

基于动态阈值的视频场景切换检测^①



孙孟寒¹, 任维政²

¹(北京邮电大学 电子工程学院, 北京 100876)

²(北京邮电大学 自动化学院, 北京 100876)

通信作者: 任维政, E-mail: renwz@bupt.edu.cn

摘要: 视频场景切换检测在视频处理领域十分重要, 优秀的视频场景切换检测算法对于视频的后续处理, 包括信息标注与特征提取等都具有很大的意义. 本文以传统的图像处理方法为基础, 结合 ORB (oriented FAST and rotated BRIEF) 快速特征描述子, 分别实现图像的相似度估计. 通过动态融合因子, 将多种基本算法得到的相似度在完整算法中的价值量化, 并根据邻帧相似度计算动态阈值, 从而提高了算法的稳定性; 另外通过流程规划, 减少了算法中耗时长的部分的调用次数, 最终实现了高效的视频场景切换检测. 实验结果表明, 视频场景切换检测的准确率有一定程度地提升.

关键词: 场景切换检测; 图像特征; ORB; 特征点提取; 相似度估计; 动态因子

引用格式: 孙孟寒, 任维政. 基于动态阈值的视频场景切换检测. 计算机系统应用, 2023, 32(2): 234-241. <http://www.c-s-a.org.cn/1003-3254/9004.html>

Video Scene Switching Detection Based on Dynamic Threshold

SUN Meng-Han¹, REN Wei-Zheng²

¹(School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China)

²(School of Automation, Beijing University of Posts and Telecommunications, Beijing 100876, China)

Abstract: Video scene switching detection is important in video processing. An excellent algorithm for video scene switching detection is of great significance for subsequent video processing, including information annotation and feature extraction. In this study, on the basis of the traditional image processing methods and oriented FAST and rotated BRIEF (ORB), the similarity estimation of images is realized separately. Through the dynamic fusion factor, the value of similarity obtained by various basic algorithms in the complete algorithm is quantified, and the dynamic threshold is calculated according to the similarity of neighboring frames to improve the stability of the algorithm. In addition, by flow planning, the number of calls of the time-consuming part of the algorithm is reduced, and efficient video scene switching detection is finally realized. The experimental results show that the accuracy of video scene switching detection is improved to a certain extent.

Key words: scene switching detection; image features; oriented FAST and rotated BRIEF (ORB); feature point extraction; similarity estimation; dynamic factor

在社会高度发展的今天, 视频已经成为人们生活中不可或缺的一部分. 大小机构也开始了对视频内容的解构分析与对价值的深度挖掘, 例如视频索引建立、视频分类、视频分割、视频信息提取等. 而对视

频场景的切换进行检测往往是进行各种操作的基础, 好的视频场景切换检测算法可以为视频下一步的处理提供更优秀的更具有识别性的基本数据集. 所以对视频场景切换检测方法的研究有很大的意义.

① 收稿时间: 2022-06-13; 修改时间: 2022-07-12, 2022-08-15, 2022-09-27; 采用时间: 2022-09-30; csa 在线出版时间: 2022-12-09

CNKI 网络首发时间: 2022-12-13

近些年来,随着机器学习算法的发展,基于学习和模型的方法在该领域也取得了突出的成果.有学者通过三重损失学习的方法,将视频场景切换的问题转换为对场景的二分类问题^[1];还有学者将深度神经网络-隐马尔科夫等模型用于视频场景分割算法^[2],同样获得了优秀的检测效果;文献[3]结合CNN和RNN的能力,突破了单一架构的局限性;文献[4]中通过利用图像的正交矩作为特征值,同时采用快速块处理方法和嵌入正交多项式算法进行特征提取从而提高了算法处理速度.既然基于机器学习的方法进步得如此迅猛,对传统算法的学习与研究的意义是什么呢?首先机器学习算法为了服务于特定的场景一般都需要经过大量的数据训练,同时针对不同的使用场景,需要的训练集合也不尽相同,而利用图像的基本特征的传统算法无需训练,可以做到拿来即用;另外机器学习算法也可以参考传统算法中的一些内容,来建立更加完善的模型.所以对传统算法的研究仍然有很大的意义^[5].

传统算法的做法一般是对相邻图像的信息进行定量表示,并对定量信息进行相似度估计,最终根据设定的阈值进行场景切换判断.将信息量化表示的经典方法有像素比较法、直方图比较法、基于边缘的方法等^[6].其中像素比较法抗干扰性差,容易受噪声影响;直方图比较法较为稳定,但容易忽略图像的纹理信息;基于边缘的方法运算量大,时效性差.

为更好地使用传统算法实现视频场景切换的检测,国内外研究人员采用了各种不同的处理方式.文献[7]在使用基于直方图的方法时,通过在图像上建立矩形环的方式,在一定程度上保留了图像的空间特征.而文献[8]中则提出了一种基于ORB的场景切换检测方法,证明了基于图像特征点判断场景切换的可行性.

根据上述情况,笔者结合视频图像中的颜色信息与特征点信息^[9],设计了一套动态因子,通过使用这些动态因子,可以缓解各种基本方法的缺陷给算法带来的影响;同时动态规划算法的执行步骤,最终实现一种融合多种方法的规范化处理流程.经过实验验证,该方法的速度稳定,且拥有较好的检测效果.

1 基本特征相似度估计

1.1 HSV 相似度估计

HSV是图像的一种三维的颜色空间模型,由色调(H)、饱和度(S)、明度(V)这3个参数对颜色进行综合描述.HSV颜色空间可以通过RGB线性颜色空间变

换得到,人眼对该种模型的感知度较高^[10].由于邻帧图像的色调与饱和度变化的参考性较强,而明度变化的参考性最弱,为保证方法效率,计算方法保留色调与饱和度两个属性.将两个属性按照数值各自分为16个等长区间,并根据式(1)确定像素所属的直方图坐标(直方图宽度为256).式(1)中 P 为像素点的直方图坐标, H 为将色调按大小均分为16份后,像素点的色调值, S 为将饱和度按大小均分为16份后,像素点的饱和度值.

$$P = H \times 16 + S \quad (1)$$

计算图像中各个像素点的直方图坐标,最终可以生成图像的颜色特征直方图.对于两个不同图像的颜色直方图,采用式(2)计算图像的色彩相似度 S_h .相似度越接近1,代表图像越相似,相似度越接近0,代表图像间差异越大.其中 n 为直方图宽度, a_i, b_i 分别为两幅图像的颜色直方图在点 i 上的值.

$$S_h = \frac{1}{n} \times \sum_{i=1}^n 1 - \frac{|a_i - b_i|}{\max(a_i, b_i)} \quad (2)$$

基于图像颜色信息的直方图相似度估计方法具有抗干扰性强的特点,但使用该方法会忽略图像纹理信息,造成颜色相似的视频切换点的漏判.

1.2 哈希相似度估计

该方法的基本概念是使用感知哈希的方式生成相邻图像的指纹对,并对生成的指纹对进行基于汉明距离的相似度估计.

这里使用的感知哈希算法流程为:

(1) 首先将图像进行压缩,压缩至 32×32 ,从而消除图片尺寸比例带来的差异,并在一定程度上减小高频和细节上的差异所造成的影响.

(2) 将图像的三色通道转换成单一灰色通道,并提取颜色信息生成 32×32 大小的灰度矩阵.

(3) 对灰度矩阵使用DCT(离散余弦变换)进行有损的图像压缩,提取矩阵左上角 8×8 的矩阵作为最终的参考矩阵.

(4) 求出该区域内的平均灰度值.

(5) 根据矩阵中各点的灰度值与平均灰度的大小关系,确定二进制数的该位信息,从而生成64位指纹.

得到图像的指纹之后,可以根据指纹的汉明距离对图像进行相似度估计,采用式(3)计算图像的纹理相似度 S_p .相似度越接近1,代表图像越相似,相似度越接近0,代表图像差异越大,其中 n 为指纹的长度.

$$S_p = \frac{1}{n} \sum \begin{cases} 0, & \text{different} \\ 1, & \text{same} \end{cases} \quad (3)$$

该方法可以在一定程度上保留图像的纹理信息,但对旋转放缩等类型的图像剧烈变化,稳定性远不如基于颜色的直方图相似度估计方法,可能会造成误判。

1.3 融合因子与动态阈值

由于视频的种类繁多,且相同种类的视频也会有不同的背景变动情况,所以对于不同视频应选用不同的相似度阈值进行场景切换的判定.文献[11]中通过建立动态阈值的方法,实现针对不同视频的使用不同的阈值.参考该方法,笔者结合本文算法,重新设计了动态因子。

由于相邻帧有很大可能具有相似的特征,所以帧图像的相似度参考值 S 也应该考虑该帧图像的前置区间内各帧的图像相似度,故在完成每幅图像的场景切换判定后,根据式(4),计算新相似度参考值.这里的相似度参考值即为判断该帧图像是否经过场景变换的一个参考值.其中 S_o 为旧的相似度参考值,而 S_n 为最新帧的图像相似度, a 为传播因子, a 越小代表过去的帧图像的相似度对相似度参考值的影响越小.经过测试, a 值选取 0.9 效果较好,选取 0.9 时,当前帧向前 20 帧的帧图像的相似度对相似度参考值的影响值小于 $0.9^{20} \approx 12\%$,且旧帧与当前帧越接近,其相似性的对参考值的影响也越强。

$$S = S_o \times a + S_n \times (1 - a) \quad (0 \leq a \leq 1) \quad (4)$$

考虑到基于颜色信息与纹理信息对图像进行相似度评估的方法各自的优点,将使用第 1.1 节与第 1.2 节中方法计算得到的相似度做融合处理,根据式(5)对相似度进行融合.其中 M 为融合因子, M 越高代表基于颜色信息的相似度 S_h 参考性越强,基于纹理信息的相似度 S_p 参考性越弱。

$$S_n = M \times S_h + (1 - M) \times S_p \quad (0 \leq M \leq 1) \quad (5)$$

实验发现, M 的优秀值并不固定,与两种方法计算得到的相似度的可靠性有关,即使用颜色直方图方法得到的相似度的可靠性越高,则在融合时该方法应该占有的比重就越大,于是引入动态融合因子的概念.为此,首先将式(4)中的相似度参考值进行概念上的扩展,见式(6).其中 C_a 、 C_j 、 T_a 、 T_j 分别代表颜色特征参考值、颜色特征跳变参考值、纹理特征参考值、纹理特征跳变参考值.跳变的判定见后文。

$$\begin{cases} C_a = C_a \times a + S_h \times (1 - a) \\ C_j = \begin{cases} C_j \times a + S_h \times (1 - a), & \text{可能跳变} \\ C_j, & \text{不可能跳变} \end{cases} \\ T_a = T_a \times a + S_p \times (1 - a) \\ T_j = \begin{cases} T_j \times a + S_p \times (1 - a), & \text{可能跳变} \\ T_j, & \text{不可能跳变} \end{cases} \end{cases} \quad (6)$$

拥有各项参考值之后,可通过式(7)得到融合因子:

$$M = \frac{C_a / C_j}{C_a / C_j + T_a / T_j} \quad (7)$$

同样我们可以根据式(8)计算获得相似度参考值:

$$S = M \times C_a + (1 - M) \times T_a \quad (8)$$

至此我们得到了帧图像相似度,与该图像对应的相似度参考值.根据式(9)进行场景切换的判断.其中 Q 为阈值因子,该因子决定了判断场景切换的严格程度, Q 值越大表示对于场景切换的判定越严格. S_n 即为上文中提到的该帧图像的相似度。

$$\begin{cases} \text{不存在跳变, } S_n > S \times Q \\ \text{存在跳变, } S_n \leq S \times Q \end{cases} \quad (9)$$

实验发现, Q 值在选取定值时的效果并不好,所以引入动态阈值 Q 进行判断.动态阈值的计算参考式(10):

$$Q = M \times \frac{C_j}{C_a} + (1 - M) \times \frac{T_j}{T_a} = \frac{2}{C_a / C_j + T_a / T_j} \quad (10)$$

根据式(11)可以计算得到动态阈值区间,其中 Q_h 为区间上限, Q_l 为区间下限.经过实验在 m 选取 2, n 选取 1.5 时效果较好.根据式(12)可以对场景切换进行更灵活的判定,为接下来的流程规划做出先行准备。

$$\begin{cases} Q_h = \sqrt[m]{Q} \\ Q_l = Q^n \end{cases} \quad (11)$$

$$\begin{cases} \text{不存在跳变, } S_n > Q_h \\ \text{可能跳变, } Q_l < S_n \leq Q_h \\ \text{一定跳变, } S_n \leq Q_l \end{cases} \quad (12)$$

2 特征点相似度估计

2.1 ORB 基本信息

ORB (oriented FAST and rotated BRIEF) 是 oriented FAST 与 rotated BRIEF 的缩写,是目前比较完善的特征点提取与描述算法.与其他特征点提取算法相比,例如 SIFT、SURF,ORB 的最大优势在于速度.在相同的实验条件下,ORB 的处理速度要比 SURF 快一个数量级^[12]。

将 ORB 算法拆分成两部分, 分别是 FAST 快速角点检测算法和旋转 BRIEF 特征点描述子计算算法。

基本的 FAST 快速角点检测算法只需两步, 常用的 FAST 算法包括 FAST-9、FAST-11、FAST-12, 这里以 FAST-12 为例。对于某个像素点, 该算法选取以该像素点为圆心, 半径为 3 的圆上的 16 个点做参考点, 如图 1。当存在连续的 12 个点上的像素值与中心点的像素值差值的绝对值大于设定阈值, 则判定该中心点为一个特征点。为进一步加快速度, FAST-12 算法会先对中心点正上方、正右侧、正下方、正左侧的 4 个像素点, 即 1、5、9、13 这 4 个像素点进行判断, 若不存在连续 3 个点上的像素值与中心点的差值大于设定阈值, 则直接否决该点, 否则继续进行全 16 点的判断。

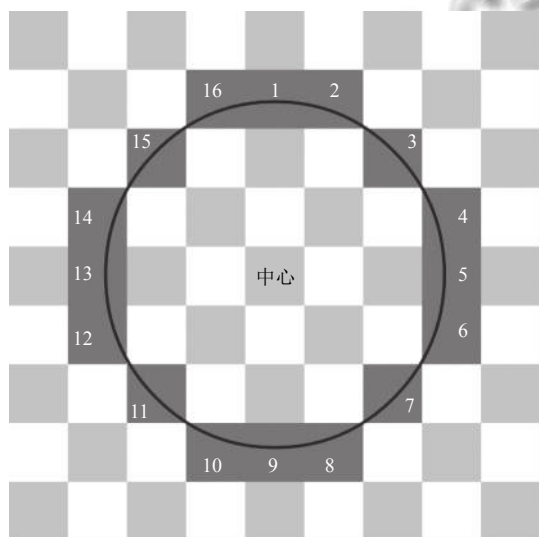


图 1 FAST 参考点

计算得到特征点后, 若存在相邻的特征点, FAST 会对相邻的特征点做非极大值抑制, 具体做法为分别计算两点的像素值与周围 16 点的像素值差和, 记为该特征点的能量值, 并保留能量值较高的特征点。

基本的 FAST 方法并不具有尺度不变性与旋转不变性, 为此 ORB 对其进行了一定的改进^[13]。针对尺度不变性, ORB 参考 SIFT 的处理方式建立了图像金字塔, 对不同尺度的图像分别进行了高斯模糊与特征点检测, 将各尺度图像的特征点集作为该幅图像的特征点集。而针对旋转不变性, ORB 通过使用灰度质心法, 在特征点周围选取一块特定大小的区域, 计算区域内的灰度质心, 并记录特征点到灰度质心的方向作为该特征点的方向。

BRIEF 是一种对特征点进行描述的二进制描述子, 基本逻辑为在特征点邻域内反复选取点对, 根据点对中两点的像素值差异, 为二进制描述子赋值。ORB 在简单 BRIEF 的基础上通过引入 oriented FAST 和统计学习的方法, 解决了旋转特征描述困难和邻域点对选取困难的问题。

2.2 特征点提取

在 ORB 提取特征点的过程中, 即使已经对相邻点做了非极大值抑制的操作, 仍然会有特征点冗余的情况出现。即在某区域内, 往往可以使用少量特征点描述该区域, 但由于角点并不完全相邻, 会在某小范围内提取出大量特征点。在拥有字幕等独立图像信息时, 该缺陷会更加明显。由于字幕等独立图像信息的边界点大部分都为特征点, 且这些点的能量一般更高, 当设定了 FAST 的提取特征点上限后, 这些对场景切换检测作用较小的图像信息可能会占据绝大多数特征点, 这会极大地降低最终得到的图像相似度的可信度。

所以在进行特征点提取时采用分组提取与四叉树非极大值抑制^[14]的方式, 通过对特征点进行均匀化处理, 从而降低独立图像信息的影响。特征点提取的步骤大致如下。

(1) 首先将图像进行均匀分块, 经实验, 当图像分为 4×4 的时候效果最好。

(2) 在每一块中分别进行特征点提取, 舍弃其中对应原图像的外边缘点的特征点, 便于后续计算特征点描述子。

(3) 判断提取的特征点个数, 当小于一定值 A 时, 适当降低 ORB 的 FAST 算法阈值, 并设定最大提取数量为 A , 重新提取特征点, 同样舍弃其中对应原图像的外边缘点的特征点。

(4) 记录各区域信息, 包括区域特征点序列与区域分裂次数, 将其加入区域队列中, 舍弃没有特征点的区域。

(5) 按照特征点数量对区域进行排序, 将特征点最多的区域从队列中取出, 若该区域分裂次数大于一定值, 或该区域特征点唯一, 直接将该区域中能量值最高的特征点记入最终的特征点列表。否则对该区域进行均匀四叉树分割处理, 记录 4 个小区域的特征点信息, 将分裂次数加一, 并将特征点数大于 0 的小区域分别加入区域队列。

(6) 重复步骤 (5), 直到队列长度与最终的特征点列表长度和达到要求或区域队列为空。

(7) 得到了最终的经过均匀化处理后的特征点列表. 特征点提取的大致流程见图 2, 流程左边为步骤

(1)–(4), 用于特征点的基本提取, 右边为步骤 (5)–(7), 用于特征点的均匀化处理.

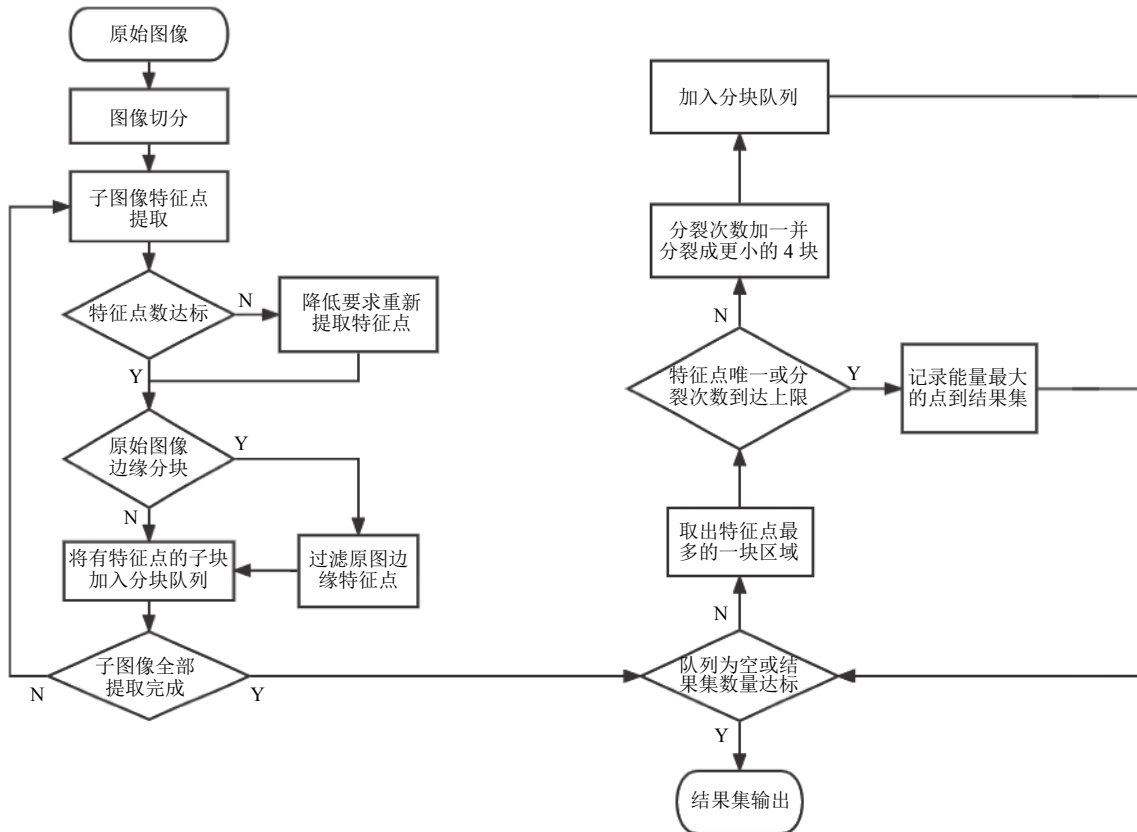


图 2 特征点提取流程图

2.3 ORB 相似度估计

通过第 2.2 节中方法得到经过均匀化的特征点集合, 记为均匀特征点; 同时将完成前 3 个步骤得到的特征点合并, 得到未经过均匀化处理的特征点列表, 记为完全特征点. 判断图像相似性的步骤大致如下.

(1) 分别对两幅图像进行如上操作, 可以得到两幅图像的均匀特征点集 A_1 、 A_2 , 完全特征点集 T_1 、 T_2 .

(2) 使用 FlannBasedMatcher 对 A_1 和 T_2 、 A_2 和 T_1 进行限制距离的特征点匹配, 得到匹配点集 F_1 、 F_2 .

(3) 对于匹配信息 F_1 , 记录匹配信息中参与匹配点数量为 M_1 , 同时根据各点的匹配数量进行不同操作. 当匹配点数量大于 0 时, 记录为成功匹配点, 当匹配点数量等于 1, 或匹配点数量大于 1 且匹配距离满足式 (13) 时, 记录为良好匹配点, 式 (13) 中 D_1 为最佳匹配的距离, D_2 为次佳匹配的距离, r 为比例因子, 经实验验证, 当 r 为 0.75 时效果最佳. 分别记录成功匹配点与良好匹配点的数量为 S_1 、 G_1 . 对匹配信息 F_2 进行同样的操

作, 得到 M_2 、 S_2 、 G_2 .

(4) 至此我们分别拥有两幅图像的参与匹配点数量 M_1 、 M_2 , 成功匹配点数量 S_1 、 S_2 , 与良好匹配点数量 G_1 、 G_2 . 根据式 (14) 可以计算得到两幅图像的相似度 S_t , 其中 a 、 b 、 c 、 d 、 e 分别为各项对相似度的影响因子, 经过实验分别设定为 0.3、0.1、0.1、0.2、0.3.

$$D_1 \leq D_2 \times r \tag{13}$$

$$\begin{cases} d_1 = 1 - \frac{|M_1 - M_2|}{\max(M_1, M_2)} \\ d_2 = 1 - \frac{|S_1 - S_2|}{\max(S_1, S_2)} \\ d_3 = 1 - \frac{|G_1 - G_2|}{\max(G_1, G_2)} \\ d_4 = \frac{S_1 + S_2}{M_1 + M_2} \\ d_5 = \frac{G_1 + G_2}{S_1 + S_2} \\ S_t = d_1^a + d_2^b + d_3^c + d_4^d + d_5^e \end{cases} \tag{14}$$

3 算法流程

按序读取视频帧序列, 根据人眼视觉暂留时长与刷新频率, 帧读取间隔为 0.2 s. 算法流程见图 3.

具体的算法流程如下.

(1) 将视频读入容器之中, 指针指向 0, 读取帧图像作为参考图像; 并将指针向后推移 0.2 s.

(2) 读取帧图像作为目标图像, 首先根据第 1 部分中的方法对参考图像与目标图像进行基本的相似度评估, 得到相似度 S_n , 并根据式 (12) 决定处理流程, 若不存在跳变则直接执行第 (4) 步; 若一定跳变则记录该帧时间与相似度到转换列表, 并直接执行第 (4) 步; 若可能跳变则正常执行第 (3) 步.

(3) 根据第 2 部分中的方法对参考图像与目标图像进行基于特征点的相似度评估得到相似度 S_t , 在同

时保留两种算法相似度的条件下, 根据式 (15) 对相似度进行重新计算, 其中 Q_o 为 ORB 方法的相似度阈值, I 为置信区间, 表示使用 ORB 算法得到的相似度的可靠区间大小, p 为非置信区间的降维参数, 使处于非置信区间的 ORB 相似度对整体相似度的影响降低. 经实验 Q_o 、 I 、 p 分别为 0.7、10、0.1 时效果良好. 使用式 (9) 对场景切换进行最终判定, 若判定存在场景切换则记录该帧时间与相似度到转换列表.

(4) 使用该帧图像覆盖参考图像, 并将指针向后推移, 返回步骤 (2), 直到读取完全部视频帧.

(5) 对最终的场景转换列表做非极大值抑制处理, 即当在同一时间段 T 内存在多个场景切换帧, 只保留相似度最低的一帧.

$$\begin{cases} S_n - Q_o + S_t, & \text{if } (Q_o - I) \leq S_t \leq (Q_o + I) \\ S_n - p \times (Q_o - S_t), & \text{other} \end{cases} \quad (15)$$

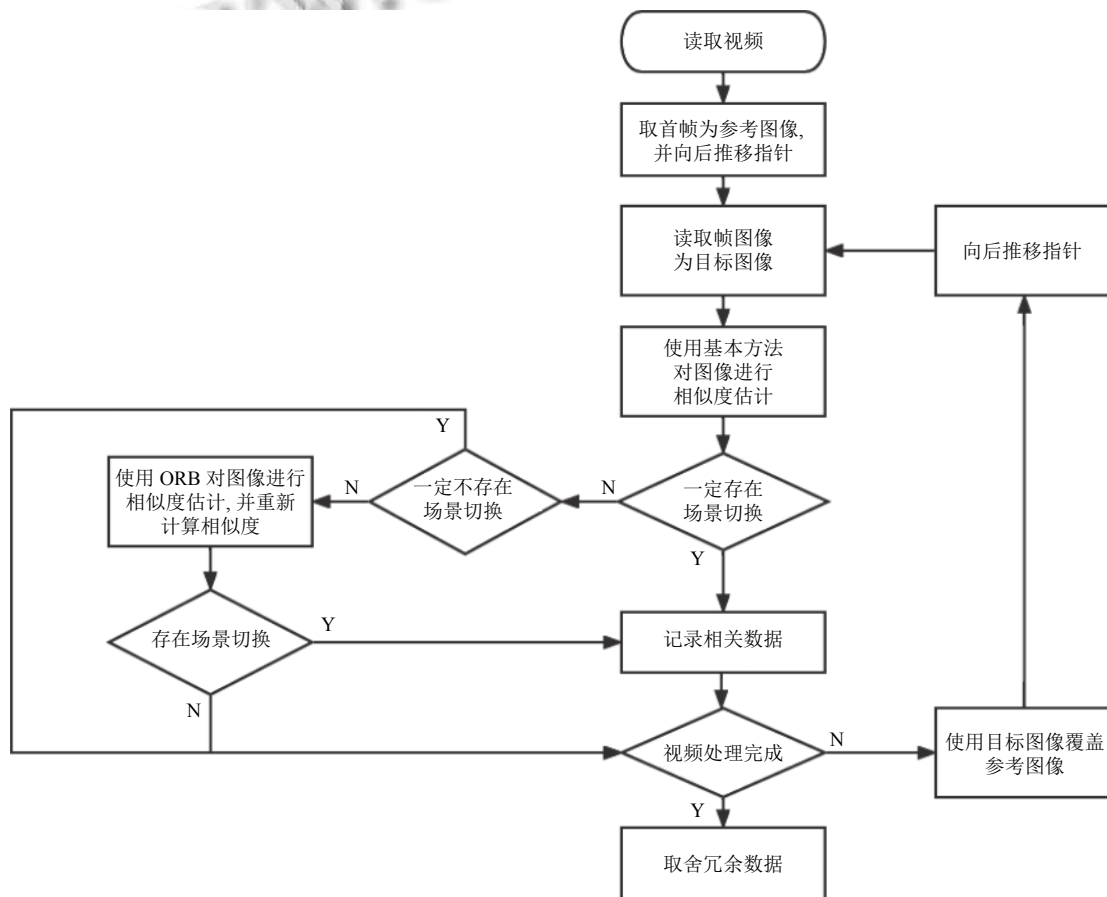


图 3 方法流程图

4 实验结果

4.1 试验条件

实验保证各方法的基本条件一致. 为保证工程的

易迁移性, 使用 Java+OpenCV 4.5.4 进行开发. 选取的视频样本大多为场景切换较为频繁的视频, 包括环境拍摄、人物讲述、游戏介绍、动漫混剪、电影速看等

多种类型,能够保证实验样本的种类丰富.对文献[9]中的算法进行参数调优,将计算自适应阈值的阈值因子设定为3.0,使其漏检与误检数据处于较均衡状态,并与本文算法进行相同的实验.

实验中,分别使用研究中各阶段的算法,即只使用颜色特征的方法、只使用纹理特征的方法、融合颜色与纹理特征的方法、只使用ORB的方法、完整方法等,同时使用文献[9]中的算法,对数据集中的视频进行场景切换检测,将场景切换时间集合作为数据结果,与人工标注的场景切换时间集合进行比对,最终输出算法处理时间,及算法的漏检、误检时间集合.

4.2 算法效率

使用各方法对同一视频进行处理,记录处理时长,数据见表1.可以看出,通过对方法流程的设计,减少了需要进行ORB相似度评估的图像数量,保证了加入ORB特征检测算法后的整体运行效率.对比文献[9]的算法,时效性有一定的提升.

表1 算法处理效率(检测帧数为1285)

方法	处理时间(s)	平均检测效率(f/s)
颜色特征判断算法	897.848	1.43
纹理特征判断算法	894.076	1.44
颜色+纹理特征判断算法	1022.537	1.26
ORB特征点判断算法	3311.962	0.39
本文算法	1674.553	0.77
文献[9]中的算法	1974.277	0.65

4.3 算法准确性

选取的视频样本中共519个切换点,1619个切换检测点,表2为各方案的实验数据,其中方案1只使用颜色直方图进行相似度判断,并设定相似度阈值为0.7;方案2融合颜色特征与纹理特征,设定融合因子为0.6、阈值为0.7;方案3为两种特征融合并使用动态融合因子,阈值定为0.7;方案4为两种特征融合并使用动态融合因子与动态阈值;方案5为两种特征融合,并使用动态融合因子和阈值,且加入ORB算法,即完整算法;方案6为文献[9]中的算法.表中的查全率与查准率计算参考式(16),其中R为查全率,P为查准率,C为正确检出数,M为漏检数,E为误检数.

$$\begin{cases} R = \frac{C}{C+M} \\ P = \frac{C}{C+E} \end{cases} \quad (16)$$

从表2中可以看到,只使用颜色特征时准确性较

低,在融合了纹理特征后准确度有了小幅度的提升,通过使用动态因子,准确性又有了一些提升,完整算法在保证查准率的前提下,查全率有极大的提升.而对比文献[9]中的算法,通过多种特征融合与动态因子的使用,算法准确性也有所提升.

表2 各方案检测结果

方案	漏检数M	误检数E	检出数C	查全率R(%)	查准率P(%)
方案1	198	45	321	61.850	87.705
方案2	184	41	335	64.547	89.096
方案3	173	40	346	66.667	89.637
方案4	170	46	349	67.245	88.354
方案5	103	54	416	80.154	88.511
方案6	113	86	406	78.227	82.520

可以看到,在加入动态阈值因子对算法进行优化后,算法的整体准确率没有提升.经过分析,原因在于选取动态阈值时并不总是能命中效果最好的阈值,但从整体来看总是能保持在较好的阈值区间.表现在各个视频上,加入动态阈值后能够在一定程度上提高检测的稳定性,避免极端情况下出现大量漏检误检.以表3中的信息为例,该表中记录了使用确定阈值与动态阈值各自的算法稳定性,稳定性通过算法的查全率方差 V_R 与查准率 V_P 方差来表示,方差的计算公式如式(17):

$$\begin{cases} \bar{R} = \frac{100}{n} \sum R \\ \bar{P} = \frac{100}{n} \sum P \\ V_R = \frac{1}{n} \sum (R \times 100 - \bar{R})^2 \\ V_P = \frac{1}{n} \sum (P \times 100 - \bar{P})^2 \end{cases} \quad (17)$$

从结果上可以看到在未使用动态阈值因子时,查全率或查准率总有一项的波动较大,这样使得很难找到一个合适的定阈值,能够同时满足高查全率与查准率.在使用了动态阈值后,可以做到在满足一定需求的情况下,兼顾查全率与查准率.

表3 加入动态阈值前后的查全率与查准率方差

阈值	查全率方差 V_R	查准率方差 V_P
0.85	38.651	129.811
0.8	117.145	94.899
0.75	277.278	70.830
0.7	633.378	93.070
0.65	657.362	92.946
0.6	419.171	223.088
动态阈值	29.657	103.378

5 结论

本文提出了一种融合多种方式的视频场景切换检测方法. 通过对检测的完整流程进行合理的分支规划, 并充分利用图像的基本特征, 规避了单一图像特征的缺陷. 同时设计了一套动态因子的计算与使用方式, 降低了各方法的缺陷对整体算法的影响, 从而提高了方法的稳定性, 并在一定程度上提高了方法的有效性, 最后结合 ORB 算法, 实现了在适当牺牲运行效率的前提下, 提升视频场景切换检测算法准确性的目标. 算法利用视频图像的基本信息, 舍弃了高级语义信息, 保证了算法迁移的便捷. 经过实验验证, 该算法具有良好的检测性能.

参考文献

- 1 黄清, 丰洪才, 刘立. 基于卷积神经网络的多模态视频场景分割优化算法. 计算机应用研究, 2022, 39(5): 1595–1600.
- 2 杜洋. 视频场景分割及暴力内容检测算法的研究与实现 [硕士学位论文]. 北京: 北京邮电大学, 2021.
- 3 Trojahn TH, Goularte R. Temporal video scene segmentation using deep-learning. Multimedia Tools and Applications, 2021, 80(12): 17487–17513. [doi: 10.1007/s11042-020-10450-2]
- 4 Abdhussain SH, Al-Haddad SAR, Saripan MI, *et al.* Fast temporal video segmentation based on Krawtchouk-Tchebichef moments. IEEE Access, 2020, 8: 72347–72359. [doi: 10.1109/ACCESS.2020.2987870]
- 5 任绍美, 都云程, 王涛, 等. 基于均值漂移的自适应视频场景分割. 2011 年全国电子信息技术与应用学术会议论文集. 上海: 上海禹捷文化传播有限公司, 2011. 605–610.
- 6 李枫, 赵岩, 王世刚, 等. 结合 SIFT 算法的视频场景突变检测. 中国光学, 2016, 9(1): 74–80.
- 7 徐迎春, 丰洪才, 刘立. 基于双重检测模型的农业气象科普视频镜头分割算法. 软件导刊, 2022, 21(9): 159–166. [doi: 10.11907/tjdk.221551]
- 8 马彦卓, 常义林, 杨海涛. 应用于视频编码的实时多测度联合突变场景切换检测算法. 光子学报, 2010, 39(6): 1058–1063.
- 9 蔡轶珩, 胡朝阳, 崔益泽. 融合颜色信息与特征点的镜头边界检测算法. 计算机应用, 2017, 37(S2): 95–98, 111.
- 10 刘林. 基于多颜色空间和统计直方图的场景分类和目标检测研究 [硕士学位论文]. 济南: 济南大学, 2011.
- 11 周丽君. 基于内容的视频镜头边界检测与关键帧提取研究 [硕士学位论文]. 天津: 河北工业大学, 2014.
- 12 索春宝, 杨东清, 刘云鹏. 多种角度比较 SIFT、SURF、BRISK、ORB、FREAK 算法. 北京测绘, 2014, (4): 23–26, 22. [doi: 10.3969/j.issn.1007-3000.2014.04.006]
- 13 赵宇琦, 连晓峰, 谭励, 等. 一种均匀抗噪自适应的 ORB 特征提取算法. 计算机仿真, 2022, 39(4): 321–327. [doi: 10.3969/j.issn.1006-9348.2022.04.063]
- 14 秦志钢. 改进的 ORB 特征提取及描述算法. 集成电路应用, 2022, 39(2): 132–133.

(校对责编: 牛欣悦)