

部分最优化动态规划轮廓检测算法¹

杨 莉 李玉山

(西安电子科技大学电路 CAD 研究所 西安 710071)

摘 要: 在研究动态轮廓模型原理及其求解算法的基础上提出了一种新的部分最优化动态规划轮廓检测算法, 保留了动态规划算法的性能和优点, 但综合运算复杂度降低, 运算速度大大提高。

关键词: 轮廓检测, 动态轮廓模型, 动态规划, 计算机视觉

中图分类号: TP391 **文献标识码:** A **文章编号:** 1009-5896(2004)06-0923-05

Partial Optimal Dynamic Programming Contour Detection Algorithm

Yang Li Li Yu-shan

(The Institute of Electronic CAD, Xidian Univ., Xi'an 710071, China)

Abstract Basing on studying the principle of active contour model and some solutions to it, a new partial optimal dynamic programming contour detection algorithm is presented and the results comparable to the dynamic programming algorithm are given. Experiment shows such new algorithm is much faster.

Key words Contour detection, Active contour model, Dynamic programming, Computer vision

1 引言

在图像处理和计算机视觉领域, 用复杂轮廓来表示场景中的对象是一个重要的问题。Kass 等人提出的动态轮廓模型^[1], 是在图像域内定义的可变形曲线 $v(s) = (x(s), y(s))$, $s \in [0, 1]$ 为归一化曲线长度, 通过对其能量函数的最小化, 动态轮廓逐步调整自身形状与对象轮廓相一致。最小化能量函数的方法很多, 有变微分法^[1], 贪婪 (greedy) 算法^[2,3], 动态规划算法^[4,5] 等。其中, Amini 提出的动态规划 (Dynamic Programming, DP) 算法采用离散多阶段决策过程对动态轮廓模型能量最小化问题建模, 采用时间延迟离散动态规划算法来求解, 并且可选择适当的约束来控制轮廓的特征, 能够得到最优全局解, 与国内外同类研究成果相比, 算法的可靠性高, 稳定性好。但它存在的一个最主要的问题就是计算复杂, 运算量大, 不适合实际应用。

本文通过研究发现 DP 算法至少存在两个问题: (1) 在 DP 算法迭代执行过程中, 动态轮廓逐步调整自己, 其上的控制点间距也在扩大或缩小, 已不能较好地反映曲线的特征; (2) DP 算法通过迭代使动态轮廓逐步逼近真实轮廓, 每一次迭代都采用时间延迟离散动态规划算法求解, 复杂度为 $O(nm^3)$, (n 为此次迭代动态轮廓上的控制点数, m 为控制点可取的方向数), 而实际上, 在轮廓运动初期, 完全可以使用一种较快、较简单的算法使之尽快逼近真实轮廓, 而在其运动后期, 才需采用 DP 算法调整细节使之与真实轮廓匹配。根据上述想法, 本文提出部分最优化动态规划 (Partial Optimal Dynamic Programming, PODP) 轮廓检测算法, 即首先不

¹ 2003-03-22 收到, 2003-09-05 改回

国家自然科学基金 (No.60172004)、北京大学视觉与听觉信息处理国家重点实验室基金 (No.2001-03)、博士点基金 (20010701003) 资助课题

考虑曲线的平滑性, 采用本文提出的一种比较简单快速的方法, 将初始轮廓尽快地收缩到真实轮廓附近; 然后, 为了保证控制点仍能较好地反映曲线的特征, 对得到的粗略轮廓重新取控制点; 最后, 使用 DP 算法, 调整轮廓细节, 优化轮廓, 使之最终与真实轮廓相匹配。实验证明, 本文所提出的 PODP 算法, 在保留 DP 算法性能和优点的同时, 综合运算复杂度降低, 计算速度大大提高。

2 动态轮廓能量最小化原理及其求解算法

动态轮廓模型参数化表示: $v(s) = (x(s), y(s))$, $s \in [0, 1]$ 为归一化曲线长度。动态轮廓的总能量函数表示为 $E_{\text{total}} = \int_0^1 E(v(s))ds = \int_0^1 [E_{\text{int}}(v(s)) + E_{\text{ext}}(v(s))]ds$, 其中 $E_{\text{int}}(v(s)) = (\alpha(s)|v_s(s)|^2 + \beta(s)|v_{ss}(s)|^2)$ 为动态轮廓的内部能量, $v_s(s)$ 和 $v_{ss}(s)$ 表示曲线的一、二阶导数, 因为平面曲线的一阶导模反映曲线的连续性, 而其二阶导模反映曲线的平滑性, 曲线的各阶导模值越小, 曲线的状态就越稳定。所以, 对平面闭合曲线, 如果使 $E_{\text{int}}(v(s))$ 最小化, 曲线将有平滑地由外向内自发收缩的趋势。 $\alpha(s)$ 和 $\beta(s)$ 控制曲线不同位置处轮廓收缩的状态。外部能量 $E_{\text{ext}}(v(s))$ 则引导动态轮廓向图像的某种特征移动, 这里定义 $E_{\text{ext}}(v(s)) = -r|\nabla f(v)|^2$ 为梯度模的负值, $E_{\text{ext}}(v(s))$ 的最小化将动态轮廓吸引到图像强边缘处。所以, 使定义的动态轮廓模型的能量最小化, 就可以将初始轮廓平滑地吸引到目标轮廓处, 从而匹配目标轮廓。

从 Kass 等提出动态轮廓能量最小化概念至今, 出现了许多求解算法, 如梯度下降法^[1]; DP 算法^[4,5]; 贪婪算法^[2,3]及其它一些算法^[6-8]。经研究发现, DP 算法虽然计算复杂, 运算量大, 但它可靠性高, 稳定性好, 而且能量随时间单调递减, 动态轮廓在有限次迭代后总能收敛, 可以获得最优化解。DP 算法概述如下, 离散化能量函数:

$$E_{\text{total}} = \sum_{i=1}^n (E_{\text{int}}(v_i) + E_{\text{ext}}(v_i)), v_i \text{ 为轮廓上的控制点, 共取 } n \text{ 个} \quad (1)$$

其中 $E_{\text{int}}(v_i) = (\alpha_i|v_i - v_{i-1}|^2 + \beta_i|v_{i+1} - 2v_i + v_{i-1}|^2)/2$, $E_{\text{ext}}(v_i) = -r|\nabla f(v_i)|^2$ 。每个控制点允许有 m 个可能值。一种找到式 (1) 最小值的方法为穷举法, 复杂度为 $O(m^n)$, 而另一种更有效的求解方法就是借助离散动态规划, 将式 (1) 的最小化看作一个离散多阶段决策过程。即从轮廓上某一控制点出发, 将式 (1) 的最小化看作有限阶段 (i_1, i_2, \dots, i_n) 中每个阶段的最小化, 再从一组有限的可能策略中作出决策。最小化问题可以写成:

$$E_{\text{total}}(v_1, v_2, \dots, v_n) = E_1(v_1, v_2, v_3) + E_2(v_2, v_3, v_4) + \dots + E_{n-2}(v_{n-2}, v_{n-1}, v_n) \quad (2)$$

其中 $E_{i-1}(v_{i-1}, v_i, v_{i+1}) = E_{\text{ext}}(v_i) + E_{\text{int}}(v_{i-1}, v_i, v_{i+1})$ 。为了求得式 (2) 的最小化能量, 生成具有两个变量的优化函数序列, 即能量矩阵:

$$S_i(v_{i+1}, v_i) = \min_{v_{i-1}} \{S_{i-1}(v_i, v_{i-1}) + \alpha|v_i - v_{i-1}|^2 + \beta|v_{i+1} - 2v_i + v_{i-1}|^2 + E_{\text{ext}}(v_i)\} \quad (3)$$

计算完所有控制点得到当前轮廓所能达到的最小能量 $E_{\text{min}} = \min_{v_{n-1}} S_{n-1}(v_n, v_{n-1})$, 完成一次迭代。在计算过程中, 还存在一个位置矩阵, 存放最小化式 (3) 所得到的控制点移动的位置。一次迭代结束后, 逆向跟踪位置矩阵, 可得到一次迭代后的优化轮廓, 若有 n 个点, 每个点可取 m 个方向, 单次迭代的复杂度为 $O(nm^3)$, 为了保证得到最优轮廓, 多次迭代直到总能量不能减少为止。

3 部分最优化动态规划轮廓检测算法

通过研究 DP 算法发现, 计算时, 通常首先将初始轮廓放在需要检测的轮廓附近, 然后采用离散多阶段决策过程使动态轮廓由外向内收缩。在每次迭代的第 i 阶段, 曲线一阶导模有两个状态

变量 v_{i-1}, v_i 单次迭代计算复杂度为 $O(nm^2)$ ，而曲线的二阶导模有 3 个状态变量 v_{i-1}, v_i, v_{i+1} ，单次迭代计算复杂度 $O(nm^3)$ ，梯度模单次迭代计算复杂度只有 $O(nm)$ ，所以综合来说，能量最小化式 (2) 单次迭代的复杂度为 $O(nm^3)$ 。如果初始轮廓离开真实轮廓稍远，迭代次数增加，整个算法的运行时间就会成倍增长，而实际上在轮廓距真实轮廓比较远的时候，只需要强调将它迅速地移向真实轮廓，而不必注重轮廓的平滑，只在轮廓比较接近真实轮廓的时候，才需要采用 DP 算法，以将它平滑、紧密地调整与真实轮廓相一致。

基于上述思想，本文提出了新的 PODP 轮廓检测算法，将式 (2) 的能量最小化过程分两个阶段进行。第一阶段，将初始轮廓尽可能迅速地移向真实轮廓，而不必过于注重轮廓的平滑。DP 算法对动态轮廓的能量最小化过程采用离散多阶段决策过程建模，可以获得全局最优解，但如前所述，移动一次轮廓的计算复杂度为 $O(nm^3)$ ，为实际应用带来一定的问题。为了提高速度，先不考虑全局最优解，对每个控制点，只考虑它的局部能量最小值，即对图像初始轮廓中的 (v_1, v_2, \dots, v_n) 等 n 个控制点，对每个控制点 v_i ，采用式：

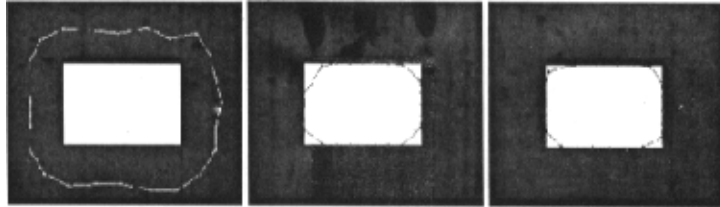
$$E(v_i) = \min_{k=0; m-1} \{[\alpha_i |v_i + k - v_{i-1}|^2 + \beta_i |v_{i+1} - 2(v_i + k) + v_{i-1}|^2] / 2 - r |\nabla f(v_i + k)|^2\} \quad (4)$$

v_{i-1}, v_{i+1} 固定，采用当前轮廓上的值，只在当前变量 v_i 的 m 个可能值中考虑式 (4) v_i 的局部最小能量，若 k 不等于 0，则将 v_i 移到新的位置，逐点计算，依次类推，处理完整个轮廓上的点，就得到一条新的轮廓，完成一次迭代，计算复杂度 $O(n)$ 。这样处理轮廓上的控制点，虽然轮廓也会向内收缩，但没有考虑控制点之间的联系，得到的曲线不具有平滑性，但计算复杂度降低了 $O(m^3)$ ，计算速度大大提高。对得到的新轮廓继续进行上述操作，直到没有轮廓点移动为止。这时得到的轮廓已经比较接近真实轮廓，但曲线不平滑，而且控制点有的粘连在一起，有的相距比较大，既不能反映曲线的特征，对后续计算也有一定的影响，所以这时对得到的新的轮廓重新取控制点，并进入能量最小化过程的第二阶段，采用 DP 算法，调整轮廓细节，优化轮廓，使之最终得到的轮廓与真实轮廓相匹配。

图 1，图 2，图 3 为采用 PODP 算法和 DP 算法对 rectangle, bottle, car 等 3 幅图进行轮廓检测相比较的结果。表 1 为检测过程中所得到的比较数据。初始轮廓与参数已预先确定。rectangle 图像大小为 128×128 像素，中心矩形像素值为 255，背景像素值为零，初始轮廓与 DP, PODP 算法轮廓检测结果如图 1 所示。可见，由于在第一阶段结束时，PODP 算法重新取控制点，在一定程度上提高了轮廓检测的性能。DP 算法选择控制点 23 个，迭代 31 次获得最小能量 -26384，耗时 357s (程序采用 matlab6.0 实现，程序没有优化，耗时仅供比较)。而采用 PODP 算法在对 rectangle 图初始轮廓快速收缩后，重新对所得轮廓取控制点 23 个，由于已经比较接近真实轮廓，到达最小能量的迭代次数仅 4 次，最小能量 -25211，总耗时仅 76s，比 DP 算法运算速度大大提高。图 2，图 3 为拍摄的真实场景，可见，由于采用了预先快速收缩，对同一幅图，采用时间延迟离散动态规划算法导致单次迭代复杂度为 $O(nm^3)$ 的迭代次数减少，使得轮廓检测的总运行时间缩短一半甚至更多，初始轮廓离真实轮廓越远，这种优势的体现越明显。而且在轮廓接近真实轮廓时对轮廓又重新取控制点，检测性能也略有增强。

表 1 PODP 算法与 DP 算法的结果比较

		控制点数	迭代次数	最小能量	耗时 (s)
rectangle	DP 算法	23	31	-26384	357
	PODP 算法	23	4	-25211	76
bottle	DP 算法	33	20	-14362	333
	PODP 算法	33	10	-14510	158
car	DP 算法	30	21	-18443	310
	PODP 算法	30	10	-15311	150



(a) 初始轮廓 (b) DP 算法检测的轮廓 (c) PODP 算法检测结果

图 1 rectangle 轮廓检测比较



(a) 初始轮廓 (b) DP 算法检测的轮廓 (c) PODP 算法检测结果

图 2 bottle 轮廓检测比较



(a) 初始轮廓 (b) DP 算法检测的轮廓 (c) PODP 算法检测结果

图 3 car 轮廓检测比较

4 结论

本文在研究动态轮廓原理及其求解算法的基础上提出了一种新的部分最优化动态规划轮廓检测算法 (PODP), 针对 DP 算法存在的运算复杂, 计算量大等缺点, 找出一种比较简单快速的方法, 将初始轮廓尽快地收缩到真实轮廓附近, 再对重新取控制点的粗略轮廓使用 DP 算法, 调整轮廓细节, 优化轮廓, 使之最终与真实轮廓相匹配。经 matlab6.0 仿真证明, 本文所提出的 PODP 算法, 比起 DP 算法, 在保留原有性能的前提下, 运算速度得到大大提高, 为实时应用打下基础。

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] Williams D J, Shah M. A fast algorithm for active contours. In Proc. Third Int. Conf. Computer Vision, Osaka Japan, Dec. 4-7, 1990: 592-595.
- [3] Williams D J, Shah M. A fast algorithm for active contours and curvature estimation. *CVGIP: Image Understanding*, 1991, 55(1): 14-26.
- [4] Amini A, Tehrani S, Whemouth T. Using dynamic programming for minimizing the energy of active contours in the presence of hard constraints. In Proc. Second Int. Conf. Computer Vision. Tarpon Springs, FL, Dec. 5-8, 1988: 95-99.
- [5] Amini A, Whemouth T, Jain R. Using dynamic programming for solving variational problems in vision. *IEEE Trans. on Pattern Anal. Machine Intell.*, 1990, PAMI-12(9): 855-867.
- [6] Mokhtarian F, Mohanna F. Fast active contour convergence through adaptive curvature scale space smoothing. In Proc. Int. Conf Computer Vision, Pattern Recognition and Image Processing. Durham, North Carolina, 2002: 724-727.
- [7] Park H, Schoepflin T, Kim Y. Active contour model with gradient directional information: directional snake. *IEEE Trans. on Circuits and Systems for Video Technology*, 2001, 11(2): 252-256.
- [8] Kang D J. A fast and stable snake algorithm for medical images. *Pattern Recognition Letters*, 1999, 1(1): 507-512.

杨 莉: 女, 1976 年生, 博士, 主要从事计算机低层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, 主要从事电子设计自动化、机器视觉等方面的研究.