DOI: 10.13382/j.jemi.2017.11.002

二值图像截断四叉树编码及快速逻辑运算方法*

梁梦霞 郭斯羽 刘 敏 凌志刚 温 和
(湖南大学电气与信息工程学院 长沙 410082)

摘 要:提出了二值图像的截断四叉树结构,限定四叉树展开到指定的小图像块而非单像素水平;二值图像主体结构通过较少的树节点表示,而非单色小图像块则利用原始图像像素表达。通过四叉树和小图像块的结合,使得截断四叉树表达的总空间效率高于完全四叉树和原始二值图像。按深度优先的方式遍历截断四叉树,并分别对四叉树节点和小图像块的原始像素进行编码以获得基于截断四叉树的二值图像编码。给出了用于截断四叉树编码的逻辑运算方法。在实际图像上的实验表明,选择适当的截断层数,可以使截断四叉树编码的空间开销降到原始二值图像的 1/10 ~ 1/3,而逻辑运算的运行时间则普遍降到了逐像素逻辑运算的 1/5 以下。所提方法适用于二值图像待进行的逻辑运算次数远大于编解码次数的应用场合。

关键词: 四叉树;截断四叉树;二值图像;逻辑运算;深度搜索
中图分类号: TP391.4;TN919.8 **文献标识码**: A **国家标准学科分类代码**: 510.4050

Encoding and fast logical operation method for binary image based on truncated quadtree

Liang Mengxia Guo Siyu Liu Min Ling Zhigang Wen He
(College of Electrical and Information Engineering, Hunan University, Changsha 410082, China)

Abstract: The structure of truncated quadtree for binary image is proposed. The collapse of quadtree is performed until the level of small image block of a given size rather than single pixel. The main structure of a binary image is represented by a relative small number of quadtree nodes, and for the small image blocks with mixed colors, they are represented by their raw image pixels. The combination of quadtree and raw image blocks leads to the higher representation efficiency of the truncated quadtree. The truncated quadtree is traversed in a depth-first manner, and the quadtree nodes and raw image blocks are respectively encoded in the order of traversal, resulting in the truncated-quadtree-based codes of the binary image. The equivalent forms of binary image logical operations on the codes are presented. The experimental results on real-world images show that, with a proper truncating level setting, the memory cost of the truncated-quadtree-based code is 1/10 to 1/3 of the cost of the raw binary image, and the logical operation time on the code is below 1/5 of that of the pixel-by-pixel approach. The proposed methods are suitable for applications where the number of logical operation is significantly larger than the number of encoding and decoding.

Keywords: quadtree; truncated quadtree; binary image; logical operations; depth-first search

0 引 言

四叉树是一种经典的图像多层表示方式,其逐层细化的图像表达自然而然地提供了一种图像多分辨率分析的结构,并因此被广泛应用于图像压缩[1-4]、图像分割[5-7]以及更为复杂的图像分析与理解的任务之中,例如 Chen 等人[8]利用四叉树分解来区分图像中的平滑与细节丰富的区域,进而通过对细节丰富区域进行编码和匹配,以完成图像检索的任务;De 等人[9]以及 Bai 等人[10]均在多聚

收稿日期:2017-03 Received Date: 2017-03
* 基金项目:国家自然科学基金(61471167,61301254,61471166)资助项目

焦图像的融合中使用了四叉树结构来确定清晰区域与模糊区域;而 Padma 等人[11]以及 Kanchana 等人[12]则分别将四叉树用于提取埃纳德语字符和掌纹的特征提取之中。

二值图像的逻辑运算是图像处理过程中频繁出现的任务,其效率的提高将使广泛领域内的应用均可受益。提高二值图像逻辑运算效率的一个直接的思路,便是采用某种二值图像的压缩表达形式,并在该表达形式之上完成与逻辑运算等价的操作,通过二值图像信息空间开销的降低,来减少所需的等价操作的运算量,从而达到二值图像逻辑运算的时间效率和空间效率同时提高的目的。

Spiliotis 和 Mertzios[13]以及 Huang 和 Chung[14]较早使用这一思路来改进二值图像的逻辑运算,他们分别使用了二值图像的分块表示和基于插值的二分树表示。文献[15]使用了完全四叉树来对二值图像进行编码,并在此基础上给出了逻辑运算方法。更多的二值图像表示方法可见文献[16]。

本文在文献[15]的基础上进行了进一步改进,通过引入截断四叉树,给出了基于截断四叉树的二值图像编码,这种编码利用四叉树节点编码来表示图像中的大块子图像,并利用原始像素编码来表示图像中的小块非单色子图像。给出了操作于截断四叉树编码之上的二值图像常用逻辑运算的方法。

1　基于截断四叉树的二值图像编码

在文献[15]中给出的基于四叉树的二值图像编码的主要思路如下:

1)利用四叉树对二值图像进行表示;

2)对构造好的四叉树使用深度优先的方式进行遍历;

3)对每个树节点用 1Byte 进行编码,其定义方式如图 1 所示[15]。四叉树所有节点的编码按节点访问顺序以序列(数组)形式组织,即为二值图像的四叉树编码序列表达。

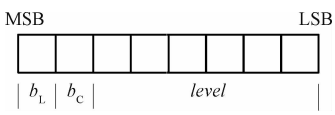

图 1　四叉树节点单字节编码的定义

Fig. 1　Definition of the one-byte code for quadtree node

四叉树表达方式对于大块单色区域是高效的,但对细碎黑白区域间杂的子图像块,其表达效率不高。以大小为 $2^n \times 2^n$ 的图像块为例,在最糟糕的情况下(在行和列两个方向上,黑白像素均相间出现),其四叉树节点数为 $4^0 + 4^1 + \cdots + 4^n = (4^{n+1} - 1)/3$,即四叉树编码的空间开销为 $(4^{n+1} - 1)/3$ Byte,而原始图像的空间开销(1 位/像素)仅为 2^{2n-3} Byte。文献[17]给出了四叉树节点数量 N_q 的一个上界为:

$$N_q \leqslant 16q - 11 + 6\sqrt{2}p \sim O(6\sqrt{2}p) \quad (1)$$

式中:q 为四叉树的深度,p 为区域轮廓点数量。可见,当图像大小不变时,p 越大,则四叉树节点数量越多;而 p 越大便意味着区域轮廓的"平滑"程度越差,即在轮廓附近存在着大量的细节性结构。

由于四叉树表达对于大块单色区域更为有效,而在区域轮廓附近其表达效率可能逊于二值图像的原始像素表达,因此可以将这两种方式结合运用:在某个图像粒度水平之上用四叉树表达,而在该水平之下采用原始像素表达。具体的做法如下。

给定截断层数 L_T;构造二值图像的四叉树,直到最小的图像分块大小为 $2^{L_T} \times 2^{L_T}$;从四叉树的角度来看,若最小的图像分块大小为 $2^{L_T} \times 2^{L_T}$,则意味着完全四叉树最底部的 L_T 层节点将不会被展开,故称此四叉树是截断层数为 L_T 的截断四叉树。

对每个四叉树节点用上述的字节编码进行编码,这类编码称为"节点编码";但若某个最小分块并非单色块,则再以 1 位/像素的方式,按某个给定顺序将分块中的像素组织为位编码序列,称为"像素编码"。最终的二值图像的截断四叉树编码以二元组 (V,P) 给出,其中 V 是截断四叉树各节点的编码按被访问的顺序所组成的节点编码序列,而 P 是相同遍历过程中所有非单色最小分块的像素编码依序组成的像素编码序列。

本文统一按由上至下再由左至右的方式遍历非单色块中的像素。显然,一个像素编码需要 $2^{L_T} \times 2^{L_T}$ 位的存储空间。选择适当的无符号整型,以最少的整型单元数来表示该序列:当 $L_T = 1$ 时,最小分块大小为 2×2,此时仅需 4 位即可,但仍用 1 个字节来表示;当 $L_T = 2$ 时,用 1 个 16 位无符号整型表示;当 $L_T \geqslant 3$ 时,对于 32 位平台,每个最小分块以 2^{2L_T-5} 个 32 位无符号整型来表示。

文献[15]中的四叉树编码过程是首先显式地构造出四叉树,然后再进行编码。本文给出一种在隐式构造四叉树的过程中同时完成编码的方法,其效率较前更高。

算法 1　基于截断四叉树的二值图像编码算法

输入:大小为 $2^N \times 2^N$ 的二值图像 $\boldsymbol{I}$;截断层数 L_T。

输出:$\boldsymbol{I}$ 的截断四叉树编码 (V,P)。

步骤如下:

1)初始化空的编码序列 V 和 P;初始化一个空的堆栈 B;B 中的元素为描述一个正方形图像分块的四元组 $b = (r,c,s,L)$,其中 r 和 c 分别为该分块的左上角像素

所在的图像行号和列号，s 表示该正方形分块的边长，而 L 表示该分块对应的节点在四叉树中的层数，显然四叉树的根节点即原始的图像的分块四元组为 $b_{\text{root}} = (0,0,2^N,0)$；$B.\text{push}\ b_{\text{root}}$。

2）若 B 为空，则编码结束，返回 (V,P)；否则，置 $b = B.\text{pop}()$。

3）根据 b 所对应的图像块（即 $\boldsymbol{I}$ 中第 $b.r$ 到 $b.r+s-1$ 行和第 $b.c$ 到 $b.c+s-1$ 列所构成的子图像）的单色与否和颜色以及 $b.L$ 构造节点编码，并将其加入 V 中。

4）若子块 b 为单色图像块，则至步骤 2）；否则，若 b 已是最小（截断）尺寸，即 $b.s = 2^{L_T}$ 或 $b.L = N - L_T$，则构造其像素编码，并将其加入 P 中，至步骤 2）；否则便按照四叉树子节点访问顺序的逆序，将当前图像块的 4 个子块的信息压入 B 中。本文按左上、左下、右上、右下的顺序访问当前图像块的子块，因此逆序依次将 $(b.r+b.s/2,\ b.c+b.s/2,\ b.s/2,\ b.L+1)$、$(b.r,\ b.c+b.s/2,\ b.s/2,\ b.L+1)$、$(b.r+b.s/2,\ b.c,\ b.s/2,\ b.L+1)$ 和 $(b.r,\ b.c,\ b.s/2,\ b.L+1)$ 压入堆栈 B 中，至步骤 2）。

算法 2 基于截断四叉树的二值图像重构算法

输入：被编码的二值图像的高 H 和宽 W；该二值图像的截断四叉树编码 (V,P)；截断层数 L_T。

输出：重构的二值图像 $\boldsymbol{I}$。

步骤如下：

1）初始化 $\boldsymbol{I}$ 为 H（W 的二值图像；置 $N = \lceil \log_2(\max\{H,W\}) \rceil$，其中 $\lceil\ \rceil$ 表示向上取整；初始化空堆栈 B，其元素为三元组 $b = (r,c,s)$，其中 r、c 和 s 的含义与编码算法中相同；$B.\text{push}\ (0,0,2^N)$；将序列 V 和 P 各自的读取游标均置于第 1 个编码之前。

2）若 B 为空，则重构结束，返回 $\boldsymbol{I}$；否则，置 $b = B.\text{pop}()$。

2 截断四叉树编码的空间开销

由式（1）可以看出，截断四叉树编码的长度也与区域轮廓的长度有关，不过此时的区域轮廓是一个较粗粒度下的轮廓。当截断四叉树的截断层数为 L_T 时，构建四叉树的分块过程最多进行到 $2^{L_T} \times 2^{L_T}$ 大小的图像块时即告结束，因此对于截断四叉树的节点数量而言，相当于将图像分为 $2^{L_T} \times 2^{L_T}$ 大小的网格之后，以该大小的图像块作为“像素”，然后再进行常规的四叉树的构建。此时如果某个图像块中包含了原图像区域中的轮廓点，该图像块所对应的粗粒度“像素”即可视为粗粒度下的轮廓点。设粗粒度下的轮廓点的个数为 p_c，于是有截断四叉树的节点数量 $N_T \sim O(6\sqrt{2}p_c)$，因此节点编码序列的空间开销（以字节计）为 $\|V\| \sim O(6\sqrt{2}p_c)$。

另一方面，对于粗粒度下的背景像素和区域内像素，它们均被编码为节点编码，因此不会产生像素编码方面的开销，但是，轮廓像素除开产生节点编码之外，还会产生像素编码。每个粗粒度轮廓像素对应的像素编码长度为 2^{2L_T-3} Byte（最小为 1 Byte），因此总的像素编码序列的空间开销（以字节计）为 $\|P\| = \max\{1,2^{2L_T-3}\} \cdot p_c$。

因此基于截断四叉树的编码的总空间开销为：

$$\|V\| + \|P\| \sim O[(6\sqrt{2} + \max\{1,2^{2L_T-3}\})p_c] \quad (2)$$

3 基于截断四叉树编码的二值图像逻辑运算

与文献[15]一样，本文仍然考虑二值图像 $\boldsymbol{A}$ 的逻辑非（$\neg\boldsymbol{A}$，NOT）以及两幅相同大小的二值图像 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 的逻辑与（$\boldsymbol{A}\cap\boldsymbol{B}$，AND）、逻辑或（$\boldsymbol{A}\cup\boldsymbol{B}$，OR）、逻辑异或（$\boldsymbol{A}\oplus\boldsymbol{B} = (\boldsymbol{A}\cap\boldsymbol{B})\cup(\neg\boldsymbol{A}\cap\neg\boldsymbol{B})$，XOR）和集合差（$\boldsymbol{A}-\boldsymbol{B} = \boldsymbol{A}\cap\neg\boldsymbol{B}$，SUB）运算。在截断四叉树编码上进行的二值图像逻辑运算，其基本思路与文献[15]中是一致的：将二值图像之间的逻辑运算分解为四叉树中各个对应节点之间的运算，并且其中至少一个节点为单色节点，此时的逻辑运算将可简化为四叉树中对应子树的输出或反色输出的操作；然后利用编码排列顺序与深度优先访问四叉树的顺序相同的事实，将四叉树子树的输出进一步转变为当前编码序列中若干连续编码的输出。

不过对于截断四叉树，有一个需要额外处理的情况，即当被考察的一对节点均非单色节点，同时它们又处在截断四叉树的截断层即最底层，此时对应的两个子图像块的逻辑运算，要由像素编码的直接逻辑运算来完成，输出的也是结果子图像块的像素编码，而非树节点。

算法 3 基于截断四叉树编码的二元逻辑运算

输入：两幅相同大小的待运算二值图像的截断四叉树编码 $Q_1 = (V_1,P_1)$ 和 $Q_2 = (V_2,P_2)$。

输出：运算结果图像的截断四叉树编码 $Q = (V,P)$。

步骤如下：

1）置 V 和 P 为空序列；设 V_1 中的当前节点为 v_1，V_2 中的当前节点编码为 v_2。

2）若 V_1 中编码已遍历，返回 Q；否则至 3）。

3）若 v_1 对应单色节点，则至步骤 4）；否则，若 v_2 对应单色节点，则至步骤 7）；否则，至步骤 10）。

4）若 v_1 对应节点为白色，至步骤 5）；否则至步骤 6）。

5）（v_1 节点为白色）对 AND，将 Q_2 中的当前子树加入 Q，略过 Q_1 中的当前子树；对 OR，将 Q_1 中的当前子树加入 Q，略过 Q_2 中的当前子树；对 XOR 和 SUB，将 Q_2 中

的当前子树反色后加入 Q,略过 Q_1 中的当前子树;至步骤 2)。

6)(v_1 节点为黑色)对 AND 和 SUB,将 Q_1 中的当前子树加入 Q,略过 Q_2 中的当前子树;对 OR 和 XOR,将 Q_2 中的当前子树加入 Q,略过 Q_1 中的当前子树;至步骤 2)。

7)若 v_2 对应节点为白色,至步骤 8);否则至步骤 9);

8)(v_2 节点为白色)对 AND,将 Q_1 中的当前子树加入 Q,略过 Q_2 中的当前子树;对 OR,将 Q_2 中的当前子树加入 Q,略过 Q_1 中的当前子树;对 XOR,将 Q_1 中的当前子树反色后加入 Q,略过 Q_2 中的当前子树;对 SUB,将 Q_2 中的当前子树反色后加入 Q,略过 Q_1 中的当前子树;至步骤 2)。

9)(v_2 节点为黑色)对 AND,将 Q_2 中的当前子树加入 Q,略过 Q_1 中的当前子树;对 OR、XOR 和 SUB,将 Q_1 中的当前子树加入 Q,略过 Q_2 中的当前子树;至步骤 2)。

10)若 v_1 对应节点所在层是截断四叉树的截断层,则至步骤 11);否则至步骤 12)。

11)(已到达截断层)将 v_1 加入 V,然后对 AND、OR、XOR 和 SUB,分别将 P_1 和 P_2 中的当前像素编码 p_1 和 p_2 进行 p_1 & p_2、p_1 | p_2、p_1 ^ p_2 和 p_1 & ~p_2 运算,其中 &、|、^和 ~分别表示位与、位或、位异或和位取反运算,像素编码的计算结果加入 P;至步骤 2)。

12)(未到达截断层)置 v_1 和 v_2 分别为 V_1 和 V_2 中的下一个节点编码;至步骤 2)。

由上述步骤可见,基于截断四叉树编码的二元逻辑运算的关键在于子树的操作,即输出子树、略过子树和输出反色子树。由于截断四叉树编码是根据以深度优先方式遍历四叉树时的顺序依次存放的,因此,一个子树的所有节点编码是以子树根节点编码为首,以连续的方式存放在编码序列之中的。

算法 4　输出子树的截断四叉树编码

输入:待输出子树的截断四叉树编码序列 $Q = (V,P)$ 及该子树的根节点编码 v;接受输出编码的序列 $Q_O = (V_O,P_O)$。

步骤如下:

1)置 $L = v.level$;将 v 加入 V_O,并置 v 为 V 中的下一个节点编码。

2)若 $v.level = L$,则结束;否则至步骤 3)。

3)将 v 加入 V_O;若 v 处于截断层,则将 P 中的当前像素编码加入 P_O;置 v 为 V 中的下一个节点编码;至步骤 2)。

请注意,在上述输出子树的步骤中,对于根节点 v 本身就处于截断层的情况,实际上是在二元逻辑运算步骤 11)中处理的。

略过子树的操作基本与输出子树相同,只要去掉输出代码的步骤即可;输出反色子树的操作也基本相同,只是在输出节点编码时需要将 b_C 位取反,而在输出像素编码时,需要将像素编码按位取反。

对逻辑非操作,可简单地用输出反色子树的方式,将整个编码序列从根节点开始反色输出即可。

4　截断四叉树编码的压缩

上述二元逻辑运算方法不保证所得结果图像的截断四叉树编码是“最紧凑”的。若参与运算的两幅图像的对应子块恰为互补,那么在经过 AND、OR、XOR 之后,该子块将成为一个单色子块,从而在四叉树中合并为一个叶节点。但上述运算方法不会自动察觉这一点,这时就会出现编码使用多于必要数量的节点来表示图像子块的情况。

可以在若干次二元逻辑运算后对编码进行压缩,方法是先将截断四叉树编码重构为完全四叉树结构,然后折叠那些被不必要展开的子树,之后再重新编码。具体步骤在此略去。

5　实验结果与讨论

利用 Microsoft VS2005 进行编译,以 C + + STL 的方式实现了本文的基于截断四叉树的二值图像编码及逻辑运算方法(TQTBYC),并生成了 MATLAB MEX 可执行模块。作为对比,用相同的方式实现了逐像素的二值图像逻辑运算(PBYP)和基于完全的四叉树的二值图像编码及逻辑运算方法(QTBYC)[15]。

实验运行的硬件环境为 Pentium ® Dual-Core E5200 2.5 GHz、2 GB RAM 的 PC,软件环境为 Microsoft Windows XP Professional SP3、MATLAB 7.1。实验使用了 3 个图像库,分别包含 40 张自然景观图像(nature40)、城市建筑图像(building40)和卡通图像(cartoon40)。nature40 中图像大小为 2 560 × 1 440,其余两个为 2 048 × 1 536。图像经 Otsu 阈值分割后,用不同半径 r_{SE} 的圆盘状结构元素进行开、闭运算的后处理,得到实验用的二值图像。

5.1　编码的空间开销实验

第 1 个实验考察 TQTBYC 编码的空间开销。分别取截断层数 $L_T = 1, 2, 3, 4$。对于 $L_T = 1$,每个像素编码用 1 Byte 表示;对于 $L_T = 2$,用 1 个短整型(2 Byte)表示;对于 $L_T = 3$ 和 4,则分别用 2 个和 8 个 32 bit 整型(8 Byte 和 32 Byte)表示。TQTBYC 的编码空间开销 n_m 以节点编码和像素编码总共耗费的字节数表示。原始的二值图像的空间开销以 1 pixel/bit 计算,即图像像素数量/

8 Byte。不同 L_T 下的压缩比定义为：

$$\text{压缩比} = \frac{n_m}{\frac{HW}{8}} \tag{3}$$

式中：H 和 W 分别为图像的高和宽。不同 L_T 和不同后处理结构元素半径 r_{SE} 下的压缩比如图 2 所示，其中 $r_{SE}=0$ 表示未进行后处理。

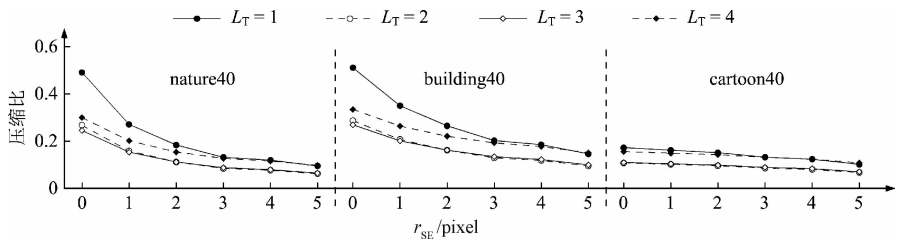

图 2　截断四叉树编码的压缩比

Fig. 2　Compression ratio of the truncated quadtree based encoding

由图 2 可见，在所有情况下，TQTBYC 都是原始二值图像的一种压缩表达。当截断层数 LT 由 1 变为 2 时，压缩比有明显改善。这是由于对于较小的截断层图像块，用直接像素编码比使用基于四叉树的编码更为高效；特别注意到，这一改善在 nature40 和 building40 两个图像库的较小 r_{SE} 值下更为明显，因此时图像区域轮廓存在大量细节性结构，所以若使用完全四叉树编码，在区域轮廓处往往需要展开到单像素级别，此时四叉树编码的空间效率不高。由式(1)也可以看出，区域轮廓细节丰富将导致区域轮廓点数量 p 增加，因此四叉树节点数量也随之增加，造成节点编码开销增大。随着 r_{SE} 不断增加，后处理实际上对区域轮廓进行了平滑，消除了细节性结构，减少了轮廓点数。因此这时可以明显看出压缩比的改善有限。对于 cartoon40 图像库，由于其内容为卡通图像，因此在经过阈值分割后，轮廓本来就比较平滑，细节较少，因此压缩比的改善不如前两者明显。而当 L_T 进一步增加时，对于减少截断四叉树的节点数量及节点编码的空间开销贡献相对不大，而像素编码的空间开销则以 4 的幂次增加，因此压缩比不降反升。

为进一步验证 TQTBYC 编码空间开销与二值图像区域轮廓点数之间的关系，我们取未经后处理的二值图像，分别计算其 TQTBYC 编码空间开销 n_m 和区域轮廓点数 p_c，并绘制散点图如图 3 所示。

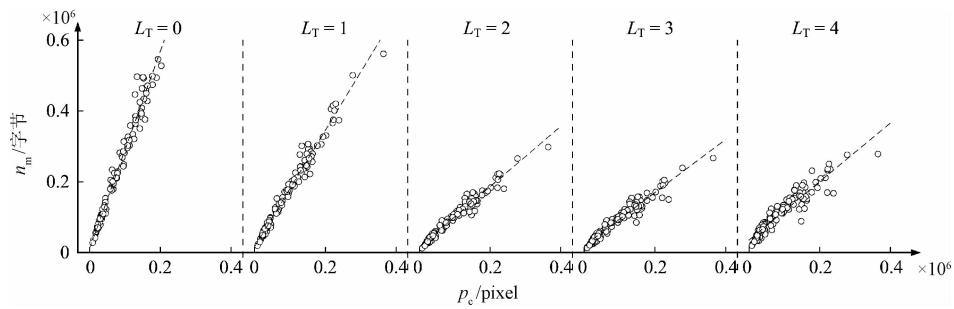

图 3　截断四叉树编码空间开销与区域轮廓点数的关系

Fig. 3　Relationship of memory cost of the truncated quadtree based encoding and the number of contour pixels

由图 3 可见，n_m 和 p_c 之间呈现出较好的线性关系，验证了式(2)所给出的关系。另一方面，从图 2 和 3 都可以看出 n_m 有随着 L_T 的增加而增加的趋势，这一点同样在式(2)中得到了反映。

由于 $L_T=2$ 时的压缩比在多数情况下为最佳，因此在后续的实验中，固定选择截断层数为 2。

5.2　编码与逻辑运算实验

在本实验中，各图像库中的图像被随机分为 20 对，然后对它们进行二元逻辑运算。NOT 运算则对每幅图像执行。PBYP、QTBYC 和 TQTBYC 等 3 种方法的逻辑运算的运行时间如图 4 所示。该运行时间都是单纯用于逻辑运算上的时间，不包括编解码的时间。为提高结果的可靠性，每种方法的每个逻辑运算均进行了 10 次，然后取平均值。

由图 4 可见，QTBYC 和 TQTBYC 的逻辑运算时间均优于 PBYP 方法，尤其是 TQTBYC，在所有后处理方式中，运行速度均较 PBYP 快 5 ~ 10 倍。当 r_{SE} 较小、轮廓平滑作用不强时，TQTBYC 也明显优于 QTBYC，r_{SE} 较大时两

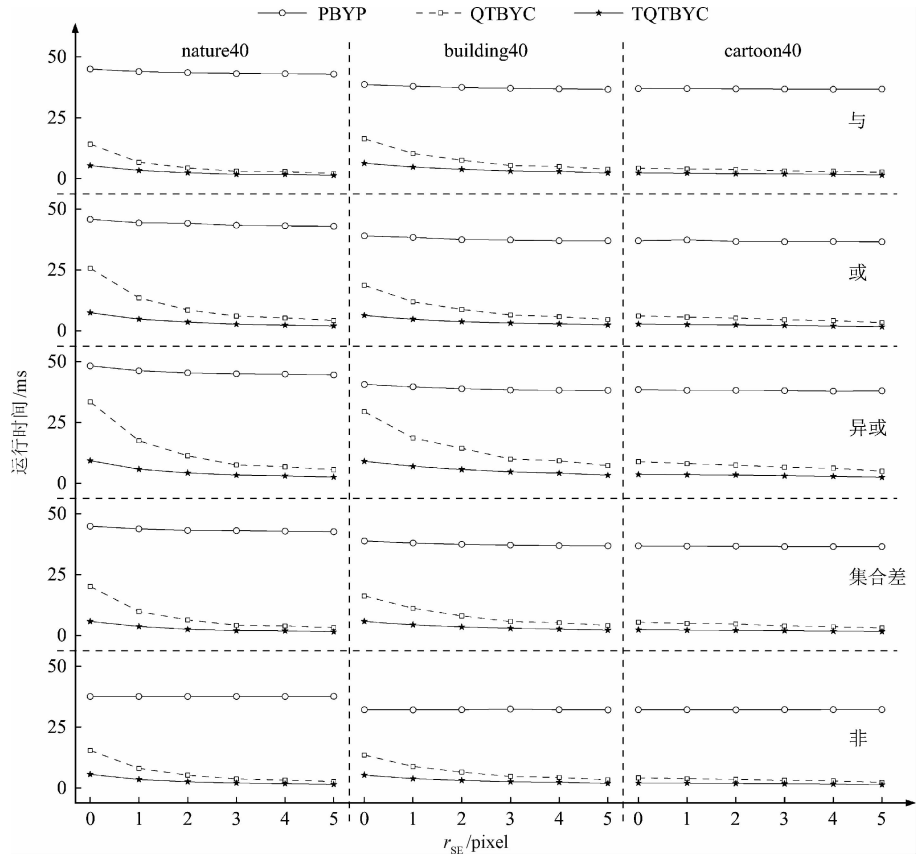

图 4　各算法执行逻辑运算的时间

Fig. 4　Execution time of logical operation by each algorithm

者则十分接近,但 TQTBYC 也仍然略占优势。

QTBYC 和 TQTBYC 各自的编码和图像重建的运行时间如图 5 所示。TQTBYC 的编解码时间仍优于 QTBYC,在后处理平滑力度较弱时优势明显。

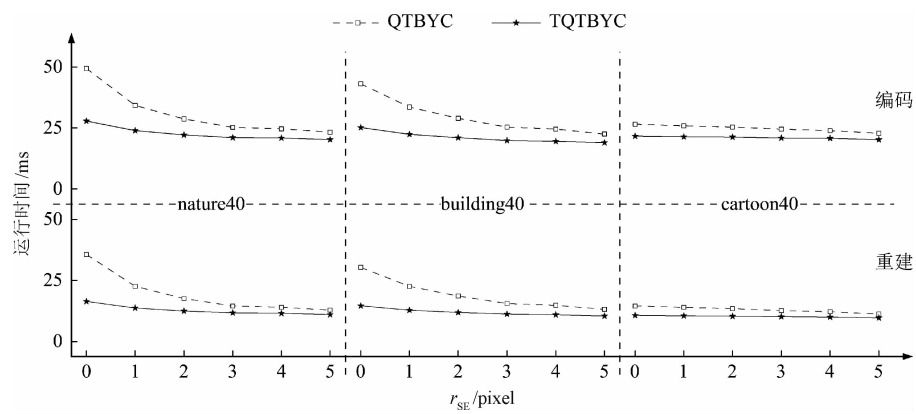

图 5　基于完全四叉树和截断四叉树的编码和图像重建的时间

Fig. 5　Encoding and decoding time of the schemes based on full and truncated quadtrees

此外,由图 4 和 5 可见,在所有情况下, TQTBYC 的编解码和逻辑运算的总时间仍然可以达到与 PBYP 相当的程度。但如文献[15]所述,QTBYC 和 TQTBYC 这类基于某种特定压缩表达的逻辑运算方法,它们最为适用的应用场景,是需要在相对固定的二值图像上重复大量逻辑运算的情况,如文献[18]中给出的应用实例;或者是在构建一个涉及二值图像的应用系统时,从最初就采用二值图像的压缩表达形式,而后的操作均在这一表达上完成。这时,少量编解码的时间开销将很快为其他操作速度的增加所抵消。另外,当二值图像尺寸大、数量多,

使得存储空间成为一个重要考虑因素时，QTBYC 和 TQTBYC 均是二值图像的压缩表达的这一特点将赋予它们以额外的优势和适用性。

5.3　编码压缩实验

为了考察编码的非最小性对于性能的影响，将各个图像库中的第 1 幅图像依次与第 2 幅，第 3 幅，…，第 40 幅图像连续进行二元逻辑运算，并获取各个连续运算步骤下未压缩的编码和压缩后的编码之间的空间开销之差和逻辑运算运行时间之差。在不同的后处理情况下，这两者各自的平均值和最大值如图 6 所示。

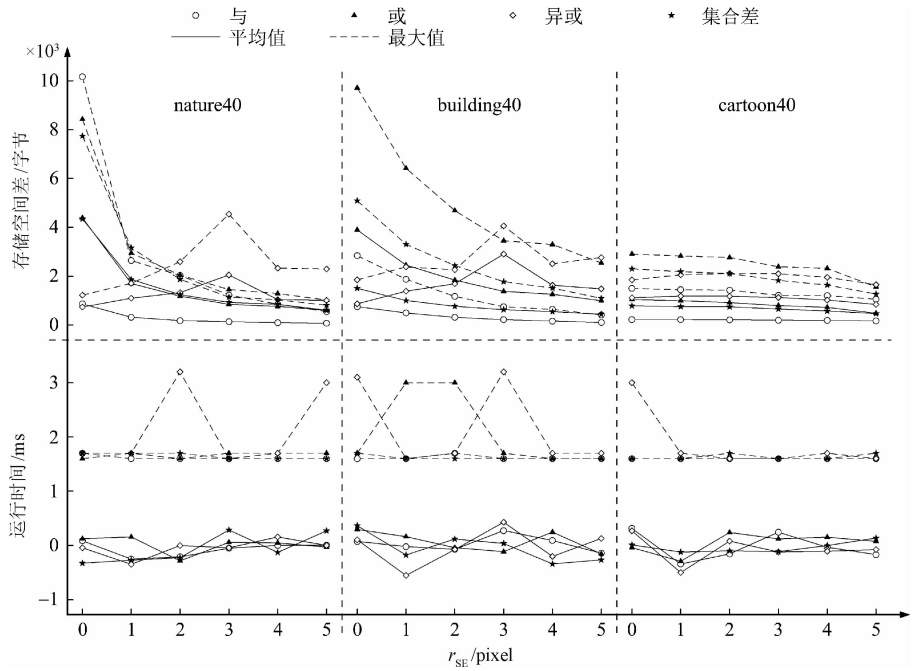

图 6　未压缩和压缩后的截断四叉树编码空间开销之差和逻辑运算时间之差

Fig. 6　Differences in memory cost and logical operation execution time between uncompressed and compressed TQTBYC codes

由图 6 可见，相比于 TQTBYC 编码本身的大小，未压缩和压缩后的编码差相对而言很小，而从运行时间差也可以看出，运行时间的差别并不显著，甚至还有压缩后的运行速度更慢的情况。不过这很可能是由于该时间差已经达到了所使用的计时器的分辨率极限，因此只能定性地来看待。由于运行时间主要与编码长度有关，因此这也体现了编码空间开销的差别并不显著。考虑到编码压缩操作本身是相对十分费时的，因此通常可以不考虑在逻辑运算过程中执行编码压缩。

5.4　应用实验

以文献[18]中的植物叶片中轴检测应用来测试和对比基于完全四叉树编码和截断四叉树编码的方法。实验的设置请参见文献[18]，不再赘述，只不过在所涉及的二值图像表达和逻辑运算上，文献[18]中的方法(LMA-QTBYC)使用完全四叉树编码，本文方法(LMA-TQTBYC)则使用截断四叉树编码。两种方法的运行时间对比如表 1 所示。由表 1 可见，使用截断四叉树编码将运行时间减少了约 10%，表明方法具有一定的实用价值。

表 1　应用实验中各算法的总运行时间

Table 1　Execution time of algorithm in the application experiment

算法	运行时间/s
LMA-QTBYC	78.325
LMA-TQTBYC	71.871

6　结　论

本文提出了一种基于截断四叉树的二值图像编码和逻辑运算的方法。截断四叉树是一种未完全展开的四叉树结构，它以四叉树表示图像大范围结构，以原始像素表示图像局部细节，从而能够更高效地表达具有丰富轮廓细节的二值图像。

通过对截断四叉树以深度优先的方式遍历，按照遍历顺序依次将树节点和截断层以下的原始图像像素分别编码为节点编码和像素编码，便得到了截断四叉树编码。在这一编码基础上给出了逻辑与、逻辑或、逻辑异或、集合差和逻辑非等 5 个常用的二值图像逻辑运算的运算方法。

在实际图像库上的实验表明,截断四叉树编码是二值图像的一种有效的压缩方式,能显著缩短逻辑运算的时间。考虑到编码和解码的运行时间开销,截断四叉树编码适用于相对固定的图像进行大量逻辑运算的场合,或者是在开发针对二值图像的应用系统时,从最初就直接引入截断四叉树编码来表达和操作二值图像,这样可以最大限度地发挥本方法在存储空间和运行时间两方面的优势。

参考文献

[1] CHEN F, LI P, PENG Z, et al. A fast inter coding algorithm for HEVC based on texture and motion quad-tree models [J]. Signal Processing: Image Communication, 2016, 47(9):271-279.

[2] KAMBLE S D, THAKUR N V, MALIK L G, et al. Color video compression based on fractal coding using quadtree weighted finite automata [C]. Information Systems Design and Intelligent Applications, 2015: 649-658.

[3] 王相海,张智迪,宋传鸣. 四叉树分块的高光谱图像分布式无损编码[J]. 中国图象图形学报, 2015, 20(8):1102-1109.
WANG X H, ZHANG ZH D, SONG CH M. Lossless distributed source coding of hyperspectral images based on quadtree segmentation [J]. Journal of Image and Graphics, 2015, 20(8):1102-1109.

[4] YUEN C H, LUI O Y, WONG K W. Hybrid fractal image coding with quadtree-based progressive structure [J]. Journal of Visual Communication and Image Representation, 2013, 24(8):1328-1341.

[5] 喻金桃,郭海涛,李传广,等. 四叉树与多种活动轮廓模型相结合的遥感影像水边线提取方法[J]. 测绘学报, 2016, 45(9):1104-1114.
YU J T, GUO H T, LI CH T, et al. A waterline extraction method from remote sensing image based on quad-tree and multiple active contour model [J]. Acta Geodaetica et Cartographica Sinica, 2016, 45(9): 1104-1114.

[6] 郭海涛,孙磊,申家双,等. 一种四叉树和测地线活动轮廓模型相结合的海陆影像分割方法[J]. 测绘学报, 2016, 45(1):65-72.
GUO H T, SUN L, SHEN J SH, et al. An island and coastal image segmentation method based on quadtree and GAC model [J]. Acta Geodaetica et Cartographica Sinica, 2016, 45(1):65-72.

[7] 胡志立,郭敏. 基于四叉树分解与图割的彩色图像快速分割[J]. 计算机工程与科学, 2015, 37(2): 390-396.
HU ZH L, GUO M. Fast segmentation in color image based on quadtree decomposition and graph cuts [J]. Computer Engineering & Science, 2015, 37(2): 390-396.

[8] CHEN H H, DING J J, SHEU H T. Image retrieval based on quadtree classified vector quantization [J]. Multimedia Tools and Applications, 2011, 72(2): 3625-3628.

[9] DE I, CHANDA B. Multi-focus image fusion using a morphology-based focus measure in a quad-tree structure [J]. Information Fusion, 2013, 14(2): 136-146.

[10] BAI X, ZHANG Y, ZHOU F, et al. Quadtree-based multi-focus image fusion using a weighted focus-measure [J]. Information Fusion, 2015, 22(3): 105-118.

[11] PADMA M C, PASHA S. Quadtree based feature extraction technique for recognizing handwritten Kannada characters [C]. Emerging Research in Electronics, Computer Science and Technology, 2012:725-735.

[12] KANCHANA S, BALAKRISHNAN G. Quadtree decomposition for palm print feature representation in palmprint recognition system [C]. IEEE International Conference on Advanced Communication Control and Computing Technologies, 2012:291-294.

[13] SPILIOTIS I M, MERTZIOS B G. Fast algorithms for basic processing and analysis operations on block-represented binary images [J]. Pattern Recognition Letters, 1996, 17(14):1437-1450.

[14] HUANG C Y, CHUNG K L. Fast operations on binary images using interpolation-based bintrees [J]. Pattern Recognition, 1995, 28(3):409-420.

[15] 郭斯羽,周卫方,温和,等. 利用四叉树编码的快速二值图像逻辑运算方法[J]. 电子测量与仪器学报, 2016, 30(6):845-853.
GUO S Y, ZHOU W F, WEN H, et al. Fast binary image logical operation method using quadtree coding [J]. Journal of Electronic Measurement and Instrumentation, 2016, 30(6):845-853.

[16] SUK T, HÖSCHL IV C, FLUSSER J. Decomposition of binary images: A survey and comparison [J]. Pattern Recognition, 2012, 45(12):4279-4291.

[17] GERARD Y, VACAVANT A, FAVREAU J M. Tight bounds in the quadtree complexity theorem and the maximal number of pixels crossed by a curve of given length [J]. Theoretical Computer Science, 2016, 624(4):41-55.

[18] 郭斯羽,董红霞,张翌. 一种用于植物叶片图像骨架提取的去毛刺方法[J]. 电子测量与仪器学报, 2013, 27(1):52-56.
GUO S Y, DONG H X, ZHANG Y. Pruning method for skeletonization of plant leaf images [J]. Journal of Electronic Measurement and Instrument, 2013, 27(1): 52-56.

作者简介

梁梦霞,1992 年出生,毕业于长江大学,现为湖南大学硕士研究生,目前主要研究方向为图像处理。

Liang Mengxia was born in 1992, and graduated from Yangtze University. And now, she is a M. Sc. candidate in Hunan University. Her present research interest is image processing.

郭斯羽,1975 年出生,毕业于浙江大学,现任湖南大学副教授,目前主要研究方向为图像处理与机器视觉、系统建模与仿真。

E-mail: syguo75@163.com

Guo Siyu was born in 1975, and graduated from Zhejiang University. And now, he is an associate professor in Hunan University. His present research interests focus on image processing, computer vision, and system modeling and simulation.

罗德与施瓦茨公司在移动网络测试方面引入行业内具有权威性的合作伙伴共同迎接网络优化中带来的挑战

罗德与施瓦茨移动网络测试部门正在引入行业内有权威性的合作伙伴来应对移动网络测试过程中面临的复杂问题。通过加强与网络测试服务提供商和网络测试服务公司合作伙伴的合作,罗德与施瓦茨在移动网络测试中为客户提供最优质的解决方案,并促使这种解决方案在行业内得以持续的传播。罗德与施瓦茨移动网络测试及它的认证机构可以为运营商提供更好的端到端的服务和专业的移动网络测试分析与评估。

随着复杂多样化的射频无线环境增加,网络性能测试迎来了前所未有的挑战,一方面是由于通讯领域先进的技术快速的发展,这包括连接着数十亿的终端和海量的数据传输的物联网。另一方面,智能手机的速度越来越快,容量越来越大,越来越多的消费需求和新的手机应用软件迫使运营商减少开支,节约成本。作为在移动网络测试领域的引领者,罗德与施瓦茨加强和这些专业的有资质的合作伙伴合作,来迎接运营商面临的网络优化方面的挑战。

罗德与施瓦茨负责移动网络测试领域的副总裁 Hanspeter Bobst 说到:“和这些服务商共同合作对各方而言都是一个双赢,通过这种紧密的合作,各方特别是研发可相互获取新的技能和知识,这有利于我们和合作伙伴共同为运营商提供更专业和更具有针对性的解决方案来解决错综复杂的网络问题。这也使我们对各种网络问题具有可预见性,长期的知识共享使我们更具有专业技能,使供应商在短时间内用少量的开销来达到预期的效果。”

罗德与施瓦茨移动网络测试的合作伙伴会在早期参与到开发中来,并且会接触到公司的新产品并且了解其特性。这些专业的视角以及知识经验的共享,大大缩短了一个新的解决方案流向市场的时间。同样,这也使得罗德与施瓦茨公司的合作商和运营商在第一时间获取这些新的技术。

目前罗德与施瓦茨正在和几个合作伙伴谈判,谈判结果会很快在罗德与施瓦茨网站上移动网络测试版块发布,详情见网址:https://www.mobile-network-testing.com/en/about/certified-partners/.

关于罗德与施瓦茨在网络运营商、网络服务供应商和网络监管商的优化方案和材料,请访问 https://www.mobile-network-testing.com and https://blog.mobile-network-testing.com.