SKELETAL STROKES

S.C. Hsu,   I.H.H. Lee and   N.E. Wiseman

Computer Science Department	Mechanical Engineering Dept.	Computer Laboratory
Chinese University of Hong Kong	University of Hong Kong	University of Cambridge
Hong Kong	Hong Kong	United Kingdom
schsu@cs.cuhk.hk	*p92hhlee@hkumea.hku.hk*	*new1@cl.cam.ac.uk*

ABSTRACT

A *skeletal stroke* is a kind of general brush stroke for changing the shape of pictures as if by bending, shearing, twisting, while conservating the aspect ratio of selected features on the picture. It is neither a simple warping nor texture mapping technique, but a new method for controlling the deformation of a picture. A deformation model of a coordinate system has been proposed taking into account cases of discontinuous or extreme bending. Complicated pictures can be built up hierarchically by defining higher order strokes and recursive strokes. It is therefore a powerful general drawing tool and extended image transformation instrument. The use of skeletal strokes as a replacement for affine transformations in IFS coding has been explored. A novel general anchoring mechanism is proposed, which allows arbitrary control of any point in the picture. This control flexibility is particularly desirable in computer animation and digital typography. As a result, virtual '2-D models' of cartoon characters as well as pseudo 3-D objects can be created and manipulated with ease.

KEYWORDS:

Virtual device interfaces, skeletal strokes, user interaction, picture transformation, animation, fractals, digital typography.

INTRODUCTION

The brush and stroke metaphor has been used in computer based painting and drawing systems for quite a long time. Many of them are simply automated rubber stamps which build up images by repeating copies of some static or simply derived pattern.

Existing Brushes and Strokes

In [26], for example, a filtered image is used as the brush to be dragged along a path to create anti-aliased line drawings with a 3-D appearance.

In other systems, the shape of the brush can be chosen from a table of predefined brush patterns based on brush pressure and brush tilt[8]. Many painting programs like MacDraw offer 'air-brushes' which simulate a spray of ink by painting pixels in the vicinity of the brush. In [25], the brush painting process is digitally recreated to generate many interesting effects. Although this kind of brush stroke is powerful and expressive, it requires considerable amount of control with each stroke. For example, the pressure of the brush at each position along the stroke has to be specified. These methods and many others[23, 24, 14] are all based on a brush-trajectory model of drawing.

The concept of a thick stroke can be found in most graphics description systems or graphics function libraries such as GKS[2], POSTSCRIPT[1] and X library[13]. A thick stroke can be rendered stippled, pattern filled, dashed and drawn with different joints and stroke-end forms. Yao and Rokne[27] elaborated on the joining of two fat curves. However, these rather simplistic 'brush strokes' are useful only as low level graphics operations.

Skeletal Strokes

This paper describes a new type of stroke, called *skeletal strokes*, not using arrays of bristles or repeated patterns but arbitrary pictures as 'brushes'. The use of skeletal strokes is a user-interaction technique to specify how to transform a predefined picture along a path. It is a simple way to create expressive strokes and to abstract pictures for various applications. Effects that can be achieved range from marks made by a bristled brush (Figure 17) or a flat nibbed pen to a novel pseudo 3-D transformation of imaginary models.

Transformation of a picture along a path may involve bending, twisting, non-uniform compression and stretching of the picture. These problems were not addressed in [10] where a skeleton technique is employed for keyframe animation; the technique is actually an image warping technique. It allows the animation of a skeleton to be interactively controlled and interpolated, but the deformation of the coordinate frames around the skeleton may have to be adjusted manually. Furthermore, the positions where the skeleton can be bent are limited by the number of segments defined at the beginning. Barr[6], on the other hand, provided a general treatment on global and local deformations (twisting, bending and tapering) of 3-D objects based on

Jacobian techniques. However, the problems arising from discontinuous or extreme bending have not been discussed.

In this paper, we are proposing a new deformation model to deal with the above problems. How to define a skeletal stroke and what effects can be achieved with it will be described in the next section. The mechanisms behind and the applications of skeletal strokes to 2-D character animation and digital typography shall also be treated.

USING SKELETAL STROKES

Using skeletal strokes is analogous to using a schematic sketch of 'stick figures'. An artist represents a figure by a simple skeleton to plan the layout and posture. Flesh and details are then added. Artists instinctively define the appearance of the flesh in their minds while we have to specify the relation between skeleton and flesh explicitly.

Stroke Definition

Defining a skeletal stroke simply requires drawing an instance of the flesh, which could be any arbitrary picture, around a reference line with a right-angled bend (Figure 1). Our implementations support pictures composed of points, closed polygons or Bézier splines. Sampled images can be used too, as will be discussed in a later section.

The reference backbone provides a reference x-axis for the points specifying the position of the flesh; the reference thickness provides a scale to specify the lateral distance of these points from the reference backbone. A picture so placed around a single reference backbone and reference thickness line defines a *skeletal stroke*.
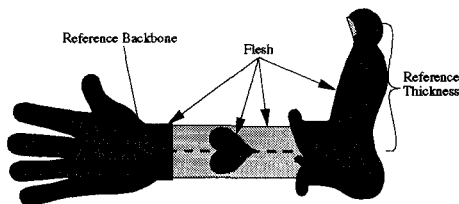

Figure 1: Definition of a skeletal stroke. The reference backbone and the reference thickness are shown in dashed lines.

Stroke Application

Once a skeletal stroke has been defined, it can be applied along any arbitrary path by simply drawing the path with a mouse or stylus. The reference backbone in the stroke definition is aligned with the given path and the flesh is distorted (stretched, compressed, bent and/or sheared) accordingly. In our implementations, the path could be a piecewise cubic spline or a polyline. Two other attributes, the width of the stroke and the shear angle of the flesh along the path, are also left at the user's discretion.

In this paper, we shall display the application path as a solid line/curve. The width and shear angle are indicated by an arrowed straight line attached to the beginning of the application path.

Sausage Style and Ribbon Style

Skeletal strokes can be applied in a *sausage* style or *ribbon* style as selected by the user. Figures 2 and 3 illustrate the difference between the two styles. In the sausage style, the shear angle is kept constant with the global coordinate system, while in the ribbon style, the shear angle is relative to the tangent of the path. Strokes drawn in the ribbon style look as if they are drawn with a flat nibbed pen.
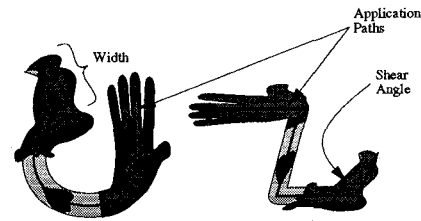

Figure 2: Application of the previous skeletal stroke using *sausage* style.

Figure 3: Application of same stroke using *ribbon* style.

More Control Over Stroke Application

Sometimes we may like to stretch or shrink different parts of a stroke in different amounts. Take the stroke in Figure 1 as an example. We may not want the 'hand' or 'foot' to be deformed as in Figures 2 and 3, but wish to keep their original proportions. We may even want to stretch the 'hand' and 'foot' to different aspect ratios but shrink the 'heart'.

Described below are the different types of control we can apply to a skeletal stroke to create various interesting effects.

Anchoring of Flesh

By *anchoring* part of the flesh to a particular point (called the *anchor-origin*) along the path, we can retain the aspect ratio of that part while the rest of the flesh is stretchable and shrinkable. To do this, the user chooses an *anchor-origin* to which a selected group of features (points) on the flesh are anchored. The anchor-origin, as the name implies, is taken as the origin of a local coordinate system which, in stroke application, is scaled in both axes by the width of the stroke being applied. The anchored features, drawn using this local coordinate system, can thus retain its aspect ratio.

Different effects of anchoring are illustrated in Figure 4 using the same stroke definition in Figure 1 and the same sausage style stroke application as in Figure 2. The anchoring of the flesh is particularly useful for typographical features like serif's, heads, tails, ears etc., which retain their shapes regardless of the stretchability of the strokes they are attached to.

We shall see in the section 'Anchoring Mechanism' how a generalized anchoring mechanism can be used to control the shape of a piece of flesh and how it could be used to create and manipulate pseudo 3-D objects that could not exist in real life.
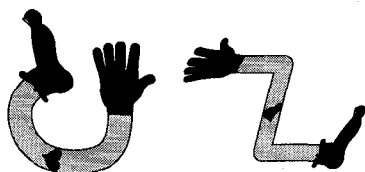

Figure 4: The 'hand' and the 'foot' have been anchored to the two ends of the stroke; the heart to the middle of the stroke. Notice that the aspect ratios of these features remain the same regardless of the length of the applied stroke, though they are still susceptible to distortion due to shearing and bending.
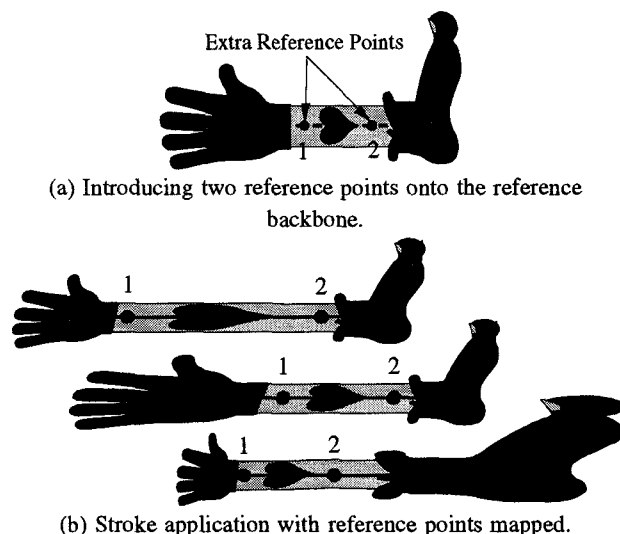

(a) Introducing two reference points onto the reference backbone.

(b) Stroke application with reference points mapped.

Figure 5: Mapping of reference points on the reference backbone to those on the stroke application path.

Controlled Mapping of Reference Points

Suppose we now want to stretch or shrink different parts of a stroke by different amounts. This can be done by introducing extra reference points on the reference backbone and dragging the corresponding reference points on the stroke application path (Figure 5). This method has effectively split the reference backbone into sections, each of which can then be controlled independently.

Variable Width and Shear Angle

If the width and the shear angle of an application path can be made to vary along its length, we achieve full expressiveness by using a single stroke. Such a stroke could simulate a soft brush, though we probably would need a more powerful input device such as a pressure and tilt sensitive stylus or the Drawing Prism[16].

Twisting is an interesting effect which is easy to specify. By allowing the width of stroke application to vary as

a cosine function along the path and applying a sequence of bending and twisting to a simple skeletal stroke, the effect of a Möbius strip in the style of Barr's[6] is achieved (Figure 6). Note that no 3-D information is specified in our skeletal strokes.
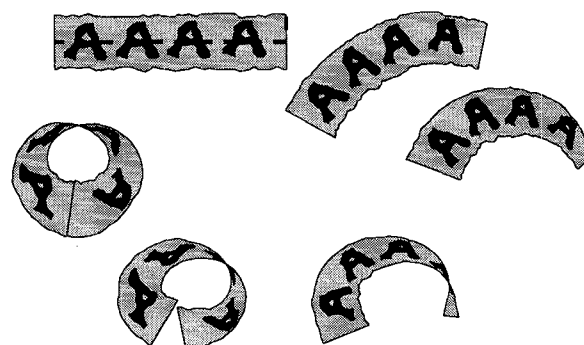

Figure 6: A '2-D' Möbius Strip.

Higher Order Stroke Definition

Higher order skeletal strokes can be created by specifying a reference backbone on a picture made up of stroke applications and simple objects like polygons, curves, lines etc. Such a stroke can be re-applied in the same way as any normal stroke defined purely by simple objects (Figure 7).

Recursive skeletal strokes can be defined to generate fractals in a single stroke application. A pure recursive skeletal stroke (i.e., one defined entirely in terms of itself) actually defines an extended form of IFS (Iterated Function Systems) codes[5, 4, 3]. The only difference is that the more powerful skeletal stroke applications instead of simple affine transformations are used to transform pictures. The average contractivity condition[4] still has to be obeyed. Such a recursive skeletal stroke can be realized using a random iteration algorithm similar to the Measure Rendering Algorithm for IFS codes. Here, one of the stroke applications in the stroke definition is picked randomly as one of the affine transformations would be. The algorithm is sketched below.

```
let p = (0,0);
    /* parametric coordinate system (see Sec.3.1)*/
for i=1 to n do
{
    define a stroke D with only point p on it;
    randomly pick a stroke application;
        (based on their relative coverage)
    apply the stroke application with D as
        the definition;
    let p = parametric position (relative to the
        reference backbone) of the point from the
        stroke application;
}
```

A recursive skeletal stroke, if defined to contain other objects besides stroke applications alone, is analogous to IFS codes with condensation sets[4]. Figure 8 shows some examples of recursive skeletal strokes.
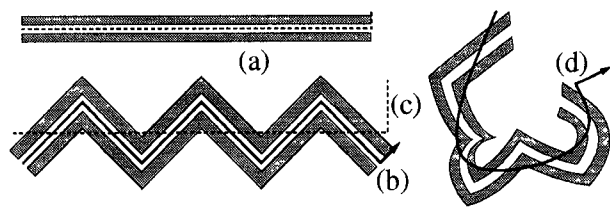
Figure 7: Higher order skeletal strokes: (a) a simple stroke is defined; (b) the stroke having been applied along a zig-zag path; (c) a new skeletal stroke is defined with a new reference backbone; (d) application of the new stroke along a curly path.
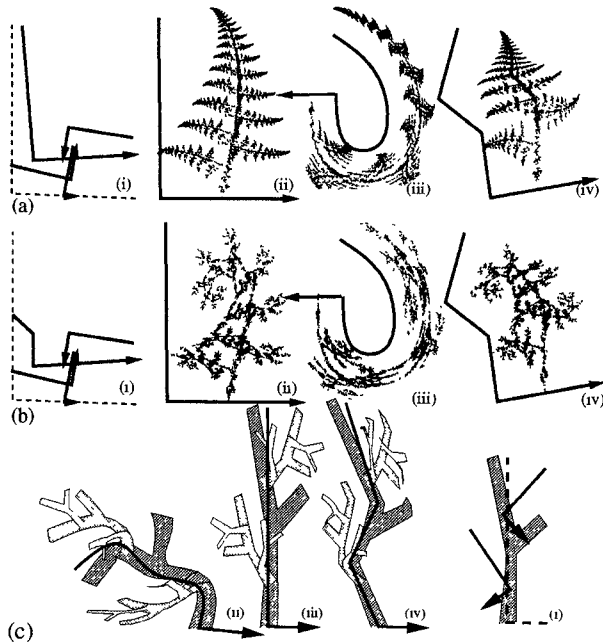

Figure 8: (a) & (b) Pure recursive skeletal strokes drawn with a random iteration algorithm. (a) The classic fern drawn along straight and curved/crooked application paths: (i) Definition of the 'fern' stroke in term of 4 applications of itself; (ii)–(iv) applying the stroke defined in (i). (b) Another species: (i) Stroke definition is identical with (a)(i) except that a bend has been introduced to the longest path in (a)(i); (ii)–(iv) Applying the stroke defined in(b)(i) to the same paths in (a)(ii)–(iv). (c) Trees generated from recursive strokes with condensation set (the twig): (i) Stroke definition; (ii)–(iv) applied on different paths. Shading of the twig is reduced by 10% on each recursive level.

MECHANISMS
Parametric Coordinates

When a skeletal stroke is defined, the outline of the flesh is recorded in parametric coordinates. The reference backbone and the reference thickness specify the [0, 1][0, 1] range of the parametric coordinate system for the whole stroke. This approach is similar to the skeleton coordinate system used in [10], except that the latter used different explicit coordinate frames for different parts of the picture.

The process of applying a skeletal stroke along an arbitrary path is actually a transformation of the parametric coordinate system so that the x-axis is aligned with the path and the y-axis is the instantaneous normal to the path.

Shrinkage and Expansion On Bending

On bending a physical elastic object, the material on the concave side of the bend will be compressed axially while that on the convex side stretched axially. The way and amount in which the material shrinks or expands depend on the physical properties of the material. The elasticity of real life materials would cause non-localized deformation. Buckling, lateral compression and expansion may occur. In a brush stroke, however, we may not want such consequences.

A simple method to deal with bending is to draw the flesh using the normal to the application path as the local y-axis. However, the flesh may look wrinkled, or may overlap or fold back on itself (Figure 9(b)).
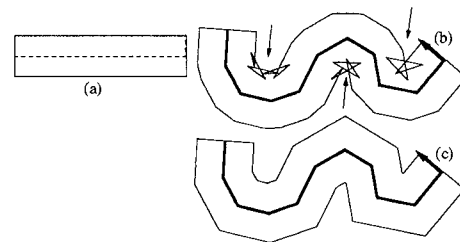

Figure 9: (a) Stroke definition. (b) Application of stroke by redrawing all the points along the normals. Wrinkled areas are indicated by arrows. (c) The same stroke redrawn with mitre joints based on our deformation model for mitre joints.

In Barr's paper[6] on Jacobian techniques for local deformation of the coordinate system, these effects and the cases of sharp bends were not addressed. Features on the concave side and beyond the centre of curvature of the bend are inverted and expanded in his method.

Deformation Model

Instead of using finite element theory to calculate the deformation of skin around a complicated bone structure[15] or using an algorithm based on kinesiology[11], we are proposing a deformation model of the coordinate system suitable for 2-D graphic objects.

We want to make it clear that it is the deformation of the cartesian space as a whole that we are modelling; we are not handling flesh and empty space individually. The latter, in fact, involves a complicated model for composite material; consider deforming a bowl of jelly with cocktail fruit suspended in it.

Consider an imaginary material which stretches like rubber and compresses like a piece of sponge does. It is infinitely stretchable and compressible and has a Poisson ratio of 0, i.e. no lateral changes are induced on axial deformation.

On bending, no lateral expansion is observed on the concave side and compression is localized near the bend only. On the convex side, the material expands without lateral compression. Because of spatial constraints on the concave side of the bend, all material located further than the centre of curvature of the bend will be compressed down to the centre of curvature (Figure 10). Bending a rectangular strip of this material will give a form identical to a thick stroke of the same width drawn with the line-drawing functions in POSTSCRIPT[1] and X library[13].
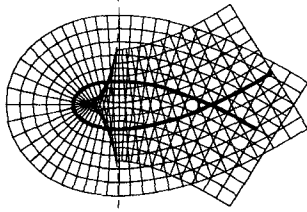

Figure 10: Deformed parametric coordinate system of a curved path segment. The dashed line is the locus of the centre of curvature. Note the spatial constraints.

Continuous Paths

Implementation of skeletal strokes based on this model is straightforward for first derivative continuous paths (e.g. Bézier curves, cubic splines or conic splines), because the locus of the centre of curvature is also continuous and the radius of curvature can be determined[19] thus:

$$R = \frac{[1 + (dy/dx)^2]^{\frac{3}{2}}}{(d^2y/dx^2)}$$

or in parametric coordinates:

$$R = \frac{[(dx/dt)^2 + (dy/dt)^2]^{\frac{3}{2}}}{(dx/dt)(d^2x/dt^2) - (dy/dt)(d^2y/dt^2)}$$

Polyline Paths

For polyline paths, however, the radius of curvature of all points along the path is infinite except at the bend where it is zero. The flesh is still spatially constrained by that on the segment on the other side of the bend. Consider a sharp bend formed by two straight path segments. The flesh whose y-position lies beyond the angle bisector of the two path segments will be compressed down to the angle bisector as shown below.
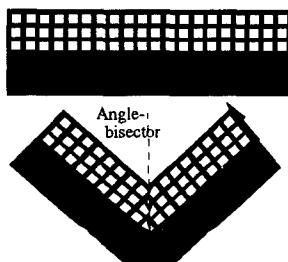
This type of ideal localized compression is similar to the compression of a piece of sponge and is, in fact, a clipping. The features in the compressed region is lost completely leaving a discontinuous appearance across the angle bisector.

Modified Deformation Model

We have adopted a modified version of such a material to allow for non-localized deformation on axial compression. Axial deformation of the material is spread linearly across the material between the two angle bisectors at the ends of each path segment. For the first and last path segments (which have no angle bisectors at the free end(s)), the effect of deformation ceases at the normal(s) at the free end(s). As can be seen from Figure 14, the amount of deformation increases towards the point the two angle bisectors (or normals) meet. This intersection point can therefore be viewed as a *macroscopic centre of curvature* (MCC for short) for that particular path segment. The MCC for a polyline path segment is analogous to the real centre of curvature for continuous curves described above. The convergence of coordinate space towards the MCC is also consistent with the case for continuous curves.

Note that even with such a deformation model we cannot completely eliminate the overlapping of flesh. In fact, we probably would not want to, as this would mean any crossing over of a stroke could not be realized. We have only allowed adjacent path segments to affect the deformation of the coordinate system of any one segment and have not attempted to propagate deformation effects from segments further away.

Adding Extra Points To The Flesh

The original points used to define a piece of flesh may be sufficient to represent the outline of the undeformed flesh. The shape of the deformed flesh, however, is usually more complicated, so the original number of points is most likely insufficient to specify the new shape. Extra points are hence introduced.
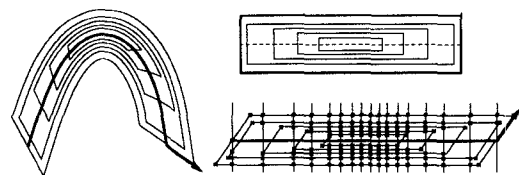

Figure 11: Each path segment is divided into portions of sizes inversely proportional to the local radius of curvature (or perpendicular distance to the MCC) of the path segment. If the two ends of a curve/line of the flesh lie in different portions, it is intersected with the y-axes at the portion boundaries. The intersection points thus form the extra points on the flesh.

It is obvious that the amount of distortion increases with the curvature of the application path, so more extra points are introduced around regions of high curvature. Figure 11 shows how this is done. This process is performed after the flesh is sheared but before the coordinate system is being

deformed. After deformation, the final set of points is joined to form the distorted outline of the flesh.

Reparameterization On Stroke Application

We still need to identify the positions of any joints (as specified on the application path) on the reference backbone. For polyline paths, the fractional positions of these joints are mapped back onto the reference backbone, which is then partitioned at these mapped points. Within each partition, the x-coordinates are reparameterized to a local coordinate system where x ranges from 0 to 1, while the y-coordinates remain unaffected. All partitions can now be treated independently. Piecewise curve paths are partitioned in the same way at the junction of curve segments.

Joint Types

The convex side of a sharp bend in a polyline path is drawn similar to that of thick strokes drawn with POSTSCRIPT. The bend could have a bevel joint, a round joint or a mitre joint.

Mitre joint At the convex side of a mitre joint, straight lines parallel to the path segment remain all the way straight and parallel. Therefore the convex and concave sides can be handled in exactly the same way, except that the convex side does not have the spatial contraints due to the convergence of coordinate space (Figure 12).
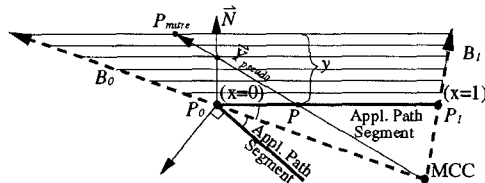

Figure 12: Mitre joint coordinate system.

A point on the flesh of parametric coordinates (x, y), after being applied to a polyline path with mitre joints, will be transformed to a new position (P_{mitre}) according to the following formula:

$$P_{mitre} = P + \frac{y}{\vec{Y}_{pseudo} \cdot \vec{N}} \vec{Y}_{pseudo}$$

where P is the point at a fraction x between the two ends, P_0 and P_1, of the path. $\vec{Y}_{pseudo} = (1 - x)\vec{B}_0 + x\vec{B}_1$, $\vec{N}$ is the unit normal vector to the path and $\vec{B}_0$ and $\vec{B}_1$ are the unit vectors along the angle bisectors at the joints at P_0 and P_1 respectively.

Bevel and Round Joints Bevel and round joints, on the other hand, would cause a straight line parallel to the path segment to bend abruptly or to bend as an arc at the joint. If we consider one side of the angle bisector of such a joint, this bending region is, to more specific, the triangular region bounded by the angle bisector itself and the normal of the path segment. We shall call it the *joint zone*. Here is the region that distinguishes the three types of joints from each

other, so in dealing with bevel and round joints, we could treat them as mitre joints first and then adjust the coordinate system in the joint zone accordingly.
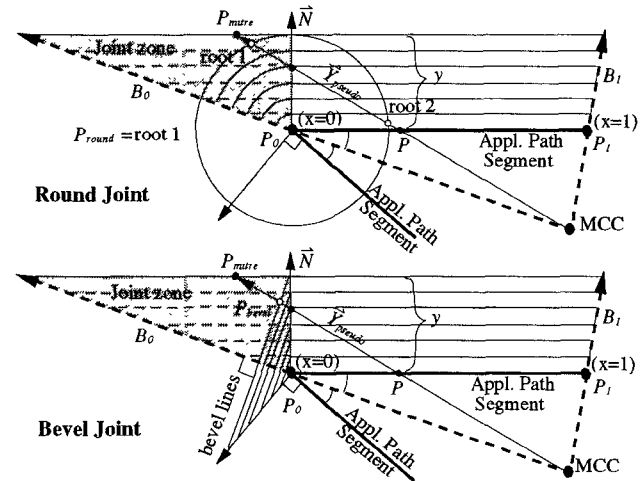

Figure 13: Adjusting the coordinate system of a mitre joint to that of a round or bevel joint.

Since for every path segment there is a joint zone at either end, we shall, without loss of generality, consider the joint zone at P_0 only. To find out whether a point P_{mitre} (established from the mitre joint method) falls within the joint zone, the following test is used (Figure 13). If the distance between P and the intersection of $\vec{Y}_{pseudo}$ and $\vec{N}$ is longer than that between P and P_{mitre}, the test is positive and the point should be drawn using the appropriate modified coordinate system.

In the case of a round joint, lines of constant y are circular arcs centred at the joint. $\vec{Y}_{pseudo}$ is therefore intersected with a circle of radius y and the root closer to the point P_{mitre} is taken as the adjusted position.

For a bevel joint, lines of constant y are straight lines perpendicular to the bisector and they span the normals on either side of the angle bisector. We shall call these lines *bevel lines*. $\vec{Y}_{pseudo}$ is intersected with the bevel line at a distance $y(\vec{B}_0 \cdot \vec{N})$ from the joint P_0. The intersection point is again taken as the adjusted position.

The effects of the three types of joint on bending a skeletal stroke with a regular grid pattern are shown in Figure 14.
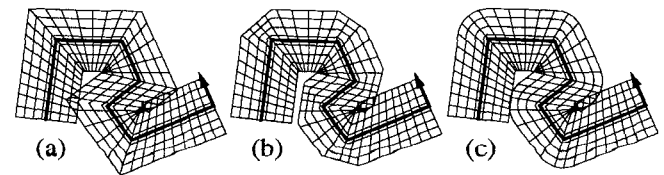

Figure 14: Three types of joint. Mitre joint (left), bevel joint (middle) and round joint (right).

Texture Mapping

If we use the undistorted skeletal stroke as a texture coordinate space[9], sampled images can be deformed with

a stroke application using standard texture mapping techniques. Since the relation between the original (defined by the reference backbone) and the distorted (specified in the stroke application) coordinate systems is well defined, proper filtering of the texture map can be performed easily.

Anchoring Mechanism

In the following stroke definition, a point p is anchored to the anchor-origin a. After having been applied along a straight path, the point p is transformed to a new point P and the anchor-origin is now at A. For convenience, we shall use small letters to denote positions on the reference backbone and capitals positions on the application path. The subscripts x and y denote the x-coordinate and the y-coordinate respectively.

<center>**Definition**   **Stroke Application**</center>
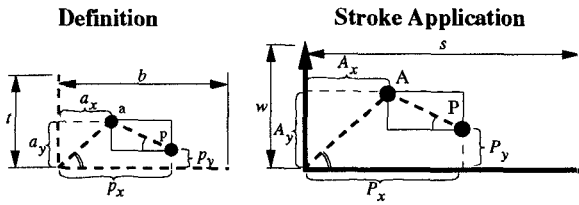
On stroke application, the position of the anchor-origin is scaled by the factor s/b. The x and y distances between the anchor-origin and the the general point is scaled by the factor w/t.

$$A_x = (s/b)a_x$$
$$(P_x - A_x) = (w/t)(p_x - a_x)$$

rearranging, $P_x = K[a_x(\alpha-1)+p_x]$ (1)

Similarly, $P_y = K[a_y(\alpha-1)+p_y]$ (2)

where s is the length of application path, w its width, b the length of reference backbone and t the reference thickness. K is the ratio w/t, the global scaling factor for distances to the anchor-origin; it is dependent on w only since t is fixed. α is the ratio $\frac{s/w}{b/t}$, or the ratio of the aspect ratio of the applied stroke, s/w, to the original aspect ratio b/t.

Intuitively, the term $(\alpha-1)$ (which we shall call Δ) can be interpreted as the amount of change in aspect ratio on stroke application. Note that Δ is dependent on the aspect ratio of the stroke application only, since the geometry of the reference backbone is fixed. When Δ is zero, i.e., the aspect ratios of the stroke application and the reference backbone are the same, A will be on the same fractional position as a.

We can see from Equations (1) and (2) that P is a linear function of Δ in both x and y directions. Different points on a flesh outline, if anchored to different anchor-origins, will form different linear relationships with Δ. This implies that the shape of a piece of anchored flesh can be distorted dramatically beyond the limits of stretching, shrinking, shearing, twisting and bending.

Higher order anchoring refers to the anchoring of anchor-origins themselves to other anchor-origins. It is a more powerful way to control the flesh points on the application path.

a^1 is an anchor-origin, itself being anchored to $\bar{a}^o$. On stroke application, a^1 is transformed to the position $A^{1'}$. $A^{1'}$ is obtained from Equations (1) and (2):

$$A_x^{1'} = K(\bar{a}_x^o\Delta + a_x^1) \quad (3)$$
Similarly, $A_y^{1'} = K(\bar{a}_y^o\Delta + a_y^1) \quad (4)$

A 1st order anchor-origin, $\bar{a}^1$, is created by mapping $A^{1'}$ back to the reference backbone. It is obvious that the scaling factor to use is $\frac{1}{K}$.

$$\bar{a}_x^1 = \frac{1}{K}A_x^{1'}$$
From (3), $\bar{a}_x^1 = \bar{a}_x^o\Delta + a_x^1$
Similarly from (4), $\bar{a}_y^1 = \bar{a}_y^o\Delta + a_y^1$

Now other points or even anchor-origins can be anchored to $\bar{a}^1$. In general, a point p anchored to an nth order anchor-origin $\bar{a}^n$ will result in P given by:

From (1), $P_x = K(\bar{a}_x^n\Delta + p_x)$
where $\bar{a}_x^n = \bar{a}_x^{n-1}\Delta + a_x^n$
and $\bar{a}_x^o = a_x^o$

Expanding,

$$P_x = K(a_x^o\Delta^{n+1} + a_x^1\Delta^n + a_x^2\Delta^{n-1} + \cdots + a_x^n\Delta + p_x)$$

Similarly,

$$P_y = K(a_y^o\Delta^{n+1} + a_y^1\Delta^n + a_y^2\Delta^{n-1} + \cdots + a_y^n\Delta + p_y)$$

P is now an $(n+1)^{th}$ order polynomial of Δ whose coefficients are the intermediate anchor-origins at various levels of the anchoring process. As these anchor-origins are independent of each other, we have total control over each term in the polynomial and hence over the behaviour of the resulting points by specifying the a values.

Figure 15: Pseudo 3-D transformation with 1st order anchoring of points. The visibility of the faces is determined by a hidden surface removal trick: if the ordering of the corner points of a face is reversed, that face is hidden.

Figure 15 shows a skeletal stroke (the flesh is made of three closed quadrilaterals) that *looks* like a cube. The corners of the quadrilaterals have been anchored to different anchor-origins which have in turn been anchored to other anchor-origins. Changes in aspect ratio in the stroke application have not resulted in a stretching or shrinking of

---

the picture but rather an effect of the cube being rotated. The anchoring mechanism is a trade off of the independent control of x and y scaling in return for control over the final shape of the flesh. Nonetheless, the flexibility of skeletal strokes in bending, shearing and twisting has not been sacrificed (Figure 16).
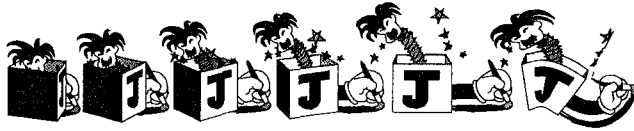

Figure 16: An animation sequence encapsulated in a single skeletal stroke. The hand and the brush are anchored to the end of the stroke. The dash of ink is freely stretchable while other objects are anchored to different 1st order anchor-origins.

The effect of 0th order anchoring is intuitive but that of higher order anchoring is not so. However, the need to determine where to put the higher order anchor-origins can be avoided. We can specify by examples how the picture should look like at particular values of Δ, just as we would specify the keyframes at particular times in keyframe animation. The system would then calculate automatically where the anchor-origins should be by solving the set of linear equations. Note that information on how a picture looks like and how it should deform is all recorded as positions in a stroke!

Figure 17: Chinese calligraphy in skeletal strokes. The part in the definition that is shaded in gray is anchored to the end of the stroke.

APPLICATIONS

Brush strokes, pen strokes and calligraphic strokes are different classes of strokes, each with its own characteristic features. The way in which skeletal strokes is used to abstract a drawing encapsulates these characteristic features in a single stroke definition. A picture created with skeletal strokes can thus be modified by a global alteration in the stroke definition level without touching the stroke applications at all. The user interface is through the drawing of the stroke width and application path, so provides a clean and expressive way to draw pictures.

General Drawing Tool

As illustrated by our example pictures, the skeletal stroke technique has proved itself to be capable of meeting professional requirements in high quality illustration[7]. The time

required to generate the stroke application for a rather complicated stroke (e.g. Figures 16) is less than half a second on a DECstation 5000, therefore interactive use is possible. The time required depends on the length and croakedness of the path. IFS strokes like those in Figure 8 take longer; typically 10000 random iterations can be performed in around 5 seconds.

2-D Models and Animation

Many published 'skeleton' based 3-D animation techniques [28, 11, 15] are based on physical deformations of 3-D realizable models. Cartoon characters, however, often cannot be realized as ordinary 3-D objects since they are designed as 2-D forms in the first place. Their artists only define a few pictures of how they look from different angles. How they look in between these angles is left to the imagination of the reader. Snoopy is a typical example. The shape of Snoopy's head stays almost unchanged even when it is turned through over 60 degrees. Only when he is looking straight to us does his head change shape (Figure 19(b)). Another example is Mickey Mouse. The slant of ears were defined to be 45 degrees unless in the front view when they are at the same vertical position (Figure 19(a)).
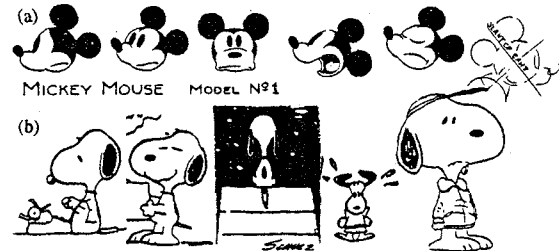

Figure 19: (a) Mickey Mouse by Walt Disney (from [12]). (b) Snoopy by Charles Schulz.

In the animation of such cartoon characters, the use of 3-D models is not practical. Instead, '2-D models' may be a more natural way to construct and manipulate such characters. Using skeletal strokes to represent a character allows us to distort different features on the character to different amounts according to the aspect ratio of the stroke application (Figure 20). To control one whole character by only one skeletal stroke is of course not flexible enough. We could use different strokes to model different parts of the character while the relative positions of these parts can still be controlled by the anchoring mechanism. In this way, 2-D virtual objects can be manipulated interactively as we would manipulate 3-D objects.
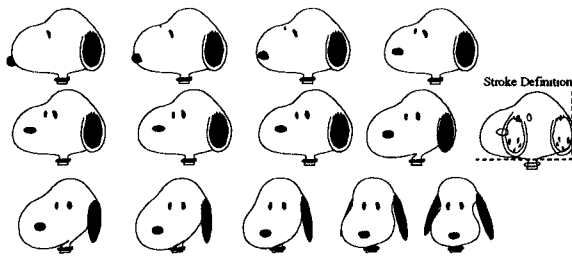
Figure 20: Snoopy's head modelled with one skeletal stroke. It turns from a side view to a front view as the length of the application path doubles.

Type-face Design

As stated in [17], the design quality of a type-face, i.e. the qualities of its aesthetic form, is governed generally by three criteria: (a) the formulation and development of its stroke-endings; (b) the formal proportions of the type image and (c) the character width, the consecutive rhythm of the type-face.

Skeletal strokes provide a natural modularization of these considerations, allowing the designer to concentrate on a particular design aspect. Features like serif's, ears, spurs and terminals can be anchored to the ends of stems and handled without having to worry about the overall proportions of the character itself. The control of weight and blackness is independent of the design of its form and style. These new techniques are far more expressive than Knuth's use of elliptical brushes[20, 21, 22, 23].

Type-face Design System Based on Skeletal Strokes

A skeletal stroke based type-face design system[18] has been developed. A design or meta-design[21] of a character is composed of basic skeletal strokes and sub-characters. Figure 21 shows a test set of characters designed from such a meta-design.

Figure 21: Roman characters test set. All four designs are generated from the same meta-design in which 4 different skeletal strokes have been used for the hairline, stem, arch and upper serif bracket. The top two designs have large stem-widths but small x-heights, in contrast to the lower two designs.

Modifications to existing designs are easy to make due to the high level of abstraction achieved in skeletal strokes. In Figure 22, for example, we have replaced the lower serif's of all the stems of the test set in Figure 21 by shoes, except

at the right-hand stem of the letter 'h', where a cursive stroke has replaced the serif design.
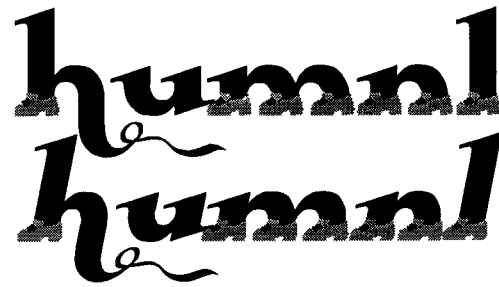

Figure 22: Modification of the design and introducing exceptions.

This approach to type-face design is particularly useful for Chinese type-faces. With the concept of a generic composite character[18], a single generic meta-design can be used to generate many designs for all Chinese characters with the same composite structure (Figure 23).
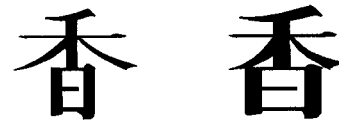

Figure 23: Three instances of a meta-design for the character 'fragrant' generated automatically from the same generic composite design for vertically stacked characters.

Storage Efficiency

The use of skeletal strokes implicitly extracts similarities of features from characters, reducing the storage space to a minimum. With a library of predefined skeletal strokes in hand, the storage of a character design just requires the storage of the stroke applications. This opens up new possibilities for storing more and larger founts in printers.

CONCLUSION

Skeletal strokes form a new framework for picture manipulation. Their power lies in their ability to abstract an image or an object into a single skeleton. Since both the definition and the application of a stroke is specified with a simple line or curve, this makes the design process intuitive and easy to control. Pictures can even be built up by the use of higher order strokes.

As a general drawing tool, skeletal strokes offer a variety of effects: bending (in sausage or ribbon style), shearing, twisting and retaining of aspect ratios of selected parts in a picture. As a transformation method, it is more powerful than the simple affine transformations. The ability of skeletal strokes to perform bending and twisting opens up new opportunities for more compact IFS encoding of shapes that were difficult using IFS with affine transformations (Figure 24). In digital typography, stroke-end forms and type-face proportions can be designed separately, while

no explicit programming is needed (c.f.[23]). In animation, separate control over character appearance and positioning is made possible by the anchoring mechanism. Flexibility in manipulation of different parts of a character is achieved by higher order anchoring. Being more powerful than an image interpolator, skeletal strokes can be further used to create and animate 2-D models and pseudo 3-D objects.
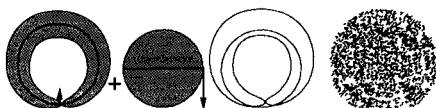

Figure 24: Extended IFS encoding of a circle using two recursive stroke applications.

References

[1] Adobe Systems Inc. POSTSCRIPT *Language: Reference Manual*. Addison-Wesley, 1989.

[2] American National Standard Institute. ANSI Standard X3.124-1985 Graphical Kernel System — GKS., 1985.

[3] Michael F. Barnsley. *Fractals Everywhere*. Academic Press, 1988.

[4] Michael F. Barnsley, Arnaud Jacquin, Francois Malassenet, Laurie Reuter, and Alan D. Sloan. Harnessing Chaos for Image Synthesis. In *SIGGRAPH Conference Proceedings*, volume 22, August 1988.

[5] Michael F. Barnsley and Alan D. Sloan. A Better Way to Compress Images. *BYTE*, January 1988.

[6] Alan H. Barr. Global and Local Deformations of Solid Primitives. In *SIGGRAPH Conference Proceedings*, volume 18, July 1984.

[7] Richard Beach and Maureen Stone. Graphical Style Towards High Quality Illustrations. In *SIGGRAPH Conference Proceedings*, volume 17, July 1983.

[8] Teresa W. Bleser, John L. Sibert, and J. Patrick McGee. Charcoal Sketching: Returning Control to the Artist. *ACM Transactions on Graphics*, 7(1):76–81, January 1988.

[9] J. F. Blinn and M. E. Newell. Texture and Reflection in Computer Generated Images. *Communications of the ACM*, 19, October 1976.

[10] N. Burtnyk and M. Wein. Interactive Skeleton Techniques for Enhancing Motion Dynamics in Key Frame Animation. *Communications of the ACM*, 19, October 1976.

[11] John E. Chadwick, David R. Haumann, and Richard E. Parent. Layered Construction for Deformable Animated Characters. In *SIGGRAPH Conference Proceedings*, volume 23, July 1989.

[12] Christopher Finch. *The art of Walt Disney*. Abrams, 1975.

[13] Jim Gettys, Ron Newman, and Robert W. Scheifler. *Xlib - C Language X Interface*. Massachusetts Institute of Technology, 1988.

[14] Pijush K. Ghosh and S. P. Mudur. The Brush-Trajectory Approach to Figures Specification: Some Algebraic-Solutions. *ACM Transactions on Graphics*, 3(2):110–134, April 1984.

[15] Jean-Paul Gourret, Nadia Magnenat Thalmann, and Daniel Thalmann. Simulation of Object and Human Skin Deformations in a Grasping Task. In *SIGGRAPH Conference Proceedings*, volume 23, July 1989.

[16] Richard Greene. The drawing prism: a versatile graphic input device. *SIGGRAPH Conference Proceedings*, 19(3):103–110, July 1985.

[17] André Gurtler and Christian Mengelt. Fundamental Research Methods and Form Innovations in Type Design Compared to Technological Developments in Type Production. *Visible Language*, 19(1), 1985.

[18] Siu Chi Hsu. *Computer Support for Large Character Set Languages*. PhD thesis, University of Cambridge, Computer Laboratory, December 1991.

[19] Glenn James and Robert C. James, editors. *Mathematical Dictionary*. D. Van Nostrand Company Inc., 1949.

[20] Donald E. Knuth. *TₑX and METAFONT, New Directions in Typesetting*. American Mathematical Society and Digital Press, 1979.

[21] Donald E. Knuth. The Concept of a Meta-Font. *Visible Language*, 16(1), 1982.

[22] Donald E. Knuth. Lessons Learned from METAFONT. *Visible Language*, 19(1), 1985.

[23] Donald E. Knuth. *The METAFONT Book*. Addison Wesley, 1989.

[24] K. C. Posch and W. D. Fellner. The Circle-Brush Algorithm. *ACM Transactions on Graphics*, 8(1):1–24, January 1989.

[25] Steve Strassmann. Hairy Brushes. In *SIGGRAPH Conference Proceedings*, volume 20, August 1986.

[26] Turner Whitted. Anti-Aliased Line Drawing Using Brush Extrusion. In *SIGGRAPH Conference Proceedings*, volume 17, July 1983.

[27] C. Yao and J. Rokne. Fat curves. *Computer Graphics Forum*, 10:237–248, 1991.

[28] D. Zeltzer. Motor Control Techniques for Figure Animation. *IEEE Comp. Graphics and Appl.*, 2:53–59, November 1982.