

增强图像处理算法在在线加工圆形零件轮廓识别中的应用研究

边培莹, 徐平

(西安文理学院 陕西省表面工程与再制造实验室, 西安 710065)

摘要: 针对加工中心圆形零件在线检测的生产需求, 及目前存在的轮廓识别算法冗余、误差漂移的问题, 利用 Opencv 计算机视觉库, 采用理论与实验结合方法, 选取圆形多孔零件作为研究对象, 通过 CCD 图像采集搭建图像处理软件编码调试的手段, 开发了一套实用的图像处理软硬件系统。经过对零件图像的高斯滤波, 轮廓增强, 霍夫圆检测, 尺寸计算等算法处理, 最终的实验表明零件识别曲线尺寸精度达到 ± 0.01 mm, 验证了图像处理技术在多轮廓、曲线形零件轮廓识别中的可靠性与准确性; 软件开发集成环境采用 Visual Studio 2013, 程序设计语言采用 C++ 语言。

关键词: 圆形零件; 图像处理; 高斯滤波; 尺寸识别; 软件仿真

Applied Research about Enhanced Image Processing Algorithm in Contour Identification of Circular Online Processing Parts

Bian Peiying, Xu Ping

(Provincial Key Laboratory for Surface Engineering and Remanufacturing of Shaanxi Province, Xi'an University, Xi'an 710065, China)

Abstract: For the production requirements of the on-line inspection of the round parts in the processing center, and the problems of the redundancy and error drift at the existing contour identification algorithm, this research using the Opencv library computer vision, which is based on theoretical and experimental methods, selection of circular parts with hole as the research object, through CCD image acquisition structures, image processing software code debugging method, have developed a set of practical image processing software and hardware of the system. It is verified the reliability and accuracy through parts of the image Gaussian filtering, contour enhancement, Hoff circle detection, size calculation algorithm processing, such as the final curve parts size accuracy to ± 0.01 mm in image processing technology at a contour recognition of a multi-contour and curvy part. Simultaneously, the software development of integration environment use Visual Studio 2013, and the programming language is in C++ language.

Keywords: round parts; image processing; gaussian filtering; size calculation algorithm; simulation software

0 引言

传统的零件测量方法多种多样, 包括卡尺测量、量规测量、轮廓仪测量、万能工具显微镜、坐标测量机等等, 这些传统的测量方法在过去的工业生产中扮演着重要的角色, 但随着现代化工业水平的提高, 对零件生产加工的精度要求也越来越高, 传统的测量方法已经不能够满足某些特定场景下的测量要求, 例如卡尺、量规等操作简单, 但精度上达不到要求; 万能工具显微镜、坐标测量机等精度较高, 但受到适用环境的限制且操作过程繁琐; 此外不可避免的人工因素例如疲劳程度, 专业技能的掌握程度等这些共同影响了零件测量的精度。同时传统的测量方法在某些工业领域不能满足实时检测、实时控制的要求, 随着工业化进程的不断推进, 提高零件检测技术的精度和效率是至关重要的一步。

图像处理源于 20 世纪 80 年代, 随着计算机技术的迅速发展, 图像处理这一学科开始形成。早期的图像处理主要用于改善图像质量以达到人眼的视觉要求, 输入低质量的图像, 输出符合人眼视觉效果的高质量图像。到了 21 世纪, 图像处理技术也已经被应用到了工业检测领域。如 2009 年史博等^[1]在西安交通大学 863 计划项目提出一种分层图像识别算法, 对有景深的零件进行在线监测。2011 年黄品松等^[2]研究了加工零件图像监测数据处理与高精度检测算法。2016 年水恒华等^[4]通过提取零件直线参数获得了零件的像素空间尺寸。同时文献^[5-8]中有较多的硕博士展开进一步的高效算法及仿真软件开发。但目前的研究针对于多轮廓、曲线形等图像处理算法及高效算法仿真的应用尚有差距。

本文针对数控加工中心多工序加工的需求, 研究在线加工零件的多个圆形轮廓识别与尺寸算法。一方面, 对采集的图像进行预处理; 第二方面, 对处理后的图像进行图像边缘增强; 第三方面, 查找圆并计算出各圆定形尺寸。同时将算法设计并实现了一个与用户交互的 MFC 应用程序。

1 图像采集与滤波处理

1.1 图像采集

图像采集通常采用 CCD 摄像头捕获图片, 且有闪光灯同

收稿日期: 2017-06-05; 修回日期: 2017-09-05。

基金项目: 西安市科技计划项目资助 (2016CXWL28; CXY1531WL24)。

作者简介: 通信作者: 边培莹 (1980-), 女, 副教授。主要从事机械 CAD/CAM、先进制造与增材制造技术方向的研究。

时开启提高图像质量与对比度。其基本处理流程如图 1 所示。

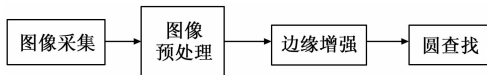


图 1 图像处理流程

当用户在客户端操作拍照时, 系统就将这一时刻的实时画面图像保存到本地的指定文件夹中, 后续处理的原始图像就是该图像。载入该图像用到的是 Opencv 开源库中的 cvLoadImage 函数, Opencv 可以处理的常见图像文件格式有以下几种: JPEG, JPG, JPE, PNG, BMP, DIB, PBM, PPM, PGM, RAS 等, 论文所拍摄的图像格式均为 JPG 格式, 载入的原图与灰度图对比如图 2 所示。

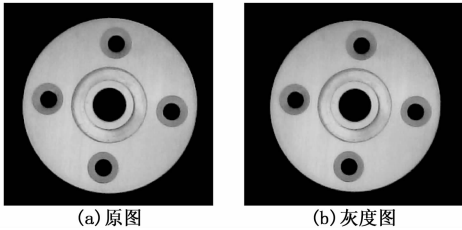


图 2 被测零件采集图片

1.2 图像滤波

对图像进行预处理是图像处理的必要环节, 目的是去除图像中的干扰因素, 主要包括颜色、空间变换和去噪处理。常用的图像噪声主要有高斯噪声以及脉冲噪声, 去除图像中的噪声, 即滤波处理。滤波处理是一种简单并且实用的预处理方式, 常用于减少图像中的噪点, 是一种线性平滑滤波。

1) 高斯滤波。

高斯滤波的原理就是对整个图像进行加权平均, 根据每个点的像素值, 由该点以及该点的邻域内的其他点的像素值通过加权平均得到。具体操作是: 用一个模板对图像中的每个像素点进行扫描, 用模板所确定的邻域内的所有像素点的加权平均灰度值去代替模板中心点的像素值。其直观效果就好比隔着一块毛玻璃观察图像。从数学角度来说, 该过程就是用图像与正态分布 (高斯分布) 作卷积, 故称这种滤波方式为高斯滤波。由于正态分布函数的傅里叶变换也是正态分布函数, 所以高斯滤波对于图像来说是一种低通滤波。

高斯滤波用于抑制服从正态分布的噪声 (高斯噪声) 的效果是非常好的, 对于图像处理而言, 一般用二维 0 均值离散正态分布函数作为滤波器。二维正态分布函数为式 (1)。

$$G_0(x, y) = Ae^{-\frac{(x-\mu_x)^2}{2\sigma_x^2} - \frac{(y-\mu_y)^2}{2\sigma_y^2}} \quad (1)$$

基于上述滤波器函数, 在 Opencv 中, GaussianBlur 函数就是用高斯滤波器来对图片进行滤波操作。函数声明如下:

```
void GaussianBlur( InputArray src, OutputArray dst, Size ksize,
double sigmaX, double sigmaY=0,
int borderType=BORDER_DEFAULT);
```

参数说明:

① InputArray 类型 src 参数, 表示待处理的源图像, 通常传入 Mat 类对象。

② OutputArray 类型 dst 参数, 表示生成的最终目标图像, 要跟待处理的源图像有相同的尺寸和类型, 为了保证这一要

求, 一般可以采用 Mat::Clone 函数, 以待处理的源图像为模板初始化目标图像。

③ Size 类型 ksize 参数, 包含 ksize.width 和 ksize.height, 分别表示宽度和高度, 用来描述所需要用到的高斯核的大小, 宽度和高度可以不一样, 但是必须是正奇数或者 0。

④ 两个 double 类型 sigmaX 和 sigmaY 参数。分别表示高斯核函数在 x 方向和 y 方向的标准偏差。

⑤ int 类型 borderType 参数, 带有默认值 BORDER_DEFAULT, 通常不需要用到。

2) 均值滤波。

均值滤波是一种线性滤波方式, 采用了邻域平均算法, 算法描述如下: 用某块面积内的所有像素点的像素平均值来替换这块面积中各个像素点的像素值, 可以达到像素模糊的目的。通常实现这一算法的方式是采用一个模板, 用该模板在图像上进行扫描, 扫描到某个位置时, 就将模板内除去某个点, 一般为模板中心点以外的所有点的像素值替换为该模板内所有像素点的平均值, 最后再将该点的像素值替换为平均值。但是这样的一种简单的滤波方式也存在一定的缺陷, 它不能有效保护图像的细节特征, 噪点跟图像的细节特征同时被模糊了。

在 Opencv 中, blur 函数就是用均值滤波器来对图片进行滤波操作。从式 (2) 表达就可看出其操作就是求均值。

$$k = \frac{1}{ksize.width \times ksize.height} \begin{bmatrix} 1 & \dots & \dots & 1 \\ \dots & \dots & & \\ 1 & \dots & \dots & 1 \end{bmatrix} \quad (2)$$

函数声明如下:

```
void blur(InputArray src, OutputArraydst, Size ksize,
Point anchor=Point(-1, -1), int borderType=BORDER_DEFAULT);
```

参数说明:

① InputArray 类型 src 参数, 表示待处理的源图像, 通常传入 Mat 类对象。

② OutputArray 类型 dst 参数, 表示生成的最终目标图像, 要跟待处理的源图像有相同的尺寸和类型, 为了保证这一要求, 一般可以采用 Mat::Clone 函数, 以待处理的源图像为模板初始化目标图像。

③ Size 类型 ksize 参数, 包含 ksize.width 和 ksize.height, 分别表示宽度和高度, 用来描述所需要用到的高斯核的大小, 宽度和高度可以不一样, 但是必须是正奇数或者 0。

④ Point 类型 anchor 参数, 带有默认参数 Point (-1, -1), 表示默认以模板中心点的邻域去进行滤波处理。

⑤ int 类型 borderType 参数, 带有默认值 BORDER_DEFAULT, 通常不需要用到。

两种滤波方式都是通过像素分布函数的方式进行滤波, 本论文通过实验对比的方式, 对两种不同的滤波方式在处理本论文所研究的问题上效果的差别, 将两种滤波方式处理的正反面零件轮廓图进行对比, 确定较优的滤波方式, 两种方案下的处理效果如图 3 至图 4 所示。

通过对比均值滤波与高斯滤波最终处理的正反面效果图不难发现, 均值滤波在正面的尺寸检测中出现了明显的偏差, 因而针对本论文所研究的对象, 高斯滤波要优于均值滤波, 故选

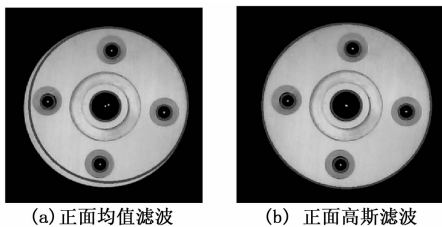


图 3 零件正面滤波效果对比

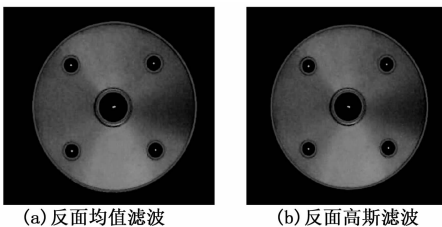


图 4 零件反面滤波效果对比

用高斯滤波。

1.3 高斯核设置

在确定了滤波方式为高斯滤波后，对高斯核的大小确定采用同样的方法。分别采用 3×3 、 5×5 的高斯核试验进行对比，效果如图 5 所示。

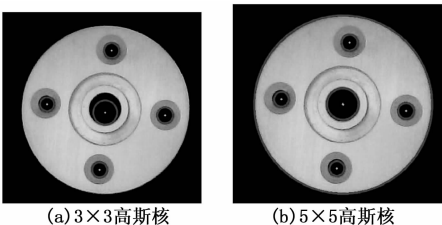


图 5 高斯核效果图

从 3×3 高斯核效果图与 5×5 高斯核效果图中不难看出，相对于 3×3 的高斯核而言， 5×5 高斯核的处理效果要更好一些。故选用 5×5 高斯核进行高斯滤波。

综上所述，本文图像处理最终采用的滤波方式为高斯滤波，所选用的滤波模板的大小为 5×5 的高斯核。

2 零件轮廓提取以及尺寸算法

在经过预处理后的图像中，图像噪声已经被大大降低了，这样的图像已经初步满足了下一步的轮廓提取以及尺寸计算处理的要求。从直观的视觉效果上来看，预处理后的图像变得“模糊了”，因此高斯滤波也称为高斯模糊，原图中的微弱噪点，经过这样“模糊”处理后就几乎不存在了，这样的预处理目的是为了保证下一步的边缘（轮廓）检测准确性，原图像是不能直接用于边缘检测的，因为图像边缘检测算法是基于图像强度的一阶导数、二阶导数。由于噪声对一阶导数和二阶导数的影响是非常大的，所以必须使用滤波的手段来降低噪声对边缘检测的影响，这也正是预处理必须进行滤波的具体原因所在。

2.1 图像轮廓增强

在图像轮廓增强前，本文将轮廓信息提取了出来，将提取出来的轮廓信息图与预处理后的灰度图进行加权处理，对预处

理后的灰度图像中的轮廓细节信息进行增强。轮廓增强后的图像与原灰度图像的对比如图 6 所示。在 Opencv 中，提供给开发者的加权操作的函数声明如下：

```
void addWeighted(InputArray src1, double alpha,
InputArray src2, double beta,
double gamma, OutputArray dst, int dtype=-1);
```

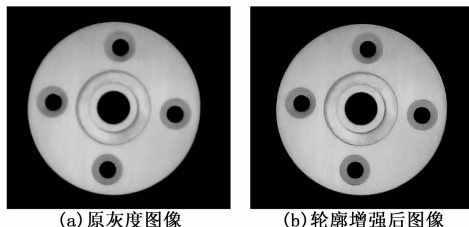


图 6 轮廓增强效果图

2.2 Hough 圆检测

对图像轮廓进行增强后的任务就是对增强的轮廓信息进行提取。针对本论文所研究的零件的具体操作就是查找出零件图像中的所有圆并计算各圆的尺寸，因此采设计了 Hough 圆检测的方式，进行零件图像中的圆查找。

Opencv 为开发者提供的霍夫圆变换函数声明如下：

```
void HoughCircles(InputArray image,OutputArray circles,
int method, double dp, double minDist,
double param1=100,double param2=100,
int minRadius=0, int maxRadius=0 );
```

通过该函数变换即可找到圆心和半径，然后通过自定义的 draw_circles 函数即可在原灰度图中画出圆心及轮廓。自定义 draw_circle 函数实现：

```
void draw_circles(Mat &image, vector<Vec3f> &circles)
{
for( size_t i = 0; i < circles.size(); i++ )
{
//提取出圆心坐标
Point center(cvRound(circles[i][0]), cvRound(circles[i][1]));
//提取出圆半径 , int &-tmp_radius
int radius = cvRound(circles[i][2]);
Vec_int.push_back(radius);
//圆心、圆心
circle(image, center, 2, Scalar(0,255,0), -1, 8, 0 );
circle(image, center, radius, Scalar(0,0,255), 3, 8, 0 );
}
return ;
}
```

自定义的 draw_circles () 有两个参数，第一个参数传入所要画出轮廓的图像，第二个参数传入霍夫圆变换函数的 circles 参数，画出的轮廓效果图如图 7 所示。

2.3 尺寸计算

轮廓提取后的最后一个关键工作就是计算各几何轮廓的尺寸。因为拍摄图片大小受摄像头距离工件远近及角度误差的影响较大，本文采用的是当前零件标定法，即在计算各轮廓圆的尺寸时采用的是实际尺寸比与像素比的比值权值进行计算。如在已知最外圆轮廓的半径后，将霍夫圆变换中用于保存像素半

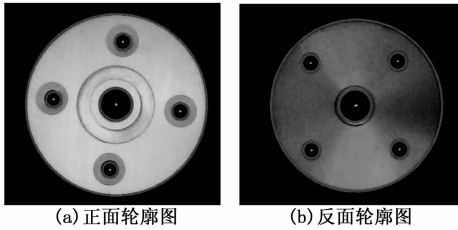


图7 圆形轮廓检测结果

径的 circles 中的最外侧大圆轮廓像素提取进行比值权值计算, 其他小圆像素半径信息提取出来, 通过此比值权值即可计算出实际其他小圆的轮廓半径。实际使用中只要摄像头与被测零件距离及垂直角度固定, 一次标定就可。

3 实验结果与分析

上述算法在 Visual Studio 2013 集成环境开发平台下利用了 MFC 设计并实现了 windows 下的用户交互界面。界面 UI 中使用了静态文本控件, 文字编辑控件, 按钮控件, 图片显示控件等, 如图 8 所示。仅仅有控件只能产生视觉上的效果, 为了使控件工作, 还需要进行下一步的操作, 添加响应函数, 给这些控件分配各自的处理任务。



图8 界面 UI 图

测试实验是将 CCD 摄像头安装于机床主轴箱外有壳体保护, 同时配有固定光源, 以保证零件曝光的稳定。该采集装置的工作流程如下: 当零件加工完部分工序置于工作台上时, 主轴停转切削液停止, 气枪吹干净零件表面油污后, 位于装置内部的摄像头即可对该零件进行实时监测, 并需要将当前的实时画面反馈到用户客户端时, 用户客户端可根据实时画面选择“拍照”进行图像采集。

在图 8 的操控界面下, 系统首先确定“选取摄像头”, 即图像的通道数在此设置为 0, 后用户点击“拍照”按钮, 这一时刻的实时画面图像将保存到本地的指定文件夹中, 后续处理的原始图像就是该图像。此时载入该图像用到的是 Opencv 开源库中的 cvLoadImage 函数, 该函数用于从文件中读取图像, 有两个入参: 图像通道数和文件名。图像的通道数有 3 种入参选择, 第一种传入大于 0 的参数时, 以 3 通道彩色图像方式载入该图像; 第二种传入小于 0 的参数时, 以保持图片本身不变的方式载入图像; 第三种传入 0 时, 以单通道灰度图像的方式载入。文件名即需要被读取的文件名, 本论文中即代码目录下的 front_image 以及 reverse_image 文件夹, 所拍摄的零件图像就保存于这两个文件夹中。由于本论文主要采取的方法是对灰度图像进行处理, 所以在载入图像时的通道设置为参数 0,

即单通道灰度图像。

对应的图形处理算法运行效果如图 9 中各部分信息所示, 图中将图像轮廓加强显示, 按照尺寸算法计算, 同时显示操作面板及几何尺寸。最后数据采集结果对比实物人工测量, 经过内径千分尺实际测量对比图中 5 个内孔尺寸验证, 所计算出圆的半径精度平均达到了 ± 0.01 mm。

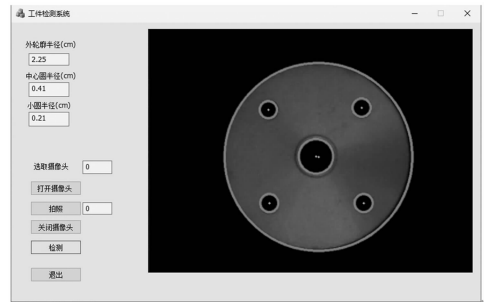


图9 图像处理算法仿真运行结果图

4 总结

本文针对在生产制造过程中对加工零件的在线检测需求, 具体以圆形零件为例进行图像处理与轮廓提取研究。通过对图像预处理的两种不同滤波方式的对比研究, 提出效果较好的高斯滤波算法, 并设计了图像轮廓增强算法、Hough 圆的查找方法、图像几何轮廓尺寸算法。然后基于 MFC 编程框架开发了相应的检测程序, 设计了简单易用的图形化界面, 该图像处理软件搭载 CCD 摄像头实现了在线加工零件的图像采集。最后通过该图像处理软件实验分析, 论证了图像轮廓增强对于轮廓提取的重要性及上述各算法的有效性。

参考文献:

- [1] 史博, 汪霖, 曹建福. 立体零件加工质量的在线图像检测方法 [J]. 仪器仪表学报, 2009, 30 (10): 2039-2042.
- [2] 黄品松, 徐杜, 蒋永平, 等. 零件尺寸图像检测数据处理与高精度检测方法 [J]. 计算机系统应用, 2011, 20 (9): 248-250.
- [3] 耿春明, 蔡东宝. 基于计算机视觉的机械零件检测系统设计 [J]. 计算机测量与控制, 2012, 20 (1): 38-40.
- [4] 水恒华, 赵永, 李刚, 等. 基于图像处理的在线零部件检测技术 [J]. 上海电力学院学报, 2016, 32 (6): 574-577.
- [5] 张道德. 机械零件图像识别的关键技术研究 [D]. 武汉: 华中科技大学, 2008.
- [6] 田小娟. 基于图像识别的零件几何量检测 [D]. 西安: 西安工业大学, 2009.
- [7] 施超. 基于计算机视觉的零配件监测系统研究与应用 [D]. 长春: 吉林大学, 2014.
- [8] 梁中源. 基于图像识别的零件在线自动监测系统 [D]. 济南: 山东科技大学, 2016.
- [9] Zhang Wei, Liu Hantao. Learning picture quality from visual distraction: Psychophysical studies and computational models [J]. Neurocomputing, 247 (2017) 183-191.
- [10] Arun Mohan, Sumathi Poobal. Crack detection using image processing: A critical review and analysis [J]. Alexandria Engineering Journal, (2017) 1-12.