

基于 OpenCV 的人脸识别应用^①

陶颖军

(河海大学 能源与电气学院, 南京 211100)

摘要: 在 Linux 平台下开发一人脸识别系统, 通过 QT 来开发用户界面, 调用 OpenCV 图像处理库对相机进行采集和处理采集图像, 从而实现了人脸检测、身份识别、简单表情识别的功能。

关键词: OpenCV; 人脸识别; QT; Linux

Face Recognition Based on OpenCV

TAO Ying-Jun

(College of Energy and Electrical Engineering, Hehai University, Nanjing 211100, China)

Abstract: By using QT to develop the user interface and by calling the OpenCV image processing library to process the captured images, we develop a face recognition system in linux platform With face detection, identification and simple expression recognition function.

Key words: OpenCV; face recognition; QT; Linux

人脸识别的研究可以追溯到上个世纪六、七十年代, 经过几十年的曲折发展已日趋成熟, 构建人脸识别系统需要用到一系列相关技术, 包括人脸图像采集、人脸定位、人脸识别预处理、身份确认以及身份查找等。而人脸识别在基于内容的检索、数字视频处理、视频检测等方面有着重要的应用价值, 可广泛应用于各类监控场合, 因此具有广泛的应用前景。OpenCV 是 Intel 公司支持的开源计算机视觉库。它轻量级而且高效——由一系列 C 函数和少量 C++ 类构成, 实现了图像处理和计算机视觉方面的很多通用算法, 作为一个基本的计算机视觉、图像处理和模式识别的开源项目, OpenCV 可以直接应用于很多领域, 其中就包括很多可以应用于人脸识别的算法实现, 是作为二次开发的理想工具。

1 系统组成

本文的人脸识别系统在 Linux 操作系统下利用 QT 库来开发图形界面, 以 OpenCV 图像处理库为基础, 利用库中提供的相关功能函数进行各种处理:通过相

机对图像数据进行采集, 人脸检测主要是调用已训练好的 Haar 分类器来对采集的图像进行模式匹配, 检测结果利用 PCA 算法可进行人脸图像训练与身份识别, 而人脸表情识别则利用了 Camshift 跟踪算法和 Lucas-Kanade 光流算法。

2 搭建开发环境

采用德国 Basler acA640-100gc 相机, PC 机上的操作系统是 Fedora 10, 并安装编译器 GCC4.3, QT 4.5 和 OpenCV2.2 软件工具包, 为了处理视频, 编译 OpenCV 前需编译 FFmpeg, 而 FFmpeg 还依赖于 Xvid 库和 X264 库。

3 应用系统开发

程序主要流程如图 1 所示。

3.1 图像采集

图像采集模块可以通过 cvCaptureFromAVI() 从本地保存的图像文件或 cvCaptureFromCam() 从相机得到图像, 利用 cvSetCaptureProperty() 可以对返回的

^① 基金项目:安徽省教育厅自然科学基金(2005KJ004ZD)

收稿时间:2011-07-04;收到修改稿时间:2011-09-07

结构进行设置^[1]:

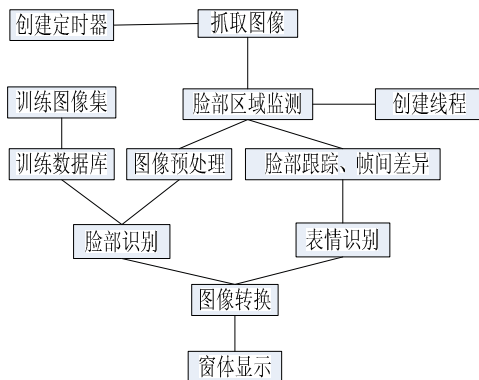


图 1 程序流程(visio)

```
IplImage *frame;CvCapture* camera = 0;
camera = cvCaptureFromCAM( 0 );
cvSetCaptureProperty(camera,
CV_CAP_PROP_FRAME_WIDTH, 320 );
cvSetCaptureProperty(camera,
CV_CAP_PROP_FRAME_HEIGHT, 240 );
```

然后利用 start() 函数开启 QTimer 定时器,每隔一段时间发送信号调用自定义的槽函数,该槽函数用 cvGrabFrame() 从视频流中抓取一帧图像放入缓存,再利用 CvRetrieveFrame() 从内部缓存中将帧图像读出用于接下来的处理与显示。在 qt 中显示之前,需将 IplImage* source 转换为 QPixmap 类型。

```
uchar *qImageBuffer = NULL;
/*根据图像大小分配缓冲区*/
qImageBuffer = (uchar*) malloc(source->width *
source->height * 4 * sizeof(uchar));
/*将缓冲区指针拷贝到存取 Qimage 的指针中*/
uchar *QImagePtr = qImageBuffer;
/* 获取源图像内存指针 */Const uchar*
IplImagePtr=reinterpret_cast<uchar*>(source->imageData);
```

```
/*通过循环将源图像数据拷贝入缓冲区内*/
for (int y = 0; y < source->height; ++y){
for (int x = 0; x < source->width; ++x){
QImagePtr[0] = IplImagePtr[0];
QImagePtr[1] = IplImagePtr[1];
QImagePtr[2] = IplImagePtr[2];
QImagePtr[3] = 0;
```

```
QImagePtr += 4;
```

```
IplImagePtr += 3; }
```

```
IplImagePtr+=source->widthStep-3*source->width; }
```

```
/*将 Qimage 转换为 QPixmap*/QPixmap local =
QPixmap::fromImage(QImage(qImageBuffer,source->width,source->height, QImage::Format_RGB32));
```

```
/*释放缓冲区*/
```

```
free(qImageBuffer);
```

最后利用 QLabel 的 setPixmap() 函数进行显示^[2]。

3.2 图像预处理

由于大部分的脸部检测算法对光照,脸部大小,位置表情等非常敏感,当检测到脸部后需利用 cvCvtColor() 转化为灰度图像,利用 cvEqualizeHist() 进行直方图归一化处理^[3]。

3.3 脸部检测方法

OpenCV 采用一种叫做 Haar cascade classifier 的人脸检测器,他利用保存在 XML 文件中的数据来确定每一个局部搜索图像的位置,先用 cvLoad() 从文件中加载 CvHaarClassifierCascade 变量,然后利用 cvHaarDetectObjects() 来进行检测,函数使用针对某目标物体训练的级联分类器在图像中找到包含目标物体的矩形区域,并且将这些区域作为一序列的矩形框返回,最终检测结果保存在 cvRect 变量中^[4]。

3.4 脸部识别方法

识别步骤及所需函数如图 2 所示。

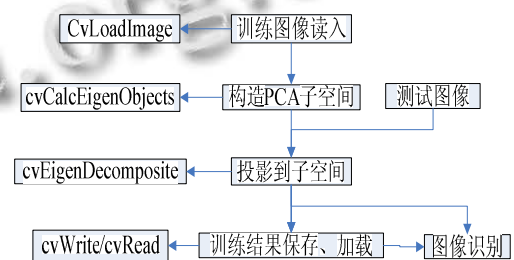


图 2 识别步骤(visio)

PCA 方法(即特征脸方法)是 M.Turk 和 A.Pentland 在文献中提出的,该方法的基本思想是将图像向量经过 K-L 变换后由高维向量转换为低维向量,并形成低维线性向量空间,即特征子空间,然后将人脸投影到该低维空间,用所得到的投影系数作为识别的特征向量。识别人脸时,只需将待识别样本的投影系数与数据库中目标样本集的投影系数进行比对,以确定与哪

一类最近^[5]。

PCA 算法分为两步：核心脸数据库生成阶段，即训练阶段以及识别阶段。

3.4.1 训练阶段

主要需要经过如下的几步：

(1) 需要一个训练人脸照片集。

(2) 在训练人脸照片集上计算特征脸，即计算特征值，保存最大特征值所对应的的 M 张图片。这 M 张图片定义了“特征脸空间”（原空间的一个子空间）。当有新的人脸添加进来时，这个特征脸可以进行更新和重新计算得到。

(3) 在“特征脸空间”上，将要识别的各个个体图片投影到各个轴（特征脸）上，计算得到一个 M 维的权重向量。简单而言，就是计算得到各个个体所对应于 M 维权重空间的坐标值。

OpenCV 实现为：先用 `cvLoadImage()` 载入图片并利用 `cvCvtColor()` 转换为灰度图片，建立自定义的迭代标准 `CvTermCriteria`，调用 `cvCalcEigenObjects()` 进行 PCA 操作，计算出的 Eigenface 都存放在向量组成的数组中，利用 `cvEigenDecomposite()` 将每一个训练图片投影在 PCA 子空间(eigenspace)上，结果保存在矩阵数组中，用 `cvWrite<datatype>()` 将训练结果保存至 XML 文件中。下面图 3 为训练得到的部分特征脸图像。

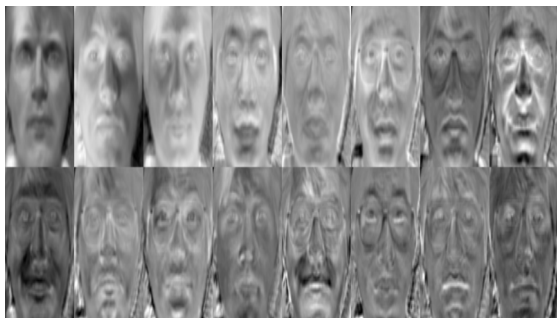


图 3 特征脸图像

3.4.2 身份识别阶段

在识别新的人脸图片时，具体的操作方法流程如下：

(1) 基于前面得到的 M 个特征脸，将新采集的图片投影到各个特征脸，计算得到一个权重集合（权重向量）。

(2) 判断新图片是否是一幅人脸图像，即通过判断图像是否足够靠近人脸空间。

(3) 如果是人脸图像，则根据前面计算的权重集合（权重向量），利用权重模式将这个人脸分类划归到初始时计算得到的各个个体或者是成为一个新的个体照片。简单而言，就是计算新权重到原来各个个体权重的距离，选择最近的，认为是识别成这个个体；如果最近的距离超出阈值，则认为是一个新的个体。

(4) 更新特征脸或者是权重模式。

(5) 如果一个未知的人脸，出现了很多次，也就意味着，对这个人脸没有记录，那么计算它的特征权重（向量），然后将其添加到已知人脸中^[6]。

OpenCV 实现调用 `cvRead<datatype>()` 加载训练结果 XML 文件，调 `cvEigenDecomposite()` 将采集图片映射至 PCA 子空间，利用最近距离匹配方法 `Squared Euclidean Distance`，计算要识别图片同每一个训练结果的距离，找出距离最近的即可。

3.5 脸部表情识别

脸部运动跟踪利用了 Camshift 算法，该算法利用目标的颜色直方图模型将图像转换为颜色概率分布图，初始化一个搜索窗的大小和位置，并根据上一帧得到的结果自适应调整搜索窗口的位置和大小，从而定位出当前图像中目标的中心位置。Camshift 能有效解决目标变形和遮挡的问题，对系统资源要求不高，时间复杂度低，在简单背景下能够取得良好的跟踪效果^[7]。

Camshift 的 OpenCV 实现分以下几步：

(1) 调用 `cvCvtColor()` 将色彩空间转化到 HSI 空间，调用 `cvSplit()` 获得其中的 H 分量。

(2) 调用 `cvCreateHist()` 计算 H 分量的直方图，即 1D 直方图。

(3) 调用 `cvCalcBackProject()` 计算 Back Projection。

(4) 调用 `cvCamShift()` 输出新的 Search Window 的位置和面积。

我们利用光流算法评估了两帧图像之间的变化，Lucas-Kanade 光流算法是一种两帧差分的光流估计算法。它计算两帧在时间 t 到 $t+\delta t$ 之间每个每个像素点位置的移动。是基于图像信号的泰勒级数，就是对于空间和时间坐标使用偏导数^[8]。

首先要用到 shi-Tomasi 算法，该算法主要用于提取特征点，即图中哪些是我们感兴趣需要跟踪的点，对应函数为 `cvGoodFeaturesToTrack()`，可以自定义第一帧特征点的数目，函数将输出所找到特征值。接下来是

cvCalcOpticalFlowPyrLK 函数, 实现了金字塔中 Lucas-Kanade 光流计算的稀疏迭代版本。它根据给出的前一帧特征点坐标计算当前视频帧上的特征点坐标。输入参数包括跟踪图像的前一帧和当前帧, 以及上面函数输出的前一帧图像特征值, 自定义的迭代标准, 输出所找到的当前帧的特征值点^[9]。这些点可以确定面部局部区域的特征 如眼部, 鼻子高度与宽度, 嘴部两侧与底部的夹角等等, 利用与前一帧的特征比较, 可得出反应脸部动态变化的参数, 这些数据可以与脸部的一些简单表情相关联。下面图 4 为跟踪眼睛上下眨动的图像。

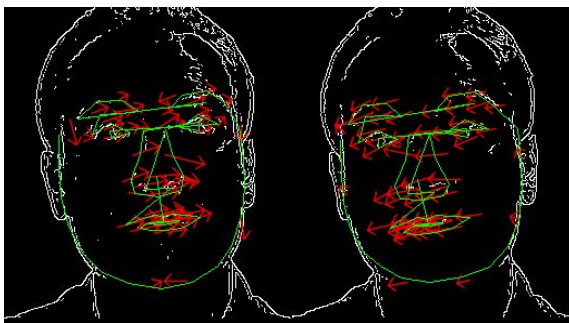


图 4 跟踪眼部上下眨动图像

4 总结

本文以 OpenCV 图像处理库为核心, 以 QT 库所提供的界面框架为基础, 设计实现了一人脸识别系统,

(上接第 170 页)

缩小了 28.36%, 同样得知中心矩也缩小了 22.43%; 而图 1 中数据表明, 混合粒子群算法可以使粒子更加快速有效地找到优化解, 能有效避免算法在迭代后期存在全局搜索能力不足缺陷, 也就是说使用混合粒子群算法的效果要优于粒子群算法。

5 结语

本文针对粒子群优化算法的早熟收敛和搜索精度低问题, 将混沌及和声搜索算法思想引入粒子群算法中, 提出了一种基于 Tent 映射的混沌和声搜索的混合粒子群算法, 利用混沌特性提高种群的多样性和粒子搜索的遍历性, 同时采用和声策略对解空间进行开发, 采用云模型的自适应策略来调整惯性权重。仿真实验表明, 改进后的混合粒子群优化算法能有效避免粒子

具有较好的实用性, 可移植性。但仍有许多不足之处, 如身份与表情识别部分可以通过引入神经网络或支持向量机 SVM 进行分类, 可以使识别准确率与识别种类数得到提高, 这些也是后续工作中需要改进的。

参考文献

- 1 Bradski G, Kaehler A. Learning OpenCV: Computer Vision with the OpenCV. O'Reilly Media, 2008.
- 2 蔡志明, 等. 精通 Qt4 编程. 北京: 电子工业出版社, 2008.
- 3 沈庭芝, 王卫江, 闫雪梅. 数字图像处理及模式识别. 北京: 北京理工大学出版社, 2007.
- 4 Wilson PI, Fernandez J. Facial feature detection using haar classifiers. Journal of Computing Sciences in Colleges, 2006, 21(4):127-133.
- 5 刘青山, 卢汉清, 马颂德. 综述人脸识别中的子空间方法. 自动化学报, 2003, 29(6):900-911.
- 6 Turk M, Pentland A. Face recognition using eigenfaces. Journal of cognitive neuroscience 1991, 3(1):71-86.
- 7 余九华, 王敬东, 李鹏. 基于 Camshift 的人脸跟踪算法. 计算机技术与发展, 2008, 18(9):12-15.
- 8 张会军. 基于图像的动目标检测技术. 微计算机信息, 2007, 23(22).
- 9 聂伟乐, 瞿建荣. 基于 OpenCV 的运动目标光流算法仿真. 应用光学, 2008, 29(6):867-869.

群算法的早熟收敛问题, 能在确保全局收敛性的基础上, 快速搜索到高质量的优化解。

参考文献

- 1 张学良, 温淑花. 基于 Tent 映射的混沌粒子群优化算法及其应用. 中国机械工程, 2008, 19(17):2108-2112.
- 2 蒋春明, 阮米庆. 汽车机械式变速器多目标可靠性优化设计. 汽车工程, 2007, 29(12):1090-1093.
- 3 Kennedy J. Bare bones particle swarms. Proc. of the IEEE Swarm Intelligence Symposium (SIS 2003), Indianapolis, Indiana, 2003, 80-87.
- 4 李俊青, 王玉亭. 混合离散和声搜索算法求解旅行商问题. 微电子学与计算机, 2009, 26(3):17-21.