



(12) 发明专利申请

(10) 申请公布号 CN 101917627 A

(43) 申请公布日 2010.12.15

(21) 申请号 201010262087.1

(22) 申请日 2010.08.20

(71) 申请人 浙江大学

地址 310027 浙江省杭州市西湖区浙大路  
38号

(72) 发明人 陈耀武 周怡然 蒋炜

(74) 专利代理机构 杭州天勤知识产权代理有限公司 33224

代理人 胡红娟

(51) Int. Cl.

H04N 7/64 (2006.01)

H04N 7/26 (2006.01)

H04N 7/32 (2006.01)

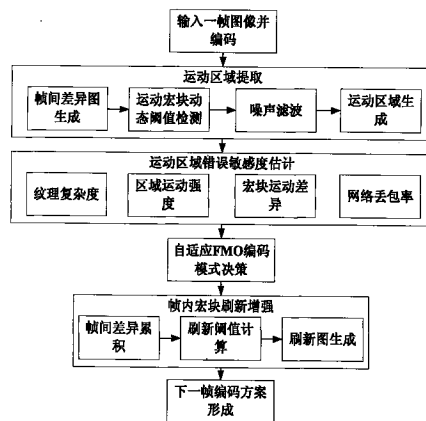
权利要求书 2 页 说明书 8 页 附图 4 页

(54) 发明名称

一种基于自适应灵活宏块次序的视频容错编码方法

(57) 摘要

本发明公开了一种基于自适应灵活宏块次序的视频容错编码方法,包括:输入一帧图像并编码;提取图像的运动区域,估计运动区域的错误敏感度,决策最优的灵活宏块次序编码模式;计算所有宏块位置的累积帧间差异,当运动区域内宏块位置的累积帧间差异大于指定的刷新阈值时,所述的宏块位置在下一帧时被强制帧内刷新,由此计算得到运动区域的帧内刷新宏块的数量与位置;根据得到的灵活宏块次序编码模式和宏块刷新的计算结果,形成下一帧图像的编码方案;重复上述步骤,直至编码全部结束。本发明的视频容错编码方法既能够满足差异化的网络传输要求,又能够充分发挥 FMO 本身的容错功能,实现较好的容错效果,且计算资源消耗少。



1. 一种基于自适应灵活宏块次序的视频容错编码方法,其特征在于,包括以下步骤:

(1) 输入一帧图像并编码;

(2) 提取所述的图像的运动区域,估计运动区域的错误敏感度;依据运动区域的错误敏感度,决策最优的灵活宏块次序编码模式;

(3) 计算所有宏块位置的累积帧间差异,当运动区域内宏块位置的累积帧间差异大于指定的刷新阈值时,所述的宏块位置在下一帧时被强制帧内刷新,由此计算得到运动区域的帧内刷新宏块的数量与位置;

(4) 根据步骤(2)得到的灵活宏块次序编码模式和步骤(3)得到的帧内刷新宏块的数量与位置,形成下一帧图像的编码方案;

(5) 重复步骤(2)~(4),直至编码全部结束。

2. 根据权利要求1所述的视频容错编码方法,其特征在于,所述的步骤(2)中,所述的提取所述的图像的运动区域,包括:

(2.1) 利用运动搜索产生的帧间差异信息对宏块进行初始运动检测,输入帧和参考帧之间差异大于运动检测阈值的宏块视为初始运动宏块;

(2.2) 分析宏块的编码比特值,对所述的初始运动宏块进行滤波,确定运动宏块,并确定涵盖所有运动宏块的最小的矩形为运动区域。

3. 根据权利要求2所述的视频容错编码方法,其特征在于,所述的步骤(2.1)中,所述的运动检测阈值随着不同序列的特征和同一序列不同帧的特征而进行动态调整。

4. 根据权利要求2所述的视频容错编码方法,其特征在于,所述的步骤(2.2)中,对所述的初始运动宏块进行滤波的过程如下:

计算每个初始运动宏块周边的四个宏块中初始运动宏块的数量  $N_{neb}$ ;

对于  $N_{neb}$  值为 0 或 1 的初始运动宏块,其最终的运动标识  $M_{final}$  定义为:

$$M_{final}(x, y, n) = \begin{cases} 1, & \text{if } B_{mb}(x, y, n) \geq [\lambda_{filter} \times B_{avg}(n)] \\ 0, & \text{otherwise} \end{cases} \quad (III)$$

式(III)中,  $B_{mb}$  表示一个宏块的编码比特值,  $B_{avg}$  表示所有  $N_{neb}$  大于 1 的初始运动宏块编码比特值的平均值;  $\lambda_{filter}$  为固定值参数,对于  $N_{neb}$  为 0 和 1 的初始运动宏块,  $\lambda_{filter}$  分别设为 1.4 和 0.7;当宏块的编码比特值大于滤波阈值  $[\lambda_{filter} \times B_{avg}(n)]$  时,最终运动标识  $M_{final}$  为 1,为运动宏块;反之,最终运动标识  $M_{final}$  为 0,为非运动宏块;

对于非初始运动宏块和  $N_{neb}$  值大于 1 的其它初始运动宏块,  $M_{final}$  和  $M_{initial}$  的值相等;

由此,得到所有宏块的  $M_{final}$  值,所有  $M_{final}$  值为 1 的宏块为运动宏块,涵盖所有运动宏块的最小的矩形为运动区域。

5. 根据权利要求1所述的视频容错编码方法,其特征在于,所述的步骤(2)中,所述的运动区域的错误敏感度,采用如下式(IV)所示的计算式估计:

$$S_{MA}(n) = p(n) \times T(n) \times \sigma_{mb}(n) \times \sigma_{frame}(n) \quad (IV)$$

式(IV)中,  $p$  为码流的传输丢包率,  $T$  为运动区域的纹理复杂度,  $\sigma_{mb}$  为运动区域内宏块运动差异,  $\sigma_{frame}$  为区域运动强度。

6. 根据权利要求5所述的视频容错编码方法,其特征在于,所述的运动区域的纹理复杂度  $T$  为运动区域中所有宏块的纹理复杂度  $t_{mb}$  的平均值,而一个宏块的纹理复杂度  $t_{mb}$  计

算如下式 (V) 所示：

$$t_{mb}(x, y, n) = \frac{\sum_{i, j \in \Omega(x, y, n)} sad_{DC44}(i, j, x, y, n)}{N_{\Omega}(x, y, n)} \quad (V)$$

式 (V) 中,  $\Omega$  表示左边和上边的周边像素完备的  $4 \times 4$  块的集合,  $N_{\Omega}$  表示这个集合中  $4 \times 4$  块的数量, 坐标  $(i, j)$  表示一个  $4 \times 4$  块在其所在宏块中的位置,  $sad_{DC44}$  为一个  $4 \times 4$  块所有组成像素与左边和上边的周边像素均值的差异之和。

7. 根据权利要求 1 所述的视频容错编码方法, 其特征在于, 所述的步骤 (2) 中, 所述的依据运动区域的错误敏感度, 决策最优的灵活宏块次序编码模式的过程如下：

对非运动区域全部使用单片组编码；

对于运动区域, 当运动区域的错误敏感度  $S_{MA}$  大于设定的敏感度阈值  $S_{TH}$  时, 采用两个片组的棋盘模式编码；当运动区域的错误敏感度  $S_{MA}$  小于设定的敏感度阈值  $S_{TH}$  时, 采用一个片组的平坦模式编码。

8. 根据权利要求 1 所述的视频容错编码方法, 其特征在于, 所述的步骤 (3) 中, 所述的宏块位置的累积帧间差异由下式 (VI) 得到：

$$D_{acc}(x, y, n) = \sum_{t=n_0}^n p(t) \times sad_{mv00}(x, y, t) \quad (VI)$$

式 (VI) 中,

$n_0$  表示宏块位置  $(x, y)$  前次进行帧内预测编码时的图像序列号,  $n_0$  小于或等于  $n$ ;  $p$  为码流的传输丢包率；

所述的刷新阈值为：

$$D_{TH}(n) = \frac{C_{IR}}{\sigma_{frame}(n)} \quad (VII)$$

式 (VII) 中,  $C_{IR}$  为固定值参数,  $\sigma_{frame}$  为区域运动强度。

## 一种基于自适应灵活宏块次序的视频容错编码方法

### 技术领域

[0001] 本发明涉及视频编解码领域,具体涉及一种基于自适应灵活宏块次序(FMO)的视频容错编码方法。

### 背景技术

[0002] 随着计算机技术与网络通信技术的快速发展,简单的文本信息形式已不能满足人们的需求,更为丰富的视频信息已经逐渐走进人们日常生活的各个方面。有限的传输带宽和存储空间必然要求对数据量巨大的原始视频进行压缩,因此视频编解码技术与标准已成为多媒体应用中不可或缺的组成部分。现有的视频编解码标准非常依赖于码流的完整性。一旦在传输过程中出现错误就会导致视频解码后的观看质量下降,影响用户感受。因此,在视频编码过程中增强码流抵御错误的能力,即改善码流的容错性,是十分必要的。

[0003] 灵活宏块次序FMO(Flexible Macroblock Ordering)是H.264/AVC标准采纳的一种非常有效的新容错机制。宏块不再必须是严格依据其在图像中的光栅扫描次序被分配入片中。通过宏块分配图(Macroblock allocation map, MBAmap),一个宏块可以被分配入任何一个片组中去。FMO机制可以用在视频容错中的原因主要有两个方面。第一个方面是其非常适合于存在差别化传输保护的的网络环境。由于可以对宏块进行任意形式的片组分配,因此重要或脆弱的区域就能够被分配在传输相对于可靠的片组中,增强了保护的针对性,提高了保护的效率。另一方面,FMO本身也具备较强的容错功能。因为丢失的宏块可能会被其周边其它没有丢失的宏块所包围,因此利用周边宏块的像素或信息可以有效恢复丢失的宏块,十分有利于解码器的错误恢复算法。

[0004] 现有利用FMO进行容错编码的方法虽然取得了一定的效果,但仍然存在的问题主要包括:没有考虑传输特性;需要进行第二次编码或采用率失真优化算法,计算量和结构不适合于实际应用;只能在离线编码中进行非实时操作;在关注差别化传输的同时,忽视了发挥FMO本身的容错功能,或者相反。

### 发明内容

[0005] 本发明提供了一种基于自适应灵活宏块次序(FMO)的视频容错编码方法,既能够满足差异化的网络传输要求,又能够充分发挥FMO本身的容错功能,实现有效的视频容错;同时,由于主要计算信息来自于编码域,即编码过程中产生的信息,因此计算效率较高。

[0006] 一种基于自适应灵活宏块次序(FMO)的视频容错编码方法,包括以下步骤:

[0007] (1) 输入一帧图像并编码;

[0008] (2) 提取所述的图像的运动区域,估计运动区域的错误敏感度;依据运动区域的错误敏感度,决策最优的灵活宏块次序编码模式;

[0009] (3) 计算所有宏块位置的累积帧间差异,当运动区域内宏块位置的累积帧间差异大于指定的刷新阈值时,所述的宏块位置在下一帧时被强制帧内刷新,由此计算得到运动区域的帧内刷新宏块的数量与位置;

[0010] (4) 根据步骤 (2) 得到的灵活宏块次序编码模式和步骤 (3) 得到的帧内刷新宏块的数量与位置, 形成下一帧图像的编码方案;

[0011] (5) 重复步骤 (2) ~ (4), 直至编码全部结束。

[0012] 步骤 (2) 中, 所述的提取所述的图像的运动区域, 包括:

[0013] (2.1) 利用运动搜索产生的帧间差异信息对宏块进行初始运动检测, 输入帧和参考帧之间的差异大于运动检测阈值的宏块视为初始运动宏块;

[0014] 第  $n$  帧图像中坐标为  $(x, y)$  的宏块的初始运动标识  $M_{\text{initial}}$  如式 (I) 所示:

$$[0015] \quad M_{\text{initial}}(x, y, n) = \begin{cases} 1, & \text{if } \text{sad}_{\text{mv00}}(x, y, n) > [\beta(n) \times \text{sad}_{\text{avg}}(n)] \\ 0, & \text{otherwise} \end{cases} \quad (\text{I})$$

[0016] 式 (I) 中,  $\text{sad}_{\text{mv00}}$  为  $16 \times 16$  运动搜索时获得的在零运动矢量处的 SAD (Sum of Absolute Difference) 编码信息, 反映宏块在输入帧和参考帧之间的差异;  $\text{sad}_{\text{avg}}$  表示第  $n$  帧图像所有宏块  $\text{sad}_{\text{mv00}}$  的平均值;  $\beta(n)$  为运动检测阈值参数,  $[\beta(n) \times \text{sad}_{\text{avg}}(n)]$  为运动检测阈值。

[0017] 当宏块的  $\text{sad}_{\text{mv00}}$  大于运动检测阈值时, 初始运动标识  $M_{\text{initial}}$  为 1, 宏块视为初始运动宏块; 反之, 初始运动标识  $M_{\text{initial}}$  为 0, 宏块视为非初始运动宏块。

[0018] 进一步, 所述的运动检测阈值随着不同序列的特征和同一序列不同帧的特征进行动态调整, 其中, 运动检测阈值参数  $\beta(n)$  如式 (II) 所示:

[0019]

$$\beta(n) = \begin{cases} \beta_H, & \sigma_{\text{frame}}(n) > \sigma_{\text{MA}_H} \\ k \times \left[ \frac{\sigma_{\text{frame}}(n) - \sigma_{\text{MA}_L}}{\sigma_0} \right] + \beta_L, & \sigma_{\text{frame}}(n) \in (\sigma_{\text{MA}_L}, \sigma_{\text{MA}_H}] \\ \beta_L, & \sigma_{\text{frame}}(n) \leq \sigma_{\text{MA}_L} \end{cases} \quad (\text{II})$$

[0020] 式 (II) 中,  $\sigma_{\text{frame}}$  表示第  $n$  帧图像所有宏块  $\text{sad}_{\text{mv00}}$  的标准差;  $k$ 、 $\sigma_0$ 、 $\beta_H$ 、 $\beta_L$ 、 $\sigma_{\text{MA}_H}$  和  $\sigma_{\text{MA}_L}$  均为固定值参数;

[0021] (2.2) 分析宏块的编码比特值, 对所述的初始运动宏块进行滤波, 确定运动宏块, 并确定涵盖所有运动宏块的最小的矩形为运动区域;

[0022] 计算每个初始运动宏块周边的四个宏块中初始运动宏块的数量  $N_{\text{neb}}$ ,

[0023] 对于  $N_{\text{neb}}$  值为 0 或 1 的初始运动宏块,

[0024] 其最终运动标识  $M_{\text{final}}$  为:

$$[0025] \quad M_{\text{final}}(x, y, n) = \begin{cases} 1, & \text{if } B_{\text{mb}}(x, y, n) \geq [\lambda_{\text{filter}} \times B_{\text{avg}}(n)] \\ 0, & \text{otherwise} \end{cases} \quad (\text{III})$$

[0026] 式 (III) 中,  $B_{\text{mb}}$  表示一个宏块的编码比特值,  $B_{\text{avg}}$  表示所有  $N_{\text{neb}}$  大于 1 的初始运动宏块编码比特值的平均值;  $\lambda_{\text{filter}}$  为固定值参数, 对于  $N_{\text{neb}}$  为 0 和 1 的初始运动宏块,  $\lambda_{\text{filter}}$  分别设为 1.4 和 0.7。

[0027] 当宏块的编码比特值大于滤波阈值  $[\lambda_{\text{filter}} \times B_{\text{avg}}(n)]$  时, 最终运动标识  $M_{\text{final}}$  为 1, 为运动宏块; 反之, 最终运动标识  $M_{\text{final}}$  为 0, 为非运动宏块。

[0028] 对于非初始运动宏块和  $N_{\text{neb}}$  值大于 1 的其它初始运动宏块,

[0029]  $M_{\text{final}} = M_{\text{initial}}$

[0030] 由此,得到所有宏块的 $M_{\text{final}}$ 值,所有最终运动标识 $M_{\text{final}}$ 值为1的宏块为运动宏块,涵盖所有运动宏块的最小的矩形为运动区域。

[0031] 步骤(2)中,所述的运动区域的错误敏感度,是指运动区域在解码器中进行错误恢复的难度,采用如下式(IV)所示的计算式估计:

$$[0032] \quad S_{\text{MA}}(n) = p(n) \times T(n) \times \sigma_{\text{mb}}(n) \times \sigma_{\text{frame}}(n) \quad (\text{IV})$$

[0033] 式(IV)中, $p$ 为码流的传输丢包率,较严重的传输丢包会导致较大的错误恢复难度; $T$ 为运动区域的纹理复杂度,复杂的纹理会加大错误恢复的难度; $\sigma_{\text{mb}}$ 为运动区域内宏块运动差异,宏块间的运动情况差异越大,错误恢复的难度也会越大; $\sigma_{\text{frame}}$ 为区域运动强度,运动区域的总体运动强度相对于背景越大,错误恢复的难度也会越大(区域运动强度 $\sigma_{\text{frame}}$ 与式(II)中 $\sigma_{\text{frame}}$ 相同,可以利用之前对第 $n$ 帧图像所有宏块 $\text{sad}_{\text{mv}00}$ 的标准差的计算结果)。上述分析因素可以适应大多数错误恢复方法的算法特征。式(IV)中结果越大,说明运动区域的错误敏感度越高,即错误恢复难度越大。

[0034] 所述的运动区域的纹理复杂度 $T$ 为运动区域中所有宏块的纹理复杂度 $t_{\text{mb}}$ 的平均值,而一个宏块的纹理复杂度 $t_{\text{mb}}$ 计算如下式(V)所示:

$$[0035] \quad t_{\text{mb}}(x, y, n) = \frac{\sum_{i, j \in \Omega(x, y, n)} \text{sad}_{\text{DC44}}(i, j, x, y, n)}{N_{\Omega}(x, y, n)} \quad (\text{V})$$

[0036] 式(V)中, $\Omega$ 表示左边和上边的周边像素都完备的 $4 \times 4$ 块的集合, $N_{\Omega}$ 表示这个集合中 $4 \times 4$ 块的数量,坐标 $(i, j)$ 表示一个 $4 \times 4$ 块在其所在宏块中的位置, $\text{sad}_{\text{DC44}}$ 为模式决策时产生的帧内 $4 \times 4$ 直流模式的SAD信息,为一个 $4 \times 4$ 块所有组成像素与周边像素(左边和上边的像素)均值的差异之和,反映该 $4 \times 4$ 块内部各个像素之间的差异性,差异越大纹理越复杂。

[0037] 步骤(2)中,所述的依据运动区域的错误敏感度,决策最优的灵活宏块次序编码模式的过程如下:

[0038] 对非运动区域全部使用单片组编码;

[0039] 对于运动区域,当运动区域的错误敏感度 $S_{\text{MA}}$ 大于设定的敏感度阈值 $S_{\text{TH}}$ 时,采用两个片组的棋盘模式编码;当运动区域的错误敏感度 $S_{\text{MA}}$ 小于设定的敏感度阈值 $S_{\text{TH}}$ 时,采用一个片组的平坦模式编码。

[0040] 步骤(3)中,所述的宏块位置的累积帧间差异由下式(VI)得到:

$$[0041] \quad D_{\text{acc}}(x, y, n) = \sum_{t=n_0}^n p(t) \times \text{sad}_{\text{mv}00}(x, y, t) \quad (\text{VI})$$

[0042] 式(VI)中, $n_0$ 表示宏块位置 $(x, y)$ 前次进行帧内预测编码时的图像序列号, $n_0$ 小于或等于 $n$ ; $p$ 为码流的传输丢包率;

[0043] 步骤(3)中,所述的指定的刷新阈值 $D_{\text{TH}}(n)$ 为变值,由下式(VII)计算:

$$[0044] \quad D_{\text{TH}}(n) = \frac{C_{\text{IR}}}{\sigma_{\text{frame}}(n)} \quad (\text{VII})$$

[0045] 式(VII)中, $C_{\text{TR}}$ 为固定值参数, $\sigma_{\text{frame}}$ 为区域运动强度。对于运动强度大的帧给予更多的刷新数量,以取得更好的自适应性。

[0046] 本发明中,对于第一帧图像不采用 FMO 进行编码,从第二帧开始,均是通过对前一帧编码域信息的分析来决定下一帧的容错编码模式。主要计算信息都来自于编码域,也就是编码过程中所产生的信息,避免了多次编码,有效节省了计算资源。

[0047] 本发明中,在提取运动区域时,采取  $16 \times 16$  运动搜索时在零运动矢量处的编码信息 SAD ( $sad_{mv00}$ ) 进行初步运动检测,由于  $sad_{mv00}$  能够反映输入帧和参考帧之间的差异且可以直接从大多数搜索算法中获得,因此可以高效率地检测运动区域;在进行了初步运动检测后,还采用宏块编码比特值对初始运动宏块进行滤波,剔除初始运动宏块中的噪声宏块,除去残余的噪声。由于噪声宏块往往处于孤立位置,因此,通过分析孤立宏块的编码比特值相对于主要运动区域的编码比特值大小,来确定噪声宏块。剔除噪声宏块后的初始运动宏块被确定为运动宏块。本发明是以视频监控场景为基础,将其最核心的运动区域(无论是人还是物体)作为优先传输和重点保护的对象,并确定涵盖所有运动宏块的最小的矩形为运动区域。

[0048] 本发明中,依据码流的传输丢包率  $p$ 、运动区域的纹理复杂度  $T$ 、运动区域内宏块运动差异  $\sigma_{mb}$  和区域运动强度  $\sigma_{frame}$  等多个因素评估运动区域的错误敏感度,能适应大多数错误恢复方法的算法特征。此外,在纹理信息的获取时,利用模式决策时产生的帧内  $4 \times 4$  直流模式的 SAD 信息 ( $sad_{dc44}$ ),提高了计算效率;而且,运动区域的纹理复杂度  $T$  和运动区域内宏块运动差异  $\sigma_{mb}$  的计算都是限定在运动区域中,降低了计算的复杂度。

[0049] 本发明中,最优的 FMO 编码模式的选择充分考虑到运动与非运动区域的区别、以及不同错误敏感度的运动区域的区别。在各种标准 FMO 类型中,两片组的类型 2 无疑是非常适合区分运动(前景)与非运动区域(背景)的一种类型,特别适合于对图像不同区域的差异化传输保护。然而,类型 2 在一些情况下对运动区域的保护并不足够,因为它只采用了一种平坦式的编码方式,也就是单片组编码运动区域。但是该类型具有较好的压缩效率;相反,当采用棋盘式的类型 1 时,容错能力会显著增强。但是因为相邻宏块之间的信息共享性显著降低,类型 1 的压缩效率较差。因此,本发明采用了对背景全部使用单片组以提高压缩效率。这一方面是因为背景区域并不是关注的重点区域,另一方面是因为少动的背景相对来说并不容易受到传输错误的影响。而为了能在压缩效率和容错能力之间取得平衡,依据前景的错误敏感度,用一个或两个片组对前景进行编码。本发明中 FMO 机制可以给解码器提供一个友好的宏块分布结构,因此能够减轻错误蔓延。但是,FMO 机制本身并不能完全终止这种错误蔓延,因此,本发明进一步利用帧内宏块刷新机制去增强其容错性,合理计算刷新的位置和数量。

[0050] 综上所述,本发明的基于自适应灵活宏块次序(FMO)的视频容错编码方法,既能够满足差异化的网络传输要求,又能够充分发挥 FMO 本身的容错功能,实现有效的视频容错;同时,由于主要计算信息来自于编码域,即编码过程中产生的信息,因此计算效率较高。

## 附图说明

- [0051] 图 1 为本发明的流程图;
- [0052] 图 2 为提取图像的运动区域的流程示意图;
- [0053] 图 3 为自适应 FMO 编码模式决策示意图;
- [0054] 图 4 为 Salesman 序列的 PSNR 结果;

[0055] 图 5 为 Container 序列的 PSNR 结果；

[0056] 图 6 为 Hall 序列的 PSNR 结果。

### 具体实施方式

[0057] 下面结合实施例和附图来详细说明本发明,但本发明并不仅限于此。

[0058] 如图 1 所示,一种基于自适应灵活宏块次序 (FMO) 的视频容错编码方法,包括以下步骤:

[0059] (1) 输入一帧图像并编码;

[0060] 对于第一帧图像不采用 FMO 进行编码,而是采用正常编码方式。

[0061] (2) 提取所述的图像的运动区域,估计运动区域的错误敏感度;并依据运动区域的错误敏感度,决策最优的灵活宏块次序编码模式;

[0062] 其中,所述的图像的运动区域的提取,包括:

[0063] (2.1) 利用运动搜索产生的帧间差异信息对宏块进行初始运动检测,输入帧和参考帧之间的差异大于运动检测阈值的宏块视为初始运动宏块;第  $n$  帧图像中坐标为  $(x, y)$  的宏块的初始运动标识  $M_{initial}$  如式 (I) 所示:

$$[0064] \quad M_{initial}(x, y, n) = \begin{cases} 1, & \text{if } sad_{mv00}(x, y, n) > [\beta(n) \times sad_{avg}(n)] \\ 0, & \text{otherwise} \end{cases} \quad (I)$$

[0065] 式 (I) 中,  $sad_{mv00}$  为  $16 \times 16$  运动搜索时在零运动矢量处的编码信息,反映输入帧和参考帧之间的差异,可以直接从大多数搜索算法中获得;  $sad_{avg}$  表示第  $n$  帧图像所有宏块  $sad_{mv00}$  的平均值;  $\beta(n)$  为运动检测阈值参数,  $[\beta(n) \times sad_{avg}(n)]$  为运动检测阈值。

[0066] 当宏块的  $sad_{mv00}$  大于运动检测阈值时,初始运动标识  $M_{initial}$  为 1,宏块视为初始运动宏块;反之,初始运动标识  $M_{initial}$  为 0,宏块视为非初始运动宏块。

[0067] 进一步,所述的运动检测阈值随着不同序列的特征和同一序列不同帧的特征进行动态调整,其中,运动检测阈值参数  $\beta(n)$  如式 (II) 所示:

[0068]

$$\beta(n) = \begin{cases} \beta_H, & \sigma_{frame}(n) > \sigma_{MA\_H} \\ k \times \left[ \frac{\sigma_{frame}(n) - \sigma_{MA\_L}}{\sigma_0} \right] + \beta_L, & \sigma_{frame}(n) \in (\sigma_{MA\_L}, \sigma_{MA\_H}] \\ \beta_L, & \sigma_{frame}(n) \leq \sigma_{MA\_L} \end{cases} \quad (II)$$

[0069] 式 (II) 中,  $\sigma_{frame}$  表示第  $n$  帧图像所有宏块  $sad_{mv00}$  的标准差;  $k$ 、 $\sigma_0$ 、 $\beta_H$ 、 $\beta_L$ 、 $\sigma_{MA\_H}$  和  $\sigma_{MA\_L}$  均为固定值参数;

[0070] (2.2) 在进行了初步运动检测后,还要对残余的噪声进行滤波,将一些初始运动宏块剔除出去,因此,利用宏块的编码比特值,对初始运动宏块进行滤波,确定运动宏块,并确定涵盖所有运动宏块的最小的矩形为运动区域:

[0071] 噪声宏块往往处于孤立位置,因此,先计算每个初始运动宏块周边的四个宏块中初始运动宏块的数量  $N_{neb}$ ,找到处于孤立位置的初始运动宏块为孤立宏块,也即:  $N_{neb}$  值为 0 或 1 的初始运动宏块;进一步,如果孤立宏块相对于主要运动区域的编码比特值不够大,孤立宏块就将被认定为噪声宏块。即:



[0072] 对于  $N_{neb}$  值为 0 或 1 的初始运动宏块,

[0073] 最终运动标识  $M_{final}$  为:

$$[0074] \quad M_{final}(x, y, n) = \begin{cases} 1, & \text{if } B_{mb}(x, y, n) \geq [\lambda_{filter} \times B_{avg}(n)] \\ 0, & \text{otherwise} \end{cases} \quad (III)$$

[0075] 式 (III) 中,  $B_{mb}$  表示一个宏块的编码比特值,  $B_{avg}$  表示所有  $N_{neb}$  大于 1 的初始运动宏块编码比特值的平均值;  $\lambda_{filter}$  为固定值参数, 对于  $N_{neb}$  为 0 和 1 的初始运动宏块,  $\lambda_{filter}$  分别设为 1.4 和 0.7。

[0076] 当宏块的编码比特值大于滤波阈值  $[\lambda_{filter} \times B_{avg}(n)]$  时, 最终运动标识  $M_{final}$  为 1, 为运动宏块; 反之, 最终运动标识  $M_{final}$  为 0, 为噪声宏块 (非运动宏块)。

[0077] 对于非初始运动宏块和  $N_{neb}$  值大于 1 的其它初始运动宏块,

$$[0078] \quad M_{final} = M_{initial}$$

[0079] 由此, 得到所有宏块的  $M_{final}$  值, 所有最终运动标识  $M_{final}$  值为 1 的宏块为运动宏块, 涵盖所有运动宏块的最小的矩形定义为运动区域。

[0080] 上述步骤 (2.1) 和 (2.2) 为运动区域提取的全部过程, 如图 2 所示。在本实施例中, 以视频监控场景为基础, 将其最核心的运动区域 (无论是人还是物体) 作为优先传输和重点保护的對象, 最终只有一个运动区域被提取出来并作为前景, 所有其它宏块属于背景, 即运动区域与前景具有完全相同的含义。

[0081] 步骤 (2) 中, 所述的运动区域的错误敏感度, 是指运动区域在解码器中进行错误恢复的难度, 采用如下式 (IV) 所示的计算式估计:

$$[0082] \quad S_{MA}(n) = p(n) \times T(n) \times \sigma_{mb}(n) \times \sigma_{frame}(n) \quad (IV)$$

[0083] 式 (IV) 中,  $p$  为码流的传输丢包率, 较严重的传输丢包会导致较大的错误恢复难度;  $T$  为运动区域的纹理复杂度, 复杂的纹理会加大错误恢复的难度;  $\sigma_{mb}$  为运动区域内宏块运动差异, 宏块间的运动情况差异越大, 错误恢复的难度也会越大;  $\sigma_{frame}$  为区域运动强度, 运动区域的总体运动强度相对于背景越大, 错误恢复的难度也会越大 (区域运动强度  $\sigma_{frame}$  与式 (II) 中  $\sigma_{frame}$  相同, 可以利用之前对第  $n$  帧图像所有宏块  $sad_{mv00}$  的标准差的计算结果)。上述分析因素可以适应大多数错误恢复方法的算法特征。式 (IV) 中结果越大, 说明运动区域的错误敏感度越高, 即错误恢复难度越大。

[0084] 所述的运动区域的纹理复杂度  $T$  为运动区域中所有宏块的纹理复杂度  $t_{mb}$  的平均值, 而一个宏块的纹理复杂度  $t_{mb}$  计算如下式 (V) 所示:

$$[0085] \quad t_{mb}(x, y, n) = \frac{\sum_{i, j \in \Omega(x, y, n)} sad_{DC44}(i, j, x, y, n)}{N_{\Omega}(x, y, n)} \quad (V)$$

[0086] 式 (V) 中,  $\Omega$  表示左边和上边像素都完备的  $4 \times 4$  块的集合, 不考虑左边或上边像素不可获得的少数划分块, 如处在图像或片的边界的块;  $N_{\Omega}$  表示这个集合中  $4 \times 4$  块的数量, 坐标  $(i, j)$  表示一个  $4 \times 4$  块在其所在宏块中的位置,  $sad_{DC44}$  为模式决策时产生的帧内  $4 \times 4$  直流模式的 SAD 信息, 为一个  $4 \times 4$  块所有组成像素与周边像素 (左边和上边) 均值的差异之和, 反映该  $4 \times 4$  块内部各个像素之间的差异性, 差异越大纹理越复杂。类似于  $\sigma_{frame}$ ,  $\sigma_{mb}$  是一个标准差, 但是它的计算范围限定在运动区域而不是整个图像。  $T$  和  $\sigma_{mb}$  二

者的计算都是限定在运动区域中,以降低计算的复杂度。

[0087] 步骤(2)中,所述的依据运动区域的错误敏感度,决策最优的灵活宏块次序编码模式的过程如下:

[0088] 对非运动区域(背景)全部使用单片组编码以提高压缩效率,因为背景区域并不是关注的重点区域,而且少动的背景相对来说并不容易受到传输错误的影响;

[0089] 对于运动区域,依据运动区域(前景)的错误敏感度,用一个或两个片组对前景进行编码,以便在压缩效率和容错能力之间取得平衡。具体实现以FMO的类型6为基础。当运动区域(前景)的错误敏感度 $S_{MA}$ 大于设定的敏感度阈值 $S_{TH}$ 时,采用两个片组的棋盘模式编码;当运动区域的错误敏感度 $S_{MA}$ 小于设定的敏感度阈值 $S_{TH}$ 时,采用一个片组的平坦模式编码,此时与两片组的类型2的结构完全相同。这两种FMO编码模式如图3所示。图中的阿拉伯数字表示每个宏块所属的片组号,片组的边界用粗线条表示,粗线条内部为提取出的运动区域。

[0090] (3)计算所有宏块位置的累积帧间差异,当运动区域内宏块位置的累积帧间差异大于指定的刷新阈值时,所述的宏块位置在下一帧时被强制帧内刷新,由此计算得到运动区域的帧内刷新宏块的数量与位置;

[0091] 为不显著增加计算量,前述的 $sad_{mv00}$ 信息在此将继续被使用;同时,还需要考虑丢包率的因素,因为严重的丢包会导致较严重的错误蔓延。所述的宏块位置的累积帧间差异由下式(VI)得到:

$$[0092] \quad D_{acc}(x, y, n) = \sum_{t=n_0}^n p(t) \times sad_{mv00}(x, y, t) \quad (VI)$$

[0093] 式(VI)中,所述的指定的刷新阈值 $D_{TH}(n)$ 为变值,由 $n_0$ 表示宏块位置 $(x, y)$ 前次进行帧内预测编码时的图像序列号, $n_0$ 小于或等于 $n$ ;  $p$ 为码流的传输丢包率;

[0094] 所述的刷新阈值由式(VII)计算;

$$[0095] \quad D_{TH}(n) = \frac{C_{IR}}{\sigma_{frame}(n)} \quad (VII)$$

[0096] 式(VII)中, $C_{IR}$ 为固定值参数, $\sigma_{frame}$ 为区域运动强度。对于运动强度大的帧给予更多的刷新数量,以取得更好的自适应性。

[0097] 为了节省比特资源,帧内刷新宏块只从更为重要的运动区域中选择。帧内宏块刷新机制可以终止错误蔓延,具有较强的容错能力,可进一步增强FMO机制的容错效果。

[0098] (4)根据步骤(2)得到的灵活宏块次序编码模式和步骤(3)得到的宏块刷新的计算结果,形成下一帧图像的编码方案;

[0099] (5)重复步骤(2)~(4),直至编码全部结束。

[0100] 实验效果:采用本发明方法(AEFMO)对Salesman、Container和Hall三种标准视频序列进行容错编码。与H.264/AVC模型标准参考模型JM15.1中的FMO1方法(H.264/AVC标准中规定的FMO类型1)和FMO2方法(H.264/AVC标准中规定的FMO类型2)进行对比。实验丢包率为10%,固定值参数 $k$ 、 $\sigma_0$ 、 $\beta_H$ 、 $\beta_L$ 、 $\sigma_{MA_H}$ 、 $\sigma_{MA_L}$ 、 $S_{TH}$ 和 $C_{IR}$ 分别设为0.1、100、1.9、1.3、800、300、 $5 \times 10^5$ 和 $2 \times 10^5$ 。图4为Salesman序列的峰值信噪比(PSNR)实验结果,曲线1、2和3分别表示FMO1、FMO2和AEFMO三种方法的结果。图5为Container序列的峰值信噪比(PSNR)实验结果,曲线1、2和3分别表示FMO1、FMO2和AEFMO三种方法

的结果。图 6 为 Ha11 序列的峰值信噪比 (PSNR) 实验结果, 曲线 1、2 和 3 分别表示 FM01、FM02 和 AEFMO 三种方法的结果。从图 4 ~ 6 中可以看到, 本发明相较于 FM01 可以提高质量 0.03-3.18 分贝 (dB), 相较于 FM02 可以提高质量 0.23-5.10 分贝 (dB), 取得了较好的容错效果。

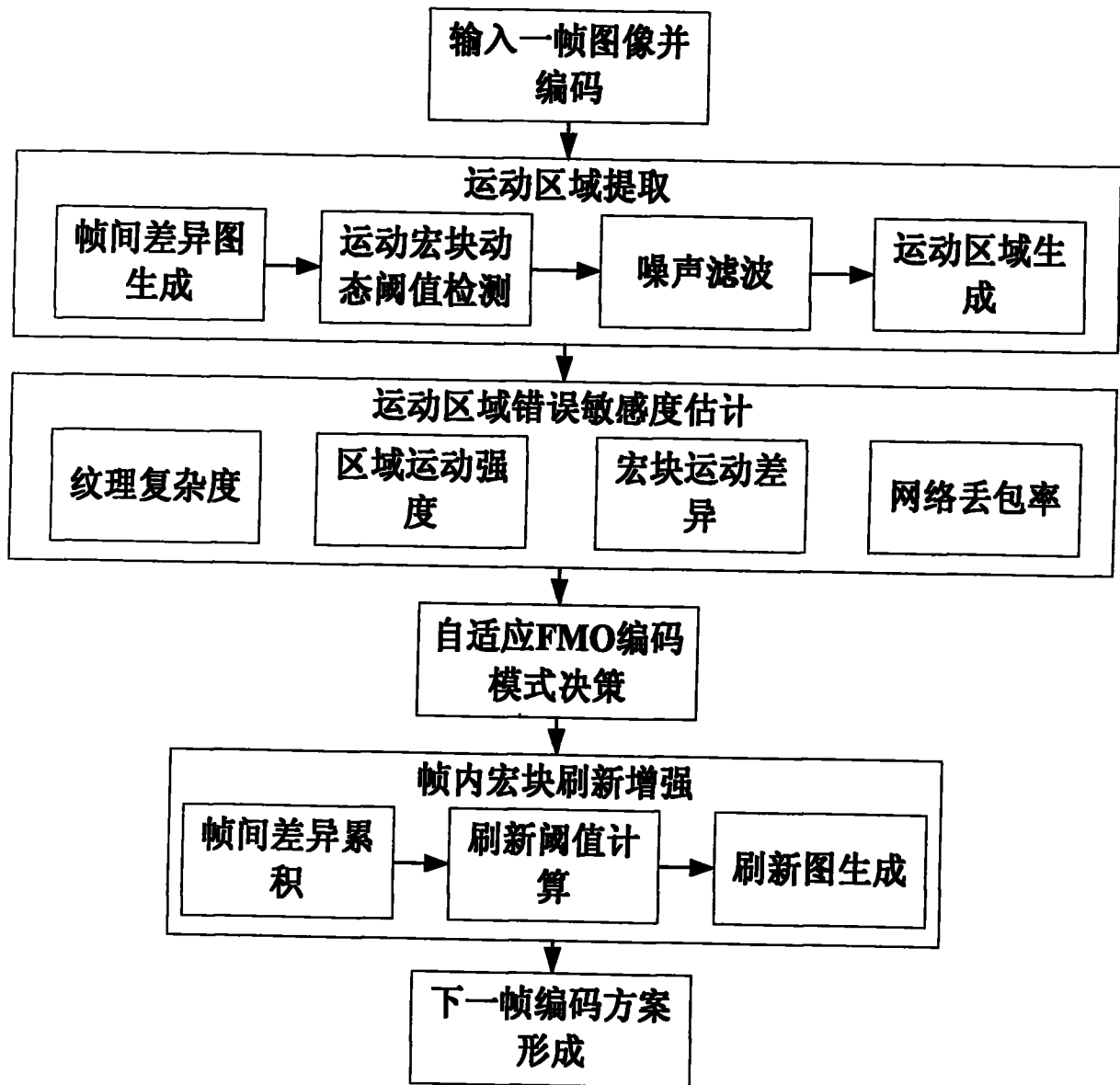


图 1

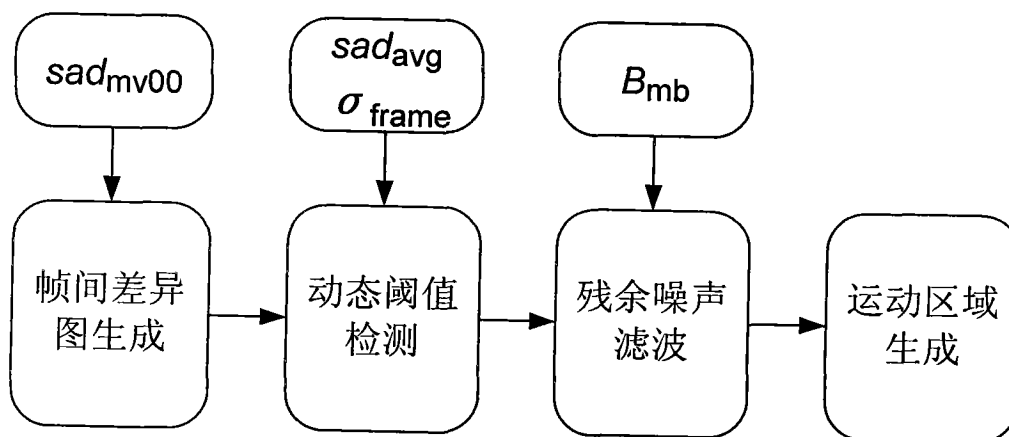


图 2

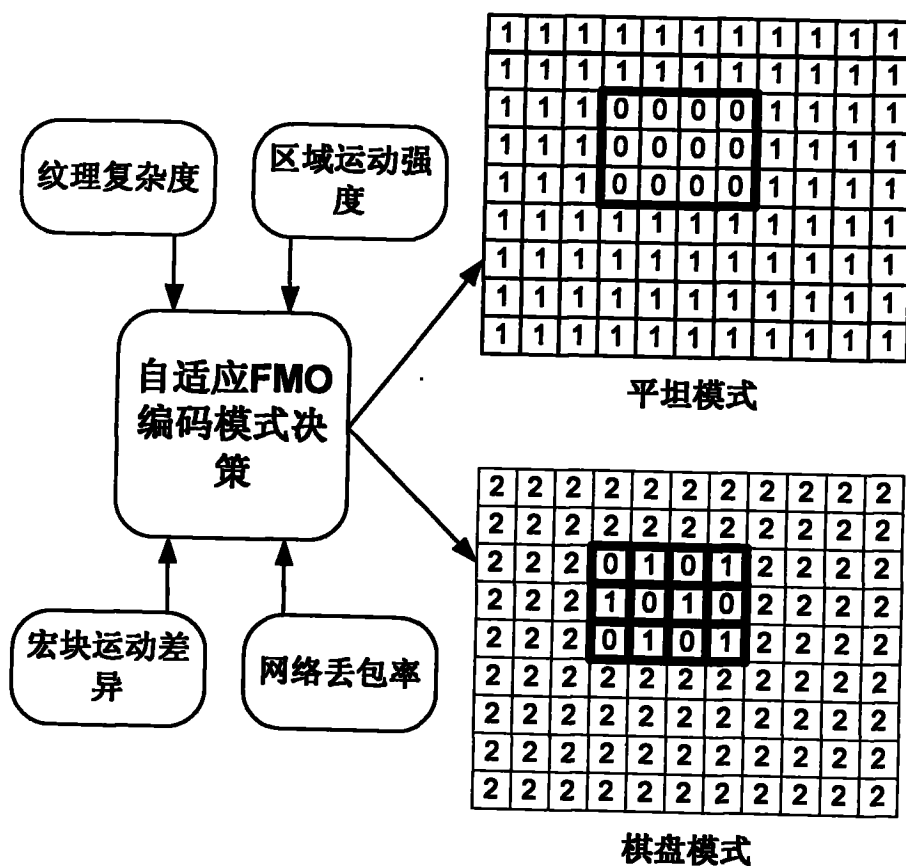


图 3

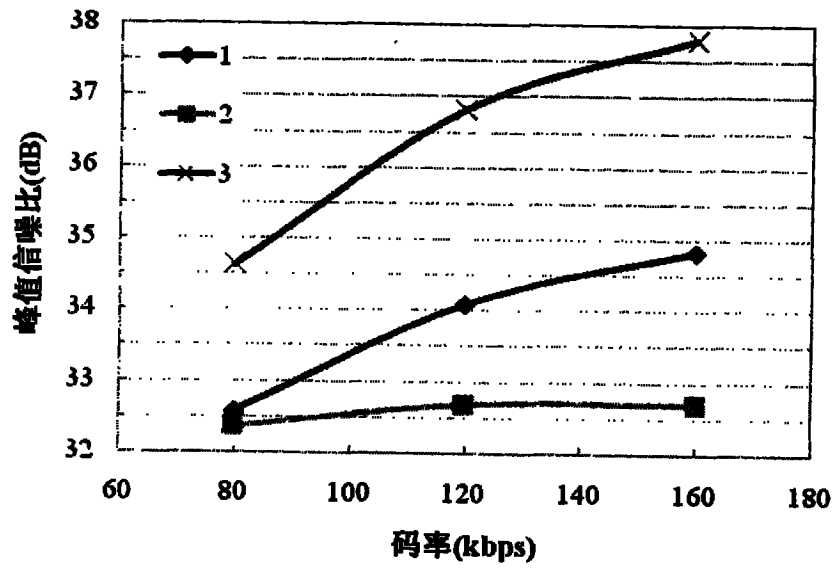


图 4

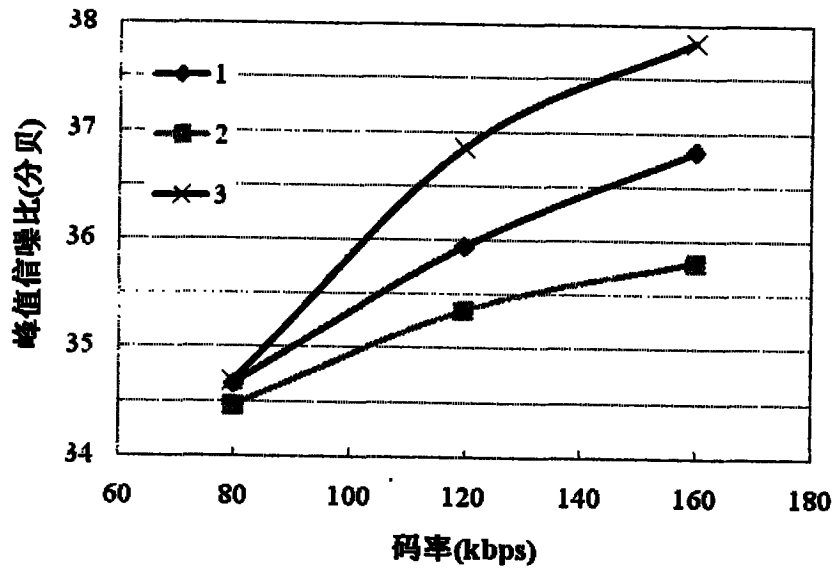


图 5

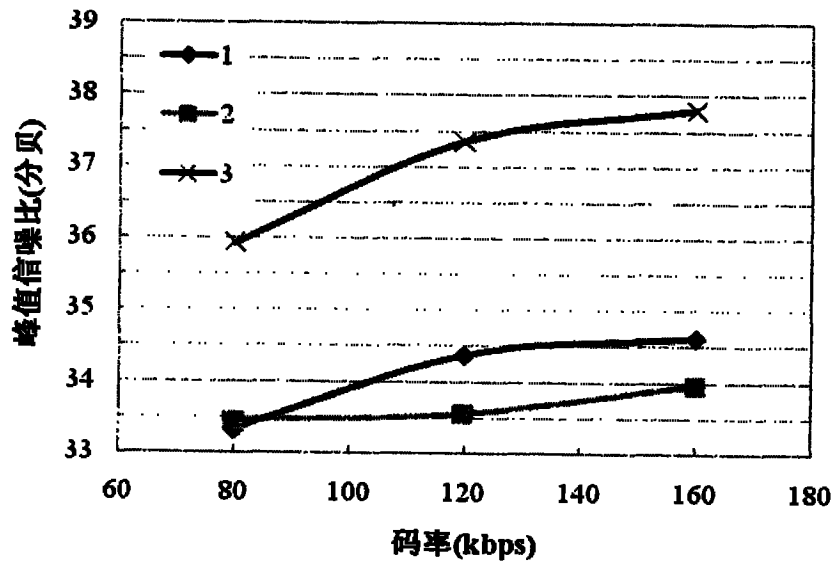


图 6