

分层六角网格中物体边缘检测的算法^{*}

汤浩钧¹ 陈明² Richard W. Stimets³ J. T. Hsu³

(上海交通大学计算机科学系 上海 200030)¹ (同济大学交通运输与工程学院 上海 200092)²

(College of Arts and Sciences, University of Massachusetts Lowell, USA 01854)³

摘要 本文提出了在分层六角网格中基于像素灰度实现边缘检测的边缘点确定算法和边缘构造算法。文中首先介绍了六角网格结构的优点和 1:3 面积比、1:4 面积比两类六角网格金字塔,而后给出了相应的实验论证。实验结果显示应用六角网格金字塔的算法对图像数据中的缝隙与噪声具有鲁棒性,1:4 面积比六角金字塔在边缘检测中优于 1:3 面积比金字塔。

关键词 六角网格,金字塔,像素灰度,边缘检测,机器视觉

Algorithms for Boundary Detection within a Pyramid of Hexagonal Pixel Grids

TANG Hao-Jun¹ CHEN Ming² Richard W. Stimets³ J. T. Hsu³

(Department of Computer Science, Shanghai Jiao Tong University, Shanghai 200030)¹

(College of Transportation and Engineering, Tongji University, Shanghai 200092)²

(College of Arts and Sciences, University of Massachusetts Lowell, USA 01854)³

Abstract In this paper we present algorithms for determining edge points and for constructing boundaries in a hexagonal neighborhood based on pixel gray-level within a pyramid of hexagonal pixel grids. We first introduce advantages of hexagonal grid structure over square grid structure and triangular grid structure, and 1-to-3 area ratio and 1-to-4 area ratio hexagonal pyramids, and then give out corresponding experimental demonstrations. The results show that the algorithms using hexagonal pyramid are robust to gaps and noise in the image data. Besides, boundary detection in a 1-to-4 area ratio hexagonal pyramid is better than in a 1-to-3 area ratio pyramid.

Keywords Hexagonal grid, Pyramid, Gray level, Edge detection, Machine vision

1 引言

人通过眼睛观察他们周围的世界,人眼是洞悉大脑工作的窗口。视网膜作为眼球内的一层薄膜组织把原始光线转换为神经信号,人脑接受这些信号并解释成图像,机器视觉的开发基于人眼视觉系统的工作机制。视网膜完成这些工作的能力超过了当今最强大的超级计算机,然而视网膜中单个神经细胞的操作要比一般的电子器件慢 100 万倍,消耗大约 1000 万分子一的等值能量,其精确度也要比当今的数字计算机差得多。显然生物计算与数字计算是十分不同的,Mahowald 和 Mead 在神经架构与视网膜功能的启迪下尝试设计了一种硅芯片^[1]。

当今常规的单 CPU 串行计算机对于执行大多数视觉识别操作实在是太慢了,机器视觉需要以更快的速度识别更复杂的物体,这些物体可能以无穷地不可预测的方式改变(例如,奔跑着的动物、表情的变化等)。只有高度并行的算法结合类似的并行程序在相当的并行计算机上执行才能达到每帧 10~50ms 的识别速率,作为候选解决方案之一的六角并行处理首先由 Golay 提出^[2],稍后 Deutsch 作了进一步的阐述^[3]。然而,在涉及数字图像的减薄或骨架算法的大量文献中,最常使用的网格类型仍然是方形网格,Deutsch 证明比较减薄算法在方形、六角、三角网格中的处理,六角网格在处理图像不规则与噪声、在削减多余边缘点和处理速度方面优于其他两类网格^[4]。虽然六角网格的采样密度比方形网格的采样密度

要好,但目前图像输入输出设备只支持矩形点阵结构的数字图像^[5]。由于传统的方形网格最易于构造光电探测器的矩形点阵,且自然地与用于电视成像的光栅扫描系统联结,到目前为止方形网格是最常用于图像处理的。

金字塔是用于存储与处理多层分辨率图像的数据结构。金字塔底层通常储存原始图像,而金字塔高层则呈现分辨率降低的图像。层与层之间的像素大小通常以 2 倍因子缩减,同一因子用以缩减金字塔的每一层。Watson 和 Ahumada 在视觉皮层中采用六角金字塔作为图像呈现的模型^[6],Watson 为六角图像金字塔发展了算法^[7]。

虽然金字塔处理的基本思想已存在多年了,但以此基本思想为依据而设计并验证的具体方案却较少发现。六角网格金字塔是此项研究所采用的图像处理模型,六角网格不仅是最近的人眼视觉模型的研究热点,有一些证据显示人眼视网膜感光器被排列成向外扩展的六角点阵,而且六角采样图像对于信号处理有许多的优势。

2 六角网格结构的优点

网格中的边缘点定义为相邻两个像素之间的点,由当前像素和它的相邻像素的灰度差值确定。边缘点的可能位置为像素的中心和像素的棱边,图 1 内的黑色小圆点显示了在方形网格、三角网格和六角网格中的可能的边缘点位置。边缘元素定义为连接两个相邻边缘点的一段直线,通过链接边缘

^{*} 中国博士后科学基金资助(编号:2005037260)。汤浩钧 美国麻省大学博士,主要研究方向:分六角网格中图像处理与模式识别、应用语音识别技术的 E-Learning 课件与系统、网格计算的算法、验证与模型研究;陈明 博士后,主要研究方向:工程智能 CAD 技术、多媒体技术、计算机支持的协同工作;Richard W. Stimets MIT 博士,麻省大学罗威尔分校教授,主要研究方向:图像处理与模式识别、光学;J. T. Hsu 是第一作者在麻省大学罗威尔分校从事研究时的同事,主要研究方向:图像处理与模式识别。

元素构成数字图像中三维物体的边缘。

与方形网格和三角网格相比,六角网格结构具有以下 3 个优势:

第一,六角网格中允许的边缘点位置均匀地覆盖整个图像,而方形网格和三角网格中所允许的边缘点并不均匀地覆盖整个图像。

第二,六角网格中只有一个描述方向变化的角度单位。图 1(a)显示了在六角网格中,从中央像素的中心边缘点到它的最近的 6 个相邻像素的中心边缘点所构成的边缘元素有 6 个可能的方向,到 6 个相邻像素的棱边边缘点所构成的边缘元素有 6 个可能的方向,总共从中央像素的中心边缘点出发所构成的边缘元素有 12 个可能的方向,任何两个相邻的可能边缘元素之间的角度改变是唯一的,它的值 $\alpha=30^\circ$ 。描述方形网格中的方向变化需要两个不同的角度,图 1(b)显示了在方形网格中,从中央像素的中心边缘点到它的最近的 8 个相邻像素(水平的、垂直的和对角的)的中心边缘点所构成的边

缘元素有 8 个可能的方向,到 8 个相邻像素的棱边边缘点所构成的边缘元素有 8 个可能的方向。总共所构成的边缘元素有 16 个可能的方向,任何两个相邻可能边缘元素之间的角度改变不是唯一的,有两种角度改变: $\alpha=\tan^{-1}(1/2)\approx 26.565^\circ$, $\beta=45^\circ-\alpha\approx 18.435^\circ$ 。在三角网格中,需要四个不同的角度来描述方向的变化。图 1(c)显示,从中央像素的中心边缘点到它的最近的 3 个相邻像素的中心边缘点所构成的边缘元素有 3 个可能的方向,到次级最近的 6 个相邻像素的中心边缘点所构成的边缘元素有 6 个可能的方向,到再次级最近的 3 个相邻像素的中心边缘点所构成的边缘元素有 3 个可能的方向,到最近和次级最近的 6 个相邻像素的棱边边缘点所构成的边缘元素有 6 个可能的方向,到次级最近和再次级最近的 6 个相邻像素的棱边边缘点所构成的边缘元素有 6 个可能的方向,总共所构成的边缘元素有 24 个可能的方向,任何两个相邻可能边缘元素之间的角度改变不是唯一的,有 4 种角度改变: $\alpha\approx 19.107^\circ$, $\beta\approx 10.893^\circ$, $\gamma\approx 16.102^\circ$, $\delta\approx 13.898^\circ$ 。

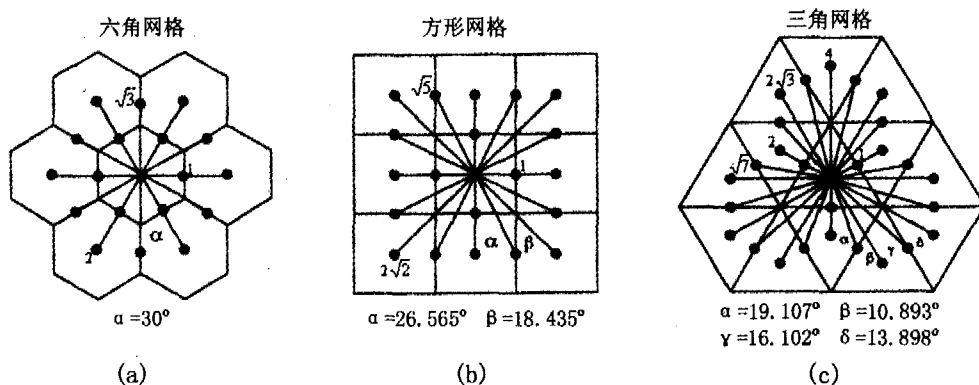


图 1 在(a)六角网格、(b)方形网格、(c)三角网格中的角度单位和边缘元素长度

第三,六角网格中的边缘元素的长度几乎是一致的,如图 1(a)所示,有两种边缘元素的长度:1 和 $\sqrt{3}$ 。因此,所有边缘元素的长度都精确等于 1 倍或近似等于 2 倍的基本单位长度。在方形网格中,有 3 种边缘元素长度:1, $\sqrt{2}$ 和 $\sqrt{5}$,如图 1(b)所示。在三角网格中,有 4 种边缘元素长度:1, $\sqrt{7}$, $2\sqrt{3}$ 和 4,如图 1(c)所示。不比六角网格,方形与三角网格中的边缘元素长度都不接近基本单位长度的整数倍。

3 边缘点确定算法

确定边缘点位置的算法如下所示。如图 2 考虑一个六角像素和它的六个相邻像素,对应的六角像素灰度值分别为 $f_0, f_1, f_2, f_3, f_4, f_5$ 和 f_6 ,中央像素的六条棱边分别为“a”、“b”、“c”、“d”、“e”和“f”。假设灰度差的阈值为 Δf_0 (可调参量),比率阈值 $r_0=3.0$ 。

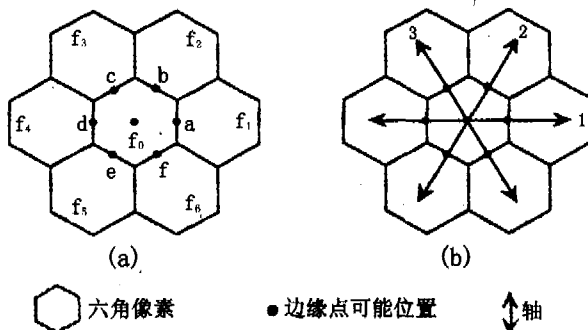


图 2 (a)六角像素中边缘点的可能位置;(b)六角网格的三条主轴

1) 计算相邻六角像素之间的灰度差值:

$$\Delta f_{10} = f_1 - f_0 \quad \Delta f_{20} = f_2 - f_0 \quad \Delta f_{30} = f_3 - f_0 \\ \Delta f_{04} = f_0 - f_4 \quad \Delta f_{05} = f_0 - f_5 \quad \Delta f_{06} = f_0 - f_6$$

2) 确定边缘点,考虑沿主轴 1 的 Δf_{10} 和 Δf_{04} :

a) IF Δf_{10} 与 Δf_{04} 的符号相反, THEN

(1) IF $|\Delta f_{10}| \geq \Delta f_0$, THEN 置放边缘点于当前像素的棱边“a”上;

(2) IF $|\Delta f_{04}| \geq \Delta f_0$, THEN 置放边缘点于当前像素的棱边“d”上。

b) IF Δf_{10} 与 Δf_{04} 的符号相同, THEN

(1) IF $|\Delta f_{10}| + |\Delta f_{04}| \geq \Delta f_0$, THEN

IF $\frac{|\Delta f_{10}|}{|\Delta f_{04}|} \geq r_0$, THEN 置放边缘点于当前像素的棱边“a”上

ELSEIF $\frac{|\Delta f_{04}|}{|\Delta f_{10}|} \geq r_0$, THEN 置放边缘点于当前像素的棱边“d”上

ELSE 置放边缘点于当前像素的中心;

(2) IF $|\Delta f_{10}| + |\Delta f_{04}| < \Delta f_0$, THEN 没有边缘点。

应用上述步骤 a、b 于沿主轴 2 的 Δf_{20} 、 Δf_{05} , 和沿主轴 3 的 Δf_{30} 、 Δf_{06} 。

3) 如果具有最大强度 $|\Delta f|$ 的边缘点位于像素的中心, 那么

a. 削除该像素棱边上的所有的可能边缘点;

b. 标记该中心边缘点的方向具有最大的 $|\Delta f|$ 。

4) 最终像素棱边上的边缘点必须由分享棱边的相邻像素确立。

4 六角网格金字塔

六角网格金字塔使系统具有鲁棒性, 即对图像数据中缝隙和噪声的容忍性, 例如, 若一个数字图像中的物体边缘出现缝隙, 则在较大六角网格中构成的物体边缘可用于指导此边缘缝隙的插值。

六角金字塔底层呈现分辨率最高的数字图像。例如六角网格 1, 金字塔较高层呈现对应的分辨率降低的数字图像, 层次越高, 数字图像的分辨率越低。六角像素的大小在不同的层次是不同的, 层次越高, 六角像素的尺寸越大。在我们的实验中使用两类六角金字塔, 1:3 面积比的金字塔和 1:4 面积比的金字塔。

对于 1:3 面积比的金字塔, 不同层六角网格中的像素方位是不同的。假设较小尺寸六角像素在六角网格 n 中, 下一层较大的六角像素在六角网格 $n+1$ 中, 六角网格 n 和 $n+1$ 中的像素方位相差 30° , 而六角网格 n 和 $n+2$ 中的像素具有

相同的方位。对于 1:4 面积比的金字塔, 不同层六角网格中的像素都具有相同的方位。

4.1 1:3 面积比金字塔-交替的方位

图 3(a) 表示了 1:3 面积比金字塔中一个六角像素和下一层较大六角像素之间的相互关系。假设图 3(a) 中的小像素在六角网格 n 中, 大像素在六角网格 $n+1$ 中, 则网格 $n+1$ 中的一个像素由网格 n 中的一个中央像素和六个环绕的 $1/3$ 相邻像素组成, 总共网格 $n+1$ 中的一个像素包含了 3 个网格 n 中的像素, 因此, 当前层的像素与下一层较大像素的面积比为 1:3。

网格 $n+1$ 中每一个像素灰度值可以从网格 n 中的像素灰度值计算得到, 其公式如下:

$$F = \frac{1}{3} \times [f_0 + \frac{1}{3} \times (f_1 + f_2 + f_3 + f_4 + f_5 + f_6)]$$

其中, F = 网格 $n+1$ 中的一个六角像素灰度数值, f_0 = 网格 n 中的中央六角像素灰度数值, $f_1, f_2, f_3, f_4, f_5, f_6$ = 网格 n 中的中央像素的六个相邻像素灰度数值。

4.2 1:4 面积比金字塔-相同的方位

图 3(b) 表示了 1:4 面积比金字塔中一个六角像素和下一层较大六角像素之间的相互关系。假设图 3(b) 中的小像素在六角网格 n 中, 大像素在六角网格 $n+1$ 中, 网格 $n+1$ 中的一个像素由网格 n 中的一个中央像素和六个环绕的 $1/2$ 相邻像素组成, 总共网格 $n+1$ 中的一个像素包含了 4 个网格 n 中的像素, 因此, 当前层的像素与下一层较大像素的面积比为 1:4。

网格 $n+1$ 中每一个像素灰度值可以从网格 n 中的像素灰度值计算得到, 其公式如下:

$$F = \frac{1}{4} \times [f_0 + \frac{1}{2} \times (f_1 + f_2 + f_3 + f_4 + f_5 + f_6)]$$

其中, F = 网格 $n+1$ 中的一个六角像素灰度数值, f_0 = 网格 n 中的中央六角像素灰度数值, $f_1, f_2, f_3, f_4, f_5, f_6$ = 网格 n 中的中央像素的六个相邻像素灰度数值。

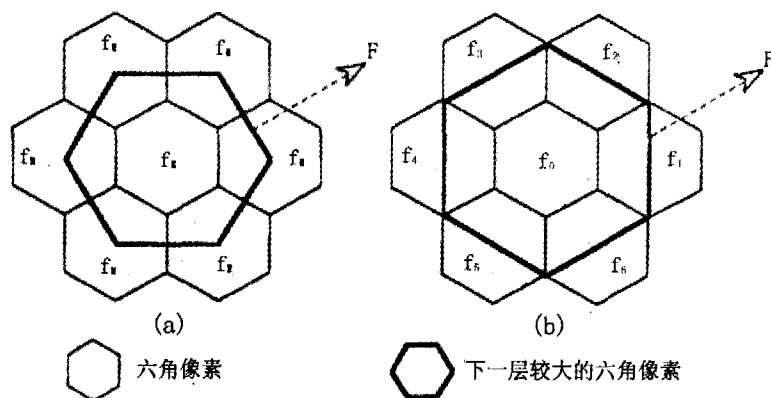


图 3 (a)1:3 面积比和(b)1:4 面积比六角金字塔中网格 n 和网格 $n+1$ 的相互关系及其像素灰度数值

5 六角邻域内的边缘构造算法

六角邻域定义为由一个中央六角像素和六个 $1/2$ 相邻像素组成, 如图 4 所示。

5.1 从六角邻域内的极大值边缘点出发搜索和构造边缘的第一分枝

1) 在指定的六角邻域内查找强度为极大值的边缘点位置。例如在图 4 中, 该六角邻域的极大强度值边缘点是 63, 这一极大值边缘点位于中央像素的中心。

2) 查找极大值边缘点附近最大的边缘点, 它的方向与极大值边缘点相差 60° 或更小, 从极大值边缘点连接这个边缘点构成一个边缘元素。在图 4 中, 极大值边缘点附近最大的边缘点是 62。另外, 强度 62 边缘点的方向与极大值边缘点相差 60° , 因此从极大值边缘点连接强度 62 的边缘点产生一个边缘元素, 这个边缘元素在图 4 中以虚线表示。

3) 在步骤 2 中找到的边缘点附近查找最大的相邻边缘点, 并且在它们之间产生边缘元素, 在查找这个最大相邻边缘点时必须遵从下面的规则:

a)这个边缘点的方向与步骤2中找到的边缘点相差 30° 或更小;

b)这个边缘点与步骤2中找到的边缘点的方向都不平行于由这两个边缘点产生的边缘元素的方向。

在图4中,由于强度62边缘点附近所有的相邻边缘点都不遵从上述规则,这个步骤没有找到新的边缘点。在某些情况下新的边缘点可能在这个步骤中找到。

4)基于步骤3中的规则继续查找边缘点直到不再找出新的边缘点为止。

5.2 从六角邻域内的极大值边缘点出发搜索和构造边缘的第二分枝

5)查找极大值边缘点附近最大的相邻边缘点,并且在它们之间产生一个边缘元素,在查找最大的相邻边缘点时应该遵从下面的规则:

a)这个新的最大边缘点并不贡献于边缘的第一分枝;

b)这个边缘点的方向与极大值边缘点方向相差 30° 或更小;

c)这个边缘点与极大值边缘点的方向都不平行于这两个边缘点产生的边缘元素方向;

d)从极大值边缘点察看本步骤产生的边缘元素的方向和步骤2产生的边缘元素的方向,这两个方向之间相差 60° 或更大。

图4中,在极大值边缘点附近并不贡献于边缘的第一分枝的最大强度边缘点是55,这个边缘点的方向与极大值边缘点相同,强度55边缘点和极大值边缘点之间产生的边缘元素的方向与步骤2中产生的边缘元素的方向相差 150° ,这个边缘点和极大值边缘点的方向都不平行于由这两个边缘点产生的边缘元素的方向,因此找到了这个最大相邻边缘点,并且在强度55边缘点和极大值边缘点之间产生一个边缘元素。

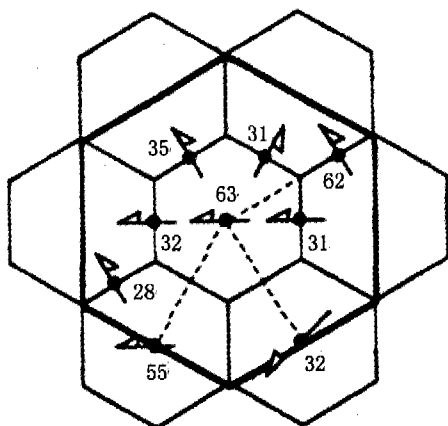


图4 六角邻域内的边缘构造

6)在步骤5中找到的边缘点附近查找最大相邻边缘点,并且在它们之间产生一个边缘元素,查找这个最大相邻边缘点时应该遵从下面的规则:

a)这个新的最大边缘点并不贡献于边缘的第一分枝和步骤5中产生的边缘元素;

b)这个边缘点的方向与步骤5中找到的边缘点相差 30° 或更小;

c)这个边缘点和步骤5中找到的边缘点的方向都不平行于由这两个边缘点之间产生的边缘元素。

图4中,由于强度55边缘点附近的所有相邻边缘点都不遵从上述规则,这个步骤没有找到新的边缘点。在某些情况

下新的边缘点可能在这个步骤中找到。

7)基于步骤6中的规则继续查找边缘点直到不再找出新的边缘点为止。

5.3 从六角邻域内的极大值边缘点出发搜索和构造边缘的第三分枝

8)查找极大值边缘点附近最大的相邻边缘点,并且在它们之间产生一个边缘元素,在查找最大的相邻边缘点时应该遵从下面的规则:

a)这个新的最大边缘点并不贡献于边缘的第一分枝和第二分枝;

b)这个边缘点与极大值边缘点的方向相差 30° 或更小;

c)这个边缘点与极大值边缘点的方向都不平行于这两个边缘点产生的边缘元素方向;

d)从极大值边缘点察看本步骤产生的边缘元素的方向与步骤2产生的和步骤5产生的边缘元素的方向都相差 60° 或更大。

图4中,在极大值边缘点附近并不贡献于边缘的第一和第二分枝的最大强度边缘点是32,这个边缘点的方向与极大值边缘点相差 60° ,强度32边缘点和极大值边缘点之间产生的边缘元素的方向与步骤2中产生的边缘元素的方向相差 90° ,与步骤5中产生的边缘元素的方向相差 60° ,这个边缘点和极大值边缘点的方向都不平行于这两个边缘点产生的边缘元素的方向,因此找到了这个最大相邻边缘点,且在强度32边缘点和极大值边缘点之间产生一个边缘元素。

9)在步骤8中找到的边缘点附近查找最大的相邻边缘点,并且在它们之间产生一个边缘元素,查找这个最大相邻边缘点时应该遵从下面的规则:

a)这个新的最大边缘点并不贡献于边缘的第一和第二分枝和步骤8中产生的边缘元素;

b)这个边缘点的方向与步骤8中找到的边缘点相差 30° 或更小;

c)这个边缘点和步骤8中找到的边缘点的方向都不平行于由这两个边缘点之间产生的边缘元素。

图4中,由于强度32边缘点附近的所有相邻边缘点都不遵从上述规则,这个步骤没有找到新的边缘点。在某些情况下新的边缘点可能在这个步骤中找到。

10)基于步骤9中的规则继续查找边缘点直到不再找出新的边缘点为止。

6 实验结果与分析

6.1 池形物体不同分辨率的边缘检测

池形物体外形呈矩形但左右两边为圆弧状,其内部是白色的,外部是黑色的。第一步是数字化池形物图像,在六角网格上覆盖池形物图像,估测对应的像素灰度值,为方便起见,该六角网格称为“六角网格1”。池形物图像白色区域内的每个像素灰度值指示为100,黑色区域内每个像素灰度值指示为0,由于图像边界上的像素灰度值与其在图像理想边界内的面积成正比^[8],例如,若一个像素的40%面积在白色区域,60%的面积在黑色区域,那么此像素的灰度值为 $100 \times 40\% + 0 \times 60\% = 40$ 。图5显示了六角网格1中的池形物数字图像。

池形物图像的边缘通过链接边缘元素得到,因此第二步根据第3节的边缘点确定算法查找池形物数字图像中的边缘点。图5(b)显示了在六角网格1中池形物数字图像对应的边缘点,箭头代表边缘点的方向,箭头的方向指向相邻像素灰

度值递增的方向。图 5(b)内所示的插值点被置放于长度为 $\sqrt{3}$ 和 2 的每个边缘元素的中间,因此从边缘点到边缘点或者从边缘点到插值点的长度都精确等于或近似等于 1。

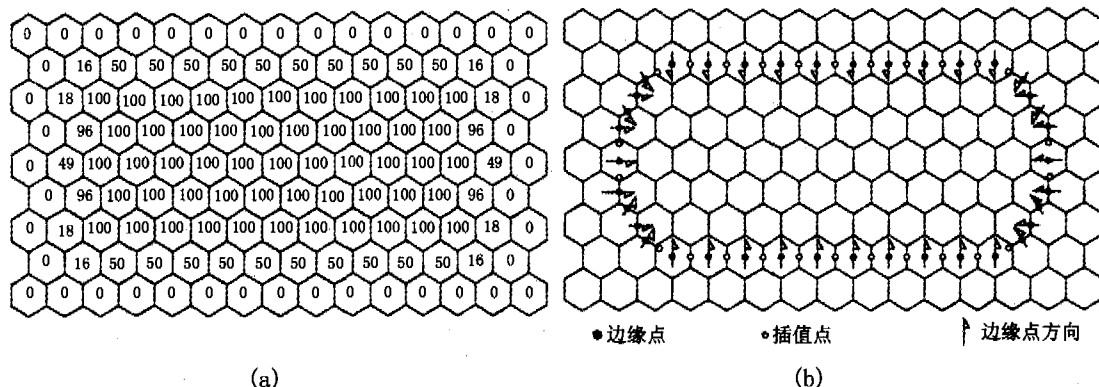


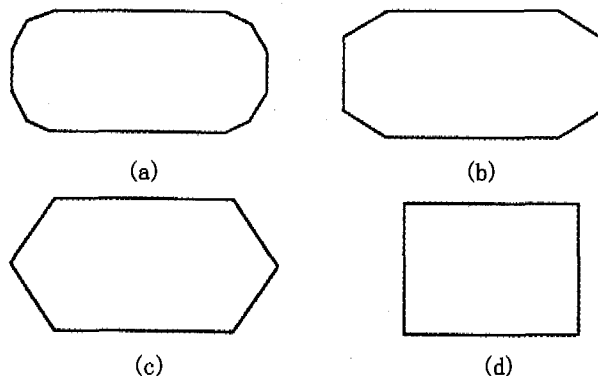
图 5 六角网格 1 中的(a)池形物数字图像,像素内的数字代表对应的灰度值;(b)带有边缘点的池形物图像

用相同的方法,基于 1:3 面积比六角金字塔在其他层六角网格中构成池形物边缘,图 6 比较了在六角网格 1、2、3 和 4 中的四种池形物图像的边缘,图 6(a)表示六角网格 1 中的池形物边缘,边缘的左边和右边各有六个顶点;图 6(b)表示六角网格 2 中的池形物边缘,边缘的左边和右边各有四个顶点;图 6(c)表示六角网格 3 中的池形物边缘,边缘的左边和

边缘点确立后,下一步是根据六角领域内的边缘构造算法连接相邻边缘点产生边缘元素最终生成六角网格 1 中池形物图像的边缘如图 6(a)所示。

右边各有三个顶点;图 6(d)表示六角网格 4 中的池形物边缘,边缘的左边和右边各有两个顶点。

六角网格 1 和 2 中构成的边缘可以被识别为池形物,而六角网格 3 和 4 中构成的边缘不能被识别为池形物,图 6 中的对比显示了较小像素六角网格构成了分辨率较高的池形物边缘。



(a)六角网格 1;(b)六角网格 2;(c)六角网格 3;(d)六角网格 4 中的池形物边缘。

图 6 六角网格 1、2、3 和 4 中的 4 种边缘的比较

曲率是其中最简单的边缘描述符,曲率定义为坡度的变化比率,沿池形物边缘的曲率的确定如下所示。 $[\theta_i]$ 定义为沿边缘反时针方向相邻边缘元素形成的角度的集合, $[\Delta\theta_i]$ 是曲率的集合,定义为

$$\Delta\theta_i = \theta_i - \theta_{i-1}$$

同时引入平均曲率的概念,平均曲率的集合 $[\overline{\Delta\theta_i}]$ 定义为

$$\overline{\Delta\theta_i} = \frac{1}{4} \Delta\theta_{i-1} + \frac{1}{2} \Delta\theta_i + \frac{1}{4} \Delta\theta_{i+1}$$

Stimets, Wu 和 Rajasekharan 的研究工作显示,由于边缘元素的位置错误,导致 $\Delta\theta$ 曲线中的噪声较大,而 $\overline{\Delta\theta}$ 曲线中的噪声显著降低,曲线峰谷的识别变得容易得多^[9]。

图 7 显示了从原池形物图像底边中点出发沿着图像边界的反时针方向的曲率相对边界位置的绘图。图 8(a)、(b)、(c)和(d)分别显示了六角网格 1、2、3 和 4 中沿池形物边缘的曲率 $\Delta\theta$ 相对边缘位置的绘图。

图 8(a)显示了六角网格 1 中的 12 个相等的尖峰,每个峰表示在该位置两个相邻边缘元素之间有一个角度差,每个峰

值为 1 意指角度差为 30° ,很明显。这 12 个峰可以分成两个对等的组,每组有 6 个峰。每组中两个峰间值为 0 的位置表示图 6(a)中池形物左侧和右侧边缘的平坦部分,这两组峰分别对应边缘的左、右侧,两组峰之间值为 0 的位置表示边缘的顶端或底部上的平坦区域。

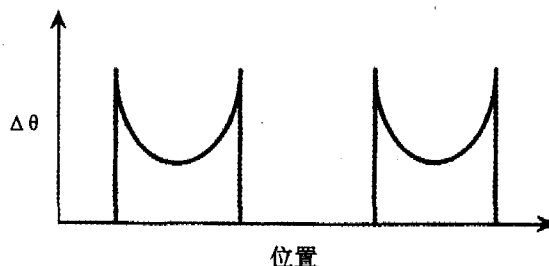


图 7 原池形物图像边缘的曲率

图 8(b)显示六角网格 2 中有 8 个尖峰,意味着沿池形物边缘相邻两个边缘元素之间共有 8 个角度差。很明显,这 8 个峰可以分成两个对等的组,每组有 4 个峰。每组中两个峰

值为 2 代表角度差为 60° ，外侧两个峰值为 1 代表角度差为 30° ，每组中两个峰间值为 0 的位置表示图 6(b) 中池形物左侧和右侧边缘的平坦部分，这两组峰之间值为 0 的位置表示边缘的顶端或底部上的平坦区域。

图 8(c) 显示六角网格 3 中有 6 个尖峰，意味着共有 6 个角度差，六个峰值为 2 代表角度差为 60° ，很明显，这 6 个峰可

以分成两个对等的组，每组有 3 个峰。两组峰之间值为 0 的位置表示边缘的顶端或底部上的平坦区域。

图 8(d) 显示六角网格 4 中有 4 个尖峰，意味着共有 4 个角度差，四个峰值为 3 代表角度差为 90° ，很明显，这 4 个峰可以分成两个对等的组，每组有 2 个峰。两组峰之间值为 0 的位置表示边缘的顶端或底部上的平坦区域。

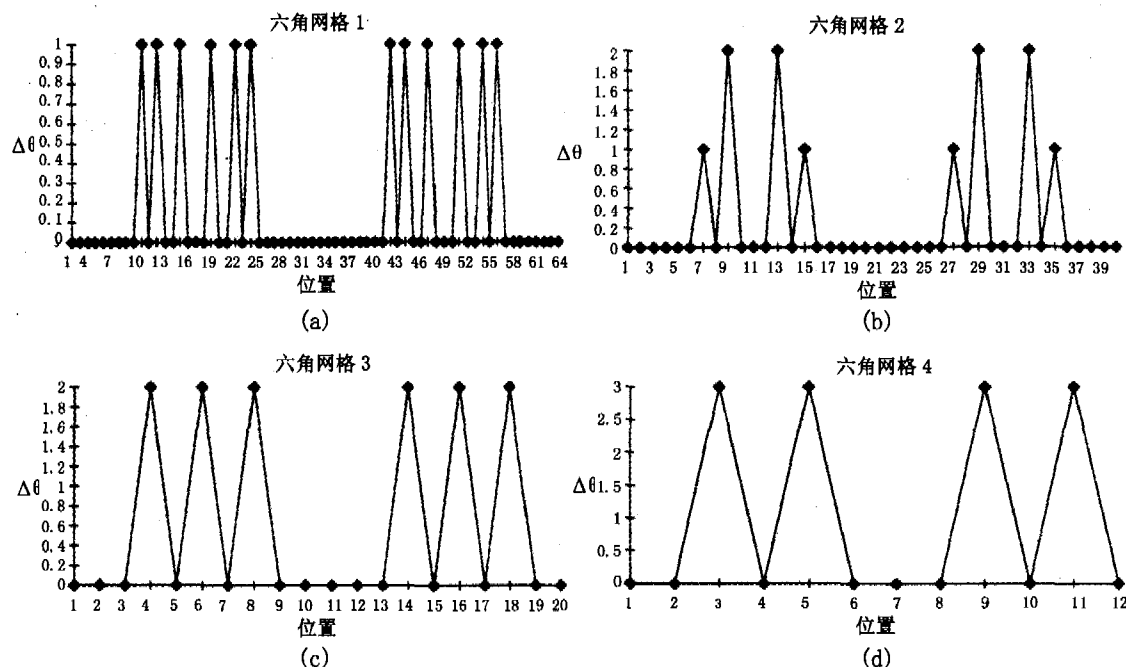


图 8 (a)六角网格 1; (b)六角网格 2; (c)六角网格 3; (d)六角网格 4 中的池形物边缘的曲率

图 9(a)、(b)、(c) 和 (d) 表示了池形物边缘的平均曲率，在图 7 中理想的原池形物图像的曲率绘图可以观察到两个主峰都有最小值，这两个最小值也可以在对应于六角网格 1、2

的图 9(a)、图 9(b) 中观察到，但是不能在对应该于六角网格 3、4 的图 9(c)、图 9(d) 中观察到。因此，在池形物图像中似乎至少需要 3 个像素才能看到这个最小值。

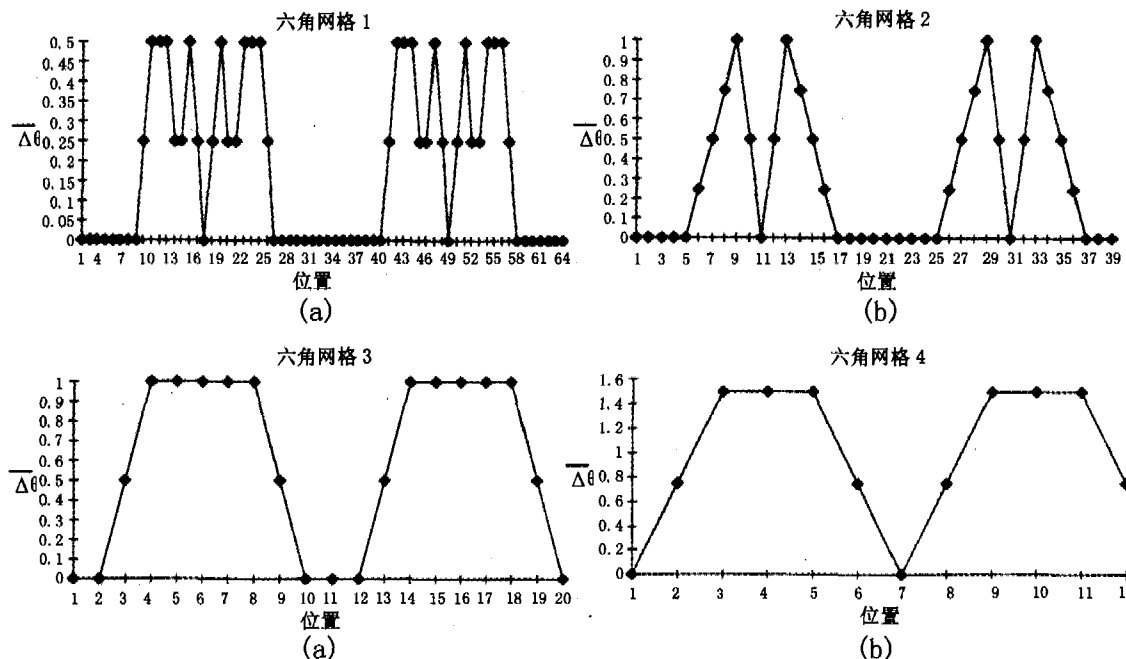


图 9 (a)六角网格 1; (b)六角网格 2; (c)六角网格 3; (d)六角网格 4 中池形物边缘的平均曲率

6.2 三维凸缘物体图像的边缘检测

三维的凸缘物体图像用于构成六角数字图像，图 10 显示

了此凸缘图像。凸缘顶部有一个空芯的圆柱体，有三个小的椭圆是凸缘的螺孔，图像中凸缘的背景为白色。

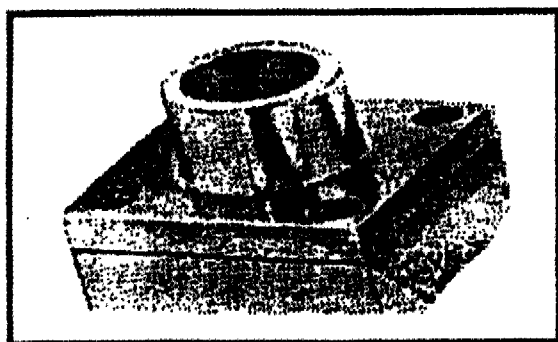
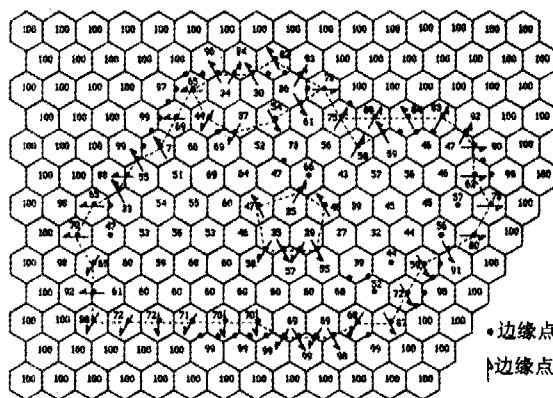


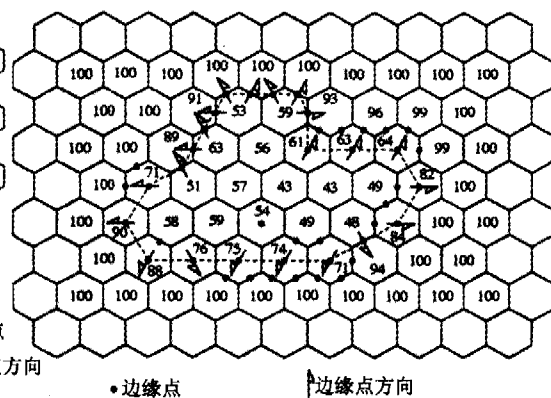
图 10 凸缘图像

凸缘图像首先被数字化,以便于计算机的边缘检测,由于凸缘图像比池形物更复杂,并不是所有的边缘点都贡献于构造凸缘图像的边缘,依照上述 A 中的处理步骤应用边缘点确定算法和六角领域内的边缘构造算法,图 11 显示了六角网格 1 中的凸缘边缘和对应的边缘点,凸缘顶部的椭圆容易看到,外部轮廓的拐角也相当明显。

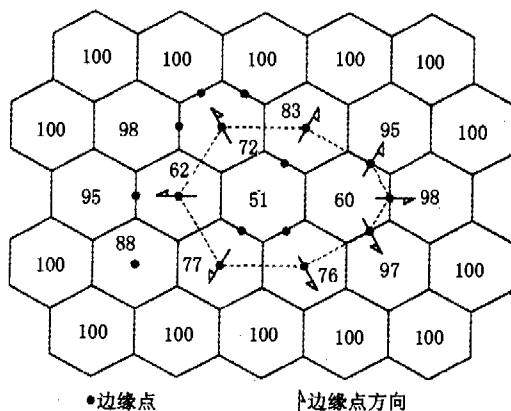
基于 1:4 面积比六角金字塔,在其他层六角网格中构造凸缘的边缘,图 12(a)显示了在六角网格 2 中凸缘数字图像的边缘,凸缘的外部轮廓的拐角落变得模糊,椭圆依然存在,除了椭圆,六角网格 1 中的边缘内消失的螺孔在此出现了。



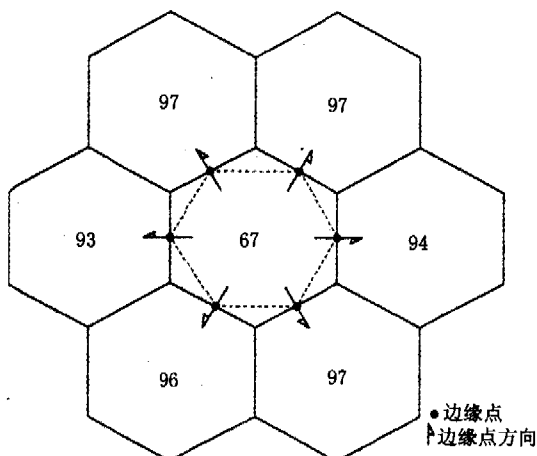
(a)



(b)



(c)

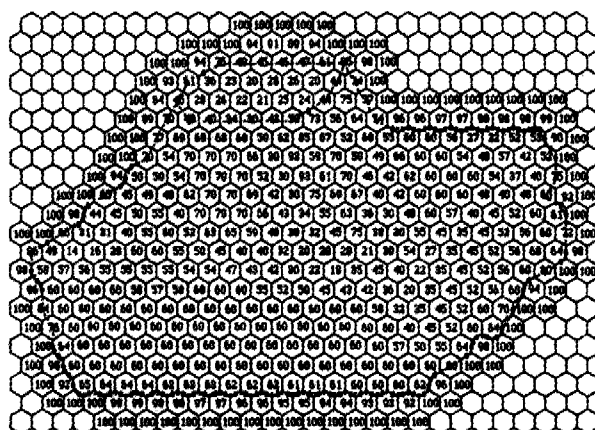


(d)

虚线代表链接边缘元素所构成的凸缘边缘。

图 12 (a)六角网格 2;(b)六角网格 3;(c)六角网格 4;(d)六角网格 5 中带有边缘点的凸缘数字图像

中间螺孔形状没有在六角网格 1 中形成的原因是由于螺孔周围特殊反射条纹带来的噪声,虽然六角网格 1 的像素尺寸较小,可以得到更具体的信息,但同时也得到了更强的噪声,而六角网格 2 的像素尺寸较大,中间螺孔周围的噪声不足以严重影响中间螺孔形状的构造。



虚线代表链接边缘元素所构成的边缘。

图 11 六角网格 1 中带有边缘点的凸缘数字图像

图 12(b)显示了六角网格 3 中凸缘数字图像的边缘,凸缘的外部轮廓已变得更加模糊,并且椭圆消失了,凸缘图像的边缘内部没有构造任何形状。图 12(c)显示了在六角网格 4

中的凸缘图像的边缘,凸缘图像的轮廓已不能识别。图 12(d)显示了六角网格 5 中的凸缘图像的边缘,凸缘的边缘变成六角形了,并且包含在一个六角像素内。

表1给出了基于上述讨论从网格1到网格5的不同层六角网格中的凸缘边缘的比较。共有7个凸缘图像的特征:外形轮廓、顶部的椭圆、圆柱体、圆柱体的阴影、左边的螺孔、中间的螺孔和右边的螺孔。

通常,从较低分辨率层到较高分辨率层操作,也就是从网格5到网格1,所检测到的图像边缘越来越精确,伴随各种不同图像特征的新边缘出现了。然而,我们的实验遇到一个情形,在较低分辨率层检测到的一个特征,即网格2上的中间螺孔,并没有在较高分辨率层检测到,因为网格1中的噪声足以抑制中间螺孔的出现。一个更包容的方案可以让较低分辨率层得到的结果指导在较高分辨率层中的边缘的构造,使得此类“消失”的特征不再发生。

表1 三维凸缘物体在六角网格金字塔结构中的分析

边缘特征	在第几层网格观察到
外形轮廓	5(没有形状);4(右边拉伸); 3(只有顶端的外形轮廓);2;1
顶部的椭圆	2;1
圆柱体	
圆柱体的阴影	
左边的螺孔	1
中间的螺孔	2
右边的螺孔	

结论和建议 在此项研究中,我们介绍了1:3和1:4面积比两类六角网格金字塔,并且分别用于实验。1:4面积比金字塔较1:3面积比金字塔更好,这是由于基于1:4面积比金字塔,六角像素在各层网格中的方向是不变的,为一层网格开发的算法可以应用于金字塔中的所有层网格。

较小像素的六角网格中可以识别物体边缘更多的特征,在池形物体边缘检测的实验中,六角网格1,即最小像素的六角网格中构成的物体边缘最接近于真实的池形物轮廓;而在六角网格4,即最大像素的六角网格中构成的物体边缘与真实池形物轮廓最不同,仅从六角网格4得到的物体边缘几乎

不能识别池形物。在三维凸缘的边缘检测实验中,应用边缘检测算法可以在较小像素网格中识别凸缘较多的特征,凸缘边缘可以在六角网格1,2和3的数字图像上得到识别,凸缘的圆柱体顶部的椭圆可以在六角网格1和2中被识别,凸缘右边的螺孔只能在六角网格2中被观察到。

反射带来的明亮斑点和条纹作为噪声抑制了较高分辨率层六角网格中的特征检测。在实验中,凸缘的其中一个特征不能在高分辨率层网格构成的图像上观察到,而能在较低分辨率层网格构成的图像上看到。因此,特别是对于含有大量混乱细节和/或精细纹理的图像,有必要使用较低分辨率层获取的结果来指导搜索较高分辨率层中的模式。

此项研究中的边缘点确定算法需对一维物体的特殊情况加以补充。举例来说,如果图像中包含窄带形状,窄带的宽度稍稍比一个像素的尺寸小一些,而边缘点只要其方向彼此相反既可位于像素的中心,也可位于像素的棱边。这一规则的增加会生成更加精确的窄带检测。

参考文献

- 1 Mahowald M A, Mead C. The Silicon Retina. *Scientific American*, 1991, 264(5): 76~82
- 2 Golay M J E. Hexagonal parallel pattern transformations. *IEEE Transactions on Computers*, 1969, C-18(8): 733~740
- 3 Deutsch E S. On parallel operations on hexagonal arrays. *IEEE Transactions on Computers*, 1970, C-19(10): 982~983
- 4 Deutsch E S. Thinning algorithms on rectangular, hexagonal, and triangular arrays. *Communications of the ACM*, 1972, 15(9): 827~837
- 5 LU Xi-Qun, CHEN Chun. Research on Hexagonal Sampled Digital Images. *Journal of Image and Graphics*, 2004(6)
- 6 Watson A B, Ahumada A. A hexagonal orthogonal-oriented pyramid as a model of image representation in visual cortex. *IEEE Transactions on Biomedical Engineering*, 1989, 36(1)
- 7 Watson A B. Recursive, in-place algorithm for the hexagonal orthogonal oriented quadrature image pyramid. *SPIE Advances in image compression and automatic target recognition 1099*, 1989
- 8 Crow F C. The aliasing problem in computer-generated shaded images. *CACM*, 20(12)
- 9 Stimets R W, Wu Liying, Rajasekharan K. Rapid recognition of object outlines in reduced resolution images. *Pattern Recognition*, 1986, 19(1): 21~33

(上接第241页)

从图3(a)可见,当包数 N 从40~200变化,HLP-LS1算法相对文[6]算法的期望PSNR增加量近似从0.45dB单调递减到0.05dB;而HLP-LS2算法相对文[6]算法的期望PSNR增加量在0.05dB左右变化。因此,小的包数、较大的包长度有利于HLP-LS1算法获得较大幅度的期望PSNR性能改进;而HLP-LS2算法的性能改进幅度较小,基本上稳定在0.05dB左右。另外,由图3(b)、(c)可以看出,当信道条件变化时,本文HLP-LS1和HLP-LS2算法的期望PSNR值能够稳定地优于文[6]算法。当平均突发长度由小变大时,该两种算法的性能改进有明显增加的趋势。

结论 本文在原ULP/HLP方案的基础上,通过对码率游程内的信源符号进行重新排列,增加了接收端用于解码的信源符号总数,从而提高了重建图像的质量,在此改进的基础上,导出了重排时的期望PSNR计算式,提出了在性能优先和速度优先两种情况下的重排策略,以及考虑重排效果和最小质量要求的基于局部搜索的信源/信道码率分配算法——HLP-LS1和HLP-LS2。实验结果表明,两种算法的期望PSNR性能都优于文[6]算法,其中HLP-LS2算法的改善幅度较小,但运行速度快,而HLP-LS1算法的改善幅度较大,但运行速度稍慢,同时发现HLP-LS1算法有利于辅助信息

的减少。

参考文献

- 1 Mohr A E, Riskin E A, Ladner R E. Unequal loss protection: graceful degradation of image quality over packet erasure channels through forward error correction [J]. *IEEE J Select Areas Commun*, 2000, 18(7): 819~828
- 2 Mohr A E, Ladner R E, Riskin E A. Approximately optimal assignment for unequal loss protection [A]. In: *ICIP'2000, Vol 1* [C]. Vancouver: IEEE press, 2000. 367~370
- 3 Puri R, Ramchandran K. Multiple description source coding through forward error correction codes [A]. In: *Proc. 33rd Asilomar Conf. Signals, Systems, and Computers, Vol 1* [C]. Pacific Grove, CA: IEEE Press, 1999. 342~346
- 4 Dumitrescu S, Wu X, Wang Z. Globally optimal uneven error-protected packetization of scalable code streams [J]. *IEEE Trans Multimedia*, 2004, 6(2): 230~239
- 5 Stankovic V, Hamzaoui R, Xiong Z. Real-time error protection of embedded codes for packet erasure and fading channels [J]. *IEEE Trans Circuits Syst Video Tech*, 2004, 14(8): 1064~1072
- 6 Grangetto M, Magli E, Olmo G. Ensuring quality of service for image transmission: hybrid loss protection [J]. *IEEE Trans Image Processing*, 2004, 13(6): 751~756
- 7 Horn U, Stuhlmüller K, Link M, et al. Robust Internet video transmission based on scalable coding and unequal error protection [J]. *Image Communication (Special Issue on Real-time Video over the Internet)*, 1999, 15(1-2): 77~94
- 8 Taubman D, Marcellin M. *JPEG2000: Image Compression, Fundamentals, Standards, and Practice* [M]. USA: Kluwer Academic Publishers, 2002
- 9 Said A, Pearlman W A. A new fast and efficient image codec based on set partitioning in hierarchical trees [J]. *IEEE Trans Circuits Syst Video Tech*, 1996, 6(3): 243~250