第 16 卷 第 5 期
1990 年 9 月

自 动 化 学 报
ACTA AUTOMATICA SINICA

Vol. 16, No. 5
Sept., 1990

局部统计量的快速计算算法

董莉敏　　韦　钰
（南京师范大学）　　（东南大学）

摘　要

本文提出了在领域中计算局部统计量的一个快速算法. 根据局部统计量的性质，设计卷积算子，将该算子与图象作循环卷积以达到快速计算的目的. 比较了循环卷积算子和线性卷积算子在计算局部统计量上的异同点. 本算法的最大特点在于计算速度与运算空间不随计算窗口的变化而变化. 植物染色体图象的实验表明：本算法较在时域中直接运算约快 6 倍.

关键词——算子，卷积，局部统计量.

一、引　言

局部统计量通常指局部均值、方差等，它们是数字图象处理中经常用到的特征量. 假设有一幅大小为 $M \times N$ 的图象 $[G]$，点 (i, j) 在大小为 $(2L+1) \times (2L+1)$ 窗口下的局部统计量的计算可一般地表示为

$$\begin{aligned} w(i, j) = {} & a(-L, -L)g(i-L, j-L) + \cdots + a(-L, L)g(i-L, j+L) \\ & + a(-L+1, -L)g(i-L+1, j-L) + \cdots \\ & + a(-L+1, L)g(i-L+1, j+L) + \\ & \cdots\cdots \\ & + a(L, -L)g(i+L, j-L) + \cdots + a(L, L)g(i+L, j+L). \end{aligned} \tag{1}$$

式中 $w(i, j)$ 代表局部统计量，运算系数 $a(i, j)$ 由局部统计量的性质决定. 通常，$w(i, j)$ 是直接在时域中利用(1)式计算得到的. 不难看出，随着窗口不断加大，计算一个点的局部统计量所作的加法和乘法次数成比例地增加. 文献 [1,2] 指出局部统计量的计算也可用线性卷积来完成，并且给出了相应的卷积算子. 但是，利用线性卷积算子计算局部统计量，无论是计算速度还是运算空间都随着窗口的扩大而增长. 本文提出的快速算法是利用循环卷积算子而不是线性卷积算子，它的计算速度与运算空间与窗口大小无关.

二、循环卷积的数学模型

设图象 $[G]$ 可表示为

本文于 1987 年 11 月 10 收到.

$$[G]=\begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1N} \\ g_{21} & g_{22} & \cdots & g_{2N} \\ & \cdots\cdots & & \\ g_{M1} & g_{M2} & \cdots & g_{MN} \end{bmatrix},$$

将它进行周期性延拓，则有

$$\begin{array}{ccc} \vdots & \vdots & \vdots \\ \cdots\cdots\begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1N} \\ g_{21} & g_{22} & \cdots & g_{2N} \\ & \cdots\cdots & & \\ g_{M1} & g_{M2} & \cdots & g_{MN} \end{bmatrix} & \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1N} \\ g_{21} & g_{22} & \cdots & g_{2N} \\ & \cdots\cdots & & \\ g_{M1} & g_{M2} & \cdots & g_{MN} \end{bmatrix} & \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1N} \\ g_{21} & g_{22} & \cdots & g_{2N} \\ & \cdots\cdots & & \\ g_{M1} & g_{M2} & \cdots & g_{MN} \end{bmatrix}\cdots\cdots \\ \cdots\cdots\begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1N} \\ g_{21} & g_{22} & \cdots & g_{2N} \\ & \cdots\cdots & & \\ g_{M1} & g_{M2} & \cdots & g_{MN} \end{bmatrix} & \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1N} \\ g_{21} & g_{22} & \cdots & g_{2N} \\ & \cdots\cdots & & \\ g_{M1} & g_{M2} & \cdots & g_{MN} \end{bmatrix} & \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1N} \\ g_{21} & g_{22} & \cdots & g_{2N} \\ & \cdots\cdots & & \\ g_{M1} & g_{M2} & \cdots & g_{MN} \end{bmatrix}\cdots\cdots \\ \vdots & \vdots & \vdots \end{array}$$

假如把式(1)中的运算系数进行适当排列以构造矩阵 $[A]_{(2L+1)\times(2L+1)}$ 如下：

$$[A]=\begin{bmatrix} a_{-L,-L} & a_{-L,-L+1} & \cdots & a_{-L,L} \\ a_{-L+1,-L} & a_{-L+1,-L+1} & \cdots & a_{-L+1,L} \\ & \cdots\cdots & & \\ a_{L,-L} & a_{L,-L+1} & \cdots & a_{L,L} \end{bmatrix}, \tag{2}$$

那么，图象[G]中任一点 (i, j) 的局部统计量计算可以看作是把 $[A]$ 中的 $(0, 0)$ 点与 $]G]$ 中的 (i, j) 点对齐．根据式(1)，把窗口内相应位置上的元素相乘，然后将乘积相加．矩阵 $[A]$ 在图象中不断移动，就可完成所有点的局部统计量计算．下面，用图象空间的平面图说明这一过程．

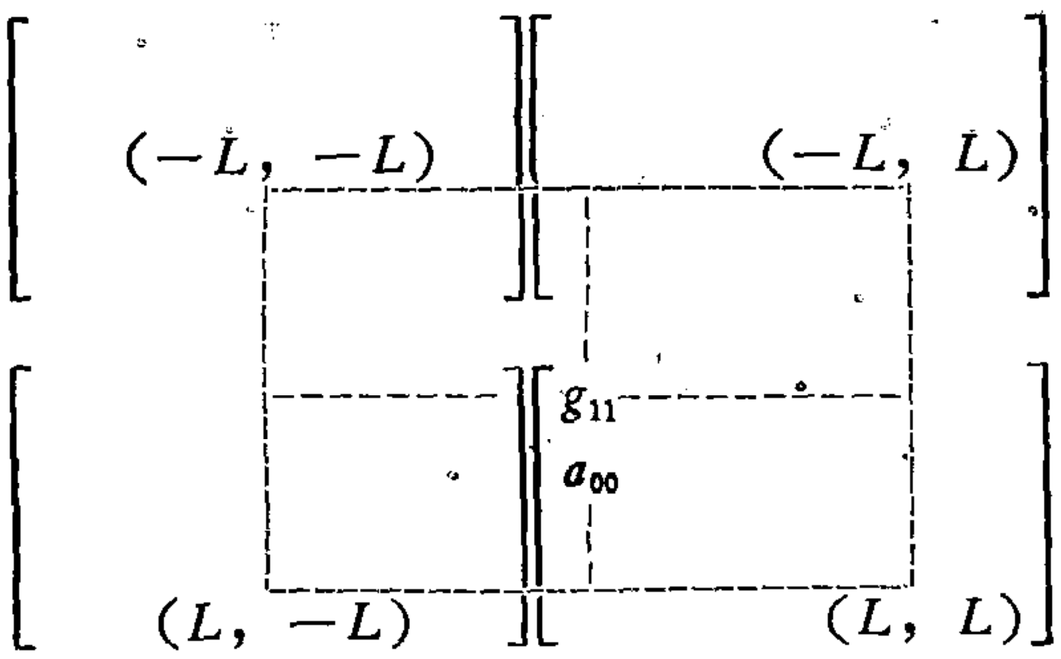

图 1

如图 1 所示，假如要计算点 $(1, 1)$ 的局部统计量，根据上述方法，只要把 $[G]$ 中的点 $(1, 1)$ 与 $[A]$ 中的点 $(0,0)$ 对齐，计算对应位置上的元素乘积，并将乘积相加即可．图 1 中虚线方框包围的区域就是乘积运算的范围．可以看出，用这种方法算出来的结果与直接利用(1)式得到的结果是完全一致的．但由于 CPU 空间有限，不可能把延拓图象全部存于计算机中；可是，$[G]$ 中有些点的局部统计量计算又要用到延拓后的图象象素值．为解决这一矛盾，必须寻找一种新的有效途径来替代延拓图象．首先把 $[A]$ 分成四块（见图 1），由于 $[G]$ 是周期延拓的，所以 $[A]$ 中右上角利用的图象象素可用 $[G]$ 中左下角的象素来替代，$[A]$ 中左下角利用的图象象素可用 $[G]$ 中右上角的象素来替代．同样，$[A]$ 左上角用到的图象象素可用 $[G]$

中右下角的象素来替代。 由于 $[A]$ 中右下角用到的象素已在 $[G]$ 中，故没有替代的必要。考虑到循环卷积的性质，可以设计循环卷积算子 $[P]_{M\times N}$ 如下：

$$[P]=\begin{bmatrix}\begin{bmatrix}a_{0,0} & \cdots & a_{0,-L}\\ \cdots & \cdots & \cdots\\ a_{-L,0} & \cdots & a_{-L,-L}\end{bmatrix} & 0 & \begin{bmatrix}a_{0,L} & \cdots & a_{0,1}\\ \cdots & \cdots & \cdots\\ a_{-L,L} & \cdots & a_{-L,1}\end{bmatrix}\\ 0 & & 0\\ \begin{bmatrix}a_{L,0} & \cdots & a_{L,-L}\\ \cdots & \cdots & \cdots\\ a_{1,0} & \cdots & a_{1,-L}\end{bmatrix} & 0 & \begin{bmatrix}a_{L,L} & \cdots & a_{L,1}\\ \cdots & \cdots & \cdots\\ a_{1,L} & \cdots & a_{1,1}\end{bmatrix}\end{bmatrix}, \tag{3}$$

式中，虚线框以外的元素为零。 这样，$[P]_{M\times N}$ 与 $[G]_{M\times N}$ 的循环卷积就可以完成图象中任一点的局部统计量计算，即：

$$w(i,j)=\left[\sum_{m=1}^{M}\sum_{n=1}^{N}\tilde{g}(m,n)\tilde{p}(i-m,j-n)\right]\cdot\eta_{M,N}(i,j). \tag{4}$$

其中，符号“～”表示周期图象，$\eta_{M,N}(i,j)$ 定义如下：

$$\eta_{M,N}(i,j)=\begin{cases}1, & 1\leqslant i\leqslant M,\ 1\leqslant j\leqslant N,\\ 0, & \text{其它点}.\end{cases}$$

式(4)可用熟知的 FFT 进行快速计算。

三、与线性卷积的比较

局部统计量的计算同样可通过构造线性卷积来完成。 循环卷积与线性卷积相比较，两者有二点不同：

1) 图象的结构不一样。 利用线性卷积计算，要求把 $[G]_{M\times N}$ 延拓成 $[G']_{(M+2L+1)\times(N+2L+1)}$，其中，$[G']$ 中的元素定义为

$$g'(i,j)=\begin{cases}g(i,j), & 1\leqslant i\leqslant M,\ 1\leqslant j\leqslant N,\\ 0, & \text{其它点}.\end{cases}$$

利用循环卷积计算，没有转换的必要。

2) 卷积算子构造不一样。 循环卷积算子构造如前所述，而线性卷积算子 $[P']_{(M+2L+1)\times(N+2L+1)}$ 的构造为

$$[P']=\begin{bmatrix}\begin{bmatrix}a_{L,L} & \cdots & a_{L,-L}\\ a_{L-1,L} & \cdots & a_{L-1,L}\\ \cdots & \cdots & \cdots\\ a_{-L,L} & \cdots & a_{-L,-L}\end{bmatrix} & 0\\ 0 & \end{bmatrix}, \tag{5}$$

式中虚线框外的元素为零。

它们的共同之处是：循环卷积利用 FFT 完成，线性卷积也是转换成循环卷积，利用 FFT 完成.

根据上述分析，对于同样大小的图象与窗口，在运算空间这一点上，显然循环卷积优于线性卷积. 从运算时间上分析，计算一幅大小为 $M \times N$ 图象的局部统计量，循环卷积所花费的时间是 $O(MN\log_2\max(M, N))$；线性卷积是 $O((M+2L+1)(N+2L+1)\log_2\max((M+2L+1),(N+2L+1)))$；直接在时域中计算所花的时间是 $O(MN(2L+1)^2)$. 三者的优劣次序为：循环卷积优于线性卷积，线性卷积优于时域中直接计算. 三者之中只有循环卷积的运算空间和计算时间与窗口大小无关.

四、实 验 结 果

在植物染色体图象的预处理中，应用上述快速循环卷积算法计算了其中的局部均值和局部方差[4]. 实验在 VAX 11/730 机上进行，图象大小为 64 × 64，局部均值和局部方差在 7 × 7 的窗口下计算. 图 2(a) 和 (b) 给出了基于本算法的大麦染色体复原前后的图象. 实验表明：利用循环卷积算法比在时域中直接计算约快 6 倍. 当窗口大时，本算法的效果将更为明显.

图 2 大麦染色体图象

参 考 文 献

[1] 奥本海姆，A. V.，谢弗，R. W.，数字图象处理学，科学出版社，1983.

[2] Gongzalez, R. C. and Wintz, P., Digital Image Processing, Addison-Wesley Publishing Company, Inc., 1977.

[3] Lee, J. S., Computer Graphics and Image Processing, **15**(1981), 380—389.

[4] 董莉敏、韦钰，植物染色体图象的自适应复原，中国生物医学工程学报，**8**(1989)，2，102—109.

A FAST ALGORITHM FOR LOCAL STATISTICS

DONG LIMIN
(*Nanjing Normal University*)

WEI YU
(*Southeast University*)

ABSTRACT

A fast algorithm for calculating local statistics in frequency domain is presented. The convolution operator is constituted by means of the nature of local statistics. It is circularly convoluted with images to complete fast operations. The circular convolution operator is compared with the linear convolution operator in the calculation of local statistics. The distinguishel feature of the present algorithm lies in that its computing time and storage requirement is independent of the size of the local window. The experiment on plant chromosome images shows that the present algorithm is about 6 times as fast as the direct space domain calculation.

Key words ——Operator; convolution;local statistics.