Abstract

Most illustration systems need a lot of user strokes to generate natural-looking pen-and-ink illustrations. In order to reduce the number of user strokes necessary, we propose a new method for pen-and-ink illustrations using a stroke morphing concept. For this, we introduce a general stroke morphing procedure, which consists of both flow-oriented morphing and shape-oriented morphing. Using this morphing technique, we can make more natural-looking pen-and-ink illustration with fewer user strokes. This work can be applied to generate simplified pictures for dictionary typesetting. The main purpose of this paper is to describe this method, which requires fewer user strokes than other previous methods. Experimental results are given in the final section.

Keywords: Illustration, Stroke Morphing

1. Introduction

Scientific, technical, or medical documents have many illustrations, sketches or other line-oriented drawings. Though traditional printing techniques have been developed increasingly in our century, illustrated pictures are also used since that abstract drawings can convey complex information better than realistic images[2]. Abstract drawings like pen-and-ink illustrations offer numerous advantages over photorealism and they include their ability to omit extraneous details, to focus attention on relevant features, to clarify and simplify shapes, or to expose parts that are hidden[9].

1.1 Previous works

In recent years, a number of systems which automatically generate pen-and-ink illustrations have been developed. These systems can be classified into two broad, input-based categories: geometry-based systems, which take 3D descriptions as input; and image-based systems, which produce illustrations from greyscale images[7]. Geometry-based systems have been studied to render silhouettes of 3D objects[4]. In one such study, strokes were drawn on silhouette images by texture mapping[8], while another study used intersection lines[2]. However, it is hard to extract the boundaries and features of surfaces completely using image-based methods. As a result, most recent researchers have chosen methods whereby strokes are specified through numerous of user interactions, in image-based systems.[6, 7, 9]

A number of previous researchers have built their systems to generate illustrations which have been beautiful, artistic and similar to natural objects, therefore they have had to use many strokes per illustration. However, if you inspect dictionaries and textbooks, you can see that many practical illustrations are somewhat different from illustrations generated using other previous systems. For clarity, dictionary illustrations express only important characteristics of objects. As a result, many features of an object are simplified or omitted and fewer strokes are used. Fig.1 shows two types of illustrations. Fig.1(a) shows an illustration by an automatic drawing system[6] and Fig.1(b) shows a hand-drawn illustration printed in a dictionary. In this paper, we propose a new method for generating illustration strokes by applying a stroke morphing technique, which is a kind of field morphing.

1.2 System overview

The procedure of our system consists of 3 steps: boundary extraction, input of user strokes, and generation of artificial strokes. In the first step, the boundaries of objects are extracted and smoothed by using the B/zier curve. User strokes are then drawn by the user to generate artificial strokes. In the final step, we use stroke morphing techniques to generate artificial strokes. Each step is described in the following section in detail. Fig.2 shows the illustration generating procedure using our system. At first, an original 2D image is given, as in Fig.2(a). Then, our system extracts
Figure 1. Two types of illustration (a) computer-based illustration by a previous work (having numerous strokes) [6] (b) hand-drawn illustration printed on dictionary (having few strokes)

Figure 2. Illustration generating procedure: (a) an original image (b) boundary extraction (c) user strokes (d) illustration result

2. Boundary processing for illustration

In general, it is difficult to generate a good illustration if only a small number of strokes are used. To overcome this drawback we exploit the boundary of an object. This phase consists of two parts; boundary smoothing and image contour construction.

2.1. Boundary smoothing

The first step of the boundary extraction is to construct boundary points with a Laplacian filter and define the boundary using polylines [1, 3, 5]. At this stage, the boundary will be rough and complex. We then use the smoothing phase to refine the rough boundary. The smoothing phase removes redundant and noisy points in the polylines. After the smoothing phase, the boundary becomes more simple and the curves smoother. This process is illustrated by Fig. 3. Fig.3(a) and (b) show a raw boundary and its noisy points, which are determined by the area of three adjacent points (local triangle). Then, the points where the area of the local triangle is less than \( \theta \) are regarded as noise and eliminated. Fig.3(c) shows the example of redundant points, which have angles greater than \( \theta \). These points are also eliminated. Fig.3(d) shows the result of this process.

Figure 3. Boundary smoothing: (a) finding out noisy points and removing \( p_2 \), \( p_6 \), and \( p_8 \), (b) removing \( p_2 \) and \( p_7 \), (c) removing redundant point \( p_4 \), (d) result boundary

2.2. Image contour construction

Following boundary smoothing process, the simplified, shortened boundary-lines must be connected as a single closed curve. There are three steps, in this contour construction process: separation, merging, and the deletion of polylines.

If the difference between the inclinations of two adjacent line segments in a polyline is larger than a given threshold,
Figure 4. Two types of stroke morphing methods: (a)–(b) the flow-oriented stroke morphing procedure, (c)–(d) the shape-oriented stroke morphing procedure

we divide the polyline at the point that connects the adjacent line segments. On the other hand, if two distinct polylines are closer than a given threshold, we merge them in order to make the polylines into one. After the separation and merging, the image contour construction process removes noisy polylines that are shorter than the user-defined threshold.

3. Stroke morphing

Stroke morphing is a technique used to automatically generate artificial strokes based on given user strokes. We use two kinds of methods: flow-oriented stroke morphing and shape-oriented stroke morphing. Flow-oriented stroke morphing is suitable when strokes are intended to show flow or direction, like smoke or water. Shape-oriented stroke morphing, on the other hand, is suitable when strokes are influenced more by shape than flow, so this can be applied to drawing leaves or feathers.

Fig.4 shows the two different drawing stroke morphing methods, both with the same number of user strokes. Fig.4 (a)–(b) show the flow-oriented stroke morphing procedure, and Fig.4 (c)–(d) show the shape-oriented stroke morphing procedure. The morphing method employed by the user can be chosen based upon the characteristics of each illustration image.

Fig.5 shows a notation for the structure of a user stroke: control points, angles, length of composing vector, direction vectors, and a base vector shown as a dotted line. Let \( s_i \) be the \( i \)-th user stroke. The \( s_{i,j} \) denotes the \( j \)-th control point of the user stroke \( s_i \). The \( \text{angle}(s_{i,k}) \) in this figure denotes \( \angle s_{i,k-1}s_{i,k}s_{i,k+1} \) of \( s_i \). The \( \text{length}(s_{i,k}) \) denotes the edge length of \( (s_{i,k}, s_{i,k+1}) \), and the \( \vec{v}_{i,j} \) means a \( j \)-th direction vector of \( s_i \). The base vector is defined as \( b_i = s_{i,m} - s_{i,1} \), where \( s_{i,m} \) is the last control point of \( s_i \).

Figure 5. The structure of the user stroke for morphing

3.1. User stroke

In the image-based illustration system, little information is supplied, so user strokes are used to compensate for insufficiencies.

A user provides user strokes by drawing polylines directly on an image with a mouse or a stylus pen. The direction and number of user strokes must be carefully determined, since three parameters (direction of strokes, their direction, and their length) are crucial to the quality of the final illustration. Then, user strokes are simplified and smoothed, to eliminate noisy and redundant control points.

Our system accepts two types of user stroke: the shape user stroke and the direction user stroke. Fig.7 shows how this system generates artificial strokes when the shape user strokes (a) and the direction user strokes (b) are given. For some objects, the direction user strokes do not need to be given. Fig.7 shows two different cases: in Fig.7(a), only shape user strokes are used, and in Fig.7(b), both shape and direction user strokes are used.

Figure 6. User strokes and artificial strokes: (a) \( s_s \): user stroke for the shape (b) \( s_d \): user stroke for the processing direction (c) artificial strokes = \( s_s \oplus s_d \)
3.2. Flow morphing

In many cases, the direction of user strokes is important to generate realistic illustrations. The flow morphing method preserves the direction of user strokes during the stroke morphing phase. For this, each user stroke is divided into several direction vectors. All direction vectors have the same length. Let $s_{new}$ denote an artificial stroke to be generated by flow morphing technique. A $p(x,y)$ denotes the position where stroke $s_{new}$ is started at. Then we can determine, $n(s_{new})$, the number of direction vectors of the line segment of polyline $s_{new}$ as follows:

$$n(s_{new}) = \frac{\sum_{i=1}^{n} (\text{dist}(s_{i,1}, p)^{-1} \cdot n(s_{i}))}{\sum_{i=1}^{n} \text{dist}(s_{i,1}, p)^{-1}}$$

2

,where $\text{dist}(s_{i,1}, p)$ denotes the distance between points $s_{i,1}$ and $p$, $n(s_{i})$ denotes the number of direction vectors of a user stroke $s_{i}$ and $n$ denotes the number of user strokes.

After determining $n(s_{new})$, we must generate each direction vector of the stroke $s_{new}$. The $k$-th direction vector of $s_{new}$, $\vec{v}_{new, k}$, is generated by following equation.

$$\vec{v}_{new, k} = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n(s_{i})} (\text{dist}(s_{i,j}, p)^{-1} \cdot \vec{v}_{i,j})}{\sum_{i=1}^{n} \sum_{j=1}^{n(s_{i})} \text{dist}(s_{i,j}, p)^{-1}}$$

3

,where $\text{dist}(s_{i,j}, p)$ denotes the distance between points $s_{i,j}$ and $p$, $n(s_{i})$ denotes the number of direction vectors of the user stroke $s_{i}$ and $n$ denotes the number of user strokes.

Through this equation, we can generate the direction vector of a new artificial stroke which started from a point $p$. After generating the direction vector, the point $p$ is modified to $p + \vec{v}_{new, k}$ and the generating process of a direction vector is repeated until $n(s_{new})$ vectors are generated. Then the same process is repeated for the next starting point $(x', y')$.

Fig. 8 shows an artificial stroke $s_{new}$, and many notations used in equations. The $s_{new}$ is generated by the flow morphing technique with two user strokes, $s_{1}$, $s_{1+1}$.

3.3. Shape morphing

The aim of shape morphing is to best preserve the shape of user strokes. For this morphing technique, each user stroke is divided into the same number of direction vectors. Next, the user generates the base vector, $\vec{b}_{new}$, of an artificial stroke $s_{new}$ which starts at a point $p$, as in the following equation. Note that $p$ is the starting point of a new artificial stroke.

$$\vec{b}_{new} = \frac{\sum_{i=1}^{n} (\text{dist}(s_{i,1}, p)^{-1} \cdot \vec{b}_{i})}{\sum_{i=1}^{n} \text{dist}(s_{i,1}, p)^{-1}}$$

4

,where $n$ is the number of user strokes, $\vec{b}_{i}$ denotes the base vector of a user stroke $s_{i}$.
After generating the base vector of $s_{new}$, the user must determine direction vectors of $s_{new}$. Next the angle and length of direction vectors in each user stroke need to be computed. The $k$-th direction vector, $v_{new,k}$, of $s_{new}$ is defined as in the following equation.

$$\bar{v}_{new,0} = b_{new}$$ (4)

$$\bar{v}_{new,k} = \text{length}(s_{new,k}) \cdot \text{rotate} (\text{angle}(s_{new,k}), v_{new,k-1})$$ (5)

$$\text{angle}(s_{new,k}) = \frac{\sum_{i=1}^{n} (\text{dist}(s_{i,1}, p)^{-1} \cdot \text{angle}(s_{i,k}))}{\sum_{i=1}^{n} \text{dist}(s_{i,1}, p)^{-1}}$$ (6)

$$\text{length}(s_{new,k}) = \frac{\sum_{i=1}^{n} (\text{dist}(s_{i,1}, p)^{-1} \cdot \text{length}(s_{i,k}))}{\sum_{i=1}^{n} \text{dist}(s_{i,1}, p)^{-1}}$$ (7)

In the previous equation, $\text{rotate}(\text{angle}(s_{new,k}), \bar{v}_{new,k-1})$ denotes the function which returns a normalized vector rotated by $\text{angle}(s_{new,k})$ from $\bar{v}_{new,k-1}$ toward $b_{new}$. By repeating the above equations, artificial strokes generated by using shape-oriented morphing technique can be obtained.

Fig.10 shows notations used in the equations and an artificial stroke generated by the shape morphing method.

Fig.11 shows an illustration image generated by using the shape morphing method. Fig.11 shows that the shape of user strokes is preserved in the resulting illustration image.

3.4. Drawing artificial strokes

So far, it has been explained how to make artificial strokes (direction, number of piecewise line segment, etc.) from given user strokes. Since artificial strokes also have to show the characteristics of input images, image intensity must also be considered. In order to generate an illustration, only a pen with a fixed drawing width is used. To illustrate the dark side of an image, one must will generate many strokes, while reducing the number of artificial strokes in the bright side of an image.

For determining the stroke density of a subregion of an image, the number of artificial strokes can be controlled by adjusting the pixel intensity of the starting point $p$ of an artificial stroke. Let $I_{p}$ be the pixel intensity of a point $p$. In our experiments, this is an integer $k$, where $0 \leq k \leq 255$. $I_{p} = 255(=0)$ means a white space (=black) pixel. Let $BOX_{a}(p)$ be a "small enclosing box" with $d$ width and height at the center point $p$, and let $I_{BOX_{a}(p)}$ be the average pixel intensity of pixels in $BOX_{a}(p)$. It must be determine...
probabilistically if the point $p$ can be a starting point of a new artificial stroke or not. Let $Pr(p)$ be the probability of an artificial stroke starting at a point $(x, y)$. Then we can calculate $Pr(p)$ as follows:

$$Pr(p) = \frac{(255 - BOX_x(p))}{(t_1 + 255)}$$ if $BOX_x(p) < t_0$

$$Pr(p) = 0$$ if $BOX_x(p) \geq t_0$,

where $t_0$ and $t_1$ are control constants. In this experiment, $t_1 = 4$. So, if a pixel is surrounded with a perfect black box, then about 25% of such black pixels might be starting points of any artificial strokes. Selecting a good $t_0$, $t_1$ and $d$ of $BOX_x(p)$ is very important in generating a realistic illustration. Also we can generate many different illustrations by adjusting these control variables $t_0$, $t_1$, and $d$.

4. Experimental results and conclusion

This illustration system has been implemented on a Pentium III, 600MHz. All input images are 512x512 black and white images. The execution time depends on the density of the objects in the input image and the number of user strokes. In our experiments, the average execution time was about one minute. The final results were translate to the postscript format with a B'zier curve to make more refined, high resolution images.

Fig.12 shows result illustrations based on the number of user strokes. If there are too few user strokes, the system can not generate good results. However, it is of no use to give too many strokes, as Fig.12(c). It can be seen that results of Fig.12(b)($n_u=5$) and Fig.12(c)($n_u=15$) are not so different.

Fig.13 shows the result illustrations of two knot images and an example of dictionary typesetting. Fig.14 shows two typical illustrations produced using the stroke morphing method. Fig.14(a) is a given 2D image. In Fig.14(b), a human illustrator creates user strokes within each image. The final results are shown in Fig.14(c), where the wolf and the prairie dog illustrations are completed in postscript format. In case of the wolf image, 13 shape user strokes are given and 308 artificial strokes are generated in the result illustration image. For prairie dog, 17 shape user strokes are given and 421 artificial strokes are generated. Fig.15 shows the result when our illustration method is used for dictionary typesetting.

This paper presents a new illustration method that highlights the important features of original images. The improved features of our method can be summarized as follows:

- Illustrations can be created by specifying a small number of simple user strokes.
- The proposed technique enables users to easily control the illustration effects by adjusting the user strokes.
- This is the first illustration technique that utilizes stroke morphing.

References


knot  - 1.  A compact intersection of interlaced material, such as cord, ribbon, or rope.  
   a.  A fastening made by tying together lengths of material, such as rope.  
   b.  A decorative bow of ribbon, fabric, or braid.  
   c.  A tight cluster of persons or things: a knot of onlookers.  
   e.  A complex problem, hard place or lump, especially on a tree, at a point from which a stem or branch grows.  

(a) slip knot  
(b) overhand knot

Figure 13. Illustrations output: two kinds of knots

Figure 12. Results based on the number of user strokes: solid lines in left figures denote user strokes.  
(a) too few user strokes, (b) proper number of user strokes, (c) too many user strokes

Real-time nonphotorealistic rendering.  

Interactive segmentation with intelligent scissors.  

Interactive pen-and-ink illustration.  

Orientable textures for image-based pen-and-ink illustration.  

Rendering parametric surfaces in pen-and-ink.  

Computer-generated pen-and-ink illustration.  
Figure 14. The three steps of illustrations building: (a) 2D images (b) boundary extraction and user strokes (c) final results of the processing. \( n_a \) is the number of artificial strokes, \( n_u \) denotes the number of user strokes.

hand /ˈhænd/ 1. [C] either of the movable parts at the end of a person’s arm, including the fingers: She had a gun in her hand. [=She was holding a gun.] I’ve got a nasty cut on my left hand. I held the child by the hands (with each other). He left the child by the hand. 2. [C] a pointer or needle on a clock, machine or measuring instrument:

(a) Original image (second author’s left hand) (b) An example for dictionary illustration

Figure 15. a hand example for dictionary illustration