

滑动窗应用场合下的2维几何矩快速算法

柯 颀 叶 斌 胡修林

(华中科技大学电子与信息工程系, 武汉 430074)

摘 要 传统的2维几何矩算法着眼于单个矩形窗口,但当关心的矩形区域在大地图上滑动时,传统算法效率不高。为提高2维几何矩运算速度,提出了一种新的快速迭代算法。由于该算法能够充分利用相邻滑动窗重叠的像素信息,从而可以大大提高2维几何矩的计算效率。该算法所需的乘法和加法运算复杂度完全与滑动窗尺寸 $N \times L$ 无关,都为 $O(1)$ 。与传统算法的2维几何矩运算复杂度 $O(N \times L)$ 相比,该算法运算速度可以比传统算法提高接近 $N \times L$ 倍。计算机仿真结果验证了该结论。该速度可以满足大多数实时应用的需要。

关键词 几何矩 滑动窗 快速算法 2维

中图法分类号:TP391.41 文献标识码:A 文章编号:1006-8961(2008)06-1166-05

Fast Algorithm for 2-D Geometric Moment in Sliding-window Applications

KE Ting, YE Bin, HU Xiu-lin

(Department of Electronic Science and Engineering, Huazhong University of Science and Technology, Wuhan, 430074)

Abstract The traditional algorithms for geometric moment focus on the single rectangle windows. It has very low efficiency in the sliding-window applications. This issue prompts a new fast algorithm for 2-D geometric moment. It could efficiently reuse the overlapping pixels between conjoint sliding-windows, and achieves much higher computation efficiency. The average number of the multiplications and additions is $O(1)$, regardless of the sliding-windows size. Compared to the operation complexity of the traditional algorithm, known as $O(N \times L)$ in 2-D case, the new algorithm derived by this paper can improve its speed by the factor of nearly $N \times L$. A number of computer experiments are performed to validate this conclusion. This operation speed of this new algorithm can satisfy the requirements of most real time applications.

Keywords geometric moment, sliding window, fast algorithm, 2-D

1 引言

众所周知,(非中心)几何矩(geometric moment)^[1]是重要的图像区域特征描述算子。由于它提出的时间早、形式简单,而且利用几何矩可以快速构造出许多常用的不变矩函数,如Hu矩^[1],Zernike矩^[2]等,因此研究几何矩的快速算法具有普遍意义,即一旦几何矩的快速算法确定,其他不变矩的快速算法也随之确定。

2维几何矩在矩形窗口中的定义。国内外对单窗口情况研究比较充分,已提出了众多快速算法,如

Delta方法^[3]、格林定律法^[4]、基于多边形和三角形的方法^[5]、基于图像变换的方法^[6]以及基于IIR滤波法^[7]等。

但在地形匹配等诸多应用领域,常常将矩形匹配模板在大图像中滑动,用于提取覆盖区域内的矩信息,该匹配模板就被称作滑动窗。当窗口在图像中滑动时,相邻窗口具有大量的重叠像素,但新窗口内的“新息”比较少。尽管信息内在的冗余性决定了在滑动窗应用场合必定存在矩函数快速算法,但是这个研究领域却鲜有人涉足,仅有Martinez等人做过一些探索^[8]。Martinez等人提出几何矩与线性矩(linear moment)不仅具有对应关系,并且已推导出相邻滑动

窗线性矩的迭代更新公式。其算法的乘法复杂度为 $O(N)$, 加法复杂度为 $O(1)$, 其中 N 为窗口尺寸。Martinez 等人的文章晦涩难懂, 结论也不便运用。

本文对滑动窗应用场合中几何矩快速计算问题进行了新的探索。Canterakis 在研究 2 维 Zernike 矩快速计算问题时, 已得到了 2 维几何矩的一个变换公式^[2], 而本文则从 Canterakis 的研究成果出发, 重点研究了其变换公式中一个关键变量在相邻滑动窗中的迭代关系, 提出了 2 维几何矩快速算法, 并给出了详细运算流程。通过性能分析可见, 该算法的乘法和加法复杂度都与窗口尺寸无关, 都为 $O(1)$ 。但是与传统的单窗口算法相比, 该算法需要使用额外的存储空间, 其存储量与图像的最短边长成正比。

该算法是用空间换取时间, 特别适用于存储空间足够, 但却关心运算速度的应用场合。计算机仿真表明, 该算法的运算速度完全能够满足大多数实时应用的需要。

2 几何矩的定义和 Canterakis 的研究成果

图 1 是滑动窗搜索示意图, 图像的原点在左上角, 纵轴为 x 轴, 横轴为 y 轴, 坐标 (x, y) 处的像素灰度值记作 $f_{x,y}$ 。将左上角在 (a, b) 处, 高度为 N , 宽度为 L 的滑动窗记作 $w_{a,b,N,L}$ 。

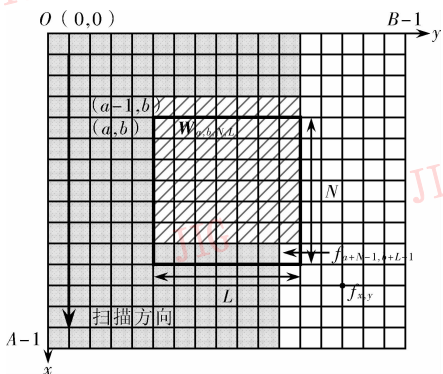


图 1 滑动窗搜索示意图

Fig. 1 Sketch for searching in sliding-window

$w_{a,b,N,L}$ 用矩阵形式表示为

$$w_{a,b,N,L} = \begin{bmatrix} f_{a,b} & f_{a,b+1} & \cdots & f_{a,b+L-1} \\ f_{a+1,b} & f_{a+1,b+1} & \cdots & f_{a+1,b+L-1} \\ \vdots & \vdots & \ddots & \vdots \\ f_{a+N-1,b} & f_{a+N-1,b+1} & \cdots & f_{a+N-1,b+L-1} \end{bmatrix} \quad (1)$$

滑动窗 $w_{a,b,N,L}$ 上的 $p+q$ 阶 2 维几何矩定义为^[1]

$$m_{p,q,a,b} \triangleq \sum_{n=0}^{N-1} \sum_{l=0}^{L-1} n^p l^q w_{a,b,N,L}(n,l) = \sum_{n=0}^{N-1} \sum_{l=0}^{L-1} n^p l^q f_{a+n,b+l} \quad (2)$$

几何矩的阶数增加反映了图像从粗糙到精致, 显示了在局部细节表征能力上的提高。为了全面描述图像的区域特征, 需要使用从低阶到高阶的一组几何矩信息。设用户关心的 2 维几何矩的最高阶数为 $P+Q$, 因此可通过构造 $(P+1) \times (Q+1)$ 维的 2 维几何矩矩阵 $m_{p,q,a,b}$ 来全面表征图像的区域特征, 其元素囊括了 $p=0, 1, \dots, P, q=0, 1, \dots, Q$ 的所有 $p+q$ 阶 2 维几何矩 $m_{p,q,a,b} \circ m_{p,q,a,b}$ 构造如下:

$$m_{p,q,a,b} = \begin{bmatrix} m_{0,0,a,b} & m_{0,1,a,b} & \cdots & m_{0,Q,a,b} \\ m_{1,0,a,b} & m_{1,1,a,b} & \cdots & m_{1,Q,a,b} \\ \vdots & \vdots & \ddots & \vdots \\ m_{P,0,a,b} & m_{P,1,a,b} & \cdots & m_{P,Q,a,b} \end{bmatrix} \quad (3)$$

直接推导相邻滑动窗内 $m_{p,q,a,b}$ 的迭代关系比较困难, 需要做一些变换。Canterakis 在研究 2 维 Zernike 矩快速计算问题时, 得到了一个 2 维几何矩的变换公式^[2], 即

$$m_{p,q,a,b} = T_p(B_N V_N w_{a,b,N,L} V_L B_L^T) T_Q^T \quad (4)$$

式中, $V_N \triangleq \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \\ 1 & & & \end{bmatrix}$, $T_p \triangleq \begin{bmatrix} 1 & & & \\ 0 & t_{1,1} & & \\ \vdots & \vdots & \ddots & \\ 0 & t_{p,1} & \cdots & t_{p,p} \end{bmatrix}$

其中, $t_{p,q} = (-1)^p \cdot \sum_{v=1}^q (-1)^v \binom{q}{v} v^p$;

$B_N \triangleq \begin{bmatrix} e_{1,N}^T + e_{N,N}^T C_N \\ e_{N,N}^T C_N^2 \\ e_{N,N}^T C_N^3 \\ \vdots \\ e_{N,N}^T C_N^{p+1} \end{bmatrix}$, 其中 $e_{1,N} \triangleq \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}_{N \times 1}$,

$e_{N,N} \triangleq \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}_{N \times 1}$, $C_N = \begin{bmatrix} 1 & & & \\ 0 & 1 & & \\ 0 & 1 & 1 & \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 1 & \cdots & 1 \end{bmatrix}_{N \times N}$

C_N^p 表示方阵 C_N 的 p 次幂, $\binom{\cdot}{\cdot}$ 表示组合数,

即 $\binom{n}{m} = \frac{n!}{m!(n-m)!}$ 。 V_L, C_L, T_Q 的定义类似可

得。本文中,记

$$\mathbf{D}_{a,b} = \mathbf{B}_N \mathbf{V}_N \mathbf{w}_{a,b,N,L} \mathbf{V}_L \mathbf{B}_L^T \quad (5)$$

将式(5)代入式(4)中可得

$$\mathbf{m}_{p,Q,a,b} = \mathbf{T}_p \mathbf{D}_{a,b} \mathbf{T}_Q^T \quad (6)$$

矩阵 $\mathbf{D}_{a,b}$ 是本文关注的对象。本文将重点推导相邻滑动窗内矩阵 $\mathbf{D}_{a,b}$ 的迭代更新关系。注意到式(6)中的 \mathbf{T}_p 和 \mathbf{T}_Q 都是维数很小的常数矩阵,因此计算 $\mathbf{m}_{p,Q,a,b}$ 的时间复杂度主要由 $\mathbf{D}_{a,b}$ 决定。

3 应用于滑动窗的几何矩快速算法

3.1 滑动窗应用场合下快速计算的内在机理

对图 1 中图像进行自上而下的列搜索,当前列搜索完后,再将滑动窗向右移到后一列的第 1 行,继续新的列搜索。当搜索新的(非边界处)滑动窗 $\mathbf{w}_{a,b,N,L}$ 时,由于只有像素 $f_{a+N-1,b+L-1}$ 从未曾在以前的搜索中出现过,因此窗口 $\mathbf{w}_{a,b,N,L}$ 的“新息”为 $f_{a+N-1,b+L-1}$,其信息量为 1pixel。由信息论的知识可知,由于“新息”的信息量决定了算法问题时间复杂度的理论下界,因此时间复杂度为 $O(1)$ 的快速算法总是存在的。这就为本文的快速计算问题提供了理论根据。该算法的优势在于能够充分利用历史处理中的冗余信息来实现快速计算,其劣势是必须开辟额外的存储空间,以记住过往的历史。因此算法的关键在于用空间换时间。

3.2 关键计算公式和算法流程

为简洁起见,本文只研究列扫描情况,行扫描结论类似可得。当作列扫描时,窗口由 $\mathbf{w}_{a,b,N,L}$ 移动到 $\mathbf{w}_{a+1,b,N,L}$ 。重点研究 $\mathbf{D}_{a+1,b}$ 和 $\mathbf{D}_{a,b}$ 的迭代关系。由于详细公式推导十分繁琐,受篇幅所限,因此这里只给出关键计算步骤。

定义 $(Q+1) \times 1$ 维的列向量 $\mathbf{r}_{a,b}$,其元素为

$$r_{a,b}(q) = \sum_{k=1}^{L-1} f_{a,b+L-k} \binom{q+L-1-k}{q}, q=0, \dots, Q \quad (7)$$

$\mathbf{r}_{a,b}$ 具有横向更新公式,即

$$\begin{cases} r_{a,b+1}(0) = f_{a,b+L} - f_{a,b+1} + r_{a,b}(0) \\ r_{a,b+1}(q) = \binom{q+L-2}{q} f_{a,b+L} + r_{a,b}(q) - r_{a,b}(q-1) \\ q=1, \dots, Q \end{cases} \quad (8)$$

$\mathbf{D}_{a,b}(p,q)$ 有纵向更新公式

$$D_{a+1,b}(0,0) = D_{a,b}(0,0) + r_{a+N,b}(0) - r_{a,b}(0) +$$

$$f_{a+N,b} - f_{a,b}$$

$$D_{a+1,b}(0,q) = D_{a,b}(0,q) + r_{a+N,b}(q) - r_{a,b}(q) \\ q=1, \dots, Q$$

$$D_{a+1,b}(1,0) = D_{a,b}(1,0) - D_{a,b}(0,0) + \\ (N-1)r_{a+N,b}(0) + r_{a,b}(0) + \\ (N-1)f_{a+N,b} + f_{a,b}$$

$$D_{a+1,b}(1,q) = D_{a,b}(1,q) - D_{a,b}(0,q) + (N-1) \times \\ r_{a+N,b}(q) + r_{a,b}(q), q=1, \dots, Q$$

$$D_{a+1,b}(p,0) = D_{a,b}(p,0) - D_{a,b}(p-1,0) + \\ \binom{p+N-2}{p} r_{a+N,b}(0) + \binom{p+N-2}{p} \times \\ f_{a+N,b}, p=2, \dots, P$$

$$D_{a+1,b}(p,q) = D_{a,b}(p,q) - D_{a,b}(p-1,q) + \\ \binom{p+N-2}{p} r_{a+N,b}(q) \quad (9)$$

$$p=2, \dots, P; q=1, \dots, Q$$

图 2 为快速算法流程图。图 2(a) 为主程序,图 2(b) 为列扫描子程序。算法流程为:先用定义式计算第 1 列的 $\mathbf{r}_{a,b}$;然后利用式(8)迭代更新后续列的 $\mathbf{r}_{a,b}$ 。在扫描每一列时,先用定义式计算第 1 行的 $\mathbf{D}_{0,b}$,然后利用式(9)迭代更新后续行的 $\mathbf{D}_{a,b}$,最后利用变换式(6)通过 $\mathbf{D}_{a,b}$ 来计算当前滑动窗的 2 维几何矩阵 $\mathbf{m}_{p,Q,a,b}$ 。

4 算法性能分析与仿真

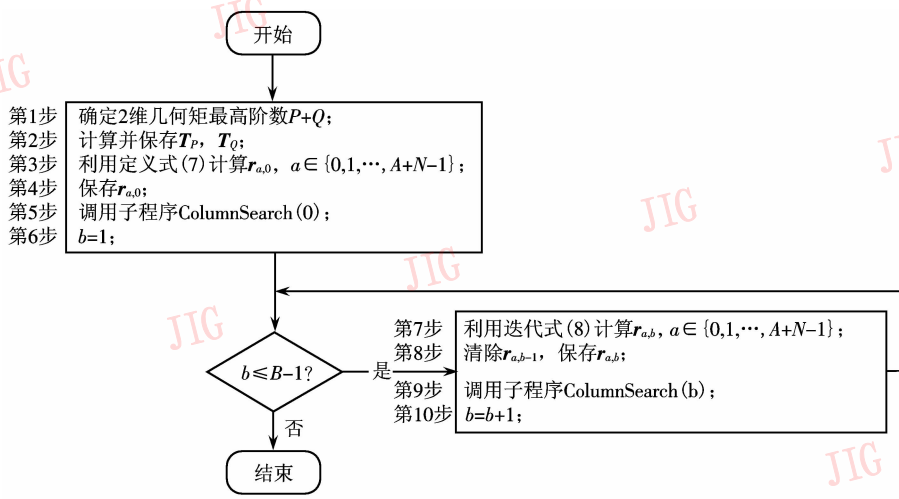
4.1 算法性能分析

假设图像扫描区域尺寸 $A \times B$ (见图 1) 远大于窗口大小 $N \times L$, 因此,在分析算法时间复杂度时,搜索图像边界(图 2 中主程序和子程序的初始化部分)的耗时可以忽略,而只需要关心算法流程图中的循环主体部分。文中还假设窗口大小 $N \times L$ 远大于几何矩最高阶数 $P+Q$, 那么主程序第 7 步利用迭代式(8)更新整列的 $\mathbf{r}_{a,b}$, $a \in \{0, 1, \dots, A+N-1\}$, 总共需要 $Q(A+N)$ 次乘法, $(2Q+2)(A+N)$ 次加法。如果将总时间分摊到 A 个滑动窗上,则每次计算约花费 Q 次乘法, $2(Q+1)$ 次加法;子程序第 15 步利用迭代式(9)更新 $\mathbf{D}_{a,b}$, 需要 $P(Q+2)$ 次乘法, $\frac{(2P+3)(2Q+3)+3}{2}$ 次加法;第 17 步利用式(6)计算 $\mathbf{m}_{p,Q,a,b}$, 需要 $\frac{P(P+1)+Q(Q+1)}{2}$ 次乘法, $\frac{P(P-1)+Q(Q-1)}{2}$ 次加法,而总运算时间则为上

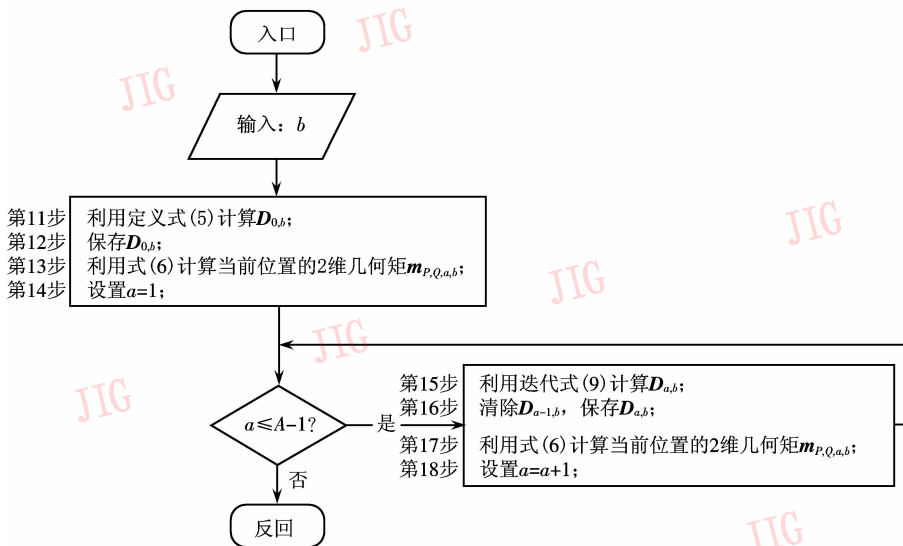
述 3 个步骤耗时的总和。显然,该算法迭代更新 $m_{P,Q,a,b}$ 所需的乘法和加法次数都是 P 和 Q 的函数,而与窗口大小 $N \times L$ 无关,其乘法和加法时间复杂度都是 $O(1)$ 。这个结论与扫描方式无关。

下面分析存储空间使用情况。主程序第 2 步创建 $(P+1)^2 + (Q+1)^2$ 个存储单元,用于存放 T_P 和 T_Q ;第 4 步创建 $(Q+1)(A+N)$ 个存储单元,用于存放 $r_{a,b}$;第 8 步复用该存储单元,并且不断更新 $r_{a,b}$ 。

子程序第 12 步创建 $(P+1)(Q+1)$ 个存储单元,用于存放 $D_{a,b}$;第 16 步复用该存储单元,并且不断更新 $D_{a,b}$ 。由于 $A \gg N \gg P$ 和 Q ,因此总占用空间约为 $(Q+1)A$ 个存储单元。同理可得,采用行扫描方式,总占用空间约为 $(P+1)B$ 个存储单元。为了减少存储空间使用量,需要优选扫描方式,即当 $A \leq B$ 时,应采用列扫描方式;当 $A > B$ 时,应采用行扫描方式。这时,总的存储空间使用量与图像的最短边长成成正比。



(a) 主程序流程图



(b) 列扫描子程序流程图

图 2 快速算法流程图

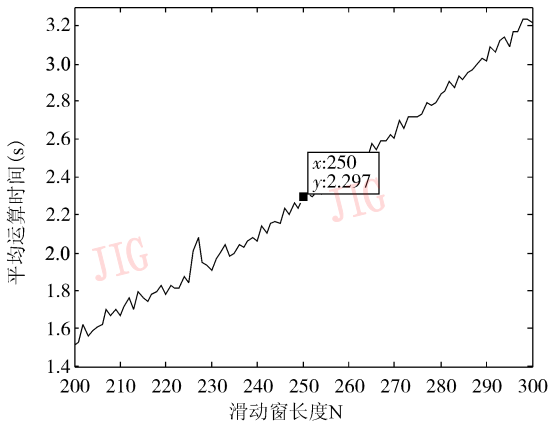
Fig. 2 Operation flow for proposed fast algorithm

4.2 计算机仿真验证

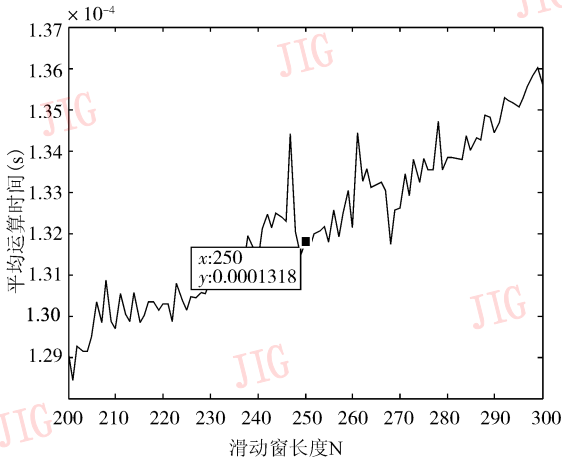
计算机仿真的目的有以下两个:一是验证计算的正确性;二是展现本文提出的新算法的时间优越性。为此,将该新算法与利用定义式(2)和式(3)在

单窗口内直接计算 2 维几何矩的传统算法相比较。显然,传统算法的乘法和加法时间复杂度都是 $O(N \times L)$ 。实验中,使用 512×512 大小的数字地形图作为图像扫描区域。滑动窗大小为 $N \times N$, N 的取值可

变。用本文及传统两种算法计算每个滑动窗内 $5 + 4$ 阶的 2 维几何矩阵 $m_{5,4,a,b}$ 。实验表明,使用两种算法获得的 $m_{5,4,a,b}$ 在数值上完全相同,从而检验了本文算法的正确性。图 3 为两种算法在计算单窗口 2 维几何矩时的平均运算时间随窗口尺寸 N 变化的关系曲线。根据图 3 可以定量比较两种算法在运算速度上的差距。如图 3 所示,当 $N = 250$ pixel 时,使用传统算法需要用时 2.297s,而使用本文算法仅需要用时 0.13ms。与传统算法相比,本文算法的运算速度提高了 4 个数量级,接近 N^2 倍。该运算速度显然能够满足大多数实时应用的需要。



(a) 传统算法



(b) 本文提出的算法

图 3 平均运算时间随滑动窗尺寸变化的关系曲线

Fig. 3 The curve of average running time with windows size

5 结 论

本文从信息论角度出发,阐述了在滑动窗应用场合下,必定存在矩函数的快速迭代算法,并且对 2 维几何矩快速算法进行了研究。与传统的单窗口算法相比,本文提出的快速算法是用存储空间换取运算速度。该算法的乘法和加法复杂度都为 $O(1)$,且与窗口尺寸无关,其所需要的存储量与图像的最短边长成正比。计算机仿真验证了算法的正确性,并且表明其运算速度能够满足大多数实时应用的需要。该算法尤其适合在存储空间足够,但关心运算速度的场合中使用。

参考文献 (References)

- Gonzalez R C, Woods R E. Digital Image Processing (2nd Edition) [M]. New York: Prentice-Hall, 2002.
- Canterakis N. Fast 3D Zernike Moments and Invariants[R], Technical Report 5-97. Institute for Computer Science, University of Freiburg, Freiburg, Germany, 1997.
- Zakaria M F, Vroomen L J, Zsombor-Murray P J A, et al. Fast algorithm for the computation of moment invariants [J]. Pattern Recognition, 1987, 20(6): 639 ~ 643.
- Li B C. A new computation of geometric moments [J]. Pattern Recognition, 1993, 26(1): 109 ~ 13.
- Jiang X Y, Bunke H. Simple and fast computation of moments [J]. Pattern Recognition, 1991, 24(8): 801 ~ 806.
- Shen Tak-Wai, Lun Daniel P K, Siu W C. Fast algorithm for 2-D image moments via the radon transform [A]. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing [C], Atlanta, GA, USA, 1996, 3: 1327 ~ 1330.
- Hatamian M A. Real-Time two dimensional moment generating algorithm and its signal chip implementation [J]. IEEE Transactions on Acoustics, Speech and Signal Processing, 1986, 34(3): 546 ~ 553.
- Martinez J, Staffetti E, Thomas F. A recursive updating rule for efficient computation of linear moments in sliding-window applications [A]. In: Proceedings of the 13th International Conference on Pattern Recognition [C], Los Alamitos, CA, USA, 1996, 2: 295 ~ 299.