

彩色图像序列中运动目标外轮廓的自动提取

刘洋 李玉山 张大朴

(西安电子科技大学电路CAD研究所 西安 710071)

摘要 该文提出了一种新的多运动目标精确外轮廓自动提取算法。算法的主要创新点是：采用指数滤波计算彩色图像的梯度强度，改善了彩色图像梯度局部极值点的定位精度，所得到的梯度强度和运动边缘位置信息供随后的轮廓迭代中使用，成功地避免了动态轮廓的过收缩。此外，充分利用轮廓的方向信息，又有效地克服了动态轮廓迭代中的外扩展。从而，算法自动地提取出一个真正紧贴的目标外轮廓。分析和实验表明，该算法抗干扰能力强，对复杂背景及不重叠物体，可以精确定位并提取出各个运动目标的外轮廓。

关键词 动态轮廓，运动检测，指数滤波，背景减除

中图分类号：TP391.41

文献标识码：A

文章编号：1009-5896(2006)09-1550-05

Contour Extraction of Moving Objects in Color Image Sequences

Liu Yang Li Yu-shan Zhang Da-pu

(Institute of Electronic CAD, Xidian University, Xi'an 710071, China)

Abstract A new algorithm to extract accurate contours of multiple moving objects is proposed in this paper. The main contributions of this paper include two parts: firstly, exponential filter is used to calculate the gradient magnitude of color image and the localization accuracy of local maximum of gradient in color image is increased, the gradient magnitude and the motion edges are used subsequently in contour iteration to prevent the contour from shrinking successfully; secondly, the directions of contour are used to prevent the active contour from expanding effectively. As a result, the real compact contours would be extracted automatically by using this algorithm. The experiments show the high efficiency of this algorithm for contours extraction of moving objects from color image sequence with complex background.

Key words Active contour, Motion detection, Exponential filter, Background subtraction

1 引言

视频运动对象的分割技术在运动视觉检测和新的MPEG-4视频编码标准中都占有十分重要的地位，但复杂背景下精确地分割出多运动目标仍然存在许多问题。近年来，在研究人员提出的各种运动目标分割算法中，动态轮廓模型，也称为Snake模型，受到了广泛的关注。Kass等人^[1]提出的动态轮廓模型，是在图像域内定义的可变形曲线 $v_i = (x_i, y_i)$ ，通过对其能量函数的最小化，逐步调整自身形状与目标轮廓相一致。但目标初始轮廓需要人为定义，不能直接用于运动目标轮廓检测^[2]，而且，动态轮廓算法的性能受到初始轮廓点位置选取以及轮廓点局部梯度强度估计误差的干扰。

基于以上研究，本文采用指数滤波比较精确地估计彩色图像的梯度强度；利用运动边缘信息和轮廓的方向信息，对原始动态轮廓算法的能量函数进行了修改。我们综合运动区域检测和动态轮廓模型的特点，以提高实时性并能精确检测运动目标轮廓为原则，提出了一种新的精确提取多运动目标轮廓的算法。其基本思路为：(1) 利用高阶统计理论，建立

背景约束条件，运用块处理技术实现快速背景重建，采用基于区域的背景模型更新方法进行背景更新，消除虚假目标区域；(2) 依靠重建背景进行运动区域检测，采用形态滤波填充运动对象二值模板中的空洞，消除小噪声及平滑边缘；采用连通区域算子，清除大的背景噪声；运用外轮廓跟踪算法自动标注出真实运动目标的初始轮廓；(3) 利用彩色信息，采用指数滤波精确估计彩色图像的梯度强度，在动态轮廓模型的外部约束力中加入边缘点硬性约束和轮廓方向信息，采用动态轮廓收缩算法精确提取多运动目标的外轮廓。

2 运动目标的初始轮廓生成

在摄像机静止的条件下，背景减除是运动分割中最常用的一种方法。这种方法计算速度很快，并且一般能够给出最完整的特征数据，但对于动态场景的变化，如突发亮点，噪声等比较敏感，在实际应用中需采用一定的算法进行背景模型的动态更新，以适应环境变化。由于噪声具有高斯特性而运动目标具有很强的结构性^[3]，我们利用高阶统计理论建立背景约束条件。根据图像序列的前 N 帧图像，采用块处理技术对序列场景进行快速背景重建。

假设给定 N 帧图像序列 $\{f_c(x, y, t)\}_{t=1}^N$ ， $c \in \{R, G, B\}$ ， t 和 $t+1$ 时刻彩色图像帧之间的帧差用 $Df_c(x, y, t) = |f_c(x, y, t+1) - f_c(x, y, t)|$ 表示。将 $f_c(x, y, t)$ 和 $Df_c(x, y, t)$ 按

2004-12-27 收到，2005-11-14 改回
国家自然科学基金(60172004)和教育部博士点基金(20010701003)
资助课题

从上到下, 从左到右的顺序划分成 $K \times K$ 大小的序列子块, 分别用 $\{f_c(x, y, t, m)\}_{t=1, m=1}^{N^M}$ 和 $\{Df_c(x, y, t, m)\}_{t=1, m=1}^{N^M}$ (M 为子块的个数) 表示。在 $Df_c(x, y, t)$ 中, 两帧之间空间位置对应于共同背景区的子块内的随机噪声点序列通常呈高斯分布。而对应于运动目标区的子块, 由于目标轮廓依然存在, 其分布不具有高斯性。高斯分布随机变量的一个重要性质是其四阶统计量为零, 因此, 对应于共同背景区的子块内 $Df_c(x, y, t, m)$ 的四阶统计量为零或很小。通常情况下, 直接计算高斯分布随机变量的四阶统计量比较复杂, 可以利用四阶统计量与四阶中心矩和方差的关系进行快速求解。在 N 帧图像序列中, 对应于彩色图像每个子块, 我们选取使子块 $Df_c(x, y, t, m)$ 的各个颜色分量四阶统计量之和最小的两帧图像子块 $f_c(x, y, t, m)$ 和 $f_c(x, y, t+1, m)$ 内对应于同一空间位置的两个像素点的均值作为当前位置的背景像素点, 即

$$B_c(x, y) = (f_c(x, y, t) + f_c(x, y, t+1)) / 2 \quad (1)$$

根据实验结果, 采用 8×8 大小的块, 30 帧图像 (N 取为 30) 就可以取得理想的效果。但是如果运动目标特别均匀或目标较多的情况下, 在背景重建时可能会造成少许误差。同时由于场景的动态变化, 例如, 场景中静止的汽车突然开始运动, 会造成采用背景减除方法检测出的运动区域, 一方面是由于目标运动而引起的真实运动区域, 记为 TMVO; 另一方面也有可能是由于场景的变化以及背景重建的误差造成的虚假运动区域 FMVO。为了区分真实和虚假的运动区域, 并对场景的动态变化作出快速响应, 我们采用基于区域的背景模型更新方法^[4], 计算采用背景减除方法检测出的运动区域中每个像素点的光流值, 真实运动区域 TMVO 应该具有较大的平均光流值, 而虚假运动区域 FMVO 的平均光流值近似为 0。根据当前像素点属于真实运动区域还是虚假运动区域来决定当前像素点的更新方式, 背景模型的更新方程如下:

$$B_c(x, y, t+1) = \begin{cases} B_c(x, y, t), & f_c(x, y, t) \in \text{TMVO} \\ (f_c(x, y, t) + f_c(x, y, t+1)) / 2, & \\ f_c(x, y, t) \in \text{FMVO} \\ (1-\lambda)B_c(x, y, t) + \lambda f_c(x, y, t), & \text{其他} \end{cases} \quad (2)$$

这里, λ 为背景模型的更新率。根据重建出来的背景图像作为参考图像, 确定出亮度变化较大的区域, 并加以阈值化来检测运动区域。在检测运动区域时, 为了达到一定的实时性, 而又压缩噪声, 对两帧图像序列差分进行高斯平滑滤波处理。

$$GDF_c(x, y, t) = G(x, y) * |f_c(x, y, t) - B_c(x, y, t)| \quad (3)$$

$$BDF(x, y, t) = \begin{cases} 1, & GDF_R(x, y, t) > \tau \vee GDF_G(x, y, t) > \tau \\ & > \tau \vee GDF_B(x, y, t) > \tau \\ 0, & \text{其他} \end{cases} \quad (4)$$

其中 $G(x, y)$ 为二维高斯平滑滤波器, $*$ 表示卷积运算(以下同), τ 为门限阈值。 $BDF(x, y, t)$ 为对帧间差阈值化后形成的二值模板; 若为 1, 则像素 (x, y) 属于运动对象, 否则属于背

景或对象的静止部分。

由于噪声的影响, 得到的二值化分割图像可能会产生一定微小的空洞和杂点, 可以采用形态滤波方法对其进行进一步处理, 填充内部空洞和消除杂点。但是在复杂背景下, 可能存在较大的背景噪声, 如果在形态滤波时选用较小的结构元, 则无法消除噪声点; 如果选用较大的结构元, 则会导致目标结构的改变和形状的扭曲。基于这一问题, 我们假定目标的区域面积要大于背景噪声的区域面积, 在对运动区域形态滤波以后, 使用连通区域算子清除大的背景噪声; 采用前面讲述的方法消除虚假的运动区域, 进行背景更新; 由于在帧差过程中, 为了抑制噪声干扰对帧差进行了高斯平滑滤波, 并采用形态滤波对运动区域进行了膨胀平滑, 得到的运动区域比实际的运动目标面积稍大。为了使每个运动目标都有精确、平滑的外轮廓, 对每个包含运动目标外轮廓的运动区域进行外轮廓跟踪, 自动标注出真实运动目标的初始轮廓; 采用动态轮廓算法对运动区域外轮廓进行平滑收缩, 以便精确提取运动目标。

3 多运动目标轮廓提取

动态轮廓是由内部约束力引导和外部约束力影响的能量最小化曲线。最小化能量函数的方法很多, 有变微分法^[1], greedy 算法^[5], 动态规划算法^[6]等。其中, greedy 算法保留了动态规划算法可靠性高, 稳定性好的优点, 而且计算简单。但是, 原始 greedy 算法对于动态轮廓的初始位置很敏感, 由于 α , β , γ 系数的不确定, 内部收缩力很难和图像特征吸引力达到平衡^[7]; 而且, greedy 算法采用高斯滤波估计图像的梯度强度, 容易造成目标边缘定位不准确。基于这些问题, 我们采用指数滤波比较精确地估计彩色图像的梯度强度, 改善了彩色图像梯度局部极值点的定位精度, 并且在外部约束力中加入边缘点硬性约束和轮廓方向信息, 有效地克服了动态轮廓提取中的过收缩和向外扩展的问题。

3.1 指数滤波取代高斯滤波

无限对称指数滤波器是一种具有诸多优点的阶跃边缘平滑滤波器^[8]。由于其独特的可递推机制以及实际可利用数据和计算量与滤波器宽度参数不相关的优势, 在本算法的使用中明显比高斯滤波器快速和简便。相反, 高斯滤波器的可利用数据, 正比于高斯滤波器的宽度, 高斯滤波的计算量随高斯滤波器的宽度而线性增加。同时, 固定大小的高斯滤波器无法满足检测具有不同尺度大小的边缘结构的要求。综合考虑, 我们选取指数滤波进行图像梯度强度的估计。一维指数滤波器表示如下:

$$f(x) = ca_0(1-a_0)^{|x|} = f_1(x) * f_2(x) = c(f_1(x) + f_2(x) - a_0\delta(x)) \quad (5)$$

其中 $0 \leq a_0 \leq 1$, $c = 1/(2-a_0)$ 。可以推出

$$\left. \begin{aligned} f_1(x) &= a_0(1-a_0)^x U(x) \\ f_2(x) &= a_0(1-a_0)^{-x} U(-x) \end{aligned} \right\} \quad (6)$$

由式(5), 式(6)可得出

$$d(f(x))/dx = f_2(x) - f_1(x) \quad (7)$$

(在这里, 我们略去了对结果不产生影响的系数, 以下同)。

对于一维来说, 令 $s(x)$ 为输入信号, $f(x)$ 为无限窗口指数滤波器, $s_0(x)$ 为输出信号, 即

$$s_0(x) = s(x) * f(x) = s(x) * f_1(x) * f_2(x) \quad (8)$$

根据式(6), 式(8), 可以得到 $s_0(x)$ 的递推计算公式

$$\left. \begin{aligned} s_1(x) &= s(x) * f_1(x) = a_0 s(x) + (1 - a_0) s_1(x-1) \\ s_0(x) &= s_1(x) * f_2(x) = a_0 s_1(x) + (1 - a_0) s_0(x+1) \end{aligned} \right\} \quad (9)$$

同理, 对于二维图像, 设 $s(x, y)$ 为输入图像, $f(x, y)$ 为二维指数滤波器, 输出为 $s_0(x, y)$, 则由指数滤波的可分离性, 即 $f(x, y) = f(x)f(y)$, 可以得出

$$\begin{aligned} s_0(x, y) &= s(x, y) * f(x) * f(y) \\ &= s(x, y) * f_1(x) * f_2(x) * f_1(y) * f_2(y) \end{aligned} \quad (10)$$

根据前面的分析结果, 滤波后图像的一阶 x , y 方向导数为^[8]

$$\left. \begin{aligned} S_{0x} &= d(s_0(x, y))/dx = s(x, y) * [f_2(x) - f_1(x)] * f_1(y) * f_2(y) \\ S_{0y} &= d(s_0(x, y))/dy = s(x, y) * f_1(x) * f_2(x) * [f_2(y) - f_1(y)] \end{aligned} \right\} \quad (11)$$

使用上述算法得到的沿 x 和沿 y 方向的导数结果, 可以检测出图像的梯度幅值及梯度方向:

$$\text{梯度幅值: } M(x, y) = \sqrt{S_{0x}^2(x, y) + S_{0y}^2(x, y)} \quad (12)$$

$$\text{梯度方向: } \theta(x, y) = \arctan(S_{0y}(x, y)/S_{0x}(x, y)) \quad (13)$$

边缘一般定义为梯度的局部最大值点, 可以通过检测梯度局部最大值或沿梯度方向的二阶导数的过零点来检测边缘。在这里, 本论文认为, 若 (x_0, y_0) 为边缘点, 则 $M(x, y)$ 在 (x_0, y_0) 处沿 θ 方向取得极大值。

3.2 能量函数及其最小化

为了使算法更具有鲁棒性, 我们所定义的能量函数利用了彩色信息, 采用指数滤波的方法代替了目前大多数文献中的高斯滤波来估计图像的梯度强度, 并且在外部约束力中加入边缘点硬性约束和轮廓方向信息。动态轮廓的总能量函数表示为

$$E_{\text{snake}}(v) = \sum_{i=0}^{n-1} (E_{\text{int}}(v_i) + E_{\text{image}}(v_i)) \quad (14)$$

其中 $E_{\text{int}}(v_i) = \alpha_i E_{\text{cont}}(v_i) + \beta_i E_{\text{curv}}(v_i)$ 为动态轮廓的内部能量; $E_{\text{image}}(v_i) = \gamma_i E_{\text{coledge}}(v_i)$ 为动态轮廓的外部能量。在这里, α , β , γ 系数的相对大小分别控制动态轮廓的平滑程度、可弯曲程度以及图像特征吸引力。式(14)中内部能量的第 1 项和第 2 项分别表示轮廓的可拉伸程度和可弯曲程度, 其标准化形式分别为

$$\left. \begin{aligned} E_{\text{cont}}(v_i) &= \frac{\bar{d} - |v_i - v_{i-1}|}{\max_j \{|\bar{d} - |v_{i,j} - v_{i-1}||\}} \\ E_{\text{curv}}(v_i) &= \frac{|v_{i-1} - 2v_i + v_{i+1}|^2}{\max_j \{|v_{i-1} - 2v_{i,j} + v_{i+1}|^2\}} \end{aligned} \right\} \quad (15)$$

在这里, $\{v_{i,j} | j = 0, 1, \dots, 8\}$ 表示当前轮廓点 $v_i = v_{i,0}$ 和其

8 邻域轮廓点 $v_{i,j} (j \neq 0)$ 。 $|v_i - v_{i-1}|$ 表示相邻两个轮廓点之间的距离, \bar{d} 是动态轮廓上所有轮廓点之间的平均距离。

由于 $E_{\text{cont}}(v_i)$ 使得轮廓点在轮廓曲线上均匀地分布, $E_{\text{curv}}(v_i)$ 是轮廓曲线局部曲率的近似估计。式(14)中的外部能量是图像约束力, 在这里, 我们利用了图像的彩色信息, 并考虑到 greedy 算法对于动态轮廓的初始位置敏感这一问题, 提出了一种利用运动边缘和动态轮廓的方向进行轮廓收缩的算法, 命名 MEDG(Motion Edge and Directional Greedy) 算法, 在原始 greedy 算法的基础上增加外部约束力: 运动边缘点约束力和动态轮廓方向约束力。运动边缘点约束力防止轮廓向内产生过收缩, 而动态轮廓方向约束力则是利用轮廓的方向, 控制轮廓的收缩方向。我们将垂直于当前轮廓点 v_i 向内收缩的方向定义为

$$\theta_N(v_i) = \theta_C(v_i) + \pi/2 \quad (16)$$

其中 $\theta_C(v_i)$ 是按逆时针计算出来的当前轮廓点 v_i 的轮廓方向, 根据其前一轮廓点 v_{i-1} 和后一轮廓点 v_{i+1} 来进行估计

$$\theta_C(v_i) = \frac{1}{2} \left(\arctan 2 \left(\frac{y_{i+1} - y_i}{x_{i+1} - x_i} \right) + \arctan 2 \left(\frac{y_i - y_{i-1}}{x_i - x_{i-1}} \right) \right) \quad (17)$$

在这里, (x_{i-1}, y_{i-1}) , (x_i, y_i) 和 (x_{i+1}, y_{i+1}) 分别表示轮廓点 v_{i-1} , v_i 和 v_{i+1} 的二维空间坐标, $\arctan 2$ 表示的角度范围为 $[0, 2\pi]$ 。根据计算得到的 $\theta_N(v_i)$, 指定当前轮廓点 v_i 的牵引方向, 如图 1 所示。

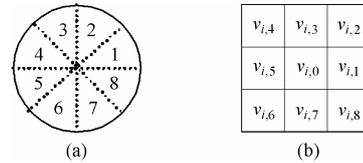


图 1 根据 $\theta_N(v_i)$ 划分的扇区以及当前轮廓点 $v_{i,0}$ 及其 8 邻域 (a) 根据 $\theta_N(v_i)$ 划分的扇区 (b) 当前轮廓点 $v_{i,0}$ 及其 8 邻域

Fig.1 Sectors partitioned according to $\theta_N(v_i)$ and the current contour point and its 8-neighbors (a) Sectors partitioned according to $\theta_N(v_i)$ (b) Current contour point and its 8-neighbors

将 $\theta_N(v_i)$ 的变化范围划分成 8 个扇区 $\text{sec}_k (k = 1, 2, \dots, 8)$, 每个扇区 $\text{sec}_k (k = 1, 2, \dots, 7)$ 分别对应当前轮廓点 $v_{i,0}$ 以及 $\theta_N(v_i)$ 方向上的两个点 $v_{i,k}$ 和 $v_{i,k+1}$; 当 $k = 8$ 时, $\theta_N(v_i)$ 方向上的两个点为 $v_{i,8}$ 和 $v_{i,1}$ 。将每个扇区所对应的轮廓点集合(包括当前轮廓点 $v_{i,0}$ 以及 $\theta_N(v_i)$ 方向上的两个点)记为 C 。那么, 加入轮廓方向信息的彩色图像约束力可以写为

$$E_{\text{coledge}}(v_{i,k}) = \begin{cases} \frac{\min - \text{mag}(v_{i,k})}{\max - \min}, & v_{i,k} \in C \\ 0, & \text{其他} \end{cases} \quad (18)$$

将图像转换到 YUV 颜色空间对图像进行指数滤波, \max 和 \min 分别表示当前轮廓点及其 8 邻域轮廓点彩色梯度强度的最大值和最小值。根据上面所讲述的指数滤波的方法, 按式(19)计算彩色梯度强度:

$$\text{mag}(v_i) = \max(M_V^i, M_U^i, M_V^i) \quad (19)$$

在对初始轮廓进行收缩时，首先将运动区域与当前帧的边缘图相与，得到运动边缘点。然后，对每一个轮廓点判断它在运动边缘图中相应的点处是否已经注册；若已注册，认为此轮廓点已经达到了它最后的位置，不再运动；若没有注册，判断它在运动边缘图相应的位置处是否为运动边缘点，若是，则对此轮廓点注册，并认为它已达到最后的位置，不再运动，否则，采用式(15)和式(18)计算当前点和其周围 8 邻域的能量函数，将轮廓点移动到能量函数最小的位置上去，采用能量最小化准则平滑连接真实运动轮廓点之间的断点。

4 实验结果与分析

为了验证算法的有效性，本文针对刚体、非刚体，单目标、多目标，室内、室外多种图像测试序列，以动态轮廓逼近目标真实轮廓的程度来衡量算法的性能，并计算多帧图像轮廓收缩算法的平均迭代计算时间 \bar{T} 。动态轮廓和目标真实轮廓之间的相对误差定义为

$$\Delta = \frac{(TO \cup AC) - (TO \cap AC)}{TO} \quad (20)$$

其中 TO 表示真实目标区域的面积， AC 表示用动态轮廓收缩的目标区域面积。 Δ 越小，表示动态轮廓越逼近目标的真实边界。在奔腾4 2.4G微处理器上，用matlab语言验证算法的有效性，对测试图像序列分别等间隔地手工分割出目标区域各30帧图像，计算其相对误差 Δ 和平均迭代时间 \bar{T} 。

实验结果如图2-图4所示，参数 $\alpha=1$, $\beta=1$, $\gamma=1.5$ 。其中，图2为室内刚体多目标的情况，背景较为简单。图2(b)是采用传统的greedy算法进行轮廓收缩的结果，可以看出，左边汽车的尾部和前部均出现了不同程度的过收缩现象，右边汽车的顶部轮廓和目标的真实边界有少许偏差。而采用本文提出的MEDG轮廓收缩算法，可以很好地逼近目标的真实轮廓，结果如图2(d)所示。

图3为室内柔体单目标的情况，背景较为复杂，目标会产生投射阴影，并且目标的手臂处有一定的凹轮廓处。采用greedy算法会造成动态轮廓向外扩展的情况，如图3(b)的目标的手臂和肩膀部分。这主要是因为目标外部也存在较强的边缘，而greedy算法并没有考虑轮廓收缩的方向，从而造成轮廓收敛到错误的局部最小值。而采用MEDG算法考虑了轮廓的方向性信息，针对凹轮廓处也不会造成轮廓向外扩展，可以取得很好的效果，见图3(d)。

图4为室外刚体多目标的情况。采用本文的算法也可以取得比greedy算法优越的结果。

表1为对上述3种测试视频序列进行greedy及MEDG算法轮廓收缩总体性能以及计算时间的比较结果。可见，采用MEDG算法可以获得比greedy算法更优越的总体性能。而且，由于MEDG算法采用运动边缘信息作为先验条件，阻止了动态轮廓的过收缩过程，利用方向性信息防止动态轮廓向外扩展，从而在计算速度上也有一定的提高。

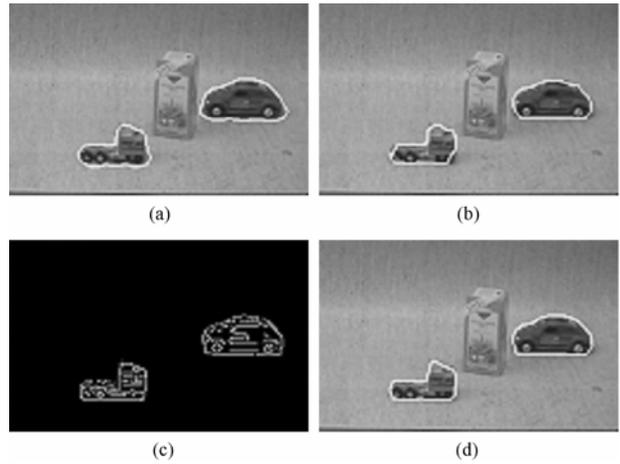


图2 “Truck”图像序列 greedy, MEDG 算法轮廓收缩比较结果
(a) 初始轮廓 (b) Greedy 算法轮廓收缩
(c) 指数滤波提取运动边缘 (d) 本文 MEDG 算法轮廓收缩

Fig.2 Comparison between greedy and MEDG algorithm in “Truck” video sequence
(a) Initial contour (b) Convergence result obtained by greedy algorithm
(c) Moving edge extracted by exponential filter
(d) Convergence result obtained by MEDG algorithm

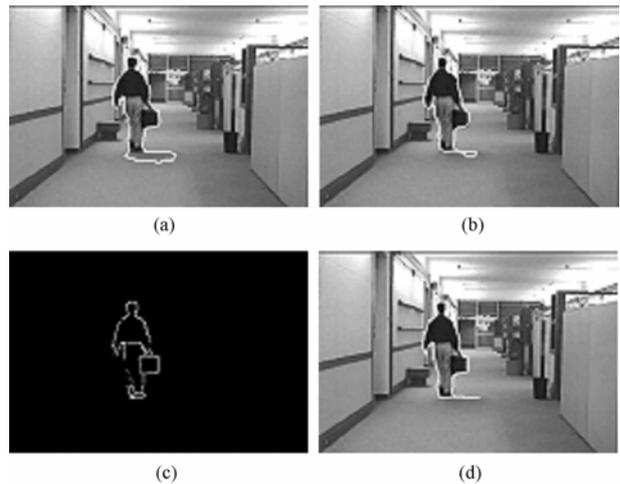


图3 “Hall-monitor”图像序列 greedy, MEDG 算法轮廓收缩比较结果
(a) 初始轮廓 (b) Greedy 算法轮廓收缩
(c) 指数滤波提取运动边缘 (d) 本文 MEDG 算法轮廓收缩

Fig.3 Comparison between greedy and MEDG algorithm in “Hall-monitor” video sequence
(a) Initial contour (b) Convergence result obtained by greedy algorithm
(c) Moving edge extracted by exponential filter
(d) Convergence result obtained by MEDG algorithm

表 1 轮廓收缩算法相对误差和平均计算时间比较

Tab.1 Comparison of relative error and average computational time between greedy and MEDG algorithm

	测试序列(参数 $\alpha=1, \beta=1, \gamma=1.5$)					
	Truck		Hall-monitor		Highway	
	Δ (%)	\bar{T} (ms)	Δ (%)	\bar{T} (ms)	Δ (%)	\bar{T} (ms)
Greedy	10.57%	33.8	15.63%	44.8	13.19%	41.2
MEG	9.14%	11.4	12.51%	32.6	9.01%	14.7

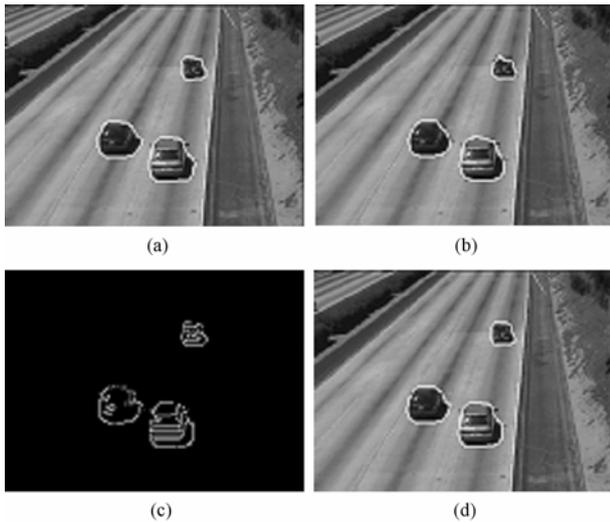


图 4 “Highway”图像序列 greedy, MEDG 算法轮廓收缩比较结果
(a)初始轮廓 (b)Greedy 算法轮廓收缩
(c) 指数滤波提取运动边缘 (d) 本文 MEDG 算法轮廓收缩

Fig.4 Comparison between greedy and MEDG algorithm in “Highway” video sequence

- (a) Initial contour (b) Convergence result obtained by greedy algorithm
(c) Moving edge extracted by exponential filter
(d) Convergence result obtained by MEDG algorithm

5 结束语

在研究多种运动目标轮廓检测算法的基础上, 采用指数滤波精确估计彩色图像梯度强度, 提高了抗噪声干扰能力和目标边界定位的精度。并利用运动边缘作为先验条件防止动态轮廓向内过收缩, 同时考虑轮廓的方向性信息, 针对凹轮廓处也不会造成轮廓向外扩展, 从而有效地防止了动态轮廓收敛到错误边界。理论分析和实验证明, 对复杂背景及不重叠物体, 可以精确定位并提取出各个运动物体的外轮廓。此外, 该算法复杂度较低, 利于实现实时图像处理。目前, 本文提出的算法是建立在初始轮廓在目标真实轮廓以外的基础上的。如果目标部分区域颜色值和背景颜色值十分接近而

造成运动边缘以及区域检测的局部失误, 则必然会造成轮廓收缩产生一定的误差。这是基于边缘的轮廓收缩算法中需要设法解决的问题, 也是今后研究的内容之一。

参考文献

- [1] Kass M, Witkin A, Terzopoulos D. Snake: Active contour models. *International Journal of Computer Vision*, 1988, 1(4): 321-331.
- [2] Rhee P K, La C W. Boundary extraction of moving objects from image sequence. Proc. of the IEEE Region 10 Conference TENCON99, Cheju-Korea, 1999, 1(15-17): 621-624.
- [3] Kim Munchurl, Choi Jae Gark, Kim Daehee, Lee Hyung, Lee Myoung Ho, Ahn Chieteuk, Ho Yo-Sung. A VOP generation tool: Automatic segmentation of moving objects in image sequences based on spatio-temporal information. *IEEE Trans.on Circuit and System for Video Technology*, 1999, 9(8): 1216-1226.
- [4] Cucchiara R, Grana C, Piccardi M, Prati A. Detecting moving objects, ghosts, and shadows in video streams. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2003, 25(10): 1337-1342.
- [5] Williams D J, Shah M. A fast algorithm for active contours and curvature estimation. *CVGIP: Image understanding*, 1991, 55(1): 14-26.
- [6] Amini A, Whemouth T, Jain R. Using dynamic programming for solving variational problems in vision. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1990, 12(9): 855-867.
- [7] 李玉山, 杨莉, 刘洋, 张大朴. 多个运动物体的外轮廓自动提取. 智能视觉监测研究进展, 北京: 科学出版社, 2003: 87-91.
- [8] Shen Jun, Castan S. An optimal linear operator for edge detection. Proc. CVPR'86, Miami, 1986: 109-114.

刘洋: 男, 1980年生, 博士, 主要从事计算机运动视觉算法与实现等方面的研究.

李玉山: 男, 1945年生, 西安电子科技大学电路CAD所所长, 教授, 博士生导师, 主要从事视觉芯片设计, 机器视觉等方面的研究.

张大朴: 男, 1977年生, 博士, 主要从事摄像机标定、运动机器视觉等方面的研究.