利用 Hough变换实现直线的快速精确检测

滕今朝[1] 邱 杰[2]

[1] (威海职业学院机电工程系, 威海 264210) [2] (海军航空工程学院, 烟台 264000)

摘 要 利用 Hough变换对直线进行检测, 通常存在“速度缓慢、结果不够精确”的问题, 本文提出了“分式查表法”, 能在大幅度减少 Hough变换的总计算量的情况下, 检测精度保持最高, 从而使超大型图像中, 直线的实时、精确检测成为可能。

关键词 Hough变换 参数空间 精度 分式查表法

中图法分类号: TP391.41 **文献标识码**: A **文章编号**: 1006-8961(2008)02-0234-04

Fast and Precise Detection of Straight Line with Hough Transform

TENG Jin-zhao[1], QIU Jie[2]

[1] (*Electromechanical Engineering Department, Weihai Vocational College, Weihai* 264210)

[2] (*College of Navy Aviation Engineering, Yantai* 264000)

Abstract Problems as low speed or inaccurate results in the process of line detection with Hough Transform remain unsatisfactorily solved. This paper puts forward a new look-up table to decrease the computation distinctly and keeping the highest precision. It offers the possibility in real-time applications especially in large image.

Keywords Hough Transform, parameter space, precision, table look-up

1 引 言

Hough变换具有优异的鲁棒性和极佳的抗干扰能力, 利用 Hough变换进行直线检测, 是图像分析和计算机视觉的一个重要内容。但是 Hough变换的计算量往往非常大, 从而阻碍了其在快速、精确检测直线方面的应用。

本文提出的新方法, 不仅能大幅度减少 Hough变换的总计算量, 而且在像素允许的情况下, 直线斜率的检测精度保持最高, 这对于超大型图像中直线的实时、精确检测, 具有重要的实用价值。

2 Hough变换检测直线的原理

选取图像空间中一条直线 L 的某些特征, 作为参数空间的一个点 M, 并且该直线 L 上所有点, 通过某种算法, 都能够对应着这些特征, 从而在图像空间和参数空间之间, 建立起“线—点”的对偶性。Hough变换就是根据这种对偶性, 将图像空间中直线的检测问题, 转化为参数空间中点的检测问题, 而后者的处理比前者要简单得多, 进行累加统计即可。

2.1 用极坐标建立参数空间

常用的 Hough变换检测直线的方法, 是运用下式在图像空间和参数空间之间, 建立对偶变换。

$$\rho = x\cos\alpha + y\sin\alpha$$

式中, ρ为极径; α为极角, α取 0~180°; x 为像素点相对图像原点的行坐标; y 为像素点相对图像原点的列坐标。

为了检测出直角坐标系中, 由非零点所构成的直线, 需要根据检测分辨率的要求, 将 α离散化为 N_α 个参数区间, 将 ρ离散化为 N_ρ 个参数区间, 也就是说将极坐标系量化成许多小格, 建立参数空间。

这种方法被称为标准 Hough变换方法 (standard

收稿日期: 2006-04-25; 改回日期: 2006-10-17

第一作者简介: 滕今朝 (1970~), 男, 讲师。2007年于海军航空工程学院获测试计量仪器与科学专业硕士学位。主要从事电气自动化、检测技术方面的教学和研究。E-mail: twrite@163.com

hough transform, SHT)。其优点是: 无论直线怎样变化, 参数空间中 α和 ρ的取值范围是有限的。所以, 目前的直线检测大多数都是基于这种方法。

但是, 这种方法在 N_α 值较大的情况下, 存在以下两个缺陷:

(1) 计算量大

N_α 越大, α的步长越小, 计算量就越大[1]。在要求检测精度很高的场合, N_α 的值往往非常大, 这样会使计算量大增。

(2) 需要大的存储空间

如果 α和 ρ都占 4个字节, 参数空间所需要的存储空间的字节数 S 可由下式求出:

$$S = 4N_\alpha N_\beta \qquad (1)$$

式中, N_α 为 α在 [0, π)间取的离散值的个数; N_ρ 为 ρ的采样个数。

对较大的图像, S 将大于数千兆字节 (GB), 单靠物理内存, 难以满足这样的要求。

为了减少这种 Hough变换的计算量, 减小所需要的存储空间, 在此基础上出现了很多改进的 Hough变换算法, 例如分块检测法[1]、两次检测法[2]、全整数 Hough变换[3]等, 为讨论方便, 将其统称为正弦 Hough变换方法。但这些方法由于受 α的步长限制, 有时候检测结果不尽人意, 难以实现对任意斜率直线的快速、精确检测, 尤其是对超大型图像 (例如像素在 2 048 × 2 048以上)。

如果能够建立一个不受 α步长限制的参数空间, 就有可能实现对任意斜率直线的快速、精确检测。

2.2 用直线的斜率 k 和截距 b 建立参数空间

图像空间中, 直线 L_0: $y = k_0x + b_0$ 上每一个点, 在参数空间中都代表一条直线, 这些直线都相交于一点 $M(k_0, b_0)$。

与用极坐标建立参数空间的方法相比, 这种方法不受 α的步长限制, 检测了所有的可能出现的直线, 不会有任何遗漏, 在像素允许的情况下, 能精确地检测出图像中的任意直线。这种优点决定了它非常适合用来对直线进行精确检测。

为便于讨论, 将其称为 kb Hough变换方法。

虽然这种方法在原理上非常明确, 但在具体实现过程中, 如果简单地用浮点数进行斜率和截距的计算, 还有以下 3个问题需要解决:

(1) 如果同时计算斜率和截距, 参数空间结构数组可能异常庞大, 而且计算量非常大

1幅像素为 $m \times n$ 的图像, 假设图像中可能出现的不重复的斜率有 u 种 (u 大于图像的像素总数)。若这些数据都占 4个字节, 那么要求的计算机内存为

$$w_1 = 8umn \qquad (2)$$

对较大的图像, 要求的计算机内存超过上千兆字节 (GB), 这显然是不现实的。

如何解决这个问题?

很多时候, 如果将一个 2维问题分解为两个 1维问题来解决, 往往非常方便。

参数空间数组之所以异常庞大, 是因为同时包含了待测的斜率和截距, 而斜率和截距的组合, 决定了其数组的元素个数必然非常多。

如果将斜率 k 和截距 b 分步进行检测, 先检测斜率, 找出出现次数最多的斜率 k_0, 然后再检测截距, 找出斜率为 k_0 的, 出现次数最多的直线的截距 b_0, 也就是说, 将一个 2维问题分解为两个 1维问题。由于斜率为 k_0 的二点对的数量往往非常有限, 这样就能有效地减小参数空间。

但是, 即使将斜率 k 和截距 b 分步进行检测, 如果用浮点数进行计算斜率, 并采用普通的 Hough变换的投票方式, 即先计算完由非零点组成的所有的二点对的斜率, 再对斜率进行统计, 也可能需要很大的参数空间。1幅 800 × 600的图像, 如果非零点非常多, 需要的参数空间仍然可能超过数千兆字节 (GB)。

所以, 用浮点数计算斜率, 要减小参数空间, 只能一边计算斜率, 一边统计斜率相同的情况 (与前面的斜率进行比较)。即边投票, 边计票。但是, 这样计算量非常大。

(2) 程序的复杂程度高

根据定量度量程序的复杂程度的 McCabe方法, 流图 (也称为程序图)的环形复杂度可由下式求出:

$$V(G) = R + 1 \qquad (3)$$

式中, $V(G)$ 为环形复杂度; R 为流图中的判定节点数目。通常 $V(G) \leqslant 10$ 为宜。

根据 McCabe方法, 求出该方法的流图的判定节点数 R 为 10, 由式 (3)可知, 其环形复杂度为 11, 显然复杂程度太高。

(3) 如果直接采用浮点数计算斜率, 无法处理斜率为无穷大的垂直直线

斜率为无穷大, 给计算带来不便, 这正是极少有人采用这种方法的原因。

综上所述, 除了需要分步检测斜率和截距外, 还不能简单地采用浮点数计算斜率。因此采用了斜率分式查表方法来解决这些问题。

3 斜率分式查表法

斜率分式查表法具体方案如下:

将斜率用分式表示, 分子、分母都是整数。但是有时候, 由不同的整数表示的斜率可能是相同的, 例如“3/9”和“4/12”实际是相同的, 所以要进行化简等处理。

事先要建立表格, 可以在初始化中进行: 以一幅 $m \times n$ 的图像为例 (假设 $m > n$)。

(1) 将 1~ 800的整数分解, 存入数组 ***bd*** 中。例如 14被分解为“1×14”、“2×7”, 将 10分解为“1×10”, “2×5”。

(2) 将 1~ 800的整数两两相除, 利用第 1步的结果进行化简, 化简的结果存入结构数组 ***reduct***, 以建立表格。

例如“14/10”, 利用第 1步的结果, 化简为“7/5”, 记为 *reduct*[14]. *sr*[10][0] = 7, *reduct*[14]. *sr*[10][1] = 5, *sr* 为结构数组 ***reduct*** 内的数据成员。将来在检测斜率的时候, 用数组 ***reduct*** 对斜率进行化简: 如果遇到斜率为“14/10”, 通过查表 *reduct*[14]. *sr*[10]立刻知道, 该斜率的化简结果为“7/5”。

建立好表格之后, 在检测斜率时, 用结构数组 ***reduct*** 来查表、化简, 根据结果, 将参数空间的结构数组 ***count*** 中的相应计数器的值加 1, 即结构数组 ***count*** 保存该斜率出现的次数。

对 1幅像素为 $m \times n$ 的图像, 由于结构数组 ***reduct*** 存储化简之后有分子、分母, 所以设定其元素个数为 $2mn$。考虑到斜率有正负, 设定 ***count*** 数组元素个数为图像的像素数的 2倍, 即 $2mn$。若忽略结构数组 ***bd*** 占用的存储空间 (对于一幅 800×600的图像, 只有 0.256MB), 则求斜率需要的内存字节数为

$$w_k = 16mn$$

求出斜率之后, 再求截距。假设斜率为 k_0 的直线有 mn 条 (实际上不可能有这么多)。考虑到截距有正负, 则设定截距数组元素个数为 $2mn$。求截距需要的内存字节数为

$$w_b = 8mn$$

至此, 参数空间总共需要内存字节数为

$$w_2 = w_k + w_b = 24mn \qquad (4)$$

这个数值与 *kb* Hough变换方法需要的内存相比如何呢?

为讨论方便, 假设 $m = n$。将式 (4)与式 (2)相比:

$$w_2 / w_1 = 3/u$$

u 大于图像的像素总数 n^2, 即 $u > n^2$, 故

$$w_2 / w_1 < 3/(n^2)$$

斜率分式查表法需要的内存, 不到 *kb* Hough变换方法的 $3/(n^2)$。可见, 图像越大, 斜率分式查表法的优势越明显。

与标准 Hough变换方法需要的内存相比, 情况又如何呢?

将式 (4)与式 (1)相比:

$$w_2 / S = 6n^2 / (N_\alpha N_\rho)$$

而 $N_\alpha = u > n^2$, $N_\rho > 2n$, 故

$$w_2 / S < 3/n$$

若 $n = 1\,000$, 斜率分式查表法所需的内存, 不足标准 Hough变换方法的 3/1 000, 优势相当明显。以像素为 800×600的图像为例, 上述参数空间总共需要的内存小于 12MB。

同时, 还解决了前面 *kb* Hough变换方法的以下两个难题:

(1) 关于减小计算量, 降低程序的复杂程度。

由上述分析可知, 斜率分式查表法只是在初始化时建立表格, 而在此后的计算过程中, 完全不需要将每一个二点对的斜率, 与参数空间中的各个斜率逐一比较, 极大地减小了计算量。

根据 McCabe方法, 可以求出这种方法的流图的判定节点数 R 为 6, 由式 (3)可知, 其环形复杂度为 7, 可见采用斜率分式查表法与 *kb* Hough变换方法相比, 大大降低了程序的复杂程度。

(2) 关于斜率可能为无穷大的问题。

对于垂直状态的直线斜率为无穷大的情况, 只需要在参数空间结构数组中, 特设一个数组元素, 例如 *count*[1]. *sc*[0]作为计数器, 专门对无穷大的斜率进行计数即可。

至此, 斜率分式查表法解决了用直线的斜率 k 和截距 b 建立参数空间所遇到的难题, 为快速、精确地检测出图像中的任意直线, 提供了有力工具。

4 测试与验证

运用斜率分式查表方法, 能快速、精确地检测出

图像中的任意直线。这一点已经在个人计算机上经过充分的验证, 并与标准 Hough变换方法进行了比较。所用的软硬件环境如下:

硬件平台: CPU 为 Intel Celeron(TM), 主频 433MHz, 内存是 128MB。

软件平台: 操作系统为中文 Windows2000, 算法程序语言是 C++, 编译器用 Visual C++ 6.0。

表 1列出了一组各种尺寸的含有椒盐噪声图像的直线测试结果。

表 1　测试结果

Tab. 1　Test result

图像尺寸 (pixel)	分辨率要求 (°)	标准 Hough 变换方法所用时间 (s)	斜率分式查表法所用时间 (s)
128×128	0.009	6.01	0.31
256×256	0.002 27	360.35	0.73
800×600	0.000 3	4 962.28	1.32
1 024×1 024	0.000 14		2.15
2 048×2 048	0.000 035 3		3.73

测试表明, 当检测精度要求较低时 (α步长大于 0.01°), 标准 Hough变换方法能够较为迅速地对目标进行检测; 当检测精度要求较高时 (α步长小于 0.01°), 标准 Hough变换方法的检测速度非常慢。

以 1幅 800×600的图像为例, 要精确地检测像素为 800×600的图像中的每一条直线, 分辨率不能低于 0.000 3°。用极坐标为参数空间的标准 Hough变换方法, 耗时 4 962.28s, 而运用斜率分式查表方法, 耗时 1.32s, 后者比前者要快上千倍。而对于在标准 Hough变换方法基础上改进的各种正弦 Hough变换方法来说, 速度能提高 100倍的, 已经算是非常快的了。

对于尺寸在 1 024×1 024以上的大型图像, 分辨率不能低于 0.000 14°, 用极坐标为参数空间的标准 Hough变换方法, 运行数小时, 仍然没有结果, 已失去了实用意义; 运用斜率分式查表方法, 仍然较为迅速, 而且随着图像尺寸的增大, 所用时间增加得并不明显。

5　结　论

通过进一步优化程序, 提高计算机配置, 斜率分式查表的 Hough变换方法, 能快速、准确地检测出目标直线, 而且图片越大, 这种查表方法的优越性就越明显, 从而使超大型图像中直线的实时、精确检测成为可能。

参考文献 (References)

1　Qiu Liwei, Song Zishan, Shen Weiqun. Kind of fast Hough transform used in line detection[J]. Journal of Beijing University of Aeronautics and Astronautics, 2003, 8(29): 741~744. [邱力为, 宋子善, 沈为群. 直线参数检测的快速 Hough变换[J]. 北京航空航天大学学报, 2003, 8(29): 741~744.]

2　Zhang Yong-zhong, Zhu Ying. Application of Hough transform in edge detection[J]. China Science and Technology Information, 2006, 13: 47~48. [张永忠, 朱英. Hough变换在物体边缘检测上的应用[J]. 中国科技信息, 2006, 13: 47~48.]

3　Ohna G, Magli E. All-integer Hough transform: performance evaluation[A]. In: Proceedings of International Conference on Image Processing[C], Thessaloniki, Greece, 2001: 338~341.