

易错环境中视频传输空域差错恢复的优化方法

谭 歆¹ 杨茂斌¹ 张 希¹ 刘 萍² 尹 刚³

(重庆邮电学院通信与信息工程学院 重庆400065)¹ (四川联通雅安分公司市场部 四川雅安625000)²
(重庆移动通信规划设计有限责任公司 重庆400041)³

摘要 在易错环境如因特网、无线通信网中传输视频流,误码率相当高。讨论了一种有效的在易错网中传输视频信号时,提高空域差错恢复能力的方法。同步标记可限制空域差错在视频帧之间传播。但是,如果其使用太频繁,会导致不必要的开销。为优化同步标记插入过程,给出一个同步标记插入算法。最后,为验证该算法,提供了仿真实验结果。

关键词 视频传输,空域误码,恢复

Spatial Error Resilience Optimization of Video Transmission in Error-prone Environment

TAN Xin¹ YANG Mao-Bin¹ ZHANG Xi¹ LIU Ping² YIN Gang³

(Department of Communication and Information Engineering, Chongqing Post and Telecommunication College, Chongqing 400065)¹
(Market Department, Shichuan Ya'an Branch Co., China Unicom Co., Ltd, Ya'an 625000)²
(Design of Chongqing Mobile Communication Co., Ltd, Chongqing 400041)³

Abstract Bit error rates will be quite high when transmit video coding in error-prone environments such as Internet and wireless communication network etc. Describe an effective method for increasing the spatial error recovery of video transmission over bit error prone networks. Synchronization markers limit the propagation of spatial errors within video frames. However, if used too frequently, it will introduce an unnecessary increase in bit rate. In order to optimize placement of synchronization markers, the synchronization marker insertion algorithms are introduced. Finally, for the purpose of verifying this method, experimental results are presented.

Keywords Video transmission, Spatial error, Recovery

1 引言

对于那些由一个或多个具有独立处理能力的计算机所构成的自主系统(Autonomous System)而言,由它们所构成的采用报文分组传送的网络,特别是基于互连网协议(IP—Internet Protocol)的报文分组(简称IP包)传送的网络中,各自主系统之间的连接采用的是IP协议。IP协议是面向无连接的,它仅仅能提供尽力“发送服务”(Best-effort)。这种尽力发送的机制不提供服务质量保证,因而网络层无法保证一些传输参数,如包丢失率(Loss)、带宽(Bandwidth)、端到端延迟(Delay)、延迟抖动(Delay Jetter)等能得到良好的控制,而这些参数对于某些应用业务,特别是视频实时网络传输应用往往是至关重要的。因此视频内容经过诸如Internet、移动网络等易错网传输后其质量就不能够得到有效的保证。当然,在这样的易错环境中传输视频流时,要想在接收端得到效果较好的再生图像,就必须对传输过程中产生的差错进行控制及恢复。本文引入信息隐藏技术,给出一种易错环境中的视频传输空域差错恢复的改进方法,并进行了仿真。

2 视频编码标准

多媒体编码技术的蓬勃发展,造就了产业化的光明前景,而产业化的大规模实施则是以标准化为前提的。因此,压缩编码技术的各种国际标准也就被制定、实施以及不断地更新、发

展。现在针对视频编码的国际标准主要有两大系列,即ISO(International Standardization Organization)所属的MPEG(Moving Picture Experts Group)系列与ITU(International Telecommunication Union)所属的H.26X系列,MPEG系列主要是针对运动图像的压缩进行研究,而H.26X系列主要是针对可视电话与视频会议领域进行压缩研究,下面分别予以概述。

MPEG-1(ISO IS 11172)采用基于DCT(Discrete Cosine Transform)的混合块运动补偿编码^[18],编码结果为单层位流,压缩位率约为1.5Mbps,主要应用于VCD等数字化存储介质,对SIF(Sequencing Intermediate Format)逐行格式(Progressive Format)进行了优化。

MPEG-2(ISO IS 13818 ITU-T H.262)也是基于DCT的混合块运动补偿编码,但更为复杂,更具有通用性。它首次定义了位流结构层次语法,其GOP(Groups of Pictues)可被单独解码。为了完整语体系及扩展,首次提出了框架及级别概念,它包含了通用的视频编码算法^[1,7]。主要应用于数字化电视HDTV(High-Definition Television)能及DVD等,输入格式支持4:2:0、4:2:2等,支持逐行扫描与隔行扫描。

MPEG-4(ISO IS 14496)主要是基于对象编码标准,其运动补偿算法仍是其于DCT。它包含很多新技术与工具,如:对任意形状对象等的支持,其中最引人注目的是FGS精细粒度编码技术被吸纳在标准中,并可处理自然与合成的音视频。

谭 歆 硕士,研究方向为网络视频流传输;杨茂斌 硕士,研究方向为视频编码及压缩;张 希 硕士,研究方向为网络传输技术;刘 萍 工程师,研究方向为电子商务及移动网络视频传输;尹 刚 工程师,研究方向为网络视频流传输。

MPEG-7标准主要是为了解决对多媒体信息的检索,称之为多媒体内容描述接口。此外,MPEG正在筹备制定的MPEG-21标准,其期望目标是从消费者的角度上定义多媒体应用框架结构,而不仅仅是个多媒体压缩标准,具体细节正在讨论中。

H. 261 (ITU-T H. 261)是基于DCT的混合块进行运动补偿编码的,并只使用了I帧与P帧,以整数像素为单位进行运动补偿,主要用于综合业务数字网ISDN(Integrated Service Digital Network)视频会议,支持逐行视频格式:QCIF(Quarter Common Intermediate Format)与CIF(Common Intermediate Format),其压缩位率大约在64kbps~2Mbps,现正被H. 263(ITU-T H. 263)取代。

H. 262与MPEG-2相同^[7],主要用于ATM/宽带视频会议。H. 263基本与H. 261一样但更为先进,其运动补偿精度更高1/2像素。为了提高编码效率增加了PB帧,对非零DCT系数进行哈夫曼(Huffman)编码,每个宏块(Macroblock)有1到4个运动矢量,主要用于POTS(Plain Old Telephone Service, i. e., analog phone line)以及PSTN与ISDN上的视频会议,其压缩位率大约在16kbps~2Mbps,支持输入格式:Sub-QCIF到16CIF^[7]。随后的H. 263+与H. 263++都是H. 263的改进,主要表现为对帧内编码加强了隔行扫描(Alternate scanning),对帧间编码使用了两个可变长编码表(VLC table: Variable Length Coding)。

最新的标准H. 26L(ITU-T H. 26L)采用了基于块的混合运动补偿及包含更多位的字,运动补偿精度高达1/8像素,采用了整数变换编码及统一VLC与自适应算术编码,其目标压缩位率为:8kbps~20Mbps,目前仍处于制订过程中,还未大量应用于实践。

在这些标准中,都没有对视频编码器进行定义,主要是为了保持良好的标准间的兼容性与标准良好的扩展性。它只定

义了一些常用的核心算法以及被一些标准测试位流验证了的并得到MPEG组织认可的一些编码技术与方法,主要是从语法与语义的角度定义了解码器的解码语法。因此,可以有各种各样的编码器结构与编码方法,但其最终编码的结果一即视频位流必须符合标准所规定的解码语法与语义,并可用标准的解码器进行解码显示。在对其作研究性的工作时,也可在标准解码语法的基础上,实现自己的编码方法,如果有必要,也可对标准的解码语法作适当修改以适合自己所提出的编码方法。

3 视频编码的基本内容

视频编码分为帧内编码与帧间预测编码,对于帧内编码的目的主要是去除空间域冗余,对帧间进行预测编码的目的主要是去除时间域冗余,将视频进行有损编码及时域可分级编码主要是利用了人的视觉特性。

视频编码时通常将视频划分成多个宏块,对于每个宏块选则不同的编码模式。如果是帧内编码模式,则将宏块再划为4个块,对每个块进行DCT变换,然后将DCT体系数进行量化,对量化后的系数进行熵编码。为了能进行下一帧的帧间编码,在将DCT系数进行量化后,要作一系列的逆过程,先进行反量化IQ,再对DCT进行反变换IDCT以构成该帧的重构帧对下一帧进行运动预测。

如果是帧间预测模式编码,则首先产生宏块的运动补偿预测值,进行差分编码DPCM将当前宏块的实际数据减去预测值得到预测误差信号。然后将该宏块的预测误差分成4个8×8的块,与帧内编码模式一样进行DCT变换,再对系数进行量化,最后,对量化后的系数进行熵编码。

3.1 信源编码算法

信源编码算法的一般形式如图1所示,主要由帧间预测、帧内分块和量化组成。

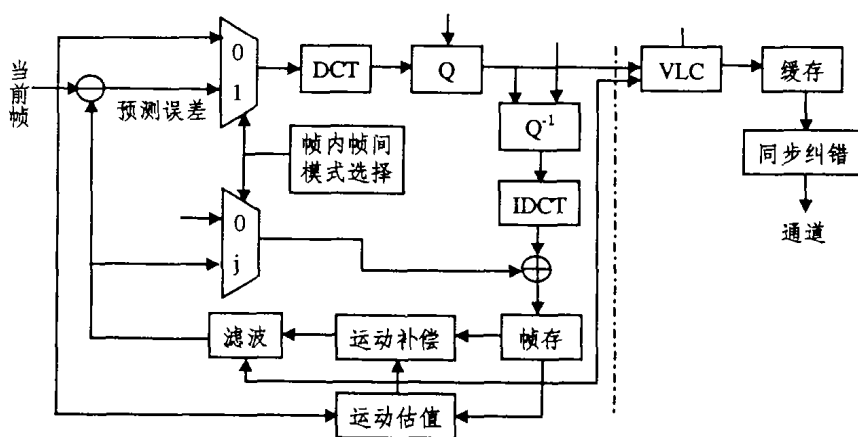


图1 通用活动图像编码框图

首先将预测误差或输入图像划分成为8×8的像素块。进一步,将四个亮度像块和两个在空间位置上与之行营的色差像块符合成一个16×16的宏块(MB)。信源编码器对每秒出现30000/1001=29.97次的非隔行扫描画面进行运算。画面按一个来年高度分量和两个色差分量进行编码。

对于帧序列中的第一幅图像或景物变换后的第一幅图像,采用帧内变换编码:利用8×8的DCT实现。各DCT系数经过线性量化、变长编码后进入缓冲器,根据缓冲器的上溢和下溢,来反馈调节量化器的量化步长,以控制视频编码位流使之与信道速率相匹配。帧间预测采用混合方法:利用运动补偿

预测,当预测误差超过某个门限后,对误差做DCT、视觉加权量化(必要时可经过递归滤波器滤除高频噪声,以改善图像质量)及熵编码。运动矢量信息编码后也送到缓冲器中。

DCT去除空间冗余度,而使用有运动补偿的帧间预测来去除时间上的冗余。这是一个典型的帧内/帧间自适应预测加DCT变换的混合算法。在帧间模式时,两个开关都打到下面。输入与经过运动补偿的前帧信号相减,差值经DCT,量化并进行可变字长编码后输出。从图中可以看到,在反馈环中有一个解码器即先反量化,反变化,然后再加上经运动补偿的前一帧,作为下一帧的预测信号。在帧内模式时,开关都打到上面。

这时输入信号直接跳过减法器,经DCT量化和VLC输出。这时运动检测和运动补偿不再起作用,只作帧内处理。

帧内编码的DCT直流系数仅用一个量化器,而交流系数则采用31个量化器。在一个宏块内,除了帧内直流系数外,所有的系数均采用一个量化器。量化器的判决电平并没有明确规定。

对于帧内编码模式,帧内的六个像块的系数均要传输;而在其他情况下,有宏块的类型和编码块接收来决定哪个像块要发送。量化后的系数按ZigZag方法读出后,最后再加一个像数块结束标记EOB后发送。数据读出时,按游程长度编码,最常见的方法是按连零个数(Run)和随后的系数值(Level)作可变字长编码。另一种是20比特固定长度码字,6比特退出码(Escape),6比特连零数(Run)和8比特系数值(Level)。

运动补偿用于帧间编码;运动估值找到最佳匹配的云矢量,计算预测误差,这是帧间编码的基本过程;滤波方式采用环路滤波器,消除高频噪声。为了避免变化系数幅值的量化失真在编码器和解码器环路内引起算术溢出,插入一个限幅功能。对于重建图像,它简单地将超出0~255范围的像素值限制在这期间。

3.2 图像复用编码

源编码后进行频分复用编码,实际上是把比特流分成图像(Picture),像块组(GOB:Group of blocks),宏块(MB:Macroblock)和像块(Block),并附加相应的信息。按照CIF格式,每帧CIF图像包含12个GOB,每个GOB包含33个MB,每个MB包含4个亮度数据块和各1个Cb、Cr色度块,每个B包含8*8像素GOB和MB都是以一定的顺序存放的。

图像复用编码器把上述层次的数据按照一定的方式连列起来,就构成了1帧数据流。编码图如下:

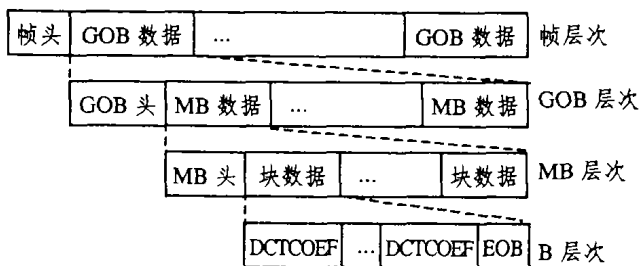


图2 数据流结构

3.3 传输缓冲器

传输缓冲器的目的是协调编码输出率和传输网络位率,使得充分利用网络的传输位率。当网络传输位率过高时,缓冲区用做存储器,同时通知编码器提高量化步长,从而降低位率;当编码器位率过低时,缓冲器根据存储余量通知编码器降低量化步长提高位率。

3.4 传输编码器

传输编码器的作用是传输前进行纠错编码,目的是当图像码字在传输过程中出现误差时,则在解码中予以纠正。这里使用的纠错码为BCH码(循环冗余校验码)。

3.5 视频帧的空间域统计特性

视频帧作为一幅图像它的空间域统计特性主要体现在像素取值的概率分布和标志像素间关系的相关函数或条件概率等表示。为了简化起见,下面通过图像的灰度直方图来对其进行介绍。

灰度直方图是一个表示一幅数字图像中具有每个灰度级

的像素数的函数,通常用离散的曲线图或条形图表示,例如,图3是一幅取值范围为[0,255]的灰度图像。图的左半部分为原始图形,右半部为其灰度直方图。如果用 $H(f)$ 表示灰度直方向,则图像中像取值为 f 的频率就是 $h(f)$ 除以图像中像素的总数,通常可用该频率近似代替图像中像素的取值概率 $P(f)$ 。

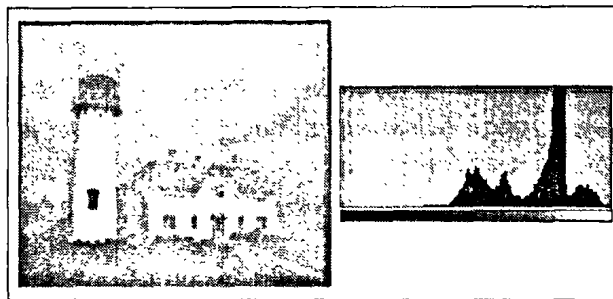


图3 图片及其直方图

通过对其进行观察,我们发现,相邻或相近的像素,其灰度值总是相近,或者说,一个像素的取值通常趋向于和它周围的像素取相同的值,这种现象,图像的像素间存在着很强的相关性。这种相关性,通常可以通过图像的相关函数、条件概率和差值分布等进行研究。例如,将图像中相邻的两个像素相减,所得差值的概率分布集中在0附近。

图像的这种空间域统计特性,对图像的变换提供了理论依据。因为图像的空间域与频率域有着对应关系。在频率域上,图像表现为不同频率分量系数的分布。空间域中的强相关性,即图像中存在量的平坦区域,反映在频率域中,就是图像的能量集在低于低频部分。通过常用的一些正交变换如:离散余弦变换、小波正交变换等可去除它的空间冗余,从而减少表达图像的数据,起到压缩作用。需要重新表达这些图像时,就对这些变换进行逆变换,将变换后的数据进行还原操作。如果在数学上不能进行严格的逆变换则该压缩为有损压缩,否则为无损压缩。

4 空域差错恢复的优化方法

利用同步标记,目前的视频编码标准支持空域差错定位。这一技术可限制误码在帧间传播。视频传输中比特差错可能导致两种同步错误:解码器解码视频流时可能失去比特同步;解码器重现视频帧时可能失去空域同步。使用同步标记可以解决这样的问题,它使解码器可与比特流一直保持比特同步(失步时重置被解码帧的空间位置)^[2],同时阻止空域预测差错的传播。但是,如果其使用过于频繁,会导致比特率不必要的增加,加大系统开销。以下给出一个同步标记插入算法,对此过程予以优化,从而提高系统的空域差错恢复能力。

视频流的一个切片(Slice)理论上可包含任意数目的已编码视频块。在可变长码比特流中,切片头提供一个唯一的同步点。我们假定切片中任意位置发生错误,整个切片将被丢弃。

首先给出两个参数:切片长度 $L(s)$ 和信道比特差错率 P_e ,以此计算切片差错率SER(Slice Error Rate),并以此作为差错产生的概率 P 。我们可得下式:

$$SER = P = 1 - (1 - P_e)^{L(s)} \quad (1)$$

假设块中比特数为均匀分布,切片长度的数学期望(以块(block)为单位)为 $E[l_{block}(s)]$,它是 P 的期望 $E[P]$ 的函数,即:

$$E[L_{block}(s)] = \frac{\log(1-E[P])}{\log(1-P_s)} \times \frac{1}{L_{bit}(block)} \quad (2)$$

式中 $L_{bit}(block)$ 代表视频块的平均比特长度, 其受限于比特率。

当 SER 超过 20% 时, 接收端视频再生的信噪比将低于 20dB, 这会导致非常糟糕的视频效果, 为形象表达 SER 与 $L(s)$ 的关系, 我们得到如图 4 的实验结果。

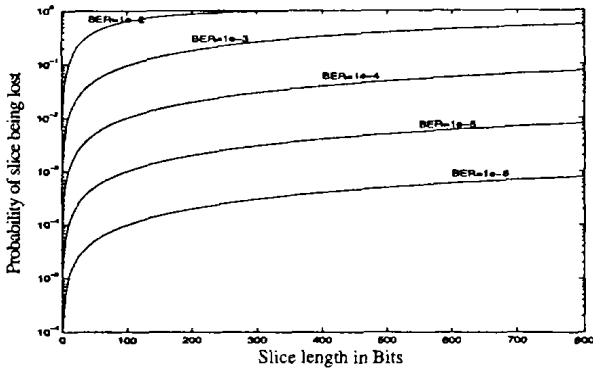


图4 SER 与切片长度的关系

为估计一个切片中能包含多少个视频块, 分别对于帧间编码模式和帧内编码模式也进行了结果如图 5 和图 6 的实验, 实验中我们得到, 平均来说, 一个切片最大可包含大约 5 个帧间编码块 (每块 45 比特) 或者 2 个帧内编码块 (每块 112 比特)。

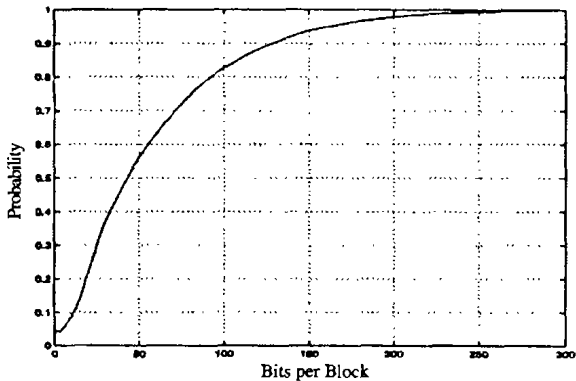


图5 帧间编码块与差错率

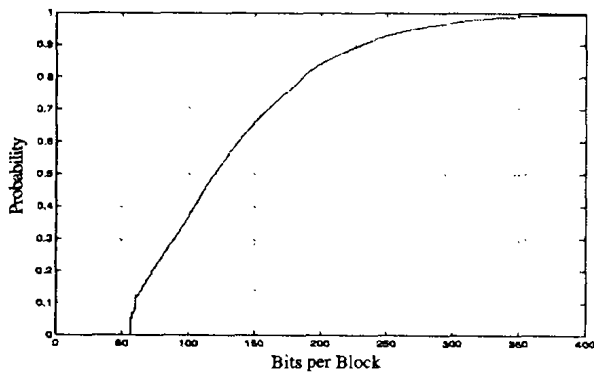


图6 帧内编码块与差错率

在拉格朗日最小化模式中, 必须考虑切片中同步标记的码率。插入同步标记时, 如果其比特率有一个较小的提高, 解码器端的视频再生质量就会得到巨大提升。因此, 我们给出一个拉格朗日开销函数, 用于切片头定位:

$$J_{sync}(n, (x, y)) = D(n, (x, y), \text{mod } e, p, sync) + \lambda_{sync} R(sync) \quad (3)$$

式中, $sync \in \{0, 1\}$, 0 代表切片中没有插入同步头, 1 代表插入了同步头, 解码器中的失真 D 归结为两类: 由某一已经接收的视频块 (可能来自对某一已隐藏的块的预测) 所产生的失真 $D1$; 由某一已丢失并被隐藏的块所产生的失真 $D2$ 。插入切片头导致的同步码率, 会产生多重码率开销, 这就需要在插入同步标记和提升空域差错恢复能力之间作出平衡, 进行优化。在最小化模式 J_{sync} 下, 在 $R(sync)$ 一个切片中各块中分布, 其增加切片的码率。为得到不同信道比特差错率 BER 下的拉格朗日参数 λ_{sync} 的优化值, 我们进行了结果如图 5 的实验, 图中显示了切片长度 (以已编码块为单位) 与 λ_{sync} 的关系。实验中码率为 48 kbps, 空域解析为 QCIF。

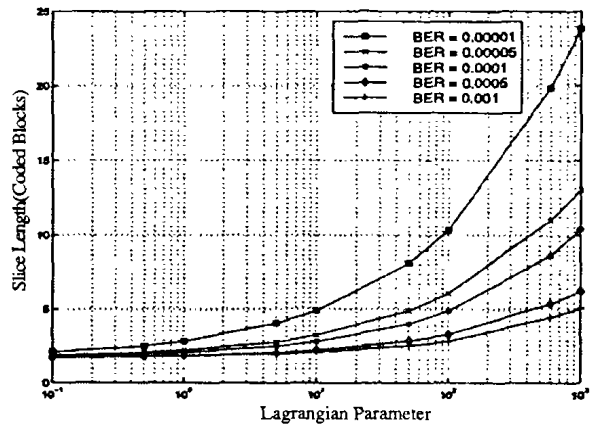


图7 λ_{sync} 与切片长度的关系

由图 7 可见, BER 较高时, λ_{sync} 对切片长度影响较小, 而 BER 较低时则影响显著。因此我们可以考察 λ_{sync} 与 BER 的关系。同样, 设比特在各视频块中均匀分布, 同时在块差错率的数学期望 $E[P]$ 小于 5% 的条件下, 通过实验, 我们给出 λ_{sync} 计算的经验公式:

$$\lambda_{sync} = \beta \times \max(-10^7 P_s + 1200, 1) \quad (4)$$

其中函数 $\max(x, y)$ 代表取 x, y 中的最大值。 β 代表一帧中被编码的块的比例, 它是以前所有帧中已编码块的平均值的函数, 随每一已编码帧变化。

5 优化方法的仿真实验

对优化方法的仿真, 我们采用 H. 223 仿真程序^[7], 视频信号的空域解析度为 QCIF (176x144 像素)。我们对编码中切片同步头插入使用本文所给出的算法与未使用该算法的效果。仿真结果见图 8。图中横坐标为比特差错率 BER (bit-error rates), 纵坐标为亮度信噪比 (Y-PSNR)。

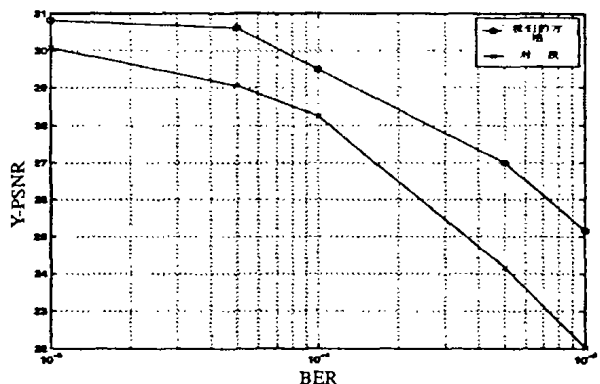


图8 仿真结果

由图 8 可见, 在相同空域空域解析度条件下, 相应于同一 (下转第 218 页)

A)把所记录的软件缺陷按软件开发的各个阶段进行分类,选择序号刻度作为刻度类型;然后统计各类缺陷的数目,选择绝对刻度作为刻度类型。

通过对图1的分析,按照 Pareto 守则可以发现:缺陷主要来自“设计”阶段。

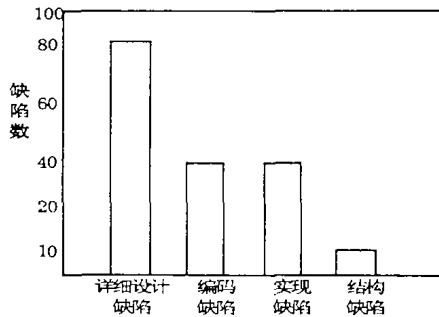


图1 编译程序缺陷类型

B)对“设计”阶段的错误进行更详细地分析。对设计阶段的缺陷类型细分,选择序号刻度作为刻度类型;然后统计细分缺陷的数目,选择绝对刻度作为刻度类型。结果图2所示。

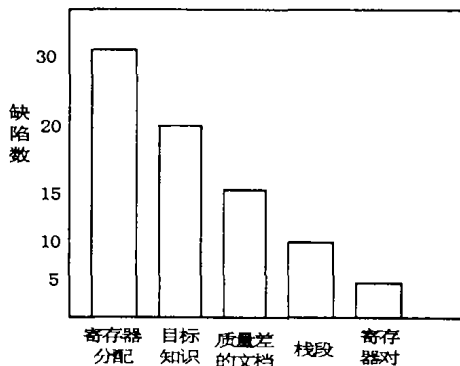


图2 编译程序设计缺陷类型

C)分析缺陷产生之原因。如寄存器分配缺陷如图3所示。

D)找到了上述原因之后,在后续的项目开始之前,采取积极的步骤对员工进行相关培训,借此来改进随后的软件项目的开发工作。

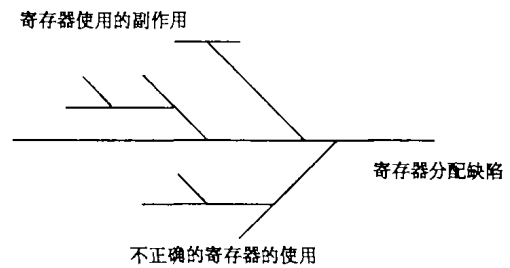


图3 因果图

小结 不同的软件测量刻度对测量意义的判定起决定作用,同时测量刻度亦影响在数字系统上所进行的数学运算。因此在建立测量环境时,对测量刻度的选择是非常重要的。

参考文献

- 1 Mari L. The meaning of "quantity" in measurement, Measurement, 1996, 17: 127~138
- 2 Muravyov S V, Savolainen V. Special interpretation of formal measurement scales for the case of multiple heterogeneous properties, Measurement, 2002, 29: 209~223
- 3 Fenton N E. Software Metrics: A Rigorous & Practical Approach. 清华大学出版社, 2003
- 4 McGarry J, Card D, Jones C, et al. Practical software Measurement: Objective Information for Decision Makers. Pearson Education, Inc., 2001
- 5 Fenton N E. Software metrics: successes, failures and new directions. The Journal of Systems and Software, 1999, 47: 149~157
- 6 Cropley D H. Towards formulating a semiotic theory of measurement information, measurement, 1998, 24: 237~262
- 7 Muravyov S V, Savolainen V. Representation theory treatment of measurement semantics for ratio, ordinal and nominal scales, Measurement, 1997, 22: 37~46

(上接第119页)

比特差错率 BER,使用本文所给出的算法,空域误码有相应的改善,视频流的亮度信噪比 Y-PSNR 有明显的提高。

结论 视频信息在易错环境如 Internet、移动网络中传输时,需对其空域和时域差错进行有效控制,否则,接收端得到的视频再生质量将难以接受。

利用同步标记对空域差错定位,其阻止空域预测差错的传播。但是,如果其使用过于频繁,会加大系统开销。本文给出的同步标记插入算法,对此过程予以优化,改善了系统的空域差错恢复能力,提高了接收端再现图像的质量。

参考文献

- 1 Talluri R. Error resilient video coding in the MPEG-4 standard. IEEE Commun. Mag., 1998, 26: 112~119
- 2 ITU T Rec H. 263, Version 2. Video coding for low bitrate communication. Jan. 1998
- 3 Wenger S, Knorr G, Ott J, et al. Error resilience support in H. 263+. IEEE Trans. Circuits Syst. Video Technol., 1998, 8: 867~877
- 4 Girod B, Farber N. Feedback-based error control for mobile video transmission. In: Proc. IEEE, Special Issue Video Mobile Multi-

media, 1999, . to appear

- 5 Wieg T, Lightstone M, Mukherjee D, et al. Rate-distortion optimized mode selection for very low bit rate video coding and the emerging H. 263 standard. IEEE Trans. Circuits Syst. Video Technol., 1996, 6: 182~190
- 6 Schuster A, Katsaggelos A. Fast and efficient mode and quantizer selection in the rate-distortion sense for H. 263. In: SPIE Proc. Visual Commun. Image Processing, vol. 2727, 1996. 784~795
- 7 Sun H, Kwok W. Concealment of damaged block transform coded images using projection onto convex sets. IEEE Trans. Image Processing, 1995, 4: 470~477
- 8 Ohm J-R, van der Schar M. Scalable Video Coding. Tutorial Material SM. T3, IEEE Intl. Conf. in Image Processing ICIP'2001, Thessaloniki, Greece, 2001
- 9 ISO/IEC IS 13818-2 / ITU-T Rec. H. 262. Generic coding of moving pictures and associated audio, part 2: video. Nov. 1994
- 10 Haskell B G, Puri A, Netravali A N. Digital video, an introduction to MPEG-2. New York: Chapman & Hall, Sep. 1996
- 11 Skarbek W. Multimedia. Algorytmy i standardy kompresji. Akademicka Oficyna Wydawnicza PLJ, Warszawa, 1998
- 12 Li W. Fine Granularity Scalability Using Bit-Plane Coding of DCT Coefficients. ISO/IEC JTC1/SC29/WG11, MPEG98/M4204, Dec. 1998
- 13 Li W. Overview of Fine Granularity Scalability in MPEG-4 Video Standard. IEEE Trans. on Circuits and Systems for Video Technology, 2001, 11(3): 301~317