

基于众核处理器和 GPU 的视频快速检测方案

杨 娟 曾苗祥 徐 晶 许 炜

(华中科技大学电子与信息工程系湖北省智能互联网技术重点实验室 武汉 430074)

摘 要 目前基于普通架构的视频检测速度较慢,难以满足网络视频实时监测的要求,为此提出一个基于众核处理器和图形处理单元(GPU)的视频检测方案。该方案基于众核处理器实现视频解码,基于 GPU 实现 SURF(Speed Up Robust Features)和 SVM(Support Vector Machine)的图像检测算法。与基于普通 PC 架构的视频检测方案相比,该方案的视频检测性能提升了 10 倍以上。

关键词 视频检测,众核处理器,GPU,SURF,SVM

中图法分类号 TP391.4 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.3.055

Fast Video Detection Scheme Based on Multi-core Processor and GPU

YANG Juan ZENG Miao-xiang XU Jing XU Wei

(Internet Technology and Engineering R&D Center, Department of Electronics and Information Engineering,
Huazhong University of Science and Technology, Wuhan 430074, China)

Abstract At present, the speed of video detection based on general structure is very slow, and it is difficult to meet the requirement of real-time network video monitoring. This paper showed a new video detection method based on multi-core processor and graphic processing unit (GPU). This method uses multi-core processor to realize video decoding, and uses the GPU to realize the SURF (Speed Up Robust Features) and SVM (Support Vector Machines) algorithm to detect the image. Compared with video detection scheme based on general PC architecture, the performance of the method based on multi-core processor and GPU can be improved over 10 times.

Keywords Video detect, Multi-core processor, GPU, SURF, SVM

1 引言

在现今信息化的时代,互联网给人们带来了越来越便捷舒适的生活,与此同时也给很多不法分子通过互联网迅速传播不健康、低俗的视频内容提供了便利,这对人们尤其是青少年的身心健康成长带来了极大的负面影响,所以应重视对网络视频的内容检测。同时,随着网络视频通信的发展,实时、快速地检测网络上传播的视频内容对保持良好的网络环境尤为重要。

基于普通 PC 架构的视频检测流程分两个步骤:将视频解码成图像和对图像分类检测。众所周知,视频是由一系列图片通过压缩编码得到的,因此经过解码解压缩可将视频转换为一系列连续的图片序列。目前常用的视频解码方法是 directshow 解码或 ffmpeg 解码。早期的不良图片检测普遍采用肤色点检测方法^[1],在 HSV 颜色空间上,因肤色点的像素主要集中在一个范围内,通过像素点的色度信息可判别是否为肤色点。基于肤色区域的视频内容检测能过滤正常图片,是后期精确检测不良视频的预处理过程^[2];于是 Hogen Lee 提出在获取图像后,依据两类特征判断图像是否含有不

良内容,使准确率得到了一定的改善^[3];2008 年 Chang-Yul Kim 等人在文献[4]提出视频解码后,利用分场景的思想获取图像的移动趋势,分离出背景,然后根据人体区域的肤色、纹理和形状特征来判断图像是否含有不健康的内容,它在一定程度上能检测出视频中是否含不良内容;文献[5]指出首先提取视频的关键帧,根据移动前景区间的帧间差,获取完整的前景区域,并同时得到皮肤区域,在该方法基础上分辨肤色上的不良信息能极大提高检测精度;上述视频检测方案均基于非机器学习,2012 年 Victor M. Torres Ochoa 提出了基于机器学习的视频检测方法,在获取图像后,使用支持向量机(Support Vector Machine, SVM)的方法检测图像中是否含有低俗内容,使准确率得到了很大的提升^[6]。

随着高清和标清视频在网络中的传播,实时快速地检测视频内容对算法性能的要求越来越高,基于普通 PC 架构的视频检测方案检测所需时间较视频本身长度还长(见表 1)。此时,众核处理器和 GPU 的出现给视频内容的快速检测带来了契机,原因如下:众核处理器中单个处理器的计算能力不如普通 PC 的处理器,但是当多个处理器对视频并行解码时,平均视频解码率显著提高;与 CPU 相比,GPU 精于计算,特

到稿日期:2014-04-28 返修日期:2014-09-17 本文受“十二五”国家科技支撑计划课题(2011BAK08B00)资助。

杨 娟(1990—),女,硕士生,主要研究方向为图像处理、视频检测,E-mail:csu2008yang@163.com;曾苗祥(1991—),男,硕士生,主要研究方向为图像处理、视频编解码;徐 晶(1979—),男,博士,讲师,主要研究方向为无线网络通信、网络安全;许 炜(1977—),男,博士,副教授,主要研究方向为网络内容管理。

别是其并行浮点计算,该优势已经被越来越普遍地应用到图像处理、生物信息计算等方向。基于上述两点,本文提出了基于众核处理器和 GPU 的视频检测方案,其中众核处理器用于视频解码,GPU 用于图像检测。众所周知,视频解码获取的图像数量很多且重复性很大,一定范围内的抽帧比例不会影响最后的检测结果^[7,8],本文根据两个模块解码获得并处理同一帧图像的速度,以此来设置合适的抽帧比例,以便最大程度地提升系统的速度。接下来本文将先介绍基于众核处理器和 GPU 的快速视频检测模块的框架设计,再介绍系统各个模块的实现,接着展示基于硬件的视频检测系统的测试结果,最后对测试结果进行分析总结。

表 1 一个 avi 格式视频在 PC 上处理的耗时

	系统模块检测/s		系统检测/s
	视频解码	图像检测	
480×320	23.5	280.7	283
640×480	45.7	513.8	521
1280×720	123	1497.3	1531
1920×1080	266	3127	3180

2 基于硬件的视频检测框架设计

2.1 基于普通 PC 架构的视频检测方法

基于普通 PC 架构的视频检测方法由两个模块构成:视频解码和图像检测。其中视频解码模块将视频解码成图片序列,图像检测模块用于获取图像的特征描述子,并用 SVM 对特征子分类。流程如图 1 所示。

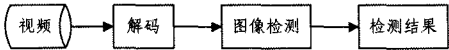


图 1 现有的视频检测方法

随着流经网络服务器的视频越来越多,视频检测的实时性越来越难得到满足。本文针对基于普通 PC 架构的视频检测方法做了一组检测速度测量,测量对象是一个 72s 的 avi 格式的视频,视频的分辨率分别为 480×320、640×480、1280×720、1920×1080。分别测量系统各模块的耗时以及系统总耗时,测量结果如表 1 所列。其中,时间单位为秒(s),视频解码和图像检测由两个不同的进程实现,两个进程之间使用命名管道传递图像数据。

分析表 1 可知,当分辨率为 480×320 时,时长约一分钟的视频的处理时间大于 1 分钟;分辨率越高,处理时间越长。即使对视频解码模块进行抽帧处理,当分辨率高于 1280×720 时,检测时间也高于视频总时长,显然这难以满足视频的实时性检测需求。

针对上述方案存在的问题,本文利用众核处理器的并行处理功能以及 GPU 的并行浮点处理优势^[9],将众核处理器和 GPU 应用到网络视频检测中,以提高视频检测系统的速度。接下来介绍基于众核处理器和 GPU 的快速视频检测系统的框架。

2.2 基于众核处理器和 GPU 的视频检测框架

基于众核处理器和 GPU 的快速视频检测系统也由两部分组成:视频解码和图像检测。在视频检测模块中,本文采用 ffmpeg 开源包对视频解码,将其放在 Tilera TM64 众核处理器上,多个处理器并行对多个视频同时解码,以此降低视频解码消耗的平均时间从而提高视频解码的速度;在图像检测模块,基于普通 PC 架构的 SUR 图像检测算法不能满足系统需

要,因此将提取图像特征的 SURF 算法移植到 GPU 中实现,通过 GPU 在并行浮点计算上的速度优势来提升图像检测的速度。

基于众核处理器和 GPU 方案的系统流程如图 2 所示。

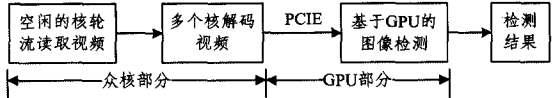


图 2 基于硬件的视频检测策略

在解码视频之前需要先将存放在硬盘上的视频通过 PCIE 传输到众核处理器上,设置一个处理器负责接收视频文件,并将视频写到某个指定的目录文件中。以便其他空闲处理器开始对视频处理,每个处理器在一段时间内只负责一个视频,通过对视频加锁的方式避免其他的处理器对该视频重复处理。当视频解码结束时,将该视频删除,接着处理目录文件中存在的其他的未被加锁的视频。

在 Tilera TM64 核处理器上,能够供开发者使用的处理器核的数目为 53,在本实验中使用 50 个处理器对多路视频同时解码。本方案通过多路视频的平稳并行解码,降低视频解码消耗的平均时间,以此提升系统的性能。在 SURF 算法中,图像的各个像素或像素区域在算法上独立并行,因此使用 GPU 可以在一定程度上提升算法的速度。

接下来描述系统的模块实现。

3 基于硬件的视频内容检测模块设计

前文已经介绍,在本方案中,使用 Tilera TM 众核处理器对视频进行解码,使用 GPU 进行图像检测。接下来对这两个模块进行详细的介绍。

3.1 基于众核处理器的视频解码模块

3.1.1 视频解码流程

本系统采用 ffmpeg 开源库解码视频,ffmpeg 是一个开源、免费、跨平台的视频和音频流方案,它提供了录制、转换以及流化音视频的完整解决方案。虽然视频编码格式不同,但其区别仅在于解码时调用的 API 不同,解码的步骤不会发生改变。基于 ffmpeg 视频解码的固定流程如图 3 所示。

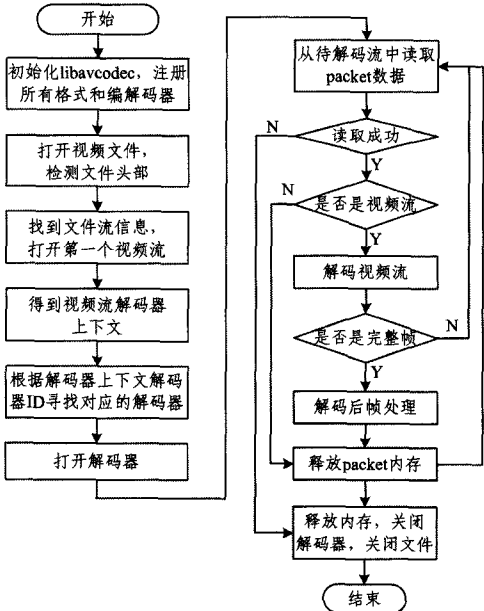


图 3 ffmpeg 解码流程

图 3 详细显示了使用 ffmpeg 从打开视频文件到得到一帧一帧的图像的过程,这里不再赘述。

3.1.2 基于众核处理器的视频解码

前文已经提到在视频解码模块中,首先设置一个处理器来接收从硬盘上传来的视频文件,当一个视频接收结束后,将该视频写在指定的目录下,该进程继续接受其他视频。接着众核处理器中的其他空闲处理器轮流读取指定目录下的视频文件,若文件已被上锁,则表示它正在被其他处理器解码,不对这个视频文件进行处理;否则将处理轮询到的未被处理的视频文件并将其上锁,防止其他处理器解码这个视频文件,在之后的一段时间内,该处理器仅对该视频解码直到视频解码结束。

在视频解码的过程中,首先读取视频的头文件信息,在视频头文件中包含有视频的时长、帧数、分辨率、文件编码格式、时间戳等重要信息;接着根据读取得到的文件格式找到相对应的解码器。前文提到一个视频将在一个处理器中解码完全,即在得到一帧图像后,会将图像数据通过 PCIe 传递到宿主主机中的图像检测模块,同时根据视频帧数据的上下文指针是否为空判断该图像是否为视频的最后一帧,若为最后一帧,表示该视频已被处理完全,然后在目录中将该视频删除,防止该视频被其他处理器重复处理;否则继续读取视频数据包还原视频帧,直到视频读取完全。根据前文描述,基于众核处理器的视频解码流程如图 4 所示。

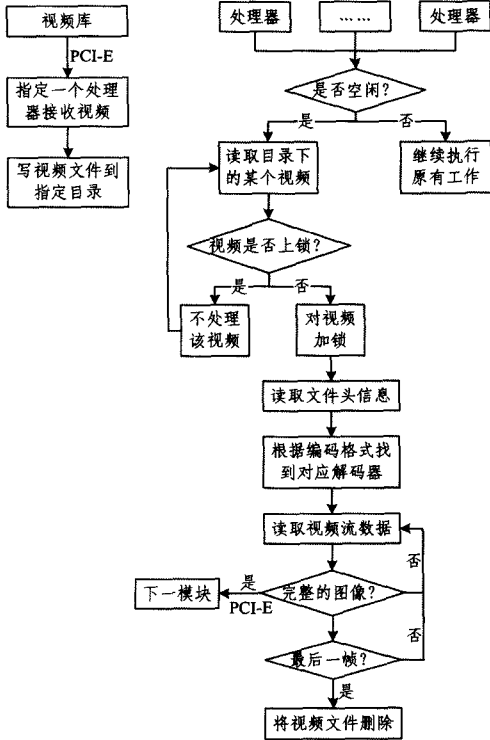


图 4 基于众核处理器的视频解码流程

经过上述的步骤,一个个视频就转化为了一系列的图像。

3.2 基于 GPU 的图像不良内容检测模块

3.2.1 图像检测原理

常用的不良图像检测算法多采用 SURF 算法和 SVM 算法。其中, SURF 算法用于提取图像的局部特征,对图像的旋转、尺度缩放、高亮变化保持不变性; SVM 算法能高效地实现从训练样本到预测样本的转导推理,在解决小样本、非线性及

高维模式识别中表现出许多特有的优势。

图像检测流程如图 5 所示。

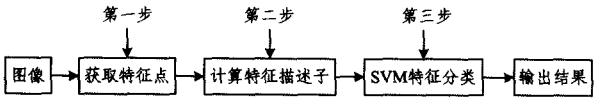


图 5 图像检测流程图

图像检测主要由上述 3 步组成,第一步获取图像特征点,第二步计算得到图像特征描述子,这两步都是 SURF 算法的内容,第三步使用 SVM 算法分类 SURF 算法计算得到的图像特征,分类结果即为图像检测的结果。表 2 显示了这 3 步在计算上的耗时情况,实验环境为 Centos5.9 64-Bit 操作系统,处理器为 i3-2120,测试的图像为 bmp 格式,表格中的结果为平均时间,时间单位为 ms。

表 2 普通方法图像检测 3 步的平均耗时

	第一步/ms	第二步/ms	第三步/ms
480×320	62.9	60	0.15
640×480	106	102	0.15
1280×720	274	251	0.15
1920×720	717	603	0.15

由表 2 可知,图像检测的耗时主要在第一步和第二步,这两步由 SURF 算法实现。为了加快检测速度,本文将 SURF 算法移植到 GPU 中实现。利用 GPU 的高速计算特性,加快图像特征描述子获取的速度,进而提升系统性能。

3.2.2 基于 GPU 的图像特征提取

在计算机系统中, CPU 主要用于算法逻辑与控制,其计算速度较慢,而 GPU 精于计算,尤其是大规模并行浮点运算。在 SURF 算法中合理利用 CPU 端和 GPU 端的资源优势提取图像的特征描述子,提升图像检测的速度。SURF 的实现技术已经十分成熟,本文不做介绍。

基于 GPU 的图像特征计算的框架如图 6 所示。

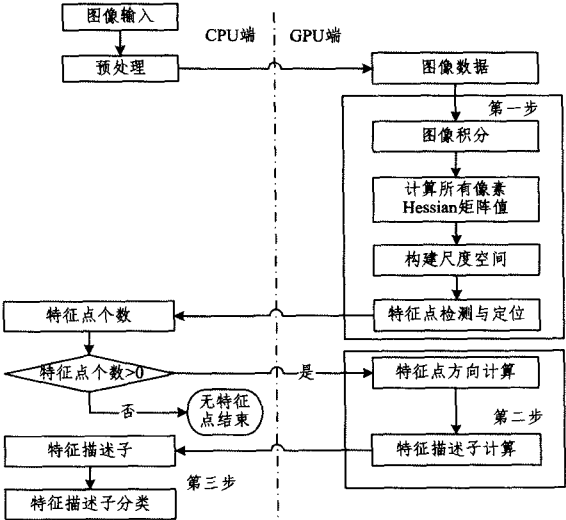


图 6 基于 GPU 的 SURF 算法实现流程

图 6 显示了基于 GPU 的 SURF 特征提取的流程,下面将介绍各个步骤的实现策略。

3.2.3 基于 GPU 的图像特征提取各个步骤的实现

1. 基于 GPU 的图像积分

计算积分图像,需要对图像中的所有像素的灰度值求和^[10,11],这些像素的灰度值是相互独立的,可以并行计算,因

此本文使用 GPU 对图像积分。

首先将图片的像素表示为子集形式,即每一个子集有 m 像素值,所有子集的并集为整张图片的像素,且子集之间不含重复像素。这样所有像素点的灰度值保存在 $sw[t][m]$ (t 为子集的个数)数组中并导入 GPU。接着在 GPU 中进行图像积分运算。

GPU 中的每个线程处理 m 个像素的灰度值。GPU 中执行的线程的个数为 t 。等待每个线程执行结束之后,再将所有线程的灰度值相加,即为图像的积分图像。

2. 基于 GPU 的 Hessian 矩阵行列式计算

尺度空间的计算返回值为某一组某一层的滤波器框的边长^[10],计算步骤简单,但在求 Hessian 矩阵行列式值中起着关键的作用,将其放在 GPU 中执行,可减少 CPU 与 GPU 数据传输的延时。

在算法中需要计算所有尺度空间中的图像每个像素点的 Hessian 矩阵行列式值,而图像中每个像素点的 Hessian 矩阵计算都是独立的,与其他像素的值无关,因此使用 GPU 计算 Hessian 矩阵行列式值能在一定程度上提升算法性能。

3. 基于 GPU 的特征点获取

为了得到图像的所有特征点,需对图像中的每个像素点进行计算,像素点是否为特征点仅与其相邻层次的 9 个临近像素点和同一层次临近的 8 个像素点有关^[10]。每个像素点是否为特征点在计算上是独立的,本文利用 GPU 大量线程的并行性计算这一步骤。

将每个像素灰度以及其相邻的 26 个像素灰度组成一个数组传递到 GPU 中,每个线程对一定个数的像素点是否为兴趣点做出判断,在线程中将该像素与临近的 26 个像素的灰度值和 Hessian 阈值相比,如果大于临近的 26 个像素灰度值和阈值,则该像素点为特征点。

4. 基于 GPU 的特征点方向计算

为了使特征描述算子具有旋转不变的性能,算法为每个特征点赋予了方向。在算法中以兴趣点为圆心,以 $6S$ (S 为兴趣点对应的尺度)为半径的圆形邻域里,用尺寸为 $4S$ 的 Haar 小波模板对图像进行处理,即得到了该邻域内每个点所对应的 x, y 方向的响应,然后以兴趣点为中心的高斯函数 ($\sigma=2s$) 对这些响应进行加权。接着使用圆心角为 60 度的扇形以特征点为中心旋转一周,计算该扇形处于每个角度时,它所包括的图像点的 Haar 小波响应和。把扇形区域环绕一周所形成的矢量都记录下来,取长度最大的矢量,其方向即为该特征点所对应的方向^[10]。

大多数图像特征点数目比较多,使用 GPU 计算特征点方向能优化运行速度。在算法中先计算得到特征点的尺度因子,再计算高斯尺度大小,接着得到特征点的矢量长度最大的方向。使用 GPU 可以将每个特征点的计算置于一个线程中执行,这样可以提升特征点的方向计算速度。

5. 基于 GPU 的特征描述子计算

为构建特征点描述子向量,先确定一个以特征点为中心的正方形邻域。把该邻域分成 4×4 个子块区域,对每个子块区域的 $dx, dy, |dx|, |dy|$ 求和 (dx 表示水平方向的 Haar 小波响应, dy 表示竖直方向的 Haar 小波响应),得到一个 4 维向量,把 4×4 个子块的向量连接起来即得到一个 64 维向量,此向量就是描述该特征点的描述子特征向量。得到特征点描述子向量后,需将其标准化,才能使用 SVM 对特征描述子进

行分类^[10]。

在这一过程中,每个特征点对应的特征描述子的计算过程是一样的,针对这一特性,可以将该算法移植到 GPU 中执行来提升速度。

4 实验方案与结果

4.1 实验方案

为了准确评估基于众核处理器和 GPU 的视频检测方案对系统性能的提升,本文进行了 4 组实验,分别从视频解码速度(独立)、图像检测速度(独立)、视频检测系统中两个模块的处理速度以及系统总体性能几个方面来验证本文提出的方案。实验均以基于普通 PC 架构的视频检测方法作为参照。根据二者对视频的检测结果和检测时间来对系统性能进行评估。

实验采用软件环境为 CentOs5.9 64 位操作系统,硬件环境为 Intel(R)Core(TM)i3-2120 型主频为 3.3GHz 的 CPU, NVIDIA Geforce GTS450 型 GPU,其显存为 1GB,显存位宽 128bits,总线接口 PCI Express 2.0 16X,有 192 个 1.242GHz 的流处理器内核,Tilera TM 64 核平台。

4.2 实验结果

实验 1 测量视频解码模块的速度,采用众核处理器和普通 PC 单处理器同时对 50 个大小相同、分辨率相同的视频进行解码。以 avi 格式为例,记录解码 50 个视频所需的总时间,单位为秒(s),检测结果如表 3 所列。

表 3 视频解码速度对比

	众核解码/s	PC 解码/s
480×320	114	1175
640×480	225	2285
1280×720	586	6155
1920×1080	1208	13300

由表 3 可知,对不同分辨率的视频文件,基于众核处理器的视频解码速度较普通 PC 单处理器均快十倍。

实验 2 对图像处理速度以及性能进行测试。实验分别采用 .bmp 和 .jpg 格式的图像库,选择 4 种不同分辨率的正常图像 1000 张和含有不良内容的图像 1000 张。分别从图像检测的速度和准确率两个方面进行评估。检测结果如表 4—表 6(表 5、表 6 显示的是每张图像检测的平均时长,时间单位为 ms)所列。

表 4 图像检测准确率对比

	GPU 检测方法 (检出数/检测数)	普通检测方法 (检出数/检测数)
正常图像	1000/1000	1000/1000
不良图像	989/1000	989/1000

由表 4 的结果可知,使用 GPU 做图像检测与普通方案检测图像得到的结果是一样的。在检测不良图像时,存在 1.1% 的漏检率,实验结果显示二者漏检的图像是相同的。在不良视频检测上,1.1% 的漏检不会对视频检测的结果造成影响。

表 5 正常图像的检测速度对比

图像分辨率	执行时间/ms	
	CPU 版本	GPU 版本
480×320	130	25
640×480	238	37
1280×720	676	88
1920×1080	1400	158

表6 不良图像的检测速度对比

图像分辨率	执行时间/ms	
	CPU 版本	GPU 版本
480×320	122	23
640×480	208	30
1280×720	507	65
1920×1080	1318	140

由表4可知,基于普通PC架构的视频检测方法和基于众核处理器和GPU的视频检测方案的准确率都为100%,二者的图像检测准确率相同,均能满足不良图像检测需求。

从表5和表6可以得出,无论是正常图片还是不良图片,基于GPU的图像检测速度远远高于基于普通CPU的,且随着视频分辨率的提高,基于GPU的图像检测速度较基于CPU的速度快。

实验3 测试视频检测系统中各模块的检测速度,前文提到视频的连续性,视频中存在大量重复的帧,可以根据视频解码和图像检测的时间差异,采取图像抽帧处理,使得两种方案的性能都达到最优。为了确定抽帧的比例,设计了如下实验,以avi格式的视频为例,分别测量分辨率为480×320、640×480、1280×720、1920×1080时的情况。视频解码得到一帧图像以及图像检测模块检测一张图像的耗时结果如表7(表中时间单位为ms)所列。

表7 avi格式视频的各个模块耗时对比

	硬件方案/ms		普通方案/ms	
	解码	检测	解码	检测
480×320	0.98	24	10.88	126
640×480	1.94	35	21.17	227
1280×720	5.68	75	56.97	595
1920×1080	12.82	147	123.21	1367

从表7的结果可以看出,在两个系统中,视频解码获取一帧图像的时间远小于检测一帧图像的时间,且随着视频图像分辨率的提高,时间差异越来越大。视频解码模块和图像检测模块由两个不同的进程实现,二者采用进程间通信的方式传递数据,可实现并行操作,在不影响图像检测准确性的前提下,对视频解码得到的图像采用一定的抽取策略,以匹配两者的速度,实现系统性能的最大化。

实验4 测量在两个检测系统的性能都达到最优时的性能。以avi格式的视频为例,视频时间为50个×72s,每个视频帧数为2159帧,分别测量分辨率为480×320、640×480、1280×720、1920×1080的视频在抽帧率为1/15、1/30、1/60时的情况,检测结果如图7所示。

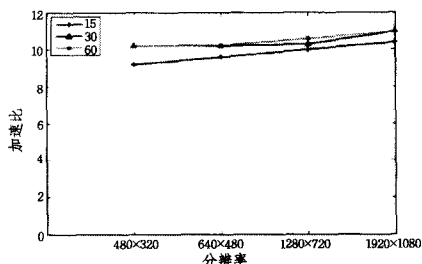


图7 avi格式不同抽帧时加速比曲线

如图7所示,在抽帧率为1/15时,性能提升随着分辨率的增加而增加;在抽帧率分别为1/15、1/30、1/60时,基于硬件的系统相对于普通检测系统的性能提升倍数随着抽帧率的提升逐渐增加,抽帧率为1/30和1/60时性能提升差别很小。

为了性能提升的同时保证视频检测的准确率,本文采用的是1/30的抽帧策略。

根据该抽帧方案得到的基于硬件的检测系统与普通方案的检测结果如表8所列(表中时间单位为s)。

表8 两种方案总体性能对比

	硬件系统/秒	普通系统/秒
480×320	140	1466
640×480	275	2855
1280×720	769	8114
1920×1080	1725	19250

由表8显示的检测结果可以发现,在测试avi格式的视频时,当基于硬件的不良视频检测系统的性能和基于PC的视频检测系统的性能都达到最优时,两者的性能差异高达10倍。

结束语 本文设计了一个基于众核处理器和GPU的快速视频检测方案,该方案能快速、准确地检测出网络中的不良视频。较基于普通PC架构的传统视频检测系统,该系统性能提升了十倍以上,在满足准确率的同时也满足了对实时性的需求。多组实验的结果验证了方案的有效性。在以后的工作中我们将进一步地研究算法在性能上提升的可能性,以获得更快的网络图像不良内容的实时检测系统,满足日益快速传输网络中的视频检测需要。

参考文献

- [1] Zhao G, Wang S, Wang T, et al. HSV color space and face detection based objectionable image detecting[C]//2008 Second International Conference on Future Generation Communication and Networking Symposia. 2008,3:107-110
- [2] Yu J J, Han S W. Skin detection for adult image identification [C]//2014 16th International Conference on Advanced Communication Technology (ICACT). IEEE, 2014:645-648
- [3] Lee H, Lee S, Nam T. Implementation of high performance objectionable video classification system[C]//The 8th International Conference Advanced Communication Technology, 2006 (ICACT 2006). IEEE, 2006,2:4-962
- [4] Kim C Y, Kwon O J, Kim W G, et al. Automatic system for filtering obscene video[C]//10th International Conference on Advanced Communication Technology, 2008(ICACT 2008). IEEE, 2008,2:1435-1438
- [5] Yu W, Qu Z, Jin Y. A Pornographic Video Detection Method Based on Semi-supervised Learning on Graphs[C]//2013 Sixth International Symposium on Computational Intelligence and Design (ISCID). IEEE, 2013,2:347-350
- [6] Ochoa V M T, Yayilgan S Y, Cheikh F A. Adult video content detection using Machine Learning Technology[C]//2012 Eighth International Conference on Signal Image Technology and Internet Based System(SITIS). 2012:967-974
- [7] Esmaeili M M, Fatourechi M, Ward R K. A robust and fast video copy detection system using content-based fingerprinting[J]. IEEE Transactions on Information Forensics and Security, 2011,6(1):213-226
- [8] Endeshaw T, Garcia J, Jakobsson A. Classification of indecent videos by low complexity repetitive motion detection[C]//37th IEEE Applied Imagery Pattern Recognition Workshop, 2008 (AIPR'08). IEEE, 2008:1-7

(下转第295页)

CMF_LRC 算法针对眼镜遮挡识别率最高,达到 100% 的识别率,比之文献[14]的方法有更进一步地提高;而且本文方法因为误差图像的求取过程十分简单,因此其计算复杂度比文献[14]的方法也低。OMF_RSRC 算法针对围巾遮挡情况,识别率达到 99.50%,相比于同类算法,识别率提升较为显著。尽管当测试图像存在光照影响时,本文采用的水平集分割方法得到的分割准确性受到影响,但是本文算法仍表现出较好的识别性能提升表现。但是,这时的识别性能改善效果还不够显著。究其原因可能在于,在存在单侧光照影响时人脸的左右像素相差较大,从而使得背景区域内像素相差较大、区域连通性欠佳,最终导致图像遮挡区域和非遮挡区域不能有效分割。针对既有遮挡又有光照影响的人脸识别,探求一种更具鲁棒性的遮挡区域检测方法,是我们下一步继续努力的方向。

表 7 与其它同类算法的识别性能比较

识别方法	无光照影响人脸库		有光照影响人脸库	
	眼镜遮挡	围巾遮挡	眼镜遮挡	围巾遮挡
CMF_LRC	100.00	95.00	65.67	73.18
OMF_RSRC	97.00	99.5	84.67	84.83
OMF_RCRC	96.50	97.00	86.00	91.17
LRC_LRC ^[14]	99.00	95.00	84.50	76.67
RPCA_RAT ^[15]	93.00	15.5	72.70	13.70
RPCA_WE ^[15]	94.00	40.00	73.30	58.00

结束语 针对实际人脸识别中的待识别图像可能存在眼镜、围巾等连续遮挡这一问题,提出了一种改进的基于异值区域检测消除的连续遮挡人脸识别方法。首先采用整体样本平均脸或是类平均脸与测试图像进行差值运算得到误差人脸图像,然后利用选择性局部或全局分割方法对误差人脸图像进行分割得到人脸图像存在的遮挡区域,最后利用遮挡区域消除后的样本进行识别,其识别性能能够得到有效提高。基于 AR 和 Yale B 人脸库的实验仿真结果证明了本文方法的有效性。

参 考 文 献

[1] Zhao W,Chellappa R,Rosenfeld A, et al. Face Recognition; A Literature Survey[J]. ACM Computing Surveys, 2003, 35(4): 399-458

[2] Hua G,Yang M H, et al. Introduction to the special section on real-world face recognition[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011, 33(10): 1921-1924

[3] 彭义刚,索津莉,戴琼海,等. 从压缩传感到低秩矩阵恢复:理论与应用[J]. 自动化学报, 2013, 39(7): 981-994

[4] Wright J, Yang A Y, Ganesh A, et al. Robust face recognition via sparse representation[J]. IEEE Transactions on Pattern A-

nalysis and Machine Intelligence, 2009, 31(2): 210-227

[5] Meng Y,Zhang L. Gabor feature based sparse representation for face recognition with gabor occlusion dictionary[C]//Proceeding of Computer Vision on European Conference on Computer Vision. 2010; 448-461

[6] Zhang L,Meng Y,Feng X C. Sparse representation or collaborative representation; Which helps face recognition? [C]// Proceedings of IEEE International Conference on Computer Vision. 2011; 471-478

[7] Zhang L, Yang M, Feng X, et al. Collaborative representation based classification for face recognition[OL]. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.363.8790>

[8] Naseem I, Togneri R, Bennamoun M. Linear regression for face recognition[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010, 32(11): 2106-2112

[9] 胡正平,李静. 基于低秩子空间恢复的联合稀疏表示人脸识别算法[J]. 电子学报, 2013, 41(5): 987-991

[10] 杜杏菁,郭明雄. 人脸识别中遮挡区域恢复算法研究[J]. 计算机科学, 2013, 40(5): 307-310

[11] Zhou Z,Wagner A,Mobahi H, et al. Face recognition with contiguous occlusion using markov random fields[C]//Proceedings of the 12th International Conference on Computer Vision. 2009: 1050-1057

[12] Morelli Andrés A,Padovani S,Tepper M, et al. Face recognition on partially occluded images using compressed sensing[J]. Pattern Recognition Letters, 2014, 36: 235-242

[13] Meng Y,Zhi Zhao-Feng,Shiu S C K, et al. Fast and robust face recognition via coding residual map learning based adaptive masking[J]. Pattern Recognition, 2014, 47(2): 535-543

[14] Luan X, Fang B, Liu L, et al. Face recognition with contiguous occlusion using linear regression and level set method[J]. Neurocomputing, 2013, 12(2): 386-397

[15] Luan X, Fang B, Liu L H, et al. Extracting sparse error of robust PCA for face recognition in the presence of varying illumination and occlusion[J]. Pattern Recognition, 2014, 47(2): 495-508

[16] Zhang K,Zhang L,Song H, et al. Active contours with selective local or global segmentation; A new formulation and level set Method[J]. Image and Vision Computing, 2010, 28(4): 668-676

[17] Lee K C, Ho J, Kriegman D. Acquiring linear subspaces for face recognition under variable lighting[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005, 27: 684-698

[18] Aleix M, Robert B. The AR face database[R]. CVC Technical Report 24. 1998

(上接第 270 页)

[9] Wu J,Wang C F. Fast computation of cylindrical Green's functions with graphic processing unit[C]//Antennas and Propagation Society International Symposium (APSURSI). 2013; 1884-1885

[10] Mirollo A C,Guerrero J J,Sagues C. SURF features for efficient robot localization with omnidirectional image[C]// 2007 IEEE

International Conference on Robotics and Automation. 2007: 3901-3907

[11] Szczuko P. Influence of image transformations and quality degradations on SURF detector efficiency[C]//Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA), 2013. IEEE, 2013; 285-290