

一种新的 B-Snake 算法在目标轮廓跟踪中的应用

邱书波^{1,2)} 王化祥¹⁾ 梁志伟²⁾

¹⁾(天津大学电气与自动化工程学院,天津 300072) ²⁾(山东轻工业学院自动化研究所,济南 250100)

摘要 用 Snake 算法跟踪移动目标时,有时会产生目标轮廓与背景中另一些物体靠近的现象,如果不加约束,那么一些控制点将可能因被吸引到背景边缘而产生错误。另外,目标产生移动或旋转时,因轮廓控制点会产生交叉而导致轮廓跟踪失败。为克服此类问题,提出了一种新的有效的目标跟踪算法,可以跟踪静止或运动目标的 2 维轮廓曲线。该算法先利用三次 B 样条曲线连接控制点,以初始化物体的边界,然后通过 Snake 最小能量函数把控制点吸引到轮廓的边界。在 Snake 跟踪算法中,由于是采用和下一点相连向量与控制中心相连向量的夹角正弦值的正负来避免控制点的交叉,而控制中心的位置则通过 $\alpha\text{-}\beta\text{-}\gamma$ 滤波能有效地估计出来,因而能准确地跟踪运动的目标。该算法运用到静止的或移动的目标轮廓跟踪的实验证明,该算法不仅对目标轮廓跟踪具有较好的效果,且运算时间短、复杂度低。

关键词 Snake 算法 B 样条 $\alpha\text{-}\beta\text{-}\gamma$ 滤波 目标跟踪 活动轮廓

中图分类号: TP391.41 **文献标识码**: A **文章编号**: 1006-8961(2005)05-0585-05

Tracking Object Contour Using A Novel B-Snake Algorithm

QIU Shu-bo^{1,2)}, WANG Hua-xiang¹⁾, LIANG Zhi-wei²⁾

¹⁾(School of Electrical Engineering & Automation, Tianjin University, Tianjin 300072)

²⁾(Automation Lab., Shandong Institute of Light Industry, Jinan 250100)

Abstract In some cases, the object contour under observation may be adjacent to some other objects in the background when using Snake algorithm to track moving object. If not be constrained, some control points might be drawn to the background boundary, which result in an error. On the other hand, if the object is moved or rotated, contour control points may result in crossing, which leads to failure of contour tracking. Thus, a novel and efficient algorithm is proposed in this paper to solve those problems mentioned above. The proposed algorithm can be used to track 2D contour for both static and moving objects. The contour is initialized by connecting control points using cubic B-Spline curve. Then control points are drawn to the contour boundary using minimum energy function. The B-Snake tracking algorithm adopts the sine value of the angle between the vector representing the next point and the vector representing the control center point to prevent control points from crossing. And the position of the control center point can be estimated efficiently by $\alpha\text{-}\beta\text{-}\gamma$ filtering. Therefore, moving object contour can be tracked precisely. Experimental results of applying the proposed algorithm in static or moving human head contour tracking show that the proposed algorithm yields good effect and saves time.

Keywords Snake algorithm, B-splines, $\alpha\text{-}\beta\text{-}\gamma$ filter, object tracking, active contour

1 引言

在计算机视觉中,移动或静止物体轮廓的跟踪是一个重要的研究领域。所谓跟踪就是给定一个目标的图像序列,通过跟踪目标物体的轮廓,在图像中

找出感兴趣的部分,例如物体随时间变化的形状和当前的位置等一些重要信息。在跟踪目标轮廓的算法中,先后出现了可变型模板跟踪^[1]和 Snake 跟踪^[2]两种比较典型的算法。可变型模板跟踪的不足之处在于它依赖于被跟踪物体形状的先验知识,且当有轻微的扰动时,物体的轮廓就会被歪曲。

收稿日期:2004-07-05;改回日期:2004-11-30

第一作者简介:邱书波(1963~),男,1989年获山东工业大学(现山东大学)硕士学位,现为天津大学在职博士生,山东轻工业学院教授,硕士生导师。主要从事计算机视觉检测及智能仪器仪表理论研究,完成国家高技术研究计划课题一项,获山东省科技进步奖二等奖一项。
E-mail: qsb2000@jnncc.com

Snake 算法(又称活动轮廓模型)是用能量最小法来达到计算机视觉跟踪的目的,即当给定一个运动物体的一系列按时间变化的图像,则物体的形状和目前的位置等重要信息就能被获得。Snake 算法及其变形在静止图像处理方面主要用于检测图像的边缘、曲率、边界以及区域分割和骨架提取等^[2-4]。用 Snake 算法对时间序列图像进行目标跟踪的思想源于 Kass 提出的算法^[2],且 Kass 先用 Snake 算法跟踪人的唇动。Terzopoulos 和 Waters 则利用多个 Snake 跟踪脸部特征^[5]。Curwen 等人利用 Snake 算法跟踪 3 维物体的封闭轮廓线^[6]。Snake 算法从 1987 年由 Kiss 等提出后,先后又出现了 AI-Snake、G-Snake、AI-EigenSnake 等形式^[7-9],应用表明,Snake 算法适合跟踪刚体和非刚体的运动^[10,11]。

本文中,作者从普遍适用的角度出发,提出了一个新的有效的跟踪算法,其在没有任何先验知识的前提下,可以跟踪静止或运动的 2 维物体的轮廓曲线。该算法利用三次 B 样条曲线连接控制点来初始化物体的边界,然后通过 Snake 算法的最小能量函数将控制点吸引到边界曲线。在 Snake 跟踪算法中,则采用和下一点相连向量与控制中心相连向量的夹角正弦值的正负来避免控制点的交叉。在轮廓的跟踪中,控制中心的位置通过 α - β - γ 滤波能有效地估计出来,最后通过人头部轮廓跟踪的实验证明了算法的有效性。

2 Snake 算法

用于轮廓跟踪的最小能量函数是外力和内力的加权和,其中内力由 Snake 曲线的形状计算得到,而外力则是从图像中获得或是从更高级的图像处理中得到。参数化的 Snake 曲线被定义为 $V(s) = [x(s), y(s)]$,这里 $x(s)$ 和 $y(s)$ 是轮廓的坐标值,其中弧 $s \in [0, 1]$,其最小化的能量函数可以写成下面的形式^[1]:

$$\begin{aligned} \hat{E}_{\text{Snake}} &= \int_0^1 E_{\text{Snake}}(V(s)) ds \\ &= \int_0^1 \{ [E_{\text{int}}(V(s))] + [E_{\text{image}}(V(s))] + \\ &\quad [E_{\text{cons}}(V(s))] \} ds \end{aligned} \quad (1)$$

其中,Snake 曲线因为弯曲产生的内部能量可以表示为

$$E_{\text{int}} = \alpha(s) \left| \frac{dV(s)}{ds} \right|^2 + \beta(s) \left| \frac{d^2V(s)}{ds^2} \right|^2 \quad (2)$$

式中,第 1 项为 Snake 曲线内部能量函数的一阶项,用于控制轮廓线的长度,并使其尽可能短;第 2 项为

Snake 曲线内部能量函数的二阶项,其用于控制轮廓线曲率,使轮廓线尽可能平滑,而 Snake 曲线外部能量 $E_{\text{ext}} = E_{\text{image}} + E_{\text{cons}}$ 来自图像和外部的约束,因为 $E_{\text{ext}}(V(s))$ 为外部能量函数,多以图像的梯度充当,它是动态轮廓线的收缩限制力,其用于控制轮廓线,使其接近梯度变化大的区域。该能量定义为

$$E_{\text{ext}}[V(s)] = -|\nabla I(V(s))|^2 \quad (3)$$

变分学中 Euler-Lagrange 条件表明,最小化能量 \hat{E}_{Snake} 的曲线 $V(s)$ 必须满足

$$\frac{d}{ds} E_{V_s} - E_{V_s} = 0 \quad (4)$$

这里, E_{V_s} 是 E 对 $dV(s)/ds$ 的偏微分,而 E_V 是 E 对 $V(s)$ 的偏微分。利用式(2),并令 $E_{\text{ext}} = E_{\text{image}} + E_{\text{cons}}$,则式(3)可以简化为

$$\begin{aligned} -\frac{d}{ds} \left[\alpha(s) \frac{dV(s)}{ds} \right] + \frac{d^2}{ds^2} \left[\beta(s) \frac{d^2V(s)}{ds^2} \right] + \\ \nabla E_{\text{ext}}[V(s)] = 0 \end{aligned} \quad (5)$$

为了求解 Euler-Lagrange 方程,可以利用解的一个初始条件。这样就可以建立如下的随时间演变的方程:

$$\begin{aligned} \frac{\partial V(s,t)}{\partial t} - \frac{\partial}{\partial s} \left[\alpha(s) \frac{\partial V(s,t)}{\partial s} \right] + \\ \frac{\partial^2}{\partial s^2} \left[\beta(s) \frac{\partial^2 V(s,t)}{\partial s^2} \right] + \nabla E_{\text{ext}}[V(s,t)] = 0 \end{aligned} \quad (6)$$

如果 $\frac{\partial V(s,t)}{\partial t} = 0$,那么就得到了问题的解。

3 改进的 B-Snake 跟踪算法

很明显,尽管用更多的边缘控制点会得到目标轮廓好的描述,但另一方面则会导致计算量的增加。因此要兼顾这两方面,在 B-Snake 控制算法^[10]中,用适当的控制点来初始化目标的边界,其处理速度可达实时处理的要求。

在算法中,仅仅用 Snake 曲线的外部能量来跟踪目标轮廓,也就是仅仅依据边界检测,在一帧图像中依据像素灰度的梯度在边缘任何点是局部最大这个事实,容易得出如下图像外部能量的表达式

$$e = -\|\nabla(G_\sigma \times I)\| \quad (7)$$

这里, I 是像素的灰度值, G_σ 是方差为 σ 的 1 维高斯函数, ∇ 是 1 维 Sobel 边缘检测算子。

当跟踪的目标移动时,有时它的轮廓与背景中另一些物体靠近,如果不加约束,则某些控制点就可

能被吸引到背景的边缘,那样就会产生错误。为了克服这个问题,可在每个控制点的搜索区域加上一个阈值。如果在这个区域中没有找到边界,就使这个控制点无效,则随后用它左边的控制点得到边界,而在序列图像的下一幅图像的目标轮廓跟踪中,则需在邻近区域增补一个控制点,以保持控制点的总数不变。

方法中用到的三次 B 样条定义如下

$$C_i[t] = \frac{1}{6} [t^3 \quad t^2 \quad t \quad 1] \begin{bmatrix} -1 & 2 & -2 & 1 \\ 2 & -5 & 2 & 0 \\ -2 & 0 & 2 & 0 \\ 1 & 3 & 3 & 0 \end{bmatrix} \begin{bmatrix} P_{i-1} \\ P_i \\ P_{i+1} \\ P_{i+2} \end{bmatrix} \quad (8)$$

P_i 是控制点,为了构成一个闭合的样条曲线,起始的 3 个点用了两次。在跟踪模型中,每帧图像中的跟踪目标初始轮廓可通过 B 样条来构造,即首先用 $\bar{l} = \frac{1}{n} \sum_{i=1}^n l_i$ 表示这些曲线段的平均长度,然后依据 \bar{l} 重新分配控制点,并使每两个控制点的轮廓长度为 \bar{l} ,在每一次分配完后就计算它的最小能量,最后得到一组具有最小能量的控制点,而控制点的数目可以依据需要来进行调节,这样就构建了自适应的 B-Snake 模型。

在相邻两幅图像控制点的搜索过程中,有时控制点会交叉(如图 1 所示),为了避免这种情况,作者定义了向量的夹角 θ (如图 2 所示),图中一个向量表示该点指向下一个点的向量,由于目标曲线的质心被所有的控制点包围,所以质心应该在一个控制点指向下一个控制点向量的同一侧,而另一个向量则表示该点指向物体质心的向量。如果该 θ 的正弦值大于零,则控制点没有交叉;如果小于零,则说明控制点交叉,这样就要更换两个控制点的位置。

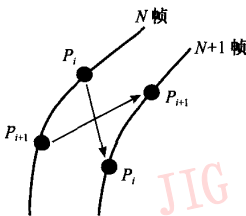


图 1 搜索中的交叉

Fig. 1 Crossing in the search process for control point

$\sin\theta$ 的表达式为

$$\sin\theta = \frac{| \overrightarrow{P_i P_{i+1}} \times \overrightarrow{P_i O} |}{| \overrightarrow{P_i P_{i+1}} | | \overrightarrow{P_i O} |} = \frac{| \mathbf{M} \times \mathbf{N} |}{| \mathbf{M} | | \mathbf{N} |}$$

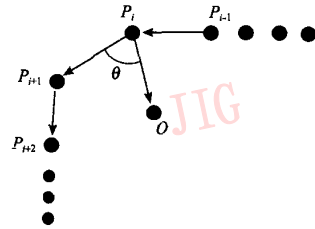


图 2 控制中心和夹角 θ

Fig. 2 The control center point and the included angle θ

$$= \frac{\begin{vmatrix} M_x & M_y \\ N_x & N_y \end{vmatrix}}{\sqrt{(M_x^2 + M_y^2)} \sqrt{(N_x^2 + N_y^2)}} \quad (9)$$

在每一个采样周期内(大约 100ms),可以设目标的运动加速度是恒定的,这里由于加加速度被看成高斯噪声,因此可以采用 Kalman 滤波中的 α - β - γ 滤波来估计物体的位置。假设采样周期内物体沿某一直线运动,第 k 个周期,即 t_k 时刻的位移、速度、加速度、加加速度分别为 d_k 、 v_k 、 a_k 、 g_k ,采样周期为 T ,则物体的运动方程为

$$\begin{cases} d_k = d_{k-1} + v_{k-1}T + \frac{a_{k-1}T^2}{2} \\ v_k = v_{k-1} + a_{k-1}T \\ a_k = a_{k-1} + g_{k-1}T \end{cases} \quad (10)$$

若取状态向量为 $\mathbf{X}_k = [d_k \quad v_k \quad a_k]^T$,则状态方程为

$$\mathbf{X}_k = \Phi \mathbf{X}_{k-1} + \Gamma g_{k-1} \quad (11)$$

Φ 和 Γ 是由式(10)得到的矩阵。

如果只对物体的位置做观测,观测值为 $Z_k = S_k + n_k$,则观测方程为

$$Z_k = H_k X_k + n_k \quad (12)$$

式中, $H_k = [1 \quad 0 \quad 0]$,向量 n_k 是观测噪声。若取 $K_k = [\alpha \quad \beta \quad \gamma]^T$,则由 α - β - γ 滤波可得

$$\hat{X}_k = \Phi \hat{X}_{k-1} + K_k [Z_k - H \Phi \hat{X}_{k-1}] \quad (13)$$

如此即可估计出物体的运动位置 \hat{S}_k 。

在 B-Snake 轮廓跟踪算法中,在两次采样的间隔时间内,可假设物体没有或仅仅发生轻微的旋转,这时能用物质心的运动代表每个控制点的运动,而轮廓质心的坐标则可通过轮廓上的每个点的坐标计算得到

$$\begin{cases} X = \left(\sum_{i=1}^n \sum_{j=1}^m x_j B[i,j] \right) / \left(\sum_{i=1}^n \sum_{j=1}^m B[i,j] \right) \\ Y = \left(\sum_{i=1}^n \sum_{j=1}^m y_j B[i,j] \right) / \left(\sum_{i=1}^n \sum_{j=1}^m B[i,j] \right) \end{cases} \quad (14)$$

其中, $B[i,j] = \begin{cases} 1 & \text{轮廓上的点} \\ 0 & \text{图中另外点} \end{cases}$

跟踪目标轮廓时,首先在每一帧 $m \times n$ 大小的图像中计算目标质心的位置,然后根据 α - β - γ 滤波得到下一帧图像中目标轮廓的质心位置,最后依据两帧连续图像质心位置的不同,就可以得到每个控制点的预测位置。

4 试 验

为验证本文算法的有效性,选择一组人头图像进行了目标轮廓跟踪实验。在试验中,作者用本文算法跟踪一个静止和运动人的头部形状,首先用嵌入式计算机通过图像采集卡采集摄像机传来的图像序列,计算机 CPU 为 P4/1.7G,图像卡为 PC104 标准的 VT121,摄像机为标准黑白摄像机 MTV1881EX。

所采集的图像序列,经过 Sobel 边缘检测后,再采用手动设置初始的控制点,然后用式(8)的三次样条插值形成一条闭合的初始轮廓曲线,再经过新的 B-Snake 算法就可以准确地跟踪到人的头部形状。图 3 的第 1 幅图是用新的 B-Snakes 算法得到的头部初始化轮廓曲线,第 2 幅图是经过 50 次迭代运算的头部形状逼近曲线,第 3 幅图是达到最小能量的头部最终形状曲线。从图 3 可以看出,改进的 B-Snake 算法可以很好地跟踪到静止头部的形状。



图 3 B-Snake 算法对静止头部曲线轮廓的逼近
Fig.3 Approach a static head contour by B-Snake

在对运动物体的跟踪中,当物体以随机的速度移动或发生稍微的旋转,改进后的 B-Snake 算法首先对连续帧的图像质心位置进行估计计算,然后质心带动控制点一起移动,就能初步跟踪到物体的轮廓,最后 B-Snake 最小能量曲线只需经过迭代 20 次就可以达到轮廓的边界,其效果如图 4 所示。而普通的 B-Snake 算法由于对每一幅图像只是利用能量最小来逼近边界,因此每幅图像需要迭代约 200 次,其效果如图 5 所示。在试验中,经过改进的 B-Snake 算法的跟踪时间,除了第 1 帧图像外,其他图像采用目标质心运动估计可以达到 0.1s,而普通的 B-Snake

算法的运算时间为 0.5s。由此可见,改进后的 B-Snake 算法在跟踪算法的复杂度上大大降低,同时跟踪时间也随之减少。



图 4 新 B-Snake 算法对运动中的头部轮廓曲线的逼近
Fig.4 Approach a moving head contour by novel B-Snake



图 5 普通 B-Snake 算法跟踪示意
Fig.5 Illustration of general B-Snake algorithm in contour tracking

另一方面,由于本文采用了 α - β - γ 滤波和避免交叉点的 θ 算法,即通过连续帧交叉控制点的互换来避免轮廓搜索中的曲线交叉,而普通 B-Snake 算法在跟踪动态目标轮廓时,则容易发生交叉。图 4 的前两幅图说明了改进的 B-Snake 算法对运动中的头部轮廓的跟踪,由图 4 可见,改进算法能很好地跟踪到头部的形状,而第 3 幅图则说明了当背景发生变化时,由于采用了有阈值的搜索区域,因此作者将其定义成以该控制点为中心 5×5 大小像素区域,本算法仍能有效地跟踪到头部的形状;而对于普通的 B-Snake 算法,从图 5 中可以明显看出,即使背景没有发生变化,只要头部稍微有些移动,便会使控制点产生交叉,这将容易造成跟踪曲线发生畸变而导致跟踪失败。从两者的对比中不难看出,新的 B-Snake 算法不但对运动的目标有良好的适应性,且跟踪精度高。

5 结 论

本文提出一种新的 B-Snake 目标轮廓跟踪算法,该算法是通过 Snake 最小能量来逼近物体的轮廓,并且每一帧图像中的目标轮廓可以被三次 B 样条曲线准确地表示。由于 B-Snake 算法依据最小能量原则最终可以跟踪到目标的形状,因此在运动目标的轮廓

跟踪中,作者对 B-Snake 算法进行了改进,即通过 θ 算法和阈值搜索避免了交叉,同时采用了 α - β - γ 滤波进行目标质心的位置估计,以便能准确、快速地跟踪到目标的轮廓。实验证明,该算法是有效的。

参考文献 (References)

- 1 Lai K F, Chin R T. Deformable contours: modeling and extraction [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1995, 17(11):1084 ~ 1090.
- 2 Kass Michael, Witkin Andrew, Terzopoulos Demetri. Snakes: Active contour models[J]. International Journal of Computer Vision, 1987, 1(4):321 ~ 331.
- 3 Shih Frank Y, Zhang Kai. Efficient contour detection based on improved snake model [J]. International Journal of Pattern Recognition and Artificial Intelligence, 2004, 18(2):197 ~ 209.
- 4 Pavlids T, Liow Y T. Intergrating region growing and edge detection [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1990, 12(3):225 ~ 233.
- 5 Terzopoulos D, Waters K. Analysis of dynamic facial images using physical and anatomical models [A]. In: Proceedings of third International Conference on Computer Vision (ICCV '90) [C], Osaka, Japan, 1990: 727 ~ 732.
- 6 Curwen R M, Blake A, Cipolla R. Parallel implementation of lagrangian dynamics for real-time Snakes [A]. In: Proceedings British Mathine Vision Conference [C], London, 1991:29 ~ 35.
- 7 Horace H S, Shen Ding-gang. An affine-invariant active Contour model (AI-Snake) for model-based segmentation [J]. Image and Vision Computing, 1998, 16(2):135 ~ 146.
- 8 Xu Chenyang, Prince Jerry L. Snakes, Shapes and Gradient vector flow [J]. IEEE Transactions on Image Processing, 1998, 7(3): 359 ~ 369.
- 9 Xue Zhong, Li Stan Z, Teoh Eam Khwang. AI-EigenSnake: An affine-invariant deformable contour model for object matching [J]. Image Vision Computing, 2002, 20(2):77 ~ 84.
- 10 Wang Yue, Teoh Earn Khwang, Shen Dinggang. Structure-adaptive B-Snake for segmenting complex objects [A]. In: IEEE International Conference on Image Processing (ICIP '01) [C], Thessaloniki, Greece, 2001, 2:769 ~ 772.
- 11 Patrick Brigger, Jeff Hoeg, Michael Unser. B-Spline Snakes: A flexible tool for parametric contour detection [J]. IEEE Transactions on Image Processing, 2000, 9(9): 1484 ~ 1496.