

多尺度并行坐标插值实时图像克隆算法

沈晔湖 蒋全胜 汪帮富 朱其新

(苏州科技大学机械工程学院, 苏州, 215009)

摘要: 作为一种重要的数字图像合成编辑技术, 图像克隆能够将源图像中的克隆区域平滑自然地嵌入目标图像的指定位置。传统的图像克隆技术通过分析图像的梯度域并求解泊松方程来获取图像信息, 具有算法复杂度高、内存消耗大的缺点, 从而限制了其在高分辨率图像中的实时应用。本文提出了一种改进的均值坐标计算方法, 将图像克隆转化为高效且易于实现的图像插值流程。此外本文还采用多尺度和GPGPU并行计算技术进一步提升了算法的整体运算效率, 从而能够实现对100万像素图像区域的实时克隆。

关键词: 图像克隆; 均值坐标; 并行加速; 图像金字塔

中图分类号: TP391.41 **文献标志码:** A

Real Time Image Cloning Based on Multi-scale Parallel Coordinates Interpolation

Shen Yehu, Jiang Quansheng, Wang Bangfu, Zhu Qixin

(College of Mechanical Engineering, Suzhou University of Science and Technology, Suzhou, 215009, China)

Abstract: As an important digital image manipulation technique, image cloning can naturally and smoothly embed the cloned region in the source image into the specified region in the target image. In traditional image cloning technique, image information can be obtained by analyzing the gradient domains and solving Poisson equations. Its shortcomings of great algorithm complexity and large memory consumption limit the real application of traditional algorithm in high-resolution image. An improved mean-value coordinates algorithm is proposed in this paper, which transforms image cloning into efficient image interpolation procedure easy to implement. Moreover, the multi-scale and GPGPU based parallel computation technique is applied to further improve the overall operation efficiency of the proposed algorithm. Experimental results show that the proposed algorithm can realize the real-time cloning for the 1 mega pixel image region.

Key words: image cloning; mean-value coordinates; parallel acceleration; image pyramid

引 言

作为一种图像编辑技术, 图像克隆能够将源图像中的克隆区域嵌入目标图像的指定位置, 并且实现该区域与目标图像背景的自然融合, 使用户在观察目标图像时不会产生明显的突兀感。

对人类视觉系统的研究表明: 人类视觉系统对场景局部亮度差异的敏感度要远高于亮度的绝对值本身^[1]。基于该理论, 传统的图像克隆技术在保持克隆区域内部颜色梯度分布的前提下保证克隆区域

的边缘与目标图像对应位置颜色的相似性,这一般通过求解泊松方程来实现。Perez等^[2]证明求解泊松方程等价于求解包含狄利克雷边界条件的拉普拉斯方程,该边界条件设置为克隆区域边界在源图像和目标图像处的色彩差异。上述问题最终可通过求解大规模稀疏线性系统来得到。Adobe公司从Photoshop 7开始支持类似于图像克隆效果的修复笔刷工具。该笔刷工具通过求解四阶偏微分方程的手段保证克隆区域内部梯度值的连续性^[3]。Agarwala等^[4]提出将多张图像中的内容融合的算法,采用Graph-cut优化技术选择不同图像中的最优融合边界,图像融合仍通过求解泊松方程来实现。Jia等^[5]发现泊松图像编辑的效果依赖于克隆图像区域的边界,因此提出了采用最短闭合路径算法来提取最优的克隆区域边界。

求解泊松方程的运算复杂度和内存消耗均较高。为了克服上述缺陷,Agarwala^[6]利用自适应四叉树细分技术构建了约简的线性系统进行求解,该方法仅可以用于加速特定问题如图像拼接领域泊松方程的求解。McCann和Pollard^[7]提出了基于GPU的多重网格泊松求解器,可以实现快速的图像克隆,然而该算法的具体实现难度较大且运行时对显存的需求较高,作者在文中指出当算法对内存的需求超过显存容量时,算法性能会大幅度下降。

除了上述复杂的求解泊松方程的方案,Farbman等^[8]发现泊松图像克隆的本质是构建一个调和插值从而将区域边界的不一致性扩散到区域内部。他们利用均值坐标技术直接通过图像插值产生平滑的内部区域,从而避免了求解泊松方程。由于该方法利用了图像插值技术,所以算法实现非常容易。但是该方法插值系数的构造计算量较大,导致其运算复杂度很高。为此,作者提出了通过自适应网格等预处理方法,从而在一定程度上降低了图像插值阶段的运算复杂度。但预处理手段的引入导致图像克隆整体过程无法实现实时运算,而且预处理步骤较复杂,使得该算法实现的难度大幅上升。

近年来,图像克隆研究主要集中在改善图像克隆视觉效果方面。如Tanaka等^[9]通过一种改进的泊松方程实现了在克隆和背景区域颜色差异较大情况下的平缓过渡;李贝等^[10]在均值坐标技术基础上提出了利用局部亲和传播算法改善图像克隆中的颜色不协调问题;黄美玉等^[11]通过构建光照一致的数据约束和马尔科夫随机场函数进行求解的方式解决了源图像和目标图像光照环境不一致的问题;文献^[12]通过引入颜色不变量模型得到图像光照判断参数,进而对克隆选区进行自适应的颜色匹配。上述算法运算复杂度均较高,处理100万像素的图像克隆区域的时间均在秒级。

鉴于目前较少有针对图像克隆算法的简易实时化研究,本文基于文献^[8],提出了一种改进的均值坐标权重系数的计算策略,大幅降低权重系数的计算复杂度,避免了实现较复杂的预处理步骤。此外结合多尺度图像处理技术以及并行加速技术进一步实现加速,从而在当前主流GPU上实现了对100万像素图像区域的实时克隆。定性和定量实验结果均表明本文提出的方法得到的结果与文献^[8]十分接近。

1 均值坐标图像克隆

1.1 均值坐标简介

为保证本文的可读性,首先简要介绍2D均值坐标的基本概念,详细内容参见文献^[13]。

如图1所示,对于2D图像内的一个区域 P ,定义该区域的边缘为 ∂P ,包含 m 个像素: p_0, p_1, \dots, p_{m-1} , $p_i \in \mathbb{R}^2$ 。为不失一般性,定义: $p_m = p_0$ 。从而区域 P 内的像素 x 相对于 ∂P 的均值坐标可定义为

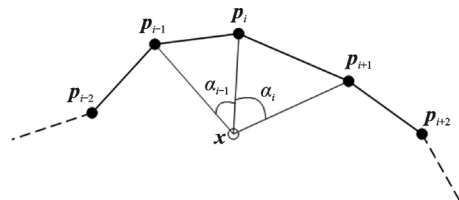


图1 图像均值坐标的定义

Fig. 1 Definition of mean-value coordinates

$$\lambda_i(x) = \frac{\omega_i}{\sum_{j=0}^{m-1} \omega_j} \quad i = 0, \dots, m-1 \quad (1)$$

式中

$$w_i = \frac{\tan(\alpha_{i-1}/2) + \tan(\alpha_i/2)}{\|p_i - x\|} \quad (2)$$

式中 $\|\cdot\|$ 表示L2范数。

1.2 均值坐标图像克隆算法流程

实现图像克隆算法的基本流程如下^[8]

步骤1 计算均值坐标。对于区域 P 内的每一个像素 x ,根据式(1,2)计算其相对于 ∂P 的均值坐标并保存。

步骤2 计算区域边缘处的前背景色彩差。对于 ∂P 中的每一个像素 p_i ,通过式(3)计算前背景色彩差并保存。

$$\text{diff}_i = I_t(p_i) - I_s(p_i) \quad (3)$$

式中 $I_t(p_i)$ 和 $I_s(p_i)$ 分别为 p_i 处目标图像和源图像的RGB值组成的矢量。

步骤3 对于每一个区域 P 内的像素 x ,计算其均值坐标插值结果。首先利用下式计算颜色校正矢量

$$r(x) = \sum_{i=0}^{m-1} \lambda_i(x) \cdot \text{diff}_i \quad (4)$$

像素 x 的最终颜色值的计算方法为

$$I_c(x) = I_s(x) + r(x) \quad (5)$$

2 本文方法

图2给出了本文算法的总体流程图。对于1.2节的算法,其运算复杂度最高的部分是步骤1。假设区域 P 内的像素个数为 n ,其边缘 ∂P 包含 m 个像素,通常 $n \gg m$ 。步骤1中调用式(2)计算次数的数量级为 $O(nm)$ 。但是式(2)中需要计算两个角的正切值,而且两个角度本身还需通过余弦定理计算得到,因此复杂度较高,造成算法无法实现实时处理。为此,文献[8]提出利用预处理算法减少 n 和 m ,从而降低式(1)的运算复杂度以及调用式(2)的次数。但是为了减少 n 和 m 引入的预处理算法本身复杂度不低,文献[8]中报告对于100万像素的克隆区域的预处理时间在1s以上,因此导致图像克隆算法整体无法实现实时处理。

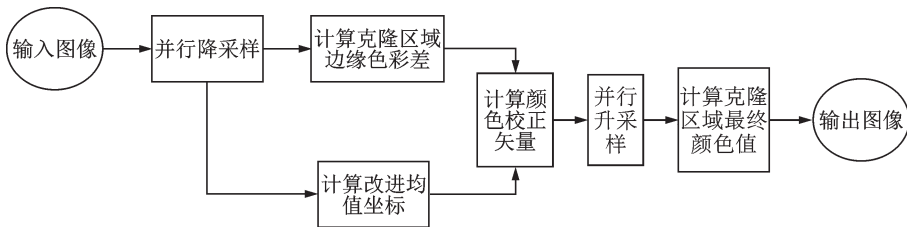


图2 算法总体流程图

Fig. 2 Flowchart of the proposed algorithm

2.1 改进的均值坐标

与文献[8]不同,本文提出的算法通过合理的近似技术直接简化式(2)的计算,以降低整体的运算复杂度。

根据麦克劳林公式,当 x 接近0时对 $\tan(x)$ 进行级数展开并取一阶近似可得到

$$\tan(x) = x + O(x) \quad (6)$$

将式(6)代入式(2)可得

$$w_i \approx \frac{\alpha_{i-1} + \alpha_i}{2 \cdot \|p_i - x\|} \quad (7)$$

为不失一般性,假设 ∂P 是由4-连通的像素组成,因此对于 ∂P 上的任意两个相邻像素 p_i 和 p_{i-1} ,它们之间的距离恒为1。可以验证当区域 P 内的像素 x 与像素 p_i 和 p_{i-1} 之间距离在5个像素以上时,角度 α_i 接近0,上述近似的误差约为0.33%。因此对于区域 P 内的绝大多数像素 x ,式(7)是对式(2)的较优近似。

式(7)仍需要计算两个角度 α_{i-1} 和 α_i ,为了进一步简化式(7)的计算,本文提出了如下算法。如图3所示,顶点 p_{i-1} , p_i 和 x 组成三角形, $\alpha_{i-1} = \angle p_i x p_{i-1}$,虚线 xq 为 α_{i-1} 的角平分线,它与边 $p_{i-1}p_i$ 的交点为 p' 。圆弧 p_iq 是以 x 为圆心, $\|p_i - x\|$ 为半径的圆弧段。根据圆弧的定义,有

$$\frac{\alpha_{i-1}}{2} = \frac{\|p_i q\|_a}{\|p_i - x\|} \quad (8)$$

式中: $\|p_i q\|_a$ 为圆弧 p_iq 的长度。如前所述, $\|p_i - p_{i-1}\| = 1$ 并且 $\alpha_{i-1}/2$ 接近于0,因此可以将 $\|p_i q\|_a$ 近似为 $\|p_i - p_{i-1}\|$ 的一半也即0.5,从而式(8)可以近似为(请参考附录A关于该近似的可行性验证)

$$\frac{\alpha_{i-1}}{2} \approx \frac{1}{2\|p_i - x\|} \quad (9)$$

类似地可以得到

$$\frac{\alpha_i}{2} \approx \frac{1}{2\|p_i - x\|} \quad (10)$$

将式(9,10)代入式(7)可得

$$w_i \approx \frac{1}{\|p_i - x\|^2} \quad (11)$$

比较式(11)和式(2)可知式(11)只需计算 p_i 和 x 的距离平方,无需计算任何角度,因此比式(2)节约了2次角度计算、2次正切计算和1次开方计算,从而实现了大幅度的简化。

由于最终的克隆区域颜色校正结果可根据式(4)均值坐标相对权重的线性组合来得到,而通常情况下边缘 ∂P 包含的像素个数 m 在 10^3 数量级,因此通过式(11)计算得到的少量权重误差通过上述线性组合的平均效应后不会对最终的校正结果产生大的影响。这将在实验结果部分作详细验证。

2.2 多尺度并行加速

根据式(5),克隆区域内像素的最终颜色值取决于颜色校正矢量 $r(x)$ 。 $r(x)$ 是克隆区域边缘像素处颜色差的线性组合。根据式(1,4,11),在克隆区域内部 $r(x)$ 是平缓变化的,受文献[14]的启发,本文提出采用多尺度并行的手段进一步加速计算。首先将源图像和目标图像降采样^[15];其次对降采样的克隆区域用并行加速手段计算每个像素的颜色校正矢量 $r(x)$;然后将克隆区域对应的颜色校正矢量 $r(x)$ 进行升采样^[15],使得分辨率达到初始状态;最后利用 $r(x)$ 计算克隆区域内像素 x 的最终颜色值。其具体步骤如下:

步骤1 利用并行双线性插值算法^[16]对 I_t 和 I_s 进行降采样,降采样的尺度为 s ,本文中 $s = 1/2$,得到

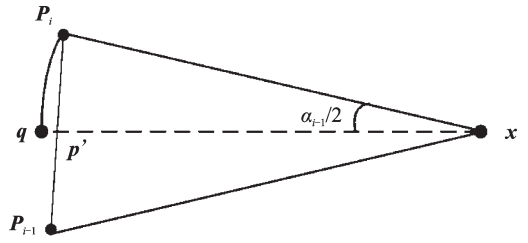


图3 均值坐标的简化计算

Fig. 3 Simplified calculations of mean-value coordinates

降采样后的图像 I_t^d 和 I_s^d 。

步骤 2 利用并行加速手段计算降采样图像对应的颜色校正矢量 $r_d(x)$ 。该步骤又可分为如下两步:

(1) 根据式(3)计算克隆区域边缘处前背景色彩差并保存;

(2) 通过 GPU 并行加速技术^[17]计算降采样的克隆区域 P_d 内部每个像素 x 对应的颜色校正矢量 $r_d(x)$, 具体算法如本节后面描述。

步骤 3 利用并行双线性插值算法^[16]对 $r_d(x)$ 进行升采样, 升采样的尺度为 $1/s$, 得到初始分辨率下的颜色校正矢量 $r(x)$ 。

步骤 4 计算克隆区域内每个像素 x 的最终颜色值。

对于步骤 2 的第 2 步, 可将每个线程设置为利用式(5)计算每个像素 x 对应的颜色校正矢量 $r_d(x)$, 算法步骤如下:

输入: 区域的边缘像素集合 ∂P_d , 区域内部像素集合 P_{in}^d , 区域边缘前背景色彩差集合 $diff_d$ 。

输出: 颜色校正矢量 $r_d(x)$ 。

(1) 计算线程 ID 号 i ;

(2) 提取第 i 个区域内部像素 $x = P_{in}^d(i)$;

(3) 对于 ∂P_d 中的每个像素 p_j , 利用式(11)计算 w_j 并保存;

(4) 根据式(1)计算 x 对应的均值坐标并保存;

(5) 根据式(4)计算颜色校正矢量 $r_d(x)$ 。

综合上述算法描述, 其伪代码如下。

输入: 源图像 I_s 、目标图像 I_t 、克隆区域内部像素掩模图像 P_{in}

输出: 克隆图像 I_c

```

 $I_s^d = \text{downsample}(I_s, s)$       #利用文献[16]算法对  $I_s$  进行并行降采样, 降采样的尺度为  $s$ 
 $I_t^d = \text{downsample}(I_t, s)$       #利用文献[16]算法对  $I_t$  进行并行降采样, 降采样的尺度为  $s$ 
 $P_{in}^d = \text{downsample}(P_{in}, s)$     #利用文献[16]算法对  $P_{in}$  进行并行降采样, 降采样的尺度为  $s$ 
for  $i = 1$  to  $m_d$ 
     $diff_i = I_t^d(p_i) - I_s^d(p_i)$     #根据式(3)计算降采样后克隆区域边缘处前背景色彩差并保存
end for
parallel for  $i = 1$  to  $n_d$            #对每个像素并行处理
     $x = P_{in}^d(i)$                    #提取第  $i$  个克隆区域内部像素
    sum_w = 0
    for  $j = 1$  to  $m_d$ 
         $w_j = \frac{1}{\|p_j - x\|_2}$       #对于  $\partial P_d$  中的每个像素  $p_j$ , 利用式(11)计算  $w_j$  并保存
        sum_w = sum_w +  $w_j$ 
    end for
     $r_d(x) = 0$                        #初始化为零向量
    for  $j = 1$  to  $m_d$ 
         $\lambda_j(x) = \frac{w_j}{sum\_w}$     #根据式(1)计算  $x$  对应的均值坐标并保存
         $r_d(x) = r_d(x) + \lambda_j(x) \cdot diff_j$     #根据式(4)计算颜色校正矢量  $r_d(x)$ 
    end for
end parallel

```

```

end for
end parallel for
r = upsample(ra, 1/s) #利用文献[16]算法对 ra 进行并行升采样,升采样的尺度为 1/s。
parallel for i = 1 to n #对每个像素并行处理。
    x = Pin(i) #提取第 i 个克隆区域内部像素。
    Ic(x) = Is(x) + r(x) #利用式(5)计算克隆图像 Ic 内部像素的颜色值。
end parallel for
    
```

3 实验结果

3.1 定性分析

本文构建了一个包含 20 组测试图像的数据集,每组测试图像包含源图像、目标图像、手工标记的源图像中需要克隆的区域和手工标记的目标图像克隆放置区域。

图 4—7 展示了 4 组测试图像的例子,比较图 4—6 中的 (d), (e) 以及 (f) 可知本文提出的算法和文献

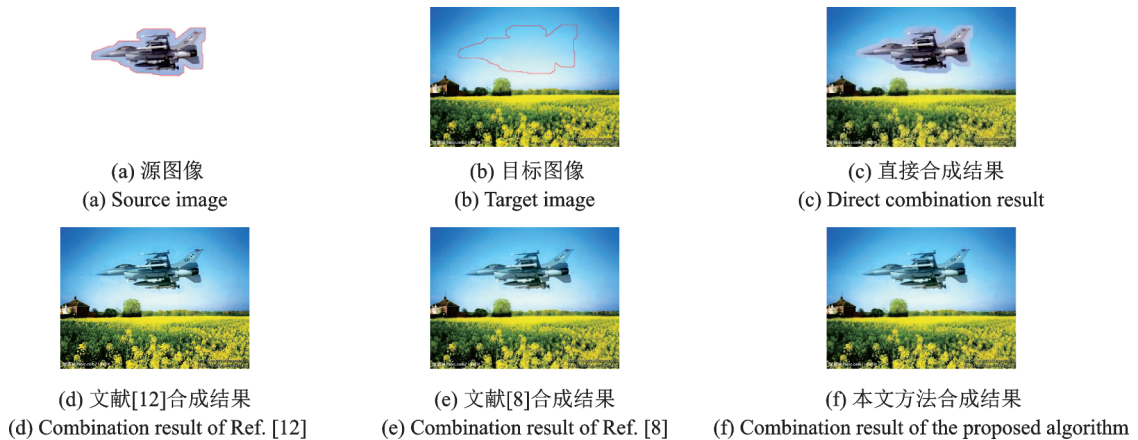


图 4 第 1 组测试图像实验结果

Fig. 4 Results of test example 1

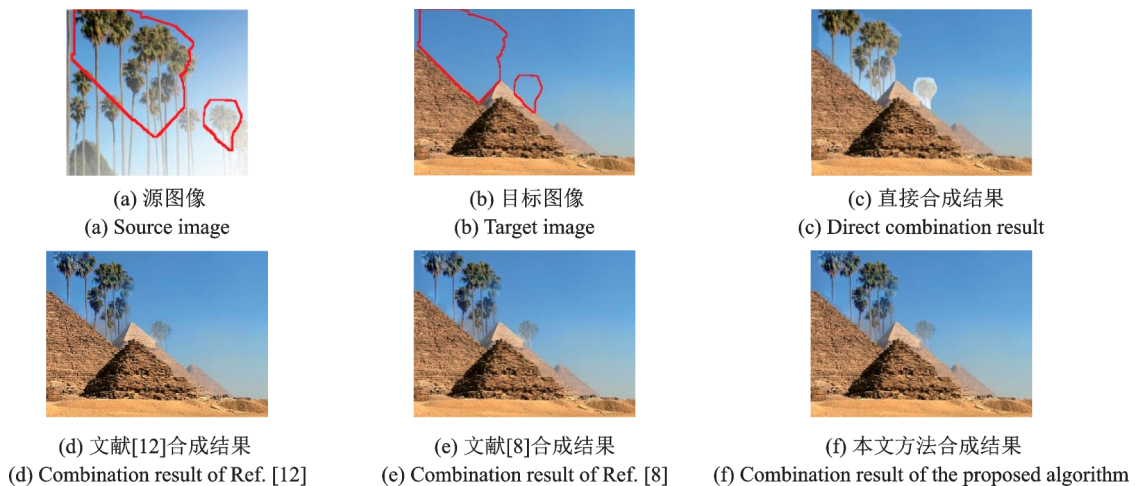


图 5 第 2 组测试图像实验结果

Fig. 5 Results of test example 2

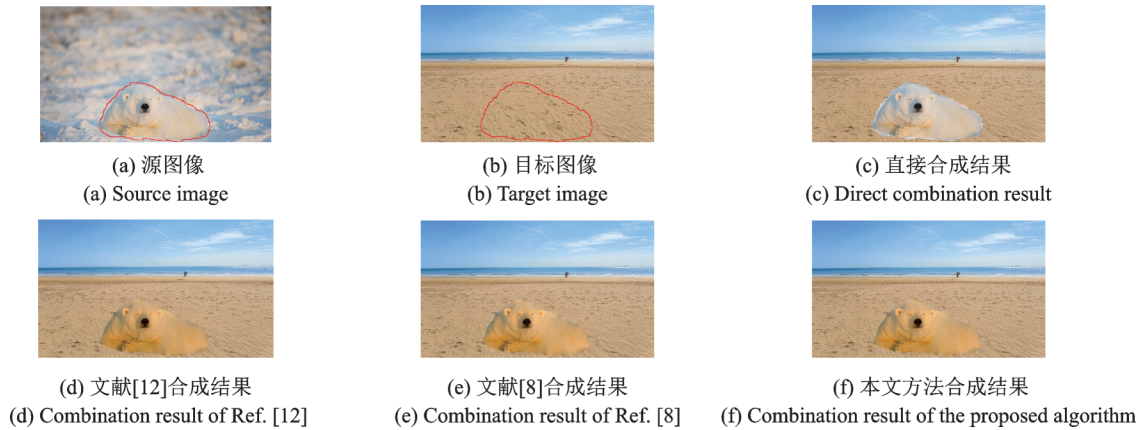


图6 第3组测试图像实验结果
Fig. 6 Results of test example 3

[12]以及文献[8]中的算法结果差别细微,肉眼几乎难以察觉。这也从定性角度验证了本文改进均值坐标方法的可行性。

图7展示了“变脸”特效,美国总统特朗普和俄罗斯总统普京的脸部区域互换后的效果自然,验证了本文算法可以实现逼真的视觉特效。

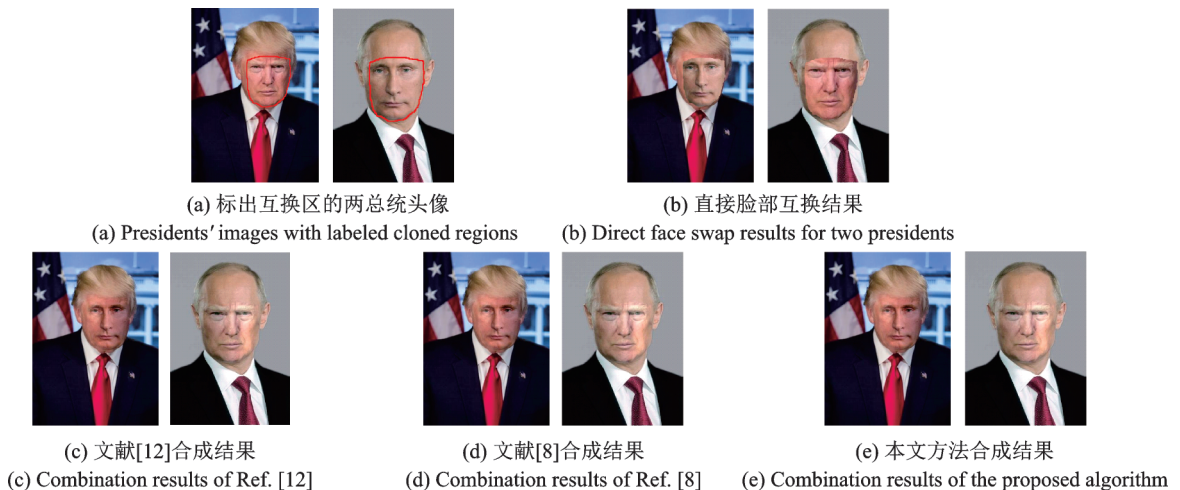


图7 “变脸”特效
Fig. 7 "Face/off" effect

3.2 客观评价

本节以文献[8]合成的克隆区域图像为基准,将本文提出算法合成的克隆区域图像与其进行比较,计算颜色误差。图8分别显示了第1组和第3组测试图像中的克隆区域颜色误差分布图像。在计算下述图像中每个像素的误差值时采用如下方式:用本文提出算法和文献[8]合成算法计算每个像素处的颜色误差,该颜色误差的取值为 R, G, B 三个分量中误差绝对值的最大值(每个像素三通道亮度级范围是 $0\sim 255$)。本文还统计了20组测试图像中克隆区域的平均颜色误差值分布统计直方图,结果如图9所示。

从图 8, 9 可以看到, 克隆区域内大部分像素和文献[8]算法的颜色误差在 2 个亮度级以内, 误差在 4 个亮度级以上的像素占比在 0.9% 以内。同时还计算了本文算法合成结果相对于文献[8]合成算法在克隆区域内的峰值信噪比(Peak signal to noise ratio, PSNR)值^[18], 结果如表 1 所示。从表 1 中的统计结果可知, 本文方法克隆结果和文献[8]克隆结果极为相似, 这也从定量角度印证了 3.1 节中的定性分析结果。

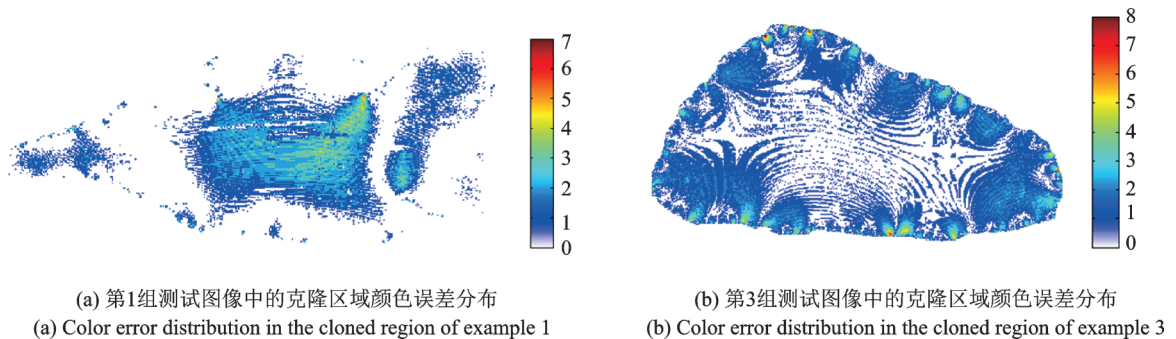


图 8 克隆区域内部的颜色误差分布图像

Fig. 8 Color error distribution images in the cloned region

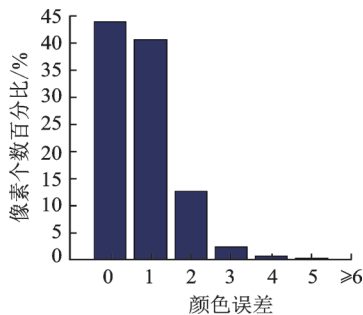


图 9 克隆区域内部平均颜色误差值分布统计直方图

Fig. 9 Color error distribution histogram in the cloned region

表 1 20 幅测试图像中克隆区域的 PSNR 值
Tab. 1 PSNR in cloned regions of twenty test images

PSNR			dB
PSNR	PSNR	PSNR	
均值	最大值	最小值	
45.11	46.97	43.29	

3.3 主观评价

为了对比文献[8]中算法与本文算法的主观视觉效果, 选取了 20 名被试者对 20 组测试图像分别用两种算法得到的结果进行评分, 评分范围是 1~10 分。20 名被试者男女各半, 年龄范围是 19~40 周岁, 涵盖了本科生、研究生以及学校教师, 并且都与本文项目无关。

实验中每次在计算机屏幕上左右分别显示一组测试图在文献[8]中算法与本文算法的结果。为了消除被试者打分的倾向性, 在每组测试结果显示过程中随机设定两种算法结果的左右显示位置。被试者分别对左图和右图的主观结果进行打分。

实验结果表明 14 人对 20 幅测试图像中两种算法的结果打分完全相同, 也就是说有 14 人主观评价认为两种方法克隆图像的效果完全一致, 另外 6 人的结果如表 2 所示。

由表 2 可知, 6 人中对两种算法结果的评价分最大只相差 1 分, 且两种方法在 20 幅测试图像中的平均得分非常接近, 这也从一个侧面表明文献[8]中算法与本文算法的结果非常类似。

表2 20幅测试图像中两种算法主观评分结果比较

Tab. 2 Comparisons of subjective scorins of two algorithms for twenty test images

被测者序号	文献[8]算法均分	本文算法均分	两种算法在20组测试图像中得分的最大差值
1	8.10	8.05	1
2	7.45	7.30	1
3	8.25	8.20	1
4	7.90	7.95	1
5	8.85	8.90	1
6	7.55	7.45	1

3.4 降采样尺度 s 的设定

在2.2节中,本文算法需要确定一个超参数降采样尺度 s ,为此本节实验选择了不同的 s 值,分别为1,1/2,1/4,1/8和1/16,对每组测试图像分别获得5张不同的克隆图像结果。与3.2节类似,计算并比较了20组测试图像在不同 s 值下的平均PSNR均值,结果如图10所示。

由图10可知当 s 取1/2时,PSNR均值仍在45 dB以上,这与不进行降采样(即 $s = 1$)的结果接近。但当 s 取1/4及以上时,PSNR均值有大幅衰减,已经影响到主观的评价结果。综合考虑图像克隆效果与运算复杂度,本文中所有实验均取 $s = 1/2$ 。

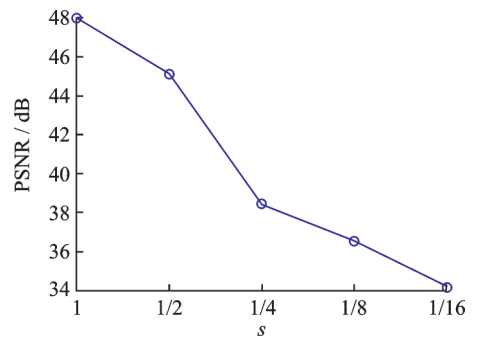


图10 不同 s 值时的 PSNR 均值比较

Fig. 10 Comparison of average PSNR under different values of s

3.5 执行时间分析

实验平台配置为 Intel 3.4 GHz i7-4770 CPU,显卡为 NVidia Geforce GTX 960,内存为 8 GB。本节中比较了文献[12]、文献[8]方法和本文提出算法的执行时间。由于文献[8,12]并未开源其代码,因此本文实现了其算法。所有算法均用 C++ 实现,其中 GPU 并行加速部分代码利用 CUDA 实现。

图11展示了平均的单次式(2)执行时间和本文简化后的单次式(11)在CPU上执行时间的对比。由图11可见,本文简化后的式(11)执行时间约为3.5 ns,从而达到了约30倍的速度提升,证明了本文算法对于简化均值坐标权重计算的有效性。

表3展示了不同图像克隆区域尺寸下本文算法运行时间和文献[8]以及文献[12]算法之间的比较。表中第3列和第4列分别为文献[12]和文献[8]算法实现图像克隆完整流程所需的时间,其中括号内的数字为整个流程中预处理算法的耗时,本文算法提出的算法无需任何预处理步骤。由表3可见本文方法实现了对100万像素的图像区域的实时克隆,即使当克隆区域增大至400万像素的高分辨率时仍能达到5帧/s左右的速度。

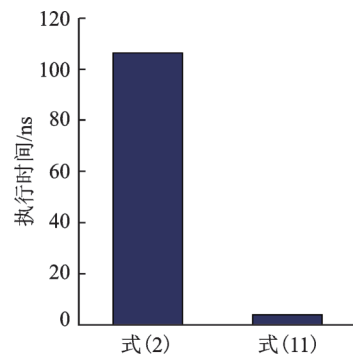


图11 式(2)和式(11)计算执行时间对比

Fig. 11 Comparisons of execution time for Eq.(2)and Eq.(11)

表3 不同算法的运算执行时间结果比较

Tab. 3 Comparison of execution time for different algorithms

克隆区域内部 像素个数	克隆区域边缘 像素个数	文献[12]算法执行 时间/ms	文献[8]算法执行 时间/ms	本文算法执行 时间/ms
260 100	2 044	551(147)	425(408)	3.1
556 516	2 988	912(409)	675(651)	9.5
1 044 484	4 092	2 032(890)	1 246(1 215)	24.2
2 076 481	5 768	3 998(1 574)	2 043(2 002)	68.9
4 186 116	8 188	7 856(2 876)	3 893(3 842)	196.8

4 结束语

本文分析了传统基于均值坐标的图像克隆算法的运算瓶颈,通过提出一种简化的均值坐标计算方案在无需任何预处理步骤的前提下大幅降低了均值坐标的运算复杂度,并进一步利用多尺度GPU并行加速方案实现了对100万像素图像区域的实时克隆。客观和主观实验结果表明本文算法在克隆结果与文献[8]中方法结果差别细微的前提下可以大幅提升高分辨率图像的克隆效率。

参考文献:

- [1] Palmer S. Vision science: Photons to phenomenology [M]. Cambridge: The MIT Press, 1999.
- [2] Perez P, Gangnet M, Blake A. Poisson image editing [J]. ACM Transactions on Graphics, 2003, 22(3): 313-318.
- [3] Georgiev T. Photoshop healing brush: A tool for seamless cloning[C]// Workshop on Applications of Computer Vision (ECCV 2004). Heidelberg: Springer, 2004: 1-8.
- [4] Agarwala A, Dontcheva M, Agrawala M, et al. Interactive digital photomontage [J]. ACM Transactions on Graphics, 2004, 23(3): 294-302.
- [5] Jia J, Sun J, Tang C, et al. Drag-and-drop pasting [J]. ACM Transactions on Graphics, 2006, 25(3): 631-637.
- [6] Agarwala A. Efficient gradient-domain compositing using quadtrees [J]. ACM Transactions on Graphics, 2007, 26(3): 94-98.
- [7] McCann J, Pollard N. Real-time gradient-domain painting [J]. ACM Transactions on Graphics, 2008, 27(3): 93-99.
- [8] Farbman Z, Hoffer G, Lipman Y, et al. Coordinates for instant image cloning [J]. ACM Transactions on Graphics, 2009, 28(3): 1-9.
- [9] Tanaka M, Kamio R, Okutomi M. Seamless image cloning by a closed form solution of a modified Poisson problem [C]// Proceedings of 5th ACM Conference and Exhibition on Computer Graphics and Interactive Techniques in Asia. New York: ACM, 2012.
- [10] 李贝, 汪浩, 于佳骏, 等. 颜色协调的鲁棒式图像克隆[J]. 系统仿真学报, 2015, 27(4): 723-730.
Li Bei, Wang Hao, Yu Jiajun, et al. Robust image cloning with color harmonization [J]. Journal of System Simulation, 2015, 27(4): 723-730.
- [11] 黄美玉, 陈益强, 纪雯. 马尔科夫随机场化的光照一致图像合成方法[J]. 计算机辅助设计与图形学学报, 2015, 27(4): 691-702.
Huang Meiyu, Chen Yiqiang, Ji Wen. Markov random field based image composition with luminance consistence [J]. Journal of Computer-Aided Design & Computer Graphics, 2015, 27(4): 691-702.
- [12] 程皓楠, 王凯, 刘世光. 颜色风格自适应的图像克隆算法[J]. 图学学报, 2017, 38(5): 700-705.
Cheng Haonan, Wang Kai, Liu Shiguang. Adaptive color-style-aware image cloning[J]. Journal of Graphics, 2017, 38(5): 700-705.
- [13] Floater M. Mean value coordinates [J]. Computer Aided Geometric Design, 2003, 20(1): 19-27.
- [14] He K, Sun J. Fast guided filter [EB/OL]. <https://arxiv.org/abs/1505.00996>, 2015.
- [15] 钟宝江, 陆志芳, 季家欢. 图像插值技术综述[J]. 数据采集与处理, 2016, 31(6): 1083-1097.

- Zhong Baojiang, Lu Zhifang, Ji Jiahuan. Review on image interpolation techniques[J]. Journal of Data Acquisition and Processing, 2016, 31(6): 1083-1097.
- [16] 王平, 全吉成, 赵柏宇. 基于双线性插值的图像缩放在GPU上的实现[J]. 微电子学与计算机, 2016, 33(11): 129-132.
Wang Ping, Quan Jicheng, Zhao Boyu. Realization of image zooming in GPU based on bilinear interpolation [J]. Microelectronics & Computer, 2016, 33(11): 129-132.
- [17] Storti D, Yurtoglu M. CUDA for engineers: An introduction to high-performance parallel computing [M]. Boston: Addison-Wesley Professional, 2015.
- [18] Sonka M, Hlavac V, Boyle R. Image processing, analysis, and machine vision [M]. 3rd ed. Stanford: Cengage Learning, 2007.

附录 A 式(8)近似可行性的验证

由于 p_i 和 p_{i-1} 为两个相邻像素, 不失一般性, 假设在图 3 中三角形两边长相等, 即 $\|p_i - x\| = \|p_{i-1} - x\| = a$ 。又由于 $\|p_i - p_{i-1}\| = 1$, 根据余弦定理可得

$$\alpha_{i-1} = \arccos\left(\frac{2a^2 - 1}{2a^2}\right) \quad (\text{A1})$$

因此, 对式(8)用式(9)进行近似的相对误差为

$$e = \frac{\arccos((2a^2 - 1)/2a^2) - 1/a}{\arccos((2a^2 - 1)/2a^2)} \quad (\text{A2})$$

图 A1 显示了相对误差值 e 和三角形边长 a 之间的关系。

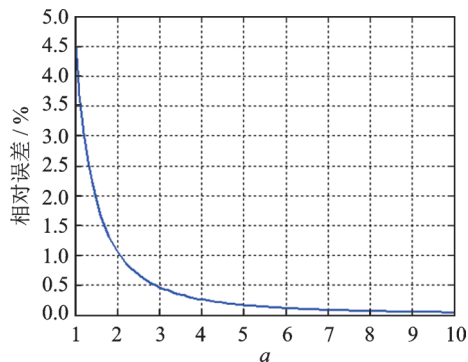


图 A1 相对误差值 e 和三角形边长 a 之间的关系

Fig.A1 Relation between relative errors e with the triangle edge length a

由图 A1 可知, 当区域 P 内的像素 x 与像素 p_i 和 p_{i-1} 之间距离在 3 个像素以上时, 上述近似的误差已经低于 0.5%, 由此验证了式(9)近似式(8)的可行性。

作者简介:



沈晔湖 (1982-), 男, 讲师, 研究方向: 图像处理、计算机视觉, E-mail: yehushen@mail.usts.edu.cn.



蒋全胜 (1978-), 男, 副教授, 研究方向: 信号与信息处理。



汪帮富 (1978-), 男, 讲师, 研究方向: 机电一体化。



朱其新 (1971-), 男, 教授, 研究方向: 先进控制理论及应用。