

[19] 中华人民共和国国家知识产权局

[51] Int. Cl⁷

G06K 9/36

G06K 9/46



[12] 发明专利说明书

[21] ZL 专利号 98806856.7

[45] 授权公告日 2004 年 4 月 14 日

[11] 授权公告号 CN 1145906C

[22] 申请日 1998.5.6 [21] 申请号 98806856.7

[30] 优先权

[32] 1997.5.7 [33] US [31] 60/045,915

[86] 国际申请 PCT/US1998/009290 1998.5.6

[87] 国际公布 WO98/50886 英 1998.11.12

[85] 进入国家阶段日期 2000.1.3

[71] 专利权人 界标制图有限公司

地址 美国得克萨斯州

[72] 发明人 I·D·黑尔

审查员 哈雅坤

[74] 专利代理机构 上海专利商标事务所

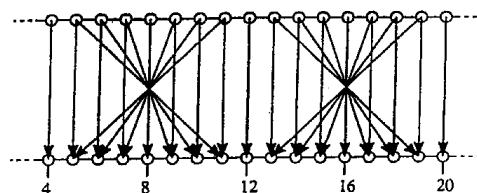
代理人 李 湘

权利要求书 2 页 说明书 14 页 附图 6 页

[54] 发明名称 数据压缩方法

[57] 摘要

在根据本发明对 JPEG 压缩算法的适用性修改中可以重复使用算法中的许多部分。通过简单地将正和逆变换互换就可以对长度为 8 的 DCT 重复使用 JPEG 方法。JPEG 量化方法经过修改避免了对小波数的优先处理，并且处理了数据幅值在块与块之间的变化。在正 DCT 之前也可以跨越块边界折叠样本值，并且在逆 DCT 之后展开这样的值。这种折叠与展开抑制了 JPEG 算法压缩图像内可觉察的分块效应。



I S S N 1 0 0 8 - 4 2 7 4

1. 一种进行地震勘探的数据压缩方法，其特征在于，它包含下述步骤：

收集由 N 个地震信号组成的一维阵列 X，

5 使所述地震信号阵列数字化，生成所述一维阵列的 N 个取样，并通过下述方法，压缩所述阵列，

(a) 将所述阵列 x 分割为具有 M 个样本的块，这里 $M < N$ ；

(b) 根据下列方程折叠跨越每个块边界 1 的样本 $x_1(j) = x(/M+j)$

$$\begin{aligned} 10 \quad y_1(j) &= f(j) x_1(j) + f(-j) x_1(-j), \\ y_1(-j) &= f(j) x_1(-j) - f(-j) x_1(j), \\ I &= 1, 2, \dots, N/M-1, \\ j &= 1, 2, \dots, M/2-1, \\ y_1(j) &= x_1(j), \quad \text{否则} \end{aligned}$$

15 这里

$$f(j) = \sin \left[\frac{\pi}{4} \left(1 + \frac{2j}{M} \right) \right],$$

(c) 根据下列方程式将阵列 y 每个块内的折叠样本进行变换

$$\begin{aligned} 20 \quad z(k) &= \sum_{j=0}^{M-1} y(j) b(j) \sqrt{\frac{2}{M}} \cos \left[\frac{\pi(2k+1)j}{2M} \right], \\ k &= 0, 1, \dots, M-1 \end{aligned}$$

这里

$$b(j) = \begin{cases} \frac{1}{\sqrt{2}}, & j=0 \\ 1, & \text{否则} \end{cases}$$

25 (d) 对阵列 z 的每个块内的已变换样本进行量化以生成整数；以及

(e) 将所述整数编码为代表压缩阵列的位流。

2. 如权利要求 1 所述的方法，其特征在于，将所述方法应用于多维阵列，其中，沿阵列的每个维数应用步骤(a)分块、(b)折叠和(c)变换作为操作的级

联。

3. 如权利要求 2 所述的方法，其特征在于，压缩所述多维阵列的子集。

4. 如权利要求 1 所述的方法，其特征在于，进一步包含解压缩所述压缩阵列的步骤，它包括与步骤(a)~(e)顺序相反的步骤。

5 5. 如权利要求 4 所述的方法，其特征在于，将所述方法应用于多维阵列，其中沿阵列的每个维数以相反的顺序应用步骤(a)分块、(b)折叠和(c)变换作为操作级联。

6. 如权利要求 5 所述的方法，其特征在于，解压缩所述多维阵列的子集。

7. 如权利要求 6 所述的方法，其特征在于，所述多维阵列经解压缩，并
10 绘制成图以代表地球壳层的一部分。

数据压缩方法

5 发明领域

本发明涉及图像压缩方法，特别涉及用于地震数据图像压缩的改进 JPEG 方法。

背景技术

10 图像压缩的 JPEG 标准算法 (Pennebaker 和 Mittchell, 1993) 由下列 3 个针对二维阵列的每个 8×8 像素块的步骤组成：

1. 利用离散余弦变换对 8×8 像素块进行变换。
2. 将变换系数量化 (标度和舍入) 为小整数。
3. 对比特进行编码，从而利用几个比特表示出现频率最高的整数。

15 解压缩算法逆向完成这些步骤。两种算法可以容易地扩展到压缩与解压缩任意维数的阵列。

图 1 示出了未经压缩的地震数据的二维阵列。图 1 的 32 位浮点二维阵列是从三维地震勘察数据中抽取的等时层面。图 2 示出了同一阵列的放大子集。

图 3 示出了利用 JPEG 型算法压缩与解压缩整个二维阵列之后的同一放大
20 子集。整个阵列的压缩比为 103:1，这意味着 32 位浮点的原始阵列包含了 103 倍于压缩阵列的比特。

对于如此大的压缩比，这种 JPEG 型算法产生了图 3 所示的可见分块效应。在较低压缩比之下，块与块之间的这些不连续性不易发觉，但是仍然不能忽视，特别是在解压缩之后作进一步的处理或判读工作时。

25 由于在独立地压缩与解压缩每个 8×8 样本块时没有考虑保持块之间连续性，所以形成了图 3 内的分块效应。

尽管会产生分块效应，但是将如此小的数据子集独立地压缩与解压缩仍然是需要的。特别是这可以访问大压缩阵列的一小部分而无需解压缩整个阵列。这还使得压缩算法适于处理数据幅值和频谱的空间变化。在基于子波变换的压缩方法 (例如 Bradley 等人, 1993; Wickerhauser, 1994) 中没有的这样的特点。
30 本申请着手解决的问题是既具备这些特点，又没有分块效应。

解决此问题的一种方案是利用交叠的块进行数据压缩，因此可以不借助邻近块边界的数值就可以计算出解压缩的样本值。Yeo 和 Liu(1995)为使 JPEG 算法适于三维医学图像的空间表示，采用了该方案。遗憾的是，利用交叠块增加了压缩块数，这增加了计算时间并降低了压缩比。

5

说明书内引用的现有技术材料包括：

Bradley, J. N. , C. M. , 和 Hopper, T. , 1993, 用于灰度指纹图像压缩的 FBI 子波/标量量化标准: Visual Information Processing II, SPIE Proceedings, 293-304. (<ftp://ftp.c3.lanl.gov/pub/WSQ.>)

10

Jawerth, B. , 和 Sweldens, W. , 1995, 双正交平滑局部三角基: J. Fourier Anal. Appl. , 2(<http://cm.belllabs.com/who/wim/papers/-papers.html>)

Jawerth, B. , Liu, Y. 和 Sweldens, W. , 1996, 借助平滑局部三角基的信号压缩: (<http://cm.belllabs.com/who/wim/papers/-papers.html>)

15

Malvar, H. S. , 和 Staelin, D. H. , 1989, 无分块效应的 LOT 变换编码: IEEE Transactions on Acoustic, Speech, and Signal Processing, 37, no.4, 553-559

20

Malvar, H. S. , 1990, 高效变换/子带编码的重叠变换: IEEE Transactions on Acoustic, Speech, and Signal Processing, 38, no.6, 969-978

Pennebaker, W. B. , 和 Mitchell, J. L. , 1993, JPEG 静止图像数据压缩标准: Van Nostrand Reinhold.

25

Prince, J. P. , 和 Bradley, A. B. , 1956, 基于抵消时域混叠的分析/合成滤波器组设计: IEEE Transactions on Acoustic, Speech, and Signal Processing, 34, no. 5, 1153-1161

30

Wickerhauser, M. V. , 1994, 自适应子波分析，从理论到软件: A. K. Peter

Yeo, B., 和 Liu, B., 1995, 基于 DCT 压缩三维标量数据的空间表示: IEEE Transactions on Visualization and Computer Graphics, 1, no. 1, 29-43

5 发明内容

本发明的主要目标是提供一种用于数据压缩的改进 JPEG 算法。

本发明的一个重要目标是提供一种改进的 JPEG 数据压缩算法, 它可以用于压缩小子集数据但不会产生分块效应。

本发明的数据压缩方法利用了针对离散余弦变换和量化变换系数的霍夫曼编码的 JPEG 算法。

本发明提供了一种进行地震勘探的数据压缩方法, 它包含下述步骤: 收集由 N 个地震信号组成的一维阵列 X; 使所述地震信号阵列数字化, 生成所述一维阵列的 N 个取样, 并通过下述方法, 压缩所述阵列:

(a) 将所述阵列 x 分割为具有 M 个样本的块, 这里 $M < N$;

15 (b) 根据下列方程折叠跨越每个块边界 1 的样本 $x_1(j) = x(/M+j)$

$$y_1(j) = f(j)x_1(j) + f(-j)x_1(-j),$$

$$y_1(-j) = f(j)x_1(-j) - f(-j)x_1(j),$$

$$I = 1, 2, \dots, N/M-1,$$

$$j = 1, 2, \dots, M/2-1,$$

20 $y_1(j) = x_1(j),$ 否则

这里

$$f(j) = \sin\left[\frac{\pi}{4}\left(1 + \frac{2j}{M}\right)\right],$$

(c) 根据下列方程式将阵列 y 每个块内的折叠样本进行变换

$$z(k) = \sum_{j=0}^{M-1} y(j)b(j)\sqrt{\frac{2}{M}}\cos\left[\frac{\pi(2k+1)j}{2M}\right],$$

25 $k = 0, 1, \dots, M-1$

这里

$$b(j) = \begin{cases} \frac{1}{\sqrt{2}}, & j = 0 \\ 1, & \text{否则} \end{cases}$$

(d) 对阵列 z 的每个块内的已变换样本进行量化以生成整数; 以及

(e)将所述整数编码为代表压缩阵列的位流。

附图的简要说明

图 1 示出了从三维地震勘察数据中抽取的等时层面，这里的图像是未经压
5 缩的 32 位浮点二维阵列；

图 2 为图 1 二维阵列的放大子集；

图 3 示出了直接利用 JPEG 算法压缩与解压缩后的图 2 数据，图中出现分
块效应，即单独压缩的 8×8 样本块之间的不连续性；

图 4 示出了利用抑制分块效应的 JPEG 型算法压缩与解压缩后的图 2 数据；

10 图 5 示出了与长度为 32 的逆 DCT-III 对应的矩阵 C_{III}^{-1} ，这里的黑色像素
对应正值，白色像素对应负值；

图 6 示出了与长度为 8 的逆分块 DCT-III 对应的矩阵；

图 7 示出了图 6 逆分块 DCT-III 矩阵的列 11 和 19，在这里诸如样本 15 与
16 之间的不连续性在 JPEG 压缩时产生分块效应，并且由于下面方程(1c)所定
15 义的函数 $b(j)$ 内的不连续性，每条余弦内的第一样本都偏离曲线；

图 8 示出了通过展开图 7 所示截断余弦曲线获得的平滑渐变余弦曲线，它
们是图 9 逆折叠 DCT-III 矩阵中的列 11 和 9；

图 9 示出了对应逆折叠 DCT-III 的矩阵，这里矩阵的列在块与块之间交
叠，其数值平滑地减少为零，因此这些列的加权和消除了分块效应；

20 图 10 示出了展开逆分块 DCT-III 操作的矩阵，在这里图 9 的矩阵等于此
处矩阵与图 6 所示矩阵之积；

图 11 示出了展开操作，它对跨越 DCT 块边界的样本值作配对组合与替换，
折叠是该操作的逆操作；

图 12 示出了解压缩块 A 内高亮度显示样本的步骤，首先需要对块 A、B、C
25 和 D 进行解码、去量化和逆 DCT-III 处理，随后将 4 个高亮度显示样本展开，
在这里折叠与展开操作是将跨越 DCT 块边界的样本作配对混合。

实施发明的较佳方式

本发明是一种改进的数据压缩方法，它在用 JPEG 型算法压缩期间将相邻
30 块的数据复合。许多作者以各种形式描述了 JPEG 型数据压缩方法(例如
Princen 和 Bradley, 1986；Malvar 和 Staelin, 1989；Malvar, 1990；

Wickerhauser, 1994; Jawerth 等人, 1995)。JPEG 型数据压缩的优点是可提供压缩比, 但是仅仅略微增加计算时间。

图 4 示出了压缩与解压缩之后图 1 二维阵列的同一子集, 利用的是基于下述本发明方法的第二 JPEG 型算法。整个阵列的压缩比为 112:1。解压缩数据中
5 没有分块效应。

现有 JPEG 型算法与本发明新提出的 JPEG 型算法之间的差别在于对 JPEG 标准定义的分块离散余弦变换作了改进。本发明算法中的改进减轻了分块效应, 与此同时又保持了现有 JPEG 型算法的所需特征。

现有 JPEG 压缩中所用的变换是称为 DCT-II 的离散余弦变换。按照本发
10 明, 采用了称为 DCT-III 的不同的离散余弦变换。(要了解离散余弦变换的种类, 可参见 Wickerhauser, 1994, P84)。

DCT-III

正向 DCT-III 定义为

15

$$z(k) = \sum_{j=0}^{M-1} y(j)b(j)\sqrt{\frac{2}{M}}\cos\left[\frac{\pi(2k+1)j}{2M}\right],$$

$$k=0, 1, \dots, M-1 \quad (1a)$$

而相应的逆变换为

20

$$y(j) = \sum_{k=0}^{M-1} z(k)b(j)\sqrt{\frac{2}{M}}\cos\left[\frac{\pi(2k+1)j}{2M}\right],$$

$$j=0, 1, \dots, M-1 \quad (1b)$$

这里

$$b(j) = \begin{cases} \frac{1}{\sqrt{2}}, & j=0 \\ 1, & \text{否则} \end{cases} \quad (1c)$$

25

正变换与逆变换可以用矩阵乘积表示, 如 $z=C_{III}y$ 和 $y=C_{III}^{-1}z$, 这里矩阵 C_{III} 内的单元

$$C_{III}(k, j) = b(j)\sqrt{\frac{2}{M}}\cos\left[\frac{\pi(2k+1)j}{2M}\right] \quad (2)$$

图 5 示出了逆变换矩阵 C_{III}^{-1} , $M=32$ 。任意 32 个实数矢量都可以表示为该矩阵列的加权和。

在一维压缩算法中, 样本值矢量可以只用矩阵 C_{III}^{-1} 的几列逼近。每列的权重由矢量 z 的变换系数给定。高压缩比要求 z 内许多系数接近零。这种小系数将被量化为零, 从而可以利用少量位高效编码。

JPEG 标准描述了二维图像而不是一维矢量的压缩。JPEG 压缩中所用现有的 DCT-II 为二维变换。但是由于多维数据的离散余弦变换相当于沿每个数据维一维变换的级联, 所以为简化起见只描述一维变换。

10 分块 DCT-III

对于诸如断面序列的长数据矢量, 对整个矢量进行一个离散余弦变换不可能生成许多微小的变换系数。因此与 JPEG 一样采用分块 DCT, 变换长度 $M=8$, 并且填充数据矢量设定为零, 或根据需要设定为 8 的整数倍的长度 N 。图 6 示出了对应逆分块 DCT-III 的矩阵, $N=32$ 并且 $M=8$ 。除了更适合压缩之外, 分块变换的效率也更高, 而计算代价仅随数据矢量长度 N 线性增加。

与 JPEG 压缩中所用的现有 DCT-II 一样, DCT-III 具有几个有用的性质。首先, DCT-III 将实数变换为实数, 因此无需复数运算。而且与现有的 DCT-II 一样, DCT-III 为酉变换: $C_{III}^{-1}=C_{III}^T$ 。[由上述方程(1)可预见该性质]。对应图 6 的正分块 DCT-III 矩阵仅仅是逆分块 DCT-III 的转置。

20 另一个有用的性质是逆 DCT-III 等价于正 DCT-II: $C_{III}^{-1}=C_{II}$ 。DCT-III 与 DCT-II 之间的这种关系和选择变换长度 $M=8$ 可以通过简单交换正算法与逆算法便能使用 JPEG 压缩中高度优化的 DCT-II 算法。

上述所有这些信息都是有用的, 但是为什么不简单采用现有 JPEG 的 DCT-II 呢? 答案是按照本发明 DCT-III 的改进避免了 JPEG 压缩引起的分块效应。

25 只有当在矢量逼近中采用图 6 所示矩阵列的子集时才出现分块效应。例如假设仅仅利用两列高度压缩这样的矢量, 其指数 $k=11$ 和 $k=19$ 。如图 7 所示, 这两列的任意非平凡组合都在逼近中样本指数 $j=15$ 与 $j=16$ 之间产生不连续性。这种不连续性在利用 JPEG 算法压缩的图像中形成可以觉察到的分块效应。

30 折叠 DCT-III

为了避免因在压缩中采用分块 DCT-III 引起的分块效应, 采用图 8 所示平

滑渐变余弦曲线代替图 7 所示分块余弦曲线。将图 9 所示的完全逆变换矩阵与图 6 所示逆分块 DCT-III 矩阵进行比较。每条余弦曲线内的样本数增加至 16 (尾部的样本不算)，因此相邻块内的余弦曲线现在交叠在一起，并且每条余弦曲线平滑地趋于零。这些余弦曲线的加权和不会产生图 7 所示的不连续性。

5

对应这些渐变余弦的变换经常被称为局部余弦变换或者交叠正交变换 (Wickerhauser, 1994, P. 370; 图 8 与 Wickerhauser 的图 11.5 有一定的相似之处)。这些变换被称为折叠余弦变换，其名称反映了它们的计算方法。具体而言，按照本发明的压缩算法中所用的变换是折叠 DCT-III。对应图 9 所示矩阵的逆变换称为逆折叠 DCT-III。

折叠是一种利用高度优化的分块 (M=8) DCT 算法实现平滑、渐变的 16 样本余弦的方法。Wickerhauser (1994, p. 103) 将该方法描述为“由几个人独立作出的惊人观察”，并且讨论了在压缩中的应用。本发明压缩中采用的折叠操作虽然是 Wickerhauser 描述的多种方式中的一种，但是也受到了 Jawerth 和 Swelsdens (1995) 和 Jawerth 等人 (1996) 著述的启发。后者讨论了特别与压缩相关的折叠操作。

折叠的巧妙和效率在于图 9 所示逆折叠 DCT-III 矩阵是图 10 所示展开矩阵与图 6 所示逆分块 DCT-III 矩阵之积。由于展开矩阵的每行与每列包含了不超过 2 个的非零元素，所以除了逆分块 DCT-III 的计算代价以外，展开的计算代价微不足道。

特别是图 6、9 或 10 所示的矩阵实际上从没构造出来。操作是在样本值上进行的，其效果与这些矩阵的乘积是一样的。对于图 6 所示的逆分块 DCT-III 矩阵，该操作是 JPEG 压缩中所用的高效正分块 DCT-II 算法。对于图 10 所示的展开矩阵，如图 11 所示，该操作是跨越指数 $j=8, 16\cdots$ 的块边界样本值的配对混合。

折叠操作是简单的逆展开操作，它将同一样本值作不同的配对复合。由于折叠与展开边界的中心在块边界周围，所以这些操作可以用偏移样本值最简洁地定义为 $y_I(j)=y(IM+j)$ 。符号 I 是块指数而 j 是块内样本指数。

随后借助下列方程完成折叠

$$\begin{aligned} y_I(j) &= f(j) x_I(j) + f(-j) x_I(-j), \\ y_I(-j) &= f(j) x_I(-j) - f(-j) x_I(j), \end{aligned}$$

$$I=1, 2, \dots, N/M-1,$$

$$j=1, 2, \dots, M/2-1,$$

$$y_I(j)=x_I(j), \quad \text{否则} \quad (3a)$$

并借助下列方程完成展开

$$5 \quad x_I(j)=f(j)y_I(j)-f(-j)y_I(-j),$$

$$x_I(-j)=f(j)y_I(-j)-f(-j)y_I(j),$$

$$I=1, 2, \dots, N/M-1,$$

$$j=1, 2, \dots, M/2-1,$$

$$x_I(j)=y_I(j), \quad \text{否则} \quad (3b)$$

10 这里的折叠函数 $f(j)$ 被定义为

$$f(j)=\sin\left[\frac{\pi}{4}\left(1+\frac{2j}{M}\right)\right], \quad (3c)$$

15 将方程 (3b) 应用于图 6 所示的矩阵列计算得到了图 8 的平滑渐变余弦曲线与图 9 的这种余弦曲线的矩阵。

首先借助方程 (3a) 计算 $y(j)$ ，然后借助方程 (1a) 计算 8 个样本中每个块的 $z(k)$ 就确定了样本值 $x(j)$ 的正折叠 DCT-III。同样，首先借助方程 (1b) 计算每个块的 $y(j)$ ，然后借助方程 (3b) 计算 $x(j)$ 就确定了逆折叠 DCT-III。

20 如 Wickerhauser(1994, p. 105)、Jawerth 和 Sweldens(1995) 以及 Jawerth 等人(1996)所述，可以采用虽然不同但是原理相似的折叠操作。选择方程 (3) 定义的折叠操作基于两个理由。

25 首先折叠操作是酉操作。图 10 所示展开矩阵是对应折叠矩阵(未画出)的转置。为了确证这种性质，请注意方程 (3c) 的折叠函数满足 $f^2(j)+f^2(-j)=1$ ，并且将方程 (3a) 和 (3b) 的折叠与展开操作表示为 2×2 矩阵的乘积。解析地求出 2×2 折叠矩阵的逆矩阵以考察其是否等于转置矩阵，而转置矩阵等于 2×2 的展开矩阵。由于矩阵由跨越边界的 2×2 样本值混合组成，因此整个折叠操作是酉操作。

30 其次，按照本发明的折叠操作确保了常数函数(例如 $x(j)=1$)在每个块内产生仅对于 $k=0$ 是非零的变换系数 $z_1(k)$ ，因此提高了恒定的(或者变化缓慢的)数据的压缩比。Jawerth 等人(1996)称正折叠 DCT-III 实现了“恒定的分辨

率”。

为了确证折叠 DCT-III 的第二种性质，解析地将方程 (3a) 的折叠操作应用于恒定样本值 $x_1(j)=1$ ，并确证结果是 $y_1(j)=\sqrt{M} C_{III}(k, j)$ ，这里 $C_{III}(k, j)$ 在方程 (2) 中定义。换句话说，选定的折叠函数使得恒定样本值折叠为精确地 (在标度因子 $\sqrt{M}$ 内) 与 DCT-III 的第一 ($k=0$) 余弦值匹配。由于该余弦与变换的所有其它余弦正交，所以在正折叠 DCT-III 之后只有每个块内 $k=0$ 的变换系数非零。每个非零系数的值为 $z_1(0)=\sqrt{M}$ 。

更为确切地说，除了第一和最后的块，其他块都保持这种性质。虽然可以通过将折叠与展开矩阵修改为矩阵角部不为单位值从而使得第一和最后块的分辨率恒定，但是这些块内的压缩比小一些是能够接受的，并且还保持了严格的酉折叠与展开操作。

与用于 JPEG 压缩的 DCT-II 之比较

由于 DCT-II 的第一余弦 (例如如图 5 内矩阵的第一行) 为常数，所以 JPEG 压缩所用的正 DCT-II 无需折叠就实现恒定的分辨率。实际上，这就是 JPEG 压缩标准指定 DCT-II 而非 DCT-III 的理由。由于无需修改标准的 JPEG 算法就可以在 JPEG 压缩之前进行折叠并在 JPEG 解压缩之后进行展开，所以可以考虑对 DCT-II 利用折叠技巧。

遗憾的是，方程 (3) 的折叠与展开操作对 DCT-II 并不适合。具体而言，方程 (3b) 不会产生如图 8 所示的平滑渐变余弦曲线。

Jawerth 等人 (1995) 描述了适合于 DCT-II 的不同折叠与展开操作，这些操作保持了恒定的分辨率。但是这些操作不是酉操作。实际上它们是病态条件的，将会放大块边界附近样本值的不连续性，从而减小压缩效率。

相反，与方程 (1) 的正和逆 DCT-III 一样，方程 (3) 的折叠与展开操作是酉操作。如果 F 表示折叠矩阵，则正折叠 DCT-III 可以表示为 $z=C_{III}Fx$ ，并且注意到

$$\begin{aligned} z^T z &= x^T F^T C_{III}^T C_{III} F x \\ &= x^T F^{-1} C_{III}^{-1} C_{III} F x \\ &= x^T x \end{aligned}$$

$$\sum_{k=0}^{N-1} z^2(k) = \sum_{j=0}^{N-1} x^2(j) \quad (4)$$

换句话说，折叠 DCT-III 变换之后的样本值平方和等于折叠 DCT-III 前的和。该信息可以用来估计量化变换系数过程中引入的解压缩数据的失真。

5

量化和编码

如上所述，本发明压缩算法中所用的变换仅仅是 JPEG 压缩中所用变换的逆变换，但是引入的折叠抑制了分块效应。所用的量化和编码方法也由 JPEG 压缩中所用方法修改而来。以下描述 JPEG 方法与本发明之间的差异。

10

量化

由于 JPEG 压缩是针对图像的，所以压缩之前的数据不是 8 位就是 12 位（彩色图像用红绿蓝三重色样本表示）。在借助 DCT-II 进行二维变换之后，由于二维 DCT-II 之后最大的样本值比变换前最大 8 倍，所以每个样本的 8 位数据可能
15 可能需要 11 位。通常情况下，DCT-II 之后的最大样本值 最多是变换前的 $M^{D/2}$ 倍，这里 D 为变换的维数。换句话说，

$$|z|_{\max} \leq M^{D/2} |x|_{\max} \quad (5)$$

20 当数据与 DCT-II 中所用其中一个余弦（例如恒定的第一余弦 $C_{11}(k=0, j)$ ）匹配时达到上限。

同样的因子 $M^{D/2}$ 也可应用于本发明压缩算法所用的折叠 DCT-III 上。因此借助折叠 DCT-III 变换 8 位整数数据之后，本发明的方法对二维压缩中的 11 位整数和三维压缩中的 13 位整数进行量化和编码。

25 JPEG 压缩算法与本发明方法中的量化步骤是一种减少编码位数的标度操作，而且正是这种位数的减少导致压缩和随后解压缩数据的失真。在这种有损压缩方法中，相对高压缩比而言这样的失真是可以接受的。

在 JPEG 压缩中，每块内的变换系数的量化是不同的，表示大波数（空间频率）所用的位数比小波数的少。JPEG 压缩中与波数相关的标度通常针对人类视觉
30 觉而优化。

在本发明的压缩方法中，波数的量化是相等的。因此不会引入与波数相关的量化才有的误差。等量化的理由是对于地震数据，经常对大波数感兴趣。例如地震数据的子表面断层图像对应于大波数。另一个理由是地震数据通常由计算机算法分析，它们与人的视觉系统无关。

5

局部量化

当压缩 8 位图像数据时，样本值介于-128 与+127 之间；并且可以假定小幅值的数据块是无意义的并在压缩期间将它们量化为零而不会有问題。特别是 JPEG 标准假定对于整幅图像只允许有一组量化标度因子。如上所述，这些标度因子相对块内不同的系数(不同的波数)变化，但是对于每个块都采用同样的一组标度因子。就这个意义而言，JPEG 量化是全局性的。

当压缩 32 位浮点数据时，比较好的是进行局部量化，其标度因子随着块的不同而不同。在压缩这类数据之前，最大样本幅值 $|x|_{\max}$ 可能是未知的，并且在压缩前读取每个样本以确定该值的代价是高昂的。另外小幅值可能不能假定为无意义的；特别是地震数据经常在确认假定有效之前需要作可观的处理。因此虽然对于块内所有变化系数的量化采用的是一个标度因子，但是块与块之间的标度因子是变化的。具体而言，在上述方程(5)中， $|z|_{\max}$ 表示每个变换块内的最大系数，并且对于每块计算不同的标度因子。

局部量化产生比全局量化更低的压缩比(每个样本的位数更多)。显而易见的原因是需额外的位来存储每个被压缩块的量化标度因子。比较间接的原因是局部量化可以量化为零的样本比全局量化更少。因此在本发明的压缩算法中既允许选择局部量化也允许选择全局量化。

局部量化无法处理单个块内动态范围较大的情况。例如，未处理地震数据中的低幅值反射可能隐藏在高幅值表面波的下面。在混杂高幅值噪声的块内，即使是局部量化，本发明的压缩算法也可能将低幅值信号量化为零，从而无法在解压缩之后恢复这样的信号。因此在压缩之前应该衰减高幅值噪声。

压缩的虚拟存储器

虚拟存储器给人以存储器空间大于物理可用空间的假象。这种假象特别适合于这样的应用，它们访问的数据常常在最近被访问过的其他数据附近。这类应用具有良好的引用局域性。

借助二维或三维阵列进行处理的应用常常具有良好的引用局域性。例如地震数据解释应用显示了来自三维阵列的连续二维层面数据。借助足够快的局部解压缩算法可以解压缩包含所需分层样本的块而无需解压缩整个三维阵列。本发明的压缩算法特别适用于这类应用。

- 5 这类应用的关键是能够压缩或解压缩大阵列的一个子集而无需压缩或解压缩整个阵列。对于基于分块 DCT-II 的压缩而言，就象 JPEG 压缩中所用的方法一样，很容易达到这种要求。具体而言，为了解压缩一个样本，只需解码、去量化和逆 DCT-II 包含该样本的块。一旦完成一个样本的解压缩，则就完成了块内所有样本的解压缩。假定引用局域性，则节省了解压缩该块其他样本的计算代价。

- 10 对于本发明基于折叠 DCT-III 的压缩算法，需要一些额外的处理。考虑二维压缩和图 12 所示的四块 8×8 样本。为了解压缩对应块 A 内实心圆圈的样本，需要(1)解码、去量化和逆 DCT-III 所有四块(A、B、C 和 D)，以及(2)展开对应实心圆圈的四个样本。跨越块边界的展开如图 11 所示并且由方程(3b)描述，
- 15 它首先对一个维数上的第一个块展开，然后再展开另一个块。虽然需要四个块，但是一旦完成块 A 内样本的解压缩，则解压缩邻近样本所需的大部分工作即告完成。而且如果假定引用局域性，则无需额外的计算。

- 本发明的量化步骤与 JPEG 另一个差异源于本发明需要量化 32 位浮点数据，这些数据的动态范围大于 8 或 12 位的图像数据。为了将浮点值 z 量化为
- 20 具有 $B+1$ 位(包括符号位)的整数 i ，采用了以下算法：

$$i = \begin{cases} zxs + 1/2, & z \geq 0, \\ zxs - 1/2, & z < 0, \end{cases} \quad (6)$$

- 这里 s 为量化标度因子。为了避免溢出，要求 $|i| < 2^B$ 。该限制与方程(6)
- 25 导出下列标度因子方程：

$$s = \frac{(2^B - 1/2)(1 - \epsilon)}{|z|_{\max}} \quad (7)$$

- 这里 ϵ 为浮点小正数，即利用浮点运算从 1 减去的最小正数，用于获得不
- 30 等于 1 的数。

霍夫曼编码

JPEG 压缩标准可以有两种编码量化生成整数的方法,即霍夫曼编码和算术编码,其他许多编码方法也是可行的。由于 JPEG 的霍夫曼编码计算速度快,
5 所以被用于本发明的压缩算法中,并且这种编码实现简单,可以免费使用。

但是霍夫曼编码和解码算法遵循的是 JPEG 规范。JPEG 标准规定了 DC(k=0)变换系数的特殊编码。JPEG 压缩充分利用了相邻块之间的这些 DC 系数常常高度相关的特点。基于两个理由,本发明的方法未采用这种特殊处理:(1)特殊处理引入了块与块的依赖性,这使单个块的压缩与解压缩复杂化;以及(2)地震数据的 DC 系数一般比较小(虽然小,但是由于采用了短 M=8 变换,所以很少
10 能忽略)。因此 DC 系数就象其他(AC)系数那样编码。

局部压缩

JPEG 型算法的主要优点是无需处理全部多维阵列即可压缩或使用其一部分。相反,基于小波变换的压缩算法缺乏这个特点(例如参见 Bradley 等人,1993;Wickerhauser,1994)。虽然 JPEG 标准并不直截了当地支持这种特征,但是 JPEG 压缩中采用的分块 DCT-II 使得这种特征的实现更为容易。该能力在本发明的基于折叠 DCT-III 的压缩算法中得到了充分发挥。

阵列中的每个数据块可以被类推为虚拟存储器的页面。对于二维压缩,每个页面包含 $64=8\times 8$ 个样本;对于三维压缩,每个页面包含 $512=8\times 8\times 8$ 个样本。当样本页面被调用时进行解压缩,并在调出页面时压缩波数。解压缩页面的工作组保存在存储器内,而大多数页面仍然处于压缩状态并且存储在存储器或磁盘内(如果存储在磁盘上,则可以对页面进行组合以提高 I/O 效率)。如果工作组的长度足够,并且应用具有良好的引用局域性,则可以将压缩和/或
25 解压缩每个页面的计算代价分摊到这些页面内大多数样本的访问上。

总结

在根据本发明对 JPEG 压缩算法的适用性修改中可以重复使用算法中的许多部分。通过简单地将正和逆变换互换就可以对长度为 8 的 DCT 重复使用 JPEG
30 方法。JPEG 量化方法经过修改避免了对小波数的优先处理,并且处理了数据幅值在块与块之间的变化。在正 DCT 之前也可以跨越块边界折叠样本值,并且在

逆 DCT 之后展开这样的值。这种折叠与展开抑制了 JPEG 算法压缩图像内可觉察的分块效应。

- 5 为了比较本发明方法与其他算法的性能，采用特定压缩比下计算时间和失真作为性能比较的两个指标。作为初步基准程序，我们倾向于采用前述本发明算法的实现方案。在较宽的压缩比范围内，本发明方法的计算时间大约是小波算法的一半而失真基本相同(Diller 1997, 私人通信)。本发明方法的计算时间比小波方法短的结果是可信的(因为本发明采用了较少的乘法和加法运算，并且更多地以局部方式使用存储器)。

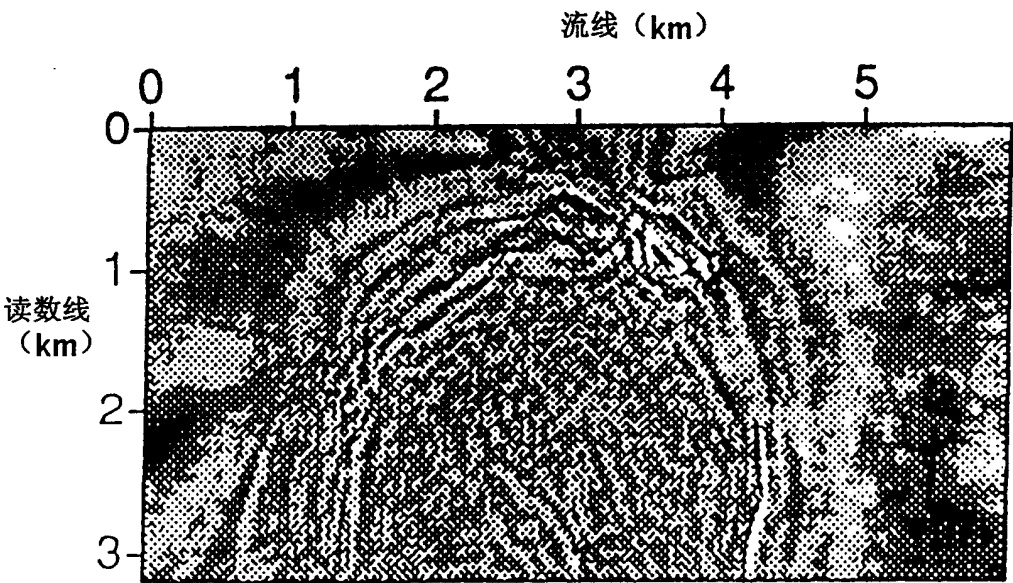


图 1

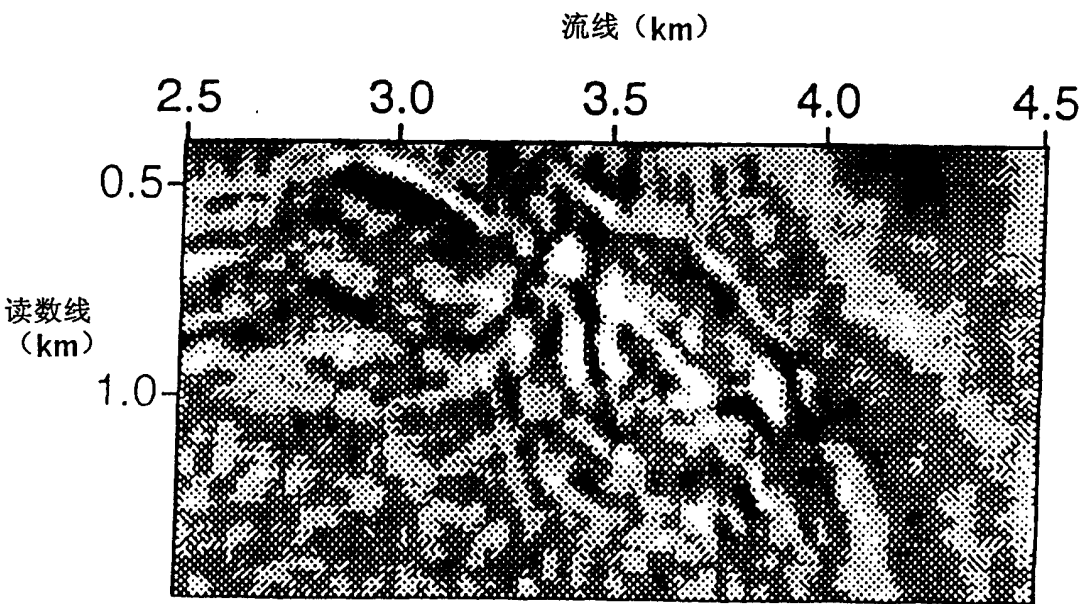


图 2

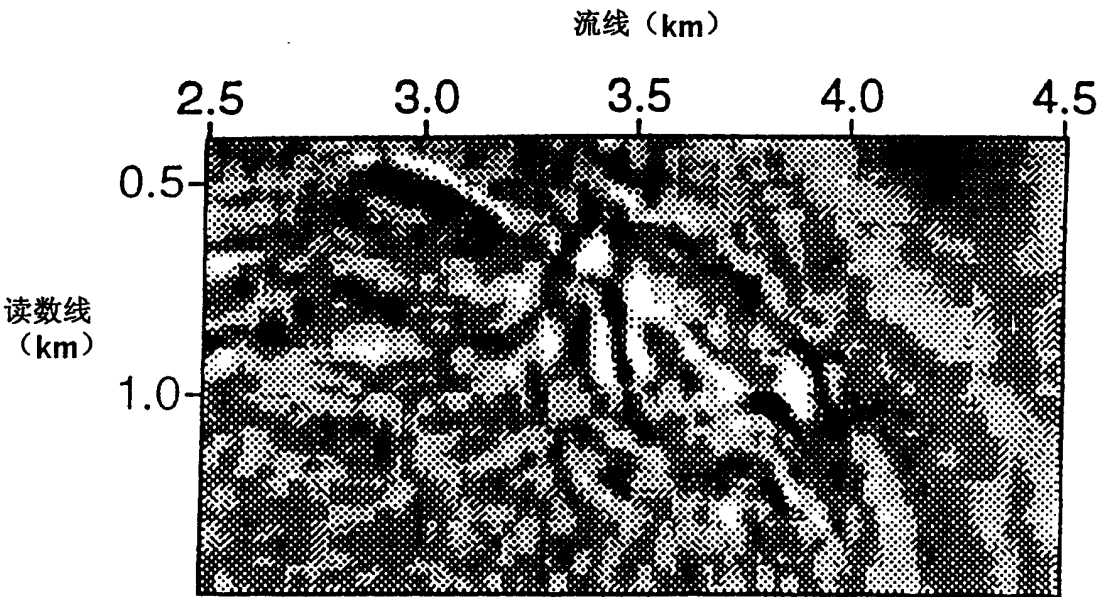


图 3

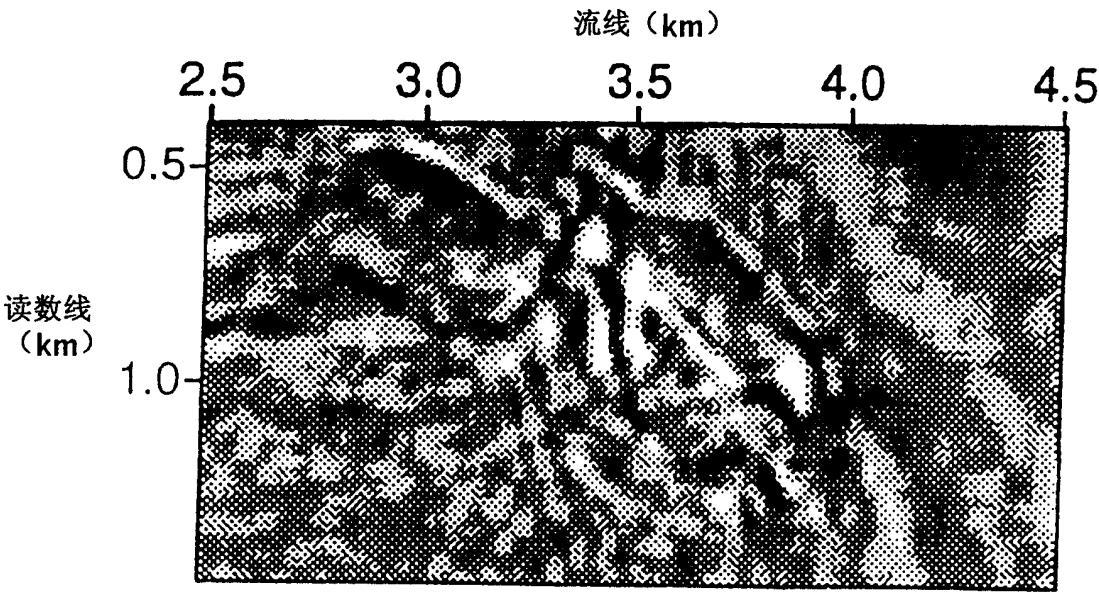


图 4

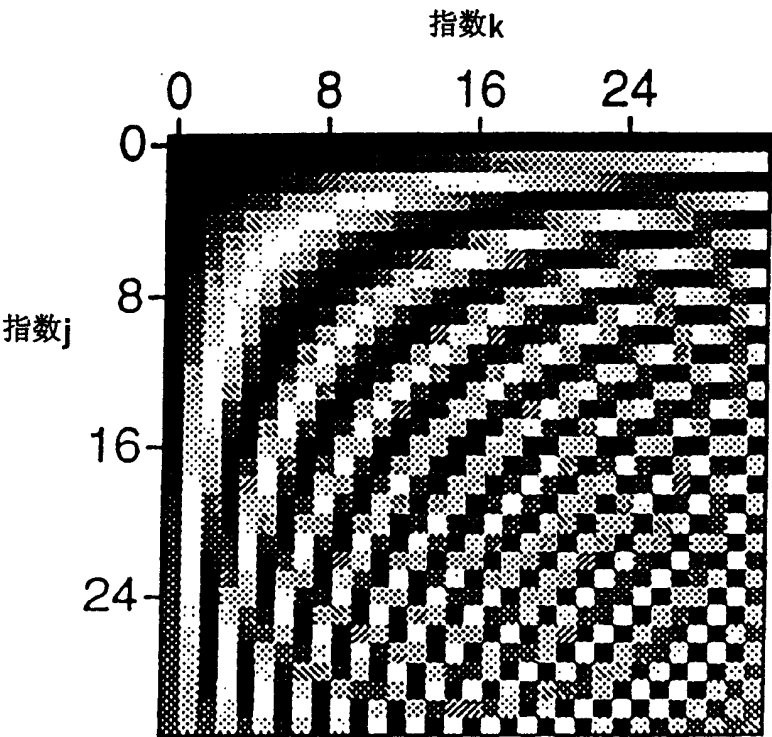


图 5

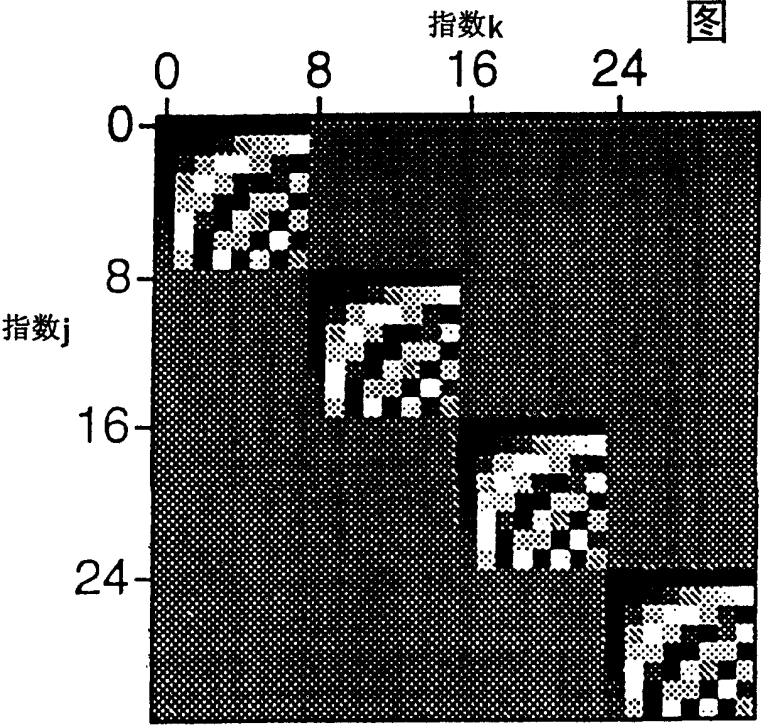


图 6

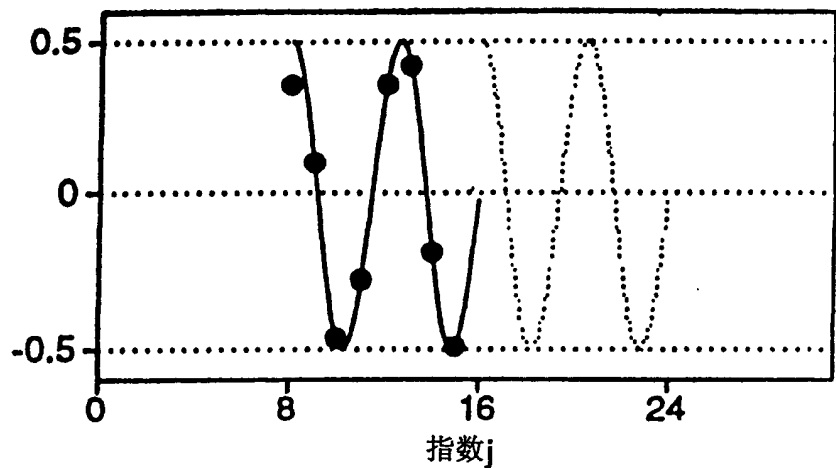


图 7

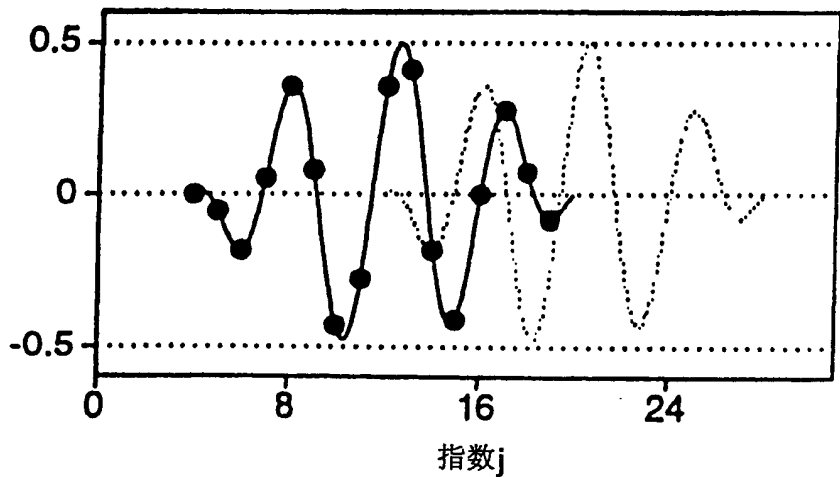


图 8

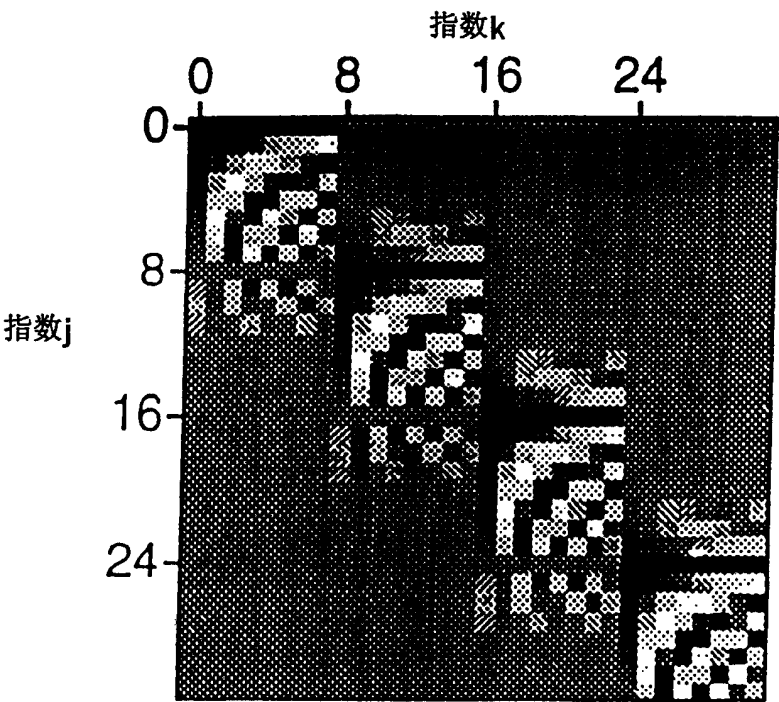


图 9

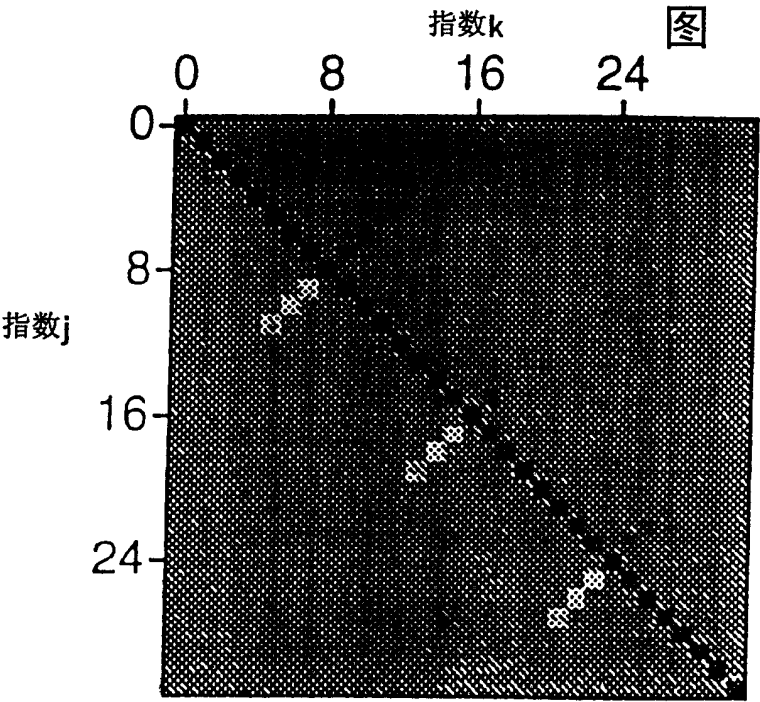


图 10

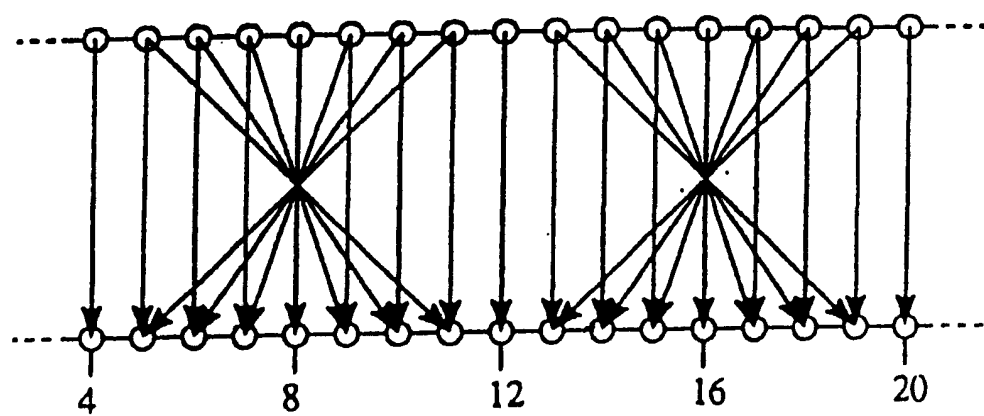


图 11

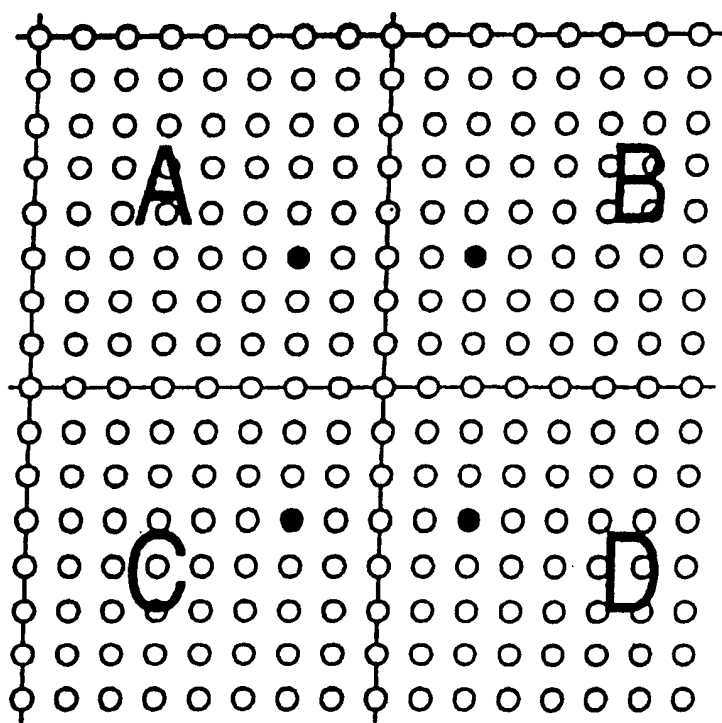


图 12