梯形子模式非对称逆布局二值图像表示方法*)

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 虽然树形分层结构在图像表示和处理中有很多优点,但是它们都过分地强调分割的对称性和节点的对称性,因此不是最优的图像表示方法。借鉴 Packing 问题研究的方法,基于非对称逆布局模式表示模型(Non-symmetry Anti-packing pattern representation Model,NAM),提出了一个梯形子模式非对称逆布局二值图像表示方法,给出具体的编码算法和解码算法,并分析了算法的时空复杂度和表示的数据量。理论分析和试验结果表明,与流行的基于分层结构的线性四元树表示方法相比,梯形子模式非对称逆布局二值图像表示方法在子模式数量和数据量方面具有较大的优势。

关键词 图像表示,布局问题,梯形子模式,线性四元树,图像复杂度

Trapezium-based Non-symmetry Anti-packing Representation Method for Binary Images

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 Although the hierarchical structures have many merits in image representation and processing, their compactness is impaired because of excessive emphases upon the symmetry of divisions and the symmetry of nodes. Inspirited by the concept of the famous Packing Problem, based on non-symmetry anti-packing pattern representation model, a trapezium-based non-symmetry anti-packing representation method for binary images is presented. After giving encoding and decoding algorithms, this paper analyses the time-and space-complexity. Theoretical analysis of the data amounts and the experiments show that this representation is greatly superior to linear quadtree in the aspects of data amounts and sub-patterns' quantities.

Keywords Image representation, Packing problem, Trapezium, Linear quadtree, Image complexity

1 引言

图像表示在计算机图形学、图像处理、模式识别、计算机视觉和机器人等研究领域内有重要的研究意义。好的图像表示方法能够极大地减少图像操作的时空开销,方便进一步的处理。基本的二维数组表示使用第 i 行,第 j 列的元素表示图像坐标(i,j)处的像素值,这种方式的最大优点是直观,但是由于这种方法不利用像素与像素之间的相关性,因而需要较大的存储空间且常用的图像处理运算效率不高,不能满足应用的需要。边界链码[1-3]和行程编码[4,5]等表示方法利用了图像像素间的相关性,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[6-10]。

图像的分层数据结构[11-13]将图像描述成若干不同粒度的层次,使算法可以根据需要将处理控制在不同的层次上,只有在必要时,处理才会在最小粒度的层次上进行。同时,由于图像运算被转化为对树结构的追迹,因而图像处理运算可以快速实现[14-17]。另外,分层数据结构利用了图像数据的二维相关性,从而大大降低了图像表示的冗余度,减少了表示的数据量。然而,尽管分层数据结构有许多优点,由于过于强调分割的对称性和节点的对称性,它们仍然不是最优的表示方法。

借鉴 Packing 问题的思维方法,考虑了图像的非对称本性,在非对称逆布局模式表示的基础上,提出了一种新的梯形子模式非对称逆布局二值图像表示方法,给出了具体的编码算法和解码算法,并分析了算法的时间和空间复杂度。数据量分析和试验结果表明,梯形子模式非对称逆布局二值图像表示方法较之于常用的线性四元树表示方法,在子模式数量和数据量方面都具有较大的优势。

2 非对称逆布局模式表示模型

非对称逆布局模式表示模型(Non-symmetry Anti-packing pattern representation Model,NAM)是著名的 Packing 问题的反问题。基本的 Packing 问题是一个判定问题:给定一个容器和若干对象,问这些对象能否不重叠地放入容器中。如果不能放入,则拒绝,否则接受并给出一个满足条件的布局。而非对称逆布局模式表示模型则是给定一个模式,找到一组子模式,使得这组子模式的某个布局可以完全与原模式重合。

非对称(Non-symmetry)是指子模式及子模式布局方法是不对称的,这种不对称性是相对于当前众多的对称模式表示方法而言的。比如在图像表示中,树形分层编码强调节点对应的区域是对称的(正方形),区域的分割也是对称的。

设原模式为 Γ,Γ 的一个 NAM 表示为 Γ′,则 NAM 表示

*)国家高技术研究发展计划(2006AA04Z211)。黄 巍 博士研究生,研究方向为图像处理与模式识别;陈传波 博士,教授,博士研究生导师,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。

模型是一个由 Γ 到 Γ' 的变换 $\Gamma'=T(\Gamma)$，其中 T 称为编码函数。正向编码过程记为 $\Gamma\rightarrow\Gamma'$，其中

$$\Gamma'=\bigcup_{i=1}^{k}p_i(v,A\mid A=\{a_1,a_2,\cdots,a_{m_i}\}) \quad (1)$$

是 Γ 的一个 NAM 表示，k 是用于表示模式 Γ 的子模式个数，p_i 是预先定义的子模式类型集合 $P=\{P_1,P_2,\cdots,P_n\}$ 中某个子模式类型 P_j 的实例，v 是 p_i 的值，A 是 p_i 子模式的参数集合，$a_r(1\leqslant r\leqslant m_i)$ 是子模式 p_i 的形状参数。

本文重点讨论如何使用梯形子模式类型表示二值图像。

3 梯形子模式类型

子模式类型的选择问题是非对称逆布局二值图像表示的一个重要问题。好的子模式类型可以大幅度地减少子模式数量和总的数据量，然而，由于图像的多样性，不可能找到一个对所有图像都最优的子模式类型。通常说来，好的子模式类型必须满足以下两个性质：

性质 1：子模式必须能够表示一个像素。

性质 2：单个子模式应该有能力覆盖尽可能多的像素。

性质 1 保证了子模式无损表达图像的可能性。由于这种子模式类型能够表达一个像素，则在最坏情况下，可以为每个像素生成一个子模式。性质 2 保证了该子模式类型表达图像的高效性。由于该子模式类型的一个实例能够表达大量的像素，则表示图像的子模式数量和全部的数据量将大大减少，进而提高图像操作的时空效率。

本文使用的梯形子模式类型是一个顶边和底边与图像顶边平行的四边形。由于只需要记录黑色的区域数据，因此梯形子模式不需要包含颜色数据，而只需要记录梯形左上角的坐标和形状参数。因而梯形子模式的定义为一个五元组 $(sp,width,height,left\_indent,right\_indent)$，其中 sp 是左上角坐标，$width$ 是梯形顶边长度，$height$ 是梯形的高，$left\_indent$ 和 $right\_indent$ 分别是左缩进和右缩进。左缩进指下一行最左像素的横坐标相对于上一行最左像素横坐标的偏移量，右缩进指下一行最右像素的横坐标相对于上一行最右像素横坐标的偏移量，负数表示向左偏移，正数表示向右偏移。图 1 所示为梯形子模式及其变体。在图 1 上排最左边的梯形子模式中，$width$，$height$，$left\_indent$ 和 $right\_indent$ 分别为 4，7，−1 和 2。

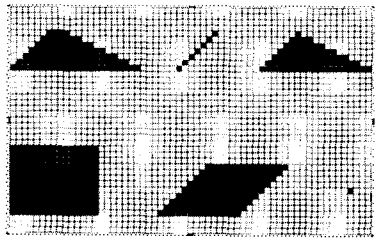

图 1 梯形子模式及其变形

由于一个 $width$ 和 $height$ 都为 1 的梯形子模式可以表示一个像素，因此梯形子模式类型满足上述性质 1 的可能性要求。又由于梯形的形状参数调整使梯形可以变为点、直线、矩形、三角形和平行四边形，这就使得梯形子模式具有较大的适应性，有能力在一个子模式中覆盖尽可能多的像素，因此梯形子模式类型满足上述性质 2 的高效性要求。

图 2 所示为梯形子模式的存储结构，5 个域分别存储五元组的 5 个分量。域的存储长度值得进一步讨论：一方面，选择更大的存储长度有利于覆盖更大的区域，从而减少子模式数量；另一方面，由于图像中往往存在大量较小尺度的区域，以较大的存储长度来表达这些较小的区域会浪费大量的存储空间，使得表示的空间效率降低，因此存储长度的选择依赖于具体的应用。Huffman 编码等变长编码技术在一定意义上可以缓解这种矛盾，然而本文不讨论这类变长编码技术，而是采用一种折中方案。

sp	width	height	left_indent	right_indent

图 2 梯形子模式的存储结构

在统计意义下，$2^n\times2^n$ 的图像中梯形子模式的顶边长度和高不超过 $2^{\lfloor n/2\rfloor}-1$ 个像素（$\lfloor x\rfloor$ 表示不超过 x 的最大整数），因而 $width$ 和 $height$ 域可以使用 $\lfloor n/2\rfloor$ 个比特表示。如果实际梯形子模式的顶边长度或高度超过了 $2^{\lfloor n/2\rfloor}-1$ 个像素，则将该梯形子模式分割为多个梯形子模式。相应地，将起点坐标 sp 采用增量方式（即与上一个子模式起点坐标的差值）表示，其横纵坐标的存储长度都为 $\lfloor n/2\rfloor$ 个比特，如果实际梯形子模式的坐标增量超过了 $2^{\lfloor n/2\rfloor}-1$，则在两个梯形子模式中插入形状参数全为 0 的梯形子模式。选择较大的左右缩进可能会破坏梯形子模式的连通性，比如当 $width$ 为 1，$height$ 大于 1，而 $left\_indent$ 和 $right\_indent$ 分别为 −3 和 −2 时，实际表示的梯形子模式在四邻接和八邻接意义上都不连通。为了保持梯形子模式的块状性质，即保持梯形子模式的八邻接连通性，$left\_indent$ 和 $right\_indent$ 被分别限制到区间 $[-2,1]$ 和 $[-1,2]$ 上，因而 $left\_indent$ 域和 $right\_indent$ 域各需要 2 个比特。综上所述，一个梯形子模式需要使用 $4\times\lfloor n/2\rfloor+2\times2$ 个比特的存储空间。

4 编码和解码算法

在讨论梯形子模式非对称逆布局二值图像表示的编码算法和解码算法之前，先介绍一种与笛卡尔坐标系统不同的新坐标系统。设 G 是 $2^n\times2^n$ 的二值图像，则 G 上任意一点的笛卡尔坐标的二进制形式为 $((x_{n-1}x_{n-2}\cdots x_0)_2,(y_{n-1}y_{n-2}\cdots y_0)_2)$，在 G 上定义一维坐标系统 K，使得对于任意的 $((x_{n-1}x_{n-2}\cdots x_0)_2,(y_{n-1}y_{n-2}\cdots y_0)_2)\in G$，其 K 轴上的坐标为 $k=(x_{n-1}y_{n-1}x_{n-2}y_{n-2}\cdots x_0y_0)_2$，称为 K 码，并称 $K:(x,y)\rightarrow k$ 为降维变换，$K^{-1}:k\rightarrow(x,y)$ 为升维变换。图 3 是 K 码的走向示意图。坐标系统 K 考虑了二维空间的相关性，因而较适应图像数据的块状本性。梯形子模式的起点采用 K 码表示时，同样采用增量方式，sp 域的长度仍为 $2\times\lfloor n/2\rfloor$ 比特。

y\x	000	001	010	011	100	101	110	111
000								
001								
010								
011								
100								
101								
110								
111								

图 3 K 码走向示意图

4.1 编码算法和解码算法

梯形子模式非对称逆布局二值图像表示的编码算法将二值图像分割为多个形状参数可变的梯形，并将这些梯形按起

始点坐标递增序排列在线性表 Q 中。算法主要分为两个阶段：起始点查找阶段和梯形子模式逆布局阶段。起始点查找阶段的目标是确定梯形子模式的起始点坐标。梯形子模式逆布局是在起始点查找阶段找到的起始点处，在二值图像上匹配出所有可能的梯形子模式（包括变体），使用子模式评价标准选择一个最优的子模式作为入选的子模式，并将其加入到表示图像的子模式线性表 Q 中。本文简单地选择覆盖最大像素的梯形子模式作为最优子模式。在梯形子模式逆布局阶段需要注意的是：为了确保入选的梯形子模式能够使用 $4\times\lfloor n/2\rfloor+2\times2$ 个比特的存储空间表示，需要控制梯形子模式的起点坐标增量不超过 $2^{2\times\lfloor n/2\rfloor}-1$，顶边长度和高不超过 $2^{\lfloor n/2\rfloor}-1$，左右缩进分别控制在闭区间[−2,1]和[−1,2]上。梯形子模式非对称逆布局二值图像表示的解码算法相对编码算法简单得多，仅需要遍历子模式线性表 Q，并在二值图像 G 上将所有梯形子模式包含的像素设为黑色即可。

编码算法的具体步骤如下：

算法 1 编码算法

输入：二值图像的像素数组表示 G 和分辨率 n。

输出：二值图像的梯形子模式非对称逆布局表示 Q。

Step1 将 K 码 *previous_sp* 设为 0，即 *previous_sp*←0，初始化梯形子模式的线性表 Q 为空。

Step2 以 *previous_sp* 为起点，以 K 码递增方式，在 G 中查找第一个黑色像素，令其坐标为 (x,y)，并求其 K 码 *sp*，即 *sp*←K(x,y)。如果这种黑色像素没有找到，则算法结束，并返回 Q。

Step3 如果 *sp*-*previous_sp*$\geqslant 2^{2\times\lfloor n/2\rfloor}$，则执行 Step4，否则跳转到 Step5。

Step4 将形状参数全为 0 的梯形子模式 $(2^{2\times\lfloor n/2\rfloor}-1,0,0,0,0)$ 加入到 Q 的末尾，同时 *previous_sp*←*previous_sp*+$2^{2\times\lfloor n/2\rfloor}-1$，跳转到 Step3。

Step5 以 (x,y) 为最左上角坐标，以顶边和底边平行与图像的上下边界，并将梯形的顶边长度 *width* 和高度 *height* 限制在闭区间 $[1,2^{\lfloor n/2\rfloor}-1]$ 上，将 *left_indent* 和 *right_indent* 分别限制在闭区间[−2,1]和[−1,2]上，匹配出所有的不包含白色像素的梯形（包含变形）。

Step6 在 Step5 匹配的所有梯形中，选择面积最大的一个梯形作为入选的梯形，将该梯形子模式（*sp*-*previous_sp*, *width*, *height*, *left_indent*, *right_indent*）加入到 Q 的末尾，并在 G 上将该梯形覆盖的像素设为白色。

Step7 *previous_sp*←*sp*，跳转到 Step2。

解码算法的具体步骤如下：

算法 2 解码算法

输入：二值图像的梯形子模式非对称逆布局表示 $Q=(p_1,p_2,\cdots,p_k)$ 和分辨率 n。

输出：二值图像的像素数组表示 G。

Step1 分配 $2^n\times2^n$ 的数组 G，将 G 中每个元素的值设为白色，并将梯形子模式索引 i 设为 1，即 i←1，将 K 码 *previous_sp* 设为 0，即 *previous_sp*←0。

Step2 从 Q 中取出梯形子模式 p_i，从 p_i 中取出该梯形子模式的上底宽度 *width*，高 *height*，左缩进 *left_indent*，右缩进 *right_indent* 以及该梯形的起始点 K 码相对于前一个梯形的起始点 K 码 *previous_sp* 的增量 *increment*。

Step3 *sp*←*previous_sp*+ *increment*，在 G 中以坐标 $K^{-1}(sp)$ 为最左上角坐标，将与梯形（*width*, *height*, *left_indent*, *right_indent*）覆盖的像素点对应的数组 G 中的元素的值设为黑色。

Step4 i←i+1，*previous_sp*←*sp*，如果 $i\leqslant k$，跳转到 Step2，否则算法结束，返回 G。

4.2 算法复杂性分析

设 $2^n\times2^n$ 图像的像素总数为 N_f，图像的解析度为 n，则 K 码升维变换或 K 码降维变换的时间复杂度为 $O(n)$。又设全部的梯形子模式数为 N_T，全部原始图像的黑色像素的个数为 N_{black}。编码算法中 Step1 消耗常数时间，Step2 最多对每个像素执行一次降维变换，需要时间 $O(nN_f)$，Step3 和 Step4 最多执行 N_T 次，需要的时间都为 $O(N_T)$，Step5 为 N_T 个起始点匹配所有可能的梯形，每个起始点最多对应 $2^{\lfloor n/2\rfloor}\times2^{2\times\lfloor n/2\rfloor}\times4\times4$ 个梯形，而匹配梯形（*width*, *height*+1, *left_indent*, *right_indent*）时只需要在梯形（*width*, *height*, *left_indent*, *right_indent*）的基础上检查下一行对应的像素是否为黑像素，而梯形下底的最大长度为 $5\times2^{\lfloor n/2\rfloor}-9$，所以需要的时间最多为 $O(2^{3n/2}N_T)$，Step6 将所有的黑像素改为白色，需要的时间为 $O(N_{black})$，Step7 执行 N_T 次，需要的时间为 $O(N_T)$。将各步骤的运行时间求和，并注意到 $N_{black}\leqslant N_f$，$N_T<N_f$，得到编码算法的时间复杂度为 $O(N_f^{7/4})$。在空间开销方面，编码算法只增加非常少的中间变量，且生成的梯形子模式数 N_T 小于 N_f，所以其空间复杂度为 $O(N_f)$。解码算法的时间复杂度则更低，只对图像中的黑像素遍历一次，所以时间复杂度为 $O(N_f)$。在空间上几乎没有新的额外空间开销，所以空间复杂度也为 $O(N_f)$。

5 数据量计算与分析

在比较梯形子模式非对称逆布局二值图像表示和线性四元树表示方法之前，先给出图像复杂度和压缩比的定义。

定义 1 图像复杂度是图像客观存在的复杂程度。设图像模式 G 的大小为 $2^n\times2^n$，则像素总数为 $N_f=4^n$，又设该图像的线性四元树表示的黑色节点数为 N_{LQT}，则图像复杂度定义为

$$C_p(G)=\frac{\text{图像 }G\text{ 用线性四元树表示的块数}}{N_f}=\frac{N_{LQT}}{4^n} \quad (2)$$

简记为 C_p，显然有 $0<C_p\leqslant1$。

定义 2 压缩比由下式确定：

$$\eta=\frac{\text{压缩前图像表示的总比特数}}{\text{压缩后图像表示的总比特数}} \quad (3)$$

对于梯形子模式来说，设编码后的子模式数为 N_T，存贮一个子模式使用 $4\lfloor n/2\rfloor+4$ 比特，则其总数据量 H_T 为：

$$H_T=(4\lfloor n/2\rfloor+4)N_T=(4\lfloor n/2\rfloor+4)4^nC_P\frac{N_T}{N_{LQT}} \quad (4)$$

表 1 梯形 NAM 与线性四元树的对比

图像	C_p	NAM 表示子模式数	线性四元树表示节点数	NAM 表示比特数	线性四元树表示比特数	压缩因子 η_{TLQT}
Building	0.0702	633	4602	12600	105846	7.98
Flight	0.0841	1378	5512	27560	126776	4.60
Lena	0.0981	1617	6430	32340	147890	4.42
Drawing	0.1701	1546	11150	30920	256450	8.30

线性四元树表示存贮一个节点使用 $3(n-1)+2$ 比特[18]，则线性四元树的总数据量 H_{LQT} 为：

$$H_{LQT}=[3(n-1)+2]N_{LQT}=(3n-1)4^nC_P \quad (5)$$

设 η_T，η_{LQT} 和 η_{TLQT} 分别为梯形子模式 NAM 表示的压缩比、线性四元树表示的压缩比和梯形子模式 NAM 表示相对

于线性四元树表示的压缩比，则有

$$\eta_T=\frac{N_f}{H_T}=\frac{4^n}{(4\lfloor n/2\rfloor+4)4^nC_P\frac{N_T}{N_{LQT}}}=\frac{N_{LQT}}{(4\lfloor n/2\rfloor+4)C_pN_T} \tag{6}$$

$$\eta_{LQT}=\frac{N_f}{H_{LQT}}=\frac{4^n}{(3n-1)4^nC_p}=\frac{1}{(3n-1)C_p} \tag{7}$$

$$\eta_{TLQT}=\frac{H_{LQT}}{H_T}=\frac{(3n-1)4^nC_P}{(4\lfloor n/2\rfloor+4)4^nC_p\frac{N_T}{N_{LQT}}}\geqslant(\frac{3}{2}-\frac{7}{2n+4})\frac{N_{LQT}}{N_T} \tag{8}$$

式(4)和式(5)反映的是梯形子模式 NAM 表示和线性四元树表示的绝对数据量。可以看出，无论采用何种编码方式，总数据量都与图像模式的复杂度 C_p 有关。一般说来，图像越复杂，则表示该图像的数据量越大。式(6)和式(7)反映了梯形子模式 NAM 表示和线性四元树表示对像素数组表示的压缩比。在式(6)中，由于梯形子模式 NAM 表示没有强调分割对称性和模式对称性，因此一般说来 $N_T<<N_{LQT}$，因而梯形子模式 NAM 表示的压缩比应该大于线性四元树表示的压缩比。式(8)进一步说明，当图像较大时(即 n 较大)，理论上梯形子模式 NAM 表示的数据量比线性四元树表示的数据量减少了将近 50%。由于 $N_T<<N_{LQT}$，实际上减少的数据量要多得多，下节的试验结果也证实了这一点。

6 实验结果分析与比较

图 4 是实验使用的四幅二值图像，它们的分辨率都为 8，这些图像具有典型的代表性。四幅图像以图像复杂度递增的顺序排列，Building 的图像复杂度最低，而 Drawing 的图像复杂度最高。

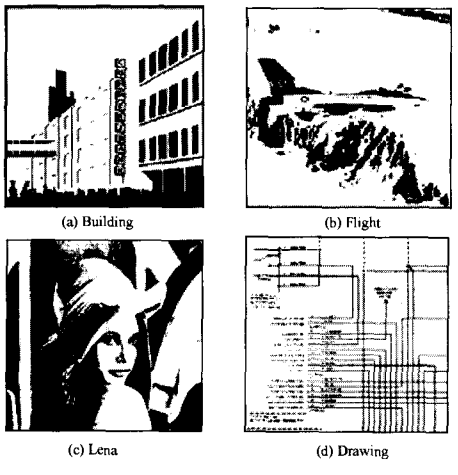

图 4 用于实验的二值图像

表 1 中给出了梯形子模式 NAM 表示和线性四元树表示的子模式数及比特数。一般说来，随着图像复杂度的提高，无论是梯形 NAM 表示的子模式数量还是四元树表示的节点数量都会增加。然而 Building 和 Drawing 的子模式数量，特别是 Drawing 的子模式数量却存在一点异常：虽然 Building 的图像复杂度与 Flight，Lena 的图像复杂度相当，然而 Building 的子模式数量却不到 Flight 和 Lena 的子模式数量的一半；另外，Drawing 比 Lena 的图像复杂度高，然而，Drawing 的子模式数量却比 Lena 的子模式数量少。这是因为，Building 和 Drawing 含有较多规则的几何形体，所以梯形子模式 NAM 表示方法对它们特别有效。但从总体上看，梯形子模式 NAM 表示的子模式数量和所需的比特数都比线性四元树表示的节点数和所需的比特数少。

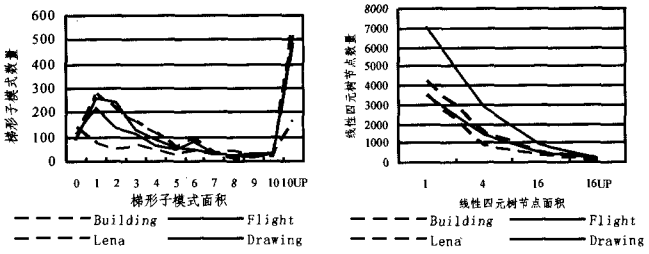

图 5 子模式或节点分布折线图

图 5 是四幅实验图像的梯形子模式 NAM 表示的子模式分布折线图和线性四元树表示的节点分布折线图。横坐标的 10UP 表示面积大于 10 的所有梯形子模式的总和，16UP 表示面积大于 16 的线性四元树的节点总和。从图中可以看出，梯形子模式 NAM 表示的子模式的面积可以连续出现，而线性四元树表示的节点面积是跳跃出现的，它只能按照 4^i 大小出现，这是因为线性四元树表示强调节点的对称性。另外，由于线性四元树表示过分强调分割的对称性和节点的对称性，很难产生大面积的节点，其节点分布曲线呈单调下降的趋势。梯形子模式 NAM 表示的子模式数量分布具有澡盆特性，大块面积的子模式在全部子模式中所占的比重相当可观，因而表示的效率更高。试验使用的四幅二值图像的梯形子模式 NAM 表示的子模式分布折线具有相似的形状，说明编码方法对于各类图像具有相当强的稳定性。

结束语 借助 Packing 问题研究的思维方法，提出了一种梯形子模式非对称逆布局二值图像表示方法，在给出了具体的编码算法和解码算法之后，分析了算法的时间复杂度和空间复杂度。在数据量分析方面使用图像复杂度和压缩率的概念，与流行的线性四元树表示方法进行理论比较，从理论上证明了本方法的优势所在。最后，从实验的角度比较了梯形子模式非对称逆布局二值图像表示方法和线性四元树表示方法的结果，实验数据进一步证实了梯形子模式非对称逆布局二值图像表示方法在子模式数量和数据量方面的优势是明显的。

参 考 文 献

[1] Dokladal P, Enficiaud R, Dejnozkova E. Contour-based object tracking with gradient-based contour attraction field [C]// Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing. 2004: 17-20

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

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

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

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

[6] 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 [J]. International Journal of Climatology, 2004, 24(7): 807-822

[7] Nagy Z, Zeger K. Bit-stuffing algorithms and analysis for run-length constrained channels in two and three dimensions [J].

IEEE Transactions on Information Theory, 2004, 50(12): 3146-3169
[8] Makinen V, Navarro G, Ukkonen E. Approximate matching of run-length compressed strings [J]. Algorithmica, 2003, 35(4): 347-369
[9] Radson D, Boyd H A. Graphical representation of run length distributions [J]. Quality Engineering, 2005, 17(2): 301-308
[10] Amir A, Landau M G, Sokol D. Inplace run-length 2d compressed search [J]. Theoretical Computer Science, 2003, 290 (3): 1361-1383
[11] Flusser J. Refined moment calculation using image block representation [J]. IEEE Transactions on Image Processing, 2000, 9 (11): 1977-1978
[12] Malo J, Epifanio I, Navarro R, et al. Nonlinear image representation for efficient perceptual coding [J]. IEEE Transactions on Image Processing, 2006, 15(1): 68-80
[13] Kharinov V M. Representation of image information for machine computation [J]. Pattern Recognition and Image Analysis, 2005, 15(1): 212-214
[14] Wundrich J I, Von der Malsburg C, Wurtz P R. Image representation by complex cell responses [J]. Neural Computation, 2004, 16(12): 2563-2575
[15] Liu Y, Ranganath S. Wavelet in POCS for image segment representation [J]. Electronics Letters, 2003, 39(19): 1379-1380
[16] Huang K, Wu Z, Wang Q. Image enhancement based on the statistics of visual representation [J]. Image and Vision Computing, 2005, 23(1): 51-57
[17] Jinsung O. Structuring element representation of an image and its applications [J]. International Journal of Control, Automation, and Systems, 2004, 2(4): 509-515
[18] Gargantini I. An Effective Way to Represent Quadtrees [J]. Communications of the ACM, 1982, 25(12): 905-910

(上接第 207 页)

对于每一帧输入图像 A，计算其在背景特征空间 U_d 上的投影，然后在图像空间重建图像。在每个像素点 (x,y) 处，计算重建图像和原始图像的欧氏距离 $d(x,y)=\|A(x,y)-\tilde{A}(x,y)\|$，如果该距离大于某个阈值，即 $d(x,y)>T$ 成立，则该像素点 (x,y) 被认为是前景目标，否则就被认为是背景点。

第 3 步　更新子空间

每累积 N 帧新图像，按(6)式更新协方差矩阵并对其进行特征值分解而得到新的投影矩阵。

第 4 步　转到步骤 2。

4　实验结果与分析

为了验证算法的有效性，我们选择了两段有较多场景变化的视频来进行测试。第一段视频是海边的场景，包括海浪的起伏、草的摆动、光照的变化以及地面的阴影等。第二段视频是喷泉的场景，包括水的运动、光线的变化和树叶的运动引起的场景变化。两段视频的格式为：分辨率 320×240，每秒 30 帧，RGB 3 通道。

为了消除光照等因素对图像的影响，克服不同图像之间的平均灰度差异，这里首先将输入图像由初始的 RGB 色彩空间转换到一个归一化的色彩空间，转换方法如下：$s=R+G+B$，$r=R/s$，$g=G/s$，此外，为了进一步降低计算量，每帧图像都分成 20×20 大小相同的图像块，每个图像块独立运行上面提出的算法。

4.1　实验结果

经过简单形态学处理后的前景检测结果如图 1 和图 2 所示，从实验结果可以看到，提出的方法可以在复杂变化的动态背景下较完整地检测出运动目标。算法采用 Matlab 编程实现，在 Pentium 4 3.0 GHz 机器上运行。在未经任何优化的情况下，平均处理速度达到约每秒 10 帧，经过优化后有望接近实时处理。

4.2　与其他方法的比较

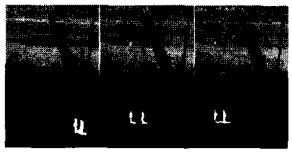

图 1　视频一结果截图

图 2　视频二结果截图

我们将提出的方法与两种主流运动目标检测方法：混合高斯法[1,2]和核密度估计法[3,4]进行了比较。实验中，目标检测后，三种算法都采用了简单的滤波处理。为了对三种算法的性能进行比较，通过逐渐改变算法的阈值，我们绘制了这三种算法的 ROC 曲线。如图 3 和图 4 所示，曲线的横轴表示前景的误判率，纵轴表示前景的检测率，其中作为参照的真实前景区域事先由人工标定。从图 3 和图 4 中可以看出，在相同的误判率下，我们提出的算法的检测率要高于其他两种算法。

结束语　本文提出了一种基于二维主成分分析的自适应运动目标检测方法。和传统的像素点建模方法不同，该方法采用二维主成分分析来建立背景模型，加入了二维空间上像素点之间的相关性信息，得到了更好的建模效果。为了能够适应不断变化的复杂环境，提出的增量算法能够在线地更新背景模型，并有效降低运行过程中的存储量和计算量。从实验结果来看，在复杂变化的动态背景下，本文提出的方法能够得到令人满意的检测性能。

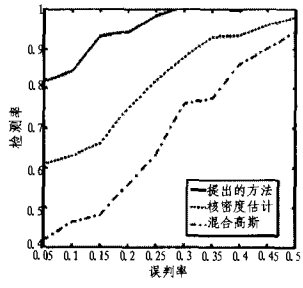

图 3　视频一算法比较 ROC 图

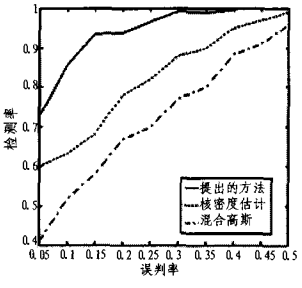

图 4　视频二算法比较 ROC 图

参 考 文 献

[1] Stauffer C, Grimson E. Adaptive background mixture models for real-time tracking// Proceeding of IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 1999: 246-25
[2] Friedman N, Russell S. Image segmentation in video sequences: A probabilistic approach// Proceeding of Conference on Uncertainty in Artifical Intelligence. 1997: 175-181
[3] Mittal A, Paragios N. Motion-based background subtraction using adaptive kernel density estimation// Proceeding of IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2004: 302-309
[4] Elgammal A, Harwood D, Davis L. Non-parametric model for background subtraction// Proceeding of European Conference on Computer Vision. Springer
[5] YangJian, ZhangDavid, YangJingyu. Two-DimensionalPCA: Anewapproach to appearance based face representation and recognition. IEEE Trans Pattern Analysis and Machine Intelligence, 2004(1): 131-137