

中图分类号: TP391.4 文献标志码: A 文章编号: 1006-8961(2011)11-2074-07

论文索引信息: 梁波, 戴芳, 赵凤群. 边缘跟踪算法的图像线条画生成 [J]. 中国图象图形学报, 2011, 16(11): 2074-2080

边缘跟踪算法的图像线条画生成

梁波, 戴芳, 赵凤群

(西安理工大学理学院, 西安 710054)

摘要: 给出一种基于边缘跟踪算法的图像线条画生成算法。该算法由边缘跟踪和线条画绘制两部分组成。在边缘跟踪中, 提出基于相异性度量的边缘跟踪算法, 以使利用边缘检测算子得到的边缘能够归类、连接; 在线条画绘制过程中, 引入非均匀 B 样条对不连续边缘进行插值并使用高斯平滑以获得连续光滑的边缘线条, 然后以线条的曲率为依据, 生成画笔, 实现线条画的绘制。实验给出了本文方法生成的图像线条画的结果, 实验结果表明, 本文方法能够迅速快捷的生成较高质量的线条画。

关键词: 线条画; 边缘跟踪; 非均匀 B 样条插值; 高斯平滑; 相异性度量

Images line drawings generation by edge tracking algorithm

Liang Bo, Dai Fang, Zhao Fengqun

(School of Science, Xi'an University of Technology, Xi'an 710054 China)

Abstract: This paper presents a method which is based on edge tracking algorithm to generate line drawings from images. The algorithm consists of two parts: edge tracking and line drawing painting. For edge tracking, we propose an edge tracking algorithm based on dissimilarity measure, so that the edges that be from the edge detection operator are able to be classified and connected. In the line drawing painting process, we employ a non-uniform B-spline to interpolate for the discontinuous edge and the Gaussian function to obtain a continuous smooth lines. Then the brush is generated based on the curvature of the lines, and the line drawing of images is obtained. The experiment results of image line drawings, which is generated by our method, are demonstrated in this paper, and experiment results show that our method is able to quickly generate higher-quality line drawings.

Keywords: line drawings; edge tracking; non-uniform B-spline; Gaussian smoothing; dissimilarity

0 引言

图像线条画是一种图像的稀疏表示方法, 是使用几何线条将图像的主要特征结构关系勾勒出来并将其组织起来进行绘制的一种技术。研究图像线条画生成技术, 不仅为计算机视觉问题提供了解决的途径和可能, 而且为图像视频信息在计算机中的快

速解读和传输提供了可能的解决方案。

图像线条画是一个新的研究课题。目前图像线条画的研究分为两类: 一类是研究不同风格的线条画转换, 主要是研究线条画笔画的形状特征, 运用数学理论对笔画的形状进行模拟, 以得到不同风格的线条画^[1], 或通过对样本风格的学习, 将样本风格转换到目标图上^[2]。另一类是研究从图像中获得线条画, 主要是研究画家绘画的过程, 并使用计算机

收稿日期: 2009-10-29; 修回日期: 2011-01-18

基金项目: 国家自然科学基金项目(60971127); 陕西省教育厅基金项目(09JK611); 西安理工大学博士启动基金项目(108-210905); 西安理工大学科技创新计划基金项目(108-210811)。

第一作者简介: 梁波(1985—), 男, 西安理工大学计算数学专业硕士研究生, 主要研究方向为图像处理。

E-mail: yeliang2007@gmail.com。

进行模拟。由于自然场景图像内容千差万别,使得线条画的绘制成为一件非常困难的事情。人脸具有特定的几何结构,更有利于线条的获取,因此现有的图像线条画研究大都集中在人脸图像的线条画。起初 Rhodes^[3] 等人研究了如何识别人脸的特征并希望能够解开人脸识别的秘密。研究发现即便是少数几根线条组成的线条画仍然可以保留人脸的视觉特征。后来一些研究者开始尝试利用人机交互方式来生成人脸的线条画及漫画。如 Li 等人^[4] 提出的自动人脸线条画生成系统,主要讨论使用对称算子、矩形滤波器和特征轮廓来检测和定位人脸的特征点。但是这类方法无论是在操作上还是在所生成的效果上都不是理想;陈洪等人^[5-6] 提出“基于样本学习的线条画”和“基于样本学习的肖像画绘制”算法,将统计学习理论应用到图像线条画绘制中,获得了人脸肖像画,然而这类方法需要有较为专业的美工人员参与手工绘制训练样本,同时要在多种假设的前提下进行实验。Son 等人^[7] 突破人脸图像的限制,提出一个新颖的图像线条画生成机制,在该文中他们给出一个基于似然函数估计的线条提取算法,并有效地提取了真实图形边界,但是这类方法计算上较为复杂。为了降低计算的复杂度以及减少人工干预给出一种基于边缘跟踪算法的图像线条画生成方法。该算法由边缘跟踪和线条画绘制两部分组成,易于实现,可以通过调节部分系数,生成不同风格的线条,而且计算迅速。

1 边缘跟踪

边缘是指图像局部亮度变化最显著的部分,是图像亮度不连续的结果。作为重要的图像特征,提取边缘往往是图像线条画、图像分割、目标区域识别、区域形状提取等图像分析领域必要一步。边缘检测算子众多,相比而言,Canny 算子^[8] 是一种较为理想的边缘检测算子,它能在噪声抑制和边缘检测之间取得较好的平衡。

利用 Canny 算子检测得到的边缘图像是分布在图像区域中一系列的离散点。为了跟踪到较为光滑的边缘曲线,定义两边缘点间合适的相异性度量是必需的。文献 [9] 中给出基于共圆性 (cocircularity)、光滑性 (curve smoothness) 和接近度 (proximity) 的相异性度量定义。考虑到两边缘的梯度方向与两边缘连线夹角对相异度的影响,对上述

定义进行了修改,给出两边缘点间的相异性度量 D 的定义为

$$D_{ij} = 1 - S_{ij}^C S_{ij}^S S_{ij}^P \quad (1)$$

式中,共圆性度量 S_{ij}^C 的定义为

$$S_{ij}^C = [1 - (\sin \theta_{ij}' - \sin \theta_{ij}'')^2] \cdot \exp\left[-\frac{1}{\sigma_C^2}(\sin \theta_{ij}' - \sin \theta_{ij}'')^2\right] \quad (2)$$

光滑性度量 S_{ij}^S 的定义为

$$S_{ij}^S = (1 - \sin^2 \theta_{ij}') (1 - \sin^2 \theta_{ij}'') \exp\left[-\frac{1}{\sigma_S^2}(\sin \theta_{ij}' - \sin \theta_{ij}'')^2\right] \quad (3)$$

接近性度量 S_{ij}^P 的定义为

$$S_{ij}^P = \exp\left(-\frac{1}{\sigma_P^2}d_{ij}^2\right) \quad (4)$$

而 θ_{ij}' 为边缘点 O_i 的梯度方向与两边缘点连线的夹角, θ_{ij}'' 为边缘点 O_j 的梯度方向与两边缘点连线的夹角, d_{ij} 为边缘点 O_i 到 O_j 的距离, σ_C 、 σ_S 、 σ_P 为高斯方差,它们分别控制着共圆性 (cocircularity)、光滑性 (curve smoothness) 和接近度 (proximity) 3 个性质的衰减程度,它们的取值由实验经验得到。

可见 $D_{ij} \in [0, 1]$, 且 $D_{ij} = D_{ji}$ 。该相异性度量的优点是可以对分叉边缘点及噪声点处的边缘跟踪,便于在这些点处找到正确的跟踪方向,因此,该相异性度量具有一定的抗噪性。

按照式 (1) 的相异性度量定义,给出边缘跟踪算法,其步骤如下:

设原始图像为 I , 使用 Canny 算子进行边缘检测后的图像为 E , O_i 表示第 i 个边缘点 (其中 $i = 1, 2, \dots, M$ (M 为图像中边缘点的个数))。

1) 使用 O_{i1} 标记第 i 个边缘点是否被处理过 (如果第 i 个边缘点未被处理,则为 0, 否则为 1), 使用 O_{i2} 表示第 i 个边缘点是否为端点 (如果第 i 个边缘点不是端点,则记为 0, 否则记为 1, 端点是指在其 8 邻域内除了该点外只有一个边缘点)。

2) 在边缘点中搜寻 $O_{p1} = 0$ 且 $O_{p2} = 1$ 的边缘点 O_p , 如果未找到边缘点 O_p , 则转 5)。

3) 令 $O_{p1} = 1$, 将下标 p 存入 $newpath$ ($newpath$ 为当前曲线路径)。

4) 计算未被处理且在 N_p (O_p 的 8 邻域) 内图像边缘点的个数 N , 如果 $N = 1$, 则设 N_p 内的边缘点为 O_q , 令 $p := q$, 转步骤 3)。如果 $N > 1$, 则计算 N_p 内的边缘点与 O_p 的相异度, 并进行排序选择, 取与 O_p 相异度最小的边缘点, 设该点为 O_q , 令 $p := q$, 转

3);如果 $N = 0$, 则当前曲线路径 *newpath* 记录完成, 将 *newpath* 存入 *paths* 中 (*paths* 用于存储所有边缘跟踪路径), 清空 *newpath*, 转步骤 2)。

5) 搜寻边缘图像 E , 如果存在 $O_{q_l} = 0$ 的点, 则令 $p: = q$, 转步骤 3), 否则停止曲线跟踪。

6) 剔出长度小于某个预先给定的阈值 T 的曲线路径。

2 线条画绘制

在得到图像的边缘路径后, 采用非均匀 B 样条^[10], 以曲线上点到起始端点的弧长作为参数, 得到插值边缘曲线, 并利用高斯滤波器对曲线进行平滑处理, 以提高画笔生成的质量。

以插值后的一条边缘曲线为例, 设该边缘曲线为 $r(t) = [x(t), y(t)]$, 其中 t 为弧长参数, $x(t)$ 和 $y(t)$ 分别为 t 处的横纵坐标, 将 $r(t)$ 与高斯函数 $G(t, \sigma)$ 进行卷积得平滑后的曲线^[11]

$$R(t, \sigma) = [X(t, \sigma), Y(t, \sigma)] \quad (5)$$

对 $R(t, \sigma)$ 使用如下公式计算其曲率^[11]

$$k(t, \sigma) = \frac{|X_t(t, \sigma)Y_u(t, \sigma) - X_u(t, \sigma)Y_t(t, \sigma)|}{[X_t(t, \sigma)^2 + Y_t(t, \sigma)^2]^{\frac{3}{2}}} \quad (6)$$

式中, $X_t(t, \sigma)$ 和 $Y_t(t, \sigma)$ 分别表示 $X(t, \sigma)$ 和 $Y(t, \sigma)$ 对 t 的一阶偏导, $X_u(t, \sigma)$ 和 $Y_u(t, \sigma)$ 分别表示 $X(t, \sigma)$ 和 $Y(t, \sigma)$ 对 t 的二阶偏导。以步长 h 剖分参数 t , 即 $t_i = ih, i = 0, 1, \dots, N_i$, 则 t_i 处的曲率为

$$k_i = k(t_i, \sigma) = \frac{|X_t(t_i, \sigma)Y_u(t_i, \sigma) - X_u(t_i, \sigma)Y_t(t_i, \sigma)|}{[X_t(t_i, \sigma)^2 + Y_t(t_i, \sigma)^2]^{\frac{3}{2}}} \quad (7)$$

式中

$$X_t(t_i, \sigma) = x(t_i) * G_t(t_i, \sigma) = \Delta \sum_{u=i-m}^{i+m} \left[x(uh) \left(\frac{-(i-u)h}{\sigma^3} \frac{1}{\sqrt{2\pi}} \right) \exp\left(-\frac{(i-u)^2 h^2}{2\sigma^2}\right) \right]$$

$$Y_t(t_i, \sigma) = y(t_i) * G_t(t_i, \sigma) = \Delta \sum_{u=i-m}^{i+m} \left[y(uh) \left(\frac{-(i-u)h}{\sigma^3} \frac{1}{\sqrt{2\pi}} \right) \right]$$

$$\exp\left(-\frac{(i-u)^2 h^2}{2\sigma^2}\right)]$$

$$X_u(t_i, \sigma) = x(t_i) * G_u(t_i, \sigma) = \Delta \sum_{u=i-m}^{i+m} \left[x(uh) \left(\frac{(i-u)^2 h^2 - \sigma^2}{\sigma^5} \frac{1}{\sqrt{2\pi}} \right) \exp\left(-\frac{(i-u)^2 h^2}{2\sigma^2}\right) \right]$$

$$Y_u(t_i, \sigma) = y(t_i) * G_u(t_i, \sigma) = \Delta \sum_{u=i-m}^{i+m} \left[y(uh) \left(\frac{(i-u)^2 h^2 - \sigma^2}{\sigma^5} \frac{1}{\sqrt{2\pi}} \right) \exp\left(-\frac{(i-u)^2 h^2}{2\sigma^2}\right) \right]$$

$$\Delta = \sigma \sqrt{2\pi} / \left(\sum_{i=1}^m 2 \exp\left(-\frac{(ih)^2}{2\sigma^2}\right) + 1 \right)$$

根据高斯函数的性质, 对于上述公式中的 m , 有如下约束

$$m \geq \frac{\sigma}{h} \sqrt{\ln\left(\frac{2\pi}{\sigma^2 \varepsilon^2}\right)} \quad (8)$$

式中, ε 为一个给定的公差, 取 $m = 3 \lceil \sigma/h \rceil$ 。

在跟踪到所要刻画物体的边缘信息后, 就可以进行画笔模拟了。模拟算法如下:

对平滑后的边缘曲线 $R(t, \sigma) = [X(t, \sigma), Y(t, \sigma)]$, 按下面的算法计算该曲线的左右轮廓线。设 t_i 处对应的左右轮廓线上的点分别为 o_{0i} 和 o_{1i} , 其计算公式如下^[12]:

$$o_{ni} = R_i - f(k_i) \phi_n(k_i) w(l(r_i)) d_i$$

$$o_{((n+1)\%2)i} = R_i + f(k_i) \times \phi_{((n+1)\%2)}(k_i) w(l(r_i)) d_i \quad (9)$$

式中

$$n = \begin{cases} 1 & d_i \text{ 在 } (r_{i+1} - r_i) \text{ 的右方} \\ 0 & \text{其他} \end{cases}$$

$R_i = R(t_i, \sigma) = (X(t_i, \sigma), Y(t_i, \sigma))$, $n = \{0, 1\}$ 表示原曲线的左右侧, d_i 表示 t_i 点处的法方向。式(9)中的其他几个函数及其意义解释如下:

$f(k_i)$ 用来控制曲线上各点左右轮廓线宽度, 函数形式为^[12]

$$f(k_i) = \begin{cases} (W_{\max} - W_{\min}) \left(-\frac{1}{2} \cos(\pi \ln(1 + \alpha k_i))\right) + \frac{1}{2} + W_{\min} & \ln(1 + \alpha k_i) < 1 \\ W_{\max} & \text{其他} \end{cases} \quad (10)$$

式中, W_{\max} 和 W_{\min} 均为常数, 用于确定画笔的最大宽度和最小宽度, α 用来控制线条夸张度, 为预先给定的常数, 该常数由实验者自己根据图像绘制情况自行给定, 式(10)中对数底的作用跟 α 是一样的, 但是作用没有 α 明显, 本文在实验中取对数底为 e 。

ϕ_n 是用来控制两条轮廓线扩张比率的, 函数形式如下^[12]

$$\begin{aligned} \phi_0 &= \frac{1}{2} + \frac{1}{2}g(h(k_i)) \\ \phi_1 &= \frac{1}{2} - \frac{1}{2}g(h(k_i)) \end{aligned} \quad (11)$$

式中

$$g(x) = 2\left(\frac{1}{\gamma^{-x}} - 0.5\right)$$

$$h(k_i) = \begin{cases} \left(-\frac{1}{2}\cos(\pi\ln(1 + \alpha k_i)) + \frac{1}{2}\right)^\beta & \ln(1 + \alpha k_i) < 1 \\ 1 & \text{其他} \end{cases}$$

式中, γ 和 β 为用来控制线条轮廓外形的系数, 是预先给定的常数, 与 α 相同, 这两个常数也是由实验者自己根据图像绘制情况自行给定, 上式中对数底也为 e 。

$w(l(R_i))$ 用于控制曲线两端某一个阈值范围内轮廓线的宽度, $l(R_i)$ 为 i 点到起始端点的弧长,

其表达式为^[12]

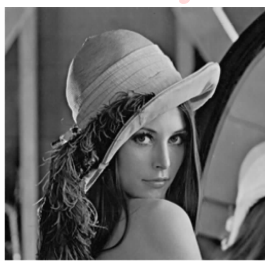
$$w(x) = \begin{cases} 1 - \left(1 - \frac{x}{Th}\right)^\delta & x < Th \\ 1 & Th \leq x \leq len - Th \\ 1 - \left(1 - \frac{len - x}{Th}\right)^\delta & len - Th \leq x \end{cases} \quad (12)$$

式中, Th 和 δ 是用来控制线条末端形状的参数, 是常数, 也是由实验者自己根据图像绘制情况自行给定; len 为曲线总的弧长。

由式(9), 可以将曲线 $R(t, \sigma)$ 上所有点处的左右轮廓线的坐标计算出来, 然后将左右轮廓点连接起来形成左右轮廓线, 由轮廓线包围的区域是一个封闭的区域, 最后使用多边形填充算法^[13], 向这个封闭的区域填充颜色, 形成画笔。

3 实验结果

本文方法实现环境: 1.6 GHz Intel Core 2 处理器、1 GB 内存、Windows XP 操作系统、MATLAB 7.4 开发环境。图 1 显示了 Lena 图像的线条画绘制过程。实验中取共圆性度量函数的方差 $\sigma_c = 1/2$, 光滑性度量函数的方差 $\sigma_s = \sqrt{2}/2$, 接近性度量函数



(a) 原始图像



(b) 边缘图像



(c) 曲线拟合高斯平滑后图像



(d) $W_{\max} = 1, W_{\min} = 0.5$ 时线条画



(e) $W_{\max} = 2, W_{\min} = 1$ 时线条画



(f) $W_{\max} = 4, W_{\min} = 2$ 时线条画

图 1 实验结果图

Fig. 1 Experimental results graph

的方差 $\sigma_p = 15$, 控制曲线上各点左右轮廓线宽度函数中用以控制线条夸张度的系数 $\alpha = 25$, 用以控制线条轮廓外形的系数 $\beta = 0.055$ 、 $\gamma = 35$, 用以控制线条末端形状的系数 $Th = 10$, $\delta = 1.5$ 。图1(d)(e)(f)是分别赋予 $W_{max} = 1$ 、 $W_{min} = 0.5$ 、 $W_{max} = 2$ 、 $W_{min} = 1$ 、 $W_{max} = 4$ 、 $W_{min} = 2$ 画笔后的线条画绘制结果。图2为文献[12]的画笔生成的线条画, 对比图1本文实验效果与其几乎相同, 其画笔与本文画笔的不同主要是在对画笔生成形状的理念不同, 不同处如图3所示, 图中中间白线为原始线条。图4为不同参数的画笔生成线条画, 参数

的不同产生不同风格的画笔, 由于本文算法在较粗的线条中更能够反应线条风格, 但是过粗的线条又会影响本文的线条画的绘制效果, 所以本文选择了 $W_{max} = 4$, $W_{min} = 2$ 的线条宽度。从图中可以看出3幅图像生成的线条其主要差异可以从线条的两端看出, 如图4(a)中的线条两端几乎都是平的, 没有形成尖端; 图(b)中的线条并不是两端都是尖端形式; 图(c)中可以看出线条又与前两幅图有所不同, 其线条是两端都形成尖端形式。图5是多组线条画绘制结果, 左列是原图像, 右列是本文方法绘制的线条画结果。

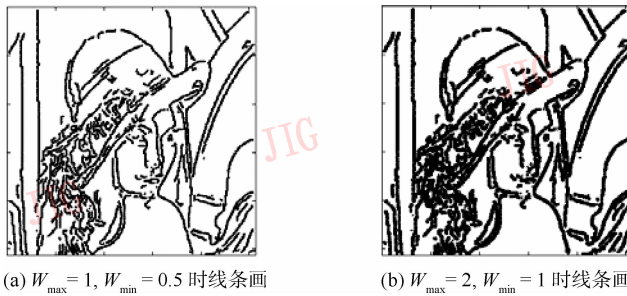


图2 文献[12]的画笔生成结果

Fig. 2 The results generated by the brush of the literature [12]

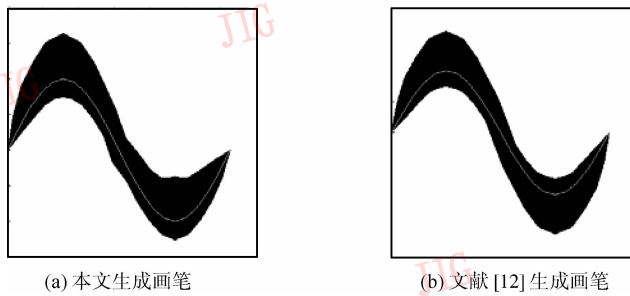


图3 本文画笔与文献[12]画笔

Fig. 3 The brush of our paper and the literature [12]

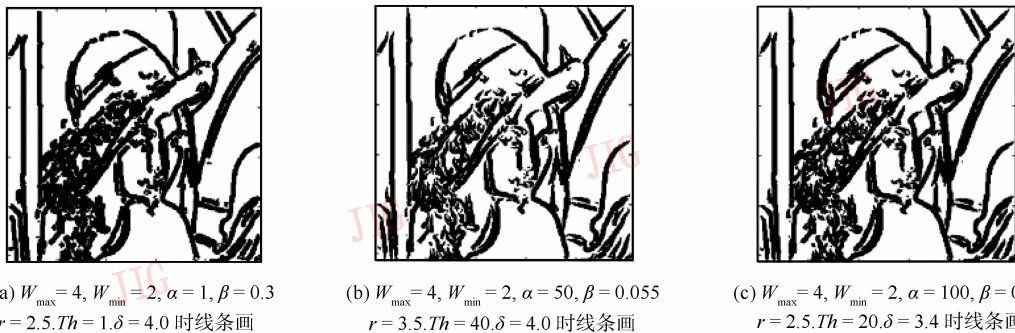


图4 不同画笔参数下的实验结果

Fig. 4 The experimental results of different parameters



(a) 原图

(b) 赋予不同宽度的线条画绘制结果

图 5 更多的绘制结果

Fig. 5 More drawing results

4 结 论

给出一个基于边缘跟踪算法的图像线条画生成方法,用于模仿人类进行绘画的过程。首先使用 Canny 算子结合边缘跟踪算法提取出边缘曲线,并运用曲线插值和高斯滤波器对边缘曲线进行连接和光滑。在此基础上给出一个基于曲率的线条绘制算法,对图像的线条画进行了较为满意的模拟绘制。然而本文方法受背景边缘和纹理边缘的影响较大,对此需要进一步探讨,以获得高质量的图像线条画。

参考文献 (References)

[1] Strassmann S. Hairy brushes[J]. Computer Graphics(Proceedings of SIGGRAPH' 86),1986,20(4):225-232.
 [2] Tu Changhe, Sun Yuhong, Meng Xiangxu. Example-based style translation and customization for line drawings [J]. Chinese

Journal of Computers,2005,28(6):965-971. [屠长河,孙玉红,孟祥旭. 基于样本的线条画风格转换与定制方法的研究[J]. 计算机学报,2005,28(6):965-971.]
 [3] Rhodes G. Secrets of the face [J]. New Zealand Journal of Psychology,1994,23(1):3-17.
 [4] Li Y, Kobatake H. Extraction of facial sketch based on morphological processing [C] //Proceedings of the 1997 IEEE International Conference on Image Processing. Santa Barbara, California, USA: IEEE Computer Society,1997,316-319.
 [5] Chen Hong, Zheng Nanning, Xu Yingqing, et al. An example-based facial sketch generation system [J]. Journal of Software, 2003,14(2):202-208. [陈洪,郑南宁,徐迎庆,等. 基于样本学习的人像线条画生成系统 [J]. 软件学报,2003,14(2):202-208.]
 [6] Chen Hong, Zheng Nanning, Liang Lin, et al. Example-based automatic portrai-ture [J]. Chinese Journal of Computers,2003,26(2):147-152. [陈洪,郑南宁,梁林,等. 基于样本学习的肖像画自动生成算法 [J]. 计算机学报,2003,26(2):147-152.]
 [7] Son M, Kang H, Lee Y, et al. Abstract line drawings from 2D images [C] //Proceedings of the IEEE Pacific Conference on Computer Graphics and Applications, Maui, Hawaii, USA: IEEE

- Computer Society, 2007, 333-342.
- [8] Canny J. A computational approach to edge detection[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1986, 8(6): 679-698.
- [9] Fischer B, Buhmann J M. Path based clustering for grouping of smooth curves and texture segmentation [J]. IEEE Transactions on Pattern Analysis and Machine, 2003, 25(4): 1-6.
- [10] Shi Fazhong. Computer-Aided Geometric Design and Non-Uniform Rational B-Spline[M]. Beijing: Beijing University of Aeronautics and Astronautics Press, 1994. [施法中. 计算机辅助几何设计与非均匀有理 B 样条[M]. 北京: 北京航空航天大学出版社, 1994.]
- [11] Zhu Yanjuan, Zhang Liyan, Zhou Laishui. Contour extraction and curvature calculation for fragment reassembly[J]. Journal of Southeast University; English Edition, 2004, 20(2): 181-186.
- [12] Saito Suguru, Kani Akane, Chang Youngha et al. Curvature-based stroke redering[J]. Visual Computer, 2008, 24(1), 1-11.
- [13] He Bin, Ma Tianyu, Wang Yunjian, et al. Visual C++ Digital Image Processing [M]. Beijing: Posts & Telecom Press, 2002: 444-448. [何斌, 马天予, 王运坚, 等. Visual C++ 数字图像处理[M]. 北京: 人民邮电出版社, 2002: 444-448.]