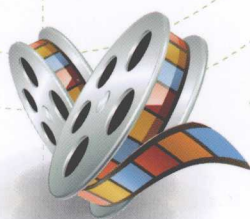




包含书中所有案例的源文件

Broadview[®]
www.broadview.com.cn

MPEG-4/H.264



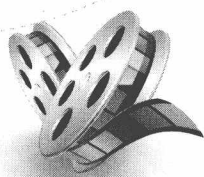
视频编解码 工程实践

路锦正 ©编著 |



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

MPEG-4/H.264



视频编解码 工程实践



路锦正 ©编著 |



YZLI0890106587

电子工业出版社

Publishing House of Electronics Industry

北京•BEIJING

内 容 简 介

数字视频编解码已经形成了一个巨大的产业,视频编解码算法是其技术核心。本书介绍了基于Visual C++2005及TI DSP平台进行视频算法优化的实现过程和技术难点,并提供了书中所有源代码。主要内容包括:视频编解码原理及视频图像的捕获与显示、基于Visual C++2005分析MPEG-4/H.264编解码原理、基于TI DSP设计和优化开源工程(Xvid/x264/ffmpeg)、视频算法的标准接口xDAIS开发和视频监控中心软件设计等。

本书中的案例均来源于笔者开发中的工程实践,具有很强的实用价值。读者既可以基于案例进行特定的开发,也可以直接应用。

附书光盘包含书中所有案例的源文件。

本书内容充实,工程实用性强。面向高年级本科生、工程/专业/学术硕士、算法工程师等致力于视频编解码算法开发的人员,为其课程设计、毕业设计和项目研发等提供实战指南和案例参考。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

MPEG-4/H.264 视频编解码工程实践 / 路锦正编著. —北京: 电子工业出版社, 2011.11
ISBN 978-7-121-14787-6

I. ①M… II. ①路… III. ①视频编解码器 IV. ①TN762

中国版本图书馆 CIP 数据核字 (2011) 第 206069 号

责任编辑: 徐津平

印 刷: 北京天宇星印刷厂

装 订: 三河市鹏成印业有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编: 100036

开 本: 787×1092 1/16 印张: 24 字数: 499 千字

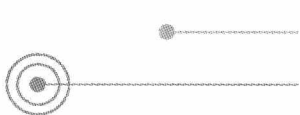
印 次: 2011 年 11 月第 1 次印刷

印 数: 4000 册 定价: 59.00 元 (含光盘 1 张)

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。



前言



视频的编解码是数字视频开发应用中的最主要方面，海量的视频数据为便于存储和传输、节约资源，必须进行一定程度的编码压缩。视频编解码在数字多媒体应用中占有极为重要的比重，已经形成了一个巨大的产业。从 VCD、DVD 到现在的 MP4、IPTV、硬盘录像机、可视电话、高清电视等；从早期的闭路电视监控 CCTV（Closed Circuit Television）到现在的数字视频录像 DVR（Digital Video Recorder）等。编解码算法是上述系统中的核心技术，因此掌握自主产权的视频算法就极其重要。

由于视频编解码原理较复杂，涉及了许多数学知识——从数据的 DCT/DWT 变换，到基于图像块的运动估计和运动补偿，均匀量化和非均匀量化技术，还有可变长度熵编码如 Huffman 编码和算术编码等。视频编解码协议复杂，系统代码量大，功能模块多。幸运的是，大量开源视频算法为我们学习和实践视频算法提供了实用的算法工程。主流视频编解码算法中当数 MPEG-4/H.264 算法标准，但其协议标准仅规定了码流语义，并没有明确如何进行工程实现。Xvid 是公认优秀的 MPEG-4 开源视频编、解码算法工程，X264 是公认优秀的 H.264 开源视频编码算法工程。这些开源算法一方面主要采用跨平台的 C 语言编程，另一方面做核心模块面向常用 CPU（Intel/AMD）的多媒体汇编指令做了系列优化。

基于可编程芯片（如 DSP/FPGA）实时实现视频编解码算法是一件非常具有挑战性的工作，一方面需要对复杂算法本身深入理解；另一方面需要对研发平台的数据处理特点及汇编指令等有相当经验的把握。但对于一般的高年级本科生甚至研究生来说，同时熟练掌握算法和实现平台可能就不是一件容易的事情了。笔者以自己多年从事音视频编解码算法开发的亲身体会出发，总结开发技术要点、凝练开发技巧、分享实战项目经验，为致力于视频编解码算法优化的研究开发人员提供指南和参考。

从目前市面上有关视频编解码的书籍来看，原理介绍性的多，工程实用性的少；技术规格翻译的多，项目实战经验共享的少。笔者几年前就想编写一本视频编解码实用性强的著作。正是基于上述各种因素的分析，终于催生了本书。笔者期待读者快速步入视频编解码开发的技术殿堂。

同时，笔者也真诚期望本书能够抛砖引玉，与相关技术人员共同探讨和学习，信息共享、共同进步。

读者对象

- 高年级本科生在毕业设计阶段，其工程能力弱，时间短任务重，迫切需要该类指南；
- 我国硕士研究生，特别是工程硕士，实际在校时间短，理论基础相对薄弱而工程能力较强，迫切需要项目实践经验参考；
- 专业/学术硕士的课题一方面有视频编解码的直接应用，另一方面又有基于视频编解码的其他应用，如目标识别或跟踪，实用工程案例可提供支持；
- 公司企业的算法优化工程师、新员工，短期内掌握并提高质量完成视频算法优化技术难度较大，而视频算法是系统核心技术，为提升员工自身价值需掌握核心算法优化技术。

本书约定

本书所使用的软件有：Windows XP (SP3)、Visual C++ 2005 简体中文专业版、CCS 3.3。

致谢说明

本书由路锦正编著，同时参加编写的还有李伟、张强林、吴艳、万雷、王呼佳、赵腾伦、王斌、魏雪辉、赵会春、王晓、余松、夏慧军、陈军等老师。本书吸收了由笔者主编的《Visual C++ 音频/视频处理技术及工程实践》一书中的部分章节，并对多数章节做了修改、补充，同时吸收了前书的读者反馈意见，使本书更贴近实用。本书是笔者在假期完成的，故牺牲了陪家人的时间，因此，特别感谢妻子郭晶女士的无私奉献和默默支持，感谢女儿带给我的无穷快乐和对工作的无穷动力。感谢成都道然科技有限责任公司（www.dozan.cn）以及徐津平老师对著书工作的大力支持。另外，本书参考了网络上的部分公开资源，在此对其所有者一并表示感谢。

由于时间仓促，加之水平有限，书中不足之处在所难免，恳请读者批评指正。最后还需要感谢电子工业出版社辛苦工作的老师们，他们为本书的顺利出版付出了很多辛勤劳动。

路锦正

2011年8月

目 录

第 1 章 视频编解码基础	1
1.1 视频编码的必要性和可行性	2
1.2 视频编码压缩原理及主流算法	3
1.3 视频解码原理及主流解码器	16
1.4 ASIC 与可编程芯片	17
1.5 开源的视频 CODEC	18
1.6 本章小结	19
第 2 章 建立和生成编解码视频序列	21
2.1 视频数据来源	22
2.2 DirectShow 实现视频采集	23
2.3 VFW 技术实现视频采集	42
2.4 YUV 序列图像显示	52
2.5 本章小结	62
第 3 章 Xvid 实现 MPEG-4 视频编解码	63
3.1 概述	64
3.2 Xvid 视频编码分析	65
3.3 Xvid 视频解码分析	79
3.4 Xvid 的 MMX/SSE 技术优化	91
3.5 运行 Xvid 系统	100
3.6 Xvid 编解码实验	103
3.7 本章小结	105
第 4 章 数字媒体处理器 TI DM64x 开发平台	106
4.1 概述	107

4.2	适于数字媒体处理的 DSP	108
4.3	DSP 开发环境 CCS	114
4.4	视频算法 DM64x 平台优化	137
4.5	数字媒体处理器 TI DM642DSP	154
4.6	本章小结	158
第 5 章	DM642 平台优化设计 Xvid 视频编解码	159
5.1	优化准备	160
5.2	DM642 视频算法优化策略	167
5.3	DM642 平台下优化 Xvid 视频编码	169
5.4	DM642 平台下优化 Xvid 视频解码	195
5.5	平台优化实验及分析	199
5.6	本章小结	202
第 6 章	x264 实现 H.264 视频编码	203
6.1	H.264 概述	204
6.2	H.264 视频算法原理	205
6.3	x264 视频编码分析	217
6.4	编译运行 x264	239
6.5	系统效果展示	244
6.6	本章小结	246
第 7 章	ffmpeg 实现 H.264 视频解码	247
7.1	ffmpeg 应用基础	248
7.2	ffmpeg 工作流程分析	249
7.3	H.264 解码过程剖析	254
7.4	H.264 视频解码实验	278
7.5	本章小结	278
第 8 章	媒体处理器 DM642 优化 H.264 视频编码	280
8.1	概述	281
8.2	创建 H.264 视频编码器	281
8.3	使用 H.264 视频编码器	288
8.4	销毁 H.264 视频编码器	317
8.5	实验 H.264 视频编码器	318
8.6	本章小结	321

第 9 章 eXpressDSP 算法接口标准开发	322
9.1 xDAIS 开发基础	323
9.2 H.264 编码算法的 xDAIS 封装	324
9.3 使用 xDAIS 封装的 H.264 编码库	342
9.4 本章小结	348
第 10 章 视频监控中心软件设计	349
10.1 概述	350
10.2 位图界面设计	350
10.3 多路图像的高效显示	367
10.4 使用视频监控中心软件	373
10.5 本章小结	374
参考文献	375

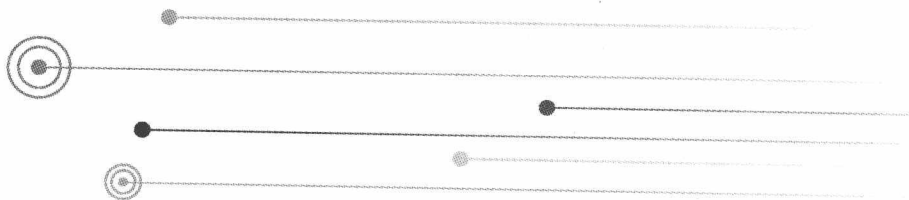
第1章

视频编解码基础

本章要点:

- ▶ 视频编码的必要性和可行性
- ▶ 视频编码压缩原理及主流算法
- ▶ ASIC 与可编程芯片
- ▶ 开源的视频 CODEC

视频编解码处理是数字视频应用的最主要方面。海量的视频数据在存储、传输时必须做编码压缩；视频的帧内空间局部连续性和帧间运动连贯性以及人眼视觉机制等为视频数据的编码压缩提供了可能；主流的视频编码标准 MPEG-4 及 H.264 均为混合编码技术框架，且适于不同的视频应用领域；专用芯片 ASIC (Application Specific Integrated Circuits) 与 CPU/GPU/DSP/FPGA 等可编程芯片构成视频技术的实现平台；开源算法为基于可编程芯片实现可定制视频 CODEC 系统提供了蓝本。



1.1 视频编码的必要性和可行性

在应用目前的视频捕获设备时,获得的图像数据量是巨大的,因此,为节约存储空间、降低传输带宽占用,一般做编码压缩处理。在不影响图像客观信息表达的前提下,我们经常采取有损编码(不可逆)处理。图像的局部空间在一定意义上是光滑连续的,特别是灰度一致不变的区域;同时,图像帧间除了目标运动或少许的光照变化之外,帧间的大部分残差是很小的,甚至趋于零值。生理实验表明,人眼对亮度比色度更加敏感。上述因素促成视频图像编码的必要性和可行性。

1.1.1 编码压缩的必要性

以视频监控的主流分辨率 CIF(352×288, PAL 制)格式为例,考虑帧频为 25f/s、RGB24 位的彩色图像,则 1 小时的数据量为: $352 \times 288 \times 25 \times 3 \times 3600 = 25.49\text{G}$, 即 25G 字节。尽管目前的硬盘容量已发展到 T 数量级,但是持续存储多天的视频数据仍然是个难题,而通常情况下,视频监控一般只保存最近三个月的历史记录。CIF 分辨率 1 秒钟的视频数据量是 58M bit, 尽管目前已有千兆光纤网,但普通的 2M 网络带宽远远无法满足实时传输需求。

随着计算机、电子技术的迅猛发展以及人们对的高分辨率、高质量图像的迫切需求, CIF 近乎成为过去式, D1(720×576, 又称标清)、D3(1920×1080, 又称高清)日渐成为视频主流格式。简单换算可知, D1 及以上大小的视频图像在存储和传输等各种应用中, 进行编码压缩处理更是必然的。

1.1.2 编码压缩的可行性

视频图像是立体场景的二维空间数据记录,在空间上是局部连续的、光滑的,而在时间上是运动持续的、连贯的。空间中的区域一致性或局部连续性允许采用少量的主信息来表达其他信息,即预测编码;时间上的运动渐变性允许帧间残差表示目标位移。因此说视频图像在空间及时间上是相关的、冗余的,编码压缩就是利用相关性去除冗余,保留图像的主要信息。

据生理学研究,人眼具有度冗余特性,即人眼对亮度信息的分辨率高于对颜色信息的分辨率,则处理视频数据时我们就可以采取全亮度信息而部分色度信息的数据格式(如 YUV420/I420, 即 4 个 Y、1 个 U 和 1 个 V)的方式。也就是说部分亮度信息对人眼来说是冗余的。

所以无论是从视频数据自身特点还是图像信息接受者的生理特性考虑,对海量的视频数据做编码压缩以降低空间占用等方法都是可行的。

1.2 视频编码压缩原理及主流算法

视频编解码已经发展成为一个巨大的产业，同时产生了很多多种算法标准。国际上起草视频标准的主要有国际标准化组织 ISO (International Organization for Standardization) 和国际电信联盟 ITU (International Telecommunication Union) 两大组织，他们制定了多种应用成功的视频编码压缩标准协议。如 ISO 的 MPEG-1 (VCD)、MPEG-2 (DVD、DVB)、MPEG-4/7/21 等标准，ITU 的 H.261/H.263 标准，ISO 的 MPEG 小组和 ITU 的 VCEG 小组合作成立的联合视频编码组 JVT (Joint Video Team) 制定了 H.264/AVC 标准。MPEG-4 视频编码的应用较为普及，H.264 视频标准的开发较为广泛。国内的 AVS (Audio and Video coding Standard) 编码效率与 H.264 标准相当，但复杂度版权费较低。相对来说，MPEG-4 的开发和应用较成熟，H.264 是目前研发和市场关注的热点，AVS 的吸引力相比 H.264 较弱。因此目前主流的视频编解码算法标准是 MPEG-4 和 H.264。因此，本书针对 MPEG-4/H.264 视频编码技术以工程实践的角度讲解相关的技术。

1.2.1 视频编码压缩原理

当前的大部分视频编码算法，包括 MPEG-4/H.264 以及 AVS 等，其工作机制均是混合编码，处理模块主要包含：预测、运动估计、变换、量化和熵编码等。不同的视频算法是前述各模块的不同实现方法，而工作原理大同小异。总体来看，图像帧的编码模式主要有帧内和帧间两种方式。帧内编码又包括预测、变换、量化和熵编码等，帧间编码包括运动估计、运动补偿、变换、量化和熵编码等。帧内预测和帧间运动估计统称预测编码。

预测编码是视频编码的最主要方法，简单而有效。编码理论和实际均已表明，图像的当前像素点与帧内相邻的上、下、左、右等位置的像点相比，灰度值变化很小，相关性极强，即图像存在空间冗余。视频帧是活动的图像序列，每秒含有多帧图像（要求不低于 25f/s），当前图像点（块）与前、后帧的对应位置或附近的图像点（块）非常相似，相关性也很强，即序列图像存在时间冗余。预测编码就是减少上述的空间冗余和时间冗余，并分别对应帧内编码和帧间编码。

视频编码压缩的一般工作原理可用图 1-1 来说明，其中 K 表明帧内或帧间编码的开关。下面分别对上述功能模块进行逐一介绍。

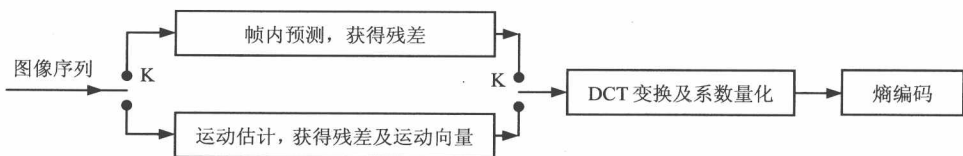


图 1-1 视频编码压缩原理图

1. 帧内预测编码

帧内预测编码是对图像数据本身作编码,如静态图像编码、I 帧(或关键帧 Key Frame)编码都是帧内编码。由于空间像素值不会突变,则我们在处理当前图像块时,可以以先前已编解码的图像作参考,则参差的零值或较小的值会很多,起到了编码压缩的目的。帧内编码的优点是错误累积少、图像清晰,缺点是压缩倍率较小。在视频编码中,关键帧即 I 帧的间隔分固定和自动两种方式,固定间隔如 $25\text{f/s} \times 10\text{s} = 250$ 帧,即每隔 10 秒出现一个 I 帧的帧内编码;自动间隔即编码系统的码率控制器根据设定的码流阈值及宏块的 SAD 值综合判断。图 1-2 是帧内预测编解码原理图。

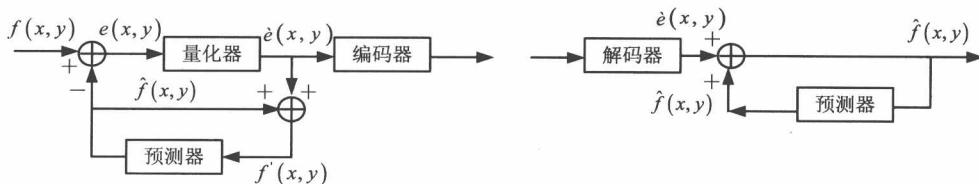


图 a 预测编码

图 b 预测解码

图 1-2 帧内预测编解码原理图

上图中的 $f(x, y)$ 为当前像素值, (x, y) 为像素点水平和垂直坐标。其预测值 $\hat{f}(x, y)$ 可以由当前位置 (x, y) 的相邻像素预测得到:

$$\hat{f}(x, y) = \sum_{(k, l) \in Z} \alpha_{k, l} f(x - k, y - l) \quad (1-1)$$

$\alpha_{k, l}$ 为预测系数, Z 为预测区域, (k, l) 分别为对当前点进行预测的像素点的水平和垂直位置坐标。 $e(x, y)$ 是当前像素值和预测值的差值, 差值经过量化、编码形成最后的码流。用户可以发现量化器置于预测环路之内, 量化误差被反馈回环路, 从而使残存的量化误差减小。

上面是帧内预测编码的原理, 但是视频编码的预测是对变换、量化后的系数做预测的, 并且根据不同的视频压缩算法标准, 预测的方式也有多种。如 MPEG-1/2/4 标准的帧内预测是利用当前块的上方、左方、斜上方等已编码块的第一行或第一列, 来预测当前块的第一行和第一列, 同时除去第一个直流分量; 而 H.264 标准的帧内预测则根据块的大小, 16×16 亮度块有 4 种预测模式, 4×4 亮度块有 9 种预测模式, 8×8 亮度块有 4 种预测模式。预测值和当前值相减, 得到预测残差, 然后代替当前系数值, 从而增加零值和较小值的概率, 达到一定的压缩效果。但是, 如果图像细节丰富, 帧内预测编码效率往往并不高。

2. 帧间预测编码

帧间预测编码是视频编码的主要方法, 帧间编码即 P 帧(前向预测)编码、B 帧(双向预测)编码。帧间的预测不是在当前图像帧内, 而是在已编码重建帧的对应位置, 或其周围作预测、搜索。因为帧间预测是图像在时间先后上的视频帧处理, 所以又非常像一个物体在运动, 体现在各个视频帧中。帧间预测就是在一定窗口内搜索到最佳匹配宏块, 然

后这两个宏块相减得到预测残差。

由于物体的实际运动轨迹并不是整数像素大小,为了更精确描述物体的运动轨迹,视频算法提出了 1/2 像素、1/4 像素、1/8 像素的运动估计,则此时得到的匹配块和运动矢量就相对更准确。另外由于物体如果快速运动,那么在一定的窗口内则无法收集到最佳匹配块和最真实运动矢量。但是,算法若在整个重建帧内搜索,计算量剧增,从而使算法不实用。这里主要介绍 P 帧的 1/2 像素菱形运动估计、插值补偿的帧间预测。其他更高分数精度像素和复杂运动估计算法,可参阅相关资料。

(1) 帧间预测基本原理

基本原理框图如图 1-3 所示。当前帧图像 $f_t(x, y)$ 与预测图像 $\hat{f}_t(x, y)$ 相减后的帧误差 $e_t(x, y)$,经量化器量化后输出 $e'_t(x, y)$,进行熵编码。预测图像 $\hat{f}_t(x, y)$ 与 $e'_t(x, y)$ 相加,得 $f'_t(x, y)$,当不计量化失真时, $f'_t(x, y)$ 即当前的 $f_t(x, y)$ 。把当前帧 $f'_t(x, y)$ 与帧存储器输出的前一帧 $f_{t-1}(x, y)$ (也称重建参考帧)同时输入运动参数估值器,经搜索、比较得到运动矢量 MV (Motion Vector)。此 MV 输入运动补偿预测器,得到预测图像 $\hat{f}_t(x, y)$ 。

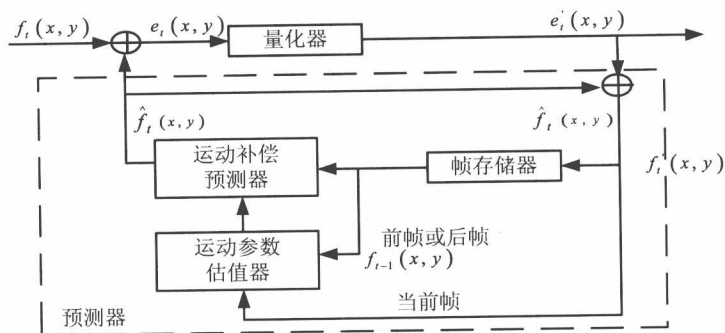


图 1-3 前向预测编码

预测图像 $\hat{f}_t(x, y)$ 不可能完全等同于当前图像 $f_t(x, y)$,无论预测得如何精确,总存在帧误差 $e_t(x, y)$ 。由上述可知,利用上一帧的图像经运动矢量位移作为预测值的方法称为单向预测,则:

$$\hat{f}_t(x, y) = f_t(x + i, y + j) \quad (1-2)$$

其中, (i, j) 即为当前宏块的运动矢量。预测图像与当前图像对应相减得到残差,对残差 $e'_t(x, y)$ 作熵编码,同时编码运动矢量。

(2) 重建图像的插值

连续运动的物体,其轨迹是连续的,而像素是离散的。所以只用整数像素来表示运动物体显然是不准确的,因此有非整数像素运动轨迹之称。对于前向预测编码的参考帧而言,只有重建参考帧,就需要在此参考帧的基础上插值出多帧非整数像素运动轨迹的参考帧。如 1/2 像素运动轨迹则有四个参考帧:重建参考帧、重建参考帧的水平插值参考帧、重建参考帧的垂直参考帧、重建参考帧的斜角参考帧。如 MPEG-4 算法中 1/2 像素插值用到的底层函数 `Interpolate16x16_halfpel_h`、`interpolate16x16_halfpel_v`、`interpolate16x16_halfpel_hv`

分别得到三个方向的插值参考帧。更高精度的插值则是前面较低精度的基础上继续在各个方向上插值。

(3) 运动估计算法

在以上参考帧中对当前宏块搜索最佳匹配宏块, 估计出最佳运动矢量大小。这里介绍的运动估计算法是基于块匹配的运动估值算法, 即对当前编码宏块寻找最佳匹配图像宏块, 同时就找到了最佳运动矢量 $mv(dx, dy)$ 。运动估计的方法有多种, 搜索的范围、大小也有多种; 另外这里的最佳也是有限范围内的最佳, 并且最佳准则也有多种。所以运动估计要解决两个问题: 匹配准则和搜索算法。

① 匹配准则

常用的块匹配判断准则有平均绝对差 (MAD)、差的绝对值的和 (SAD), 公式分别见 (1-3.a) 和 (1-3.b)。

$$MAD = \frac{1}{I \times J} \sum_{|i| \leq \frac{I}{2}} \sum_{|j| \leq \frac{J}{2}} |f(i, j) - g(i - d_x, j - d_y)|, I = J = 16 \quad (1-3.a)$$

$$SAD = I \times J \times MAD \quad (1-3.b)$$

由于 SAD 只有乘加没有除法, 所以 SAD 准则更为常用。如 SAD 值越小, 表明这两个块越相似; MAD 值的意义判断同 SAD。

② 搜索算法

对当前预测编码图像的宏块在参考帧中, 在一定窗口内, 搜索判断, 根据 SAD 准则在窗口限制范围内搜索最佳匹配宏块 $f(x, y)$ 和运动矢量 $mv(dx, dy)$, 运动估计原理图如图 1-4 所示。根据搜索的路径不同有多种运动估计算法。

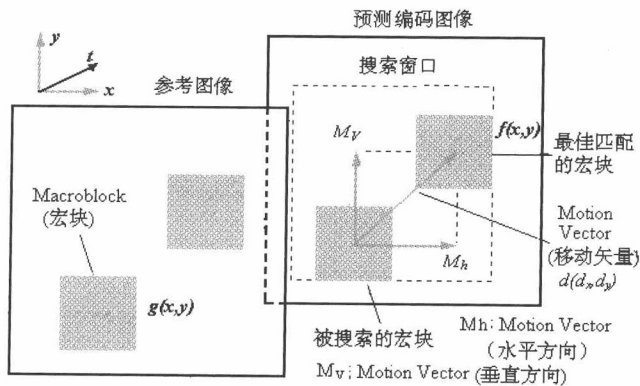


图 1-4 运动估计原理图

最简单的运动估计算法有三步搜索法、对数搜索法、四步搜索法, 或者菱形搜索、大/小钻石搜索、六角形搜索等方法。

三步搜索算法如图 1-5 所示: 每一步搜索 9 个位置点, 搜索步长为 3。对每一个位置计算 MAD/SAD 值, 确定最小的失真方向, 在最小失真方向上搜索区域减少一个搜索步长, 下一步搜索新的 9 个位置点。直到搜索步长为 1, 此时的宏块即为用户认为的最佳匹配块,

并记录当前运动矢量。

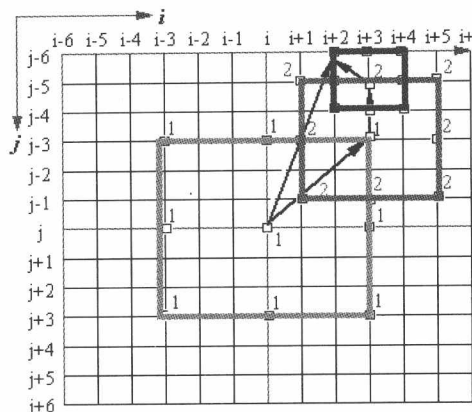


图 1-5 三步搜索算法示意图

对数搜索法如图 1-6 所示：每一步搜索由区域中心位置和与中心相联的四个边缘点组成，连续搜索直到寻找的区域成为 3×3 矩形区域为止。最后一步计算 3×3 矩形区域的 9 个位置点，MAD/SAD 为最小值时对应的宏块就是要寻找的最佳匹配块。

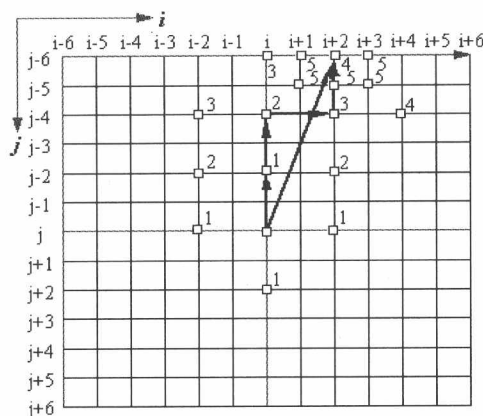


图 1-6 对数搜索示意图

由于实际物体的运动千变万化，很难用一种简单的模型去描述，也很难用一种单一的算法来搜索最佳运动矢量，因此实际上大多采用多种搜索算法相组合的办法，可以在很大程度上提高预测的有效性和鲁棒性。如图 1-7 为 H.264 视频标准推荐的运动估计算法，全称是：非对称十字型多层次六边形格点运动搜索算法。

H.264 的运动估计算法利用了多种搜索方式。

第一步：非对称十字形搜索，水平搜索多于垂直方向；

第二步：正方形区域搜索；

第三步：中六边形模板、十字模板。

有时根据性能需求，简化该图，只采用部分搜索方式。经过实验证明对于中小运动序

列, 其速度可接近全局搜索算法的 400 倍, 接近三步搜索算法的 4 倍; 而对于大运动序列, 该算法则会自动调节搜索点数以适应复杂的运动。

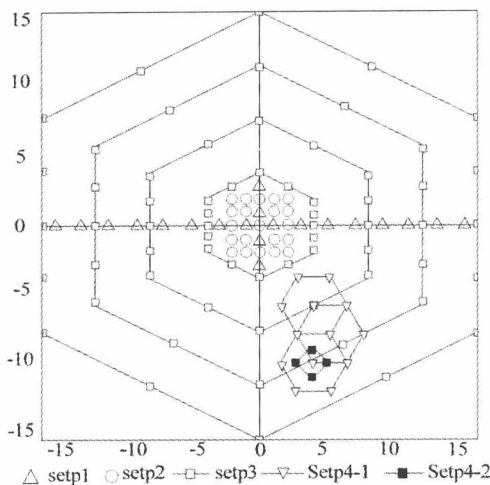


图 1-7 H.264 混合运动估计算法搜索示意图

(4) 运动补偿插值

根据匹配准则得到最佳匹配宏块及其运动矢量, 作减运算得到残差。在计算残差时, 需要根据运动矢量和参考帧类型作插值补偿, 即在参考帧中, 对应宏块位置偏移运动矢量大小得到匹配宏块, 然后根据运动矢量选择插值帧: 水平插值、垂直插值、斜角插值, 最后做减法运算, 完成运动补偿。

事实上, 在进行运动估计时也并不是单一使用上述某一类搜索算法, 而是根据各类算法的优点灵活组合。在运动幅度比较大的情况下可以采用自适应的菱形搜索法和六边形搜索法, 这样可以大大节省码率而图像质量并未有所下降。在运动图像非常复杂的情况下, 采用全局搜索法, 在比特数相对来说增加不多的情况下, 可使图像质量得到保证。

3. 变换编码

变换编码是把时域或空域的信号通过数学工具转换到另外一个域内, 而在这个新域中, 信号的能量重新分布, 且能量更集中, 从而便于编码压缩。根据统计和经验得知: 平坦区域或内容缓慢变化区域占据一幅图像的绝大部分, 而细节区域或内容突变区域则占小部分。也可以说, 图像中直流和低频分量占大部分, 高频分量占小部分。这样, 空间域的图像变换到频域, 即所谓的变换域, 会产生相关性很小的一些变换系数, 并可对其进行压缩编码。视频编码中变换编码的数据源, 包括两种: 一是图像数据本身, 二是图像残差。

变换中的正交变换, 可用于图像编码。自 1968 年利用快速傅立叶变换 (FFT) 进行图像编码以来, 出现了多种正交变换编码方法, 如 K-L 变换、离散余弦变换 (DCT)、小波变换等等。其中, 编码性能以 K-L 变换最理想, 但缺乏快速算法, 且变换矩阵随图像而异, 不同图像需计算不同的变换矩阵, 因而只用来作参考比较。DCT 编码性能略逊组最接近于

K-L 变换, 具有快速算法, 广泛应用于图像编码, 如 MPEG-1/2/4、H.261/263 的浮点 DCT 变换和 H.264 中的整数 DCT 变换。JPEG 采用了 DCT 变换, JPEG2000 采用了小波变换。视频算法中的 DCT 原理如下表示, 其中 $f(x, y)$ 为 $N \times N$ 数字图像序列, DCT 和 IDCT 变换核如下公式所示。

二维 DCT 变换表示为:

$$F(u, v) = \frac{2}{N} c(u) c(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos \left[\frac{\pi}{2N} (2x+1)u \right] \cos \left[\frac{\pi}{2N} (2y+1)v \right] \quad (1-4)$$

$$c(u) = \begin{cases} \frac{1}{\sqrt{2}} & u=0 \\ 1 & u \neq 0 \end{cases}, \quad c(v) = \begin{cases} \frac{1}{\sqrt{2}} & v=0 \\ 1 & v \neq 0 \end{cases}, \quad \text{其中 } u, v = 0, 1, \dots, N-1 \quad (1-5)$$

二维 IDCT 变换表示为:

$$f(x, y) = \frac{2}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} c(u) c(v) F(u, v) \cos \left[\frac{\pi}{2N} (2x+1)u \right] \cos \left[\frac{\pi}{2N} (2y+1)v \right] \quad (1-6)$$

$c(u)$ 和 $c(v)$ 同式 (1-5), 其中 $x, y = 0, 1, \dots, N-1$ 。

DCT 和 IDCT 也可写成矩阵形式:

$$F = AfA^T \quad (1-7)$$

$$f = A^T F A \quad (1-8)$$

对图像做变换编码, 最理想的变换运算应对整个图像进行, 以便去除所有像素间的相关性。但这样的操作计算量太大, 需要的临时存储区太大。实际上, 往往把图像分为一定大小的若干块, 如 16×16 的宏块, 然后以宏块为单位进行 DCT 变换。视频算法中通常为 16×16 或 8×8 的宏块。但是也正因为是以宏块为处理单位, 经编解码后的图像有明显的块效应, 这就需要用户在解码低码流的视频时增加后处理以提高图像质量。H.264 视频编码算法就在编码内部及解码端开启了环路滤波功能, 提高参考帧或解码帧的图像质量。

传统的 DCT 不是完全可逆的, 所以解码出的图像有“拖尾效应”及误差积累, 即 I 帧间隔不能太大, 否则 DCT 的不可逆操作 IDCT 会产生更多的图像误差积累。H.264 提供了整数 DCT/IDCT, 其反变换是完全可逆的。H.264 图像在同码流下, 明显优于其他以往标准算法, 很重要的一个特点就是变换编码采用了整数 DCT。

4. 量化编码

量化编码是把 DCT 系数除以一个常量, 经过量化后的结果是量化步长的整数倍或为更多的零值, 从而达到了压缩的目的。如 H.263 的编码方式是 16×16 宏块的 DCT 系数除以同一个量化步长, 即均匀量化; 而 MPEG 的量化方式是采用量化矩阵的方法, 即不同重要性的 DCT 系数除以不同的量化步长。在反量化时, 由于量化过程取整或四舍五入, 而无法完全恢复原 DCT 系数, 从而产生失真。量化是视频编码失真的根本原因。量化的原理公式如 (1-9) 所示: