



(12) 发明专利

(10) 授权公告号 CN 101162470 B

(45) 授权公告日 2011.04.20

(21) 申请号 200710177523.3

(22) 申请日 2007.11.16

(73) 专利权人 北京交通大学
地址 100044 北京市西直门外上园村3号北京交通大学科技处

(72) 发明人 赵耀 刘楠 朱振峰

(74) 专利代理机构 北京众合诚成知识产权代理有限公司 11246

代理人 李光松

(51) Int. Cl.

G06F 17/30 (2006.01)

G06K 9/46 (2006.01)

(56) 对比文件

CN 1118967 A, 1996.03.20, 全文.

CN 1841371 A, 2006.10.04, 全文.

审查员 董泽华

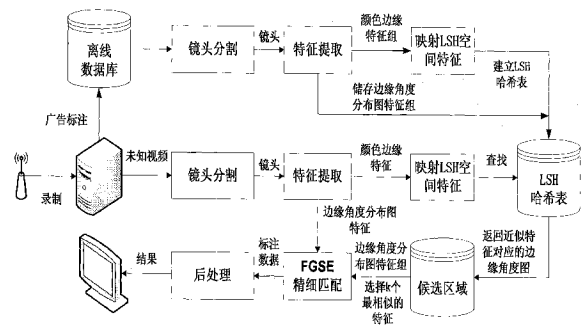
权利要求书 3 页 说明书 12 页 附图 9 页

(54) 发明名称

一种基于分层匹配的视频广告识别方法

(57) 摘要

一种基于分层匹配的视频广告识别方法,通过局部敏感性哈希算法 (Locality Sensitive Hash, LSH) 和精细尺度连续过滤算法 (Fine Granularity Successive Elimination, FGSE), 利用广告的新颖的视频特征结合后处理技术开发出一套能够快速检测电视节目是否存在数据库中的广告。技术方案是: 其特征在于包含以下步骤: 离线数据库建立与在线监测, 其中离线数据库建立分为离线预处理与哈希表建立; 在线监测部分分为在线预处理、由粗糙到精细的二级匹配和后处理。



1. 一种基于分层匹配的视频广告识别方法，其特征在于包含以下步骤：离线数据库建立与在线监测，其中离线数据库建立分为离线预处理与离线建立哈希表；在线监测部分分为在线预处理、由粗糙到精细的二级匹配和后处理；

所述离线预处理包括数据库广告镜头分割与特征提取，具体内容包括：对于离线数据库中每一个广告，将每一帧由 RGB 颜色空间转化为灰度空间，将灰度图像从中心区域划分为 8 个 64*64 像素的小块，通过计算每个小块对应的灰度直方图之间的差异，乘以相应的权值并求和，得到帧间灰度变化率，如果这个变化率大于设定的阈值，则视为两帧之间发生了镜头的切变，并记录镜头变化的位置，将每一个广告分为一系列的镜头序列；

对于每一个镜头，以镜头长度 25% 的位置作为区间起点，75% 的位置作为区间终点，提取这段区间内所有视频帧，对这些视频帧进行累加平均，获得代表镜头内容的平均关键帧，保留关键帧中心 160*160 像素区域的子图像，用作特征提取，在子图像中提取 36 维 HSV 颜色空间直方图与 64 维边缘角度直方图，以及边缘角度分布图特征，将每个图像的颜色直方图与边缘角度直方图分别归一化，组成 100 维特征向量用于建立局部敏感哈希表，同时，将存储所有镜头的 100 维联合特征以及边缘角度分布图特征，用于在线监测时，与未知视频进行镜头分割后所提取的 100 维联合特征以及边缘角度分布图特征进行比较；

所述离线建立哈希表包括：将获得的离线数据库中所有镜头产生的 100 维联合特征构造局部敏感性哈希表，并将 100 维联合特征、从上述所有镜头中提取的边缘角度分布图特征和提取出上述边缘角度分布图特征和 100 维联合特征的镜头所对应广告镜头名称和镜头编号，存储到所构造的局部敏感性哈希表中，具体为：设定子哈希表的个数与哈希键长，对于每一个子哈希表产生过程为，根据初始设定的两个变量，利用随机函数产生不同的哈希系数，利用这些系数将 100 维联合特征映射到汉明空间，转化为 0 和 1 的组合序列，根据序列中的 0 和 1 的个数，选择不同的哈希表位置，将所有产生的子哈希表组成局部敏感性哈希表，将离线数据库中所有镜头产生的 100 维联合特征以及边缘角度分布图特征和与之对应的标注信息存入哈希表中；

所述在线预处理，包括镜头分割与特征提取，具体内容包括：对于输入的视频文件，将每一帧由 RGB 颜色空间转化为灰度空间，将灰度图像从中心区域划分为 8 个 64*64 像素的小块，通过计算每个小块对应的灰度直方图之间的差异，乘以相应的权值并求和，得到帧间灰度变化率，如果这个变化率大于设定的阈值，则视为两帧之间发生了镜头的切变，并记录镜头变化的位置，将每一个广告分为一系列的镜头序列；对于每一个镜头，以镜头长度 25% 的位置作为区间起点，75% 的位置作为区间终点，提取这段区间内所有视频帧，对这些视频帧进行累加平均，获得代表镜头内容的平均关键帧，保留关键帧中心 160*160 像素区域的子图像，用作特征提取，在子图像中提取 36 维 HSV 颜色空间直方图与 64 维边缘角度直方图，以及边缘角度分布图特征，将每个图像的颜色直方图与边缘角度直方图分别归一化，组成 100 维特征向量，与边缘角度分布图特征一起，作为在离线数据库建立过程中产生的局部敏感性哈希表中进行搜索的查询向量；

所述由粗糙到精细的二级匹配，包括在局部敏感性哈希表中查询与精细尺度连续过滤两部分，具体内容包括：将获得的一个镜头的联合特征向量通过局部敏感性哈希方

法, 将其映射到数据库的 10 个哈希表对应的条目中, 获得与查询向量最为相近的 50 个候选向量及其对应的镜头与边缘分布图, 比较这 50 个镜头与查询镜头包含的帧数的差异, 如果相差小于 25 帧, 就利用精细尺度连续过滤算法 (FGSE) 对二者分别对应的边缘角度分布图进行更为精细的匹配, 在设定的阈值下, 如果匹配成功, 即可判断该镜头为数据库中某个广告的一部分, 将数据库中对应的广告镜头的名称和序号标注到查询镜头上, 如果输入的镜头是视频文件中的最后一个镜头, 进行后处理, 否则读入下一个视频镜头;

所述后处理内容包括: 首先, 由粗糙到精细的二级匹配过程将广告名称以及序号标注到了未知视频的查询镜头中, 这些标注信息构成了标注序列, 将标注序列中具有不同标注广告名称的镜头划分到若干个子段, 对每一个子段, 根据镜头在原始文件中的起始位置, 在时间轴上依次排列, 利用标注的广告镜头序号与起始点之间的位置关系消除其中的漏检与误检, 获得连续完整的标注信息; 经过纠正的子段, 如果其播放长度大于数据库中该广告长度的一半, 同时检测到的镜头数量大于数据库中广告镜头个数的三分之一, 则认为标注成功, 处理完所有子段后, 对结果按照时间先后顺序排序, 输出。

2. 根据权利要求 1 所述的基于分层匹配的视频广告识别方法, 其特征在于所述平均关键帧计算方法, 包括如下步骤:

(1) 判断镜头中包含的视频帧的个数, 如果等于 1, 将该帧作为平均关键帧, 结束; 否则, 转步骤 (2);

(2) 以镜头长度 25% 的位置作为区间起点, 75% 的位置作为区间终点, 提取这段区间内所有视频帧, 其数量占镜头中视频帧总数的 50%;

(3) 累加图像像素的值, 求平均, 获得平均关键帧。

3. 根据权利要求 1 所述的基于分层匹配的视频广告识别方法, 其特征在于所述平均关键帧中提取 64 维边缘角度直方图特征, 其步骤如下:

(1) 将 RGB 彩色图像转化为灰度图像; 在平均关键帧中, 对同一幅图像分别从中心位置截取 160×161 和 161×160 像素的两张子图;

(2) 将 160×161 像素的子图像, 按照从左到右的顺序, 用每一个像素减去右方相邻像素, 得到 160×160 的水平方向上的边缘分布 dx ; 将 161×160 像素的子图像, 按照从上到下的顺序, 用每一个像素减去下方相邻像素, 得到 160×160 的垂直方向上的边缘分布 dy ;

(3) 计算当 $dx \neq 0$ 且 $dy \neq 0$ 时, 每个像素的边缘角度: $Ang = \arctan(dy/dx)$, 得到 160×160 的边缘角度分布图; 统计每个角度区间中像素的个数;

(4) 归一化, 将特征中每一维除以该特征中的最大值, 得到全局边缘角度的分布。

4. 根据权利要求 1 所述的基于分层匹配的视频广告识别方法, 其特征在于所述在平均关键帧中提取边缘角度分布图特征, 其步骤如下:

(1) 将 RGB 彩色图像转化为灰度图像; 在平均关键帧中, 对同一幅图像分别从中心位置截取 160×161 和 161×160 像素的两张子图;

(2) 将 160×161 像素的子图像, 按照从左到右的顺序, 用每一个像素减去右方相邻像素, 得到 160×160 的水平方向上的边缘分布 dx ; 将 161×160 像素的子图像, 按照从上到下的顺序, 用每一个像素减去下方相邻像素, 得到 160×160 的垂直方向上的边缘分布

dy ;

(3) 计算当 $dx \neq 0$ 且 $dy \neq 0$ 时, 每个像素的边缘角度: $Ang = \arctan(dy/dx)$, 得到 $160*160$ 的边缘角度图;

(4) 将边缘角度图平均分为 $4*4$ 个子块, 统计每一幅子图中, 每个角度区间中的像素点的个数; 每一个子块中的区间个数为 16, 区间间隔为 $360/16$;

(5) 将每个子块中的 16 位一维直方图按照 $4*4$ 的顺序重新排列, 根据子图在边缘角度分布图中的位置, 可以得到 $16*16$ 维边缘角度分布图特征。

5. 根据权利要求 1 所述的基于分层匹配的视频广告识别方法, 其特征在于所述将精细尺度连续过滤算法 FGSE, 应用于图像识别中; 其步骤如下:

(1) 初始化分割的层数 $l = 0$, 子图像序号 order 为 0, 子图像的大小 M 等于输入图像的高度乘以宽度, 即 $n*n$; 在第 0 层中子图像的个数 S_l 为 1, 分割的总次数 $L = n*n+3$, 初始化子图的高度与宽度均为 $m = n$;

(2) 计算两个图像中各个子图像对应的 BV_l :

$$BV_l = \sum_{k=0}^{S_l-1} |Q_l^{(k)} - R_l^{(k)}|$$

其中 $Q_l^{(k)}$ 和 $R_l^{(k)}$ 分别对应查询图像和候选图像在第 l 层时对应的第 k 个子图像中各像素值的和;

(3) 如果 BV_l 大于等于阈值 SAD_{min} 跳转至 (5), 否则置 $l = l+1$; 对图像进行分割, 每次分割, 仅对图像中的第一个未分割的子图像进行分割:

a. 将大小为 $m*m$ 的第 i 个子图分割为四个大小为 $\frac{m}{2} * \frac{m}{2}$ 的子图, 更新 $S_l = S_l+3$, order 中第 i 块及以前的块序号保持不变, 后面的序号依次加 3;

b. 判断是否所有的长度为 $m*m$ 的子图像均完成了分割, 如果完成, 令 $i = 0$, $m = m/2$, 否则 $i = i+4$ 指向下一个大小为 $m*m$ 的子图像;

(4) 重复 (2), (3) 直到最后一层 L 分割完成, 如果 $BV_l < SAD_{min}$, 则令 $SAD_{min} = BV_l$;

(5) 检测是否还有待检测图像未被检测的候选图像, 如果存在未被检测的图像, 跳转至步骤 (1), 否则输出是否匹配的结果。

一种基于分层匹配的视频广告识别方法

技术领域

[0001] 本发明属于视频广告识别方法领域，尤其涉及一种基于分层匹配的视频广告识别方法。

背景技术

[0002] 随着科学技术和信息技术的进步，特别是计算机技术、网络技术和大容量存储技术的不断发展，人们已经运用各种手段大量的采集和生产了各种类型的多媒体信息数据。多媒体已成为信息高速公路上传送数据的主要组成部分。在这些多媒体信息中，数字化媒体结合了视觉和听觉，有效地向人们提供了大量的信息，具有重要的应用和研究价值。

[0003] 多媒体数据流中，既包括结构化的数据也含有非结构化数据。在多媒体数据中除了包含传统的文本和数字外，其它主要包含的数据类型还有：图形、图像、动画、视频、音频以及上述各种媒体的组合。为了能够在数据量巨大和结构复杂的多媒体数据中快速的定位感兴趣的信息，人们提出了比较有效的检索算法——基于内容的多媒体信息检索 (Content-based multimedia information retrieval, CMIR)。所谓基于内容的多媒体检索是指对多媒体数据（如视频、音频流等）中所蕴藏的物理和语义内容进行计算机分析和理解，以方便用户查询，其本质是对无序的多媒体数据流结构化，提取语义信息，保证多媒体内容能被快速检索。多媒体检索与文本信息检索相比要复杂的多。

[0004] 广告视频检索作为多媒体视频检索的一个方面，正引起人们越来越多的关注。这是因为广告视频在日常的信息传递中占据着越来越重要的角色，随着商业化进程的加快，广告越来越频繁的出现在电视、广播、互联网等多个媒体平台，潜移默化的影响着人们的生活方式。广告作为媒体中商业信息的主要载体在数字化的今天依然扮演着不可或缺的角色。伴随着网络技术及多媒体技术的日益普及，数字电视技术在个人与专业应用领域得到了广泛的应用，例如电视节目转录软件，电视节目监测系统。如何通过计算机自动甄别广告与正常的视频节目成为了迫切的市场需要。

[0005] 通过观察，广告的受众可以分为一般观众群体、广告公司和监测机构，不同的人群对广告内容的需求不尽相同。对于观众来说，一方面广告影响了正常的视频节目的播放，扰乱了人们正常地观看模式，有时用户在录制电视节目的过程中，也不得不花费大量的存储空间，录制这些对自己没有用处的信息。另一方面，观众如何从浩如烟海的众多广告中，快速寻找自己感兴趣的内容，也已成为一个亟待解决的问题。对于广告公司而言，广告视频在商业信息的传递中占据着越来越重要的地位，他们通过对广告的分析，帮助客户了解竞争对手的产品情况，检测电视台是否完全履行广告合同或是通过监测其他广告公司的广告提高自身的广告创作技术。监测机构往往是隶属于国家的专业管理部门，他们实时监测广告，保留违法的证据与信息用以查处违反国家政策法规的公司与个人，维护社会的正常秩序。每一天，各个电视台都会产生海量的视频数据，如果对广告的监测、过滤、分析等工作，完全通过人工来完成，处理一个电视台一天的广告数

据，往往需要耗费多名工作人员四五天的工作量，如果考虑到全国多个电台的监测，工作量更是呈现指数级的增长。如何开发一种能够自动识别广告的电视监测系统，分析、监测、滤除广告，越来越受到了各方面的重视与关注。

[0006] 针对广告视频和普通节目视频之间的特征区别，人们提出了很多的广告检索算法。根据检索算法基于的特征不同，比较实用的广告监测系统的检索算法大体上可以分成以下三类：

[0007] 1、基于标识的方法 (Logo-based methods)

[0008] 此方法是最早被提出用于广告视频定位的方法。它主要是利用电视台的台标来进行检测的。电视台在播放一般的节目时会将自己的台标显示出来，而当播放广告时一般就会隐去台标，这使得我们可以通过检索台标的存在与否来区分广告视频和一般视频。台标一般分为三种：静态台标、半透明台标和动态台标。针对不同形态的台标，都有相应的检索算法，以实现广告视频的定位。其中对于静态台标的研究是比较深入的，而且应用得很广泛。但是对于后两种台标，因为其自身的制作手法和演示特点，检测起来比较困难。所以还没有比较成熟的检测算法。

[0009] 2、基于规则的方法 (Rule-based methods)

[0010] 基于规则的系统大多是利用广告镜头信息来进行检测，通过多组特征和规则区分广告视频与一般电视节目视频。由于广告视频和普通的电视节目在某些特征方面存在着比较明显的差异，因此通过定义和利用这些区分性比较强的特征，就可以实现对广告视频的检测。比如在视频方面可以通过提取一段视频帧的平均边缘变化率 A-ECR (Average of Edge Change Ratio) 和边缘变化方差 V-ECR (Variance of Change Ratio) 以及平均帧差 A-FD (Average of Frame Difference) 和帧方差 V-FD (Variance of Frame Difference) 来实现检测。在音频方面，广告视频部分的音频内容和普通的节目部分的音频信息也存在一些明显的特征，比如音频多维频率谱系数 (Mel-frequency Cepstral Coefficient) 和音频信息熵来实现对视频广告的分割检测。不过音频方面的特征一般都是用于辅助视频特征检测方面，通过两者的结合实现对广告段更加精确的分割检测。在近期的研究中，很多基于规则的检索方法中有引入了分类器算法，通过对样本的训练，获得分类性比较好的分类器，然后对广告镜头和普通节目镜头进行分类，从而得到比较精确的检测结果。

[0011] 3、基于识别的方法 (Recognition-based methods)

[0012] 此方法利用视频数据库预先储存大量的已知广告的特征信息，采用相应的匹配算法计算待确定镜头和广告样本镜头之间的相似度，从而确定其是否为数据库中的广告。在实际的检索系统中首先检测出一些潜在的、可能是广告的视频或音频段，然后将其特征和广告数据库里面的广告视频段或音频段的特征逐一进行比较，然后可以判断出此部分是否为数据库里面的广告。

[0013] 但是由于广告数据是一种非格式化、时空并存、数据量大的信息，对于它的检测就会涉及到很多的问题。此外由于不同的广告视频在制作方式和表现手法上的多种多样，使得上述三大类广告检测算法在对广告视频检测上都存在着很多问题。

[0014] (1)、对于基于标识的检测算法来说，它只是简单的依靠台标的存在来判断广告视频的存在，因此首先必须能正确的检测出台标。但是现在很多电台在电视节目里

面插播广告视频的时候也往往会将台标显示出来，这样基于标识的检测算法就会失效；另外，现在电视台的台标越来越复杂，从早期的静态台标演变成现在的动态的、半透明的、时隐时现的台标，这些使得台标的检测越来越困难。

[0015] (2)、对于基于规则的方法来说，此方法需要定义多组用来区分广告和普通节目的特征和阈值。因此区分特征和阈值的选择是十分重要的。但是由于广告是多格式化的，风格不同的广告之间的区别也十分明显，因此很难找到通用的区分特征。另外检测系统可能对于某些阈值比较敏感，所以对于所有类型的视频很难找到一组统一的阈值。而且这类方法过分依赖于各国各地的法律法规，剪切手段，以及各种节目的制作风格，不能够形成一种通用的电视监测系统。监测的正确率和查全率往往受到一些视觉信息或听觉信息相类似的节目的干扰，正确率与查全率往往不高。

[0016] (3)、对于基于识别的方法来说，此方法虽然只能检索事先已经获知的数据库中的广告，但是这种系统的正确率与查全率可以达到较高的水平，能够满足实用的需要，而且这种方法并不需要依靠各种广告法规或制作技术等主观的人为因素，完全通过客观信息进行检测，尽可能大的排除了人为因素的干扰，可以跨越不同电视机构平台构建通用的监测系统。同时人们也有理由相信，在高检出率与正确率的前提下，可以通过融合第一第二类技术或者结合对重复音视频探测技术，实现对未知的广告探测，因此的其应用前景十分的广阔。在这类技术中，关键的瓶颈问题在于匹配的速度与数据库存储的大小，如何在数据库很大的情况下，节省存储空间，保持搜索速度，显得尤为重要。

发明内容

[0017] 本发明正是基于第三类方法，提出一种基于分层匹配的视频广告识别方法，通过局部敏感性哈希算法 (Locality Sensitive Hash, LSH) 和精细尺度连续过滤算法 (Fine Granularity Successive Elimination, FGSE)，利用广告的新颖的视频特征结合后处理技术开发出一套能够快速检测电视节目中是否存在数据库中广告的方法。

[0018] 本发明的目的是通过如下技术方案实现的。

[0019] 一种基于分层匹配的视频广告识别方法，其特征在于包含以下步骤：离线数据库建立与在线监测，其中离线数据库建立分为离线预处理与哈希表建立；在线监测部分分为在线预处理、由粗糙到精细的二级匹配和后处理。

[0020] 所述离线预处理包括数据库广告镜头分割与特征提取，具体内容包括：对于离线数据库中每一个广告，将每一帧由 RGB 颜色空间转化为灰度空间，将灰度图像从中心区域划分为 8 个 64*64 像素的小块，通过计算每个小块对应的灰度直方图之间的差异，乘以相应的权值并求和，得到帧间灰度变化率，如果这个变化率大于设定的阈值，则视为两帧之间发生了镜头的切变，并记录镜头变化的位置，将每一个广告分为一系列的镜头序列；

[0021] 对于每一个镜头，以镜头长度 25% 的位置作为区间起点，75% 的位置作为区间终点，提取这段区间内所有视频帧，对这些视频帧进行累加平均，获得代表镜头内容的平均关键帧，保留关键帧中心 160*160 像素区域的子图像，用作特征提取，在子图像中提取 36 维 HSV 颜色空间直方图与 64 维边缘角度直方图，以及边缘角度分布图特征，将每个图像的颜色直方图与边缘角度直方图分别归一化，组成 100 维特征向量用于建立局

部敏感哈希表，同时，将存储所有镜头的 100 维联合特征以及边缘角度分布图特征，用于在线监测时，与未知视频进行镜头分割后所提取的 100 维联合特征以及边缘角度分布图特征进行比较。

[0022] 所述离线建立哈希表包括：将获得的离线数据库中所有镜头产生的 100 维联合特征以及边缘角度分布图特征和标注信息，即与之对应的广告镜头名称和镜头编号，用于构造局部敏感性哈希表。设定子哈希表的个数（10 个）与哈希键长（键长为 20），对于每一个子哈希表产生过程为，根据初始设定的两个变量，利用随机函数产生不同的哈希系数，利用这些系数将 100 维联合特征映射到汉明空间，转化为 0 和 1 的组合序列，根据序列中的 0 和 1 的个数，选择不同的哈希表位置，将所有产生的子哈希表组成局部敏感性哈希表，将离线数据库中所有镜头产生的 100 维联合特征以及边缘角度分布图特征和与之对应的标注信息存入哈希表中。

[0023] 所述在线预处理，包括镜头分割与特征提取，具体内容包括：对于输入的视频文件，将每一帧由 RGB 颜色空间转化为灰度空间，将灰度图像从中心区域划分为 8 个 64*64 像素的小块，通过计算每个小块对应的灰度直方图之间的差异，乘以相应的权值并求和，得到帧间灰度变化率，如果这个变化率大于设定的阈值，则视为两帧之间发生了镜头的切变，并记录镜头变化的位置，将每一个广告分为一系列的镜头序列。对于每一个镜头，以镜头长度 25% 的位置作为区间起点，75% 的位置作为区间终点，提取这段区间内所有视频帧，对这些视频帧进行累加平均，获得代表镜头内容的平均关键帧，保留关键帧中心 160*160 像素区域的子图像，用作特征提取，在子图像中提取 36 维 HSV 颜色空间直方图与 64 维边缘角度直方图，以及边缘角度分布图特征，将每个图像的颜色直方图与边缘角度直方图分别归一化，组成 100 维特征向量，与边缘角度分布图特征一起，作为在离线数据库建立过程中产生的局部敏感性哈希表中进行搜索的查询向量。

[0024] 所述由粗糙到精细的二级匹配，包括在局部敏感性哈希表中查询与精细尺度连续过滤两部分，具体内容包括：将获得的一个镜头的联合特征向量通过局部敏感性哈希方法，将其映射到数据库的 10 个哈希表对应的条目中，获得与查询向量最为相近的 50 个候选向量及其对应的镜头与边缘分布图，比较这 50 个镜头与查询镜头包含的帧数的差异，如果相差小于 25 帧，就利用精细尺度连续过滤算法（F6SE）对二者分别对应的边缘角度分布图进行更为精细的匹配，在设定的阈值下，如果匹配成功，即可判断该镜头为数据库中某个广告的一部分，将数据库中对应的广告镜头的名称和序号标注到查询镜头上，如果输入的镜头是视频文件中的最后一个镜头，进行后处理，否则转否则读入下一个视频镜头。

[0025] 所述后处理内容包括：首先，由粗糙到精细的二级匹配过程将广告名称以及序号标注到了未知视频的查询镜头中，这些标注信息构成了标注序列，将标注序列中具有不同标注广告名称的镜头划分到若干个子段，对每一个子段，根据镜头在原始文件中的起始位置，在时间轴上依次排列，利用标注的广告镜头序号与起始点之间的位置关系消除其中的漏检与误检，获得连续完整的标注信息；经过纠正的子段，如果其播放长度大于数据库中该广告长度的一半，同时检测到的镜头数量大于数据库中广告镜头个数的三分之一，则认为标注成功，处理完所有子段后，对结果按照时间先后顺序排序，输出。

[0026] 所述平均关键帧计算方法，包括如下步骤：

[0027] (1) 判断镜头中包含的视频帧的个数，如果等于 1，将该帧作为平均关键帧，结束；否则，转步骤 (2)；

[0028] (2) 以镜头长度 25% 的位置作为区间起点，75% 的位置作为区间终点，提取这段区间内所有视频帧，其数量占镜头中视频帧总数的 50%；

[0029] (3) 累加图像像素的值，求平均，获得平均关键帧。

[0030] 所述平均关键帧中提取 64 维边缘角度直方图特征，其步骤如下：

[0031] (1) 将 RGB 彩色图像转化为灰度图像。在平均关键帧中，对同一幅图像分别从中心位置截取 160*161 和 161*160 像素的两张子图；

[0032] (2) 将 160*161 像素的子图像，按照从左到右的顺序，用每一个像素减去右方相邻像素，得到 160*160 的水平方向上的边缘分布 dx。将 161*160 像素的子图像，按照从上到下的顺序，用每一个像素减去下方相邻像素，得到 160*160 的垂直方向上的边缘分布 dy；

[0033] (3) 计算当 $dx \neq 0$ 且 $dy \neq 0$ 时，每个像素的边缘角度： $Ang = \arctan(dy/dx)$ ，得到 160*160 的边缘角度分布图。统计每个角度区间中像素的个数；

[0034] (4) 归一化，将特征中每一维除以该特征中的最大值，得到全局边缘角度的分布。

[0035] 所述在平均关键帧中提取边缘角度分布图特征，其步骤如下：

[0036] (1) 将 RGB 彩色图像转化为灰度图像。在平均关键帧中，对同一幅图像分别从中心位置截取 160*161 和 161*160 像素的两张子图；

[0037] (2) 将 160*161 像素的子图像，按照从左到右的顺序，用每一个像素减去右方相邻像素，得到 160*160 的水平方向上的边缘分布 dx。将 161*160 像素的子图像，按照从上到下的顺序，用每一个像素减去下方相邻像素，得到 160*160 的垂直方向上的边缘分布 dy；

[0038] (3) 计算当 $dx \neq 0$ 且 $dy \neq 0$ 时，每个像素的边缘角度： $Ang = \arctan(dy/dx)$ ，得到 160*160 的边缘角度图；

[0039] (4) 将边缘角度图平均分为 4*4 个子块，统计每一幅子图中，每个角度区间中的像素点的个数。每一个子块中的区间个数为 16，区间间隔为 360/16。

[0040] (5) 将每个子块中的 16 位一维直方图按照 4*4 的顺序重新排列，根据子图在边缘角度分布图中的位置，可以得到 16*16 维边缘角度分布图特征。

[0041] 所述将精细尺度连续过滤算法 FGSE，应用于图像识别中。其步骤如下：

[0042] (1) 初始化分割的层数 $l = 0$ ，子图像序号 order 为 0，子图像的大小 M 等于输入图像的高度乘以宽度，即 $n*n$ 。在第 0 层中子图像的个数 S_l 为 1，分割的总次数 $L = n*n+3$ ，初始化子图的高度与宽度均为 $m = n$ ；

[0043] (2) 计算两个图像中各个子图像对应的 BV_l ：

$$[0044] \quad BV_l = \sum_{k=0}^{S_l-1} |Q_l^{(k)} - R_l^{(k)}|$$

[0045] 其中 $Q_l^{(k)}$ 和 $R_l^{(k)}$ 分别对应查询图像和候选图像在第 l 层时对应的第 k 个子图像中各像素值的和；

[0046] (3) 如果 BV_l 大于等于阈值 SAD_{\min} 跳转至 (5)，否则置 $l = l+1$ 。对图像进行分

割，每次分割，仅对图像中的第一个未分割的子图像进行分割：

[0047] a. 将大小为 $m*m$ 的第 i 个子图分割为四个大小为 $\frac{m}{2} * \frac{m}{2}$ 的子图，更新 $S_i = S_i + 3$ ，order 中第 i 块及以前的块序号保持不变，后面的序号依次加 3；

[0048] b. 判断是否所有的长度为 $m*m$ 的子图像均完成了分割，如果完成，令 $i = 0$ ， $m = m/2$ ，否则 $i = i+4$ 指向下一个大小为 $m*m$ 的子图像；

[0049] (4) 重复 (2)，(3) 直到最后一层 L 分割完成，如果 $BV_i < SAD_{\min}$ ，则令 $SAD_{\min} = BV_i$ ；

[0050] (5) 检测是否还有待检测图像未被检测的候选图像，如果存在未被检测的图像，跳转至步骤 (1)，否则输出是否匹配的结果。

[0051] 所述后处理过程，还包括如下步骤：

[0052] (1) 统计标注结果中的不同名称的广告个数，并把具有相同广告名称的标注镜头放入同一子段；

[0053] (2) 根据标注镜头的在原视频中起始位置的时间信息，将标注镜头按照时间顺序排序，如果两个连续镜头之间存在未标注的视频帧，则在其中补充一个空白的镜头，其中记录未标注视频帧的数量与起始位置。该步骤名称为时间扩充；

[0054] (3) 得到了时间扩充的标注子段后，标注的时间序号与空白镜头之间存在着一定关系，如果两个相邻的镜头之间的序号连续，则记录这些正确的镜头，如果不连续且中间存在空白镜头，则计算空白镜头中包含的未标注视频帧的数量，并与标注镜头间所缺失的视频帧数量进行比较，如果两者差距在 25 帧以内，可以进行如下操作并记录这些纠正后的镜头：对于出现错误的 134、1034、1094、1324、1294、1023、1293 分别纠正为 1234、1234、1234、1234、1234、123 和 123；其中普通字体数字序号 1, 2, 3, 4 代表标注正确的时间序号，粗字体序号 9 代表错误标注的镜头，0 代表在时间扩充过程中添加的空白镜头，32 代表需交换的时间序号。

[0055] 如果两个镜头之间没有必然的联系，则认为其间存在错误，放弃记录；直到这个子段中所有镜头检测完毕为止；

[0056] (4) 根据数据库中广告的时间长度，判断该文件中可能存在的广告个数，广告个数为识别到的视频帧数量与数据库的对应广告的视频帧数量之比，并向上取整；根据获得的广告个数，将标注序列进一步分割为几个子序列，分别与阈值进行比较，判断广告是否存在；本方法中包含两个阈值，分别为数据库中广告所包含的视频帧数量的二分之一与镜头个数的三分之一；

[0057] (5) 判断是否所有不同广告名称的序列都已经被检测，检测未完转 (2)，检测完成转 (6)。

[0058] (6) 根据时间顺序将获得结果排列，输出。

[0059] 本发明的效果是：与现有的广告监测技术相比，本发明通过一种对广告视频进行优化的镜头分割算法，将视频分割为一组镜头序列，利用镜头中的一组帧提取特征，消除关键帧选取的影响，对提取的特征通过在局部敏感性哈希表中的检索，获得与检索特征相似的候选特征向量，并利用一种快速的特征匹配算法——精细尺度连续过滤技术，在本方法提出的边缘角度分布图之间进一步匹配获得精确的标注结果，在方法最后

通过后处理模块对标注信息进行校正。和以往广告监测方法不同的是，本发明以一种鲁棒的广告识别技术，通过一个由粗糙到精细的二级匹配过程逐渐滤除差异很大的候选特征，通过有效的后处理模块，获得最后的结果。本方法的优点是只需要很低的 CPU 负荷，就可以计算输入视频的特征，通过两层筛选过程有效的减少匹配次数，利用后处理进一步提高检测的正确率与查全率，并且通过对数据库构造局部敏感性哈希表，大大减少所需要的存储空间。具体来说，用户输入一段视频，本方法自动将这段视频分割为若干个镜头，在镜头中，利用中间 50% 的视频帧，提取 36 维 HSV 颜色空间直方图与 64 维边缘角度直方图，组成 100 维特征，在局部敏感性哈希表的映射空间中，查找与之最为相近的 K 个特征向量的位置。同时在镜头中，提取 16*16 的边缘角度分布图特征，查询镜头将与这些候选镜头利用边缘角度分布图在精细尺度连续过滤模块中进行比较，快速输出在规定阈值下与查询镜头最为近似的结果。在所有镜头的匹配进行完成后，利用后处理技术判断标注的内容是否可以组成一个相对完整的广告。

[0060] 为了验证本发明的有效性和优点，使用本发明对其识别的性能与鲁棒性进行了测试。我们从北京、安徽、四川等多个电视台录制了一定量的电视数据，存储为 240*320, 25fps 的 avi 文件。从中手工挑选了 170 个左右，长度从 5 秒至 120 秒不等的广告，建立离线数据库，数据库的存储空间大约在 635MB 左右。我们利用时间总长度近 4 小时的视频，作为测试文件，对方法的性能进行测试，要特别说明的是，测试文件中包含近 1 个小时的广告，这些广告并不简单的是数据库中广告的拷贝，其中的 20% 是在不同时间或不同频道中录制下来的，与数据库中颜色或边缘特征存在着差异。与其它方法相比，我们所设计的广告识别方案在大大减轻计算机与处理复杂度的情况下，取得了优异的识别性能，正确率与查全率均达到了 100%，通过数据库建立的哈希表与特征向量组所占用的存储空间仅为 2MB，节约了 99.9% 以上的数据存储空间。如附图 5 所示，我们进行了 FGSE 阈值对本方法识别性能影响的实验，其中黑色曲线代表正确率，灰色代表查全率。通过附图我们可以观察到，随着 FGSE 阈值的不断加大，本方法性能在阈值为 4800 时，达到最优；伴随着阈值增大，误检的广告个数增多，本方法性能缓慢下降。如附图 6 所示，我们对本方法的特征选取进行了测试，其中黑色曲线代表了本方法中所采用的颜色加边缘构造局部敏感哈希表的方法，浅灰色曲线代表单纯使用颜色构建哈希表的方法，灰色代表单纯使用边缘构造的方法。通过实验我们观察到，边缘特征比颜色特征具有更强的表达能力。在融合两种特征以后，我们的方法在查全率方面有显著的提高，而在正确率方面，由于小阈值可以为本方法带来更高的执行效率，通过观察实验结果，可以看到，融合的方法在阈值很小时具有优异的效果。我们同样对边缘角度分布图的性能进行了测试，如附图 7 所示，其中黑色曲线代表本方法所采用的边缘角度分布图的正确率与查全率，灰色曲线代表使用边缘角度直方图作为 FGSE 匹配特征时系统本方法的性能，通过实验，我们可以看到由于考虑了图像的空间信息，边缘角度分布图比一维边缘特征具有更好的效果。附图 8 是对本方法建立颜色边缘联合哈希表和独立建立颜色哈希表、边缘哈希表方法之间的对比。我们可以看到，采用联合建立哈希表的方法，在阈值很小的情况下可以得到更好的性能。同时我们对于本方法标注广告的起始点位置与真实的位置之间的差距进行了统计，经过本方法标注，起始点与真实情况完全一致的广告个数在 86.55% 左右，与实际位置差异在 0.4s 以内的广告个数占总体的在 96.7%

以上，相差 1s 以内的广告个数的比例则达到了 99% 以上。

[0061]	边界点 D 位置	起点 (个)	起点 (百分比)	终点 (个)	终点 (百分比)
[0062]	D = 0.0s	202	93.5	172	79.6
[0063]	0.0s < D ≤ 0.1s	7	3.2	23	10.6
[0064]	0.1s < D ≤ 0.2s	3	1.4	10	4.6
[0065]	0.2s < D ≤ 0.4s	1	0.5	4	1.9
[0066]	0.4s < D ≤ 1.0s	2	0.9	6	2.8
[0067]	D > 1.0s	1	0.5	1	0.5
[0068]		216	100	216	100

附图说明

- [0069] 图 1 本发明的工作流程框图；
- [0070] 图 2 局部敏感性哈希算法方案框图；
- [0071] 图 3 精细尺度连续过滤技术滤除过程示意图；
- [0072] 图 4 后处理方案框图；
- [0073] 图 5 本方法随阈值变化的整体性能实验图；
- [0074] 图 6 选取不同特征构造哈希表时的本方法性能实验图；
- [0075] 图 7 选取不同特征进行 FGSE 匹配时的本方法性能实验图；
- [0076] 图 8 利用不同方法构造哈希表时的本方法性能实验图；
- [0077] 图 9 本方法识别结果图。

具体实施方式

[0078] 下面结合附图和具体实施方式对本发明作进一步的描述。

[0079] 根据以上介绍的技术方案，我们可以很容易将本发明应用于广告监测中，为用户提供精确的广告识别服务。结合附图，我们对本发明的具体实施方式作详细阐述。

[0080] 1、广告镜头分割

[0081] 在本发明中，由于视频分割的目的并不是将视频分为语义单元，而是希望通过将相似的视频帧进行聚类，减少计算复杂度。本方法使用经过优化的单阈值突变镜头分割方法。其操作过程如下：

[0082] (1) 按帧的格式读取输入的视频文件，将读入的每一帧由 RGB 空间转化为灰度空间，剪切图像，仅保留图像中心位置 8 个 64*64 的小块。

[0083] (2) 计算连续的两帧之间所有对应的 64*64 小块的灰度直方图，并计算其差值。对于位于图像中心处的 4 个小块的差值乘以 0.6 的权重，其他的小块乘以 0.4 的权重，求和。得到两帧之间的直方图之差。

[0084] (3) 将差值与设定的阈值进行比较，如果大于阈值，证明该位置为切变发生位置。记录该位置。

[0085] (4) 转 (2)，继续比较视频中剩余的连续帧之间直方图的差值，直到所有帧之间比较完成。

[0086] 2、平均关键帧提取方法：

[0087] 为了避免关键帧选取的差异给识别方法带来影响，本方法采取利用平均关键帧的方法，表达镜头的内容。本方法中使用镜头中心位置向前 25% 和向后 25% 的视频帧构造平均关键帧

[0088] (1) 判断镜头中包含的视频帧的个数，如果等于 1，将该帧作为平均关键帧，结束。否则，转 (2)

[0089] (2) 以镜头 25% 的位置作为区间起点，75% 的位置作为区间终点，提取这段区间内所有视频帧（如果提取的视频帧数量包含小数，则向上取整），其数量占镜头中视频帧总数的 50%。

[0090] (3) 累加图像像素的值，求平均，获得平均关键帧。

[0091] 3、边缘角度直方图提取

[0092] 考虑到不同电视台在播放相同广告时会其中加入一些滚动字幕或数字时钟，我们仅提取平均关键帧中心位置的 160*161 和 161*160 像素的区域作为子图像，提取特征。64 维边缘角度直方图提取过程如下：

[0093] (1) 将 RGB 彩色图像转化为灰度图像。在平均关键帧中，对同一幅图像分别从中心位置截取 160*161 和 161*160 像素的两张子图。

[0094] (2) 将 160*161 像素的子图像，按照从左到右的顺序，用每一个像素减去右方相邻像素，得到 160*160 的水平方向上的边缘分布 dx 。将 161*160 像素的子图像，按照从上到下的顺序，用每一个像素减去下方相邻像素，得到 160*160 的垂直方向上的边缘分布 dy 。

[0095] (3) 计算当 $dx \neq 0$ 且 $dy \neq 0$ 时，每个像素的边缘角度： $Ang = \arctan(dy/dx)$ 。统计每个角度区间中像素的个数，本方法采用的为 64 维特征，故区间的间隔为 360/64。

[0096] (4) 归一化，将特征中每一维除以该特征中的最大值，得到全局边缘角度的分布。

[0097] 4、特征合并：

[0098] 本方法采用 36 维 HSV 颜色直方图与 64 维边缘角度直方图作为在局部敏感哈希算法的输入向量，在平均关键帧中心位置的 160*160 像素的区域作为子图像，提取归一化的 36 维 HSV 颜色直方图特征。将两个经过归一化的特征按顺序拼接成 100 维联合特征，并将其扩大 10000 倍，以便在局部敏感哈希算法中得到更好的效果。

[0099] 5、边缘角度分布图的提取：

[0100] (1) 将 RGB 彩色图像转化为灰度图像。在平均关键帧中，对同一幅图像分别从中心位置截取 160*161 和 161*160 像素的两张子图。

[0101] (2) 将 160*161 像素的子图像，按照从左到右的顺序，用每一个像素减去右方相邻像素，得到 160*160 的水平方向上的边缘分布 dx 。将 161*160 像素的子图像，按照从上到下的顺序，用每一个像素减去下方相邻像素，得到 160*160 的垂直方向上的边缘分布 dy 。

[0102] (3) 计算当 $dx \neq 0$ 且 $dy \neq 0$ 时，每个像素的边缘角度： $Ang = \arctan(dy/dx)$ ，得到 160*160 的边缘角度图。

[0103] (4) 将边缘角度图平均分为 4*4 个子块，统计每一幅子图中，每个角度区间中的像素点的个数。每一个子块中的区间个数为 16，区间间隔为 360/16。

[0104] (5) 将每个子块中的 16 位一维直方图按照 4*4 的顺序重新排列, 根据子图在边缘角度分布图中的位置, 可以得到 16*16 维边缘角度分布图特征。

[0105] 6、局部敏感性哈希算法:

[0106] 在高维特征空间中, 由于“维数灾难”的原因, 对于一个查询, 寻找一个精确的匹配往往需要耗费大量的计算, 直接影响了索引的速度。局部敏感性哈希算法提供了一种快速的近似性返回机制, 而且索引速度不会因为维数的增长发生过多的恶化。局部敏感性哈希算法的主要思路是使用不同的局部敏感性哈希函数, 根据特征之间的相似性, 将特征映射到多个哈希表的不同条目中去。这些哈希表组成构成一个局部敏感性哈希表。如果两个特征被分到一个条目中的概率越大, 就说明这两个特征向量越接近。

[0107] 假设查询向量为 Q_m , 局部敏感性哈希函数可以将该特征向量通过第 i 个哈希表的哈希函数 $h_i(\cdot)$, 映射到条目 $h_i(Q_m)$ 中。假设若干个哈希函数 $h_i(\cdot)$ 构成了局部敏感性哈希函数族 H , 对于查询向量 Q_m 与数据库中向量 R 在距离测度 l 下, 可以得到局部敏感性哈希的基本数学思想:

[0108] $\text{If} \|Q-R\|_l \leq r_1, P_{\text{hit}}[h(Q) = h(R)] \geq p_1,$

[0109] $\text{If} \|Q-R\|_l > r_2, P_{\text{hit}}[h(Q) = h(R)] \leq p_2.$

[0110] 其中 r_1, r_2, p_1 和 p_2 为常数, 且 $p_1 > p_2, r_1 < r_2$ 。对于函数族 H 它保证了在一定距离测度下, 向量 Q 和 R 的距离小于一定的阈值时, 其落入同一条目的概率大于 p_1 。在本系统方法中, 使用的距离测度为欧式距离, 那么两个向量之间的相似度度量方法为:

$\sum_{i=1}^n |Q(i) - R(i)|$ 。对于查询向量 Q 映射到第 i 个哈希表的条目 $h_i(Q_m)$ 中, 其中有相似的特征向量 R_{ij} , 我们将所有哈希表对应条目中的所有相似特征向量集合在一起, 选出最相近的 K 个候选特征向量。

[0111] 建立局部敏感性哈希表的操作过程如下:

[0112] (1) 获得数据库的特征向量组, 将特征向量数组的各个特征值整数化, 初始化局部敏感性哈希表中包含的哈希表的个数, $l = 10$, 哈希主键 (Hash Key) 的长度, $hk = 20$, 以及特征向量数组中的最大值 C 。

[0113] (2) 将每一个特征向量 $p_j = (x_1, \dots, x_{100})$ 映射到汉明空间 $H^{d'}$ 中, 其中 $d' = C * 100$, 映射为 $p_j' = \text{Unary}_C(x_1), \dots, \text{Unary}_C(x_{100})$, 其中 $\text{Unary}_C(x)$ 表示对于特征向量中将每一维的特征值 x 映射为 x 个 1 与 $C-x$ 个 0 的序列组合。

[0114] (3) 对于 $i = 1, \dots, l$, 将映射后的每一个特征向量通过哈希表 T_i 的哈希函数 $g_i(\cdot)$, 根据 hk 的值映射到对应的条目中存储, 构建哈希表。

[0115] 在 LSH 哈希表中查询近似向量的过程, 如附图 2 所示:

[0116] (1) 设定返回的相似向量的个数 K , 本方法设置为 50。设置相似向量的缓冲区 $S = \phi$ 。

[0117] (2) 对于输入的查询向量 q , 映射到汉明空间 q' 。

[0118] (3) 进入每一个哈希表 $T_i, i = 1, \dots, l$ 。利用各个哈希表的哈希函数 $g_i(\cdot)$ 将 q' 映射到哈希表的条目中, 将条目中的向量存入 $S \leftarrow S \cup \{p \text{ in } g_i(q) \text{ of } T_i\}$ 。

[0119] (4) 在向量缓冲区 S 中使用最近邻线性搜索算法, 寻找 50 个最相近的向量并返回这些向量。

[0120] 7、精细尺度连续过滤技术算法

[0121] 精细尺度连续过滤技术应用于视频压缩中，宏块间寻找最优的运动向量，本方法将该技术改进后用于快速的模式匹配过程。假设 Q 和 R 分别表示查询与候选的边缘角度分布图，两个图像相互之间每一像素的差值和为：

$$SAD = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |Q(i, j) - R(i, j)|,$$

用 Q_0 和 R_0 表示两个边缘角度分布图各像素之和，根据不等式 $|a+b| \leq |a|+|b|$ ，可以得到 $|Q_0 - R_0| \leq SAD$ 。其中：

$$[0122] \quad Q_0 = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} Q(i, j)$$

$$[0123] \quad R_0 = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} R(i, j)$$

[0124] 假设 SAD_{\min} 为阈值，如果出现 $|Q_0 - R_0| \geq SAD_{\min}$ ，就可以从这一等级判断两个边缘图是不匹配的。通过对图像进行不断的分块，并且计算所有子块差值的和，在不同的分段层次 l 和 $l+1$ 时，差值和有如下关系：

$$[0125] \quad BV_l = \sum_{k=0}^{S_l-1} |Q_l^{(k)} - R_l^{(k)}|$$

$$[0126] \quad BV_{l+1} = \sum_{k=0}^{S_{l+1}-1} |Q_{l+1}^{(k)} - R_{l+1}^{(k)}|$$

$$[0127] \quad BV_{l+1} = \sum_{k=0}^m |Q_l^{(k)} - R_l^{(k)}| + \sum_{k=m+4}^{S_{l+1}-1} |Q_l^{(k)} - R_l^{(k)}|$$

$$[0128] \quad \text{因为：} + \sum_{i=m}^{m+3} |Q_{l+1}^{(i)} - R_{l+1}^{(i)}|$$

[0129] 所以： $BV_l \leq BV_{l+1}$

[0130] 可以看到在不断的细分过程中，差值和 BV 是不断的增大，如果 $BV_l \leq BV_{l+1} \geq SAD_{\min}$ ，就可以滤除这些不匹配的边缘角度分布图。只有最优匹配的图像，可以细分到图像的像素水平，并且保证 $BV < SAD_{\min}$ 。

[0131] 图 3 给出了本方案的滤除原理图以及分块示例过程：

[0132] (1) 初始化分割的层数 $l = 0$ ，子图像序号 order 为 0，子图像的大小 M 等于输入图像的高度乘以宽度，即 $n*n$ 。在第 0 层中子图像的个数 S_l 为 1，分割的总次数 $L = n*n+3$ ，初始化每次分块的大小 $m = n$ 。

[0133] (2) 计算两个图像中各个子图像对应的 BV_l ：

$$[0134] \quad BV_l = \sum_{k=0}^{S_l-1} |Q_l^{(k)} - R_l^{(k)}|$$

[0135] 其中 $Q_l^{(k)}$ 和 $R_l^{(k)}$ 分别对应查询图像和数据库图像在第 l 层时对应的第 k 个子图像中各像素值的和。

[0136] (3) 如果 BV_l 大于等于阈值 SAD_{\min} 跳转至 (5)，否则置 $l = l+1$ 。对图像进行分割。每次分割，仅对图像中的第一个未分割的子图像进行分割：

[0137] a. 将大小为 $m*m$ 的第 i 个子图分割为四个大小为 $\frac{m}{2} * \frac{m}{2}$ 的子图，更新 $S_l = S_l+3$ ，

order 中第 i 块及以前的块序号保持不变，后面的序号依次加 3。

[0138] b. 判断是否所有的长度为 $m*m$ 的子图像均完成了分割，如果完成，令 $i = 0$ ， $m = m/2$ ，否则 $i = i+4$ 指向下一个大小为 $m*m$ 的子图像。

[0139] (4) 重复 (2)，(3) 直到最后一层 L 分割完成，如果 $BV_1 < SAD_{min}$ ，则令 $SAD_{min} = BV_1$ 。

[0140] (5) 检测是否还有待检测图像未被检测，如果存在未被检测的输入图像，跳转至 (1)，否则输出是否匹配的结果。

[0141] 7、后处理实施方案

[0142] 图 4 给出了本方案操作过程

[0143] (1) 统计标注结果中的不同名称的广告个数，并把具有相同广告名称的标注镜头放入同一子段。

[0144] (2) 根据标注镜头的在原视频中起始位置的时间信息，将标注镜头按照时间顺序排序，如果两个连续镜头之间存在未标注的视频帧，则在其中补充一个空白的镜头，其中记录未标注视频帧的数量与起始位置。该步骤名称为时间扩充。

[0145] (3) 得到了时间扩充的子段后，标注的时间序号与空白镜头之间存在一定关系，如果两个相邻的镜头之间的序号连续，我们记录这些正确的镜头，如果不连续且中间存在空白镜头，我们计算空白镜头中包含的未标注视频帧的数量，并与标注镜头间所缺失的视频帧数量进行比较，如果两者差距在 25 帧以内，可以进行如下操作并记录这些纠正后的镜头：

[0146]

错误情况	纠正前的序号排列	纠正后
漏检	1 3 4	1 2 3 4
漏检	1 0 3 4	1 2 3 4
漏检	1 0 9 4	1 2 3 4
交叉错误	1 3 2 4	1 2 3 4
误检	1 2 9 4	1 2 3 4
误检	1 0 2 3	1 2 3
误检 r	1 2 9 3	1 2 3

[0147] 其中普通字体数字序号 1, 2, 3, 4 代表标注正确的时间序号，粗体字序号 9 代表错误标注的时间序号，0 代表在时间扩充过程中添加的空白镜头，32 代表需交换的时间序号。如果两个镜头之间没有必然的联系，则认为其间存在错误，放弃记录。直到这个子段中所有镜头检测完毕为止。

[0148] (4) 根据数据库中广告的时间长度，判断该文件中可能存在的广告个数，广告个数为识别到的视频帧数量与数据库的对应广告的视频帧数量之比，并向上取整。根据获得的广告个数，本方法将标注序列进一步分割为几个子序列，分别与阈值进行比较，判断广告是否存在。本方法中包含两个阈值，分别为数据库中广告所包含的视频帧数量的二分之一与镜头个数的三分之一。

[0149] (5) 判断是否所有不同广告名称的序列都已经被检测，检测未完转 (2)，检测完成转 (6)。

[0150] (6) 根据时间顺序将获得结果排列，输出。

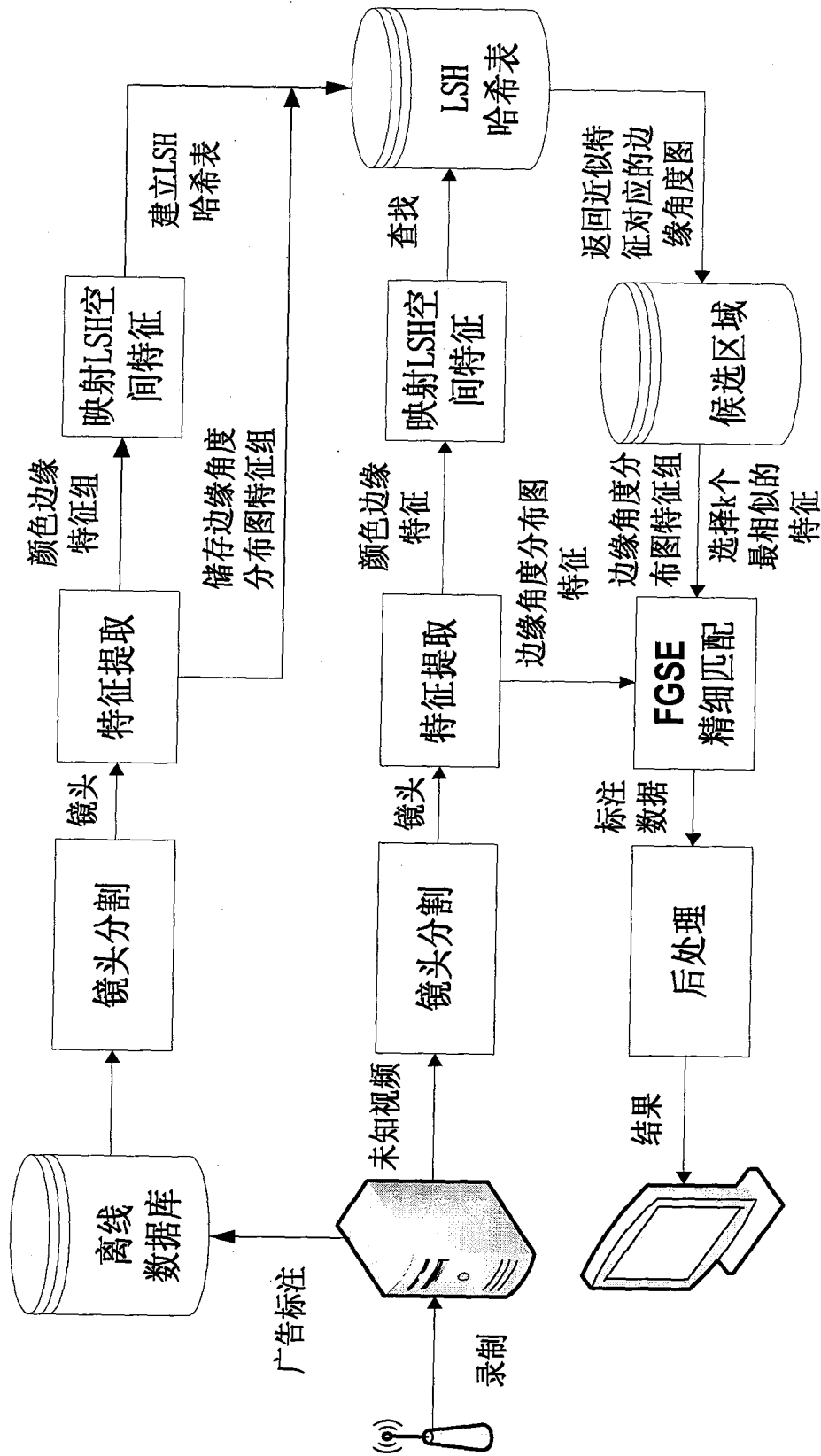


图 1

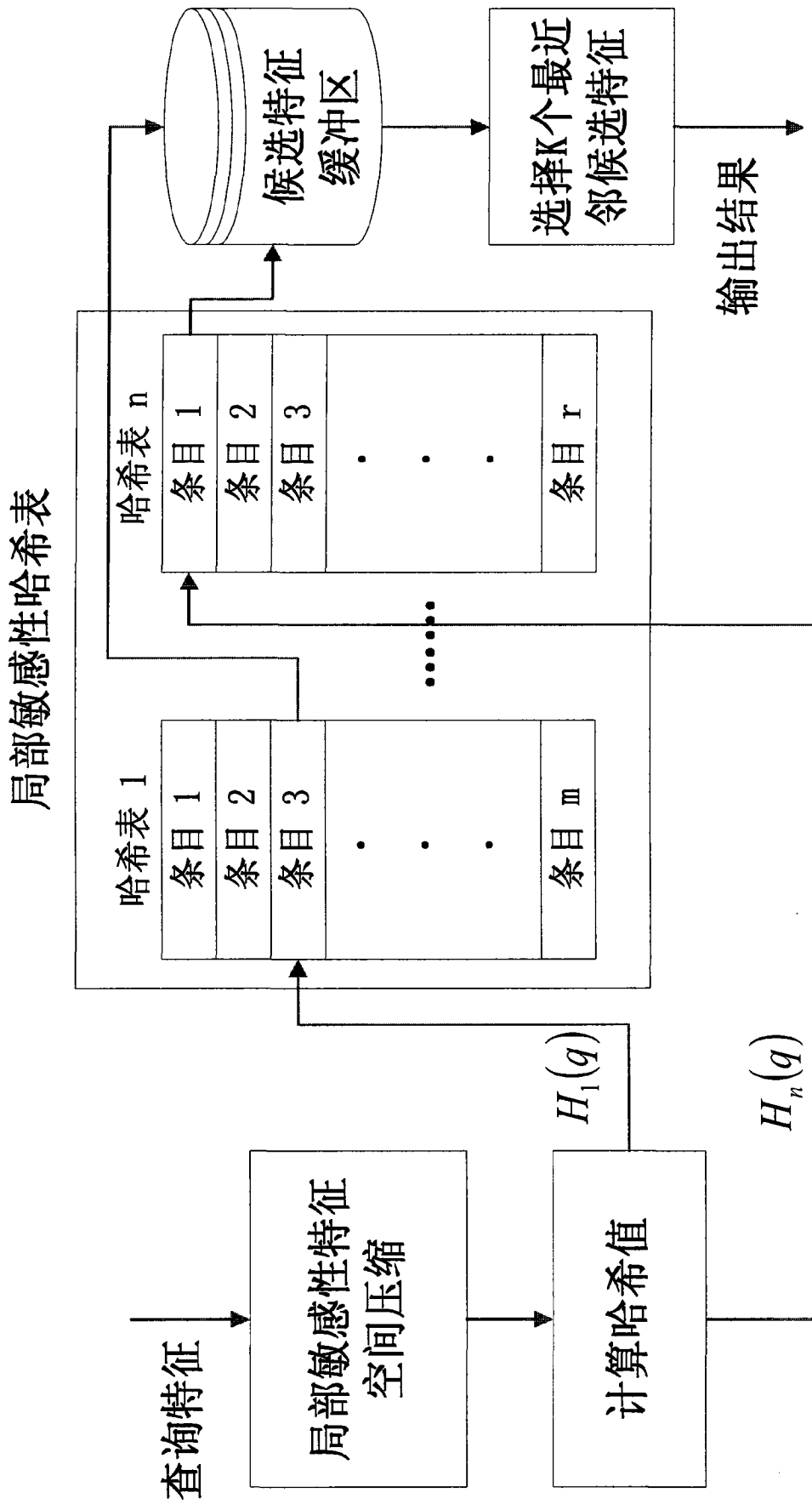


图 2

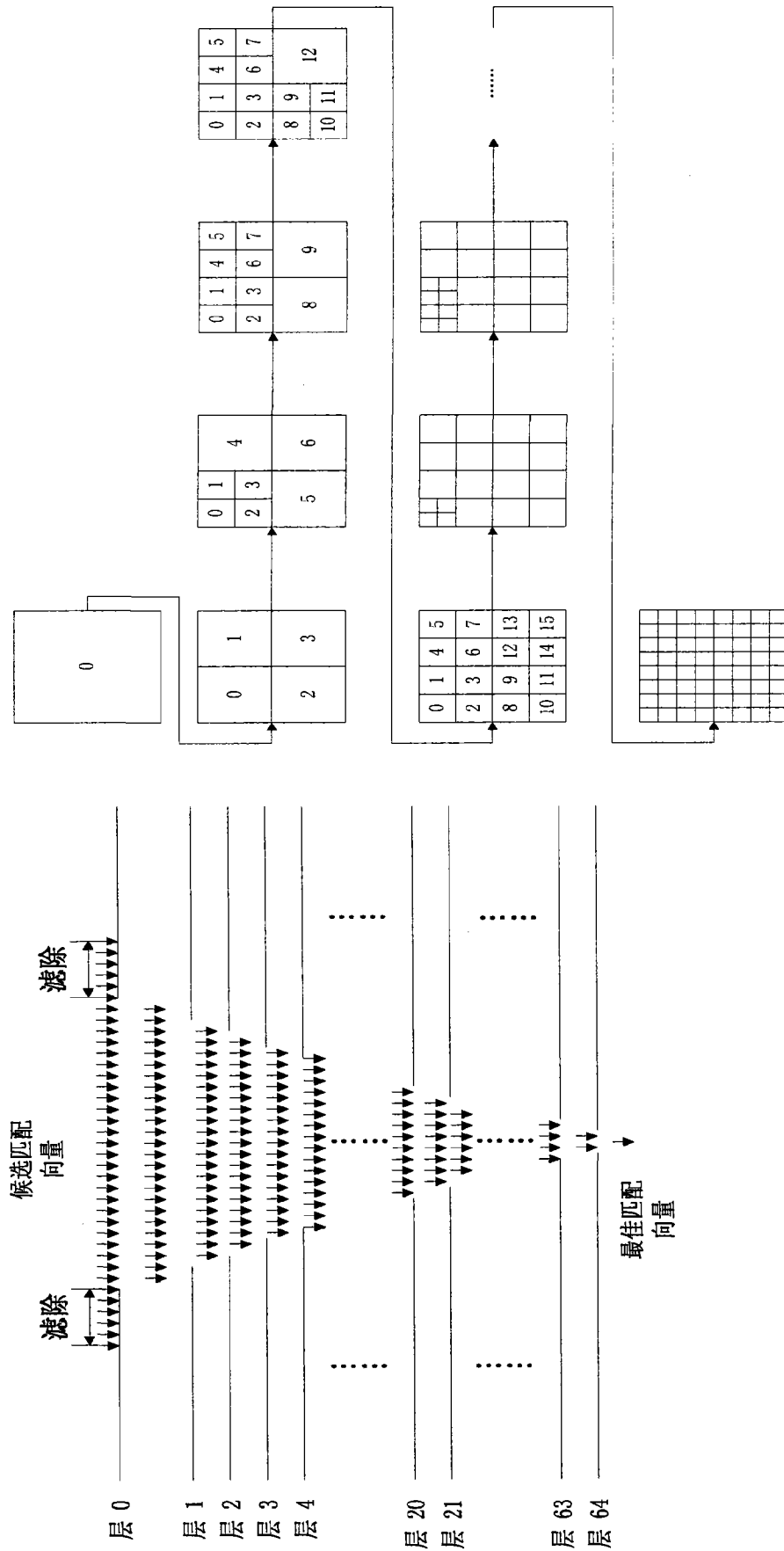
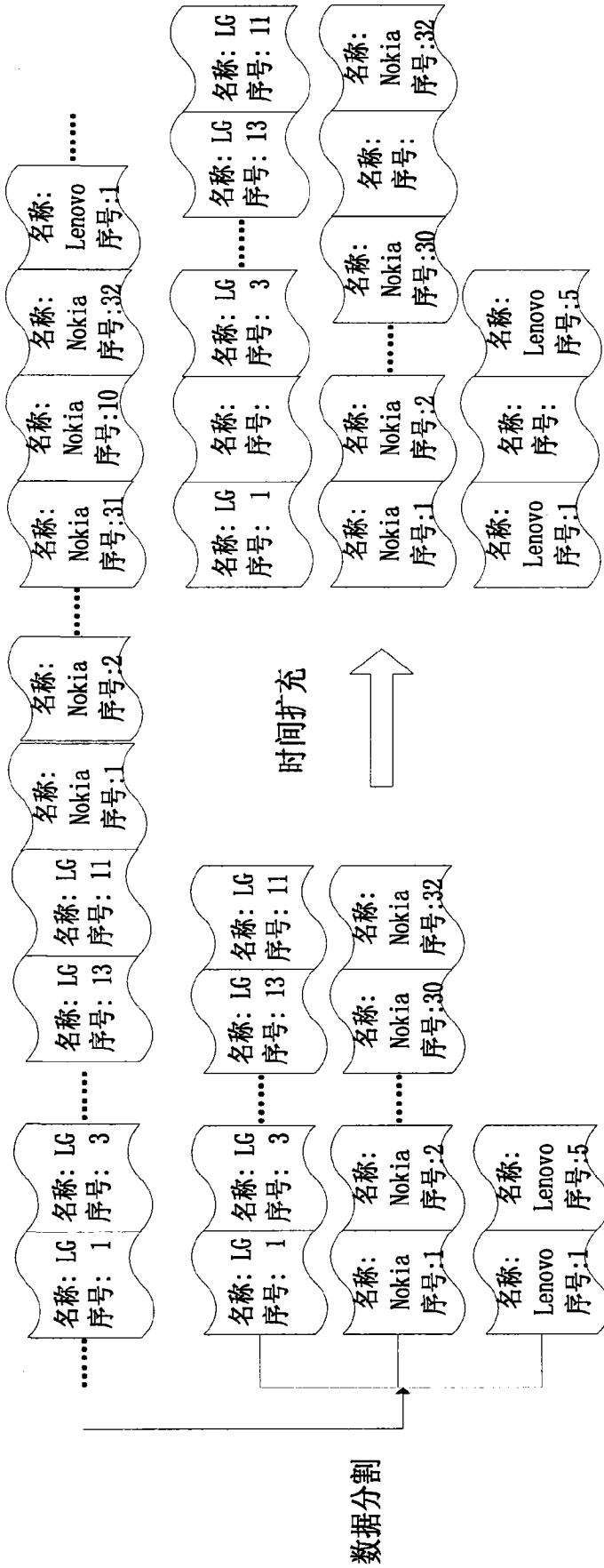


图 3

原始标注数据



后处理后的结果

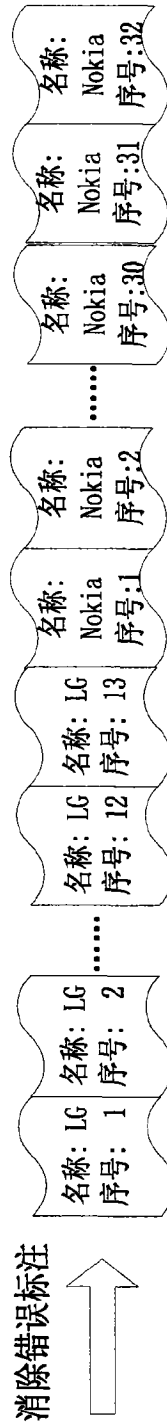


图 4

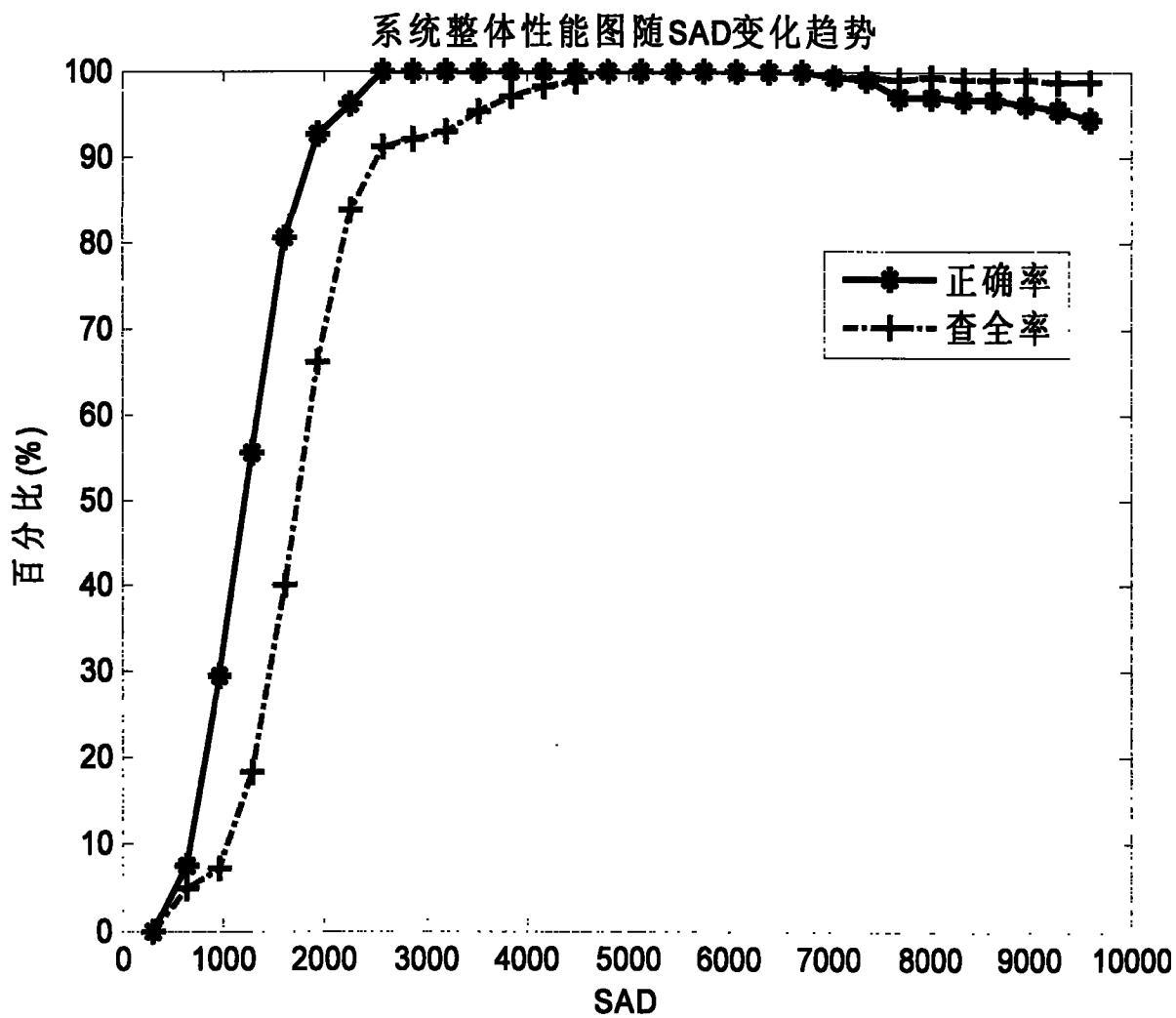


图 5

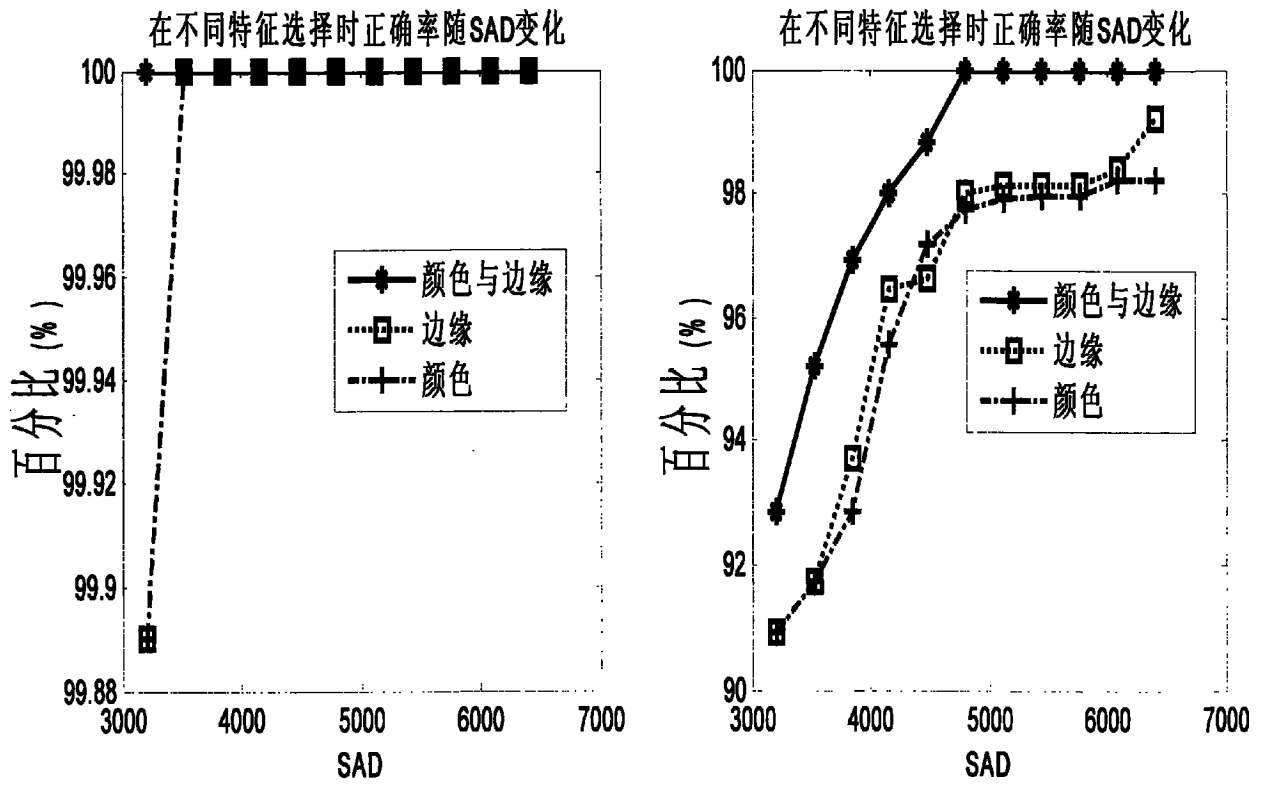


图 6

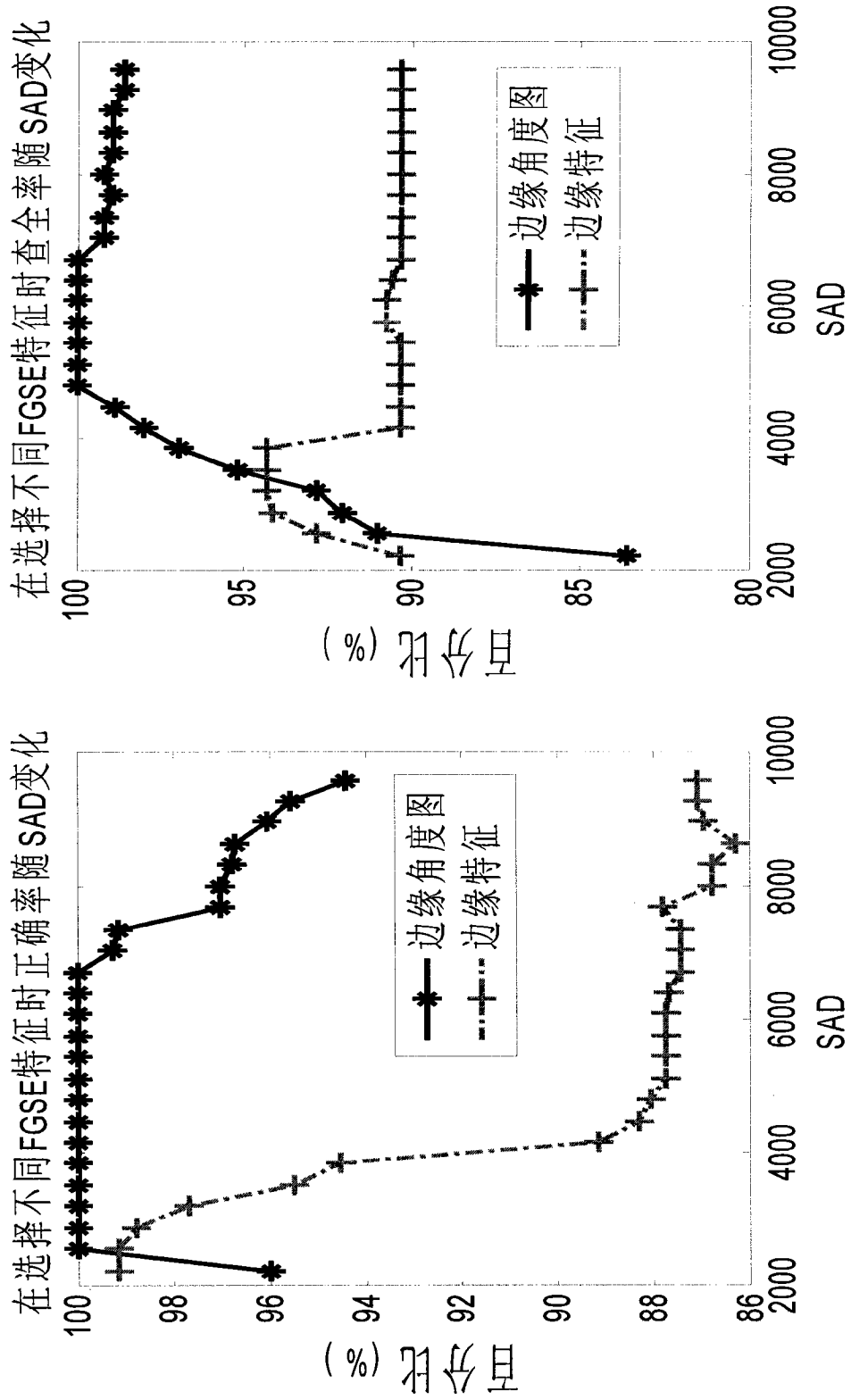


图7

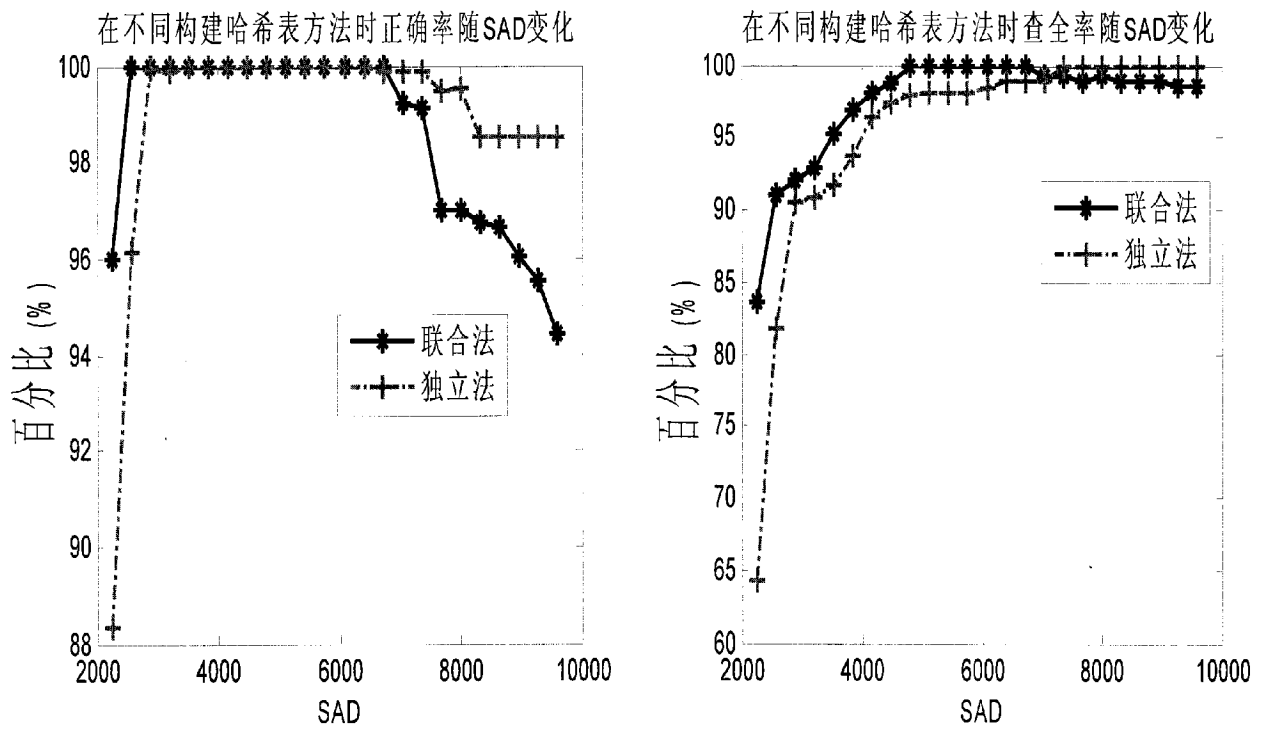


图 8

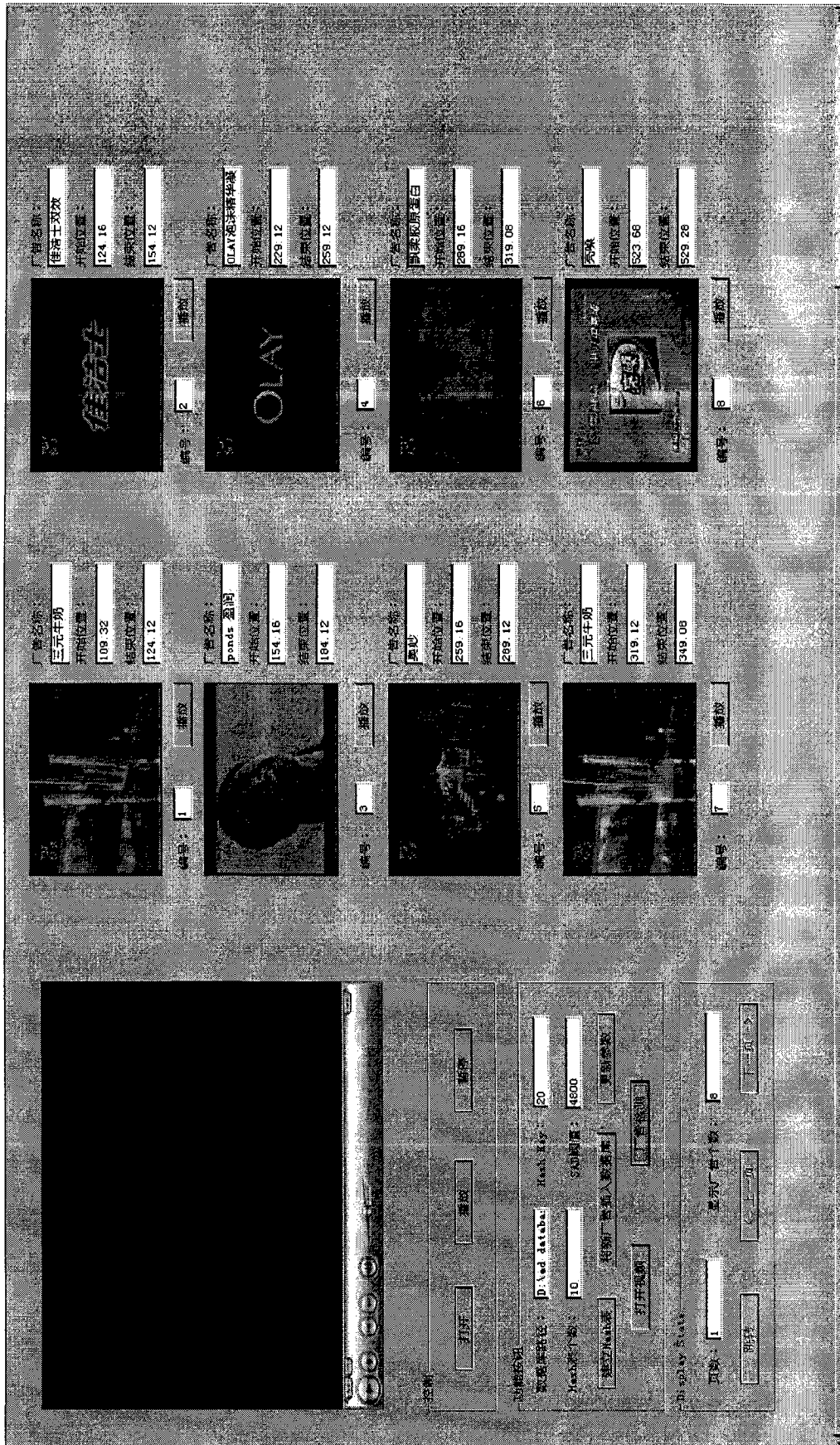


图 9