

清 华 大 学

综 合 论 文 训 练

题目：基于内容的视频复制检测

系 别：计算机科学与技术系

专 业：计算机科学与技术

姓 名：梁颖宇

指导教师：张钹 教授

辅导教师：李建民 助理研究员

2008 年 6 月 13 日

关于学位论文使用授权的说明

本人完全了解清华大学有关保留、使用学位论文的规定，即：学校有权保留学位论文的复印件，允许该论文被查阅和借阅；学校可以公布该论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存该论文。

(涉密的学位论文在解密后应遵守此规定)

签 名：梁颖宇 导师签名：张敏 日 期：2008年6月13日

中文摘要

该论文进行了对基于内容的视频复制检测的研究。在给出视频复制检测的定义及其广泛应用后，文章从视频复制检测使用的特征出发，介绍了目前国内外相关领域的两类主要算法。之后论文提出了基于顺序度量的直方图匹配算法，并在 CIVR 提供的数据集 MUSCLE-VCD-2007^[1]上进行测试，结果显示其时间复杂性低，但不能检测到某些变换后的复制视频。因此我们提出了基于 SURF^[19]特征的图片序列匹配算法，实验显示其性能和复杂度都令人满意，而且这个算法表现出较高的局部精确度，并可以通过后续处理获得进一步改善。

关键词：复制视频 视频复制检测 SURF CIVR

ABSTRACT

This thesis's topic is Content-Based Video Copy Detection. After bringing out the definition of the task and its mass application, we introduce two kinds of existing mainstream algorithms from the view of features used. Then, an algorithm based on ordinal measure and statistics histogram is proposed and tested on the database MUSCLE-VCD-2007^[1] provided by CIVR. The result shows it represents relatively low time complexity but fails to detect certain kinds of video transformations. So we propose an image-sequence matching algorithm based on SURF^[19]. Its performance and complexity are very satisfying. Especially, this algorithm represents high local accuracy, which can be further improved by a successive step.

Keywords: Video Copy Video Copy Detection SURF CIVR

目 录

第 1 章 背景介绍和任务描述	1
1.1 背景介绍	1
1.2 复制视频和视频复制检测的定义	1
1.2.1 复制视频的定义	1
1.2.2 视频复制检测的定义	5
1.2.3 基于内容的视频复制检测	6
1.3 视频复制检测的应用	6
1.3.1 版权保护	6
1.3.2 视频跟踪	7
1.3.3 冗余去除	7
1.3.4 语境关联	7
1.4 论文结构介绍	7
第 2 章 相关工作	9
2.1 概述	9
2.2 特征抽取	9
2.2.1 全局特征	9
2.2.2 局部特征	12
2.3 搜索	13
2.4 总结	13
第 3 章 基于顺序度量的直方图匹配算法	15
3.1 算法概述	15
3.2 算法流程	16
3.2.1 帧级别特征提取	16
3.2.2 镜头级别特征生成	19
3.2.3 建立索引和查找相似镜头	20

3.2.4 筛选候选视频	21
3.2.5 子序列匹配	21
3.3 算法测试	23
3.3.1 数据库介绍	23
3.3.2 离线处理工作	25
3.3.3 第一类查询测试结果	25
3.3.4 第二类查询测试结果	26
3.3.5 结果讨论	26
3.4 算法总结	27
第 4 章 基于 SURF 的图片序列匹配算法	29
4.1 算法概述	29
4.2 算法介绍	30
4.2.1 局部特征提取	30
4.2.2 统计分布直方图	34
4.2.3 建立索引和查找相似镜头	36
4.2.4 筛选候选视频	36
4.2.5 改善局部精确度	42
4.3 算法测试	42
4.3.1 离线处理工作	42
4.3.2 第一类查询测试结果	43
4.3.3 第二类查询测试结果	43
4.3.4 结果讨论	44
4.4 算法总结	44
第 5 章 总结与展望	47
5.1 已完成的工作	47
5.2 主要成果	47
5.3 下一步工作	48
5.3.1 镜头划分	48
5.3.2 局部特征的选取	48
5.3.3 局部特征的提取	48

5.3.4 基向量的选取	48
5.3.5 使用更大更复杂的测试集	48
插图索引	49
表格索引	52
参考文献	53
致 谢	55
声 明	56
附录 A 外文资料的阅读调研报告或书面翻译	57

主要符号对照表

ANN	近似近邻 (Approximate Nearest Neighbor)
SURF	快速鲁棒特征 (Speeded Up Robust Features)

第1章 背景介绍和任务描述

1.1 背景介绍

随着视频技术的快速发展和日益广泛的使用，通过电视频道、计算机网络和多媒体数据库可以获得的视频文件正以爆炸式的速度增加。越来越普及的便携式摄像机和各种多媒体流通渠道使得视频的获取变得轻而易举，而利用视频处理软件，如Adobe公司的Premier系列软件、微软的Movie Maker软件等，人们可以很方便地对视频材料进行处理，并将处理后的视频在好友间分享，或者将它们上传到视频网站如www.youtube.com、www.youku.com等。因此在网络上或者各种视频数据库中存在着大量相互之间内容基本相同但是可能经过各种变换处理的复制视频，使得视频复制检测成为迫切的需要。

1.2 复制视频和视频复制检测的定义

复制视频或者视频复制检测的一般意义是不言自明的，但是仍然缺乏公认的准确定义。

1.2.1 复制视频的定义

在[2]中，复制视频是指对源视频进行各种处理之后得到的视频。在[3]中给出了复制视频的一个类似的定义：视频 P 称为视频 O 的一份复制，如果 $P = t(O)$ ， $t \in T$ ， T 为可忍受的视频变换的集合，即经过这些视频变换处理后得到的视频仍然能为人辨认出其内容来自于源视频。

定义中视频变换需要进一步的说明。早期的一些研究工作主要关注视频的编码格式变换、视频图像的颜色和分辨率调整等比较温和的变换。如[10]中考虑了视频的编码格式变换：

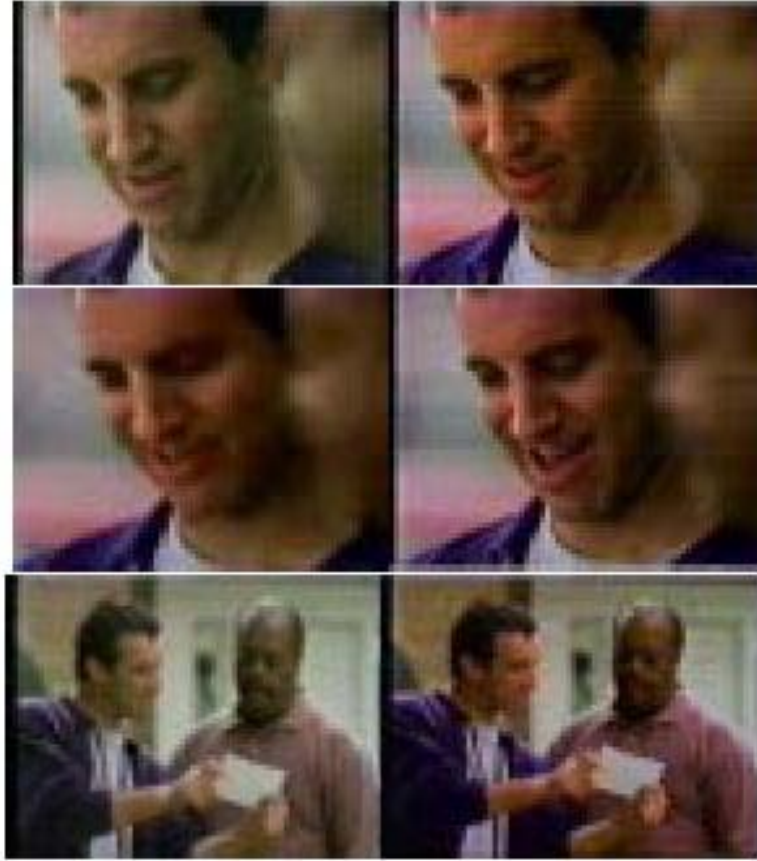


图1.1 [10]复制视频示例。左列为源视频，右列为视频编码格式变换得到的复制视频。各视频的格式如下。左上：Mpeg1，右上：Avi，左中：RealVideo128k，右中：RealVideo512k，左下：Mpeg1，右下：Avi。

随着这些变换得到了很好的处理，后续的一些研究工作开始关注一些更加严厉的变换，如在[6]中考虑了帧率的改变。其他包括插入标志和字幕、画面的偏移、放缩和剪裁、帧的丢失和插入等。



(a) *La Télé des Inconnus* P. Lederman 1990 (c).



(b) *Celine Dion Music Video* 2002 (c).



(c) *The Full Monty* 1997 (c) 20th Century Fox.

图1.2 [3]中复制视频示例。其中(a)为颜色调整、模糊和噪声变换的示例，(b)为放缩、偏移和插入字幕的示例，(c)为垂直方向拉伸和剪裁的示例。

近期进入研究范围的一些变换有水平翻转、camcoding等。其中camcoding是指在屏幕上放映源视频时，使用便携式摄像机拍摄得到的视频。下面给出MUSCLE-VCD-2007^[1]中的一些复制视频的示例。

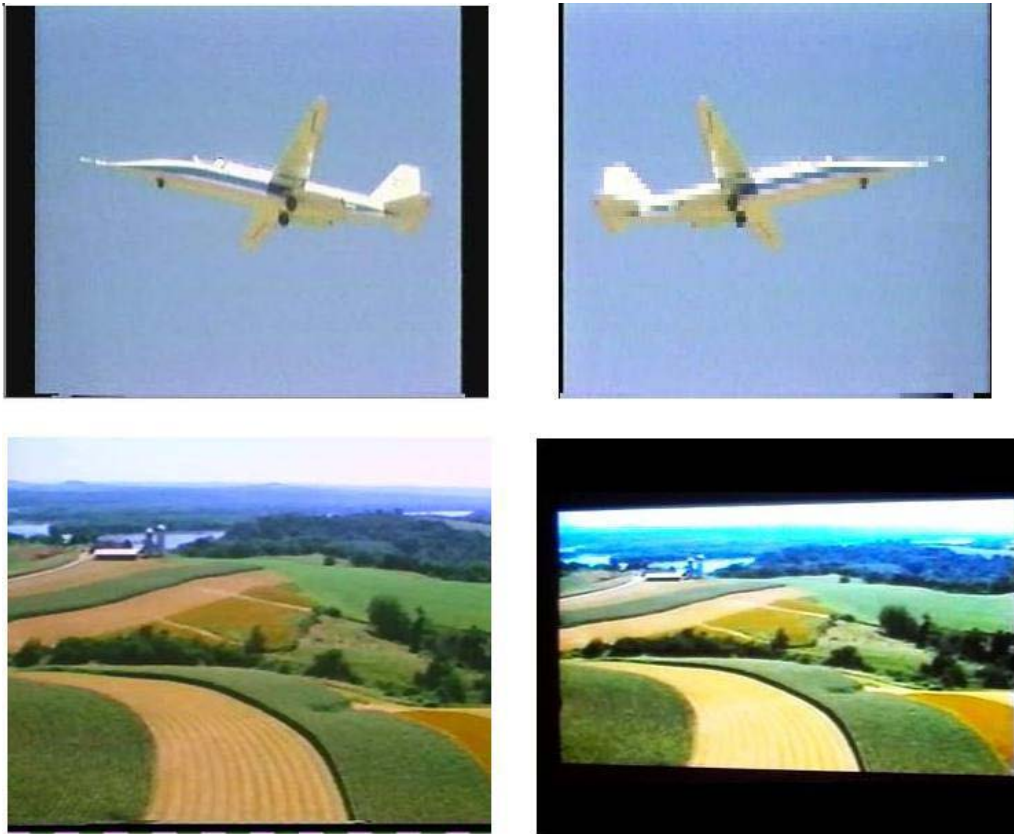


图1.3 MUSCLE-VCD-2007^[1]中复制视频示例。右上为左上经过水平翻转和剪裁得到的视频，右下为左下经过camcoding得到的视频。

综上所述，本论文考虑的视频变换处理包括以下方面：

- 1) 视频格式变换：编码格式、帧率、画面分辨率等；
- 2) 颜色变换：颜色值、对比度、gamma值等；
- 3) 画面几何变换：放缩、剪裁、偏移、翻转等；
- 4) 插入变换：标志、边框、字幕、画中画等；
- 5) 随机变换：帧丢失、插入、画面噪声等；
- 6) camcoding 变换；

值得注意的是，一些最近的研究关注更为严厉的转变。由于这些变换在现实中很少见，甚至在变换后画面变化太大，以至于人们难以识别出得到的视频是否和源视频内容相同，所以在本论文中没有考虑这些变换。



图1.4 严厉变换示例^[3]。第一行为虚拟背景变换，第二行为插入大标志和颜色变换。

1.2.2 视频复制检测的定义

将视频 V 视为一个帧序列 $\{V_i, 1 \leq i \leq N\}$ 。基于复制视频的概念，视频复制检测的形式定义如下：

给定一个视频集合 $\{R^j\}$ ，和一个查询视频 $Q = \{Q_i, 1 \leq i \leq N\}$ ，视频复制检测的任务为对于 $\{R^j\}$ 中的每个视频 $R^j = \{R_i^j, 1 \leq i \leq M^j\}$ ，找出可能存在的 $1 \leq u < v \leq N$ 和 $1 \leq x < y \leq M^j$ ，满足 $\{Q_i, u \leq i \leq v\}$ 是 $\{R_i^j, x \leq i \leq y\}$ 的一份复制。一般只考虑视频片段 $\{Q_i, u \leq i \leq v\}$ 和 $\{R_i^j, x \leq i \leq y\}$ 足够长的情况，在本论文中为大于等于 10 秒。



图1.5 视频复制检测定义的图示

许多参考文献只考虑 $u = 1, v = N$ 的情况，如[4]-[7]。有些参考文献关注一个更加特殊的情况： $u = 1, v = N$ 并且 $x = 1, y = M^j$ ，如[7]和[8]。这些情况都包含在上面给出的一般性的定义中，后者更加复杂，但也具有更强的现实意义。因此本论文采用的是一般性的定义。

1.2.3 基于内容的视频复制检测

视频复制检测有两种途径，分别是基于关键字的视频复制检测、基于内容的视频复制检测。基于关键字的视频复制检测是指利用视频的标题、标注等附加的文字信息判断是否为复制视频的方法。很明显，这种方法具有非常大的局限性，现实意义不大，因此在本论文中没有考虑。基于内容的视频复制检测

(Content-Based Video Copy Detection) 是指利用视频本身的信息判断是否为复制视频的方法，是本论文的研究主题。为了陈述方便，后面提到基于内容的复制检测均简称视频复制检测。

1.3 视频复制检测的应用

视频复制检测有广泛的应用前景。

1.3.1 版权保护

这是视频复制检测的原动力，也是目前比较成熟的应用。主要有两种方法可以应用于视频的版权保护，分别是数字水印和基于内容的视频复制检测。在数字水印方法中，在视频发布之前在视频中嵌入额外的信息，并利用这些额外的信息来证明版权的合法性。基于内容的视频复制检测的思想是“视频本身即为水印”，即通过提取视频本身的信息判断是否为复制视频。与数字水印相比，它的一个主要优点是已经发布的视频也可以进行保护，即便这些视频中并没有嵌入水印。

1.3.2 视频跟踪

当需要跟踪某个视频的应用时，可以使用视频复制检测的方法。例如，某公司需要确保它的广告确实在某几个电视频道中播放。又如，需要跟踪竞争对手的广告时间安排，以便分析对手的商业策略。

1.3.3 冗余去除

在某些情况下，复制视频是多余甚至是有害的。如果能够检测出这些复制视频，则可以进行冗余去除。例如在视频搜索中，同一视频的多个版本会削弱搜索结果的有效性，降低用户搜索效率。而通过视频复制检测，可以将多余的复制视频删除，只返回最有效的版本，则可以在很大程度上改善用户体验。另一个应用涉及视频数据库，其中的冗余视频会显著地增大数据库的大小和降低系统的性能。

1.3.4 语境关联

这是一个在[11]中新提出的应用。在这种应用中，视频复制检测系统生成大量的语境关联，如视频的出现频率、出现的持续时间、地理分布等。这些语境关联包含了大量的信息，可以通过数据挖掘方法进行利用。[11]中给出了语境关联的两个例子：在同一天中出现在多个国家的电视频道中的视频对应了一个国际事件；有大量复制分布在一个较长时间段的视频对应了一个重要历史事件。

1.4 论文结构介绍

本论文其余部分的内容如下进行组织。

第2章从使用的特征出发介绍了国内外在基于内容的视频复制检测的一些主流算法。

在已有算法的基础上，我们提出了一个新的算法，首先提取顺序度量特征，统计分布直方图，在搜索出相似镜头后筛选候选视频，并利用动态规划进行局部匹配。第3章中给出了这个算法的细节，并在 MUSCLE-VCD-2007^[1]数据集上测试了算法的性能。

为了进一步提高检测的准确度和召回率，在第4章中提出了另一个新的算法。这个算法基于 SURF 特征，利用图片序列模糊匹配的方法筛选出复制视频，并可以通过后续的简单处理改进其局部精确度。在 MUSCLE-VCD-2007^[1]数据集上的测试表明这一算法的性能优越。

最终，在第 5 章中我们对整篇论文进行了总结，并提出了下一步工作的展望。

第2章 相关工作

2.1 概述

在基于内容的视频复制检测中，通常有两个主要步骤：第一步是从视频中抽取特征；第二步是在特征集合中进行搜索以找到查询视频的复制。下面分别对这两个步骤已有的相关工作进行详细说明。

2.2 特征抽取

根据[3]的分类，应用于视频复制检测的主要有两种特征，分别是全局特征和局部特征。

2.2.1 全局特征

这类特征表示的是视频的帧序列中各帧的全局空间特性，或者序列的时间特性，或者是两者的结合。

空间特征主要是图像处理中一些传统的特征，如图像的像素本身、RGB 直方图、HSV、梯度方向、边缘表示和不变矩等。在[10]中对这些特征和相应的距离度量在视频复制检测中的性能进行了比较，实验结果表明边缘表示及其 Hausdorff 距离性能最优，紧随其后的是 HSV 直方图及其相交距离。但是这些特征的性能都不是很好。在[10]的后续工作[13]中，对运动方向特征（应该归类为时间特征）、顺序度量特征和颜色直方图特征进行了实验比较。实验结果表明顺序度量有最好的性能，其次是运动方向特征，最后是颜色直方图特征。由于相对比较简单而且性能优越，最近的使用全局特征的算法通常都使用顺序度量特征或者其变种。

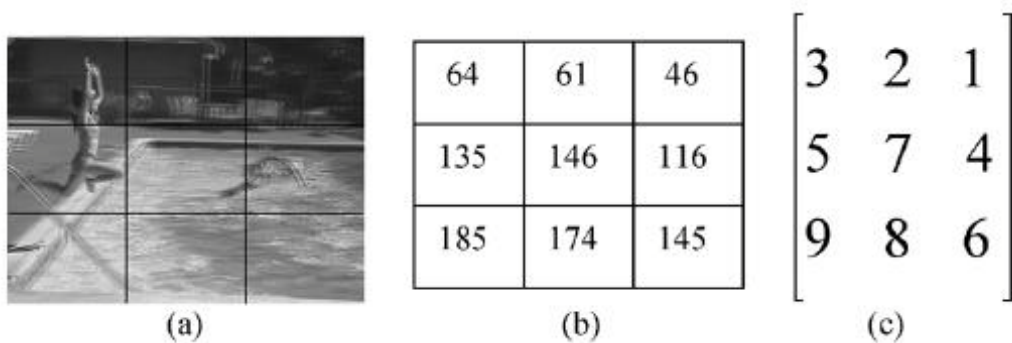


图2.1 [4]中顺序度量的图示。(a) 将图像划分为 m 行 n 列的子区域，图中为3行3列；(b) 各个子区域中灰度的平均值；(c) 灰度值的排序结果即顺序度量值。

时间特征关注的是帧序列中相邻帧之间的不同。在[3]中，参与实验比较有一个简单的时间特征，它是通过将相邻两帧各个像素的颜色差累加而得。在[13]中参与实验比较的运动方向特征也属于时间特征。将每一帧划分为 M 行 N 列的子区域；对于每个子区域，在后续一帧中搜索出与其最相似的区域，这样就得到了该子区域的运动方向；将所有子区域的运动方向视为该帧的运动方向特征。时间特征的性能不是很好，在实际中较少用到。

时空特征同时利用了空间信息和时间信息。在[13]中提出的时间顺序度量方法使用帧序列中沿着时间方向的各个子区域的灰度值排序结果，而不是使用同一帧中各个子区域的灰度值排序结果作为特征。在[4]中，两个视频的顺序度量之间的不同，和各个子区域在时间上变化的不同进行加权平均，作为两个视频之间的距离，其性能比原来的顺序度量好。

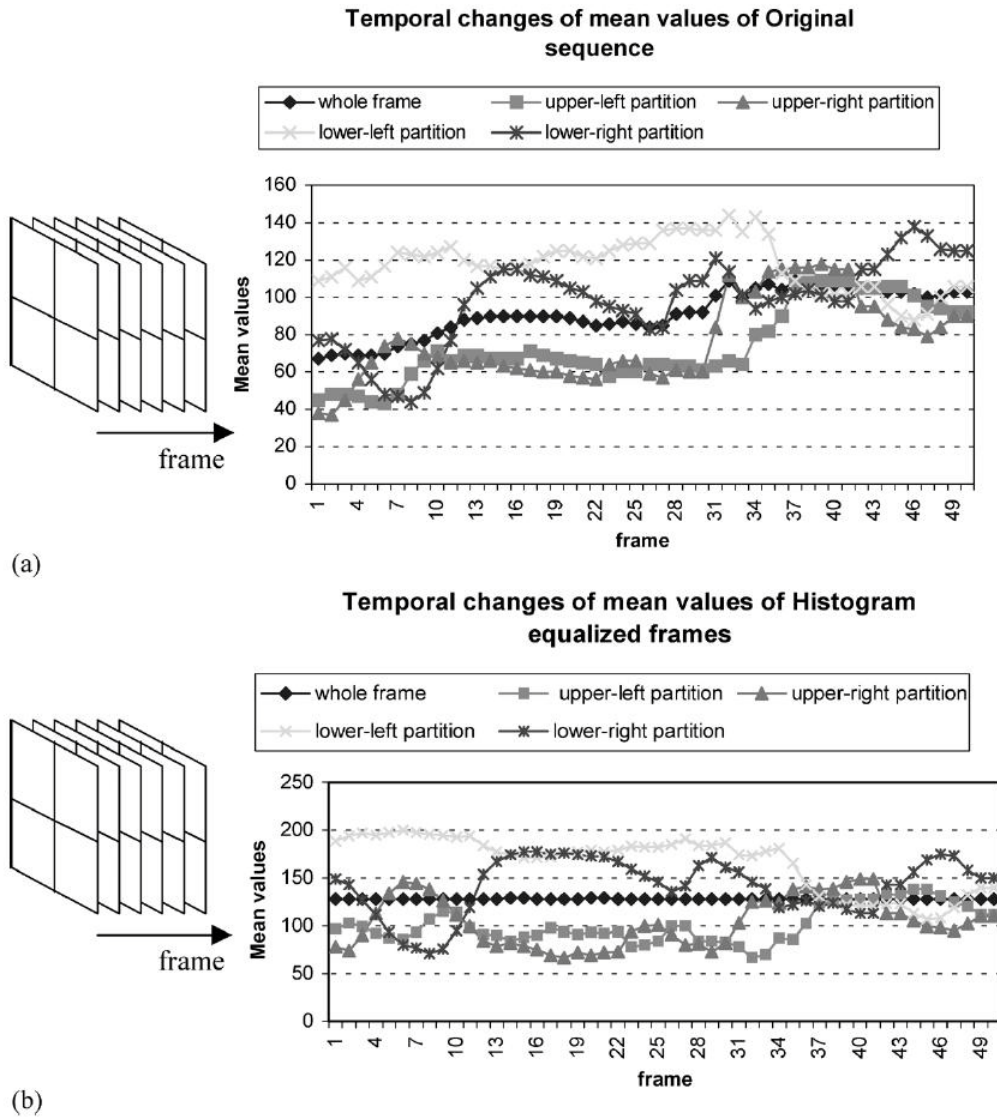


图2.2 [4]中给出的时空特征图。图中的曲线为各个子区域的平均灰度值随着时间的波动。两个对齐的视频片段之间的距离为各帧之间的距离，和图中曲线变化趋势之间的距离两者的加权平均。

同样使用时空特征的还有[7]和[8]。在[7]中，视频中所有帧的顺序度量值被压缩为分布直方图。在[8]中，在特征空间中选取基向量；将每个帧的特征向量投射到与其最近的基向量，在此基向量上累加权重；各个基向量上的权重作为视频的特征。这两种特征都是应用于检测整个视频的复制，不能直接应用于 1.2.2 中的一般情形；但是其处理方法大大减少了特征的时间和空间复杂度，启发我们学习其思想，降低算法的复杂度。第 3 章和第 4 章中将给出在算法中使用其思想的具体细节。

2.2.2 局部特征

局部特征是指各帧图像中的兴趣点。[11]中的视频复制检测系统使用了 Harris 兴趣点的一个改进版本作为检测器，并在检测到关键帧上的兴趣点后，通过对兴趣点所在的局部区域进行微分而得到视频的特征。[15]中使用的特征与[11]中的类似，但在兴趣点附近局部区域的选择上有不同，而且将兴趣点关联起来形成轨迹用于比较。除了这两个特征之外，参与[3]中比较的还有[16]中提出的时空兴趣点。关于这些特征的具体细节请参考相关文献。

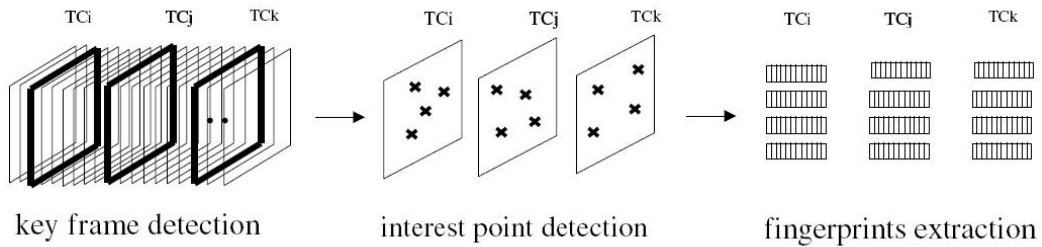


图2.3 [11]中的特征提取的图示

[3]实验结果表明顺序度量在处理大多数的单一变换时性能良好，复杂度低，但是在放缩、剪裁、插入边框、画面偏移等单一变换以及多种变换混合的情况下表现不佳。局部特征在单一变换和混合变换等情况下表现优越，但是复杂度高。但是测试数据库相对较小，且实验针对 1.2.2 中的一般定义，所以在数据库较大或者检测整段视频的复制等情况下，可能其他的特征有更好的性能。而且 Harris 兴趣点并不是很强的图像不变特征，可以考虑使用更强的局部特征以提高系统的性能。

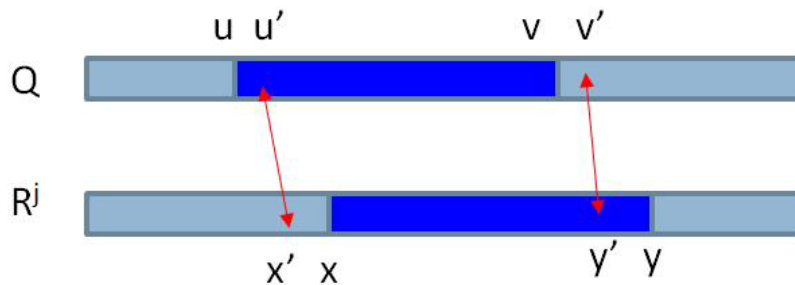


图2.4 局部精确度图示。 u, v, x, y 为正确的边界， u', v', x', y' 为实际检测时判断的边界。

值得注意的是在已有的工作中通常都没有考虑局部精确度。局部精确度是指帧级别的定位精度，即 1.2.2 给出的定义中 u, v, x, y 的判断精度，如图 2.4 所示。实际上，局部精确度也是非常重要的性能指标，而且在实际中有重要应用，在目前视频级别精确度已经可以做得较好的情况下，可以考虑改进局部精确度。

2.3 搜索

在得到视频的特征之后，视频可以表示为一系列的特征向量，则问题变为在数据库中找到与查询视频特征序列有相似子序列的视频并进行定位。需要利用查询视频的特征进行搜索，找到可能的视频片段。搜索一般可以分为两种，一种是暴力搜索，一种是在索引结构中搜索。

对于较小的数据集，可以使用暴力搜索。但是当数据集变大，暴力搜索所需的时间剧增，这对于其应用是灾难性的。因此近来的相关工作中，都是使用了各种索引结构进行搜索。如在[8]中，使用了 B^+ 树及其变种作为索引结构。在[11]中，利用了 Hilbert space-filling curve 对特征进行索引。由于索引的选择比较自由，还可以根据实际情况考虑使用其他一些合适的索引结构。

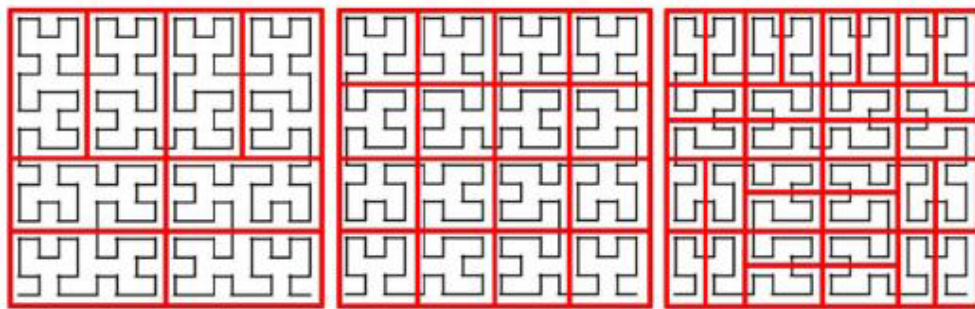


图2.5 使用Hilbert space-filling curve对二维空间进行分割的图示^[11]。从左到由分割深度为3, 4, 5。

2.4 总结

综上所述，关于视频复制检测任务，现在已经有许多比较好的算法。但是这些算法都有其缺陷，比如使用顺序度量的算法，对于某些变换性能很差，又比如使用局部特征的算法，其复杂度相对较高。而且在已有的算法中，并没有将局部

精确度作为性能的评测指标，有可能算法在局部精确度方面表现有待改进。这些缺陷都需要有新的思路和方法来解决。

第3章 基于顺序度量的直方图匹配算法

3.1 算法概述

在上一章中提到现有的算法都没有考虑局部精确度。实际上在目前的工作中，视频级别的精度和召回率都可以做得比较好，因此除了继续提高视频级别的精度和召回率、降低复杂度外，还有两个研究方向，一是考虑更多更复杂的变换，二是提高局部精确度。0 中已经提到，有些最近的工作考虑的是更加复杂的变换，但是在实际应用中，很少遇到太过复杂的变换。在 1.2.1 中限定了本论文考虑的变换，这些变换基本上可以覆盖应用中的大多数情况。另一方面，局部精确度在实际应用中也有重要意义。如禁止某段视频的流通，需要去除嵌入到其他视频中的复制片段，如果定位不精确，则有可能导致监控的力度不够，或者误删过多无关片段。实际上在[2]中已经提到，TRECVID2008 的视频复制检测任务将会把局部精确度列为视频级性能、时间之外的第三个算法评测指标，表明了局部精确度逐渐得到认可。综合以上考虑，我的想法是继续提高精度和召回率、降低复杂度，同时将局部精确度纳入性能评测标准中，尝试改进算法的局部精确度。

但是已有的工作不强调局部精确度是有原因的：为了降低复杂度，一般只提取关键帧的特征用于搜索；而关键帧之间相隔较远，只有关键帧的特征难以改进局部精确度。如果有每个帧的特征或者时间轴上特征相对密集，则在搜索出有复制片段的视频后，使用子序列匹配的算法可以改进局部精确度。

可是特征密集也存在问题，因为数目剧增的特征会导致搜索的复杂度增高。解决方法是以镜头为单位将帧的特征压缩为镜头级别的特征用于搜索，而帧级别的特征则用于之后的子序列匹配。

因此，算法可以粗略地分为三个阶段：提取特征，粗定位搜索和子序列匹配。第一阶段提取帧级别特征用于子序列匹配，同时生成镜头级别特征用于粗定位搜索。其中第三阶段子序列匹配是可选的，用于改进局部精确度，当然也可以放宽第二阶段粗定位搜索的尺度，在子序列匹配时再去除一些误判，以改进算法的召回率和精度。

在本章中介绍了基于顺序度量的直方图匹配算法，帧级别特征为顺序度量，镜头级别特征为顺序度量的分布直方图。算法的流程图如图 3.1 所示。

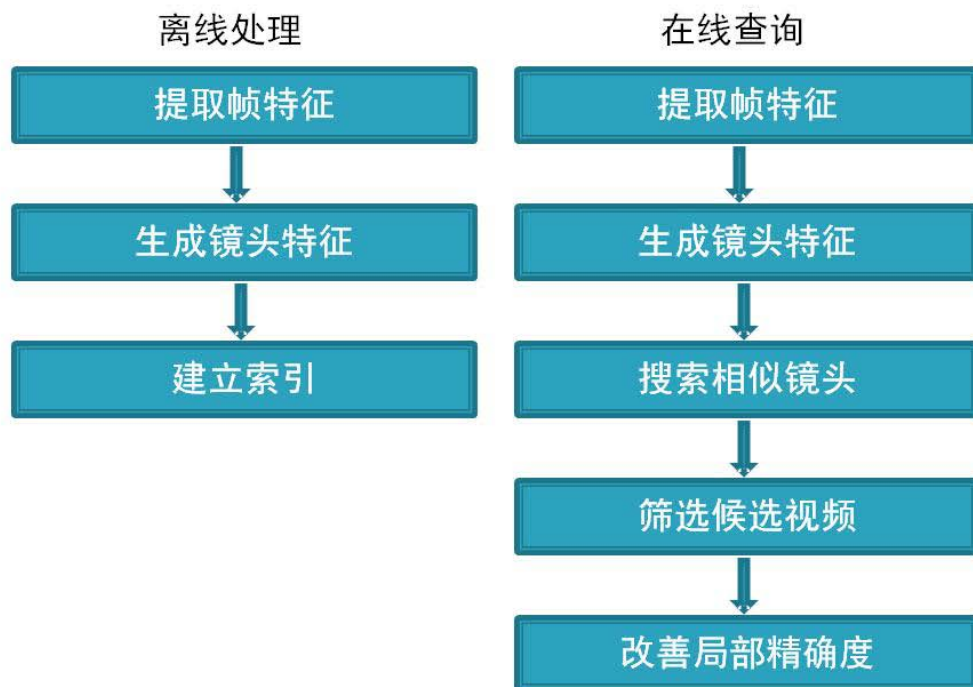


图3.1 基于顺序度量的直方图匹配算法的流程图

3.2 算法流程

下面对算法的各个步骤进行详细介绍。

3.2.1 帧级别特征提取

由于要提取每一帧的特征，为了避免时间和空间复杂度过高，需要选择一种比较简单的特征；同时为了保证性能，也需要特征有较好的分辨性和鲁棒性。分辨性好是指非复制视频的特征之间的差异较大，鲁棒性是指复制视频的特征之间的差异较小。只有同时保证特征的分辨性和鲁棒性，才能准确地搜索出复制视频。

根据[13]和[3]的实验结果，我们决定选择顺序度量作为帧级别的特征。顺序度量计算比较简单，时间复杂度低；每一帧的顺序度量可以用一个整数表示，即便是提取所有帧的特征，空间复杂度也比较低；而且顺序度量对于大多数的变换都有较好的分辨性和鲁棒性。

顺序度量提取的第一步是将图像划分为 m 行 n 列的子区域。经过对复杂度和性能的权衡，我们将图像划分为 2 行 3 列的子区域。参照[17]的做法，考虑到视频画面可能会加入标志或者字幕，为了排除这些干扰，可以只关注画面的中心区域。如图 3.2 所示，只考虑画面中心区域可以避免左下角的标志的干扰。



图3.2 顺序度量提取时的子区域划分

顺序度量提取的第二步是计算各个子区域的颜色平均值。可以对 RGB 三个通道分别求其平均值，也可以只计算灰度的平均值。因为后者对于颜色调整的变换鲁棒性更好，复杂度也较前者低，我们在算法中采用后者。但是即便只计算灰度的平均值，顺序度量对于某些变换的鲁棒性依然很差，如图 3.3 所示的放缩和插入边框变换。



图3.3 放缩和插入边框示例

如果变换前后子区域中的点完全一样,那么顺序度量的鲁棒性就能得到保证。但实际上点在变换后并不一定仍然处于原来的子区域中,这是有一个概率的。如果变换后点仍然处于原来的子区域的概率较大,那么增加它对平均值的贡献就能增强特征鲁棒性。所以我们对子区域中各点的灰度值进行加权平均,权重为变换后点仍处于原来的子区域的概率。形式地,记子区域 R 中的点 (x, y) 在复制视频中的对应点为 (x', y') ,那么其权重为 (x', y') 仍处于 R 中的概率,即

$$w_{x,y} = \iint_R p(x', y') dx' dy' \quad (3-1)$$

其中 $p(x', y')$ 是概率密度函数,假定为正态分布,期望为 (x, y) ,方差根据实验结果进行调节。在实验中发现只要落在一定范围内,方差对实验结果影响不是很大,因此最终确定为使得偏移10个像素时 $p(x', y') = 0.1$ 。

实验结果显示在添加了权重之后,特征的鲁棒性有了提高;时间增长0.5倍。由于顺序度量特征的计算时间较短,时间上0.5倍的增加是可以接受的。

顺序度量提取的第三步是对平均值进行排序,将排序结果作为特征。有些参考文献如[17]直接将平均值作为特征,而没有排序这一步。出于提高特征的鲁棒性和降低复杂度的考虑,我们选择用排序结果作为特征。我们采用的方法是对两行的灰度平均值分别排序,排序结果分别映射到0到5之间的整数 S, T ,最后将 $6S + T$ 作为最终结果,如图3.4所示。

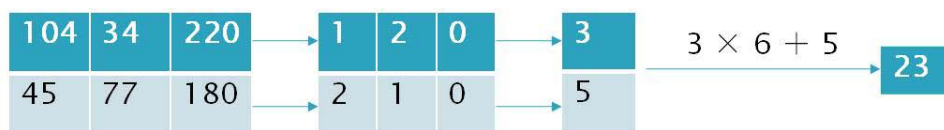


图3.4 顺序度量提取时的排序

3.2.2 镜头级别特征生成

在镜头级别上可以考虑使用与帧级别完全不同的特征，也可以使用帧级别特征作为源数据进行压缩得到。受到[7]的启发，我们采用使用帧级别特征作为源数据，压缩成分布直方图。在视频级别上，[7]的实验结果表明分布直方图易于计算，占用空间小，而保持了顺序度量的性能；在镜头级别上的有效性则可能因为帧比较少，统计信息不足而有所降低。考虑到可以结合上下文信息筛选候选视频，我们最终采用了分布直方图的方法。

镜头级别生成的第一步是划分镜头。现有的镜头边界检测算法的精确度相当高，在[17]中提到现有技术对于 CUT 类型的镜头边界检测（即通过一帧完成的镜头转换，而不是渐隐渐显或者是一定时间长度的转场）基本上可以达到 90% 以上的高正确率。而且在这里对镜头边界检测的要求可以进一步降低，只要满足以下两个要求：1) 正确找出嵌入的复制视频的边界；2) 对于复制片段和对原片段的镜头划分相同。第一个要求几乎都能够满足，因为嵌入的片段和被嵌入的视频通常边界清晰。至于第二个要求，在实验中发现除了遇到一些严厉的变换如图 3.5，通常情况下也能够满足。



图3.5 使得镜头划分不同的严厉变换。左边的源视频宽和高都缩小为0.5倍得到右边的复制视频，两者的镜头划分不相同。

镜头级别生成的第二步是对镜头内的帧特征进行直方图统计。为了提高分辨率，边界内的帧都没有参与统计，而每个镜头分为两个子镜头进行统计。另外对于一些帧很少的镜头（少于 10 帧），镜头中的帧画面高度相似，帧特征通常一样，统计得到的直方图会在某个维度上数值很大，其他维度数值都很小，这样在查找相似镜头时会找到很多类似的镜头，引入了较大干扰，因此我们将这些镜头去掉不予考虑。帧特征为 0 到 35 之间整数，统计子镜头内各个整数出现的次数并归一化为频率，得到 36 维的直方图，如图 3.6 所示。

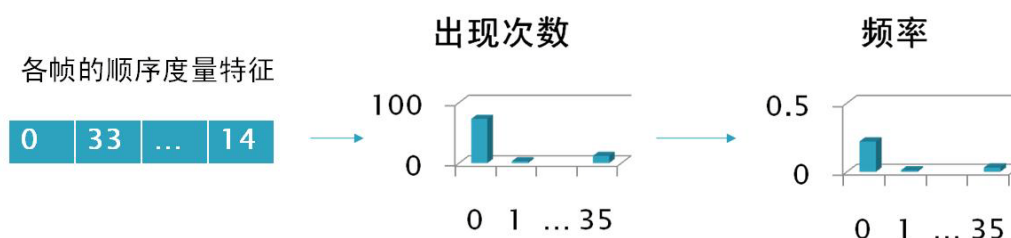


图3.6 统计分布直方图

3.2.3 建立索引和查找相似镜头

在生成了帧级别和镜头级别的特征后，需要查找查询视频的相似镜头（实际上实现时是采用子镜头，但是为了陈述方便，以下都称为镜头）。当数据库较小时，可以使用暴力搜索的方法，但是数据库较大时，需要建立索引，降低搜索的复杂度。建立索引可以离线进行，而大大降低关键的在线搜索的时间。

经过调研，我们决定采用 ANN (Approximate Nearest Neighbor)^[18]索引结构。ANN 将特征所在的空间分割成小块，以 kd-tree 或 bd-tree 的方式组织起来进行搜索，如图 3.7 所示。

ANN 离线预处理的时间复杂度为 $O(dn \log n)$ ，空间复杂度为 $O(dn)$ ，其中 d 为特征维数， n 为特征数目。搜索的时间复杂度为 $O(d(1 + 6d/\epsilon)^d \log n)$ ，其中 ϵ 是误差范围，即算法返回的第 i 个最近邻与查询点的距离，和真正的第 i 个最近邻与查询点的距离，其比例不超过 $(1 + \epsilon)$ 。

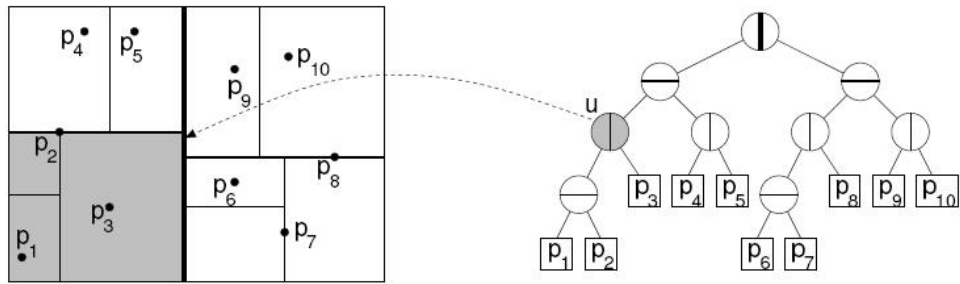


图3.7 ANN索引结构^[18]

3.2.4 筛选候选视频

在搜索到查询视频的相似镜头后，就可以利用这些信息筛选出候选视频。如果数据库中视频 R_i 有多个几乎连续的镜头级特征和查询视频 Q 中某段一一对应，则这些镜头的相似度之和为 R_i 的得分，如图 3.8 所示。所谓几乎连续是指对应序列中允许出现两个对应的镜头特征相似度较小（如图 3.8 中的 D 和 R ），但是不允许有连续的这样的镜头对出现。

得分超过阈值 $2*0.8$ 的视频作为候选视频。

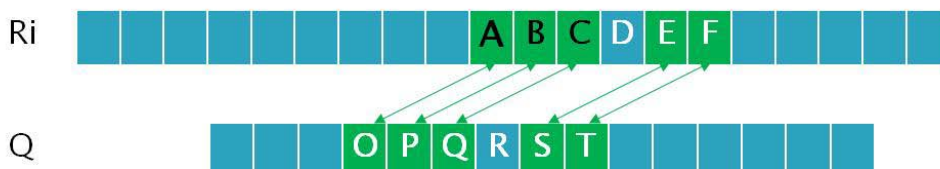


图3.8 筛选候选视频

3.2.5 子序列匹配

得到候选视频后，可以直接将这些候选视频作为最终的检测结果，也可以利用子序列匹配改善局部精确度，并筛去潜在的误判。

子序列匹配是将候选视频和查询中的相似段向两端扩展 3 个镜头，然后将扩展后两段视频进行帧级别的序列匹配，使用动态规划找出其中的最佳匹配子序列。考虑到有可能复制时并没有按照源视频的边界进行裁剪，复制段的开始和结束的镜头并不是源视频中的一个镜头，而只是镜头的一部分，如图 3.9 所示。这时这些镜头将不能匹配上，因此向两端扩展 3 个镜头再进行子序列匹配。

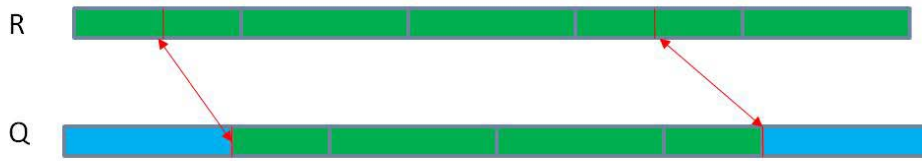


图3.9 复制片段边界不是源视频中的边界

动态规划算法找最佳匹配子序列如下：

$$DP[i][j] = \max \begin{cases} DP[i][j-1] + score[-][om[j]] \\ DP[i-1][j] + score[om[i]][-] \\ DP[i-1][j-1] + score[om[i]][om[j]] \\ 0.0 \end{cases} \quad (3-2)$$

其中 $DP[i][j]$ 表示以 i, j 结束的最佳匹配子序列的得分； $score[x][y]$ 表示特征 x, y 的相似度，即 x, y 匹配的得分； $score[x][-], score[-][x]$ 表示 x 与空帧匹配的得分。计算时可以同时记录 i, j 结束的最佳匹配子序列的起始点，计算结束后找到最大的 $DP[i][j]$ ，即为最佳匹配子序列的得分，并根据记录可知匹配的起始点。

$score[s][t]$ 可以离线计算获得。如图 3.10 所示，首先在 MUSCLE-VCD-2007^[1] 中选取一些视频，分别做各种变换；然后统计变换后特征 s 变为 t 的次数 $f[s][t]$ ，将次数对数化即得到 $score[s][t]$ 。

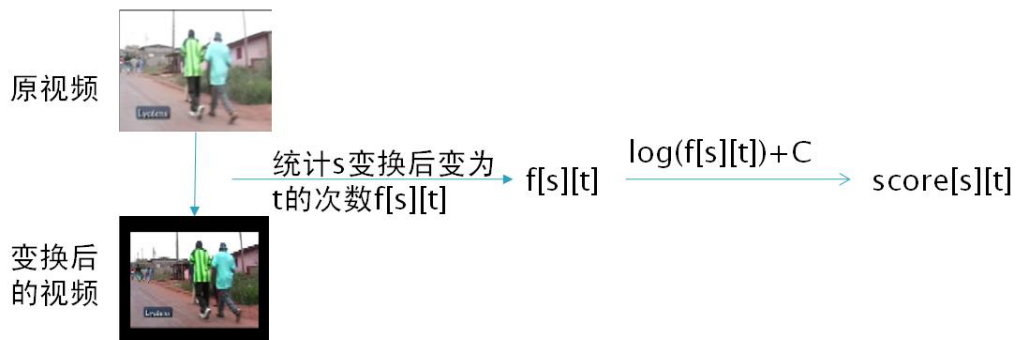


图3.10 $score[s][t]$ 的离线计算

3.3 算法测试

在设计和实现算法之后，我们将其在 MUSCLE-VCD-2007^[1]数据集上进行测试。测试平台为 Pentium(R) 4 CPU 2.80GHz; 2.79GHz, 2.00GB 内存。

3.3.1 数据库介绍

MUSCLE-VCD-2007^[1]是由 CIVR 提供的用于视频复制检测评估的数据集。数据集包括约 100 小时的视频，内容广泛，格式和质量多样，可以很好地反映视频复制检测的应用背景。

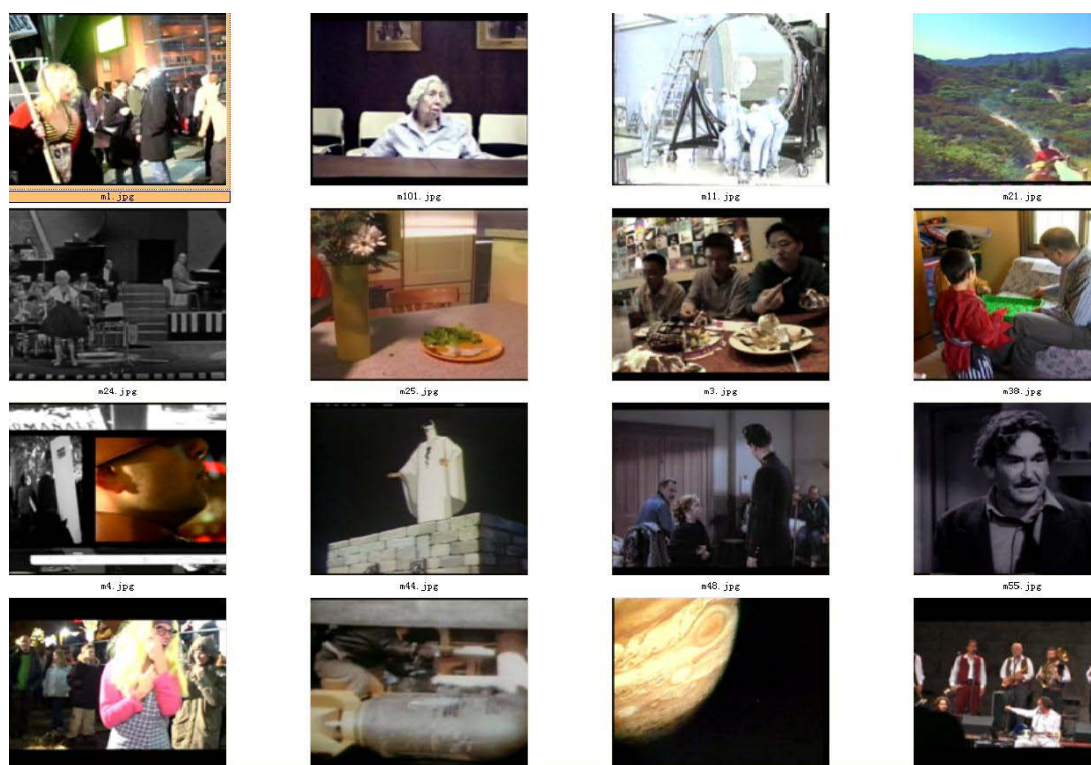


图3.11 MUSCLE-VCD-2007^[1]

数据集还包括两类查询，供使用者评估算法。

第一类查询中有 10 个复制视频，5 个非复制视频，时长从 5 分钟到 40 分钟不等，复制时包括各种变换。具体信息如表 3.1 所示。这类查询有两个评估指标：*Quality* 和查询时间。*Quality* 计算如下：

$$Quality = N_{correct} / N_{queries} \quad (3-3)$$

其中 $N_{correct}$ 指判断正确的查询数目， $N_{queries} = 15$ 是指查询总数。

表3.1 MUSCLE-VCD-2007^[1]中的第一类查询

查询视频	时长	变换
ST1Query1	6:57	color adjustment + Blur
ST1Query2	5:48	not_in_db
ST1Query3	6:18	reencoding + color adjustment + cropping
ST1Query4	5:25	not_in_db
ST1Query5	7:43	reencoding with strong compression
ST1Query6	6:02	frontal camcording + subtitles
ST1Query7	7:00	not_in_db
ST1Query8	8:00	not_in_db
ST1Query9	9:13	colors phase modification + color adjustment
ST1Query10	11:33	non frontal camcording
ST1Query11	11:46	frontal camcording
ST1Query12	14:41	not_in_db
ST1Query13	17:27	flip
ST1Query14	26:19	resizing + subtitles
ST1Query15	42:59	resizing

第二类查询中有 3 个视频，每个 15 分钟，每个视频为非复制视频中嵌入 6 至 8 段复制片段。复制片段的时长从 20 秒到 120 秒不等。这一类查询有三个衡量指标： $QualitySegment$ 、查询时间和 $QualityFrame$ 。 $QualitySegment$ 和 $QualityFrame$ 计算如下：

$$QualitySegment = (N_{correct} - FalseAlarm) / N_{segment} \quad (3-4)$$

$$QualityFrame = 1 - N_{mis} / N_{frames} \quad (3-5)$$

其中 $N_{correct}$ 是判断正常的复制片段的数目， $FalseAlarm$ 是误判为复制片段的视频数， $N_{segment} = 21$ 是复制片段的总数； N_{mis} 是没能查出的复制帧或者误判为复制的帧的总数， N_{frames} 是复制帧的总数，由于提供的答案的局部精确度精确到秒，所以 N_{mis} 和 N_{frames} 在计算中用秒数来代替。

CIVR在<http://www-rocq.inria.fr/imedia/civr-bench/Results.html>上给出了 2007 年参加评估的各个单位的测试结果，可以作为对比算法集。对比集第一类查询结果和第二类查询结果分别如图 3.12 和图 3.13 所示。

Team - run	ST1 score	ST1 search time
advestigo	0,86	64 min
Chinese academy of sciences - 1	0,46	41 min
Chinese academy of sciences - 2	0,53	14 min
City university of Hong Kong	0,66	45 min
IBM - 1	0,86	44 min
IBM - 2	0,73	68 min
IBM - 3	0,8	99 min

图3.12 对比集算法第一类查询的测试结果

Team	ST2 Recall	ST2 search time
Considering segments		
City university of Hong Kong	0,86	35 min
advestigo	0,33	33 min
Considering frames		
City university of Hong Kong	0,76	35 min
advestigo	0,17	35 min

图3.13 对比集算法第二类查询的测试结果

3.3.2 离线处理工作

离线处理工作包括三部分。

帧级别特征提取需要 5.5 小时，速度约 1.5Mb/s。

镜头级特征生成共需要 5 分钟，共生成 51142 个镜头级特征。

建立 ANN 索引需要 20 秒。

3.3.3 第一类查询测试结果

第一类查询的测试结果如下：

$Quality = 12/15 = 0.8$;

查询时间 27 分钟，其中 22 分钟用于生成查询视频的特征。

与对比集中的结果（如图 3.12 所示）相比，可以看出我们的算法时间复杂度低，性能也比较好。

3.3.4 第二类查询测试结果

第二类查询的测试结果如下：

$$QualitySegment = (15 - 1) / 21 = 0.67 ;$$

时间为 8 分钟，其中 6 分钟用于提取查询视频的特征；

$$QualityFrame = 1 - 500 / 1006 = 0.59 。$$

与对比集中的结果（如图 3.13 所示）相比，可以看出我们的算法时间复杂度很低，但是性能却有欠缺。*QualityFrame* 很低，但是我们注意到如果复制片段判断正确，则其局部精确度在 1 秒以内，也就是说，算法的局部定位确实比较好，但是由于视频级性能太差，导致 *QualityFrame* 较低。

3.3.5 结果讨论

与对比集算法的结果（图 3.12 和图 3.13）相比，我们的这个算法的结果有两个明显的特点：一是查询时间明显比其他算法的短；二是帧级别精度高但是召回率低。

第一个特点是因为采用了顺序度量这种比较简单的特征。实际上，查询时的大部分时间是用于提取视频的特征，如在第一类测试中，总时间 27 分钟，而提取特征（包括帧级别和镜头级别特征）就用去 22 分钟。而提取顺序度量是相对比较快的，即便还需要通过统计直方图的方法将各帧的特征压缩为镜头级特征，时间也要对比集算法的短。由于提取特征的时间与查询视频总长度成正比，可以预见即便查询增多，这个算法也保持着较大的时间优势。剩下的时间主要用于搜索相似镜头。由于采用了 ANN 索引结构，所以即使是处理更大的数据集（如超过 10000 小时），这部分时间也不会急剧增加。

第二个特点中，精度高（第一类查询中为 90%，第二类查询中为 94%）主要是在筛选候选视频之后还可以利用子序列匹配将误判剔除，从而使得误判的数目非常少。但是评估指标并没有超过对比集算法中最优者，是因为召回率偏低（第一类查询中为 80%，第二类查询中为 71%）。经过检查，发现没有检测到的复制视频都有共同的特点：复制时的变换都带有画面的几何变换。详细地说，施加于这些视频的变换或者是放缩剪裁、或者是画面偏移、或者是水平翻转。而这

些变换恰好是顺序度量特征不能保持鲁棒性的变换，说明召回率偏低是因为顺序度量缺乏鲁棒性所致。筛选候选视频的结果也证明了这一点：没有检测到的复制视频在筛选候选视频时就没有找到，召回率低与后面的子序列匹配步骤无关。因此要提高性能，必须考虑使用更鲁棒的特征。

除了以上两个特点之外，我们还需要看看局部精确度是否如期得到改善。第二类查询中反映局部精确度的指标 *QualityFrame* 显得很低，但是这是因为召回率低造成的，而召回率低是与子序列匹配无关的。为了评估子序列匹配的有效性，我们需要比较在检测复制视频正确时，检测到正确帧占总帧数的比例。CIVR 提供的结果中没有这个指标，我们用 $QualityFrame / QualitySegment$ 来近似。对比集算法中最好结果为 $0.76/0.86=88\%$ ，而我们的算法为 $0.59/0.67=88\%$ ，两者相近，说明子序列匹配确实改善了局部精确度。

3.4 算法总结

基于顺序度量的直方图匹配算法的突出优点在于它的时间复杂度低。与对比集算法相比，在第一类查询中，这个算法的查询时间约为其他算法的一半，而在第二类查询中，优势更加明显，时间仅为其他算法的 $1/4$ 。需要说明的是，在第一类中的时间优势不如在第二类中明显，是因为我们测试第一类时也执行了子序列匹配步骤，而对比集算法在处理第一类查询时可能对局部精确度并不关注，所以时间要比处理第二类查询快。时间复杂度低的特点使得这个算法在对时间要求比较严格或者召回率要求不是太严格的应用中有较大优势。比如在数据库的冗余去除中，并没有要求去除所有的冗余成分，而要求能够快速检测出新添加的视频是否带有复制片段并进行后续处理，以便响应接下来对数据库的请求。这种情况下，基于顺序度量的直方图匹配算法显得比较合适。

这个算法的优点来自于使用的特征，缺点同样也来自于特征。由于顺序度量特征对画面偏移、放缩剪裁和水平翻转等几何变换鲁棒性差，对源视频和复制视频提取的特征相似度小，无论后续步骤做得多好，也难以准确地检测出这些变换后的复制视频。为了进一步提高性能，需要用到其他更强的特征。调研时已经发现，局部特征是目前比较好的特征，但是其复杂度要高得多，因此换用局部特征需要处理的主要问题将是如何降低复杂度而保持较高的性能。

这个算法框架的总体设计思路是首先筛选候选视频，然后进行局部精确度改善。对测试结果的分析反映了这个思路的可行性。筛选候选视频后，改善局部精

确度较为容易，而且候选视频不多，如果精度高，则候选视频的数目大约等于正确复制视频的数目，因此可以使用较复杂的方法而不用担心时间复杂度剧增。但是测试结果同样也反映了这个思路的一个瓶颈：如果筛选候选视频这一步做得不好，那么即使后续步骤确实能改善局部精确度，也无法改变总体性能偏低的事实。所以下一步工作中将使用更强的局部特征，以提高测试的评估指标。

第4章 基于SURF的图片序列匹配算法

4.1 算法概述

在上一章中，我们介绍和测试了基于顺序度量的直方图匹配算法。这个算法虽然速度快，但是评估指标偏低，原因是顺序度量特征对某些变换的鲁棒性差，导致复制片段召回率偏低。为了提高性能，我们考虑使用更强的局部特征。

使用局部特征的主要困难在于其复杂度高。在图像处理领域，提取一张图片的局部特征，其时间约需 1 秒。如果对视频中的每一帧都提取局部特征，以视频中每秒有 25 帧计算，离线处理时提取 100 小时数据集的特征需要 2500 小时；在线查询时提取一个 15 分钟视频的特征需要 375 分钟。这样长的时间是难以接受的。因此不可能提取每一帧的局部特征，我们最终决定简单地提取每个镜头中部一帧的局部特征。

使用局部特征的第二个困难在于如何计算两帧图像局部特征的相似度。与顺序度量或者其分布直方图不同，局部特征的维数通常是不固定的。一般地，一帧图像的局部特征可以看成是 n 个 m 维向量，其中 n 不固定， m 固定。这样，不能直接计算两幅图像局部特征的距离，也就不能使用 ANN 索引结构，因为 ANN 索引需要计算特征之间的 Minkowski 距离。因此，需要找到计算两帧图像局部特征的相似度的有效方法，同时不能减弱其性能。

本章将介绍基于 SURF 的图片序列匹配算法。算法的流程如图 4.1 所示。

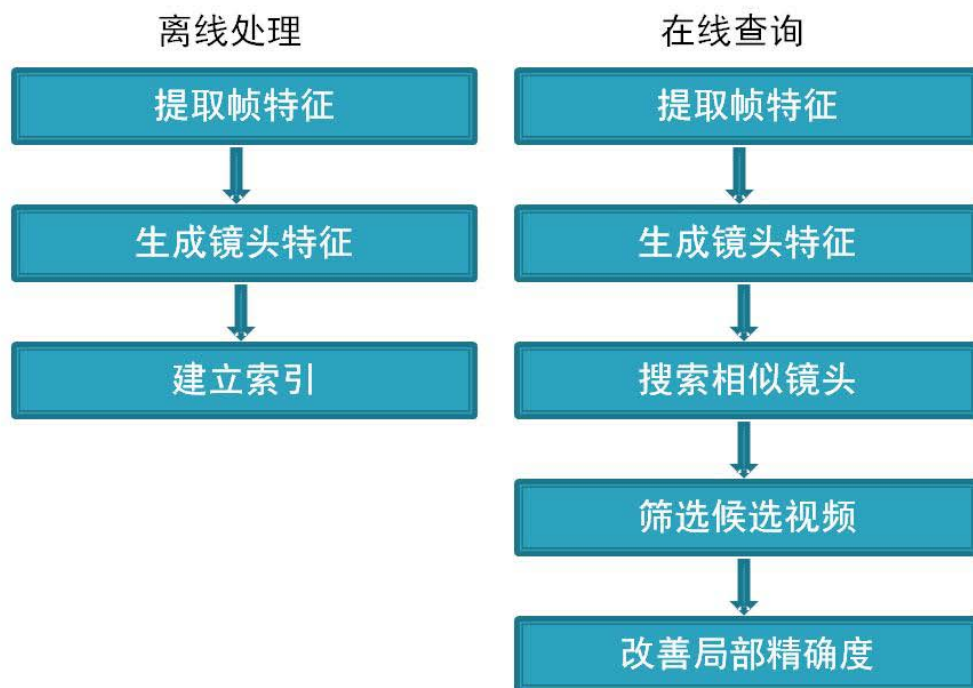


图4.1 基于SURF的图片序列匹配算法的流程图

4.2 算法介绍

下面对算法的各个步骤进行详细介绍。

4.2.1 局部特征提取

由于局部特征的高复杂度，我们选择每个镜头只提取其中一帧的局部特征。所以提取局部特征的第一步是划分镜头。这一步的工作和 3.2.2 中划分镜头的工作是一样的，所以不再详细说明。同样，为了排除干扰，镜头边界和帧数太少（少于 10 帧）的镜头都不予考虑。

第二步是选择镜头帧序列中点的那一帧，并提取其局部特征。在之前，需要选择提取哪种局部特征。参考文献[3]中比较了三个采用局部特征的视频复制检测算法，其中两个[11][15]采用 Harris 关键点的一个改进版本作为检测器，并在关键点周围选取 4 个点进行微分得到特征；第三个算法采用[16]中提出的时空兴趣点（Space-time interest point）。后者的时间复杂度太高，相比而言，前者更合适。

但是 Harris 兴趣点是已经提出比较久的特征，我们考虑使用在图像处理领域一些新的特征。

经过调研，我们发现 SURF^[19]特征和 SIFT^[20]特征比较合适。SURF 的性能较 SIFT 稍好，更重要的是，提取 SURF 的时间比 SIFT 要少，因此我们选择提取 SURF 特征。

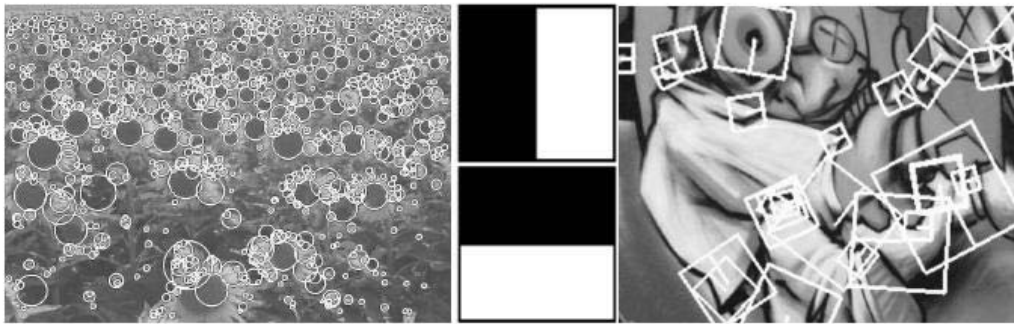


Fig. 2. Left: Detected interest points for a Sunflower field. This kind of scenes shows clearly the nature of the features from Hessian-based detectors. Middle: Haar wavelet types used for SURF. Right: Detail of the Graffiti scene showing the size of the descriptor window at different scales.

图4.2 SURF特征示例^[19]

提取一幅图像的 SURF 特征一般需要检测 300 到 500 个点（如图 4.3 所示），在每个点附近的局部区域提取一个 64 维的向量，整个过程需 0.2 到 0.5 秒。

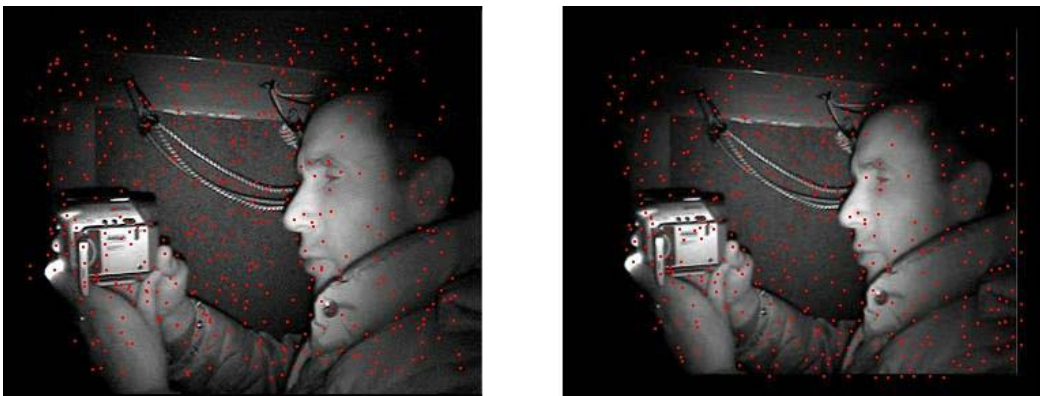


图4.3 放缩前后帧中检测到的SURF点

由于特征的重要作用，下面对 SURF 特征进行进一步介绍。

SURF 特征原来的用途是图像处理的物体识别，对图像变换有很强的鲁棒性。如图 4.4 所示，即便对雕像的拍摄角度不同、光照等画面背景因素不同，两幅图像的 **SURF** 特征也是相近的。这使得在本算法中，我们可以简单地选择位于镜头帧序列中点的帧，不用过多考虑帧偏移的问题。所谓帧偏移，是指在源镜头中提取的帧和在复制镜头中提取的帧在时间上有微小的偏移，如图 4.5 所示。比较图 4.4 和图 4.5，可以发现两者的相似性，即 **SURF** 特征在帧偏移时也仍然有很好的鲁棒性，因此我们在镜头中选取帧时不必采用复杂的方法，避免了复杂度的增高。



图4.4 物体识别示例^[19]

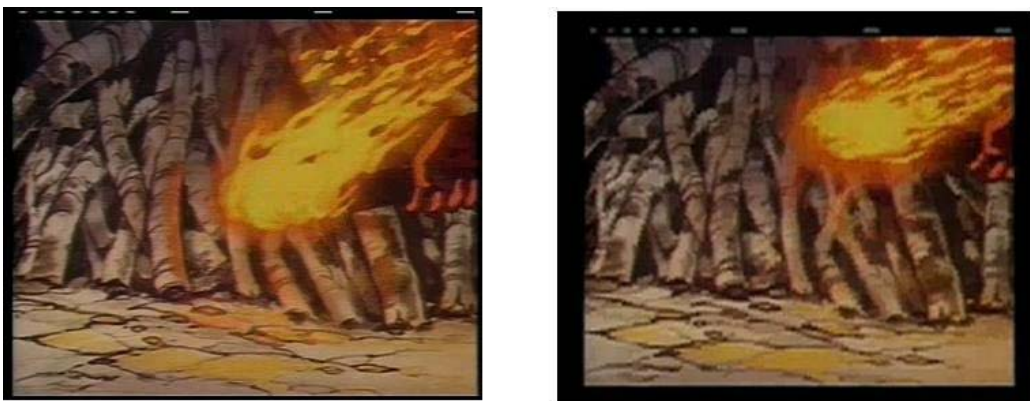


图4.5 帧偏移

虽然 **SURF** 特征在物体识别中表现出很好的性能，但是它毕竟不是为视频复制检测量身定做的，可能并不适合用于检测。比如在物体识别中，对于每类物体，可以基于正负例的 **SURF** 特征训练出一个分类器；查询时，将各类的分类器应用于查询图像，并将图像划分为得分最高的一类。但是在视频复制检测任务中，视

频是图片序列，类的确定是困难的，所以不能直接应用这样的思路。为了保证其有效性，我们需要检验 SURF 在视频复制检测任务中的性能。

考虑两帧图像 Q 和 R ，其 SURF 特征可以看成多个 64 维的点。对于 Q 中每个点 Q_i ，计算与 R 中每个点 R_j 的欧式距离 $d[i][j]$ ，如果存在 $d[i][j]$ 小于阈值（设为 0.3），则认为 Q_i 在 R 中再现。 Q 和 R 的相似度定义为

$$sim(Q, R) = \frac{\#\{Q_i \mid Q_i \text{ 在 } R \text{ 中再现}\}}{\#\{Q_i\}} \quad (4-1)$$

其中 $\#S$ 表示集合 S 中元素的个数。如图 4.6 所示，图像的横轴为 4 个源视频的帧，纵轴为对应的复制视频的帧（施加的变换包括颜色调整、水平翻转、画面偏移和放缩剪裁等），图中各点的亮度与两帧相似度成正比。理想情况是黑色背景上有一条白色对角线。可以看出，图像跟理想情况是很接近的，这表明 SURF 特征性能优越。

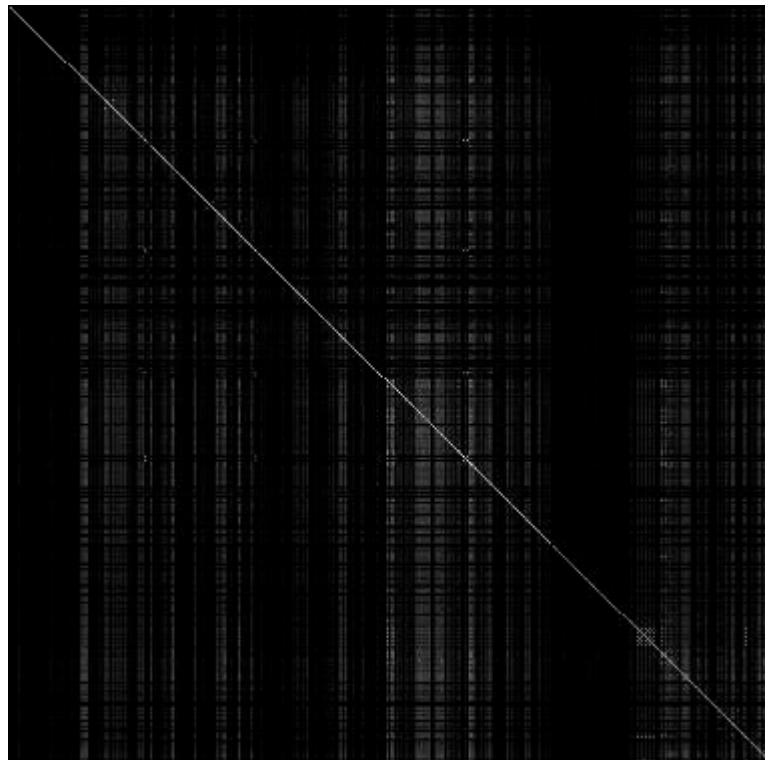


图4.6 SURF相似度图像

4.2.2 统计分布直方图

得到图像的 SURF 特征后,我们需要确定如何计算两帧 SURF 特征的相似度。虽然在检验 SURF 性能时使用的相似度定义可以很好地比较两帧图像的 SURF 特征,但是这种定义计算复杂度很高,不适合用于查询。[8]在特征空间中选取基向量;将每个帧的特征向量投影到与其最近的基向量,在此基向量上累加权重;归一化各个基向量上的权重作为视频的特征,如图 4.7 所示。这启发我们选取合适的基向量,对帧中各个 SURF 点做类似的处理,最终得到一个固定维数(基向量个数)的向量。这实际上是基向量集合上的分布直方图。

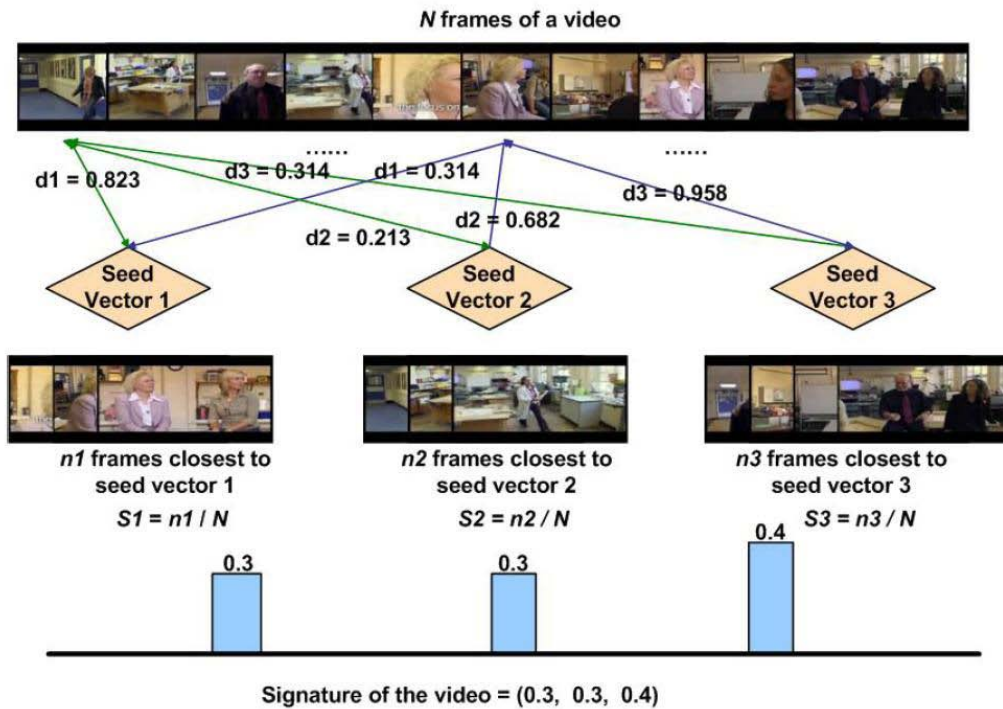


图4.7 基向量上的分布直方图^[8]

统计分布直方图有两个问题需要解决。第一个问题是选取多少个基向量。基向量的个数决定了得到的特征的维数。一般而言,维数越高,表现力越强,性能越好,同时复杂度越高。因此这是一个对复杂度和性能进行权衡的问题。我们选取了从 64、256、1024、4096 共 4 个维数进行测试。

另一个问题是如何选取基向量。可以简单地指定基向量的值,如直接使用 SURF 点的自然基。其距离图像如图所示,图中点的亮度跟两帧图像直方图之间的

相似度成正比。从图中看出可以其分辨性和鲁棒性都很差。后续的实验也证实了这样做会使得系统的性能低下。因此我们参考[9]中方法，使用聚类的方法获得基向量集合。在 SURF 的物体识别应用中也使用聚类的方法获得基向量并统计直方图，不过其目的是为了增强识别的鲁棒性，而我们的主要目的是较快地进行图像相似度计算。在聚类基向量数目为 64 时直方图的相似度图像如图 4.9 所示，比图 4.8 好，但比图 4.6 差。

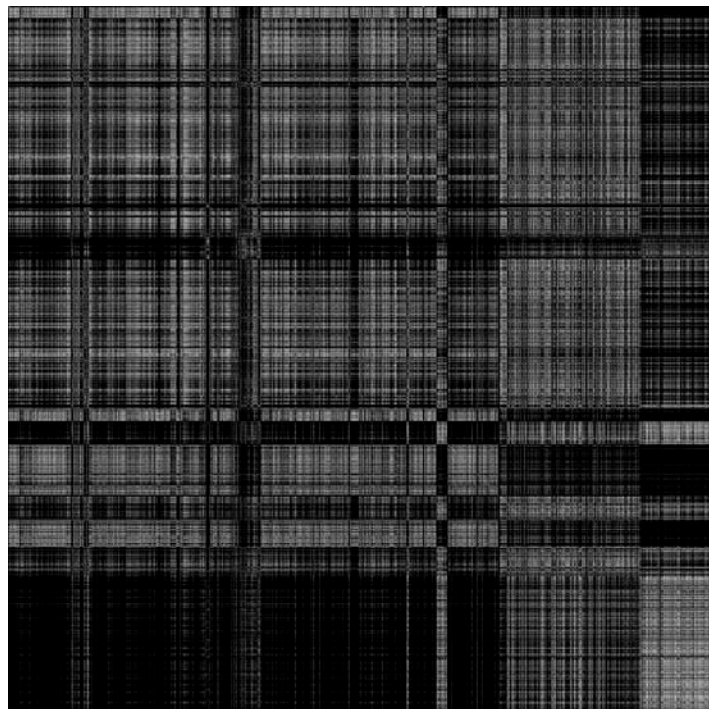


图4.8 自然基上分布直方图的相似度图像

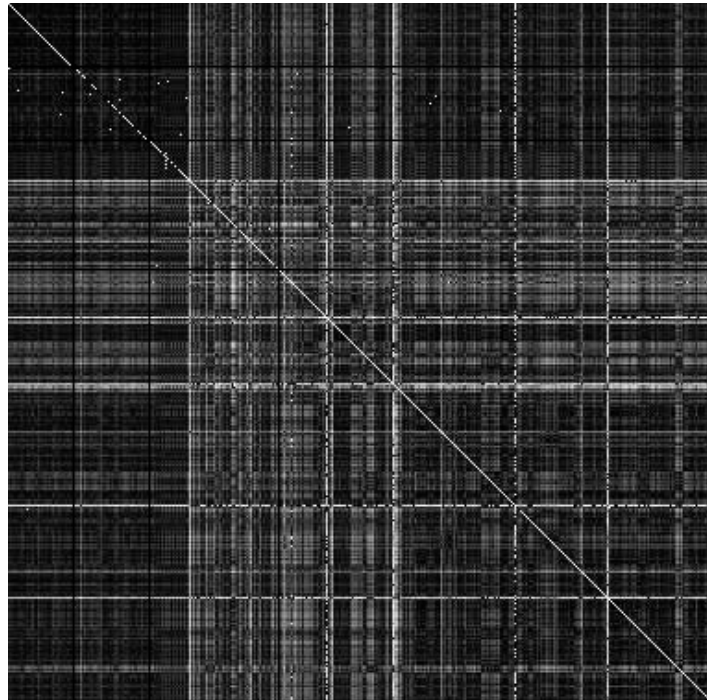


图4.9 64个聚类基向量上直方图的相似度图像

4.2.3 建立索引和查找相似镜头

这一步和 3.2.3 中一样，采用了 ANN 索引结构，因此在这里不再详细说明。有一点不同的是，在查找相似镜头的时候，在上一章中我们采用了固定半径的搜索方式，即查找距离小于指定阈值的所有点。但是在测试本算法时发现，查询图片的直方图特征离其近邻的距离起伏较大，如果指定阈值，则有些图片找到太多的近邻，而另外一些则找不到近邻。因此我们改用固定近邻数目的搜索方式，即对于每个查询，都返回相同数目的近邻。经过后续实验的测试，发现返回近邻太多时，干扰比较大，所以我们选择了只返回距离最小的最近邻。

4.2.4 筛选候选视频

相比其顺序度量，SURF 特征分辨性更强，而且我们选择了返回很少数目的近邻，因此在筛选候选视频时上一章的方法并不合适，需要设计新的方法。

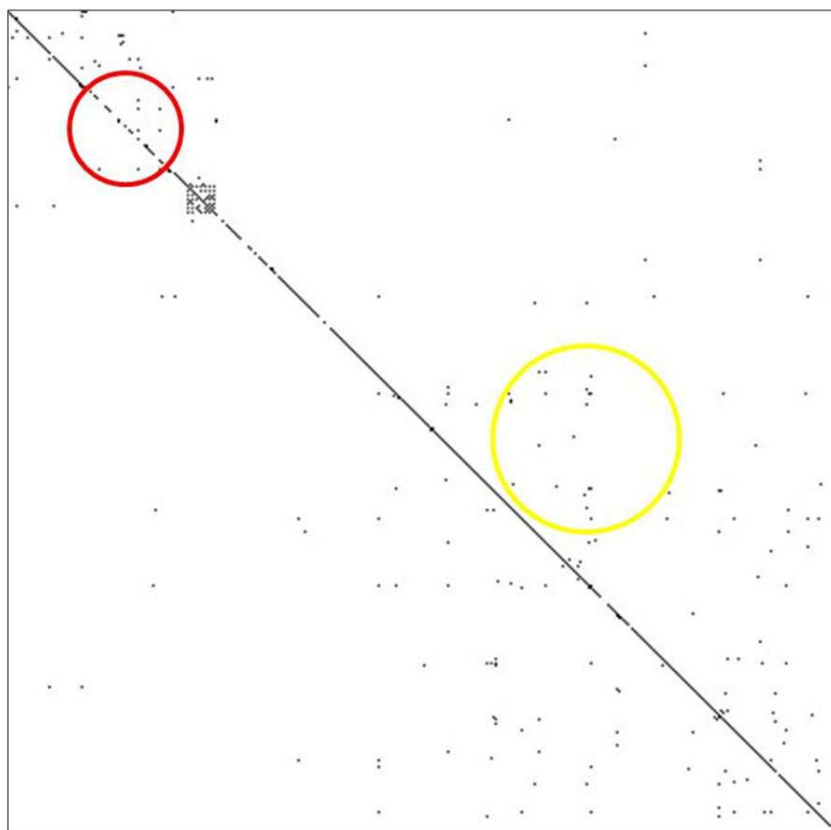


图4.10 近邻图中的断点和干扰。直方图维数为64。

筛选遇到的第一个问题是断点和干扰很多，即在复制视频的帧序列中，会出现连续多个镜头在搜索时没有查找到源视频的对应镜头。如图 4.10 所示，图中纵轴为查询视频的帧，横轴为对应源视频的帧，黑点表示搜索时返回的最近邻。红色圆圈中有明显的断点；不在对角线上的点都是干扰点，如黄色圆圈中的点。在严厉的变化下，这个问题更加突出，增加维数虽然能够减缓，但是问题仍然比较严重，如图 4.11 所示。

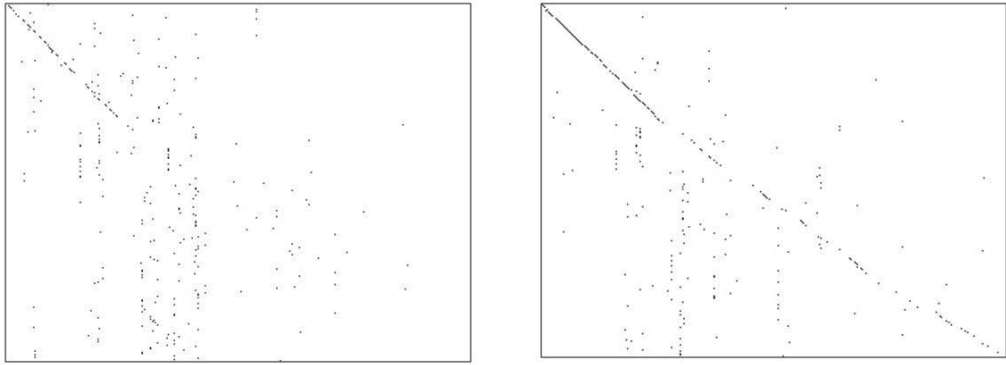


图4.11 严厉变换下的近邻图像。左图为64维，右图为1024维。

第二个问题是镜头划分时对源片段和复制片段的镜头划分不一致，如图 4.12 所示。在查询视频中多划分了一个镜头，这就造成对应镜头的偏移不一致，反映在最近邻图中会出现线段的错位，如图 4.13 中红色圆圈中的一段。



图4.12 镜头划分不一致示例

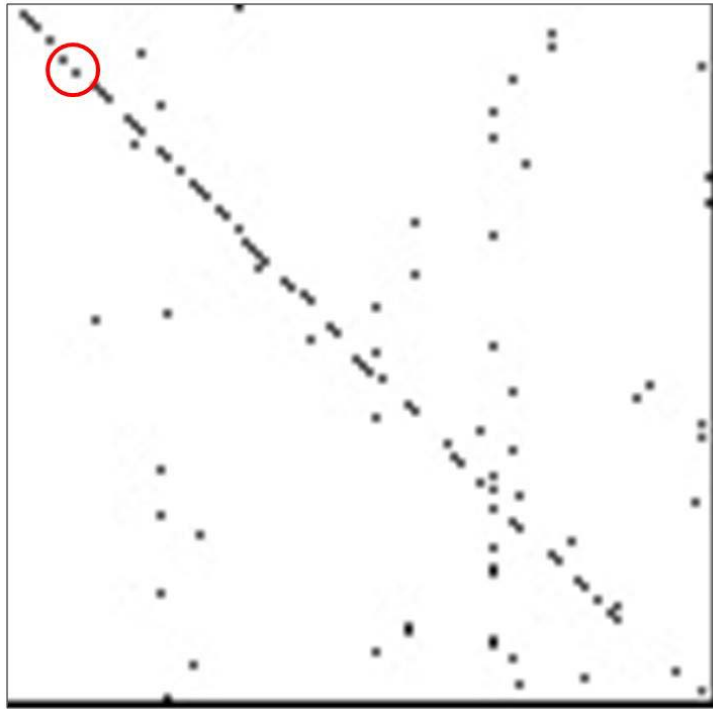


图4.13 近邻图中的错位

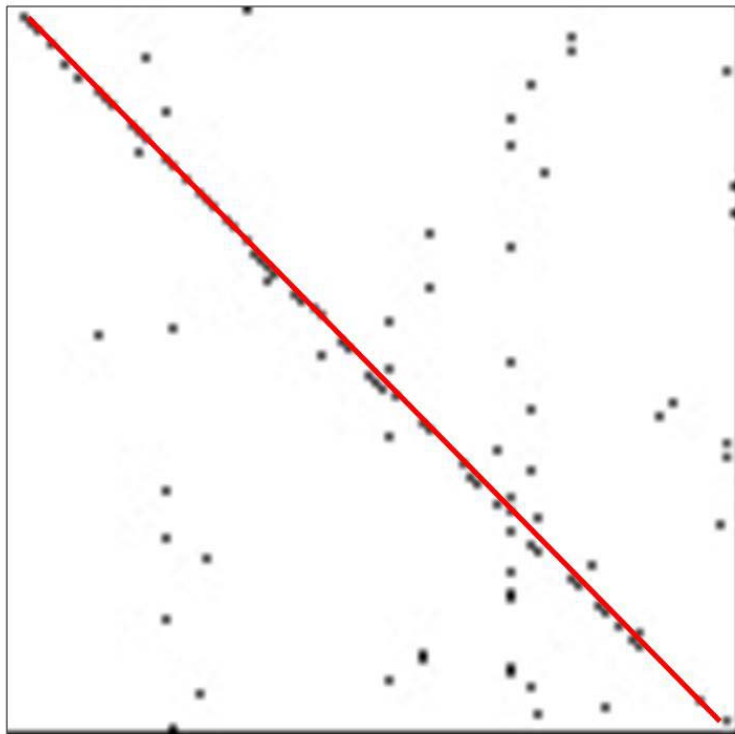


图4.14 需要找到的直线

筛选候选视频实际上是需要最近邻图中找到可能存在的近似直线 $y = x + b$ 。比如需要在图 4.13 中找到对角线，如图 4.14。

比较源视频和非源视频的最近邻图像，可以发现非源视频的最近邻图像基本上是随机分布，而源视频中则形成一条直线，因此我们设计了以下的图片序列匹配方法。

首先去除一些明显不正确的最近邻点。如图 4.15 中的两种情况，至少有一个近邻点是不正确的匹配，我们将这些近邻点都去掉。

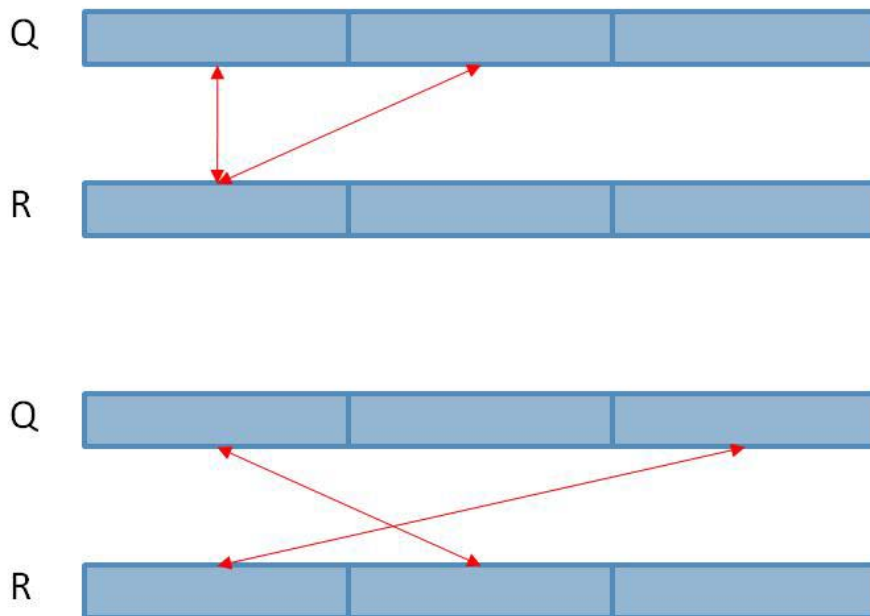


图4.15 不正确的镜头匹配

接下来估计直线 $y = x + b$ 中的参数 b 。计算最近邻图中所有点 (x, y) 的 $y - x$ 值，求平均值作为 b 的近似 b' 。预期干扰点的影响将相互抵消， b' 值比较准确。

然后将 $y - x$ 偏离 b' 太多 ($|y - x - b'| > 10$) 的点去掉，剩下的点视为正确匹配的最近邻。所有这些匹配的相似度之和为视频的得分，阈值超过 $2 * 0.8$ 的作为候选视频。

这个步骤的伪代码如下：


```
Screening(ANNpoints matchPts)
```

```
/* eliminate invalid points */
```

```
for i = 1 to matchPts.size
```

```
  for j = 1 to i-1
```

```
    if( (matchPts[i].y-matchPts[j].y)/(matchPts[i].x-matchPts[j].x)<=0 )
```

```
      matchPts[i].x = matchPts[i].y = -1; /* eliminate point i */
```

```
      matchPts[j].x = matchPts[j].y = -1; /* eliminate point j */
```

```
/* estimate parameter b */
```

```
sum = 0.0;
```

```
num = 0;
```

```
for i = 1 to matchPts.size
```

```
  if( matchPts[i].x != -1 )
```

```
    sum += matchPts[i].y - matchPts[i].x;
```

```
    num ++;
```

```
b = sum/num;
```

```
/* eliminate invalid points */
```

```
MAX_OFFSET = 10;
```

```
for i = 1 to matchPts.size
```

```
  if( matchPts[i].x != -1 && |matchPts[i].y-matchPts[i].x-b|>MAX_OFFSET )
```

```
    matchPts[i].x = matchPts[i].y = -1; /* eliminate point i*/
```

上面是一般情况。在变换比较温和的情况下，复制视频中有连续的镜头和查询视频中的一一对应。我们可以直接用这些镜头对应的点的 $y - x$ 值作为 b 的近似 b' ，这个估计值更加准确。

4.2.5 改善局部精确度

与上一章中的算法一样，这一步是可选的。不同的是，为了降低复杂度，本算法提取的特征在时间轴上非常稀疏，密度为一个镜头提取一帧的特征，而改善局部精确度需要比较密集的特征，因此不能使用上一章的动态规范算法。

在实验中我们发现筛选候选视频时，对复制片段的定位已经比较准确，因此可以只在定位边界处抽取密集的特征进行比较，从而避免了复杂度剧增的问题。进一步检查筛选结果发现，对复制片段的定位有偏差，通常是因为靠近复制片段边界的镜头未能检测出，因此只需要向两端扩展，抽取特征进行对比即可。

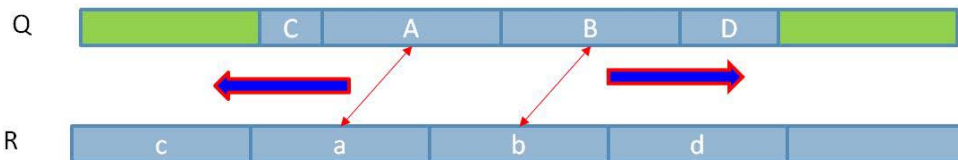


图4.16 改善局部精确度。其中已匹配A和B，向两端扩展匹配C和D以改善局部精确度。

4.3 算法测试

我们使用 MUSCLE-VCD-2007^[1]数据集对算法进行测试。测试平台为 Pentium(R) 4 CPU 2.80GHz; 2.79GHz, 2.00GB 内存。

测试第一类查询时，没有执行改善局部精确度的步骤。测试第二类查询时，对不执行和执行改善局部精确度步骤都做了统计。

4.3.1 离线处理工作

离线处理工作包括三部分。

聚类生成基向量集合所需时间与基向量个数成正比，生成 1024 个基向量需要约 2 小时；

提取 SURF 特征并生成直方图需要约 6 小时，共 25571 个直方图；

建立 ANN 索引时间少于 1 分钟。

4.3.2 第一类查询测试结果

第一类查询的结果如表 4.1 所示，可知算法的性能非常好，在维数比较高的情况下，性能和时间超过了对比集算法结果（如图 3.12 中所示）中最优者的性能和时间。

表4.1 第一类查询测试结果

Dimension	Quality	Time(min)
64	0.8	30
256	0.80	33
1024	0.93	40
4096	1.00	65

4.3.3 第二类查询测试结果

第二类查询结果如表 4.2 和表 4.3 所示。结果显示 *QualitySegment* 对比集算法结果（如图 3.13 所示）中的最优者稍差，但是 *QualityFrame* 比较好，时间复杂度也较低。由表 4.2 可知，不执行局部精确度改善时，算法的时间优势很明显，而且局部精确度也已经比较好。而从表 4.3 可知，执行局部精确度改善后，*QualityFrame* 进一步提高。如果比较 *QualityFrame / QualitySegment*，可以看出本算法明显优于对比集算法。

表4.2 第二类查询测试结果（不执行局部精确度改善）

Dimension	QualitySegment	QualityFrame	Time(min)
64	0.72	0.61	8
256	0.72	0.61	11
1024	0.81	0.70	15
4096	0.72	0.63	24

表4.3 第二类查询测试结果（执行局部精确度改善）

Dimension	QualitySegment	QualityFrame	Time(min)
64	0.72	0.68	52
256	0.72	0.68	54
1024	0.81	0.79	60
4096	0.72	0.71	75

4.3.4 结果讨论

与对比集算法相比，本算法表现了优越的性能和较低的复杂度。

在第一类查询测试结果中，我们注意到随着维数的增大，评估指标 *Quality* 也逐渐增大。在 SURF 的物体识别应用中，也有类似的趋势，但是当维数很大时，其性能将会下降，因为物体识别中同类物体的图片可能差别比较大，过于精细的空间划分会导致同类物体特征差异增大，从而降低了鲁棒性。但是在视频复制检测中，源帧和复制帧之间的差别不会非常大，因此可以推测可以使用更高维数的直方图来提高性能。随着维数增大，统计直方图的时间将增多，不过占总时间大部分的提取特征时间并不会增多，因此使用一万维左右的直方图是可行的。

在第二类查询中，不执行局部精确度时时间优势非常明显，复制片段判断正确时的局部精确度也和已有算法中最优者的持平，所以能够满足现在大多数应用的需要，尤其是适用于时间要求较高的应用。执行局部精确度改善能使得局部定位更加准确，*QualityFrame / QualitySegment* 能达到约 0.95 的水平，适用于对局部精确度要求很高的应用。如果执行这一步，时间增加约一倍，显得稍微多了些。但是注意到这一步的复杂度与数据库大小无关，近似于与查询视频中的复制片段数目成正比，因此在查询视频不变时，即便数据库急剧增大（如达到一万小时），这一步的时间仍然为 20 分钟左右，因此应该说它对算法总的的时间复杂度影响较小。

测试结果的精度很高，基本没有出现误判，而且非源视频的得分非常低（通常为 0，最高者为 0.9，即匹配到个镜头），说明特征的高分辨率和筛选算法的有效性。相比之下，召回率显得比较低。查看未能检测到的视频，发现施加的变换通常是与放缩相关（放缩剪裁、camcoding）。其近邻图中断点和干扰严重，说明 SURF 特征的直方图对此类变换的鲁棒性较差。值得注意的是，随着维数的增大，一些这样的视频能被检测出来，说明问题主要不在于 SURF 特征本身，而在于统计直方图的方法。另外近邻图中也有比较多的错位，增大维数也不能改善，说明这是镜头划分不一致造成的。

4.4 算法总结

基于 SURF 的图片序列匹配算法性能优越，时间复杂度较低，而且局部精确度明显优于对比集算法，基本达到了预期目标。

算法的优越性能主要来自于性能较强的 SURF 特征。可以推测使用其他分辨性和鲁棒性较强的局部特征也可以用于视频复制检测的任务,与使用 SURF 一样,能使得系统表现出优越的性能。但是局部特征高复杂度的缺陷也很明显,因此本算法的成功在于控制了使用 SURF 特征带来的高复杂度。为了做得这一点,不能提取每一帧的特征,只能提取在时间轴上比较稀疏的特征,如每个镜头只提取一帧的特征。这种做法会带来一些问题。比如如何选择镜头中的这一帧。我们注意到 SURF 特征对于帧偏移的鲁棒性,因而使用了简单的选择方法,避免复杂度进一步提高。在计算两帧图像 SURF 特征的相似性时,我们使用了统计直方图的方法,也是为了避免直接比较高复杂度。而主要问题出现在筛选候选视频时:断点、干扰和错位都使得筛选工作比较困难。由于干扰点随机分布,我们预期在估计匹配直线 $y = x + b$ 中的参数 b 时其影响将相互抵消,因此得到比较准确的估计值 b' ,再利用 b' 清除干扰点,这样就完成了模糊匹配。这种方法简单快捷,性能也非常好。

算法的另一个优点是高局部精确度。实际上,由于特征性能优越、直方图的模糊匹配也较好,即便不执行改善局部精确度的步骤,结果也具有很好的局部定位。我们也可以通过向两端扩展的方式进一步提高局部精确度,将指标 *QualityFrame/QualitySegment* 提高到约 0.95 的水平,而时间并没有急剧增加。这表明我们改进局部精确度的想法是可行的,而先筛选候选视频再改善局部精确度的总体思路是有效的。

算法的缺陷在于相比而言,召回率偏低。一个原因是 SURF 特征本身对于放缩变换的鲁棒性比其他变换(如水平翻转等)的鲁棒性要差,不过依然能够保持较好的不变性,但是压缩为直方图特征后,对放缩变换的鲁棒性进一步被削弱,导致一些复制视频没有被检测出来。这可以通过提高维数来改善,但复杂度也会相应地提高。另一个原因是镜头划分也缺乏鲁棒性。现在使用的镜头划分算法着重于划分的准确性,这会带来源视频和复制视频镜头划分不一致的问题。比如在比较严厉的缩小剪裁变换后,由于画面中很大一部分区域为黑色边框,帧和帧之间画面变动小,源视频中的多个镜头在复制视频中可能会被视为一个镜头。实际上在视频复制检测任务中,只需要满足检测到复制片段边界、对变换有鲁棒性两点要求即可,划分是否准确并不是最重要的,因此有可能通过修改镜头划分算法使之更适用于视频复制检测任务。还有一个可能的原因是,如果复制片段比较短,划分的镜头只有很少的几个,同时复制的变换比较严厉或者源片段和复制片段的镜头划分不一致,导致搜索最近邻时并没有找到正确的对应镜头,那么筛选候选

视频时很有可能将这个片段去掉。处理复制片段比较短的情况一方面需要改善特征和镜头划分的鲁棒性；另一方面需要每个镜头提取多帧的特征，但是这会明显增加处理的时间。

第5章 总结与展望

5.1 已完成的工作

首先我们对基于内容的视频复制检测这个题目进行了调研，总结了现有的相关工作，包括使用的特征、搜索方法等（第2章）。

注意到改善局部精确度正成为视频复制检测中的一个重要的研究方向，而现有算法并没有太多关注，因此我们考虑继续提高性能和复杂度，同时改善局部精确度。为此，我们提出了先筛选候选视频，然后进行局部精确度改善的总体思路。

在第3章中，我们提出了基于顺序度量的直方图匹配算法。该算法使用顺序度量作为帧级别特征，并利用统计直方图的方法得到镜头级别的特征；利用后者筛选候选视频，进而利用前者改善局部精确度。在 MUSCLE-VCD-2007^[1]上的测试表明其时间优势明显，而且局部精确度确实得到了改善，但是限于特征对某些变换鲁棒性较差，整体的性能指标偏低。

因此在第4章中，我们提出了基于 SURF 的图片序列匹配算法。该算法在每个镜头中选取一帧提取 SURF 特征，并统计基向量上的直方图；利用图片序列匹配方法筛选候选视频，进而向两端扩展改善局部精确度。在 MUSCLE-VCD-2007^[1]上的测试表明其性能优越，复杂度较低，局部精确度高，基本达到了预期目标。

5.2 主要成果

我们工作的主要成果是，基于先筛选候选视频后改善局部精确度的总体思路，提出了两个算法，并在 MUSCLE-VCD-2007^[1]上对算法进行了评估。

基于顺序度量的直方图匹配算法充分利用了顺序度量复杂度低的优势，对每一帧都提取特征，使得在筛选出候选视频后可以进行帧级别上的精确匹配；同时使用统计直方图的方法，避免了直接使用帧级别特征的高复杂度，最终获得了明显的时间优势和不错的性能。

基于 SURF 的图片序列匹配算法则充分利用了 SURF 特征具有优越性能的优势，并利用统计基向量上的直方图、图片序列模糊匹配的方法，在控制了复杂度的同时保持了良好的性能。测试表明这个算法性能优越，复杂度较低，局部精确度高，基本达到预期目标，验证了我们总体思路的有效性。

5.3 下一步工作

基于 SURF 的图片序列匹配算法是比较好的算法，下一步工作可以基于这个算法做进一步的提高。下面给出了一些可能的方向。

5.3.1 镜头划分

如 4.4 所说，现在使用的镜头划分算法并不完全适合于视频复制检测任务。如果能提高镜头划分对变换的鲁棒性，则可以提高筛选候选视频的性能。

5.3.2 局部特征的选取

虽然 SURF 特征性能优越，但它并非为视频复制检测任务量身定做，因此可能会有一些缺陷，比如第 4 章中提到，SURF 特征的直方图对于放缩变换缺乏鲁棒性；又比如，SURF 特征的分辨性和鲁棒性可能超过了我们的需要，而同时伴随着过高的复杂度。如果能够为视频复制检测专门设计局部特征，则有可能得到更好的性能和更低的复杂度，从而使得整个算法的性能得到提高。

5.3.3 局部特征的提取

注意到查询的大部分时间都用于提取局部特征，因此如果能够改进局部特征提取的速度，则可以使得查询的时间有显著的降低。比如可以对提取局部特征的过程进行简化，去掉对视频复制检测无用的特点。

5.3.4 基向量的选取

现在使用的基向量都是单码本，可以考虑使用多码本，即将特征空间进行从粗到细的划分，得到不同粒度划分上的统计直方图。这样或许能对算法的性能有所提高。

5.3.5 使用更大更复杂的测试集

现在使用的 MUSCLE-VCD-2007^[1]有 100 个小时，是比较大的数据集；但是对比应用中可能遇到的上万小时或更多的数据量（如网络视频监控），则又显得比较小。另外供测试的查询视频比较少，可能不能暴露算法的一些缺陷。比如说，算法可能在复制片段比较短的情况下性能急剧下降，但提供的查询中，复制片段都比较长，因此不能暴露这个缺陷。更大更复杂的数据集将能暴露算法的缺陷，从而为进一步改进指明方向。

插图索引

- 图 1.1 [10]复制视频示例。左列为源视频，右列为视频编码格式变换得到的复制视频。各视频的格式如下。左上：Mpeg1，右上：Avi，左中：RealVideo128k，右中：RealVideo512k，左下：Mpeg1，右下：Avi。
..... 2
- 图 1.2 [3]中复制视频示例。其中(a)为颜色调整、模糊和噪声变换的示例，(b)为放缩、偏移和插入字幕的示例，(c)为垂直方向拉伸和剪裁的示例。
..... 3
- 图 1.3 MUSCLE-VCD-2007^[1]中复制视频示例。右上为左上经过水平翻转和剪裁得到的视频，右下为左下经过 camcoding 得到的视频。 4
- 图 1.4 严厉变换示例^[3]。第一行为虚拟背景变换，第二行为插入大标志和颜色变换。 5
- 图 1.5 视频复制检测定义的图示 6
- 图 2.1 [4]中顺序度量的图示。(a)将图像划分为 m 行 n 列的子区域，图中为 3 行 3 列；(b)各个子区域中灰度的平均值；(c)灰度值的排序结果即顺序度量值。 10
- 图 2.2 [4]中给出的时空特征图。图中的曲线为各个子区域的平均灰度值随着时间的波动。两个对齐的视频片段之间的距离为各帧之间的距离，和图中曲线变化趋势之间的距离两者的加权平均。 11
- 图 2.3 [11]中的特征提取的图示 12
- 图 2.4 局部精确度图示。 u, v, x, y 为正确的边界， u', v', x', y' 为实际检测时判断的边界。 12
- 图 2.5 使用 Hilbert space-filling curve 对二维空间进行分割的图示^[11]。从左到由分割深度为 3, 4, 5。 13
- 图 3.1 基于顺序度量的直方图匹配算法的流程图 16

图 3.2	顺序度量提取时的子区域划分	17
图 3.3	放缩和插入边框示例	18
图 3.4	顺序度量提取时的排序	19
图 3.5	使得镜头划分不同的严厉变换。左边的源视频宽和高都缩小为 0.5 倍得到右边的复制视频，两者的镜头划分不相同。	19
图 3.6	统计分布直方图	20
图 3.7	ANN 索引结构 ^[18]	21
图 3.8	筛选候选视频	21
图 3.9	复制片段边界不是源视频中的边界	22
图 3.10	$score[s][t]$ 的离线计算	22
图 3.11	MUSCLE-VCD-2007 ^[1]	23
图 3.12	对比集算法第一类查询的测试结果	25
图 3.13	对比集算法第二类查询的测试结果	25
图 4.1	基于 SURF 的图片序列匹配算法的流程图	30
图 4.2	SURF 特征示例 ^[19]	31
图 4.3	放缩前后帧中检测到的 SURF 点	31
图 4.4	物体识别示例 ^[19]	32
图 4.5	帧偏移	32
图 4.6	SURF 相似度图像	33
图 4.7	基向量上的分布直方图 ^[8]	34
图 4.8	自然基上分布直方图的相似度图像	35
图 4.9	64 个聚类基向量上直方图的相似度图像	36
图 4.10	近邻图中的断点和干扰。直方图维数为 64。	37
图 4.11	严厉变换下的近邻图像。左图为 64 维，右图为 1024 维。	38
图 4.12	镜头划分不一致示例	38

图 4.13 近邻图中的错位	39
图 4.14 需要找到的直线	39
图 4.15 不正确的镜头匹配	40
图 4.16 改善局部精确度。其中已匹配 A 和 B，向两端扩展匹配 C 和 D 以改善局部精确度。	42

表格索引

表 3.1	MUSCLE-VCD-2007 ^[1] 中的第一类查询.....	24
表 4.1	第一类查询测试结果	43
表 4.2	第二类查询测试结果（不执行局部精确度改善）	43
表 4.3	第二类查询测试结果（执行局部精确度改善）	43

参考文献

- [1] <http://www-rocq.inria.fr/imedia/civr-bench/index.html>
- [2] <http://www-nlpir.nist.gov/projects/tv2008/tv2008.html>
- [3] Law-To, and Julien. Video copy detection: a comparative study. Conference On Image And Video Retrieval: Proceedings of the 6th ACM international conference on Image and video retrieval; 09-11 July 2007.
- [4] C. Kim and B. Vasudev. Spatiotemporal sequence matching techniques for video copy detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 1(15):127–132, Jan. 2005.
- [5] X.-S. Hua, X. Chen, and H.-J. Zhang. Robust video signature based on ordinal measure. In *International Conference on Image Processing*, 2004.
- [6] Chih-Yi Chiu et al. A Time Warping Based Approach for Video Copy Detection. In *Proceedings of ICPR 2006*.
- [7] J. Yuan, Q. Tian, and S. Ranganath. Fast and robust search method for short video clips from large video collection. *International Conference on Pattern Recognition*, 2004.
- [8] Lu Liu, Wei Lai, Xian-Sheng Hua, Shi-Qiang Yang. Video Histogram: A Novel Video Signature for Efficient Web Video Duplicate Detection. In *Proceeding of MMM2007(The International MultiMedia Modeling Conference)*.
- [9] Heng Tao Shen, Xiaofang Zhou, Zi Huang, and Jie Shao. Statistical summarization of content features for fast near-duplicate video detection *International Multimedia Conference: Proceedings of the 15th international conference on Multimedia; 25-29 Sept. 2007*.
- [10] Arun Hampapur, Rudolf M. Bolle. Comparison of Distance Measures for Video Copy Detection. In *Proceedings of ICME 2001*.
- [11] A. Joly, O. Buisson, and C. Frélicot. Content-based copy detection using distortion-based probabilistic similarity search. *IEEE Transactions on Multimedia*, 2007.
- [12] A. Joly, C. Frélicot, and O. Buisson. Robust content-based video copy identification in a large reference database. In *Proc. Int. Conf. Image and Video Retrieval*, 2003.
- [13] A. Hampapur, K.-H. Hyun, and R. M. Bolle. Comparison of sequence matching techniques for video copy detection. *The SPIE Conference on Storage and Retrieval for Media Databases*, 2002.

- [14] L. Chen and F. W. M. Stentiford. Video sequence matching based on temporal ordinal measurement. Technical report no. 1, UCL Adastral, 2006.
- [15] J. Law-To, O. Buisson, V. Gouet-Brunet, and N. Boujema. Robust voting algorithm based on labels of behavior for video copy detection. In ACM Multimedia, MM'06, 2006.
- [16] I. Laptev and T. Lindeberg. Space-time interest points. In International Conference on Computer Vision, 2003.
- [17] 付滨.复制/近似复制的视频检测.清华大学, 2007.
- [18] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Wu. An optimal algorithm for approximate nearest neighbor searching. J. ACM, 45:891-923, 1998.
- [19] Herbert Bay, Tinne Tuytelaars, Luc Van Gool. SURF: Speeded Up Robust Features. In Proceedings of the ninth European Conference on Computer Vision, 2006.
- [20] David G. Lowe. Distinctive image features from scale-invariant keypoints. In International Journal of Computer Vision, 2004.

致 谢

感谢我的指导老师张钹老师。在开题时，张老师指出了任务定义的模糊，使得我重新考虑任务的定义，避免了走不必要的弯路。谢谢您。

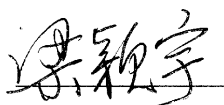
感谢我的辅导老师李建民老师。李老师工作十分繁忙，但他还是经常抽出宝贵的时间来指导我的研究。对于在毕业设计中遇到的各种问题，或者在我没有思路时，他总是能够给我很有帮助的建议，使得我顺利完成了毕业设计。另外，我这学期下午不常去实验室，感谢李老师对我的包容和理解。谢谢您。

感谢组里的师兄们，在我有不懂的问题，需要帮助的时候，师兄们总是给予我热情的帮助。感谢王栋师兄，感谢袁进辉师兄，感谢刘啸冰师兄，感谢王知昆师兄。

感谢和我同组的曹彬彬同学和王端鹏同学，我们一起进行毕业设计，彼此分享了很多信息，也相互讨论交流。

声 明

关于论文内容没有侵占他人著作权的声明，放在致谢页后。请认真阅读声明内容，全面审视自己的论文，是否严格遵守《中华人民共和国著作权法》，对他人享有著作权的内容是否都进行了明确的标注，慎重签名。

签 名： 日 期：2008年6月13日

附录A 外文资料的阅读调研报告或书面翻译

Literature Review

This literature review is organized in this way: Section I provides some background information about content-base video copy detection (CBVCD, or shorten as CBCD in this article) and details the definition, along with some promising applications; Section II discusses previous work related to CBCD; Section III shows some of my ideas about the CBCD issue; and finally, Section IV lists some important and valuable references.

I. Background Introduction

Nowadays more and more digital videos are available through the TV channel, on the web and in the multimedia database due to the development and widespread use of video technology. Since videos are easily got by an increasing use of personal handle-held cameras and through various kinds of multimedia channel, and easily edited by video processing software such as Adobe Premier and Windows Movie Maker, there are many videos which are copies of each other which have nearly the same content but usually been practiced various transformations. To manipulate such copies brings CBCD to a critical issue.

A. Definition of CBCD

The general meaning of copy or CBCD is self-explained but the precise definition is still ambiguous. In [1], a copy is defined as a segment of video derived from another video, usually by means of various transformations such as addition, deletion, modification (of aspect, color, contrast, encoding...), camcording, etc. In [3], a similar definition is proposed: a document O_1 is a copy of a document O if $O_1 = t(O)$, $t \in T$, where T is a set of tolerated transformations, which indicate those transformations that do not prevent the transformed video from being recognizable.

Note each video V as a frame sequence $\{V_i, 1 \leq i \leq N\}$. Based on the notion of copy, a general definition of CBCD can be described as follows. Given a set of videos $\{R^j\}$ and a query video $Q = \{Q_i, 1 \leq i \leq N\}$, the task of CBCD is to determine for each $R^j = \{R_i^j, 1 \leq i \leq M^j\}$, all $1 \leq u < v \leq N$ and $1 \leq x < y \leq M^j$, if there exist, satisfying that $\{Q_i, u \leq i \leq v\}$ is a copy of $\{R_i^j, x \leq i \leq y\}$. Generally, only the situations when $\{Q_i, u \leq i \leq v\}$ and $\{R_i^j, x \leq i \leq y\}$ are long enough are considered, and here they are constrained to longer than 10 seconds.

Many references consider the situation when $u=1$ and $v=N$, such as [7]-[10]. Some focus on an even more restricted situation when $u=1, v=N, x=1, y=M^j$, such as [11], [12]. These situations are included in the more general definition described above, which is more complex but practically called for in real applications. Thus, the general definition is used here for CBCD.

B. Application

Here are some possible applications of CBCD.

1. Copyright Protection

There are basically two approaches to detecting copies of digital video, watermarking and CBCD. In watermarking, information is embedded into the video before distribution and extracted to prove ownership. CBCD is an alternative approach to watermarking and its primary thesis is “the media itself is the watermark”. One of the main advantages of CBCD is that the already existing video materials can be tracked even if no watermark is embedded into the materials.

2. Usage Tracking

CBCD can be applied when the usage of a certain video material need to be tracked. For example, a company wants to ensure that its advertisement has been broadcast in the state channel. Also, the advertisements of the competitive companies can be tracked to infer some useful business information.

3. Copy Purge

Copies can be unnecessary or even undesirable in some cases, and if they can be detected, copy purge can be applied to the videos. In the video search scene, the copies may underplay the effectiveness of the search result and weaken the focus of users. Copies can be simply deleted or link to the most promising one to enhance the

performance of the search system. Another case involves the video database, where copies may significantly increase the size of the database and decrease the efficiency.

4. Contextual Linking

This is newly advocated application in [3]. In this case, the CBCD system generates a lot of contextual links (frequency, persistence, geographic dispersion, etc.) that are highly informative and can be exploited by the data mining methods. Two television scenarios are provided in [3] to illustrate the potential information in the links:

- A video which is broadcast the same day on several foreign channels refers to an international event.
- A video already having a lot of copies distributed among a large period refers to a major historical event.

II. Related Work

There are generally two main steps in CBCD: the first is to extract a certain kind of descriptor, or called feature or signature, from the video, and the second is to search the set of descriptors extracted to find the possible (global or partial) matches of the query.

II-I. Feature extraction

There are mainly two kinds of features according to [2].

A. Global Descriptors

These descriptors are extracted by considering the global spatial features of frames in the sequence, or the temporal features of the sequence, or the combine of the two.

1. Spatial

The spatial features used are the traditional features in image management, including image pixels itself, histogram of RGB, HSV or Gradient Direction, edge representation and invariant moments compared in [6]. More details about the feature and the distance measures can be found in the thesis. The result of the compare favors the edge representation and its Hausdorff distance measure, followed by HSV

histogram and its intersection distance measure. However, the performance of these features is not very promising.

In [5], a successive thesis of [6], the motion direction descriptor (a temporal descriptor), the ordinal descriptor, and the color histogram descriptor are compared, all using the sequence matching. The results of the comparison indicate that the ordinal feature has superior performance followed by the motion direction, and then the color histogram. Recent studies of global descriptors in CBCD mainly use the ordinal measure or its variances as the descriptor due to their relatively simplicity and superior performance, so some possible beneficial details about it are given below.

The ordinal measure was originally proposed by Bhat and Nayar in [13] for computing image correspondence. Mohan use an adaptation of the original measure in [14] for video sequence matching. Basically, the extraction are performed as follows: first, the frame is divided into $N=N_x*N_y$ blocks, then the average gray level in each block is computed, and finally the average intensities are sorted and the array of ranks is used as the descriptor of the frame.

2. Temporal

The temporal descriptor focuses on the difference of the succeeding frames in the sequence. In [2], a simple temporal descriptor in the comparison is computed by summing up the differences of the intensity of each pixel in neighboring frames. In [5], the motion direction descriptor can also be viewed as a kind of temporal descriptors. The frame is divided into $N=N_x*N_y$ blocks, and the most similar area near each block in the succeeding frame is searched and thus the directions of the local optical flow can be computed as the descriptor.

3. Spatiotemporal

The spatiotemporal descriptors utilize both the spatial and temporal information by combining the two in the match of video sequences. The temporal ordinal method proposed in [15] uses the rank of blocks along the time instead of using the rank in the image. In [7], ordinal measure distance and temporal differences of each block in succeeding frames are summed up by weight to get a better distance measure of the video sequence.

In [10], the ordinal array of all frames in the sequence is compressed into the distribution histogram. In [11], seed vectors are selected and each frame feature vector

adds weight to its nearest seed vector, and the array of weights of the seed vectors serves as the descriptor. These two descriptors are representation of a whole video and cannot be applied directly to the general definition of CBCD.

B. Local Descriptors

The local descriptors used are interest points. The framework for CBCD in [3] use an improved version of the Harris interest point in the key-frames as the local indicator, and compute the descriptor by differentiating the local region around each interest point. The descriptor used in [17] is similar to that of [3] but different in the selection of local region around the interest point and an association of the points selected to form trajectories which do not appear in [3]. Besides the two above, one other descriptor involved in the comparison in [2] is the space-time interest point proposed in [16]. More details about these descriptors can be found in the corresponding references.

The results of the comparison in [2] indicate that the ordinal measure is very efficient in most of the single transformations but presents poor performance in zooming, cropping, letter-box transformation and also in random mixed transformations. Local descriptors have good performance in both single and random mixed transformations, but also more computational costs. However, the comparison is carried out in a relatively small database and all methods are tested by the general definition of CBCD, while in some other thesis, performance is considered in a large database or in the restricted definition of CBCD. Also, the localization accuracy concerning frame-level precision is not evaluated. The selection of descriptors needs to take these factors into account.

Some other features are also used such as the audio feature in the fine search step of the algorithm in [10]. Details can be found in the reference and thus are omitted here.

II-II. Search

Two kinds of search methods are presented: exhaustive and indexing, but different implementations are employed in different definitions.

A. Most Restricted Definition

In this case, $u=1$, $v=N$, $x=1$, $y=M^j$. Thus, it is the simplest, which means the performance is relatively easy to guarantee. Therefore, the studies generally focus on the scalability and complexity of the algorithms. Large database is employed to test the algorithm, emphasizing on the reduction of storage and time. In order to achieve the goal, a video is usually summarized as one signature, not a sequence of signatures, and thus the problem reduces to find the neighbors of the query signature.

In [11], exhaustive search is performed, but this thesis focuses on the uniqueness and robustness of the signature proposed and in future work, more efficient approach may be used. In [12], a B^+ -tree of the signatures of the videos in the database is constructed based on one-dimension transformation by computing their distances to a reference signature. Then the distance d of the query signature to the reference signature is computed, and a range query $[d-r, d+r]$ in B^+ -tree is performed, where r is a given search radius. A two-dimension transformation using two reference signatures is proposed in this thesis, the principle of which is similar to the one-dimension transformation.

B. Restricted Definition

In this case, $u=1$ and $v=N$. Since it is not the matches between the query video and a whole target video in the database, the summarization of a whole video as one signature cannot be practiced. Instead, a video is represented as a sequence of signatures, usually one signature extracted from one key-frame. Thus the problem reduces to find the neighboring subsequences in the database similar to the query sequence.

Generally, exhaustive search is performed by aligning the query sequence to different possible positions in different sequences in the database, though sometimes some optimization may be employed. [7] and [8] exercise the exhaustive search. In [10], while sliding the aligning window in the target video for the query, the sliding process is accelerated by skipping unnecessary steps if the similarity of the current window is too low. In [9], before matching is performed, start candidates C_{start} that are similar to the first frame signature of the query are searched from the target sequence; similarly, C_{end} are found. According to C_{start} and C_{end} , possible subsequences are selected and compared to the query. This optimization reduces the computation significantly.

In this case, the localization accuracy that does not occur in the most restricted definition comes into consideration. To improve this frame-level precision and also the video-level precision, a fine step may be practiced after the coarse search. The audio feature is used in the fine step in [10] while in [8], dynamic programming technique is applied.

C. General Definition

In this case, no restriction is imposed on the definition. As in B, the video is represented as a sequence of signatures, but the search step is to find neighboring subsequences similar to a subsequence in the query.

Generally, signatures are organized in an index structure. Neighbors of each signature in the query sequence are searched in the index structure. [4] uses the Hilbert-filling curve structure proposed in [18] as the index and performs range query in the structure. In the succeeding thesis [3], the same index is used but a distortion-based probabilistic similarity search is performed instead of rang query.

Based on the matching information between the query signatures and those in database, a voting process is then performed to determine if the video in the database contains a copy of a subsequence in the query. The voting algorithm used in [2]-[4] are based on the matches geometrically-consistent with a global transform model. For each video in database, a global transform model is estimated and the number of geometrically-consistent matches is computed as a criterion for judging whether the video contains a copy.

The recent studies on this case emphasize on the video-level precision and computational costs. Localization accuracy is not discussed in the references.

III. My Ideas

A. Definition

The selection of the definition of CBCD influences the study greatly since different definitions provide different task scenarios and emphasize on different goals. I aims at the general definition for it includes the other two and in practical usages this definition would be desired. For example, in enforcing copyright, copies of a segment in the original video are probably embedded into other irrelevant videos. The situation can only be dealt with by algorithms for the general definition.

B. Descriptor

Among global descriptors, ordinal measure has been proved to be a simple feature excellent in both uniqueness and robustness though it still has some flaws in dealing with the zooming, cropping and letter-box and mixed transformations. One difficulty is that one ordinal measure for each frame will result in too much space took up by the descriptors, which reduces the scalability significantly. A possible solution would be to divide the video into shots, and ordinal measure descriptors within one shot can be compressed into the distribution histogram. The shot boundary techniques are well-developed, and also the accuracy in the CBCD scene is not demanded to be very precise. Further, the compression will not degrade the performance of the descriptor significantly since the information lost is about the time position of the frames within the shot and has been proved to be redundant in [10]. However, the effectiveness of this approach needs to be verified in experiment.

An alternative choice is the local descriptor induced by interest points. Similarly, the local descriptors within one shot can be compressed into distribution histogram.

Which descriptor to choose depends on the results of experiment carried to test the performance of the two at shot level.

C. Search

A two-step approach, including a coarse search step and a fine step, can be designed to guarantee both low time cost and high precision.

In the coarse search step, the shot-level descriptors are used and the index structure should be constructed to reduce the time complexity and thus enhance scalability of the system. The index structure can be designed to cooperate with the performance of the descriptor selected.

Voting process is performed while searching: some score is added to the video if it contains a match shot to one in the query. And in the end, the videos with high scores are considered to contain copies of some subsequence in the query.

In the fine step, frame-level descriptors are used and dynamic programming technique or other subsequence matching algorithms (possible approximate algorithms) can be applied to further improve video-level precision and determine the matching subsequence in the query and the target video.

Reference

- [1] <http://www-nlpir.nist.gov/projects/tv2008/tv2008.html>
- [2] Law-To, and Julien. Video copy detection: a comparative study. Conference On Image And Video Retrieval: Proceedings of the 6th ACM international conference on Image and video retrieval, 2007.
- [3] A. Joly, O. Buisson, and C. Frélicot. Content-based copy detection using distortion-based probabilistic similarity search. IEEE Transactions on Multimedia, 2007.
- [4] A. Joly, C. Frélicot, and O. Buisson. Robust content-based video copy identification in a large reference database. In Proc. Int. Conf. Image and Video Retrieval, 2003.
- [5] A. Hampapur and R. Bolle. Comparison of sequence matching techniques for video copy detection. In Conference on Storage and Retrieval for Media Databases, 2002.
- [6] A. Hampapur and R. M. Bolle. Comparison of distance measures for video copy detection. In Proc. Of Int. Conf. on Multimedia and Expo, Aug. 2001.
- [7] C. Kim and B. Vasudev. Spatiotemporal sequence matching techniques for video copy detection. IEEE Transactions on Circuits and Systems for Video Technology, 2005.
- [8] X.-S. Hua, X. Chen, and H.-J. Zhang. Robust video signature based on ordinal measure. In International Conference on Image Processing, 2004.
- [9] Chih-Yi Chiu et al. A Time Warping Based Approach for Video Copy Detection. In Proceedings of ICPR, 2006.
- [10] J. Yuan, Q. Tian, and S. Ranganath. Fast and robust search method for short video clips from large video collection. In International Conference on Pattern Recognition, 2004.
- [11] Lu Liu, Wei Lai, Xian-Sheng Hua, and Shi-Qiang Yang. Video histogram: a novel video signature for efficient Web video duplicate detection Advances in Multimedia Modeling. In 13th International Multimedia Modeling Conference, MMM 2007.
- [12] Heng Tao Shen, Xiaofang Zhou, Zi Huang, and Jie Shao. Statistical summarization of content features for fast near-duplicate video detection. In International Multimedia Conference: Proceedings of the 15th international conference on Multimedia, 2007.
- [13] D. Bhat, and S. Nayar. Ordinal measures for image correspondence. In IEEE Transactions on Pattern Analysis and Machine Intelligence, 1998.
- [14] R. Mohan. Video sequence matching. In Proceedings of the International Conference on Audio, Speech and Signal Processing, IEEE Signal Processing Society, 1998.
- [15] L. Chen and F. W. M. Stentiford. Video sequence matching based on temporal ordinal measurement. Technical report no. 1, UCL Adastral, 2006.

- [16] I. Laptev and T. Lindeberg. Space-time interest points. In International Conference on Computer Vision, 2003.
- [17] J. Law-To, O. Buisson, V. Gouet-Brunet, and N. Boujema. Robust voting algorithm based on labels of behavior for video copy detection. In ACM Multimedia, MM'06, 2006.
- [18] Lawder, J.K., King, P. J.H.: Querying multi-dimensional data indexed using the Hilbert space-filling curve. In SIGMOD, 2001.
- [19] 付滨.复制/近似复制的视频检测.清华大学, 2007.

综合论文训练记录表

论文题目	基于内容的视频复制检测
主要内容以及进度安排	<p>基于内容的视频复制检测在版权保护、视频跟踪、冗余去除等方面有重要应用。较多已有工作集中于检测整个查询视频的复制，而检测查询视频某个片段的复制没有得到重点关注，尤其是此时定位复制片段的局部精确度问题没有很好的处理方法。</p> <p>本毕业设计的主要内容是设计并实现新的算法来解决基于内容的视频复制检测任务，重点是在保证效率的前提下利用子序列匹配算法改进检测结果的局部精确度。</p> <p>进度安排如下：</p> <p>1-4 周：调研和算法框架设计</p> <p>5-8 周：算法的基本实现，争取完成基本的实验。</p> <p>9-13 周：基于实验结果分析对算法进行改进，和/或实现并检验新思路</p> <p>14-15 周：总结并撰写论文。</p> <p style="text-align: right;">指导教师签字： <u>张敏</u></p> <p style="text-align: right;">考核组组长签字： <u>李建元</u></p> <p style="text-align: right;">2008年3月21日</p>
中期考核意见	<p style="text-align: center;">较好地完成了工作安排中的任务。</p> <p style="text-align: right;">考核组组长签字： <u>李建元</u></p> <p style="text-align: right;">2008年4月21日</p>

<p>指导教师评语</p>	<p>论文分析了视频复制检测的两个主要算法,在此基础上提出一种新的算法,在MUSCLE-VCD-2007数据集上测试实验,表明新算法在性能与计算复杂度上均有改进。</p> <p style="text-align: right;">指导教师签字: <u>张敏</u></p> <p style="text-align: right;">2008年6月13日</p>
<p>评阅教师评语</p>	<p>论文针对视频复制检测这一重要技术进行了大量调研。在此基础上提出并实现了两种检测方法,对现有方法进行了改进。在CIVR 2007 video copy detection数据集上测试,两种方法均表现出较好性能。特别是基于局部特征的方法,可以大大改善局部精确度。</p> <p>论文工作认真,实验充分,为进一步提高性能打下了很好的基础。</p> <p style="text-align: right;">评阅教师签字: <u>李建元</u></p> <p style="text-align: right;">2008年6月14日</p>
<p>答辩小组评语</p>	<p>视频复制检测在版权保护、案情分析等应用中起到越来越重要的作用。论文针对这一重要的技术进行了大量调研工作。在此基础上提出并实现了两种检测方法。在CIVR 2007 video copy detection数据集上的测试表明,两种方法均在某些方法表现出较好性能,为进一步提高性能打下了很好的基础。</p> <p>论文工作认真细致,较好地完成了毕业论文。</p> <p style="text-align: right;">答辩小组组长签字: <u>李建元</u></p> <p style="text-align: right;">2008年6月14日</p>

总成绩: 88.85

教学负责人签字: 张

2008年6月21日
年 月 日