

MPEG-4 自然视频编码工具和算法

MPEG-4 Natural Video Coding Tools and Algorithms

王相海 张福炎

(南京大学计算机软件新技术国家重点实验室 南京大学计算机系 南京 210093)

Abstract Although video compression standards, MPEG-1 and MPEG-2, perfectly well suite in environments for which they were designed, they are not necessarily flexible enough to efficiently address the requirement of multimedia applications. Hence, MPEG (Moving Picture Experts Group) committed itself to the development of the MPEG-4 standard, providing a common platform for a wide range of multimedia applications. This paper summarizes the major natural video coding tools and algorithms, and explains figuratively some important concepts as defined in the MPEG-4.

Keywords MPEG-4, Natural video coding, VOP, Sprite coding, Scalability

1 引言

作为一个多媒体标准往往被期待着能对大量的应用提供很好的支持。MPEG-1^[1]和 MPEG-2^[2]这两个视频压缩标准以其严谨的风格、完备的特性以及强大的功能,被广泛地应用于计算机多媒体数据压缩的编码和解码系统中,对计算机多媒体数据编、解码技术的发展做出了巨大的贡献。然而,随着网络、有线和无线通信系统的迅猛发展,交互式计算机和交互电视技术的普遍应用,以及视频、音频数据综合服务等技术的发展趋势,对计算机多媒体数据压缩编、解码技术以及所遵循的标准提出了更高的要求。有许多要求是 MPEG-1 和 MPEG-2 标准难以支持的,这样 MPEG-4 便应运而生,它为多媒体数据编码提供了一个更为广阔的平台^[3]。ISO 的 MPEG-4 标准的第一版和第二版已分别于 1999 年 1 月和 12 月正式公布^[4,5]。本文系统地讨论了标准 ISO/IEC14496-2(MPEG-4)^[6]所提供的自然视频的编码工具和算法,同时对所涉及的一些重要概念进行形象化的解释。

本文讨论了与自然视频编码相关的一些重要概念,对每一概念都结合相应的图示给出了形象化的解释;并对标准中自然视频编码的一些工具和算法,诸如形状编码、运动估计和补偿、纹理编码、Sprite 编码和可分级性编码等进行了分析和讨论。

2 基本概念

2.1 视频对象 (Video Object)

王相海 博士后,讲师,主要研究领域为计算机图形学、视频图像处理。张福炎 教授,博士生导师,主要研究领域为计算机图形学、多媒体技术。

视频对象(VO):视频流中用户可以获取和操作的个体称为视频对象。它由时间上连续的帧画面序列构成。一个视频对象通常与场景中一个特定的 2-D 对象相对应,它的构成通常依赖于具体应用和系统实际所处的环境,最简单的情况,它可以是一个矩形帧,这种情况保证了 MPEG-4 标准与前两个标准的兼容性(参见图 2.1);它也可以是场景中任意形状的一个对象(参见图 2.2),正是这种情况导致了 MPEG-4 与 MPEG-1 和 MPEG-2 最根本的区别:MPEG-4 是基于内容的压缩编码方法。

在 MPEG-4 中,视频对象(VO)通常又被分为自然视频对象和合成视觉对象,前者是指从自然图像和视频分割出来的某一视频区域,图 2.1 和图 2.2 中的 VO 即为这种情况;合成视觉对象是指表面通过若干顶点和曲面片定义的、由计算机产生的 2-D/3-D 对象。

在 MPEG-4 中,对自然视频对象和合成视觉对象进行编码的方式不尽相同,在第 3 节我们主要针对自然视频对象的编码情况进行了讨论。



图 2.1 VO 为矩形帧情况



图 2.2 VO 为任意形状情况图

2.2 视频对象平面 (Video Object Plane)

视频对象平面 (VOP): 是指在某一时刻、某一帧画面的视频对象 (VO) (如图 2.3)。对 VOP 作以下说明 (参见图 2.4);

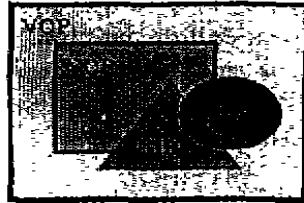


图 2.3 VOP 举例

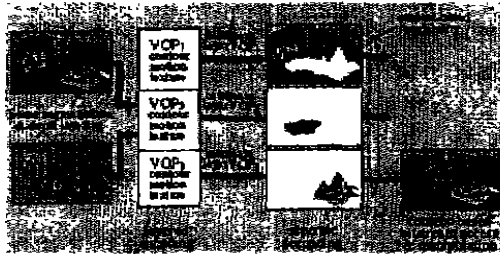


图 2.4 MPEG-4 VOP 的编码使得在解码端可进行基于内容操作的示意图

1. VOP 是 MPEG-4 中最重要、最引人注目的一个新概念,它直接导致了基于内容的压缩,为提供更高的压缩比打下了基础;

2. VOP 编码即针对某一时刻该帧,对画面 VO 的形状、运动、纹理等三类信息进行编码,每个 VOP 可独立编码、存储和传输;每个 VOP 的标识、多个 VOP 在接收端被重新组合为原始序列的相关信息也将传送给解码器;

3. 在解码端可以有选择地对一个或几个 VOP 进行解码,从而可实现基于内容的分级 (Scalability) 和访问 (有关“分级”的概念见 2.4 节)。

值得注意的是,由于 VOP 的获取往往涉及图像分割、运动图像序列分析 (诸如光流场、运动矢量分析) 等经典难题,所以目前在一般情况下还无法完全由算法来自动完成。但是对于象 Video Phone、Video Conference 这样图像内容比较简单的视频流, VOP 的完

全自动获取是有可能的。

2.3 Sprite 图像

Sprite 图像是指在一帧 video clip 中所有可见的、属于某一视频对象的像素所组成的图像,主要是针对背景对象的特点提出的。对 Sprite 作如下说明。

1. 背景视频对象的特点,在许多应用场合,背景视频对象自身是没有任何局部运动的 (或者说局部运动是可以忽略的),其每帧所产生的变化是由于前景物体的运动,一部分背景被掩盖,而另一部分背景又显露出来;或者是由于摄像头的运动,如平移、旋转、缩放所产生的比较复杂的变化;

2. 为了有效地编码背景视频对象,可以将其在一段时间的内容拼接成一幅完整的背景图像,该视频对象在某一帧出现过的像素点,在这幅大的背景图像中都能找到对应点,这样的图像就叫 Sprite 图像 (参见图 2.5);



图 2.5 运动场序列的 Sprite 图像举例

3. Sprite 技术编码机制: Sprite 图像只需要编码传输一次并存放在解码端,随后的图像只需要传输摄像机相对于背景的运动参数,就可以从 Sprite 上恢复所有的图像的背景 (参见图 2.6);

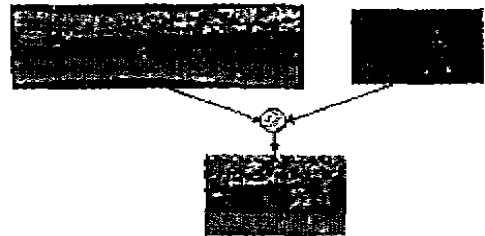


图 2.6 Sprite 编码示意图

4. MPEG-4 定义了两种 Sprite,一种是静态 Sprite (Off-line Sprite): 在整个编码过程中保持不变的 Sprite。在编码前 Sprite 图像和每帧图像重构点都已经知道了,在解码端就能通过解得的重构点来重构当前帧,例如在视频会议和视频电话等场景固定的应用中,背景全景图就可采用静态 Sprite 方式;另一种是动态 Sprite (On-line Sprite): 在编码过程中动态产生的不断更新的 Sprite。由于视频对象的运动变化,不能把每帧图像直接拼接到 Sprite 中去,因此在动态生成 Sprite 图像前应对当前帧图像进行全局运动估计,计算出视频对象的变换参数,将当前帧的图像还原后才能用于

动态 Sprite 图像的更新。有关动态 Sprite 的编码算法在第 3 节进行讨论。

Sprite 图像作为 MPEG-4 的重要概念之一,极大地提高了编码的效率,然而实现 Sprite 编码必须满足两个前提条件,一个是前景对象与背景对象要能很好地分开;另一个是要做到无痕迹地从一段视频或一些图像中拼接出 Sprite 图像。

2.4 可分级性(Scalability)

一个对象的比特流的可分级性是指利用该比特流的一部分(或者说子集)就可以产生该对象的一个有用的表示的能力。基于对象的分级功能是 MPEG-4 提供的又一个新的功能,它同时兼容于 MPEG-2 中的图像分级功能。对可分级性作如下说明:

1. 视频分级编码的目的:一是满足以不同的带宽及不同的显示能力和显示要求进行视频数据库的浏览,以及在多媒体环境下视频内容的多分辨率播放;二是提供分层的视频流以适应按优先顺序进行视频传输的要求;

2. MPEG-4 提供了两种基本的分级方法,空域分级(Spatial Scalability)和时域分级(Temporal Scalability),允许以不同的空域和时域分辨率对 VOP 进行基于内容的传输和访问。为了实现分级操作,视频序列至少要被分为两层:基本层和增强层。基本层提供了视频序列的基本信息,它可以单独传输和解码;增强层提供了视频序列更高的分辨率和细节,它必须与基层一起传输和解码。

• 空域分级:用增强层来增加基本层的空域分辨率(参见图 2.7)。

• 时域分级:由增强层来增加基本层中感兴趣区域的时域分辨率,即帧率。这样,利用增强层将改进指定 VO 运动的“光滑度”。

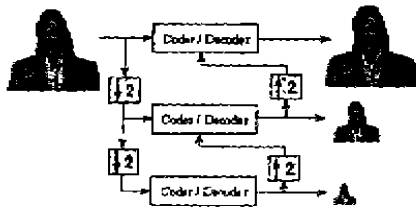


图 2.7 任意形状 VOP 的空域分级方法

2.5 MPEG-4 视频流的逻辑结构

一个完整的视频序列通常由几个视频段(Video Session, VS)构成;每个 VS 由一个或多个视频对象 VO 构成;每个 VO 又由一个或多个视频对象层 VOL 构成;每个 VOL 代表一个层次,即基本层和增强层;每个层表示某一种分辨率。在每个层中,都有时间上连续的一系列 VOP(参见图 2.8)。

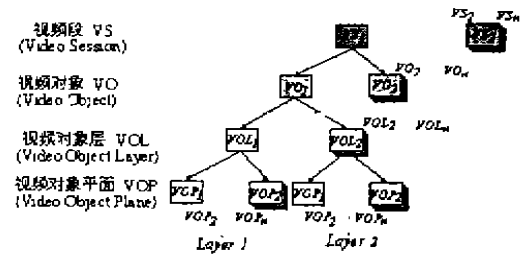


图 2.8 MPEG-4 视频流逻辑结构图

3 MPEG-4 自然视频编码工具和算法

MPEG-4 中每个 VO 由三类信息来描述:运动信息、形状信息和纹理信息,MPEG-4 标准的自然视频编码就是针对这三种信息的编码技术。图 3.1 给出了 MPEG-4 中自然视频编码器的一个基本框图^[3],它包含了对 VO 的形状编码、运动补偿和纹理编码等基本的编码工具。

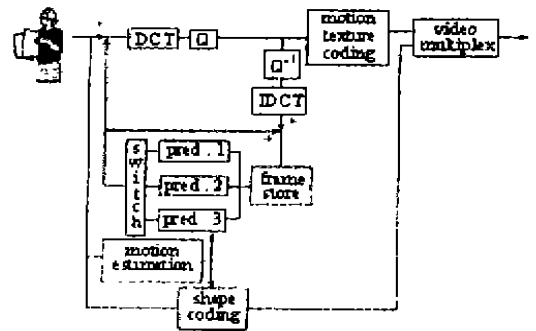


图 3.1 MPEG-4 视频编码器基本框图

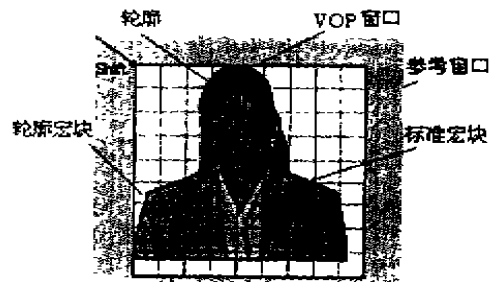


图 3.2 形状自适应的宏块网格示意图

由于 VOP 的位置、形状和大小往往会随时间发生变化,为了实现任意形状 VOP 的形状、运动和纹理信息的编码算法,MPEG-4 视频验证模型 VM(Verification Model)引进了所谓的“形状自适应的宏块网格”^[3,10](参见图 3.2)。其中,参考窗口,原始图像边界;

VOP 窗口, 上边和左边分别与 VOP 的最上和最大左边界相切, 长和宽均为 16 整数倍的最小窗口, 它通过参数“Shift”来确定其相对于参考窗口的的位置; 轮廓宏块: 部分在 VOP 内, 部分在 VOP 外的宏块; 标准宏块: 全部在 VOP 内的宏块。

3.1 形状编码

在 MPEG-1 中有两种形状信息, 即二值和灰度值。为了描述视频对象的形状, MPEG-1 专门定义了与图像大小相同的 α 平面 (有时也将形状信息描述成为 α 平面)。MPEG-4 标准形状编码通常采用下面的位图法。

1) 二值 α 平面 它分别用 0 和 255 表示在一个图像区域内所有像素点的归属。用白色表示该区域内的像素属于当前对象区域, 黑色表示该区域的像素不属于当前对象 (参见图 3.3)。



图 3.3 与图 2.1 相对应的前景对象的二值形状信息

2) 二值 α 平面的编码 仿照图 3.2 的“形状自适应的宏块网格”的划分, 将二值 α 平面分为一个个 16×16 的网格块 (如图 3.4), 边界信息包含在块中。二值 α 平面的编码采用基于“形状块”的运动补偿和二进制算术编码相结合的方法。



图 3.4 二值 α 平面的 16×16 网格分割图

3) 灰度形状信息的编码 仿照二值 α 平面建立灰度 α 平面 (此时 VOP 区域内的像素将是 0~255 之间的灰度值), 其“网格分割图”见图 3.2。对灰度 α 平面的编码采用基于“形状块” (在图 3.4 中称为“轮廓宏块”) 的运动补偿和 DCT 编码方法 (可参见 3.3 节的“纹理编码”)。

以上位图法并不是 VOP 形状编码的唯一方法, 比如将 VOP 用梯度图表示, 形状边界被抽出, 用边界跟踪方法进行编码也很方便, 可以断定, 未来标准中将会引入其它基于几何轮廓的编码技术^[11]。

3.2 运动估计和补偿^[11, 12]

• 4 •

MPEG-4 标准中 ME、MC 方法与其他视频编码标准的 ME、MC 区别不是很大, 主要的不同在于 MPEG-4 中 ME、MC 是基于 VOP 结构的, 这样需要在 VOP 的边缘做特殊考虑。

1) VOP 的三种帧格式 类似于 MPEG-1 的三种帧格式, MPEG-4 中的 VOP 也有三种相应的帧格式 (参见图 3.5), 即帧内 VOP (I-VOP): 本身独立地进行编码的 VOP; 预测 VOP (P-VOP): 基于先前已经被解码的 VOP 来实现预测编码的 VOP; 双向预测 VOP (B-VOP): 基于先前和后续已经被解码的 VOP 来实现预测编码的 VOP。

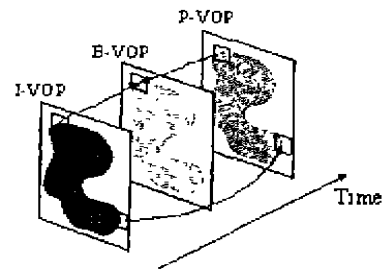


图 3.5 VOP 编码的三种模式

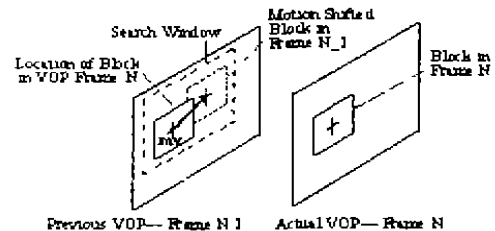


图 3.6 标准宏块的运动估计匹配方法

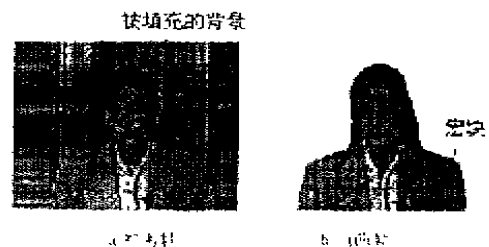


图 3.7 预测参考帧的填充图形示意图

2) VOP 的运动预测与补偿模式 显然, 预测仅仅在对 P-VOP 和 B-VOP 编码时能够用到。运动估计和补偿既可基于 16×16 的宏块, 也可基于 8×8 的块。如果一个宏块为标准宏块, 则运动预测可按通常的方式进行 (参见图 3.6), 此时根据需要, 一个宏块可能使用一个运动矢量, 也可能将宏块的 4 个 8×8 子块每个各使用一个运动矢量; 如果一个宏块为轮廓宏块, 则首先

使用重复填充技术对前面预测参考帧中 VOP 以外的区域进行填充(图 3-7a 给出了对预测参考帧的填充图形, b 为当前帧, 具体填充算法在本节 3.10 中), 然后采用“修改的块(多边形)匹配”(Modified Block (Polygon) Matching)技术来预测运动矢量, 即通过计算宏块内属于 VOP 的像素的绝对误差和(SAD—Sum of Absolute Difference)的最小值来进行预测估计。

3) 重复填充技术 (Repetitive Padding Technique)^[2] 采用该技术对某一帧中 VOP 以外的区域进行填充需要以下五个步骤

步骤 1: 首先将帧中 VOP 以外区域均填为 0 (为了显示方便, 假如填充的为白色)(参见图 3-8);

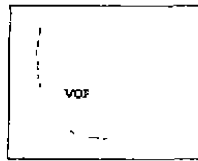


图 3-8 原图

步骤 2: 按自上而下进行水平扫描, 每条扫描线可能被分成两种线段, 一种是 0 段(即该线段上的像素均为 0); 另一种是非 0 段。对于 0 段按下列三种情况进行处理:

- 如果该扫描线上没有任何非 0 段, 则该对扫描线不予填充;
- 如果两端点一个位于扫描线的端点(即起始点或终结点), 另一个位于非 0 段的端点, 则用非 0 段端点的像素值填充此 0 段上的所有像素;
- 如果两端点均为非 0 段的端点, 则用这两个端点的平均值填充此 0 段上的像素(参见图 3-9)。

步骤 3: 仿照步骤 1、步骤 2 按列进行填充(参见图 3-10);



图 3-9 按水平方向填充的情况 图 3-10 按垂直方向填充的情况



图 3-11 由左上角或右下角混合填充的情况

步骤 4: 如果一个 0 像素按步骤 2、步骤 3 进行了填充, 则该像素最后的填充值为这两个填充值的平均

值(参见图 3-11);

步骤 5: 按上述步骤没有填充到的 0 像素, 分别按水平或垂直方向寻找第一个非 0 像素, 将二者的平均值作为该像素的填充值(参见图 3-12)。

3.3 纹理编码^[2]

VOP 的纹理信息有两种形式, 一是帧内 VOP (I-VOP) 中的像素值(亮度和色度成分), 二是帧间 VOP (P-VOP, B-VOP) 的预测误差值。

1) VOP 纹理编码的基本思想 MPFG-4 对两种形式的 VOP 均采用基于分块的 DCT 变换方法, DCT 变换基于 8×8 块, 仍有三种情况(参见图 3-13), 一是 VOP 外、VOP 窗口内的宏块, 对这种宏块不进行编码; 二是标准宏块, 对其采用传统的 DCT 方法编码; 另一种是轮廓宏块, 首先对该块位于 VOP 外的部分进行填充(具体: 如果当前 VOP 为 I-VOP 情况, 则按下面介绍的“低通外推”法进行填充; 如果是 P-VOP 或 B-VOP 情况则只需填零), 然后再用 DCT 进行编码。

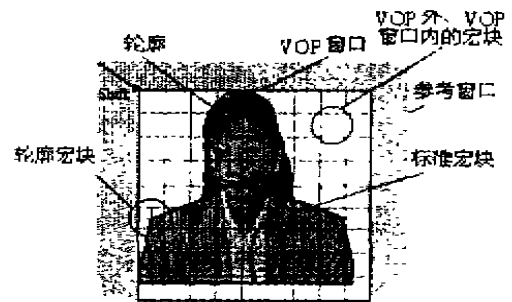


图 3-13 形状自适应的宏块网格示意图 (用于纹理编码情况)

2) 轮廓宏块填充的“低通外推”方法 对于轮廓宏块中非 VOP 的像素采用称作“低通外推”法进行填充。该方法包括以下两个步骤:

• 计算轮廓宏块内属于 VOP 的像素的平均值作为该块内非 VOP 像素的填充值, 即

$$f_{x,y} |_{x,y \notin VOP} = \sum_{(x,y) \in VOP} f_{x,y} / N$$

其中 N 为轮廓宏块内属于 VOP 的像素总数。

• 利用下列“十字”平均运算对上步的填充值进行修正(外推):

$$f_{x,y} |_{x,y \in VOP} = (f_{x,y-1} + f_{x,y+1} + f_{x-1,y} + f_{x+1,y}) / 4$$

具体运算时可能在边界处需要对分母进行调整。

3) VOP 纹理编码的过程(见图 3-14)

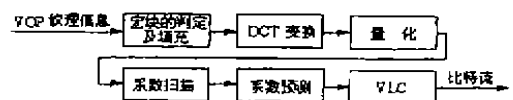


图 3-14 VOP 纹理编码的过程

3.4 Sprite 编码

Sprite 的编码可以分为两类: 静态 Sprite 和动态

Sprite, 前者只需对静态 Sprite 图像以及当前帧在 Sprite 图像中的对应点和范围的参数进行编码。

下面来讨论动态 Sprite 编码的情况^[6]。动态 Sprite 技术的编码过程如图 3.15 所示。其中, 全局运动估计: 计算当前帧的图像相对于前一帧重构图像所发生的运动, 采用的是块匹配算法, 因而只能估计图像的平移运动; 运动参数计算: 在镶嵌和更新 Sprite 前, 为了恢复由于图像的旋转和缩放等复杂运动所带来的扭曲和变形, 需要计算射影变换的 8 个参数; 参考点计算: 计算当前帧图像的左上角位于 Sprite 图像的位置; 运动补偿: 主要是基于不断更新的 Sprite 所进行运动补偿, 因而又被称为全局运动补偿。下面主要介绍全局运动估计和变换参数计算的算法实现思想,

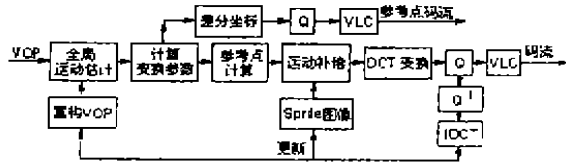


图 3.15 动态 Sprite 编码框图

1) 全局运动估计 全局运动估计的目的是确定当前要编码的 VOP 在 Sprite 图像中的位置。由于 Sprite 图像的尺寸要比 VOP 大许多, 如果直接用 Sprite 图像作全局运动估计其计算量将很大, 因而通常采用前一帧的重构 VOP 作参考来估计当前的 VOP 运动。具体方法是:

• 为了进一步减少计算量, 对参考 VOP 和当前编码 VOP 均作两次如式(1)的滤波, 产生类似金字塔结构的三层图像(参见图 3.16)。最底层是原图像, 最上层图像的长和宽都只是原图像的四分之一, 图像的像素点减少了 16 倍。

$$\begin{aligned} f(x, y) &= [f(x, y_{2j-1}) + 2f(x, y_{2j}) + f(x, y_{2j+1})] / 4 \\ f(x, y) &= [f(x_{2i-1}, y) + 2f(x_{2i}, y) + f(x_{2i+1}, y)] / 4 \end{aligned} \quad (1)$$



图 3.16 原图像经变换(1)后所形成的分层图像示意图

• 图像的运动估计是在最上层图像用三步搜索算法完成的。具体定义了误差估计函数式(2), 并按式(3)是搜索得到最佳运动矢量 $(\nabla x_{min}, \nabla y_{min})$ 。

$$SAD_{\nabla x, \nabla y}(x, y) = \sum_i \sum_j |f(x+i, y+j) - f(x+i+\nabla x, y+j+\nabla y)| / N \quad (2)$$

$$(\nabla x_{min}, \nabla y_{min}) = \min_{\substack{-1 \leq \nabla x \leq 1 \\ -1 \leq \nabla y \leq 1}} \min_{\substack{-1 \leq \nabla x \leq 1 \\ -1 \leq \nabla y \leq 1}} SAD_{\nabla x, \nabla y}(x, y) \quad (3)$$

2) 变换参数计算 前面的全局运动估计只能估计图像的平移运动, 而摄像头的运动除了平移外, 往往还会产生旋转和缩放等运动, 为了估计这样的运动, 通常按照下面式(4)、(5)的运动模型(二维射影变换的非齐次形式)求解图像的 8 个运动参数。

$$y = (a_4x + a_5y + a_6) / (a_7x + a_8y + 1) \quad (4)$$

$$x = (a_1x + a_2y + a_3) / (a_7x + a_8y + 1) \quad (5)$$

如果用 $f(x, y)$ 和 $f'(x', y')$ 分别表示当前编码图像和参考图像, 那么两幅图像坐标变换的矩阵表示形式如(6)所示。

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix} \quad (6)$$

定义误差函数式(7), 则满足误差最小的变换参数 $a_r (r=1, \dots, 8)$ 即为所求得变换的参数。

$$E = \sum_i |f'(x'_i, y'_i) - f(x_i, y_i)| = \sum_i e_i^2 \quad (7)$$

3.5 分级

MPEG-4 使用了一个具有一般性的分级框架来描述分级的过程^[12](参见图 3.17)。

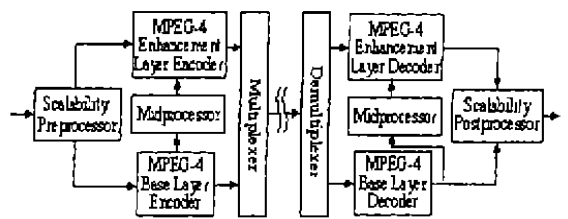


图 3.17 MPEG-4 一般分级框架框图

这里, 分级预处理器对所输入的 VOP 完成分级工作, 具体, 在空域分级情况, 分级预处理器对输入的 VOP 进行降取样(down-sample)以产生基层 VOP(该基层 VOP 由 VOP 编码器进行处理), 中间处理器对新建构的基层 VOP 进行升取样(up-sample), 并以此作为增强层的预测 VOP, 原始 VOP 与预测 VOP 误差估计值作为增强层处理器的输入信息进行编码; 对于时域分级情况, 分级预处理器将输入帧分成两个数据流, 分别用于基层编码器和增强层处理器的处理, 此时中间处理器不再使用。

1) 空域分级 在空域分级中, 基层 VOP 的编码与前面讨论的不分级 VOP 的编码情况相同。而增强层的 VOP 不再存在 I-VOP 的情况, 也就是说增强层的 VOP 均以 P-VOP 或 B-VOP 的形式进行编码(参见图 3.18)。

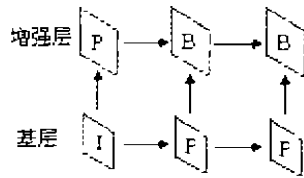


图 3.18 空域分级中基层和增强层的情况说明

如果增强层中的一个 VOP 在时间上与基层上的一个 I-VOP 相对应,那么它将以 P-VOP 的形式进行编码;增强层中在时间上与基层上的 P-VOP 相对应的 VOP,将以 B-VOP 的形式进行编码;由于基层被看作是增强层的预测层,所以基层上的 VOP 必须先于增强层中与之相对应的 VOP 进行编码。

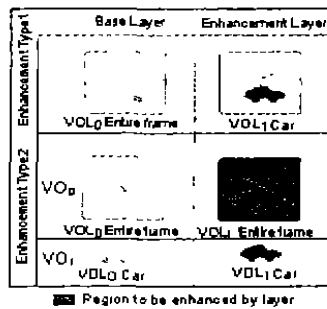


图 3.19 MPEG-4 时域分级的两种增强类型

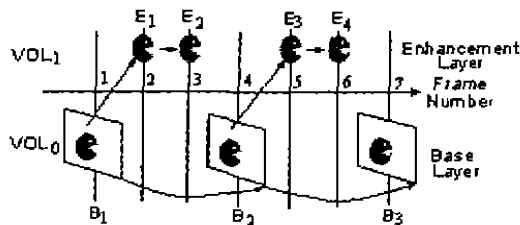


图 3.20 时域分级中增强类型 1 的基层和增强层的情况

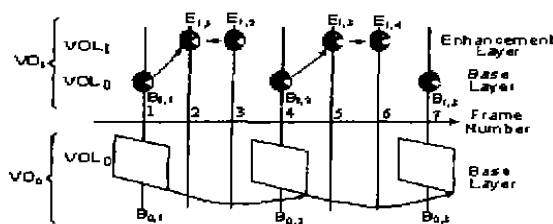


图 3.21 时域分级中增强类型 2 的基层和增强层的情况

2) 时域分级 在时域分级中,增强层可以两种类型来提高基层的帧率,一种类型是增强层仅仅提高基

层中部分 VOP 的帧率,另一种类型是增强层提高整个基层的帧率(参见图 3.19、图 3.20 和图 3.21)。

结束语 MPEG-4 标准使用了基于对象的视频序列表示,从而可方便地对视频帧中任意形状的区域进行访问和操作,基于视频对象(VO)的结构直接支持了一个非常重要的功能:基于对象的可交互性;空间可分级性和时间可分级性在 MPEG-4 中也被支持,它是通过对信息进行分层这一机制来实现的,其中基层提供了视频序列的基本信息,增强层提供了视频序列更高的分辨率和细节,前者可单独传输和解码,而后者则必须与基层一起传输和解码;Sprite 和图像纹理的编码也是 MPEG-4 所支持的两个新的特征。文中通过大量的图示对以上功能和实现算法以及一些重要概念进行了介绍和讨论。

MPEG-4 编码还具有鲁棒性和纠错功能,它是通过再同步、数据恢复和错误隐藏这三个策略来实现的,限于篇幅在文中没有介绍,具体可参见文[9,12]。

参考文献

- 1 MPEG-1 Video Group. Information Technology-Coding of Moving Pictures and Associated Audio for Digital Storage Media up to about 1.5 Mbit/s; Part 2-Video. ISO/IEC 11172-2. International Standard, 1993
- 2 MPEG-2 Video Group. Information Technology-Generic Coding of Moving Pictures and Associated Audio; Part 2-Video. ISO/IEC 13818-2. International Standard, 1995
- 3 Changhone L. MPEG and Multimedia Communications. IEEE Transactions on Circuits and Systems for Video Technology, 1997, 7(1): 5~18
- 4 Overview of the MPEG-4 Standard, ISO/IEC JTC1/SC29/WG11, N2323. Dublin, Ireland, 1998
- 5 Overview of the MPEG-4 functionalities supported in MPEG-4 Version 2. ISO/IEC JTC1/SC29/WG11, N2197 Tokyo, Japan, 1998
- 6 MPEG-4 Video Group. Generic Coding of Audio-Visual Objects; Part 2 - Visual, ISO/IEC JTC1/SC29/WG11, N1902. FDIS of ISO/IEC 14496-2. Atlantic City, Nov. 1998
- 7 Lee M C, Chen W G. A layered video object coding system using sprite and affine motion model. IEEE Transaction on Circuit and Systems Video Technology, 1997, 7(1): 130~145
- 8 吴枫,高文,陈大童. 动态 Sprite 编码的研究与改进. 计算机学报, 1999, 22(3): 262~268
- 9 Overview of the MPEG-4 Standard, ISO/IEC JTC1/SC29/WG11 N2725, March, 1999
- 10 Sikora T. The MPEG-4 video standard verification model. IEEE Trans. on Circuits and Systems for Video Technology, 1997, 7(1): 19~31
- 11 钟玉琢,蔡莲红,等. 多媒体计算机技术基础及应用. 北京:高等教育出版社, 1999
- 12 Touradj E, Caspat H. MPEG-4 Natural Video Coding-An overview, 1999