

彩色视频图像卡通风格化研究与实现

王巧玉, 陈锻生, 吴扬扬

(华侨大学 计算机科学与技术学院, 福建 厦门 361021)

摘要: 提出一种彩色视频图像卡通风格化的新方法, 将均值漂移滤波与高斯图像金字塔结合起来使用, 在对彩色视频图像进行颜色聚类 and 噪声消除的同时, 提高了处理的速度. 运用 DoG 算子对滤波后的图像进行边缘检测; 再对滤波后的图像进行色彩量化; 最后将量化后的图像与边缘图像进行融合. 融合时根据边缘处梯度的大小设置边缘颜色的深度, 使得最终得到的卡通化图像不仅在高对比度区域得到加强而且减少了噪声, 边缘更加自然. 实验结果表明: 本算法与传统的双边滤波和 DoG 算子相比, 获得了更好的卡通风格化效果和更快的运行速度, 并且可以直接将输入的视频文件转换为卡通化的视频文件输出.

关键词: 图像卡通化; 均值漂移滤波; 高斯金字塔; 色彩量化

中图分类号: TP 391

文献标志码: A

传统的卡通制作需要花费动画师大量的时间和精力, 为了提高动画的制作效率, 1963 年, 贝尔实验室的 Knowlton^[1] 开发了一个名为 BEFLIX 的二维动画制作系统. 2000 年, 于金辉等^[2] 使用计算机模拟出了卡通烟雾效果, 采用简单的正弦波模拟出了卡通流水效果. 2001 年, 于金辉等^[3] 又引入了一个用于立体动画的卡通水模型. 2002 年, de Carlo 等^[4] 提出了采用图像区域分割及着色技术来对二维图像进行卡通风格化渲染. Wang 等^[5] 和 Collomosse 等^[6] 将 Mean-shift 图像分割用到了视频抽象化中. 国内也有人对这种基于图像分割的抽象化方法进行了一些研究^[7-9], 但是这种方法通常在区域分割时存在分割不准确, 分割线不平滑, 算法花费时间长等缺点. 2006 年, Winnemöller 等^[10] 提出了一个全自动的基于视频和图像的卡通风格化渲染框架, 但耗时也比较长. 后来的一些卡通化处理方法也大都基于该框架进行改进的^[11-13]. 双边滤波是保边去噪的滤波器, 它在去除图像的噪声的同时能够很好的保留其边缘. 常见的保边去噪滤波器^[14] 还有均值漂移滤波^[15]、桑原滤波^[16] 以及广义的桑原滤波^[17] 等. 本文将均值漂移滤波与高斯金字塔结合起来使用, 大大缩短了处理时间.

1 相关技术与流程设计

1.1 均值漂移滤波

均值漂移(Mean-shift)是一种无参统计滤波算法, 主要是找个种子点, 然后开始在该种子点邻域内寻找其目标点的密度中心, 那么种子点到密度中心点的向量方向就是密度上升方向. 更新密度中心点为种子点, 迭代, 直到收敛或者达到最大迭代次数. Mean-shift 算法为

$$Mh(x) = mh(x) - x = \frac{\sum_{i=1}^n x_i g(\frac{x - x_i}{h}) w(x_i)}{\sum_{i=1}^n g(\frac{x - x_i}{h}) w(x_i)} - x. \tag{1}$$

式(1)中: $Mh(x)$ 为 Mean-shift 向量; $w(x_i) \geq 0$ 为赋给采样点 x_i 的权重; s 为被平滑的像素值; $g(x)$ 为

核函数.

若允许误差为 a , 则 Mean-shift 算法循环的执行下面的 3 个步骤, 直到达到结束条件.

步骤 1 计算 $mh(x)$.

步骤 2 将 $mh(x)$ 的值赋给 x .

步骤 3 若 $|mh(x) - x| < a$, 则结束循环, 否则执行步骤 1.

由式(1)可以看出: $mh(x) = x + Mh(x)$. 故上面的步骤就是沿着概率密度上升的方向不断移动, 在满足一定的条件的情况下, Mean-shift 算法一定会收敛到该点附近的峰值.

一幅图像可以表示成一个二维网格点上的 P 维向量, $p=1$, 表示一幅灰度图; $p=3$ 表示一幅彩色图. 用一个 $P+2$ 维的向量 $x(x^s, x^r)$ 来表示一幅图像, 其中 x^s 表示网格点的坐标, x^r 表示该网格点上的 P 维向量特征. Comanicu 等^[15] 提出均值漂移算法用于图像平滑时来估计 x 的分布的核函数 $g(x)$, 其具体形式为

$$g(x) = \frac{c}{h_s h_r} k\left(\left\|\frac{x^s}{h_s}\right\|^2\right) k\left(\left\|\frac{x^r}{h_r}\right\|^2\right). \tag{2}$$

式(2)中: h_s 和 h_r 控制着平滑的解析度, 它们的值越大, 平滑后的图像保留的细节就越少, c 是一个归一化常数.

分别用 x_i 和 $z_i (i=1, 2, \dots, n)$ 表示原始的图像和平滑后的图像, 则对图像采用 Mean-shift 算法进行平滑的具体有如下 3 个步骤.

步骤 1 初始化 $j=1$, 并使 $y_{i,1} = x_i$.

步骤 2 用 Mean-shift 算法计算 $y_{i,j+1}$, 直到收敛, 将收敛后的值记为 $y_{i,c}$.

步骤 3 滤波后的像素 $z_i = (x^s, y_{i,c}^r)$.

1.2 高斯金字塔

图像金字塔是以多分辨率表示图像的一种结构, 它是一个图像集合, 集合中所有的图像都来源于同一个原始图像, 它们都是通过对原始图像进行连续下采样获得的. 高斯金字塔^[18] 最低一层(第 0 层)图像是原始图像, 随后每上升一层, 图像缩小为原来的 $1/4$.

假设 $G_i(x, y)$ 表示第 i 层高斯金字塔图像, $G_0(x, y)$ 表示最底层的图像, 则构造第 i 层高斯金字塔的过程为: 首先将第 $i-1$ 层图像与低通滤波函数 $w(m, n)$ 进行卷积, 然后, 对卷积后的结果图像进行隔行隔列下采样, 具体公式为

$$G_i(x, y) = \sum_{m=-2n=-2}^2 \sum_{n=-2n=-2}^2 w(m, n) G_{i-1}(2x+m, 2y+n), \quad 0 < i \leq J, \quad 0 \leq x \leq r, \quad 0 \leq y \leq c. \tag{3}$$

式(3)中: J 是高斯金字塔的层数; r 和 c 分别为第 i 层金字塔图像的行数和列数; $w(m, n)$ 是一个 $m \times n$ 的低通模板, $m \times n$ 通常取 5×5 .

1.3 DoG 边缘

常见的边缘检测算子有 Sobel 算子, Laplace 算子, Canny 算子等^[19], 文中采用的是高斯差分算子. 基于高斯差分算子(DoG)边缘检测^[20] 的主要思想是用两个不同参数的高斯函数的差对图像做卷积.

首先分别用核宽度为 δ_1 和 δ_2 的两个高斯核函数 $G_{\delta_1}(x, y)$ 和 $G_{\delta_2}(x, y)$ 来对图像 $l(x, y)$ 做卷积, 得到两幅平滑后的图像, 即

$$g_i(x, y) = G_{\delta_i}(x, y) * l(x, y), \quad i = 1, 2, \tag{4}$$

$$G_{\delta_i}(x, y) = \frac{1}{\sqrt{2\pi}\delta_i} \exp\left[-\frac{x^2 + y^2}{2\delta_i^2}\right]. \tag{5}$$

检测到的边缘图像为

$$g(x, y) = g_2(x, y) - g_1(x, y) = G_{\delta_2}(x, y) * (x, y) - G_{\delta_1}(x, y) * (x, y) = \text{DoG}^*(x, y). \tag{6}$$

参数 δ_1 和 δ_2 决定了边缘检测的空间尺度, 它们的取值越大, 则检测到的边缘曲线就会越粗糙, 然而噪声点会越少. 实验取 $\delta_1=1.2, \delta_2=1.92$. DoG 边缘检测流程, 如图 1 所示. 由图 1 可知: 得到的边缘图像很不明显. 为了得到一个更加明显、平滑的边缘, 增强自适应性, 将得到的边缘图像 $g(x, y)$ 乘以 n , 得到一幅边缘增强的图像, 如图 2 所示. 图 2 中: p 为边缘图像中像素的最大值; $n=255/p$. 边缘图像的

取值范围被扩展到了 $[0, 255]$, 再对该图像进行一次中值滤波去除一些噪声, 使边缘变得更加的平滑。



(a) $g_1(x, y)$ (b) $g_2(x, y)$ (c) DoG 边缘检测结果

图 1 DoG 边缘检测流程图

Fig. 1 Flow chart of edge detection by DoG



图 2 自适应增强后的 DoG 边缘图像

Fig. 2 Edge image of DoG after adaptive enhancement

1.4 色彩量化

色彩量化^[10]可以减少图像中不同颜色的数量, 为了使生成的图像具有类似卡通画的效果, 对图像进行色彩量化, 即

$$Q(x, q, \phi_q) = q_{\text{nearest}} + \frac{\Delta q}{2} \tanh(\phi_q \cdot (l(x) - q_{\text{nearest}})). \quad (7)$$

式(7)中: Q 是量化后的图像; Δq 是相邻色阶之间的宽度; q_{nearest} 是最接近 $l(x)$ 的那个色阶; ϕ_q 为锐化参数, 是一个色阶到另一个色阶传递的控制锐化参数。

如将图像的 L 通道的亮度分为 5 个范围, 则 $\Delta q = 20$, q_{nearest} 的取值为 0, 20, 40, 60, 80, 100. 如果 ϕ_q 为一个固定的值, 则可能会在图像的大平滑区域造成许多明显的变化. 为了降低色阶在传递过程中的变化程度, 将 ϕ_q 定义为图像亮度的梯度函数。

Winnemöller 等^[10]定义了一个梯度范围 $[\Lambda_\delta, \Omega_\delta]$ 和一个锐化范围 $[\Lambda_\phi, \Omega_\phi]$. 首先计算像素的亮度梯度, 然后将梯度规范化到梯度范围 $[\Lambda_\delta, \Omega_\delta]$, 再通过线性变换得到一个在锐化范围 $[\Lambda_\phi, \Omega_\phi]$ 内的锐化参数, 具体的变换公式为

$$\phi_q = \text{grad} \frac{\Omega_\phi - \Lambda_\phi}{\Omega_\delta - \Lambda_\delta} + \Lambda_\phi. \quad (8)$$

式(8)中: grad 为规范化到梯度范围 $[\Lambda_\delta, \Omega_\delta]$ 的梯度值, 取 $q \in [8, 10]$, 梯度范围为 $[\Lambda_\delta = 0, \Omega_\delta = 2]$, 锐化范围为 $[\Lambda_\phi = 0.3, \Omega_\phi = 1.4]$.

1.5 基于图像梯度的边缘融合

以往的卡通化渲染方法中, 最后得到的卡通化图像通常采用的是黑色边缘, 这样在有些地方会显得边缘不够连续, 且噪声比较明显. 针对这些问题, 提出了一种新的边缘融合方法, 即根据边缘处梯度的大小来设置其边缘颜色的深浅。

具体实现方法为: 在得到增强的 DoG 边缘灰度图像之后, 用色彩量化后的图像减去边缘图像, 即可得到最终的卡通效果图. 由于梯度大的边缘处经过增强后其灰度值比较大, 相减之后最终图像上相应位置的灰度值比较小, 故其颜色为黑色或接近黑色; 而其他梯度比较小的边缘的灰度值相对比较小, 相减之后边缘的颜色与原始图像相近, 但比原始图像的颜色更深. 在增强对比度的同时, 也使得图像更加自然. 在边缘图像中的非边缘部分, 由于其灰度值为 0, 所以相减后还是色彩量化后的图像的像素值。

1.6 流程设计

卡通化渲染流程, 如图 3 所示. 首先对原始的彩色图像建立高斯图像金字塔, 接着对塔顶的图像进行均值漂移滤波, 并对结果图像进行上采样. 具体方法为: 首先将结果图像的行和列都扩大 2 倍, 将结果图像中位置为 (x, y) 的像素映射到扩大后的图像的 $(2x+1, 2y+1)$ 位置, 其他位置插入 0; 然后对得到的图像用指定的滤波器进行高斯卷积, 其中滤波器乘以 4 作为插值; 对下一层图像中偶数行或偶数列位置上与上采样得到的图像相应像素颜色值相差超过一定值的像素进行均值漂移滤波, 重复该过程, 直到对塔底图像完成均值漂移滤波, 然后对均值漂移滤波的结果图进行 DoG 边缘检测以及色彩量化, 最后用色彩量化后的图像减去边缘图像来进行边缘融合。

对视频的卡通风格化渲染过程, 如图 4 所示. 由图 4 可知: 首先将输入的视频逐帧分解成视频图像;

然后对每一帧视频图像采用上述流程进行卡通化处理;最后将处理好的每一帧卡通化的图像写入到新的视频文件中,其中新视频文件的帧数和帧速率与原始视频文件相同. 本实验是在 VS2010 平台上利用 OpenCV 动态函数库实现的,视频的输入有两种形式:一种是通过摄像头直接的实时获取视频;另一种是输入 AVI 格式的视频文件.

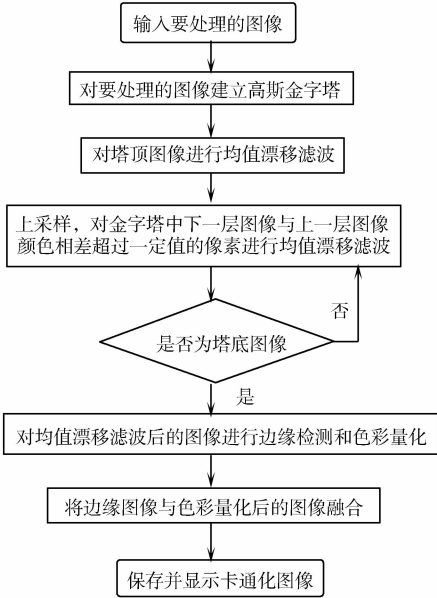


图 3 卡通化流程图

Fig. 3 Flow chart of

the cartoon-like stylization

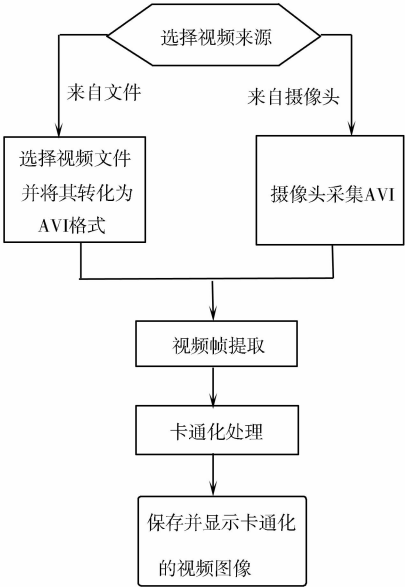


图 4 视频卡通化处理预定义过程

Fig. 4 Predefined process of

the cartoon-like stylization for video

2 实验及结果分析

结合不同层数高斯金字塔的最终卡通化效果,如图 5 所示. 由图 5 可知:随着金字塔层数的增加,越来越多的细节被平滑掉.



(a) 2 层金字塔

(b) 3 层金字塔

(c) 4 层金字塔

(d) 5 层金字塔

图 5 结合不同层数高斯金字塔的卡通化效果

Fig. 5 Cartoon-like stylization effects combined with different layers of gaussian pyramid

从图 5(a)可知:采用 2 层高斯金字塔会有一些噪声无法去除掉(如右边的墙壁). 由图 5(b)可知:采用 3 层高斯金字塔,右边墙壁上的大部分噪声被平滑掉了,但是其他地方如人物背部衣服的纹理也有部分被消除了. 由图 5(c)可知:采用 4 层高斯金字塔,图像中人物背部衣服以及地板的纹理几乎全部被消除了. 由图 5(d)可知:采用 5 层高斯金字塔,相比图 5(c)椅子下半部分已经被平滑的看不清了.

结合不同层数 (n) 的高斯金字塔进行均值漂移滤波所花的时间,如图 6 所示. 图 6 中:横坐标表示高斯金字塔的层数,1 层表示原始图像即没有用金字塔;5 条线代表 5 幅不同大小的图像;最上面的那条线所代表的图像尺寸最大为 1 024 px \times 768 px. 由图 6 可知:对于不同大小的图像随着金字塔层数的增加其处理所花费的时间都会减小,但基本上从第 3 层开始,再增加其层数处理时间也不会减少. 综合考虑卡通化的效果和处理速度,本实验选择高斯金字塔的层数为 3.

Winnemöller 方法^[10]和文中的方法在图像卡通化的效果上的比较,如图 7 所示. 由图 7(b)和图 7(c)可知:采用文中提出的基于高斯差分算子进行的边缘检测,由于对边缘进行了中值滤波,所以得到的

边缘更加完整、平滑, 显著性效果更强.

彩色视频图像卡通风格化渲染算法执行时间的比较, 如图 8 所示. 图 8 中: 横坐标为 4 种视频图像的大小; 纵坐标为连续 15 帧视频图像卡通风格化渲染所消耗的时间的平均值. 由图 8 可知: Winnemöller 等提出的双边滤波虽然在图像尺寸大于 $640\text{ px}\times 480\text{ px}$ 时比均值漂移滤波快, 但是和文中提出的引入图像金字塔的均值漂移滤波相比速度要慢. 文中提出的卡通化渲染算法的处理速度相比另外两种方法都要快很多, 对于一个 $1\,920\text{ px}\times 1\,080\text{ px}$ 的高清视频图像, 文中提出的结合高斯金字塔的均值漂移滤波平均可以在 0.8 s 之内完成; 对于一个 $640\text{ px}\times 480\text{ px}$ 的视频的每一帧图像, 其平均处理时间不超过 0.2 s .



(a) 原图 (b) Winnemöller 方法 (c) 文中方法

图 7 图像卡通化效果比较

Fig. 7 Cartoon-like stylization results comparison

3 结论

卡通化渲染技术将图像金字塔和均值漂移滤波结合起来, 与 Winnemöller 等所用的双边滤波和 DoG 算子相比, 获得了更好的卡通风格化效果和更快的运行速度. 文中对 DoG 算子检测到的图像进行了中值滤波和灰度范围扩展, 可以检测到更加完整的、平滑的边缘线条; 将图像金字塔运用到均值漂移滤波中, 大大提高了算法运行的速度.

在最后边缘与图像融合的时候, 本算法没有采用传统的单一的黑色线条, 而是根据边缘处梯度的大小来设置其颜色的深浅, 不仅增强了对比度, 而且使图像的卡通化效果有另一种的选择. 此外, 文中的方案可以直接应用到对视频图像的处理, 可以直接将一部视频渲染成卡通风格的视频后输出, 后期可采用 GPU 加速技术对文中提出的方法进一步优化.

文中方法的缺点是相同物体在处理后的不同帧之间的可能存在一些差异, 后期将考虑采用直方图匹配等方法来缩小不同帧中同一物体之间的差异.

参考文献:

[1] KNOWLTON K C. A computer technique for producing animated movies[C]// Proceedings of the 1964 Spring Joint Computer Conference. Washington D C: ACM, 1964: 67-87.

[2] 于金辉, 徐晓刚, 彭群生. 计算机生成卡通烟雾动画[J]. 计算机学报, 2000, 23(9): 987-990.

[3] 于金辉, 徐晓刚, 彭群生. 用随机正弦波拟合卡通流水[J]. 计算机研究与发展, 2001, 38(5): 519-523.

[4] de CARLO D, SANTELLA A. Stylization and abstraction of photographs[J]. ACM Transactions on Graphics, 2002, 21(3): 769-776.

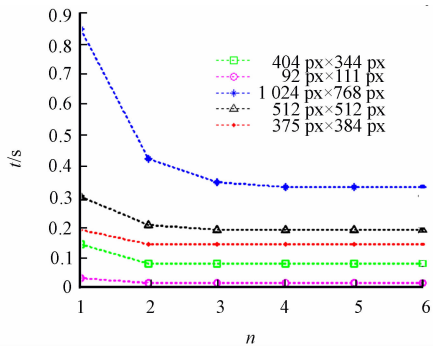


图 6 进行均值漂移滤波所花的时间

Fig. 6 Time consumption of Mean-shift filter

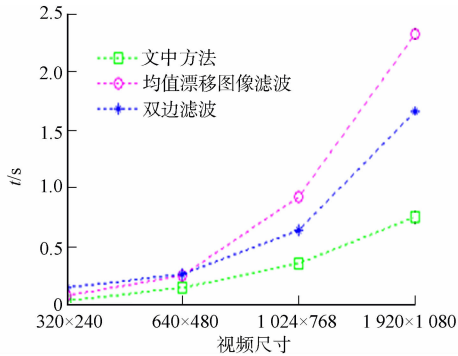


图 8 卡通风格化渲染算法执行时间比较

Fig. 8 Cartoon-like stylization time consumption comparison

- [5] WANG Jue, XU Ying-qing, SHUM H Y, et al. Video tooning[J]. ACM Transactions on Graphics, 2004, 23(3): 574-583.
- [6] COLLOMOSSE J P, ROWNTREE D, HALL P M. Stroke surfaces: Temporally coherent artistic animations from video[J]. IEEE Transactions on Visualization and Computer Graphics, 2005, 11(5): 540-549.
- [7] 周冲. 二维彩色图像的卡通风格实现[D]. 大连: 大连理工大学, 2005: 28-53.
- [8] 高宝玉. 二维彩色图像卡通风格化方法研究[D]. 青岛: 青岛大学, 2010: 7-47.
- [9] 郭纳. 图像的卡通风格渲染技术[D]. 青岛: 青岛大学, 2009: 16-38.
- [10] WINNEMÖLLE H, OLSEN S C, GOOE H B. Real-time video abstraction[J]. ACM Transactions on Graphics, 2006, 25(3): 1221-1226.
- [11] TOMASI C, MANDUCHI R. Bilateral filtering for gray and color images[C]// Proceedings of the 1998 IEEE International Conference on Computer Vision, Bombay: IEEE Press, 1998: 839-846.
- [12] SHAHCHERAGHI Z, SEE J. On the effects of pre-and post-processing in video cartoonization with bilateral filters [C]// 2013 IEEE International Conference on Signal and Image Processing Applications (ICSIPA), Melaka: IEEE Press, 2013: 37-42.
- [13] LU Li-wen, PU Yuan-yuan, ZHANG Heng, et al. A non-photorealistic rendering algorithm for cartoons[C]// 2013 6th International Congress on Image and Signal Processing (CISP), Hangzhou: IEEE Press, 2013: 680-685.
- [14] ROSIN P, COLLOMOSSE J. Image and video-based artistic stylization[M]. London: Springer-Verlag, 2013: 77-91.
- [15] COMANICIU D, MEER P. Mean shift: A robust approach toward feature space analysis[J]. IEEE Transaction on Pattern Analysis and Machine Intelligence, 2002, 24(5): 603-619.
- [16] KUWAHARA M, HACHIMURA K, EIHO S, et al. Processing of RI-angiocardigraphic images[M]. New York: Springer US, 1976: 187-203.
- [17] PAPARI G, PETKOV N, CAMPISI P. Artistic edge and corner enhancing smoothing[J]. IEEE Transactions on Image Processing, 2007, 16(10): 2449-2462.
- [18] BURT P J, ADELSON E H. The Laplacian pyramid as a compact image code[J]. IEEE Transactions on Communications, 1983, 31(4): 532-540.
- [19] 李娅娅, 李志浩, 郑海旭, 等. 图像边缘检测算法的比较与实现[J]. 计算机工程与设计, 2010, 31(9): 1971-1975.
- [20] WINNEMÖLLER H, KYPRIANIDIS J E, OLSEN S C. XDoG: An extended difference-of-gaussians compendium including advanced image stylization[J]. Computers and Graphics, 2012, 36(6): 740-753.

Research and Implementation of Cartoon-Like Stylization for Color Video Image

WANG Qiao-yu, CHEN Duan-sheng, WU Yang-yang

(College of Computer Science and Technology, Huaqiao University, Xiamen 361021, China)

Abstract: A new approach for cartoon-like stylization of color video image is proposed. The Mean-shift filter and gaussian pyramid are combined in this approach to cluster the color and eliminate the noise of the color video image and improve the speed of processing at the same time. Then, the DoG is used to detect edges of the image which has been filtered by Mean-shift filter. After that, the soft quantization is applied to the image which has been filtered by Mean-shift filter. Finally, the edge curves and the quantitative image are fused to generate a personalized cartoon image. In this step, the color of edges in the cartoon image is based on the edge gradient. This makes the cartoon image not only be enhanced in the regions of high contrast, but also have less noise and seems more natural. In comparison with the approach for cartoon-like stylization of image based on the bilateral filter and DoG, the experimental results show that our approach can produce a better cartoon effect in shorter time, and can directly convert an input video to a cartoonish video as an output file.

Keywords: cartoon-like image; Mean-shift filter; gaussian pyramid; color quantization

(责任编辑: 陈志贤)

英文审校: 吴逢铁)