

可重叠矩形多值图像表示及其上的几何矩生成^{*})

黄巍 陈传波 郑运平 吴雪丽

(华中科技大学计算机科学与技术学院 武汉 430074)

摘要 为了支持快速的多值图像运算,提出了一种无损多值图像表示方法,称为可重叠矩形多值图像表示(Overlapping Rectangle Multi-valued Image Representation,ORMIR)。ORMIR采用递归方式将一幅多值图像分割为具有不同基础颜色的可重叠矩形,并使用孩子兄弟树来组织这些矩形,通过弱化二值图像块表示中同一矩形所覆盖的所有像素必须具有相同颜色的约束,ORMIR能够使用较少的矩形无损地表示一幅多值图像,因而基于ORMIR的多值图像运算能够被快速实现。基于ORMIR,提出了一个多值图像几何矩生成算法,该算法首先生成多个仅包含一个矩形区域的二值图像的几何矩,然后将这些几何矩加权求和得到原始多值图像的几何矩。试验结果表明,基于ORMIR的几何矩生成算法能够以每秒50帧以上的速度计算8比特位深的512×512的灰度图像直到3+3阶的几何矩,从而满足实时应用的需要。

关键词 数据结构,多值图像表示,图像块表示,可重叠矩形,几何矩

Overlapping Rectangle Multi-valued Image Representation and its Application in Geometric Moments Generation

HUANG Wei CHEN Chuan-bo ZHENG Yun-ping WU Xue-li

(School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China)

Abstract A lossless multi-valued image representation, referred as to Overlapping Rectangle Multi-valued Image Representation (ORMIR) was presented in this paper for the fast generation of geometric moments of multi-valued images. ORMIR divides a multi-valued image into some overlapping rectangles with some basic colors and settles these rectangles in a child-sibling tree. By impairing the constraint in Image Block Representation where all the pixels covered with a distinct rectangle must have the same color, ORMIR has reduced significantly the number of the rectangles required to represent a multi-valued image, hence is able to speed up image operations such as geometric moment generation. Based on ORMIR, a fast algorithm to generate geometric moments of multi-valued images was presented, which firstly generates all the geometric moments of a corresponding set of binary images with only one rectangular region and then sums up these moments by some weight in order to obtain the geometric moments of the original multi-valued image. The experimental results show that the algorithm is capable of generating the geometric moments up to order 3+3 of a 512×512 8 bits gray image beyond a speed of 50 frames per second and therefore, it is competent to geometric moments generation in real-time applications.

Keywords Data structure, Multi-valued image representation, Image block representation, Overlapping rectangle, Geometric moments

1 引言

图像表示在计算机视觉、机器人、计算机图形学、图像处理、模式识别等领域内有重要的研究意义。通常说来,研究图像表示的目的无外乎以下三种:其一是支持特定的设备,比如行程编码^[1,2]特别适合光栅设备的输入输出;其二是用于图像压缩,提高图像的传输和存储效率,这类图像表示方法虽然能够得到很高的压缩比,但是一般不能直接进行图像运算,而且为了追求高压缩比还会有意损失部分图像信息,JPEG图像格式是这类图像表示方法的典型代表;其三是支持特定的图像运算,比如边界链码表示^[3,4],四元树数据结构^[5-7]和图像块表示^[8-11],由于这类表示方法能够在一个表示单位(比如

一个四元树节点或一个图像块)中编码多个像素的信息,因而某些图像运算(比如集合运算)能够快速实现。本文的主要目的在于第三种,即提供一种高效的多值图像表示方法以支持快速的多值图像运算。

虽然行程编码、边界链码、四元树数据结构和图像块表示的编码方式和编码效率各不相同,但是它们都具有一个共同的特点,即将满足特定条件的具有相同颜色的像素编码到一个表示单位中。由于二值图像只具有黑白两种颜色,因此这些表示方法对二值图像具有较高的表示效率。然而多值图像由于其像素位深较大,很难有大面积的像素具有相同的颜色,导致这些表示方法需要使用较多的表示单位来无损地表示一幅多值图像,结果基于这些表示方法的多值图像运算通常需

^{*} 基金项目:国家高技术研究发展计划(2006AA04Z211)。黄巍 博士生,研究方向为图像处理与模式识别;陈传波 博士,教授,博士生导师,研究方向为图像处理与模式识别;郑运平 博士研究生,研究方向为模式识别与多媒体信息处理;吴雪丽 博士研究生,研究方向为图像处理与模式识别。

要处理大量的表示单位,因而耗费相对较长的运算时间。

自从 Hu 提出矩的理论以来^[12],矩和矩不变量作为模式描述子,在模式识别和图像分析中得到了广泛的应用。几何矩由于具有具体的物理意义,得到了特别的关注。 $M \times N$ 的数字图像的 $p+q$ 阶几何矩 m_{pq} 由下式确定:

$$m_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} x^p y^q f(x, y) \quad (1)$$

其中 $f(x, y)$ 将像素坐标 (x, y) 映射为该像素的颜色值。显然,由式(1)直接生成一个 $M \times N$ 的数字图像的一个几何矩需要 $2MN$ 次幂运算, $2MN$ 次乘法运算和 $MN-1$ 次加法运算。过多耗时的幂运算和乘法运算限制了几何矩的应用,特别是在实时环境中的应用。近年来,大量的研究工作致力于提高几何矩生成的速度^[11,13-20]。由于早期计算机的计算能力有限,只能通过设计专用的硬件来实现快速的几何矩生成,这类研究通常使用大量的加法运算来减少耗时的幂运算和乘法运算,并使算法的 VLSI 实现使用最少的晶体管数量和最小的芯片面积。借助于单芯片实现, Hatamian 的算法能够实时地(每秒 30 帧)生成 8 比特位深的 512×512 的灰度图像直到 3 阶(即 $p+q \leq 3$) 的几何矩^[13]。Delta 方法利用求和变量的可分离特性,将二维空间上的求和转化为多个一维求和,根据 Delta 方法, $M \times N$ 的数字图像的 $p+q$ 阶几何矩的计算公式为:

$$m_{pq} = \sum_{y=0}^{N-1} y^q \sum_{x=0}^{M-1} x^p f(x, y) \quad (2)$$

Delta 方法特别适合以行程编码表示的二值图像,当图像的行程数为 Nr 时,可以使用 $6Nr$ 次乘法和 $17Nr$ 次加法计算凸对象的直到 3 阶的几何矩^[14]。格林理论将区域上的积分转化为区域边界上的积分,利用离散化的格林理论, Yang 提出了一个快速精确的几何矩生成算法^[15],然而,由于在多值图像中区域的边界不能完全确定区域的信息,因此采用离散格林理论的几何矩生成方法通常并不适合多值图像几何矩的生成。图像变换也常常被采用以减少几何矩生成的运算次数^[16,17],为了计算 $N \times N$ 数字图像直到 k 阶的几何矩,文献^[16]将二维几何矩的计算转化为 $k+1$ 个一维几何矩的计算,然后通过线性变换,求得原始图像直到 k 阶的几何矩。文献^[17]采用了另外一种变换方法,称为 Biaxial 变换,它将位深为 m 的 $N \times N$ 数字图像转换为一个位深为 $m+2$ 的 $(N/2) \times (N/2)$ 数字图像,并将原始图像的几何矩生成转化为对变换后图像的几何矩生成,通过这种转换,与几何矩生成直接相关的算术运算减少为原来的 $1/4$ 。采用变换方式生成图像几何矩方法的不足之处在于变换过程本身需要大量的加法运算。王冰等人将二个数组对应元素的乘积之和转换为一个数组的差分与另一个数组累计的乘积之和,通过这种方法,他们将灰度图像质心计算所需的乘法运算数量减少到 3 次^[17],然后他们的方法不适合 2 阶以上的几何矩生成。也有大量研究致力于开发多值图像几何矩的并行生成算法^[18],通常这些算法的时间复杂度为 $O(1)$,但是这些算法不适合在序列计算机上实现。

本文提出了一种支持快速运算的多值图像无损表示方法,称为可重叠矩形多值图像表示(Overlapping Rectangle Multi-valued Image Representation, ORMIR), ORMIR 能够在—个矩形块中表示原始图像的多个像素,甚至这些像素在 4 邻接和 8 邻接意义上都不连通。由于 ORMIR 极大地减少了表示一幅多值图像所需的矩形数量,因此,基于 ORMIR,多值图像的几何矩能够快速生成。试验结果表明,纯粹软件实

现的基于 ORMIR 的几何矩生成算法能够在个人计算机上以每秒 50 帧以上的速度生成 8 比特位深的 512×512 数字图像直到 3+3 阶的几何矩。

2 可重叠矩形多值图像表示

可重叠矩形多值图像表示(Overlapping Rectangle Multi-valued Image Representation, ORMIR)是二值图像块表示(Image Block Representation, IBR)方法^[8-11]在多值图像上的扩展。IBR 将一个二值图像分割成多个不重叠的矩形,要求被同一个矩形覆盖的所有像素具有相同的颜色。由于多值图像的颜色非常丰富,直接地将 IBR 应用于多值图像,将导致块的面积过小,因而表示效率不高。为了提高多值图像表示效率,直观上应该尽可能地增加矩形的面积,减少矩形块的数量。

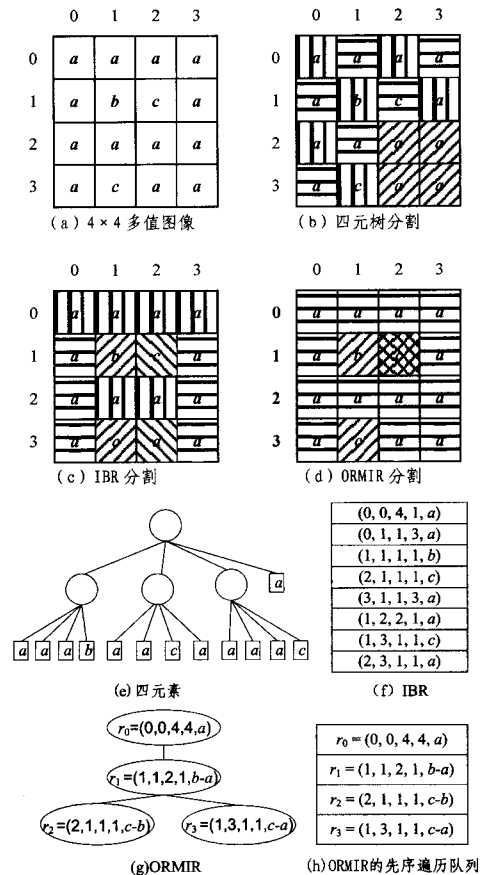


图 1 多值图像的不同分割方法及对应的表示

在图 1(a) 所示的一个 4×4 的多值图像中,除了像素(1, 1)处的颜色值为 b , 像素(2, 1)和(1, 3)处的颜色值为 c 以外,其他所有像素的颜色值都为 a , 不失一般性,假设 $a < b < c$ 。为了表示这幅多值图像,图 1(b) 所示的四元树分割方法将该图像分割为 13 个正方形,其对应的四元树如图 1(e) 所示,即使采用线性四元树编码该图像,仍需要 13 个线性四元树节点。图 1(c) 所示的 IBR 分割方法将该图像分割为 8 个矩形,其对应的 IBR 如图 1(f) 所示,IBR 采用一个队列来表示原始图像,队列中的任一元素是一个用于表示矩形的 5 元组 (x, y, l, w, col) ,其中 x, y, l, w 和 col 分别表示矩形起点(即矩形最左上像素)横坐标,矩形起点纵坐标,矩形的长,矩形的宽和矩形覆盖的所有像素的颜色值。显然,IBR 分割由于取消了四元树分割所要求的两个约束条件,即块必须为正方形和块

的起点必须在 $(2^i, 2^i)$ 处,因而分割一幅多值图像所得到的矩形块数量少于同一图像的四元树分割所得到的正方形块数。

为了进一步减少表示一幅多值图像的矩形块的数量,可以弱化 IBR 中规定的同一矩形块中所有像素的颜色必须相同的约束,而为每块规定一个基础颜色值,并规定包含在一个矩形块中的所有像素的颜色值必须大于或等于该块的基础颜色值。基础颜色值的选择显著地影响编码效率,但是为了简化讨论,本文简单地选择每个矩形块的起点的颜色值作为该块的基础颜色值。事实上,注意到为了表示图 1(a)所示的原始图像,只需要指出整个图像的基础颜色值为 a ,并标明所有颜色值不为 a 的像素的位置及其颜色值与基础颜色值 a 的差值即可。根据这个思想,图 1(a)所示的原始图像被看作一个基础颜色值为 a 的矩形 $r_0 = (0, 0, 4, 4, a)$,为了标识 r_0 中颜色值不为 a 的像素,需要指出 r_0 中还包含 2 个基础颜色值不为 a 的矩形块:第一块是由像素 $(1, 1)$ 和像素 $(2, 1)$ 所构成的一个长为 2, 宽为 1 的矩形块 r_1 , 它的基础颜色值是 b , 与 r_0 的基础颜色值 a 的差值为 $b-a$, 所以 $r_1 = (1, 1, 2, 1, b-a)$; 第二块是由像素 $(1, 3)$ 所构成的一个长宽都为 1 的矩形块 r_3 , 它的基础颜色值是 c , 与 r_0 的基础颜色值 a 的差值为 $c-a$, 所以 $r_3 = (1, 3, 1, 1, c-a)$ 。同样地,由于 r_1 中仍包含颜色值不为基础颜色值 b 的像素 $(2, 1)$, 因此需要使用另外一个矩形块 $r_2 = (2, 1, 1, 1, c-b)$ 来标识它。这样就形成了一个如图 1(d)所示的递归分割,从图 1(d)中可以看出,使用四元树分解和 IBR 分解都无法包含在一块中的像素,比如像素 $(0, 0)$ 和 $(3, 3)$, 现在都包含在同一个矩形块 r_0 中,从而达到了减少表示一幅多值图像所需的矩形块数量的目的。由于在这种分割方法中,一些矩形块,比如 r_1 和 r_2 , 在空间位置上存在重叠,因此称这种表示方法为可重叠矩形多值图像表示(Overlapping Rectangle Multi-valued Image Representation, ORMIR)。相应地,在 ORMIR 中使用的带有基础颜色值差值信息的矩形块称为 ORMIR 矩形块。ORMIR 分割所得到的所有矩形构成了一个出度不固定的树形结构,为了方便,可以使用如图 1(g)所示的孩子兄弟树来表示这个出度不固定的树形结构。为了从 ORMIR 中解码原始图像,只需要先序遍历如图 1(g)所示的树形结构,获取每个树节点对应的 ORMIR 矩形 (x, y, l, w, col) , 并将其覆盖的像素的颜色值加 col 即可。进一步,为了节省存储空间,还可以消除图 1(g)所示的 ORMIR 中的指针,而以 ORMIR 的先序遍历队列来表示 ORMIR, 如图 1(h) 所示。

ORMIR 编码算法的递归描述如下:

算法 1 ORMIR 编码算法

输入: $M \times N$ 的多值图像 G , ORMIR 矩形 $r = (x, y, l, w, col)$ 。

输出: r 所覆盖的 G 的子区域的 ORMIR 的先序遍历队列 O 。

Step 1 初始化 O 为一个空队列;

Step 2 初始化候选起点横坐标 sp_x 为 x , 纵坐标 sp_y 为 y ;

Step 3 从像素 (sp_x, sp_y) 开始, 在矩形 r 所覆盖的图像 G 的子区域内, 以光栅扫描顺序寻找第一个未被 O 中任何矩形所覆盖的, 且颜色值大于 col 的像素 p , 若这样的像素 p 没有找到, 则跳转到 Step 8;

Step 4 设 p 的横坐标, 纵坐标和颜色值分别为 p_x , p_y 和 p_col , 以 p 为起点, 匹配出所有的边与图像 G 的边界平行的矩形, 要求被这些矩形覆盖的像素未被 O 中任何矩形所覆盖且这些像素的颜色值大于或等于 p_col ;

Step 5 在 Step 4 得到的所有矩形中, 选择面积最大的一个矩形, 设其长和宽分别为 l' 和 w' , 记矩形 r' 为 $(p_x, p_y, l', w', p_col)$, 将 r' 的以基础颜色值差值方式表达的 ORMIR 矩形 5 元组 $(p_x, p_y, l',$

$w', p_col)$ 添加到 O 的末尾;

Step 6 若 r' 所覆盖的像素中包含颜色值大于 p_col 的像素, 则以 G 和 r' 为参数, 递归调用本算法, 获得其返回值 O' , 并将 O' 中的元素按其本身的顺序添加到 O 的末尾;

Step 7 以光栅扫描顺序在 r 所覆盖的 G 的子区域上查找 p 的下一个像素 p' , 如果这样的 p' 不存在, 则跳转到 Step 8, 否则令 sp_x 和 sp_y 分别为 p' 的横坐标和纵坐标, 跳转到 Step 3;

Step 8 返回 O 。

为了获得 $M \times N$ 的多值图像 G 的 ORMIR 的先序遍历队列 O , 只需将 $(0, 0, M, N, 0)$ 加入到空队列 O 中作为 O 的对首元素, 并使用参数 G 和 $(0, 0, M, N, 0)$ 调用 ORMIR 编码算法, 将返回的队列中各元素按其本身的顺序添加到 O 的末尾即可。

根据 ORMIR 编码算法, 不难看出, 当将该算法应用到二值图像上时, 编码结果与 IBR 一致, 因此, 可以将 IBR 看作 ORMIR 在二值图像表示上的特例。

3 基于 ORMIR 的几何矩生成

开发 ORMIR 的目的是支持快速的多值图像运算, 本节将 ORMIR 应用到多值图像的几何矩生成中, 得到基于 ORMIR 的几何矩生成算法。

根据式(1), 图 1(a)所示多值图像的 $p+q$ 阶几何矩 m_{pq} 为:

$$m_{pq} = \sum_{x=0}^3 \sum_{y=0}^3 x^p y^q f(x, y) \quad (3)$$

将式(3)展开, 并交换加法运算的次序:

$$\begin{aligned} m_{pq} &= \sum_{y=0}^3 a 0^p y^q + a 1^p 0^q + b 1^p 1^q + a 1^p 2^q + c 1^p 3^q + a 2^p 0^q + \\ &\quad c 2^p 1^q + a 2^p 2^q + a 2^p 3^q + \sum_{y=0}^3 a 3^p y^q \\ &= \sum_{y=0}^3 a 0^p y^q + \sum_{y=0}^3 a 3^p y^q + a 1^p 0^q + [a + (b-a)] 1^p 1^q + \\ &\quad a 1^p 2^q + [a + (c-a)] 1^p 3^q + a 2^p 0^q + [a + (b-a) + \\ &\quad (c-b)] 2^p 1^q + a 2^p 2^q + a 2^p 3^q \\ &= (\sum_{y=0}^3 a 0^p y^q + \sum_{y=0}^3 a 3^p y^q + a 1^p 0^q + a 1^p 1^q + a 1^p 2^q + \\ &\quad a 1^p 3^q + a 2^p 0^q + a 2^p 1^q + a 2^p 2^q + a 2^p 3^q) + (b-a) \\ &\quad (1^p 1^q + 2^p 1^q) + (c-b) 2^p 1^q + (c-a) 1^p 3^q \\ &= a \sum_{x=0}^3 \sum_{y=0}^3 x^p y^q + (b-a) \sum_{x=1}^2 \sum_{y=1}^1 x^p y^q + (c-b) \sum_{x=2}^2 \sum_{y=1}^1 x^p y^q + \\ &\quad (c-a) \sum_{x=1}^1 \sum_{y=3}^3 x^p y^q \end{aligned} \quad (4)$$

考虑到在二值图像上, 由式(1)确定的 $M \times N$ 的数字图像的 $p+q$ 阶几何矩可由下式确定:

$$\mu_{pq} = \sum_x \sum_y x^p y^q \forall x, y: f(x, y) = 1 \quad (5)$$

可以由图 1(a)所示的原始图像派生出与原始图像具有相同分辨率的四个二值图像 B_0, B_1, B_2 和 B_3 , 其中, B_i 对应原始图像的 ORMIR 的第 i 个矩形 r_i , 在 B_i 中, 除了 r_i 覆盖的像素全为 1 以外, 其他像素都为 0, 称这些二值图像为原始图像的伴随图像, 显然, 所有的伴随图像都是只包含唯一矩形区域的二值图像。由式(5)可知, 伴随图像 B_i 的 $p+q$ 阶几何矩 μ_{pq}^i 为:

$$\mu_{pq}^i = \sum_x \sum_y x^p y^q (x, y) \in r_i \quad (6)$$

比较式(6)和式(4), 可得:

$$m_{pq} = a \mu_{pq}^0 + (b-a) \mu_{pq}^1 + (c-b) \mu_{pq}^2 + (c-a) \mu_{pq}^3 \quad (7)$$

由式(7)可知, 图 1(a)所示原始图像的 $p+q$ 阶几何矩等于其所有伴随图像的 $p+q$ 阶几何矩的加权和, 而权重即为伴随

图像对应的 ORMIR 矩形块以增量方式表达的基础颜色值。

对于任意 $M \times N$ 的数字图像 G , 设其 ORMIR 矩形块数为 K , 根据 ORMIR 矩形块的定义, 第 i 个 ORMIR 矩形块 r_i 的以增量方式表达的基础颜色值为 $r_i.col$, 则可由式(7)推广得到 G 的 $p+q$ 阶几何矩 m_{pq} 的计算公式如下:

$$m_{pq} = \sum_{i=0}^{K-1} r_i.col \times \mu_{pq}^i \quad (8)$$

可以使用任何二值图像几何矩生成算法生成伴随图形 B_i 的 $p+q$ 阶几何矩 μ_{pq}^i , 然而, Iraklis 等人提出的算法特别适合仅包含一个矩形区域的二值图像的几何矩生成^[11]。令 $r_i.x, r_i.y, r_i.l$ 和 $r_i.w$ 分别是 ORMIR 矩形块 r_i 的起点横坐标, 起点纵坐标, 长和宽, 则根据 Iraklis 的方法并利用求和变量的可分离特性, 由式(6)确定的伴随图像 B_i 的 $p+q$ 阶几何矩 μ_{pq}^i 可进一步变为:

$$\mu_{pq}^i = \left(\sum_{\mu=r_i.x}^{r_i.x+r_i.l-1} u^p \right) \left(\sum_{v=r_i.y}^{r_i.y+r_i.w-1} v^q \right) \quad (9)$$

式(9)说明, 为了计算 μ_{pq}^i , 只需要计算:

$$\begin{aligned} S_{r_i.x, r_i.x+r_i.l-1}^p &= \sum_{u=r_i.x}^{r_i.x+r_i.l-1} u^p, S_{r_i.y, r_i.y+r_i.w-1}^q \\ &= \sum_{v=r_i.y}^{r_i.y+r_i.w-1} v^q \end{aligned} \quad (10)$$

利用解析公式:

$$\begin{aligned} S_{1,n}^1 &= \frac{n(n+1)}{2}, S_{1,n}^2 = \frac{n(n+1)(2n+1)}{6} \\ S_{1,n}^3 &= \frac{n^2(n+1)^2}{4}, S_{1,n}^4 = \frac{n(n+1)(2n+1)(3n^2+3n+1)}{30} \end{aligned} \quad (11)$$

以及当指数 t 大于 4 时的递推公式:

$$C_{t+1}^1 S_{1,n}^t + C_{t+1}^2 S_{1,n}^{t-1} + \dots + C_{t+1}^t S_{1,n}^1 = (n+1)^{t+1} - (n+1) \quad (12)$$

可以使用一个解析公式计算多个幂运算的和, 从而实现仅包含一个矩形区域的二值图像的几何矩的快速生成。

综上所述, 为了计算由 K 个 ORMIR 矩形块表示的 $M \times N$ 的数字图像 G 的 $p+q$ 阶几何矩 m_{pq} , 首先利用 Iraklis 的算法, 即式(9)-(12), 生成 G 的所有伴随图像 B_0, B_1, \dots, B_{K-1} 的 $p+q$ 阶几何矩 $\mu_{pq}^0, \mu_{pq}^1, \dots, \mu_{pq}^{K-1}$, 然后利用式(8)将所有的 μ_{pq}^i 加权求和即可。

表 1 直接法, Delta 算法以及基于 ORMIR 的算法生成 $M \times N$ 的数字图像直到 $(L-1)+(L-1)$ 阶的几何矩所需计算次数比较

方法	幂运算	乘法运算	加法运算
直接法	$2L^2MN$	$2L^2MN$	$L^2(MN-1)$
Delta 法	$L^2N(M+1)$	$L^2N(M+1)$	$L^2(MN-1)$
基于 ORMIR 的算法	$4LK$	$(3L^2-L)K$	$2L^2K-LK-L^2$

注: K 表示图像的 ORMIR 矩形块的数量

文献[11]已证明, 使用式(9)-(12), 生成仅包含一个矩形区域的二值图像直到 $(L-1)+(L-1)$ 阶的几何矩, 需要 $4L$ 次幂运算, $2L^2-L$ 次乘法运算和 L^2-L 次加法运算, 所以生成 K 个伴随图像直到 $(L-1)+(L-1)$ 阶的几何矩共需要 $4LK$ 次幂运算, $(2L^2-L)K$ 次乘法运算和 $(L^2-L)K$ 次加法运算。另外, 使用式(8)加权求和这些伴随图像的几何矩需要额外的 L^2K 次乘法和 $L^2(K-1)$ 次加法, 因此, 基于 ORMIR 的几何矩生成算法生成直到 $(L-1)+(L-1)$ 阶的几何矩需要的幂运算, 乘法运算和加法运算次数分别为 $4LK, (3L^2-L)K$ 和 $2L^2K-LK-L^2$ 。

表 1 列出了使用直接法、Delta 方法以及基于 ORMIR 的几何矩生成算法生成 $M \times N$ 的数字图像直到 $(L-1)+(L-1)$ 阶的几何矩所需计算次数比较, 表中 K 表示图像的 ORMIR 矩形块的个数。在三种运算中, 幂运算是最耗时的, 乘法运算其次, 而加法运算最快, 从表中数据来看, 基于 ORMIR 的算法需要的幂运算最少。为了便于量化比较, 将一次幂运算看作一次乘法运算, 并且不考虑加法运算的影响, 则直接法、Delta 法和基于 ORMIR 的几何矩生成算法生成 $M \times N$ 的数字图像直到 $(L-1)+(L-1)$ 阶的几何矩所需乘法运算次数分别为 $4L^2MN, 2L^2N(M+1)$ 和 $3(L^2+L)K$ 次。记方法 B 相对方法 A 的理论加速因子 $\eta_{A|B}$ 为:

$$\eta_{A|B} = \frac{\text{方法 A 的乘法次数}}{\text{方法 B 的乘法次数}} \quad (13)$$

则基于 ORMIR 的几何矩生成算法相对直接法和 Delta 方法的理论加速因子 $\eta_{\text{Dir}|ORMIR}$ 和 $\eta_{\text{Delta}|ORMIR}$ 分别为:

$$\eta_{\text{Dir}|ORMIR} = \frac{4L^2MN}{3(L^2+L)K} \quad (14)$$

$$\eta_{\text{Delta}|ORMIR} = \frac{2L^2N(M+1)}{3(L^2+L)K} \quad (15)$$

从式(14)和式(15)可以看出, 基于 ORMIR 的几何矩生成算法相对其他方法的理论加速因子反比于表示图像的 ORMIR 矩形块个数。当生成 512×512 的数字图像直到 3+3 阶的几何矩时, 只有 $K < 279620$ 时, 才有 $\eta_{\text{Dir}|ORMIR} > 1$, 而实际上, 由于 512×512 的数字图像最多有 262144 个像素, 因此 $K < 279620$ 肯定成立, 则基于 ORMIR 的几何矩生成算法, 在生成 512×512 的数字图像直到 3+3 阶的几何矩时, 始终比直接算法快。同样, 如果 $K < 140083$, 则基于 ORMIR 的几何矩生成算法, 在生成 512×512 的数字图像直到 3+3 阶的几何矩时, 比 Delta 方法快。由于 ORMIR 具有较高的编码效率, 通常, 对于非特殊构造的图像, $K < 140083$ 是成立的, 因此, 基于 ORMIR 的几何矩生成算法, 在生成 512×512 的非特殊构造的数字图像直到 3+3 阶的几何矩时, 比 Delta 方法快。

4 试验结果

实验环境为 Intel Pentium4 1.8GHz 处理器, 512MB 内存、Windows XP Professional with SP2 操作系统, 使用 Visual Studio 2005 开发环境和 VC7.0 程序设计语言实现了基于 ORMIR 的多值图像几何矩生成算法和作为对比的直接计算算法, Delta 算法以及基于 Biaxial 变换的算法。图 2 是实验使用的 4 幅多值医学图像, 它们的分辨率都为 9, 位深都为 8, 图 2(a)和图 2(b)是两幅计算机断层扫描图像, 图 2(c)和图 2(d)是两幅 X 光照片。表 2 显示了 ORMIR 和线性四元树的编码效率对比, 表 3 给出了基于 ORMIR 的多值图像几何矩生成算法、直接计算算法、Delta 算法以及基于 Biaxial 变换的算法的运行时间比较。

表 2 ORMIR 矩形数与线性四元树节点数对比

图像	线性四元树节点数	ORMIR 矩形块数	线性四元树节点数 / ORMIR 矩形块数
大脑	143521	76633	1.44
肝脏	145753	96717	1.51
肺部	242152	89927	2.69
手臂	135151	69698	1.94

表 3 几何矩生成算法的运行时间比较

图像	ORMIR 方法				实际加速因子	
	直接法 (毫秒)	Delta 方法 (毫秒)	Biaxial 变换法 (毫秒)	ORMIR 方法 (毫秒)	Delta 方法	Biaxial 变换
大脑	84.43	48.87	39.21	12.74	1.73	2.15
肝脏	84.91	47.56	38.89	16.38	1.79	2.18
肺部	83.85	48.37	39.84	15.07	1.73	2.10
手臂	83.84	47.45	39.14	11.38	1.77	2.14

(下转第 214 页)

结束语 论文针对恢复综合孔径微波辐射计展源图像时,标准 CLEAN 算法恢复的图像存在着条纹问题,提出了借鉴最大熵图像复原的算法思想,改进 CLEAN 算法。仿真结果表明,图像周围的条纹有了一定程度的改善。

参考文献

[1] 章勇,李兴国,王华力.毫米波合成孔径成像辐射计,1997
 [2] Högborn J A. Aperture synthesis with non-regular distribution of interferometer baselines. *Astron. Astrophys. Suppl.*, 1974, 15:417-426
 [3] Schwarz U J. Mathematical- statistical description of the itera-

tive beam re-moving technique (method CLEAN) [J]. *Astron. Astrophys.*, 1978, 65(2):345-356
 [4] Clark B. An Efficient Implementation of the Algorithm "CLEAN". *Astronomy & Astrophysics*, 1980, 89 (3):377-378
 [5] Segalovitz A, Frieden B. A CLEAN-type Deconvolution Algorithm. *Astronomy & Astrophysics*, 1978, 70 (3):335-343
 [6] Camps A, Bara J, Torres F, et al. Extension of the Clean Technique to the Microwave Imaging of Continuous Thermal Sources by Means of Aperture Synthesis Radiometers
 [7] Cornwell T. A Method of Stabilizing the CLEAN Algorithm. *Astronomy & Astrophysics*, 1983, 121(2):281-285
 [8] 陈春涛,黄步根,高万荣,等.最大熵图像复原及其新进展,2004

(上接第 207 页)

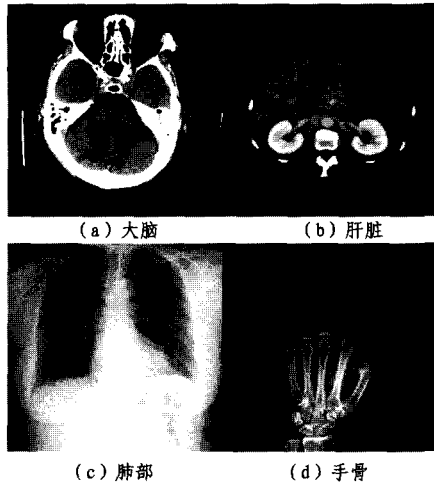


图 2 用于试验的 4 幅 8 比特位深 512×512 的医学图像

从表 2 中可以看出,ORMIR 表示一幅 8 比特位深 512×512 的数字图像需要的矩形块数量比线性四元树表示同样的图像需要使用的节点数少得多,对测试图像而言,线性四元树节点数量是 ORMIR 矩形块数量的 1.44 到 2.69 倍,表明 ORMIR 的编码效率高于线性四元树的编码效率。

表 3 给出了直接法、Delta 方法、基于 Biaxial 变换的算法以及基于 ORMIR 的几何矩生成算法的执行时间对比,实际加速因子是相应算法的执行时间相对直接法执行时间提高的倍数,由直接法的执行时间除以相应算法的执行时间获得。虽然在基于 Biaxial 变换的算法中,几何矩生成直接相关的运算次数相对直接法减少了 $3/4$ ^[17],然而由于变换本身需要大量加法,从测试数据上看,基于 Biaxial 变换的算法的实际加速因子比 2 略大。基于 ORMIR 的算法的实际加速因子最小也为 5.18,是所有算法中最快的。更重要的是,在测试图像上,基于 ORMIR 的算法最多耗时 16.38 毫秒,即基于 ORMIR 的几何矩生成算法能够以超过 50 帧/秒的速度计算 8 比特位深 512×512 的数字图像直到 3+3 阶的几何矩,是所有方法中唯一满足实时应用要求的纯软件实现的算法。

结束语 通过弱化二值图像块表示中同一矩形所覆盖的所有像素必须具有相同颜色的约束,提出了一种支持快速运算的无损多值图像表示方法,即可重叠矩形多值图像表示 (Overlapping Rectangle Multi-valued Image Representation, ORMIR)。ORMIR 采用递归方式将一幅多值图像分割为具有不同基础颜色的可重叠矩形,并使用孩子兄弟树来组织这些矩形。ORMIR 能够使用较少的矩形无损地表示一幅多值图像,取得了较高的编码效率,并使基于 ORMIR 的多值图像运算能够快速实现。基于 ORMIR,提出了一个多值图像几何矩生成算法,试验结果表明,基于 ORMIR 的几何矩生成算

法能够以每秒 50 帧以上的速度计算 8 比特位深的 512×512 的灰度图像直到 3+3 阶的几何矩,满足实时应用的需要。

参考文献

[1] Golomb S W. Run-length Encodings. *IEEE Transactions on Information Theory*, 1966, IT-12:399-401
 [2] Radson D, Boyd H A. Graphical Representation of Run Length Distributions. *Quality Engineering*, 2005, 17(2):301-308
 [3] Kaneko T, Okudaira M. Encoding of Arbitrary Curves Based on the Chain Code Representation. *IEEE Transactions on Communications*, 1985, COM-33(7):697-707
 [4] Wagenknecht G. A Contour Tracing and Coding Algorithm for Generating 2D Contour Codes from 3D Classified Objects. *Pattern Recognition*, 2007, 40(4):1294-1306
 [5] Hunter G M. Operations on Images Using Quad Trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1979, 1(2):145-153
 [6] Samet H. Region Representation: Quadrees from Binary Arrays. *Computer Graphics and Image Processing*, 1980, 13(1):88-93
 [7] Gargantini I. An Effective Way to Represent Quadrees. *Communications of the ACM*, 1982, 25(12):905-910
 [8] Aoki M. Rectangular Region Coding for Binary Image Data Compression. *Pattern Recognition*, 1979, 11(5/6):297-312
 [9] Mohamed S A, Fahmy M M. Binary Image Compression Using Efficient Partitioning into Rectangular Regions. *IEEE Transactions on Communications*, 1995, 43(5):1888-1893
 [10] Zahir S, Naqvi M. A New Rectangular Partitioning Based Lossless Binary Image Compression Scheme // *Proceedings of Conference on Electrical and Computer Engineering*. Saskatoon, Canada, 2005:281-265
 [11] Spiliotis I M, Mertzios B G. Real-time Computation of Two-Dimensional Moments on Binary Images Using Image Block Representation. *IEEE Transactions on Image Processing*, 1998, 7(11):1609-1615
 [12] Hu M-K. Visual Pattern Recognition by Moment Invariants. *IRE Transactions on Information Theory*, 1962, 8(2):179-182
 [13] Hatamian M. A Real-Time Two-Dimensional Moment Generating Algorithm and Its Single Chip Implementation. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1986, ASSP-34(3):546-553
 [14] Li Bincheng. A New Computation of Geometric Moments. *Pattern Recognition*, 1993, 26(1):109-113
 [15] Yang L, Albrechtsen F. Fast and Exact Computation of Cartesian Geometric Moments Using Discrete Green's Theorem. *Pattern Recognition*, 1996, 29(7):1061-1073
 [16] Shen T W, Lun D P K, Siu W C. Fast Algorithm for 2-D Image Moments via the Radon Transform // *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*. 1996, 3:1327-1330
 [17] Belkasim S, Kamel M. Fast Computation of 2-D Image Moments using Biaxial Transform. *Pattern Recognition*, 2001, 34(9):1867-1877
 [18] 王冰,职秦川,张仲选,等.灰度图像质心快速算法. *计算机辅助设计与图形学学报*, 2004, 16(10):1360-1365
 [19] Wu C H, Horng S J, Lee P Z. A New Computation of Shape Moments via Quadtree Decomposition. *Pattern Recognition*, 2001, 34(7):1319-1330