Computer Simulation of Calligraphic Pens and Brushes

Karen Donoghue & Ken Knowlton
Freestyle/Advanced Technologies Group
Wang Laboratories, Inc.

ABSTRACT:

A computer graphic method is described for simulation of calligraphic pens and brushes. Unlike a paint system which uses a mouse, here the artist uses a force-transducing pen to create realistic pen or brush strokes in real time. The system has been used for the production of western-style calligraphy and for writing with brush-based alphabets such as kanji.

Successive values of x, y, and pen-tablet force are measured, and others are derived--such as velocity, accumulated stroke length, and stationary dwell time. These are all used to define the momentary geometric footprint of pen or brush; the incremental geometry of the stroke depends in turn on rules for connecting successive such footprints. Additional rules may apply at the beginning or end of the stroke.

Crucial to the real-time aspects of the method are a very fast means for computing the momentary footprint, and for defining and filling the area connecting the current footprint with the previous one. This high-speed processing permits regular, uninterrupted acquisition of input data (thus avoiding improper polygonal faceting of curved strokes) and it produces a screen display without noticeable delay.

Later, hard copy output mat be produced at higher resolution by recomputing from the same data.

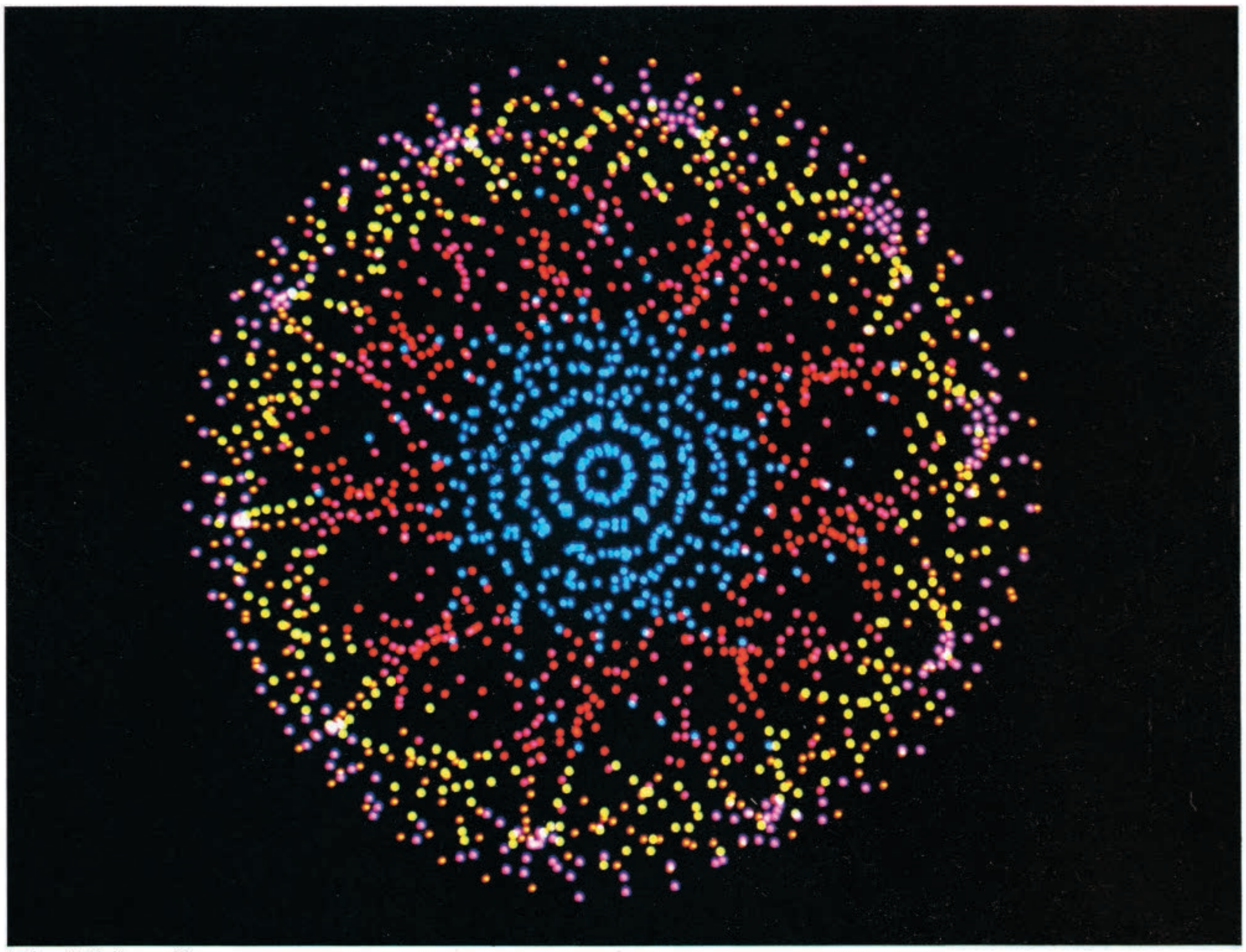
Early History of the Project:

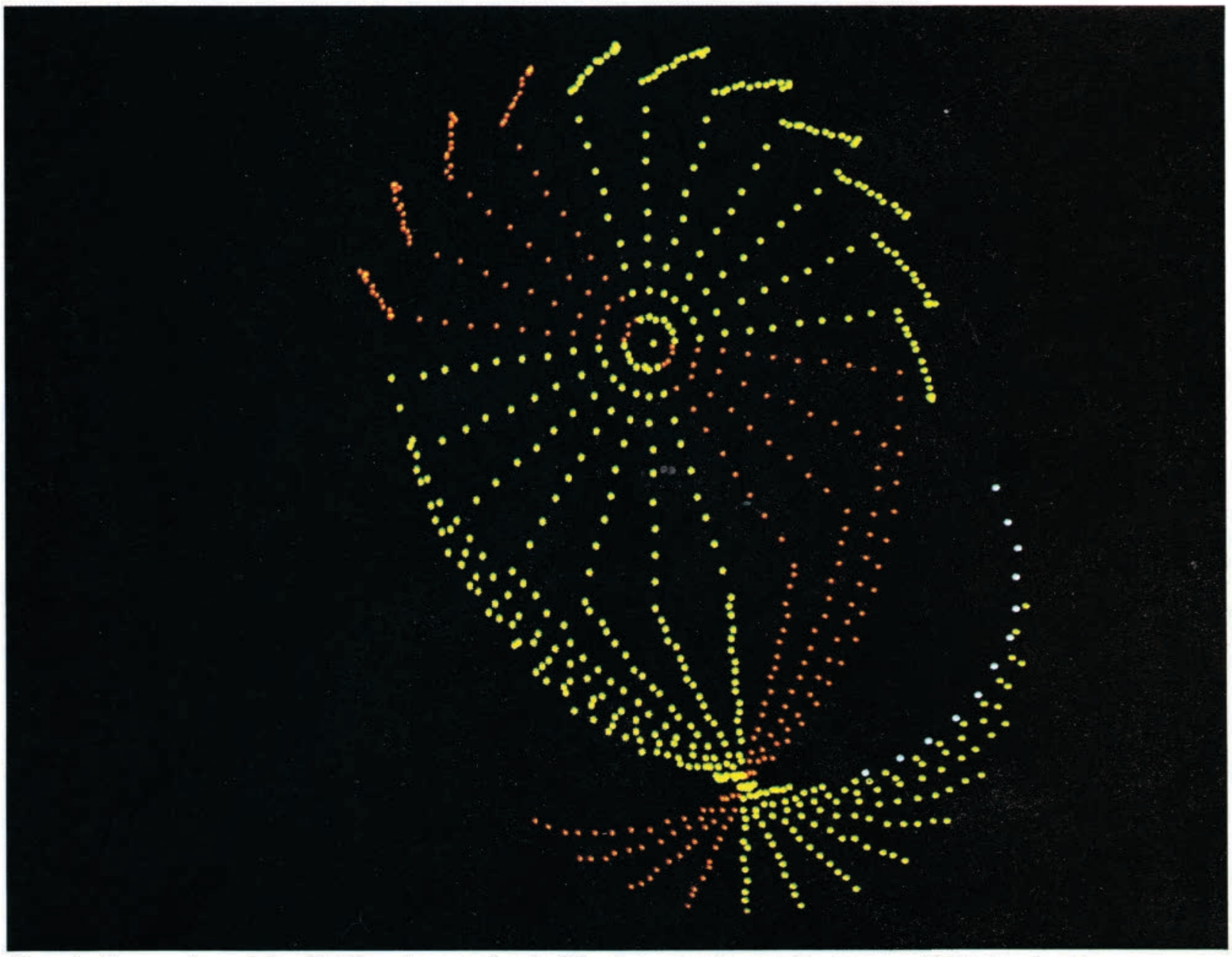
This project arose as a result of talks regarding force sensitive annotation capability within the Freestyle* annotation environment, and the desire to be able to vary stroke width and shape dynamically as a function of force. Initially, a force sensitive pen was implemented as a rigid broad-edged Western style calligraphy pen. The underlying software used for this pen was then built upon to model the Asian style brushpen described in the following paper.

Reasons for Doing This Project:

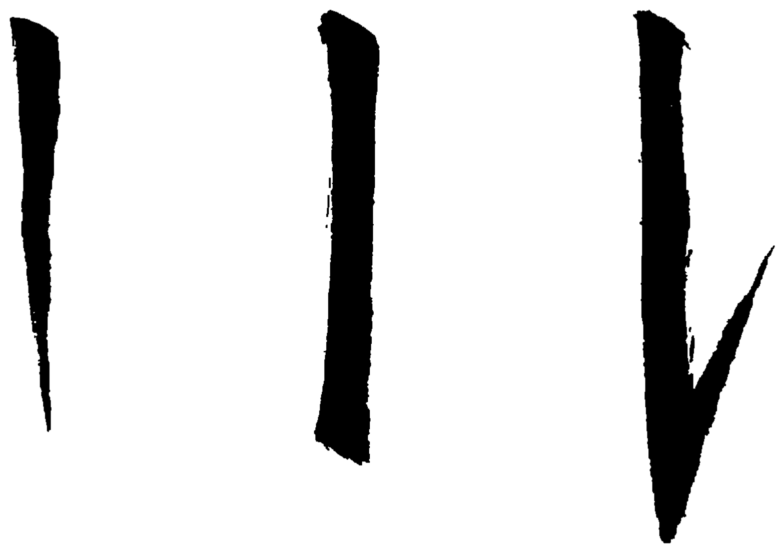
A computer model of an Asian brush pen is described, using the Freestyle force sensitive pen and tablet as input devices. Though the underlying software can be used to model any type of brush, we have chosen at this time to model a brush pen that one might use to do Chinese or Japanese calligraphy (kanji). This type of Asian brush writing is a good model to simulate, as it consists of strictly defined stroke forms, and strictly defined techniques for stroke creation. This type of computer model is useful as an alternative method of annotation in a document processing system, particularly in the Far East, where stroke weight (i.e. - line width and shape) are important to the aesthetics of the writing style as well as to human readability. In addition, it is a useful method of graphically-based input for systems used in cultural areas where the

*__Freestyle__ is a registered of Wang Laboratories, Inc.

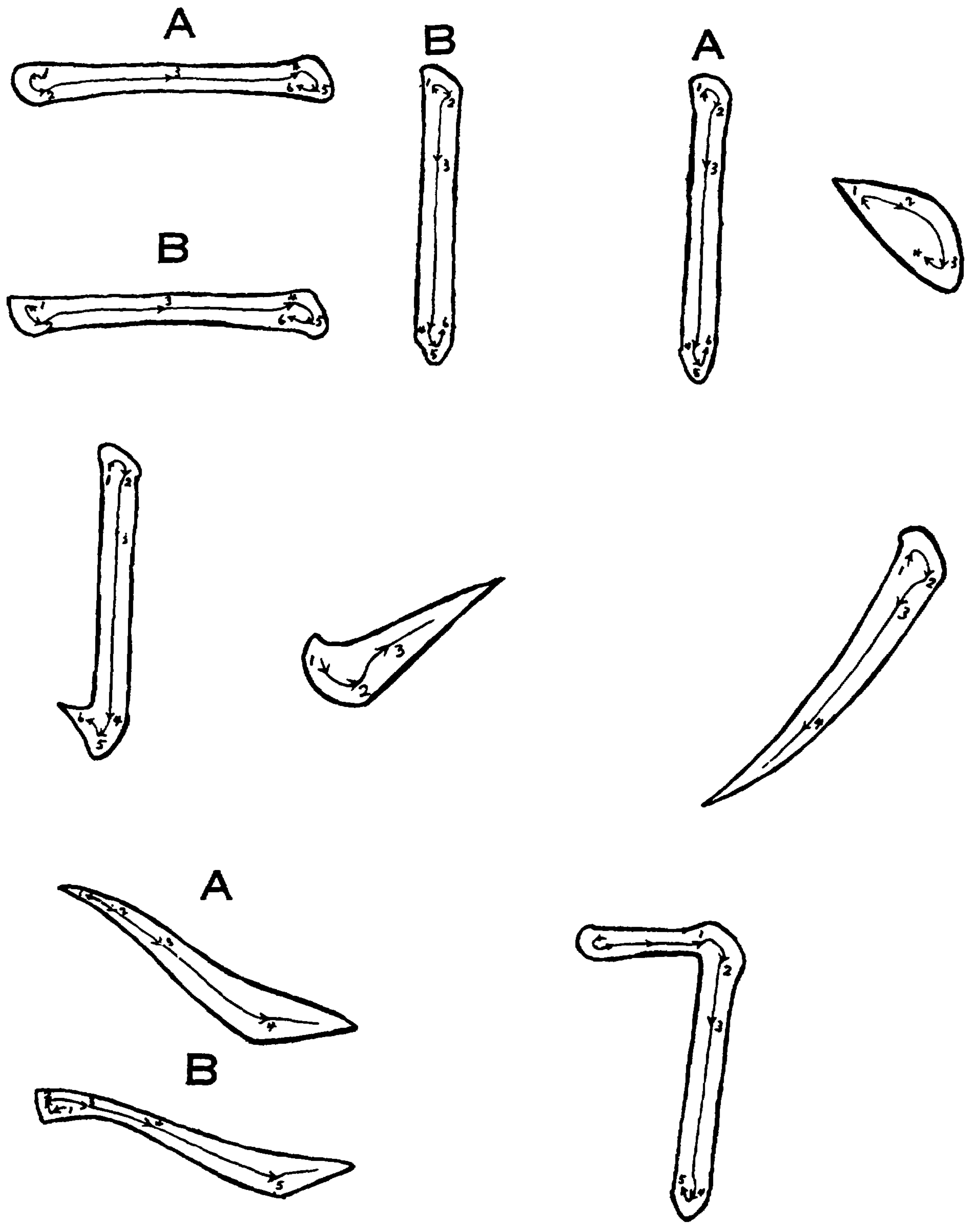

John Whitney Sr



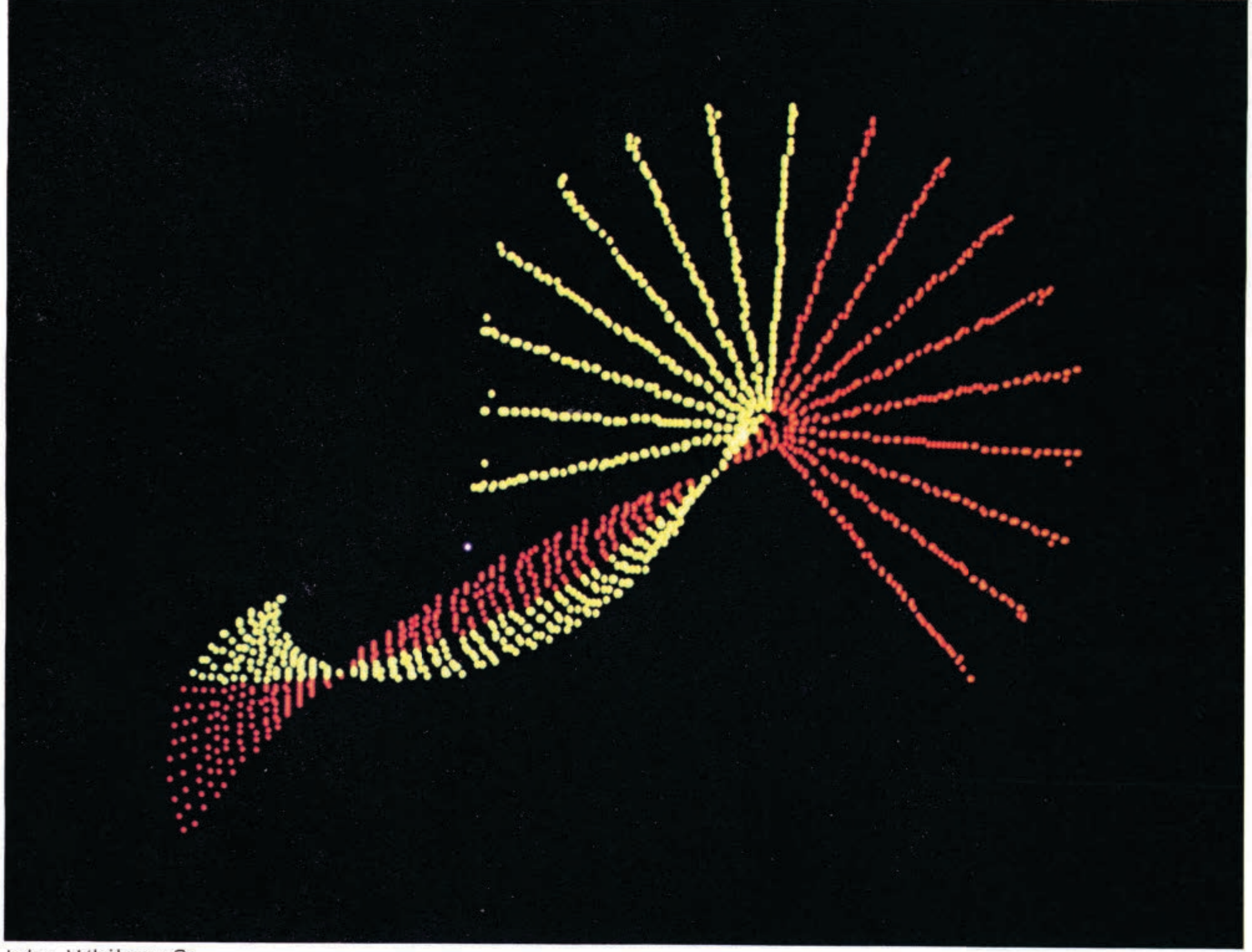
Sample frames from John Whitney's recent set of twelve music-graphic compositions inspired by early native American pottery, textiles, pictographs and artifacts.

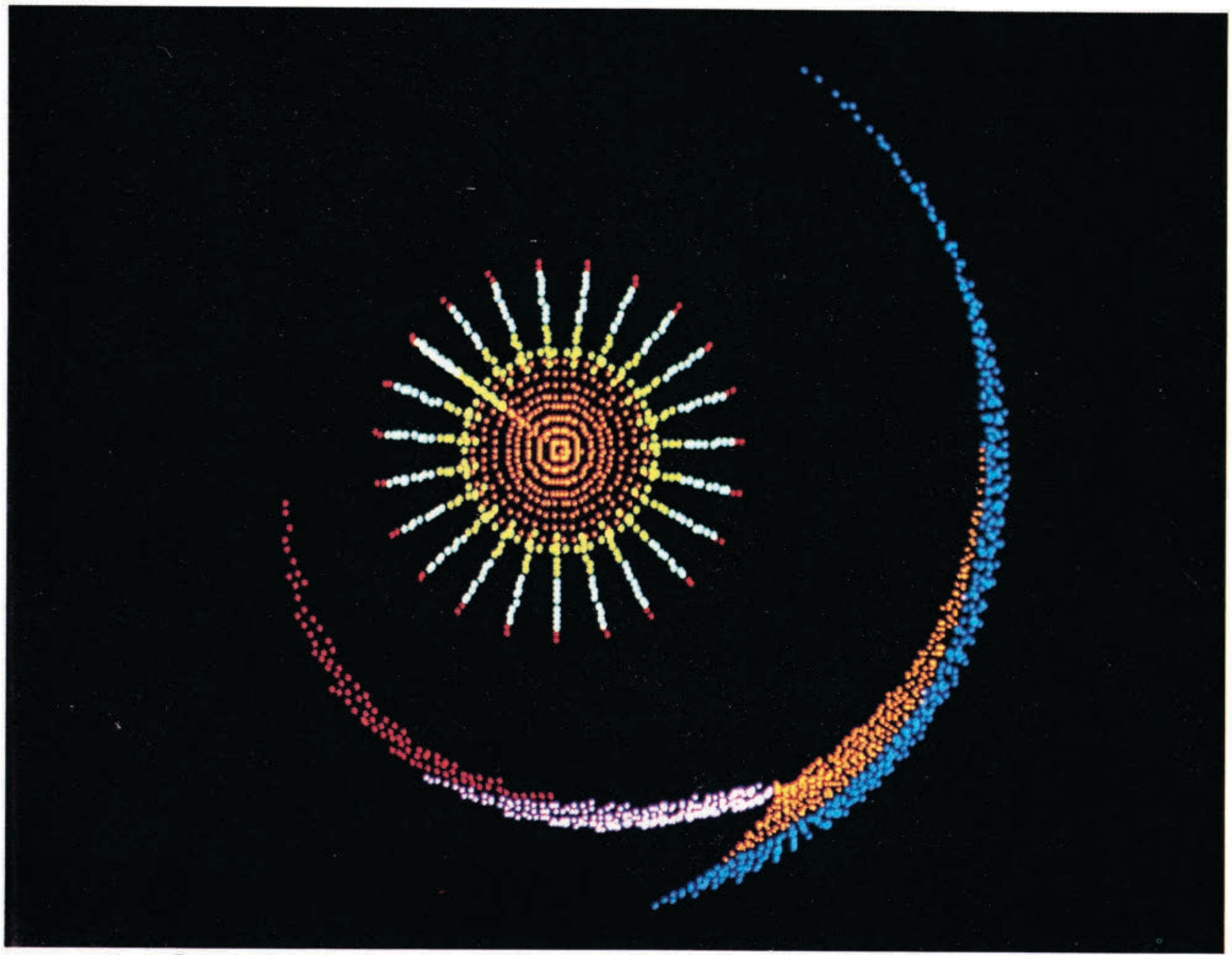


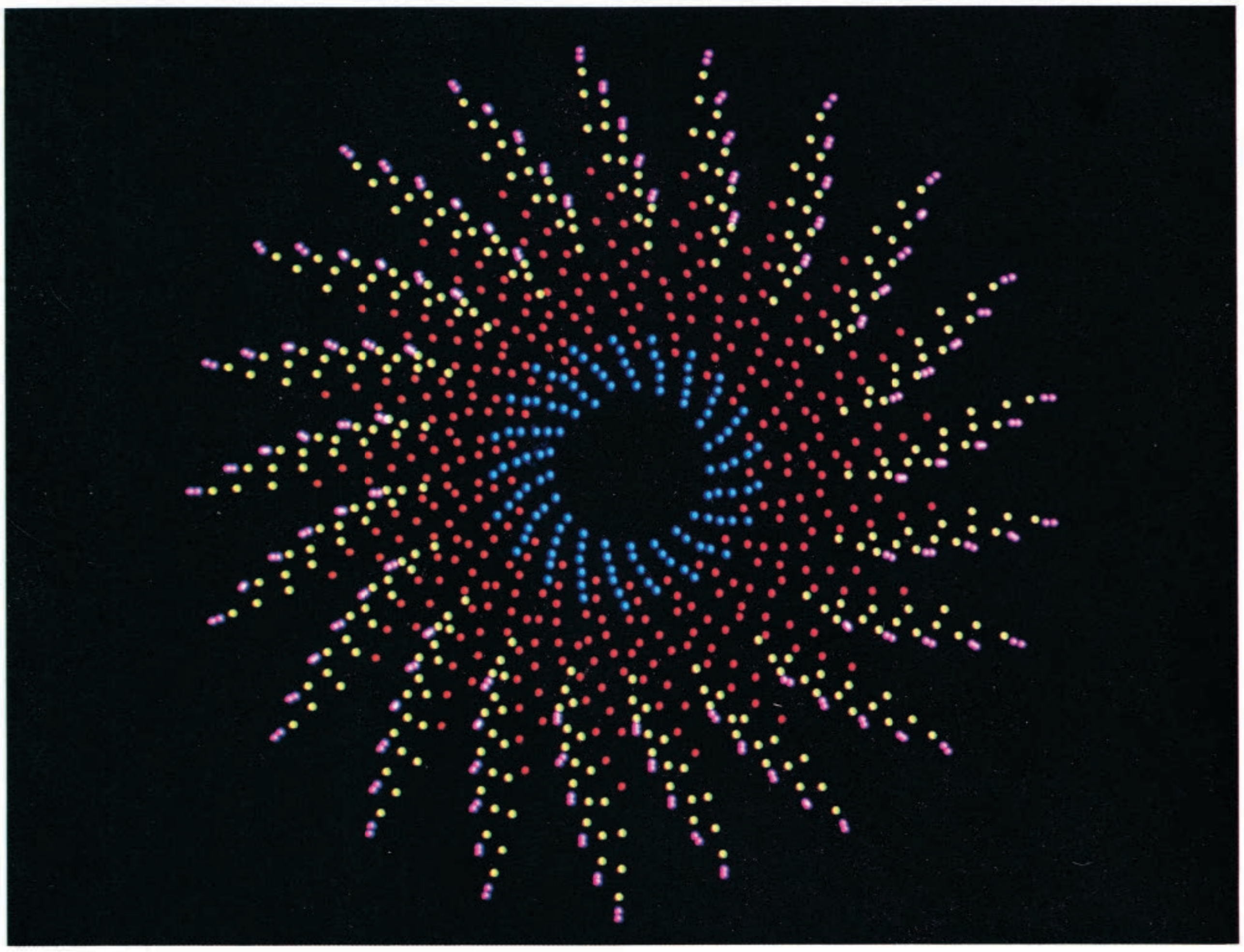
Karen Donoghue & Ken Knowlton
fig 1- The 3 basic kanji stroke styles, from left to right, linear lift, lift at 135 degree angle and lift with a backward sweep.

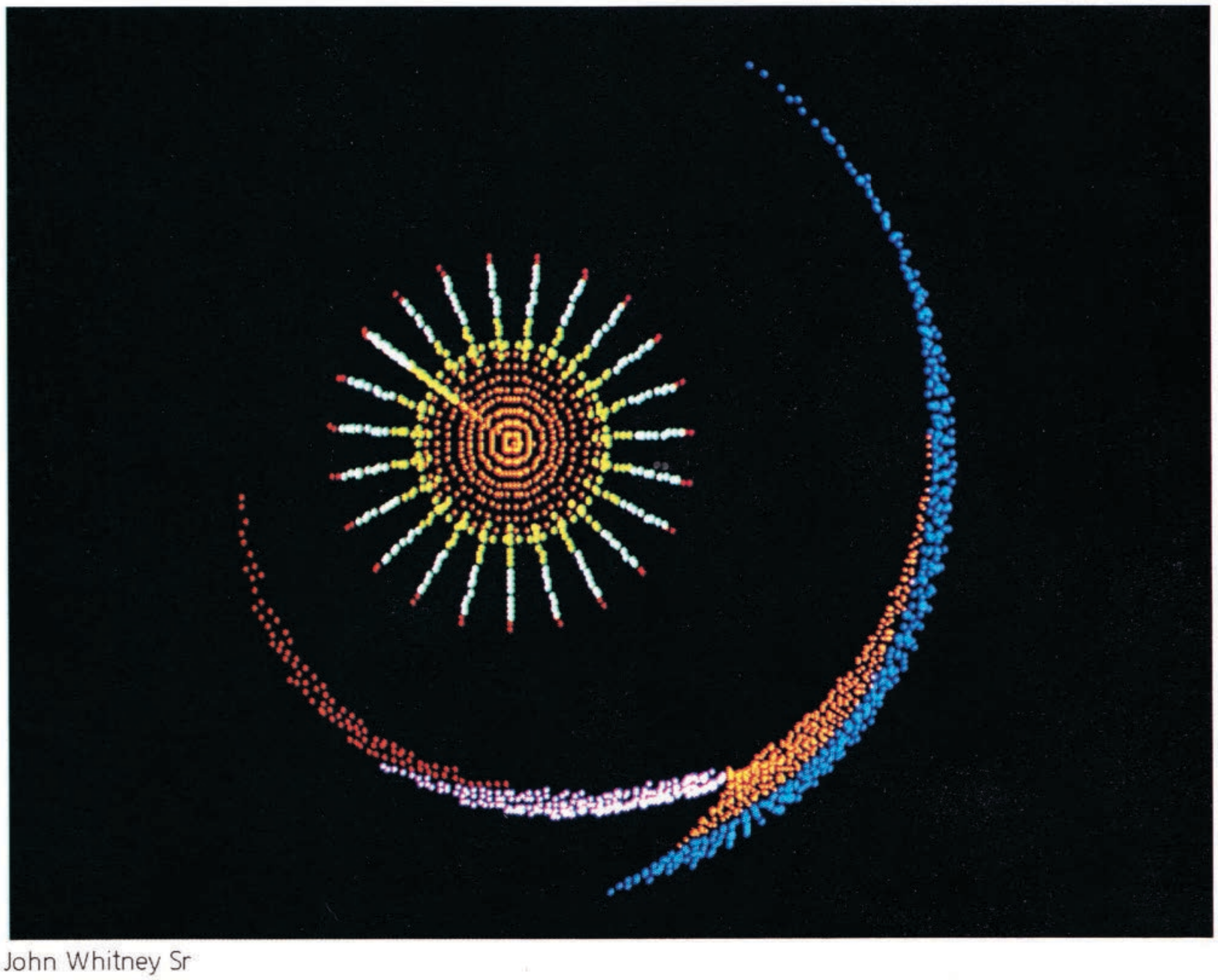


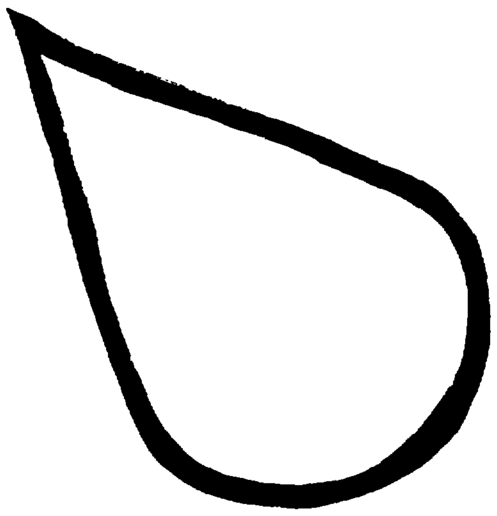
Karen Donoghue & Ken Knowlton
fig 2- The set of all basic kanji strokes.


John Whitney Sr

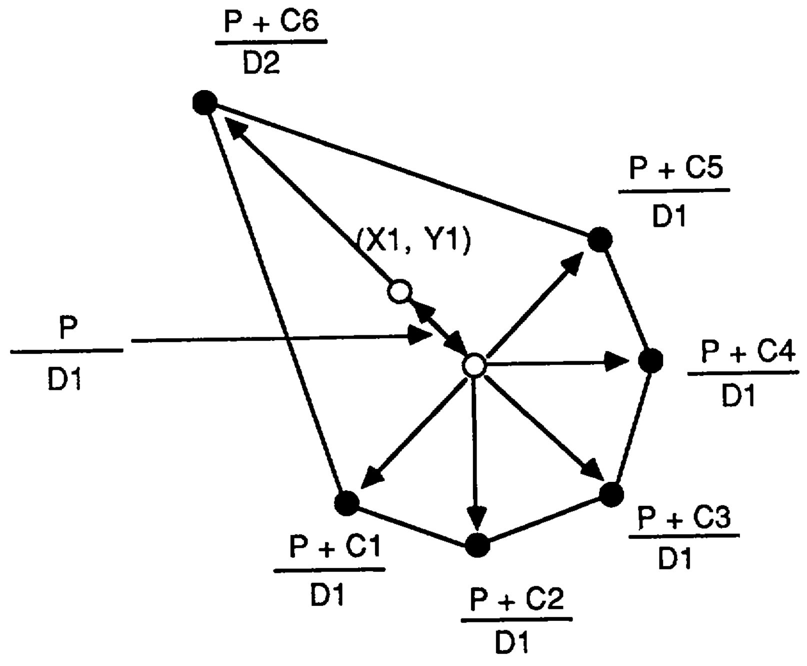

John Whitney Sr


John Whitney Sr

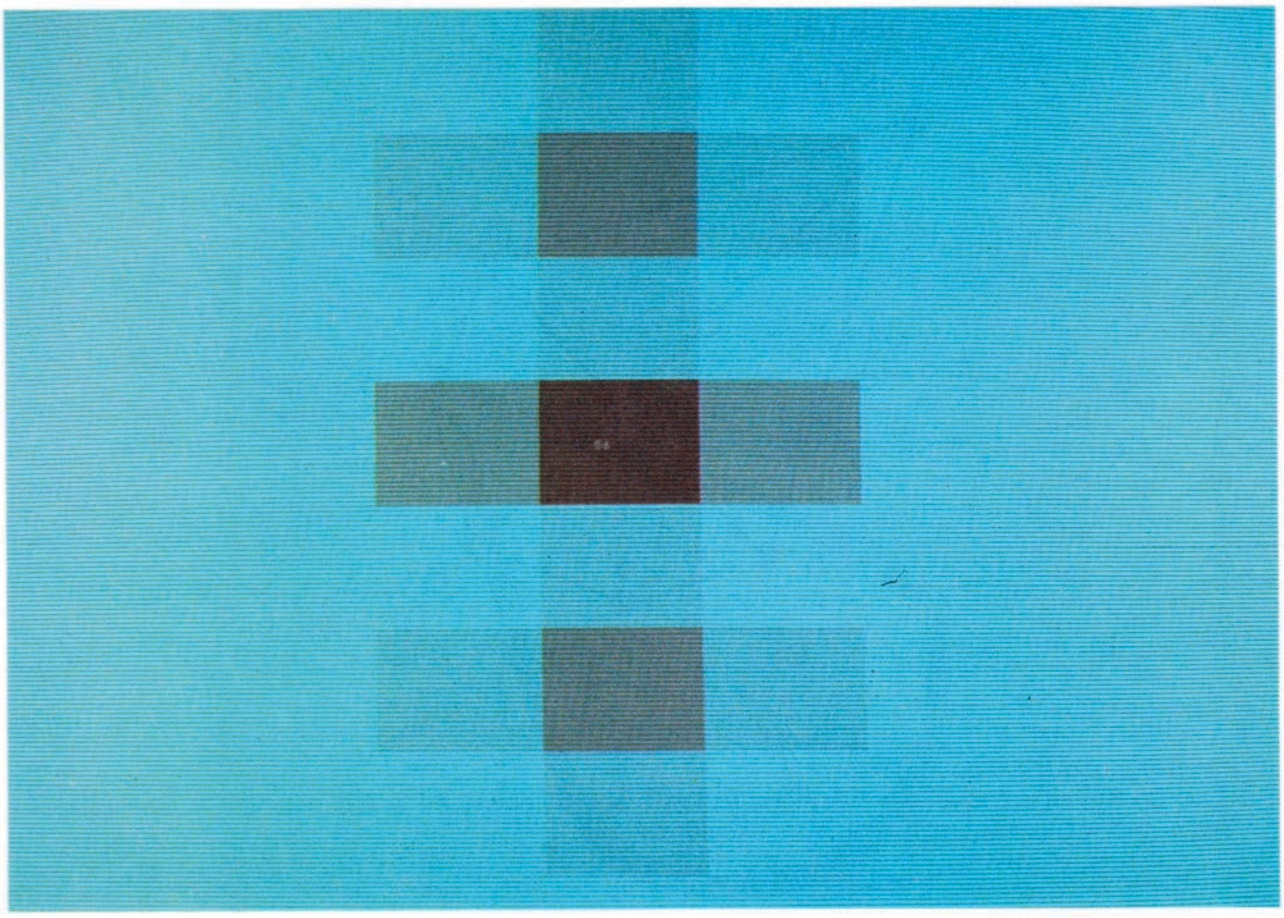

John Whitney Sr



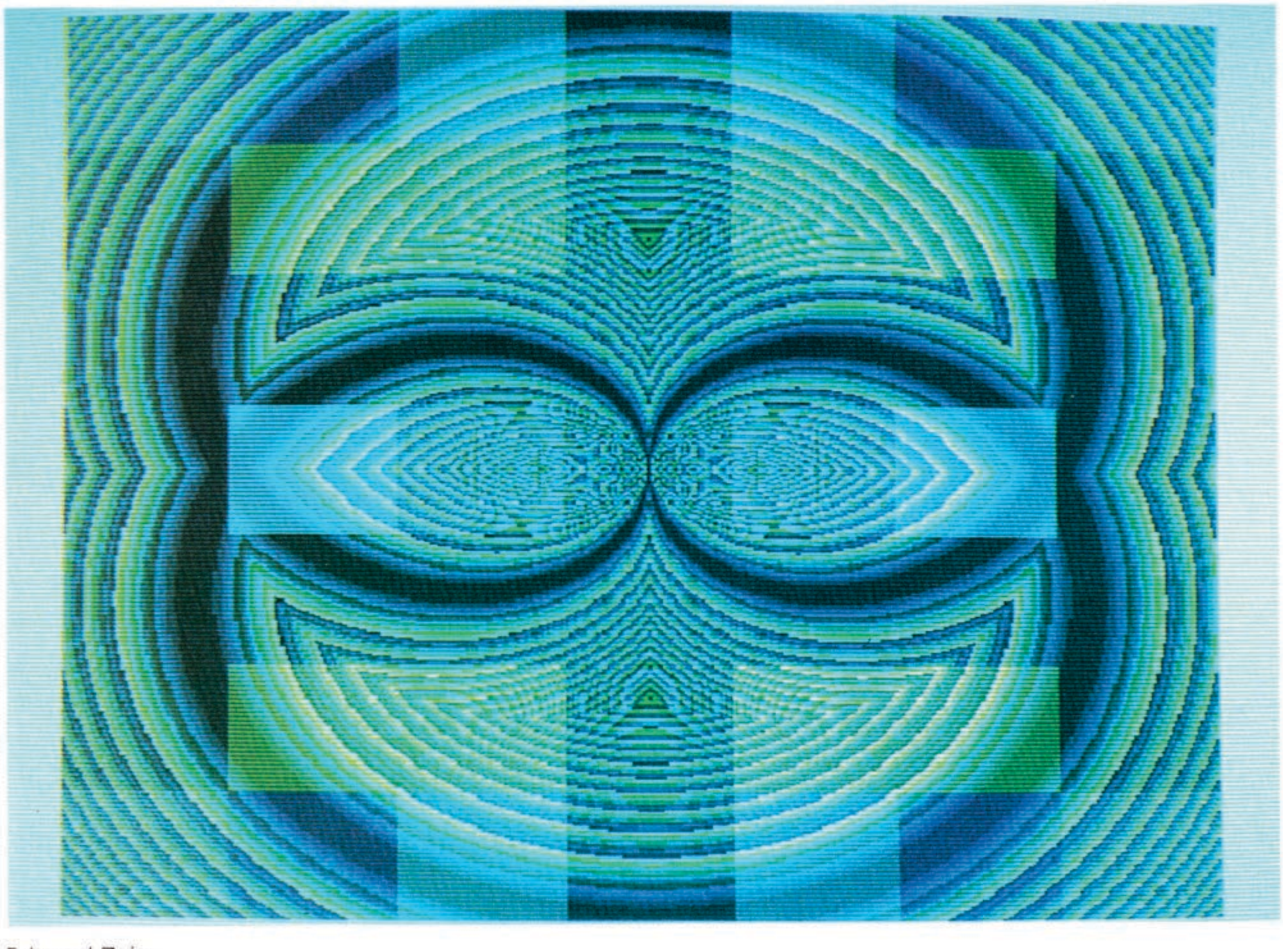
Karen Donoghue & Ken Knowlton
fig 3- The idealized Brush Pen teardrop primitive.



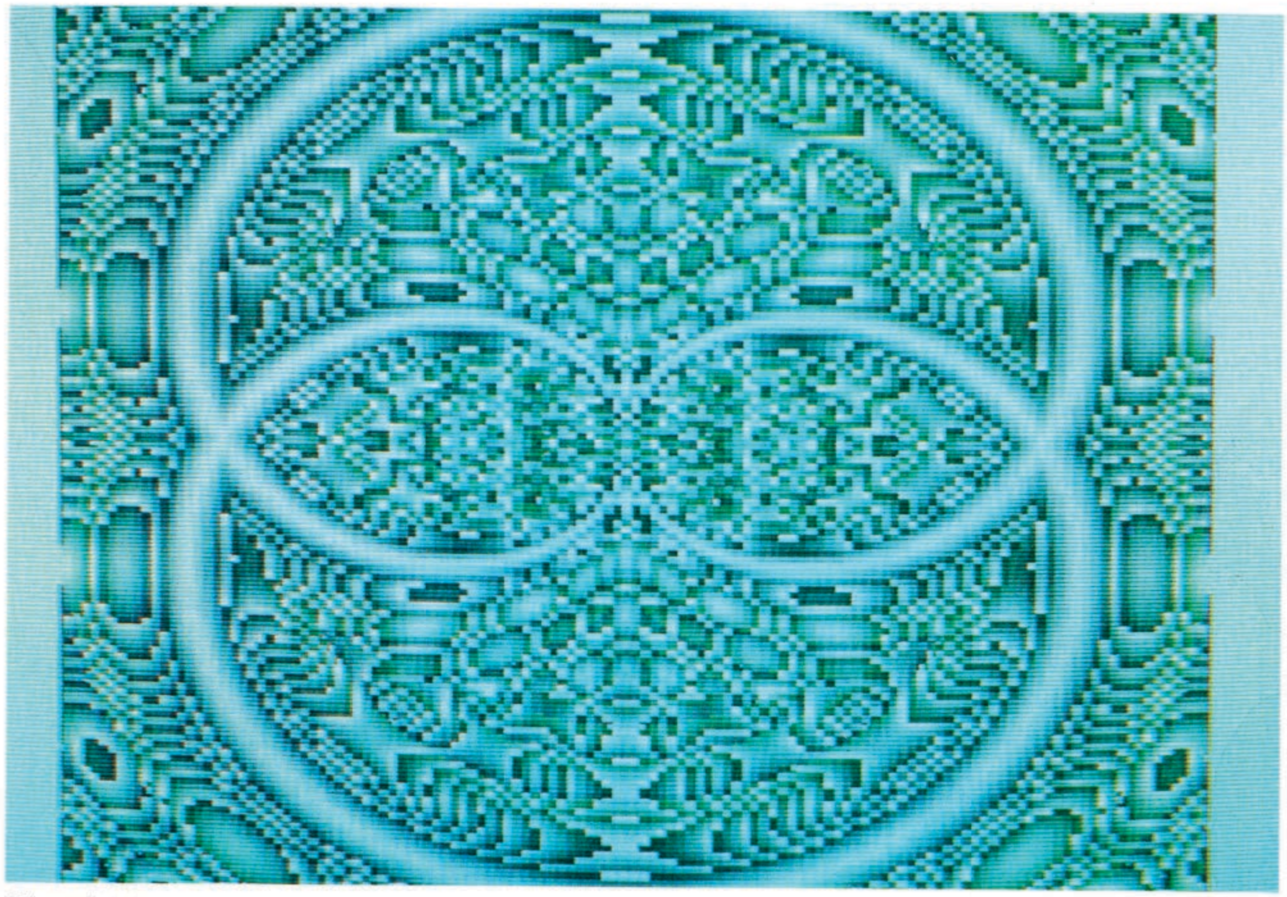
Karen Donoghue & Ken Knowlton
fig 4- Brush Pen primitive with weighting factors based on pressure P and constants Cn, Dn. Note that each Cn is distinct, and D1 is a multiple of D2.



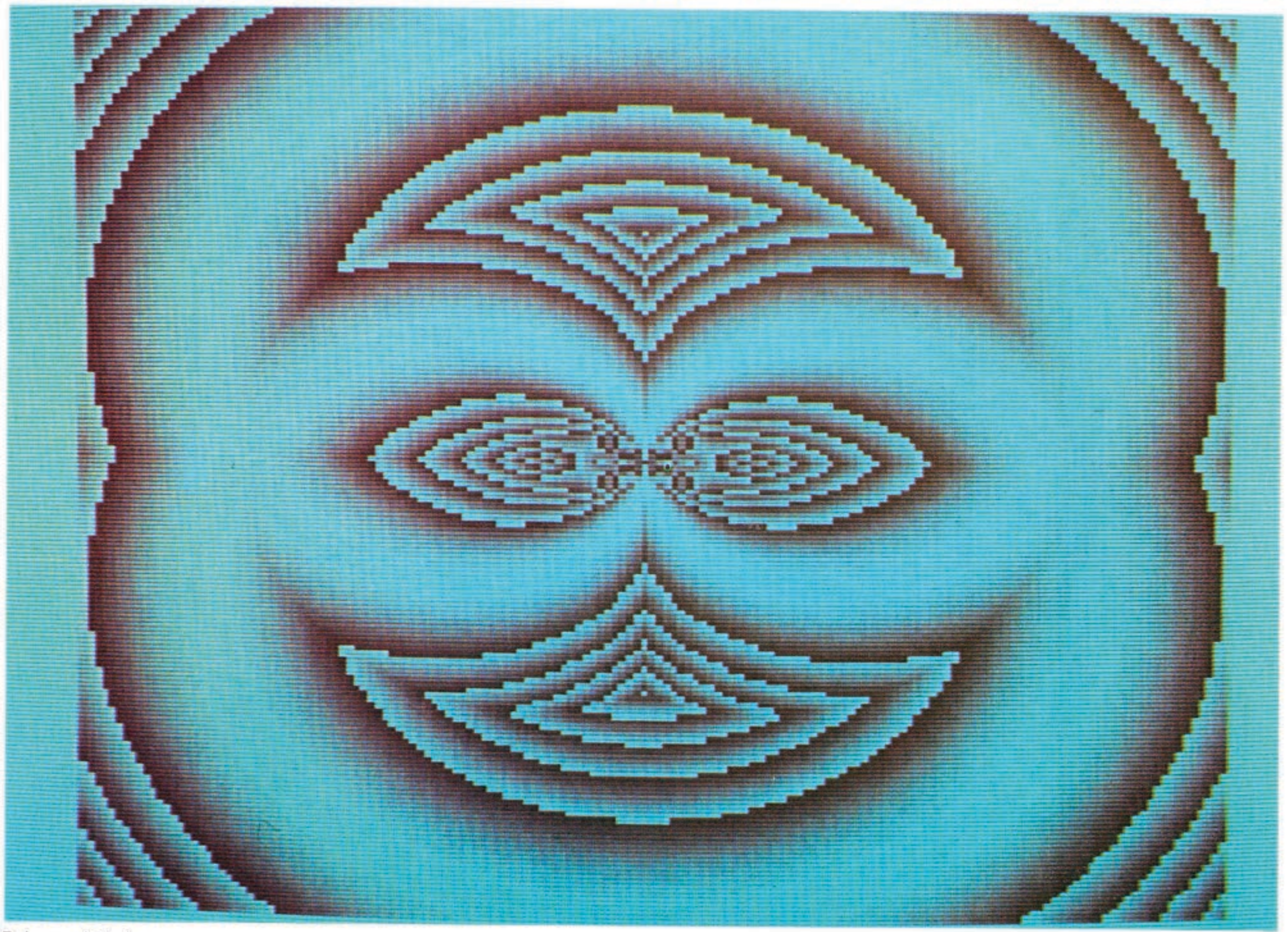
Edward Zajec
fig 1a The curve at scale 8x8 the components are large, colour retains a high degree of saturation



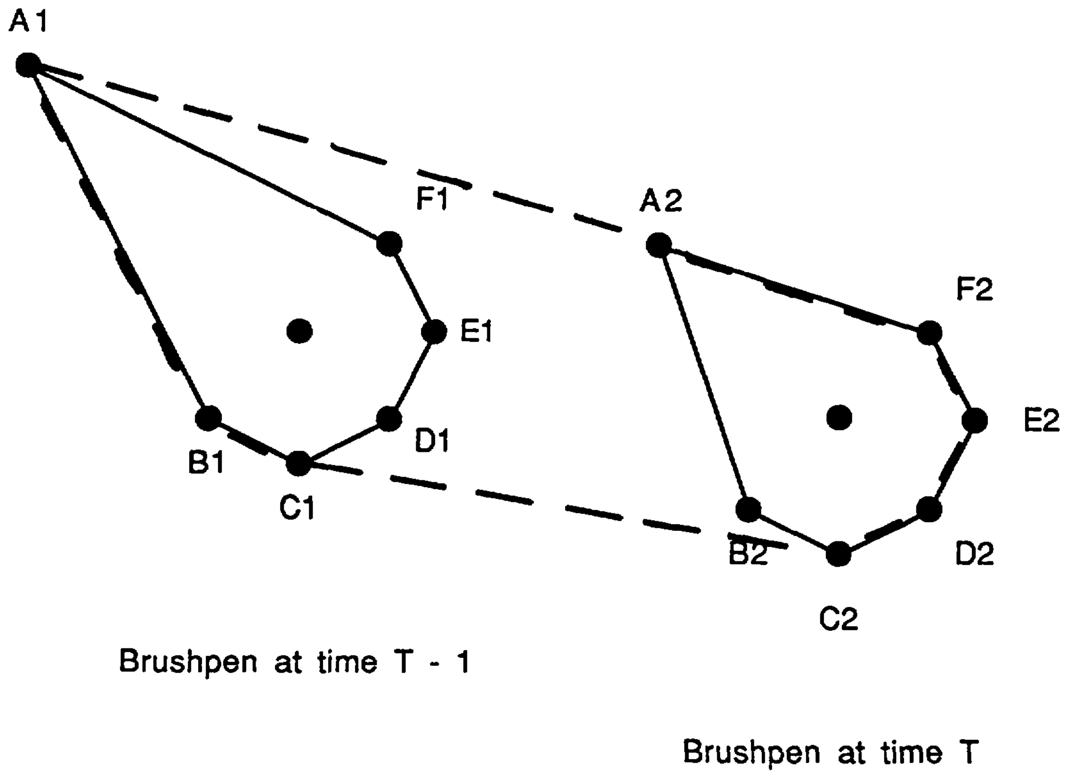
Edward Zajec
fig 1b The curve at scale 512x512. The tonal order is not progressively graded showing a rather shallow green texture.



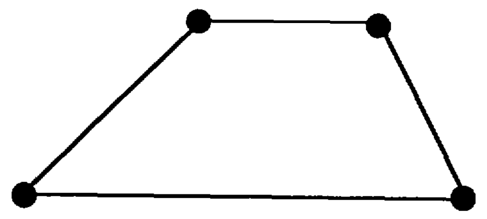
Edward Zajec
fig 1c The curve at scale 512x512. The progressive tonal gradation gives an illusion of volume, even though the curve itself is linear.



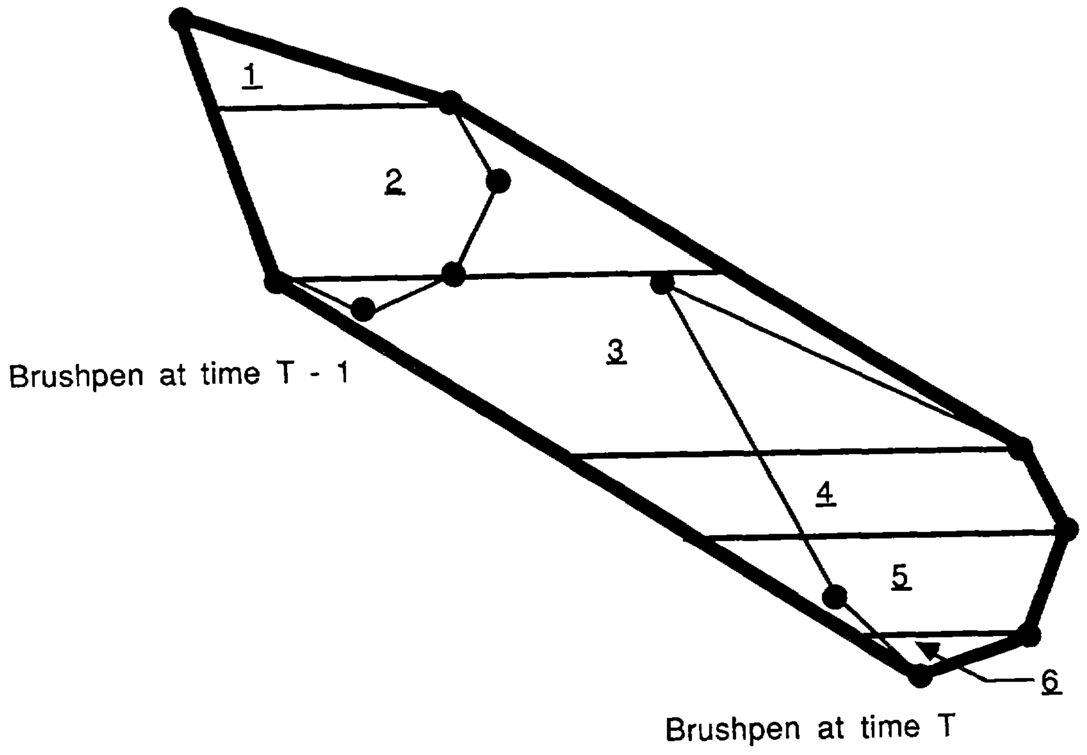
Edward Zajec
fig 1d An overlay of two transparancies (scale 4x4 and 128x128), simultaneously evidences thematic continuity as given by the curve at different scales, and colour depth in the modulation from green to red.



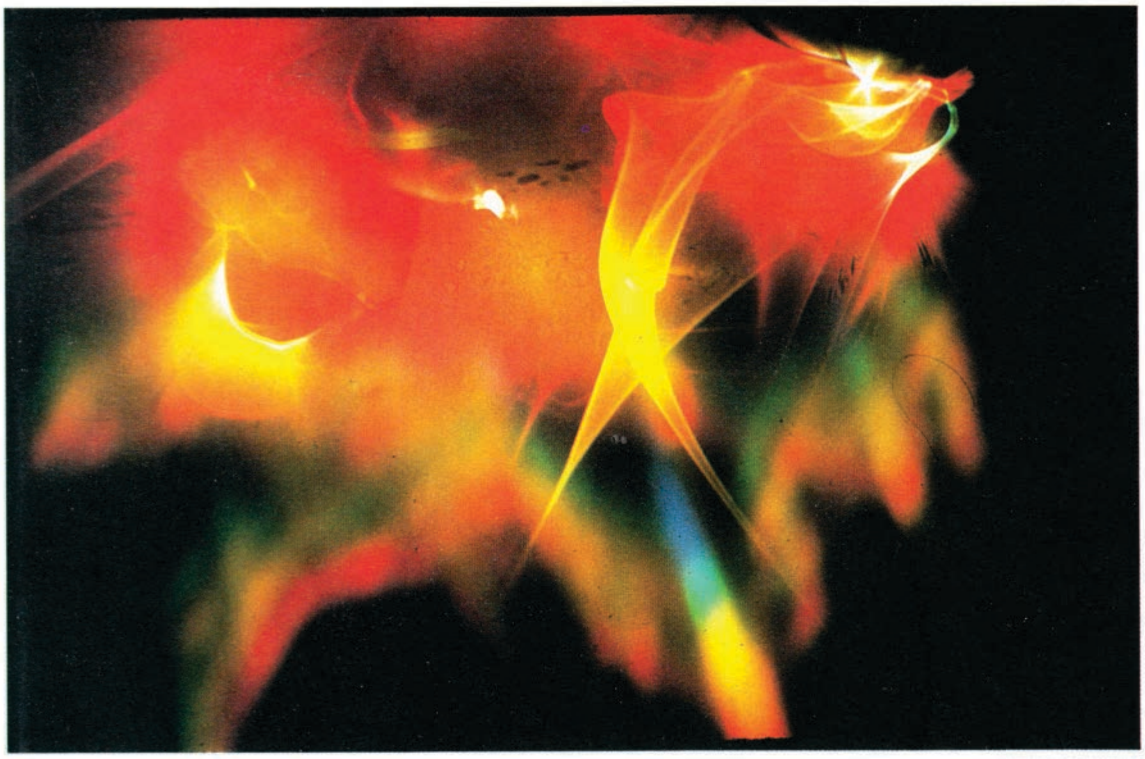
Karen Donoghue & Ken Knowlton
fig 5- The dotted line depicts the convex hull formed by Brush Pen primitives at time (T-1) and time T.

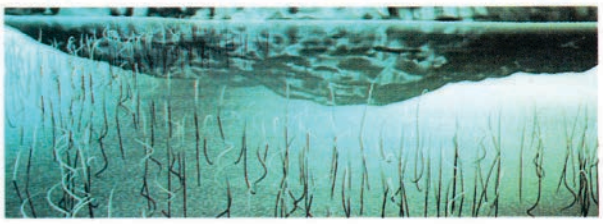


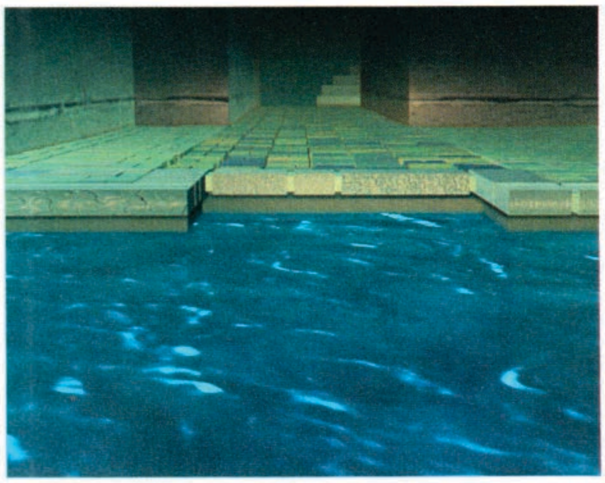
Karen Donoghue & Ken Knowlton
fig 6a- A typical trapezoid consisting of two X values at each of two Y levels.

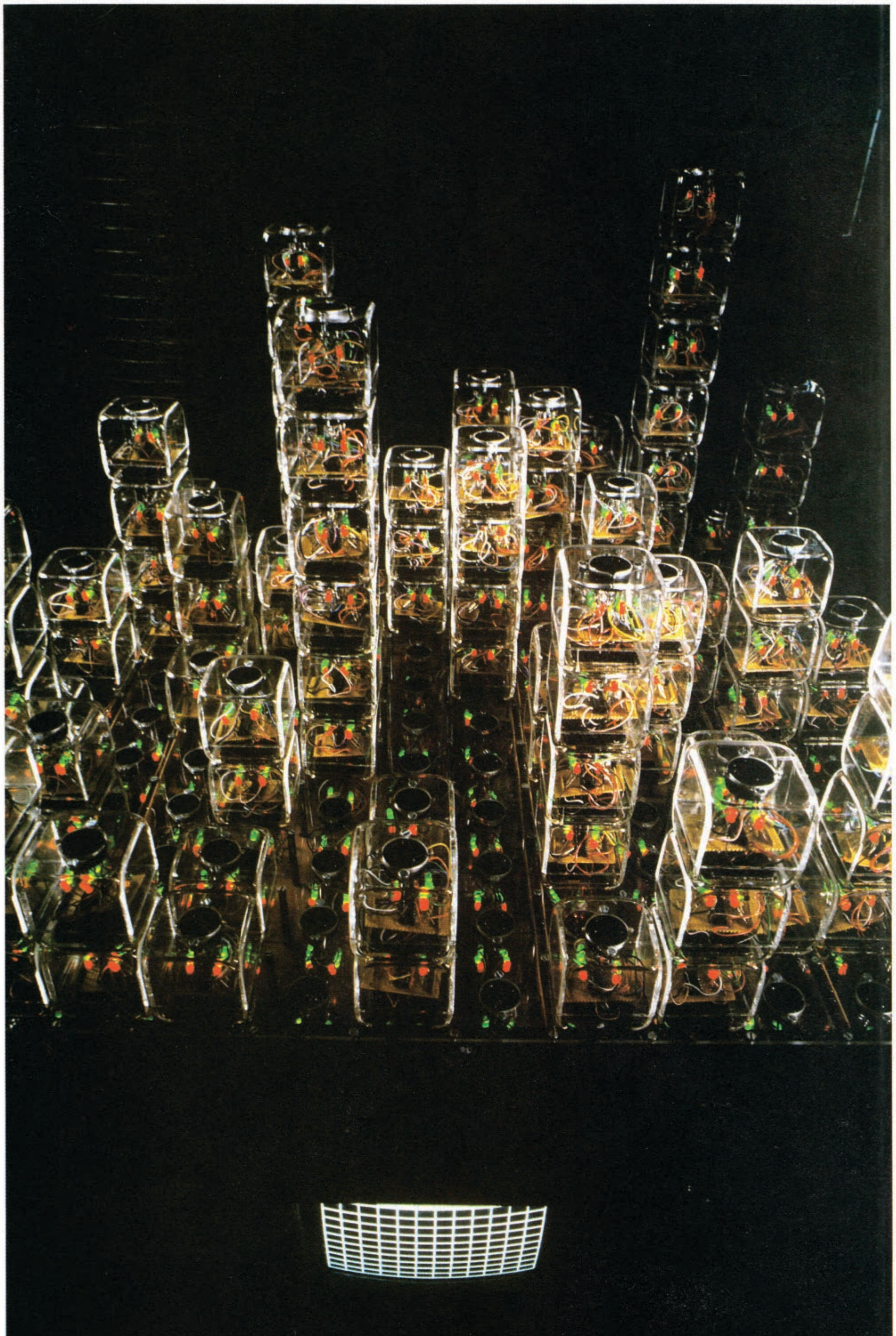


Karen Donoghue & Ken Knowlton
fig 6b- Brush Pen primitives at time (T-1) and time T, showing convex hull and corresponding decomposition to trapezoids, numbered here in order of filling.

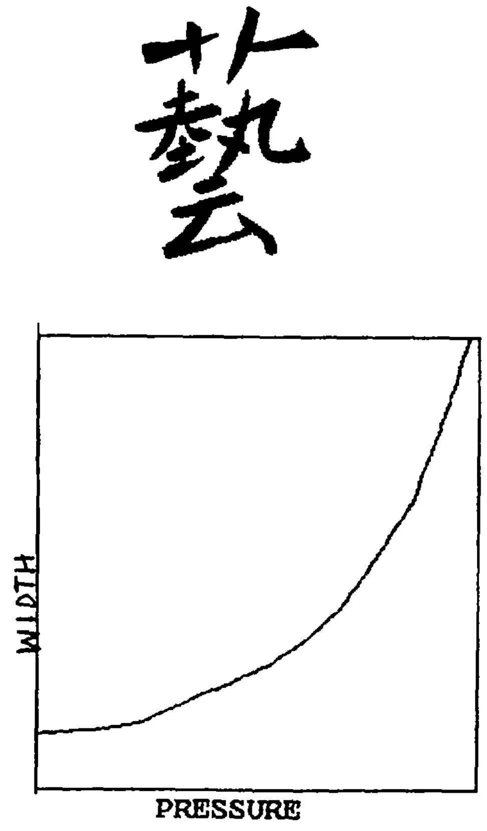

P K Hoenich
fig 1


Delle Maxwell
Sketches of Venice

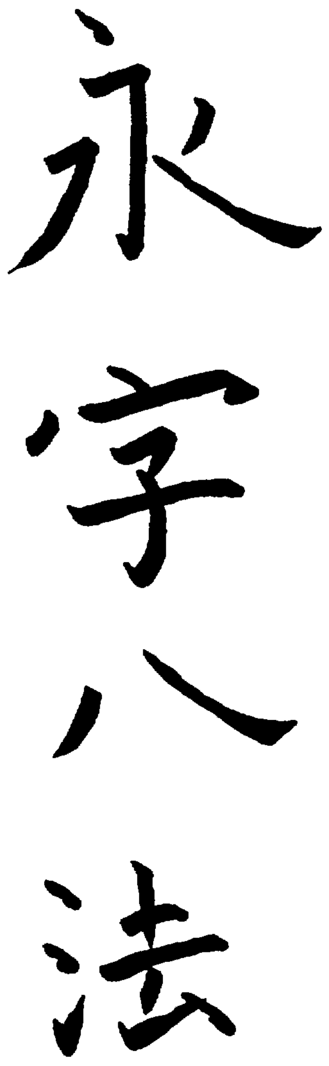

Delle Maxwell
Reflection



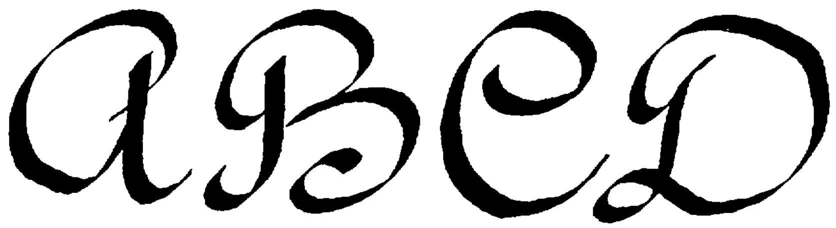
J H Frazer
UNIVERSAL CONSTRUCTOR photograph by Geoff Beekman, Building Design

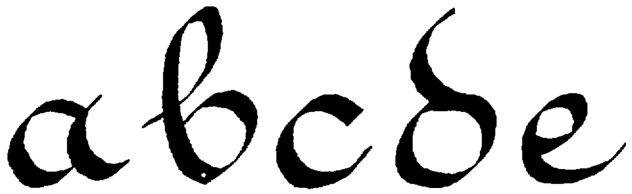


Karen Donoghue & Ken Knowlton
fig 7-Example of screen showing user interface with pop-up window for redifining Brush Pen pressure transfer function.



Karen Donoghue & Ken Knowlton
fig 8- Example of Brush Pen hardcopy, printed on a 300 dots per inch laser printer.





Karen Donoghue & Ken Knowlton
fig 9- Examples of 300 dots per inch output from computer model of a rigid broad-edged pen common Western calligraphy.

Sally Pryor
"Thinking of Myself as a Computer"

character set cannot be easily and quickly entered through a keyboard.

A.) Introduction and Background:

Nowadays in China and Japan, writing for practical purposes is done using pencil or pen, though brush and ink calligraphy is still widely practiced as a separate art. Until modern times, the brush was the main writing implement for writing the Chinese language, which consists of pictographic and ideographic characters written in succession downwards in vertical lines arranged right to left. It is from Chinese that the Japanese imported their written language, and have added phonetic symbols called *kana* used in conjunction with the basic Chinese characters. The first Chinese characters were primitive pictographs (called *shokei moji*), though now several types of characters exist. One such character type is that of combined-meaning characters (*kai-i moji*), consisting of several sub-characters consolidated into one form. Another type is the form-and-sound character (*keisei moji*), comprised of two parts, one denoting pronunciation, the other denoting meaning. *Kashaku* characters are borrowed from the words they represent, and *tenchu* , which are characters that differ from the original ideas that they represented due to an extension of their meaning. These six principles of construction constitute the structure on which the Chinese (and the Japanese) style of calligraphy referred to as *kanji.* A *kanji* is comprised of three elements: form, sound, and meaning. Form is the most important element to us, as we try to model the graphical behavior of a brush pen using the Freestyle environment.

B.) Rules and Characteristics of Kanji:

Each kanji character is comprised of a number of strokes, where the ordering of the strokes and their form is strictly defined by the type of script to which the character belongs. Some characters are so complicated that they are comprised of more than 30 strokes. Wang (1) asserted that Chinese characters are artistically elegant and rich in imagery, and therefore possess semantic meaning that is directly depicted by appearance or syntax. Proper visual feedback is then important in creating kanji characters, as both aesthetic and semantic information is contained in the strokes of the character.

To create a kanji characters, the brush pen is usually held perpendicular to the paper, with the writing hand sometimes placed atop the other hand for support and to keep the writing hand from smudging the ink. There is no specific type of brush used for a particular style of kanji, hence the brush used for kaisho (a popular block script dating from the 4th century A.D) can also be used to create characters in the gyosho (or semi-cursive) script. Instructional books on *kanji* (2) recommend that the calligrapher keep vertical strokes absolutely vertical, and that all horizontal strokes be kept at the same slope during the creation of a single character. At the beginning of a stroke, more force applied to the brushpen's bristles should result in their forming an angle of 135 degrees from the right hand horizontal in a Cartesian coordinate system (3). During lift, three basic techniques exist. These are a.) linear lift, b.) lift at 135 degree angle, and c.) lift and sweep backwards. These stroke styles are depicted in Figure 1. Each stroke is created using slow and deliberate motion, holding the pen rigidly at the same angle throughout the creation of the stroke. There is no turning or twisting of the brushpen between the fingers, as it common in Western calligraphy as a technique used to vary stroke width. In kanji, stroke width is a function of stroke direction and force, where an increase in force may result in more bristles coming in contact with the paper, or a spreading of the bristles so that they cover a larger area of the paper at a given moment. Direction can also be used to vary stroke width, as the angle of 135 degrees made by the bristles means that diagonals downwards to the right (and closely related strokes) are somewhat thinner than strokes in other directions. The set of basic strokes used to create all possible kanji characters is small, as depicted in Figure 2. However, these are consolidated together to create thousands of possible characters.

One of the problems of data processing in Asian cultures is the difficulty in entering data through the keyboard. A single character is broken down into its constituent strokes, and each stroke corresponds to a keystroke on a typical keyboard. Depending on the complexity of the character, there can be few or many keystrokes needed to enter a single character. Multiplying the complexity of an average kanji character by the thousands of characters in the language makes inputting data for a form, for example, an extremely difficult task for someone who is not an expert kanji typist. Kanji characters made with a ball point pen (or non force-sensitive writing implement) lack proper visual feedback. Leedham and Downton (4) showed that this affects writing dynamics and changes writing style, as well as the process of stroke creation. Using a ball point pen to create a syntactically correct, force-sensitive kanji stroke would require the writer to scribble over the same line to make it more distinct. Of course, force is not enough to provide perfectly realistic visual feedback, but helps in providing visual feedback closer to what the writer expects. Therefore, writing dynamics should be less distorted.

C.) The Computer Model:

1.) The Brushpen "Footprint":

Because the set of all possible kanji strokes is so small, it is fairly simple to characterize a kanji stroke based on three parameters: initial contact of the brushpen with the paper at the beginning of the stroke, stroke body, and stroke termination. We studied the technique of real (human) brushpen calligraphers to collect data on stroke creation and calligraphy technique. We determined the shape of the area formed by the brush bristles touching the paper to be a teardrop shape leaning diagonally to the left, symmetric about an axis that is 135 degrees from the right hand horizontal. Hence, we decided to use the teardrop shape as our basic drawing primitive, or brushpen "footprint". This footprint shape allows for the creation of all basic stroke forms, both narrow and wide (See Figure 3). The footprint size is related to the amount of force exerted on the pen, and allows the user to create pressure-determined variable width strokes. The Freestyle system is especially useful for modeling the creation of kanji, since the stylus is force sensitive so that the strokes constituting a kanji character can be rendered using stroke width as a function of force.

The angle at which the brushpen handle is held is always constant, perpendicular to the paper, hence there is no need to sense any angle data in the computer model. The Freestyle stylus is physically the same basic shape and size as a typical brushpen, so strokes created using the computer model are created just as they are in the real world - using the same hand motions and muscles, and using force to vary the width of the strokes.

2.) Stroke Creation:

Most brushes in computer-based paint systems use a single, static drawing primitive repeatedly drawn as a function of the movement of the input device. In some systems such as FullPaint (from Ann Arbor Softworks, Inc.) and SuperPaint (copyright 1986, Bill Snyder) available for the Macintosh (registered trademark of Apple Computer, Inc.) as the input device (a mouse) moves along from point P1 to P2, the entire path joining the two points is filled with paint. The user cannot dynamically change path stroke easily during the creation of a stroke, except by going to the menu and choosing a sub-menu for brush shape and/or size. In other systems, the single primitive itself is drawn at each sample point, giving a non-continuous series of dots or filled circles, or whatever the shape of the drawing primitive happens to be. Paint systems that use a 2D mouse as their input device are not capable of expressing stroke as a function of force. MacCalligraphy (copyright 1986-87, Enzan-Hoshigumi Co., Ltd.) allows the user to create kanji brush strokes using a mouse, however thickness of stroke is a function not of force but of the length of time that the mouse button is depressed. Realistic computer modeling of an application like kanji requires an input device which realistically refelects the dynamics of the hand motions of the artist - with variations in force and speed resulting in realistic feedback. Strassman (5) modeled a non real-time brush as a compound object composed of bristles, keeping track of the physical state of the brush and remaining amount of ink over the course of the stroke. Strassman's "stroke" is described as a path whose constituting control points each have a force value associated with them. The force value for each control point is input by the user through the keyboard. Ware and Baxter (6) described a paint program using a 6 degree of freedom "flying mouse" input device measuring x, y, z, and three variables of twist, allowing the artist to control form and color by moving the mouse in 3D space.

We chose to create a brushpen that combines some ideas of the above mentioned methods into a real-time brush whose resulting stroke is sensitive to dynamic force changes made by the user. As the stylus moves, the path traversed by the stylus determines the placement of the stroke on the screen, and the force exerted on the stylus determines the stroke shape and width.

For current use of the system as a brush pen to do Asian calligraphy, our method is good enough to create very realistic kanji strokes, according to our expert kanji calligraphers. One of our users, an expert painter, has also commented that the visual feedback from the system is excellent, and could easily be extended to model any type of paint brush, including brushes of different shapes and used with paper of different textures. This is an area for future research.

We have developed a fast method for creating a real-time kanji stroke using the Freestyle stylus. We approximate the convex hull of the teardrop drawing primitive using six control points. Six was determined, based on user feedback, to be the minimum number of points for an acceptable approximation for the teardrop shape. The arrangement of the six points is determined by adding a user-definable constant "C" to the raw force data "P" of the current stylus position at time T-1. Pressure data is in the range from 0 to 500 grams of force. After adding the constant "C" to the current force "P", the resulting value is divided by a non-zero user-definable constant "D". This result is then used as a "weighting factor" used to calculate the coordinates of each of the six control points in relation ot the current position of the pen at (X,Y). This (X,Y) data is obtained from the current tablet data packet. The points of the convex hull are calculated using (X,Y) at the very center of the brushpen footprint, to allow an even weight change in the stroke as force is changed linearly. Each control point has its own distince "C" value independent of all the other "C"s used in calculating the other control points. Each "C" is distinct so that, with slowly increasing force, the radial spokes of Figure 4, when pixel-quantized, do not increment simultaneously. (See Figure 4). Unlike the charcoal sketch program by Bleser et al (7), which uses pressure as an index into an array of predefined brush shapes, our method allows the convex hull of the drawing primitive to increase and decrease dynamically in a staggered manner, allowing for smooth gradations as force changes.

After the convex hull of the current drawing primitive is determined, the points are sorted in increasing Y, duplicates are thrown out, and points with the same Y value are culled - of these we keep only the leftmost and, if it is distinct, the rightmost. The remaining points are then fed to an N point Polygon fill routine, which fills the current primitive with paint (or ink). As the user moves the stylus to a new position at time T, the new tablet data packet is determined, and the new brushpen primitive is derived using the current force value and the series of "C' and "D" constants as stated above. The convex hull formed by both brushpen primitives at times T-1 and T is determined. These points for T-1 and T are each sorted in the manner stated above, and the remaining sorted points from both primitives are then merge sorted and fed to an N point Polygon fill routine. As the stylus moves, the last brushpen primitive for position (X,Y) at time T is saved and used in conjunction with the current brushpen primitive (based on most recently received tablet data) to determine the current stroke. (See Figure 5)

The filling of a stroke section between data packets happens within 5 milliseconds (running on a WANG 286), which is the time between tablet points at the fastest data rate possible. Because kanji is done at a somewhat slower pace than normal handwriting, and because of the importance placed on stroke quality, 5 milliseconds is enough time to fill the current stroke path with ink. This level of temporal resolution (200 points per second) gives very realistic kanji stroke feedback and is within the acceptable level for handwriting at normal speeds as described by Phillips (8). The current software allows the user to slow the tablet data rate as desired via the keyboard, and this may come in handy later when larger or more complicated shapes are used as footprints.

We developed a fast polygon fill algorithm for filling the convex hull of an N point polygon, which breaks convex hull of a stroke into its constituent trapezoids, each having horizontal top and bottom. (A trapezoid consists of 1 or 2 points on 1 or 2 possible Y levels. See Figure 6.) Each of these trapezoids is then fed to a trapezoid fill routine, which fills the trapezoid with horizontal lines, using Bresenham's line algorithm to determine the successive line endpoints. Figure 6B shows the convex hull of a brush stroke broken down into its constituent trapezoids.

The speed of the algorithm enables real-time strokes to be created with no jaggies, except when the pen is moved in a curve at very high speed. The code is currently implemented in Microsoft C running under Freestylus 1.0 on a high resolution (100 dpi) monitor. The only current colors avaiable are black and white. Data from the tablet is in 1000dpi format, so when data is mapped onto the monitor's screen at 100dpi, there is some quantization, though it is not objectionable. Non-real-time hard copy output from the same data can have higher resolution and even smoother edges.

3.) User Definable Parameters:

In addition to allowing the user to define the "C" and "D" constants that determine the relationship between raw force data and stroke width, our software allows the user to interactively define a force transfer function. This resulting value is substituted into the force-to-line-width equation as the raw force. These curves can be altered to create brushpens of different types, such as a brushpen whose stroke width decreases as more force is applied, and vice versa. (See Figure 7)

Most of the users like to experiment with their own force mapping curves, though variations on the exponential curve seem to be the most favored. Some people prefer more force sensitivity at low force levels, which others want as thick a stroke as possible over the entire range of force.

4.) Output:

Output for the brushpen system is now obtained either by saving a 100dpi screen image, or by converting the stroke data to a Postscript readable file. Both methods are currently used with a WANG LCS15. The second, higher quality method, is obtained by taking the 1000dpi tablet data, running our postprocessing software with the data, and converting it to a Postscript file for printing at 300dpi. Samples of output are shown in Figure 8.

D.) Additional Applications:

Since the polygon fill can handle any number of points, we can vary the drawing primitive shape to be any convex polygon, such as a hexagon, or a square. It can even be the degenerate case of a straight line, thus simulating a rigid broad-edged pen common in Western calligraphy, whose nib components spread further apart as more force is applied to the pen, resulting in a wider stroke (See Figure 9). We can thus model many brushes of different types, including ones that do not even exist in the real world. This code could be integrated into a paint system which would allow the user to create his or her own brush, specifying its shape, size, and force sensitivity. The addition of color into the system would provide a palette for different paint colors, type, and textures.

The polygon fill code and its supporting sort routines are currently implemented in C, but to operate more quickly, they will eventually be converted to assembly language. This will allow larger strokes to be created quickly, with no aliasing effects.

Currently, we are adding some more features requested by our kanji experts. These include a "smearing" or "bleeding" function to allow the paint or ink to smear on the paper if the brushpen is held in the same position for some length of time (about 3+ seconds), primarily by increasing the size of the footprint as the brush dwells at one place without significant movement.

E.) Ackowledgements:

The authors would like to thank the many calligraphers that helped give constructive comments and suggestions to the project, especially Kenichi Akagi, Horace Tsiang, Steven Teng, and John Cataldo. Also thanks to Steve Levine, Jonathan Reduker, Maria Witt, and Jean Renard Ward.

References:

(1) Wang, P.S. "Knowledge Pattern Representation of Chinese Characters". International Journal of Pattern Recognition and Artificial Intelligence, 2, No. 1, 161 - 179 (1988).

(2) O'Neill, P.G., "Essential Kanji" (New York/Tokyo: Weatherhill Inc, 1976)

(3) Nakata, Yujiro, "The Art of Japanese Calligraphy", (New York/Tokyo: Weatherhill/Heibonsha, 1973)

(4) Leedham, C. G. & Downton, A. C., "On-line Recognition of Pitman's handwritten shorthand - an evaluation of potential", International Journal of Man - Machine Studies, 24, 375 - 393.

(5) Strassman, Steve, "Hairy Brushes", SIGGRAPH Proceedings 1986, 20, 4, 225 - 232 (1986).

(6) Ware, Colin and Baxter, Curtis, SIGCHI Proceedings 1989, 155 - 160 (1989).

(7) Bleser, T. W., Silbert, J.L., and McGee, J. P., "Charcoal Sketching: Returning Control to the Artist", ACM Transaction on Graphics, 7, 1, 76 - 81 (1988).

(8) Phillips, M.J. , "Several Simple Tests Can Help You Choose the Correct Digitizer". Computer Technology Review, 7, No. 1 (January 1987)