

复杂物体轮廓提取

徐晓刚 于金辉 马利庄

(浙江大学 CAD&CG 国家重点实验室, 杭州 310027)

摘要 图象分割是图象处理中的一项重要工作,虽然目前手工与自动相结合的分割方法已在实际工作中得到了广泛应用,但由于图象的多义性和复杂性,许多分割工作仍无法利用计算机来自动完成.为了解决这一问题,根据图象经 Marr 变换后的特征,采用新的判断准则,提出了一种复杂物体边缘定位算法,用该算法可对具有尖角特征的物体轮廓进行快速准确地提取,同时由于利用了矢量化方法来消除毛刺,从而使跟踪获得的边界更符合物体的实际轮廓特征.通过对多种图象的实验表明,该方法十分有效.

关键词 复杂轮廓提取 Snake 算法 基于动态规划图搜索算法

中图法分类号: TP751.1 文献标识码: A 文章编号: 1006-8961(2001)05-0455-05

Extraction of Complex Object Contour

XU Xiao-gang, YU Jin-hui, MA Li-zhuang

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

Abstract Image segment plays an important role in the field of image processing, and currently the hybrid approach combining the manual and automatic methods is widely used in segment practice. In this paper we present an algorithm capable of locating the target object contour of sharp tips accurately in the interactive rate. Considering that the edge are usually on the zero-crossing points after Marr transformation for most images, existing techniques tend to give undesirable results because the energy path containing more points is given less priority. In our method we specify a pointer to point the current point on a path of interest, when the energy of current path is less than the energy for the previous point, we check n latest points in the current path instead of checking only one point as existing techniques do, and, if more than m point ($m \leq n$) is zero-crossing, the pointer of the point is updated, otherwise, the pointer remains unchanged. Using this criterion we can insert new seeds automatically near the tips of the target object and the burr is eliminated by a vectorization approach. The final contour traced out fits the feature of the target object well and the effectiveness of our method is demonstrated by examples shown in the paper.

Keywords Contour detection, Snake, Graph searching formulation of dynamic programming.

0 引言

众所周知,图象分割是一项广泛应用的图象处理技术,但是由于图象的多义性和复杂性,许多分割工作仍无法依靠计算机来自动完成,而手工分割又存在工作量大和定位不准确的难题,因此,人们提出

了一些人工交互与计算机自动定位相结合的方法,并利用各自的优势,来实现目标轮廓的快速定位.纵观这些方法,它们大致可以归结为如下两类:一类为 Snake 算法或 Active Contour Models 算法^[1,2],这类算法需要首先给出初始的轮廓,然后进行迭代,使轮廓沿能量降低的方向靠近,最后即得到一个优化的边界,这里能量函数包括内外力两方面,如边界曲

率和梯度,但由于用户无法估计迭代的最后结果,且应用 Snake 算法往往需要进行多次的交互工作,特别当目标比较复杂时,或与其他物体靠得较近时,其初始的轮廓则不易确定,因而迭代的结果往往不能达到要求;另一类是基于动态规划图搜索算法^[3~8],这种图搜索算法是在全图范围内通过寻找优化的边界来进行目标轮廓的定位.与 Snake 算法不同,这种图搜索算法不是通过迭代初始轮廓即降低能量的方式,而是通过分步优化能量函数的方法来获得边界,但图搜索算法也需要用一个初始的模板来约束搜索.

Mortensen 提出了一种更方便的交互提取目标轮廓的算法“Intelligent Scissors (IS)”^[3,7],其在搜索轮廓时,虽与一般的图搜索算法有些类似,但可以更好地利用人的交互优势.该方法的基本思想是在设定种子点后,计算出图上各点到种子点的最小能量.由于边缘点多为零交叉点,其能量较邻点小,因而能使能量图呈“总线”结构,其中边缘即为代表低能量的“总线”,从而用该方法在交互引导过程中即能自动勾勒出边缘.虽然对于轮廓比较平滑的目标,文献[3]及文献[7]的方法可以获得比较好的效果,但在处理具有尖锐边角的目標时,这种方法对尖锐边角往往不能准确地定位,从而造成“割角”.为克服 IS 算法中的缺点,通过分析能量的迭代过程,本文提出了一种改进算法,该算法可对复杂物体的边缘进行比较准确的定位,同时可针对跟踪过程产生的“毛刺”现象,采用矢量化方法来消除毛刺,从而使获得的边界具有比较好的视觉效果.

1 算 法

能量的计算方法有多种,如以轮廓线的亮度、曲率和长度来计算,或以梯度值、梯度方向和闭合曲线的曲率来计算,但从运算速度考虑,本文采用零交叉点、梯度值、梯度方向来作为能量评价函数因子.

1.1 能量的计算方法

$$l(p, q) = w_z f_z(q) + w_D f_D(p, q) + w_G f_G(q) \quad (1)$$

式中, $f_z(q)$ 、 $f_G(q)$ 、 $f_D(p, q)$, 分别表示零交叉点、梯度值和梯度方向; q 为 p 的邻点; w_z 、 w_D 、 w_G 为权值, 本文计算中采用文献[4]中的取值方法, 且对大多数图象, 该权值分别取为 0.43, 0.14, 0.43.

1.1.1 $f_z(q)$ 的计算

$$f_z(q) = \begin{cases} 0 & \text{if } I_L(q) = 0 \\ 1 & \text{if } I_L(q) \neq 0 \end{cases} \quad (2)$$

其中, I_L 是对原图进行 Marr 变换得到的结果. 对于 I_L , 零交叉点的定义如下: (1) 该点为 0; (2) 若相邻两点 p, q 为从正到负, 且 $|I_p - I_q| > T$, 则取绝对值小者为零交叉点. 对大多数图象, 阈值 T 选 1.66 较有效. 实验结果表明, 采用该方法可以克服一定噪声的影响, 且比文献[3]的算法更易跟踪到物体的边界.

1.1.2 $f_G(q)$ 的计算

虽然计算梯度时, 可以采用不同的算子, 但对最终计算所得的能量影响较大的是模板的宽度, 本文采用高斯函数来计算梯度. 令 I_x, I_y 表示 x, y 方向的梯度, 则梯度 G 为

$$G = \sqrt{I_x^2 + I_y^2} \quad (3)$$

为使高梯度产生低能量, 令

$$f_G = \frac{\max(G) - G}{\max(G)} = 1 - \frac{G}{\max(G)} \quad (4)$$

最后, 若 q 是 p 的对角邻点, 则 $f_G(q)$ 不变; 若 q 是 p 的水平或垂直邻点, 则 $f_G(q)$ 除以 $\sqrt{2}$.

1.1.3 $f_D(p, q)$ 的计算

由于梯度方向能量是对变化剧烈边缘的一种平滑, 且当 p, q 两点相似时, 其梯度方向所占的能量较小, 因此这种能量因子的最终影响结果就是使相似点归于同一条能量路径.

$$f_D(p, q) = \frac{1}{\pi} \{ \cos[d_p(p, q)]^{-1} + \cos[d_q(p, q)]^{-1} \} \quad (5)$$

其中

$$d_p(p, q) = D'(p) \cdot L(p, q) \quad (6)$$

$$d_q(p, q) = L(p, q) \cdot D'(q) \quad (7)$$

$$D'(p) = (I_x(p), I_y(p)) \quad (8)$$

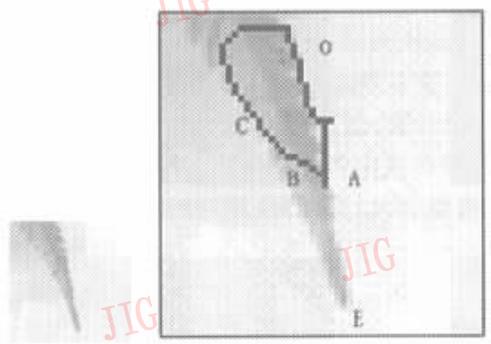
$$D'(q) = (I_y(q), -I_x(q)) \quad (8)$$

$$L(p, q) = \begin{cases} q-p & \text{if } D'(p) \cdot (q-p) \geq 0 \\ p-q & \text{if } D'(p) \cdot (q-p) < 0 \end{cases} \quad (9)$$

从上述能量的计算方法可以看出, $f_z(q)$ 、 $f_G(q)$ 、 $f_D(p, q)$ 分别起到了突出边缘、平滑边缘和归类相似点的作用, 从而使用户能够快速找到边缘点.

由于文献[3]和文献[7]中, 在计算新点的能量时, 只考虑了与当前点的能量和能量差, 也即新点的能量是当前点的能量与新增能量的能量和(这里新

点是当前点的邻点),且这种计算只考虑邻点的影响,并使实际路径距离短的能量值占优,所以在对具有尖角的物体定位时,将造成割角(如图1所示)。



(a) 原图 (b) 搜索后的定位边界放大图
图 1

本文通过跟踪能量计算过程发现,对于C点,由于通过OAEC的能量大于通过OABC的能量,从而导致无法检测到轮廓AEC(图1)。

上述问题可以从改变能量迭代算法和适当设置新的种子点两方面来解决。

(1)采用新的迭代判断准则 对大多数图象,由于经过Marr变换后,边缘点一般处于零交叉点,对于同一点的两条能量路径,若边缘点多的一条不能占优,则显然不符合要求,这时就需要添加新的约束,本文是根据能量路径上零交叉点数的多少来判断取舍,具体就是在计算获得新点的能量后,若当前计算获得的能量大于已有的能量,则反向跟踪路径的最近 n 个点,然后统计两路径的零交叉点数,若当前路径的零交叉点数大于旧路径的零交叉点数,则用当前计算所获得的能量来替换旧的能量,继续进行迭代。

(2)在尖点处自动选取种子点 由于适时地插入新的种子点可以在一定程度上避免因能量迭代误差而引起的边缘定位不准确问题,因此,在尖点处自动插入新种子点则可以改善边缘定位准确性,在交互过程中,需跟踪从种子点到当前点的路径,若发现路径方向发生剧烈的改变时,则在方向改变处设置新的种子点。而判断方向是否改变的一般方法是计算相邻点间的角度,但用这种方法在交互中显然不能满足实时性要求,因此,本文通过相邻点的变号次数(在八邻域中,相邻两点坐标通过 $+1, 0, -1$ 来表示,若两邻点“坐标值”发生变化,即为变号)来判断是否发生方向剧烈改变。另外,由于交互过程中,从边缘点到非边缘点也会发生方向剧烈改变,因此在

实际应用中,从种子点到当前点,若累计发生两次方向剧烈改变,则在第1次发生改变的位置设置种子点。

综上所述,本文的能量迭代最终算法如下:

- (1)初始化,设置种子点和标志位 $Flag=0$;
- (2)是否处理完毕,若是,转步骤(7);否则转步骤(3);
- (3)计算当前点的八邻域点能量,若 $Flag=0$,则把能量值写入能量表,若 $Flag=1$,则转步骤(5);
- (4)比较各邻域点能量,然后选取能量最小者进栈,转步骤(2);
- (5)若新计算的能量小于旧的能量,则把能量值写入能量表,转步骤(4),否则,转步骤(6);
- (6)反向跟踪新、旧能量的路径,并统计最近一段距离内过零点的数量 N_{new}, N_{old} ;若 $N_{new} > N_{old}$,则把能量值写入能量表,转步骤(4),否则,转步骤(4);
- (7)结束。

1.2 平滑处理

本文采用矢量化方法来快速消除毛刺(见图2)。毛刺是由于受噪声的干扰,在跟踪得到边缘时产生的,如图3(b)、(d)、(e)和图4(b)、(d)、(e)所示。

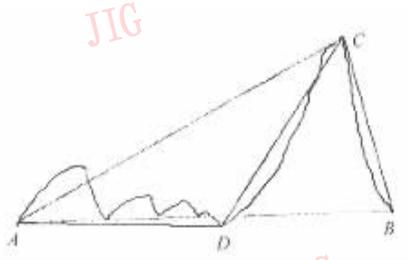


图 2 矢量化方法消除毛刺

对于图2上任意点A与B,首先检查曲线AB之间各点到直线AB距离,并取最大距离点C,连接AC,CB;然后检查曲线AC之间各点到直线AC的距离,并取最大距离点D,连接AD,DC;对曲线CB进行同样处理;当距离达到设定阈值或两点之间小于设定的点数时,则迭代结束。由于上述算法的极限收敛于原曲线,因此可以根据不同的要求来设置误差。这样迭代完成后,再通过对各线段离散化,可得到连续的边缘曲线。

2 实验结果

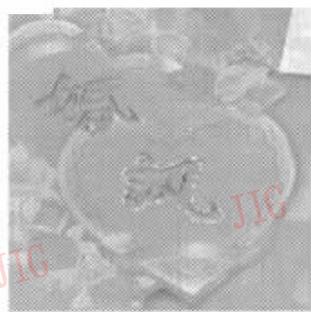
本文用毛笔字以及葵花图片对本文提出的算法进行了实验,实验结果见图3、图4。



(a) 原图



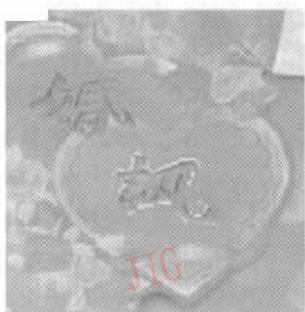
(b) 用文献[7]算法得到的轮廓



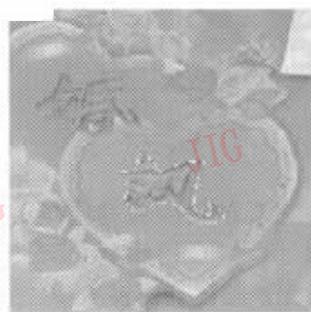
(c) 用文献[7]算法平滑后的轮廓



(d) 用文献[1]算法得到的轮廓



(e) 用本文算法得到的轮廓



(f) 用本文算法平滑后的轮廓

图 3 用不同算法处理毛笔字“讯”字提取出的轮廓



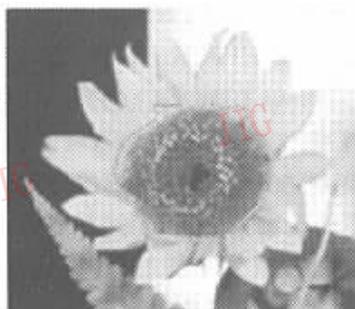
(a) 原图



(b) 用文献[7]算法得到的轮廓



(c) 用文献[7]算法平滑后的轮廓



(d) 用文献[1]算法得到的轮廓



(e) 用本文算法得到的轮廓



(f) 用本文算法平滑后的轮廓

图 4 用不同算法处理葵花图提取出的轮廓

从图 3、图 4 可见,在采用文献[7]的算法时,图 3(b)、(c)中“讯”字和图 4(b)、(c)中花瓣的几个尖角被“割”掉了;采用文献[1]的 Snake 算法,其结

果更差(见图 3(d)和图 4(d)),且无法对复杂物体进行定位,而采用本文算法则比较准确地勾勒出了“讯”字和花瓣的外轮廓(见图 3(e)、(f)和图 4(e)、(f)).

3 结束语

虽然如今 Snake 算法和图搜索算法在目标的匹配定位中已得到了广泛的应用,但尚无法对复杂物体的轮廓进行精确定位,因此本文针对复杂物体,提出了一种有效的轮廓交互提取算法,实验结果表明,算法不仅可以较好地保持目标的尖角特征,而且还能避免以往算法中的“割角”问题,从而提高了轮廓跟踪的准确性。

参考文献

- 1 Williams Donna, Shah Mubarak. A fast algorithm for active contours and curvature estimation. *CVGIP: Image Understanding*, 1992, 55(1): 14~26.
- 2 Kass M, Witkin A, Terzopoulos D. Snake: active contour models. *International Journal of Computer Vision*. 1988, 1(4): 321~331.
- 3 Mortensen E N, Barrett W A. Intelligent scissors for image composition. In: *Proceedings of the ACM SIGGRAPH 95*, Los Angeles, CA, Aug. 1995: 191~198.
- 4 Mortensen E N, Morse B S, Barrett W A *et al.* Adaptive boundary detection using 'Live-Wire' two-dimensional dynamic programming. In: *IEEE Proceedings of Computers in Cardiology*, Durham, NC, USA, 1992: 635~638.
- 5 Chien Y P, Fu K S. A decision function method for boundary detection. *Computer Graphics and Image Processing*, 1974, 3(2): 125~140.
- 6 Martelli A. An application of heuristic search methods to edge and contour detection. *Communications of the ACM*, 1976, 19(2): 73~83.

- 7 Mortensen E N, Reese L J, Barrett W A. Intelligent selection tools. In: *Proc. IEEE: Computer Vision and Pattern Recognition (CVPR'00)*, Vol. II, Hilton Head, SC, June 2000: 776~777.
- 8 Morse B S, Barrett W A, Udupa J K *et al.* Trainable optimal boundary finding using two-dimensional dynamic programming. Technical Report No. MIPG180, Department of Radiology, University of Pennsylvania, Philadelphia, PA, USA, March 1991.

徐晓刚 1967年生,博士,讲师,现为浙江大学CAD&CG国家重点实验室博士后,目前研究方向为图象处理、计算机动画、虚拟现实、系统仿真。

于金辉 1960年生,博士,副教授,研究方向为计算机辅助制作卡通动画、非真实感绘制技术、物体变形技术、计算机生成装饰物体、计算机书法、计算机美术等。

马利庄 1963年生,博士,研究员,博士生导师,国家杰出青年基金,中国青年科技奖获得者,目前研究方向为CAGD、计算机动画。