

基于 OPENCV 的图像采集及激光线形提取程序的开发方式

孙观宏, 刘江, 李忠涛, 李积云

(北京科技大学 机械工程学院, 北京 100083)

摘要:介绍了一种基于 VC 开发平台和 Intel 开源图像处理类库 OPENCV 的线激光源视觉测量系统图像的采集以及激光线形提取程序的开发方式。该方式相对于传统的采集和处理方法具有实时性好、开发成本低、开发周期短和效率高等特点。最后通过实验给予验证。

关键词: OPENCV; 逆向工程; 图像采集; 图像处理

中图分类号: TP 391 **文献标识码:** A **文章编号:** 1671-5276(2008) 02-0053-03

A Method of Developing Program of Image Acquisition and Laser Alignment Extraction Based on OPENCV

SUN Guan-hong, LU Jiang, LI Zhong-tao, LI Ji-yun

(School of Mechanical Engineering, University of Science and Technology Beijing, Beijing 100083, China)

Abstract This paper presents a new method of developing program of image acquisition and extracting laser alignment based on the Intel open-source computer vision library-OPENCV in linear laser measurement system. The new method enjoys several good factors like real-time well, low developing cost, short development cycle and high efficiency in comparison to other methods.

Key words OPENCV; reverse engineering; image acquisition; image processing

0 引言

随着科学技术的发展,逆向工程已经广泛地应用于现代先进的制造系统中,利用现有产品,通过三维轮廓测量技术,得到所需要的 CAD 模型。它一般通过三个步骤实现: 1) 产品表面数字化; 2) 特征提取; 3) 生成产品的 CAD 模型^[1]。产品表面数字化不仅是实现逆向工程系统的第一步,同样也是最重要的环节。所以一直以来,如何快速准确地获取产品表面的原始数据,即点云数据就成了研究的重点。其测量方式可以分为接触式和非接触式两种。光切法是近 10 年发展起来的一种非接触测量方法,其基本原理是用线光源产生的光平面照射被测物体,由于被测物体表面凹凸不平,在其表面形成受表面形状影响产生的光带,通过摄像头 CCD 可获得光带的图像数据,并在计算机中对得到的原始图像进行图像处理,通过诸如极值法、重心法、曲面拟合等方法对光带线形进行提取,并将一定间隔的二维截面轮廓组合起来,即得该目标物体表面的三维轮廓信息^[2]。如何实时进行图像采集以及如何快速准确地对光带线形进行提取,成为表面数字化中两个最值得关注的问题。所以选择一套高效率、低成本的三维重构系统的软件开发平台是解决问题的关键。本文首先介绍几种常见的程序开发平台,而后说明基于 OPENCV 库的开发环境的优势所在。

1 几种图像处理开发平台

1.1 直接利用 VC++ 进行开发

VISUAL C++ 语言是由美国 Microsoft 公司推出的一套可视化编程语言。它是目前综合性最高、功能最全面的开发工具之一,为用户提供了基于 MFC (Microsoft Foundation Class) 的程序开发。比如常用于表示图像的设备相关位图 CBitmap 和设备无关位图 CDib 都是 MFC 中的类。在图像处理程序的开发过程中,只要将待处理的图像作为一个 CDib 类的实例,并将其指针传递给处理函数,就可以完成所需的处理。但是 MFC 中并没有提供一些例如二值化、平滑与滤波、阈值分割、灰度变换等常用的图像处理方法,都必须由用户手动开发,因此大大延长了程序开发的周期^[3-4]。

1.2 基于 VC++ 与 Matlab 混合编程

Matlab 是 Mathworks 公司推出的一套工程计算及数值分析软件。由于其功能强大、易使用,因而得到广泛应用,特别是在图像信号处理、语音信号处理、信号分析等领域,充分展示了其强大的矩阵运算功能^[5],尤其提供了非常多的实用图像处理函数,使处理过程变得简单易懂。VC++ 与 Matlab 混合编程是一种常见的图像处理程序的开发平台。但 Matlab 强大的功能不能脱离它的环境,在 Matlab 中使用行解释方式执行代码限制了代码执行速度,

而且也增加了开发成本。也就是说, 图像程序在运行时不能脱离 Matlab 环境, 虽然 Matlab Add-in 提供了一种直接集成 M 文件到 VC++ 的混合编程的方法, 但是这种方式也是通过 Mcc 预先将 M 文件编译为 C/C++ 程序的。所以 Mcc 方式存在的缺点 (如不支持所有 Matlab 函数语句尤其很多用于图像处理的函数) 它基本上都有。

1.3 基于 VC 和 OPENCV 图像库的混合编程

计算机视觉类库 OPENCV (intel open source computer vision library) 由 Intel 公司开发, 该函数库是开放源代码的, 能够从 Intel 公司的网站免费下载得到。可以说 OPENCV 相对于其他开发方式具有其显著的不可比拟的优点, 主要体现在如下几个方面^[6]:

- a) OPENCV 是一个包含独立的图像处理函数的应用程序接口, 它不依赖于外部库, 既可以独立运行, 也可在运行时使用其他外部库;
- b) 高性能: OPENCV 中所有的算法都是基于封装于 IPL 的具有很高灵活性的动态数据结构, 而且其中有一半以上的函数在设计及汇编时被 Intel 公司针对其所生产的处理器优化, 大大提高了图像处理的速度;
- c) 平台兼容性: 在多个操作系统下如 windows, linux 下都可以进行开发;
- d) 开放性: 由于 OPENCV 的源代码完全开放, 所以程序开发者可以仔细地阅读关键算法的源代码, 来理解图像处理中很多算法的原理及整个实现过程, 从而使程序开发过程变得快速高效。

特别指出的是, 它强大的图像处理应用程序接口大大减少了程序的开发量。例如, 只要在程序中调用函数 `CvThreshold(const CvArr* src, CvArr* dst, double threshold, double max_value, int threshold_type)`, 就可以轻松完成对图像的二值化或阈值分割的处理^[7]。其中参数 `threshold_type` 表示函数的作用类型。当它等于 `CV_THRESH_BINARY`, 就是对图像进行二值化处理, 结果如图 1 所示。

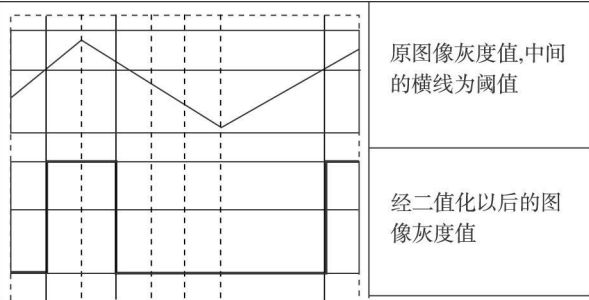


图 1 图像二值化处理

2 图像采集程序的开发

2.1 通过图像采集卡提供的 SDK 开发

图像采集卡应用了 Microview 公司的 V130 卡, 板卡附

带的二次开发包 (sdk) 中提供了用于图像视频采集的函数。其中 `MV_CaptureSingle` 是用于普通捕获操作的函数。采集例程如下:

```
MV_MAGINFO Info
MV_ScDeviceParameter( CardIndex SET_GARBMAGINFO,
(DWORD)(&Info)); PVOID pBuf1 = malloc( Info.Length);
PVOID pBuf2 = MV_CaptureSingle( CardIndex FALSE, pBuf1, Info.Length & Info).
```

这样就在 `pBuf2` 中得到了数字图像数据。但是这种捕获方法的实现依赖于视频捕获卡与摄像头的类型, 不利于灵活应用。而且, 如果想继续对得到的图像数据利用 OPENCV 中提供的函数进行处理, 就必须将图像数据转化成 OPENCV 中定义的图像数据结构 `IplImage` 类型, 这样在开发中就带来诸多不便。

2.2 通过 OPENCV 中 Highgui 提供的基于 VFW 接口的方式采集

VFW 是 Microsoft 公司推出的关于数字视频的一个软件开发包。围绕 AVI 文件标准, VFW 包含了一整套完整的视频采集、压缩、解压、回放和编辑的应用程序。编程接口利用 VFW 技术可以提高视频捕获的灵活性, 减少了对物理设备的依赖。而 OPENCV Highgui 中的视频捕获函数正是根据这一接口开发的。更关键的是, 通过 Highgui 中的采集函数捕获的视频图像数据地址可以直接传给 `IplImage` 类型的指针, 这样就简化了后续用 OPENCV 进行激光线形提取的工作。通过 Highgui 进行采集的程序如下:

```
CvCapture* capture = 0; // 定义一个视频捕获的指针
capture = cvCaptureFromCAM( 0); // 获得第一个摄像头的视频流
if( ! capture)
{
    MessageBox( "不能打开摄像头", MB_OK);
    return;
}
IplImage* frame = 0; // 定义图像指针
frame = cvQueryFrame( capture); // 捕获一帧图像
```

当然可以将采集过程放入循环过程中, 然后调用 `CvShowImage("window name", frame)`; 对采集到的视频流进行显示。实验证实, 这种采集方式不仅具有很好的硬件灵活性, 而且实时性也很好, 采集速率可以达到 20 帧 /s 以上。

3 激光线形提取方法

3.1 提取流程

经过上一步的采集, 可以得到采集到的图像数据的指针 `IplImage*`, 接下来就是根据实际情况的需要, 利用 OPENCV 提供的强大的图像处理函数对所采集的图像进行相应的处理。处理工作是在采集的图像中提取所需的激光线形 (图 2), 从而实现对被测模型表面数据的获取。图像处理的流程如图 3 所示。

图 3 中灰度变换用到的函数为 `CvCvtColor(const`



图 2 扫描激光线形

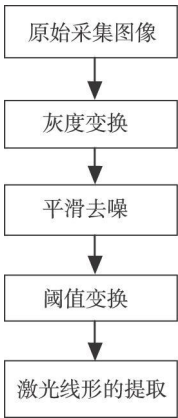


图 3 图像处理流程

`CvArr* src CvArr* dst int code);` 去除噪声用到的函数 `cvSmooth(const CvArr* src CvArr* dst int smooth_type= CV_GAUSSIAN, 3, 3)` 为高斯 3×3 滤波; 阈值变换函数则是上述的 `CvThreshold` 只是将 `threshold_type` 改为 `CV_THRESH_TOZERO`。这样, 当变换前图像中大于阈值的点的像素灰度值不变, 而小于阈值的点的灰度变为零。

而最后线形的提取需要对图像像素进行访问, 在 `IplImage` 图像结构中, 用来储存数据的成员变量是 `imageData` 所以对于八位灰度图像来说, 坐标为 x, y 点的灰度值 $I(x, y)$ 为 `((uchar*) (img->imageData+ img->widthStep* y))[x]`。这样就可以编写算法对线形进行提取了。

3 2 提取算法

a) 极值法: 极值法是将光强极大值作为光带中心。这种方法比较简单, 运算量小, 特别适合对精度要求不高但对实时性要求高的系统。极值法的算法思想是遍历图像的每一行, 找出灰度值最大的点将其保留, 将其他点的灰度值清零。

b) 重心法: 重心法是基于物理学中的刚体在 x 轴的质心演化而来的。设 x 轴上 n 个离散质点的坐标分别为 x_i , 质量为 m_i 在图像中就是像素灰度值 p_i , 则 n 个质点的质心为

$$x_c = \frac{\sum_{i=1}^n x_i p_i}{\sum_{i=1}^n p_i} \quad (1)$$

重心法具有运算速度快、精度高等特点, 提取精度可以达到亚像素级。

3 3 试验结果

根据重心法速度快、精度高等特点对线形进行提取, 结果如图 4 所示。因为提取的精度可以精确到亚像素级, 所以以相邻像素点表示, 为便于观察反色显示。

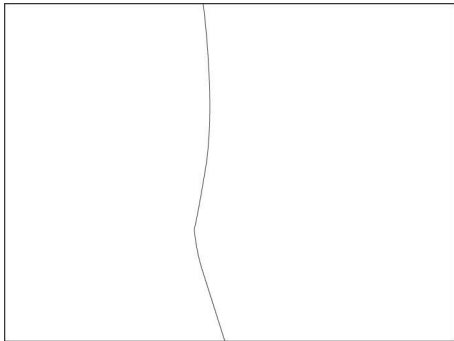


图 4 重心法提取结果

4 结论

Intel 公司开发的 OPENCV 为用 Visual C++ 开发视觉测量系统提供了极大的方便, 具有广阔的应用前景。它相对其他开发方式具有成本低、速度快、周期短等诸多优点。目前, OPENCV 在国外应用较广, 主要用于图像处理, 在国内使用还不广泛, 但是应用它开发图像处理程序必将成为一种趋势, 被越来越多的人认可。

参考文献:

[1] B dalk A, lhashdan A, SaeidMotavalli *et al* Automatic segmentation of digitized data for reverse engineering applications [J]. IE Transactions, 2000 (32): 59-69.

[2] JYan WANG Tong-qing, LIU Xiao-hui Image acquisition and processing of 3D profile measurement using light-sectioning method [J]. Journal of Computer-Aided Design & Computer Graphics, 2003, 15 (1): 43-47.

[3] 杨枝灵, 王开. Visual C++ 数字图像获取处理及实践应用 [M]. 北京: 人民邮电出版社, 2003.

[4] 侯俊杰. 深入浅出 MFC [M]. 2 版. 武汉: 华中科技大学出版社, 2001.

[5] 苏金明. MATLAB 与外部程序接口 [M]. 北京: 电子工业出版社, 2004.

[6] 刘瑞祯, 于仕琪. OPENCV 教程-基础篇 [M]. 北京: 航空航天大学出版社, 2006.

[7] Intel open source computer vision library reference manual [Z]. 2001, 12.

收稿日期: 2007-10-26