

快速最近邻矢量量化码字搜索算法

孙圣和, 陆哲明, 刘春和

(哈尔滨工业大学自动化测试与控制系, 黑龙江哈尔滨 150001)

摘要: 本文综述了各种快速最近邻矢量量化码字搜索算法, 按照算法的特点对各种快速算法进行了归类, 对各种算法的编码时间、平均失真计算次数、额外存储量和离线计算量进行了仿真、比较和分析, 并提出作者的一些改进算法或改进思路。

关键词: 矢量量化; 码字搜索

中图分类号: TN919.31 **文献标识码:** A **文章编号:** 0372-2112 (2001) 12A-1772-06

Fast Nearest-Neighbor Codeword Search Algorithms for Vector Quantization

SUN Sheng-he, LU Zhe-ming, LIU Chun-he

(Dept of Automatic Test and Control, Harbin Institute of Technology, Harbin, Heilongjiang 150001, China)

Abstract: This paper summarizes various fast codeword search algorithms for vector quantization. These algorithms are classified into several classes according to their characteristics. Simulation, comparisons and analysis are done for various fast algorithms in terms of encoding time, average calculations, extra memory and off-line computation. Some modified algorithms or trains of thoughts are also presented in this paper.

Key words: vector quantization; codeword search

1 引言

矢量量化技术 (VQ)^[1,2] 已经广泛地应用于语音编码和图像压缩系统中。矢量量化过程可以定义为从 k 维欧几里德空间 R^k 到其一个有限子集 C 的一个映射, 即 $Q: R^k \rightarrow C$, 其中 $C = \{y_0, y_1, \dots, y_{N-1} | y_i \in R^k\}$ 称为码书, N 为码书长度。该映射满足: $Q(x | x \in R^k) = y_p$, 其中 $x = (x_0, x_1, \dots, x_{k-1})$, $y_p = (y_{p0}, y_{p1}, \dots, y_{p(k-1)})$ 并满足

$$d(x, y_p) = \min_{0 \leq j \leq M-1} (d(x, y_j)) \quad (1)$$

其中 $d(x, y_j)$ 为矢量 x 与码字 y_j 之间的失真测度, 常用的失真测度为平方误差测度, 其表达式为

$$d(x, y_i) = \sum_{l=0}^{k-1} (x_l - y_{il})^2 \quad (2)$$

矢量量化编码器在码书中搜索出与输入矢量间失真最小的码字。传输时仅传输该码字索引。矢量量化解码器只要根据接收到的索引查表操作在码书中查找该码字, 并将它作为输入矢量的重构矢量。

编码时间是影响编码系统实时性的一个重要因素。矢量量化编码过程最终归结为在给定码书 C 中搜索出与输入矢量 x 最匹配的码字 y_j , 使得 y_j 与 x 之间的失真是所有码字中最小的。穷尽搜索 (Full Search, FS) 算法需要计算输入矢量与所有码字之间的失真并通过比较找出失真最小的码字。因此, 为了对矢量 x 进行编码, 它需要 Nk 次乘法, $N(2k-1)$ 次加法和 $N-1$ 次比较。由此可见, FS 算法的计算复杂度由码书尺

寸和矢量维数决定。在高效率 VQ 编码系统中, 减少码字搜索的计算负担是非常必要的。

快速搜索算法可以提高 VQ 编码效率。一种快速有效的码字搜索算法的三个必不可少的因素是: 1. 良好的初始匹配码字; 2. 强有力的码字删除准则; 3. 合理的码字搜索顺序。然而, 一种有效的码字搜索算法往往需要附加计算量 (如计算码字均值或对码书进行排序等) 和额外存储空间 (如存储码字特征值)。附加计算量分为在线计算量 (占用编码时间) 和离线计算量 (不占用编码时间)。如何尽可能减少附加计算量和额外存储空间是许多快速算法所必须面临的问题。文献中的快速算法大致可分为基于不等式判据的、基于金字塔结构的、基于变换域的三大类。下面, 本文对这三大类最近邻码字搜索算法分别进行综述, 然后进行仿真实验和总结分析。

2 基于不等式判据的码字搜索算法

2.1 部分失真搜索算法

部分失真搜索 (PDS) 算法^[3] 是一种简单有效的最近邻搜索算法。它的基本原理如下:

定理 1 设目前最小失真为 $d_{\min} = d(x, y_p)$, 若

$$\sum_{l=0}^{s-1} (x_l - y_{il})^2 \geq d_{\min} (1 \leq s \leq k) \quad (3)$$

则

$$d(x, y_i) \geq d(x, y_p)$$

因此, 在计算码字 y_i 和输入矢量 x 的失真过程中, 若满足不等式 (3), 则 y_i 肯定不是 x 的最近码字。该算法的效率有

限,但它不额外存储空间. PDS 常用于许多快速搜索算法的最后一步,以排除其它方法已经没法排除的码字.

2.2 基于绝对误差不等式的最近邻搜索算法

定理 2 假定 $d_{\min} = d(x, y_p), 1 \leq s \leq k$, 若

$$\sum_{l=0}^{s-1} e_{il} = \sum_{l=0}^{s-1} |x_l - y_{il}| \geq \sqrt{k \cdot d_{\min}} \quad (4)$$

则 $d(x, y_i) \geq d(x, y_p)$

基于定理 2 的算法称为 AEI 算法^[4], 其初始匹配码字为具有最小的最大绝对误差分量的码字. 除了定理 2 所述的删除准则, 还有 Minimax 删除方法^[5], 如下定理所述:

定理 3 设 $d_{\min} = d(x, y_p)$, 对于任何 $0 \leq l \leq k - 1$, 若

$$e_{il} = |x_l - y_{il}| \geq \sqrt{d_{\min}} \quad (5)$$

则 $d(x, y_i) \geq d(x, y_p)$

可见, 当某一绝对误差分量比所有其它绝对误差分量都大得多时, Minimax 方法比较有效. 比较式 (5) 和 (4), 当 $s = 1$ 时, AEI 算法的删除效率就没有 Minimax 算法高, s 逐步增加时 AEI 算法的删除效果才越来越明显. 为进一步提高 AEI 的删除效果, 文献[6]提出了改进的 IAEI 删除准则:

定理 4 假定 $d_{\min} = d(x, y_p), s \leq h \leq k$, 若

$$\sum_{l=0}^{s-1} e_{il} = \sum_{l=0}^{s-1} |x_l - y_{il}| \geq \sqrt{h \cdot d_{\min}} \quad (6)$$

则 $d(x, y_i) \geq d(x, y_p)$

IAEI 算法已包含 Minimax 方法 ($s = h = 1$), 通过 s 和 h 从 1 到 k 的取值, 可排除大量码字, 所以该算法效率比 AEI 和 Minimax 算法高. 基于绝对误差不等式的各种算法的共同缺点是: (1) 选取初始匹配码字需要较多的在线绝对误差运算; (2) 没有考虑码字排序和搜索顺序. 所以这类算法的效率是有限的.

2.3 基于三角不等式的最近邻搜索算法

三角不等式删除准则是一种有效的码字删除准则, 它充分利用如下的三角不等式特性:

定理 5 令 y_i, y_p 为相异码字, $0 \leq i, p \leq N - 1$, 若

$$d(y_p, y_i) / 4 \geq d(x, y_p) \quad (7)$$

则 $d(x, y_i) \geq d(x, y_p)$

定理 6 令 y_i, y_j, y_p 为三相异码字 $0 \leq i, j, p \leq N - 1$, 若

$$|\sqrt{d(y_j, y_i)} - \sqrt{d(x, y_i)}| \geq \sqrt{d(x, y_p)} \quad (8)$$

则 $d(x, y_i) \geq d(x, y_p)$

文献[7]提出了基于定理 5 的快速最近邻搜索算法 (FNNS). 文献[8]在文献[7]的基础上利用了定理 6, 提出了近似与排除搜索算法 (AES). 它们的缺点是需要大量的离线失真计算量以及 $(N(N - 1)/2)$ 个单元来存储码字间失真, 从而限制了这类算法的应用.

为了降低额外存储量, 文献[9]的算法通过存储一套控制矢量的 2-范数来缩小存储空间和搜索空间, 从而大大降低了计算量. 此外, 作者在文[10]中提出一种改进算法 (MFNNS). 该算法不需要存储任何码字间距, 只需存储 N 个码字和值, 大大降低了 FNNS 的存储负担. 它同时利用了均值不等式和三角不等式删除准则, 如下定理所述:

定理 7 假定 $d_{\min} = d(x, y_p)$, 码字 y_i 的和值为 S_i , 若

$$|S_i - S_p| \geq 2 \sqrt{k d_{\min}} \quad (9)$$

则 $d(x, y_i) \geq d(x, y_p)$

2.4 基于均值不等式的最近邻搜索算法

k 维矢量 x 的均值定义为 $m_x = (\sum_{l=0}^{k-1} x_l) / k$, 方差定义为

$v_x = \sqrt{\sum_{l=0}^{k-1} (x_l - m_x)^2} / k$. 设 R^k 上的一条直线 L 上任何点 $p = (p_0, p_1, \dots, p_{k-1})$ 都满足 $p_0 = p_1 = \dots = p_{k-1}$, 则称 L 为 R^k 的中心线, 与 L 正交的超平面称之为等均值超平面. 点 x 在中心线上的投影点为 $L_x = (m_x, m_x, \dots, m_x)$, 而 x 到该投影点的欧氏距离刚好就是矢量 x 的方差. 容易得到如下定理:

定理 8 假定 $d_{\min} = d(x, y_p)$, m_i 为码字 y_i 的均值, 若

$$|m_i - m_x| \geq \sqrt{d_{\min} / k} \quad (10)$$

则 $d(x, y_i) \geq d(x, y_p)$

令 $m_{\max} = m_i + \sqrt{d_{\min} / k}, m_{\min} = m_i - \sqrt{d_{\min} / k}$,

则基于定理 8 的等均值最近邻搜索算法 (ENNS)^[11] 将搜索范围限制到两个平行的等均值超平面内, 这两个超平面的均值分别为 m_{\max} 和 m_{\min} , 并与中心线 L 分别相交于 $L_{\max} = (m_{\max}, m_{\max}, \dots, m_{\max})$ 点和 $L_{\min} = (m_{\min}, m_{\min}, \dots, m_{\min})$ 点.

图 1 为 2 维情况示意图, 算法的搜索区域限制在直线 L_1 和 L_2 之内. ENNS 算法在开始搜索前需要按码字均值的大小对码书进行升序排列. 算法采用上下

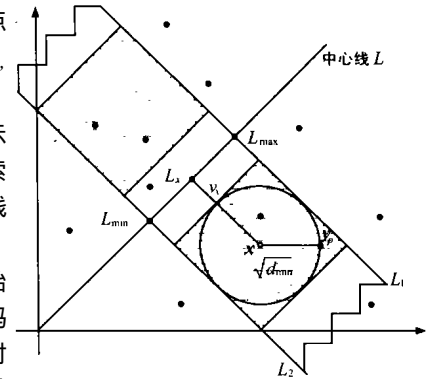


图 1 搜索算法的二维示例

搜索法, 即在码字 y_p 附近进行上下搜索, 一旦某个方向上的码字均值满足式 (10), 则该方向的搜索停止.

为了缩小 ENNS 算法的搜索范围, 文献[12]提出的改进算法 (MENNS) 把矢量分成两个子矢量, 用子矢量均值不等式删除准则和原矢量均值不等式删除准则一同进行码字删除. 它减少了码字搜索范围, 但引入一些额外子矢量均值计算量和额外存储空间. 文献[13]提出的等均值等方差最近邻搜索算法 (EENNS) 在运用均值不等式的同时采用如下方差不等式删除准则:

定理 9 假定 $d_{\min} = d(x, y_p)$, v_i 为码字 y_i 的方差, 若

$$|v_i - v_x| \geq \sqrt{d_{\min}} \quad (11)$$

则 $d(x, y_i) \geq d(x, y_p)$

注意到以中心线为轴心的超圆柱面上各点的方差相同, 所以可以称该圆柱面为等方差超圆柱面. EENNS 算法将搜索范围限制到了两个平行的等均值超平面和两个同轴等方差超圆柱面 (方差分别为 $v_{\max} = v_i + \sqrt{d_{\min}}$ 和 $v_{\min} = v_i - \sqrt{d_{\min}}$) 所围成的区域内, 如图 1 阴影所示. 虽然 EENNS 算法需要增加离线的方差计算量和在线的输入矢量方差计算, 但由于它大大降低了搜索范围, 它比 ENNS 算法有效. 此外, 文献[14]提

出的改进算法(MEENNS-1)结合了 EENNS 算法与 AEI 删除准则,刚好把图 1 中心线上方阴影部分所包含的码字排除掉,搜索范围得到进一步减少.而文献[15]提出的另一种改进算法(MEENNS-2)同样利用了均值和方差,但它把均值和方差体现在一个不等式中,称为均值-方差不等式删除准则,如下定理所述:

定理 10 设 $d_{\min} = d(x, y_p)$, m_i, v_i 为 y_i 的均值和方差,若

$$k(m_i - m_x)^2 + (v_x - v_i)^2 \geq d_{\min} \quad (12)$$

则

$$d(x, y_i) \geq d(x, y_p)$$

MEENNS-2 算法的搜索范围被限制到关于中心线对称的两个超球体内.与 EENNS 相比,该算法没有增加任何附加乘法计算量.显然,定理 8 和定理 9 只是定理 10 的推论,定理 10 能够排除更多码字.为了进一步减少 MEENNS-2 算法的搜索范围,作者在文[16]中推导出了比定理 10 更有效的不等式删除准则.在该算法中作者定义了矢量和其子矢量和和方差,结合 IAEI 算法的思想和定理 10,推导出三条均值-方差删除准则.由于比 MEENNS-2 多两条子矢量删除准则,所以它能删除更多码字.

理论分析和仿真实验均表明,基于均值不等式删除准则的各种算法是最实用的码字搜索算法.它具有良好初始匹配码字并选取过程快速,搜索范围小,搜索顺序合理,删除准则简单有效,这些优点是其它不等式删除准则无法比拟的.

2.5 基于其它不等式的最近邻搜索算法

除了上面三小节所述的三类不等式删除准则,在一些文献中提出了其它类型的不等式删除准则.文献[17]通过对平方误差的展开式进行分析,提出了双测试算法(DTA),算法基于如下定理:

定理 11 设矢量 x 和码字 y_i 的各分量均不小于零,即 $x_l \geq 0, y_{il} \geq 0, l = 0, 1, \dots, k-1$, 设 $x_{\max} = \max\{x_0, x_1, \dots, x_{k-1}\}$ 及 $y_{i\max} = \max\{y_{i0}, y_{i1}, \dots, y_{i(k-1)}\}$

$$d_1(x, y_i) = (x - y_i)^2 + (y_i - x_{\max})^2 - 2 \cdot x_{\max} \cdot y_i \quad (13)$$

$$d_2(x, y_i) = (x - y_i)^2 + (y_i - y_{i\max})^2 - 2 \cdot y_{i\max} \cdot x_i \quad (14)$$

若 $d_1(x, y_i) \geq d_{\min}$ 或 $d_2(x, y_i) \geq d_{\min}$ 则 $d(x, y_i) \geq d(x, y_p)$ (15)

为了提高双测试算法的搜索效率,文献[18]提出的算法考虑了等价的误差测度,即

$$d_1(x, y_i) = d(x, y_i) - x - \frac{y_i}{2} = y_i - \frac{x}{2} - 2 \sum_{l=0}^{k-1} x_l y_{il} \quad (16)$$

设当前的最小失真为 $d_{\min} = d_1(x, y_p)$, 若

$$y_i - (y_i - 2 \cdot x) \geq d_{\min} \quad (17)$$

则 $d_1(x, y_i) \geq d_{\min}$

此外,文献[19]提出一种用于图像矢量量化的快速码字搜索算法.该算法给出三种投影:水平投影、垂直投影和整体投影的定义,并推导出三条相应的删除准则.该算法比 EENNS 算法多用两条删除准则,所以该算法比 EENNS 算法有效.

3 基于金字塔结构的码字搜索算法

图像的金字塔数据结构早在文献[20]中提出并开始用于图像编码.假定图像 x_n 的大小为 $2^n \times 2^n$, 则 x_n 的金字塔结构可定义为一系列矩阵 $\{X_0, \dots, X_{m-1}, X_m, X_{m+1}, \dots, X_n\}$, X_{m-1} 为 X_m 分辨率降低的形式,其大小为 $2^{m-1} \times 2^{m-1}$, 其中 X_0 只有一个像素.设 $X_{m-1}(i, j)$ 为 $m-1$ 级图像 X_{m-1} 的一个像素,它可以由 m 级图像 X_m 中相邻的 2×2 个像素 $X_m(2i, 2j), X_m(2i, 2j+1), X_m(2i+1, 2j)$ 和 $X_m(2i+1, 2j+1)$ 作适当的运算得到.图像金字塔类型各种各样^[21], 其中最简单的是均值金字塔,也就是说 $X_{m-1}(i, j)$ 为 $X_m(2i, 2j), X_m(2i, 2j+1), X_m(2i+1, 2j)$ 和 $X_m(2i+1, 2j+1)$ 的均值.此外,图像的小波变换是图像的一种分级多分辨率描述,它也是一种图像金字塔.下面重点介绍均值金字塔搜索算法(MPSA)和小波金字塔搜索算法(WPSA),并介绍作者提出的均值-方差金字塔搜索算法(MVPSA).

3.1 均值金字塔搜索算法

均值金字塔的 $m-1$ 级图像 X_{m-1} 的像素 $X_{m-1}(i, j)$ 为 m 级图像 X_m 中相邻的 2×2 个像素 $X_m(2i, 2j), X_m(2i, 2j+1), X_m(2i+1, 2j)$ 以及 $X_m(2i+1, 2j+1)$ 的平均值.为描述文献[22]提出的均值金字塔快速码字搜索算法,先给出如下定理:

定理 12 定义 $k = 2^n \times 2^n$ 维输入矢量 x 和码字 y_i 对应的均值金字塔分别为 $\{X_0, X_1, \dots, X_n = x\}$ 和 $\{Y_0, Y_1, \dots, Y_n = y_i\}$, 定义 $d_m(x, y_i)$ 为 X_m 和 Y_m 间的平方误差,则

$$d(x, y_i) \geq 4 d_{n-1}(x, y_i) \geq 4^2 d_{n-2}(x, y_i) \geq \dots \geq 4^n d_0(x, y_i) \quad (18)$$

根据定理 12, 均值金字塔搜索算法过程可以描述如下:首先离线建立所有码字的均值金字塔并保存在码书中,并按照均值的大小对码书进行升序排列.搜索时,在线建立输入矢量 x 的均值金字塔,找出与输入矢量均值最近的码字 y_p 作为初始匹配码字,计算 $d_{\min} = d(x, y_p)$. 搜索顺序与 EENNS 算法相同.对于所搜索的码字 y_i , 首先判断 $4^n d_0(x, y_i) \geq d_{\min}$ 是否成立.如果成立,则码字被排除.否则继续判断 $4^{n-1} d_1(x, y_i) \geq d_{\min}$ 是否成立.如果成立,则码字被排除.否则继续判断下一级.相同的过程进行到最后一级,用 FDS 技术计算 $d(x, y_i) = d_n(x, y_i)$. 如果 $d(x, y_i) < d_{\min}$, 则更新 d_{\min} .

均值金字塔搜索算法需要较大的附加计算量和额外存储空间.该算法在大码书和高维矢量的情况下比较有效.为了降低附加计算量和额外存储空间,文献[23]提出如下所述的小波金字塔搜索算法.

3.2 小波金字塔搜索算法

设 $k = 2^n \times 2^n$ 维矢量 x 的 L 级小波变换的第 l 级, $l = -1, -2, \dots, -(L-1)$ 包含四个矢量 $x^{sl}, x^{dl,j}, j = 0, 1, 2$, 这里 x^{sl} 为第 l 级的粗糙矢量而 $x^{dl,j}$ 为第 l 级第 $j+1$ 个细节矢量.令 $d(x^{sl}, y_i^{sl})$ 为输入矢量 x 和码字 y_i 第 l 级粗糙矢量间的失真,即

$$d(x^{sl}, y_i^{sl}) = \sum_{u=0}^{2^{n+l-1}} \sum_{v=0}^{2^{n+l-1}} (x^{sl}(u, v) - y_i^{sl}(u, v))^2 \quad (19)$$

同理 $d(x^{dl,j}, y_i^{dl,j})$ 为 x 和 y_i 间第 l 级第 $j+1$ 个细节矢量间的

平方误差.容易得到如下定理:

定理 13 对于 $k = 2^n \times 2^n$ 维输入矢量 x 和码字 y_i , 设小波变换是正交的, 则

$$d(x^{s^{(l+1)}}, y_i^{s^{(l+1)}}) = d(x^{s^l}, y_i^{s^l}) + \sum_{j=0}^{2^l-1} d(x^{d^{l,j}}, y_i^{d^{l,j}}) \quad (20)$$

由定理 13 可知 $d(x^{s^{(l+1)}}, y_i^{s^{(l+1)}}) \geq d(x^{s^l}, y_i^{s^l}), l = -1, -2, \dots, -(L-1)$. 根据这个性质可以进行小波金字塔搜索算法. 由于该算法的搜索过程与均值金字塔搜索过程相似, 在此不再叙述. 小波金字塔搜索算法的优点是附加计算量小, 不需要额外存储空间. 而且式(20)表明了下级平方误差可以由上级推出, 从而进一步减少了计算量, 所以该算法非常有效.

3.3 均值-方差金字塔搜索算法

在均值金字塔搜索算法的基础上, 作者根据均值-方差不等式, 提出了一种更有效的均值-方差金字塔搜索算法^[24]. 可以定义 $k = 2^n \times 2^n$ 维输入矢量 x 和码字 y_i 对应的均值金字塔分别为 $PM_x = \{X_0, X_1, \dots, X_n = x\}$ 和 $PM_{y_i} = \{Y_0, Y_1, \dots, Y_m = y_i\}$, 对应的方差金字塔分别为 $PV_x = \{V_{(x)0}, V_{(x)1}, \dots, V_{(x)n}\}$ 和 $PV_{y_i} = \{V_{(y_i)0}, V_{(y_i)1}, \dots, V_{(y_i)m}\}$. 对于方差金字塔, 第 n 层的元素 $V_{(x)n}(u, v) = V_{(y_i)n}(u, v) = 0$, 而第 0 层只有一个元素, 如 $V_{(x)0}(0, 0)$ 为 $X_1(0, 0), X_1(0, 1), X_1(1, 0)$ 和 $X_1(1, 1)$ 的方差. 定义 $d_m(x, y_i)$ 为 X_m 和 Y_{im} 间的平方误差, $d_m(x, y_i)$ 为 $V_{(x)m}$ 和 $V_{(y_i)m}$ 间的平方误差, 即:

$$d_m(x, y_i) = \sum_{l=0}^{2^m-1} \sum_{h=0}^{2^m-1} (X_m(l, h) - Y_{im}(l, h))^2 \quad (21)$$

$$d_m(x, y_i) = \sum_{l=0}^{2^m-1} \sum_{h=0}^{2^m-1} (V_{(x)m}(l, h) - V_{(y_i)m}(l, h))^2 \quad (22)$$

令 $D_m(x, y_i)$ 为 x 与 y_i 之间的第 m 层虚平方误差, 即

$$D_m(x, y_i) = d_m(x, y_i) + d_m(x, y_i)/4 \quad (m = 0, 1, \dots, n) \quad (23)$$

根据这些定义, 容易得到如下定理:

定理 14 $d(x, y_i) = D_n(x, y_i) \geq 4 \cdot D_{n-1}(x, y_i) \geq 4^2 \cdot D_{n-2}(x, y_i) \geq \dots \geq 4^n \cdot D_0(x, y_i)$ (24)

由上述定理可得如下删除准则: 若 $4^{n-m} \cdot D_m(x, y_i) \geq d_{\min}, m = 0, 1, \dots, n$, 则可以排除 y_i , 这里假定 $d_{\min} = d(x, y_p)$. 由于搜索过程与 MPSA 算法类似, 在此不再叙述. 显然, $D_m(x, y_i) \geq d_m(x, y_i), m = 0, 1, \dots, n$, 所以定理 14 比定理 12 能删除更多的码字. 但是, 作者算法的缺点是增加了附加计算量和额外存储空间. 仿真实验表明, 作者的算法在大码书和高维矢量情况下比 MPSA 算法有效.

4 基于变换域的最近邻码字搜索算法

不等式是快速码字搜索算法的数学基础, 否则无法排除不匹配码字. 前面所述的各种快速最近邻码字搜索算法大都在空间(时)域下使用不等式来删除码字, 即输入矢量和码字都未作任何变换. 众所周知, 正交变换如 KLT, DCT 和 DWT 都具备能量集中特性, 即能量集中在少数几个变换域系数中, 且变换前后能量保持不变, 如下定理所述:

定理 15 设 A 为 $k \times k$ 正交变换矩阵, 设 x 和 y_i 的变换域矢量分别为 X, Y_i , 即 $X = Ax, Y_i = Ay_i$, 则

$$d(x, y_i) = d(X, Y_i) \quad (25)$$

由定理 15 可知, 变换域矢量间的平方误差与原矢量间的平方误差相等. 因此, 在变换域中找最近邻码字和在空间(时)域中找最近邻码字是等价的. 而变换域矢量间的平方误差往往集中在前几维变换系数上, 在这种情况下使用 PDS 算法将非常有效. 变换域搜索算法的初始匹配码字选择为与变换域输入矢量第一维分量最接近的码字. 变换域搜索算法的优点是可以不使用额外存储空间, 即可以用变换域码字覆盖原始码字(在解码端, 对得到的变换域码字作逆变换即可). 文献[25]提出了一种基于 KLT 变换的快速码字搜索算法. 该算法的缺点是预先计算量比较大, 因为 KLT 变换的特征矢量需要通过大量的训练矢量得到. 下面介绍基于小波变换的和作者的基于哈德码变换的快速码字搜索算法.

4.1 基于小波变换的快速码字搜索算法

文献[26]提出了基于小波变换的快速搜索算法(WTS). 该算法的主要思想是利用小波变换将码字的能量集中到少数几个变换系数中, 然后在这些系数上利用 PDS 算法高效地排除不匹配码字. 该算法适用在图像矢量量化中. 设 $k = 2^n \times 2^n$ 维矢量 x 的 n 级小波变换为 X . X 仍是 $2^n \times 2^n$ 维矢量, 它包含子矢量 x_{L0} (1 维) 和 $2^l \times 2^l$ 维子矢量 $x_{Vl}, x_{Hl}, x_{Dl}, l = 0, 1, \dots, n-1$. 注意到低通子矢量 x_{Ll} 以及高通子矢量 $x_{Vl}, x_{Hl}, x_{Dl}, l = 0, 1, \dots, n-1$ 可由 $x_{L(l+1)}$ 恢复得到, 这里 $x_{L,n} = x$. 而将 $x_{L(l+1)}$ 分解为四个子矢量 $x_{Ll}, x_{Vl}, x_{Hl}, x_{Dl}$ 可采用文献[27]中所述的 QMF 方法.

令 $d^m(X, Y_i) = \sum_{l=0}^{m-1} (X_l - Y_{il})^2$ 为变换域输入矢量与码字间的部分失真, 则有如下定理所述的小波变换域 PDS 准则:

定理 16 假定目前最小失真为 $d_{\min} = d(x, y_p) = d(X, Y_p), 0 \leq p \leq N-1$, 若 $d^m(X, Y_i) \geq d_{\min} (1 \leq m \leq k)$, 则 $d(x, y_i) = d(X, Y_i) \geq d(X, Y_p) = d(x, y_p)$.

WTS 算法需要离线计算所有码字的小波变换, 并按第一个系数的大小对码书进行排序. 选择初始匹配码字 y_p , 其索引为 $p = \arg \min_i d^1(X, Y_i)$, 并计算 $d_{\min} = d(X, Y_p)$. 判断码字 Y_i 是否满足不等式 $|Y_{i0} - X_0| \geq \sqrt{d_{\min}}$. 若满足, 则排除该码字; 否则采用如下 PDS 算法: 从 $m=2$ 开始, 对每一个 $m, m = 2, \dots, 2^n \times 2^n$, 计算 $d^m(X, Y_i)$. 若 $d^m(X, Y_i) \geq d_{\min}$, 则排除该码字; 否则, 继续选择下一个 m 值进行判断. 这个过程直到码字被排除或 $m = 2^n \times 2^n$ 为止. 当 $m = 2^n \times 2^n$ 时, 若 $d^m(X, Y_i) < d_{\min}$, 则 d_{\min} 更新为 $d^m(X, Y_i)$.

WTS 算法选择最简单的 Haar 小波^[27], 因为该小波变换的计算不需要乘法. 此外该算法不需要额外存储空间, 因为原码字可以用变换域码字替换存储, 而在解码端用逆变换恢复. 仿真结果表明该算法非常有效.

4.2 基于哈德码变换的快速码字搜索算法

令 H_n 为 $2^n \times 2^n$ 的哈德码矩阵, 其元素属于集合 $\{1, -1\}$. 假定以下所有矢量为 k 维矢量, $k = 2^n (n > 0)$, 令 $H_1 = \begin{Bmatrix} 1 & 1 \\ 1 & -1 \end{Bmatrix}$, $H_{n+1} = \begin{Bmatrix} H_n & H_n \\ H_n & -H_n \end{Bmatrix}$, 定义矢量 x 的哈德码变换矢量 $X = H_n x$, 则可得如下定理:

定理 17 $H_n H_n = kI_k$, 其中 I_k 是 k 阶单位矩阵.

定理 18 $X_0 = S_x$, 这里 X_0 为矢量 X 的第一维分量, S_x 为矢量 x 的和值.

定理 19 $d(X, Y_i) = kd(x, y_i)$

从上面的定义和定理可以看出, 哈德码变换虽然不是严格意义上的正交变换, 但是变换前后的能量成倍数关系, 因此在变换域中搜索最近码字与空间(时)域中搜索最近码字是等价的. 此外, 哈德码变换矩阵的元素不是 '1' 就是 '-1', 故哈德码变换不需要乘法运算. 基于这些优点, 作者提出了哈德码变换域快速码字搜索算法 (HTS)^[28].

首先离线计算所有码字的哈德码变换, 并按照变换域码字第一个分量(等于原码字的和值)的大小对码书进行排序. 搜索时, 首先计算输入矢量 x 的哈德码变换矢量 X , 然后选择第一维系数与 X_0 (或 S_x) 最近的码字 Y_p 为初始匹配码字, 计算 $d_{\min} = d(X, Y_p)$. 令 $MINSUM = X_0 - \sqrt{d_{\min}}$, $MAXSUM = X_0 + \sqrt{d_{\min}}$, 对码字 Y_i , 若有 $Y_{i0} > MAXSUM$ 或 $Y_{i0} < MINSUM$, 则排除该码字; 否则采用如下 PDS 算法: 从 $m = 2$ 开始, 对每一

个 $m, m = 2, \dots, 2^n$, 计算 $d^m(X, Y_i)$. 若 $d^m(X, Y_i) \geq d_{\min}$, 则排除该码字; 否则, 继续选择下一个 m 值进行判断. 这个过程直到码字被排除或 $m = k = 2^n$ 为止. 当 $m = k = 2^n$ 时, 若 $d^m(X, Y_i) < d_{\min}$, 则 d_{\min} 更新为 $d(X, Y_i)$, 同时更新 $MAXSUM$ 和 $MINSUM$.

与小波变换域算法相比, 作者的算法要求矢量维数 $k = 2^n$ 而小波变换域算法要求矢量维数 $k = 2^n \times 2^n$, 而且哈德码变换的计算比小波变换的计算简单得多. 另外, 哈德码变换域矢量的第一维分量等于原矢量和值的特性使算法搜索过程与 ENNS 算法极为相似, 但效率却比 ENNS 算法高得多.

5 仿真实验和结论

本文所提到的各种算法的仿真均在 IBM 奔腾 III 微机上进行. 所有算法都以 256 灰度 512 \times 512 图像的矢量量化编码为应用背景. 算法所用到的 4 个码书均以 256 灰度 512 \times 512 Lena 图像作为训练图像并采用 LBG 算法^[2]生成, 这些码书的大小为 256 和 512 两种, 码字的维数有 4 \times 4 和 8 \times 8 两种.

表 1 各种算法的离线计算量和额外存储空间比较

算法	离线计算量				额外存储空间		
	加减法	乘除法	比较	开方	均/和值	方差	绝对误差
FS	0	0	0	0	0	0	0
PDS	0	0	0	0	0	0	0
AEI	0	0	0	0	0	0	$N \cdot (k + 1)$
Minimax	0	0	0	0	0	0	$N \cdot (k + 1)$
IAEI	0	0	0	0	0	0	$N \cdot (k + 1)$
ENNS	$N(k - 1)$	N	$N(N - 1)/2$	0	N 均值	0	0
EEENS	$N(3k - 2)$	$N(k + 1)$	$N(N - 1)/2$	N	N 均值	N	0
MEENNS-2	$N(3k - 2)$	$N(k + 1)$	$N(N - 1)/2$	N	N 均值	N	0
*MEENNS-3	$N(6k - 6)$	$2Nk$	$N(N - 1)/2$	$3N$	$3N$ 和值	$3N$	0
FNNS	$N(N - 1)(2k - 1)/2$	$Nk(N - 1)/2$	0	0	$N(N - 1)/2$ 个码字间失真		
*MFFNNS	$N(k - 1)$	0	$N(N - 1)/2$	0	N 和值	0	0
DTA	$N(k - 1)$	Nk	$N(k - 1)$	N	N 个 2-范数和 N 个分量最大值		
WTS	N 个 QMF	0	$N(N - 1)/2$	0	0	0	0
*HTS	$Nk(k - 1)$	0	$N(N - 1)/2$	0	0	0	0
MPSA	$3N \sum_{i=0}^{p-1} 4^i$ $p = (\log_2 k)/2$	$N \sum_{i=0}^{p-1} 4^i$	$N(N - 1)/2$	0	$N \sum_{i=0}^{p-1} 4^i$ 均值	0	0
*MVPSA	$10N \sum_{i=0}^{p-1} 4^i$ $p = (\log_2 k)/2$	$5N \sum_{i=0}^{p-1} 4^i$	$N(N - 1)/2$	$N \sum_{i=0}^{p-1} 4^i$	$N \sum_{i=0}^{p-1} 4^i$ 均值	$N \sum_{i=0}^{p-1} 4^i$	0
*ASRSS	$N(3k - 2)$	$N(k + 1)$	$2N(N - 1)$	0	$4N$ 个特征值		

表 2 各种最近邻搜索算法的编码性能比较

算法	编码时间				平均失真计算			
	256 4 \times 4	256 8 \times 8	512 4 \times 4	512 8 \times 8	256 4 \times 4	256 8 \times 8	512 4 \times 4	512 8 \times 8
FS	6.49	6.26	12.96	12.75	256.0	256.0	512.0	512.0
PDS	1.48	1.15	2.59	1.87	39.32	34.91	69.20	57.11
ENNS	0.44	0.38	0.68	0.64	6.82	9.48	11.72	15.01
EEENS	0.28	0.25	0.38	0.36	5.01	7.04	8.15	10.65
MEENNS-2	0.27	0.22	0.36	0.33	4.31	6.07	6.99	9.19
*MEENNS-3	0.27	0.21	0.32	0.30	3.58	5.45	6.15	8.27
FNNS	1.02	0.96	1.78	1.69	14.32	18.56	21.56	29.91
*MFFNNS	0.32	0.27	0.42	0.40	6.79	9.43	11.54	14.89
MPSA	0.30	0.27	0.41	0.39	6.75	9.45	11.69	14.95
*MVPSA	0.27	0.24	0.39	0.37	4.98	7.01	8.10	10.61
WTS	0.25	0.21	0.32	0.29	4.89	6.85	7.98	9.79
*HTS	0.25	0.20	0.30	0.27	4.75	6.76	7.67	9.58

作者选择了一些典型的算法参与比较,加星号的表示作者的算法.表 1 比较了各种算法的离线计算量和额外存储空间.表中 N 为码书大小, k 为矢量维数,加星号的表示作者的算法.在不同的码书尺寸和矢量维数下,表 2 比较了各种最近邻搜索算法的编码时间和平均失真计算.码书尺寸为 256 或 512,矢量维数为 4×4 或 8×8 .从表 1 中可以看出,一种算法编码效率的提高往往需要付出大量离线计算和存储的代价.由于离线计算不占用编码时间,所以作者的某些算法需要较高的离线计算量和额外存储空间是值得的和可行的.而基于绝对误差不等式的算法虽然不需要离线计算量,但是它的效率却不高,因为其在在线计算量及搜索范围大.从表 2 中可以看出,在基于不等式的搜索算法中,基于均值不等式的算法比基于三角不等式和绝对误差不等式的算法效率高;而基于变换域的算法比基于金字塔结构的算法和基于均值不等式的算法效率高.

参考文献:

- [1] Gersho A, Gray R M. Vector Quantization and Signal Compression [M]. Boston: Kluwer Academic Publishers, 1992.
- [2] Linde Y, Buzo A, Gray R M. An algorithm for vector quantizer design [J]. IEEE Trans. ,1980, COM28(1) :84 - 95.
- [3] Bei C D, Gray R M. An improvement of the minimum distortion encoding algorithm for vector quantization [J]. IEEE Trans. ,1985, COM33(10) :1132 - 1133.
- [4] Soleymani M R, Morgera S D. An efficient nearest neighbor search method [J]. IEEE Trans. ,1987, COM35(6) :677 - 679.
- [5] Cheng D, Gersho A, Ramamurthi B, Shoham Y. Fast search algorithms for vector quantization and pattern matching [A]. International Conference on ASSP [C], 1984:9. 11. 1 - 9. 11. 4.
- [6] Pan J S, McInnes F R, Jack M A. Bound for minkowski metric or quadratic metric applied to VQ codeword search [J]. IEE Proceedings-Vision, Image and Signal Processing, 1996, 143(1) :67 - 71.
- [7] Orchard M T. A fast nearest neighbor search algorithm [A]. International Conference on ASSP [C], 1991:2297 - 2300.
- [8] Vidal E. An algorithm for finding nearest neighbors in (approximately) constant average time [J]. Pattern Recognition Letters, 1986, 54 :145 - 157.
- [9] Li W, Salari E. A fast vector quantization encoding method for image compression [J]. IEEE Transactions on Circuits and Systems for Video Technology, 1995, 5(2) :119 - 123.
- [10] 陆哲明,潘正祥,孙圣和.一种矢量量化码字搜索的快速算法[J].电子学报,2000,28(2) :133 - 135.
- [11] Guan L, Kamel M. Equal-average hyperplane partitioning method for vector quantization of image data [J]. Pattern Recognition Letters, 1992:693 - 699.
- [12] Pan J S, Huang K C. A new vector quantization image coding algorithm based on the extension of the bound for minkowski metric [J]. Pattern Recognition, 1998, 31(11) :1757 - 1760.
- [13] Lee C H, Chen L H. Fast closest codeword search algorithm for vector quantization [J]. IEE Processings-Vision, Image and Signal Processing, 1994, 141(3) :143 - 148.
- [14] Ghosh D, Shivaprasad A P. Fast codeword search algorithm for real-time codebook generation in adaptive VQ [J]. IEE Processings-Vision, Image and Signal Processing, 1994, 144(5) :278 - 284.
- [15] Baek S J, Jeon B K, Sung K M. A fast encoding algorithm for vector quantization [J]. IEEE Signal Processing Letters, 1997, 4(12) :325 - 327.
- [16] Xu R S, Lu Z M, Xu X M, Zhang W D. An efficient fast encoding algorithm for vector quantization [J]. Journal of Shanghai JiaoTong University, 2000, E-5(2) :23 - 27.
- [17] Torres T, Huguet J. An improvement on codebook search for vector quantization [J]. IEEE Trans. ,1994, COM42(2) :208 - 210.
- [18] Wu K S, Lin J C. Fast VQ encoding by an efficient kick-out condition [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2000, 10(1) :59 - 62.
- [19] Lin Y C, Tai S C. A fast Linde Buzo Gray algorithm in image vector quantization [J]. IEEE Trans. Circuits and Systems I: Analog and Digital Signal Processing, 1998, 45(3) :432 - 435.
- [20] Burt P J, Adelson E. The Laplacian pyramid as a compact image code [J]. IEEE Trans. ,1983, COM31(4) :532 - 540.
- [21] Wang L, Goldberg M. Reduced-difference pyramid: a data structure for progressive image transmission [J]. Optical Engineering, 1989, 28(7) :708 - 716.
- [22] Lee C H, Chen L H. A fast search algorithm for vector quantization using mean pyramids of codewords [J]. IEEE Trans. ,1995, COM43(2/3/4) :1697 - 1702.
- [23] Hwang W J, Chen B Y, Jeng S S. A fast vector quantization encoding method using wavelet transform [J]. Pattern Recognition Letters, 1997, 18:73 - 76.
- [24] Pan J S, Lu Z M, Sun S H. Fast codeword search algorithm for image coding based on mean variance pyramids of codewords [J]. Electronics Letters, 2000, 36(3) :210 - 211.
- [25] Lee C H, Chen L H. High-speed closest codeword search algorithms for vector quantization [J]. Signal Processing, 1995, 43:323 - 331.
- [26] Hwang W J, Jeng S S, Chen B Y. Fast codeword search algorithm using wavelet transform and partial distance search techniques [J]. Electronics Letters, 1997, 33(5) :365 - 366.
- [27] Vetterli M, Kovacevic J. Wavelets and Subband Coding [M]. New York: Prentice Hall, 1995.
- [28] Lu Z M, Pan J S, Sun S H. Efficient codeword search algorithm based on Hadamard transform [J]. Electronics Letters, 2000, 36(16) :1364 - 1365.

作者简介:

孙圣和 (见本期第 1782 页)