第15卷 第1期 自动化学报 Vol. 15, No. 1
1989年1月 ACTA AUTOMATICA SINICA Jan., 1989

四值逻辑在图象处理中的应用

张田文 李仲荣
(哈尔滨工业大学)

摘 要

本文讨论二维目标的四分形结构. 提出四分形阶的概念，直接编码包含在目标中的更高阶四分形. 给出四分形的几种算符（$\Longrightarrow$，$\cap$，#），使得目标操作如连接、相交、取补等可以用代数方法计算. 最后提出四分形在计算机中的一种有效数据结构以及这些算符在通用计算机上的实现方法.

一、引 言

区域表示法在图象处理、景物分析和计算机制图等方面都起着重要作用. 为了表示一区域有基于其边界或者骨架的各种方法. 近来，已提出的二维目标的四分树方法[1,2]有若干优点，但对四分树操作需要搜索过程，以四分树形式存取数据也较复杂，同时四分树中包含的几何信息是隐含的.

一个二维目标总可以刚好包含在一个正方形内，该正方形称为目标的整正方形，记作 U_n. 它可以分割成四个正方形，每个正方形又可以分割成四个正方形，这种过程可以继续下去，在上述分割中得到的任意正方形称为四分形. 如果一个四分形完全被目标覆盖，给它标号 BLACK；如果一个四分形完全在目标之外，给它标号 WHITE；否则给它标号 GRAY. 对具有标号 GRAY 的四分形继续分割，直至得到的四分形具有标号 BLACK 或者 WHITE，此时得到的四分形称为单位四分形，记作 c^u. 在对 U_n 分割过程中得到的具有标号 BLACK 的四分形集合构成目标区域.

二、四分形编码

编码四分形便于目标操作. 假设对整正方形 U_n 分割 n 次，其中任意四分形 c 的编码形式可以写成带权展开式 $c = c_{n-1}4^{n-1} + \cdots + c_{n-i}4^{n-i} + \cdots + c_0 4^0$，$1 \leqslant i \leqslant n$，其中 c_{n-1} 是第一次分割时 c 的编码，c_{n-i} 是第 i 次分割时 c 的编码，c_0 是第 n 次分割时 c 的编码. 各次分割得到的编码顺序排列便构成四分形 c 的编码，上式可以简写成 $c = c_{n-1}\cdots c_{n-i}\cdots c_0$. 如果第 i 次分割四分形 c^l 得到四分形 c^k，其中 c^k_{n-i} 编码如下：

本文于1986年4月1日收到.

c^k在c^l中的位置: NW NE SW SE

c^k_{n-i}: 0 1 2 3

其中 0、1、2、3 是这里所用的四值逻辑的四个逻辑符号，它们的物理意义如图 1 所示，又称它们为四分数，记作 $T=\{0,1,2,3\}$. 如果第 i 次分割得到的四分形 c^k(标号为 BLACK 或 WHITE) 不需要继续分割，则其 $c^k_j=x, n-i-1\geqslant j\geqslant 0$. 若四分形 c^k 的编码中有 p 个元素是 x，既 $c^k=c^k_{n-1}c^k_{n-2}\cdots c^k_p xx\cdots x$，其中 $c^k_{n-1},c^k_{n-2},\cdots,c^k_p\in T$，则说 c^k 是 P-四分形，P 称为 c^k 的阶. 一个 P-四分形包含 4^p 个 0-四分形，它们是用 0、1、2、3 分别代替 x 组合成的四分形. 显然，整正方形 $U_n(=xx\cdots x)$ 是 n-正方形，单位四分形 $c^u(=c^u_{n-1}\cdots c^u_{n-i}\cdots c^u_0, c^u_{n-i}\in T, 1\leqslant i\leqslant n)$ 是 0-四分形. 四分形的编码过程如图 2 所示，这里 $n=3$.

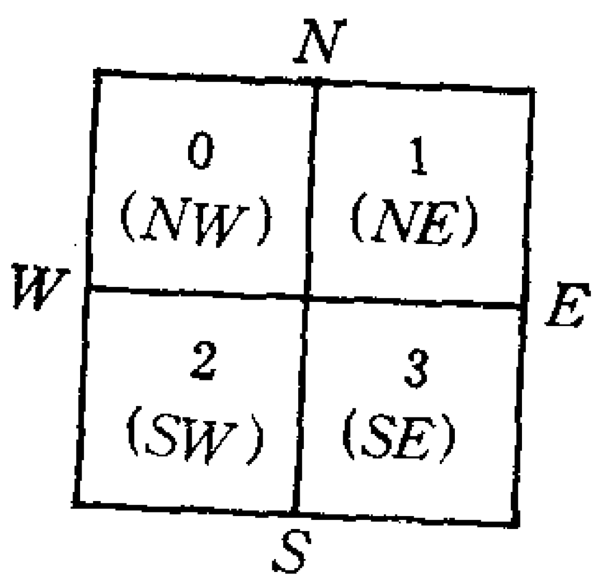

图 1 四个逻辑符号的物理意义

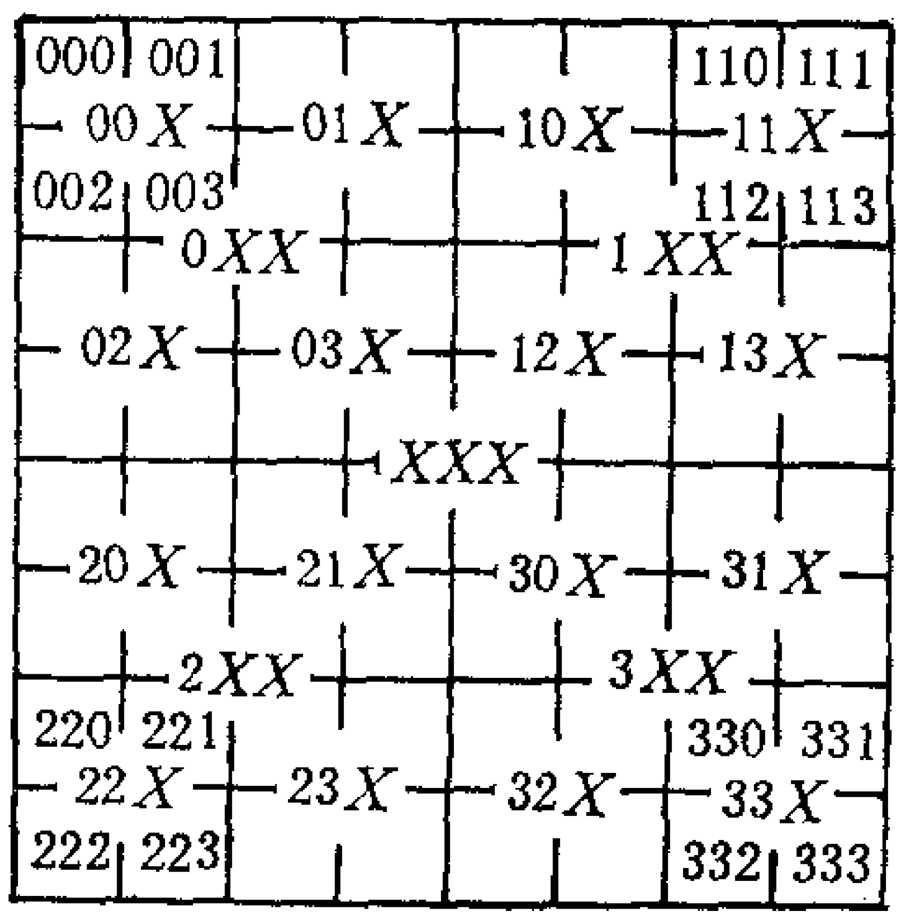

图 2 整正方形分割三次时各四分形编码过程

三、四分形算符

1. 蕴含关系

四分形 c^k 蕴含于 c^l，记作 $c^k\Rightarrow c^l$，当且仅当 c^k 所覆盖的所有 0-四分形也被 c^l 覆盖. 蕴含关系$\Rightarrow$可以通过表 1(a) 的规定来定义: 如果对任一座标 i，对应表中的交点是 ϕ，则 $c^k\not\Rightarrow c^l$; 否则 $c^k\Rightarrow c^l$. 蕴含关系是自反和传递的.

2. 相交

四分形 c^k 和 c^l 的交，记作 $c^k\cap c^l$. 它是通过表 1(b) 的规定来定义: 如果对任一座标 $i, c^k_i\cap c^l_i=\Omega$，则 $c^k\cap c^l=\phi$; 否则 $c^k\cap c^l=((c^k_{n-1}\cap c^l_{n-1})\cdot(c^k_{n-2}\cap c^l_{n-2})\cdots(c^k_0\cap c^l_0))$. 相交运算是可结合、交换和分配的.

3. #-积

四分形 c^k# -积 c^l，记作 $c^k\# c^l$. 它是通过表 1(c) 的规定来定义: 如果对任一座标 $i, c^k_i\# c^l_i=y$，则 $c^k\# c^l=c^k$; 如果对所有的 $i, c^k_i\# c^l_i=z$，则 $c^k\# c^l=\phi$; 否则 $c^k\# c^l=\bigcup_i\bigcup_{T\backslash c^l_i}(c^l_{n-1}\cdots c^l_{i+1}\alpha_i c^k_{i-1}\cdots c^k_0)$，其中 $\alpha_i=c^k_i\# c^l_i, \alpha_i\in\{T\backslash c^l_i\}$. #-积运算是

不可交换的.

表 1　定义四分形算符的坐标表

(a)

$\Rightarrow$ (c_i^k \ c_i^l)	0	1	2	3	x
0	ε	ϕ	ϕ	ϕ	ε
1	ϕ	ε	ϕ	ϕ	ε
2	ϕ	ϕ	ε	ϕ	ε
3	ϕ	ϕ	ϕ	ε	ε
x	ϕ	ϕ	ϕ	ϕ	ε

(b)

$\cap$ (c_i^k \ c_i^l)	0	1	2	3	x
0	0	Ω	Ω	Ω	0
1	Ω	1	Ω	Ω	1
2	Ω	Ω	2	Ω	2
3	Ω	Ω	Ω	3	3
x	0	1	2	3	x

(c)

$\#$ (c_k^i \ c_i^l)	0	1	2	3	x
0	z	y	y	y	z
1	y	z	y	y	z
2	y	y	z	y	z
3	y	y	y	z	z
x	α_i	α_i	α_i	α_i	z

四、目标操作

1. 连接

连接操作是指在同一整正方形内并具有相同尺寸 0-四分形的两个目标叠加. 首先将四分形之间的蕴含关系推广到四分形集合. 设C是表示目标Q的四分形集合，若有 c^k, $c^l\in C$，且 $c^k\Rightarrow c^l, c^k\neq c^l$，则 $C-c^k$ 也表示目标Q. 去掉C中所有这样的c^k之后得到C^s，显然C^s也表示目标Q. 从C求得C^s的运算称为求蕴含，记作 $C^s=s[C]$. 如果C是四个(同阶) P-四分形集合，对于任意的 c^k, $c^l\in C$，满足 $c_p^k\cap c_p^l=\Omega$ (见表 1(b)) 和 $c_i^k=c_i^l\,(n-1\geqslant i\geqslant p+1)$，则$C$可以压缩成一个 $(P+1)$-四分形 $c^m=c_{n-1}^k\cdots c_{p+1}^k xx\cdots x$. 用$c^m$代替$C$的运算称为压缩，记作 $c^m=D[C]$.

若C是两个目标C_1和C_2的连接，则 $C=D[S[C_1\cup C_2]]$. 两个目标的连接操作可以推广到多个目标.

2. 相交

该操作同样是指在同一整正方形内并具有相同尺寸 0-四分形的两个目标之交. 四分形之间的相交运算可以推广到集合: $a\cap B=\bigcup\limits_{b\in B}a\cap b$, $A\cap B=\bigcup\limits_{a\in A}a\cap B$. 两个目标$A$和$B$之交可以写成 $A\cap B=\bigcup\limits_{a\in A}\bigcup\limits_{b\in B}a\cap b$. 两个目标的相交操作同样可以推广到多个目标.

3. 取补

该操作是目标对其相关的整正方形求补. 首先将四分形之间的$\#$-积运算推广到集合: $a\# B=(\cdots((a\# b^{i_1})\# b^{i_2})\#\cdots)\# b^{i_m}$, $A\# B=\bigcup\limits_{a\in A}a\# B$，其中 $i_1, i_2, \cdots, i_m$ 是B中m个四分形的任意排列. 目标C对其U_n取补，记作C'，则 $C'=U_n\# C$.

五、数据结构

这一节要解决两个问题: (1)在计算机中如何表示四分形编码. 为了节省存储空间，

四分形编码的各分量存储在几个字内而不是一个分量占一个字，这样也便于四分形运算的并行操作，减少数据处理时间.（2）四分形算符不是通用计算机中的算术逻辑运算，为此必须解决将四分形算符转换成一般的算术逻辑运算.

1. 四分形编码的数据结构

假设机器是按字编址的 n 位字长，又设四分形 c^k 的编码为 n 个分量，既 $c^k = c^k_{n-1}\cdots c^k_{n-i}\cdots c^k_0$，$c^k_{n-i}\in\{0,1,2,3,x\}$，$1\leqslant i\leqslant n$. 由于 c^k_{n-i} 有五种可能的取值，表示 c^k_{n-i} 至少需要三个二进位. 为此用三个机器字 V^k、T^k_1 和 T^k_2 表示 c^k，其中 $V^k = v^k_{n-1}\cdots v^k_{n-i}\cdots v^k_0$，$T^k_1 = t^k_{1,n-1}\cdots t^k_{1,n-i}\cdots t^k_{1,0}$，$T^k_2 = t^k_{2,n-1}\cdots t^k_{2,n-i}\cdots t^k_{2,0}$. 并规定：

v^k_{n-i}	1	1	1	1	0
$t^k_{1,n-i}$	0	0	1	1	0
$t^k_{2,n-i}$	0	1	0	1	0
编码	0	1	2	3	x

可见 V^k 是 c^k 的特征函数，T^k_1 和 T^k_2 是 c^k 的真值表示.

2. 四分形算符的实现方法

1) $c^k \Rightarrow c^l$，当且仅当 $P_1(k,l) = V^k\cdot V^l = V^l$ 和 $P_2(k,l) = V^l\cdot[(T^k_1\oplus T^l_1)+(T^k_2\oplus T^l_2)] = \underline{0}$，其中："·"是逻辑与，"+"是逻辑或，"⊕"是逻辑异或，$\underline{0} = 00\cdots0$.

2) $c^k\cap c^l = \phi$，当且仅当 $p_3(k,l) = V^k\cdot V^l\cdot[(T^k_1\oplus T^l_1)+(T^k_2\oplus T^l_2)]\neq\underline{0}$；否则，若 $\|V^k\|\leqslant\|V^l\|$，则 $c^k\cap c^l = c^k$；若 $\|V^k\| > \|V^l\|$，则 $c^k\cap c^l = c^l$，其中 $\|A\|$ 表示 A 中 0 的数目.

3) 对于 #-积，有 $(a)c^k\# c^l = c^k$，当且仅当 $p_3(k,l)\neq\underline{0}$，$(b)c^k\# c^l = \phi$，当且仅当 $p_1(k,l) = V^l$ 和 $P_2(k,l) = \underline{0}$，(c) 否则，$c^k\# c^l = Uc^q$，其中各个 c^q 由下列步骤确定的 (V^q, T^q_1, T^q_2) 表示：

第一步：$0\rightarrow q$，$(\|V^k\| - \|V^l\|)\rightarrow M$；$(10\cdots0)\rightarrow P$

第二步：$3\rightarrow W$，如果 $P\cdot\bar{V}^k\cdot V^l = \underline{0}$ 转第五步.

第三步：$V^q = V^k + P$，$T^q_2 = T^k_2\oplus P\cdot\bar{T}^l_2$.

如果 $P\cdot T^q_2 = \underline{0}$，则 $T^q_1 = T^k_1\oplus P\cdot\bar{T}^l_1$；否则，$T^q_1 = T^k_1\oplus P\cdot T^l_1$.

第四步：$q+1\rightarrow q$，$W-1\rightarrow W$.

如果 $W\neq 0$，T^q_1、T^q_2 分别代替 T^l_1，T^l_2，并返回第三步；否则，V^q 代替 V^k，$P\cdot T^l_1 + T^k_1\rightarrow T^k_1$，$P\cdot T^l_2 + T^k_2\rightarrow T^k_2$，$M-1\rightarrow M$. 如果 $M = 0$，停止.

第五步：$P/2\rightarrow P$（P 右移一位），返回第二步.

可见，利用通用计算机很少几条指令可以实现大部分四分形算符的运算.

参 考 文 献

[1] Hunter, G. M. and Steiglitz, K., Linear Transformation of Pictures Represented by Quad Trees, Computer Graphics and Image Processing 10, 1979, 289—296.

[2] Hunter, G. M. and Steiglitz, K., Operations on Images Using Quadtrees, *IEEE Trans.* **PAMI-1**, 1979, 145—153.

THE APPLICATION OF FOUR VALUED LOGIC TO IMAGE PROCESSING

ZHANG TIANWEN　LI ZHONGRONG

(*Harbin Institute of Technology*)

ABSTRACT

In this paper, a quadrant structure for two- dimensional objects is described. The concept of quadrant order is presented. A direct procedure for encoding higher order quadrant is discussed. Operators associated with quadrants are provided, namely, the subsume relation, $\Rightarrow$, the intersect operator, $\cap$, and the # -product, #. Union, intersection, and complement of quadrants encoded objects are computed. Finally, the implementation of quadrant operators on general purpose computer as well as data structures for a quadrant are developed.