摘要

本文以视频对象分割技术为研究课题，首先介绍视频分割相关的理论与技术，然后对现有的基于运动和基于时空域相关两大类分割算法进行对比研究，并把重点放在基于 3D 区域生长的时空域分割算法的分析上。

从时空域分割要着重解决的几个关键问题入手，本文探讨了 3D 区域生长的种子分布和生成方法，给出了区域生长过程中的像素和元素之间的相似度准则和后处理过程，并构建了相应的时空域数据结构来支持生长算法的进行。通过区域生长算法，输出视频中具有颜色同质性的组件，接下来进行运动估计和元素运动轨迹分析得到这些同质组件的运动信息，并用空间聚类算法将具有运动一致性的组件合成视频对象。此外，本文还对视频时域分割、颜色空间选取、空域滤波等时空域分割要解决关键问题进行了探讨，并提出了一种自适应阈值切变镜头探测算法和加权中值滤波算法来解决这些问题。最后，将上述算法结合起来形成一个视频对象分割方案，有效地解决运动前景和背景分离的问题，并成功地完成从视频图像序列中抽取视频对象板的任务。

关键词：　　视频分割 视频对象 时空域分割 3D 区域生长 MPEG-4

Abstract

Video object segmentation techniques are discussed, theories and techniques related to video segmentation are introduced and the existing typical algorithms of motion-based and spatiotemporal segmentation are analyzed and compared with the emphasis on analysis of spatiotemporal segmentation algorithms based on 3D region growing.

Proceeding with several key problems about spatiotemporal segmentation, this paper discusses the generation and distribution of seeds in 3D region growing, provide the similarity measurement between pixel and volume, design the post processing and construct spatiotemporal data structure to support the algorithm. Homogeneous video components with similar color feature are obtained. Their motion trajectory is analyzed and motion estimation is made, and these components are clustered into objects with motion coherence. In addition, other key problems such as video temporal segmentation, color space selection and temporal filtering are discussed and an adaptive threshold video shot cut detection algorithm and a weighted median-filtering algorithm are presented as solution. At last, the algorithms are combined into an automatic video object segmentation schema, which can separate motion foreground from stationary background and extract video object plane from video image sequence in succeed.

Keyword: Video Segmentation Video Object 3D Region Growing Spatiotemporal Segmentation MPEG-4

独创性（或创新性）声明

本人声明所呈交的论文是我个人在导师的指导下进行的研究工作及所取得的研究成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其它人已发表或撰写过的研究成果；也不包含为获得西安电子科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志为本研究所做的任何贡献均已在论文中做了明确的说明并表示了谢意。

申请学位论文与资料若有不实之处，本人承担一切相关责任。

本人签名：陆璐　　日期：2004.1.4

关于论文使用授权的说明

本人完全了解西安电子科技大学有关保留和使用学位论文的规定，即：研究生在校攻读学位论文期间论文工作的知识产权单位属西安电子科技大学。本人保证毕业离校后，发表论文或使用论文工作成果时署名单位仍然为西安电子科技大学。学校有权保留送交论文的复印件，允许查阅和借阅论文；学校可以公布论文的全部或部分内容，可以允许采用影印、缩印、或其它复制手段保存论文。

本人签名：陆璐　　日期：2004.1.4

导师签名：王保保　　日期：2004.1.4.

第一章 绪论

1.1 研究背景

随着信息技术的发展，多媒体技术日益受到人们的关注。多媒体系统是数据、文字、声音、图形、图像和动画等各种媒体的有机组合，并与先进的计算机、通信技术相结合，使人们交流信息的方式获得了扩展，并影响着人们的交互方式、生活方式和工作方式。其中，数字视频是尤为重要的一种多媒体数据形式，它有着广泛的应用空间，是电影、电视、卡拉 ok、电子出版物等媒体信息进行数字化的重要基础。但是数字化的视频数据量非常巨大，这无疑给存储器的存储容量、通信干线的通道传输率以及计算机的速度都增加了极大的压力。为了解决多媒体信息在存储和传输过程的瓶颈——庞大的信息量和计算机系统的处理能力之间的矛盾，单纯用扩大存储器容量、增加传输率是不现实的，因此数字视频的压缩技术受到了前所未有的关注。所以，数字视频的编码压缩技术成为了多媒体领域的一项重要的技术，它为人们观赏、存储、交换和操纵视频信息，提供了有利的支持。新一代支持甚低码率传输的压缩标准 MPEG-4[1]，提出了基于内容编码的重要思想。正是多媒体领域产生的这种基于内容的可视信息表达方法的强烈需求，使视频对象分割技术成为一个研究热点。

视频对象分割的主要目的是通过在一系列连续图像帧中抽取感兴趣的对象，把视频表示成一个视频对象(VO),为基于对象的编码和基于内容的表达提供技术支持。视频分割有以下的重要的应用：

- 视频压缩和解压缩
- 视频对象操纵和编辑
- 视频的索引和检索
- 对象识别和鉴别
- 视频场景理解

从压缩角度看，基于对象的视频压缩标准，如 MPEG-4,需要视频对象分割技术。由于视频数据的数据量非常大，在带宽资源有限的网络上传输视频需要有效的编码技术。基于对象的表达方式可以标出图像帧中重要的部分，使得视频可以高效编码来满足传输的需要。特别是在个人通讯终端如移动电话、PDA、可视电话日益蓬勃发展的今天，强烈需要一种甚低码率的编码方式，来满足用户对多媒体信息的需求。

有了好的分割方法，就可以访问和操纵视频中的对象，这为人造场景对象和自然场景对象更好的融合在一起提供了有效的工具。实现更好的视频的非线性编

辑功能，如剪切视频中某些对象到其它的背景或场景中，就是一种很有用的功能。另外，交互式电视技术的发展，出现了对可交互的媒体[2]的需要，例如交互式的广告，指用户收看广告时可选择感兴趣的商品，然后该商品的详细信息同时呈现在用户的面前。为了实现这一点，对视频的分割是必不可少的。

目前，市面上的视频数据库只能通过像颜色、纹理和简单的运动等简单的统计特征来检索视频数据，它们或者检索能力有限或者有应用范围限制。如果视频可以独立的对象形式来存储，那么索引和检索视频信息就会象检索和索引文本信息那么简单。能从根本上管理可视信息的工具必须具有以语义方式自动描述和索引视频序列的能力。这种工具才可以在巨大的视频数据库中查询到想要的视频片断和视频对象。有效利用存储影片和探测监控视频中的特定活动都有广阔的应用空间，这需要引入对象的概念才能得到完满的解决。

许多机器视觉问题都要借助视频分割技术才能完成。安装有自动驾驶系统的汽车要通过分析视频来获取周围环境的信息。而且，它要求高层次的图像理解和解释如监控视频中的场合和特殊事件的跟踪能力。举例子来说，步行道和高速交通可以用分割出的人和车的密度来区分开。通过对象分割，还可以检测到快速移动的汽车，路上障碍物，路面上其它的异常活动等。再加上行为识别的用户接口，就可以实现禁区、停车位、电梯都可以自动监控。

尽管人类可以快速解释包含在各种形式信息的语义，但是计算机来理解可视信息还处在初级阶段。未来的标准要成功，分割工具是非常关键的。但是把图像序列自动分割成语义对象是一项很有挑战性的工作。

尽管人们已对视频信息处理的基本方法有了很好的理解，但是在这方面的还有许多问题和困难等待解决。其中视频分割是这些问题中需要首先解决的，说道视频分割，就不得不提多媒体压缩标准，因为视频分割技术的发展，跟视频的编解码标准的发展紧密相关的。

1.2 视频编码标准

未经压缩的音视频数据需要巨大的存储空间来存放，传输和处理都不方便。为了高效存储和传输视频，人们开发了各种压缩算法和压缩标准。在压缩标准中，编解码技术是最关键的，编解码技术的发展促使数字视频得到广泛应用和传播。以不同的编码技术为核心，运动图像专家组(Motion Pictures Experts Group)定义了数字多媒体内容的的编码和压缩系统,陆续推出 MPEG-1、 MPEG-2、MPEG-4 和 MPEG-7 等多媒体压缩标准。

MPEG-1 和 MPEG-2

MPEG-1 处理的是标准图像交换格式(Standard Interchange format，SIF)或者称

为源输入格式(Source Input Format，SIF)的电视，即 NTSC 制为 352 像素× 240 行/帧× 30 帧/秒，PAL 制为 352 像素× 288 行/帧× 25 帧/秒，压缩的输出速率定义在 1.5 Mbit/s 以下。这个标准主要是针对当时具有这种数据传输率的 CD-ROM 和网络而开发的，用于在 CD-ROM 上存储数字影视和在网络上传输数字影视。

MPEG-2 标准从 1990 年开始研究，1994 发布 DIS。它是一个直接与数字电视广播有关的高质量图像和声音编码标准。MPEG-2 可以说是 MPEG-1 的扩充，因为它们的基本编码算法都相同。但 MPEG-2 增加了许多 MPEG-1 所没有的功能，例如增加了隔行扫描电视的编码，提供了位速率的可变性能(scalability)功能。MPEG-2 要达到的最基本目标是：位速率为 4～9 Mbit/s，最高达 15 Mbit/s。

MPEG-1 和 MPEG-2 标准采用第一代编码技术，以信息论为理论基础，以象素块为编码实体，把图像分成许多小方块来处理，依此适应非静态图像的特性。通常采用预测编码、变换编码和统计编码等经典编码方法。虽然基于块的算法参数是可以改变的，但是现实场景中的对象可不是由方块组成的。当压缩率增加时，这种块结构在解压图像中可被人眼察觉，这就是所谓的“块效应”。

MPEG-7

MPEG-7[3]的工作于 1996 年启动，名称叫做多媒体内容描述接口(Multimedia Content Description Interface)，目的是制定一套描述符标准，用来描述各种类型的多媒体信息及它们之间的关系，以便更快更有效地检索信息。例如，用户可能想访问一张关于视频内容的表，他可以从一个条目跳到另一个条目。这就要求把视频数据按照镜头和场景结构化。

与其它 MPEG 标准一样，MPEG-7 是为满足特定需求而制定的视听信息标准。MPEG-7 标准也是建立在其它标准之上的，例如，PCM, MPEG-1, MPEG-2 和 MPEG-4 等等。MPEG-7 继承了 MPEG-4 中使用的形状描述符、MPEG-1 和 MPEG-2 中使用的运动矢量(motion vector)。

1.3 MPEG-4 与视频对象 VO

MPEG-4 从 1994 年开始工作，它是为视听(audio-visual)数据的编码和交互播放开发算法和工具，是一个甚低码率多媒体通信标准。作为新一代多媒体应用标准，它提供基于对象的高可交互性功能、通用访问机制、健壮的错误探测机制和高效的压缩。

MPEG-4 的目标是要在异构网络环境下能够高度可靠地工作，并且具有很强的交互功能。为了达到这个目标，MPEG-4 引入了对象基表达(object-based representation)的概念，用来表达视听对象(audio/visual objects，AVO)。MPEG-4 扩充了编码的数据类型，由自然数据对象扩展到计算机生成的合成数据对象，采

用合成对象/自然对象混合编码(Synthetic/Natural Hybrid Coding，SNHC)算法；在实现交互功能和重用对象中引入了组合、合成和编排等重要概念。MPEG-4 系统构造如图 1-1 所示。

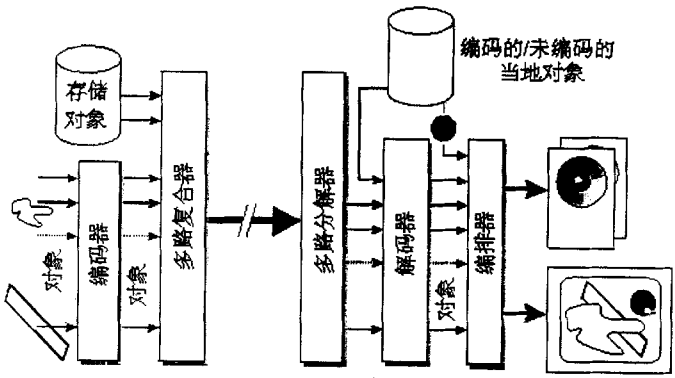


图 1-1 MPEG-4 的系统构造图

MPEG-4 最重要的特点是它引入了 VO(Video Object)的概念,并用于描述视频画面。VO 是有实际意义的物理实体，而不是出于编码效率分割出来的某些部件。在视频序列的一个画面可由单个或者多个 VOP(Video Object Plane)组成,它是 VO 在某个时刻的一个表示，场景中属于同一对象的连续的 VOP 被称作视频对象。MPEG-4 编码中最关键的部分是 VO 的形成和表示。VO 的形成要用到最先进的图像理解、识别和分割算法。MPEG-4 标准本身并不定义这些算法，而是让用户自己开发，这可能是用好 MPEG-4 最难的部分。基于对象的视频分割目的是从视频序列中抽取 VO 和 VOP，并把它们按一定的形式组织存储起来，所以说研究基于对象的视频分割技术，是有很强的现实意义的。对象概念的引入，使 MPEG-4 具有了许多新的特性：

- 交互性：提供了基于内容交互的机制，在编码、解码和物体合成阶段均可与每一个音视频对象交互，这意味着在这样的视听通信系统中，人不仅可以看见物体在什么地方，还容许我们采取行动改变它的位置；
- 通用性：能够处理各种各样的音视频对象，不仅包括图像和视频，还包括各种图形、3D 动画及文本，同时使自然目标和人工合成目标共存。而且可根据各种网络的不同特性，进行高效率低码率的信息传输。实现通用的多媒体信息的存取和传输。；

- 易用性：提出基于内容的压缩，使信息处理技术的方式更加接近人自身的信息处理方式。这就使得人在进行多媒体信息处理时，直接和场景中的物体打交道，而不是具有抽象概念的像素。

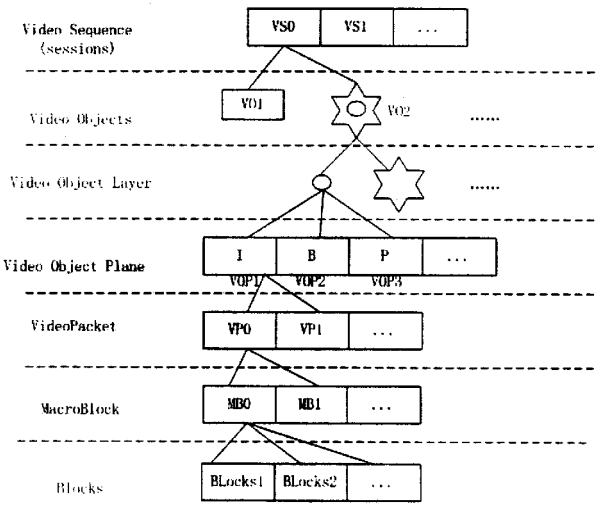


图 1-2 MPEG-4 视频层次化数据结构

不同于 MPEG-1 和 MPEG-2 那样一帧一帧进行编码，基于对象编码的 MPEG-4 用层次化的数据结构来表示视频数据(见图 1-2)，引入了下列概念，：

- 视频序列(VS: Video Session)：VS 是其它 3 层数据的入口。一个完整的视频包括多个 VS。
- 视频目标(VO: Video Object)：VO 即是场景中的特定目标。是有实际意义的物理实体，而不是出于编码效率分割出来的某些部件。
- 视频对象层(VOL: Video Object Layer)：VOL 是 VO 的时间或空间的伸缩性描述。VO 的描述可以在不同时间分辨率和空间分辨率上进行的。它可以只包括一个基本层，也可以包括多个分辨率增强层。目标的伸缩性是通过 VOL 来实现的。
- 视频对象板(VOP: Video Object Plane)：VOP 是 VO 在某个时间的存在。是 VO 在不同 VOL 层的时间序列。每一帧图像都被分割成很多任意形状的 VOP,每个 VOP 都覆盖了一个特定的感兴趣的视频内容。因此，在基于对象的编码中，输入信息不再象基于 DCT 的块编码那样，针对矩形区域进行编码。

MPEG-4 还提供“对象层”概念[1]，把不同的对象编码到不同的位流层。这个特征允许访问和操纵场景中的不同的音频对象(AO)和视频对象(VO)。为了支持分别解码不同的对象，每个对象的形状、运动、空间坐标和编码信息被分别编到不同的“对象层”。用户通过解压所有的视频对象层来重构整个场景，也可以仅解压部分对象重构场景。利用编码到不同码流的信息，操作对象进行转换、旋转、标记和缩放等成为可能。另外，不属于原始场景的新对象可以加入场景或者可以忽略原有的对象。在接收端的构造部件如图 1-3 所示。

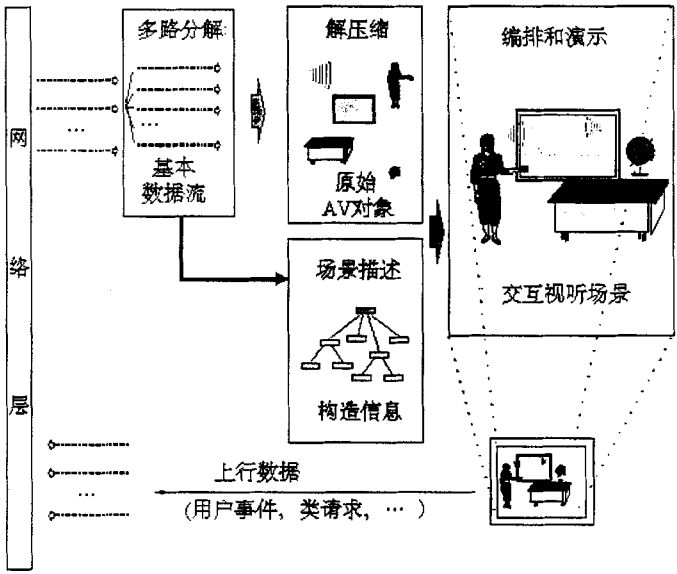


图 1-3 MPEG-4 接收端的构造部件

1.4 本文工作

本文以视频对象分割技术为研究课题，深入地进行国内外视频对象分割算法的研究，对相关分割技术进行了分类，对现有的基于运动的分割算法和基于时空相关的分割算法进行比较。以此为基础，在时空域分割方面展开研究，对基于 3D 区域生长的时空域分割方法进行了探索，并对实施算法要解决的关键问题提出了自己的解决办法。最后，将相关算法组合在一起形成了以 MPEG-4 为服务目标的视频对象自动分割方案，应用该方案进行 VOP 的抽取，能取得比较好的效果。下面介绍本文相关章节的内容安排。

第一章 绪论。这一章主要阐述视频对象分割技术的概念和应用需求，以及与视频分割技术的发展密切相关的多媒体压缩标准。由于 MPEG-4 标准是视频对象分割技术的最重要应用，所以重点介绍了该标准并引出视频对象的概念。

第二章 视频分割相关理论与技术。这一章讨论视频分割要使用的技术与理论，为后文的讨论做理论铺垫，分别讨论了运动估计、块运动分析、块匹配技术和空域图像分割技术，其中块匹配、边界分割、区域生长和空间聚类等方法是后文分割算法的重要支撑技术。

第三章 现有分割算法简介。本章对现有的分割方法进行了分类，同时介绍和比较基于运动和时空相关的两类算法。由于基于运动的方法有缺陷，所以把空域信息与运动信息相结合是很重要的，本章重点分析以变化检测模板、数学形态学为工具的时空域算法和其它混合算法。

第四章 时空域分割关键算法研究。本章重点解决时空域生长视频分割的相关问题，首先提出一种视频自适应阈值的视频分段算法把视频分成一个个镜头，在镜头内才可能对视频内容进行分析。接着讨论各种颜色空间的特性，选择 HSV 颜色空间进行视频分割。区域生长的算法对图像噪声十分敏感，本文使用快速的加权中值滤波算法去除噪声，取得了很好的效果。接下来对区域生长要解决的种子选择问题、相似性规则和后处理进行了分析和讨论，最后对生长得到的同质元素进行运动特征聚类，分割出视频对象。

第五章 时空域视频对象分割方案。在这一章中将相关的工作成果结合在一起，提出一种基于三维区域生长时空域分割方案。使用该方案可以将视频中的运动前景和背景实施分离，并抽取 VOP。最后，给出了实验结果验证方案的有效性。

第六章 总结全文内容和工作，并对需要进一步研究的问题进行了展望。

第二章 视频分割相关理论与技术

2.1 数字视频

数字视频可以采用光栅扫描或直接用数字视频摄像机获得，在多媒体信息中，它属于一种视觉媒体信息。物体在成像平面的投影被采样成离散的一幅幅数字图像，这些图像也称为帧。每一帧由水平和垂直离散化的阵列输出值组成，每一个象素点按照一定的存储结构在帧缓冲器中形成我们常说的位图。对视频信息按时间逐帧进行数字化得到数字图像序列，如图 2－1 所示。

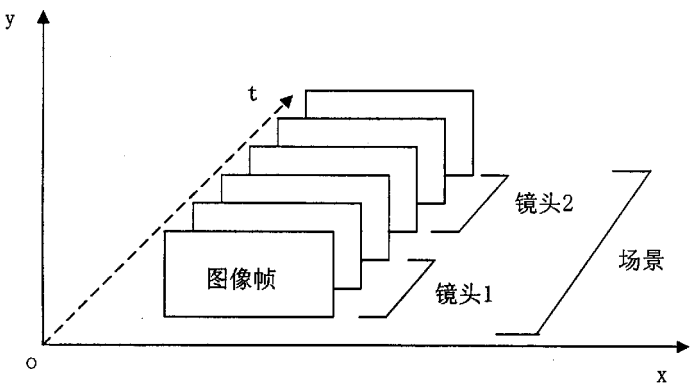


图 2－1 数字图像序列形成示意图

由上图可以看出数字视频由多幅连续的图像序列构成。其中，x 和 y 轴表示水平和垂直的空间维，t 表示时间维。沿着 t 轴方向若间隔△t，利用人类视觉暂留效应，可以形成连续的动态图像。沿着 x 轴方向的扫描行上分布有象素点，y 方向表示垂直方向的列数。这样每一个象素点的颜色或亮度 E 可以表示为 XY 平面的函数 E(x，y，t)。当在监视器上显示数字视频时，每个象素被表示为具有指定给该象素的一种恒定彩色的一个矩形区域。

2.2 运动分割理论

2.2.1 运动估计

研究表明，人眼对图像的静止部分具有较高的空间分辨力和较低的时间分辨力。利用这种人眼的这种特性，可以进行图像序列的压缩，首先将图像分割成静止部分和运动部分分别进行处理，静止部分可以重复利用上一帧的数据，而对运动部分则设法测定其相对于上一帧的位移量，用位移量进行运动部分的预测，这样就用存储的静态帧和用位移量作为补偿得到预测帧，实现帧间预测效果，构成完整的图像，把这种技术称为运动补偿技术[12]。

在运动补偿编码中，运动补偿和预测在压缩中起了占非常重要的地位。运动估计是对来自参考帧中的像素在当前帧进行的估计过程。运动估计技术是依赖于两个假设：一个是物体运动的轨道上照明是恒定的。也就是认为物体运动时照明光线的不随时间改变，只有这样才能保证图像上亮度模式的改变是由运动引起的，而不是光照改变引起的。二是没有遮挡的背景的问题。虽然这些假设不足以获得真实世界的视频序列，但是多数运动估算方法都建立在这些假设之上。运动估计的一个关键问题是如何参数化运动场，也就是如何表示运动的问题[14]。通常按照不同的运动表示法，把运动估计技术分为象素运动估计、块运动估计、区域运动估计和全局运动估计如图 2-1。其中象素运动估计用光流模型、块运动估计用块运动模型，基于对象的运动分割技术经常用到区域运动估计和全局运动估计。

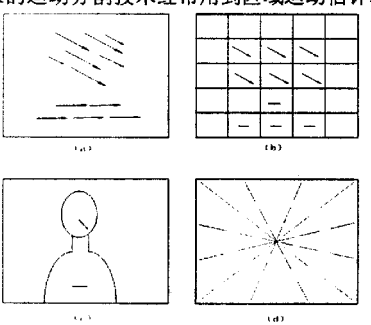


图 2-1 运动估计分类

(a)像素运动估计(b) 块运动估计

(c)区域运动估计(d)全局运动估计

2.2.2 光流模型

光流的概念是 Glibson 于 1950 年首先提出的[27]。人眼是通过在不同的时刻认出相应的一些点来感觉运动的，这种对应性通常是由假定一个点的彩色和亮度在运动以后不改变来确定的。当物体在运动时，物体表面的亮度模式发生改变，我们就感觉到了运动。光流(optical flow)是指图象亮度模式的表观运动。虽然光流可能不等同于真实的二维运动。当只能利用图像的彩色信息时，所能够得到的最精确估计就是光流。光流场(optical flow field)是一种二维瞬时速度场，其中二维运动速度矢量是三维速度矢量在成象表面的投影。光流不仅包括了被观察物体的运动信息，而且携带着有关景物结构的丰富信息。

在运动估算算法中，光流方程起着关键的作用。下面介绍一下光流约束方程。

设 I(x, y, t)是图像点(x, y)在时刻 t 的照度，如果 u(x, y)和 v(x, y)是该点光流的 x 和 y 分量，假定点在 t+∇t 时运动到(x+∇x, y+∇y)时，照度保持不变，其中 ∇x=u∇t, ∇y=v∇t，也就是

$$I(x+\nabla u\,t, y+\nabla v\,t, t+\nabla t) = I(x, y, t) \tag{2.1}$$

这一约束还不能唯一求解 u、v，通常要加上其他的约束条件，比如，运动场连续行的假设。如果亮度随着 x、y、t 光滑的变化，则可以将上式用泰勒级数展开，

$$I(x,y,t)+\nabla x\frac{\partial I}{\partial x}+\nabla y\frac{\partial I}{\partial y}+\nabla t\frac{\partial I}{\partial t}+e=I(x,y,t) \tag{2.2}$$

e 是高阶无穷小。可以推得

$$\frac{\partial I}{\partial x}\frac{dx}{dt}+\frac{\partial I}{\partial y}\frac{dy}{dt}+\frac{\partial I}{\partial t}=0 \tag{2.3}$$

上式实际上就是

$$\frac{dI(x,y,t)}{dt}=0 \tag{2.4}$$

令 $I_x=\frac{\partial I}{\partial x}, I_y=\frac{\partial I}{\partial y}, I_t=\frac{\partial I}{\partial t}, u=\frac{dx}{dt}, v=\frac{dy}{dt}$

则可得到空间和时间梯度与速度分量之间的关系,其中 u，v 分别像素点流矢量的垂直分量和水平分量：

$$I_x u+I_y v+I_t=0 \tag{2.5}$$

也可以表示成(2.6)，其中 ∇I^T 和 I_t 分别是图像序列的时域梯度和空域梯度，

$$\nabla I^T \bullet V + I_t = 0 \tag{2.6}$$

由上式可以看出我们不能单凭 ∇I^T 和 I_t 确定流矢量 V。为了解出两个未知量，必须添加附加条件。通常的约束是流矢量在空间平滑变化，使我们能利用象素周围一个小的邻域的亮度变化去估计该处的运动。一般采用再约束方程上加一个平滑量来约束速度场，这样运动场既满足光流约束又满足全局的平滑性，如 Horn-schunck 方法[13]。

(a) (b)

图 2-2 用光流约束和 Horn-Schunck 方法得到光流场分布

2.2.3 块运动分析

由于光流法的运算复杂度，难以达到实时处理的要求，况且有些情况下并不要求计算出每个象素的精确的运动矢量。因此基于块的运动分析算法，在数字视频编码技术中得到了广泛的应用。块的运动通常分为平移、旋转、仿射等运动形式，一般情况下，块运动是这些运动的组合，称为变形运动。下面我们详细讨论块的运动模型。

1、块平移

基于块的模型最简单的形式是平移的块，假设图像中每一个块都是作单纯的平移运动。在第 k 帧中的一个中心位于 X(x,y)的 N×N 块 B 被模型化成为帧 k+1(1 是整数)中同样尺寸块的一个完全位移形式。也就是说，在第 K 帧中，中心位于 X(x,y)的块 B,在第 K+1 帧时，块 B 的所有象素之间关系及其灰度值保持不变，但中心位置移到了 $X'(x+d_x, y+d_y)$,其中 d_x, d_y 是块 B 平移位移分量。

$$s(x,y,k)=s(x+d_x,y+d_y,k+1) \tag{2.7}$$

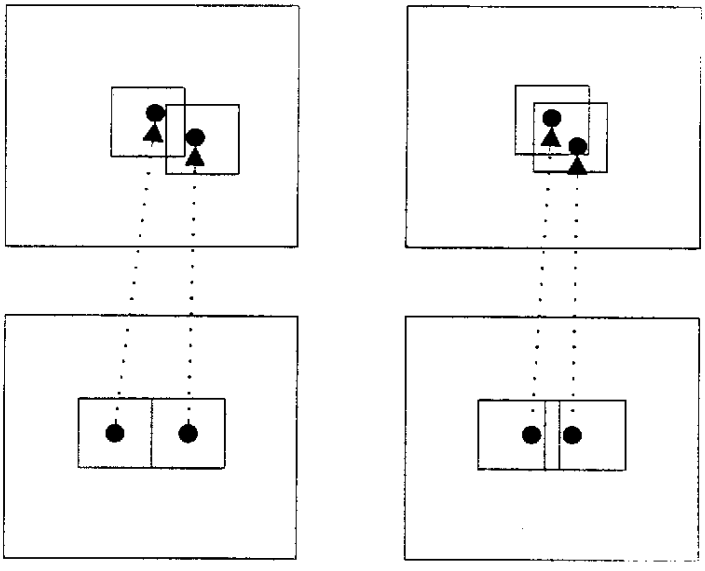


图 2-3 块运动平移图

参照图 2－3 所示，块运动可能存在两种情况：块 B 可能重叠或未重叠。在未重叠的情况下，整个块使用同一运动矢量，可以拷贝 k 帧中每个象素，使 k+1 帧中的相应块得到运动补偿。对于重叠的情况，我们可以计算重叠范围的运动矢量的平均值作为整个块的运动矢量。

基于块的模型优点在于不需要很多附加条件表示运动场，运动矢量的估算通常采用块匹配的办法，相对于光流计算上较简单。但是物体并不是由一个块组成的，特别是物体边界处容易出现“块效应”。

2、二维运动模型

物体在三维空间运动，而我们看到的图像是物体运动在摄像机平面上的投影，为了推广块运动，需要建立了二维运动模型，常见的模型由以下几种[22]：

(1)透视变换模型

假定物体在 Z 方向没有平移运动，或者当成像物体具有一个平坦表面时，透视变化可由式(2.8)来表示：

$$x'=\frac{a_0+a_1x+a_2y}{1+c_1x+c_2y},\ y'=\frac{b_0+b_1x+b_2y}{1+c_1x+c_2y} \tag{2.8}$$

这就是所谓的 8 参数模型，其中 5 个运动参数和 3 个物体表面参数。在研究帧间运动和视频配准时，这个投影映射是一个重要的关系式。

(2)仿射运动模型

仿射运动是对投影映射的近似，仿射运动具有以下的形式，就是 6 参数

模型：

$$\begin{bmatrix} d_x(x,y) \\ d_y(x,y) \end{bmatrix} = \begin{bmatrix} a_0 + a_1 x + a_2 y \\ b_0 + b_1 x + b_2 y \end{bmatrix} \tag{2.9}$$

(3)双线性模型

双线性具有以下形式：

$$\begin{bmatrix} d_x(x,y) \\ d_y(x,y) \end{bmatrix} = \begin{bmatrix} a_0 + a_1 x + a_2 y + a_3 xy \\ b_0 + b_1 x + b_2 y + b_3 xy \end{bmatrix} \tag{2.10}$$

以上介绍了块运动的 4 种基本的运动模型，实现效果参加图 2－4 所示。

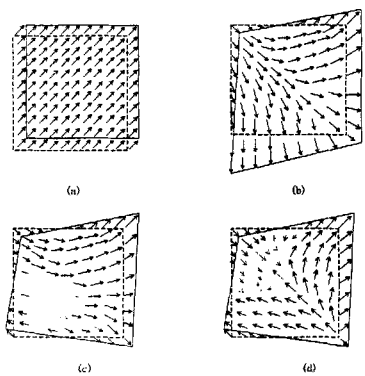


图 2－4 基本运动模型

(a)平移的；(b)仿射的；(c)双线性的；(d)投影的

2.2.4 块匹配

利用块运动估计和光流计算的不同，它不用计算每一个像素的运动，而只是计算若干象素组成的象素块的运动，对于许多图像的分析和估计应用来说，块运动分析是一种很好的近似。虽然基于平移运动的块运动补偿不适于缩放、旋转运动，但是，块匹配算法跟踪能力强，实现简单,得到了广泛的应用。

块匹配的基本思想如图2-5示，其中帧K的位移通过考虑一个中心定位于(x,y)的位移通过考虑一个中心定位于(x,y)的 $N_1 \times N_2$ 块，同时搜索帧 K+1 来找出同样大小的最佳匹配块的位置来确定。

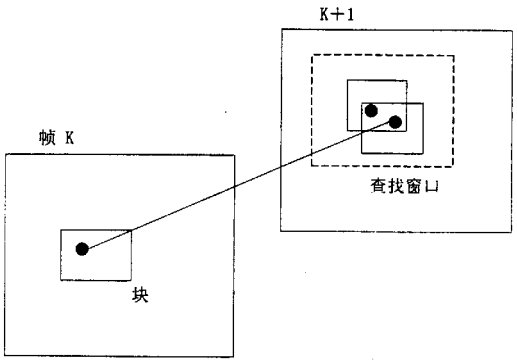


图 2－5 块匹配的基本思想示意图

为了检测当前帧的块与参考帧候选块的相似性，就定义了匹配的准则，块匹配可以依据各种准则来确定它的运动矢量大小，包括最小均方误差函数(MSE)最小平均绝对差值函数(MAD),最大匹配象素统计(MPC)。

在最小均方误差函数准则中计算 MSE,定义成：

$$MSE(d_1,d_2)=\frac{1}{N_1N_2}\sum_{(x,y)\in B}\left[s(x,y,k)-s(x+d_x,y+d_y,k+1)\right]^2 \tag{2.11}$$

其中 B 代表 $N_1\times N_2$ 块，作为可选择的运动矢量(d_x,d_y)的集合。最终块的运动矢量是使 MSE 达到最小值的运动矢量(d_x,d_y)，也就是

$$[\hat{d}_x,\hat{d}_y]^T=\arg\min_{(d_x,d_y)} MSE(d_x,d_y) \tag{2.12}$$

最小平均绝对差值函数(MAD)准则定义成：

$$MAD(d_x,d_y)=\frac{1}{N_1N_2}\sum_{(x,y)\in B}\left|s(x,y,k)-s(x+d_x,y+d_y,k+1)\right| \tag{2.13}$$

位移估算用下式给出

$$[\hat{d}_x,\hat{d}_y]^T=\arg\min_{(d_x,d_y)} MAD(d_x,d_y) \tag{2.14}$$

最大匹配象素统计准则(MPC)，在这个方法中，块 B 中每一个象素依据下式被划分成匹配象素和非匹配象素，其中 t 是估算阈值。

$$MPC(d_x,d_y)=\sum_{(x,y)\in B}T(x,y:d_x,d_y),$$

$$\text{其中}T(x,y;d_x,d_y)=\begin{cases}1 & \left|s(x,y,k)-s(x+d_x,y+d_y,k+1)\right|\le t\\ 0 & \text{其它}\end{cases} \tag{2.15}$$

位移估算用下式给出

$$[\hat{d}_x,\hat{d}_y]^T = \arg\max_{(d_x,d_y)} MPC(d_x,d_y) \tag{2.16}$$

为了得到最优的块匹配，通常依据上面所讲的评价准则，采用搜索算法来得到块运动矢量的解算。最简单的方法是全面搜索算法(EBMA)，在一个预定义大小的窗口中，对每个可能的位移应用匹配准则，这种方法很费时。为了加快搜索，在牺牲估计精度的前提下,开发了各种快匹配算法快速算法。一种常用的快速算法是三步搜索法[13]，这种搜索的步长从等于或者略大于最大搜索范围的一半开始。每一步中，比较九个搜索点。它们包括搜索正方形的中心点和八个位于搜索区边界上的搜索点。每一步以后搜索步长减小一半，至搜索步长为一个象素时结束搜索。在每一个新的搜索步中，搜索中心点移到由前一步得到的最佳匹配点。

2.3 空域分割技术

2.3.1 边界分割

图像分割是指把图像分成各自具有特性的区域并提取出感兴趣目标区域的技术和过程。这里特性可以是灰度、颜色、纹理等，目标可以对应单个区域，也可以对应多个区域。而边缘分割技术对于处理数字图像分割非常重要，因为边缘是所要提取目标和背景的分界线．分离出边缘才能将目标和背景区分开来。在图像中，边界表明一个个特征区域的终结和另一个特征区域的开始。下面从串行和并行两个方面讨论边界分割技术。

1、串行边界分割

串行边界技术指采用串行的方法通过对目标边界的检测来实现图像分割的技术。串行边界技术通常通过搜索边界点来工作，所以实现起来需要注意以下三个方面：

(1)确定起始边界点，顺序搜索从这里开始；

(2)选择合适的搜索策略，确定先前的结果对选择下一个检测象素和下一个结果的影响，并根据一定的机理依次的检测新的边界点；

(3)设定中止条件，用来结束搜索的进行所需的条件。

串行分割技术主要可采取两种策略：一、先检查边缘点,再连接它们；二、对边界点的检查和连接交叉或结合进行。

2、并行边界分割

并行边界检测技术指采用并行的方法通过对目标边界的检测来实现图像分割的技术。并行边界技术在确定图像中区域边界时是同步进行的，从某种意义上说图像大部分信息都是集中在区域的边界上,所以确定边界对于场景的理解很重要。

所涉及的算法比较多，论文中主要用到了基本的梯度算子法和流行的 canny 方法，下面分别介绍。

(1)梯度算子法

梯度对应一阶导数，梯度算子是一阶导数算子。对一个连续函数 $f(x, y)$，它在位置(x, y)的梯度可表示为一个矢量：

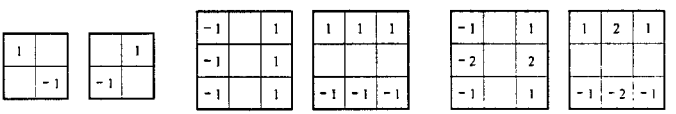
$$\nabla f(x,y)=\begin{bmatrix} G_x & G_y \end{bmatrix}^{\mathrm{T}}=\begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \end{bmatrix}^{\mathrm{T}} \tag{2.18}$$

这个矢量的幅度和方向角分别为：

$$\mathrm{mag}(\nabla f)=\left[G_x^2+G_y^2\right]^{1/2} \tag{2.19}$$

$$\phi(x,y)=\arctan\left(G_y/G_x\right) \tag{2.20}$$

在实际中常用小区域模板卷积来近似计算偏导数。对 G_x 和 G_y 各用一个模板，所以需要两个模板组合起来以构成一个梯度算子。最简单的梯度算子是 Roberts 算子，见 2－6 图(a)所示。比较常用的还有 Prewitt 算子，见 2－6 图(b)，Sobel 算子，见 2－6 图(c)，其中 sobel 算子是效果较好的一种。



(a)Roberts (b)Prewitt (c)Sobel

图 2－6 几种常见梯度算子模板

(2)坎尼算子

坎尼(Canny)把边缘检测问题转换为检测单位函数极大值的问题[25]。边缘检测是一种比较新的边缘检测算子，具有很好的边缘检测性能，得到了越来越广泛的应用。Canny 边缘检测法利用高斯函数的一阶微分，它能在噪声抑制和边缘检测之间取得较好的平衡。具体步骤如下：

- 用高斯滤波器来对图像滤波，可以去除图像中的噪声。
- 用高斯算子的一阶微分对图像进行滤波，得到每个像素梯度的大小|G|和方向 θ。

$$|G|=\left[\left(\frac{\partial f}{\partial x}\right)^2+\left(\frac{\partial f}{\partial y}\right)^2\right]^{\frac{1}{2}} \tag{2.21}$$

$$\theta=\tan^{-1}\left[\frac{\partial f}{\partial y}\Big/\frac{\partial f}{\partial x}\right] \tag{2.22}$$

其中，f 为滤波后的图像。

• 对梯度进行“非极大抑制”。

梯度的方向可以被定义为属于 4 个区之一，各个区别不同的邻近像素用来进行比较，以决定局部极大值。这 4 个区及其相应的比较方向如图表 2－7 所示。

4	3	2
1	x	1
2	3	4

图 2－7

• 对梯度取两次阈值得到两个阈值 T1 和 T2，T1＝0.4*T2。我们把梯度值小于 T1 的象素的灰度设为 0，得到图像 1。然后把梯度值小于 T2 的象素的灰度设为 0，得到图像 2。由于图像 2 的阈值较高，去除了大部分噪声，但同时也损失了有用的边缘信息。而图像 1 的阈值较低，保留了较多的信息。我们可以以图像 2 为基础以图像 1 为补充来连接图像的边缘。

• 连接边缘的具体步骤如下：

(1) 对图像 2 进行扫描，当遇到一个非零灰度的像素 P 时，跟踪以 P 为开始点的轮廓线，直到该轮廓续的终点 Q。

(2)考察图像 1 中与图像 2 中 Q 点位置对应的点 Q 的 8-邻近区域。如果 Q′ 点的 8-邻近区域中有非零像素 R′ 存在，则将其包括到图像 2 中，作为点 R。从 R′ 开始，重复第(1)步，直到我们在图像 1 和图像 2 中都无法继续为止。

(3)当完成对包含 P 的轮廓线的连接之后，将这条轮廓线标记为以访问。回到第(1)步，寻找下一条轮廓线。重复步骤(1)、(2)、(3)，直到图像 2 中找不到新轮廓线为止。

2.3.2 区域分割

1、串行区域分割

串行区域分割技术指采用串行处理的策略通过对目标区域的直接检测来实现图像分割技术。基于区域的串行分割技术有两种基本形式，一种是从单个象素出发，逐渐合并以形成所需的分割区域，称为区域生长。另一种是从全图出发，逐渐分裂切割至所需的分割区域。论文后续内容的实现采用了区域生长技术，这里我们展开介绍。

区域生长的基本思想是将具有相似性质的象素集合起来构成区域。具体先对每个需要分割的区域找一个种子象素作为生长的起点，然后将种子象素周围邻域中与种子象素有相同或相似性质的象素(根据某种事先确定的生长或相似准则来

判定)合并到种子象素所在的区域中。将这些新象素当作新的种子象素继续进行上面的过程，直到再没有满足条件的象素可被包括进来。这样一个区域就长成了。

区域生长的一个关键是选择合适的生长或相似准则，生长准则可以根据不同原则制订，而使用不同的生长准则会影响区域生长的过程。基于区域灰度差的方法主要有如下步骤：

(1)对图像进行逐行扫描，找出尚没有归属的象素；

(2)以该象素为中心检查它的邻域象素，如果灰度差小于预先确定的阈值，将它们合并；

(3)以新合并的象素为中心，返回步骤(4)，检查新象素的邻域，直到区域不能进一步扩张；

(4)返回步骤(1)，继续扫描直到不能发现没有归属的象素，结束整个生长过程。

在采用区域生长方法时，一般新象素所在区域的平均灰度值代替新象素的灰度值与邻域象素的灰度值比较，避免图像存在缓慢变化时不同区域逐步合并时有可能出现的错误。

对一个含 N 个象素的图像区域 R，其均值为

$$m = \frac{1}{N}\sum_{R} f(x,y) \tag{2.23}$$

对象素的比较测试可以用下式表示，其中 T 为阈值：

$$\max_{R}\left|f(x,y)-m\right| < T \tag{2.24}$$

2、并行区域分割

并行区域分割技术指采用并行的方法对目标区域的检测来实现图像分割技术，分割的目的是将感兴趣的区域提取出来。并行区域分割技术在实际应用中有两类：阈值化算法和特征空间聚类[49]。这里重点讨论后者特征空间聚类。

根据特征进行模式分类是指一组目标根据从它们测得的特征值将它们划分到各类中的技术。利用特征空间聚类的方法进行图像分割可看作是对阈值分割概念的推广。它将图像空间中的元素用对应的特征空间点表示，通过将特征空间的点聚集成团，然后再将它们映射回原图像空间以得到分割的结果。

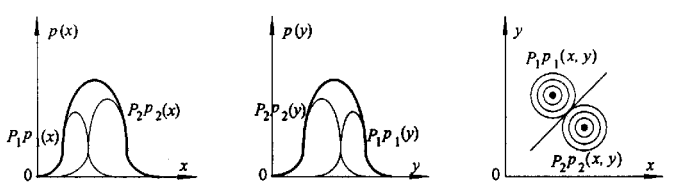


图 2—8 高维空间聚类的优点

特征空间聚类常采用多个特征。在高维特征空间聚类可克服仅用一个特征而不能解决的问题。例如在图 2－8 中，两个类团在两个方向上都有一定的交叠，所以如(a)图或(b)图那样仅用任一个特征都不能将两个类团分开，而(c)图表明在二维特征空间可以方便地分开两个类团。

聚类的方法很多，根据不同的特征进行分类，将象素看作分类的目标点。下面介绍两种常见的聚类方法。

K-均值聚类

将一幅图像分成 K 个区域的一种常用方法是 *K*-均值法。令 $\boldsymbol{x}=(x_1, x_2)$代表一个象素的坐标，$g(x)$代表这个象素的灰度值，*K*-均值法是要最小化如下指标：

$$E=\sum_{i=1}^{K}\sum_{x\in Q_j^{(i)}}\left\|g(x)-\mu_j^{(i+1)}\right\|^2 \tag{2.25}$$

其中 $Q_j^{(i)}$ 代表第 i 次迭代后赋给类 j 的象素集合，μ_j 表示第 j 类的均值。具体的 *K*-均值法步骤如下：

(1)任意选 K 个初始类均值，$\mu_1^{(1)}, \mu_2^{(1)}, \cdots, \mu_K^{(1)}$；

(2)在第 i 次迭代时，根据下述准则将每个象素都赋给 K 类之一($j=1, 2, \cdots, K$，$l=1, 2, \cdots, K$，$j\neq l$)，即：

$$x\in Q^{(i)} \quad \text{如果} \quad \left\|g(x)-\mu_l^{(i)}\right\|<\left\|g(x)-\mu_j^{(i)}\right\| \tag{2.26}$$

即将每个象素赋给均值离它最近的类。

(3)对 $j=1, 2, \cdots, K$，更新类均值 $\mu_j^{(i+1)}$：

$$\mu_j^{(i+1)}=\frac{1}{N}\sum_{x\in Q_j^{(i)}}g(x) \tag{2.27}$$

其中 N_j 是 $Q_j^{(i)}$ 中的象素个数。

(4)如果对所有的 $j=1, 2, \cdots, K$，有 $\mu_j^{(i+1)}=\mu_j^{(i)}$，则算法收敛，结束；否则退回步骤(2)继续下一次迭代。

· ISODATA 聚类

ISODATA 聚类方法一种非分层的聚类方法，其主要步骤如下：

(1)设定 N 个聚类中心位置的初始值；

(2)对每个模式(象素)求取离其最近的聚类中心位置，通过对象素赋值把图像分成 N 个区域；

(3)分别计算属于各聚类模式的平均值；

(4)将最初的聚类中心位置与新的平均值比较，如果相同则停止，如果不同返回步骤(4)继续进行。

第三章 视频对象分割算法

3.1 视频对象分割算法分类

新一代多媒体压缩标准MPEG-4的研究与发展，提出了基于内容的编码方法。VO(Video Object)是MPEG-4的新的核心概念，在VOP是VO在某个时间的存在形式，基于内容的压缩也就是基于VOP的压缩方法。视频对象分割就是研究如何将视频分割成具有语义的多个VOP，即如何从视频中提取有意义的对象。近年来，研究人员在视频对象分割领域开展了大量的研究工作。

视频对象分割技术按照用途来分，可分为面向编码的分割、视频对象操纵和编辑、视频数据库的索引和检索技术、面向监控和事件分析的技术、面向场景理解等几类。本文研究着重侧重于编码目的和视频对象操纵和编辑两方面。

按照人工参与程度可分为自动分割和交互式分割两大类。自动分割是指在不需要用参与的情况下，由计算机算法完成对视频的分割，这种方法通常讲分割问题归结为从静止的背景中分离运动的前景。这种算法不需要人工参与，适合于处理海量的数据，但由于可提供给算法的语义信息不足，分割出的对象比较粗糙。常用在视频编码和数据库索引等领域。而交互式分割是最近新兴的研究热点，算法以用户提供的线索(如对象的边界或者对象所在的区域)为出发点，辅以自动分割算法，通常能取得较好的分割效果，得到的对象的边界比较精确。

按照实现技术可分为基于运动的分割和基于时空相关性的分割。基于运动的方法，通常计算光流场或者用块匹配算法获得运动信息，按照运动模型得到相关参数，利用运动信息的聚类特性将场景分割成多个区域。视频是由一系列时间上相关的图像组成，一方面利用时间上的相关性，同时充分利用空域内的分割算法的优点，主要使用变化检测模板和其它一些空域分割算法，得到视频的最终分割结果。

3.2 基于运动的分割算法

许多算法中用光流法[38]来估计运动向量。运动分割通常把具有相同运动的像素进行聚类，再估计其密度场进行分类[33]。由于运动对象通常与背景有不一致的运动，因此可以从分析对象的运动特征入手来分割视频序列.

基于光流法的分割就是通过研究光流场,从序列图像中近似计算运动场，然后根据运动场的运动特征进行视频分割。光流法是用于估算运动场的一个较普遍的

方法，光流法使用的是与投影位移模型不同的投影速度模型。由于存在孔径问题和遮挡问题，用光流法估算二维运动场的解是不确定的，需要使用附加的假设模型来模拟二维运动场的结构,可用的模型分为参数模型和非参数模型 2 种。

参数模型是描述曲面的三维运动在图像平面上的正交或透视投影。采用基于参数模型的光流法进行分割的基本算法思想是:假设有K个相互独立的运动物体，每一个流矢量对应于单个不透明体的三维刚体运动的投影。基于这一假设,每一个不同的运动可以通过一系列映射参数来正确描述。

Wang 和 Adelson[19]提出了聚类仿射参数来实现运动分割的方法。这个方法中先把帧分成 N 个块，进行运动估计分别求它们的运动参数。然后使用迭代的 K-均值聚类把这些运动参数向量分组得到聚类中心，通过把各个象素分配到最近的聚类中，完成运动分割。这种方法对仿射运动估计中的小的误差很敏感。文献[26]提出一种改进方法。用一种新的区域标记方法来增强运动分割的空域平滑性，使参数运动模型的边界与对象的边界相匹配，利用颜色分割的区域精确性来提高运动分割产生对象的边缘平滑度。算法描述如下：

(1) 估计 t 帧到 t-1 帧的密度运动向量；

(2) 确定初始运动分类数，计算初始的运动参数集；

(3) 经过迭代地像素运动向量匹配得到最优运动参数集；

(4) 利用颜色或灰度信息进行帧内分割得到不连接的区域；

(5) 对颜色区域的运动向量匹配到运动分割中。

相对于非参数模型,参数模型受噪声的影响较小,因为参数是由多个像素结合在一起估算出来的 .但参数模型的缺点是只适用于刚体运动。

典型的非参数模型运动分割的主要有块运动模型和贝叶斯法。块运动模型主要应用在低码率视频编码应用中。基于平移的块运动模型的运动估算虽然简单，但处理逐帧的块旋转和变形时却效果不好，其分割精度由块的大小决定。现在使用空间变换的网格 (mesh)模型成为一个积极的研究领域，网格模型可以很好地应用在旋转和缩放的情况下，但运动估算的复杂度也大大增加了。

贝叶斯法是在给定光流数据的条件下，搜索分割标记的最大后验概率(MAP)，它是检测当前的分割符合被观察的光流数据的程度和当前分割与我们的期望值一致程度的方法。贝叶斯法利用随机平滑度约束条件 ,通常采取 Gibbs 随机场方法来估算位移场。MAP 分割法用分段的二次流场模拟光流数据，用 Gibbs 分布模拟分割场,通过模拟退火搜寻使后验概率最大的标记。

文献[41]提出光流场的估算和分割同时进行的贝叶斯法,把运动虑运动聚类信息,然后由密度流场对每个亮度分割标记计算仿射运动参数,仿射模型不够精确的分割标记被进一步分割成较小的标记。这样，由从单特征的分割标记可以得到多特征的分割标记。在分水岭算法中也同时考虑亮度和运动因素。最后，把相似仿

射运动的区域合并起来完成分割算法。

这种方法克服了早期非迭代光流分割算法孤立运动分割与光流场估所引起的问题。近期的研究表明，光流分割的成功与否与被估算的光流场的正确性紧密相关，反之亦然。

3.3 基于时空域相关的分割算法

用时空域相关性分割视频序列是重要的研究工作，在文献[32][34]中提出了一些算法。虽然可以使用运动信息进行分割，但这种方法有两个重要的缺陷。一是光流方法不能很好地处理快速的运动，二是具有相同运动的区域可能包括多个需要进一步分割的对象。为了克服基于运动的方法的缺点，把空域信息和运动信息结合起来显得尤为重要。一种简单的做法是先把首帧进行空域分割获得初始分割结果，然后利用运动信息进行区域匹配来分割后续帧。但是，当有新的目标进入场景时，这种方法必须处理仿射区域匹配所带来的积累误差。下面我们讨论几种常见的时空域方法。

3.3.1 变化检测模板

有些算法使用变化检测模板来代替运动向量。尽管用变化检测模板计算起来很简单，但这种方法也有缺陷。首先，只有闭塞区域被标记成已变化的，而对象内部却记为未变化的。另外，在某段时间中停止运动的对象或对象的一部分，会被当作背景而丢失。为了解决这些问题，有些算法用缓存记录前几帧对象的状态。但是，缓存的长度过长会造成已露出背景仍被当作对象处理，结果是得到的 VOP 比实际对象大一些。

在文献[36]中，自动分割定义为从静态背景中分离运动前景或物体。在预处理阶段，利用帧差的高阶统计检验，确定可能的前景区域。差分帧中零值数据，可能是噪声或者运动物体。在高斯噪声模型下，噪声是随机的，而运动物体是规则的，所以可以通过高阶统计抑制噪声。对差分帧中的所有的零值象素，计算它们的四阶矩，然后选取适当的阈值，得到运动物体和背景的初始分割图。在接下来的运动分析阶段，计算被标记为变化的区域的像素的位移。如果运动向量起始点都在变化区域内，则像素被标记为前景，否则被标记成背景。最后，对得到的分割模板应用形态学开闭算子，清除物体内部孔洞，得到连续的运动物体。由于没有利用物体灰度边界信息，所以得到的前景比实际物体稍微大一些。

Mech and WollBorn 在文献[20]中，提出了一种从估计的变化检测模板中得到 VOP 的算法。开始先计算连续两帧的帧差，用全局阈值得到变化检测模板。然后，

用局部自适应阈值松弛迭代算法精化对象，提高对象的空间连续性。同时为了避免由于视频对象或部分对象在某时刻停止时，丢失部分对象，利用变化模板缓存器来增强分割的时域稳定性，如果象素在前几帧的模板中，至少有一次被标记成变化的，那么这个象素就标记成变化的。接下了再用形态学算法，得到最终的变化检测模板。通过 CDM 消除背景后，计算帧内边缘图，调整对象模板，提高最终分割的精确性。

变化检测模板方法有时比运动场方法要有用一些。例如，在视频会议应用中，视频多是头肩序列，人的运动比较小，闭塞区域也很小，用 CDM 方法得到的 VOP 比较接近真实对象。而在这种情况下，由于运动较小，估计运动场是比较困难的。

基于 CDM 的时空域方法不需要计算光流场的估算和任何特征点匹配，但是要求背景相对稳定，同时这种算法对图像量度梯度的噪声很敏感，分割精度易受到观测噪声的影响。

3.3.2 形态学方法

形态学是以形态为基础对图象进行分析的数学工具。它的基本思想是用具有一定形态的结构元素去量度和提取图象中对应形状以达到对图像分析和识别的目的。数学形态学的数学基础和所用语言是集合论。应用形态学处理图像，可以简化图像数据，保持它们的基本形状特性，并去处不相干的结构。

在视频分割中，越来越多的用到形态学方法，如水线算法和形态学滤波器。文献[21]中描述了一种典型的形态学算法，它具有以下的步骤：

(1)用修正过的开闭算子进行形态学滤波，消除帧内的亮斑或暗斑，并保持物体的边界；

(2)确定同质区域，同时为区域作标记；

(3)以区域标记为种子，用类似于区域生长的水线算法进行区域分割；

(4)进行分割质量评价，确定区域是否需要继续分割。

以上的算法主要侧重空域分割，而文献[23]中描述了另外一种形态学算法，它在确定同质区域时，同时以亮度梯度和运动场作为相似度准则，同时考虑时域和空域的特性。对每一个要选取的标记点，从光流场中计算仿射运动参数。首先进行全局运动估计和运动补偿。再计算形态学梯度图像，然后用水线算法确定物体边界的位置。得到的区域如果有半数以上的象素在变化检测模板中，则该区域作为前景。接下来对具有相似运动参数的区域进行合并，最后用开闭算子修正边界得到最终结果。

形态学分割算法计算上比较简单，对于具有强噪声的图像有可能取得较好的效果。然而，会因选取不同的评价准则，而存在不完全分割的情况。

3.3.3 其它方法

各种理论基础的分割技术都有自身的优点，同时也存在某些缺陷。出现了混合各种技术的其它方法。文献[33]中提出了一种层次结构的两阶段算法。首先，用哈夫变换把光流场分割成相连的组件，用仿射变换为每个组件建立运动模型。相邻的组件如果具有相似的 8 参数双线性运动模型，就把它们合并成小的元素。接下来，在把具有相同 3D 运动的小元素进行合并，得到最终的合并结果。

文献[24]中提出了一种层次结构的算法。先用变化检测模板把当前帧分成变化的和未变化的区域，把连通的变化区域最为同一个对象处理。对每一个对象从亮度和梯度估算运动参数。如果运动补偿后预测误差过大，就把对象进行再分割，放到层结构的下一层进行分析，直到所有变化区域都得到精确的补偿。由于这些算法交替使用图像分析和综合，所以称这些算法为分析综合算法。

文献[34]中提出了一种图像序列的分层表达方式。用仿射模型的运动参数，对当前帧进行分割。首先估算光流场，然后把帧分成正方形的块。为每个块计算仿射运动的参数。再通过 k 均值聚类算法，把块按运动参数分成不同的层。为每个象素重新指定一个层，选择与该象素处光流运动向量差异最小的层。帧被分成几个具有相同仿射运动的对象的层。再用一个时域的中值滤波得到每一个对象的表达图像。这个算法有以下的缺点：如果视频序列中同一个对象以不同的角度呈现，这个算法就不可能用单幅图像来表示一个对象；对于非刚体运动的对象，不可能用相同的仿射运动参数来表示整个层的运动；整个算法有效性完全取决于光流场估算的精度。

3.4 交互式分割

在基于对象的视频编辑和视频数据索引过程中，需要从视频序列中提取出人感兴趣的语义视频对象，以方便检索或者制作特定的视频节目。语义视频对象是人主观上定义的，用自动的方法通过分析视频的底层特征来提取视频对象是目前是很困难的，所以要得到具有完整语义的视频对象必须借助于人的认知能力，自然就产生了交互式分割的方法。

交互式分割是与自动分割相对的概念，这种方法一般先由用户提供分割线索，再使用相应的自动方法根据分割线索把视频分割成多个视频对象，分割的同时可以为这些对象加入语义信息。交互式分割方案必须定义用户的交互操作和提供能用交互信息的自动分割方法。交互式分割通常用在对分割精度要求较高或者要求得到具有明确语义的非实时性场合。交互式分割的一般做法是用户通过图形用户界面对视频中的关键帧进行初始分割，然后用自动分割算法分割后继帧。根据交

互式分割使用的技术，可以把它们分为三大类方法:对象跟踪算法、变化检测的算法、形态学算法，下面我们就来介绍和比较采用不同技术的各类方法。

首先介绍一种使用边界跟踪算法的交互分割方案。如图 3-1 所示这个算法是MPEG-4 建议的一个交互式分割方案。首先用户先在通过在第一帧中描绘出视频对象的大致边界，然后用水线算法找到第一帧视频对象的真正边界，然后分刚体和非刚体使用不同的算法进行后续帧的分割。

对非刚体对象使用边界跟踪方案。对前一帧对象的每个边界象素都进行运动估计和运动补偿，计算出当前帧中对象的初始边界。然后利用当前帧的边缘图作为空域信息，运动补偿的误差作为时域信息，对得到的初始边界逐象素进行调整，产生最后的视频对象边界。

对刚体对象分割使用基于参数模型的运动跟踪和补偿方案，将刚体对象轮廓从一帧映射到下一帧。由前一帧的轮廓可得到对象的模板，对相继两帧的对象区域进行运动估计，将运动补偿误差过大的点去除。用迭代算法对调整后的对象模板进行六个参数运动估计，得到运动模型参数。然后，对前一帧的轮廓作运动补偿得到视频对象的新轮廓。

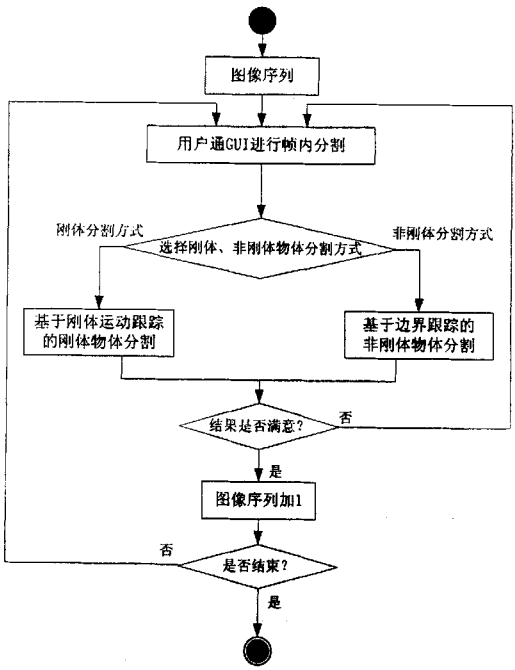


图 3-1MPEG-4 建议的半自动分割方案

变化检测模板也可以用在交互式分割中。文献[9]提出的交互式分割方案，主要使用变化检测模板来实现交互式分割。首先，用户在第一帧图像中用矩形框住对象所在的区域，使分割操作只在用户感兴趣的区域上进行。使用高阶统计的方法，消除噪声引起的亮度变化。接下来，分析多帧的变化检测值来确定当前帧中属于运动对象的像素。最后，用形态学开闭算子对运动对象所在的区域进行边界修整。这个算法的优点是定义用户感兴趣的对象，消除其它运动区域对生成对象的影响，同时分析多帧的变化检测值确定运动象素，消除了由于物体运动不连续而引起变化检测误差。但是它具有计算量较大、边界不够精确的缺点。

文献[30]提出基于形态学工具的交互式分割方法，它充分结合了人在的高层次特征上分割特长和计算机在低层次特征上的分割特长，能取得较好的分割效果。

首先用人工交互手段，描绘将处要分割的视频对象，这些视频对象通常在颜色或运动方面不具有聚类性质。对得到的对象进行进一步分割，将其按颜色、灰度聚类性质分割成不同区域。用块匹配算法估计运动向量，把这些区域通过运动补偿映射到下一帧上。这些映射区域用来作为下一帧进行水线算法所需的分割标记。接下来进行水线算法分割，同时要用变化检测模板作为参考，提高分割在时间上的稳定性，以避免静止区域的标记以溢出到变化区域中。由映射区域和区域分割可得到视频对象的分割。

交互式分割算法在视频编辑和人工视频数据库索引制作中可以取得良好的效果，但是它们共同的缺点是需要人工参与，不能用实时性要求高的环境中。

第四章 时空域分割关键算法

4.1 视频分段

4.1.1 分段原理

数字视频在进行视频对象分割之前，首先要把整个视频序列按照内容分成若干段，最小的、最自然的视频内容单元是镜头。一个镜头是由一个摄像机连续拍摄得到的时间上连续的若干幅图像组成[41]，同时若干镜头又组成一个场景（参见2.1节）。在一个镜头中对象和内容相对稳定和连续，这一点是进行视频对象分割的前提条件。将镜头提取出来和从镜头中提取视频对象，分别是对视频内容进行分析的不同层次。视频分段目的是找出视频对象在时间上都有哪些活动(session)，而视频对象分割目的则是在一个特定的活动中，确定都有哪些对象以及它们的空间位置。由于视频分段是沿时间轴对视频进行分割，所以也被称为视频的时域分割(temporal segmentation)。

镜头到镜头的转换通常有两种方式：直接切割(straight-cut)和光学切割(optical cut)[42]，分别是突变方式和渐变方式。突变指镜头间的突然变化，常在两帧图像间完成。渐变则是从一个镜头缓慢的变化到另一个镜头，通常持续十几和几十帧。

检测镜头的基本思路是寻找视频中相邻两个镜头之间明显的特征差别，计算特征值并利用特征值的不连续性，将镜头切分开来。不同的镜头中包含的内容通常不同，其中对象的像素分布差异比较大，比较相邻帧之间的差异，如果差异超过一定的阈值则认为镜头发生了切换。根据这一事实，本文提出了一种自适应阈值滑动窗口算法检测镜头突变，对视频进行时域分割，下面介绍这种算法。

4.1.2 自适应阈值视频分段算法

对多个视频序列的实验表明，镜头突变对应着帧差的一个突变(如图4-1)，检测镜头突变就是检测帧差的突变位置。视频的帧率一般是24~30fps，一个镜头的直接切割一般在5帧内完成。我们用一个大小为15的窗口，在各个帧差间滑动，如果窗口中有超过阈值T的帧差值，则认为其为可能的切换点。再根据可能点周围的情况，进行进一步判定。

算法描述如下：

（1） 计算相邻两帧之间的帧差。帧差定义为

$$D_l = \frac{1}{N}\sum_{x,y}|f_l(x,y) - f_{l-1}(x,y)|,$$
$$f_l(x,y) = \frac{R(x,y)+G(x,y)+B(x,y)}{3} \tag{4.1}$$

其中 N 是一帧图像中的像素总数。$f_l(x,y)$是则帧图像(x,y)处的灰度值。

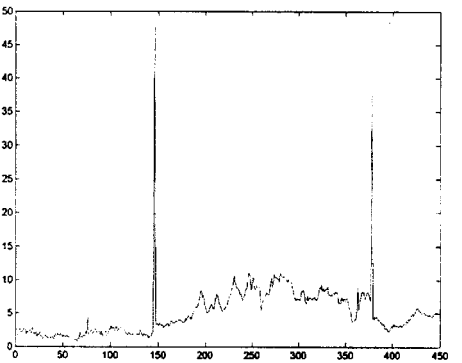


图 4-1 镜头突变在帧差序列中的特征，帧 145 和帧 377 处有镜头的直接切换

（2） 定义一个大小 W 的窗口，这里 W 取为 15，窗口左端位置 L=1；

（3） 窗口按照帧的顺序，在帧差序列上滑动，满足下列条件：

$$D_l > T,$$
$$T = \begin{cases} \alpha\mu_w & ,u_w > \mu \\ \beta\mu_w & ,\text{其它} \end{cases}, \tag{4.2}$$
$$D_l = \arg\max_{k\in W} D_k$$

其中，μ_w为窗口内帧差均值，μ为整个序列的全局均值，α,β为阈值系数，根据经验，值分别取为 2 和 5，转(4)，否则，L=L+15,转(3)；

（4） 将窗口滑动到以 l 为中心的位置，应用条件(4.2),得到$D_{l'}$,若条件得到满足，把l'加入集合 Cuts, L=l'+5;否则，L=L+15,转(3)。

判断镜头检测算法的效果目前主要有查全率(recall)和准确率(precision)两个指标来衡量[43]。设正确检测数为 N_c、漏检数为 N_m和误检数 N_f,查全率和准确率分别用(4.3)和(4.4)来表示。

$$R = \frac{N_c}{N_c + N_m} \tag{4.3}$$

$$P = \frac{N_c}{N_c + N_f} \tag{4.4}$$

	正确检测数	漏检数	错检数	查全率	准确率
故事片	20	0	0	100%	100%
MTV	32	4	1	88%	96%

表 4-1 镜头检测结果

应用上述算法，我们对有较多长镜头故事片和有较多短镜头的 MTV 其中的 3000 帧进行了分割，取得了较好的效果。效果如表 4-1 所示，MTV 中编辑特效较多，渐变切换镜头较多，故查全率降低。

4.2 颜色空间选取

在构建时空域数据以前，应选择合适的颜色空间，这是视频处理和分析要处理的第一个任务。因为颜色相似度和距离函数等的选择都要依赖于颜色空间的选择，而这直接影响着算法的效能。常见的颜色空间有 RGB、YUV 和 HSV 等。

RGB 彩色模型中，颜色由红、绿、篮三原色的强度合成。RGB 强调从物理上反映颜色的特性，如航天和卫星多光谱图像，强度图像由工作于不同光谱范围的图像传感器得到，进行合成是具有物理意义的。由于这种物理特性，使得大多数获取数字图像的彩色摄像机和显示图像监视器都使用 RGB 颜色格式。

为了在视频信号传输时，减少所需带宽并与单色电视系统兼容，定义了 YUV 颜色空间，Y 分量来确定图像的亮度，U 和 V 两个分量反映色度值。黑白电视接收端接到彩色电视信号，只处理 Y 分量，即可做到黑白电视兼容彩色信号。虽然 YUV 空间中定义了色度，但 U 和 V 两个分量正比于色差 B-Y 和 R-Y,他们不能直接反映和适应人类视觉的主观颜色。

HSV 色彩空间由分别代表色调(Hue)、色饱和度(Saturation)、亮度(Value)的成分组成。HSV 色彩模型可由六角模型表示出来，如图 4-2 所示。色度可由绕垂直轴的旋转角度定出。从红色 0 开始，在 0~360 范围之间的值，它反映颜色的光谱组成。饱和度是在 0~1 之间变化的，反映某种光的主波中纯光的比例。亮度(Value)也在 0~1 之间变化，是一个相对亮度。当 V=0 时，这个点对应于黑色。而这时的色度、饱和度是无定义和没有意义的。当 R=G=B 时，沿着 V 轴的任意一点的饱和度为 0，色度没有意义。由 RGB 空间到 HSV 空间转换公式如式(3.5)所示。

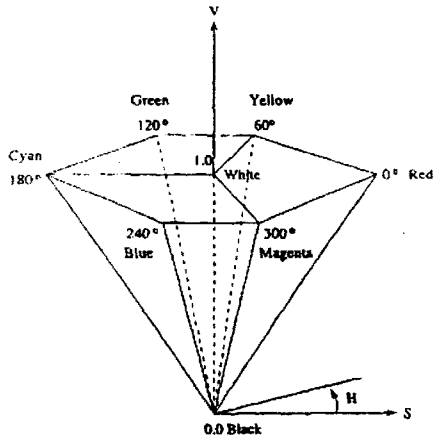


图 4-2 HSV 空间色彩模型

$$H=\begin{cases}\dfrac{\theta}{2\pi}, B\le G\\ \dfrac{2\pi-\theta}{2\pi}, B>G\end{cases}, S=\frac{Max(R,G,B)-Min(R,G,B)}{Max(R,G,B)}, V=\frac{Max(R,G,B)}{255} \tag{4.5}$$

$$\text{其中}\theta=\arccos\frac{(R-G)+(R-B)}{2\sqrt{(R-G)^2+(R-B)(G-B)}}$$

HSV 空间的特点是：去掉了强度(Intensity)成分与颜色信息的联系；色调和饱和度分量与人类获取和理解颜色的方式相近。本文提出的算法，选用 HSV 空间进行颜色分析。主要考虑到以下几点：

1. 选择颜色空间的一个重要因素是在选定空间中计算颜色距离的复杂程度。区域生长算法中，频繁的计算象素之间的颜色距离。复杂的距离计算方法将导致算法的性能严重下降。
2. 对于许多颜色相关的视觉应用不能表达颜色主观颜色如橙色、青色，这是 RGB 颜色空间的一个重要缺陷。而基于对象的分割中，同质区域的颜色相似性应主要由色调来决定。
3. 通常场景中对象在运动时，其表面亮度会发生变化，要求亮度不敏感的颜色距离来量度区域的同质性。HSV 颜色空间的 S 和 H 分量可以满足这一点。
4. HSV 色彩空间在视觉上是均匀的，与人类的颜色视觉由很好的一致性，经常用于肤色分割，对于人脸、皮肤等对象组件很容易进行聚

类。

5. HSV 空间也有不足之处，由于多数视频是以 RGB 格式存储的，所以处理颜色前要从 RGB 空间转换到 HSV 空间，而这一转换涉及到三角函数计算，比较费时。但是为了进行准确地分析，这个预处理工作是值得的。

4.3 空域滤波算法

4.3.1 滤波算法对比

在进行视频分割之前，一般先要对原始视频图像进行空域滤波。滤波有两个主要目的：去处噪声和平滑颜色组件，以避免生长算法过度分割。过度分割有很多缺点，它会降低算法的速度，增加区域的数量从而增加内存的需求量。更重要的是会把具有轻微纹理的图像，分成一个个小的区域。因此，滤波可以提高分割的速度和稳定性。尽管可以用低通滤波、中值滤波和形态学滤波等除去图像中的噪音，但这些滤波算法会或多或少破坏图像的边缘结构。下面我们看看这些滤波算法。

高斯滤波及特点

高斯滤波器时一个基本的低通滤波器，可以用来去处图像中的细节和噪声。它是一个在时域和频域都具有平滑性能的低通滤波器。它的卷积核是

$$g(x,y)=\frac{1}{2\pi\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{4.6}$$

高斯滤波器输出的像素和相邻象素的加权均值，中央的像素占的权重较大。因此它比相同大小的均值滤波器有更好的平滑性能和边缘保持性能。然而，高斯滤波器对椒盐噪声只有抑制作用，而且会削弱高频细节。另外的缺陷是它会模糊物体的边缘。处理效果见图 4-3(a2)和(b2)。

形态学滤波

形态学滤波是一种非线性滤波算法。它的特点是可以选择性的移除图像结构，这种选择性由结构元素 A 来决定。膨胀和腐蚀是灰度形态学最基本的两种运算。分别定义如下

$$D(I,A)=I\oplus A=\max_{i,j\in A}\{I(x-i,y-j)+A(i,j)\} \tag{4.7}$$

$$E(I,A)=I\ominus A=\min_{i,j\in A}\{I(x-i,y-j)-A(i,j)\} \tag{4.8}$$

一种常见的滤波算法是使用混合滤波器[28]，定义为

$$HYB(f)=[(f\Theta g)+(f\oplus g)]/2 \tag{4.9}$$

在这种。虽然它可以很好的平滑图像和保留物体的边缘信息，但是其对噪声的抑制功能是有限的。处理效果见图 4-3(a3)和(b3)。

图 4-3 滤波效果对比图

中值滤波

中值滤波是一种非线性滤波算法，在某些条件下可以做到既去处噪声由保护图像边缘。他对椒盐噪声和高斯白噪声都有很好的效果。滤波窗口为 A 的得二维中值滤波可定义为

$$y_{ij} = \underset{A}{Med}\{x_{ij}\} = Med\{x_{(i+r),(j+s)}, (r,s) \in A, (i,j) \in I^2\} \tag{4.10}$$

中值滤波可以有效的去除脉冲型噪声，而且对图像的边缘由较好的保护。但它也有固有的缺陷，如果使用不当，会损失很多图像细节。

只有保持较好的边界，在进行区域生长时才不会超出物体的边界。高斯滤波计算上比较复杂，而且会引起图像的边缘被破坏。形态学去噪的效果，取决于对结构元素的选择。它可以很好的平滑图像，但缺陷是如果算法反复进行，就会破坏物体的边界。中值滤波算子可以很好的除去椒盐噪声同时保持边缘，但用小窗口不能除去纹理。在我们的算法中使用自己设计的一种加权中值滤波算法，下面介绍这种算法。

4.3.2 加权中值滤波

通常的中值滤波算法，窗口中的元素输出的机会是均等的，为了更好的保留阶跃边缘信息，用权值使窗口中的元素输出概率不等。具体做法是改变窗口中的变量的个数，可以使一个以上的变量等于同一点的值，然后扩张后的数字集中求中值，如图 4-4(b)中数字为该位置数据扩充后的个数。对于图 4-4(c)所示数据区用不同的中值滤波输出值不同。普通的中值滤波输出 $Med\{1,2,5,2,5,5,1,5,2\} = 2$，而加权中值滤波输出为

$Weight\_Med\{1,2,5,2,5,5,1,5,2\} = Med\{1,2,2,5,2,2,5,5,5,5,5,1,5,5,2\} = 5$。对于这个数据区来说 5 是阶跃值，应该予以保留，这一点可以显示加权中值滤波的优越性。为了加快算法的速度，同时优化滤波算法的实现过程，没有对所有元素排序，而是值排序了一部分元素，就找到了输出值。算法流程见图 4-5。处理效果见图 4-3(a4)和(b4)。

1	1	1
1	1	1
1	1	1

(a)

1	2	1
2	3	2
1	2	1

(b)

1	2	5
2	5	5
1	5	2

(c)

图 4-4 (a)普通中值滤波的权值(b)带权值的模板(c)模板覆盖的数据

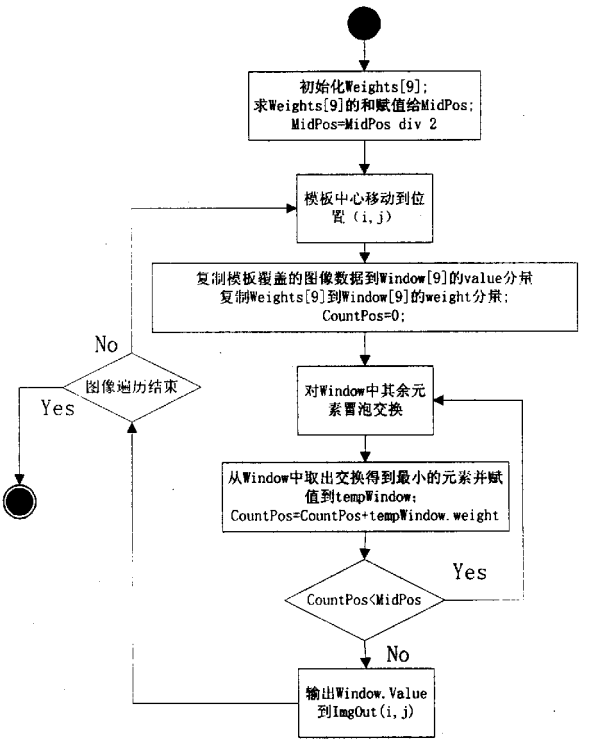


图 4-5 加权中值滤波算法流程图

4.4 时空域生长

4.4.1 3D 区域生长

在图像的空域分割算法中，区域生长算法是一类重要的串行区域分割算法，区域分割得到的区域具有空间连续性好和边界的精确性高的特点。三维区域生长较多的应用在三维医学影像处理中，它基于三维位置空间信息分布的连续性。而在分割序列图像时所说的 3D 区域生长，是指在二维图像区域生长算法的基础之上引入在时域上方向增长，是在二维区域生长的基础上发展而来的。

数字视频是一个图像序列 E(x, y, t) (参见图 2-1)，镜头所摄取的场景中运动物体在成像平面上的投影的时间采样，构成了连续的视频图像帧。由于物体运动的连续性，使得运动物体的投影在成像平面上的空间位置上有了强的相关性，时

域方向的生长正是以此为基础进行的。我们对象的立体轨迹称为元素(Volume),记作 V_i，而元素在每一帧上的截面就是我们通常说的二维区域(Region)，记作 R_i^t。可以认为元素在整个 x-y-t 空间中是连续的，即属于同一物体的帧间区域具有同质性。

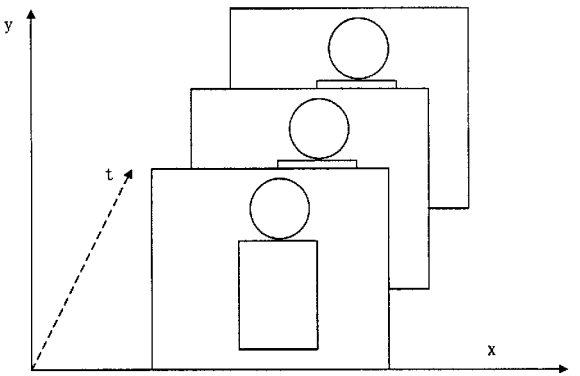


图 4-6 物体运动轨迹在各帧上的投影具有同质特性

三维区域生长法也可以称为时空域生长，它有以下优点：

（1） 同时对多个帧进行分割；

（2） 能很好的解决同一镜头中对象的出现和消失带来的问题；

（3） 从元素中可以获得元素所对应物体的平移运动信息；

（4） 充分利用了时域和空域信息

三维区域生长的基本思想同二维区域生长的基本思想是一致的。首先，从种子象素开始生长，如果相邻象素符合该元素的相似性准则，则把它加入该元素。将这些新加入的象素当做新的种子象素继续进行上面的过程，直到再没有满足条件的像素可以被包括进来。这样就形成了具有相似性质的象素集合起来构成的元素。不象二维图像中考虑的 4 邻域和 8 邻域，这里要考虑三个维度上的 8 个邻域或者 26 个邻域的信息。

与二维区域生长相同的是三维区域生长也要考虑三个问题：如何选取种子象素；象素与元素的相似性准则是什么；何时停止生长。下面的几节就来讨论这些问题。

4.4.2 种子像素选取

对所有的区域生长算法来说，种子象素的选取都是关键的第一步。种子象素直接

影响着区域生长的效果和效率，如果太多的种子不在感兴趣的区域中，生长的效率就很低；如果选在噪声点上，周围的点很可能就不会被包含到其它区域中了。常见的种子选取方法有以下几种：

- 均匀分布种子：没有先验知识时，将种子均匀分布在图像上。
- 聚类中心种子：常用在有先验知识的场合，事先知道要分割区域的个数，再根据区域特征计算出聚类中心，作为生长起点，进行区域生长完成分割；
- 梯度极小化种子：种子最好选在区域的中心位置，而一般来说区域边界的梯度较大，而内部的梯度较小，所以选择梯度极小的地方一般较好；
- 最亮点种子：常用于红外图像的检测，因为最亮点对应物体温度最高的部分。

在进行视频对象分割时，一般没有先验知识，所以一般采取均匀分布种子的办法，但是为了使区域生长效果更好，本文将多种原则结合在一起，提出了一种同时结合图像的边缘信息和时空域梯度信息的综合选种子方法。首先把帧分成边长为 L(通常 L=20)的块。分析块中的图像边缘特征决定，在块中填入 0 个 1 个或 2 个种子。然后用同样的方法在其它帧填入种子。可以用算法描述如下：

(1)用 Canny 算子得到第 i 帧的边缘模板;

(2)均匀的把当前帧分成 n 块;

(3)对第 j 块进行分析,若 j>n, i=i+1 转(1);

(4)若边缘模板中边缘点总数超过块大小的三分之一，则 j=j+1 转(3)；否则，转(5);

(5)若没有边缘点，则把块中最小梯度点定为种子点，j=j+1 转(3);否则，转(6);

(6)在块中边缘的两侧各选一个最小梯度点定为种子点，j=j+1 转(3).

上述算法中，给不同的块中填入不同个数的种子的原因在于区域生长到区域边缘处必须停止。边缘较多的块一般是细节较多的地方如人脸、头发、衣服的褶皱、细纹理区域，经常位于物体内部，对于视频分割来说，这些组件最好能被周围的平滑区域合并，如果在这里种种子，会形成很多小的区域，降低算法的效率。而把这些点留到后处理阶段由大区域合并或者周围块生长过程中合并，对于抽取平滑的同质区域轮廓有利。若边缘点较少，可能是平滑区之间的边界，投入两个种子使其分别向边缘两侧生长。

选好种子之后，就可以开始生长过程了。在这之前还应该考虑一个问题，那就是生长顺序问题。先生长的种子，一般成长性较好，更可能形成较大区域。所以在决定种子的生长顺序时，应该让我们所感兴趣的区域(ROI)中的种子优先。对于大多数的视频来说，景物和人物位于帧的中央位置，而对一个镜头或者视频序列(Video Session)来说，具有代表性的关键帧也常常在序列中处于中间位置。考虑到这一点，本文算法在遍历种子时，采取时间上空间上都由中间向两侧的顺序，一层一层向外遍历种子。本文称这种遍历方法称为“卷心菜”(Cabbage)方法。这

种遍历方法的伪代码见附录。

4.4.3 相似性准则

确定好种子像素，就要开始元素的增长了。在元素增长过程中，不断地把与元素相邻的象素合并到元素中来。假设我们从单点 p 开始，希望从种子点扩展成一个连续的元素。这就要先定义一种距离 d(p, V),它在 p 和元素 V 具有相似性时值很小，反之，则很大。如果像素 p 与元素 V 的相邻，且距离 d(p, V)< ε ,可以把 p 点合并到元素中。反复进行这个过程，直到 V 再也没有相邻的像素可以合并进来，元素 V 就生长好了。

在整个生长过程中对特征距离的定义直接影响着生长的结果。特征相似通常使用几何模型，将图像的特征看作是坐标空间中的点，两个点的接近程度通常用它们之间的距离表示，即它们之间的不相似程度。关于距离度量函数的定义通常满足距离公理的自相似性、最小性、对称性、三角不等式等条件。常用的距离定义方法有：

(1)平方距离

$$d(X,Y)=(X-Y)W^{-1}(X-Y)^T \tag{4.11}$$

(2)余弦相关

$$\cos\theta=\sum_{i=1}^{n}x_iy_i/\sqrt{\sum_{i=1}^{n}x_i^2\sum_{i=1}^{n}y_i^2} \tag{4.12}$$

(3)Minkowski 距离度量

$$d(X,Y)=(\sum_{i=1}^{n}|x_i-y_i|^r)^{\frac{1}{r}} \tag{4.13}$$

r=1 时，称为 Manhattan 距离，也叫 city-block 距离。

r=2 时，称为 Euclidean 距离，这是人们通常所说的距离概念。

r=∞时，称为 Chebyshev 距离，此时的表达式为：

$$d(X,Y)=\max_{1\le i\le n}|x_i-y_i| \tag{4.14}$$

在本文的算法中，使用改进的分段 Manhattan 距离。如式(4-15)所示：

$$d(p,V)=\begin{cases}0.1*|H_p-H_v|+0.3*|S_p-S_v|+0.6*|V_p-S_v|, & S_v<0.5\\ 0.6*|H_p-H_v|+0.3*|S_p-S_v|+0.1*|V_p-S_v|, & S_v>=0.5\end{cases} \tag{4.15}$$

H_p,S_p,V_p是元素V相邻点p的颜色分量，H_v,S_v,V_v是元素V内部颜色均值

这是同时考虑到两方面的问题，在生长过程中，每加入一个像素就要计算一

次像素到元素的距离。高次的距离计算方法将大大降低算法的速度，所以采用街区距计算方法效率比较高。另外为了充分利用视频图像中的彩色信息，我们为不同的颜色分量施以不同的权值。在饱和度较高的情况下，颜色较纯主要由色度分量来计算颜色距离，饱和度低的情况下主要由亮度来计算颜色距离。

选定了距离计算的方法，接下来就要确定阈值 ε。阈值最简单的方式是采用常量阈值，但这种方法通用性差，对自动分割算法来说，显然是不合适的。本文设计了一种全局阈值 ε_f 和局部阈值 ε_v 相结合的自适应阈值方法。

用颜色直方图为每一帧确定全局阈值 ε_f。直方图的方差和均值如公式(4.16)表示。方差显示颜色直方图的特征，高的方差表示多种颜色分布，可能是高的空间纹理和多个区域。小的方差则显示出颜色平滑分布或者较少的区域。一般地，图像中区域数量越多，颜色方差越大。如果方差较大，则说明区域较多，应选一个较小的阈值。因此，方差和阈值是相互影响的。颜色的动态范围μ(参见式 4.17)是影响阈值的另外的一个参数。在颜色聚类数目不变的情况下，当颜色动态范围变化时，各聚类中心的距离就会变化。因此，阈值应该与动态范围有关。人的视觉对小的颜色变化不敏感，一般可取可察觉的最小值作为阈值。

$$\begin{aligned}\eta_k &= \frac{1}{L}\sum_l h_k(l)\\ \sigma_k^2 &= \frac{1}{L}\sum_l (h_k(l)-\eta_k)^2\end{aligned}, L=256, 0<l<L, \tag{4.16}$$

$$\mu = c_2 - c_1, \int_0^{c_1} h_k(l)dl = \int_{c_2}^{1} h_k(l)dl = 0.05 \tag{4.17}$$

根据以上所述，全局阈值分为色度阈值 ε_{fH} 和亮度优先阈值 ε_{fV}，分别用于饱和度高和饱和度低两种情况计算距离 d(p, V)。

$$\begin{aligned}\varepsilon_{f_H} &= \lambda_h\varepsilon_h + \lambda_m\varepsilon_s + \lambda_l\varepsilon_v \quad ,\varepsilon_k = \max(\frac{u_k}{4\sigma_k^2},10), k=h,s,v;\\ \varepsilon_{f_V} &= \lambda_l\varepsilon_h + \lambda_m\varepsilon_s + \lambda_h\varepsilon_v \qquad \lambda_h=0.6, \lambda_m=0.3, \lambda_l=0.1\end{aligned}; \tag{4.18}$$

局部阈值 ε_v 是由正在生长的元素确定的。在元素的生长过程中，开始时使用较“松”的全局阈值，随着元素膨胀元素的特性也越来越稳定，就要选用一个适应当前元素的局部阈值，这个阈值比较“紧”，可以在元素膨胀到区域边界时，将元素约束住，不使其越过边界生长。这里取元素的颜色方差作为局部阈值 ε_v。

4.4.4 生长后处理

生长阶段结束后，序列图象被分成多个同质元素。这些元素中有些很小可以

忽略或者比较狭长。而且，还可能存在没有包含到任何元素中的边缘点和孤立点。如果这些种类的元素或者点继续存在的话，会占用更多的内存，影响对象抽取过程的效率，所以要对他们进行进一步处理，应该把他们合并到其它较大的元素或者合并他们成为一个较大的元素。可以把这些元素分为下面几类处理：

(1)不属于任何元素的孤立点。这些点或者是噪音点或者是由于种子位置没选好而形成的。对于这些点，直接根据其空间连通性生成新的元素。

(2)狭长的元素。通常是由物体边缘生长而成，处理办法是根据其与相邻元素的颜色距离、公共边界长度、体积对比等和相邻元素进行合并。

(3)体积较小元素。这类元素通常是物体内细节较多部分或者比较小背景物体。如果是前一种情况的话，则在其周围一定存在很多类似的小元素，可以直接合并这些小元素，否则该元素应该保留。

(4)微小的孤立元素。可以直接合并到与其相邻的颜色距离最小的元素中。

在合并过程中要注意一点的是合并得到的元素应该有紧凑的形状，以避免引起分割不足还要再次分裂的问题。通常用紧凑度(compactness)作为元素的形状约束。在进行每一次合并时，在满足其它条件的情况下，还必须满足合并后的紧凑度大于某个阈值。

紧凑度在二维区域中可定义为

$$C=\frac{R}{B^2}，其中\ R\ 使区域的面积，B\ 是区域的边界的长度 \tag{4.19}$$

在这里紧凑度定义为

$$C=\frac{1}{c_f}\sum\frac{R(f)}{B(f)^2} \tag{4.20}$$

其中 c_f 是元素在时域方向的长度，也称生命期，R(f)和 B(f)分别是元素在 f 帧上截面的面积和区域边长。

二维图形中圆的紧凑度最大值为 $1/(4\pi)$，三维图形中球体紧凑度最大值也为 $1/(4\pi)$，所以在合并区域时密集度的阈值取为 $1/(8\pi)\approx0.04$。而对于狭长元素的判断阈值是 0.02，若元素的紧凑度小于这一阈值则认为它是一个狭长元素。

后处理的过程总体上描述如下：先对孤立点进行无阈值的区域生长。多次遍历所有元素，分别处理所有的微小孤立元素，再进行较小元素的合并，接下来进行狭长元素的合并。

4.4.5 时空域数据

从以上对三维区域生长算法所涉及问题的讨论，可以看出要成功的高效实施3D区域生长算法，要以合理的时空域数据结构为基础，在这一节将介绍时空域数据结构。

在三维区域生长算法中，除了要使用转换到HSV空间的数据外，还要构建其它时空域数据。在整个算法中，元素是最核心的概念，也是计算、统计和存储的基本单位。元素是由元素的特征信息、形态信息和运动信息等组成。对于元素的结构的描述见图 4-7。另外为了提高算法效率，还要使用空间占有数组来缓存元素之间的空间位置关系。

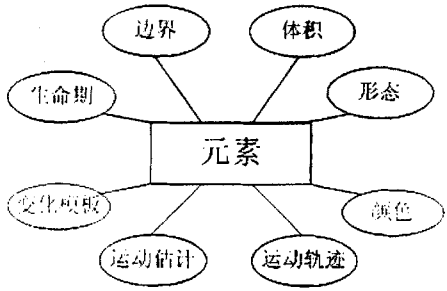


图 4-7 元素结构描述图

下面介绍一下元素的各种属性的计算方法和存储方式：

- 颜色信息：反映元素的颜色特性，包括各颜色通道(HSV)的颜色均值、方差，并在元素的生长过程中不断的更新，主要用它来衡量元素的同质特征,点与元素之间和元素之间的相似度。
- 体积：反映元素中包含的像素的多少，在元素生长阶段不断的更新，在生长和后处理阶段，用来作为阈值选择和合并的参考标准。同时它还记录了元素在各个帧的截面中面积。用大小为SL的数组存储，其中SL是要分割的视频序列(VS)的长度。
- 形态：反映元素在空间的形态，用空间紧凑度(Compactness)来表示。定义和计算方法(参见式4.20)。用来作为判别狭长性元素的标准，在后处理阶段计算。
- 边界：在后处理阶段经常会访问元素的边界信息，进行相邻元素的判定。在对元素进行统计分析时，还要对元素进行遍历，这些都要求有合理的边界存储方式。本文的做法是存储元素在各帧投影区域的边界点，具体结构如图4-8所示。VEdge代表整个元素的边界，REdge代表投影区域的边界，每个分量指向存储边界点对的链表。这样进行元素遍历、投影区域的遍历和非连续的

边界遍历，得到统计数据和分析元素之间关系就变的很方便。

- 生命期：反映元素在时域方向的大小，元素分布在多少帧中，用 VL 分量表示，VL≤SL。
- 变化检测值：根据每一帧的变化检测模板，计算落在元素中的发生变化的点的总数。用长度为 VL 数组来存储。用来作为运动前景和背景分离的参考数据。
- 运动估计：元素在各帧的截面所在的区域进行运动估计。主要在对象抽取阶段使用。用来作为运动对象聚类的依据，在对象抽取阶段进行估算。
- 运动轨迹：元素在各图像帧投影的区域的中心之间的位移差。用来作为运动对象聚类的依据。
- 空间位置：反映元素在三维数据中的位置分布，空间占有数组来存储。所谓空间占有数组，就是一个三维数组，数组每个值记录元素的标号，使用这种结构虽然空间耗费较大，但是能直接描述元素的三维信息，访问速度快，在生长阶段非常有用。

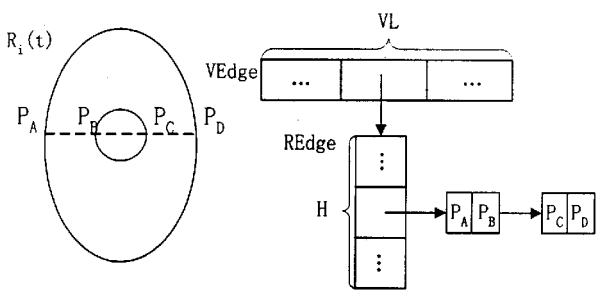


图 4-8 元素边界存储结构

4.4.6 生长过程

处理完 3D 区域生长着重解决的几个关键问题，我们对整个 3D 区域生长的过程做个总结。首先，用卷心菜遍历方法，依次遍历视频序列中的各个块，对于每一个块，按照前述的方法选取种子像素，并把他们的空间位置(f, r, c)存放到链表 Seeds 中。依次取 Seeds 中的空间位置。从区域标记模板 MaskCube 读元素的标记模板，判断种子是否有效，如果种子已经被其它元素占据，则跳过。在 MaskCube 中标记种子作为一个新的元素 V，把种子放进队列 QueNew 中，开始一个新的生长过程。从 QueNew 取出一个位置，探测这个位置的 8 个邻域像素，如果像素满

足相似度准则,则把这个位置加入 QueNew，并 MaskCube 中标记该位置为 V，重复这个过程，直到 QueNew 为空。遍历完所有的种子后，开始进行后处理，合并小的区域、孤立点和带状区域到相邻的区域中，整个生长过程就可以结束了。图 4-9 是整个过程的流程图。

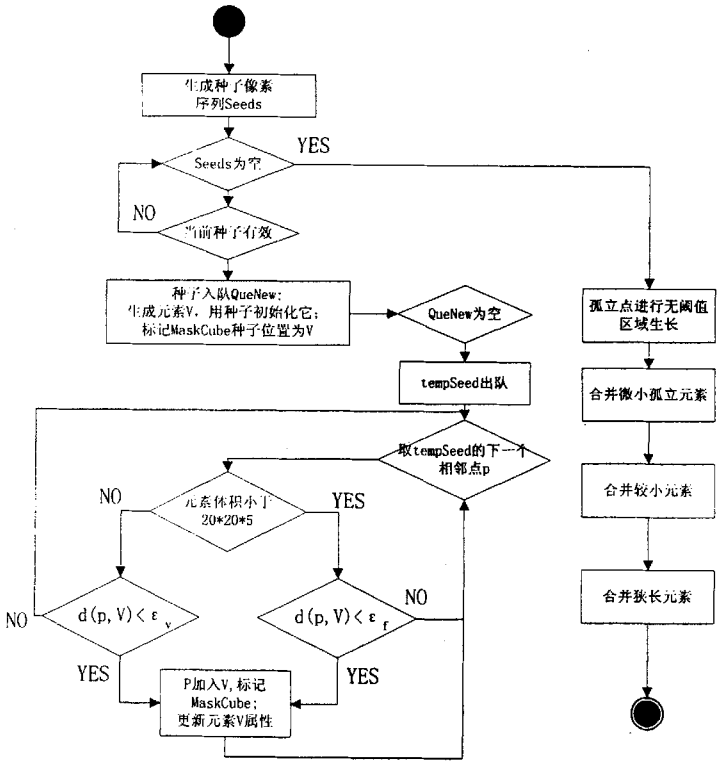


图 4-9 3D 区域生长算法流程图

4.5 视频对象抽取

4.5.1 运动分析

时空域生长输出元素形式颜色视频同质组件，但是多数视频对象如汽车，虽然有相似的运动，很可能包含不同的颜色特征，需要分析元素的运动特征，用运

动信息合成具有运动一致性的视频对象。基于时空域生长的方法有一个优点就是在简单的元素增长过程中，获得重要的平移运动信息，从而避免块运动分析所需的费时计算。

分析元素的运动轨迹可获得元素的平移运动信息。运动轨迹是与运动区域联系在一起的高层特征，是描述元素帧投影的运动特征的量，定义为各投影质心位置 $T_i(t)$的集合。一般用投影的中心来代替质心，因为对于刚体来说运动中心和质心是重合的。在这样的假定下，每一个元素 V_i,运动轨迹 $T_i(t)=[X_i(t),Y_i(t)]^T$ 可以通过平均元素在帧上的坐标计算(4.21)。

$$T_i(t)=\begin{bmatrix}X_i(t)\\Y_i(t)\end{bmatrix}=\begin{bmatrix}\frac{1}{R(i,t)}\sum x\\\frac{1}{R(i,t)}\sum y\end{bmatrix};(x,y)\in R_i' \tag{4.21}$$

区域 R_i' 是元素 V_i 在帧 t 上的投影。R(i,t)是区域 R_i' 的面积。

这种情况下区域的运动向量可以定义为

$$MV_{V_i}(t)=\begin{bmatrix}d_x(t)\\d_y(t)\end{bmatrix}=\begin{bmatrix}X_i(t+1)-X_i(t)\\Y_i(t+1)-Y_i(t)\end{bmatrix} \tag{4.22}$$

在多数情况下，运动轨迹大致可以反映元素的平移运动。这种运动很容易被人类视觉察觉，因此它对识别对象很重要。另外，平移运动是可精确估计的参数运动的一部分。利用平移运动的运动分析器一般比只使用旋转运动的分析器健壮性更好些。

用运动轨迹征帧之间的区域运动，避免了稠密的运动向量的计算,但是运动轨迹缺点在于当对象的运动引起了投影形状变化时，运动轨迹受到的干扰较大，因此需要用运动估计方法，对区域运动信息修正。光流的方法计算量很大，本文采用特征点运动块匹配算法对运动轨迹进行修正。

首先选择特征点。区域内部一般特征比较均匀，不适合作为特征点，而在区域的边缘对象的运动特征较为明显，因此本文在区域的边界内侧选择特征点。首先用锚定区域的中心位置作为原点，水平方向垂直方向作为 x, y 轴方向，建立坐标系。然后落在四个象限内的区域外边界上分别搜索变化检测值较大的 3-5 个点.以这些点为中心建立 10×10 的小窗口，使用最小平均绝对差值函数(MAD)准则(参见 2.2.4)，用 EBMA 算法进行搜索范围为±R 的块匹配。其中

$$R=\min\left\{\max\left[d_x(t),d_y(t)\right]\ ,\ 10\right\} \tag{4.23}$$

计算得到各块的运动向量 MV_i。并把它们与 MV_T 均值，作为新的平移运动向量。

4.5.2 空间聚类

分析得到各个元素的运动信息后，就可以开始进行运动对象的抽取。一个场景可以分为运动前景和静态的背景，运动前景中多个运动对象。完成运动对象的分割一般可以采用两种方法。计算整个场景的运动场直接对运动向量进行聚类，然后进行区域连通的后处理，根据运动向量不同讲运动前景分离出来。另外一种，先通过帧差进行初始分割分出可能的前景区域，然后分析前景中运动向量，进一步分割出运动前景物体。本文采用后一种方法，因为在时空域生长过程中，很容易得到区域信息和区域的变化检测值，给前背景初始分割提供了足够的信息。

首先根据变化检测模板把元素分成运动的和静止的。运动物体的的变化检测值一般出现在其边界上。统计每一帧变化检测模板落在各元素中的点数。如果总点数超过元素边界长度的三分之二，则认为元素是运动的，否则认为它是静止的，这样就以把可能的前景从背景中分离出来。从这可以看出变化检测模板的准确性，直接影响着前背景的分离效果。在变化检测模板计算上，本文做了一些研究。分析了大量数据基础统计变化值的分布，发现变化点总可以分为两类一类接近零并且不同值数量变化剧烈可认为是背景和噪音，另一类变化值较高，各值的数量分布比较平坦。求背景变化值和前景变化值的分界点，得到二值变化检测模板是很关键。由于较多的背景变化小于 10，我们求取超过 10 的点的质心位置，并求得质心与 10 的中心位置作为变化检测的阈值(4.24)，得到二值变检测模板。

$$\gamma = \frac{10 + C}{2}, C = \sum_{k=10}^{255} k * P(k) \tag{4.24}$$

其中，γ 是模板阈值，P(k)是变化值为 k 的点的个数。

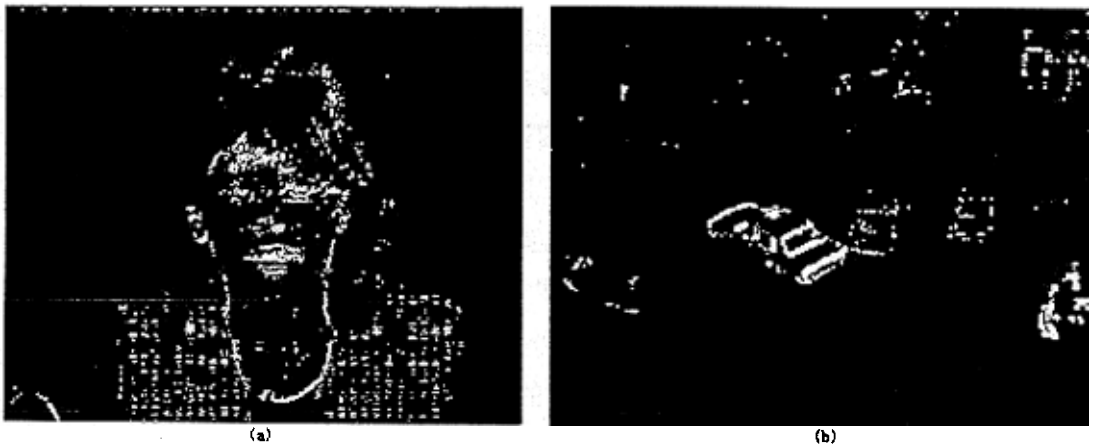


图 4-10 用质心阈值得到的二值变化检测模板

将元素分成了前景和背景两类，接下来在前景分割成不同的运动物体。首先，进行前景区域运动分析，得到各区域的运动向量 $MV_1, MV_2, \ldots, MV_n$.对这些运动向

量进行空间聚类，这里基于同一物体的运动是相似的这一假定。另一方面，仅用运动向量作为特征进行聚类，各个物体的特征平面可能存在交叠情况，这是因为不同物体中的组件可能有相似的运动。同一物体的组件距离较近，因此把区域之间的距离作为一个特征引入。定义出二维聚类空间的区域之间距离量度下式：

$$d_{AB} = \sqrt{\alpha(d_{xA} - d_{xB})^2 + \alpha(d_{yA} - d_{yB})^2 + \beta d_R^2} \tag{4.25}$$

d_R 是区域 A 和区域 B 之间的距离，定义为区域之间的最近两点的距离，d_x, d_y 分别为运动向量的分量，这些距离计算时先要作归一化处理。α 和 β 分别是两种特征的权值，通过调节它们可以决定是运动优先还是距离优先。

完成聚类空间距离量度的定义，就可以开始运动区域的空间聚类。我们使用 K 均值聚类算法(参见 2.3.2)，完成运动区域的聚类。这里对区域 A 到聚类中心 μ 距离定义为

$$d(A, \mu_j) = \sqrt{\alpha(d_{Ax} - u_{jx})^2 + \alpha(d_{Ay} - u_{jy})^2 + \beta d_{Rnew}^2} \tag{4.26}$$

其中 μ_{jx}、μ_{jy} 为聚类 j 的均值，d_{Rnew} 是加区域 A 到聚类 j 后聚类内各区域之间的平均距离。采用试探法选择 K 的值，确定不同的物体个数。一般试探 K=2-5，同时背景区域作为一个单独的聚类，加到四种聚类结果中。背景区域聚类和前景区域做成一种聚类结果。在这五种结果中，选取聚类品质最好的作为最后的分类结果。

4.5.3 对象表示

对于 MPEG-4 视频来说，VOP 是一个重要的概念。MPEG-4 视频由多个视频会话或场景组成,而视频会话是多个视频对象 VO(参见 1.3 节)的集合。视频对象由一个或者多个 VOL 分辨层来组成，VOL 序列是 VO 在不同空间分辨率上的表示形式，通常以基本层和多个增强层的形式出现。每个 VOL 用 VOP 序列反映其在时间上的采样。

MPEG-4 的视频编码是基于 VOP 进行的，主要分为 VOP 形状编码和传统的纹理、运动编码。VOP 的形状一般是任意的，是对场景分割各 VO 的描述，需要专门的编码。VOP 内部编码同 MPEG2 相似，采用帧内编码和帧间预测编码，使用宏块作为编码的基本单位。所以，获取 VOP 对于实现 MPEG-4 是非常重要的。

经过时空域生长和运动分析,本文算法将视频中的内容分成了运动前景物体和静态背景，视频对象用单一 VOL 表示，其中 VOP 由元素在各帧的投影来计算。要实现对象多个分辨率的表示,还需要进一步在已有的 VOP 内进行利用其它信息进行分析，如利用人脸模型以及人脸检测定位等技术建立增强层，这需进一步深

入的研究。

4.5.4 交互分割与语义对象

VOP 能否成功提取直接影响 MPEG-4 优越性的发挥,但由于 VOP 定义的主观性,目前国际上仍缺乏有效的语义对象提取方法。全自动方案不需要人的帮助,整个提取过程自动进行 ,这只有在已知 VOP 具有某种特定的、图像帧的其它部分区分开的特征 (如颜色或运动特征)时才可行,其适应范围窄,实用性并不是很好。另一方面, MPEG-7 多媒体内容描述用户接口的发展和应用, 进行基于内容的视频检索, 首先要解决的问题就是获得具有语义的视频对象。

为了得到语义视频对象, 经常使用交互式视频分割方案, 通过用户的参与引入语义信息, 交互式分割方案又可分为两大类:一类是重要参数辅助输入的半自动方案,一类是人工初始输入的半自动方案。前一类方案依照人对序列和分割结果的判断调整算法的某些参数,在提取过程之前需要人工输入运动滤波器和对象跟踪器的有关参数才能使结果达到最佳。第二类方案是通过人工输入确定初始帧 VOP 范围,利用一些算法边缘跟踪算法获得初始帧的 VOP,并在后继帧中自动跟踪此 VOP 的形变和运动。这类方法的优点是提取 VOP 的边缘较为精确,不但适用于运动视频对象,也适用于静止视频对象,是目前较为成熟的方法,其缺点是用户的工作量较大,无法实时进行。

本文提出的时空域分割方法, 可输出视频中的最小同质区域, 通过用户操作, 完成对这些小元素的对象化。例如可以定义如下交互:

- 选取: 在关键帧种选择一个元素的投影区域, 并进行其它操作或赋给语义。
- 合并: 将若干小的无意义区域合成一个具有语义的区域, 如面部的各器官分成的小区域合并成头一个整体区域。
- 分裂: 对于同质区域, 也可能具有较细致的语义。如头颈部具有相同的肤色, 但可以分成面部、颈部多个语义区域。
- 细化: 为了得到某些局部区域的精确边缘, 可以在局部区域应用不同阈值, 起到细化边缘的作用。
- 分层: 建立对象层, 分别容纳在不同分辨率的语义对象。

在本文方法的基础上, 引入用户交互, 可以容易得到具有语义的视频对象分割, 因此本文的方法对交互式分割研究也有一定参考价值。

第五章 时空域视频对象分割方案

5.1 分割方案

前一章我们研究了进行视频对象分割的相关问题和技术，并设计了解决问题所需的算法，我们可以把这些算法组合在一起，形成一个视频对象分割方案。这个分割方案按照实施顺序可以划分三个部分，分别是视频分段、彩色视频预处理和时空域分割。视频分段主要对数字视频或视频图像序列，进行镜头切分，按照场景和内容的不同，把视频在时间上分成若干段，后续程序对每一段分别进行处理。由于本算法在 HSV 颜色空间内处理视频图像，所以在预处理阶段先把视频由 RGB 颜色空间转换到 HSV 颜色空间，而后对得到的 HSV 空间图像进行加权中值滤波，平滑图像和消除噪声。在时空域分割阶段，先进行三维区域生长，把镜头内的视频帧分成多个具有颜色同质性的空间元素。接下来对这些元素进行运动分析，按照运动相似的原则，将他们组合成视频对象，并用视频对象板(VOP)存储和表示。整个方案的实施流程可参见图 5-1。

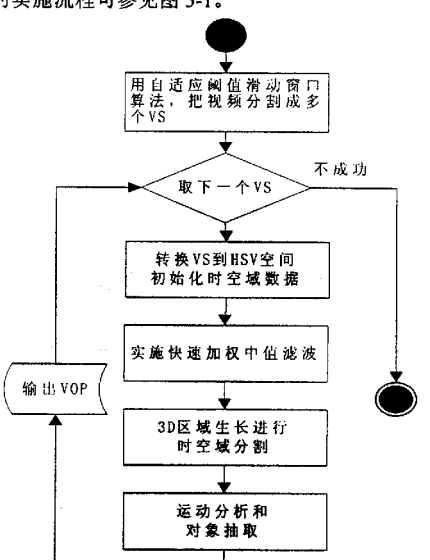


图 5-1 自动分割方案流程图

5.2 实验结果

使用 Miss American 序列对这个方案进行测试，实验在 PentiumⅣ 1.6G 和主存大小 256M 的微机上进行。实验证明，算法可以成功将运动前景的从背景中分离出来。

原始序列：

图 5-2 原始序列

区域生长结果：

图 5-3 区域生长结果伪彩图

变化检测模板

图 5-4 变化检测模板示例

运动前景提取结果：

图 5-5 视频对象板和运动前景

第六章 总结与展望

随着信息技术的发展，数字视频技术日益受到人们的关注。为了解决数字视频数据量巨大的问题，数字视频的压缩技术成为视频应用的重要支撑技术。新的压缩标准 MPEG-4，提出了基于内容编码的重要思想，把视频数据分成一个个视频对象压缩、存储和传输，促使视频对象分割技术成为一个研究热点。

本文以视频对象分割技术为研究课题，讨论视频分割相关的理论与技术，比较和研究现有的分割算法，在基于 3D 区域生长的时空域分割算法方面展开深入研究。讨论实现时空域分割要解决的几个关键问题，并给出了解决这些问题的方法，最后将上述算法结合在一起，形成了一个视频对象自动分割方案，实验证明，它有效的解决运动前景和背景分离的问题，并成功的完成从视频图像序列中抽取视频对象板的任务。本文主要的工作总结如下：

- 进行三维区域生长，首要解决的问题是如何选取生长种子，本文提出一种改进的均匀选种方法，结合局部边缘信息和空间梯度来确定种子的位置，避免噪音点和边缘点的干扰。同时基于视频目标多出现在镜头中央的事实，提出一种卷心菜算法，使得目标区域能优先生长。同传统的顺序生长相比，目标区域生长更加充分，同时避免图像外边界处噪音的干扰，改善了三维区域生长算法的效果。
- 为得到运动的前景对象，需要把同质组件的分类组合成不同的视频对象，本文使用二维空间聚类算法，同时利用运动特征和空间位置特征，既考虑到同一对象的组件具有运动相似性，又利用它们空间距离接近的特性。特征选取得当，能有效地将运动相似的相邻组件组合成视频对象。
- 把视频分成场景和内容相对固定的镜头是进行视频分割首先要解决的问题。本文提出一种自适应阈值视频分段算法，通过检测帧差的突变位置，把视频分成若干镜头，实验证明该算法具有较高的查全率和准确率。
- 组合相关时空域生长的关键算法形成了以 MPEG-4 为服务目标的视频对象自动分割方案，应用该方案进行 VOP 的抽取，能取得比较好的效果。
- 针对三维时空域数据的特点，本文构建时空域数据结构来支持生长算法的进行。借鉴游程编码的思想，实现了一种空间元素边界存储结构，使用它可以方便地进行空间元素统计和边界遍历。
- 使用三维区域生长算法进行视频分割，能充分利用时域和空域信息，同时对多个帧进行分割，具有分割边界精确，执行速度快的特点。本文探索这一领域并解决相关问题，对其它分割方法的研究也有重要的参考价值。

视频对象分割技术有非常广泛的应用领域和巨大的实用价值。它不仅是基于对象的视频压缩标准的核心，还在视频对象操纵和编辑、视频数据库检索、视频监控、视频场景理解等应用领域发挥着重要的作用。虽然，视频分割技术是非常活跃的研究领域，但是由于该问题的病态特征，至今人没有任何方案可以从根本上解决问题。所有的方法都要在一定约束条件下，才能取得较好的效果。本文提出的算法，虽然在简单场景和对象的情况下具有较好的分割效果，但是还是存在很多不尽完善的地方，应该从以下几个方面进行改进。

首先，本文算法在生成同质组件生成时，在对象的纹理特性方面考虑不足，把具有纹理的对象分成了许多小的区域处理，造成算法的性能下降，而且会对最终分割结果产生一定的影响。改进办法是在预处理中进行纹理分割，并选单一颜色作为纹理区域的掩模，然后进行后续的生长算法。

另外，本文的算法以运动相似性为测度，通过对同质视频元素进行聚类得到视频对象，因而这些对象只具有简单的语义。通过引入用户交互合并同质区域来改进，将会得到具有完整语义的视频对象，因此如何把自动方法应用到交互式分割中是本文今后的研究重点。

最后，通过底层特征得到完整语义的对象是本文未解决的问题，这是自动分割方法今后的研究方向，有待于继续探索和研究。

致谢

首先非常感谢我的导师王保保副教授。在整个研究生阶段，王老师在学习、科研方面启发我、鼓励我，使我不断进步。王老师的言传身教使我学会了很多做人的道理。王老师敏锐的思维、渊博的知识、严谨的治学态度，使我终身受益。在此，谨向王老师表示最衷心的谢意！

感谢黄凤贤同学在算法研究方面给予我的帮助。感谢西安交大的童端同学为我的论文撰写提供了许多宝贵的资料。赵君卫、张静波和宋健峰，他们丰富的知识、敏锐的思维给了我很多启迪。还要感谢舍友——张富斌、何卫、张浙峰和权宁强，和他们一起使我度过了愉快的研究生生活。

徐静女士为我的论文的排版和内容优化提出了宝贵的意见，在此表示深深的感谢。

最后，我要感谢我的父母和家人，他们一直默默地关心和鼓励我，离开他们的支持和关怀，就不可能完成研究生期间的学业和论文撰写工作。

参考文献

[1]ISO/IEC JTC1/SC29/WG11，Overview of the MPEG-4 Standard[S]，N4030，Singapore，March 2001

[2]Jonathan Dakss, Stefan Agamanolis, etc., Hyperlinked Video, SPIE Multimedia Systems and Applications, 3528, 1998

[3] ISO/IEC JTC1/SC29/WG11，MPEG-7 Requirements document[S]，N4035，Singapore，March 2001

[4]Choi J G, Lee S W, Kim S D. Spatio-temporal video segmentation using a joint similarity measure. IEEE Trans on Circuits and System for Video Technology, 1997,(7):279-186.

[5]Thomas M, King N. Automatic segmentation of moving objects for video object plane generation. IEEE Trans on Circuits and System for Video Technology,1998,8(5):525-538.

[6]Zheng Q, Chellappa R. Automatic feature point extraction and tracking in image sequential hypothesis testing, IEEE Trans Signal Processing,1991,39:1611-1629.

[7]Bouthemy P,Francois E. Motion Segmentation and qualitative dynamic scene analysis from an image sequence. Int J Computer Vision,1993,10(2):157-182.

[8]Kim M, Choi J G,Lee M H,et al. User-assistanted segmentation for moving objects of interest. ISO/IEC JTC1/SC29/WG11 MPEG97/m2803,1997.

[9]Kim M. Jeon J G, Kwak J. et al. User's guide for user-assisted video object segmentation tool. ISO/IEC JTC1/SC29/WG11 MPEG98/m3935,1998

[10] Alexandre F, Gerard M. Adaptive color background modeling for real-time segmentation of video streams. Proc. Of International on Imaging Science, System and Technology,1999:227-232.

[11]T.Merier,K.N.Ngan, Automatic segmentation of movings for video object plane generation[J].IEEE Transactions on Circuits and Systems for Video Technology,1998,8(5):525-538

[12]A.Murat Tekalp University of Rochester, Digital Video Processing 清华大学出版社，Prentice Hall 公司，1997.11

[13]koga,T.,et al. Motion-compensated interframe coding for video conferencing,In Nat.Telecommun.Conf.(Nov.1981),G5.3.1-5,New Orleans,LA.

[14]Yao Wang, Jorn Ostermann, Ya-Qin Zhang, Video Processing and

Communications,电子工业出版社，Prentice Hall，2003.6,112-113

[15]季白杨，陈纯，钱英，视频分割技术的发展，计算机研究与发展，Vol 38, No 1,Jan 2001

[16]F.Dufaux,F.Moscheni, and A.Lippman. Spatio-temporal segmentation based on motion and static segmentation. Proceedings of International Conference on Image Processing,Vol.1,1995

[17]L.Luccese and S. Mitra. Unsupervised segmentation of color images based on K-means clustering in the chromaticity plane. Proceedings of Content-based Access of Image and Video Libraries,Vol.1,1999.

[18]卢官明，用基于运动的区域生长法分割序列图像，电视技术，2000.11.

[19] Wang, J.Y.A. and Adelson, E.H.,Representing Moving Images with Layers. IEEE Transactions on Image Processing, Vol.3 No.5:625-638,Sept. 1994.

[20] R.Mech and M.Wollborn.A noise robust method for segmentation of moving objects. Proceedings of International Conference on Acoustics, Speech,and Signal Processing, Vol.4,1997.

[21] P.Salembier and M.Pardas. Hierachical morphological segmentation for image sequence coding. IEEE Transactions on Image Processing,Vol.3,1994.

[22]Yao Wang ,Jorn Ostermann,Ya-Qin Zhang,Video Processing and Communications, 电子工业出版社，Prentice Hall 公司，2003.6,101-105

[23] J.G.Choi,S.W.Lee,S. D.Kim.Spatio-temporal video segmentation using a joint similarity measure. IEEE Transactions on Circuits and Systems for video Technology,Vol.7, 1997.

[24]N. Diehl.Object-oriented motion estimation and segmentation in image sequences, Signal Processing: Image Communication,Vol.3,1991.

[25] J.Canny. A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence,Vol.8,1986.

[26] Yucel Altunbasak，P. Erhan Eren，A. Murat Tekalp，Region-based parametric motion segmentation using color information，Graphical Models and Image Processing，January 1998,Vol. 60 Issue1:13 - 23

[27] 高文，陈熙霖著.计算机视觉.清华大学出版社，1998.

[28] 崔屹，图象处理与分析——数学形态学方法及应用，科学出版社，2000

[29] B.Duc,P Schtoeter, and J.Bigun. Spatio-temporal robust motion estimation and segmentation. Proceedings of International Conference on Computer Analysis of Images and Patterns, Vol. 1,1995.

[30] Marques and C.Molina. Object tracking for content-based functionalities. SPIE

Proceedings of Visual Communication and Image Processing, Vol.3024,1997

[31]F.Porikli,"Video object segmentation by volume growing using feature-based motion estimator",Proc 16 th Internation Symposium On Computer and Information Science,Antalya,Turkey,November 2001.

[32]B.Duc, P.Schtoeter, and J.Bigun. Spatio-temporal robust motion estimation and segmentation.Proceedings of International Conference on Computer Analysis of Image and Patterns,Vol.1,1995.

[33]G.Adiv.Determining the three-dimensional motion and structure from optical flow generated by several moving objects. IEEE Transactions on Pattern Analysis Machine Intelligence, Vol. 7,1985.

[34]J.Wang and E.Adelson.Spatio-temporal segmentation of video data.Proceedings of SPIE on Image and Video Processing,Vol.2182,1996.

[35]I.Pitas and A.N. Venetsanopoulos, "Order Statistics in Digital Image Processing", Proceedings of the IEEE,Vol 80,No 12,Dec,1992

[36]A.Neri,S.Colonnese, G.Russo, and P.Talone.Automatic moving object and background separation. Signal Processing,Vol.66,1998.

[37] G.R. Aree,and R.E.Foster,"Detail-preserving Ranked-Order Based Filters for Image Processing",IEEE Trans on ASSP,Vol 37,No 1,Jan 1989

[38]Horn,B.K.P, and B.G.Schunck. Determining optical flow. Artificial intelligence, Vol. 1,1981:185-203

[39]周赟，李久贤，夏良正，基于区域生长的红外图像分割，南京理工大学学报，Vol26 , 2002.11:75-78

[40] Yong Hoon Lee,and S.A. Kassam,"Generalized Median Filtering and Related Nonlinear Filtering Techniques",IEEE Trans. On ASSP,Vol 33,NO.3,June,1985

[41]Murray DW,Buxton B F.Scene segmentation from visual motion using global optimization.IEEE-PAMI,1987.9,220~228

[42]章毓晋,正在制订的国际标准—MPEG-7.电子科技导报，1999,11:15-18

[43]Patel N V,Sethi I S,Video shot Detection and characterization for Video databases,Pattern recognition,1997,30(4):583~592

[44]M.Chang,M.I.Sezan,and A.M.Tekalp. An algorithm for simultaneous motion estimation and scene segmentation. Proceedings of International Conference on Acoustics,Speech,and Signal Processing,Vol.5,1994.

[45]M.Celenk.Color image segmentation by clustering. Proceedings of Computers and Digital Techniques,Vol.138,1991.

[46]Zabih R, Miller J, Mai K.A featured-based algorithm for detecting and classifying

scene breaks. Procededings of ACM Multimedia'95,1995,189-200
[47]钟玉琢，王琪，贺玉文，基于对象的多媒体压缩编码国际标准 MPEG-4 及其校验模型，科学出版社，2000
[48]周洞汝，胡宏斌等.视频数据库管理系统导论，科学出版社，2000
[49]章毓晋，图象分割，科学出版社，2001
[50]Kenneth.R.Castleman, Digital Image Processing,电子工业出版社，1998
[51]边肇祺，张学工等著，模式识别，清华大学出版社，2000
[52] 阮秋琪，数字图像处理学，电子工业出版社，2001

在读期间的研究成果

一、参加科研情况

参加了“梯度透镜丝径检测”图像测量算法的研究工作；

参加并负责“X-光机图像处理系统”项目开发，此系统已交付使用。

二、发表论文情况

陈博，王保保，黄凤贤，《一种高精度玻璃丝径测量算法》，

发表于《计算机仿真》杂志 2004 年第 6 期。

陈博，徐静，王保保，《自适应阈值镜头分割算法》，

发表于《微机发展》杂志，已录用，待发。

附录 卷心菜算法伪代码

```
//帧中水平方向的块数HBlocks; 垂直方向的块数VBlocks
//序列起点帧号BFrame;终点帧号EFrame
//
void Process(int i,int j,int k)//处理第k帧第i行,第j列的块

...

void SingleTravel(int FrameCode,int CRow,int CCol,int Level)//帧内单层
遍历
    int l,r,t,b,i;
    l=(CCol-Level>0)?CCol-Level:1;
    r=(CCol+Level<HBlocks)?CCol+Level:HBlocks;
    t=(CRow-Level>0)?CRow-Level:1;
    b=(CRow+Level<VBlocks)?CRow+Level:VBlocks;
    for (i=l;i<=r;i++) Process(t,i,FrameCode);
    for (i=t;i<=b;i++) Process(i,r,FrameCode);
    for (i=r;i>=l;i--) Process(b,i,FrameCode);
    for (i=b;i>=t;i--) Process(i,l,FrameCode);
    return;
}
void Travel()//卷心菜遍历

int i,j,B,E;
int CFrm=(BFrame+EFrame)/2;
int CRow=VBlocks/2;
int CCol=HBlocks/2;
int Level=1;
while (1)

if ((CFrm-Level<BFrame)&&(CFrm+Level>EFrame)&& (CRow-Level<1)
    &&(CRow+Level>VBlocks)&&(CCol-Level<1)&&(CCol+Level>HBlocks))
    return;//结束
SingleTravel(CFrm,CRow,CCol,Level);//遍历中心所在帧
B=(CFrm-Level<BFrame)?BFrame:CFrm-Level;
E=(CFrm+Level>EFrame)?EFrame:CFrm+Level;
for (i=1;i<Level;i++)

  if (CFrm-i>B) SingleTravel(CFrm-i,CRow,CCol,Level);
  if (CFrm+i<E) SingleTravel(CFrm+i,CRow,CCol,Level);

if (B==CFrm-Level)
  for (i=0;i<=Level;i++) SingleTravel(B,CRow,CCol,i);
if (E==CFrm+Level)
  for (i=0;i<=Level;i++) SingleTravel(E,CRow,CCol,i);
Level++;
```