

在线视频分割实时后处理

钟 凡¹⁾ 秦学英^{1),2)} 陈佳舟¹⁾ 莫铭臻¹⁾ 彭群生¹⁾

¹⁾ (浙江大学 CAD&CG 国家重点实验室 杭州 310027)

²⁾ (山东大学计算机学院 济南 250101)

摘 要 在线视频分割的应用通常都需要对分割结果进行后处理,以消除误分割和边沿闪烁.基于图像抠图的方法太慢,而简单的对(前/背景)边界进行模糊不仅不能消除误分割,而且会导致清晰的边界被过度模糊.为了解决上述问题,文中提出了一种新的后处理方法.对边界附近的每一像素,首先通过一种新的专门用于颜色聚类的快速聚类算法得到该像素周围的局部颜色模型,并用来重新估计像素的 alpha 值,以消除误分割.为了改善结果的一致性,再采用一种自适应的边界函数作光滑性约束.边界函数可根据像素的局部属性自适应地调整过渡区域的中心和宽度,这样就防止了将清晰的边界过度模糊.文中的算法速度很快,可以很好地满足在线视频分割的要求.

关键词 alpha 值;视频分割;抠图;快速聚类;边界模型

中图法分类号 TP391 DOI号: 10.3724/SP.J.1016.2009.00261

Real-Time Post-Processing for Online Video Segmentation

ZHONG Fan¹⁾ QIN Xue-Ying^{1),2)} CHEN Jia-Zhou¹⁾ MO Ming-Zhen¹⁾ PENG Qun-Sheng¹⁾

¹⁾ (State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou 310027)

²⁾ (Department of Computer Science, Shandong University, Jinan 250101)

Abstract Applications of online video segmentation usually need to do post-processing in order to remove mis-segmentation and suppress flicking. Traditional matting-based methods are too slow, while simply blur the (foreground/background) boundary not only cause over-blur but also can't remove mis-segmentation. This paper proposes a novel post-processing method. For each pixel around the boundary, a local color model is first estimated through a new fast clustering algorithm, which is designed specially for color clustering. Mis-segmentation is then removed by re-estimating the alpha value for each pixel according to its local color model. In order to improve the consistence of result, an adaptive edge model is applied as a smooth constraint. The edge model can adjust the center and width of the transition region according to the local context of each pixel, and this way prevents the boundary from being over-blurred. The proposed method is very fast, and can meet the requirement of online video segmentation very well.

Keywords alpha value; video segmentation; matting; fast clustering; edge model

1 引 言

提取图像、视频中的前景物体近年来一直是计

算机图形学和计算机视觉研究的热门课题.分割与抠图是前景提取的两类主要方法,其区别在于分割只是简单地将像素分为前景与背景两类^[1],而抠图则要求为每一像素估计一连续变化的 alpha 值^[2].

收稿日期:2008-08-09;最终修改稿收到日期:2008-12-15.本课题得到国家“九七三”重点基础研究发展规划项目基金(2002CB312101)、国家“八六三”高技术研究发展计划项目基金(2007AA01Z326)资助.钟 凡,男,1982 年生,博士研究生,主要研究方向为图像视频分割与编辑. E-mail: zhongfan@cad.zju.edu.cn. 秦学英,女,1966 年生,教授,博士生导师,研究兴趣为增强现实、视频处理、摄像机定标与三维重建、计算机图形学等. 陈佳舟,男,1984 年生,博士研究生,主要研究方向为图像视频编辑、计算机视觉. 莫铭臻,男,1986 年生,本科生,主要研究方向为图像修复与抠图. 彭群生,男,1947 年生,教授,博士生导师,主要研究领域为计算机真实感图形、虚拟现实、生物计算、计算机动画、视频处理、科学信息可视化.

最近几年,由于在视频会议、增强现实等领域的潜在应用,在线视频分割受到了越来越广泛的关注^[3-6].由于在线视频分割不能有用户交互,并且必须达到实时的速度,因此它比离线视频分割和抠图更加困难.为了达到实时的速度,现有的方法都对输入视频

进行逐帧的二元分割.虽然二元分割可以通过“图切割”算法(graph-cut)^[7]快速地求解,但在边界附近容易出错(如图1所示),从而导致结果的闪烁.因此基于分割的前景提取方法都需要在分割完成之后对边界附近的像素作后处理,以获得较好的视觉效果.

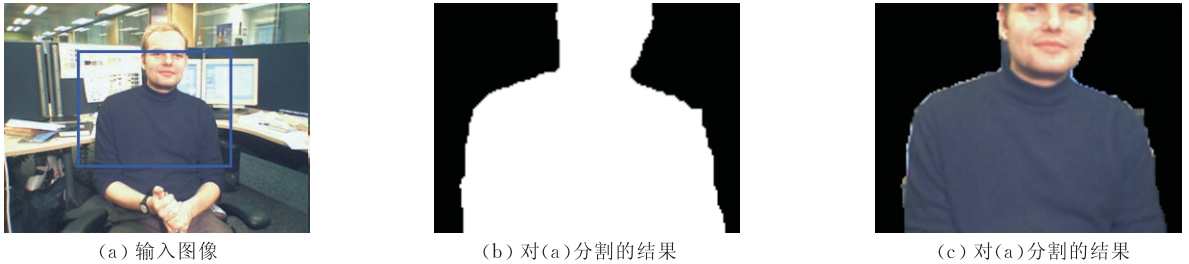


图1 二元分割在边界处存在较多错误(其中(b)(c)所示为(a)中矩形框内部分的放大图,可以看出二元分割在边界处存在较多的错误,并且边界很粗糙)

最简单的后处理是直接对图像边界进行羽化^[6].这种方法非常快,但是只能处理轻微的闪烁;并且羽化之后原本清晰的边界会变得模糊,与背景合成之后会有明显的颜色溢出,如图2(a)所示.更好的方法是采用边界抠图(border matting)^[4-5,8].该方法用一S形函数来拟合边界附近像素的 α 值,S形函数的参数需要通过动态规划全局优化得到,这样可以保证结果沿边界切线方向的光滑性.在一些交互的视频前景提取系统中^[9],边界抠图被扩展到视频体上.虽然这样有助于改善结果沿时间轴上的一致性,但在视频体上是难以进行实时优化的.此外,基于抠图的方法通常都假设前、背景之间是平滑过渡的,因此也不可避免地会在边界清晰的地方造成背景色溢出.当前、背景颜色相似时,这种错误会很严重.一致抠图(coherence matting)^[10-11]在贝叶斯抠图^[2]的基础上施加了边界约束,这样可以增强算法对相似前、背景颜色的鲁棒性.SPS方法^[3,12]也可用于边界的后处理,利用它可以处理较明显的误分割,但需要用双目视频作为输入.此外,更重要的

是,上述这些方法都只是针对图像或离线视频分割的,运行速度较慢,很难满足在线视频分割的实时性要求.

本文提出了一种专门针对在线视频分割的实时后处理算法.如图2(b)、(c)所示,本文的方法能有效地消除边界附近的误分割和闪烁,并且能够自适应地调整过渡区域的宽度,使边界清晰但不生硬,从而防止背景合成时的颜色溢出.首先以边界附近的每一像素为中心为其建立局部颜色模型,并用来计算其颜色 α 值,以消除误分割.局部颜色模型可通过对颜色空间的均匀剖分和三维连通域搜索快速地建立,在此过程中同时计算每个像素处边界的清晰度;为了改善结果的一致性,再构造一四参数的边界模型来为每个像素计算一边界 α 值.边界模型可根据边界的清晰度自适应地调整过渡区域的宽度及其中心位置,以防止边界被过度模糊;最后,根据颜色 α 值和边界 α 值的可信度将它们加权平均得到每一像素最终的 α 值.

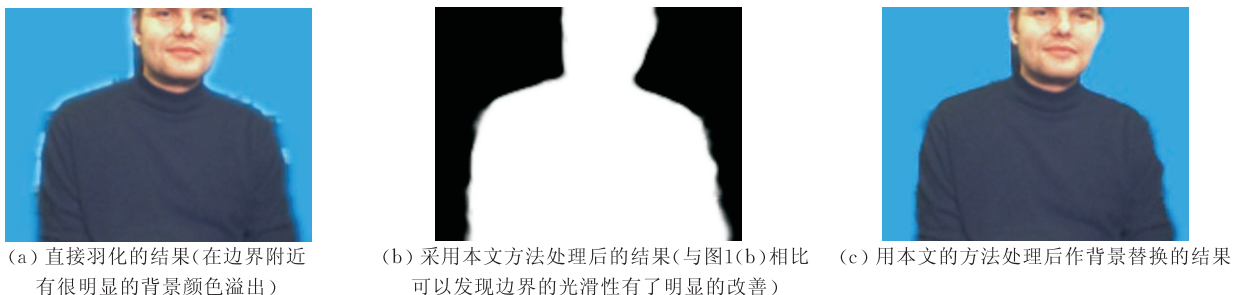


图2

2 实时后处理

2.1 计算颜色 α 值

现有的在线视频分割算法最终都是通过图切割算法来求解. 虽然图切割算法趋向于从边缘较强的地方进行分割, 但在边界附近, 前、背景颜色由于采样、压缩等原因, 会不可避免地有一定程度的混合, 因此二元分割很容易出错 (如图 1(c)). 一般来说, 由上述原因引起的错误是比较轻微的, 但在视频里会导致闪烁, 严重影响视觉效果. 在作进一步的处理之前, 必须先消除这些误分割.

后处理通常都只针对边界附近的像素进行. 记待处理像素的集合为 Ω ; 为了找出 Ω , 之前的方法都先找出前、背景的边界, 并通过距离变换计算出像素到边界的距离, 然后定义 Ω 为边界两侧某一宽度内的像素. 这种方法的缺点是速度较慢, 本文采用一种新的方式来定义 Ω :

$$\Omega(L_b) = \{p \mid \tau_0 < s_p < \tau_1\}, \quad s_p = \frac{1}{L_b^2} \sum_{x \in N_p(L_b)} \alpha_x^s \quad (1)$$

p 为一像素; $N_p(L_b)$ 是 p 的 $L_b \times L_b$ 邻域; $\alpha_p^s \in \{0, 1\}$ 为像素 p 的二元分割; s_p 为 p 的邻域内所有像素分割值之和的平均值; $\Omega(L_b)$ 为边界两侧一带状区域内的像素, 区域的宽度由参数 τ_0, τ_1 控制, $0 < \tau_0 < \tau_1 < 1$. 由于 s_p 可以通过积分图^[13] 快速地计算出来, 因此 Ω 的构造非常快.

对 Ω 中的每一像素 p , 构造一局部颜色模型来估计其颜色 α 值 $\alpha_p^c \in [0, 1]$. 局部颜色模型的选取和构造是决定后处理算法能否达到实时的关键. 边界抠图^[8] 中采用高斯模型来表示局部颜色分布, 虽然高斯模型的构造很快, 但在纹理丰富的区域会产生较大的误差. 解决这个问题的一般方法是采用高斯混合模型 (GMM), 但是估计 GMM 的参数太慢. 在这里采用一种近似的方法, 首先将样本聚类, 然后每一类用一个高斯模型来表示. 由于常用的聚类算法, 如 k -均值等, 都很难达到实时的速度, 因此本文提

出了一种基于颜色空间均匀剖分的快速聚类方法.

将颜色空间均匀剖分为 $32 \times 32 \times 32$ 个子空间 B_{ijk} . 为了对颜色样本集 X 进行聚类, 首先将它的每一个元素放入对应的子空间中, 得到一系列子样本集 X_{ijk} . 若 $|X_{ijk}| \leq \epsilon$, 则认为 X_{ijk} 及对应的子空间 B_{ijk} 都为空. 其中 $|\cdot|$ 表示集合中元素的个数. 然后找出所有非空子空间的连通域, 并把每一个连通域中包含的样本当成一类. 由于非空子集较少, 因此用三维的种子填充算法就能很快地找出连通域; 样本对应的子空间只需要对颜色值进行移位就能得到. 当 $\epsilon > 0$ 时, 很小的子样本集将会被丢弃, 这样可以避免噪声和误分割的影响, 同时可以加快求解连通域的速度.

聚类之后, 再为每一类中的颜色统计一高斯模型来表示其分布. 记样本集 X 的第 k 个子类的高斯分布为 $D_k(s|X)$, 则颜色 s 属于 X 的概率为

$$P(s|X) = \max\{D_k(s|X) \mid k=0, 1, \dots, K-1\} \quad (2)$$

其中 K 为子类的个数. 注意在这里采用的是最大概率, 而没有像 GMM 一样采用概率和, 这是因为最大概率能更准确地反映像素颜色与样本颜色集的相似度.

对 Ω 中每一像素 p , 在得到它的局部颜色模型之后, 就可以用局部颜色模型来估计它的颜色 α 值. $N_p(L_s)$ 是像素 p 的 $L_s \times L_s$ 领域. 若 $L_s > L_b$, 则 $N_p(L_s)$ 中必同时包含前景像素 ($\alpha_p^s = 1$) 和背景像素 ($\alpha_p^s = 0$). 记 $N_p(L_s)$ 中前景像素和背景像素的集合分别为 N_p^F 和 N_p^B , 则

$$\alpha_p^c = \frac{P(I_p | N_p^F)}{P(I_p | N_p^F) + P(I_p | N_p^B)} \quad (3)$$

其中, I_p 为像素 p 的 RGB 颜色.

为了改善颜色 α 值的空间一致性, 本文也用每一像素处的局部颜色模型来计算它的八邻域像素的颜色 α 值. 这样每一个像素将会得到最多 9 个颜色 α 值, 最后取这些值的平均值作为像素的颜色 α 值. 图 3(b)、(c) 是由上述算法得到的颜色 α 值及相应的前景, 与图 1(b)、(c) 比较可以发现大部分误分割都已经被消除了.



(a) 边界像素 (黑线和暗影线条之间的像素为待处理的边界像素 ($L_b=7$; $\tau_0=0.1$; $\tau_1=0.9$))



(b) 颜色 α 值



(c) 通过 (b) 中的 α 值得到的前景, 与图 1(c) 相比可以发现大多数误分割都已经被消除

2.2 计算边界 alpha 值

如图 3(b) 所示, 在前景、背景颜色很接近的区域, 颜色 alpha 值会有较明显的错误, 导致过宽的半透明区域和边界不光滑. 在这种情况下应该遵从二元分割的结果. 边界 alpha 值即是根据像素离分割边界的距离计算的 alpha 值. 一般来说, 像素离边界的距离可通过距离变换^[14]较快地求得, 但是在这里采用一种更加简便快速的方法. 注意到在式(1)中, $s_p \in [0, 1]$ 可当作是像素 p 处前景像素分布的核密度, 它可以反映像素 p 离边界的距离. 记像素 p 的边界 alpha 值为 α_p^b , 边界函数为 $f(x; \Theta_p)$, 其中 Θ_p 为 p 处边界函数的参数, 则 $\alpha_p^b = f(s_p; \Theta_p)$. 本文采用在文献[15]中提出的四参数边界模型:

$$f(x; \Theta_p) = \frac{a_p}{1 + e^{(c_p - x)/\sigma_p}} + b_p \quad (4)$$

其中 $\Theta_p = \{a_p, b_p, c_p, \sigma_p\}$ 是 4 个待确定的参数. 该函数是一 S 形函数, a_p, b_p 可控制其值域, σ_p 可控制其陡峭程度, c_p 可控制其中心. 文献[15]通过对所有边界点进行全局优化来获取它们的参数, 这样做可以获得很好的效果, 但是速度较慢. 本文提出一种根据边界清晰度和控制关键点来计算边界函数参数的方法.

首先确定 σ_p . 注意到 σ_p 应该是与边界的清晰度相关的, 边界越清晰, σ_p 应该越小. 而边界的清晰度在一定程度上反映在边界两侧前、背景颜色的分布, 前、背景颜色在颜色空间的交迭越少, 则边界越清晰. 基于此可以根据上一节中对颜色空间的剖分来估计边界的清晰度. 若颜色子空间 B_{ijk} 同时包含 N_p^F 和 N_p^B 中的元素, 则认为它是二义的. 记 \bar{n}_p 为所有二义颜色子空间包含的 N_p^F 和 N_p^B 中颜色样本的总数, 则像素 p 处边界的清晰度为

$$\varphi_p = 1 - \frac{\bar{n}_p}{L_s^2} \quad (5)$$

然后如下计算 σ_p :

$$\sigma_p = \frac{1}{2}(1 - \varphi_p) + \sigma_0 \quad (6)$$

σ_0 可防止 σ_p 太小, 在本文的实验中取 $\sigma_0 = 0.1$.

为了得到 Θ_p 中的其它参数, 注意到边界 alpha 值应该在与背景和前景相接的地方光滑过渡, 于是有 $f(0; \Theta_p) = 0; f(1; \Theta_p) = 1$. 此外, 令 $f(0.5; \Theta_p) = \mu_p$. 由于 $s_p = 0.5$ 的像素可被认为是刚好在边界线上, 因此通过 μ_p 可以控制边界函数的中心. 由以上 3 个约束可以解得

$$a_p = \frac{(1+h)(1+g^2h)}{(1-g^2)h}, \quad b_p = \frac{1+g^2h}{(g^2-1)h}, \quad c_p = \sigma_p \log h \quad (7)$$

其中

$$g = e^{-\frac{1}{2\sigma_p}}, \quad h = \frac{1}{g} \left(\frac{g-1}{g-(g+1)\mu_p} - 1 \right) \quad (8)$$

注意为了使 $\log h$ 有意义, h 必须大于 0, 又由于 $0 < g < 1$, 于是 μ_p 必须满足 $\frac{g}{g+1} < \mu_p < \frac{1}{g+1}$. 为了方便控制 μ_p , 可将它写为如下的形式:

$$\mu_p = \frac{g + \nu(1-g)}{g+1}, \quad 0 < \nu < 1 \quad (9)$$

若 $\nu = 0.5$, 则 $c_p \equiv 0.5$ 与 σ_p 无关, 此时边界函数的中心刚好落在前、背景的边界线上; 若 $\nu < 0.5$, 则 $c_p > 0.5$, 中心在前景区域; 否则 $c_p < 0.5$, 中心在背景区域. 如图 4 所示, 通过 ν 可以控制像素的边界 alpha 值. 为了使边界函数的中心尽量落在由颜色 alpha 值(而不是二元分割)决定的前、背景边界上, ν 应该根据颜色 alpha 值作相应的调整:

$$\nu = 0.5 + \frac{1}{2|\Omega|} \sum_{q \in \Omega} (\alpha_q^c - \alpha_q^s) \quad (10)$$

注意 ν 对 Ω 中所有像素都是相同的, 虽然直观上根据每个像素的局部属性计算的 ν 会更好, 但实际上这样做会引入太多的不一致. 因此在这里 ν 仅用来作全局的控制, 像素之间的差异将通过 σ_p 表现出来. 给定 ν 和 σ_p , Θ_p 中的其它参数可很容易地计算出来, 然后可通过边界函数来计算像素 p 的边界 alpha 值 α_p^b .



(a) $\nu = 0.2$, 过多前景像素被当成背景



(b) $\nu = 0.5$, 边界函数中心刚好在二元分割边界线上



(c) $\nu = 0.8$, 过多的背景像素被当成前景

图 4 通过 ν 控制边界 alpha 值

2.3 计算 alpha 值

颜色 alpha 值可以有效地消除误分割,但在前、背景颜色相似的地方容易出错;边界 alpha 值不受颜色相似性的影响,且与前、背景之间过渡更光滑,但没有考虑误分割.因此最后本文通过对两者加权平均来计算各像素最终的 alpha 值 α_p ,权重将通过颜色 alpha 值和边界 alpha 值的可信度来计算.

记颜色 alpha 值 α_p^c 的可信度为 ω_p^c ,它实际上取决于像素 p 周围前、背景颜色在颜色空间的分离程度,因此可直接用边界的清晰度 φ_p 作为 ω_p^c 的值, $\omega_p^c = \varphi_p$.相应地,记边界 alpha 值的可信度为 ω_p^b ,它应该取决于 p 周围二元分割的错误率,因此可以通过颜色 α 值和二元分割结果的差来计算

$$\omega_p^c = \frac{1}{9} \sum_{q \in N_p(3)} |\alpha_q^c - \alpha_q^s| \quad (11)$$

$N_p(3)$ 表示像素 p 的 3×3 邻域. 像素 p 的 alpha 值 α_p 于是可以计算为

$$\alpha_p = \omega_p \alpha_p^c + (1 - \omega_p) \alpha_p^b, \quad \omega_p = \frac{\omega_p^c}{\omega_p^c + \omega_p^b} \quad (12)$$

下面总结一下本文算法的步骤:

1. 构造二元分割的积分图,确定待处理的边界像素集合 Ω ;
2. 对 Ω 中的每一个像素 p ,以其邻域 $N_p(L_s)$ 中像素的颜色值为样本建立局部颜色模型,并计算 p 的颜色 alpha 值 α_p^c 和边界清晰度 φ_p ;
3. 计算参数 ν ;
4. 对 Ω 中的每一像素 p ,计算 σ_p ,然后根据 ν 和 σ_p 计算 Θ_p ,获取边界 alpha 值 α_p^b ;

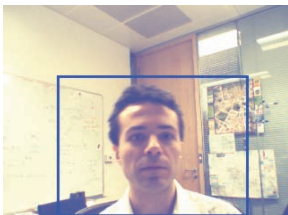
5. 计算可信度 ω_p^c, ω_p^b 以及权重 ω_p ,然后对 α_p^c 和 α_p^b 加权平均得到像素 p 的最终 alpha 值 α_p .

3 实验结果与讨论

在实验中我们首先采用文献[3]中的方法对输入视频进行二元分割,然后用本文提出的方法进行后处理.

我们在 2.2GHz CPU 的 PC 上用 C++ 实现了本文中的算法,表 1 是对运行时间的统计.输入都是 320×240 的视频序列.总平均时间带 * 号的行对应的参数为最后采用的参数,其余两行仅用于说明参数对运行时间的影响.参数 L_b, τ_0, τ_1 会影响需要处理的边界像素的数量,而 L_s 决定了用于建立局部颜色模型的样本集的大小,因此会对运行时间有很大的影响.可以看出本文的算法速度非常快,每秒可处理 40~70 帧.

图 5 是对序列 AC 中某一帧处理的结果.从 (b)、(c) 可以看出二元分割在边界附近存在很大的问题,不仅误分割较严重,而且即使在清晰的边界上也会有较多的锯齿.(d) 是通过图像抠图获得的结果^[16],虽然在大部分区域都很好,但在前、背景过于相似的地方颜色溢出现象会很严重.这是所有图像抠图算法应用于分割后处理的一个共同缺点,此外,抠图算法在速度上很难达到实时系统的要求.(e)、(f) 是用本文的方法处理后的结果.可以看出本文的方法有效地消除了误分割,并且没有颜色溢出现象.



(a) 输入帧



(b) 二元分割结果



(c) 二元分割结果



(d) 采用文献[16]的方法抠图的结果
(可见在某些区域有明显的颜色溢出现象)



(e) 用本文方法处理后的结果



(f) 用本文方法处理后的结果

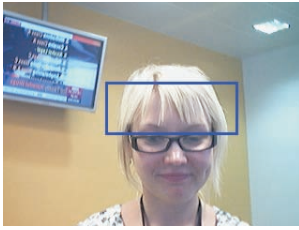
图 5 对序列 AC 中某一帧处理的结果

表 1 不同参数下处理每 10 帧的平均时间以及总平均时间

输入	参数				时间/ms									平均
	L_b	τ_0	τ_1	L_s	~10	~20	~30	~40	~50	~60	~70	...		
MS	7	0.3	0.7	15	15.1	14.1	14.1	14.1	14.6	14.1	14.6	...	14.8*	
	15	0.3	0.7	21	42.2	41.2	41.7	39.6	32.2	41.7	42.2	...	42.4	
	7	0.1	0.9	15	27.6	26.6	27.1	25.5	28.1	26.6	27.6	...	27.3	
AC	7	0.3	0.7	21	13.5	13.6	13.6	13.0	13.0	13.5	12.5	...	13.1	
	15	0.3	0.7	21	36.5	36.0	36.0	34.4	34.9	36.4	31.8	...	34.7	
	7	0.1	0.9	15	23.4	23.9	24.0	23.0	23.0	24.0	21.9	...	22.9*	

从图 6 可以看出本文方法有效地保持了边界的清晰度. 在这里误分割主要集中在前景的左边界附近,

此区域内过渡较明显, 而右边界则很清晰, 但却没有如分割结果一样的锯齿.



(a) 输入帧



(b) 上面一行为二元分割结果, 下面一行为用本文方法处理后的结果(可以看出本文的方法有效地消除了误分割(左边界), 并保持了边界的清晰度(右边界))



(c) 上面一行为二元分割结果, 下面一行为用本文方法处理后的结果(可以看出本文的方法有效地消除了误分割(左边界), 并保持了边界的清晰度(右边界))

图 6 对序列 56 中某一帧处理的结果

由于后处理的目的是消除闪烁, 因此本文的方法对大块的误分割是无能为力的. 这一方面是由于它只处理边界附近的像素; 另一方面则是因为在本文方法中用来消除误分割的局部颜色模型是建立在二元分割结果基础之上的, 当分割出现严重错误时, 局部颜色模型会很不准确. 因此大块的误分割一般只能通过改进分割算法来消除.

4 总 结

本文提出了一种对在线视频分割结果进行实时后处理的算法, 以消除二元分割造成的误分割和闪烁, 改善视觉效果. 首先通过局部颜色模型为待处理像素计算一颜色 alpha 值, 并估计边界的清晰度. 通过颜色 alpha 值可以有效地消除误分割, 而边界的清晰度可以控制过渡区域的宽度. 为了快速地建立局部颜色模型, 本文提出了一种基于三维颜色空间均匀剖分和连通域搜索的快速样本聚类方法. 其次, 根据四参数的边界模型, 为待处理像素计算一边界 alpha 值. 边界 alpha 值具有很好的光滑性, 并且不受前、背景相似颜色的影响. 边界函数可根据边界处的清晰度自动地调整过渡区域的宽度, 并根据误分割的程度调整其中心位置. 这种自适应能力使得本

文算法只在前、背景相似或误分割严重的边界附近产生明显的过渡, 从而防止了颜色溢出. 最后, 根据颜色 alpha 值和边界 alpha 值的可信度将它们加权平均得到最终的 alpha 值, 这样既有效地消除了误分割, 又保证了结果的光滑性. 本文的方法速度很快, 可以满足在线视频分割的实时性要求. 此外, 本文的方法是不依赖于分割算法的, 因此它适用于任何分割算法; 这种独立性也使得它更适合于与分割一起并行处理.

参 考 文 献

- [1] Blake A, Rother C, Brown M, Perez P, Torr P. Interactive image segmentation using an adaptive GMMRF model//Proceedings of the European Conference on Computer Vision. Prague, Czech Republic, 2004: 428-441
- [2] Chuang Y Y, Curless B, Salesin D H, Szeliski R. A bayesian approach to digital matting//Proceedings of the Conference on Computer Vision and Pattern Recognition. Hawaii, 2001: 264-271
- [3] Kolmogorov V, Criminisi A, Blake A, Cross G, Rother C. Bi-layer segmentation of binocular stereo video//Proceedings of the Conference on Computer Vision and Pattern Recognition. San Diego, CA, USA, 2005: 407-414
- [4] Criminisi A, Cross G, Blake A, Kolmogorov V. Bilayer segmentation of live video//Proceedings of the Conference on

- Computer Vision and Pattern Recognition. New York, 2006; 53-60
- [5] Yin P, Criminisi A, Winn J, Essa I. Tree-based classifiers for bilayer video segmentation//Proceedings of Conference on Computer Vision and Pattern Recognition. Minneapolis, MN, 2007; 1-8
- [6] Sun J, Zhang W, Tang X, Shum H Y. Background cut//Proceedings of the European Conference on Computer Vision. Graz, Austria, 2006; 628-641
- [7] Boykov Y, Veksler O, Zabih R. Efficient approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2001, 20(11): 1222-1239
- [8] Rother C, Kolmogorov V, Blake A. Grab-cut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 2004, 23(3): 309-314
- [9] Wang J, Bhat P, Colburn R A, Agrawala M, Cohen F-M. Interactive video cutout//Proceedings of the ACM SIGGRAPH. Los Angeles, 2005; 585-594
- [10] Shum H-Y, Sun J, Yamazaki S, Li Y, Tang C-K. Pop-up light field: An interactive image based modeling and rendering system. *ACM Transactions on Graphics*, 2004, 23(2): 143-162
- [11] Li Y, Sun J, Shum H-Y. Video object cut and paste//Proceedings of the ACM SIGGRAPH. Los Angeles, 2005; 595-600
- [12] Criminisi A, Blake A. The SPS algorithm: Patching figural continuity and transparency by split-patch search//Proceedings of the European Conference on Computer Vision. Washington, DC, 2004; 342-349
- [13] Viola P, Jones M. Robust real-time object detection. *International Journal of Computer Vision*, 2004, 57(2): 137-154
- [14] Rucklidge W-J. *Efficient Visual Recognition Using the Hausdorff Distance*. NY: Springer-Verlag, 1996
- [15] Mortensen Eric N, Barrett William A. Toboggan-based intelligent scissors with a four parameter edge model//Proceedings of the ACM SIGGRAPH. Fort Collins, CO, USA, 1999; 451-459
- [16] Wang J, Cohen M. An iterative optimization approach for unified image segmentation and matting//Proceedings of the International Conference on Computer Vision. Beijing, 2005; 936-943



ZHONG Fan, born in 1982, Ph. D. candidate. His research interests include image and video segmentation, editing, etc.

QIN Xue-Ying, born in 1966, professor, Ph. D. supervisor. Her major research interests include augment reality, video processing, computer vision and computer graphics, etc.

Background

High-quality online video segmentation is very difficult due to the scene complexity and limited computational resource. Recent works almost all regard it as a MRF-based optimization problem, and then solve with max-flow/min-cut, which yields a binary segmentation that is global optimal but less satisfying nearby the foreground/background boundary, so post-processing must be applied in order to suppress flicking. Previous works do post-processing by either feathering or matting, both are not designed specially for online video segmentation and perform not very well.

This paper proposed a post-processing method that fully

CHEN Jia-Zhou, born in 1984, Ph. D. candidate. His research interests include image and video editing, computer vision, etc.

MO Ming-Zhen, born in 1986, undergraduate student. His research interests include image completion and matting.

PENG Qun-Sheng, born in 1947, professor, Ph. D. supervisor. His major research interests include realistic image synthesis, virtual reality, video processing, computer animation, scientific visualization, etc.

considering the requirement of online video segmentation. The proposed method can effectively remove the mis-segmentation nearby the boundary and adaptively smooth the boundary without causing over-blur. In addition, it is very fast and brings only a little computational burden for real-time applications.

This paper is supported by the National High Technology Research and Development Program (863 Program) of China under grant No. 2007AA01Z326; the National Basic Research Program (973 Program) of China under grant No. 2002CB312101.