一种改进的基于 NAM 的彩色图像表示算法*)

郑运平　陈传波　黄　巍
(华中科技大学计算机科学与技术学院　武汉 430074)

摘　要　图像表示在机器人、图像处理、模式识别等领域里是一个非常重要的研究内容之一。尽管基于 NAM 的彩色图像表示方法是彩色图像模式的一种良好表示方法，但是该方法是建立在单类型子模式(矩形)基础之上，因而还有更进一步的优化空间。通过对多子模式类型的组合(矩形和三角形)逆布局作进一步的研究，提出了一种改进的基于 NAM 的彩色图像表示算法，并给出了该算法的存储结构和数据量分析。理论分析和实验结果表明：与基于 NAM 的彩色图像表示算法和流行的基于线性四元树的彩色图像表示方法相比，改进的基于 NAM 的彩色图像表示方法能更有效地减少子模式数(节点数)和数据存储空间，是彩色图像模式表示的一种更优的表示算法，为逐步逼近彩色图像模式的最优化表示提供了一种重要的研究途径。

关键词　图像表示，彩色图像，NAM，INAM，线性四元树，逆布局问题，图像复杂度

Improved Algorithm for NAM-based Color Image Representation

ZHENG Yun-ping　CHEN Chuan-bo　HUANG Wei
(School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China)

Abstract　The image representation is one of the most important research contents in robotics, image processing, pattern recognition, and so on. Although a NAM-based color image representation method is a better method to represent the color image pattern, the method is only based on single subpatterns (rectangles) and there is still much room left to optimize the method. Therefore, instead of single subpatterns (rectangles), by adopting multi-subpatterns (such as some combination of rectangles and triangles), an improved algorithm for NAM-based color image representation was presented. Also, the storage structures and the total data amount of the proposed algorithm were analyzed. The theoretical and experimental results presented show that the proposed algorithm can greatly reduce the numbers of subpatterns or nodes and simultaneously save the storage room much more effectively than the algorithms of the popular linear and the NAM with single rectangular subpatterns. Therefore, the proposed algorithm is a better method to represent the color image pattern and it provides an important research approach for approximating the optimal representation of the color image pattern.

Keywords　Image representation, Color image, NAM, INAM, Linear quadtree, Anti-packing problem, Image complexity

1　引言

图像表示在机器人、图像处理、模式识别等领域里是一个非常重要的研究内容之一[1-3]。在 Internet 已成为最主要的信息传输途径的今天，由于图像信息所具有的大量性，其快速、实时传输的要求已得不到满足，它成为制约 Internet 发展的一个难题。许多实际的应用由于大量的图像信息得不到快速的传输而使系统的实时效果得不到满足。因此图像表示的研究就变得非常重要，它是目前最活跃的研究领域之一。传统的、直观的区域表示方法是二维数组表示。这样，二维数组的空间效率及其"样点-样点"的运算方式已不能适应发展的需要。取而代之的应该是既紧凑又便于做各种图像处理运算的表示方法。为此，人们相继提出了边界链码[4-8]，行程编码[9-11]等表示方法。这些表示方法虽然具有一定的紧凑性，但一些基本图像处理运算却变得难以实现[12,13]。例如，边界链码很难做区域的交、并运算；行程编码难以确定区域边界特征等等[14-16]，鉴于以上方法存在的不足，人们提出了更多的图像表示方法[17-22]。

四元树是图像分层表示的一种形式，它是基于图像数组的四元分割。四元树表示是研究得最早的，也是研究得最多的一种分层表示形式[23-25]。早期的四元树表示都是基于指针的四元树结构，为了进一步减少存贮空间，Gargantini 消除了指针方案，提出了称之为线性四元树的表示方法[26]。一般情况下，线性四元树可节省 66%的存贮空间；特殊情况下，可节省高于 90%的存贮空间。线性四元树的建立，在四元树表示的历史中是一个大的转折。然而，尽管线性四元树表示有许多优点，但是它们过于强调分割的对称性，因此不是最优的表示方法。借助于 Packing 问题的思想，以寻找分割最大化的非对称分割方法为目标，文献[27]提出了一种基于非对称逆布局的模式表示模型(NAM)的彩色图像表示方法，该方法与流行的基于线性四元树的彩色图像表示方法相比，不仅前者的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。另外，在医学图

*) 基金项目：国家高技术研究发展计划(863)(2006AA04Z211)。**郑运平**　博士研究生，主要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，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；**黄　巍**　博士研究生，主要方向为计算机图形图像处理。

像模式的表示方面，文献[28]提出了一种医学图像的直接NAM表示方法并同时给出了一些基于直接NAM的图像操作和计算算法，该医学图像的直接NAM表示方法在数据表示和操作上均优于线性四元树的表示方法，是医学图像模式表示的一种有效方法。然而文献[27,28]的图像模式表示方法均是基于矩形子模式的，这两种表示方法比较适合图像具有一定的块状性，事实上，对于图像的块状性不是很强或非块状类图像，可以考虑用非块状类子模式，比如：三角形子模式，有鉴于此，文献[29]提出了一种新的三角形非对称逆布局的模式表示模型(TNAM)的灰度图像表示算法，该算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算法。

然而，尽管文献[27-29]针对NAM做了较为深入的研究，但是这些结果还是建立在单类型子模式(矩形或者三角形)基础之上的，即使是那些情况下也能充分证明NAM表示方法的效能，对于多子模式类型，文献[27]在展望部分指出：对于多子模式类型(如矩形与三角形等的组合)，作者预期能产生更少的子模式数并具有更强的数据表示及操作能力。因此，多子模式类型的组合逆布局是需要进一步研究的内容，其结果将会使得基于NAM的彩色图像表示方法更加具有生命力。因此，本文的工作就是对这一部分内容继续研究，以更进一步地减少子模式的数量和数据存储空间，逐步逼近模式的最优化表示，而这也正是NAM表示的最终目的。

为叙述方便起见，在本文，基于NAM的彩色图像表示算法[27]简称为NAM算法，而改进的NAM算法则简称为INAM算法。

本文第2节分析了NAM算法的局限性及其改进策略，提出了一种INAM算法。第3节对INAM算法作了具体的描述。第4节分析了INAM算法的存储结构。第5节分析并计算了INAM算法的总数据量。第6节从实验的角度来说明了INAM算法相对于NAM算法和流行的线形四元树算法的明显优势。最后是结论。

2 NAM算法的局限性及其改进策略

在分析NAM算法的局限性及其改进策略之前，我们先对NAM的思想作一下简单的描述。

2.1 NAM的思想描述

NAM是Non-symmetry and Anti-packing pattern representation Model(非对称逆布局的模式表示模型)的简称，它是一种全新的模式表示方法，其基本思想是：给定一个已经布局好的容器(模式)和n个预先定义的不同形状的物体(n个子模式)，现在要从这个布局好了的容器(给定的模式)中抽出这些物体(子模式)，用这些物体(子模式)的组合来表示已布局的容器(给定的模式)。

2.2 NAM算法的局限性分析

由于NAM算法本质上是先将一幅彩色图像(灰度级参数为m)经BPD分解为$3m$幅二值图像，然后对每一幅二值图像进行逆布局，因此下面以二值图像(如图1所示)为例来说明NAM算法改进的可能性和必要性。在图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，且线性四元树算法、NAM算法、INAM算法均仅需对黑色像素进行表示。

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)是用流行的线性四元树算法对图1(a)进行表示的结果。由于存贮一个黑色节点块占$3(n-1)+2$位[26]，且图1(b)共有31个黑色节点块，因此用线性四元树表示方法表示图1(a)时总共需要$(3(n-1)+2)\times 31=8\times 31=248$位。

图1(c)是用NAM算法对图1(a)进行逆布局结果。在图1(c)中，预先定义的子模式集合为一个矩形子模式$p=\{rectangle \mid rectangle=(sp,length,width)\}$，其中$sp$是矩形左上角的坐标，$length$和$width$分别代表该矩形的长与宽(孤立点是长和宽同时为1的矩形，线段是长或宽为1且长和宽不同时为1的矩形)。由于存贮一个矩形记录占$2n$位[27]，且在图1(c)中共有12个矩形，也即总共需要$2n\times 12=6\times 12=72$位来表示图1(a)。

图1(d)是用INAM算法对图1(a)进行逆布局结果。在图1(d)中，预先定义的子模式集合为一个矩形和一个三角形子模式。矩形子模式还是用$p=\{rectangle \mid rectangle=(sp,length,width)\}$来表示，但是这里的矩形不包括线段和孤立点，也即$length$和$width$必须同时大于或等于2。这里的三角形子模式也是带参数的，即用$P=\{triangle \mid triangle=(flag,point1_hyp,point2_hyp)\}$表示一个三角形子模式，其中$flag$为三角形子模式类型的标识符，$point1_hyp$和$point2_hyp$表示斜边的两个端点。由于形成一个矩形至少需要4个点而形成一个三角形至少需要3个点，因此在INAM逆布局结果中也有可能会出现线段和孤立点的情况。从下文第4节的分析可知，存贮一个矩形记录占$2n$位，一个三角形记录占$2n+2$位，一条线段记录占$2n$位，一个孤立点记录占n位。由于在图1(d)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和4个矩形子模式，所以总共需要$(2n+2)\times(1+1+1+1)+2n\times 4=8\times 4+6\times 4=56$位来表示图1(a)。

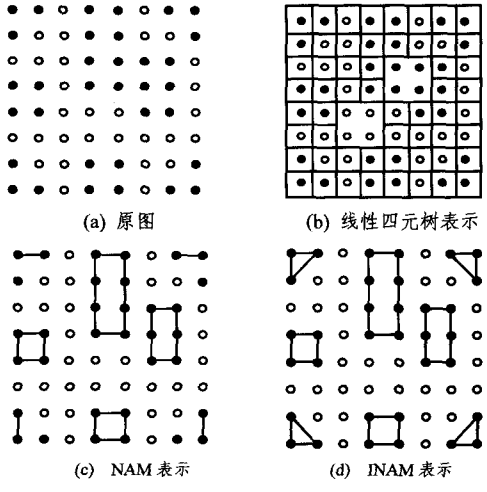

图1 原始图像及其用不同表示方法的结果

2.3 NAM算法的改进策略

由2.2节的例子可以看出，逆布局的方式不是唯一的，不同的划分(逆布局)方法，其表示效率是不一样的，显然，INAM算法的表示能力要优于线性四元树算法和NAM算法的表示能力。

因此，鉴于NAM算法的局限性，本文提出了一种改进的NAM算法，也即INAM算法。在INAM算法中，预先定义的子模式集合是矩形和三角形，而且，为了进一步降低彩色图像模式表示的总数据量，我们将三角形子模式分成了四类，

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，不需要存储三个顶点的坐标，而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。

3 INAM算法描述

由于在INAM算法中用到了K码变换规则和灰度图像的BPD性质定理[27]，因此在描述算法之前，有必要对两者进行简单的介绍。

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，并且 $F=\{f(x,y)\}$ 的大小为 $2^n\times 2^n$。

令 $x=(x_{n-1}x_{n-2}\cdots x_1x_0)_2$ 和 $y=(y_{n-1}y_{n2}\cdots y_1y_0)_2$，构造一个一维的坐标变量 k，有：$k=(y_{n-1}\ x_{n-1}\ y_{n-2}\ x_{n-2}\cdots y_1x_1y_0x_0)_2$。

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 $(x,y)=K^{-1}(k)$。

3.2 灰度图像的BPD性质定理

定理1 给定任意具有 $K=2^m(m>1)$ 个灰级的灰度图像 MP，存在 m 个二值图像 $BP_0,BP_1,\cdots,BP_{m-1}$，满足下面的关系：

$$MP=\sum_{i=0}^{m-1}2^iBP_i$$

其中 $BP_i(0\leqslant i\leqslant m-1)$ 是从灰度图像 MP 中灰级值的第 i 个二进制位平面复制而得到的二值图像，并规定：如果第 i 位为0（或1），则 BP_i 对应的那个像素值也为0（或1）。

3.3 INAM算法描述

INAM算法中被逆布局的子模式对象是任意大小的矩形 $p=\{rectangle\mid rectangle=(sp,length,width)\}$ 和三角形 $p=\{triangle\mid triangle=(flag,point1\_hyp,point2\_hyp)\}$。设已经布局好的的彩色图像模式 CIP 的大小为 $2^n\times 2^n\times 3$，灰度级别参数为 m，其分解后的3幅灰度图像模式为 $MP[1]$，$MP[2]$和 $MP[3]$，大小均为 $2^n\times 2^n$。

为方便起见，在位平面二值图像中，假定“0”为“*black*”，即黑色，“1”为“*white*”，即白色。黑色表示区域，白色表示背景点。本算法只记录“*black*”像素点。由于这种算法是可逆的，因而给出了相应的解码算法。下面将分别给出编解码算法的具体步骤。

编码算法的具体步骤：

/*给定一个 $2^n\times 2^n\times 3$ 的彩色图像模式 CIP 以及其灰度级别参数 m，将编码结果存入4个队列集合 $Q\_rect=\{Q\_rect_0,Q\_rect_1,\cdots,Q\_rect_{3m-1}\}$，$Q\_tri=\{Q\_tri_0,Q\_tri_1,\cdots,Q\_tri_{3m-1}\}$，$Q\_line=\{Q\_line_0,Q\_line_1,\cdots,Q\_line_{3m-1}\}$ 和 $Q\_point=\{Q\_point_0,Q\_point_1,\cdots,Q\_point_{3m-1}\}$ 中。*/

Step 1 对于一个给定的大小为 $2^n\times 2^n\times 3$ 的彩色图像模式 CIP，分别取得它的 r,g,b 颜色分量组成的大小为 $2^n\times 2^n$ 的灰度图像模式 $MP[1]$，$MP[2]$和 $MP[3]$，并把矩形、三角形、线段和孤立点的计数变量 *rect_num*，*tri_num*，*line_num* 和 *single_point* 均赋值为0。

Step 2 用灰度图像的BPD方法依次将3幅灰度图像模式 $MP[1]$，$MP[2]$和 $MP[3]$ 各自分解为 m 幅二值图像 $BP_i(0\leqslant i\leqslant m-1)$，$BP_i(m\leqslant i\leqslant 2m-1)$ 和 $BP_i(2m\leqslant i\leqslant 3m-1)$，并将这 $3m$ 幅二值图像 $BP_i(0\leqslant i\leqslant 3m-1)$ 连续存储。最后将位面号 i 赋值为0。

Step 3 按光栅扫描的顺序，从二值图像模式 BP_i 的第一个入口开始，首先确定一个未被标识的矩形子模式的起始点 (x,y)，再根据矩形子模式的匹配（逆布局）算法来追迹相应的矩形子模式。

Step 4 根据矩形子模式的效率尺度（即：矩形子模式的面积）确定一个面积最大的矩形子模式，并将这个最大的矩形子模式在二值图像模式 BP_i 中作标识，以便下一个起始点的寻找。

Step 5 将矩形子模式的计数变量 *rect_num* 的值加1，并记录此最大矩形子模式的3个参数，即：起始点坐标 (x,y)、长度值 *length* 和宽度值 *width*，然后对起始点坐标 (x,y) 作K码降维变换，即：$sp\leftarrow K(x,y)$。最后将 *sp*，*length* 和 *width* 这3个变量存储到队列 Q_rect_i 中，即有 $Q\_rect_i\{rect\_num\}\leftarrow\{sp,length,width\}$。

Step 6 循环执行 *Step* 3 到 *Step* 5，直到不能形成新的矩形子模式为止。

Step 7 按光栅扫描的顺序，从标记过的二值图像 BP_i 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（这里三角形特指图1(d)中提到的4种三角形，也即此三角形包含的像素总数为3）。

Step 8 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 $flag=0$，如果是下三角形则令 $flag=1$，如果是对称上三角形则令 $flag=2$，如果是对称下三角形则令 $flag=3$。最后将这个三角形子模式在 BP_i 中作标识。

Step 9 将三角形的计数变量 *tri_num* 的值加1，记录斜边端点的两个坐标 (x_1,y_1) 和 (x_2,y_2)，然后对斜边的两个端点作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_i 中，即有 $Q\_tri_i\{tri\_num\}=\{flag,point1\_hyp,point2\_hyp\}$。

Step 10 循环执行 Step 7 到 Step 9，直到不能形成新的三角形子模式为止。

Step 11 按光栅扫描的顺序，从标记过的二值图像 BP_i 的第一个入口开始，首先确定一个未被标记的点，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，记录线段端点的两个坐标 (x_1,y_1) 和 (x_2,y_2)，然后对线段的两个端点作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_i 中，即有 $Q\_line_i\{line\_num\}=\{point1\_line,point2\_line\}$，且将存储过的此线段在 BP_i 中作标识。否则，说明只能形成孤立点，执行 Step 12。

Step 12 直接存储孤立点的坐标 (x,y)，将孤立点的计数变量 *point_num* 的值加1，然后将这个点作K码降维变换，即：$single\_point\leftarrow K(x,y)$。最后将 *single_point* 这个变量存储到队列 Q_point_i 中，即有 $Q\_point_i\{point\_num\}=\{single\_point\}$，并将此点在 BP_i 中作标识。

Step 13 循环执行 Step 11 到 Step 12，直到不能形成新

的线段和孤立点为止。

Step 14　$i=i+1$。If $i\leqslant 3m-1$, then go to Step 3, else go to Step 15。

Step 15　依次输出矩形、三角形、线段和孤立点的编码结果 $Q\_rect=\{Q\_rect_0, Q\_rect_1, \cdots, Q\_rect_{3m-1}\}$, $Q\_tri=\{Q\_tri_0, Q\_tri_1, \cdots, Q\_tri_{3m-1}\}$, $Q\_line=\{Q\_line_0, Q\_line_1, \cdots, Q\_line_{3m-1}\}$ 和 $Q\_point=\{Q\_point_0, Q\_point_1, \cdots, Q\_point_{3m-1}\}$。

解码算法的具体步骤：

/ * 给定 4 个编码队列集合 $Q\_rect=\{Q\_rect_0, Q\_rect_1, \cdots, Q\_rect_{3m-1}\}$, $Q\_tri=\{Q\_tri_0, Q\_tri_1, \cdots, Q\_tri_{3m-1}\}$, $Q\_line=\{Q\_line_0, Q\_line_1, \cdots, Q\_line_{3m-1}\}$ 和 $Q\_point=\{Q\_point_0, Q\_point_1, \cdots, Q\_point_{3m-1}\}$，以及图像的灰度级别参数 m 以及分辨率 n，将解码结果存入彩色图像模式 CIP 中，本算法中不妨以矩阵形式存放图像模式。 * /

Step 1　将 3 幅灰度图像模式 $MP[1]$, $MP[2]$和 $MP[3]$（它们分别代表彩色图像的 r, g, b 颜色分量）和 $3m$ 幅二值图像模式 BP_i ($0\leqslant i\leqslant 3m-1$) 分别赋值为 $2^n\times 2^n$ 大小的全 0（黑色）和全 1（白色）的矩阵，且将位面号 i 赋值为 0，3 幅灰度图像模式组成的数组 $MP[j]$ 的下标变量 j 赋值为 1。

Step 2　根据给定的矩形、三角形、线段和孤立点的编码队列集合 Q_rect, Q_tri, Q_line 和 Q_point，分别取得各自第 $i+(j-1)\times m$ 个位平面二值图像的编码结果，即：$Q\_rect_{i+(j-1)\times m}$, $Q\_tri_{i+(j-1)\times m}$, $Q\_line_{i+(j-1)\times m}$ 和 $Q\_point_{i+(j-1)\times m}$。

Step 3　根据 $Q\_rect_{i+(j-1)\times m}$ 计算出该位平面二值图像的矩形子模式的总数 $rect\_num$，对于其中的每个矩形，依次获取矩形的起始点坐标 sp、长度值 $length$ 和宽度值 $width$，即有：$\{sp, length, width\}\leftarrow Q\_rect_{i+(j-1)\times m}\{rect\_num\}$，然后对起始点坐标 sp 作 K 码升维变换，即：$(x, y)\leftarrow K^{-1}(sp)$。最后根据矩形子模式的 3 个参数 (x, y)、$length$ 和 $width$ 即可解码出该矩形子模式，并将矩阵 $BP_{i+(j-1)\times m}$ 对应元素赋值为 0（即黑色）。

Step 4　根据 $Q\_tri_{i+(j-1)\times m}$ 计算出该位平面二值图像的三角形子模式的总数 tri_num，对于其中的每个三角形，依次获取三角形的类型的标识符 $flag$ 及斜边的 2 个端点 $point1\_hyp$ 和 $point2\_hyp$，即有：$\{flag, point1\_hyp, point2\_hyp\}\leftarrow Q\_tri_{i+(j-1)\times m}\{tri\_num\}$，然后将这 2 个点作 K 码升维变换，即：$(x_1, y_1)\leftarrow K^{-1}(point1\_hyp)$, $(x_2, y_2)\leftarrow K^{-1}(point2\_hyp)$，得到斜边的这 2 个端点坐标后，当 $flag=0$ 时即可解码出上三角形，当 $flag=1$ 时即可解码出下三角形，当 $flag=2$ 时即可解码出对称上三角形，当 $flag=3$ 时即可解码出对称下三角形，并将矩阵 BP_i 对应元素赋值为 0。

Step 5　根据 $Q\_line_{i+(j-1)\times m}$，计算出线段的总数 $line\_num$，对于其中的每条线段，依次获取每一条线段的两个端点 $point1\_line$ 和 $point2\_line$，即有：$\{point1\_line, point2\_line\}\leftarrow Q\_line_{i+(j-1)\times m}\{line\_num\}$，然后将这两个点作 K 码升维变换，即：$(x_1, y_1)\leftarrow K^{-1}(point1\_line)$, $(x_2, y_2)\leftarrow K^{-1}(point2\_line)$，得到线段的这两个端点坐标后即可解码出线段，并将矩阵 BP_i 对应元素赋值为 0。

Step 6　根据 $Q\_point_{i+(j-1)\times m}$，计算出孤立点的总数 $point\_num$，对于其中的每个孤立点，依次获取每一个孤立点 $single\_point$，即有：$\{single\_point\}\leftarrow Q\_point_{i+(j-1)\times m}\{point\_num\}$，然后将这个点作 K 码升维变换，即：$(x, y)\leftarrow K^{-1}(single\_point)$，得到孤立点的坐标，并将矩阵 BP_i 对应元素赋值为 0。

Step 7　$i=i+1$。If $i\leqslant m-1$, then go to Step 2, else go to Step 8。

Step 8　根据 $MP[j]=\sum_{i=0}^{m-1}2^iBP_{i+(j-1)\times m}$（定理 1），即可解码出原始多值图像模式 $MP[j]$，并将位面号 i 赋值为 0。

Step 9　$j=j+1$。If $j\leqslant 3$, then go to Step 2, else go to Step 10。

Step 10　根据三幅代表 r, g 和 b 颜色分量的灰度图像模式 $MP[1]$, $MP[2]$和 $MP[3]$ 即可解码出彩色图像模式 CIP。

4　INAM 算法的存储结构分析

INAM 算法中被逆布局的子模式对象是任意大小的矩形 $p=\{rectangle \mid rectangle=(sp, length, width)\}$ 和三角形 $p=\{triangle \mid triangle=(flag, point1\_hyp, point2\_hyp)\}$。由于形成一个矩形至少需要 4 个点而形成一个三角形至少需要 3 个点，因此在 INAM 算法的逆布局结果中除了矩形和三角形外，还有可能会出现线段和孤立点的情况。下面分别讨论矩形、三角形、线段和孤立点的存储结构。

4.1　矩形的存储结构

对于矩形队列集合 Q_rect 来说，INAM 算法的输出是 $3m$ 个顺序存储的矩形队列，即：$Q\_rect=\{Q\_rect_0, Q\_rect_1, \cdots, Q\_rect_{3m-1}\}$，其中 Q_rect_i ($0\leqslant i\leqslant 3m-1$) 分别对应于 $3m$ 个位面的二值图像模式的 INAM 表示。Q_rect_i 的构成主要由三个元素组成，即矩形的起始点 sp 以及矩形的长 $length$ 和宽 $width$。由于只记录“$black$”像素，因此起始点坐标不能省。对于一个 $2^n\times 2^n$ 图像模式来说，sp 即为一个坐标对 (x, y)，x 和 y 的二进制码长度都为 n。具体存储记录用 K 码表示。其存储结构如图 2 所示。

K 用相对值来记录，本次记录的 K 域用本次 K 码减去上一个 K 码的差值来记录，即 $\Delta K=K_i-K_{i-1}$。在统计意义下其长度为 n，在实际情况下，如果 ΔK 的长度却实超过了 n，则可以将该块拆分为两个块，用两个记录来表示。按照 K 的定义，length 和 width 的最大长度为 $n/2$ 位。因此，存储一个矩形子模式占 $2n$ 位。

图 2　矩形的存储结构

4.2　三角形的存储结构

对于三角形队列集合 Q_tri 来说，INAM 算法的输出是 $3m$ 个顺序存储的三角形队列，即：$Q\_tri=\{Q\_tri_0, Q\_tri_1, \cdots, Q\_tri_{3m-1}\}$，其中 Q_tri_i ($0\leqslant i\leqslant 3m-1$) 分别对应于 $3m$ 个位面的二值图像模式的 INAM 表示。Q_tri_i 中的每一个记录的构成由三个元素组成，即三角形形状的标识符 $flag$ 以及斜边的两个端点 $point1\_hyp$ 和 $point2\_hyp$。因此，一个三角形记录可表示为：$p=\{triangle \mid triangle=(flag, point1\_hyp, point2\_hyp)\}$，其存储结构如下图 3 所示，其中 $flag$ 占 2 个 bit，$flag=0$ 时表示上三角形，$flag=1$ 时表示下三角形，$flag=2$ 时表示对称上三角形，$flag=3$ 时表示对称下三角形。

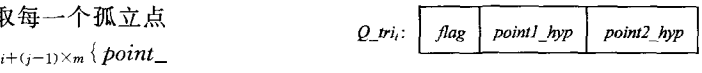

图 3　三角形的存储结构

对于一个 $2^n\times 2^n$ 图像模式来说，$point1\_hyp$ 和 $point2\_hyp$ 均为坐标对(x_i,y_i)，x_i 和 y_i 的二进制码长度都为 n。两个点的具体存储记录用 K 码表示，且也用相对值来记录，在统计意义下其长度均为 n。因此，按照 K 的定义，存储一个三角形子模式占 $2n+2$ 位。

4.3 线段和点的存储结构

线段和点的存储结构相对较为简单，对于线段和点的队列集合 Q_line 和 Q_point 来说，INAM 算法的输出分别是 $3m$ 个顺序存储的线段和点队列，即：$Q\_line=\{Q\_line_0,Q\_line_1,\cdots,Q\_line_{3m-1}\}$ 和 $Q\_point=\{Q\_point_0,Q\_point_1,\cdots,Q\_point_{3m-1}\}$。对于 $Q\_line_i(0\leqslant i\leqslant 3m-1)$ 中的每一个记录来说，只需存储线段的两个端点 $point1\_line$ 和 $point2\_line$，而对于 $Q\_point_i(0\leqslant i\leqslant 3m-1)$ 中的每一个记录来说，直接存储孤立点 $single\_point$ 即可。并且线段的两个端点和孤立点用 K 码来表示，且也用相对值来记录，在统计意义下其长度均为 n。按照 K 的定义，存储一条线段占 $2n$ 位，存储一个孤立点占 n 位。相应的存储结构分别如图 4 和图 5 所示。

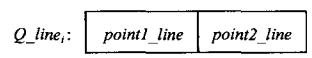

图 4 线段的存储结构

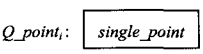

图 5 孤立点的存储结构

5 INAM 算法的数据量分析

本节重点计算并分析 INAM 算法的数据量。通过与流行的线性四元树表示算法作比较，从理论上证明了 INAM 表示算法在数据表示方面的优越性。

在具体分析 INAM 算法的数据量之前，我们先引用一个定理[27]。

定理 2 位平面二值图像 BP_i 的复杂性 $CF(BP_i)$ 小于灰度图像 MP 的复杂性 $CP(MP)$。换句话说：对 BP_i 的逆布局所得到的子模式数少于对 MP 的四元分割所得到的块数，即：$\sum_{i=0}^{m-1}N_r(i)<N_{LQT}$。其中 $N_r(i)$ 表示逆布局后第 i 个位面的子模式总数，N_{LQT} 为 MP 用线性四元树表示时的总块数，亦即，BP_i 扩展了或保持了 MP 的块状结构。

设彩色图像模式 CIP(灰度级参数为 m)的大小为 $2^n\times 2^n\times 3$，经彩色图像的 BPD 分解后，可以将其分解为 3 幅大小为 $2^n\times 2^n$ 的灰度图像模式或者 $3m$ 幅大小为 $2^n\times 2^n$ 的位平面二值图像模式。设第 i 个色彩分量的第 j 个位平面逆布局后的总数据量以及矩形、三角形、线段、孤立点的子模式数分别为 $S(i,j),N_r(i,j),N_t(i,j),N_l(i,j)$ 和 $N_p(i,j)$，其中 $1\leqslant i\leqslant 3$，且 $0\leqslant j\leqslant m-1$。

根据第 4 节的分析可知，对于 INAM 算法来说，存贮一个矩形记录占 $2n$ 位，一个三角形记录占 $2n+2$ 位，一条线段记录占 $2n$ 位，一个孤立点记录占 n 位，则 CIP 逆布局后的 $3m$ 个位面的总数据量 S_{bpd} 为：

$$S_{bpd}=\sum_{i=1}^{3}\sum_{j=0}^{m-1}S(i,j)=\sum_{i=1}^{3}\sum_{j=0}^{m-1}[2nN_r(i,j)+(2n+2)N_t(i,j)+2nN_l(i,j)+nN_p(i,j)] \tag{1}$$

对于基于线性四元树的彩色图像表示方法来说，存贮一个节点占 $3n-1+m$ 位，则线性四元树的总数据量 S_{LQT} 为：

$$S_{LQT}=\sum_{i=1}^{3}(3n-1+m)N_{LQT}(i) \tag{2}$$

其中 $N_{LQT}(i)$ 表示第 i 个色彩分量用线性四元树表示的总节点数。

设 λ_{LQTbpd} 为线性四元树的总数据量与 INAM 的总数据量的比值，则有：

$$\lambda_{LQTbpd}=\frac{S_{LQT}}{S_{bpd}}=\frac{\sum_{i=1}^{3}(3n-1+m)N_{LQT}(i)}{\sum_{i=1}^{3}\sum_{j=0}^{m-1}[2nN_r(i,j)+(2n+2)N_t(i,j)+2nN_l(i,j)+nN_p(i,j)]}>\frac{(3n-1+m)\sum_{i=1}^{3}N_{LQT}(i)}{(2n+2)\sum_{i=1}^{3}\sum_{j=0}^{m-1}[N_r(i,j)+N_t(i,j)+N_l(i,j)+N_p(i,j)]} \tag{3}$$

通过 λ_{LQTbpd} 这一比值，可以比较 INAM 相对于线性四元树的优劣。由于线性四元树的分割是对称形的分割，其分割方式受到很大的限制，而 INAM 的分割是非对称的分割，分割是最大限度地形成预先定义的子模式，分割的灵活性更大，由定理 2 可知，$\sum_{j=0}^{m-1}[N_r(i,j)+N_t(i,j)+N_l(i,j)+N_p(i,j)]<N_{LQT}(i)$，从而可推出 $\sum_{i=1}^{3}\sum_{j=0}^{m-1}[N_r(i,j)+N_t(i,j)+N_l(i,j)+N_p(i,j)]<\sum_{i=1}^{3}N_{LQT}(i)$，因此 $\lambda_{LQTbpd}>(3n-1+m)/(2n+2)>1$，比如：在下节实验中，当 $n=9,m=8$ 时，从理论上来说，$\lambda_{LQTbpd}>1.7000>1$。

综上所述，理论分析表明：对彩色图像模式而言，与流行线性四元树表示方法相比，INAM 表示方法能够更有效地减少数据存储空间。

6 实验结果及分析

为了确证 INAM 算法的理论结果，本节从实验的角度来说明 INAM 算法相对于 NAM 算法和线性四元树算法的明显优势。实验中用来测试的彩色图像模式的分辨率参数 $n=9$，即大小为 $2^9\times 2^9\times 3$ 的彩色图像模式，灰度级参数 $m=8$，即 $2^8=256$ 级。在分析实验结果之前，先介绍一下彩色图像模式的复杂度的定义[27]。彩色图像模式的复杂度是参照线性四元树的块数来定义的，其目的主要是为了和线性四元树进行比较。

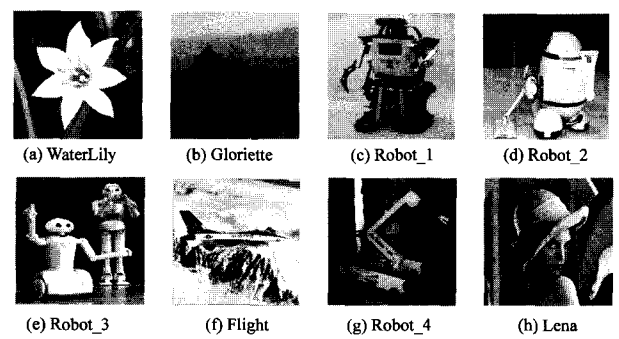

图 6 八幅测试图像

定义 1 彩色图像模式的复杂度 Cc 定义为：$Cc=(N_r+N_g+N_b)/(3N_f)$，其中 N_r,N_g 和 N_b 分别为彩色图像分解后的三幅代表 r,g,b 颜色分量的灰度图像用线性四元树表示时的总块数，N_f 为单幅灰度图像的像素总数，显然有 $0<Cc\leqslant 1$。

实验中用来测试的 8 幅彩色图像如图 6 所示，这些图像的复杂度各不相同，具有较好的代表性，能够说明 INAM 算

法的适应性。

通过在 Matlab6.5 上编程，我们分别实现了 INAM 算法、NAM 算法、线性四元树算法，对这三种算法的实验结果进行了比较，相应的比较数据如表 1 所示。

表 1　INAM，NAM 和 Quadtree 编码的实验数据比较

图像	Cc	N			$Diff$	φ_{bpdLQT}	λ_{LQTbpd}
		Quadtree	NAM	INAM			
WaterLily	0.1953	153621	57078	53223	3855	5.0839	5.9628
Gloriette	0.2216	174309	61140	57241	3899	5.3851	6.7200
Robot_1	0.7415	583128	445378	422040	23338	2.4734	3.1043
Robot_2	0.8245	648378	392350	375463	16887	3.1216	3.8189
Robot_3	0.8351	656712	426388	410156	16232	2.9088	3.5713
Flight	0.9119	717153	589685	556730	32955	2.2973	2.9070
Robot_4	0.9254	727746	661414	616746	44668	2.0783	2.7019
Lena	0.9862	775593	757957	703449	54508	1.9325	2.5285

注：Cc：彩色图像模式的复杂度；N：子模式或节点个数；NAM：矩形 NAM 表示；INAM：矩形和三角形 NAM 表示；Quadtree：线性四元树表示；$Diff$：NAM 与 INAM 的子模式之差；φ_{bpdLQT}：Quadtree 与 NAM 的总数据量之比；λ_{LQTbpd}：Quadtree 与 INAM 的总数据量之比

表 1 中给出了 INAM，NAM 和 Quadtree 的子模式数（节点数）的数据，从实验数据来看，INAM 和 NAM 方法在数据量方面的效果均是非常明显的，其子模式数均远小于 Quadtree 方法的节点数，对图像的适应性也非常好，而且从表 1 中的 Diff 值可知，INAM 方法的子模式数比 NAM 方法的子模式还要少 3855～54508 个。因此，与 NAM 方法和 Quadtree 方法相比，INAM 方法能够更有效地减少子模式的数量。另外，表 1 中也给出了 Quadtree 与 INAM 的总数据量之比 λ_{LQTbpd}，对于给定的 8 幅图像而言，Quadtree 的总数据量是 INAM 的 2.5285～6.7200 倍，显然，这些图像均证实了理论分析的结果，即：当 $n=9, m=8$ 时，即 $\lambda_{LQTbpd}>1.7000>1$。

表 2 中给出了 8 幅测试图像用 INAM 算法表示时的子模式数的分布情况，从 *rect_per*，*tri_ per*，*line_ per*，*point_ per* 的数值来看，矩形和孤立点所占的百分比是较大的，其次是三角形和线段，且三角形占 5.32%～13.62%。由于在 NAM 算法中，每个三角形是用两个矩形来存储的，即每个三角形占用 $4n$ 位，而在 INAM 算法中，每个三角形仅占用 $2n+2$ 位，显然如果将这些三角形用矩形来表示是相当浪费存储空间的，并且从表 1 也不难看出，λ_{LQTbpd} 的值是总高于 φ_{bpdLQT} 的，这表明：在数据表示方面，INAM 对彩色图像的表示能力总要强于 NAM 的表示能力，且能够比 NAM 更有效地减少数据存储空间，因此 INAM 是一种比 NAM 更优的彩色图像表示算法，具有更强的数据表示能力，为逐步逼近彩色图像模式的最优化表示提供了一种重要的研究途径。

表 2　INAM 的子模式数的分布情况

图像	INAM 子模式数分布					INAM 子模式数分布的百分比			
	rect_num	tri_num	line_num	point_num	N	rect_per	tri_ per	line_ per	point_ per
WaterLily	35375	2832	5272	9744	53223	66.47	5.32	9.91	18.31
Gloriette	26822	4695	8189	17535	57241	46.86	8.20	14.31	30.63
Robot_1	159264	51978	64858	145940	422040	37.74	12.32	15.37	34.58
Robot_2	156933	40353	59647	118530	375463	41.80	10.75	15.89	31.57
Robot_3	164966	45605	63773	135812	410156	40.22	11.12	15.55	33.11
Flight	203279	70393	86000	197058	556730	36.51	12.64	15.45	35.40
Robot_4	207411	81670	93407	234258	616746	33.63	13.24	15.15	37.98
Lena	225606	95781	112613	269449	703449	32.07	13.62	16.01	38.30

注：*rect_num*，*tri_num*，*line_num*，*point_num*：分别表示 *INAM* 算法中矩形、三角形、线段和孤立点的总数；*rect_per*，*tri_ per*，*line_ per*，*point_ per*：分别表示 INAM 算法中矩形、三角形、线段和孤立点各自所占的百分比

综上所述，理论分析和实验结果均表明：与 NAM 算法和流行的线性四元树算法相比，INAM 算法能够更有效地减少子模式数（节点数）和数据存储空间，是彩色图像的一种更优的表示算法。因此，从这个意义上说，多子模式类型的组合逆布局是优于单子模式类型的逆布局的。

结束语　通过对多子模式类型的组合逆布局作进一步的研究，提出了一种改进的基于 NAM 的彩色图像表示算法。理论分析和实验结果均表明：与基于 NAM 的彩色图像表示算法和流行的线性四元树表示算法相比，本文提出的算法具有明显的优势，是彩色图像的一种更优的表示方法。这种表示方法可以应用于图像模式表示的各个方面，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。

参 考 文 献

[1] Wang S J, Kuo L C, Jong H H, et al. Representing images using points on image surfaces. IEEE Trans. Image Process, 2005, 14(8): 1043-1056

[2] Hong W, John W, Huang K, et al. Multiscale hybrid linear models for lossy image representation. IEEE Trans. Image Process, 2006, 15(12): 3655-3671

[3] Naik S K, Murthy C A. Distinct multicolored region descriptors for object recognition. IEEE Trans. Pattern Analysis and Machine Intelligence, 2007, 29(7): 1291-1296

[4] Cohen D L. Multiple contour finding and perceptual grouping using minimal paths. Journal of Mathematical Imaging and Vision, 2001, 14(3): 225-236

[5] Liu Y K. Research on the compression algorithm for Freeman chain code. Chinese Journal of Computers, 2001, 24 (12): 1294-1298

[6] Park S J, Han H J. Euclidean reconstruction from contour matches. Pattern Recognition, 2002, 35(10): 2109-2124

[7] Liu Z Y, Sun Z Q, Xu L, et al. Contour representation based on wedgelet. Journal of Systems Engineering and Electronics, 2006, 17(2): 251-257

[8] Liu Y K, Wei W, Guo H. Research on compressed chain code. Chinese Journal of Computers, 2007, 30(2): 281-287

[9] Meyr H, Rosdolsky G H, Huang T S. Optimum run length codes. IEEE Trans. on Communications, 1974, 22(6): 826-835

[10] Yeh R H, Ho W T, Tseng S T. Optimal production run length for products sold with warranty. European Journal of Operational Research, 2000, 120(3): 575-582

[11] Vasic B, Pedagani K. Run-length-limited low-density parity check codes based on deliberate error insertion. IEEE Trans. Magnetics, 2004, 40(3): 1738-1743

[12] Peel C M, Pegram S, Mcmahon A T. Global analysis of runs of annual precipitation and runoff equal to or below the median: run length. International Journal of Climatology, 2004, 24(7): 807-822

[13] Nagy Z, Zeger K. Bit-stuffing algorithms and analysis for run-length constrained channels in two and three dimensions. IEEE Trans. Information Theory, 2004, 50(12): 3146-3169

[14] Makinen V, Navarro G, Ukkonen E. Approximate matching of run-length compressed strings. Algorithmica, 2003, 35(4): 347-369

[15] Radson D, Boyd H A. Graphical representation of run length distributions. Quality Engineering, 2005, 17(2): 301-308

[16] Amir A, Landau MG, Sokol D. Inplace run-length 2 d compressed search. Theoretical Computer Science, 2003, 290(3): 1361-1383

[17] Monasse P, Guichard F. Fast computation of a contrast-invariant image representation. IEEE Trans. Image Processing, 2000, 9(5): 860-872

[18] Flusser J. Refined moment calculation using image block representation. IEEE Trans. Image Processing, 2000, 9(11): 1977-1978

[19] Voronin V. Holographic representation in image processing tasks. Pattern Recognition and Image Analysis, 2001, 11(1): 265-267

[20] Liu Y, Ranganath S, Zhou X. Wavelet-based image segment representation. Electronics Letters, 2002, 38(19): 1091-1092

[21] Kharinov V M. Representation of image information for machine computation. Pattern Recognition and Image Analysis, 2005, 15(1): 212-214

[22] Malo J, Epifanio I, Rafael N, et al. Nonlinear image representation for efficient perceptual coding. IEEE Trans. Image Process, 2006, 15 (1): 68-80

[23] Samet H. Data structures for quadtree approximation and compression. Communications of the ACM, 1985, 28(9): 973-993

[24] Samet H, Webber R E. Storing a collection of polygons using quadtrees. ACM Trans. on Graphics, 1985, 4(3): 182-222

[25] Li S X, Loew M H. The quadcode and its arithmetic. Communications of the ACM, 1987, 30(7): 621-626

[26] Gargantini I. An effective way to represent quadtrees. Communications of the ACM, 1982, 25(12): 905-910

[27] Zheng Y P, Chen C B. Study on a NAM-based color image representation method. Journal of Software, 2007, 18(11): 2932-2941

[28] Chen C B, Hu W J, Wan L. Direct non-symmetry and anti-packing pattern representation model of medical images // Proceedings of the First International Conference on Bioinformatics and Biomedical Engineering. Wuhan, China, July2007: 1011-1018

[29] Zheng Y P, Chen C B, Sarem M. A novel algorithm for triangle non-symmetry and anti-packing pattern representation model of gray images // Proceedings of the Third International Conference on Intelligent Computing ICIC'07, LNCS 4681. Qingdao, China, August 2007: 832-841

(上接第 199 页)

的应用场合下，实验中使用的 Logistic 回归模型及其参数估计方法较 SVM 有更快的算法收敛速度、更高的运行效率。

表 6 Logistic 回归与 SVM 运行效率的比较

Model	Train Time(s)	Test Time(s)	Total(s)
LR(C=4, w_i=1:3, w_m=1:1)	34(30-38)	16(13-18)	50(43-56)
LR(C=256, w_i=1:3, w_m=1:5)	43(39-46)	16(17-15)	59(56-61)
SVM(C=4, w_i=1:3, w_m=1:1)	2115 (1858-2372)	157 (127-186)	2272 (1985-2558)
SVM(C=256, w_i=1:3, w_m=1:5)	1548 (1460-1635)	115 (111-118)	1663 (1571-1753)

结束语 本文实验结果表明，Logistic 回归模型具有较少的调节参数，能够在中文垃圾邮件过滤应用中取得很好的分类效果。与 SVM 相比，无论是在 ROC 分类指标上，还是在运行效率上，Logistic 回归模型都要优于后者。未来的主要工作是，如何将 Logistic 回归模型应用于在线中文垃圾邮件过滤系统中，并验证其效果。

参 考 文 献

[1] Hulten G, Goodman J. Tutorial on junk e-mail filtering // Proceedings of 21st International Conference on Machine Learning (ICML'2004). Banff, Canada, July 2004: 45-57. http://www.research.microsoft.com/~joshuago/tutorialOnJunkMailFilteringjune4.pdf

[2] Hsu C, Chang C, Lin C. A practical guide to support vector classification, 2007. http://www.csie.ntu.edu.tw/~cjlin

[3] Lynam T R, Cormack G V. On-line spam filter fusion // Proceedings of 29th Annual International ACM SIGIR Conference on Research and Development on Information Retrieval (SIGIR'2006). Seattle, Washington, August 2006

[4] Goodman J, Yih W. Online discriminative spam Filter training // Proceedings of 3rd Conference on Email and Anti-Spam (CEAS'2006). July 2006: 113—116. http://www.ceas.cc/2006/allpapers.pdf

[5] [英]Andrew R W. 统计模式识别(第二版). 王萍，等译. 北京：电子工业出版社，2004

[6] Lin C, Weng R C, Keerthi S S. Trust region Newton method for large-scale logistic regression // Proceedings of 24th International Conference on Machine Learning (ICML'2007). June 2007. http://www.machinelearning.org/ proceedings/icml2007/papers/114.pdf

[7] Zhang J, Yang Y. Robustness of regularized linear classification methods in text categorization // Proceedings of 26th Annual International ACM SIGIR Conference on Research and Development on Information Retrieval (SIGIR'2003). Toronto, Canada, July 28-August 1, 2003: 191-192. http://net.pku.edu.cn/~wbia/2004/ public_html/Readings/tmp/reading-2/Robustness%20of%20Regularized%20Linear%20Classification%20Methods%20in%20Text%20Categorization.pdf

[8] 陈宝林. 最优化理论与算法(第 2 版). 北京：清华大学出版社，2005

[9] Sebastiani F. Machine learning in automated text categorization. ACM Computing Surveys, 2002, 34(1): 1-47

[10] 华南理工大学信息网络工程研究中心和广东省计算机网络重点实验室邮件评测小组. SEWM2007 垃圾邮件过滤系统评测，2007, 3. http://www.cwirf.org/ 2007WebTrack/cct/SEWM07_SPAMoverview.ppt