

基于对象的编码模式优化选择算法^{*})

张 剑 郑心炜

(华中科技大学计算机科学与技术学院国家外存储系统专业实验室 武汉430074)

摘 要 各种视频压缩编码标准都是根据人们在不同领域中对声像数据的要求所制定的,并且随着人们的需求不断地发展。目前,视频压缩编码研究主要分为两个方向:一是基于传统的DCT混合编码方案;另一个是基于第二代图像编码技术而提出的基于对象的编码方案。其中,基于对象的编码方法不仅能满足进一步获得更大的图像数据压缩比的要求,而且能够实现人机对话的功能,所以,我们认为它将是未来视频压缩编码的发展方向。本文对基于对象的编码模式选择算法进行了研究,借用了MPEG-4中视频对象的概念,提出了一种视频分割的方法。

关键词 率失真, H. 264编码, 对象, 混合编码

The Optimization Selection Algorithm of Coding Model Based on Object

ZHANG Jian ZHENG Xin-Wei

(Huazhong University of Science and Technology, Wuhan 430074)

Abstract Various video compressed coding standards are established in the sphere of difference according to people for the requirement of the data of voice and image, and developed continuously along with the demand of people. Now, video compressed coding research divides into two directions mainly. One is that DCT hybrid encoding scheme based on tradition; another is the coding scheme based on object of putting forward based on 2 generation image coding technology. The coding method based on object, can not only satisfy the function that gets the requirement of larger image data compression ratio further and can realize man-machine interaction, so we think it will be future video compressed coding develop direction. In this paper, we study selection algorithm of the coding model based on object, and give the method of video cuts apart using the concept of video object in MPEG-4.

Keywords Rate-distortion, H. 264 coding, Object, Hybrid encoding

1 混合编码模式方法

混合编码框架的编码模式选择方法主要有两种,一种是以编码前数据的绝对误差为依据,另一种则是以率失真的准则为依据。它们都需要对每一种可能的模式分别进行检查,其中后一种方法还需要按照各种可能的模式做一次实际编码,来计算该模式下的失真和比特率^[1],这使得编码过程异常复杂,大大增加了编码所需要的时间。譬如在H. 264预测模式中,利用了相邻像块的6种相邻像素相关性,将QCIF(176×144)格式的图像分成99个16×16宏块,然后分成4×4像块做预测编码,组成6种4×4亮度像块的帧内预测模式,另外,增加了帧间预测模式的数量和精度,有7种不同的宏块分割方法,从而又形成7种不同的帧间预测模式,显然编码模式近二十种,这令其模式选择过程占用了大量的编码时间,为了解决这个问题,本文对基于对象的编码模式选择算法进行了研究,借用了MPEG-4中视频对象的概念,提出了一种视频分割的方法,将待编码序列分作若干个视频对象,对每个视频对象采用不同的模式选择方法,只在空间域上进行编码算法,除去相邻块之间的空间冗余度,尽可能在保证编码性能不受影响的条件下提高编码速率,取得更为有效的压缩。

2 视频分割

视频分割是指将视频序列按照一定标准分割成多个区

域,目的是为了从视频序列中分离出有一定意义的实体,而这种有意义的实体便是视频对象^[2]。视频分割的方法和准则有很多,具体选用哪种方法或准则主要取决于分割的目的和用途。用于编码目的的视频分割一般基于视频图像低层次的特征,其自动分割技术相对来说是比较容易实现的。本文讨论的视频分割主要是为了便于编码模式的选择,同时利用帧间位移差(DFD)进一步对分割进行完善和细化。整个视频分割过程可以分为:负责确定视频对象的视频对象划分;负责对视频对象状态随时间的变化进行估计的视频对象跟踪。

2.1 视频对象划分

在对第一个INTER帧进行编码时,依据H. 264中提供的编码模式,基本编码单位是16×16宏块,H. 264每个视频对象是若干个宏块组成的一个集合。我们将输入的视频序列分作三个视频对象:背景区域、亚背景区域和运动前景,即采用16×16帧间预测模式或16×16帧内预测模式宏块集合的背景区域;采用16×8及8×16帧间预测模式宏块集合的亚背景区域;采用8×8帧间模式和各种子模式宏块集合的运动前景。又由于INTRA帧中禁止了多种编码模式,且不能作为划分的依据,为了保证视频对象划分的准确性,该帧的编码模式选择过程必须检查所有编码模式并选取最佳的模式,如图1所示,给出了对Miss America序列进行视频对象划分的结果,图中,■为运动背景;//为亚背景区域;□为背景区域。

^{*})本课题得到全国教育科学十五规划重点课题(NO: AYA010034)和国家自然科学基金(NO: 69735020)资助。张 剑 副教授,高级工程师,博士研究生,研究方向:计算机多媒体视频压缩算法及其实现。郑沁炜 博士研究生,研究方向:计算机多媒体视频压缩算法及其实现。

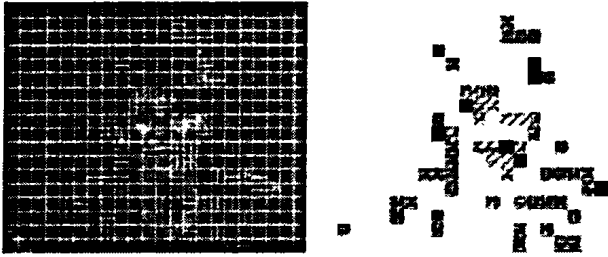


图1 视频对象划分结果图

另外,对于图像对象采用的编码模式是与其内容有关系的,譬如,静止区域和运动简单的区域采用了匹配块较大的帧间编码模式;运动复杂的区域采用了匹配块较小的帧间编码模式。

2.2 视频对象跟踪

利用视频对象先前的状态来估计其当前状态的过程是视频对象跟踪的过程,对于视频对象 VO,其状态可以用一个随时间变化的宏块集合 $\Phi_{VO}(t)$ 来描述。

2.2.1 背景区域的跟踪方法

定义1 假设 $\Phi(x, y)$ 位于图像 (x, y) 左上角像素处的一个宏块, $\Phi_{BG}(t)$ 是用来描述背景区域状态的宏块集合。若 $\Phi(x, y) \in \Phi_{BG}(t)$, 且对于所有 $x + \Phi V_x \leq \Phi \leq x - \Phi V_x$, $x + \Phi V_y \leq n \leq x - \Phi V_y$, 都存在 $\Phi(\Phi, n) \in \Phi_{BG}(t)$, 则 $\Phi(x, y) \in \Phi_{BG}(t+1)$, 即 $\Phi(x, y)$ 在下一帧图像中依然属于背景区域。

其中, ΦV_x 和 ΦV_y 分别为运动矢量在 x 方向和 y 方向的最大值。在 $\Phi V_x = \Phi V_y = 16$ 的情况下, 如果宏块 $\Phi(x, y)$ 及其周围的8相邻宏块在时间 t 均属于背景区域, 则 $\Phi(x, y)$ 在时间 $t+1$ 也属于背景区域。由此可见, 视频对象状态是具有时间相关性的, 视频对象的运动仅限于平移假设, 依据定义1对背景的跟踪是非常有效的, 如图2所示。

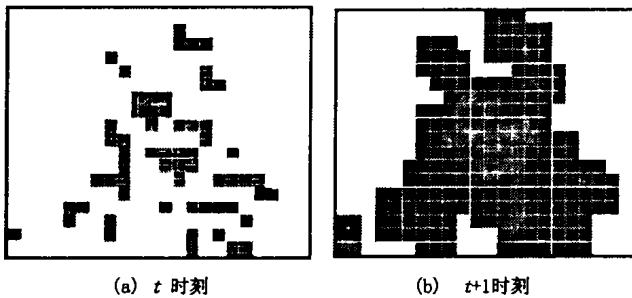


图2 白色部分为背景区域的跟踪

但是, 这种跟踪方法属于一种保守的跟踪方法^[3], 它发生运动的仅仅是亚背景部分和运动前景, 而没有考虑背景区域的漂移, 因此, 对于亚背景部分和运动前景的侵蚀作用, 背景区域的面积将会不断缩小。为了解决这个问题, 还需要在编码结束之后根据实际的模式选择结果进行一次补充性的视频对象划分。

2.2.2 运动前景的确定方法

定义2 假设描述运动前景状态的宏块集合为 $\Phi_{FG}(t)$, 通过估计得到的运动前景状态集合子集为 $\Phi'_{FG}(t)$ 。对于宏块 $\Phi(x, y)$, 若 $\Phi(x, y) \in \Phi_{FG}(t)$, 且对于所有 $x + \Phi V_x < \Phi < x - \Phi V_x$, $x + \Phi V_y < n < x - \Phi V_y$, 满足 $\Phi(\Phi, n) \in \Phi_{FG}(t)$, 则认为 $\Phi(x, y) \in \Phi'_{FG}(t)$, 即 $\Phi(x, y)$ 属于运动前景部分。

其中, ΦV_x 和 ΦV_y 依然是运动矢量在 x 方向和 y 方向的最大值, 与背景区域不相邻的部分都是运动前景。对于满足 $\Phi(x, y) \in \Phi_{FG}(t)$, 但不满足 $\Phi(\Phi, n) \in \Phi_{FG}(t)$ 宏块的 $\Phi(x, y)$, 需要根据帧间差值来判断其所属的视频对象。

文[1]中提到利用帧间差值进行运动对象的分割的方法, 但需要计算多个帧间差值的高阶统计量, 这样的计算复杂度很高, 不适合实时视频编码的要求, 因此, 本文利用易于计算的 SAD 值来进行判断。

定义3 对于满足 $\Phi(x, y) \in \Phi_{FG}(t)$, 不满足 $\Phi(\Phi, n) \in \Phi_{FG}(t)$, 必然存在 $x + \Phi V_x \leq \Phi \leq x - \Phi V_x$, $x + \Phi V_y \leq n \leq x - \Phi V_y$, 满足 $\Phi(\Phi, n) \in \Phi_{BG}(t)$, 则:

$$SAD_{AVG} = \frac{1}{N} \sum_{i=1}^{N-1} SAD_{ms}(X_i)$$

其中, $X_i \in \{\Phi(\Phi, n) | x + \Phi V_x \leq \Phi \leq x - \Phi V_x, y + \Phi V_y \leq n \leq y - \Phi V_y\} \cap \Phi_{BG}(t)$, $SAD_{ms}(X_i)$ 是宏块 X_i 与参考图像最佳匹配区域的 SAD 值, N 为满足以上条件的 X_i 的个数。在 $\Phi V_x = \Phi V_y = 16$ 的情况下, N 的最大值为8, 但由于 H. 264 编码过程的限制, 其中只有当前宏块左方、左上方、上方和右上方4个相邻宏块的 SAD 值可以得到, 如图3所示。若 $SAD_{ms}(\Phi(x, y)) > SAD_{AVG}$, 则认为 $\Phi(x, y) \in \Phi_{FG}(t)$ 。

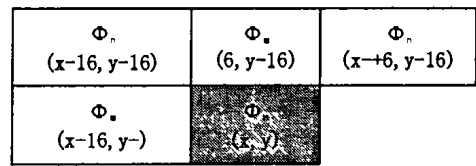


图3 相邻宏块的 SAD 值进行运动前景估计

当背景区域和运动前景确定之后, 剩下的部分便是亚背景区域。

3 编码模式选择过程的优化

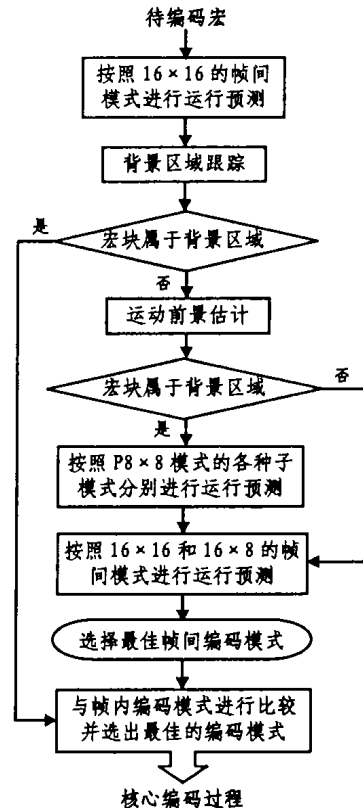


图4 优化后的 H. 264 模式选择过程

视频分割可以实现基于对象的编码模式选择, 即对于不同的视频对象, 采用不同的选择编码模式集合^[4]。背景区域可选择: 16×16 帧内预测模式、16×16 帧间预测模式; 亚背景区域可选择: 4×4 帧内预测模式、16×16 帧内预测模式、16×16 帧间预测模式、16×8 帧间预测模式、8×16 帧间预测模式; 运

动前景可选择:4×4帧内预测模式、16×16帧内预测模式、16×16帧间预测模式、16×8帧间预测模式、8×16帧间预测模式、4×4帧间预测模式及各种子模式。为此,H.264的编码模式选择流程得到了一定程度的优化,优化后的框图如图4所示^[1]。

4 帧间预测程序

帧间预测编码过程包括:modon_search()计算运动矢量、LumaResidualCoding_P()变换编码残差图像。其中运动矢量的获取,需要进行不同块大小的矢量搜索、多参考帧的选取、高精度的运动估计等H.264所特有的帧间编码策略。

```
//P 帧运动矢量估计
int motion_search (inttot_intra_sad)
{
int predframe_no,mjn_inter_sad,hv,i,j;
predframe_no = UnifiedMOdOnSearch (tot_intra_sad, refFrArr,
tmp_mv,img_mv,
&img_mv_mode,&img_mv_blc_size_v,&img_mv_multiframe_no,
PFRAME,&min_inter_sad,img_mv_all_mv);
//tot_intra_sad 是帧内编码的最小 SAD
//min_inter_sad 是帧间编码的最小 SAD
//根据 SAD 大小决定采用哪种编码方式
if (tot_intra_sad<min_inter_sad)
{
img_mv_mb_data [img_mv_current_mv_nr].intraOrInter=MTER_MB
img_mv_imod=INTRA_MB_INTER;//设置帧间编码模式
```

```
for(hv=0;hv<2;hv++)
for(i=0;i<4;i++)
for(j=0;j<4;j++)
在此不再详细说明
```

程序中计算 SAD 的函数定义如下,static intMVCost(int Parameter Resolution,in Bitstream Resolufion,intBlocktype, int qp,int Pred x,int Pred_y,int Candidate_x,intCandidate_y.各种精度的矢量搜索方法同快速搜索方法类似,整数像素的运动矢量搜索是基于整数位置图像 Refbunl_P 进行的,对半像素精度、1/4像素精度和1/8像素精度的运动矢量搜索,输入参考图像均为经过1/4插值的图像 mref_P,因此需要对1/2精度搜索和1/8精度搜索需要做相应处理,才能获得参考图像数据。

5 实验结果及分析

最后,利用本文提出的优化的编码模式选择方法,对H.264的测试模型 JM61e 进行了改进^[5],表1、表2和图5给出相应的实验结果:根据实验结果可以看出,与 JM61e 相比,改进后的编码程序可以在信噪比降低小于0.05db,比特率增加不超过1%的前提下,节约10%~20%的整体编码时间。对于运动较小的低比特率序列,改进效果尤其显著。

表1 QP=28,不使用率失真优化方法编码时间比较

序列	h.264 测试程序 JM61e			采用本文中的算法					
	PSNRY	比特率	时间	PSNRY	比特率	时间	PSNRY	比特率	时间
	dib	kbps	ms	dib	kbps	ms	down	increase	节约
Container_qcif(10fbs,100 frame)	35.73	25.60	96.44	35.69	25.80	82.44	0.02	0.63%	15.02%
MissA_cif(30fbs,100 frame)	38.35	158.46	399.99	38.30	158.80	339.11	0.03	0.20%	22.6%
Football_cif(30fbs,80 frame)	36.85	969.15	318.48	36.79	969.52	289.10	0.01	0.13%	13.03%

表2 QP = 28,使用率失真优化方法编码时间比较

序列	h.264 测试程序 JM61e			采用本文中的算法					
	PSNRY	比特率	时间	PSNRY	比特率	时间	PSNRY	比特率	时间
	dib	kbps	ms	dib	kbps	ms	down	increase	节约
Container_qcif(10fbs,100frame)	35.88	23.82	164.61	35.81	23.88	147.04	0.04	0.63%	11.02%
MissA_cif(30fbs,100 frame)	38.35	130.49	644.17	38.52	130.10	534.11	0.03	0.20%	16.60%
Football_cif(30fbs,80 frame)	37.27	942.52	533.45	37.12	940.92	473.10	0.03	0.13%	11.39%

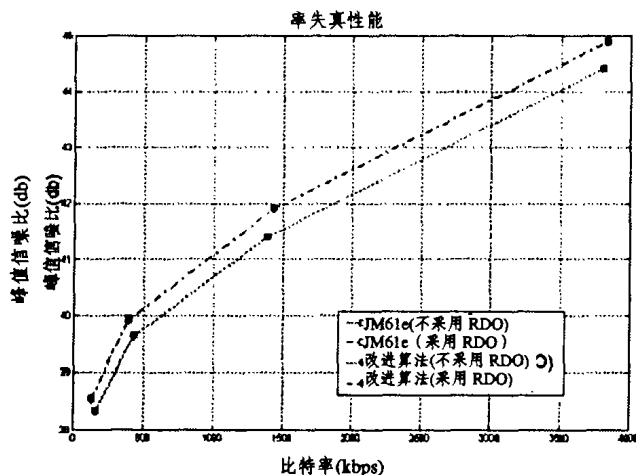


图5 Miss America CIF 序列率失真性能比较

结论 通过对序列进行视频分割,将其划分为背景区域、亚背景区域和运动前景三种视频对象,可以实现基于对象的编码模式选择,从而优化了模式选择的过程,使整体的编码速度得到提高。实验表明,编码模式优化能降低运动矢量和块模

式传输的开销,从而提高压缩效率、降低编解码实现的复杂度。另外,上述实验情况,编码器的率失真性能没有明显下降,这是由于运动的复杂性,需要提高当前混合编码框架的编码速度,确定一种快速、准确的运动前景跟踪方法。因此,在运动前景追踪的方向上还需要作进一步的研究,本文仅将基于视频对象的编码思想用于编码模式选择的过程中,而如何实现整个编码过程的面向对象化将是下一步的研究方向。

参考文献

- 1 Wiegand T. Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG. 7th Meeting: Pattaya, Thailand, March 2003
- 2 Hallapuro A, Karczeicz M, Malvar H. Lo complexity transform and quantization. JVT of ISO/IEC MPEG and ITU-T VCEG, Docs. JVT-B038 and JVT-B039, Jan. 2002
- 3 Wiegand T. Joint Video Team (JVT) of ISO/IECMPEG and ITU-T VCEG. Geneva, Switzerland, January 29-February 1, 2002
- 4 Zhang Jian. A Study on the Source Extension Based on H.264e-Entropy. Journal of Shanghai University, 2005(2)
- 5 Wiegand T. Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC). JVT-G050, March 2003