

改进的 TNAM 二值图像表示方法

方少红¹ 郑运平² 陈传波¹

(华中科技大学软件学院 武汉 430074)¹ (华南理工大学计算机科学与工程学院 广州 510006)²

摘要 提出了一种改进的 TNAM 二值图像表示方法,简称 ITNAM 表示方法。给出了 ITNAM 算法的形式化描述,并对其存储结构、总数据量和时空复杂性进行了详细的分析。以图像处理领域里惯用的标准‘Lena’,‘Baboon’和‘Peppers’等二值图像作为典型测试对象,对其进行了理论分析和实验,结果表明,与新提出的 TNAM 方法和经典的线性四元树表示方法相比,ITNAM 表示方法具有更少的子模式数(或节点数),能够更有效地减少数据存储空间,是二值图像模式的一种更优的表示方法。

关键词 二值图像表示,三角形,NAM,线性四元树

中图分类号 TP391 文献标识码 A

Binary Image Representation Method Using Improved TNAM

FANG Shao-hong¹ ZHENG Yun-ping² CHEN Chuan-bo¹

(College of Software Engineering, Huazhong University of Science and Technology, Wuhan 430074, China)¹

(College of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China)²

Abstract A binary image representation method using the improved TNAM (Triangle Non-symmetry and Anti-packing pattern representation Model), which is called the ITNAM representation method, was proposed. Also, a concrete algorithm of the ITNAM for binary images was presented and the storage structure, the total data amount, and the time and space complexities of the proposed algorithm were analyzed in detail. By taking some idiomatic standard images, such as ‘Lena’, ‘Baboon’ and ‘Peppers’, in the field of image processing as some typical test objects, and comparing the algorithm of the ITNAM with those of the classic linear quadtree (LQT) and the latest TNAM, the theoretical and experimental results show that the former is obviously superior to the latter with respect to the numbers of subpatterns or nodes and the data storage, and therefore it is a better method to represent binary images.

Keywords Binary image representation, Triangle, Non-symmetry and anti-packing model, Linear quadtree

1 引言

图像表示是计算机图形学、计算机视觉、机器人、图像处理和模式识别等领域里的一个重要问题^[1],图像表示方法的研究受到人们越来越多的重视^[2-4]。四元树表示是研究得最早的也是研究得最多的一种图像表示方法^[5]。早期的四元树表示都是基于指针的四元树结构。为了进一步减少存储空间,Gargantini 消除了指针方案,提出了线性四元树(LQT)表示方法^[6]。一般情况下,LQT 表示方法可节省 66% 的存储空间;特殊情况下,可节省高于 90% 的存储空间。借助于布局和填充问题的思想,文献[7]提出了一种基于非对称逆布局的模式表示模型(Non-symmetry and Anti-packing pattern representation Model,简称 NAM)的彩色图像表示方法,该方法与流行的基于 LQT 表示的彩色图像表示方法相比,不仅前者的矩形子模式数远小于后者的节点数,而且前者的总数据量也远小于后者的总数据量,因此前者能更有效地减少数据

存储空间,是彩色图像模式表示的一种良好方法。不同于文献[7]中的矩形子模式,文献[8]提出了一种基于 K-line 子模式的彩色图像表示方法。矩形子模式和 K-line 子模式比较适合具有一定块状性的图像。对于块状性不是很强或非块状性的图像,可以考虑用非块状类子模式,如三角形子模式。有鉴于此,文献[9]提出了一种三角形 NAM 的二值图像表示方法,简称 TNAM 表示方法,该方法对非块状类图像具有较强的适应性,是非块状类图像的一种高效表示方法。TNAM 表示方法直接对三角形子模式进行逆布局,且将三角形子模式分为两种类型,即上三角形和下三角形。事实上,在图像模式中,不仅客观上存在上三角形和下三角形子模式,还客观存在着对称上三角形和对称下三角形子模式,但在 TNAM 表示方法中没有考虑到后面两类三角形。若图像模式中考虑对称上三角形或对称下三角形,则对 TNAM 逆布局算法是不利的,因为表示一个对称上三角形或对称下三角形子模式需要至少两个上三角形或下三角形或线段或孤立点子模式。因

到稿日期:2009-05-05 返修日期:2009-07-16 本文受 863 国家重点基金项目(2006AA04Z211),国家自然科学基金(60873031)资助。

方少红(1968—),女,博士生,副教授,主要研究方向为计算机图形图像处理等,E-mail:flaviaf@263.net;郑运平(1979—),男,博士,主要研究方向为计算机图形图像处理与模式识别等;陈传波(1957—),男,博士,教授,博士生导师,主要研究方向为计算机网络与信息工程、图像处理与模式识别等。

此,尽管 TNAM 表示方法是二值图像的一种良好的表示方法,但其仍然有进一步的优化空间。

为了更有效地减少子模式数(节点数)和数据存储空间,针对 TNAM 表示方法,本文提出了一种改进的 TNAM 二值图像表示方法,简称 ITNAM 表示方法。ITNAM 表示方法相对于 TNAM 表示方法的改进之处主要表现在,将三角形子模式的类型从原来的两类(上三角形和下三角形)扩展成 4 类(上三角形、下三角形、对称上三角形和对称下三角形)。实验结果表明,与目前新提出的 TNAM 表示方法和经典的 LQT 表示方法相比,ITNAM 表示方法具有更少的子模式数(或节点数),能够更有效地减少数据存储空间,是二值图像模式的一种良好的表示方法。

2 ITNAM 表示方法描述

在 ITNAM 表示方法中,预先定义的子模式集合也是三角形。但为了进一步降低二值图像模式表示的总数据量,将三角形子模式分成了 4 类,即上三角形、下三角形、对称上三角形和对称下三角形。这样,对于任何一个三角形子模式,不需要存储 3 个顶点的坐标,只要存储斜边的两个端点和一个用于标识三角形子模式类型的标识符(比如上三角形、下三角形、对称上三角形和对称下三角形分别用“0”、“1”、“2”和“3”这 4 个数来标识)即可。相反,以斜边的两个端点和三角形类型的标识符也可以非常简单地解码出三角形子模式来。

图 1 说明了这种表示方法的思想及其相对于 TNAM 表示方法的优势。图 1 中白色点表示图像背景点的像素值为 1,黑色点表示图像区域的像素值为 0,且 TNAM 算法和 ITNAM 算法均仅需对黑色像素进行表示。

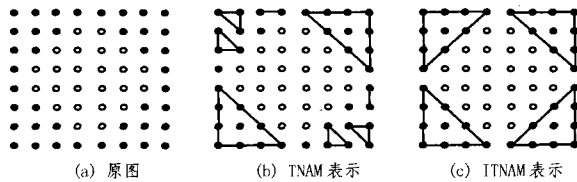


图 1 原始图像及其不同的表示方法

图 1(a)是给定的待逆布局的二值图像模式,其大小为 $2^n \times 2^n$ ($n=3$)。

图 1(b)是用 TNAM 算法^[9]对图 1(a)进行逆布局的结果(按光栅扫描的顺序寻找子模式的起始点)。在图 1(b)中,预先定义的子模式集合为 1 个三角形子模式。三角形子模式用 $t = \{\text{triangle} \mid \text{triangle} = (\text{flag}, \text{point1_hyp}, \text{point2_hyp})\}$ 来表示,其中 flag 为三角形子模式类型的标识符, point1_hyp 和 point2_hyp 表示斜边的 2 个端点。由于形成 1 个三角形需要 3 个点,因此在 TNAM 逆布局结果中也有可能会出现线段和孤立点的情况。从文献^[9]的分析可知,存储 1 个三角形记录占 $2n+1$ 位,1 条线段记录占 $2n$ 位,1 个孤立点记录占 n 位。由于在图 1(b)中共有 3 个上三角形、3 个下三角形、2 条线段和 4 个孤立点子模式,因此总共需要 $(2n+1) \times (3+3) + 2n \times 2 + n \times 4 = 42 + 12 + 12 = 66$ 位来表示图 1(a)。

图 1(c)是用 ITNAM 算法对图 1(a)进行逆布局的结果(按光栅扫描的顺序寻找子模式的起始点)。在图 1(c)中,预先定义的子模式集合为 1 个三角形子模式。这里的三角形子模式也是带参数的,即用 $t = \{\text{triangle} \mid \text{triangle} = (\text{flag}, \text{point1_hyp}, \text{point2_hyp})\}$ 表示 1 个三角形子模式,其中 flag 为三角形子模式类型的标识符, point1_hyp 和 point2_hyp 表示斜边的 2 个端点。从下面的分析可知,存储 1 个三角形记录

占 $2n+2$ 位,1 条线段记录占 $2n$ 位,1 个孤立点记录占 n 位。由于在图 1(c)中共有 1 个上三角形、1 个下三角形、1 个对称上三角形、1 个对称下三角形子模式,因此总共需要 $(2n+2) \times (1+1+1+1) = 8 \times 4 = 32$ 位来表示图 1(a)。

综上所述,图 1(a)用 TNAM 和 ITNAM 表示时所需的子模式总数分别为 12 个和 4 个,各自所占的总存储空间分别为 66 位和 32 位。因此,与 TNAM 表示方法相比,ITNAM 表示方法能够更有效地减少子模式的数量且能够更有效地节省存储空间。

由这个例子可以看出,逆布局的方式不是唯一的,不同的划分(逆布局)方法表示效率是不一样的,因此逆布局算法的优化是必要的。显然,ITNAM 的表示能力要优于 TNAM 的表示能力。

3 ITNAM 表示的编解码算法

ITNAM 表示的编解码算法中用到了 K 码变换规则^[7-9],因此在描述算法之前,有必要对 K 码变换规则进行简单介绍。

3.1 K 码变换规则

设已经布局好了的图像模式用 $F = \{f(x, y)\}$ 来表示,假定 $F = \{f(x, y)\}$ 为二值图像,即 $f(x, y) \in \{0, 1\}$,并且 $F = \{f(x, y)\}$ 的大小为 $2^n \times 2^n$ 。为方便起见,假定“1”为“white”,即白色;“0”为“black”,即黑色。白色表示背景点,黑色表示区域。

令 $x = (x_{n-1} x_{n-2} \dots x_1 x_0)_2$ 和 $y = (y_{n-1} y_{n-2} \dots y_1 y_0)_2$,构造 1 个一维的坐标变量 k ,有 $k = (y_{n-1} x_{n-1} y_{n-2} x_{n-2} \dots y_1 x_1 y_0 x_0)_2$ 。这样就将 1 个二维图像模式转换为 1 个保持块状性质的一维序列,即 $F = \{f(x, y)\} = \{g(k)\}$,简称这种一维的表示为 K 码。这样做的好处在于子模式的块状性质会更好,可以大大减少子模式的数量。由二维到一维的降维变换称为 K 码正变换,记为 $k = K(x, y)$ 。由一维到二维的升维变换称为 K 码的反变换,记为 $(x, y) = K^{-1}(k)$ 。 K 码的正变换和反变换都是二进制数按位拼接过程。

3.2 ITNAM 表示算法

本算法中被逆布局的子模式对象是任意大小的三角形 $t = \{\text{triangle} \mid \text{triangle} = (\text{flag}, \text{point1_hyp}, \text{point2_hyp})\}$,其中 flag 是三角形子模式类型的标识符,当 $\text{flag}=0$ 时为上三角形,当 $\text{flag}=1$ 时为下三角形,当 $\text{flag}=2$ 时为对称上三角形,当 $\text{flag}=3$ 时为对称下三角形; point1_hyp 和 point2_hyp 表示三角形斜边的 2 个端点坐标。由于形成 1 个三角形需要 3 个点,因此在 ITNAM 算法的逆布局结果中除了三角形外,还可能会出现线段和孤立点的情况。

ITNAM 编码算法的具体步骤如下。

Input: 一幅 $2^n \times 2^n$ 的二值图像 f 。

Output: $Q = \{Q_{tri}, Q_{line}, Q_{point}\}$,其中 Q_{tri} , Q_{line} 和 Q_{point} 分别表示三角形、线段和孤立点的编码队列集合。

Step 1 将三角形、线段和孤立点的计数变量 tri_num , $line_num$ 和 $point_num$ 均赋值为 0。

Step 2 按光栅扫描的顺序,从标记过的二值图像 f 的第一个入口开始,首先确定一个未被标识的三角形子模式。

Step 3 根据找到的三角形子模式的类型,将 flag 赋为相应的值,即如果是上三角形,则令 $\text{flag}=0$;如果是下三角形,则令 $\text{flag}=1$;如果是对称上三角形,则令 $\text{flag}=2$;如果是对称下三角形,则令 $\text{flag}=3$ 。最后将这个三角形子模式

在 f 中做标识。

Step 4 将三角形的计数变量 tri_num 的值加 1, 记录斜边端点的 2 个坐标 (x_1, y_1) 和 (x_2, y_2) , 然后对斜边的 2 个端点做 K 码降维变换, 即 $point1_hyp \leftarrow K(x_1, y_1)$, $point2_hyp \leftarrow K(x_2, y_2)$ 。最后将 $flag, point1_hyp$ 和 $point2_hyp$ 这 3 个变量存储到队列 Q_tri 中, 即有 $Q_tri\{tri_num\} = \{flag, point1_hyp, point2_hyp\}$ 。

Step 5 循环执行 Step 2 到 Step 4, 直到不能形成新的三角形子模式为止。

Step 6 按光栅扫描的顺序, 从标记过的二值图像 f 的第 1 个入口开始, 首先确定 1 个未被标记的点, 再根据子模式的匹配(逆布局)算法来尽可能地形成最长的线段。如果能形成线段, 则将线段的计数变量 $line_num$ 的值加 1, 记录线段端点的 2 个坐标 (x_1, y_1) 和 (x_2, y_2) , 然后对线段的 2 个端点做 K 码降维变换, 即 $point1_line \leftarrow K(x_1, y_1)$, $point2_line \leftarrow K(x_2, y_2)$ 。最后将 $point1_line$ 和 $point2_line$ 这 2 个变量存储到队列 Q_line 中, 即有 $Q_line\{line_num\} = \{point1_line, point2_line\}$, 且将存储过的此线段在 f 中做标识。否则, 说明只能形成孤立点, 执行 Step 7。

Step 7 记录孤立点的坐标 (x, y) , 将孤立点的计数变量 $point_num$ 的值加 1, 然后将这个孤立点做 K 码降维变换, 即: $single_point \leftarrow K(x, y)$ 。最后将变量 $single_point$ 存储到队列 Q_point 中, 即有 $Q_point\{point_num\} = \{single_point\}$, 并将此点在 f 中做标识。

Step 8 循环执行 Step 6 到 Step 7, 直到不能形成新的线段和孤立点为止。

Step 9 输出二值图像 f 的编码队列集合 $Q = \{Q_tri, Q_line, Q_point\}$ 。

解码算法是编码算法的逆过程, 相对于模式逆布局算法来说要简单得多, 只要顺序地从队列 Q 中读取元素并逐一解码即可。

3.3 ITNAM 表示算法的复杂度分析

假定二值图像 f 的规模为 M , 即 f 中元素的总数为 M 。对基于 ITNAM 的二值图像表示算法来说, 编码所需的时间正比于 $M\xi$, 其中 M 是二值图像的像素数, ξ 表示图像中每个像素平均分割的次数, 且 ξ 的上限为 $O(\log M)$ 。因此, 在最坏情况下编码算法时间复杂度为 $O(M \log M)$ 。解码所需的时间正比于二值图像的像素数 M 。因此, 解码算法的时间复杂度为 $O(M)$ 。

在空间开销方面, 编码算法除原图像矩阵外只增加了为数非常少的中间变量, 因而其空间复杂度与图像的大小成正比, 即 $O(M)$; 解码算法除了原图像矩阵外只需增加几个缓存变量, 因而其空间复杂度也与图像的大小成正比, 即 $O(M)$ 。

4 ITNAM 表示的存储结构和数据量分析

4.1 ITNAM 表示的存储结构

与 TNAM 算法的输出一样, ITNAM 算法的输出 Q 也是 3 个编码队列的集合, 即 $Q = \{Q_tri, Q_line, Q_point\}$ 。且从文献[9]易知, 对于 TNAM 表示方法来说, 存储 1 个三角形记录占 $2n+1$ 位, 1 条线段记录占 $2n$ 位, 1 个孤立点占 n 位。ITNAM 表示的存储结构除了三角形与 TNAM 表示的不一致外, 线段和孤立点的存储结构是一样的, 因此本节将重点讨论三角形的存储结构。

对于三角形队列集合 Q_tri 来说, ITNAM 算法的输出是

1 个顺序存储的队列, 它主要由 3 个元素组成, 即三角形子模式的类型标识符 $flag$ 以及斜边的 2 个端点坐标 $point1_hyp$ 和 $point2_hyp$ 。由于算法中三角形子模式被分成了 4 种类型, 即上三角形、下三角形、对称上三角形和对称下三角形, 因此存储 $flag$ 需占 2 位。此外, 斜边的 2 个端点的坐标 $point1_hyp$ 和 $point2_hyp$ 也用 K 码来表示, 按照 K 的定义^[7], 存储这两个端点占 $2n$ 位, 因此, 存储一个三角形子模式占 $2n+2$ 位。其存储结构如图 2 所示。

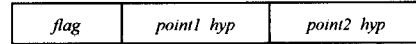


图 2 三角形的存储结构

4.2 ITNAM 表示的数据量

设二值图像模式的大小为 $2^n \times 2^n$, 逆布局后的三角形、线段、孤立点的子模式数分别为 N_t, N_l 和 N_p , 其总数据量记为 H_{ITNAM} 。设 LQT 表示的黑色节点数为 N_{LQT} , 总数据量为 H_{LQT} 。

对于 ITNAM 表示方法来说, 存储 1 个三角形记录占 $2n+2$ 位, 1 条线段记录占 $2n$ 位, 1 个孤立点占 n 位。则其逆布局后的总数据量 H_{ITNAM} 为:

$$H_{ITNAM} = (2n+2)N_t + 2nN_l + nN_p \quad (1)$$

对于 LQT 表示方法来说, 存储一个节点占 $3(n-1)+2$ 位^[6], 则 LQT 表示的总数据量 H_{LQT} 为:

$$H_{LQT} = (3n-1)N_{LQT} \quad (2)$$

设 $\varphi_{LQT, ITNAM}$ 为 LQT 的总数据量与 ITNAM 的总数据量的比值, 则有:

$$\varphi_{LQT, ITNAM} = \frac{(3n-1)N_{LQT}}{(2n+2)N_t + 2nN_l + nN_p} > \frac{(3n-1)N_{LQT}}{(2n+2)(N_t + N_l + N_p)} \quad (3)$$

通过式(3)可以比较 LQT 表示相对于 ITNAM 表示的优劣。由于 LQT 表示是对称形的分割, 其分割方式受到很大的限制, 而 ITNAM 表示是非对称分割, 分割是最大限度地形成子模式, 且分割的灵活性更大。一般情况下, ITNAM 表示逆布局后的子模式总数比 LQT 表示的节点数少得多, 即 $N_t + N_l + N_p < N_{LQT}$, 因此 $\varphi_{LQT, ITNAM} > (3n-1)/(2n+2)$ 。比如, 当 $n=8$ 时, $\varphi_{LQT, ITNAM} > 1.2778 > 1$, 即 LQT 表示的总数据量至少是 ITNAM 表示总数据量的 1.2778 倍以上。

因此, 与经典的 LQT 表示方法相比, 从理论上来说, 二值图像的 ITNAM 表示方法在子模式数和存储空间上具有明显的优势。

5 实验与分析

为了验证 ITNAM 表示方法的理论结果, 本节从实验的角度来说明其相对于 TNAM 表示方法和经典的 LQT 表示方法的明显优势。实验中用来测试的二值图像模式的分辨率参数 n 为 8, 即图像模式的大小为 $2^8 \times 2^8$ (如图 3 所示)。这些图像的复杂度^[7]和纹理各不相同, 具有较好的代表性, 说明了 ITNAM 表示方法的适应性。实验中的机器配置: CPU 为 Celeron(R)2.4 GHz, 内存为 Kingston DDR 2 GB, OS 为 MS-Windows XP Service Pack 2。编程环境为 Matlab 6.5。

通过编程, 分别实现了 ITNAM, TNAM 及 LQT 表示算法, 并对这 3 种算法的实验结果进行了比较。相应的比较数据如表 1 所列, 其中 $Image$: 二值图像的名称; CF : 二值图像的复杂度; N : 子模式或节点个数; $TNAM$: 三角形 NAM 表示; $ITNAM$: 改进的 TNAM 表示; LQT : 线性四元树表示; D :

TNAM 与 ITNAM 的子模式数之差; ϕ_{LQT_TNAM} : LQT 与 TNAM 的总数据量之比; ϕ_{LQT_ITNAM} : LQT 与 ITNAM 的总数据量之比。

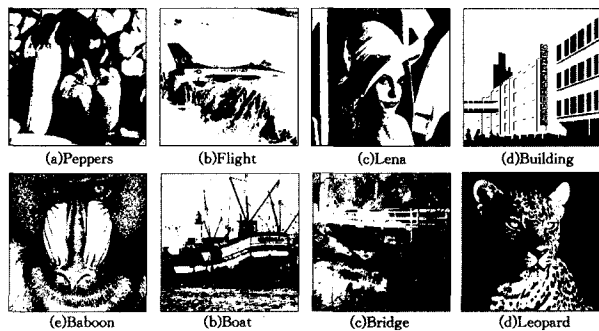


图3 8幅二值图像

表1给出了LQT, TNAM和ITNAM的子模式数(节点数)。从实验数据来看, TNAM和ITNAM在数据量方面的效果均是非常明显的, 其子模式数均小于LQT方法的节点数, 对图像的适应性也非常好。而且从表1中D的值可知, ITNAM的子模式数比TNAM的子模式还要少178~751个。因此, 与LQT和TNAM方法相比, ITNAM方法能够更有效地减少子模式的数量。

表1 LQT, TNAM和ITNAM的性能比较

Image	CF	N			D	ϕ_{LQT_TNAM}	ϕ_{LQT_ITNAM}
		LQT	TNAM	ITNAM			
Peppers	0.0856	5608	2994	2816	178	2.9916	3.1257
Flight	0.0850	5568	2526	2252	274	3.5287	3.8772
Lena	0.0981	6430	3414	2993	421	2.9902	3.3531
Building	0.0702	4602	1692	1493	199	4.0114	4.4477
Baboon	0.2601	17047	9112	8361	751	3.1227	3.2973
Boat	0.1313	8604	4041	3701	340	3.3743	3.5677
Bridge	0.1813	11881	6645	6086	559	2.8796	3.0566
Leopard	0.1059	6938	3969	3668	301	2.8103	2.9717

表1也给出了LQT与ITNAM的总数据量之比 ϕ_{LQT_ITNAM} 。对于给定的8幅图像而言, LQT的总数据量是ITNAM的2.9717~4.4477倍。显然, 这些图像均证实了理论分析的结果, 即当 $n=8$ 时, $\phi_{LQT_ITNAM} > 1.2778 > 1$ 。并且从表1中也不难看出, ϕ_{LQT_ITNAM} 总是大于 ϕ_{LQT_TNAM} , 这表明在数据存储表示方面, ITNAM比LQT和TNAM方法能够更有效地减少数据存储空间。

综上所述, 理论分析和实验结果均表明, 与LQT和TNAM方法相比, ITNAM方法能够更有效地减少子模式数(节点数)和数据存储空间。

结束语 NAM优化策略的最终目标是不断地逼近模式的最优化表示, NAM优化问题在降低存储空间、提高传输速度、加快处理过程、模式匹配等方面具有良好的理论参考意义和实际应用价值^[10,11]。本文提出了一种改进的TNAM二值图像表示方法, 简称ITNAM表示方法。给出了ITNAM算法的形式化描述, 并对其存储结构、总数据量和时空复杂性进行了详细的分析。理论分析和实验结果均表明, 与新提出的TNAM方法和经典的线性四元树表示方法相比, ITNAM表示方法具有更少的子模式数(或节点数), 能够更有效地减少数据存储空间, 是二值图像模式的一种更优的表示方法。

参考文献

- [1] Pennec E L, Mallat S. Sparse geometric image representations with bandelets[J]. IEEE Transactions on Image Processing, 2005, 14(4): 423-438
- [2] Gan T, He Y, Zhu W. Fast M-term pursuit for sparse image representation[J]. IEEE Signal Processing Letters, 2008, 15(1): 116-119
- [3] He Z. Peak transform for efficient image representation and coding[J]. IEEE Transactions on Image Processing, 2007, 16(7): 1741-1754
- [4] Tang F, Crabb R, Tao H. Representing images using nonorthogonal haar-like bases[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007, 29(12): 2120-2134
- [5] Samet H. The quadtree and related hierarchical data structures[J]. Computing Surveys, 1984, 16(2): 187-260
- [6] Gargantini I. An effective way to represent quadtrees[J]. Comm. ACM, 1982, 25(12): 905-910
- [7] 郑运平, 陈传波. 一种基于非对称逆布局模型的彩色图像表示方法[J]. 软件学报, 2007, 18(11): 2932-2941
- [8] Zheng Y P, Chen C B, Huang W, et al. A new non-symmetry and anti-packing model with K-lines and its application to color image representation[C]// Proceedings of the 2007 International Conference on Image Processing, Computer Vision, and Pattern Recognition (ICCV'07). Las Vegas: CSREA Press, 2007: 526-532
- [9] 郑运平, 陈传波, 黄巍. 一种新的基于TNAM的二值图像表示方法[J]. 计算机科学, 2008, 35(11): 220-224
- [10] 郑运平, 陈传波. 基于光栅扫描的NAM优化策略[J]. 华中科技大学学报, 2008, 36(8): 1-4
- [11] Zheng Y P, Chen C B, Sarem M. An improved algorithm for gray image representation using non-symmetry and anti-packing model with triangles and rectangles[J]. Frontiers of Computer Science in China, 2008, 2(4): 431-437
- [2] Moghaddam B, Yang M H. Gender classification with support vector machines[J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2002, 24(5): 707-711
- [3] 武勃, 艾海舟, 肖习攀, 等. 人脸的性别分类[J]. 计算机研究与发展, 2003, 40(11): 1546-1553
- [4] Lian Hui-cheng, Lu Bao-liang. Multi-view gender classification using local binary pattern and support vector machine[J]. Lecture Notes in Computer Science, 2006, 3972: 202-209
- [5] Hall M. Correlation-based Feature Subset Selection for Machine Learning[D]. Department of Computer Science, University of Waikato, 1999
- [6] Brill F, Brown D, et al. Fast genetic selection of features for neural network classifiers[J]. IEEE Transaction on Neural Networks, 1992, 3(2): 324-328
- [7] Bartlett M S. Face image analysis by unsupervised learning and redundancy reduction[D]. University of California San Diego, 1998
- [8] Hyvarinen A, Oja E, et al. Independent Component Analysis: Algorithms and Applications[J]. Neural Networks, 2000, 13(4/5): 411-430
- [9] Hyvarinen A, Oja E. A fast fixed-point algorithm for independent component analysis[J]. Neural Computation, 1997, 9(7): 1483-1492
- [10] Holland J. Adaptation in Natural and Artificial Systems[M]. MIT Press, 1992
- [11] Gao Wen, Cao Bo, Shan Shiguang, et al. The CAS-PEAL Large-Scale Chinese Face Database and Baseline Evaluations[R]. Technical report of JDL, 2004

(上接第200页)