H.323 视频会议系统中视频编解码子系统设计若干问题的研究

王中元 胡瑞敏 傅佑铭 边学工
(武汉大学国家多媒体软件工程研究中心 武汉 430072)

摘 要：该文研究了 H.323 视频会议系统中的视频编解码子系统设计中的运动估计、码率控制等问题，介绍了每个问题产生的背景，并对提出的算法原理和算法步骤进行了详细描述，对每个问题在解决方法上都有所改进或创新。研究成果应用到实际的 H.323 视频会议系统中，大幅度地提高了系统的视频性能技术指标。

关键词：视频会议；视频编解码；视频通信
中图分类号：TN919.85 **文献标识码**：A **文章编号**：1009-5896(2007)07-1596-04

Study on Video Coding Subsystem Design of H.323 Video Conference System

Wang Zhong-yuan Hu Rui-min Fu You-ming Bian Xue-gong
(*National Multimedia Software Engineering Research Center, Wuhan Univ., Wuhan* 430072, *China*)

Abstract: This paper presents eight problems of video encoding and decoding subsystem design in H.323 video conference system, such as motion estimation, rate control etc. For each problem, after introducing their background, the paper finds their solutions and algorithm steps are described in detail. Some technical renovations are discussed also. These results are incorporated into a real-life H.323 system which achieves outstanding video performance improvement.

Key words: Video conference; Video codec; Video communication

1 引言

ITU-T H.323[1]标准是IP网络多媒体通信系统最主流的标准之一，随着IP网络的发展，基于H.323 的应用越来越广泛。视频编码和通信子系统设计是研制H.323 视频会议系统中最困难的部分之一，其原因在于视频通信本质上数据量大、计算复杂度高、编码数据率的抖动、变长压缩码流对网络丢包或误码敏感等；另外视频编码协议如H.263[2]，在制定上的开放、灵活也为协议的实际应用增加了难度，所以视频编解码子系统设计的质量直接影响到H.323 系统的总体性能。笔者在研制有自主知识产权的H.323 视频会议系统的过程中，对视频编码和通信子系统主要的问题进行了归纳和总结，在参考同行研究成果的基础上，进一步对问题的多个方面提出改进，最终研制出的H.323 视频会议系统中的主客观视频性能指标都达到了要求。

2 视频信源编码

2.1 运动估计快速算法

运动估计快速算法的目标是通过某种方式确定对应于一个图像块的相对于其参考帧的运动矢量，以使基于运动补偿的帧间预测误差达到最小，同时这个过程必须是高效的，以保证视频通信中对视频编解码实时性的要求。存在多种统计指标来衡量对应于确定的运动矢量的帧间预测误差的大小，最常用的是 SAD(累加绝对差)，为了加速 SAD 的计算，本文采取了两个改进措施：首先，SAD 计算只对二分之一亚采样的图像块进行，即只考虑图像块中行列坐标同为奇数的像素点的帧间预测误差，如此计算量减至原来的四分之一；其次，考虑到 SAD 计算的目的是使 SAD 达到最小的运动矢量，在对一个特定的图像块进行运动估计的过程中随时记录此前所得到的最小 SAD 并作为一个控制阈值 SAD_Escape 参与当前的 SAD 计算，在 SAD 计算按照行来累加的过程中，每计算一行便与控制阈值 SAD_Escape 进行比较，如大于 SAD_Escape 则 SAD 计算终止。这样做，避免了不必要的 SAD 计算，但也增加了与控制阈值 SAD_Escape 进行比较的运算，从统计的角度讲，对于提高计算效率的目的而言，其利要远大于弊。

关于运动矢量的搜索策略，全搜索是最完备的，保证能找到使帧间预测误差达到最小的最佳运动矢量，但其穷举的搜索策略计算效率十分低下。本文的运动估计搜索策略基于多备选点的分层递进原则，搜索分 3 个层次由粗而细进行：每个层次保留一个或两个最优备选点作为下个层次搜索的中心，保留次优备选点的充要条件是两个备选点对应的帧间预测误差的差异小于多备选点偏好控制阈值；搜索分层递进时搜索间隔减半，最粗一层为 4 个像素，最细一层为一个或半个像素；搜索起点的确定考虑到空间位置上与当前图像块相邻的已完成运动估计的图像块的运动矢量，取其均值，理由是相邻的图像块可能属于同一运动物体，因而应具有相近

2005-12-22 收到，2006-11-03 改回

的运动矢量。

将本文方法同常用的快速搜索方法的效率进行对比，结果列于表 1，比较使用的测试序列是 300 帧CIF格式的Foreman序列，编码参数设置为 384kbps，25fps。同最新提出的、受专利保护的预测运动矢量场自适应搜索技术(PMVFAST)[3]相比，本文方法计算效率略有下降，但搜索精度略有上升。本文方法一个更突出的优点是便于并行计算，因为搜索过程中需要扫描的备选点位置在搜索前可以预知，而PMVFAST的备选点间则有递进搜索的依赖关系，利用这个特点便于在多媒体处理器上进行指令级和数据级的并行优化[4]。

表 1　搜索策略性能比较

搜索策略	相对效率	PSNR(dB)
梯度下降法	6.07	33.67
PMVFAST	5.82	35.24
本文方法	5.68	35.29
对数搜索	5.60	34.28
三步法	5.52	34.39
快速块匹配法	2.73	36.73
全搜索	1.00	36.73

2.2 码率控制

在视频会议或可视电话等实时视频通信中，公认的TMN8 码率控制算法[5]因其码率适配精确而被广泛采用。TMN8 码率控制模型适用于所有的固定比特率的应用场所，它由一个帧层和一个宏块层组成，其中，在帧层内选定当前帧的目标比特率，而在宏块层内修改量化参数以得到目标比特率。这个算法是目前唯一适用的速率控制算法。在研制视频会议系统的过程中发现，如果在如下方面对TMN8 算法进行改进，则可以取得更好的实际应用效果。

(1)金字塔加权　对于可视电话、视频会议类应用，画面的主要活动区域集中在图像中心的人头肩像部位，人眼感观视觉对这部分也最敏感，因此可以考虑对画像中心采取金字塔型加权，即加权系数的权值从中心到四周满足金字塔型分布。按照这一思路权值计算公式为，

$$\alpha_k = A_1\left(1-\frac{|r-R|}{R}\right)\left(1-\frac{|c-C|}{C}\right)+A_2$$

其中A_1为常数 1.5，A_2为常数 0.1，R为图像行宏块数的一半，C为列宏块数的一半，r为宏块行号，c为列号。图 1(a)是没有加权的图像，图 1(b)是加权后的图像，可以看出，图 1(b)的面部看起来更清晰。

(2)引入“负”码流缓冲区概念　在 TMN8 模型中目标编码比特率和帧率是两个重要的输入参数，模型的精确程度很大程度上取决于目标参数设置是否合理，例如，如果要求在 64kbps 下编码 25fps，其自适应控制出来的实际比特率是无论如何达不到 64kbps 的，因此，根据先验知识合理地设

(a)　(b)

图 1　加权视觉效果比较

置输入参数很有必要。但即使输入参数设置没有问题，在实际应用中还会有一种情况影响模型的精度，即当目标帧率高出实际采样帧率许多时，实际控制出来的比特率远小于目标比特率。而实际应用中采样帧率往往是变化的，目标帧率在TMN8 模型初始化时就确定，不太可能动态地修正目标帧率的初始化设置。进一步分析 TMN8 模型，这个问题的出现究其原因是编码比特流发送缓冲区的充满程度不允许出现负数，本文外延了这个概念，引入“负”值，即缓冲区不仅可以处于空、满、部分满等状态，而且还可以是“负空”，也就是下溢。当然，实际上只要修正 W 值定义，允许其计算时出现负值就可以，而不必过于拘泥于其物理含义。

2.3 INTRA 刷新算法

帧间预测编码会带来预测误差的累积，如果不加以处理，则视频编码器运行一段时间后(通常约 5min)图像会出现色变，所以每隔一段时间需要通过INTRA编码控制误差的累积。例如，H.261，H.263 标准定义的宏块INTRA刷新间隔是 132 次。通常控制误差累积的方法有两种[6]：

方法 1　定时编 INTRA 帧，间隔一般是 10s 左右，该方法能够有效控制预测误差累积，不足之处表现为：一是定时的 INTRA 帧会带来编码数据量的突发性；二是画面闪烁，原因是 INTRA 帧编码在量化步长不是特别小时(一般 4 以下)，编码的块效应更明显。

方法 2　为了改善 INTRA 帧数据突发性问题，不再整帧刷新，而是每次只刷新几个宏块，按照图像编码的顺序，自上而下、自左至右刷新，刷新位置逐帧顺序移动，一次刷新宏块数目按照(总宏块数/132)计算，例如 CIF 格式共有 396 宏块，则每次刷新 3 个宏块。但实践发现，该方法的不足之处是图像上局部闪烁以刷新的顺序不停地跳动，就好像有物体在移动一样。为了克服这个问题，本文提出了随机刷新的方法。

随机刷新的思路是，依然每帧只刷新(总宏块数/132)个宏块，但刷新位置是随机的，不再是有规律的顺序改变。产生刷新位置的随机函数是

seed = 31821seed + 13849，seed 初始值为 21845。

刷新算法步骤如下：

(1)令 k 为编码时的宏块物理顺序号，update_count 为刷新宏块计数器，update_pos 为刷新位置计数器，update_num 为每帧允许刷新的宏块数目

(2)编码器初始化时，初始化变量

update_pos=0， update_num $=\frac{N}{132}$；

(3)每帧编码时刷新方法，

```
update_count=0;
For(k=0;k<N;k++)
begin
    if(k=refresh_table [update_pos] and update_count< update_num) then
        Intra_Encode_MB(); update_count++; update_pos++;
        if(update_pos>=N) update_pos=0;
    endif
end
```

这个方法克服了前两个方法的缺点，取得较好的主观刷新效果。

3 音视频同步和视频抖动平滑

音视频同步在视频会议系统中就是所谓的唇形同步，指的就是口形和话音是否一致。视频抖动指的是，在视频显示时画面不流畅、时快时慢。两个问题的根源都在于网络传输过程中 RTP 包到达的时间间隔不均匀，这是 IP 网络本身固有的问题。解决问题依赖于音视频接收端的播放速率控制机制，本文提出了基于视频 RTP 时间戳驱动的显示速率控制和语音采样频率驱动的播放速率控制的解决方法。

本文提出的同步方法要点就是音视频各自同步到同一物理时钟，但同步物理时钟的方法不同，视频基于 RTP 时间戳，音频基于取样频率。将时间戳和时钟频率结合起来，而不是笼而统之地一律用时间戳原因在于，音频播放是硬件时钟频率驱动的，例如 8k 采样率语音，其播放时样点取样频率一定是 8k，该精度是硬件时钟来保障的，容不得有一点误差，无论采样什么方法来同步语音，其落脚点都是控制解码器和固定时钟频率的硬件播放中断服务程序间缓冲区的缓冲时延，在缓冲时延门限固定的情况下(几百毫秒)，只要延时没有超过这个门限，任何解码的语音数据都应该被缓冲区接收，只要超过了才丢弃；相反，一旦播放时钟中断(一般是几十毫秒中断一次)要从缓冲区读数据，即使此时缓冲区是空的，应用程序也必须人为地填充数据来保障语音 D/A 变换同步电路每个时钟周期都有数据(一般是填零信号，也可以研究专用的填充算法)。因此，语音同步即使也采用 RTP 时间戳，其最终依然要转化到取样频率上，既然这样，语音同步不如直接用取样频率来做，因为时间戳在转化为时钟频率时会出现处理精度损失。对于视频，用 RTP 时间戳同步是唯一的选择，因为视频不存在固定的帧率，其编码帧率往往本身就是变化的。

首先，为便于算法描述，定义如下变量：物理时间复位标记 time_reset_flag，参考起始时间戳 start_vid_pts，参考起始物理时间 start_vid_clktime，当前时间戳 pts_time，当前物理时间 clk_time，时间戳间隔 pts_interval，物理时间间隔 clk_interval，等待时间 wait_time。视频同步方法基本原则是“快等慢赶”，另外一个要点是时间戳不能单帧计算，而是多帧自适应补偿，适应周期直到遇到一次时钟复位为止，算法用如下步骤描述：

(1)初始化

start_vid_clktime=0; start_vid_pts=0;
time_reset_flag=1;

(2)令 pts_time 等于当前视频帧 RTP 时间戳，clk_time 等于当前物理时间，

if(time_reset_flag) then
start_vid_pts = myTime; start_vid_clktime = clk_time; time_reset_flag=0;
endif
pts_interval = pts_time-start_vid_pts, clk_interval = clk_time-start_vid_clktime;

因为物理时间可能因表示精度的原因出现时间计时器转弯，故

if(clk_interval<0) then clk_interval = pts_interval; time_reset_flag=1; endif

因为时间戳可能因表示精度的原因出现时间计数器转弯，故

if(pts_interval <0) then clk_interval= pts_interval; time_reset_flag=1; endif
wait_time = pts_interval-clk_interval;
if(wait_time>0) 执行步骤(3)
else if(wait_time< CLK_EXCEPTION_THRESHOLD) 执行步骤(4)
endif
else 执行步骤(5)endif;

(3)此种情况视频包接收、处理过快，超前于时间戳，故等待差距时间；

sleep (wait_time); 转到(6)；

(4)此种情况，时间戳滞后于物理时间太多，可能是其它原因引起的，如系统暂停、CPU 资源抢占等，故复位物理时钟；

time_reset_flag=1; 转到(6)；

(5)此种情况，视频包到达滞后于物理时间，意味着显示时间轴需要往前赶；

(6)显示解码出的视频帧；

(7)继续回到步骤(2)，直到系统退出。

对于语音，借助取样频率来达到同步目的则方法非常简单：如果播放缓冲区样点数小于延时门限(本文设成 200ms)，

表 2　视频子系统性能技术指标对比测试

测试项	128kbps		256kbps		320kbps	
	本文	BVP8770	本文	BVP8770	本文	BVP8770
帧率统计	12fps	13fps	18fps	19fps	20fps	20fps
画面节奏连贯性	可	可	良	良	优	优
清晰度	优	可	优	可	优	良
马赛克现象	优	可	良	可	良	良
画面停顿现象	良	良	良	良	可	可
时延(ms)	<500	<500	<500	<500	<500	<500
唇形同步	优	优	优	优	优	优
网络差错恢复	快	一般	快	一般	快	一般

则解码数据写入缓冲区，否则丢弃。

4　实验结果

测试方法：中国电信 512k ADSL 接入，多用户共享(同时有多人上网)，一天分上午、下午、晚上 3 次测试，每次测试 3h，共测试 1 周，测试带宽设置 128kbps，256kbps，320kbps 等 3 个档次。记录测试结果，客观指标统计得到，主观指标按优、良、可、差、坏 MOS 评分。参与对比测试是 LeadTek BVP8770 可视电话终端，将实验结果处理后列于表 2。

需要说明的是，画面节奏连贯性反映实际主观的画面连贯性，因如果画面抖动厉害，则帧率不能反映实际的连贯性，帧率仅是统计意义上的画面连贯性；画面停顿系解码器请求编码器刷新，重新锁定 INTRA 帧的原因。实验表明本文视频子系统性能在视频画质和对网络的适应性方面明显超出 LeadTek BVP8770，但在连贯性和唇形同步上基本相当；特别值得一提的是，在网络出现差错时，本文视频可以很快恢复到正常的视频表现，不会长时间停留在马赛克或图像破碎画面，而 BVP8770 则做不到，这主要得益于基于包的带差错掩蔽功能的解码器设计和精心设计的 INTRA 重同步策略。

5　结束语

从技术路线上考虑，本文提出的几个问题是研制 H.323 视频会议系统最难以回避的核心问题，另外，一个可实用的 H.323 视频会议系统在视频方面还必须考虑到其它技术难度，例如视频编解码器计算复杂度优化，除了从快速运动估计搜索、快速 DCT 变换等快速算法上降低计算复杂度以外，在具有多媒体计算能力的处理器上，如何结合处理器的硬件计算体系结构特点进行优化是一个极有前途的方向，本文在指令级并行、数据级并行、高速缓存的优化、存贮器数据流量的优化等方法上也进行了深入的研究。

通过提出的若干创新性方法对这些问题的逐一解决，笔者参与研制的 H.323 视频会议系统在视频性能技术指标上有出色的表现，但考虑到IP网络的复杂性，结合视频通信的信道特点对视频编码进行率失真优化[7]越来越引起视频通信理论研究者的兴趣，例如网络丢包或误码条件下的码率控制、运动估计、编码模式判决等的率失真模型。这是笔者下一步研究的目标。

参 考 文 献

[1] ITU-T Recommendation H.323(02/98), Packet-based multimedia communications systems.

[2] ITU-T Recommendation H.263(02/98), Video coding for low bit rate communication.

[3] Tourapis A M, Au O C, and Liou M L. Fast block-matching motion estimation using predictive motion vector field adaptive search technique (PMVFAST). ISO/IEC JTC1/SC29/WG11 M5866, Noordwijkerhout (Netherlands), March 2000.

[4] Tiehan Lv, Burak Ozer, and Wayne Wolf. Exploiting Parallelism in Media Processing Using VLIW Processor. ICIP 2003, Barcelona, Catalonia, Spain, 2003, 3: 97–100.

[5] ITU-Telecommunications Standardization Sector, STUDY GROUP 16,Video Coding Experts Group, Title: Video Codec Test Model, Near-Term, Version 8 (TMN8), Portland, 24-27 June 1997.

[6] Worrall S, Sadka A H, Sweeney P, and Kondoz A M. Motion adaptive INTRA refresh for MPEG-4. *IEEE Electronics Letters*, 2000, 36(23): 1924–1925.

[7] Peng Z, *et al*.. On the trade-off between source and channel coding rates for image transmission.Proc. of the IEEE International Conference on Image Processing, Chicago, Illinois, 1998: 118–121.

王中元：　男，1972 年生，博士生，从事多媒体信息处理、多媒体网络通信等方面研究.

胡瑞敏：　男，1965 年生，教授，博士生导师，从事多媒体网络通信、多媒体信息处理和编码等方面研究.