

文章编号 1004-924X(2018)10-2565-10

归一化互相关中计算基准子图能量的快速递推

韩 冰^{1*}, 牟忠锋¹, 乐小峰¹, 贾小志¹, 石选卫¹, 李贝贝^{1,2}

(1. 长光卫星技术有限公司, 吉林 长春 130033;

2. 中国科学院 长春光学精密机械与物理研究所, 吉林 长春 130033)

摘要:景象匹配对匹配算法的运行速度和内存占用均要求较高。为提升归一化互相关算法的运行速度并降低其内存占用率,本文重点对其中的基准子图能量计算步骤进行了加速研究。经过详细分析,积分图法具有灵活、快速的优点,但缺陷为其在快速计算的同时需花费较大内存,并不适合直接应用在嵌入式系统中。本文提出了一种快速递推算法。该算法利用相邻像素值的能量进行连续递推,计算时不必像积分图法那样给所有的图像能量都分配空间,只需预留 1 行的像素空间便能完成整个能量计算过程。实验结果表明:在时间花费方面,快速递推法具有和积分图法相当的运算速度,耗时均只为传统归一化互相关算法的 1/2;在内存占用率方面,快速递推法约为积分图法的 1/3 以下,且实时图尺寸越大,快速递推法占用的内存越小。综上所述,在归一化互相关算法中利用经典积分图法和本文提出的快速递推法计算基准子图能量,均较传统 NCC 算法有所加速,两种算法各具优点,经典积分图法快速、灵活,适用于对速度要求高,但对内存占用率要求不太高的应用场景;而快速递推法快速、省内存,更适用于嵌入式系统的应用。

关键词:归一化互相关;基准子图能量;经典积分图法;快速递推法

中图分类号:TP301.6 文献标识码:A doi:10.3788/OPE.20182610.2565

Fast recurrence algorithm for computing sub-Image energy using normalized cross correlation

HAN Bing^{1*}, MU Zhong-feng¹, LE Xiao-feng¹,

JIA Xiao-zhi¹, SHI Xuan-wei¹, LI Bei-bei^{1,2}

(1. Chang Guang Satellites Technology Co., Ltd., ChangChun 130033, China;

2. Changchun Institute of Optics, Fine Mechanics and Physics,

Chinese Academy of Sciences, ChangChun 130033, China)

*Corresponding author, E-mail: 334679010@qq.com

Abstract: Scene matching requires higher matching speed and memory usage. In order to improve the running speed of the normalized cross correlation algorithm and reduce its memory occupancy rate, this paper focus on researching the steps of fast calculating sub-image's energy. After detailed analysis, the integral graph method has the advantages of flexible and rapid, but the defect is that it needs to spend a lot of memory at the same time, while it is not suitable for the embedded system. Therefore, a fast recurrence method was proposed. In this method, the energy of adjacent pixel values is used to contin-

收稿日期:2018-02-10;修订日期:2018-04-23.

基金项目:国家重点研发计划资助项目(No. 2016YFB0502605)

uously recursive compute. It is not necessary to allocate space for all image energy as the integral image method in the calculation process. Only one row of space can be reserved for the entire energy calculation process in fast recurrence method, which greatly saves the memory usage. The fast recurrence method has the equivalent calculation speed with the integral image method, and the time consuming is only 1/2 of the traditional normalization cross correlation algorithm. In the memory occupancy rate, the fast recurrence method is less than 1/3 of the integral image method, and the larger the size of the real-time graph, the less memory occupied by the fast delivery method. In the normalized cross correlation algorithm, the classical integral graph method and the fast recursive method proposed in this paper are used to calculate the energy of the sub-image's energy, which are both faster than the traditional NCC algorithm. The two algorithms have their advantages. The classical integration image method is fast and flexible, which is suitable for the application scene with high speed requirements, but the memory occupancy rate is not very high. The fast recursive method is fast and saves memory, and is more suitable for the application of embedded systems.

Key words: Normalize Cross Correlation(NCC); sub-image's energy; classic integral image method; fast Recurrence method

1 引言

所谓景象匹配就是指将事先制备好的基准图与传感器拍摄的实时图进行相关运算,搜索实时图在基准图中位置的过程。目前众多相关算法中,效果最好的是归一化互相关算法(Normalized Cross Correlation, NCC),具有抵抗噪声和图像亮度变化的强鲁棒性,且匹配精度高等优点,因此,被广泛应用于景象匹配技术中。

NCC 算法应用于景象匹配的场景多见于执行目标识别跟踪任务的嵌入式系统^[1-4],除了对匹配精度有较高要求外,对匹配算法的运算速度和内存占用也有较高要求。对 NCC 算法进行优化加速,主要从两个方向进行,一是对其中的卷积运算进行加速,二是对其中计算基准子图能量的运算步骤进行加速。目前,国内外已有大量针对 NCC 算法中的卷积运算进行加速的研究成果^[5-7],归纳一下卷积加速通常会采用两种方法:一是时域法,即将卷积运算在时间域中进行。该方法利用数字信号处理器 DSP 或图形处理器 GPU 的并行流水结构,采用所谓的乒乓算法^[5-7]等进行加速运算。二是频域法,即卷积运算在频域中进行。该方法利用卷积定理,使用二维快速傅立叶变换 FFT2,将卷积运算从时间域变换到

频率域中进行^[8-9],从而达到加速目的。对于基准子图能量的运算步骤进行加速,目前学术成果中,最为经典的为 Lewis 提出的积分图法^[10],即将积分图像经典计算理论引入归一化积相关中,将复杂的区域像素逐一相加运算,简化成 3 个加减法,从而提升了运算速度。

本文重点针对 NCC 中计算基准子图能量的运算步骤进行加速研究。通过对经典 NCC 积分图法^[10]进行详细分析发现,虽然积分图法过程灵活,加速明显,但在内存占用较大,对于嵌入式系统的应用,无法完全满足速度高和内存占用少的要求。因此,针对以上问题,本文提出了一种快速递推算法,其主要思想是利用相邻像素值的能量进行连续递推,计算时可以不必要像积分图法那样给所有的图像能量都分配空间,只需预留 1 行的像素空间便能完成整个能量计算过程,因而即保证了快速的计算速度,又大大节省了内存占用。经过试验分析,在运行速度方面,基于快速递推法的 NCC 算法,和基于积分图法的 NCC 算法相比,两者运行速度相当,耗时均为传统 NCC 算法的 1/2;在内存占用率方面,景象匹配应用中,实时图尺寸越大,快速递推法比积分图法更加节省内存,在实时图大小为 32×32 的条件下,快速递推法的内存占用率为积分图法的 1/3;在实时图

大小为 350×350 条件下,快速递推法的内存占用率仅为积分图法的 $1/8$ 。在灵活性方面,由于快速递推算法需要连续进行递推完成,因此与经典积分图法相比,积分图法的灵活性更占优势。综上所述,本文提出的快速递推算法和经典积分图法均为加速优化传统归一化互相关算法的有效途径,两种方法各有优势,积分图法快速、灵活,但内存占用率较大,适合在非嵌入式系统进行应用;本文的快速递推法快速、内存占用小,适合在嵌入式系统中进行应用。

2 NCC 中基准子图能量的定义

归一化互相关算法(Normalized Cross Correlation, NCC)是执行景象匹配的经典算法之一,其基本原理如下:设有基准图像 f (高 H , 宽 W), 实时图像 g (高 M , 宽 N), 满足 $H \geq M, W \geq N$, 在位置 (r, c) 处, 基准图与实时图的相关系数可定义为:

$$\rho(r, c) = \frac{\sum_{i=1}^M \sum_{j=1}^N (g_{ij} - \bar{g})(f_{i+r, j+c} - \bar{f}_r)}{\sqrt{\sum_{i=1}^M \sum_{j=1}^N (g_{ij} - \bar{g})^2 \sum_{i=1}^M \sum_{j=1}^N (f_{i+r, j+c} - \bar{f}_r)^2}} \quad (1)$$

对式(1)的进行重新整理推导,可得更为简洁的形式:

$$\rho(r, c) = \frac{S_{gf} - S_g S_f / MN}{\sqrt{(S_{gg} - S_g^2 / MN)(S_{ff} - S_f^2 / MN)}} \quad (2)$$

其中:

$$S_{gf} = \sum_{i=1}^M \sum_{j=1}^N g_{ij} f_{i+r, j+c}, \quad (3)$$

$$S_{ff} = \sum_{i=1}^M \sum_{j=1}^N f_{i+r, j+c}^2, \quad (4)$$

$$S_f = \sum_{i=1}^M \sum_{j=1}^N f_{i+r, j+c}, \quad (5)$$

$$S_{gg} = \sum_{i=1}^M \sum_{j=1}^N g_{ij}^2, \quad (6)$$

$$S_g = \sum_{i=1}^M \sum_{j=1}^N g_{ij}. \quad (7)$$

公式(3)为基准图中的基准子图与实时图的卷积;公式(4)、(5)分别为基准图中基准子图的像素灰度平方和及像素灰度和,也称基准子图能量;公式(6)、(7)为实时图像像素灰度平方和及像素灰度和。

基准图与实时图在各点的相关系数可以组成大小 $(H-M+1) \times (W-N+1)$ 的相关矩阵, 如式(8)所示。通常, 相关矩阵中相关系数最大值处就可被认为是实时图在基准图中的位置。

$$R = \begin{bmatrix} \rho(0,0) & \rho(0,1) & \cdots & \rho(0,W-N) \\ \rho(1,0) & & & \cdots \\ \cdots & & & \cdots \\ \cdots & & & \cdots \\ \rho(H-M,0) & \cdots & \cdots & \rho(H-M,W-N) \end{bmatrix}. \quad (8)$$

对于式(2),最费时间的是公式(3)所示的基准子图与实时图的卷积运算、公式(4)和(5)所示的基准子图的像素灰度平方和以及灰度求和运算。对于实时系统和在线系统,上述三项运算往往成为使用的瓶颈。而公式(6)、(7)所示的实时图像像素灰度平方和以及灰度求和一般只进行一次运算,其运行时间可以忽略不计。

对于公式(3)NCC 中卷积过程的加速运算,已有许多研究成果^[5-7]。因此本文主要针对公式(4)、(5)计算基准子图能量的加速算法进行研究。分析了快速求取基准子图能量的经典积分图法的算法过程^[8],以及消耗内存大的主要问题。

3 经典积分图法原理分析

3.1 积分图像基本原理

积分图像(Integral image)^[11]由 Crow 提出,后应用在人脸识别^[12]领域中,最早将积分图像用于 NCC 基准子图能量计算的为 Lewis^[10]。积分图像的定义如图 1 所示,图像中任意一点 (i, j) 的积分值 $ii(i, j)$ 、 $ii2(i, j)$, 分别定义为原图像相应斜线区域灰度值和、灰度值平方和,即:

$$ii(i, j) = \sum_{i' \leq i, j' \leq j} p(i', j'), \quad (9)$$

$$ii2(i, j) = \sum_{i' \leq i, j' \leq j} p^2(i', j'), \quad (10)$$

其中, $p(i', j')$ 表示原图像中一点 (i', j') 的灰度

值, $p^2(i', j')$ 表示原图像中一点 (i', j') 灰度值的平方。

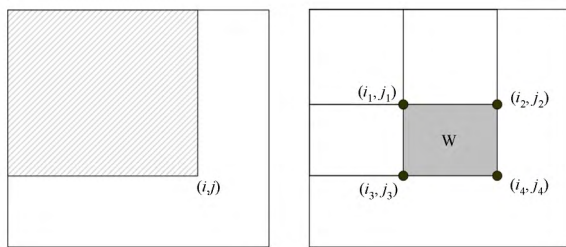


图 1 积分图像

Fig. 1 Integral image

$ii(i, j)$ 可用式(11)、(12)迭代计算得到:

$$S(i, j) = S(i, j-1) + p(i, j), \quad (11)$$

$$ii(i, j) = ii(i-1, j) + S(i, j). \quad (12)$$

灰度平方的积分图像 $ii2(i, j)$ 可用式(13)、(14)迭代计算得到:

$$S'(i, j) = S'(i, j-1) + p^2(i, j), \quad (13)$$

$$ii2(i, j) = ii2(i-1, j) + S'(i, j), \quad (14)$$

其中: $S(i, j)$ 和 $S'(i, j)$ 均表示一列的积分, 且 $S(i, -1) = 0, S'(-1, j) = 0, ii(-1, j) = ii2(-1, j) = 0$ 。

可以看出计算积分图像, 只需将原图像所有像素扫描一遍。如果基准图可以预先确定, 那么积分图像也是可以预先计算得到的, 景象匹配时可将其与基准图一起预先装载在嵌入式计算机的内存中。

3.2 积分图法快速计算基准子图能量

利用积分图像快速求取基准子图能量的算法, 称之为积分图法, 由 Lewis 于 1995 年提出^[10]。计算时主要用到了式(15)、(16)。如图 1 所示, 在求取图 1 中窗口 W 内的图像能量时, 不管窗口 W 的大小如何, 均可以用积分图像的 4 个相应点 (i_1, j_1) 、 (i_2, j_2) 、 (i_3, j_3) 、 (i_4, j_4) 快速计算得到, 即窗口 W 内像素的灰度或灰度平方和为:

$$S_f = ii(i_4, j_4) - ii(i_2, j_2) - ii(i_3, j_3) + ii(i_1, j_1), \quad (15)$$

$$S_{ff} = ii2(i_4, j_4) - ii2(i_2, j_2) - ii2(i_3, j_3) + ii2(i_1, j_1). \quad (16)$$

应用时, 需要将基准图的积分图像事先计算完成。那么, 在求取基准子图的能量运算就可转变为式(15)、(16)的 3+3 个加减法运算, 与传统 NCC 算法相比, 计算速率将得到大大提升。

3.3 经典积分图法主要问题分析

经典积分图法的计算过程灵活, 大大节省了运算时间。但是它最大的不足就是占用内存开销大。因为计算机的内存除了为必存的基准图预留空间外, 还需给基准图像素值的积分图像 ii 及像素平方和的积分图像 $ii2$ 分配相等的内存。从式(12)、(14)可知, 积分图像中的值随着行列的增加, 其值不断增大。假设一幅 1024×1024 尺寸, 具有 256 个灰度级的基准图, 考虑其中所有像素灰度值都为 255 的极限情况, 则积分图像 $ii2$ 中 $i = H-1, j = W-1$ 处像素值平方和的大小就为: $1024^2 \times 255^2 \approx 2^{10} \times 2^{10} \times 2^8 \times 2^8$, 占用内存 36 bit。那么, 用现有计算机中 4 B(32 bit) 的整型数就无法表示积分图像 $ii2$ 中的值, 而只能采用 8 B(64 bit) 的长整型数或双精度浮点数来表示。此时, 积分图像需要的内存量为图像本身的 2×8 倍。显然, 这样大的内存消耗在一些嵌入式系统或单片机上是难以承受的。

因此本文提出了 NCC 中快速计算基准子图能量的快速递推算法, 详细介绍如下。

4 快速递推算法

快速递推算法的主要思想是利用图像中相邻像素值的能量进行连续递推, 优势是在计算时可以不必要像积分图法那样给所有的图像能量都分配空间, 只需预留 1 行的像素空间便能完成整个能量计算过程, 即保证了计算速度, 又大大节省了内存占用。

4.1 基本原理

假设基准图 f 高 H 、宽 W , 实时图 g 高 M 、宽 N , 基准子图能量的定义如图 2 所示, (i, j) 处基准子图能量值 A 可定义为以 (i, j) 像素为右下角点, 高为 M , 宽为 N 区域内所有像素的灰度和或灰度平方之和。同理, 如图 2 所示, B 、 C 、 D 可分别定义为与 (i, j) 相邻的 3 个像素 $(i-1, j)$ 、 $(i, j-1)$ 和 $(i-1, j-1)$ 处的能量值。

基准子图能量 A 的计算方法如公式(17)所示。其中, B 、 C 、 D 可分别为与 (i, j) 相邻的 3 个像素 $(i-1, j)$ 、 $(i, j-1)$ 和 $(i-1, j-1)$ 处的基准子图能量值。 $f(i, j)$ 、 $f(i-M, j-N)$ 、 $f(i-M, j)$ 和 $f(i, j-N)$ 则为基准图 f 在该处的像素灰度值, M 、 N 为实时图 g 的高和宽。基于这些元素,

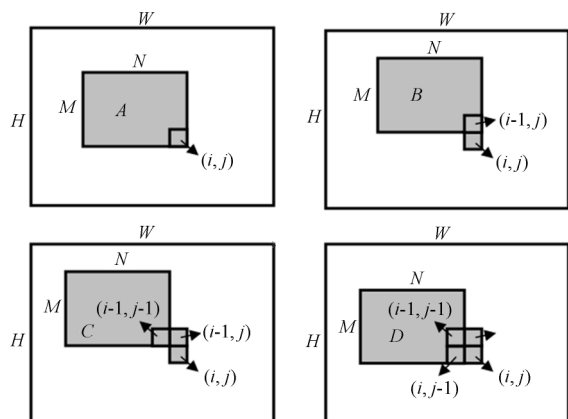


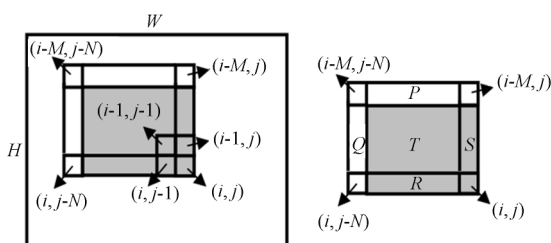
图 2 快速递推法基准子图能量定义

Fig. 2 Sub-image's energy in fast recurrence method

通过对基准图 f 进行递推遍历运算,就可以得出每个像素位置的基准子图能量。

$$A = B + D - C + f(i, j) + f(i - M, j - N) - f(i - M, j) - f(i, j - N). \quad (17)$$

A 处能量的具体计算推导过程如下。

图 3 计算 (i, j) 处基准子图能量的推导过程Fig. 3 Calculation of sub-image's energy in pixel (i, j)

如图 3 所示,图 3 右图中的灰色部分为要计算的基准子图能量 A , A 可由公式(18)计算得出:

$$A = T + R + S + f(i, j), \quad (18)$$

$$B = P + T + S + f(i - M, j), \quad (19)$$

$$C = P + Q + T + f(i - M, j - N), \quad (20)$$

$$D = Q + T + R + f(i, j - N). \quad (21)$$

计算 $B + D - C$ 可得:

$$B + D - C = T + R + S + f(i - M, j) + f(i, j - N) - f(i - M, j - N). \quad (22)$$

因此:

$$T + R + S = B + D - C - f(i - M, j) - f(i, j - N) + f(i - M, j - N). \quad (23)$$

将式(23)代入式(18)中,便可以得到公式(17),求出 A 的值。

因此,已知式(17),要求 (i, j) 处基准子图能量 A ,只需先求出与其相邻像素 $(i - 1, j)$ 、 $(i - 1,$

$j - 1)$ 和 $(i, j - 1)$ 处的能量值 B 、 C 、 D 即可。由于用到的只是相邻图像能量值,因此在计算分配内存的过程中,可以不必像积分图法那样给所有的图像能量都分配空间,而只需预留 1 行的空间便能完成整个能量计算的过程。这种求取基准子图能量的方法,称之为快速递推法,其最大优点就是占用内存少,速度快。下面将给出快速递推法的详细算法过程,如图 4 所示。

4.2 快速递推法计算基准子图能量

首先分配一个长度等于基准图 f 宽度 W 的数组 Row ,并将该数组内所有元素的值初始化为 0。在算法运行过程中需要用基准图 f 每 1 行像素所对应的基准子图能量值 S_j 不断地对 Row 进行更新。同时设置 3 个变量 $B = 0$ 、 $C = 0$ 、 $D = 0$,在计算基准图 f 中 (i, j) 处的基准子图能量时, B 、 C 、 D 将分别存储 $(i - 1, j)$ 、 $(i - 1, j - 1)$ 和 $(i, j - 1)$ 处的能量值。

将大小为 $M \times N$ 的窗口 g 从基准图 f 的 $(0, 0)$ 点处开始逐行遍历,至 $(H - 1, W - 1)$ 点结束,这期间将进行如图 4 所示的运算过程。

设相关矩阵为 R ,大小为 $(H - M + 1) \times (W - N + 1)$ 。

Step 1. 计算 $R(0, 0)$ 至 $R(0, W - N)$ 所需的图像能量。如图 4(a)、(b) 所示。

(1) 窗口 g 滑行至 $f(i, j)$ ($0 \leq i \leq M - 1, 0 \leq j \leq W - 1$) 处时,先将 $Row[j]$ 的值提取出赋给 B ,然后分两种情况更新 $Row[j]$ 的值。

如果 $j \leq N - 1$, 计算

$$Row[j] = B + D - C + f(i, j); \quad (24)$$

如果 $N \leq j \leq W - 1$, 计算

$$Row[j] = B + D - C + f(i, j) - f(i, j - N). \quad (25)$$

(2) 用 $Row[j]$ 的新值更新 D ,用 B 的值更新 C ,将 i, j 分别加 1,返回执行(1),直至 $i = M - 1, j = W - 1$ 时结束。

Step 2. 计算 $R(1, 0)$ 至 $R(H - M, W - N)$ 所需的图像能量。如图 4(c)、(d) 所示。

(1) 窗口 g 滑行至 $f(i, j)$ ($M \leq i \leq H - 1, 0 \leq j \leq W - 1$) 处时,先将 $Row[j]$ 的值提取出赋给 B ,然后也分两种情况更新 $Row[j]$ 的值。

如果 $j \leq N - 1$, 计算

$$Row[j] = B + D - C + f(i, j) - f(i - M, j); \quad (26)$$

如果 $N \leq j \leq W - 1$, 计算

$$Row[j] = B + D - C + f(i, j) - f(i - M, j) + f(i - M, j - N) - f(i, j - N). \quad (27)$$

(2) 用 $Row[j]$ 的新值更新 D ,用 B 的值更新 C ,将 i, j 分别加 1,返回执行(1),直至 $i = H - 1, j = W - 1$ 时结束。

以上只列出了求取式(5)的算法过程,求式

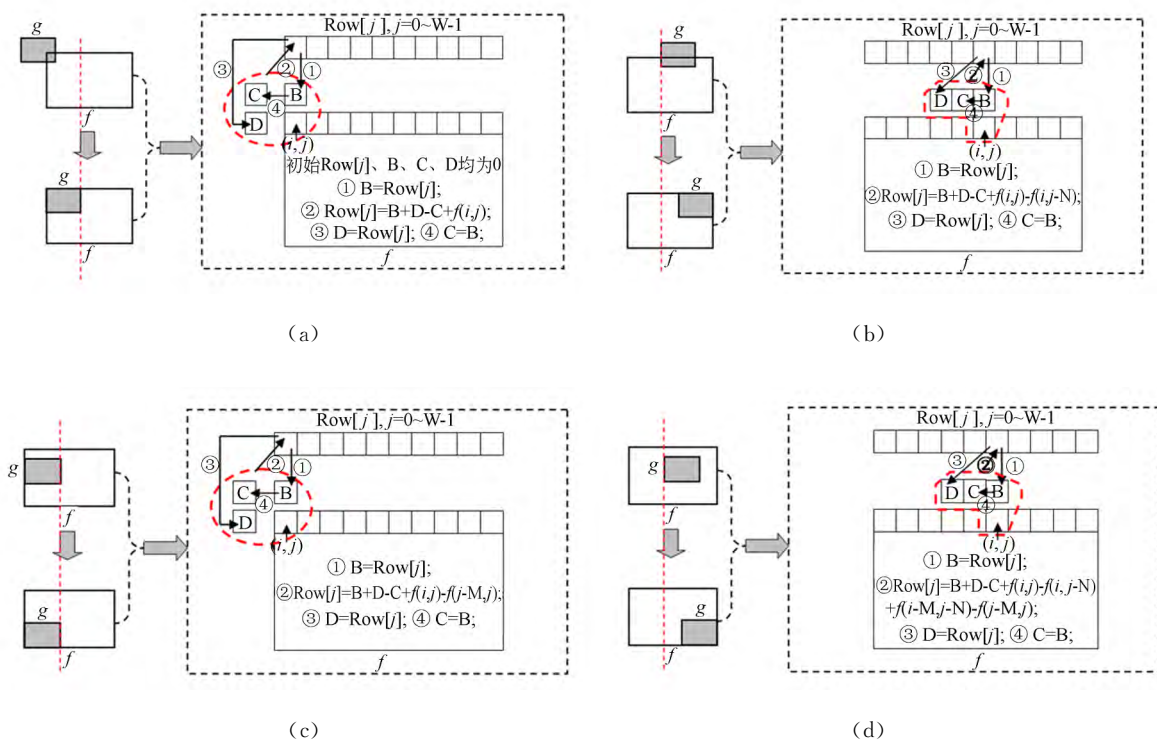


图 4 快速递推法计算图像能量的算法过程

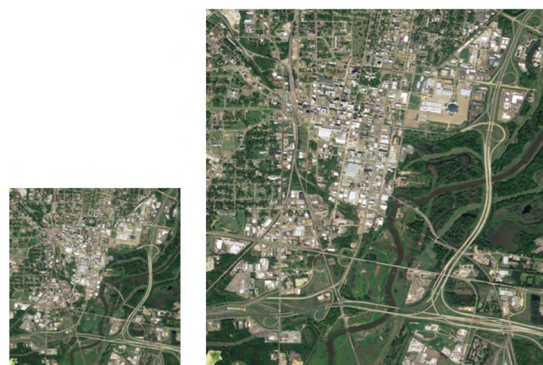
Fig. 4 Algorithm process of computing sub-image's energy by fast recurrence method

(4)的过程与之类似,将不再赘述。

以上算法执行过程中,快速递推法在计算一个基准子图能量时同经典积分图法一样只需要进行 6 个加减运算;在内存占用方面,快速递推法只需要给基准子图计算过程分配一行与基准图 f 的宽度 W 相同的存储空间,而为了保证计算速度,积分图法则需要分配 2 个与 f 尺寸 $H \times W$ 相同的空间用来分别存储灰度值和与灰度值平方和。可见,快速递推法在优化运算速度的前提下,有效降低了内存占用率。

5 实验与分析

为了验证本文快速递推算法的性能。本文在 CPU 为 Intel (R) XEON (TM) 2.4 GHz 2.37 Hz,内存 2 G 的 PC 机实验环境下,使用 Visual Studio 2008 的 C# 语言,编程实现了传统归一化互相关算法、基于经典积分图法的归一化互相关算法和基于本文快速递推法的归一化互相关算法。另外为了公平起见,编程实现时归一化互相关中的卷积运算与传统算法保持了一致,没有进行加速处理。

图 5 基准图,左图大小 256×256 ,右图 512×512 Fig. 5 Basic images, left image size 256×256 , right image size 512×512

在实验过程中首先比较了上述 3 种算法计算相关矩阵所花费的时间,然后比较了后两种算法计算相关矩阵时所占用的内存,最后分析讨论了将扩展后的快速递推法应用于跳跃式计算的情况。本文选用的测试基准图 f 分别为 256×256 、 512×512 的国外遥感卫星影像,如图 5 所示。

实时图则为分别从各自两幅图中截取的大小为 $32 \times 32 \sim 150 \times 150$ 的子图像。由于本文快速递推法和经典积分图法均是对传统归一化互相关

算法的优化加速,因此 3 种算法的匹配结果是一致的,如图 6 所示。



图 6 3 种归一化互相关算法的匹配结果示意,右图实时图尺寸为 150×150 ,左图为右图在 512×512 基准图上的匹配结果

Fig. 6 Matching result of three normalized cross-correlation algorithms, left image is real-time image which size is 256×256 , right image is result of left image matching with basic image which is size 512×512

5.1 时间花费对比试验

传统归一化互相关算法、基于经典积分图法的归一化互相关算法和基于本文快速递推法的归一化互相关算法的运算时间分别进行了测试。这里按照实际应用的情况,又将积分图法的计时方法按匹配方式分成两种。一是内含式匹配,即基准图尺寸大于实时图,该种情况下,由于基准图不变,它的积分图像可以事先算出,因此归一化互相关算法过程不需记录积分图像 ii 和 $ii2$ 计算时间;二是穿越式匹配,即基准图尺寸小于实时图,在这种情况下匹配过程就变为在实时图中寻找基准图,由于实时图是不断变化的,它的积分图像无法事先算出,因此归一化互相关算法就需要记录积分图像 ii 和 $ii2$ 的计算时间。详细的测试结果如表 1、2 所示。

由实验结果数据可以总结出以下结论:

(1) 基于快速递推法的归一化互相关算法与基于积分图法的归一化互相关算法,两者时间花费均较不做任何加速处理的传统归一化互相关算法相大大减少,因此两者都是归一化互相关的快速算法;

(2) 快速递推法花费的时间要比不记 ii 和

$ii2$ 的计算时间的积分图法略多,但是比记 ii 和 $ii2$ 的计算时间的积分图法少。

表 1 计算相关矩阵所花费的时间,基准图 f 大小为 256×256

Tab. 1 Calculation time of correlation matrix, size of basic image f is 256×256

实时图大小/pixel	原算法/ms	积分图法 (不记 ii 和 $ii2$ 的计算 时间) /ms	积分图法 (记 ii 和 $ii2$ 的 计算时间) /ms	快速递 推法 /ms
32×32	630.333	350.526	365.718	356.574
50×50	1232.693	709.840	730.213	713.026
64×64	1 770.413	985.766	994.135	989.484
72×72	2 065.552	1 135.894	1 145.940	1 139.996
90×90	2 593.192	1 454.175	1 471.212	1 461.079
120×120	4 009.575	1 744.706	1 759.837	1 745.595
150×150	5 855.010	1 851.028	1 879.011	1 860.783

表 2 计算相关矩阵所花费的时间,基准图 f 大小为 512×512

Tab. 2 Calculation time of correlation matrix, size of basic image f is 512×512

实时图大小/pixel	原算法 /ms	积分图法 (不记 ii 和 $ii2$ 的计算 时间) /ms	积分图法 (记 ii 和 $ii2$ 的 计算时间) /ms	快速递 推法 /ms
32×32	2 974.532	1 621.932	1 665.671	1 622.827
50×50	6 608.814	3 614.825	3 668.766	3 623.451
64×64	12 833.306	5 568.937	5 571.202	5 606.456
72×72	15 436.983	6 766.353	6 804.190	6 773.256
90×90	26 258.846	9 660.373	9 733.690	9 670.544
120×120	55 661.099	25 573.334	25 739.005	25 635.300
150×150	66 055.529	32 869.159	32 940.056	32 940.491

5.2 内存消耗对比试验

快速递推法比积分图法更节省内存开销,为了验证这一点,这里我们仍以一幅大小为 $1 024 \times 1 024$,灰度级为 256 的基准图为例进行说明。与积分图法不同的是,快速递推法中数组 Row 的数据类型选择与实时图像的大小有关,如果也考虑像素值都等于 255 的极限情况,那么 $i = H - 1, j = W - 1$ 处的像素值平方和就为 $M \times N \times 255^2$ 。若实时图高与宽均小于 256,如 $M \times N = 128 \times 128$,那么 $M \times N \times 255^2 = 128 \times 128 \times 255^2 \approx 2^{30}$,占用内存 30 bit,此时 Row 的数据类型就可定义为 4 B(32 bit)的长整型;若实时图高与宽均大于

256, 如 $M \times N = 512 \times 512$, 那么 $M \times N \times 255^2 = 512 \times 512 \times 255^2 \approx 2^34$, 占用内存 34 bit, 那么 Row 的数据类型就必须定义为 8 B(64 bit) 的长整型。

事实上, 无论数组 Row 的数据类型如何定义, 快速递推法的内存消耗都始终小于积分图法。如果实时图尺寸小于 256×256 , 那么快速递推法分配 $2 \times 4 \times W(B)$ 的内存给两个数组, 如果大于 256×256 , 则分配 $2 \times 8 \times W(B)$ 给数组。比较于积分图法始终需要分配 $2 \times 8 \times H \times W(B)$ 的内存给两个积分图像, 快速递推法的内存开销确实是大大减小了。

在 C# 语言编程实现时, 将基准图和实时图的数据类型定义为 8 位整型, 积分图像定义为 64 位长整型; 如果实时图小于 256×256 , 快速递推法中存储能量的数组就定义为 32 位整型, 否则为 64 位长整型; 最后相关矩阵的类型定义为 64 位双精度浮点型。因此基准图、实时图占用的内存空间分别为 $H \times W$ 、 $M \times N(B)$, 两幅积分图像为 $2 \times H \times W \times 8(B)$, 快速递推法两数组为 $2 \times W \times 4$ 或 $2 \times W \times 8(B)$, 相关矩阵为 $(H - M + 1) \times (W - N + 1) \times 8(B)$ 。

依据以上描述, 本文分别对基于积分图法和快速递推法的 NCC 算法所占用的内存进行了测试, 此外为了验证快速递推法内存消耗低的特点, 对于 512×512 的基准图, 又添加了实时图大于 256×256 的情况。测试结果如表 3、4 所示。

表 3 两种算法计算相关矩阵所占用的内存, 基准图 f 大小为 256×256

Tab. 3 Memory of computing correlation matrix, size of basic image f is 256×256

实时图大小 /pixel	积分图法 /B	快速递推法 /B
32×32	1 520 224	473 676
50×50	1 459 492	412 944
64×64	1 416 288	369 740
72×72	1 393 184	346 636
90×90	1 345 412	298 864
120×120	1 278 752	232 204
150×150	1 228 292	181 744

表 4 两种算法计算相关矩阵所占用的内存, 基准图 f 大小为 512×512

Tab. 4 Memory of computing correlation matrix, size of basic image f is 512×512

实时图大小 /pixel	积分图法 /B	快速递推法 /B
32×32	6 308 448	2 118 220
50×50	6 173 988	1 983 760
64×64	6 073 440	1 883 212
72×72	6 017 568	1 827 340
90×90	5 896 068	1 705 840
120×120	5 706 528	1 516 300
150×150	5 533 188	1 342 960
260×260	5 036 208	850 144
300×300	4 909 488	723 424
350×350	4 791 588	605 524

由表 3、4 可以看出, 无论实时图的尺寸是否大于 256×256 , 快速递推法计算相关矩阵时占用的内存总是比积分图法小很多。

5.3 灵活性对比试验

积分图法可以计算任意位置任意大小区域的像素灰度和, 而且只要 3 步加减运算, 既高效又灵活。相对而言, 快速递推法只有在连续求取基准子图能量的情况下, 才能保证与积分图法具有同样高的计算速度。因此在灵活性方面 4.2 快速递推法的算法过程不及积分图法。

为此, 为了提高快速递推法的灵活性, 快速递推法也可以做跳跃式地计算(跳跃式是指存在采样间隔的求和方式, 即每次隔几个像素, 算一次能量和)。只需要将原本表示 1 个像素值的 4 个角区域扩展为区域像素和即可, 如图 6 所示。但是这种扩展是存在一定限制的, 并且随着采样间隔的增加, 扩展后的快速递推法计算效率也会随之降低, 即快速递推法要想提高灵活性必须牺牲计算速度。

设采样间隔为 $s, s=0$ 的情况, 也就是连续计算的情况, 如图 6 中第 1 幅图所示, 此时快速递推法的效率最高, 与积分图法相当。随着 s 的不断增加, 图中 4 个角区域的像素个数也会按 $(s+1)^2$ 的顺序不断增多, 原本只表示单个像素的 4 个角区域, 现在要表示该区域的像素值和, 而这种像素值求和, 却只能按照一个一个像素依次相加的方法来做, 没有其它更好的办法, 因此随着 s 的增大, 快速递推法的运算效率会变得越来越低。如:

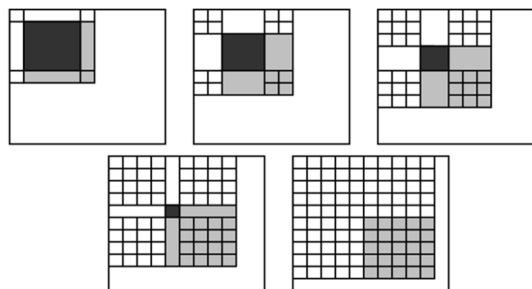


图 6 快速递推法跳跃式扩展求和

Fig. 6 Sum by jumping expansion of recurrence method

随着采样间隔按 0~4 的顺序增加,4 个角区域内的像素个数也按照 1、22、32、42 和 52 的顺序不断增多,图中黑色区域不断减小,最终将快速递推法变为最原始的求和方式。假设实时图高为 M ,宽为 N ,当横采样间隔 $s_1 \geq N-1$ 时,或者竖采样间隔 $s_2 \geq M-1$ 时,快速递推法将起不到任何快速计算的作用,因为此时的快速递推法已经变成了最原始的求和方式,如图 6 中第 5 幅图所示。

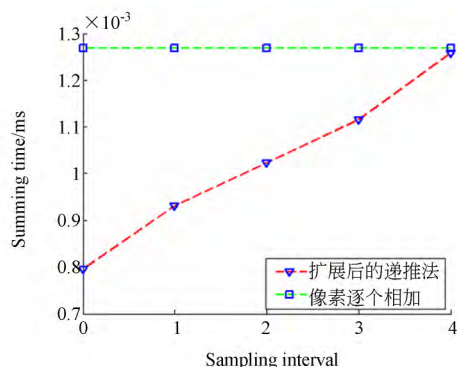


图 7 快速递推法跳跃式扩展后计算基准子图能量的时间

Fig. 7 Time of calculation for sub-image's energy by jumping expansion of recurrence method

参考文献:

- [1] TSAI D M, LIN C T. Fast normalized cross correlation for defect detection[J]. *Pattern Recognition*, 2003, 24, 2625-2631.
- [2] 徐伟, 陈彦彤, 朴永杰, 等. 基于吉林一号遥感图像的星载目标快速识别系统[J]. *光学精密工程*, 2017, 25(1): 255-262.
XU W, CHEN Y T, PIAO Y J, *et al.*. Target fast matching recognition of on-board system based on Jilin-1 satellite image[J]. *Optics and Precision En-*

我们对快速递推法进行跳跃式求和计算时的效率也进行了简单的实验测试,实验按照图 6 给出的过程实施。选取的实时图像大小为 5×5 ,采样间隔为 0~4,最后得到的图像能量计算时间如图 7 所示。

由图 7 可以看出,随着采样间隔的增加,图像能量的计算时间越来越长,到最后已经达到与最原始求和方法基本相等的时间,从而验证了以上讨论的结果。

6 结 论

快速递推法,同经典积分图法一样,同为归一化互相关中计算基准子图能量的快速算法,能够有效提升归一化互相关算法的计算速度,耗时均只为传统归一化互相关算法的 1/2。总的来说,经典积分图法计算过程快速灵活的优势,但是占用内存大。本文提出的快速递推法,其计算过程虽然不如积分图法灵活,但是占用内存小,二者各有优点。因此如果想要充分发挥它们的优势,就需要根据应用场合,恰当使用,扬长避短。通过本文的试验分析,现将积分图法和快速递推法的适用场合总结如下:

(1)快速递推法适合进行连续图像能量求取的应用场景,此时效率最高,例如飞行器嵌入式系统中的快速景象匹配应用。积分图法较为灵活,可进行跳跃式运算,更适合监控等 PC 环境下的目标识别应用;

(2)在内存容量较大的应用环境,如 PC 机环境,适合采用积分图法进行计算。而在一些对内存大小有限制的硬件环境,如 DSP 硬件、嵌入式系统等,则更适合采用快速递推法来计算。

gineering, 2017, 25(1): 255-262. (in Chinese)

- [3] 张浩鹏. 基于互相关计算加速器的实时目标跟踪系统设计与实现[D]. 浙江大学, 2014.
ZHANG H P. *Design and implementation of real-time target tracking system based on intercorrelation computing accelerators*[D]. Zhejiang University, 2014. (in Chinese)
- [4] HUI A J H, HANN C E, CHASE J G, *et al.*. Fast normalized cross correlation for motion tracking using basis functions[J]. *Computer Methods and Programs in Biomedicine*, 2006, 82(2): 144-156.

- [5] ARMELLONI E, GIOTTOLI C, FARINA A, *et al.*. Implementation of realtime partitioned convolution on a DSP board[C]. 2003 *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, 2003, 10, 19-22.
- [6] 田立, 周付根, 孟僊, 等. 互相关跟踪算法的多核 DSP 快速实现[J]. 高技术通讯, 2013, 23(12): 1248-1253.
- TIAN L, ZHOU F G, MENG S, *et al.*. Fast implementation of multi-core DSP for intercorrelation tracking algorithm[J]. *High-tech Communications*, 2013, 23(12): 1248-1253. (in Chinese)
- [7] 刘毅飞, 张旭明, 丁明跃. 归一化互相关灰度图像匹配的多核信号处理器实现[J]. 计算机应用, 2011, 31(12): 3334-3336.
- LIU Y F, ZHANG X M, DING M Y. Implementation of a Multicore Signal Processor for Normalized Intercorrelation Grayscale Image Matching [J]. *Computer Application*, 2011, 31(12): 3334-3336. (in Chinese)
- [8] JAE C Y, HAN T Y. Fast normalized cross-correlation[J]. *Circuits, Systems and Signal Processing*, 2009, 28 :819-843.
- [9] ARIE N, ALLEN T. A new distance measure based on generalized Image Normalized Cross-Correlation for robust video tracking and image recognition[J]. *Pattern Recognition Letters*, 2013, 34(3).
- [10] LEWIS J P. Fast normalized cross-correlation[J]. *in Proceedings of Vision Interface*, 1995, 120-123.
- [11] FRANKLIN C CROW. Summed-area tables for texture mapping[J]. *Computer Graphics*, 1984, 18(3).
- [12] PAUL VIOLA, MICHEAL J JONES. Robust real-time face detection[J]. *International Journal of Computer Vision*, 2004, 57(2): 137 - 154.
- [13] 卢宇, 卢荣胜, 郭广平, 等. 基于快速归一化算法的拉伸距离动态跟踪测量[J]. 传感技术学报, 2016, 29(9): 1328-1334.
- LU Y, LU R S, GUO G P, *et al.*. Dynamic tracking measurement of stretching distance based on fast normalization algorithm[J]. *Journal of sensing technology*, 2016, 29(9): 1328-1334. (in Chinese)
- [14] 陈冰倩. 基于 CPU 的归一化互相关算法波前斜率技术研究[D]. 中国科学院研究生院(光电技术研究所), 2016.
- CHEN B Q. *Research on wavefront slope based on normalized cross correlation algorithm based on CPU*[D]. Graduate University of Chinese Academy of Sciences (Institute of Optoelectronic Technology), 2016. (in Chinese)

作者简介:



韩冰(1982—),女,山东兖州人,博士,助理研究员,2007年、2011年于北京理工大学分别获得硕士、博士学位,主要从事遥感图像处理、目标自动识别与跟踪方向的研究。E-mail: 334679010@qq.com.



牟忠锋(1975—),男,吉林长春人,硕士,副研究员,2003年于北京师范大学获得硕士学位,主要从事遥感图像处理与应用方向的研究。E-mail: 13911799042@qq.com.



乐小峰(1981—),男,四川乐山人,硕士,助理研究员,2006年于北京理工大学获得硕士学位,主要从事遥感图像处理与应用、海量遥感影像数据管理等方向的研究。E-mail: 34828963@qq.com