

基于提升小波变换的视频压缩技术的研究与实现

摘 要

随着网络的日益普及和多媒体技术的不断发展,数字多媒体应用技术得到很大的发展,但仍跟不上多媒体文件体积增大的速度,经过数字化后的视频和音频信息体积具有数据海量性,这给信息的存储和传输带来了很多的困难。正因为如此,对多媒体文件压缩处理的研究一直是一个热门的研究课题,此方面的研究成果和新技术也不断推出。

本文也是从此角度出发,借鉴第一代小波视频编码成功的经验,提出了基于提升小波变换的视频压缩技术的完整模型。采用整型双正交小波作为提升小波基,使整个变换具有过程简单、复杂度低、内存少和易逆变换等优势,从而减少了在变换、量化过程中图像能量损失,并实现软件编码。然后研究块间匹配的运动估计的算法,总结现有算法的优缺点,结合整像素和半像素两种估计方式,提出了新的逐级搜索方式算法,该算法可以满足不同用户的需求;其中自适应的阈值搜索终止规则,解决搜索过程中局部最优的矛盾;并对边缘的宏块估计提出新的解决办法。经测试该算法在块匹配的搜索时间和解码后的图像质量方面都有很大的改善。最后,在 Visual C++6.0 上仿真和实现关键算法,使基于提升小波变换的视频压缩技术能应用于实际中。

关键词: 提升小波变换; 运动估计; 视频编码; 多分辨率分析; 整型双正交小波

Studying and Implementation of Video Compression Based on Lifted Wavelet Transform

Abstract

With the development of the network and multimedia technology, digital multimedia technology also has made a large progress. But it can't catch up with the speed that the volume of the multimedia file increases. The volume of video and audio information has mass data when it is digitized. It has brought much difficulty to store and transmit information. Just because of this, it is a hot subject that deals with the compression of the multimedia file for research all the time. The research results and new technology in this aspect are also coming in all the time.

Profiting from the successful experiences of video coding based on the first-generation wavelet transform, adopts to integer biorthogonal wavelet as the lifted wavelet base, which makes entire transform to have many superiorities, such as the simple process, the low complexity, the few memory and easy inverse transform, reduces energy loss in transform and quantification process, and then realizes the software coding. Then this paper researches the block match movement estimate algorithms, summarizes their merits and defects, and combines with the two estimate's methods of the entire pixel and half pixel, puts forward a new progressive search mode algorithm, the algorithm can satisfy the demand of different users; in which the auto-adapted threshold search termination rules can solve the contradiction of the local optimum in the search process, and proposes a new solution for edge macro block's estimate. According to the test results, the new algorithm makes both the search time of block match and the quality of decoded picture improved greatly. Finally this paper simulates and realizes key algorithms on the Visual C++6.0, and these algorithms enable the video compression technology based on the lifted wavelet transform to applications.

Key words: Lifted Wavelet Transform; Motion Estimate; Video Coding;
Multiresolution Analysis; Integer Biorthogonal Wavelet

独创性声明

本人声明所呈交的学位论文是在导师的指导下完成的。论文中取得的研究成果除加以标注和致谢的地方外，不包含其他人已经发表或撰写过的研究成果，也不包括本人为获得其他学位而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示诚挚的谢意。

学位论文作者签名：张延军

签字日期：2007.1.12

学位论文版权使用授权书

本学位论文作者和指导教师完全了解东北大学有关保留、使用学位论文的规定：即学校有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人同意东北大学可以将学位论文的全部或部分内容编入有关数据库进行检索、交流。

（如作者和导师同意网上交流，请在下方签名；否则视为不同意）

学位论文作者签名：张延军

导师签名：张延军

签字日期：2007.1.12

签字日期：2007.1.15

第一章 绪论

1.1 数字视频压缩技术概述

信息时代的重要特征是信息的数字化，而信息的数字化则带来了“信息爆炸”。由于计算机，数字通信系统和电信网络的出现和发展，使得信息的数字化表示、存储、处理和传输更加简便易行。如今，信息的获取和处理已经成为人们生活和工作必不可少的方式和手段了，而多媒体信息以其丰富的内容和友好的环境更加受到青睐，比如：VCD、DVD、电视会议、视频点播、可视电话、高清晰度数字电视（HDTV）以及新兴的多媒体短信（MMS）等等。在多媒体信息中，图像和视频所占的比重是最大的。但是它们的数据量太大，要求的存储空间和传输的带宽都很大，对于实际中的应用是非常不利的。比如：CIF 格式的视频，彩色 4:4:4 格式采样，每帧数据量： $352 \times 288 \times 3 = 304.128$ (KB)，每秒数据量： $304.128 \times 30 = 9.12$ (MB/s)，一片 CD-ROM 可存帧数： $650 / 0.304 = 2137.26$ (帧/片)，一片 CD-ROM 可存的节目时间： $(650 / 9.12) / 60 = 1.188$ (分钟/片)。由这个例子可以看出，数字化信息的数据量是非常庞大的。这样大的数据量给存储器的存储容量，通信线路的信道传输以及计算机的运行速度都增加了极大的压力。如果单纯地使用扩大存储器容量，增加通信干线的传输率的办法是不现实的。而数据压缩技术则是一个有效的方法，通过数字压缩手段把信息数据量降低下来，以压缩后的形式进行存储和传输，既紧缩节约了存储空间，又提高了通信干线的传输效率，同时也使计算机能实时处理音频、视频信息，并且保证播放高质量的视频、音频成为可能。因此，数据压缩的任务就是保持信号在一个可以接受的状况的前提下把需要的比特数减少到最少程度，即用尽可能少的比特数来表示信源信号并能将其还原，以减少存储和传输的成本^[1]。

由于原始的数字化视频数据中有极强的相关性，存在着大量的冗余信息，比如空间冗余，时间冗余，视觉冗余等等。数据压缩就是将庞大数据中的冗余信息去掉，保持相互独立的信息分量。以静止图像画面为例，数字化图像的灰度信号和色差信号在空域虽然属于一个随机场分布，但是它可以看成为一个平稳的马尔可夫场。图像的像素点在空域中的灰度值和色差信号值，除了边界轮廓之外，都是缓慢变化，比如一幅头肩图像，背景，人脸，头发等处的灰度，颜色都是平缓改变。相邻像素的灰度和色差值比较接近，具有强的相关性，直接用采样数据表示灰度和色差，信息有较多的冗余^[2]。因此，如何排除冗余信息，再进行编码，使得表示每个像素的平均比特数下降，就是图像压缩和视

频压缩所要解决的问题。

1948年提出电视信号数字化后,人们就开始了图像压缩编码的研究工作,至今已经将近有六十年的历史,在这期间,人们开发出了各种不同的数据压缩算法,主要分为统计编码,预测编码,变换编码等等^[3],在开始的二十年间,人们只是对帧内的线性预测编码和亚抽样内插复原法进行研究,对视觉特性也做了可贵的工作。70年代开始进行帧间预测编码的研究;80年代初开始对运动估计进行研究。变换编码是1968年由Andrews等人提出采用二维傅立叶变换,此后相继出现了用其它变换方法的变换编码,如离散余弦变换和离散小波变换。随着压缩技术的不断进步,形成了一系列的视频压缩标准,比如:H.261, H.263, MPEG-1, MPEG-2, MPEG-4^{[4][5]},以及最近出台的H.264。这些标准的主要目的就是要达到比较高的压缩效率,同时保证一定的图像保真度,以满足不同的应用需求。这些标准采用的都是混合编码技术,即将统计编码,预测编码,变换编码都融合到编码框架之中。这些标准的制订推动了数据压缩算法的发展,也为工业界形成了规范,使得人们能够充分的享受高科技所带来的乐趣。

目前,主要的数据压缩技术都是为了减少像素级别的冗余信息,属于第一代编码技术,是以信息论和数字信号处理技术为理论基础,旨在去除图像数据中的线性相关性的编码技术。随着人们对信息的处理和管理需求的多样化发展,人们对信息处理的要求越来越多,也越来越高。从电报,电话,广播的单一信息传输方式,到电视,电影的被动形式的多媒体方式,再到互联网的主动形式的多媒体传输形式,人们仍然不能满足现在的信息技术水平。数字电影制作,多媒体信息查询,数字媒体的管理,视频点播,多媒体信息的无线传输,电脑游戏等等,要求对多媒体信息进行更有效的描述,管理和综合处理,并且还要实现多媒体信息的交互式应用。这些需求促使了第二代编码技术的出现和发展。第二代编码技术并不局限于信息论的框架,它充分的利用了人的视觉生理,心理和图像信源的各种特征,以获得更高的压缩比^{[6][7]}。第二代编码技术主要有:基于分形的编码,基于模型的编码,基于区域分割的编码,基于神经网络的编码等等。

第二代编码技术就是要解决减少数据中语义级别的冗余信息的问题,以达到更高的压缩效率,并且能够实现便捷的多媒体交互式应用。基于视频对象的编码方法就属于第二代编码技术。要实现以上基于内容的多媒体应用,最重要的就是将多媒体信息有效的划分为不同的对象,最终的目标应该是语义水平的对象。对于视频或图像来说,就要将它们进行有效的分割,再对分割出来的对象分别进行编码,压缩,存储。可见,视频对象的编码是实现多媒体信息基于内容应用中的重要一环。因为只有采用有效的编码、压

缩方法, 才能实现高的压缩性能, 实现对多媒体内容的访问和控制。

1.2 数字视频压缩的标准

近年来, 一系列国际视频压缩编码标准的制定极大地促进了视频压缩编码技术和多媒体通信技术的发展。视频压缩编码标准的制定工作主要是由国际标准化组织 (International Standardization Organization, 简称 ISO) 和国际电信联盟 (International Telecommunication Union, 简称 ITU) 完成的。由 ITU 组织制定的标准主要是针对实时视频通讯的应用, 如视频会议和可视电话等, 它们以 H.26x 命名 (如 H.261, H.262, H.263 和 H.264); 而由 ISO 和 IEC (International Electrotechnical Commission, 国际电工委员会) 的共同委员会中的 MPEG 组织 (Moving Picture Export Group) 制定的标准主要是针对视频数据的存储 (如: DVD)、广播电视和视频流的网络传输等应用, 它们以 MPEG-x 命名 (如 MPEG-1, MPEG-2, MPEG-4, MPEG-7 等)^[5]。

1.2.1 H.26x 系列

(1) 视听会议压缩编码标准 H.261

H.261 是第一个广泛投入应用的数字视频编码标准, 又称为 P×64, 其中 P 为 1~30 的可变参数, P 取值较小时, 只能传清晰度不太高的图像, 适合于面对面的电视电话; P 取值较大时 (如: P>6), 则可以传输清晰度较好的会议电视图像, 最初是针对在 ISDN 上实现电信会议应用特别是面对面的可视电话和视频会议而设计的, H.261 奠定了混合编码框架的雏形, 其视频压缩算法的核心是运动估值预测和 DCT 编码, 许多技术 (包括视频数据格式、运动估算与补偿、DCT 变换、量化和熵编码) 都被后来的 MPEG-1, MPEG-2 所借鉴和采用。它属于恒定码流可变质量编码而非恒定质量可变码流编码。

(2) 低比特率视听会议压缩编码标准 H.263

H.263 是为低码流通信而设计的, 但实际上这个标准可用于很宽的码流范围, 而非只用于低码率范围, 它在许多应用中可以被用来取代 H.261。H.263 的编码算法与 H.261 一样, 但做了一些改善, 以提高性能和纠错能力。H.263 标准在低码率下能够提供比 H.261 更好的图像效果, 例如在比特率低于 30 kb/s 的应用中, 同样的视频质量前提下, H.263 的输出码率仅为 H.261 的一半甚至更少。

两者的区别包括:

- ① H.263 的运动补偿使用半像素精度, 而 H.261 则用全像素精度和循环滤波;
- ② 数据流层次结构的某些部分在 H.263 中是可选的, 使得编解码可以配置成更低

的数据率或更好的纠错能力；

- ③ H.263 包含 4 个可协商的选项以改善性能；
- ④ H.263 采用无限制的运动向量以及基于语法的算术编码；
- ⑤ 采用事先预测和与 MPEG 中的 P-B 帧一样的帧预测方法；
- ⑥ H.263 支持 5 种分辨率，即除了支持 H.261 中所支持的 QCIF 和 CIF 外，还支持 SQCIF, 4CIF 和 16CIF。

(3) H.263+/++

H.263+是 H.263 的改进版，它在保证原 H.263 标准的核心句法和语义不变的基础上，增加了若干选项以提高压缩效率或某方面的功能，它提供了 12 个新的可协商模式和其他特征，进一步提高了压缩编码性能。另一重要的改进是可扩展性，它允许多显示率、多速率及多分辨率，增强了视频信息在易误码、易丢包异构网络环境下的传输。新增选项可归纳为新的图像种类和新的编码模式两类。

H.263++是在 H.263+的基础上又增加了 3 个选项，主要是为了增强码流在恶劣信道上的抗误码性能，同时也是为了增强编码效率，这 3 个选项分别为：增强型的参考帧选择，数据分片的模式，在 H.263+的码流中增加了补充信息，保证增强型的反向兼容。

(4) 新一代视听会议压缩编码标准 H.264

H.264 同样采用 DPCM 加变换编码的混合编码模式，但它采用“回归基本”简洁设计，不用众多的选项，获得比 H.263++好得多的压缩性能；应用目标范围较宽，以满足不同速率、不同分辨率以及不同传输（存储）场合的需求。

技术特点包括：

- ① 帧间编码，可变块大小的运动补偿预测技术；
- ② 帧内编码，多方向的空间预测技术；
- ③ 环内滤波器，去除块效应；
- ④ 通 4×4 的整数正交变换及相应的量化策略；
- ⑤ 改进的运动向量预测编码；
- ⑥ 更加高效的熵编码器；
- ⑦ 基于率失真的编码优化技术。

这些措施使得 H.264 算法具有很高的编码效率，在相同的重建图像质量下，能够比 H.263 节约 50%左右的码率。H.264 的码流结构网络适应性强，增加了差错恢复能力，能够很好地适应 IP 和无线网络的应用。但从总体上说，H.264 性能的改进是以增加复杂

性为代价而获得的，其编码的计算复杂度大约相当于 H.263 的 3 倍，解码复杂度大约相当于 H.263 的 2 倍。

1.2.2 MPEG-x 系列

(1) 面向数字存储的运动图像及其伴音的编码标准 MPEG-1

MPEG-1 标准 1990 年 12 月提出标准草案，1993 年 8 月公布，是适用于传输 1.5Mbps 数据传输率的数字存储媒体运动图像及其伴音的编码标准。MPEG-1 标准可以处理各种类型的活动图像，其基本算法对于压缩水平方向 360 个像素、竖直方向 288 个像素的空间分辨力，每秒 24 至 30 幅画面的运动图像有很好的效果。与 JPEG 不同，它没有定义产生合法数据流所需的详细算法，而是在编码器设计中提供了大量的灵活性，另外定义已编码位流和解码器的一系列参数都包含在位流本身当中，这些特点允许算法可以用于不同大小和宽高比的图像，也可以用在工作速率范围很大的信道和设备上。

MPEG-1 标准完成的基本任务就是：质量适当的图像（包括伴音）数据必须成为计算机数据的一种，和已有的数据（如文字、绘图等数据）在计算机内兼容，并且这些数据必须在现有的计算机网络和广播电视等通信网络中兼容传输。

(2) 广播系统压缩编码标准 MPEG-2

MPEG-2 标准 1994 年 11 月公布，其全称为“运动图像及其伴音的编码”，主要针对高清晰度电视（HDTV）所需要的视频及伴音信号。制定 MPEG-2 的初衷是得到一个针对广播电视质量（CCIR601 格式）的视频信号的压缩编码标准，但实际上最后得到的是一个通用的标准，它能在很宽范围内对不同分辨率和不同输出比特率的图像信号有效地进行编码。

MPEG-2 核心部分与 MPEG-1 基本相同，是在 MPEG-1 基础上的进一步扩展和改进，克服并解决了 MPEG-1 不能满足日益增长的多媒体技术、数字电视技术对分辨率和传输率等方面的技术要求的缺陷。最高图像分辨率：1920×1152；普通 PAL 制式分辨：720×576；最高数据流量：80Mbps；一般数据流量：6500kbps（720×576）。它是主要针对数字视频广播、高清晰度电视和数字视盘等制定的编码标准，可以支持固定比特率传送、可变比特率传送、随机访问、信道跨越、分级编码、比特流编辑等功能。从本质上讲，MPEG-2 可视为一组 MPEG-1 的最高级编码标准，它保留了 MPEG-1 所提供的所有功能，并设计成与 MPEG-1 兼容，但又增加了基于帧/场的运动补偿、空间可伸缩编码、时间可伸缩编码、质量可伸缩编码以及容错编码等新的编码技术。MPEG-2 有不可分级和可分级两

种编码方式。它还定义了 5 个框架和 4 个级别，框架是标准中定义的语法子集，级别是一个特定框架中参数取值的集合。框架和级别限定以后，解码器的设计和校验，就可以针对限定的框架在限定的级别中进行，同时也为不同的应用领域之间的数据交换提供了方便和可行性，其中的主框架应用最广，也是最为重要的一个。

(3) 基于对象的低码率视频压缩编码标准 MPEG-4

MPEG-4 是 MPEG 组织制定的一种 ISO/IEC 标准，MPEG 组织于 1999 年 1 月正式公布了 MPEG-4 V1.0 版，1999 年 12 月又公布了 MPEG-4 V2.0 版本。MPEG 组织的初衷是制定一个针对视频会议、视频电话的超低比特率（64kbit/s 以下）编码的需求，并打算采用第二代压缩编码算法，以支持甚低码率（Very Low Bit Rate）的应用，但在制定过程中，MPEG 组织深深感到人们对多媒体信息特别是对视频信息的需求由播放型转向到基于内容的访问、检索和操作，所以修改了计划，制定了现在的 MPEG-4。MPEG-4 主要应用于数字电视，交互式图像应用，交互式多媒体（网络根据内容来分布和存储）和基于计算机网络的可视化场景应用等的有效多视角编码，加上多伴音声道编码及有效的视听同步。

MPEG-4 的最高图像分辨率：720×576；最高数据流量：5-10Mbps；普通 PAL 制式分辨率：352×288；数据流量：880kbps。

1.2.3 AVS 标准

AVS 是我国具备自主知识产权的第二代信源编码标准。顾名思义，“信源”是信息的“源头”，信源编码技术解决的重点问题是数字音视频海量数据（即初始数据、信源）的编码压缩问题，故也称数字音视频编解码技术。显而易见，它是其后数字信息传输、存储、播放等环节的前提，因此是数字音视频产业的共性基础标准，它在性能上与最新的国际标准相同或更高，兼容国际标准基本功能，所包含的技术专利不受控于境外企业与组织。

AVS 视频当中具有特征性的核心技术包括：8×8 整数变换、量化、帧内预测、1/4 精度像素插值、特殊的帧间预测运动补偿、二维熵编码、去块效应环内滤波等。

AVS 是一套适应面十分广阔的技术标准。与 MPEG-2 等现有压缩标准相比，在相同的视觉质量下，码率至少可以降低 50%，码率范围覆盖几十 Kbps 的低带宽通信到数十 Mbps 的高清晰度电视广播，可以支持低延迟模式的视频会议应用，也支持高压缩效率的视频存储应用等，明显超越了现有的所有视频编码标准。

据预测,数字音视频产业将在2008年超过通信产业,在2010年成为国民经济第一大产业。AVS作为数字音视频产业“牵一发动全身”的基础性标准,为我国构建“技术—>专利—>标准—>芯片与软件—>整机与系统制造—>数字媒体运营与文化产业”的产业链条提供了难得机遇。

1.3 工作研究的内容

小波分析,无论是作为数学理论的小波变换,还是作为技术工作和方法的离散小波变换,仍有很多理论需要进一步研究。特别是随着适于硬件实现的提升小波的出现,为小波理论的发展提供了更大的动力。图像压缩编码技术目前已经成为高技术领域的汇流点之一。正是基于此,本文作者针对小波理论,方法和提升小波的内容进行了研究,并对视频编码中运动估计算法进行了探讨和改进。

本文主要有以下几个方面的工作:

(1) 由于双正交小波的线性特性,其广泛应用于图象处理领域。小波滤波器的构造是很关键的技术。因此,选则合适的滤波器对整个变换有很大的影响。综合考虑不同小波函数的具体特性,本文提出利用双正交对称9/7小波实现提升变换。

(2) 为了减少系数变换、量化时的能量损失,得到较好的恢复,提出了基于整数提升变换。因为该方法算法简单,易于实现,且对逆变换更容易得到等优点,其结果与其它小波变换方法获得的系数相似。

(3) 运动估计在整个视频压缩系统中的时间复杂度最大,占了整个编码系统时间的50%以上,所以对运动估计的搜索速度至关重要,本文根据整数像素和半像素结合的方式,采用逐级搜索的方法,并对边缘的宏块估计提出解决办法,提高图像帧的压缩质量,并利用自适应的阈值作终止规则解决局部最优的矛盾。

(4) 研究了现有视频编码协议,并在标准协议基础上开发了视频编码器,且编码后的码流能适应于当前主要媒体播放器解码,并通过系统的测试,验证了本文提出的方法具有很好的压缩效果和实际应用能力。

1.4 本文的结构

本论文共分为五章,各章节的具体内容安排如下:

第一章:绪论,讲述了当前课题研究的背景及主流视频压缩系列标准。

第二章:理论知识,分析了小波的基本理论、提升小波变换的原理和变换中的知识
点。

第三章：算法的设计，分析当前运动估计算法的优缺点并提出新改进的算法。

第四章：算法的实现与分析，应用本文知识，利用软件仿真实现了视频文件压缩和对算法性能验证及性能分析。

第五章：总结与展望，在总结全文工作的基础上，提出了下一步工作的方向和基本思路。

第二章 理论知识

2.1 第一代小波变换的理论

2.1.1 小波变换理论的发展

傅立叶变换是信号处理领域中最完美、应用最广泛、效果最好的一种分析手段。但傅立叶变换只是一种纯频域的分析方法，它在频域的定位性是完全准确的（即频域分辨率最高），而在时域无任何定位性（或分辨能力），也即傅立叶变换所反映的是整个信号全部时间下的整体频域特征，而不能提供任何局部时间段上的频率信息。实际中，对于一些常见的非平稳信号它们的频域特性都随时间而变化，因此也可称它们为时变信号。对这一类时变信号进行分析，通常需要提取某一时间段（或瞬间）的频域信息或某一频率段所对应的时域信息。为了研究信号在局部时间范围的频率特征，1946年 Gabor 提出了著名的 Gabor 变换，之后又进一步发展发展为短时傅立叶变换（Short Time Fourier Transform，简记为 STFT，又称为加窗傅立叶变换）。虽然 STFT 已在许多领域获得了广泛的应用，但由于 STFT 的本身特点决定了其窗函数的大小和形状与时间和频率无关而保持固定不变，这对于分析时变信号来说是不利的。高频信号一般持续时间很短，而低频信号持续时间较长，因此，我们希望对于高频信号采用小时间窗，对于低频信号采用大时间窗进行分析，这种变时窗的要求同 STFT 的固定时窗的特性是相矛盾的，这表明 STFT 在处理这一类问题时已不在实用了^[4]。

小波分析（Wavelets Analysis）是近年迅速发展起来的新兴学科，具有深刻的理论意义和广泛的应用范围。小波分析是一种信号的时间—尺度（时间—频率）分析方法，它具有多分辨分析的特点，而且在时频两域都具有表征信号局部特征的能力，是一种窗口大小固定不变但其形状可以改变的时频局部化分析方法。即在低频部分具有较高的频率分辨率和较低的时间分辨率，在高频部分具有较高的时间分辨率和较低的频率分辨率，很适合于探测正常信号中夹带的瞬变反常信号并分析其成份，所以被誉为分析信号的显微镜。由于小波具有多分辨分析的能力，可以对信号和图像在不同尺度上进行分解，在小波域进行去噪、压缩处理后，作反变换得到去噪和压缩后的信号和图像。小波分析用于非平稳信号和图像的处理优于传统的傅立叶变换已被许多应用领域的事实所证实。因此，自小波分析诞生到现在不过 10 年的时间，就在诸如信号信息处理、图像处理、语音分割与合成、故障诊断等取得了很佳的应用效果。

小波变换的思想来源于伸缩与平移方法。小波分析方法的提出,最早应属 1910 年 Haar 提出的规范正交基,这是最早的小波基,但当时并没有出现“小波”这个词,小波概念的真正出现应算于 1984 年。法国地球物理学家 J.Morlet 在分析地震数据时提出将地震波按一个确定函数的伸缩、平移系展开。随后,他与 A.Grossman 共同研究,发展了连续小波变换的几何体系。1985 年,法国的大数学家 Meyer 首先提出了光滑的小波正交基,对小波理论做出了贡献。1986 年, Meyer 及其学生 Lemarie 提出了多尺度分析的思想。1987 年 Mallat 将计算机视觉领域的多尺度分析思想引入到小波分析中,提出了多分辨率分析的概念,统一了在此之前的所有正交小波基的构造,并提出了相应的分解与重构快速算法。1988 年,年轻的女数学家 Daubechies 提出了具有紧支集的光滑正交小波基—Daubechies 基,为小波的应用研究增添了催化剂,同年, Daubechies 在美国主办的小波专题讨论会上进行了十次演讲,引起了广大数学家、物理学家甚至某些企业家的重视,由此将小波的理论 and 实际应用推向了一个高潮。

小波分析的概念在 80 年代由 Morlet 首先提出,此后发展迅速,现已广泛应用于许多学科领域,特别在图像数据压缩、图像处理应用等诸多方面中显示出它独特的魅力。小波分析是一种时频局部化分析方法,能以有效信号表征方式处理图像中的非平稳信息,为多分辨率和子带分解技术提供了一个统一的理论框架^[5]。因此,小波理论迅速成为国际学术界的 research 热点。

自 Morlet 于 1981 年在分析地质数据中首先提出小波分析 (Wavelet Analysis) 这一概念,并研究了 Gabor 变换,对 Fourier 变换与加窗 Fourier 变换的异同、特点以及函数的构造做了创造性的研究后,开创了以他的名字命名的 Morlet 小波后,许多科学家们又在基于其研究领域上发展了小波分析。特别有:1985 年,法国大数学家 Meyer 首次提出光滑的小波正交基,又于 1986 年提出多尺度分析思想;1988 年,女数学家 Daubechies 提出了具有紧支集正交小波基—Daubechies 基,为小波应用研究增添了催化剂。信号分析专家 Mallat 提出了多分辨率分析的概念,给出了构造正交小波基的一般方法,在 1989 年以多分辨率分析为基础提出了著名的快速小波算法—Mallat 塔式算法 (FWT),宣告小波从理论研究走向宽广的应用研究。也是 Mallat 将小波变换引入图像处理领域,随后人们研究了不少用小波变换实现图像数据压缩的方法。

2.1.2 连续小波变换

小波是由一个函数 $\psi(x)$ 经伸缩和平移得到的一组函数:

$$\psi_{a,b}(x) = |a|^{-1/2} \psi\left(\frac{x-b}{a}\right) \quad a, b \in R, a \neq 0 \quad (2.1)$$

其中 $\psi(x)$ 就称为基本小波或母小波，它满足相容条件：

$$W_h = \int_{-\infty}^{\infty} \frac{|\psi(\omega)|}{|\omega|} d\omega < \infty \quad (2.2)$$

其中 $\psi(\omega)$ 是 $\psi(x)$ 的傅立叶变换。式 (2.1) 中， a 表示小波函数的尺度伸缩因子， b 表示小波函数的位移因子。由基本小波函数 $\psi(x)$ 和 a 、 b 的改变可以得到一组小波基底函数，信号 f 在这一组基底函数上的分解就叫小波变换：

$$W_f(a,b) = \langle f(x), \psi_{a,b}(x) \rangle = \int_{-\infty}^{\infty} |a|^{-1/2} \psi\left(\frac{x-b}{a}\right) f(x) dx \quad (2.3)$$

如果满足相容性条件，则小波变换是可逆的，其小波反变换为：

$$f(x) = \frac{1}{W_h} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} W_f(a,b) \psi_{a,b}(x) da db \quad (2.4)$$

在连续小波变换中，参数 a 的伸缩和参数 b 的平移为连续取值，它主要应用于理论分析方面。小波变换体现了一种以不同尺度观察信号，以不同分辨率分析信号的思想。小波基函数在频域上表现为一个带通滤波器，而小波变换则表现为信号被一组带通滤波器滤波。特别地，这组带通滤波器的中心频率和带宽分别随着 a 、 b 的变化而变化。所以小波变换是把信号分解到一系列非等宽频带上，正是由于这种频带的变化使小波变换可以适应信号的不规则性而局部地表征它们，即在不同尺度或分辨率上表示和分析它们。当式 (2.3) 中 a 值较小时，频率成分主要是在高频区域，频响宽度宽，同时时间位移步进很小，意味着对信号细节的观察；反之，若 a 增大，则小波膨胀，低频成分占主导，频响宽度窄，时间位移步进大，意味着对信号的大范围观察。这就是小波特有的“变焦”特性，这种特性使得小波很好地实现了时间（空间）域和频率域的局部化，可以随着频率变化实现多分辨率分析。

2.1.3 离散小波变换

在式 (2.1) 中， (a,b) 皆为连续变量且在 $f(x)$ 的连续小波变换描述中具有冗余性。因此没有必要针对所有可能的 (a,b) 值去计算 $W_f(a,b)$ ，而针对有限 (a,b) 值的计算就有实际上的必要。

当 (a,b) 为离散值时，若令 $a = a_0^j$ ， $b = kb_0 a_0^k$ 其中 $j, k \in Z, a_0, b_0 > 0$ ，则对应离

散小波函数为^{[12][13]}:

$$\psi_{j,k}(x) = a_0^{-j/2} \psi(a_0^{-j}x - kb_0) \quad (2.5)$$

离散小波变换为:

$$C_{j,k}(f) = \langle \psi_{j,k}(x), f \rangle = a_0^{-j/2} \int \psi(a_0^{-j}x - kb_0) f(x) dx \quad (2.6)$$

如果信号 $f(x)$ 也是离散的, 记为 $f(n)$, 则有:

$$C_{j,k}(f) = \sum_n f(n) \overline{\psi_{j,k}(n)} \quad (2.7)$$

离散小波变换也是一种时频分析, 它从集中在某个区间的基本函数开始, 以规定的步长向左或右移动基本波形, 并用标度因子 a_0 来加以扩张或压缩以构造其函数系。离散小波变换能为信号分析与合成提供足够的信息, 此外还可降低计算机的资源消耗和计算量。

在离散小波变换中, 滤波器将在不同的尺度条件下截断信号的某些频率成分: 信号通过不同的高通滤波器得到一系列的信号高频成分, 通过不同的低通滤波器得到一系列的低频成分, 这样便能分析信号的不同频率成分。

这种离散化的基本思想体现了小波变换作为“数字显微镜”的主要功能, 选择适当的放大倍数, 在一定特定的位置研究一个函数或信号过程, 再平移到另一个位置继续研究。

2.1.4 多分辨率分析

当式 (2.5) 中的 a_0 取值为 2, b_0 取值为 1 时, 我们就可以得到二进制小波, 用二进制小波变换所得的小波系数仍含有大量的冗余性, 这是因为二进制小波族的每个小波之间可能是线性相关的。如果能找到 $L^2(R)$ 空间的一个正交小波基, 用它来进行变换, 则所得到的的小波系数就不会有冗余了, 这就存在一个选取小波基的问题。八十年代, Mallat 提出了多分辨率分析的概念, 将那之前的所有正交小波基的构造统一起来, 并给出了信号分解为不同频率通道及重构的算法。

任何小波把 $L^2(R)$ 都分解成子空间 $W_j, j \in Z$ 的某种直和。这个意义上, 每个子空间 W_j 是函数集合 $\varphi_{j,k}(x) = 2^{-j/2} \varphi(2^{-j}x - k), j \in Z$ 的线性张成在 $L^2(R)$ 中的闭包。因此, 相应的子空间

$$V_j = \dots + W_{j-2} + W_{j-1}, j \in Z \quad (2.8)$$

形成 $L^2(\mathbb{R})$ 中的一个嵌套序列, 他们的并在空间 $L^2(\mathbb{R})$ 中是稠密的, 它们的交是零空间 $\{0\}$, 即具有下述性质:

- (1) 单调性: $\dots \subseteq V_{j-1} \subseteq V_j \subseteq V_{j+1} \subseteq \dots, j \in \mathbb{Z}$;
- (2) 逼进性: $\bigcup_{j \in \mathbb{Z}} V_j = L^2(\mathbb{R}), \bigcap_{j \in \mathbb{Z}} V_j = \{0\}$;
- (3) 伸缩性: $f(x) \in V_j \Leftrightarrow f(2x) \in V_{j+1}, j \in \mathbb{Z}$;
- (4) 平移不变性: $f(x) \in V_0 \Leftrightarrow f(x-k) \in V_0, j \in \mathbb{Z}$;
- (5) Riesz基存在性: 存在 $\phi(x) \in V_0$ 使 $\{\phi(x-k) | k \in \mathbb{Z}\}$ 构成 V_0 的Riesz基。

其中 (1) 描述的是子空间 V_j 的序列是嵌套的; (2) 描述的是 $L^2(\mathbb{R})$ 中的每个函数 $f(x)$ 能够用它在 V_j 中的投影 $p_j f(x)$ 非常接近希望的逼进。但另一方面, 如 (2) 中保证的, 通过减少 j , 投影 $p_j f(x)$ 能够具有任意小的能量。

一个函数 $\phi \in L^2(\mathbb{R})$, 如果它生成的子空间 $V_0 = \{\phi_{0,k}, k \in \mathbb{Z}\}$ 意义上满足 (1)、(2)、(3) 的闭子空间 V_j 的一个嵌套序列, 并且 $\{\phi(x-k), k \in \mathbb{Z}\}$ 是 V_0 的一组标准正交基, 那个 $\{\phi_j\}$ 是 $L^2(\mathbb{R})$ 的一个多分辨率分析 (MRA)。生成 MRA 的 V_j 称为一个尺度函数。由于 $V_0 \in V_1, W_0 \in V_1$, 而且 V_1 是用 $\{\phi_{1,k}(x) = 2^{1/2} \phi(2x-k), j \in \mathbb{Z}\}$ 生成的, 所以 V_0 和 W_0 中的任何函数都可以由 $\{\phi_{1,k}(x) = 2^{1/2} \phi(2x-k), j \in \mathbb{Z}\}$ 线性组合, 特别的:

$$\phi(x) = \sum h_n \sqrt{2} \phi(2x-n) \tag{2.9}$$

$$\phi(x) = \sum g_n \sqrt{2} \phi(2x-n) \tag{2.10}$$

系数 h_n 为 $\langle \phi(x), \sqrt{2} \phi(2x-n) \rangle$, 系数 g_n 为 $\langle \phi(x), \sqrt{2} \phi(2x-n) \rangle$ 。式 (2.9) 和式 (2.10) 称为尺度函数与小波函数的双尺度方程。

从 (3) 中可以看出 $\{V_j\}$ 任意一空间的基可由另一个空间的基经过简单的伸缩平移变换得到, 所以可以证明 $\{\phi_{j,k}(x) = 2^{j/2} \phi(2^j x - k), j \in \mathbb{Z}\}$ 是 $\{V_j\}$ 的一组标准正交基。

多分辨率分析从物理意义上来讲, 可以把信号分解成不同频率段的分量, 以便加以分析, 下面我们来看 Mallat 分解与重构。

Mallat 在 1986 年将计算机视觉领域的多分辨率分析的思想巧妙地引入小波分析中, 从而统一了在此之前各种小波基的构造方法, 并给出了一种子带滤波器结构的离散小波变换与重构算法, 即所谓的 Mallat 算法。Mallat 算法在小波分析中的地位相当于傅立叶

变换中的 FFT，奠定了离散小波变换在图像处理、图像编码等领域中的应用基础，在实际应用中非常重要。通过它可以实现信号多分辨的快速分解，使小波分析方法在信号处理领域真正得以实用化。

Mallat 算法的基本思想是将被分析的函数分解成不同尺度下的“模糊的像”和对该“像”的细节补充。Mallat 算法是一种离散小波变换算法，它比计算一组完整内积更为有效。它以迭代的方式使用双带子带编码并自底向上地建立小波变换，也就是，首先计算小尺度下的系数。

由多分辨率分析的理论可知，任意信号 $f(x) \in V_{j-1}$ 在 V_{j-1} 空间中的展开式为：

$$f(x) = \sum_k c_{j-1,k} 2^{(-j+1)/2} \phi(2^{-j+1}x - k) \quad (2.11)$$

将 $f(x)$ 分解一次（即投影到 V_j, W_j 空间），则有：

$$f(x) = \sum_k c_{j,k} 2^{-j/2} \phi(2^{-j}x - k) + \sum_k d_{j,k} 2^{-j/2} \psi(2^{-j}x - k) \quad (2.12)$$

其中， $c_{j,k}$ 是 $f(x)$ 在 V_j 空间的离散逼近，称为尺度系数。 $d_{j,k}$ 是 $f(x)$ 在 W_j 空间的离散值，称为小波系数，也就是我们要求的小波变换 $(W_{\psi,f})(j,k)$ 。

$$c_{j,k} = \langle f(x), \phi_{j,k}(x) \rangle = \int_{-\infty}^{\infty} f(x) 2^{-j/2} \overline{\phi(2^{-j}x - k)} dx \quad (2.13)$$

$$d_{j,k} = \langle f(x), \psi_{j,k}(x) \rangle = \int_{-\infty}^{\infty} f(x) 2^{-j/2} \overline{\psi(2^{-j}x - k)} dx \quad (2.14)$$

多分辨率分析理论下尺度函数 $\phi(x)$ 和小波函数 $\psi(x)$ 具有以下重要的性质：设 $\phi(x)$ 和 $\psi(x)$ 分别是尺度空间 V_0 和小波空间 W_0 的标准正交基函数，由于 $V_0 \subset V_{-1}, W_0 \subset V_{-1}$ ，所以 $\phi(x)$ 和 $\psi(x)$ 也必然属于空间 V_{-1} ，也即 $\phi(x), \psi(x)$ 可用 V_{-1} 空间正交基 $\phi_{-1,n}(x)$ 线性展开：

$$\phi(x) = \sum_n h(n) \phi_{-1,n}(x) = \sqrt{2}h(n)\phi(2x - n) \quad (2.15)$$

$$\psi(x) = \sum_n g(n) \phi_{-1,n}(x) = \sqrt{2}g(n)\phi(2x - n) \quad (2.16)$$

其中展开的系数 $h(n), g(n)$ 分别为：

$$h(n) = \langle \phi(x), \phi_{-1,n}(x) \rangle \quad (2.17)$$

$$g(n) = \langle \varphi(x), \phi_{-1,n}(x) \rangle \quad (2.18)$$

式 (2.17) 和式 (2.18) 描述的是相邻二尺度空间基函数之间的关系, 被称为二尺度细分方程^[11]。

将二尺度细分式 (2.17) 和式 (2.18) 对时间进行伸缩和平移, 得到

$$\phi(2^{-j}x - k) = \sum_n h(n)\sqrt{2}\phi(2^{-j+1}x - 2k - n) \quad (2.19)$$

$$\varphi(2^{-j}x - k) = \sum_n g(n)\sqrt{2}\phi(2^{-j+1}x - 2k - n) \quad (2.20)$$

令 $m = 2k + n$, 则式 (2.19) 和式 (2.20) 两式分别变为

$$\phi(2^{-j}x - k) = \sum_n h(m - 2k)\sqrt{2}\phi(2^{-j+1}x - m) \quad (2.21)$$

$$\varphi(2^{-j}x - k) = \sum_n g(m - 2k)\sqrt{2}\phi(2^{-j+1}x - m) \quad (2.22)$$

将式 (2.21) 和式 (2.22) 分别代入式 (2.13) 和式 (2.14) 得:

$$c_{j,k} = \sum_m h(m - 2k) \int_{-\infty}^{+\infty} f(x) 2^{(-j+1)/2} \overline{\phi(2^{-j+1}x - m)} dx \quad (2.23)$$

$$d_{j,k} = \sum_m g(m - 2k) \int_{-\infty}^{+\infty} f(x) 2^{(-j+1)/2} \overline{\phi(2^{-j+1}x - m)} dx \quad (2.24)$$

由于

$$\int_{-\infty}^{+\infty} f(x) 2^{(-j+1)/2} \overline{\phi(2^{-j+1}x - m)} dx = c_{j-1,m} \quad (2.25)$$

所以, 式 (2.23) 和式 (2.24) 分别变为:

$$c_{j,k} = \sum_m h(m - 2k)c_{j-1,m} \quad (2.26)$$

$$d_{j,k} = \sum_m g(m - 2k)c_{j-1,m} \quad (2.27)$$

式 (2.26) 和式 (2.27) 说明, V_j 的尺度系数和 W_j 空间的小波系数可由 V_{j-1} 空间的尺度系数经过滤波器系数 $h(n)$ 和 $g(n)$ 进行加权求和得到。

将 V_j 空间的尺度系数 $c_{j,k}$ 进一步按上面的两式分解下去, 可分别得到 V_{j+1} , W_{j+1} 空

间的尺度系数 $c_{j+1,k}$ 和小波系数 $d_{j+1,k}$ ，式 (2.26) 和式 (2.27) 给出的就是著名的 Mallat 塔式分解算法^[12]，也就是离散小波变换的快速算法。

用类似于分解的思路不难推出重建过程，图 2.1 给出了对一维信号进行小波分解与重构的示意图^[13]。采用可分离滤波器的形式，很容易将上述结论推广到二维情况，图 2.2 演示了二维小波变换的分解和重构。

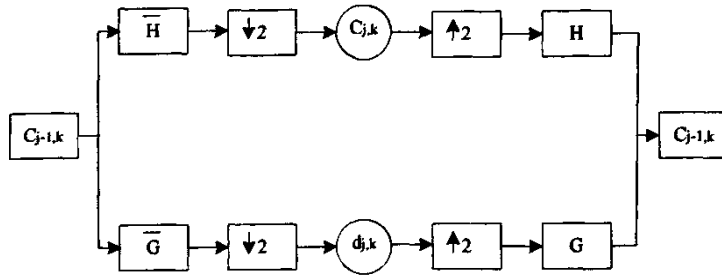


图 2.1 一维小波分解与重构结构示意图

Fig. 2.1 1-D Wavelet Decomposition and Reconstruction

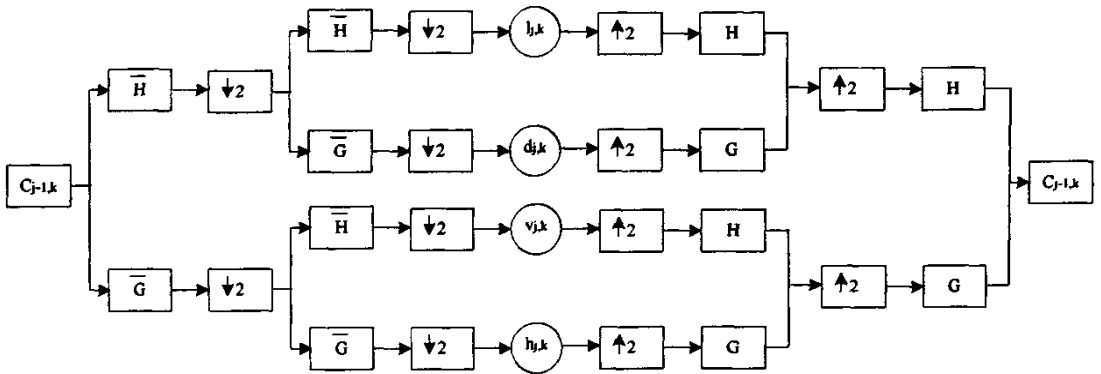


图 2.2 二维小波分解与重构结构示意图

Fig. 2.2 2-D Wavelet Decomposition and Reconstruction

2.1.5 小波基的选择

和傅立叶变换相比，小波变换具有很大的灵活性，其中一个重要的方面就是傅立叶变换具有唯一的正弦型基函数，数学性质比较简单。而小波变换理论上有无数个小波基可供选择，选用不同的小波基对于信号或图像处理的效果有很大的影响。这种灵活性一方面是小波变换性能比傅立叶变换有根本提高的原因，另一方面，也给小波变换的应用带来了一个难题。已经有很多的文献对怎样正确选择小波基的问题进行过理论上的论

述,但这个问题并没有从本质上得到解决。在很长一段时间内,小波基的选择仍然将是小波应用研究的一个热点。

同一幅图像用不同的小波基进行分解所得到的数据压缩效果是不同的。在应用于图像压缩时,希望经小波分解后得到的三个方向的细节分量具有高度的局部相关性,而整体相关性被大部分完全解除。由于小波变换是将原始图像与小波基函数以及尺度函数进行内积运算,1989年 Daubechies 基于离散滤波器迭代的方法构造了紧支集的规范正交小波基,因而内积运算转换为信号和离散滤波器的卷积运算,小波变换中的小波基的选择转换为正交镜像滤波器组的选择。

在选取小波基时,应综合考虑以下四个因素:

(1) 小波基的正则性阶数^[14]与图像数据压缩效果的关系。正则性阶数越高,图像压缩后重建的效果越好(正则性是函数光滑程度的一种描述,也是函数频域能量集中的一种度量)。

(2) 待处理图像与小波基的相似性。小波基函数的基本图像与待编码图像的结构越相似,压缩效果就越好。

(3) 由于在图像处理中数据量特别大,因而不能片面追求高压缩比,而应综合考虑压缩效率和计算复杂程度。由于图像数据压缩中的小波变换通常是由图像信号与滤波器的离散卷积实现,因而滤波器长度不能太长,否则计算量太大而没有实用价值。一般当滤波器长度大于 10 以后,图像压缩重建以后的峰值信噪比(PSNR)的改善效果就不明显了。

(4) 小波函数的能量集中特性应好。若变换的能量集中特性越好,则数据压缩的效果也越好。

2.1.6 小波的边界问题

边界失真主要是正交镜像滤波器的非线性相位特性、信号自身在边界附近的相关性以及变换结果亚抽样所造成的。在信号的子带分析中,子带分析重构系统必须满足下列两个条件:

(1) 完全重构性

原始信号可以由它的子带信号完全重构,正交镜像滤波器组组成的系统是完全重构的系统。

(2) 子带信号的数据点数的总和不应多于原始信号的数据点数

对于无限长度的信号，它们的频带是严格受限的，根据抽样定理，其子带信号进行严格抽样就能满足条件（1）和（2）。然而，对于有限长度的信号，信号经过小波变换滤波，子带信号的数据点的总和将大于原始信号的数据点数，从而引起边界外延。如果满足条件（1）的完全重构性，子带信号在严格抽样时的数据点数将增加，不能满足条件（2）。如果去掉因滤波而增加的点数以满足条件（2），则由于信息的丢失，重构信号将产生畸变，不能满足条件（1）。为了同时满足两个条件，必须对原始信号进行边界延拓，形成一无限信号，以减少信息的丢失^[20]。

延拓方法包括以下几种：零填充延拓、周期延拓、边界重复、对称周期延拓、双对称延拓。其中前两种延拓方法使边界不连续，后三种方法则能保持信号的连续性。从符合重构条件来说，只有周期延拓是符合重构条件的，因为周期延拓后的信号与有限长度滤波器卷积后的信号仍是周期信号。由于运算后仍只取一个周期，因而在复频率 Z 域仍满足重构条件。对称周期延拓时，若滤波器具有对称结构也可实现完全重构。其他几种延拓方法均不能完全满足重构原始信号的条件，但是由于周期延拓使边界不连续，对低比特率量化不利^{[6][19]}。

2.2 提升小波变换的理论

2.2.1 提升小波的发展和特点

目前图像处理常用的小波变换是通过 Mallat 算法来实现的，通常的小波滤波器都是小数形式。这样，当实际的数字图像采用整数表示的时候，滤波器的输出的结果就不是整数形式，此时的小波变换也不能实现无失真的重构。1994 年 Sweldens 提出了一种不依赖于傅立叶变换的新的的小波构造方法—提升方法，也称为第二代小波或整数小波变换，该方法能够将整数映射到整数，实现图像的无损的压缩。

提升结构的主要优点是有益于硬件实现，能进行快速原位运算。提升结构不需要额外的存储空间，运算复杂度低，易于可逆变换，能比一般卷积运算方法少的运算次数实现小波变换，而得到的小波系数与使用传统小波变换得到的结果相同。

提升算法的主要特点：

- （1）继承了第一代小波的多分辨率的特性；
- （2）不依赖傅立叶变换；
- （3）小波变换后的系数是整数；
- （4）图像的恢复质量与变换时边界采用何种延拓方式无关。

2.2.2 正变换和逆变换

基于提升的滤波由简单的序列组成，基本思想非常简单，小波分解过程正变换主要有三个步骤 分裂 (split)、预测 (predict)、更新 (update) 可以实现。如图2.3:

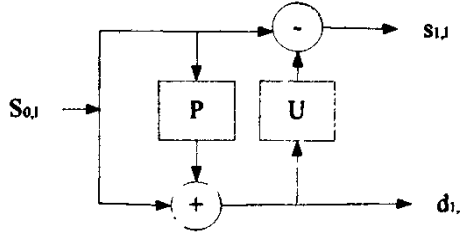


图 2.3 提升小波分解

Fig. 2.3 Lift Wavelet Decomposition

以 $s_{0,j}$ 表示原始数据， $s_{1,j}$ 和 $d_{1,j}$ 分别表示小波变换后的低通系数和高通系数。

(1) 分裂 (split)

此过程仅仅将一数据集 (以一维为例) s_j 分裂成两个较小的子集合 s_{j-1} 和 d_{j-1} ，一般情况下分裂为奇偶两个子集合，即：

$$\text{Split}(s_j) = (\text{even}_{j-1}, \text{odd}_{j-1}) = (s_{1,j}^{(0)}, d_{1,j}^{(0)}) \quad (2.28)$$

(2) 预测 (predict)

在预测阶段，应该保持偶样本不变，利用插值细分的方法预测奇样本系数，原值与预测之差作为细节系数，又称小波系数，表示为：

$$d_{j-1} = d_{j-1} - P(s_{j-1}) \quad d_{1,j}^i = d_{1,j}^{(i-1)} - \sum_k p_k^i s_{1,j-k}^{i-1} \quad (2.29)$$

式中预测算子P用来表示预测函数，预测函数可以通过插值拟合的方法拟合一个多项式，多项式的节数称为插值拟合阶矩，用N表示，常用的有分段线形插值和立方插值，也可进行更高的阶次插值， d_{j-1} 越小表示预测的越精细，拟合的越好。

(3) 更新 (update)

经过上面两个步骤还不能在子数据集 s_{j-1} 中维持原始数据集 s_j 中的某些整体性质，如均值等。因此，必须采用更新过程。更新的基本思想是找出一个更使之保持原始数据集 s_{j-1} 的一些尺度特性 $Q(\cdot)$ ，如均值、消失矩等的不变性，即：

$$Q(s_{j-1}) = Q(s_j) \quad (2.30)$$

构造一个算子U去更新 s_{j-1} ，定义如下：

$$s_{j-1} = s_{j-1} + U(d_{j-1}) \quad s_{1,j}^{(i)} = s_{1,j}^{(i-1)} - \sum_k u_k^i s_{1,j-k}^{(i-1)} \quad (2.31)$$

对于逆变换操作比较简单，主要是将操作过程逆过来，反转符号，就可以得到反变换，图2.4是利用提升方法实现小波反变换的框图。

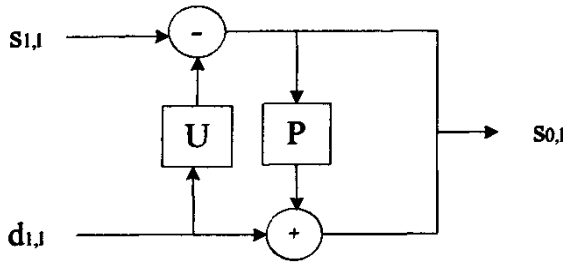


图 2.4 提升小波重构示意图
Fig. 2.4 Lift Wavelet Reconstruction

逆变换的主要过程为：

$$\begin{cases} s_{1,j}^{(N)} = Ks_{1,j} \\ d_{1,j}^{(N)} = d_{1,j} / K \\ s_{1,j}^{(i-1)} = s_{1,j}^{(i)} + \sum_k u_k^{(i)} d_{1,j-k}^{(i)} \\ d_{1,j}^{(i-1)} = d_{1,j}^{(i)} + \sum_k p_k^{(i)} s_{1,j-k}^{(i-1)} \\ s_{1,2l} = s_{1,j}^{(0)} \\ s_{1,2l+1} = d_{1,j}^{(0)} \end{cases} \quad (2.32)$$

其中利用不同的预测算子 P 和更新算子 U 可以建立不同的小波变换，K 为修正因子。

2.2.3 整型小波变换

任何小波变换都可以用提升方法实现。但从上述分析可以看出，基于提升算法的小波分解操作，其输出结果仍然为小数。上述公式同时又表明，输出结果的小数部分是由其中的提升和双提升运算引入的，对其取整数可得到整数变换的结果。取整数操作相当于对原来的小波滤波器系数做了小的改动，但是小波分解的特性仍然保留。例如下面的这个例子，在加/减之前，对每个提升运算的结果取整，小波变换就可以实现整型到整型的映射。

整型双提升可表示为：

$$s_{1,j}^{(i)} = s_{1,j}^{(i-1)} - \left[\sum_k u_k^{(i)} s_{1,j-k}^{(i-1)} + 1/2 \right] \quad (2.33)$$

$$d_{1,j}^{(i)} = d_{1,j}^{(i-1)} - \left[\sum_k p_k^{(i)} s_{1,j-k}^{(i-1)} + 1/2 \right] \quad (2.34)$$

整型的基本提升变换过程可表示为:

$$\begin{cases} s_{1,j}^{(0)} = s_{0,2j} \\ d_{1,j}^{(0)} = s_{0,2j+1} \\ \text{for}(i=1; i < N; i++) \\ \quad d_{1,j}^{(i)} = d_{1,j}^{(i-1)} - \left[\sum_k p_k^{(i)} s_{1,j-k}^{(i-1)} + 1/2 \right] \\ \quad s_{1,j}^{(i)} = s_{1,j}^{(i-1)} - \left[\sum_k u_k^{(i)} d_{1,j-k}^{(i-1)} + 1/2 \right] \\ \text{end for} \end{cases} \quad (2.35)$$

因此, 采用提升方案的整型可逆小波变换可以表示为:

$$\begin{cases} \text{for } i = N; i > 1; i-- \\ \quad s_{1,j}^{(i-1)} = s_{1,j}^{(i)} + \left[\sum_k u_k^{(i)} d_{1,j-k}^{(i)} + 1/2 \right] \\ \quad d_{1,j}^{(i-1)} = d_{1,j}^{(i)} + \left[\sum_k p_k^{(i)} s_{1,j-k}^{(i)} + 1/2 \right] \\ \text{end for} \\ s_{0,2j+1} = d_{1,j}^{(0)} \\ s_{0,2j} = s_{1,j}^{(0)} \end{cases} \quad (2.36)$$

2.2.4 双正交小波

在本节中主要讨论本论文中用到的双正交对成小波 9/7 完全重构滤波器。消失矩条件是构造小波的必要条件, 因此, 获得对称双正交小波完全重构滤波器 $\{h, g, \tilde{h}, \tilde{g}\}$ 。设 N 和 \tilde{N} 分别表示小波及其对偶的消失矩长度, 也就是 $h^{(k)}(-1) = 0, k = 0, 1, \dots, N-1$ 和 $g^{(k)}(1) = 0, k = 0, 1, \dots, \tilde{N}-1$ [4]。

对于 9/7 对成双正交完全重构小波滤波器, 设 $h_k = h_{-k}$ 和 $\tilde{h}_k = \tilde{h}_{-k}$, $k = 0, 1, 2, 3, 4$, 得到:

$$\begin{cases} h_e(z) = h_0 + h_2(z + z^{-1}) + h_4(z^2 + z^{-2}) \\ h_0(z) = h_1(z + 1) + h_3(z^2 + z^{-1}) \end{cases} \quad (2.37)$$

$$\begin{cases} g_e(z) = -\tilde{h}_0(z^{-1}) = -[\tilde{h}_1(1 + z^{-1}) + \tilde{h}_3(z + z^{-2})] \\ g_0(z) = \tilde{h}_e(z^{-1}) = \tilde{h}_0 + \tilde{h}_2(z + z^{-1}) \end{cases} \quad (2.38)$$

对于 $h_3 \neq 0$ 时, 应用 Euclidean 算法, 得到下面的提升结构:

$$P(z) = \begin{bmatrix} h_e(z) & g_e(z) \\ h_0(z) & g_0(z) \end{bmatrix} = \begin{bmatrix} 1 & s_0(1+z^{-1}) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ s_1(1+z) & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1/t_3 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1/t_3 \end{bmatrix} \quad (2.39)$$

对于任意给定的系数 $h_k = h_{-k}$ 和 $\tilde{h}_k = \tilde{h}_{-k}$, 式 (2.39) 并不是构成小波的充分条件, 为了获得 9/7 小波滤波器, 还需要引入一个新的条件: 消失矩满足 $N = 2$ 和 $\tilde{N} = 4$, 得到:

$$h^{(k)}(z)|_{z=1} = 0, k = 0, 1 \quad (2.40)$$

$$g^{(k)}(z)|_{z=1} = 0, k = 0, 1, 2, 3 \quad (2.41)$$

则式 (2.39) 可以等价于:

$$P(z) = \begin{bmatrix} 1 & \alpha_0(1+z^{-1}) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \beta(1+z) & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \gamma(1+z) & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \delta(1+z) & 1 \end{bmatrix} \begin{bmatrix} \zeta & 0 \\ 0 & 1/\zeta \end{bmatrix} \quad (2.42)$$

基于消失矩条件等式 (2.42) 和归一化条件 $h(1) = 2, \tilde{h}(1) = 1$, 得到下面 5 个方程的方程组:

$$\begin{cases} 1 + \delta(4\alpha + 4\gamma - 2) + 2(2\alpha - 1)\beta(1 + 4\gamma\delta) = 0 \\ 1 + 2\alpha + 2\gamma + 4\beta\gamma + 8\alpha\beta\gamma = 0 \\ 2 + 6\alpha + 6\gamma + 16\beta\gamma + 40\alpha\beta\gamma = 0 \\ [1 + \delta(4\alpha + 4\gamma + 2) + 2(2\alpha + 1)\beta(4\gamma\delta + 1)]\zeta = 2 \\ 1 + (4\beta - 2)\gamma - 2\alpha(1 + 4\beta\gamma) = \zeta \end{cases} \quad (2.43)$$

方程的解表示为:

$$\begin{cases} \alpha = \frac{-2t+1}{4(t-1)} \\ \beta = -(t-1)^2 \\ \gamma = \frac{1}{4t(t-1)} \\ \delta = t^3 - \frac{7}{4}t^2 + t \\ \zeta = \frac{2}{t} \end{cases} \quad (2.44)$$

从式 (2.43) ~ 式 (2.44), 得到了一个带自有变量的双正交 9/7 完全重构滤波器。

为了得到双正交 9/7 小波滤波器, 需要应用 Daubechies 不等式来确定参数的范围。

首先, 定义 $h_t(z) = (\frac{1+z^{-1}}{2})^2 F(z)$ 和 $\tilde{h}_t(z) = (\frac{1+z^{-1}}{2})^2 Q(z)$, 这里 $F(z)$ 和 $Q(z)$ 都是包含参数 t 的多项式。对于整数 k , 解下面的不等式:

$$B_k = \text{Sup}_{t \in R, |z|=1} |F(z)F(z^2)\dots F(z^{2^{k-1}})| < 2^{\frac{3}{2}} \quad (2.45)$$

$$\tilde{B}_k = \text{Sup}_{t \in R, |z|=1} |Q(z)Q(z^2)\dots Q(z^{2^{k-1}})| < 2^{\frac{7}{2}} \quad (2.46)$$

当 $k = 40$ 时, 得到当 $t \in [0.780, 1) \cup (1.1, 1.852]$, 满足式 (2.46)。我们能够得到双正交 9/7 小波。特别地, 如果 $t = 1.25$, 那么 9/7 小波系数就是:

$$\begin{cases} \{h(0), h(1), h(2), h(3), h(4)\} = \frac{1}{10} \left\{ \frac{190}{16}, \frac{86}{16}, \frac{24}{16}, \frac{6}{16}, \frac{9}{16} \right\} \\ \{\tilde{h}(0), \tilde{h}(1), \tilde{h}(2), \tilde{h}(3)\} = \left\{ \frac{18}{32}, \frac{19}{64}, -\frac{1}{32}, -\frac{3}{64} \right\} \\ h(-k) = h(k) \\ \tilde{h}(-k) = \tilde{h}(k) \end{cases} \quad (2.47)$$

从而可以实现了整数变换。

2.2.5 常见提升方法举例

下面介绍几种采用提升方法获得整型小波变换的例子^[14]

(1) S 变换

S 变换是最古老的整型小波变换, 它是 haar 变换的整型表示。图中 $\lfloor \cdot \rfloor$ 表示取整运算。

$$d_{1,j} = s_{0,2j+1} - s_{0,2j} \quad s_{1,j} = s_{0,2j} - \lfloor d_{0,j} / 2 \rfloor \quad (2.48)$$

(2) S+P 变换

Said 和 Pearlman 进一步提出了 S+P 变换, 即通过线性变换, 在低通系数 $s_{1,j}$ 的基础上, 进行线性预测, 产生一组新的高通系数。其中主要的变换有以下几种:

(a) (2, 2) 变换

$$\begin{cases} d_{1,j} = s_{0,2j+1} - \lfloor 1/2(s_{0,2j} + s_{0,2j+2}) + 1/2 \rfloor \\ s_{1,j} = s_{0,2j} - \lfloor 1/4(d_{1,j-1} + d_{1,j}) + 1/2 \rfloor \end{cases} \quad (2.49)$$

(b) (4, 2) 变换

$$\begin{cases} d_{1,j} = s_{0,2j+1} - \lfloor 9/16(s_{0,2j} + s_{0,2j+2}) - 1/16(s_{0,2j-2} + s_{0,2j+4}) + 1/2 \rfloor \\ s_{1,j} = s_{0,2j} - \lfloor 1/4(d_{1,j-1} + d_{1,j}) + 1/2 \rfloor \end{cases} \quad (2.50)$$

(c) (2, 4) 变换

$$\begin{cases} d_{1,j} = s_{0,2l+1} - \left[1/2(s_{0,2l} + s_{0,2l+2}) + 1/2 \right] \\ s_{1,j} = s_{0,2l} - \left[19/64(d_{1,j-1} + d_{1,j}) - 3/64(d_{1,j-2} + d_{1,j+1}) + 1/2 \right] \end{cases} \quad (2.51)$$

(d) (4, 4) 变换

$$\begin{cases} d_{1,j} = s_{0,2l+1} - \left[9/16(s_{0,2l} + s_{0,2l+2}) - 1/16(s_{0,2l-2} + s_{0,2l+4}) + 1/2 \right] \\ s_{1,j} = s_{0,2l} - \left[9/32(d_{1,j-1} + d_{1,j}) - 1/32(d_{1,j-2} + d_{1,j+1}) + 1/2 \right] \end{cases} \quad (2.52)$$

(e) (6, 2) 变换

$$\begin{cases} d_{1,j} = s_{0,2l+1} - \left[\frac{75}{128}(s_{0,2l} + s_{0,2l+2}) - \frac{25}{256}(s_{0,2l-2} + s_{0,2l+2}) \right. \\ \left. + \frac{3}{256}(s_{0,2l-4} + s_{0,2l+6}) + 1/2 \right] \\ s_{1,j} = s_{0,2l} - \left[1/4(d_{1,j-1} + d_{1,j}) + 1/2 \right] \end{cases} \quad (2.53)$$

(f) (2+2, 2) 变换

$$\begin{cases} d_{1,j}^{(1)} = s_{0,2l+1} - \left[1/2(s_{0,2l} + s_{0,2l+2}) + 1/2 \right] \\ s_{1,j} = s_{0,2l} - \left[1/4(d_{1,j-1}^{(1)} + d_{1,j}^{(1)}) + 1/2 \right] \\ d_{1,j} = s_{0,2l} - \left[\alpha(-1/2s_{1,j-1} + s_{1,j} - 1/2s_{1,j+1}) \right. \\ \left. + \beta(-1/2s_{1,j} + s_{1,j+1} - 1/2s_{1,j+2}) + \gamma d_{1,j+1}^{(1)} + 1/2 \right] \end{cases} \quad (2.54)$$

(g) D4 变换

$$\begin{cases} d_{1,j}^{(1)} = s_{0,2l+1} - \left[\sqrt{3}s_{0,2l} + 1/2 \right] \\ s_{1,j} = s_{0,2l} - \left[\sqrt{3}/4(d_{1,j}^{(1)} + (\sqrt{3}-2)/4d_{1,j-1}^{(1)}) + 1/2 \right] \\ d_{1,j} = d_{1,j}^{(1)} - s_{1,j+1} \\ K = (\sqrt{3}+1)/\sqrt{2} \end{cases} \quad (2.55)$$

JPEG2000 标准中建议采用 Daubechies 9/7 滤波器进行有损变换，采用 LeGall5/3 滤波器进行无损变换，他们都可以按上述的提升算法来实现。Daubechies 9/7 滤波器的计算公式为第 (g) 种变换，LeGall5/3 滤波器的计算公式为第 (a) 种变换。因此在本论文的提升算法中采用 9/7 双正交小波实现提升变换。

2.2.6 提升算法实现

本节列出主要函数模块正提升变换并调用一维地形式，其它地函数省略，采用双正交对称 9/7 小波：

函数 2.1 VCLiftWavelet2D (int *data, int height, int width, int level)

说明: data 存储块数据, height 宏块高, width 宏块宽, level 分解层次

过程:

```

for (m = 0; m < level; m++)
{
    for (n = 0; n < height; n++)
    {
        CopyRow (buffer, data + n * width, width);
        //一维提升变换
        WaveletForward1D (buffer, width);
        CopyRow (data + n * width, buffer, width);
    }
    for (n = 0; n < width; n++)
    {
        CopyCol (buffer, data + n, width, height);
        WaveletForward1D (buffer, height);
        CopyColBack (data + n, buffer, width, height);
    }
}

```

因为是整型小波变换, 变换后的结果也可以是整数, 所以在后来的量化中的损失可以避免舍去精度值, 减少整个帧图像的能量损失。

函数 2.2 VCLiftWavelet1D (int *data, int Datalen)

说明: data 存储块数据, Datalen 数据长度

过程:

```

int mid = Datalen/2;
for (c=0;c<mid;c++)
{
    * (s+c) =* (data+2*c);
    * (d+c) =* (data+2*c+1);
}

for (c=0;c<mid-1;c++)
{
    * (d+c) += (int) floor (ALPHA* (* (s+c) +* (s+c+1)) +0.5);
}
* (d+mid-1) += (int) floor (ALPHA* (* (s+mid-1)) *2+0.5);
*s+= (int) floor (BELTA* (*d) *2+0.5);
for (c=1;c<mid;c++)
{
    * (s+c) += (int) floor (BELTA* (* (d+c-1) +* (d+c)) +0.5);
}
for (c=0;c<mid-1;c++)
{
    * (d+c) += (int) floor (GAMMA* (* (s+c) +* (s+c+1)) +0.5);
}

```

```
* (d+mid-1) += (int) floor (GAMMA* (* (s+mid-1)) *2+0.5);

*s+= (int) floor (DELTA* (*d) *2+0.5);
for (c=1;c<mid;c++)
{
* (s+c) += (int) floor (DELTA* (* (d+c-1) +* (d+c)) +0.5);
}
for (c=0;c<mid;c++)
{
* (d+c) +=* (s+c);
}
for (c=0;c<mid;c++)
{
* (s+c) += (int) floor ((ROW-1) * (* (d+c)) +0.5);
}
for (c=0;c<mid;c++)
{
* (d+c) += (int) floor (0- (* (s+c) /ROW) +0.5);
* (s+c) += (int) floor ((ROW-ROW*ROW) * (* (d+c)) +0.5);
}
for (c=0;c<mid;c++)
{
* (data+c) =* (s+c);
* (data+c+mid) =* (d+c);
}
}
```

函数 2.2 为一维提升算法，具体理论内容可参考 2.2.2 节和 2.2.3 节。

第三章 算法设计

3.1 基于小波变换的视频压缩技术

基于小波的视频图像编码技术是目前国际上的一个研究热点^[21]，它不仅能提供高的编码压缩效率，而且高度支持了可扩展性编码。JPEG2000 中引进了优化截断点的嵌入块编码方法对图像进行编码，MPEG-4 在其静止图像压缩标准中采用了零树小波编码方式，新一代视频对象编码标准 MPEG-4 将基于小波的视频编码方法纳入个优选的方案。基于小波的视频编码技术主要可分为三类：

(1) 空间域运动补偿后跟小波变换

这类小波视频编码方案采用空域运动估计和运动补偿技术消除时间冗余，再对其运动补偿残差图像采用小波变换编码方法。与传统的基于 DCT 的混合编码算法的主要差别在于对残差信号的处理，它是用小波变换代替分块 DCT 变换。就运动补偿残差图像压缩，Bhutani 和 Pearlman 提出用嵌入式零树小波 (EZW) 算法对运动补偿残差图像进行编码^[22]，取得了比 MPEG-1 标准更好的性能。Martucci 等在 EZW 算法的基础上，提出了自适应系数量化编码和优化算术编码的“零树熵 (ZTE)”视频压缩编码算法^[23]。虽然 ZTE 方法失去了嵌入特性，但其压缩率较 EZW 算法有了很大的提高，同时具有空间可分级性。Marpc 等人对小波系数进一步优化分离，在改进零树映射和使用局部自适应熵编码等预处理基础上提出了 PACC (Partitioning Aggregation and Conditional coding) 视频压缩编码算法^[24]，PACC 实质上是对 EZW 和多级树集合分裂算法 (SPIHT) 综合优化的结果，其压缩性能要高于 ZTE 方法。另外，在空间域中，基于块的 MEMC 所产生的运动矢量场往往是不连续的，不可避免会产生块效应，导致运动补偿块之间的不连续性，从而增加了高频信息，使得小波变换后的高频子带系数增大，降低了编码效率。重叠块匹配运动补偿 (OBMC) 方法^[25]缓和了这个矛盾，它使用相邻块的运动补偿预测的加权来表示当前块的预测，减少了运动补偿残差帧的整体能量，在一定程度上也消除了块间不连续性造成的块效应。

这类编码器的关键是对预测误差的编码，在面向低比特率传输过程中取得了比较好的性能。但由于它没有对运动补偿技术产生任何影响，而且采用全局小波变换，变换后的系数分布比较松散，因此难以对编码性能做较大程度的改善。

(2) 小波变换后跟小波域运动补偿

这类小波视频编码方案先对视频图像进行小波变换,然后在小波变换域中进行运动估计和运动补偿,最后对小波域运动补偿残差图像进一步编码。Zhang 等人提出了一种基于小波变换域的可变块多分辨率运动补偿(MRMC)视频编码方法^[26],它利用小波金字塔结构每层间的相关性,来降低运动补偿过程的运算复杂度。这种多分辨率运动补偿方法能适应人的视觉特性,有效地降低运动搜索和匹配时间,平滑的运动矢量场能较连贯地描述物体的连续运动,较好地应用于中和高比特率传输领域。但是,小波变换的平移可变特性限制了它在低比特率传输领域的应用效果,并且这种方法运动矢量的编码将占有比较大的比例,在极低码率的视频通信中,是不可忽视的。

对于这类编码器,需要考虑小波变换平移可变特性给运动估计和运动补偿带来的影响,在编码冗余度和编码效率之间寻求最佳平衡,同时需考虑运动估计的计算复杂度,以适当减少搜索匹配时间。

(3) 三维小波变换编码

三维小波视频编码是当前的研究热点之一,这不仅是因为具有局部时频特性的小波基更能捕捉视频图象的非平稳信息,可以获得更高的压缩比编码,更重要的是二维小波分解所提供的多分辨率数据结构为支持可伸缩编码提供了一个良好条件。所谓的伸缩性是指改变一个已压缩码流的分辨率或码率以满足在视频压缩时不能预见的要求的能力,伸缩性包括空间分辨率可伸缩、时间分辨率可伸缩、码率/信噪比(SNR)可伸缩、解码复杂度可伸缩等以及它们的组合。

三维小波变换是二维变换在时间轴上的扩展,通过对视频序列进行三维小波变换,消除了其在时间和空间方向上的相关性。先对原始视频序列进行分组,T帧为一组,每一组为一个变换单元,进行时间、垂直和水平三个方向上的小波变换,得到不同空间-时间频率子带。对于运动幅度较大的视频序列,在时域分解时需结合运动补偿技术。最后根据各个子带的统计特性,分配适当的码率,以实现压缩。

三维小波视频编码器在可扩展性能方面表现了极强的特性,能灵活地结合时域、空域、信噪比等可分级性编码。但是三维小波变换在去除视频序列时间冗余技术上还不够成熟,尤其在低比特率传输下,视频序列间的剧烈运动会使编码效率降低,因此,三维小波变换视频编码方法主要应用于中高比特率场合。

3.2 运动估计

对于视频序列图像,由于相邻帧之间存在很大的时间相关性,即时间冗余,所以通

过减少时间冗余,可以大幅度提高视频压缩编码的效率。这方面一种有效的方法是基于块匹配的运动估计(Motion Estimation, ME),因其具有算法简单、便于VLSI实现等优点得到了广泛应用。它已经被许多视频编码标准所采纳,如MPEG-1、MPEG-2、MPEG-4、H.261、H.263和H.264等。其基本思想是将图像序列的每一帧分成 $M \times N$ 的宏块,然后对于当前帧中的每一块根据一定的匹配准则在前一帧或后一帧某一给定搜索范围内找出与当前块最相似的块,即匹配块,由匹配块与当前块的相对位置计算出运动位移,所得运动位移即为当前的运动矢量。运动估计得越准确,补偿的残差就越小,编码的效率就越高,解码出来的图像质量越好。并且,运动估计在整个系统中的计算复杂度最大,占了整个系统的50%以上。如何提高运动估计的效率,使运动估计算法的搜索过程更健壮、更快速、更高效成为目前研究的热点。

本章研究了基于块匹配的运动估计算法,详细总结了提高运动估计算法效率的三个主要技术:初始搜索点的选择、匹配准则、运动搜索策略;目前一些较为典型的基于块匹配的运动估计算法:全搜索法(FS)、对数法(TDL)、三步法(TSS)、交叉法(CS)、基于块的梯度下降法(BBGDS)和菱形法(DS)等;菱形搜索法虽然取得了非常大的成功,但是其性能上仍然存在缺陷,本文主要研究其中几个算法的优点和不足,提出了新的改进搜索算法。

3.2.1 运动估计算法研究

要想提高视频压缩编码的效率,关键是要有准确的运动估计,因此运动估计算法的研究成了多年来视频压缩编码的研究重点。运动估计算法多种多样,大体上可以把它们分成四类:块匹配法、递归估计法、贝叶斯估计法和光流法。块匹配运动估计算法因其具有算法简单、便于VLSI实现等优点得到了广泛的应用。

目前,块匹配运动估计算法中搜索精度最高的是全搜索法(FS),它对搜索范围内的每一个像素点进行匹配运算以得到一个最优的运动矢量。但它的计算复杂度太高,不适合实时应用,为此人们提出了许多快速运动估计算法。早期的三步法(TSS),对数法(TDL),交叉法(CS)等主要是通过限制搜索位置的数目来减少计算量。动态搜索窗调整法(DSWA)根据当前结果动态调整下一步搜索步长的大小,算法性能在一定程度上有了改进。新三步法(NTSS)、新四步法(NFSS)、基于块的梯度下降法(BBGDS)等利用运动矢量具有中心偏移的分布特性,提高了匹配速度,减少了陷入局部极小的可能性。预测搜索法(PSA)、自适应运动跟踪(AMTS)等利用相邻块的运动相关性选择

一个反映当前运动块趋势的预测点作为初始搜索点，以提高搜索速度和预测的准确性。1999年10月，菱形法(DS)被MPEG-4国际标准采纳并收入验证模型(VM)。虽然菱形法获得了非常大的成功，但是其性能上仍然存在缺陷：其一，对于运动大的序列，菱形法在搜索最佳点所在区域时，广度搜索和梯度下降搜索同时进行，即同等地对待搜索区域的各部分，这造成较大的搜索冗余，影响了算法的搜索速度；其二，对于保持静止的图像序列，即运动矢量为零的情况，菱形法要经历由大模板(LDSP)到小模板(SDSP)的变化过程，要对13个搜索点进行搜索，而理想情况是只需搜索5个点，即对于小运动序列DS算法还有待改进。为此，针对菱形算法的问题，又相继提出了正方形菱形法(SDS)和线性菱形并行搜索法(LSPS)的算法。随着新技术的出现和发明，块匹配运动估计算法的性能将不断得到提高。

3.2.2 运动估计的原理

运动图像多数情况下只是其中的很少一部分图像在运动，同一场景相邻的两幅图像之间在内容上的差异不会太大，或者说后一帧的内容与前一帧重复的部分很多。用数学术语来讲，二者相关的。

对于相关的视频图像，发送端不一定必须把每帧图像上的像素像素都传给接收端，而只要将物体(或摄像机)的运动信息告知接收端，接收端就可以根据运动信息和前一帧图像的内容来更新当前帧图像，这比全部传送每帧图像的具体细节所需的数据量要小得多。

要这样做，首先要解决的问题是如何从序列图像中提取有关物体运动的信息，这个过程称为运动估计(Motion Estimation, ME)，其表达方式是运动矢量(Motion Vector, MV)，运动估计研究的主要内容就是如何快速、有效地获得足够精度的运动矢量；而把前一帧相应的运动部分信息根据运动矢量补偿过来的过程称为运动补偿(Motion Compensation, MC)。运动估计的基本思想是将图像序列的每一帧分成许多互不重叠的宏块，并认为宏块内所有像素的位移量都相同，然后对于当前帧中的每一块到前一帧或后一帧某一给定搜索范围内根据一定的匹配准则找出与当前块最相似的块，即匹配块，由匹配块与当前块的相对位置计算出运动位移，所得运动位移即为当前块的运动矢量(Motion Vector)。利用搜索到的运动矢量在参考帧上进行运动补偿，补偿残差(Compensation Difference)经变换、量化、行程编码后与运动矢量共同经熵编码，然后以比特流形式传出去。

3.2.3 块匹配准则

运动估计算法中常用的匹配准则有三种，即最小绝对差（MAD）、最小均方差误差（MSE）和归一化互相函数（NCCF），下面分别介绍：

(1) 最小绝对差

$$MAD(i, j) = \frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N |f_k(m, n) - f_{k-1}(m+i, n+j)| \quad (3.1)$$

式中 (i_0, j_0) 为位移矢量， f_k 和 f_{k-1} 分别为当前和上一帧的像素值， $M \times N$ 为宏块的大小，若在某一点 (i_0, j_0) 处 $MAD(i_0, j_0)$ 达到最小，则该点为要找的最优匹配。

(2) 最小均方差

$$MSE(i, j) = \frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N [f_k(m, n) - f_{k-1}(m+i, n+j)]^2 \quad (3.2)$$

(3) 归一化函数

$$NCCF(i, j) = \frac{\sum_{m=1}^M \sum_{n=1}^N [f_k(m, n) - f_{k-1}(m+i, n+j)]}{\sum_{m=1}^M \sum_{n=1}^N [f_k^2(m, n)]^{1/2} * \sum_{m=1}^M \sum_{n=1}^N [f_{k-1}^2(m, n)]^{1/2}} \quad (3.3)$$

运动估计中，匹配准则对匹配的精度影响不是很大，由于MAD准则不需要做乘法运算，实现简单、方便，所以使用最多，通常使用SAD代替MAD，SAD(Sum of Absolute Difference)即求和绝对误差，定义为：

$$SAD(i, j) = \sum_{m=1}^M \sum_{n=1}^N |f_k(m, n) - f_{k-1}(m+i, n+j)| \quad (3.4)$$

3.3 算法的研究

3.3.1 全搜法

全搜法(FS, Full Search Method)是在搜索窗的每个可能位置上计算最佳匹配。全搜法计算量很大，特别是对于大尺寸的搜索窗。可以按照光栅顺序(图 3.1)或者从中心位置开始螺旋顺序(图 3.2)处理各个位置。当使用提前结束匹配判决的算法时，螺旋搜索顺序有一些优势，因为最佳匹配很有可能发生在搜索区域中心的附近。

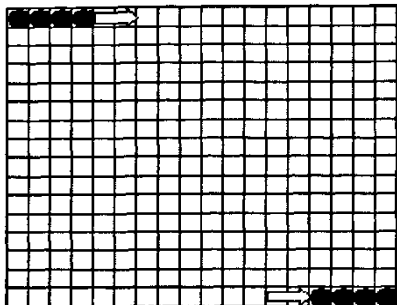


图 3.1 光栅顺序

Fig. 3.1 Grating Order

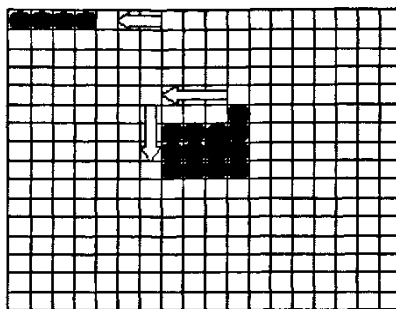


图 3.2 螺旋顺序

Fig. 3.2 Screw Order

全搜法是最简单的块匹配算法，由于可靠，且能够得到全局的最优的结果，通常是其他算法性能的比较的标准，但是它的计算量很大，这就限制了实时压缩编码的应用。

3.3.2 三步搜索法

三步搜索法 (TSS, Three Step Search)，是 T.KOLA 等人提出的，由于简单、健壮、性能良好的特点，为人们所重视。若最大搜索长度为 7，搜索精度取 1 个像素，则步长为 4, 2, 1 共需三步即可以满足精度，因此而得名三步法，如图 3.3。

TSS 算法的基本思想是采用一种由粗到细的搜索模式，从原点开始，按一定步长取周围 8 个点构成每次搜索的点群，然后进行匹配计算，跟踪最小块误差点。其算法如下：

- (1) 从原点开始，选取最大搜索长度的一半为步长，在周围距离步长的 8 个点处进行块匹配计算并比较；
- (2) 将步长减半，中心点移到上一步的最小点，重新在周围距离步长的 8 个点处进行块匹配计算并比较；
- (3) 在中心及周围 8 个点处找出 SAD 最小点，若步长为 1，该点所在位置即对应最佳运动矢量，算法结束；否则转到 (2)。

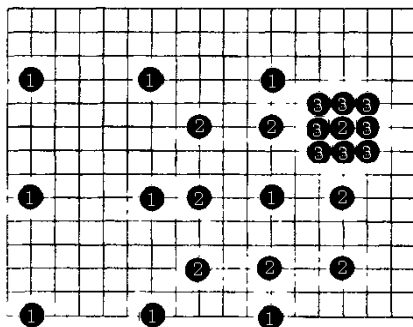


图 3.3 三步搜索法示意图

Fig. 3.3 Three Step Search Sketch Map

3.3.3 交叉搜索法

交叉搜索算法 (CSA, Cross Search Algorithm) 是 1990 年, Ghanbari 提出的它是在 TDL 和 TSS 基础上为进一步减少计算量而发展起来的快速搜索法^[33], 如图 3.4。

CSA 是从原点开始, 以“x”字形分布的五个点构成每次搜索的点群, 以 TDL 的搜索方法检测 SAD 点, 仅在最后一步采用“十”字形点群, 其算法如下:

(1) 从原点开始, 选取最大搜索长度的一半为步长, 在以“x”字形分布的五个点处进行块匹配计算并比较, 然后移动中心点;

(2) 以上一步的 SAD 点为中心, 步长减半, 继续做“x”字形的 5 点搜索。若步长大于 1, 则重复 (2); 步长为 1, 则进行 (3);

(3) 最后一步根据 SAD 点的位置, 分别做“十”字形和“x”字形搜索: 若上一步 SAD 点处于中心点、左上角或右上角, 做“十”字形搜索; 若上一步 SAD 点处于左上角或右上角, 做“x”字形搜索, 由当前 SAD 点得到最佳运动矢量, 算法结束。

CSA 的最大搜索点数为 $5 + 4\log_2 w$, 这里 w 表示最大偏移量。虽搜索速度很快, 但是运动补偿的效果不算太好, 在搜索区域的边界上有四分之一的点 CSA 没有考虑, 因此它不适用于较复杂的运动。

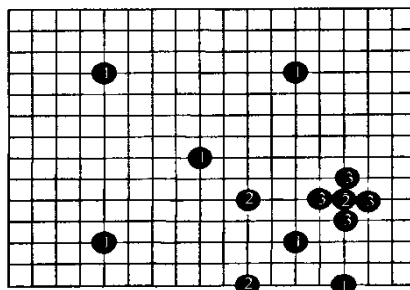


图 3.4 CSA 搜索示意图
Fig. 3.4 Cross Search Algorithm Sketch Map

3.3.4 对数法

对数法搜索由 J.R.Jain 和 A.K.Jain 提出的, 它开创了快速算法的先例, 分多个阶段搜索, 逐次减小搜索范围直到不能再小为止。其思想是从原始点开始, 以“十”字型分布的五个点构成每次搜索的点, 快速搜索 SAD 点, 如图 3.5。

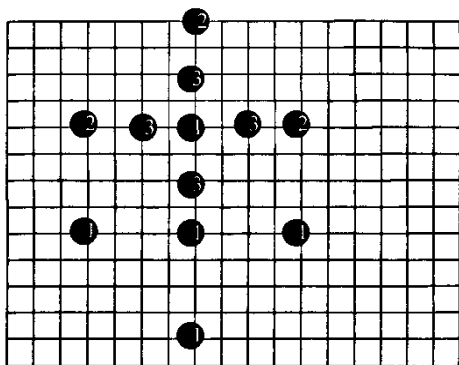


图 3.5 对数法搜索示意图

Fig. 3.5 Two-Dimensional Logarithm Search Sketch Map

算法步骤为：

- (1) 从原点开始，选取一定的步长，在以十字形分布的五个点进行快速匹配计算并比较；
- (2) 若 SAD 点在边缘四个点处，则以该点作为中心点，保持步长不变，重新搜索十字形分布的五个点；若 SAD 点位于中心点，则保持中心点位置不变，将步长减半，构成十字形点群，在五个点处计算；
- (3) 在中心及周围 8 个点处找出 SAD 点，若步长为 1，该点所在位置即为对应最佳运动矢量，算法结束；否则转到 (2)。

对数算法搜索时，最大搜索点数为 $2 + 7 \log_2 w$ ，这里 w 表示最大偏移量。若发现新的十字形点群的中心点位于搜索区的边缘，则步长也减半，后来有人提出应该在搜索的每个阶段搜将步长减半。所有这些改动都是为了使算法搜索范围很快变小，提高收敛速度。对数算法的前提是假设搜索区内只有一个谷点，如果搜索区内存在多个谷点时，该方法找到的可能时局部最小点。不能保证找到全局最优点也正是大部分快速搜索算法的通病。

3.3.5 梯度下降搜索法

基于块的梯度下降搜索法 (BBGDS, Block-Based Gradient Descent Search) 是 1996 年由 Lurng-Kuo Liu 和 Ephraim Feig 提出的。它是利用运动矢量中心分别特性，在搜索过程中使用 3×3 搜索窗。视频帧内相邻像素间具有渐变性，每一步的 SAD 点分布具有一定的方向性，也即梯度下降方向。BBGDS 算法使用梯度下降方向来决定下一步的搜索方向，如图 3.6。

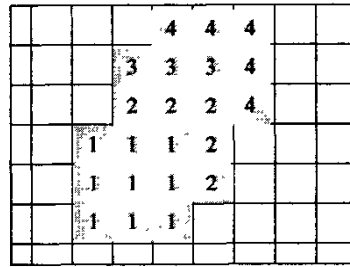


图 3.6 BBGDS 搜索示意图

Fig. 3.6 Block-Based Gradient Descent Search Sketch Map

算法描述为：

- (1) 使用 3×3 搜索窗对 9 个点进行搜索；
- (2) 若 SAD 点在搜索窗中心，则算法结束；否则以上一步 SAD 点为中心，重复第 (1) 步。

在每一步搜索过程中，BBGDS 算法使用了中心匹配块而不是匹配点，降低了陷入局部最优的可能性。引入梯度下降的方向来指导搜索方向，对该方向进行重点搜索，从而减少和避免了不必要的搜索，大大降低了算法的复杂度。

3.3.6 菱形搜索法

菱形搜索法(Diamond Search,DS)算法最早由 Shan Zhu 和 Kai-Kuang Ma 两人提出，后又经过多次改进，已成为目前快速块匹配算法中性能最优异的算法之一。1999 年 10 月，菱形法 (DS) 被 MPEG-4 国际标准采纳并收入验证模型 (VM)。如图 3.9，其基本思想是：

DS 算法采用了两种搜索模板，分别是有 9 个检测点的大模板 LDSP (Large Diamond Search Pattern) 和 5 个检测点的小模板 SDSP (Small Diamond Search Pattern)，如图 3.7 和图 3.8 所示。搜索时先用大模板计算当前最小块误差 SAD 点出现在中心点处时，将大模板 LDSP 换为 SDSP，再进行匹配计算，这时 5 个点中的 SAD 即为最优匹配点。

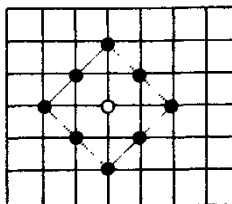


图 3.7 LDSP

Fig. 3.7 Large Diamond Search Pattern

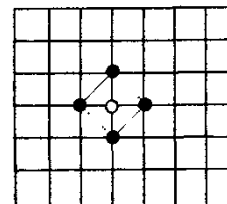


图 3.8 SDSP

Fig. 3.8 Small Diamond Search Pattern

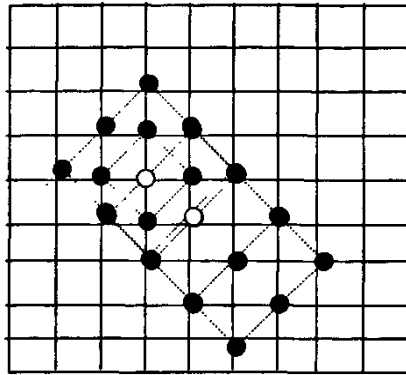


图 3.9 DS 算法搜索示意图
Fig. 3.9 Diamond Search Sketch Map

主要算法为：

- (1) 用 LDSP 在搜索区域中心及周围八个点处进行匹配计算，若 SAD 点位于中心点，则转到 (3)，否则转到 (2)；
- (2) 以上一次找到的 SAD 点作为中心点，用新的 LDSP 来计算，若 SAD 点位于中心点，在转到 (3)，否则转到 (2)；
- (3) 以上一次找到的 SAD 点作为中心点，将 LDSP 换为 SDSP，在 5 个点处计算，找出 SAD 点，该点所在位置即对应最佳运动矢量。

DS 算法的特点在于它分析了视频图像中运动矢量的基本规律，选用了大小两种形状搜索模板 LDSP 和 SDSP。先用 LDSP 搜索，由于步长大，搜索范围广，可以进行粗定位，使搜索过程不会陷入局部最小；当粗定位结束后，可以认为最优点就在 LDSP 周围 8 个点所围的菱形区域中，这时再用 SDSP 来准确定位，使搜索不至于有大的起伏，所以它的性能优于其他算法。另外，DS 搜索时各步骤之间有很强的相关性，模板移动时只需在几个新的检测点处进行匹配计算，所以也提高了搜索速度。

3.4 算法的改进

通常对基于块的运动估计快速搜索算法可以归纳为三类：(1) 通过减少搜索点的快速算法；(2) 快速块匹配误差计算方法；(3) 运动场下采样法。其中第一类方法是最高效的也是最常用的快速搜索算法。目前已有很多的搜索算法，其中全搜索算法 FS (Full Search) 的搜索精度是最高的。但全搜索算法是通过搜索窗内的所有点进行搜索，对每一个块需要计算匹配准则 $(2dx+1)^2$ 次，其中， dx 为运动矢量的最大偏移量。因此全搜索算法计算量巨大，很不利于实时应用的场合。针对该问题又提出了很多的搜索算

法：对数搜索法、三步搜索法、交叉法、梯度下降搜索法、菱形搜索法等。

通过对上述各种快速搜索算法进行深入分析，可得到如下结论：算法中采用逐步缩小搜索步长来缩小块匹配的最佳位置，或是采用大小两个不同的搜索范围，先经过大的搜索范围然后在转向小的搜索范围。而实际中很多的视频序列如视频会议、可视电话等，大部分的背景并没有发生移动。因此，这种算法对那些运动矢量近乎为零的块的运动估计搜索造成了很大的计算冗余，降低了搜索速度。而且对于上面的搜索方法都是当前偏离整数个像素的位置，实际上，对于很多块，如果在一个内插到半像素精度的区域进行搜索可以找到更好的匹配。并且这些算法还很容易陷入局部最优，找到的是匹配极小值，而非匹配最小值，有时性能与 FS 相比下降很多，缺少灵活性和自适应性。为了解决上述问题，本节利用时空域运动相关性，采用半像素估计和整数像素估计结合，为了避免陷入局部匹配，采用一种自适应的搜索终止规则，提出了一种新的搜索算法——逐级搜索算法。

3.4.1 相关性分析

在视频序列中，相邻的帧间通常都具有较高的相关性，在帧率较高时这种相关性更加明显。视频对象的运动具有连续的运动特征，因此描述视频序列运动特征的宏块运动矢量间也存在时间和空间上的相关性。尤其是在相邻的块间，运动矢量通常都具有较高的相关性，图 3.10 为运动矢量的时空域相关特性。

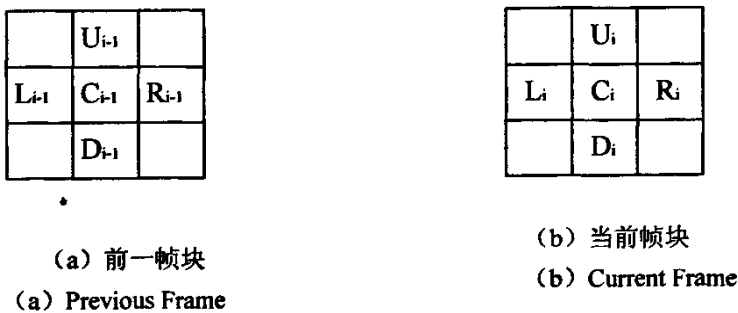


图 3.10 空域相关性

Fig. 3.10 Air Zone Relevant

设 C_i 为当前要编码的块， L_i 、 U_i 、 R_i 、 D_i 分为其左侧、上侧、右侧和下侧的相邻块， L_{i-1} 、 U_{i-1} 、 R_{i-1} 、 D_{i-1} 分别为前一帧中相对应的块，设 L_i 、 U_i 、 R_{i-1} 、 D_{i-1} 的运动矢量分别为 $V_1 \sim V_5$ ，如果各运动矢量相对于 V_3 的运动矢量差值的绝对值小于一个给定的

阈值, 则认为与 C_i 块相邻的各块间运动的相关性较高, 表明该区域的变化比较平缓。如果大于阈值, 则认为与 C_i 块相邻的各块间运动的相关性较低, 表明该区域变化比较剧烈。

3.4.2 半像素估计

半像素估计就是在原来整数像素的基础上内插进行估计, 如图 3.11 所示, 原来的整数的像素位置在块中用 a 标记, 像素 b 和 c 通过对一对整数像素进行线性内插得到的, 像素 d 是通过对周围的四个整数像素进行内插得到, 箭头表示内插方向, 增加内插的密度可以提高更好的匹配, 但计算量的代价也相应的增加, 所以本文采用一次线性插值。

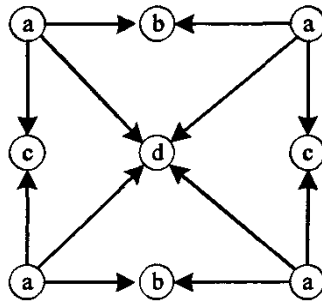


图 3.11 半像素插值
Fig. 3.11 Half Pixel Interpolation

3.4.3 终止规则

对于相邻的图像帧之间有很大的相关性, 所以存在很多运动估计向量为零或很小的搜索估计, 这样当 SAD 值满足一定的范围时, 即 $SAD < T$, 就可认为找到最佳的宏块匹配, 即可以得到相应块的估计值, 但这并不能适应于纹理变化较大的帧图像。

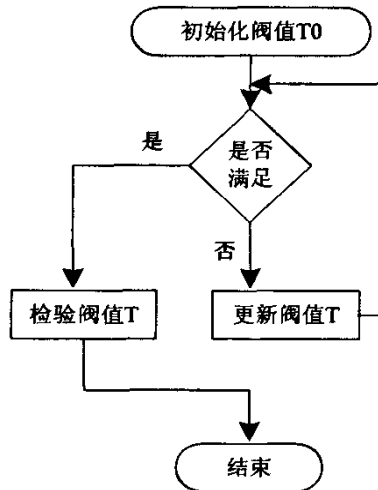


图 3.12 自适应阈值更新
Fig. 3.12 Auto-adapted Threshold Update

所以，利用自我学习特点的阈值可以更好的解决局部最小的矛盾。阈值的改变规则流程如图 3.12 所示：

3.4.4 搜索策略

在半像素的网格进行搜索显然比先前提到的整数像素搜索需要更多的计算量。为了限制计算量的增长，本文采用的方法是先在整数位置找到最佳匹配，然后在这个位置周围的半像素位置进行搜索，虽然计算量有所增长，但是半像素运动估计的性能明显好于整数像素的运动估计，因为相邻图像帧的运动物体不一定运动到整数像素的位置。所以通过两者的结合运动估计搜索，在大多数情况下能找到更好的匹配。因此对于整数像素搜索，本文提出了逐级搜索方式：即首先对一副图像块下采样本进行搜索，接着进行更高分辨率的图像搜索，直到到达原始块的大小。在最高等级使用全搜索，对于下一级，只在上一级粗糙矢量周围的 1 个像素位置进行搜索。图 3.13 显示这个算法，包含 3 个等级，最高等级的搜索窗是 3 个像素。在最高等级进行全搜时计算复杂度很低，因为在最高等级 3 每个搜索位置只需要搜索 4×4 的像素区域。最佳匹配位置 3 作为等级 2 的搜索中心，依次类推。同时，若计算得到的某个 SAD 值小于当前的阈值则直接转到半像素搜索，否则对于在最低一级得到的最小 SAD 中，再进行一次半像素搜索，即可得到半像素级的运动估计搜速。 这样来避免陷入局部最优的局面。

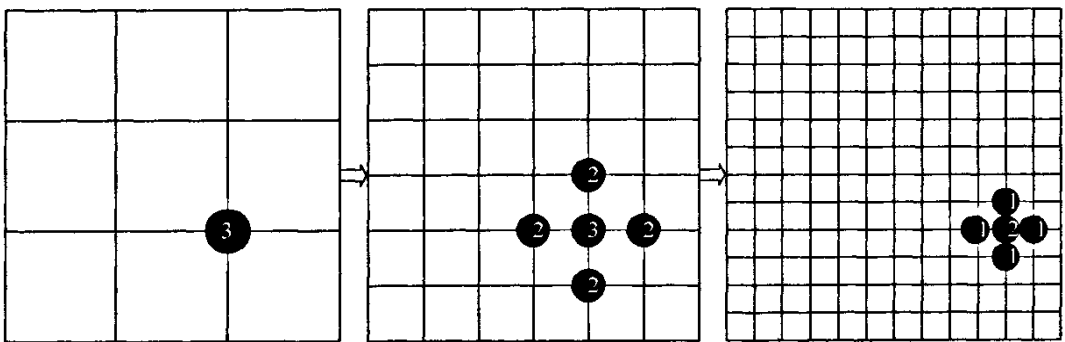


图 3.13 逐级搜索法

Fig. 3.13 Progressive Search Method

3.4.5 搜索步骤

- (1) 等级 1 包含原始尺寸的当前宏块和参考宏块，对等级 1 的水平和垂直方向进行因子为 2 的下采样得到等级 2；
- (2) 重复步骤 (1)，对等级 2 进行下采样得到等级 2，继续直达到指定的等级数；

- (3) 索最高等级的最佳匹配, 若得到的 SAD 小于当前的阈值, 则直接转到步骤(6), 否则转到步骤(4);
- (4) 在上一级粗糙的矢量附近对下一级进行搜索, 找到最佳匹配, 若得到的 SAD 小于当前阈值, 则直接转到步骤(6), 否则转到步骤(5);
- (5) 重复步骤(4) 直到找到等级 1 的最佳匹配, 然后转到步骤(6);
- (6) 在搜索到的最佳匹配附近进行一级半像素搜索; 若在半像素中找到最佳的 SAD, 则该半像素点即是最佳匹配点, 否则整数像素点为最佳匹配点。

3.4.6 算法分析

通过本算法的搜索步骤可以看出, 与其它算法相比,, 采用自适应的调整搜索阈值, 以便能较快的跳出局部最优, 以致达到全局最优。且在性能和复杂度之间作出了很好的折中, 同时又利用半像素搜索方式, 从而增加的宏块匹配的精度, 提高了图像质量。

3.4.7 边缘处理

对于运动发生宏块的边缘, 则边缘宏块的最佳匹配可能发生在参考图像帧的边缘之外。如图 3.14 显示, 当前帧中全部出现的球在参考帧中部分可见, 所以最佳的匹配块的一部分可能在参考帧的边缘之外。本文通过在参考帧的边缘内插像素, 把边缘的像素简单的外推到图像的边缘之外, 这样对于边缘运动视频序列, 块匹配效率就会得到提高。

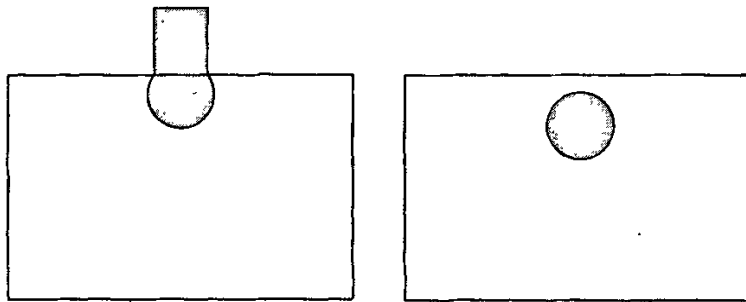
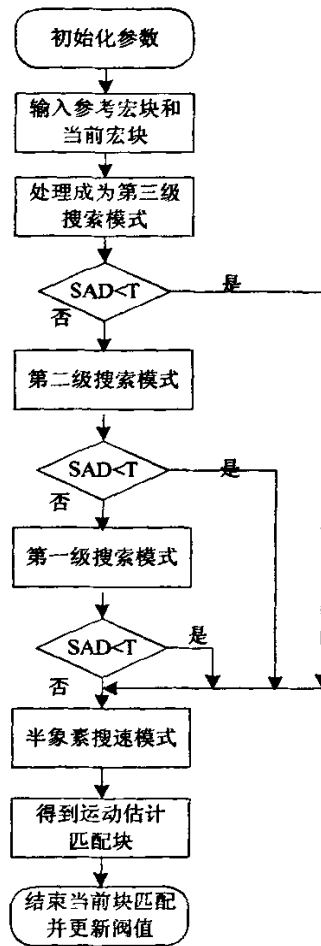


图 3.14 边缘处运动估计处理
Fig. 3.14 Process Motion Estimate Of The Edge

3.4.8 算法流程

通过重复使用已经计算过的值, 可减少运动估计过程中的计算时间, 本文通过当计算的 SAD 小于给定的阈值时, 认为已经达到了最优匹配, 这样可以提前结束运动搜索。如图 3.15 为算法流程图。对于计算 SAD 中采用单循环方式, 函数 3.1 主要是计算块的

SAD 的主要过程:



3.15 改进算法流程图

Fig. 3.15 Improvement Algorithm Flow Map

函数 3.1 VCMotionEstimate (int *CurData, int *PreData)

//i,j 为当前位置, ioffset,joffset 为参考帧块位置, N 为块大小

```

VCProgress
SumSad=0;
For j=0 to N
{
    SumSad=SumSad+abs (Col[0,j]-Row[0+ioffset,j+joffset] );
    If (SumSad<T) beark;
    .....
    SumSad=SumSad+abs (Col[15,j]-Row[15+ioffset,j+joffset] );
    If (SumSad<T) beark;
}
VCUpdate (T) ;//更新阈值
    
```

3.4.9 性能分析

在第四章, 4.2.1 节算法比较中, 对本算法的和目前较常用的搜索算法做了比较主要在搜索点数和帧 PSNR 值。

3.5 压缩编码

3.5.1 嵌入式小波编码

小波变换编码是近年来随着小波变换理论的研究而提出的一种具有很好发展前景的编码方法, 作为一种多分辨率的分析方法, 小波变换具有很好的时-频或空-频局部特性, 特别适合按照人类视觉系统特性设计图像压缩编码方案, 非常有利于图像的分层传输, 实践证明: 基于小波变换的图像编码, 在压缩比和编码质量方面有优于传统的 DCT 变换编码的巨大潜力。

随着研究的深入, 出现了许多算法, 其中以嵌入式零树小波编码算法 (EZW)、分层小波树集合分割算法 (SPIHT)、集合分裂嵌入编码 (SPECK) 和比特流优化截取的内嵌块编码算法 (EBCOT) 四种最为经典。下面将分别介绍这四种算法。

(1) 嵌入式零树小波编码算法 (EZW)

Lewis 和 Nowles 等首先提出了零树结构^[15]并且第一个实现了零树编码思想, 但是 Lewis 等人提出的算法并不完美。1993 年, Shapiro 将这种数据结构与比特平面编码技术结合起来, 提出了嵌入式零树小波 (EZW: Embedded Zero-tree Wavelet) 编码算法^[16]。EZW 算法中采用的零树结构充分利用了不同尺度间小波系数的相似特性, 有效的剔除了对高频小波系数的编码, 极大的提高小波系数的编码效率。EZW 算法以极低的复杂度获得了高效的压缩性能, 产生的码流还具有嵌入的特性, 支持渐进传输, 因此 EZW 算法在小波图像编码史上具有里程碑的意义。零树小波编码基于三个关键的思想: 用小波变换去相关; 利用小波变换的内在自相似性在各级之间预测重要的信息位置; 用自适应算术编码进行无损压缩。

(a) 零树结构 (零树预测)

一幅经过小波变换的图像按其频带从低到高形成一个树状结构, 树根是最低频子带的结点, 它有三个孩子分别位于三个次低频子带的相应位置, 如图 3.16, 其余子带 (最高频子带除外) 的结点都有四个孩子位于高一级子带的相应位置, 由于高频子带分辨率增加, 所以一个低频子带结点对应有四个高频子带结点, 即相邻的 2×2 矩阵, 见图 3.16 所示的三级小波分解就形成了深度为 4 的树。

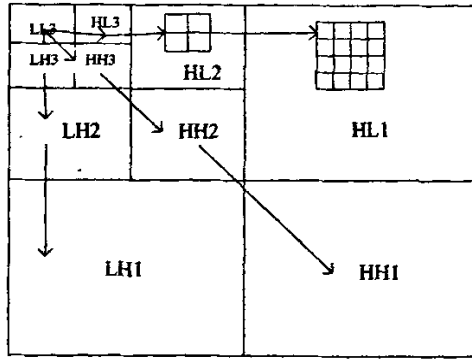


图 3.16 ZEW 的零树结构

Fig. 3.16 ZEW Zero-tree Constructure

定义一个零树的数据结构：一个小波系数 x ，对于一个给定的阈值 T ，如果 $|x| < T$ ，则称小波系数 x 是不重要的。如果一个小波系数在一个粗的尺度上关于给定的阈值 T 是不重要的，之后在较细的尺度上在同样的空间位置中的所有小波系数也关于阈值 T 是不重要的，则称这些小波系数形成了一个零树。这时，在粗尺度上的那个小波系数称为母体，它是树根，在较细尺度上相应位置上的小波系数称为孩子。正是通过这种零树结构，使描述重要系数 ($|x| < T$) 的位置信息大为减少。

(b) 用零树结构编码重要图

重要图包括三种要素：即重要系数、孤立零和零树根。其中，对于一个给定的阈值 T ，如果系数 x 本身和它的所有的子孙都小于 T ，则该点就称为零树根；如果系数本身小于 T ，但其子孙至少有一个大于或等于 T ，则该点就称为孤立零点。在编码时分别用三种符号与之对应。当编码到最高分辨率层的系数时，由于它们没有子孙，零树根不再存在，只需其余两种符号即可。为了有利于内嵌编码，将重要系数的符号与重要图一起编码，这样就要使用四种符号：零树根、孤立零、正重要系数、负重要系数。

(c) 基于SAQ的嵌入式编码

SAQ (Successive Approximation Quantization) 逐次逼近量化，内嵌编码的核心在于采用了 (SAQ)，SAQ 按顺序使用了一系列阈值 T_0, T_1, \dots, T_{N-1} 来判决重要性，其中 $T_i = T_{i-1} / 2$ ，初始阈值 T_0 按如下条件选择： $|x_i| < 2T_0$ ，其中 x_i 表示所有变换系数。

在编码过程中，始终保持着两个分离的列表：主表和辅表。主表对应于编码中的不重要的集合或系数，其输出信息起到了恢复各重要值的空间位置结构的作用，而辅表是编码的有效信息，输出为各重要系数的二进制值。编码分为主、辅两个过程：在主过程中，设定阈值 T_i ，按上述原理对主表进行扫描编码，若是重要系数，则将其幅值加入

辅表中,然后将该系数在数组中置为零,这样当阈值减小时,该系数不会影响新零树的出现:在辅过程中,对辅表中的重要系数进行细化,细化过程类似于比特平面编码。对阈值 T_i 来说,重要系数的所在区间为 $[T_i, 2T_i]$,若辅表中的重要系数位于 $[T_i, 3T_i/2]$,则用符号“0”表示,否则用符号“1”表示。编码在两个过程中交替进行,在每个主过程前将阈值减半。

(2) 分层小波树集合分割法 (SPIHT)

在 EZW 算法中首次提出了零树的概念,零树是一种有效的表示不重要系数的数据结构,但是在小波变换系数中还存在这样的树结构,它的树根是重要的,除此之外其它的结点是不重要的。对于这样的树结构零树就不是一种有效的数据结构。针对这一现象, Said 和 Pearlman 于 1996 年提出了 SPIHT 算法。

SPIHT 算法继承了 EZW 算法的小波系数的零树结构^[17]。但是它采用空间方向树,结点 (i,j) 的所有子孙集合 $D(i,j)$ 和间接子孙集合 $L(i,j)$ 来表示小波变换系数。如图 3.17 所示,空间方向树的每一个结点对应一个像素,这个结点的直接后代是位于下一个高频子带上和它具有相同空间位置的像素。对于每一个结点,要么没有直接后代,要么有四个直接后代,这四个直接后代是相邻的,组成一个 2×2 的像素块。图 3.17 中的箭头是从父结点指向它的四个直接后代。位于金字塔结构最高层的像素,也就是最低频子带的像素,是空间方向树的树根,它们也是由相邻的像素组成 2×2 的像素块,可以看到,位于像素块左上角的像素没有直接后代。当结点 (i,j) 的后代 $D(i,j)$ 中存在显著系数时,即 $D(i,j)$ 中至少有一个系数的绝对值大于给定的阈值 T ,则将其分裂为结点 (i,j) 的直接子孙集合 $O(i,j)$ 和间接子孙集合 $L(i,j)$,其中 $L(i,j) = D(i,j) - O(i,j)$ 。如果 $L(i,j)$ 中依然存在显著系数,则将 (i,j) 的直接子孙加入到结点集合中,从而实现数据集合的排序和更新。

在实现 SPIHT 算法的编码时,使用了三个链表:不重要系数链表 (LIP),重要系数链表 (LSP) 和不重要集合链表 (LIS)。LSP 初始化为空表, LIP 用最低频子带系数初始化, LIS 用每一个空间方向树的根结点来初始化。对于一个具有三级结构的空间方向树 $T(i,j)$ 来说,在初始化时它分裂为树头结点 (i,j) 和剩余集合 $D(i,j)$,对 $c(i,j)$ 判断其重要性,若重要则转到 LSP 中。对 $D(i,j)$ 进行重要性测试,若 $D(i,j)$ 是不重要的,则它用一个符号就可表示;否则若 $D(i,j)$ 是重要的,则将 $D(i,j)$ 分裂为两个集合 $O(i,j)$ 和 $L(i,j)$ 。对 $O(i,j)$ 中的每个元素进行重要性测试,将重要元素转到 LSP 中。对 $L(i,j)$ 进行重要性测试,若 $L(i,j)$ 不重要,则用一个符号就可以表示该集合;

若 $L(ij)$ 重要, 则 $L(ij)$ 分裂成四部分, 每部分由相应空间方向树上的位置上的元素构成, 每部分与 $O(ij)$ 中的四个元素分别构成四棵新树, 由于每棵新树的头结点的重要性已经判断过了, 所以只需对新树的剩余部分也就是 $L(ij)$ 分裂出的四个集合进行判断。如此重复对每一棵树进行分裂和判断, 从而找到所有的重要元素, 并把它们转到 LSP 中。需要注意的是对每棵树的分裂不是一次进行到底的, 而是按照一定的扫描顺序进行的。对由最低频子带和头结点构成的 LIP 中的元素是按从上到下, 从左到右的顺序进行扫描的, 其它子带则是以 2×2 的块为单位从上到下, 从左到右依次扫描, 对每个 2×2 块中的元素还是按照从上到下, 从左到右的顺序扫描。

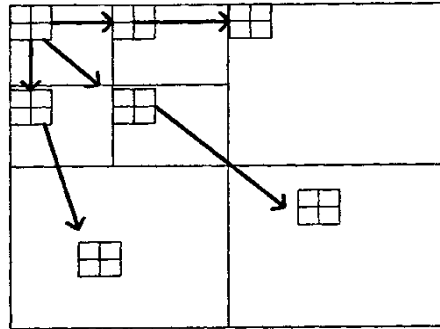


图 3.17 SPIHT算法空间方向树结构

Fig. 3.17 SPIHT Algorithm Space Direction Tree Structure

SPIHT 算法的伪代码为:

- (a) 求初始阈值 $T_0 = \lfloor \log_2(\max |c(i, j)|) \rfloor + 1$, 并输出到码流中。将坐标 $(i, j) \in H$ 加入 LIP, $(i, j) \in H$ 中有子孙的树加入 LIS, LSP 为空集。
- (b) 排序过程: 检测 LIP 中所有的系数的重要性。
 - ① 如果重要, 则输出1符号位, 并将系数移到LSP;
 - ② 如果不重要, 则输出0。
- (c) 根据树的类型检查 LIS 中所有的树的重要性。

对于D (ij) 类树:

- ① 如果重要, 则输出1, 并编码它的子节点;
 - 如果子节点重要, 则输出1, 后跟1位符号, 并将其加入到LSP中;
 - 如果不重要, 则输出0, 并将其加入到LIP的末尾;
 - 如果子节点有子孙, 则将树作为L (ij) 类移到LIS的最后, 否则从LIS中移去。

② 如果不重要，则输出 0。

对于 $L(i,j)$ 类树

① 如果重要，则输出 1，并将每个子节点作为 $D(i,j)$ 类的一项加到 LIS 末尾，并从 LIS 中移去父树；

② 如果不重要，则输出 0。

(d) 量化：阈值减 1，如果阈值为 0，则结束；否则转到第二步。

(3) 集合分裂嵌入编码 (SPECK)

SPECK (Set partitioned embedded block) 是 Islam 和 Pearlman 提出的另一个嵌入式小波编码方法^[19]。该方法充分利用小波系数的能量集中和能量尺度的增加而衰减的特点，将二叉树分裂和比特平面编码方法相结合，获得了与 SPIHT 相近的压缩性能。

基本思想：初始化时集合 X 有所有的小波子带构成，集合 S 由最低频子带初始化，而剩余集合 $I=X-S$ 如图 3.18 (a) 所示。如果在当前阈值下集合 S 是重要的，则采用二叉树分裂的方法搜索重要系数，如图 3.18 (b) 所示。如果集合 I 是重要的，则采用倍频带分裂的方法对集合 I 进行处理：集合 I 依次分解成三个 S 子带和剩余集合 I ，如图 3.18 (c) 所示。然后对这三个 S 子带和集合 I 分别进行处理。如此重复，直到把所有的子带分裂出来，剩余集合 I 变为空集。

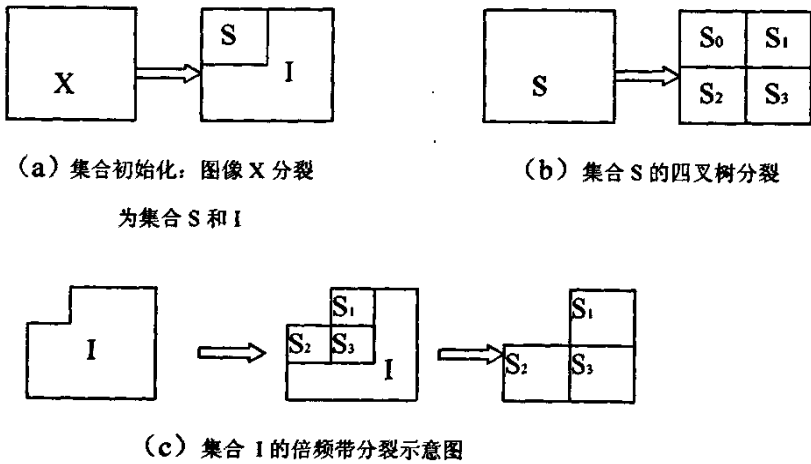


图 3.18 SPECK 结构图

Fig. 3.18. SPECK Structure

根据 SPIHT 的思想，对于一个集合 S 来说，如果 $\max_{(i,j) \in S} \{ |c_{i,j}| \} \geq 2^n$ ，则这个集合 S

是重要的，否则它就是不重要的。这样可以定义一个集合 T 的重要性：

$$\Gamma_n(T) = \begin{cases} 1, & 2^n \leq \max_{(i,j) \in T} \{ |c_{i,j}| \} < 2^{n+1} \\ 0, & \text{其它} \end{cases} \quad (3.5)$$

为了对一个 S 集合进行编码，SPECK 算法采用了与 SPIHT 相似的方法。二者的区别在于排序过程。在排序过程中，SPECK 不是采用空间树来判断重要性，而是采用 S 集合的方式来有效组织小波系数。这样做的目的是充分利用小波子带能量集中的特性，可以保证具有高信息量的小波系数能首先被编码。

与 SPIHT 算法相似，SPECK 采用了两个列表来组织小波系数：LIS 不重要集合的列表，LSP 重要像素的列表。前一个列表包括尺度可变的类型为 S 的集合，这些集合对于当前阈值来说是不重要的。后一个列表包括一些像素，这些像素在当前阈值下被检测为是重要的，LIS 列表中包括多个具有不同尺寸大小的 S 集合。为了加速编解码的处理速度，SPECK 算法将具有相同尺寸大小的集合 S 集中起来形成多个小的 LIS 列表，这些小的列表按照从小到大的顺序进行排序。

(4) 比特流优化截取的内嵌块编码算法 (EBCOT)

对于 EZW, SPIHT 及 SPECK 等小波编码器来说，虽然码流具有嵌入的特性，可以支持渐进传输，但是只能支持 SNR 可扩展，无法支持分辨率可扩展。同时，这种嵌入式的码流对传输误码非常敏感，抗误码的能力很差，一旦传输出现误码，则此误码之后的码流会失去同步，因此 Taubaman 等在早期的 LZC (Yayered Zero Coding) 算法的基础上，提出了 EBCOT (Embedded Block Coding with Optimized Truncation) 编码^[20]，这是一种内存受限的比特平面编码方法，编码时先将每个子带分成一个个相对独立的，固定尺寸的码块，然后对这些码块进行比特平面编码得到嵌入式的码流。每个比特平面进行编码时又分成四个子平面通道，每个子平面的编码结束点作为率失真截断点。然后使用优化的截断算法对码流进行截断，产生压缩码流。EBCOT 不仅具有 SNR 可扩展能力，而且具有分辨率可扩展能力，还可以支持图像的随机存储。从而称为新一代静止图像编码标准 JPEG2000 的核心算法。

3.5.2 非嵌入式小波编码

嵌入式编码支持渐进式的传输，可以在任意点停止编码，严格满足目标码率或目标失真度。与嵌入式编码相比，非嵌入式小波编码方法往往具有内存需求小，抗误码能力强等优点。下面主要介绍 SFQ 和 FFSQ。

(1) SFQ

SFQ (Space-Frequency Quantization)^[34]是 Xiong 等提出的一种非嵌入式的小波编码方法。该方法将均匀标量量化和零树量化结合在一起,采用一种联合最优方法对图像进行编码。在一定码率限制下,通过最优选择零树结构和均匀标量量化步长,产生最小的图像编码失真。

在 SFQ 算法中,剩余树是指任何父节点的所有后代组成的集合,不包括父节点。零量化是指对剩余树中所有节点的值是否设为 0。一个零树节点是指对这个节点的所有后代设为零值。当对一个剩余树进行零树量化时,只需要一个符号去表示整个量化为 0 的树集合。如果某个剩余树的节点很少的能量,那么整个剩余树就应该设为 0。

SQF 算法采用率失真准则来解决这个问题。最优的零树结构取决于标量量化步长的选择,因为剩余树设为零树的操作涉及到标量量化步长所引起的率失真平衡问题。相反,标量量化步长的选取也受到零树量化的约束。因此,选择最佳的零树量化和最佳的标量量化是相互影响的,需要一种迭代的方法来解决这问题。

(2) FSSQ 算法

FSSQ (Finite State Scalar Quantization) 有限状态标量量化,是 Naveen 和 Woods 于 1993 年最早提出的^[35]。这一方法主要利用了小波图像在同一个方向上不同分辨率各子带之间的相似性,对三个方向的塔式数据分别进行处理。

编码过程为:

(a) 对最低频子带 LL3 采用差分脉冲编码调制 (DPCM) 进行编码。

(b) 对 LH3 子带进行如下操作:

- ① 将 LL3 子带重建图像分割成尺寸为 $m \times n$ 的小块,分别计算这些小波的方差值,并根据此方差值的不同将 LL3 子带重建图像内的像素分成 S 类。

$$\text{var} = \frac{1}{mn} \sum_{i=1}^{mn} x_i^2 - \left(\frac{1}{mn} \sum_{i=1}^{mn} x_i \right)^2 \tag{3.6}$$

根据 LL3 子带与 LH3 子带的父子对应关系,可以决定 LH3 子带内像素的分类情况。

- ② 为各类像素确定不同的量化步长,分别用不同的一致阈值量化器 (UTQ:Uniform Threshold Quantizer) 进行不同状态的标量量化,即为有限状态量化。
- ③ 对量化结果进行熵编码。

(c) 对 LH2 和 LH1 子带的操作过程类似，区别只是在决定其子带内像素的分类时，要分别利用他们的父子带 LH3 和 LH2 的重建图像。

性能比较：

下面对二副 (Lena,Barbara) 具有不同纹理的标准测试图像进行比较，表 3.1 和表 3.2 所示为比较的结果。

表 3.1 Lena 图像 (512×512) PSNR 比较结果 (dB)
Table 3.1 Lena Image (512×512) PSNR Comparison Result (dB)

	0.25bpp	0.5bpp	1.0bpp
EZW	33.17	36.28	39.55
SPIHT	34.17	37.21	40.54
SPECK	34.03	37.10	40.25
EBCOT	34.16	37.29	40.48
SFQ	34.33	37.36	40.52

表 3.2 Barbara 图像 (512×512) PSNR 比较结果 (dB)
Table 3.2 Barbara Image (512×512) PSNR Comparison Result (dB)

	0.25bpp	0.5bpp	1.0bpp
EZW	26.77	30.53	35.14
SPIHT	27.78	31.50	36.51
SPECK	27.76	31.54	36.49
EBCOT	28.40	32.29	37.11
SFQ	28.29	32.15	37.03

3.5.3 编码实现

本节主要对 SPIHT 算法进行编码，其主要包括初始化集合，排序过程和量化过程三个部分，其中 PutBit () 为输出字节函数：

函数 3.1 VCSPiHTCoding ()

说明：编码主函数

过程：

```
//对其中的一些参数设置略
Initialization ();
//初始化列表
```

```

while (threshold!=0)
{
    pass_count++;
    SortingPass1 (M,threshold) ;//排序过程
    RefinementPass1 (M,threshold) ;//量化过程
    threshold >>= 1;
    if (stop==1)
        break;
}
Destroy () ;//释放内存空间

```

函数 3.1 主要对整个 SPIHT 编码进行内存分配，参数设置，错误检查，及编码后的内存释放等处理，是整个编码算法的入口。

函数 3.2 VCInitialization ()

说明：集合的初始化

过程：

```

// 创建并初始化 LSP 列表
LSP->Create ();

//创建并初始化 LIP 列表
LIP->Create ();

//创建并初始化 LIS 列表
LIS = new CLinkList;
LIS->Create ();

```

函数 3.2 主要是对三个列表进行创建内存并初始化相应的参数。并分别调用相应类的 create () 函数。完成整个编码过程中的初始操作，对于每个列表采用链表数据结构。

函数 3.3 VCSorting (CMatrix2D *m, int threshold)

说明：排序过程

过程：

```

// 处理 LIP 中的元素
LIP->Reset ();
while (LIP->current!=NULL)
{
    d = LIP->GetCurrentElement (&found) ;
    if (found==0) {
        MessageBox (NULL,"Can't retrieve current list element..",
            "Warning!",MB_OK) ;
        return;
    }
    LIPProcess (LIP) ;
}
LIP->Reset ();

```

```

// 处理 LIS 中的元素
LIS->Reset ();
while (LIS->current!=NULL)
{
    d = LIS->GetCurrentElement (&found);
    if (found==0) {
        MessageBox (NULL,"Can't retrieve current list element.",
            NULL,MB_OK);
        return;
    }
    LIS Process (LIS);
}
LIS->Reset ();

```

函数 3.3 是对列表中的元素进行排序。首先在利用链表之前，先清空原来有的的数据内容利用 Reset () 成员函数，遍历整个表，用 GetCurrentElement () 函数找到元素，并存到指针变量中，若找到则进行该类型元素的处理；否则，转到出错处理环节中。

函数 3.4 VCRefinementPass1 (CMatrix2D *m, int threshold)

说明：量化过程

过程：

```

while (p!=pNewlyAppended->next) //循环
{
    temp = threshold & abs (m->m[p->data.y][p->data.x]);
    if (temp==0)
        PutBit (0);
    else
        PutBit (1); //输出编码后的字节
    if (stop==1)
        return;
    p = p->next;
}

```

函数 3.4 是对上一步得到的数据进行量化，这样就可以得到更多的接近零的数据，达到更高的压缩效果。

第四章 算法的实现与分析

本论文在以 VC++6.0 为开发平台，在 PC 机上对论文提出的视频编码算法进行了软件仿真实验，对其中的基于自适应阈值的逐级搜索的运动估计算法、提升小波变换算法和整个方案的性能进行了分析，并在关键算法中进行了比较。机器的运行环境是：512M 内存，Pentium (R) 4 CPU 2.60GHz, Windows XP 操作系统 SP2；

实验采用的两个图像序列，一个是：一段用摄像头录制未经过压缩处理的视频帧 Test，视频文件大小为 34647k，帧格式为 176×144，背景是相对静止的，序列中的运动信息较少；一个是 4:2:0 格式的标准 QCIF Foreman 序列，Foreman 序列纹理复杂度一般，两个序列的帧率均是 30 帧/秒。

4.1 算法实现

4.1.1 框架流程图

本论文的视频编码结构和标准的结构相似，只是在具体关键算法中利用本论文提出的算法；其整个编码框架采用运动估计/补偿+分块 ILWT 变换的混合编码框架。如图 4.1 所示；对于其中每一帧图像的编码流程为图 4.2 所示：

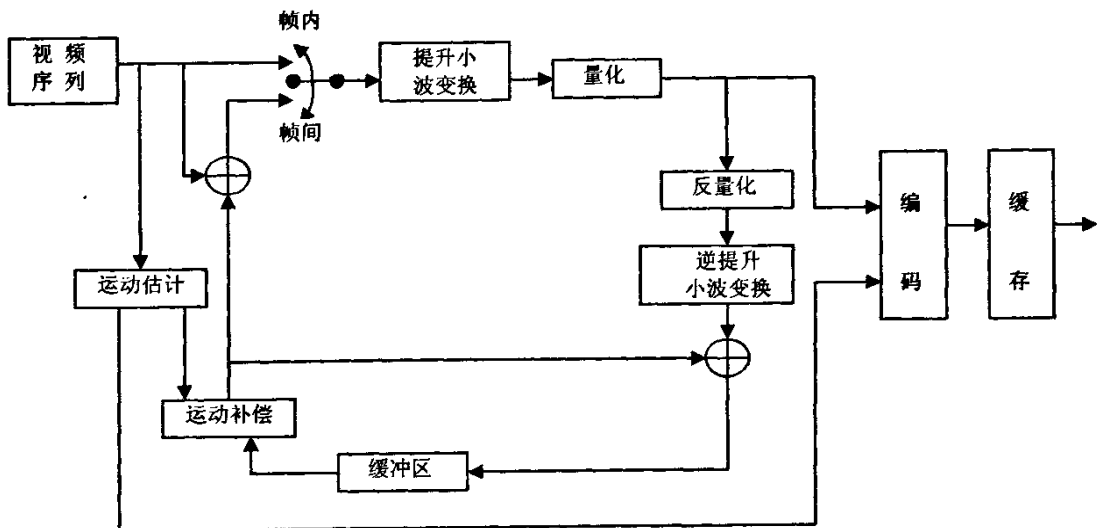


图 4.1 视频编码框架

Fig. 4.1 Video coding framework

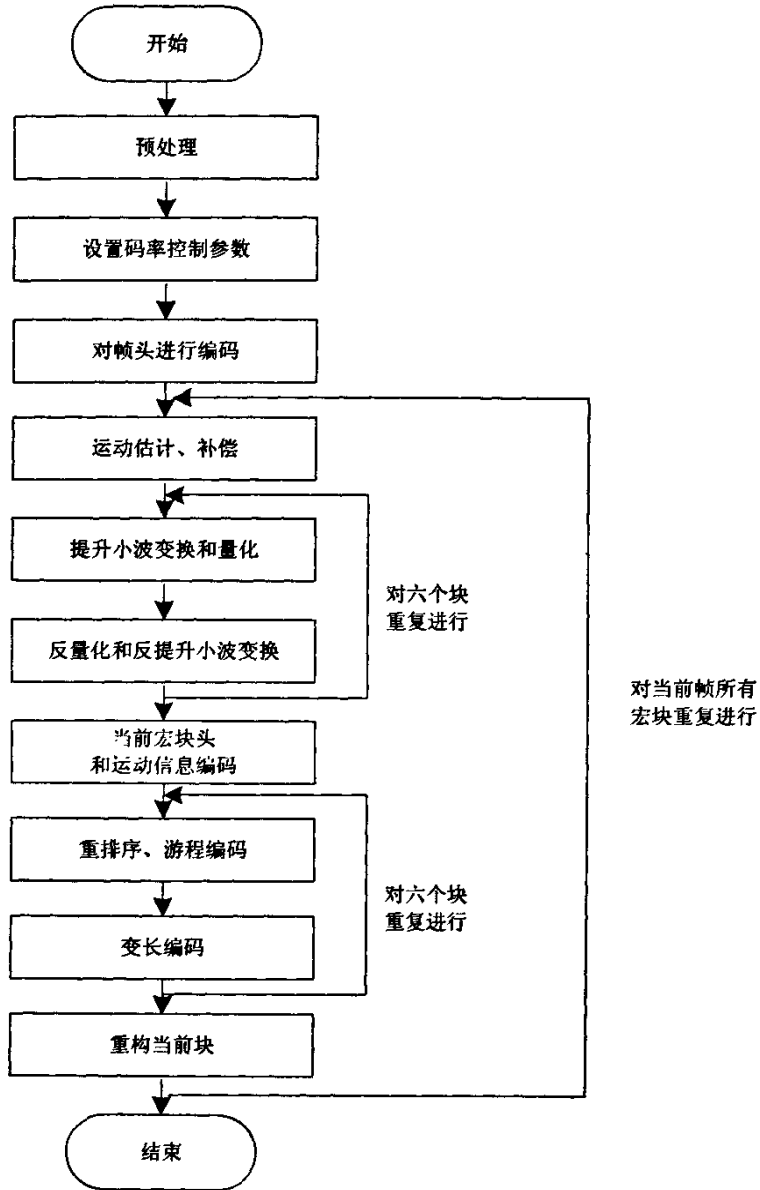


图 4.2 帧图像编码流程图

Fig. 4.2 Frame coding flow

4.1.2 参数设置

编码参数设置通过对特定的配置文件（如 encoder.cfg）进行操作完成。为方便对参数的存取，采用结构化的存储方式。即参数名称、类型、数据值。为了便于参数操作，定义结构体 Mapping:

```

Typedef struct{
    char *TokenName;

```



```

void *Place;

int  Type;

}Mapping;
    
```

创建映射表，将参数名称、数值同具体参数变量相对应。

读取配置文件中的参数主要由 Configure () 函数完成，Configure () 首先通过 GetConfigFileContent () 函数在内存中为各个参数分配空间，然后打开配置文件 encoder.cfg，主要参数设置如表 4.1，将参数读入内存；然后，通过 ParseContent () 函数解析每个参数的具体含义，将相应的参数赋值，最后通过 PatchInp () 函数检验参数的一致性。

表 4.1 参数设置
Table 4.1 Parameter Configure

参数	数值	说明
InputFile	foreman	输入序列
StartFrame	0	开始编码帧号
FramesToBeEncoded	30	被编码的帧数
FrameRate	30.0	帧率 (秒)
SourceWidth	176	帧宽
SourceHeight	144	帧高
OutputFile	Test	编码后的输出文件名
ProfileIDC	100	框架 ID
LevelIDC	40	等级级 ID
IntraPeriod	5	I 帧的间隔
QPISlice	28	I 片参数的量化值
QPPSlice	28	P 片参数的量化值
SearchRange	16	最大搜索范围
NumberReferenceFrames	1	运动估计参考帧数
Bitrate	45020	字节流
InitialQP	24	初始化第一个 I 帧的量化值
YUVFormat	1	YUV 格式 (4:2:0)
BitDepthLuma	8	亮度值字节长度
BitDepthChroma	8	色度值字节长度
CbQPOffset	0	色度 Cb 的量化偏移值

续表 4.1 参数设置
Continue Table 4.1 Parameter Configure

参数	数值	说明
CrQPOffset	0	色度 Cr 的量化偏移值
Transform8x8Mode	1	变换模式 8×8
BasicUnit	11	基本单元的宏块数
WeightedPrediction	0	P 帧权值预测不可用
WeightedBiprediction	0	B 帧权值预测不可用
UseWeightedReferenceME	0	运动估计参考帧权值不可用
PicInterlace	0	图像编码为帧编码方式
MbInterlace	0	宏块编码为帧编码方式

4.1.3 帧内编码

帧内编码主要是除去图像的空间冗余，主要函数由 CodeOneIntra () 实现，具体流程如图 4.3 所示：

预测规程中首先通过 Intrapred_Luma () 函数得到预测模式下亮度预测值，存储残差图像数据到 Img 结构中，最后通过 Idwt_luma2 () 函数变换。色度系数由 Intrapred_chroma () 函数完成。

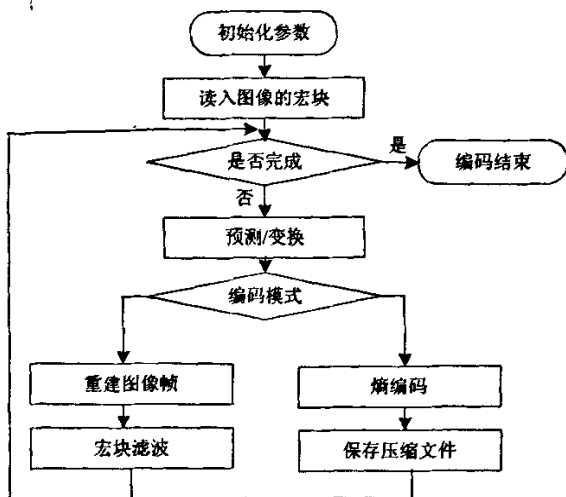


图 4.3 帧内编码
Fig. 4.3 intraframe coding

4.1.4 帧间编码

帧间编码程序主要包括 Motion_search () 计算运动矢量和 Motion_estimate () 计算运动估计组成, 其实现流程如图 4.4 所示:

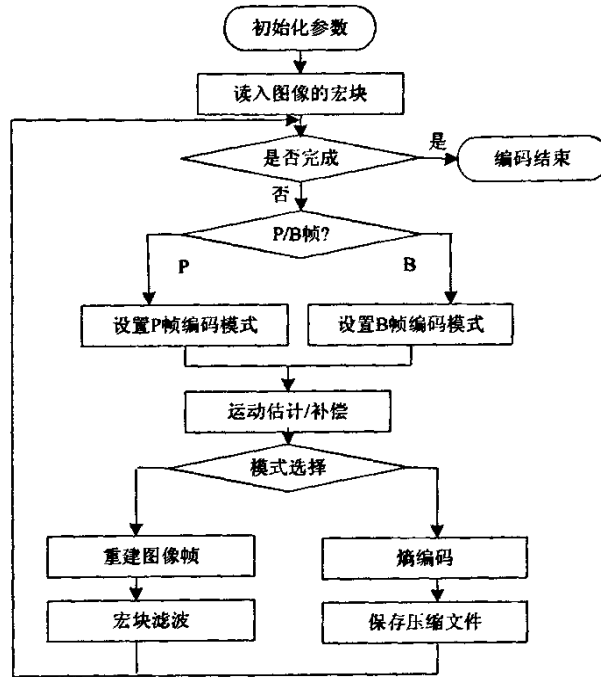


图 4.4 帧间编码

Fig. 4.4 interframe coding

4.2 算法分析

4.2.1 算法比较

在视频编码中, 运动估计算法对于块搜索的时间在整个编码过程中占具很大比例, 所以快速搜索的效率对编码器至关重要。本测试用搜索点数、图像 PSNR 来衡量空间域的全搜索 (FS)、三步法 (TSS)、交叉搜索算法 (CS)、二维对数搜索法 (2DLS)、菱形搜索法 (DS) 和本文算法对上述提供的序列进行对比测试, 以前一帧作参考帧。其中选择的宏块大小为 16×16 , FS 搜索范围是 15×15 , 匹配准则采用 SAD。表 4.2 和表 4.3 为对两种原始序列的估计及 PSNR 值, Foreman (qcif)、Test, 以 30 帧为比较对象。

表 4.2 平均每个宏块的搜索点数
Table 4.2 Average Search Points of Every Block

测试序列	FS	TSS	CS	2DLS	DS	本文算法
Foreman	255.00	25.00	15.78	9.00	13.32	29.41
Test	255.00	25.00	8.10	9.00	13.05	17.50

表 4.3 平均帧图像的峰值信噪比 PSNR (dB)

Table 4.3 The Average PSNR of Every Frame Image (dB)

测试序列	FS	TSS	CS	2DLS	DS	本文算法
Foreman	34.23	33.65	30.10	31.56	31.69	34.06
Test	39.65	38.95	38.23	38.26	38.23	39.61

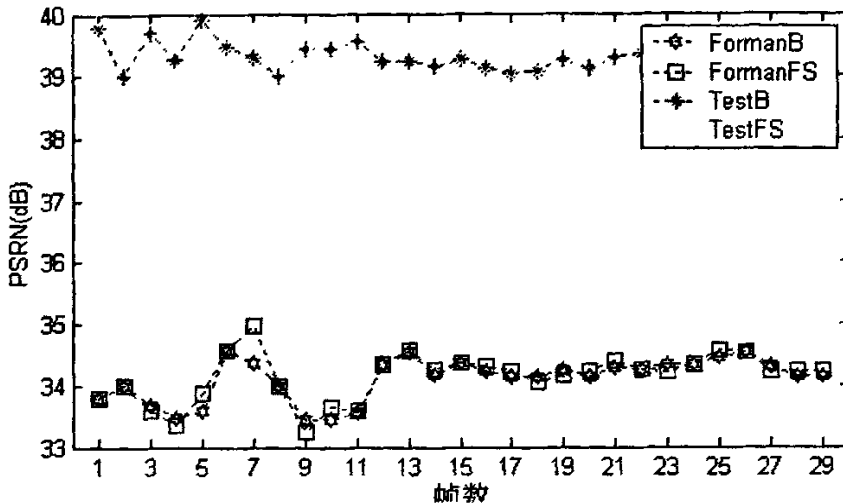


图 4.5 压缩时间比较

Fig. 4.5 Comparison of compression time

试验结果表明：在预测图像矢量方面，本文算法得出的 PSNR 值几乎等于由 FS 算法计算的 PSNR 值，对于 Test 序列，PSNR 值只比 FS 的算法计算的 PSNR 值减少 0.04dB，对于 Forman 序列，PSNR 值只比 FS 算法计算的 PSNR 减少 0.17dB；在计算量方面本算法大量减少了搜索点数，特别时对于背景变化较小的序列，虽然与一些算法相比本文算法搜索点数相对增加，但是对于得到的图像质量和 FS 相差较小，同时压缩时间也有一定的下降。对于缓慢运动的 Test 序列，本文算法的搜索点数相对更少，同时图像质量都好于其它算法，因为利用更新阈值的方法，中间出现提前结束的点较多。图 4.5 中 FormamFS, TestFS 为 FS 算法的测试结果；FormamB, TestB 为本论文提出的新算法的测试结果。

4.2.2 性能分析

对整体性能的测试，对示例的编码器的参数的主要设置是：YUV 格式比例 4:2:0；8×8 变换模式，对于编码流每隔 5 帧有一个 I 帧，对于 P 帧采用向前一个参考帧的运动估计模式，图像格式为了一致都采用 176×144，来检测算法的性能，具体其他设置参考表 4.1。

表 4.4 算法的压缩性能比较

Table 4.4 The Algorithm Compression Performance Compares

	Foreman		Test	
	压缩前 (KB)	压缩后 (KB)	压缩前 (KB)	压缩后 (KB)
H.263	2191	66	34647	592
本论文	2191	48	34647	457

表 4.5 对于不同量化因子下的 PSNR (dB)

Table 4.5 The PSNR Different Quantification Factor (dB)

	QP=24			QP =28			QP =32		
	y	u	v	y	u	v	y	u	v
Foreman	38.54	40.79	42.63	35.79	39.14	40.51	33.10	37.96	38.97
本论文	38.01	40.67	42.48	35.22	39.06	40.42	32.43	38.00	39.11

下面对两个序列的第 N 帧分别采用 H.263 标准和本文提出的算法进行压缩和提取的后的图像的截图，采用播放器 (Media Player Classic Version:6.4.8.4) 播放。通过对于编码后的图像进行播放，从主观质量上或从人眼视觉上都有一定的改善，满足实际需要。



图 4.6 Foreman 第一帧解码结果 (H.263)
Fig.4.6 The First Decoding Result of Forman (H.263)



图 4.7 Foreman 第一帧解码结果 (本论文)
Fig. 4.7 The First Decoding Result of Forman (This Paper)



图 4.8 Test 第十帧解码结果 (H.263)
Fig. 4.8 The Tenth Decoding Result of Test (H.263)



图 4.9 Test 第十帧解码结果 (本论文)
Fig. 4.9 The Tenth Decoding Result of Test (This Paper)

表 4.6 原始图像与解码后图像帧平均 PSNR 比较
Table 4.6 The Average PSNR of Original and Decoded Frames

QP=28	原始图像 PSNR (dB)	解码后图像 PSNR (dB)
Foreman	34.06	33.89
Test	39.61	39.53

从表 4.6 中得出：利用本文算法对解码前后图像帧中 PSNR 的数据进行对比，前后并没有很大的损失，这也说明了本文算法很好的重建压缩后的视频数据。

4.3 结论

通过上述实验数据，可以发现以下结论：

(1) 利用基于提升小波变换，减少了图像质量的损失，且增加的视频的清晰程度。

(2) 通过对运动估计算法的改进，能在很大情况下缩短了压缩的时间，提高了图像质量，改善了视频压缩算法。

通过上述实验对比，可以看出本系统与现在主流的压缩方法相比，基本达到了较高的压缩率并且因采用半像素估计技术使得图像质量得到一定的提高，系统设计总的来说是成功的，但也存在一些需要完善之处。具体的表现在：

(1) 压缩算法比较单一，只是在运动估计和提升小波变换方面做了改进。

(2) 对于阈值还要研究神经智能技术，增强自学习功能，更好的更新阈值，提高效率。

第五章 总结与展望

5.1 总结

小波分析于八十年代末取得突破性成就, Dauchies 提出构造具有紧支撑的光滑小波和 Mallat 的多分辨率分析及快速小波基的构造方法。目前在小波领域研究的热点问题是提升格式的小波, 即提升小波也叫第二代小波, 其不仅具有通用性和灵活性, 而且具有高效的实现方法。信号和图像是人们认识世界的主要信息源, 如何用较少的系数来表示信号和图像信息, 是许多研究领域需要解决的问题。在本文中, 由于双正交小波的线性特性, 其广泛应用于图象处理领域。小波滤波器的构造是很关键的技术。因此, 选择合适的滤波器对整个变换有很大的影响。通过综合考虑不同小波函数的具体特性, 本文选择了比较合适的小波基函数来实现视频文件的小波提升变换。

为了减少系数变换、量化时的能量损失, 得到较好的恢复, 研究了基于整数提升系数的滤波构造器。因为该方法算法简单, 易于实现, 且逆变换容易得到等优点, 其结果与其它小波变换方法获得的系数相似。

运动估计在整个视频压缩系统中的计算复杂度最大, 占了整个系统的 50% 以上。所以对运动估计的搜索速度至关重要, 本文研究块间匹配的运动估计的算法, 总结现有算法的优缺点, 结合整像素和半像素两种估计方式, 提出了新的逐级搜索方式算法, 该算法可以满足不同用户的需求; 其中自适应的阈值搜索终止规则, 解决搜索过程中局部最优的矛盾; 并对边缘的宏块估计提出新的解决办法。从测试的数据分析可知, 本文提出的算法及创新可以满足视频压缩技术能于实际应用中。

5.2 展望

本文进一步要研究的问题如下:

(1) 基于人的视觉系统特性的小波图像压缩方法。基于小波的图像压缩算法, 象 SPIHT 已经非常接近率失真极限, 其压缩比很难再有较大的提高。由于图像和视频中的冗余信息有很多种, 传统的压缩方法只是仅仅消除了部分相关性和统计冗余, 而视觉冗余是最具压缩潜力的冗余之一。本文拟进一步研究的主要内容是充分利用人的视觉系统特性, 同时结合本文的部分研究成果设计新的压缩算法。

(2) 对于阈值还要研究神经智能, 增强自学习功能, 更优的阈值更新算法, 提高效率, 并根据学习得到的结果制定出更加符合实际情况的运动估计终止规则, 以加快搜索

的速度。

(3) 对部分代码进一步优化, 尽量减少双循环结构的情况出现。

参考文献

1. Jain E.G.Richardson (著), 欧阳合, 韩军 (译). Video codec design Developing Image and Video Compression System [M], 科技大学出版社, 2005, 50-68.
2. 肖自美. 图像信息理论与压缩编码技术[M], 第一版, 广州, 中山大学出版社, 2000, 8-50.
3. 胡国荣. 数字视频压缩及其标准, 第一版[M], 北京, 北京广播学院出版社, 1999, 56-67.
4. Jerry D .Gibson. 多媒体数字压缩原理与标准[M], 第一版, 北京, 电子工业出版社, 2000, 105-107.
5. 沈兰荪. 视频编码与低速率传输[M], 第一版, 北京, 电子工业出版社, 2001, 167-122.
6. Hwang MS, Chang CC, Hwang KF. A Watermark technique based on one-way hash functions [J], IEEE Trans on Consumer Electronics, 1999, 45 (2) :286-294.
7. Khalid Sayood. Introduction to data compression [M], 1st Edition, San-Fransisco, Calif: Morgan Kaufmann publisher, 1996, 123-124.
8. Gsuh-Ming Hang, John W.Woods. Handboob of Visual Communication 1st Edition[M], SanDiego: Academic Press , 1995, 345-247.
9. Alfred Mertins. Signal analysis: Wavelet, filter banks, time-frequency transforms and applications [M] 1st Edition, Chicheser, West Sussex, England: J. Wiley, 1999, 98-100.
10. 崔锦泰. 小波分析导论[M], 程正兴译, 西安交通大学出版社, 1995, 167-203.
11. Mallat S. Multifrequency channel decompositions of images and wavelet models [J], IEEE Transaction on Acousitcs Speech and Signal Processing , 1989, 37:209-110.
12. 魏海. 基于小波的压缩域图像检索技术的初步研究 [D], 北京工业大学, 2001.
13. A R Calderbank, I Daubechies, W Sweldens, B L Yeo. Wavelet Transforms that mao intergers to integers [R]. Technical report, Department of Mathematics, Princeton University, 1996.
14. A S Lewis and G K Nowles. Imgae compression using the 2-D wavelet transform [J], IEEE Transacions on Image processing, 1992 1 (2) .
15. J M Shapiro Embedded image coding using zero trees of wavelet coefficients [J] ,IEEE Transactions on Signal Processing, 1993, 41 (12) .
16. A Said, W Pearlman . A new ,fast and efficient image codec based on set partitioning in hierarchical trees [J], IEEE Trans on Circuit and System for Video Technology , 1996, 6 (3) .

17. 沈兰荪,卓立.小波编码与网络视频传输[M],第一版,北京,科学出版社,2005,356-367.
18. W A Pearlman ,A Islam ,N Nagaraj and A Said . Efficient ,low-complexity image coding with a set-partitioning embedded block coder [M], 2004.
19. David Taybaman .High Performance Scalable Image Compression with EBCOT [J],IEEE Transactions on Image Processing,2000,9 (7) .
20. ISO/IEC JTC1/SC29/WG11, MPEG2002/N4928, The status of Interframe Wavelet Coding Exploration in MPEG [S]. Klagenfurt, July 2002.
21. G. Bhutani and W.A. Pearlman. Image sequence coding using in Proceedings of SPIE Conference Visual Communications [M] 1993, 2094: 463-471.
22. S. Martucci, I. Sodagar, T. Chiang, et al. A zerotree wavelet video coder[J]. IEEE Trans. on CSVT, 1997, 7 (1) .
23. D.Marpe and H.L. Cycon. Very Low Bit-Rate Video Coding using Wavelet-Based Techniques [J]. IEEE Trans. on CSVT, 1999, 9 (1) : 85-94.
24. M.Orchard and G.Sullivan. Overlapped block motion compensation:An estimation-theoretic approach [J].IEEE Trans. Image Processing, 1994,3:693-699.
25. Y.Zhang and S.Zafar .Motion-compensated wavelet transform coding for color video compression [J].IEEE Trans on CSVT,1992,2 (3) :285-296.
26. C.Caffario,C.Guaragnella,F.Bellifemine,etal Motion compensation and multiresolution coding.Signal Process.Image Commun[J],1994,6 (4) :123-142.
27. 27.S.Kim,T.Aboulnasr,and S.Panchathan.Adaptive Multiresolution.motion estimation techniques for wavelet based video coding Proc SPIE Visual Communications Image Processing[J],1998,3309:965-974.
28. S.Cui,Y.Wang and J.E.Fowler,Multihypothesis Motion Compensation in the Redundant Wavelet Domain Int Conf on Image Procesing [J] ,Barcelona,Spain,2003,2:53-56.
29. H. W.Park and H.S.Kim Motion estimation using low band shift method for wavelet based moving-picture coding [J].IEEE Transactions on Image Processing 2000,9 (4) :577-587.
30. F.Meyer,A. Averbush, and R. Coifman.Motion compensation coefficients for very low bit rate video coding in Proc [J], IEEE Int.Processing, Santa Barbara,CA,Oct.1997:638-641.
31. P. Cheng,J. Li, and C.J. Kuo, Multiscale video. Compensation using wavelet transform and motion Compensation in Proc [J].IEEE Int. Conf. Image. Processing 1995:606-609.
32. M.Ghanbari.The cross-search algorithm for motion estimation [J],IEEE Trans Communication,38,July 1990.
33. Z Xiong, K Ramchandran and M T orchard .Space-Frequency Quantization for wavelet

- image coding [J], IEEE Transactions on Image Processing ,1997,6 (5) .
34. T Naveen and J W Woods .Subband finite finite state scalar quantization [J] .IEEE Transactions on Image Processing ,1996,5 (1) .
35. QccPack-Quation ,Compression and coding Library[M] ,2003,122-124.
36. M.M.Reid,R,J.Millar, N.D.Black. Second-Generation Image Coding: An Overview. ACM Computation Surveys[J],March 1997,29 (1) .

致 谢

本论文是在我的导师高晓兴副教授的悉心指导和大力支持下完成的。自撰写论文以来，高晓兴老师虽然公务繁忙，仍然每周抽出时间来与我见面，询问我的研究情况和学习进展。对于我在学习和研究中遇到的困难，高老师都能够帮我耐心而细致的分析，给予我具体的指导和帮助。对我论文中的不足之处，高老师提出了许多宝贵的意见，使我少走了很多弯路。高老师谨慎的治学态度，渊博的学识，敏锐的思想，积极开拓、勇于进取的科学精神使我受益匪浅。同时，高老师也经常询问我的生活情况，在生活中也给予我很多关心和帮助。在这两年半的求学生活中，我不仅从高老师身上学到了应该怎样做学问，更学到了应该怎样做个正直的人，他将是我今后生活和工作中的榜样，在此表示衷心的感谢！

感谢以前的诸位师兄，和我同届诸位同学，以及诸位师弟、师妹们。他们或者在课题研究课题上给予了很多重要的提示和启发，或者在实验室的日常生活中给我带来了许多欢乐。

感谢我的父母，他们不辞辛劳地工作，在我读研期间给予我物质和精神上莫大的帮助，是我顺利完成学业的坚实后盾。

在这里，我还要感谢那些网上文章的撰稿人，他们经常将自己的开发或成果公布在网上，使我有许多可以参考的文档资料。