

计算旋转 Harr 型特征的积分图像算法改进

邵平^{1,2}, 杨路明¹, 曾耀荣²

(1. 中南大学 信息科学与工程学院, 湖南 长沙 410083;

2. 玉林师范学院 物理与信息科学系, 广西 玉林 537000)

摘要:通过改进计算旋转 Harr 型特征的积分图像算法,降低了其计算复杂度,实现了遍历原图像一次即可获得图像直立矩形窗口和 45° 旋转矩形窗口的积分图像,实验证明算法的改进是有效的。

关键词:Harr 型特征;积分图像;算法改进;人脸检测

中图分类号:TP391

文献标识码:A

文章编号:1673-629X(2006)11-0146-02

An Improved Algorithm of Integral Image for Computing Rotated Harr - Like Features

SHAO Ping^{1,2}, YANG Lu-ming¹, ZENG Yao-rong²

(1. Department of Information Science and Engineering, Central South University, Changsha 410083, China;

2. Department of Physics and Information Science, Yulin Normal College, Yulin 537000, China)

Abstract:An improved algorithm of integral image for computing rotated Harr - like features is proposed. Not only the calculating complexity is reduced, but also the integral images of upright windows and 45° rotated windows can be gained by searching through the origin image once. Experimental results demonstrate the efficiency of this improved algorithm.

Key words:Harr - like features; integral image; improved algorithm; face detection

0 引言

在 Crow 提出 summed-area tables 算法^[1]后, Viola 将其用于人脸图像分析,引入积分图像(Integral Image)的概念作为图像的一种中间表示,并在 Adaboost 和 Cascade 算法实现的实时人脸检测系统中用它来计算图像的 Harr 型特征^[2]。后来, Rainer Lienhart 在 Viola 的基础上又提出了一些新的旋转 Harr 型特征^[3,4]及其积分图像算法,使 boosting 算法学习得到的系统能够用于旋转人脸的快速检测,但 Rainer Lienhart 用来计算旋转 Harr 型特征的积分图像需要对原图像遍历两次才可得到^[4]。

在人脸检测中,基于 boosting 的统计学习算法训练时需要大量的样本,如文献[5]共使用了人脸样本 44000 个,而每一个样本的 Harr 型特征的计算都需先计算样本的积分图像。另外,对测试集进行测试时也同样需要先计算积分图像。文中通过改进 45° 旋转矩形窗口的积分图

像算法,实现了遍历原图像一次即可同时获得图像直立矩形窗口和 45° 旋转矩形窗口的积分图像,并减少了计算量。为描述方便,下文均设原始图像宽度为 W ,高度为 H , x 为图像水平方向坐标(向右为正方向), y 为图像垂直方向坐标(向下为正方向),即 $0 \leq x \leq W-1$, $0 \leq y \leq H-1$ 。

1 Harr 型特征及积分图像

1.1 Harr 型特征介绍

Harr 型特征是 Viola 等人提出的用于人脸图像分析的一种简单矩形特征,因类似于 Harr 小波而得名,如图 1(a)~(d)所示。Harr 型特征的定义是黑色矩形和白色矩形各自在图像子窗口中对应区域的灰度值总和之差。它反映了图像局部的灰度变化,为检测旋转人脸, Rainer Lienhart 在 Viola 的基础上提出了一些新的旋转 Harr 型特征^[3,4],图 1(c), (d)为其中的一部分,所有这些 Harr 型特征的计算都是通过积分图像实现的,因 Harr 型特征中的矩形有直立矩形和 45° 旋转的矩形两种,因此积分图像也有相应的两种。

1.2 计算 Harr 型特征的积分图像

1.2.1 直立矩形窗口的积分图像

直立矩形窗口的积分图像(以下简称积分图像)中任意一点 (x, y) 表示在原始图像中(如图 2(a) 所示)的斜线

收稿日期:2006-01-28

基金项目:广西教育厅面上项目(桂教科研[2004]20号);玉林师范学院重点科研资助项目(05DWX16);玉林师范学院科研资助项目(05YBW23)

作者简介:邵平(1970-),男,湖南浏阳人,博士研究生,研究方向为图像处理、模式识别;杨路明,教授,博士生导师,研究方向为数据库、图像处理、模式识别。

区域像素灰度值之和, 即:

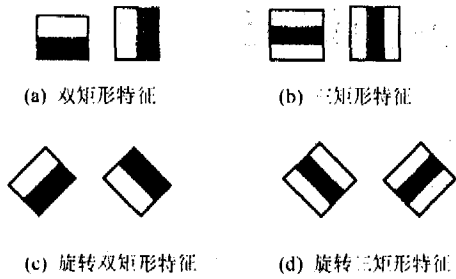


图 1 Harr 型特征示例

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \quad (1)$$

式(1)中, $ii(x, y)$ 表示积分图像(Integral Image), $i(x', y')$ 表示原图像, $ii(x, y)$ 可用式(2)、式(3)迭代进行计算^[2]:

$$s(x, y) = s(x, y-1) + i(x, y) \quad (2)$$

$$ii(x, y) = ii(x-1, y) + s(x, y) \quad (3)$$

其中 $s(x, y)$ 表示一列的积分, 且 $s(x, -1) = 0, ii(-1, y) = 0$ 。求一幅积分图像, 只需遍历一次原图像。图 2(b) 所示图像矩形窗口 D 的像素灰度值之和可以用积分图像中相应的 1, 2, 3, 4 共 4 个点计算出来, 即窗口 D 的像素灰度值之和为: $4 + 1 - (2 + 3)$ 。

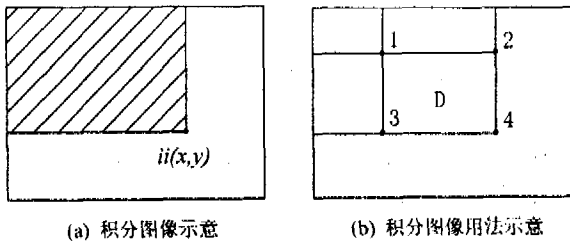


图 2 积分图像及其用法示意

1.2.2 45° 旋转矩形窗口的积分图像

45° 旋转矩形窗口的积分图像(以下简称旋转积分图像)中任意一点 (x, y) 表示原始图像中(如图 3(a) 所示)的斜线区域像素灰度值之和, 即:

$$rii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \quad (4)$$

式(4)中, $rii(x, y)$ 表示旋转积分图像(Rotated Integral Image), $i(x', y')$ 表示原图像。求旋转积分图像 $rii(x, y)$, 文献[4]需遍历原图像两次才能完成, 每次迭代需要 5 次加法和 2 次减法。图 3(b) 所示图像 45° 旋转的矩形窗口 D 的像素灰度值之和可以用旋转积分图像中相应的 1, 2, 3, 4 共 4 个点计算出来, 即窗口 D 的像素灰度值之和为: $4 + 1 - (2 + 3)$ 。

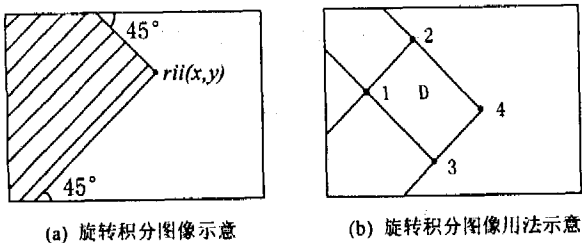


图 3 旋转积分图像及其用法示意

2 45° 旋转矩形窗口的积分图像算法改进

为了改进上述 Rainer Lienhart 的算法, 使旋转积分图像在遍历原图像一次即能完成, 并减少其计算复杂度, 对其算法进行了改进。改进的旋转积分图像中任意一点 (x, y) 表示原始图像中(如图 4(a) 所示)的斜线区域像素灰度值之和, 即:

$$irii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \quad (5)$$

式(5)中, $irii(x, y)$ 表示改进后的旋转积分图像(Improved Rotated Integral Image), $i(x', y')$ 表示原图像, $irii(x, y)$ 算法如下:

$$* y = 0 \text{ 时, } irii(x, y) = i(x, y) \quad (6)$$

$$* y > 0 \text{ 时, } irii(x, y) \text{ 可用式(7)迭代进行计算:}$$

$$irii(x, y) = irii(x-1, y-1) + irii(x+1, y-1) - irii(x, y-2) + i(x, y) + i(x, y-1) \quad (7)$$

$$\text{且 } irii(-1, -1) = 0, irii(x, -1) = 0, irii(-1, y) = 0, irii(W, -1) = 0, irii(W, y) = 0, irii(x, -2) = 0$$

改进的算法仅需从上到下遍历原图像各行一次即可获得旋转积分图像, 与文献[4]相比, 不需对原图像遍历二次, 且运算量由原来每次迭代需要 5 次加法和 2 次减法, 降低为每次迭代只需 3 次加法和 1 次减法。另外, 改进后的算法, 可使积分图像和旋转积分图像在对原图像的同一次遍历过程中获得, 因为这两种积分图像均是从左到右, 从上到下的方式扫描原图像得到的。图 4(b) 所示图像 45° 旋转的矩形窗口 D 像素灰度值之和可以用改进的旋转积分图像中相应的 1, 2, 3, 4 共 4 个点计算出来, 即窗口 D 的像素灰度值之和为: $4 + 1 - (2 + 3)$ 。

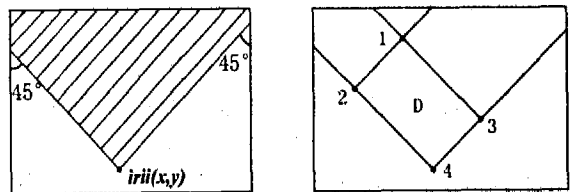


图 4 改进的旋转积分图像及其用法示意

3 实验结果

表 1 为文中的算法和 Rainer Lienhart 的算法计算速度对比, 用文中的算法计算旋转积分图像要快很多, 这是因为 Rainer Lienhart 的算法要对原图像扫描两次, 而文中的算法只需对原图像扫描一次, 且计算复杂度较小。

表 1 计算旋转积分图像的时间对比

算 法	计算旋转积分图像的时间(s)	
	Rainer Lienhart 的算法	文中的算法
CMU 集的一幅图像 (1920 × 1536 像素)	0.93	0.34
2000 幅人脸样本图像 (每幅 24 × 24 像素)	1.2	0.46

(下转第 181 页)

消费电子设备而开发的综合应用平台,它是基于 Qt/Embedded 图形界面库。Qt/Embedded 包括了窗口操作系统、游戏和多媒体、输入法、工作辅助应用程序等特性。

同 Qt/Embedded 交叉编译一样,首先需要先修改 Qt/Embedded 的配置文件,将 GCC、G++ 编译器和链接器设置为前文编译安装的交叉编译工具链。

接着是设置 Qt/Embedded 环境变量,因为 Qt/Embedded 是基于 Qt/Embedded 库的,因此需要交叉编译的动态链接库的支持,需要同时设置 Qt 的环境变量。

```
export QTDIR=/linuette/host/Qt/embedded/qt-2.3.6
export LD_LIBRARY_PATH=/usr/lib:/lib:$QTDIR/lib:
$LD_LIBRARY_PATH
export QPEDIR=/linuette/host/Qt/qpe/qpe-1.6.2
export PATH=/opt/host/armv4l/bin:$PATH
export TMAKEPATH=/usr/lib/tmake/lib/qws/linux-linuette-
g++
```

最后配置 Qt/Embedded 将 Qt/Embedded 配置为动态 SO 库形式:

```
./configure -xplatform linux-arm-g++-shared
编译 make
```

此时会出现/bin/uic: Command not found 的错误,这是因为没有指定 Qt/Embedded 的 uic 工具,uic 的工具是 Qt 专门用来将 ui(ui 文件是 Qt 图形界面文件,支持所见即所得)文件生成 .h 和 .cpp 文件的。这里可以直接使用 Qt/X11 的 uic 工具,方法如下:在/qt-2.3.6/bin 目录下建立到 Red Hat 9.0 自带的 Qt X11 的 uic 工具的链接。可以使用 Red Hat 9.0 下 Qt designer(界面与 Delphi 相类似)的应用程序开发嵌入式系统下的所见即所得的图形界面应用程序。

编译通过后会 Qt/Embedded 的路径/qt-2.3.6/lib 下生成 libqpe1.6.2 的动态链接库,同样将这些库文件拷贝到目标板文件系统中(/s3c2410pro/root/usr/qt/lib)。在目标板文件系统目录/s3c2410pro/root/usr 下新建 qpe 文件夹,将/qpe-1.6.2/apps,/qpe-1.6.2/pics,/qpe-1.

6.2/docs,/qpe-1.6.2/sounds 复制到该文件夹下。

最后修改目标板 Linux 的 /profile 文件,设置 Qt/Embedded 的环境变量:

```
export PATH=/usr/qpe/bin:$PATH
export QWS_SIZE=640*480
export PATH=/usr/qpe/bin:$PATH
```

开发板启动后就会运行 Qt/Embedded 图形界面了。

4 总 结

目前越来越多嵌入式系统要求图形显示界面,特别是在一些工业控制领域,嵌入式 Linux 系统的应用也逐渐增多。Qt/Embedded 延续了 Qt 在桌面系统的所有功能,丰富的 API 接口和基于组件的编程模型使得嵌入式 Linux 系统中的应用程序开发更加便捷。由于 Qt/Embedded 本身面向高端的手持设备和移动设备,将成为未来嵌入式系统的主流 GUI。

参考文献:

- [1] Qt Reference Documentation (Open Source Edition). Trolltech [EB/OL]. 2006. <http://doc.trolltech.com/4.1/index.html>.
- [2] 徐广毅,张晓林,崔迎伟,等. 嵌入式 Linux 系统中 GUI 系统的研究与移植[J]. 单片机与嵌入式系统应用,2004(12): 14-17.
- [3] SAMSUNG ELECTRONICS. USER'S MANUAL S3C2410 X32-Bit RISC Microprocessor[EB/OL]. 2001-05. <http://www.Samsung.com/Products/Semiconductor/SystemLSI/MobileSolutions/MobileASSP/MobileComputing/S3C2410/S3C2410.htm>.
- [4] Rubini A. LINUX 设备驱动程序[M]. 北京:中国电力出版社,2004.
- [5] 孙天泽,袁文菊,张海峰. 嵌入式设计及 Linux 驱动开发指南——基于 ARM9 处理器[M]. 北京:电子工业出版社,2005.

(上接第 147 页)

4 结 论

文中提出了计算旋转 Harr 型特征积分图像的改进算法。通过算法的改进,降低了其计算复杂度,并实现了遍历原图像一次即可同时获得计算图像直立矩形窗口和 45° 旋转矩形窗口的积分图像。实验结果证明算法改进是有效的,可用于提高对 boosting 统计学习算法中大量样本及测试集的积分图像计算速度。

参考文献:

- [1] Crow F. Summed-area tables for texture mapping[J]. SIG-GRAPH, 1984,18(3):207-212.
- [2] Viola P, Jones M. Rapid object detection using a boosted cas-

cade of simple features[C]//In Proc. of the IEEE Computer Vision and Pattern Recognition. Cambridge, Britain: [s. n.], 2001:511-518.

- [3] Lienhart R, Kuranov A, Pisarevsky V. Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection[C]//In Proc. 25th German Pattern Recognition Symposium. Magdeburg, Germany: [s. n.], 2003:297-304.
- [4] Lienhart R, Maydt J. An Extended Set of Haar-like Features for Rapid Object Detection[C]//In Proc. of the IEEE Conf. on Image Processing. New York, USA: [s. n.], 2002: 155-162.
- [5] 武勃,黄畅,艾海舟,等. 基于连续 Adaboost 算法的多视角人脸检测[J]. 计算机研究与发展, 2005,42(9):1612-1621.