

一种改进的铅笔画的生成方法

李龙生^{1,2)} 周经野¹⁾ 陈益强^{2,3)} 刘军发^{2,3)} 周仁琴^{1,2)}

¹⁾ (湘潭大学信息工程学院, 湘潭 411105) ²⁾ (上海中国科学院计算技术研究所, 上海 201203)

³⁾ (中国科学院计算技术研究所, 北京 100080)

摘 要 给出了一种改进的根据真实 2D 图像自动生成相应的非真实感铅笔画的方法。首先将彩色图像进行霓虹处理, 再进行反相计算和灰度化, 就可以产生铅笔画中的轮廓效果。其次, 为了更好地产生铅笔画的光照效果及其局部走势纹理, 采用线积分卷积 (LIC) 的方法来生成类似的效果, 并且用适当的图像分割方法来获取进行 LIC 处理的有意义的区域。实验结果表明, 本文的方法与以往的方法相比, 能生成具有不同风格的效果, 并且生成的速度更快。

关键词 铅笔画 线积分卷积 非真实感绘制 (NPR) 霓虹处理

中图法分类号: TP391.9 **文献标识码:** A **文章编号:** 1006-8961(2007)08-1423-07

A Improved Generation Method of Pencil Sketching

LI Long-sheng^{1,2)}, ZHOU Jing-ye¹⁾, CHEN Yi-qiang^{2,3)}, LIU Jun-fa^{2,3)}, ZHOU Ren-qin^{1,2)}

¹⁾ (The Institute of Information Technology, Xiangtan University, Xiangtan 411105)

²⁾ (Institute of Computing Technology, Chinese Academy of Sciences, Shanghai 201203)

³⁾ (Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080)

Abstract In this paper we propose an improved method for automatically generating pencil sketching from real 2D color images. First, we apply neon processing, then negative and gradation to an image to generate contour effects in pencil sketching. Second, to obtain better effects of light and texture of pencil sketching, we use line integral convolution (LIC) algorithm, and segment the image into significant areas for LIC using an effective segmentation algorithm. Compared to the traditional method, the experiments show that our method generates images with different style fast and effectively.

Keywords pencil sketching, line integral convolution, non-photo-realistic rendering, neon processing

1 引 言

最近, 非真实感图形学成了计算机图形学一个非常重要的研究领域, 自 20 世纪 80 年代以来, 这一领域得到了蓬勃发展, 国外许多研究人员撰写了大量的论文, 国际会议开始设立专题, 以专门讨论非真实感图形学。迄今为止, 已经有许多模拟传统艺术风格的技术被开发出来, 比如钢笔画^[1~3]、油画^[4]、水彩画^[5]、铅笔画^[6~8]。本文给出了一种根据一张 2D 的真实图片自动生成其相应铅笔画的方法。迄

今为止, 已经有几种技术用来实现铅笔画的效果, 有一种早期的 2D 绘图系统叫做 PencilSketch^[9], 使用鼠标和一个虚拟桌面, 让用户交互地给出一些参数, 比如铅笔的硬度, 使用铅笔时的压力, 条纹的方向等等, 这种技术不能做到自动化, 限制了它的应用, Sousa 和 Buchanan 使用电子显微镜扫描真实的铅笔画^[10], 也开发了一个铅笔画的产生模型, 在这个模型中, 当用户指定一些参数后, 它能把一张 2D 图片转换成一张铅笔画, 另一个模型通过指定一些参数值也可以把 3D 场景直接渲染成铅笔画^[11], 但这两种方法仍属于交互式生成铅笔画的研究, 而且 3D

基金项目: 国家自然科学基金项目 (60303018)

收稿日期: 2005-10-31; 改回日期: 2006-05-19

第一作者简介: 李龙生 (1970 ~), 男。现为湘潭大学信息工程学院计算机应用专业硕士研究生。研究方向为图像图形处理等。E-mail: aslms@tom.com

场景的需求也限制了其使用的方便性。Takagi 和 Fujishiro 提出了一种彩色铅笔画的模型^[8],属于另一种风格铅笔画方法的研究。在文献[7],[9]中,使用了线积分卷积的方法来生成铅笔纹理,但由于它们都使用了区域合并的图像分割算法,时间开销较大,如文献[7]对一幅 $1\,024 \times 768$ 的图像处理要花费 20 分钟的时间。其中绝大部分时间开销花在图像分割上。

铅笔画的生成方法通常是生成方法是否有交互,使用 3D 场景作输入还是使用 2D 图片作为输入来分类的。本文方法属于从 2D 数字图片中自动生成,与同类方法比较,本文方法有如下优点。首先,本文方法是完全自动的,可以完全不需要用户的交互,这样可以增强它的应用范围。其次,与文献[7],[9]相比,通过霓虹处理生成铅笔画中的轮廓效果,改进了 LIC 处理中白噪声图的生成方法,通过颜色缩放处理及对图像中灰度进行分层处理来控制白噪声的生成方式,从而控制图像中线积分卷积的效果,即能够表达出图像中的铅笔画的纹理,又能保留一些必要的空白区域,突出铅笔画表达的重点区域,使结果具有更好的效果。另外,选取了新的高效的图像分割的方法,很大程度上缩短了生成铅笔画的时间。

2 产生铅笔画的方法

2.1 用霓虹处理产生铅笔画中的轮廓效果

彩色图像的霓虹化首先是计算原始像素的红绿

蓝分量与相同行及相同列相邻像素的梯度,即差的平方和的平方根,再将梯度值作为新像素的红绿蓝分量值。由于霓虹对图片进行了梯度处理,因而对图像进行霓虹处理后可以产生图像的线条效应。其具体的形式化描述如下:

设原图像像素 $f(i, j)$ 的红、绿、蓝分量为 r_1, g_1, b_1 , 相同行 $f(i+1, j)$ 的红、绿、蓝分量为 r_2, g_2, b_2 , 相同列 $f(i, j+1)$ 的红、绿、蓝分量为 r_3, g_3, b_3 , 则处理后图像 $g(i, j)$ 的红、绿、蓝分量为 r, g, b , 这 3 个分量可以通过下式计算得到:

$$\begin{aligned} r &= 2 \times \sqrt{(r_1 - r_2)^2 + (r_1 - r_3)^2} \\ g &= 2 \times \sqrt{(g_1 - g_2)^2 + (g_1 - g_3)^2} \\ b &= 2 \times \sqrt{(b_1 - b_2)^2 + (b_1 - b_3)^2} \end{aligned} \quad (1)$$

通过上面的处理可以得到图 1(b) 的效果,但是其距离铅笔画还有较大的差距,经过多次实验后发现,将彩色图像进行霓虹处理后,再进行反相,然后再灰度化,最后使用窗口变换,就可以产生类似于铅笔画中的轮廓效果,图 1(c) 显示了一个处理后的例子。

从处理后的图中,可以看出其不足之处是没有表达出铅笔画中特有的铅笔纹理,而且在原图中的阴影部分也没有表现出来,也就是没有表现出原图中的光照效果。为了达到这种效果,还必须提取出必要的阴影区域,并对图片进行铅笔画的纹理处理。对图像进行线积分卷积(line integral convolution, LIC)可以做到这一点,并最后生成类似铅笔画的纹理效果。因此考虑将霓虹处理与线积分卷积相结合



图1 用霓虹处理得到图像边缘

Fig. 1 The edge by neon processing

的方法来产生铅笔画效果。

2.2 使用线积分卷积生成铅笔画的纹理

线积分卷积是 Cabral 和 Leedom 于 1993 年在 SIGGRAPH 上首先提出的^[12],由于线积分卷积能较好地对矢量场进行可视化处理,因此被应用到图像处理、计算机艺术等领域,其基本原理是基于矢量方向的相关性对噪声纹理进行低通滤波。

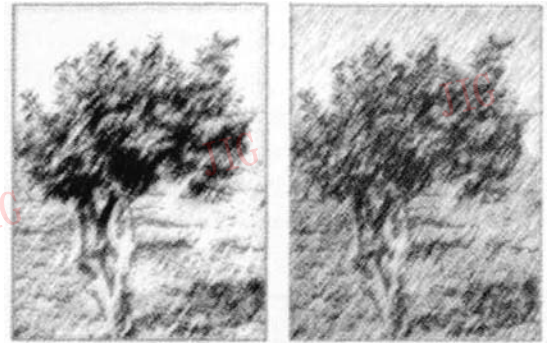
然而要将线积分卷积结合进来,还需要进行以下一些技术处理。

(1)颜色缩放处理 为了进行 LIC 处理,必须得到原图的白噪声图,但由于 LIC 处理是对全图进行,而铅笔画作为素描画的一种,从审美学的角度来说,应保留有一定的空白区域,用以激发欣赏者的联想,因此,有必要控制进行 LIC 处理的区域,对原始图片进行颜色缩放处理,色彩缩放的基本原理就是对图像的 r, g, b 3 个颜色分量分别乘以一个相同的比例值,这里放缩比例一般取图像中 r, g, b 原始大小的 1.5 倍左右,根据实际情况也可以进行调整,需要说明的是,这种处理只适合于那些背景较淡的图像。

(2)白噪声图的获取 白噪声图是一种以某种随机方式生成的黑白噪声图片,由于 LIC 处理后的色调效果与白噪声图有很大的关系,因此对白噪声的产生方式进行了改进。为了使原图的处理结果匹配于其相应的铅笔画,用原图来控制噪声点产生几率,与文献[7]、[9]不同的是这里对图像的不同灰度范围内的情况分开处理,以使得最后生成的铅笔画中的明暗对比更突出,空间层次感更强,在图 2 中把改进后方法的结果与文献[7]、[9]的白噪声处理结果进行了对比,可以看出改进后的效果在明暗对比和空间层次感方面要好很多,而且颜色缩放处理后产生了一些空白区域,突出了表达的重点,改进后的噪声图产生的方式为

$$I_{\text{noise}} = \begin{cases} I_{\text{noise1}} = \begin{cases} I_{\text{max}} & P \geq T_1 \\ I_{\text{min1}} & \text{其他} \end{cases} & I_{\text{input}} \leq L_1 \\ I_{\text{noise2}} = \begin{cases} I_{\text{max}} & P \geq T_2 \\ I_{\text{min2}} & \text{其他} \end{cases} & L_1 < I_{\text{input}} \leq L_2 \\ I_{\text{noise3}} = \begin{cases} I_{\text{max}} & P \geq T_3 \\ I_{\text{min3}} & \text{其他} \end{cases} & L_2 < I_{\text{input}} \end{cases} \quad (2)$$

其中, $T_1 = k_1(1 - I_{\text{input}}/255)$, $T_2 = k_2(1 - I_{\text{input}}/255)$, $T_3 = k_3(1 - I_{\text{input}}/255)$, P 是一个随机数, I_{input} 是输入像素的灰度值, I_{noise} 是输出的灰度值, I_{max} 是输出像素的最大灰度值,一般取 255,但根据情况也可能小于



(a) 改进后的白噪声生成方法

(b) 改进前的效果

LIC 处理后的效果

图 2 噪声产生方式改进前后对比

Fig. 2 The comparison between improved method and original method

这个值, $I_{\text{min1}}, I_{\text{min2}}, I_{\text{min3}}$ 是输出像素的灰度的最小值, L_1, L_2 是 2 个不同的灰度域值,以本文的实验结果, L_1 一般取 25 左右, L_2 一般取 80 左右,这里的 I_{min} 可以不为 0,因为考虑到处理后要与霓虹处理的结果合并,色调太深可能会影响铅笔画的效果,因此 I_{min} 可以进行适当调整。 k_1, k_2, k_3 是 3 个经验值,以本文实验的结果, k_1 一般取 0.9, k_2, k_3 一般取为 0.7。用户也可以根据实际进行调整。图 3 为对图 1(a) 进行颜色缩放处理后的图片再进行进行白噪声处理的结果。



图 3 白噪声图

Fig. 3 The white noise image

(3)图像分割 常常看到画家在铅笔画中的不同区域用不同的铅笔条纹的方向、深浅来表达物体

的明暗、空间感、质感等等。为了表现出这种效果,对图形进行分割,对不同的区域分别提取特征,再分别产生各区域的向量场。由于我们的目的是针对图片中的不同区域操作,因此选取了一种算法效率较高,而又能对图片中各个有意义的区域进行大致分割的图像分割方法,一种基于图论的彩色图像分割法^[13],在此作简单说明,图像中的所有像素点首先被作为无向有权图 $G=(V,E)$ 中所有的顶点,再连接所有的相邻边作为其边集,这里的相邻是指 8 邻域点,即一个像素与周围 8 个点相邻,边的权值是两个顶点的区别,这里一般指灰度值之差。图像的分割就是将图像最终通过相邻边的聚类成为一个森林,森林中的每棵树是一个最小生成树,最后形成的每棵树就对应一个分区,通过定义一个合并的断言,来确定两个树是否合并为一个树,断言的形式为

$$D(C_1, C_2) = \begin{cases} \text{TRUE} & \text{Dif}(C_1, C_2) \geq \text{Min}(C_1, C_2) \\ \text{FALSE} & \text{其他} \end{cases} \quad (3)$$

式中, $D(C_1, C_2)$ 指图中两个区域 C_1, C_2 合并的断言,为假则合并, $\text{Dif}(C_1, C_2)$ 指 C_1, C_2 两区域之间的区别,实际上是指两个区域之间相连的最小边的权值, $\text{Min}(C_1, C_2)$ 是两个区域 C_1, C_2 内部的区别的最小值。这种方法的优点是可以调节图像分割的粗糙度,对于图像有些区域可以忽略其细节,这对铅笔效果的表达也是有用的,另外分割算法的时间复杂度可以达到 $O(n \log n)$, n 是无向图中的边数,与文献[7]、[9]相比很大程度加快了铅笔画生成的时间,图 1(a) 图像分割的结果如图 4 所示,图中使用了不同的灰度来代表不同的分区。

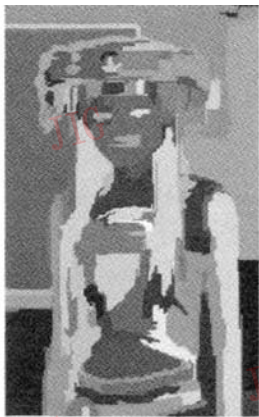


图 4 图像分割结果

Fig. 4 The result of image

(4) 局部纹理方向的确定 铅笔画中条纹的方向是很重要的,画家常常用它们来表达物体的形状、纹理等信息,如果所绘制的物体的纹理具有一定的方向特点,则铅笔画中应表达出这种特征,为了达到这种效果,使用傅里叶功率谱法分析各区域的纹理方向,这个方法的基本原理是如果图像的纹理有方向性的话,则当图像转换到频率域后,在与纹理方向垂直的方向上应该有较高的能量,由此在分析每个像素的方向时,以像素为中心建立一个边长为 M 的正方形窗口,在窗口内进行傅里变换,转换到频率域后再把频率域分成许多小的角度,分析每个部分内的能量和,找到一个能量最大的区域,再与平均能量比较,如果这个比率大于设定的阈值,则像素的方向为与这个区域的中线垂直的方向。如果用这种方法找不到方向,则考虑给这个区域一个随机的方向。

(5) LIC 处理 LIC 选择噪声(如白噪声)作为输入纹理,输出纹理的每个像素值均通过线积分卷积得到:首先基于该像素沿矢量正、反方向对称积分生成流线,流线上所有像素对应的输入噪声值按卷积核(convolution kernel)参与卷积,结果作为输出纹理的像素值。图 5 显示了 LIC 处理的一般过程,把向量场(图 5(a))和白噪声图片(图 5(b))作为 LIC 处理的输入数据,最后产生的结果是反映出向量场特征的纹理。而这种效果,正是铅笔画中所需要的,铅笔画的关键因素之一是使用线条来表达物体,画家对铅笔的灵活操作,可以通过各种线条表达出许多不同的色调,而计算机也可能通过 LIC 来模拟类似的效果。

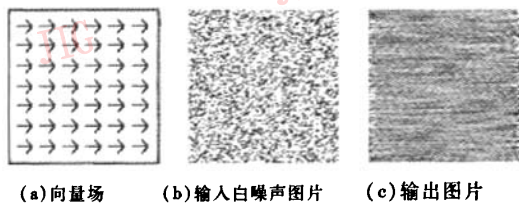


图 5 LIC 处理过程

Fig. 5 The process of LIC

设 $F_{out}(x, y)$ 为输出图像中 (x, y) 点处的像素值, $F_{in}(d_i)$ 为流线正方向上第 i 段处的像素的像素值, $F_{in}(d'_i)$ 为流线反方向上第 i 段处的像素的像素值, h_i 为流线正方向上卷积核在第 i 段上的积分结果, h'_i 为流线反向上卷积核在第 i 段上的积分结果, LIC 的处理后的结果可以用下式来表示:

$$F_{out}(x, y) = \frac{\sum_{i=0}^l F_{in}(d_i) h_i + \sum_{i=0}^{l'} F_{in}(d'_i) h'_i}{\sum_{i=0}^l h_i + \sum_{i=0}^{l'} h'_i} \quad (4)$$

其中

$$h_i = \int_{s_i}^{s_i + \Delta S_i} k(w) dw \quad (5)$$

式中, ΔS_i 是流线在第 i 段上走过的实际距离, $k(w)$ 是卷积核函数, 在这里选择汉宁窗口函数, 因为它能满足对低通滤波的要求, 具体形式为

$$\begin{aligned} k(w) &= \frac{1 + \cos(cw)}{2} \times \frac{1 + \cos(dw + \beta)}{2} \\ &= \frac{1}{4} (1 + \cos(cw) + \cos(dw + \beta) + \\ &\quad \cos(cw) \cos(dw + \beta)) \end{aligned} \quad (6)$$

式中的 c 和 d 是两个汉宁函数的扩大系数, β 是汉宁函数的相移。通过这种运算就可以将图 6 中的输入白噪声图 5(b) 根据向量场图 5(a) 的具体方向数据转化为图 5 的结果图 5(c), 图 6 给出了一张真实的铅笔的纹理图, 可以看出由 LIC 处理产生的结果与真实的铅笔纹理是有一定的相似之处的。

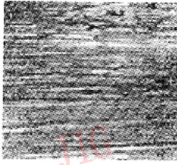


图 6 真实的铅笔纹理

Fig. 6 Realistic pencil tone

通过以上几步的处理, 就可以很好地实现图像的铅笔画效果。图 1(a) 中的彩色图像用本文方法处理后所产生的铅笔画显示如图 7 所示。

2.3 算法流程

综上所述, 可以将产生铅笔画的算法描述如下:

- (1) 对原始彩色数字图片 I_{input} 进行灰度化, 并对其进行颜色缩放处理, 得到 $I_{threshold}$ 。
- (2) 对上面颜色缩放处理的结果 $I_{threshold}$ 进行白噪声处理, 产生相应的白噪声图 I_{noise} 。
- (3) 把原始图 I_{input} 分割成为不同的区域, 得到 $\{S_i | i = 1, 2, \dots, N\}$ 。
- (4) 对分割的各区域 $\{S_i\}$ 产生向量场 $\{V_i\}$, 作为铅笔画的纹理方向 $\{D_i\}$ 。
- (5) 对白噪声图 I_{noise} 和向量场进行线积分卷



图 7 最后的铅笔画的结果

Fig. 7 The result of pencil Sketching

积, 产生铅笔画的纹理, 得到 I_1 。

(6) 对原始图片 I_{input} 进行霓虹处理, 得到 I_{neon} 。

(7) 对霓虹处理的结果进行反转, 窗口变换处理, 得到 I_2 。

(8) 合并上面的步骤 5、7 的结果, 产生铅笔画, $I_{output} = I_1 + I_2$ 。

此算法已经以 VC++ net 作为平台, 在 WindowsXP 操作系统、PIV1.7GHz 的兼容机上实现。对 1 幅 350×280 的 24 位 bmp 真彩色位图, 该算法大约需时 18s 左右, 便可将这幅 2D 图片转换为 1 幅铅笔画。对 1 幅 1024×768 的图片处理所需的时间约为 80s, 而文献[7]在 PIII Xeon 933 的图形工作站上所用的时间是 1200s, 本文算法在执行时间上快了很多。文献[9]虽然没有给出具体的执行时间, 但因其图像分割的方法相同, 而图像分割所占用的时间约为整个算法执行时间的 90% 左右, 所以与文献[7]相比执行时间上也不可能相差太大。处理的过程是全自动化的。当然, 用户也可以自己对算法中的参数, 如白噪声的最大或最小像素灰度值、LIC 处理时流线的长度等等, 事先进行调节。

3 实验结果

运用本算法对一些图像进行了实验, 产生的铅笔画效果基本令人满意。图 8、图 9、图 10 给出了几幅图像的自动生成的铅笔画。把运用本算法所产生的结果与文献[9]中的结果进行了比较, 如图 11 所示。可以看出, 文献[9]中的结果有点类似油画的

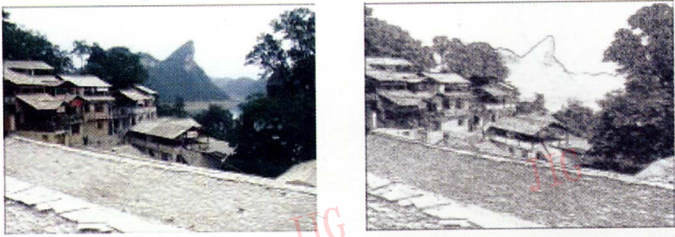


图 8 风景照片的铅笔画效果
Fig.8 The result of Landscape photo



图 9 人物照片的铅笔画效果
Fig.9 The result of human photo

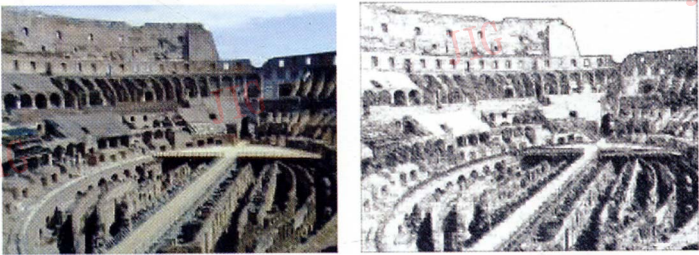
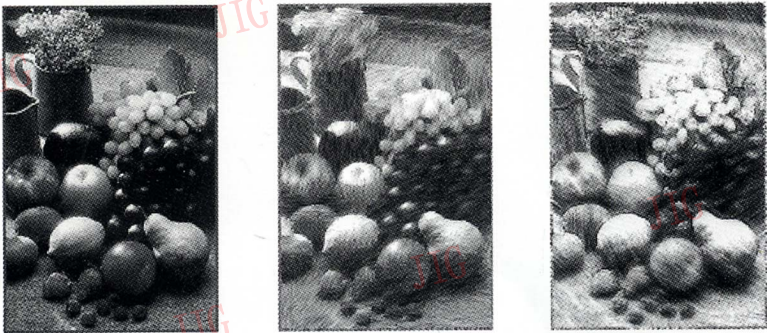


图 10 古建筑的铅笔画效果
Fig.10 The result of antique architecture photo



(a) 原始图片 (b) 文献[9]处理的结果 (c) 本文算法处理的效果

图 11 本文算法与文献[9]算法产生结果的比较
Fig. 11 The comparison between our algorithm and paper[9]

风格,本文效果更接近于手绘铅笔画的风格。

4 结 论

本文给出了一种将2D图片转换为其对应的铅笔画的基本思想和算法,从过程上来说有点类似模拟人类手工作画的过程,即先勾勒出景物的大致轮廓,再进一步深入刻画出各部分的不同色调。但是,模拟人的手工作画过程并不是一件简单的事情。因为这也意味着要模拟人类的思考和推理,特别是创造性的思考,这用算法实现是极其困难的,因此,只能尽量模拟铅笔画产生的艺术效果,如铅笔画的色调。NPR的研究不是用来替代艺术家,但是它可以为一些没有作画经验的人士提供一种工具,并在一些应用领域,如数字娱乐、动画制作等等中具有很大的应用价值。

今后的工作主要是希望能更大地改善生成的铅笔画的质量,模拟出更丰富的铅笔画的色调,找到更好的图像分割方法和控制条纹方向的方法。

参考文献 (References)

- 1 Salisbury M I P, Anderson S E, Barzel R, *et al.* Interactive pen-and-ink illustration [A]. In: Proceedings of SIGGRAPH '94 [C], Florida, USA, 1994: 101 ~ 108.
- 2 Winkerbach G, Salesin D H. Computer-generated pen-and-ink illustration [A]. In: Proceedings of SIGGRAPH '94 [C], Florida, USA, 1994: 91 ~ 100.
- 3 Salisbury M P, Wong M T, Hughes J F, *et al.* Orientable textures for image-based pen-and-ink illustration [A]. In: Proceedings of SIGGRAPH '97 [C], Los Angeles, CA, USA, 1997: 401 ~ 406.
- 4 Hertzmann A. Painterly rendering with curved brush strokes of multiple sizes [A]. In: Proceedings of SIGGRAPH '98 [C], Florida, USA, 1998: 453 ~ 460.
- 5 Curtis C J, Anderson S E, Seims J E, *et al.* Computer-generated watercolor [A]. In: Proceedings of SIGGRAPH '97 [C], Los Angeles, CA, USA, 1997: 421 ~ 430.
- 6 Mao X, Nagasaka Y, Imamiya A. Automatic generation of pencil drawing from 2D images using line integral convolution [J]. CAD/Graphics, 2001, (9): 240 ~ 248.
- 7 Takagi S, Fujishiro I, Nakajima M. Volumetric modeling of colored pencil drawing [A]. In: Pacific Graphics '99 Conference Proceedings [C], Los Angeles, CA, USA, 1999: 250 ~ 258.
- 8 Li Nan, Huang Zhong. A feature-based pencil drawing method [A]. In: Computer Graphics and Interactive Techniques [C], Australasia and South East Asia, 2003: 135 ~ 143.
- 9 Vermeulen A H, Tanner P P. PencilSketch—a pencil-based paint system [A]. Graphics Interface '89 Conference Proceedings [C], London, 1989: 138 ~ 143.
- 10 Sousa M C, Buchanan J W. Observational model of blenders and erasers in computer-generated pencil rendering [A]. In: Graphics Interface '99 Conference Proceedings [C], Kingston, Ontario, Canada, 1999: 157 ~ 166.
- 11 Sousa M C, Buchanan J W. Computer-generated graphite pencil rendering of 3D polygonal models [A]. In: EUROGRAPHICS '99 Conference Proceedings [C], London, England, 1999: 195 ~ 207.
- 12 Cabral B, Leedom C. Imaging vector field using line integral convolution [A]. In: Proceedings of SIGGRAPH '93 [C], California, USA, 1993: 263 ~ 270.
- 13 Felzenszwalb P F, Huttenlocher D P. Efficient graph-based image segmentation [J]. International Journal of Computer Vision, 2004, 59(2): 167 ~ 181.