\partial B_y}{\partial u}\left(\frac{\partial B_x}{\partial t} + \frac{\partial T_x}{\partial t}\right) = 0. \tag{3h}$$

Solving eqs. (3g) and (3h) analytically is very difficult, almost impossible even for simple cases. What we would like to try, therefore, is to see if these conditions can be simplified so that a good approximation of the actual envelope is obtained analytically. For this we make use of the classical-mechanics idea of perturbation: If

$$\frac{\partial B_x}{\partial t} \ll \frac{\partial T_x}{\partial t} \quad \text{and} \quad \frac{\partial B_y}{\partial t} \ll \frac{\partial T_y}{\partial t},$$

then we get a good approximate envelope by initially ignoring the $\partial B_x/\partial t$ and $\partial B_y/\partial t$ terms in eq. (3h) and later perturbing the envelope equation thus obtained by the exact dynamic nature of the brush. For all practical purposes, this method of perturbation works well and a very good approximation of the actual envelope is obtained. Most of the existing systems [7] use this approach to approximate the envelope for dynamic brushes.

Again, we shall consider an elliptical brush, but this time the major axis, minor axis, and the orientation of **B** will be some functions of t, that is,

$$a = f_1(t); \qquad b = f_2(t); \qquad \theta = f_3(t).$$

The equation of **B** can then be written as

$$\begin{aligned} B_x &= a \cdot \cos u \cdot \cos\theta + b \cdot \sin u \cdot \sin\theta, \\ B_y &= -a \cdot \cos u \cdot \sin\theta + b \cdot \sin u \cdot \cos\theta. \end{aligned} \tag{15}$$

The trajectory **T** is again considered as a cubic parametric curve whose equation is (12).

Following the same procedure, we get

$$\tan u = \frac{b \cdot \sin\theta \cdot m - b \cdot \cos\theta}{a \cdot \cos\theta \cdot m + a \cdot \sin\theta} = -\frac{1}{m_1}, \tag{16}$$

where m is expressed by eq. (13). Therefore, the equations of the outlines are

$$\begin{aligned} O_{1x} &= \frac{-a \cdot m_1}{\sqrt{1 + m_1^2}} \cdot \cos\theta + \frac{b}{\sqrt{1 + m_1^2}} \cdot \sin\theta + T_x\ (t = t_1), \\ O_{1y} &= \frac{-a \cdot m_1}{\sqrt{1 + m_1^2}} \cdot \sin\theta + \frac{b}{\sqrt{1 + m_1^2}} \cdot \cos\theta + T_y\ (t = t_1); \end{aligned} \tag{17a}$$

$$\begin{aligned} O_{2x} &= \frac{a \cdot m_1}{\sqrt{1 + m_1^2}} \cdot \cos\theta - \frac{b}{\sqrt{1 + m_1^2}} \cdot \sin\theta + T_x\ (t = t_1), \\ O_{2y} &= \frac{a \cdot m_1}{\sqrt{1 + m_1^2}} \cdot \sin\theta - \frac{b}{\sqrt{1 + m_1^2}} \cdot \cos\theta + T_y\ (t = t_1). \end{aligned} \tag{17b}$$

A few typical examples are shown in Figure 5.

4. DESIGNING A SPECIAL BRUSH

The brush shapes we have discussed so far are linear, circular, or elliptical. With the freedom to transform a brush dynamically, these brushes provide a powerful tool for shape description. In fact, as we show later, any pair of outlines can be generated using one of these brushes with the dynamic transformation capabilities. However, it may not be very natural from the user's point of view. Because of the basic symmetrical nature of the above brushes, there exist complex shapes (Figure 6) which they cannot easily and naturally describe. Therefore, we design a special brush that is parametrically defined using a fourth-order polynomial. Later, we impose conditions on the parameters of the brush so that the special brush approximates a line, a circle, or an ellipse, as desired.

4.1 The Special Brush Equation

Our intention is to choose a brush that satisfies the following criteria:

(1) **B** should be a closed, smooth, and convex curve.
(2) **B** should be computationally convenient and, at the same time, be flexible enough to permit variation in brush shape.
(3) **B** should be conceptually easy to comprehend.

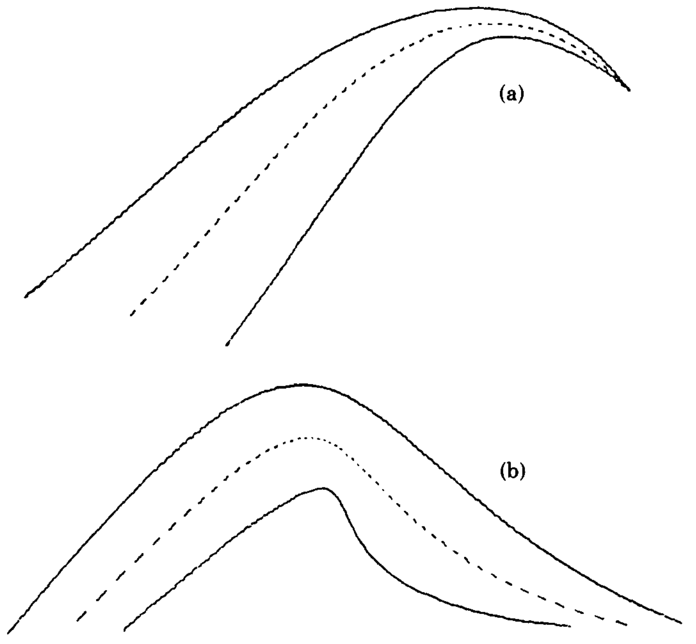


Fig. 5. (a) A dynamically transformed elliptical brush and parametric cubic trajectory. (b) Another example of a dynamically transformed elliptical brush and parametric cubic trajectory.

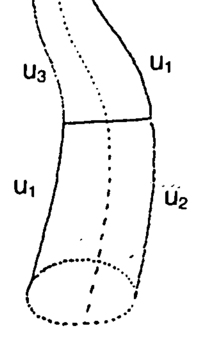


Fig. 6. A shape difficult to produce using a symmetric brushlike circle or ellipse.

Without digressing into the details of the steps involved in arriving at the present brush equation, we shall present the end results with some notes explaining how it can be used conveniently.

To make our brush equation computationally convenient, we have chosen a fourth-order polynomial. (The cubic does not provide enough design flexibility for changing the brush shape.) If we assume that the brush outline always passes through the origin, **B** can be described by the following polynomial expression:

$$\mathbf{B} = \mathbf{B}(u) = 4u(1 - u)(1 - 2u) \cdot \mathbf{B}_1 + 6u^2(1 - u)^2 \cdot \mathbf{B}_2, \tag{18}$$

where u is a scalar quantity that varies from 0 to 1, and $\mathbf{B}_1$, $\mathbf{B}_2$, and $-\mathbf{B}_1$ are the position vectors of three vertices P1, P2, and P3 of a *characteristic triangle* which determines the brush shape. It can be easily shown that the characteristic triangle is in fact a special case of the *generalized characteristic polygon* of the Bernstein–Bezier polynomial curve [2]. In order to give some idea of how the characteristic triangle determines the brush shape, we present two examples in Figure 7.

Also, it can be easily shown that **B** is closed, smooth, and convex.

4.2 Special Brush and Cubic Trajectory

We can rewrite the brush eq. (18) as

$$\begin{aligned} B_x &= 4u(1 - u)(1 - 2u) \cdot B_{1x} + 6u^2(1 - u)^2 \cdot B_{2x}, \\ \mathrm{B}_y\ \text{fl}\ &\mathrm{4u(1\ fi\ u)(1\ fi\ 2u)} \cdot \mathrm{B}_{1y}\ \text{ff}\ \mathrm{6u^2(1\ fi\ u^2)} \cdot \mathrm{B}_{2y}, \end{aligned} \tag{19}$$

and **T** is expressed by eq. (12). From eq. (4) we can write

$$\left[\frac{\partial B_y/\partial u}{\partial B_x/\partial u}\right] = \frac{24B_{2y} \cdot u^3 + (24B_{1y} - 36B_{2y}) \cdot u^2 + (12B_{2y} - 24B_{1y}) \cdot u + 4B_{1y}}{24B_{2x} \cdot u^3 + (24B_{1x} - 36B_{2y}) \cdot u^2 + (12B_{2x} - 24B_{1x}) \cdot u + 4B_{1x}} = m, \tag{20}$$

where m is given by eq. (13). Simplifying (20) we obtain

$$\alpha \cdot u^3 + \beta \cdot u^2 + \gamma \cdot u + \delta = 0, \tag{21}$$

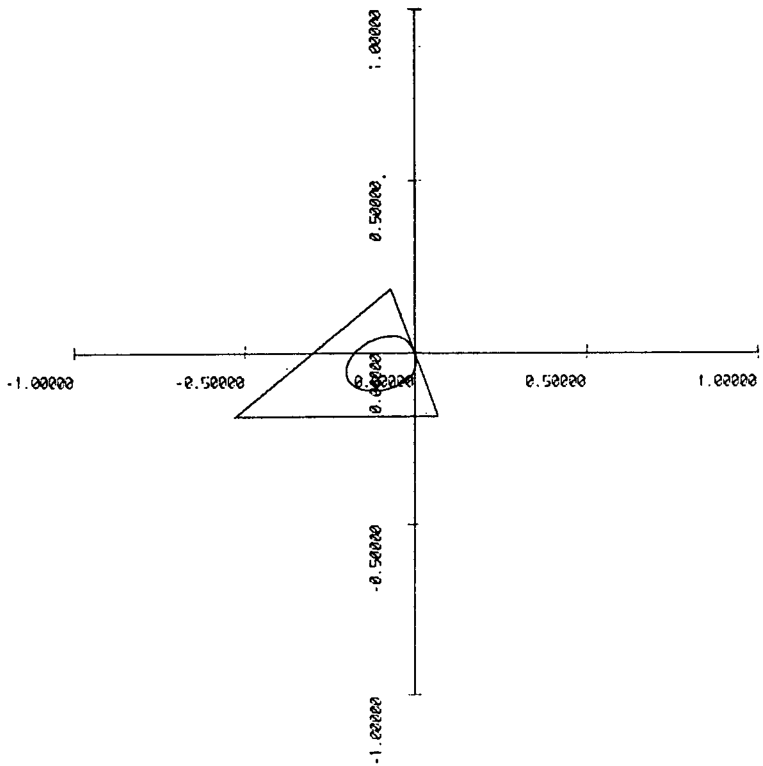
where

$$\begin{aligned} \alpha &= 24B_{2y} - 24B_{2x} \cdot m \\ \beta &= (24B_{1y} - 36B_{2y}) - (24B_{1x} - 36B_{2x}) \cdot m \\ \gamma &= (12B_{2y} - 24B_{1y}) - (12B_{2x} - 24B_{1x}) \cdot m \\ \delta &= 4B_{1y} - 4B_{1x} \cdot m. \end{aligned}$$

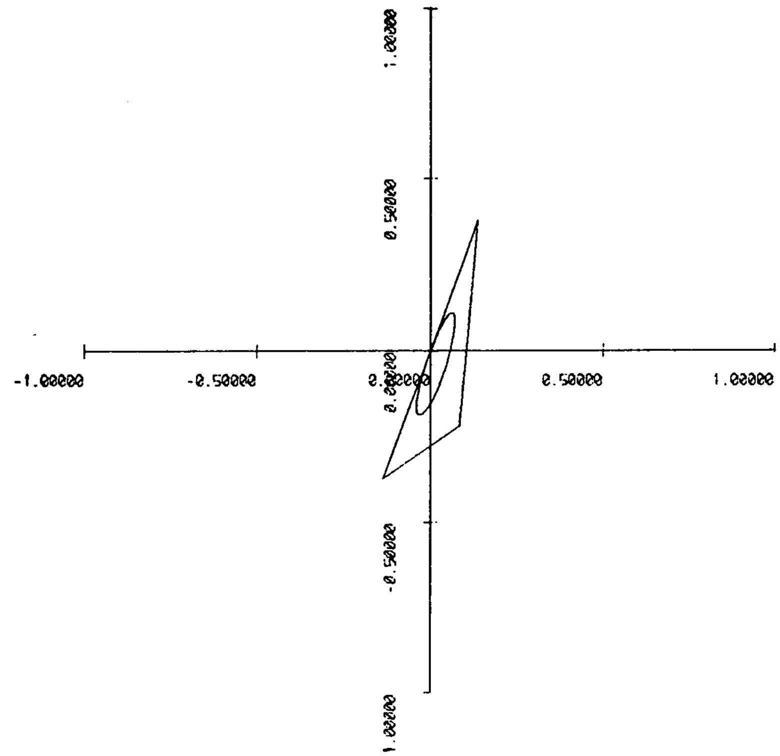
Note that the solution of eq. (21) will give rise to three roots, u_1, u_2, and u_3, of u. Any cubic equation has either one real and two imaginary roots, or three real roots. Since our brush is a closed, smooth curve, it must have at least two real roots; therefore, all u_1, u_2, and u_3 are real. But **B** is a convex curve too; therefore, it cannot have more than two parallel tangents in any given direction. This seems puzzling, but is easily explained as follows. In any given direction only two out of three roots of u will lie within the range of 0 to 1. The other has to be discarded since **B** is defined only for u varying from 0 to 1.

We can express eq. (21) in the following form:

$$v^3 + q \cdot v + p = 0, \tag{22}$$



(a)



(b)

Fig. 7. (a) The *characteristic triangle* and the special brush. (b) Another *characteristic triangle* and the corresponding brush shape

where

$$v = u + \frac{\beta}{3\alpha};$$
$$q = \frac{\gamma}{\alpha} - \frac{1}{3}\left(\frac{\beta}{\alpha}\right)^2;$$
$$p = \frac{\delta}{\alpha} - \frac{1}{3}\left(\frac{\beta}{\alpha}\right)\left(\frac{\gamma}{\alpha}\right) + \frac{2}{27}\left(\frac{\beta}{\alpha}\right)^3.$$

Since all roots of u are real, we use trigonometric methods to solve eq. (22). The roots of the cubic equation are then given by

$$n \cos \psi, \qquad n \cos\left(\frac{2\pi}{3} + \psi\right), \qquad n \cos\left(\frac{2\pi}{3} - \psi\right),$$

where

$$n = \left(\frac{-4q}{3}\right)^{1/2},$$
$$\cos \psi = -4p\left(-\frac{3}{4q}\right)^{3/2}.$$

Therefore,

$$\begin{aligned} u_1 &= n \cos \psi - \frac{\beta}{3\alpha}, \\ u_2 &= n \cos\left(\frac{2\pi}{3} + \psi\right) - \frac{\beta}{3\alpha}, \\ u_3 &= n \cos\left(\frac{2\pi}{3} - \psi\right) - \frac{\beta}{3\alpha}. \end{aligned} \tag{23}$$

The roots u_1, u_2, and u_3 are some functions of t_1. Therefore, for each point t_1 of $\mathbf{T}$, where t_1 varies from 0 to 1, we have to find out which two of the three roots lie within the range 0 to 1. Let these two roots be u' and u'' at some particular point t_1. Therefore, the equations of the outlines are

$$\begin{aligned} O_{1x} &= B_x(u = u') + T_x(t = t_1), \\ O_{1y} &= B_y(u = u') + T_y(t = t_1); \end{aligned} \tag{24a}$$

$$\begin{aligned} O_{2x} &= B_x(u = u'') + T_x(t = t_1), \\ O_{2y} &= B_y(u = u'') + T_y(t = t_1). \end{aligned} \tag{24b}$$

It is important to note at this point that even though we are able to obtain analytical solutions to u_1, u_2, and u_3 above, the need to choose two out of three makes it impossible to derive analytical forms for the outlines. Therefore, even though every point on the outlines is analytically defined, determining the locus of points has to be done through procedural means only. A second look at the labeling of the outlines generating the shape of Figure 6 illustrates this better.

4.3 Approximating a Line, Circle, or Ellipse

Forming a straight line brush is rather easy. The only condition that has to be imposed is to make the vectors $\mathbf{B}_1$ and $\mathbf{B}_2$ collinear.

The special brush cannot be made to represent an exact ellipse. It can only approximate it. For approximating an ellipse with major and minor axes as p and q, respectively, we have to impose the following conditions:

$$p = \tfrac{1}{2} \mid \mathbf{B}(\tfrac{3}{4}) - \mathbf{B}(\tfrac{1}{4}) \mid = \tfrac{3}{8} \mid \mathbf{B}_1 \mid, \tag{25a}$$

$$q = \tfrac{1}{2} \mid \mathbf{B}(\tfrac{1}{2}) - \mathbf{B}(0) \mid = \tfrac{3}{16} \mid \mathbf{B}_2 \mid, \tag{25b}$$

$$\mathbf{B}_1 \cdot \mathbf{B}_2 = 0; \tag{25c}$$

that is, the two vectors should be perpendicular.

A circle can be approximated by making p and q equal. To elaborate, once again consider the special brush equation (18):

$$\begin{array}{lll} \text{At} & u = 0, & \mathbf{B}(0) = 0. \\ \text{At} & u = \tfrac{1}{4}, & \mathbf{B}(\tfrac{1}{4}) = \tfrac{3}{8}\mathbf{B}_1 + \tfrac{27}{128}\mathbf{B}_2. \\ \text{At} & u = \tfrac{1}{2}, & \mathbf{B}(\tfrac{1}{2}) = \tfrac{3}{8}\mathbf{B}_2. \\ \text{At} & u = \tfrac{3}{4}, & \mathbf{B}(\tfrac{3}{4}) = -\tfrac{3}{8}\mathbf{B}_1 + \tfrac{27}{128}\mathbf{B}_2. \\ \text{At} & u = 1, & \mathbf{B}(1) = 0. \end{array}$$

Let $\mathbf{V}_1$ and $\mathbf{V}_2$ be the two vectors as follows,

$$\mathbf{V}_1 = \mathbf{B}(\tfrac{1}{4}) - \mathbf{B}(\tfrac{3}{4}) = \tfrac{3}{4}\mathbf{B}_1.$$

$$\mathbf{V}_2 = \mathbf{B}(\tfrac{1}{2}) - \mathbf{B}(0) = \tfrac{3}{8}\mathbf{B}_2.$$

Now let the major and minor axes, p and q, of the ellipse be

$$p = \tfrac{1}{2} \mid \mathbf{V}_1 \mid; \qquad q = \tfrac{1}{2} \mid \mathbf{V}_2 \mid.$$

It is necessary, therefore, for the vectors $\mathbf{V}_1$ and $\mathbf{V}_2$ to be perpendicular; that is, $\mathbf{B}_1$ and $\mathbf{B}_2$ should be perpendicular:

$$\mathbf{B}_1 \cdot \mathbf{B}_2 = 0.$$

The approximate coordinates of the center will be

$$(\tfrac{27}{128}B_{2x},\ \tfrac{3}{16}B_{2y}),$$

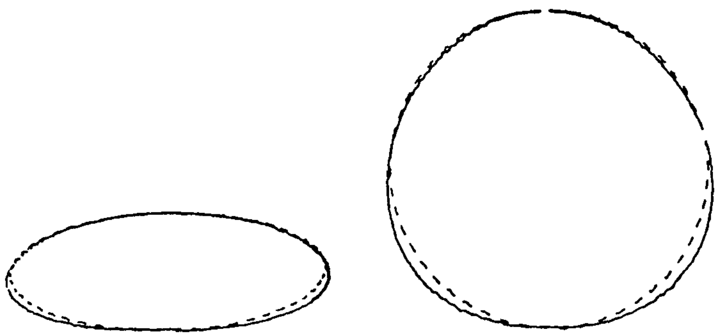
and the central axis is oriented at an angle of

$$\tan^{-1}\left(\frac{B_{1y}}{B_{1x}}\right).$$

Figure 8 compares the special brush with the exact ellipse (and circle) shown dotted.

5. THE UNCONSTRAINED BRUSH

Up to now we have chosen our brush curve to be a closed, smooth, and convex curve. This is mainly because we wished $\mathbf{B}$ to have two and only two parallel tangent lines in any given direction. As the brush is moved along the trajectory,



(a) (b)

Fig. 8. (a) The special brush approximation to an ellipse. (b) The special brush approximation to a circle.

the tangent points on the brush parallel to the direction of the trajectory are chosen to yield the outline curves. If **B** is chosen to be a more general curve, then for some particular direction(s) it is possible that **B** has no tangent line, one tangent line, or more than two tangent lines parallel to this direction. Obviously, our slope theorem fails to determine the points on the outline curves in such cases.

5.1 A Modified Slope Theorem

If **B** is any general curve (open or closed, but piecewise slope continuous), the points on the outline curves when **B** is at some point t of **T** can be conceptually determined in the following manner (Figure 9a):

For each point $b \in \mathbf{B}$ draw a line parallel to the tangent direction, L of **T** at t. Consider only those b's for which it is possible to construct a circle however small with b as center such that within this circle all other points of **B** lie entirely on one side of the line drawn at b that is parallel to L. The point $b + t$ then becomes a point on the outline curve. We call a point such as b a *supporting point* with respect to the direction L. For a closed, smooth, convex curve such a point is none other than a point at which the drawn line is tangent to **B** and is also a supporting line of the bounded, convex figure in that direction. Let us note here that a supporting line passes through at least one point of the figure and is such that the figure lies entirely on one side of it. It is well known that in each direction there can be drawn exactly two parallel supporting lines to a bounded convex figure. Since for a smooth curve these are the tangent lines, it is easy to see that this method reduces to the statement of Theorem 2.

The slope theorem can now be restated as follows:

THEOREM 2 (MODIFIED SLOPE THEOREM). *The boundary of the shape resulting from moving* **B** *along* **T** *can be expressed as a subset of the set of all points $b + t$ such that $b \in \mathbf{B}$, $t \in \mathbf{T}$, and b is a supporting point with respect to the direction of the line parallel to the tangent line of* **T** *(at t).*

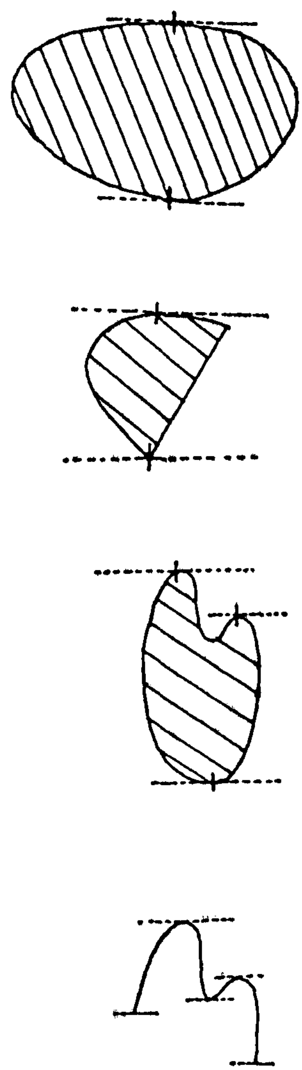
PROOF. Let $B(u_0) \in \mathbf{B}(u)$, $T(t_0) \in \mathbf{T}$, $p_0 \in \mathbf{B} + \mathbf{T}$ and $p_0 = B(u_0) + T(t_0)$. We wish to show here that if p_0 is a point on the boundary of $\mathbf{B} + \mathbf{T}$ then for some

(1)

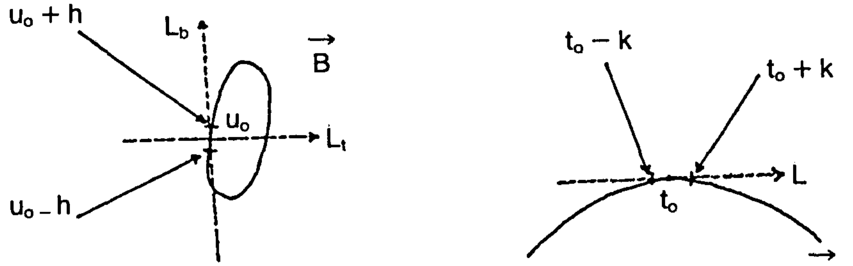
(2)

(3)

(4)



(a)



(b)

Fig. 9. (a) Examples of *supporting points* of brushes of various shapes. (1) Closed, smooth, convex region. (2) Closed, convex region (not smooth). (3) Closed, smooth, concave region. (4) Open, smooth curve. (b) Figure for modified slope theorem.

$T(t_0)$, $B(u_0)$ must be a supporting point of **B** with respect to the direction of the tangent of T at t_0, say L.

In order to prove this, let us first assume that even though p_0 is a boundary point, $B(u_0)$ is not a supporting point with respect to the direction of L. (Figure 9b) Draw two lines L_b and L_t passing through $B(u_0)$ such that L_t is parallel to L and L_b is perpendicular to L_t. Now choose h, k, however small, such that

(1) $[B(u)]_{u_0-h<u<u_0}$ is below L_t, and $[B(u)]_{u_0<u<u_0+h}$ is above L_t.

We shall use B_h to denote the part of the brush defined by $[B(u)]_{u_0-h<u<u_0+h}$.

(2) $[T(u)]_{t_0-k<t<t_0+k}$ which we shall denote as T_k, such that, the tangent on B_h for any u is not parallel to the tangent at T_k for any t. (We shall say that B_h is transverse to T_k.)

Choice (1) above is possible because **B** is position and piecewise slope continuous and $B(u_0)$ is not a supporting point. Choice (2) is possible because **T** is both position and slope continuous and L_t is not a tangent to **B** at u_0. Clearly, $B_h + T_k \subseteq \mathbf{B} + \mathbf{T}$.

Consider the following one-to-one continuous function:

$$f: D \rightarrow C,$$

where

$$D(u, t) = B_h + T_k, \qquad C(u, t) = B(u_0) + L_b(u - u_0) + L_t(t - t_0),$$

and

$$u_0 - h < u < u_0 + h, \qquad t_0 - t < t < t_0 + k.$$

Thus,

$$f(B_h + T_k) = B(u_0) + L_b(u - u_0) + L_t(t - t_0).$$

This f can be considered as composed of

$$\begin{aligned} D(u, t) = B_h + T_k &\rightarrow B_h + T(t_0) + L_t(t - t_0); \text{ because } \mathbf{T} \text{ is continuous} \\ &\rightarrow B_h + L_t(t - t_0) \\ &\rightarrow B(u_0) + L_b(u - u_0) + L_t(t - t_0) \equiv C(u, t) \\ &\qquad \text{because } B_h \text{ is continuous and transverse to } T_k. \end{aligned}$$

Clearly $C(u_0, t_0)$ is interior to $C(u, t)$. Therefore, $f^{-1}(C(u, t))$ is interior to $D(u, t)$, because inverses of continuous functions preserve interior points. Therefore, $B_h(u_0) + T_k(t_0)$ is interior to $B_h + T_k$, that is, $B(u_0) + T(t_0)$ is interior to $\mathbf{B} + \mathbf{T}$, that is, p_0 is interior to $\mathbf{B} + \mathbf{T}$. This, however, contradicts our initial premise that p_0 is a boundary point of $\mathbf{B} + \mathbf{T}$. Therefore, our assumption is incorrect and $B(u_0)$ must be a supporting point of **B** with respect to the direction of the tangent at $T(t_0)$.

5.2 The Supporting Points of a Piecewise Smooth Brush

We consider brush shapes whose boundaries can be thought of as piecewise compositions of a number of open smooth curves. By piecewise composition we

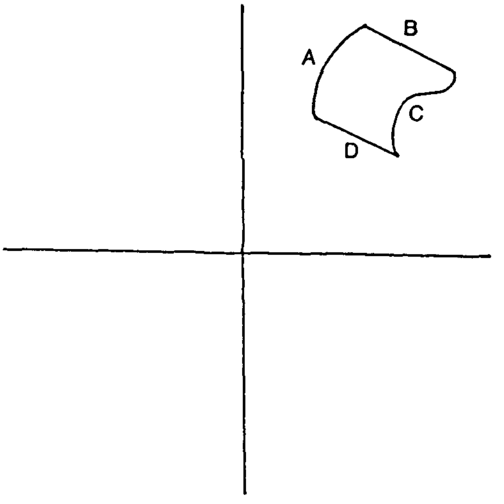


Fig. 10. Piecewise definition of a brush.

mean position continuity at the endpoints where two open curve segments meet. In general, there may be slope discontinuities at these endpoints. Figure 10 shows a shape whose boundary is defined by four open curves, A, B, C and D.

We shall take up the task of analytically determining the supporting points of B, with respect to the direction of the tangent at t of $\mathbf{T}$. Consider $\mathbf{B}_1$ as an open curve of $\mathbf{B}$. It is easy to recognize the fact that for any given direction, the supporting points are endpoints of $\mathbf{B}_1$ or points at which a line in that direction is tangential to $\mathbf{B}_1$. Hence, we adopt the following procedure:

(1) Take the two endpoints e_1 and e_2 of $\mathbf{B}_1$ as supporting points.
(2) Also determine all the points $p_1, p_2, \ldots, p_n$ on $\mathbf{B}_1$, such that the tangents at these points are all parallel to the given direction, namely, the tangent to the trajectory $\mathbf{T}$.
(3) If the brush is a closed, filled region, then from $p_1, p_2, \ldots, p_n$, include only those points at which the line in the given direction lies locally outside the brush region.

The above procedure is applied to all the curves constituting $\mathbf{B}$, taking care to see that junction vertices of $\mathbf{B}$ are considered only once.

We know that for a convex brush there can be two and only two supporting lines in any given direction. In the case in which the boundary of the brush has a linear segment in the given direction, the brush will have infinitely many supporting points in that direction, and then we choose only one point from this connected set of supporting points. Using algebraic distances of the supporting points from the tangent line to $\mathbf{T}$, we can sort the points chosen according to the earlier listed procedure and the two extremal points on either side of the tangent chosen to form the outlines. Figure 11 shows an example of a piecewise-convex brush and a cubic trajectory. Note here that this method can be used only in the case in which the brush is a closed convex region with its boundary defined in a piecewise manner.

In the case of an open or a concave brush, the loci of the supporting points are considered as the outline curves. It is important to note that, in general, we may

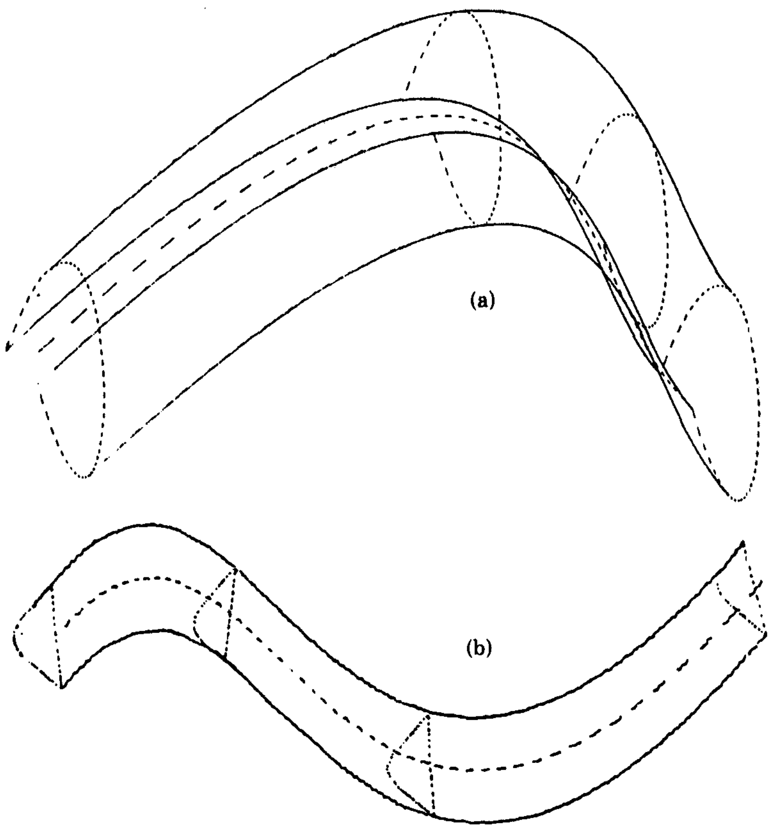


Fig. 11. (a) Open, smooth brush and parametric cubic trajectory. (b) Closed, convex brush (not smooth) and parametric cubic trajectory.

therefore have more than two outline curves; some of them possibly intersecting with one another. As noted earlier the actual boundary has to be derived from these outline curves using the winding number concept [3], or graph manipulation approach [8].

5.3 An Example: An Open Cubic Brush and a Cubic Trajectory

Let the equation of **B** be

$$\mathbf{B} = \mathbf{q}_0 + \mathbf{q}_1 \cdot u + \mathbf{q}_2 \cdot u^2 + \mathbf{q}_3 \cdot u^3 \tag{26}$$

with u varying from 0 to 1. **T** will be expressed by eq. (12). Using eq. (4) we can write

$$\dot{\mathbf{B}}(u) = \dot{\mathbf{T}}(t = t_1) = m \quad \text{or} \quad \alpha \cdot u^2 + \beta \cdot u + \gamma = 0$$

or

$$u = \frac{-\beta \pm \sqrt{\beta^2 - 4\alpha\gamma}}{2\alpha}.$$

From the above equation we can conclude that **B** can have, at most, two tangent points in the direction of m. These, together with the endpoints, give us a maximum of four supporting points with respect to any given direction. It should be remembered that only those supporting points of **B** with parameter u value within 0 and 1 are to be considered. In tracing the loci of these points, we make use of the fact that, since **T** is continuous and smooth for small increments in its parameter t, the supporting points will always lie within a small neighborhood of **B**. (Fig. 11)

6. DECONVOLUTION OF FIGURE OUTLINES

So far, we have mainly concentrated on the derivation of outlines given a brush trajectory specification. In this section, we shall briefly discuss some issues involved in the inverse problem: deriving a brush trajectory specification given its outlines. Though at first sight this may seem unnecessary, we would like to point out its importance with the following arguments:

There are many traditional fonts that have been contour coded or digitized using alphabet design systems like IKARUS [4]. If the brush trajectory specification for such fonts can be obtained, then detecting the structural similarities/differences among different fonts should become possible. Also, because of the inherently large variational capability provided by computational techniques, it should now become possible to provide variations in old designs without violating the basic design integrity as built in by the designer using manual methods of calligraphic design.

We begin by stating a few more theorems which help us in understanding the deconvolution problem. In all these cases we assume that the brush is a convex region.

6.1 A Few More Theorems

THEOREM 3 (POSITION-INVARIANCE THEOREM). *If x denotes linear translation, then*

$$(\mathbf{B}^x) * \mathbf{T} = \mathbf{B} * (\mathbf{T}^x) = (\mathbf{B} * \mathbf{T})^x. \tag{27}$$

PROOF. *Let*

$$\mathbf{B} = [B_x, B_y] \qquad \text{and} \qquad \mathbf{T} = [T_x, T_y].$$

Let us move the origin from (0, 0) to some point (α, β). Then

$$\mathbf{B}' = \mathbf{B}^x = [(B_x - \alpha), (B_y - \beta)].$$

Now $\dot{\mathbf{B}}' = \dot{\mathbf{B}}$, since (α, β) are some fixed points. Therefore, $\dot{\mathbf{B}}' = \dot{\mathbf{B}} = \dot{\mathbf{T}}(t = t_1)$.

Let u_1 and u_2 be the two antipodal points which do not change even after B moves to (α, β). Therefore,

$$\mathbf{O}_1 = [(B_x(u = u_1) + T_x - \alpha), (B_y(u = u_1) + T_y - \beta)],$$
$$\mathbf{O}_2 = [(B_x(u = u_2) + T_x - \alpha), (B_y(u = u_2) + T_y - \beta)].$$

Thus the theorem is proved.

This theorem implies that it does not matter which point of the brush is fixed with respect to the trajectory—the generated shape remains same. Only the contour moves as a whole.

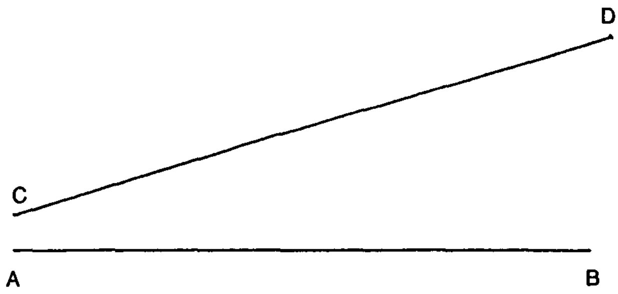


Fig. 12. Outlines which cannot be generated using any simple brush and any trajectory.

THEOREM 4 (NONUNIQUENESS THEOREM). *For any given outlines, there may exist more than one combination of brush–trajectory for generating them.*

PROOF. (See end of Theorem 6).

THEOREM 5 (SIMPLE BRUSH INSUFFICIENCY THEOREM). *If a closed, smooth, convex brush is only moved along the trajectory, without any dynamic transformation, then there may not exist any brush–trajectory combination that produces two given outlines* $\mathbf{C}_1$ *and* $\mathbf{C}_2$, *if for some point p on* $\mathbf{C}_1$, *there does not exist any point* p' *on* $\mathbf{C}_2$, where the tangents of $\mathbf{C}_1$ and $\mathbf{C}_2$ become parallel.

PROOF. We prove this by a counterexample. Consider Figure 12, which represents a pair of straight lines as outline curves. If the slopes of the lines are different, then it is not possible, without dynamic transformation to generate these as outlines using any closed, smooth, convex brush and any trajectory.

6.2 Convolution Equations for the Canonical Brush

We define the canonical brush as a straight-line brush with dynamic scaling and rotation as it is moved along the trajectory. We now try to see if this canonical brush can be used in the solution of the deconvolution problem.

Equation (17) can be used directly to get the convolution equations in this case. The ellipse transforms to a straight line if any one of its axes is made 0. Let $b = 0$. Then

$$-\frac{1}{m_1} = 0.$$

Therefore,

$$\begin{aligned} O_{1x} &= -a \cdot \cos\theta + T_x(t = t_1) = -a \cdot \cos\theta + T_x, \\ O_{1y} &= -a \cdot \sin\theta + T_y(t = t_1) = -a \cdot \sin\theta + T_y; \end{aligned} \tag{28a}$$

$$\begin{aligned} O_{2x} &= a \cdot \cos\theta + T_x, \\ O_{2y} &= a \cdot \sin\theta + T_y. \end{aligned} \tag{28b}$$

Note that a and θ can be any function of t where t is the scalar parameter of the trajectory equation $\mathbf{T}$.

THEOREM 6 (THE CANONICAL BRUSH THEOREM). *Any pair of outlines of a convex region can be described in terms of a brush–trajectory representation, if we choose the brush as a straight line with dynamic transformations allowed on it.*

PROOF. Let a pair of outlines be given by the following equations:

$$\mathbf{C}_1 = [C_{1x}, C_{1y}], \qquad \mathbf{C}_2 = [C_{2x}, C_{2y}]. \tag{29}$$

Equating (28) and (29), we can write

$$C_{1x} = -a \cdot \cos\theta + T_x, \quad C_{1y} = -a \cdot \sin\theta + T_y;$$
$$C_{2x} = a \cdot \cos\theta + T_x, \quad C_{2y} = a \cdot \sin\theta + T_y.$$

Therefore,

$$C_{2x} - C_{1x} = P = 2a \cdot \cos\theta$$
$$C_{2y} - C_{1y} = Q = 2a \cdot \sin\theta.$$

Therefore, the length a and orientation angle θ of the brush $\mathbf{B}$ will be given by,

$$a = \tfrac{1}{2} \cdot \sqrt{P^2 + Q^2}, \qquad \theta = \tan^{-1}(Q/P). \tag{30}$$

And the trajectory equation T will be

$$T_x = C_{1x} + \tfrac{1}{2}P, \qquad T_y = C_{1y} + \tfrac{1}{2}Q. \tag{31}$$

Hence Theorem 6 is proved. □

Let us take a simple example to demonstrate its working. Let the contour be a circular annulus, with M as the radius of the inner circle and N as that of the outer circle. Therefore,

$$\mathbf{C}_1 = [M\cos\psi, M\sin\psi], \qquad \mathbf{C}_2 = [N\cos\psi, N\sin\psi].$$

So,

$$a = \frac{N - M}{2}, \qquad \theta = \psi$$

And

$$T_x = \tfrac{1}{2}(N + M)\cos\psi, \qquad T_y = \tfrac{1}{2}(N + M)\sin\psi.$$

If we write $N = R + r$ and $M = R - r$,

$$T_x = R\cos\psi, \qquad T_y = R\sin\psi.$$

The length of the brush is constant; that is, $a = r$ and the orientation function θ is $\theta = \psi$. Note that the same pair of outlines would have been generated by a circular brush of radius r (see eq. (9)). This, therefore, proves Theorem 4 by example.

For describing arbitrary regions using the canonical brush, it is necessary to first generate a convex subdivision of the given region and apply the method of Theorem 6.

The canonical brush result is interesting but not really very useful. This is because during the design phase a variety of brush shapes are used. The brush shapes are chosen such that the desired shape is naturally produced. Thus the deconvolution problem should be addressed to determining the most natural brush-trajectory combination for a given outline shape. A precise specification of the most natural combination is itself an open problem for further work.

ACKNOWLEDGMENTS

We are extremely grateful to the referees for their highly perceptive comments on an earlier version of the manuscript. We also thank Shri P. A. Koparkar and Dr. Pranab Ghosh for their willingness to discuss and formulate the proofs of some of the theorems stated in this paper.

REFERENCES

1. COUEIGNOUX, PH. Character generation by computers. *Comput. Graph. Image Process. 16*, 3 (July 1981), 240–269.
2. FAUX, I. D., AND PRATT, M. J. *Computational Geometry for Design and Manufacture.* Ellis Horwood, Chisester, England 1979.
3. GUIBAS, L. J., AND STOLFI, J. Lecture notes on computational geometry, CS445. Stanford University, Stanford, Calif., 1983.
4. KAROW, P. ET AL. *Ikarus-System: Computer-controlled font production for CRT and Lasercomp,* Karow Rubow Weber GmbH of Hamburg, Germany. Sept., 1979.
5. KNUTH, D. E. Mathematical typography. *Bull. Am. Math. Soc. 1*, 2 (1979), 337–372.
6. KNUTH, D. E. METAFONT, A system for alphabet design, part 3 of TeX and METAFONT: New directions in typesetting, American Mathematical Society and Digital Press, Bedford, Mass., 1979.
7. MEI, T. Y., LCCD, A language for chinese character design. *Softw. Prac. Exper. 11* (1981).
8. WEILER, K. Polygon comparison using a graph representation. *Comput. Graph. 14*, 3 (July 1980), 10–19.

Received November 1983; accepted July 1984