第 4 卷(A 版) 第 10 期
1999 年 10 月

中国图象图形学报
Journal of Image and Graphics

Vol. 4(A), No. 10
Oct. 1999

矩特征的一种快速算法

董 武 李树祥
(第一军医大学生物医学工程系，广州 510515)

摘 要 介绍了一种基于边界点计算形状矩特征的算法，并采用该算法计算了目标的形状特征。结果表明该算法比传统的矩特征计算方法具有高的运算速度。

关键词 矩 不变矩 特征提取

0 引 言

在模式识别领域中，图象的形状特征是特征提取的重要对象。矩特征是其中广泛使用的形状特征之一，一些最基本的二维形状特征都与矩有直接的关系。图形的面积由其第(0,0)阶矩表示。而重心、关于长轴及短轴的惯性矩和一些十分有用的矩不变量都可直接由矩得到。二维连续函数 $f(x,y)$ 的第 (p,q) 阶矩定义如下：

$$m_{pq}=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}x^p y^q f(x,y)\mathrm{d}x\mathrm{d}y \qquad (1)$$

类似地，数字图象的第 (p,q) 阶矩可定义如下：

$$m_{pq}=\sum_{i=1}^{M}\sum_{j=1}^{M}i^p j^q g(i,j) \qquad (2)$$

其中 $g(i,j)$ 是图象的空间矩阵。矩通常都是根据式(2)由空间矩阵计算得到。对于分割后的二值图象，$g(i,j)$ 通常取 0 和 1。

由于利用图形的点阵计算矩特征的算法运算量大、耗时。本文介绍一种新的计算矩的方法，该法由图形的边界点而不是由空间点阵来计算矩。由于传统计算矩的方法的复杂度正比于图形中象素的个数，该算法的复杂度正比于图形边界点的数目。而边界点的数目近似正比于图形中象素总数的平方根，故该算法具有较高的计算速度。并且对象尺寸越大，这种效率优势越明显。

1 基于边界点计算矩的方法

在不失一般性的前提下，将所需计算的图形置于第一象限，将每一边界点与座标原点用线连接，如图 1 所示。相邻的两边界点与原点就形成了一个三角形。具有 n 个边界点 $(x_1,y_1),(x_2,y_2),\cdots,(x_n,y_n)$ 的图形有 n 个这样的三角形，即 $T_1,T_2,\cdots,T_n$。由于积分式(1)的运算是线性运算，一个完整图形的矩就可由这些三角形的矩求得。

假设 T 是具有角 $(a,b),(c,d)$ 和 $(0,0)$ 的三角形。用线段将点 (a,b) 与 $(a,0)$、(c,d) 与 $(c,0)$ 连接起来就形成了三个区域。假设 $a<c$，由图 2 所示区域 1

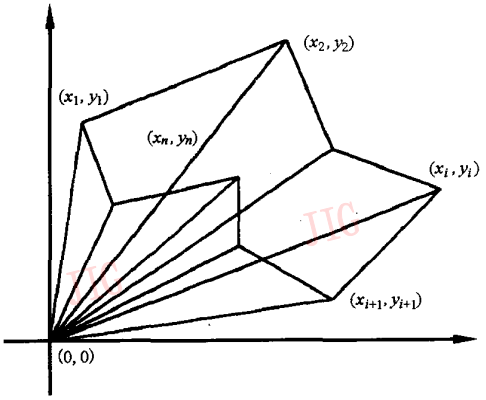


图 1 连接边界点和坐标系原点形成三角形

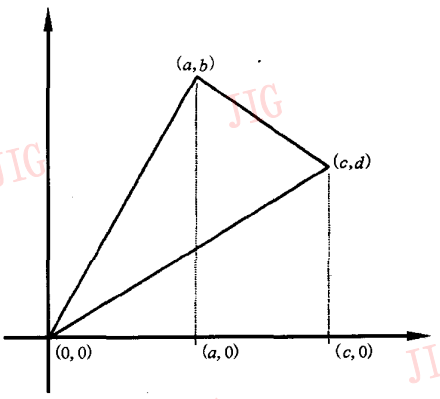


图 2 由三角形的 3 个角定义 3 个区域

收稿日期：1998-08-19；收到修改稿日期：1998-11-03

就是由角点(0,0),(a,b)及(a,0)构成的三角形,区域 2 是由角点(a,0),(a,b),(c,d)及(c,0)构成的梯形,区域 3 是由角点(0,0),(c,d)及(c,0)构成的三角形。我们感兴趣的三角形 T 可由区域 1 加区域 2 减去区域 3 而得到。由于式(1)的积分运算是线性运算,三角形 T 的(p,q)阶矩可由区域 1 的(p,q)阶矩加区域 2 的(p,q)阶矩减去区域 3 的(p,q)阶矩而得到。而这 3 个区域的(p,q)阶矩可分别由积分得到。下面给出这 3 个区域的 6 个低阶矩。对区域 1 这些矩是

$$m_{00,1}=ab/2,\quad m_{01,1}=ab^2/6,\quad m_{10,1}=a^2b/3,$$
$$m_{11,1}=a^2b^2/8,\ m_{02,1}=ab^3/12,\ m_{20,1}=a^3b/4。\tag{3}$$

对区域 2 这 6 个低阶矩是

$$m_{00,2}=(c-a)(d+b)/2,$$
$$m_{01,2}=(c-a)(d^2+db+b^2)/6,$$
$$m_{10,2}=(d-b)(c^2+ca+a^2)/3+(bc-ad)(c+a)/2,$$
$$m_{11,2}=\begin{cases}[3(c-a)^2(d^3+d^2b+db^2+b^3)-4(bc-ad)(c-a)(d^2+db+b^2)]/24(d-b)\\ b^2(c^2-a^2)/4\end{cases}$$
$$m_{02,2}=(c-a)(d^3+d^2b+db^2+b^3)/12,$$
$$m_{20,2}=(d-b)(c^3+c^2a+ca^2+a^3)/4+(bc-ad)(c^2+ca+a^2)/3。$$

由于区域 3 与区域 1 一样是直角三角形,它的 6 个低阶矩通过将公式(3)中的 a 替换成 c,b 替换成 d 而得到。

三角形 $\dot{T}$ 的(p,q)阶矩可由下式得

$$m_{pq,T}=m_{pq,1}+m_{pq,2}-m_{pq,3}$$

对每个三角形计算得到的矩需要确定正负。三角形极性的确定采取如下方法。对于由点(x_i,y_i),(x_{i+1},y_{i+1})和(0,0)组成的三角形 T_i,当以顺时针方向沿边界由(x_i,y_i)向(x_{i+1},y_{i+1})的方向跟踪时,点(0,0)位于该边界段的右边时给三角形 T_i 赋“+”号。相反,如果点(0,0)位于左边时赋“−”号。

对于点(0,0)是位于边界段的左边还是右边比较容易确定。如图 3 所示,如果由 X 轴与(0,0)至(x_i,y_i)连线的夹角大于由 X 轴与(0,0)至(x_{i+1},y_{i+1})连线的夹角,则(0,0) 位于边界段的右边。反之,如果第一个角度小于第二个角度,则该点位于左边。如果这两个角度相等,则正负无关紧要。因为这时(0,0)点位于(x_i,y_i)与(x_{i+1},y_{i+1})连线的沿长线上,这时三角形的面积为 0。相应的各阶矩也就为 0。因此符号公式如下:

$$sign(i)=\begin{cases}+1, & \tan^{-1}(y_i/x_i)>\tan^{-1}(y_{i+1}/x_{i+1})\\ -1, & 其它\end{cases}\tag{4}$$

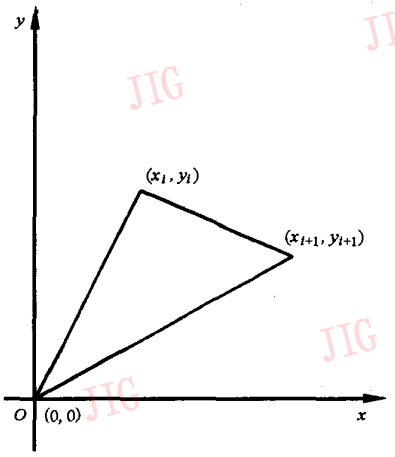


图 3　原点在边界线右边时

给三角形赋予符号的方法是基于对边界点的跟踪是沿顺时针方向。内部在右边而外部在左边。

下面讨论如何由三角形的矩得到图形的矩。图形的第(p,q)阶矩可以由下面的公式计算:

$$m_{pq,S}=\sum_{i=1}^{n}m_{pq,T_i}\cdot sign(i)\tag{5}$$

其中 $sign(i)$是赋予三角形 T_i 的符号,其表达式见公式(4)。公式(5)表明按照三角形的极性,通过加减三角形的各阶矩可以得到图形的矩。这是因为图形本身可以由三角形相加得到。下面对公式(5)加以证明。

假设图形 S 是一个有 N 个边界点的数字图形,其中 N 是有限整数。换言之,图形边界有 N 个直线边界段。仍将图形置于 x-y 平面的第一象限。也就是将x-y 平面的原点 O 置于图形外面的左下角位置。

如图 4 所示。假设 P 是图形内的一点,用线 L 将原点 O 及 P 点连接起来并延伸至图形外的一点 E。可以进一步假定线 L 不完全包含 N 条边界线中的任何一条。因为 P 是内部点,O 是外部点,线 L 在 O 点与 P 点之间与图形的边界相交 K 次,且 K 为奇数。同样,当将线 L 从 P 点延长至外部一点 E 时,它与图形边界相交 M 次,M 亦为奇数。

当从 P 点向 E 点移动时,第一次相交使得移动点处于图形外,第二次相交使得移动点又回到图形内部,如此往复。当然,最后一次相交使得移动点处于图形外。这样,在从 P 点到 E 点间线 L 与边界的 M 次交点中有$(M+1)/2$ 次移动到图形外,有$(M-1)/2$ 次移回至图形内。如图 1 所示每一边界段的两个终点与原点一起构成一个三角形。实际上从 P 点

移动到 E 点与边界有 M 个交点意谓着有 M 个包含 P 点的三角形。对于从里移动到外时所经过的边界线,其相应的三角形所对应的符号为“+”。也就是说有$(M+1)/2$ 个三角形的符号为正。这样,剩下的$(M-1)/2$ 个三角形的符号为负。因此,按照公式(5),内部点 P 对计算图形的矩只做了一次贡献,或者说每一个内部点的矩只被计算了一次。同样可以证明当 P 是外部点时,K 和 M 都是偶数。在 M 个包含 P 点的三角形中,一半取正号,另一半取负号。这样,按照公式(5),外部点 P 对计算图形的矩没有任何贡献。

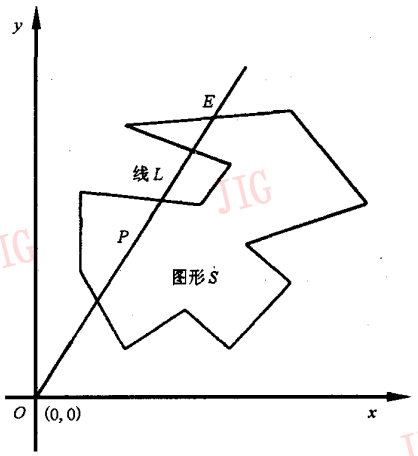


图 4 线 L 通过一内部点 P

在前面已假定线 L 与任何边界线不完全重迭。因为只有有限数目(N)的边界线,因此最多只排除了 N 条有限长线。由微积分学可知,位于图形中的有限条有限长线段的面积积分为 0。因此证明了由公式(5)可以准确计算数字图形的矩。

2 由矩计算其它形状特征

当得到了一个图形的低阶矩,就可由它们推出许多重要的形状特征。例如(0,0)阶矩给出了图形的大小。以下是一些由低阶矩推导的图形的形状特征。

中心:图形 S 的重心可由下式计算:

$$c_x=m_{10,S}/m_{00,S},\quad c_y=m_{01,S}/m_{00,S}$$

中心矩:图形 S 的第(p,q)中心矩定义如下

$$\mu_{pq,S}=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}(x-c_x)^p(y-c_y)^q f(x,y)\mathrm{d}x\mathrm{d}y$$

中心矩是当坐标原点变换到图形中心所得到的矩。可以看出 $\mu_{00}=m_{00}$,$\mu_{01}=m_{01}$。其它低阶中心矩可由普通矩得到。如下列几个

$$\mu_{11,S}=m_{11,S}-m_{10,S}m_{01,S}/m_{00,S}$$

$$\mu_{02,S}=m_{02,S}-m_{01,S^2}/m_{00,S}$$

$$\mu_{20,S}=m_{20,S}-m_{10,S^2}/m_{00,S}$$

最小和最大惯性矩:图形 S 的最小惯性矩和最大惯性矩是图形分别围绕其长轴和短轴旋转而得到的。

$$I_{\min}=\{\mu_{20,S}+\mu_{02,S}-[4\mu_{11,S}^2+(\mu_{20,S}-\mu_{02,S})^2]^{1/2}\}/2$$

$$I_{\max}=\{\mu_{20,S}+\mu_{02,S}+[4\mu_{11,S}^2+(\mu_{20,S}-\mu_{02,S})^2]^{1/2}\}/2$$

几个不变矩:

$$I_1=\eta_{20,S}+\eta_{02,S}$$

$$I_2=(\eta_{20,S}-\eta_{02,S})^2+4\eta_{02,S}^2$$

$$I_3=(\eta_{30,S}-\eta_{02,S})^2+(3\eta_{21,S}-\eta_{03,S})^2$$

$$I_4=(\eta_{30,S}+\eta_{12,S})^2+(\eta_{21,S}-\eta_{03,S})^2$$

其中,$\eta_{pq,S}=\mu_{pq,S}/(\mu_{00,S})^{(p+q+2)/2}$。

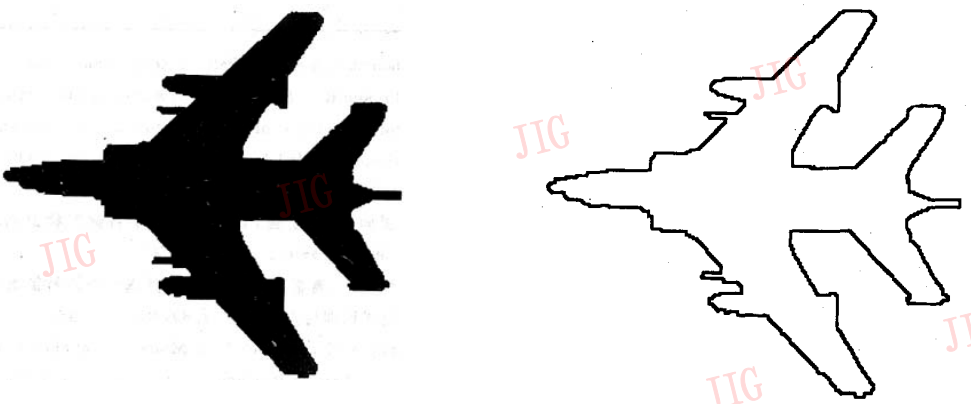
这些不变矩由于不受旋转、尺度变化的影响,因此在识别中起着十分重要的作用。

3 实 验

由公式(2)可以看出由点阵计算矩特征其复杂度正比于图形的象素个数。而由公式(5)可以看出由边界点计算矩特征其复杂度正比于边界的象素个数。这里,我们比较了分别采用图象点阵和边界点两种方法对同一图形进行计算的效率,以及周长面积比对该种算法效率的影响。为了客观比较不同算法的效率,采取以下措施。采用性能一样的微机(586×200),同一种编程语言(C 语言),有关矩及特征量采用同一种计算公式,对于采用不同方法的编程都尽量使其效率只依赖于其复杂度。对于两种算法都采用同一二值图象进行计算,因此由边界点计算矩特征需将边界跟踪所用的时间考虑进去。对图形或二值化后的图象,通过行扫描将所遇到的第一个属于对象的点作为初始点,然后按顺时针方向沿边界搜索即可获取对象的边界点。由于两种算法通过原点矩计算中心矩及不变矩的过程完全一样,因此仅以原点矩为实验计算对象。

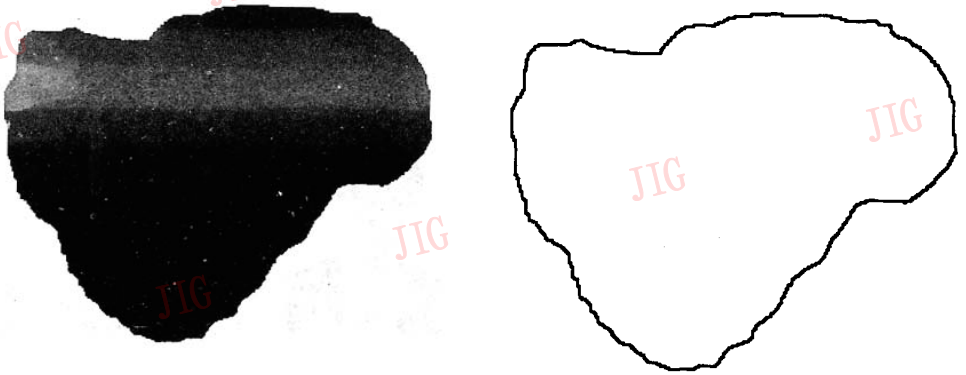
图 5 和图 6 分别为两个具有不同边界复杂程度(周长面积比)图形的点阵图(a)和边界点图(b),图象尺寸是 256×256。表 1 和表 2 分别为对这两个图形由图形点阵和边界点计算若干原点矩的结果、相对误差和耗时。从两表中的实验结果可以看出,由边界点计算所得的原点矩与用图形点阵所算的结果相对误差很小,但后者的耗时却比前者减少了一个或

接近一个数量级。通过对比两个表的参数 $R=t_a/t_e$ 可以看出,采用边界点计算矩特征时效率的提高对于第二个图形更为显著。因此,当周长面积比减小时,该算法效率的优势就更加明显。对图 5 中的图形分别在 64×64、128×128、256×256 及 512×512 尺寸的图象上进行计算,由表 3 可以看出,对同一图形,图象尺寸越大,即图形周长面积比越小时,该算法效率的提高就越显著。



(a) 飞机图形的点阵图　　(b) 飞机图形的边界轮廓图

图 5　飞机图形(256×256)



(a) 细胞核图形的点阵图　　(b) 细胞核图形的边界轮廓图

图 6　细胞核图形(256×256)

表 1　图 5 图形的原点矩、相对误差和耗时

$R=t_a/t_e=7.1$

	m_{00},s	m_{01},s	m_{10},s	m_{11},s	m_{02},s	m_{20},s	t(s)
点阵图	9.17×10^3	7.31×10^5	1.08×10^6	8.69×10^7	6.68×10^7	1.41×10^8	0.121
边界图	8.87×10^3	7.07×10^5	1.05×10^6	8.39×10^7	6.43×10^7	1.36×10^8	0.017
相对误差(%)	3.21	3.25	3.43	3.48	3.61	3.96	

表 2　图 6 图形的原点矩、相对误差和耗时

$R=t_a/t_e=21.0$

	m_{00},s	m_{01},s	m_{10},s	m_{11},s	m_{02},s	m_{20},s	t(s)
点阵图	1.81×10^4	1.42×10^6	2.19×10^7	2.38×10^8	2.24×10^8	5.09×10^8	0.231
边界图	1.76×104	1.38×10^6	2.12×10^7	2.31×10^8	2.16×10^8	4.91×10^8	0.011
相对误差(%)	2.84	2.89	2.91	2.98	3.21	3.46	

其中:相对误差(%)=|点阵图矩特征−边界图矩特征|/点阵图矩特征×100%。t_a 和 t_e 分别为采用点阵图和边界图计算原点矩的耗时矩特征的方法。对于用多边形近似误差不大的图形,可以考虑采用角点计算。有关角点检测算法及其门限的选取如何使得矩特征误差最小而计算效率最高是本方法需作

进一步完善和提高之处。

表 3 周长面积比与该算法效率的关系

	64×64	128×128	256×256	521×512
周长面积比	0.3081	0.1768	0.0893	0.0449
$R=t_a/t_e$	1.8	3.6	7.1	14.3

4 结 论

本文采用了边界点计算矩特征的方法。该方法的核心就是基于这样一个事实:二维图形可以由若干个以边界点与原点组成的三角形合成。三角形是最简单的二维形状。这些三角形的矩可以通过将其进一步分解而较容易得到。由于该算法的复杂度正比于边界点的象素数,而边界点的象素个数正比于图形总的象素个数的平方根。因此既使考虑了边界跟踪的耗时,其计算效率也远高于点阵计算。

参 考 文 献

1 Leu J G, Chen L. Polygonal approximation of 2-D shapes through boundary merging. Pattern Recognition, 1988, 7:231~238.

2 Prokop R J, Reeves A P. A survey of moment-based techniques for unoccluded object representation and recognition. CVGIP: Graphical Models and Image Processing, 1992, 54(5):438~460.

3 雷向康,周曼丽. 归一化矩用于目标形状识别. 信息与控制, 1991,6:58~62.

4 米航宇,离惠超,韦 穗. 机器视觉中矩特征的快速提取方法. 模式识别与人工智能, 1989,2(4):41~46.

5 Boyce J F, Hossack W J. Moment invariants for pattern recognition. Pattern Recognition Lett, 1983, 1:451~456.

6 Sluzek A. Using moment invariants to recognize and locate partially occluded 2D objects. Pattern Recognition Lett, 1983, 7:451~456.

董 武 1992 年毕业于长沙国防科学技术大学电子技术系,获图象分析与理解专业硕士学位。现任第一军医大学生物医学工程系医学图象实验室讲师。从事医学图象处理及模式识别研究工作。

李树祥 1963 年毕业于哈尔滨工业大学,长期从事医学图象领域的教学与研究工作,现任第一军医大学医工系主任,博士生导师,全国医学图象学会理事长,全国临床工程学会理事。1982—1985 年留学美国,从事医学图象处理与模式识别,人工智能研究。回国后主持的科研项目获国家发明奖一项、全军科技进步一等奖一项、二等奖三项。1987—1988 年、1993—1994 年又两次应邀赴美国华盛顿大学担任客座教授,1994 年回国后主持立项承担了国家"八五"攻关项目 X 刀系统的研制。

Moment Computation Through Boundary

Dong Wu, Li Shuxiang

(Department of Biomedical Engineering, The First Military Medical University, Guangzhou 510515)

Abstract The moment invariants play an important rule in feature extraction and pattern recognition. This paper presents an efficient method for moment computation. The experiment shows that the new method has higher speed than the conventional way of moment computation.

Keywords Moment, Moment invariants, Feature extraction