

H. 263 中基于全零块发现的运动搜索方法

兰军强 何 芸 曹志刚

(清华大学电子工程系, 北京 100084)

摘 要 H. 324 系统中, H. 263 图象编码速度一直是影响系统实时性能的瓶颈, 而运动搜索和 DCT 等模块又是其中运算量最大的模块. 为了进一步降低图象编码的运算量, 提出了一种基于零块发现的运动搜索方法. 该方法在对活动图象进行运动搜索时, 应用了一种基于发现全零块的中止准则, 使得 H. 263 中占用运行时间最多的 INTER 块的运动搜索, 收敛得更快, 它尤其提高了对背景图象或者变化较小的图象部分的编码效率, 同时也较大幅度地减小了 DCT 等模块所占用的计算时间. 该算法在提高速度的同时, 仍保持了 H. 263 原有压缩算法的效率和图象质量.

关键词 图象编码 H. 263 运动搜索 零块发现 离散余弦变换(DCT)

中图法分类号: TN919.81 文献标识码: A 文章编号: 1006-8961(2000)11-0925-07

Motion Search Based on Zero Block Detection in H. 263

LAN Jun-qiang, HE Yun, CAO Zhi-gang

(Department of Electronic Engineering, Tsinghua University, Beijing 100084)

Abstract To further improve the performance of H. 263's encoder in the real time H. 324 system, a motion search algorithm using zero block detection technology, is proposed in this paper. This algorithm exploits the fact that many residual blocks become all zero blocks after DCT and coarse quantization in very low bit rate video coding. To further speed up the encoder implementation, this zero detection technology is combined with motion estimation. In the motion search process, if an all zero macro-block is detected using the distortion value of the macro-block, the search process is stopped. The choice of the threshold for zero detection is critical. Simulation results show that the algorithm can dramatically decrease the motion search points. And also many blocks' quantization, DCT and other reconstruction steps are skipped. Under typical quantization parameter of 12, the average search points decrease is 30% ~ 50%, and 25% ~ 30% blocks' DCT and quantization is skipped due to the zero detection in the motion search. With the gain of encoder speed, the algorithm does not suffer the coding efficiency of the H. 263 baseline. The algorithm does not affect the reconstructed image quality either. After the algorithm is combined into a real-time H. 324 system, the performance of the whole system is improved a lot.

Keywords Image/Video coding, H. 263, Motion estimation, Zero block detection, Discrete Cosine Transform

0 引 言

ITU-T H. 324^[1]是基于通用电话网(GSTN)的可视电话协议, 在 H. 324 中, 是采用基于块运动补偿和 DCT 技术(MC+ DCT)的传统图象编码方案——H. 263^[2]作为其图象编码的标准协议. 这种编码方法由于其简单、稳健和高效性而得到了广泛的

实际应用, 已成为目前在低速率图象通信领域最为有效的图象编码方案.

在 H. 324 实时通信系统中, H. 263 图象编码通常是最消耗 CPU 时间的模块, 因而实时运算时需要对其算法进行简化^[3]. 由于编码模块中计算量最大的部分是运动搜索和 DCT、反 DCT 等模块, 因此省略 DCT 或减少 DCT 的计算时间, 便可大大提高 H. 263 编码速度. 经研究, 了解到在低速率的情况

下, H. 263 中使用的量化参数一般较大, 因而使很多编码块经过运动补偿后的残差经 DCT 量化后变为全零块. 因此利用这个特点, 一些论文提出了全零块的提前检测技术^[4,5]. 由于这些技术是在完成运动搜索和补偿后, 在对残差块做 DCT 及量化以前, 就利用这个块的一些特性来提前确认此块是否将成为全零块, 因而省去了 DCT 和量化的时间, 同时也省掉了 IDCT 的时间. 利用这个技术, 在理想的情况下, 可以省掉 60% 的 DCT 和 IDCT 模块的调用时间^[4,5], 但这些方法却都是只在 DCT 以前、运动搜索以后进行的, 并没有考虑如何利用这种零块提前检测和运动搜索的结合来, 提高运动搜索的效率.

本文提出的基于全零块发现的运动搜索算法正是一种把这种零块发现技术和运动搜索模块相结合的方法, 从而在保留文献[4]、[5]方法中的节省 DCT 和量化模块时间等优点的同时, 进一步提高了运动搜索模块的效率和速度, 从而使 H. 263 的整体运算速度又有了较大幅度的提高.

1 H. 263 和运动搜索算法

H. 263 视频编码是基于 MC+ DCT 的编码方案, 而且 H. 263 算法以分块结构为特征, 即每一帧图象被分割为基本的编码单元——宏块(16×16 个像素点的图象阵列), 而每个宏块又包含 6 个 8×8

的小块, 其中包括 4 个亮度块, 2 个色差块. H. 263 的基本系统中(不包含任何选项时), 在编码一帧图象前, 首先要确定这一帧的编码模式, 这里有两种编码模式可供选择, 即 INTRA 帧和 INTER 帧编码模式. 其中, INTRA 帧的编码与 JPEG 类似, 其编解码不需要依赖别的帧; 而编码效率较高的 INTER 帧则需要依赖参考帧; 编码时, 需依据参考帧进行运动搜索, 解码时, 也需依据参考帧进行运动补偿. 并且 INTER 帧中的每个宏块又有帧内模式(INTRA)和帧间模式(INTER)两种编码模式. 其中, INTRA 模式与前一帧无关, 即每块直接作 DCT 后、量化、编码输出, 而 INTER 模式则与其它帧有关, 它是在参考帧中搜索到最佳匹配块后, 对帧间差值作 DCT 变换和量化, 最后把各宏块的运动矢量和量化后的 DCT 系数, 经过变长编码形成码流. 显见, INTRA 帧都由 INTRA 宏块组成, 而 INTER 帧中的宏块则可以选择一种编码模式. H. 263 协议与其它传统的图象编码标准, 如 MPEG1、MPEG2、H. 261 等协议相同, 也严格地规定了解码过程. 而相对于解码过程, 编码过程就有很多开放的部分, 它们可以依据各种具体编码实现过程的不同而不同, 例如运动搜索的方法等. INTRA 帧的编码较 INTER 帧的编码相对简单, 下面以一个 INTER 帧中宏块的编码过程为例, 来介绍 H. 263 的编码过程(图 1).

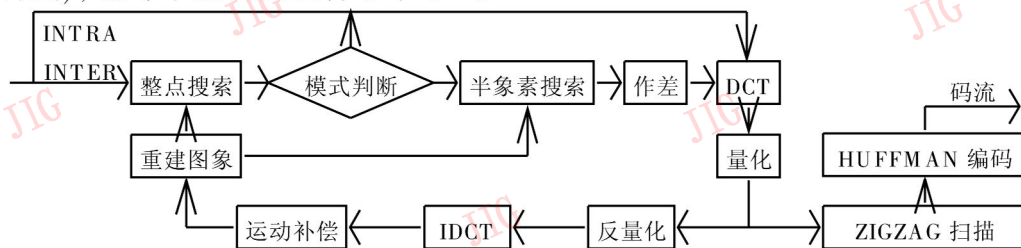


图 1 宏块编码流程图

首先对要编码的宏块进行运动匹配, 匹配之后再行模式判断, 如果作 INTRA 编码更加有效, 则选择 INTRA 模式, 否则就进行半像素运动搜索, 即在上一帧中找到与当前宏块最匹配的预测宏块, 然后从当前宏块中减去预测宏块而得到残差, 再分别对差值宏块中的每个块进行 DCT 变换和量化. 除了重建过程中的反量化、IDCT、运动补偿和形成码流的扫描和 HUFFMAN 编码等与解码端具有相同的模块外, 剩下的模块都可以根据应用场合和需求的不同, 而采用不同的实现方法. 以运动搜索为例, 编码器的目标只要能够得到一个矢量就可以, 至于矢量是否是最优,

H. 263 标准中并不规定, 因而得到矢量的方法可以有很大的区别, 而且运算的复杂度、速度和效率也可以因此而十分不同. 例如在极端的情况下, 假设 CPU 的计算速度非常慢, 但带宽却足够的情况下, 可以为提高运算速度, 而把所有的矢量都设为零(简化为 DPCM 的情况), 此时并不需要进行运动搜索, 从而节省了大量的计算时间, 但产生的码率相对较高. 由此可见, H. 263 中所得到的运动矢量并不一定是图象中每个宏块“真正”的运动方向, 而只是在一定编码条件(如运算速度和搜索算法等)限制下所能够得到的指向“最优”匹配块的运动矢量.

如前所述, H. 263 中运动搜索的目标就是在一定编码能力限制下找到与当前块“最匹配”的块, 而所有的运动搜索算法都是依据一定的失真准则 (SAE, MSE, MAD, VAR, MAX 等), 并在这个运动搜索算法所遍历的点中寻找最优的匹配点. 这些失真准则的定义如下:

SAE: mean absolute error——绝对误差准则

$$SAE = \sum_{i=1}^8 \sum_{j=1}^8 |f(i, j)|$$

其中, $f(i, j)$ 是 (i, j) 处的像素差值

MSE: mean square error——平方误差准则

$$MSE = \sum_{i=1}^8 \sum_{j=1}^8 (f(i, j))^2$$

MAD: mean absolute difference——绝对值差准则

$$MAD = \sum_{i=1}^8 \sum_{j=1}^8 |f(i, j) - \mu|$$

其中, μ 是残差块的平均值

VAR: block variance——块平方方差准则

$$VAR = \sum_{i=1}^8 \sum_{j=1}^8 (f(i, j) - \mu)^2$$

其中, μ 是残差块的平均值

MAX: maximum value——块最大误差准则

$$MAX = \max(f(i, j)), \quad i, j \in (1, 8)$$

其中, 由于 SAE 的运算只涉及加减法和求绝对值, 其运算相对简单, 而且性能和 MSE 十分接近, 同时在上述准则中的性能/计算量的比值最高, 因此, 它已成为目前使用最广泛的失真准则. 本文算法就是基于 SAE 失真准则进行讨论的(下面将把 SAE 记为 E_a). 对于给定的运动矢量 \mathbf{p} , 一个宏块的失真函数 E_a 值为

$$E_a(\mathbf{p}) = \sum_{i=1}^{16} \sum_{j=1}^{16} |f(i, j) - \hat{f}_p(i, j)|$$

其中, f 是当前宏块的像素值; \hat{f}_p 是重建帧中矢量 \mathbf{p} 指向的宏块的像素值. 那么, 运动搜索的过程就是在算法和搜索窗口所指定的矢量集合 \mathbf{V} 中寻找使失真函数达到最小的运动矢量 \mathbf{P}_{\min} . 因此有

$$\mathbf{P}_{\min} = \arg \min_{\mathbf{p} \in \mathbf{V}} E_a(\mathbf{p}) \quad (1)$$

2 零块发现技术

在文献[4]、[5]等论文中, 已经讨论了在获得最佳匹配的运动矢量之后, 就可利用这一宏块的 6 个 8×8 块的失真值进行判断, 以决定每个 8×8 块是

否为零块. 一个块 E_a 的失真值如下

$$\begin{aligned} E_a(k) &= \sum_{i=1}^8 \sum_{j=1}^8 |f^k(i, j) - \hat{f}_p^k(i, j)| \\ &= \sum_{i=1}^8 \sum_{j=1}^8 |r_p^k(i, j)|, \quad k = 1, 2, \dots, 6 \end{aligned}$$

其中, 最匹配块的运动矢量是 \mathbf{p} , f^k 是当前宏块第 k 块的像素值; \hat{f}_p^k 是最匹配宏块第 k 块的像素值; r_p^k 是它们的差, 即运动补偿后的残差. 这些残差的 DCT 系数^[6,7] 为:

$$F(u, v) = \frac{1}{4} c(u) c(v) \times \sum_{i=0}^7 \sum_{j=0}^7 r(i, j) \cos \left| \frac{(2i+1)u\pi}{16} \right| \cos \left| \frac{(2j+1)v\pi}{16} \right|$$

其中, F 是 DCT 变换后的系数; r 代表要做 DCT 的残差块的值. $c(u)$ 只有在 $u=0$ 时, 为 $\frac{1}{\sqrt{2}}$, 其它时候都是 1. 于是根据“和的绝对值小于绝对值的和”以及余弦函数小于 1 等特性可知

$$|F(u, v)| \leq \frac{1}{4} c(u) c(v) \times \sum_{i=0}^7 \sum_{j=0}^7 |r(i, j)| \begin{cases} \frac{1}{4} E_a, & \text{for } u, v \neq 0 \\ \frac{1}{8} E_a, & \text{for } u, v = 0 \end{cases} \quad (2)$$

文献[4]中, 在此时仅仅要求 $F(u, v) < 2Q_p$ 就可以了, 但是在 H. 263 中最常用的量化公式是

$$Cof(u, v) = \text{sgn} F(u, v) \frac{|F(u, v)| - Q_p/2}{2Q_p}$$

$Cof(u, v)$ 指量化后的 DCT 系数, Q_p 是量化参数. 为了保证所有的 $Cof(u, v)$ 都等于零, 从文献[5]可以得到如下公式

$$|F(u, v)| - Q_p/2 < 2Q_p$$

也就是

$$|F(u, v)| < \frac{5}{2} Q_p \quad (3)$$

通常一个变换的 DCT 直流系数最大, 因此, 只要直流系数满足式(3), 就可以保证整个量化后的系数为零. 直流系数就是式(2)中 $u, v=0$ 的情况, 而这种情况的上界是 $\frac{1}{8} E_a$, 因此如果满足

$$\frac{1}{8} E_a < \frac{5}{2} Q_p, \quad \text{即 } E_a < 20Q_p \quad (4)$$

那么, 就可以确认整个 8×8 块的系数为零了. 下面讨论本文提出的将式(4)和运动搜索模块相结合而得到的基于零块发现的运动搜索算法.

3 基于零块发现的运动搜索

运动搜索就是在搜索算法指定的路径中寻找最匹配的运动矢量(式(1)). 各种搜索算法的主要区别就在于如何选择指定搜索窗口内的搜索路径和收敛原则. 如果把某个特定算法对一个宏块所遍历的矢量集合记为 V , $V = \{p_0, p_1, \dots, p_E\}$, 其中的 $0, 1, 2, \dots$ 等序号是这个搜索算法所遍历矢量的顺序, 则所有常规的搜索算法可以用如下的算法语言描述

```
SAEMin= MAX- INTEGER;
for(i= 0; i <= E; i+ +)
{
    // 计算  $P_i$  处的 SAE, SAE( $P_i$ );
    ...
    if(SAE( $P_i$ ) < SAEMin)
    {
        SAEMin= SAE( $P_i$ );
         $P_{min}$ =  $P_i$ ;
    }
    // 决定下一个搜索位置  $P_{(i+1)}$ 
    ...
}
```

但如果被编码的宏块是一个相对静止的块, 或者只经过了小范围的平动, 那么, 经过运动补偿后的帧差会很小, 经 DCT 量化后将变为全零, 此时, 若再进行很多次搜索, 将十分浪费. 由于这时在搜索的过程中, 有可能会出现几个矢量位置的失真值相差很小, 而且这些宏块经过 DCT 以后都会变为全零宏块, 因此在这样位置中进行搜索和进行失真值的比较, 对于目前的量化器是没有效果的.

基于此, 提出本文算法的思路如下: 在搜索过程的某个矢量位置 P_j 时, 如果利用得到的失真值确认这个宏块为全零宏块的话, 就立即停止运动搜索. 设原有搜索算法在指定的搜索过程中遍历到的所有满足全零条件的矢量集合均计为 M , 那么运动搜索表述如下

$$P_{min} = P_j \begin{cases} j = \min_{p_i \in M} i, M \neq \emptyset \\ j = \arg \min_{p \in V} E_a(p), M = \emptyset \end{cases} \quad (5)$$

也就是说, 搜索过程中如果存在满足全零块的矢量, 那么最优匹配矢量就是搜索中遇到的第一个这样的矢量 P_j ; 如果不存在这样的矢量, 那么矢量搜索就和常规的运动搜索方法相同, 即同式(1). 由

此可见, 计算量的减少来自于搜索过程中存在全零块的情况. 此时, 仅从产生码流的角度而言, 如果 P_j 能使残差块成为全零块, 即使可能存在失真更小的块 $P_i (j < i)$, 目前的量化器也已经不能分辨, 因为此时产生的 DCT 码流为全零块的码流, P_i 的码流也不可能更小. 同时由于在本文描述的运动搜索算法中, 运动搜索的顺序是由近及远的, 即矢量 P_i 离 $(0, 0)$ 点的距离一般大于 P_j 离 $(0, 0)$ 点的距离, 而且相应的矢量码流一般也是 P_i 较大, 所以如果使用矢量 P_j 进行编码, 所得到的码流也将是最小.

实际图象编码中, 由于运动搜索是以 16×16 宏块单位的失真值进行比较的, 而发现零块的准则是用 8×8 块的失真值, 因此需要把一个 16×16 宏块的 E_a 运算, 分解成 4 个 8×8 块的 E_a 运算. 因为基本的加减和求绝对值运算的次数是相等的, 而分解只增加了把 4 个值相加的 3 次加法. 本来应该对 6 个块都进行全零确认, 才能判断这个宏块是否为全零块. 本算法使用简化的方法以方便运算, 即如果能够确认 4 个亮度块为全零块, 则认为两个色度块也成为全零块. 这样的简化是利用了实际图象中亮度的分辨率比色度分辨率高的特点, 即如果亮度块足够“像”(差经 DCT 量化为零), 那么色度应该也足够“像”, 即如果满足下式

$$E_{a8 \times 8}(k) < 20Q_p, k = 1 \sim 4 \quad (6)$$

那么就认为这个宏块为全零的宏块.

如果用函数 $IsZero(SAE) = 1$ 来描述满足式(6)的宏块, 而用 $IsZero(SAE) = 0$ 来描述不满足式(6)的宏块, 改进后的运动搜索算法的过程就可以用如下语句进行描述:

```
SAEMin= MAX- INTEGER;
for(i= 0; i <= E; i+ +)
{
    // 计算  $P_i$  处的 SAE, 即 SAE( $P_i$ );
    ...
    if(IsZero(SAE( $P_i$ )) == 1)
    {
        SAEMin= SAE( $P_i$ );
         $P_{min}$ =  $P_i$ ;
        break; // 停止运动搜索, 退出循环;
    }
    else if(SAE( $P_i$ ) < SAEMin)
    {
        SAEMin= SAE( $P_i$ );
         $P_{min}$ =  $P_i$ ;
    }
}
```

```

}
// 决定下一个搜索位置 P(i+1)
...
}
    
```

由此可见, 本算法实际上是在常规的运动搜索方法中加入了一个停止条件, 当运动搜索到满足式 (6) 就可以停止, 即把当前的矢量记为本块的运动矢量, 并把此时的宏块残差标记为全零. 注意, 本文中所提到的运动搜索包括整点象素搜索和半象素搜索两部分, 在这两部分中都可以使用本算法. 在把当前宏块标记为全零块的同时, 后续的 DCT、量化、反量化、IDCT 和运动补偿的相加等运算也被相应的省掉, 从而也就节省了 DCT 等模块的运算量.

4 实验结果和分析

4.1 实验环境

本文实验利用 TMN8^[8]所提供的软件, 作为编解码平台, 所有的结果比较都是在只有运动搜索模块不同的条件下进行的. 本实验对多个图象序列进行了测试和比较, 这里只选取其中的 3 个具有代表性的标准测试序列—— Miss America, Mother and Daughter, 和 Car Phone 序列来进行分析比较. 这 3 个图象序列(下简称序列)分别代表了不同复杂度的图象, 其中 Miss America 最为简单, 而 Car Phone 最复杂. 实验中选用了每个序列的 50 帧逐行的 QCIF (176×144 象素, YUV 4: 2: 0) 图象作为图象源.

TMN8 中提供的快速算法是钻石搜索法^[8], 钻石搜索法是目前 H. 263 算法中最成功的快速整点搜索算法, 该算法在将平均搜索次数降低到 20 次/宏块(整点和半象素)的同时, 能够达到与全搜索算法基本相同的压缩比和图象质量. 本文算法则是在钻石搜索法的基础上结合应用了零块发现技术, 文中还给出了和钻石搜索法的比较. 算法之间的性能比较, 是从运算复杂度、压缩比和图象质量等 3 个方面进行衡量的. 运算复杂度是以每个宏块的平均运动搜索次数和由这种算法引入的 DCT 等模块的运算跳过比例来进行比较的. 由于帧数固定, 所以码流长度与码率具有相同的效果, 所以压缩效率是以编码后码流长度来比较的. 为了去掉各种码率控制策略所引起的不同, 本实验采用了固定量化步长的方案. 图象质量是用平均峰值信噪比 $PSNR$ 来衡量的(单位是 dB), 平均峰值信噪比 $PSNR$ 就是多帧

峰值信噪比的均值.

$$\overline{PSNR} = \frac{1}{L} \sum_{k=1}^L 10 \lg \left| 255^2 \left| \sum_{i=1}^M \sum_{j=1}^N |f_k(i,j) - \hat{f}_k(i,j)| \right| \right| \quad (7)$$

4.2 运算速度

通过应用零块发现准则, 运动搜索次数有了较大程度的减少. 本文算法最理想的一种情况是, 如果连续两帧编码图象中同一位置宏块处的图象是静物或者是背景, 即两帧图象的差别很小, 那么本算法可能只要进行一次 E_a 计算, 就能确认本宏块是否是全零宏块. 此时本宏块的编码只用到一次 E_a 运算. 表 1 是 3 个序列两种算法的比较, 其中对 miss 序列, 常规钻石搜索法的平均搜索次数约为 20 次左右; 而基于零块发现的搜索法则在量化参数大于 5 以后, 就很快地降低到 10 次以下, 但 Mother & Daughter 和 Car Phone 两个序列搜索次数的减少没有 Miss 序列那样显著. 以量化参数为 10 的时候为例, Mother & Daughter 和 Car Phone 的减少次数都只有 4 次左右, 即减少比例为 25% 左右.

表 1 两种算法搜索次数的比较

| 量化参数 Q_p | 搜索次数 | | | | | |
|---------------|---------|------|--------|-------|--------|-------|
| | Miss 序列 | | M&D 序列 | | Car 序列 | |
| | 常规算法 | 本文算法 | 常规算法 | 本文算法 | 常规算法 | 本文算法 |
| 5 | 19.39 | 9.72 | 16.17 | 13.31 | 17.22 | 15.47 |
| 10 | 19.69 | 7.81 | 16.26 | 12.56 | 17.58 | 13.81 |
| 15 | 19.82 | 6.25 | 16.32 | 11.93 | 17.57 | 12.53 |
| 20 | 19.87 | 5.17 | 16.51 | 11.00 | 17.59 | 11.32 |
| 25 | 20 | 4.45 | 16.57 | 9.97 | 17.59 | 9.98 |

运算速度的提高另外一个方面体现在一旦零块被发现, 后续的 DCT 等运算也将被全部跳过. 表 2 列出了本算法在使用钻石搜索算法时, 所带来的零块跳过占所有块的百分比, 也就是后续的 DCT 等运算, 所节省的运算量的百分比. 综合以上两项运算量的节省, 可见本算法使得编码过程中运算量最大的运动搜索和 DCT 等模块的运算量均有了较大幅度的节省.

表 2 跳过零块的比例(省去 DCT 等后续运算的比例)

| 量化参数 Q_p | 跳过零块的比例(%) | | |
|------------|------------|--------|--------------|
| | Miss 序列 | M&D 序列 | Car Phone 序列 |
| 5 | 46.53 | 16.78 | 10.35 |
| 10 | 58.96 | 22.0 | 21.34 |
| 15 | 68.58 | 26.37 | 28.26 |
| 20 | 74.73 | 34.30 | 35.62 |
| 25 | 78.54 | 40.43 | 42.90 |

4.3 压缩比

由于本算法提前结束了常规的运动搜索,并将本块变为全零块,从而减小了码流.但另一方面由于所得到的运动矢量可能是次最优的,因此影响了正在编码帧的重建图象质量;又由于 H. 263 算法对过去帧的依赖性,使得总体的码率(用码流长度衡量)有时稍有增大,但从表 3 中数据可知,增大的幅度非

常的小.表 3 列出了几种典型量化参数下的码流长度的变化,其中码流长度的单位是 byte.由于各个序列相差不大,所以只列出了 Miss America 和 Car Phone 两个序列的情况.由表中数据可以看出,本算法在大大节省运算时间的情况下,并没有引起码率的显著变化,仍保持了 H. 263 算法压缩比高的特性.

表 3 零块发现搜索算法引起的码率变化

| 量化参数 Q_p | Miss 码流长度(byte) | | 码流长度变化 (%) | Car 码流长度(byte) | | 码流长度变化 (%) |
|------------|-----------------|--------|------------|----------------|--------|------------|
| | 常规算法 | 本文算法 | | 常规算法 | 本文算法 | |
| 5 | 22 241 | 22 000 | - 1.10 | 51 643 | 51 675 | + 0.06 |
| 10 | 8 461 | 8 529 | + 0.80 | 19 030 | 19 171 | + 0.75 |
| 15 | 5 602 | 5 521 | - 1.44 | 10 939 | 10 967 | + 0.26 |
| 20 | 4 460 | 4 335 | - 2.80 | 7 462 | 7 464 | + 0.03 |
| 25 | 3 836 | 3 693 | - 3.72 | 5 809 | 5 698 | - 1.91 |

4.4 图象质量

表 4 表示了各序列在两种方法下 $PSNR$ 的差别.

表 4 两种算法的 $PSNR$ (dB) 的比较

| 量化参数 Q_p | Miss 序列 | | M&D 序列 | | Car 序列 | |
|------------|---------|--------|--------|--------|--------|--------|
| | 常规算法 | 本文算法 | 常规算法 | 本文算法 | 常规算法 | 本文算法 |
| 5 | 40.603 | 40.563 | 36.835 | 36.821 | 37.421 | 37.414 |
| 10 | 37.126 | 36.937 | 32.842 | 32.833 | 33.106 | 33.062 |
| 15 | 35.405 | 34.985 | 31.002 | 30.966 | 31.052 | 30.92 |
| 20 | 34.29 | 33.6 | 29.779 | 29.731 | 29.573 | 29.351 |
| 25 | 33.429 | 32.532 | 28.935 | 28.824 | 28.594 | 28.342 |

由于随着量化参数的增大,本算法运算量有较大的下降,即跳过的零块个数增加,因此使 $PSNR$ 这样的客观图象质量度量标准必然也有一定的下降.3 个序列中 Miss 序列的 $PSNR$ 下降也就最明显,最坏的情况下达到下降 1dB 左右.其它两个图象序列 $PSNR$ 值的改变没有 Miss America 那样显著,这两个序列图象 $PSNR$ 的改变最大时只有 0.2dB 左右.并且这种最大的差别产生于量化参数较大的情况,而在这种情况下,0.2dB 的差别是较小的.下面以 Miss 序列为例,来分析由本算法引起的 $PSNR$ 改变,对实际图象质量产生的影响.对 Miss 序列来说, $PSNR$ 值的下降是由于较大比例的零块跳过引起的(例如,当量化参数为 10 时,有 60% 左右的宏块被跳过,表 2).通过选取量化参数为 10 时

$PSNR$ 相差最大的第 27 帧作为研究对象来进行对比研究.通过对比发现此帧用本算法与常规算法得到的重建图象的 $PSNR$ 值相差约 1.2dB.图 2 表示了这帧图象用本算法所跳过的零块原始图象的掩膜,白线框以内的是没有跳过的块.



图 2 跳过零块在原始图象中的掩膜

从图 2 可以看出那些没有跳过的块,是由于运动较大,其产生的失真值也始终比较大,从而使常规的运动搜索能进行到底.经过计算两帧图象掩膜以内区域的 $PSNR$,用本算法重建图象的 $PSNR$ 值还略高于用常规算法重建图象上这些区域的 $PSNR$ 值(约 0.1dB 左右).也就是说,人物所在区域的图象质量并没有下降,由此可见,本算法能保证运动较大区域或者图象变化剧烈区域的图象质量.因此,虽然重建图象 $PSNR$ 值有较大改变,但引起 $PSNR$ 下降的区域是背景区域.

图 3、图 4 分别是用本文搜索算法和常规搜索算法重建的该帧图象.比较图 3、图 4 可看出,人们



图 3 本算法重建图象



图 4 常规算法重建图象

所关心区域的图象质量并没有显著的改变. 加上实时系统中, 图象播放时的时间掩蔽效应 (Time Masking Effect), 本算法在实际应用中将不会真正引起图象质量的恶化.

4 结 论

本文提出了一种运用于低码率视频编码中的基于全零块发现的运动搜索算法. 针对可视电话中运动很小或者变化较小的图象部分用常规编码方法效率低下的缺点, 本算法把全零块发现准则应用于运动搜索, 从而大大加快了对这些图象部分的编码速度. 本算法也能够较大幅度地减少 H. 263 图象编码中最耗时间的运动搜索和 DCT 等模块的时间, 且在量化参数为 12 典型值的情况下, 平均搜索次数可以减少 30%~50%, 而跳过 DCT 等模块的运算也占到 25%~30%. 同时, 本算法对 H. 263 算法压缩效率影响很小, 并较好地保持了运动部分的图象质量. 本算法实际上是以略微牺牲不太重要的背景图象的质量为代价, 来换取图象编码速度显著提高的.

参 考 文 献

- 1 ITU-T H. 324 Recommendation. Terminal for low bit rate multimedia communication. 1998, 12.
- 2 ITU-T H. 263 Recommendation. Video coding for low bit rate communication. 1998, 01.
- 3 何芸, 翁成坚. 低数码率视频压缩编码及全软件 H. 263 视频通信系统研究. 中国图象图形学报, 1997, 2(5): 319~324.
- 4 Alice Yu, Ruby Lee, Michale Flynn. Early detection of all zero coefficients in H. 263. PCS 1997: 159~164.
- 5 ITU-T Study Group 16. Intelligent pre-quantization in motion compensated video coding. Q15-D-35. Tampere, Finland: ITU-T 1998, 04.
- 6 姚庆栋, 毕厚杰, 王兆华等. 图象编码基础. 杭州: 浙江大学出版社, 1993.
- 7 章毓晋. 图象处理和分析. 北京: 清华大学出版社, 1999.
- 8 ITU-T Study Group 16. TMN8 video codec test model near-term version 8[R]. Q15-B-46. Sunriver: ITU-T. 1997, 09.



兰军强 1997年毕业于清华大学电子工程系, 现为该系博士研究生. 本科和研究生期间参加过多项图象编码和多媒体通信领域科研项目的工作. 现在的主要研究方向为图象编码和多媒体通信.



何芸 1955年生, 清华大学电子工程系副教授, 1989年获比利时列日大学工学院 Montefiore 电学院博士学位. 目前的研究领域包括视频编码和视频通信系统、视觉信息表述、基于无线 IP 的视频信号编码和通信系统.



曹志刚 清华大学电子工程系教授, 博士生导师, 微波与数字通信国家重点实验室副主任兼学术委员会副主任. 目前主要研究领域包括通信理论、卫星通信、宽带移动通信、信号处理和远程教育等. 合著(译)专著 5 部, 发表论文 120 多篇.