

基于 MATLAB 的迈克尔逊干涉仪干涉视频处理^①

邹智星¹, 胡卫东¹, 王子涵², 谭兴文¹, 唐曦¹

1. 西南大学 物理科学与技术学院, 重庆 400715; 2. 重庆市朝阳中学, 重庆 400700

摘要: 在关于迈克尔逊干涉仪的实验中, 通常需要观察大量的干涉条纹吞吐变化以进行下一步的计算, 使用 CCD 采集到条纹变化视频信息后, 需要进一步处理以得到条纹吞吐量. 基于 MATLAB 的图像处理功能, 使用圆心搜索算法等方法完成了条纹吞吐量的自动识别. 解决了迈克尔逊干涉仪条纹变化观察困难的问题, 实现了计数自动化, 在教学科研领域均有一定的实用价值.

关键词: MATLAB; 搜索算法; 图像处理; 迈克尔逊干涉仪

中图分类号: TP274

文献标志码: A

文章编号: 1000-5471(2019)07-0052-05

迈克尔逊干涉仪是一种利用分割光波振幅的方法实现干涉的精密光学仪器, 对近代物理学的发展产生过重大影响. 由于迈克尔逊干涉仪装置简明、光路直观、精度高, 能以极高的精度测量长度或者折射率的微小变化, 它的调整和使用方法在光学技术中具有典型性和代表性, 所以它在干涉测量技术中具有重要的地位和作用, 根据迈克尔逊干涉仪的基本结构原理设计和研发的各种精密仪器已广泛应用于生产和科学研究领域^[1].

在使用到迈克尔逊干涉仪的实验中, 不可避免地要得到干涉圆环的级次变化量 Δk . 而为了提高实验结果的准确度, 需要的 Δk 从几百到几千不等, 用肉眼观察计数是不现实的. 用 CCD 采集到干涉图样变化的视频后, 可使用 MATLAB 对视频进行进一步的处理, 以得到 Δk 的值.

本文计算 Δk 的基本思路是对干涉圆环中心的明暗交替情况进行记录, 干涉圆环中心明暗交替次数的 $1/2$ 即为 Δk 的值.

由 CCD 得到的视频画面一般会含有许多噪点, 且干涉圆环的中心不一定在画面中心, 所以需要先对视频进行预处理, 并找到干涉圆环的中心.

1 预处理

由 CCD 得到的视频画面一般会含有噪声, 且干涉圆环的中心不一定在画面中心, 所以需要先对视频进行预处理, 并找到干涉圆环的中心.

1.1 中值滤波

中值滤波是一种常见的降噪算法, 其基本原理是把图像中一点的值用该点的一个邻域中各点值的中值代替, 使得该范围内的各点像素值趋于相同, 此算法可有效消除图像中孤立噪声点.

$$V_{\text{out}} = \text{median}\{a_1, a_2, \dots, a_n\} \quad (1)$$

其中, V_{out} 是待处理像素的输出值, median 是取平均值, $\{a_n\}$ 是待处理像素值某邻域中各点的像素值. 所

① 收稿日期: 2018-05-24

基金项目: 重庆市教委雏鹰计划研究项目, 西南大学"国家级大学生创新创业训练计划"项目(201710635011).

作者简介: 邹智星(1997-), 男, 大学本科学生, 主要从事物理虚拟仿真实验的研究.

以, 中值滤波后的图像 $g(x, y)$ 是 (x, y) 为中心的 $N \times N$ 窗口的各像素的灰度中间值^[2-3].

1.2 二值化

图像的二值化即把灰度图中所有像素点的值变为 0 或者 255, 进一步增强图像的对比度^[4], 消除干涉圆环边缘强度递减对搜索圆环圆心造成的影响. 图像的二值化需要确定一个阈值, 把灰度值在阈值以上的像素点的像素值变为 255, 在阈值以下的像素点的像素值变为 0, 即完成了图像的二值化.

由于本问题中的原图像不同部分灰度差异明显, 图像的直方图类型是典型的双峰直方图(图 1). 可以通过文献 5 中的方法来确定阈值的大小, 步骤为^[5]:

- 1) 计算像素灰度平均值(avg)、标准偏差(sigma);
 - 2) 以像素平均值为分界点, 分别求出左右部分最大值的位置;
 - 3) 以两峰值位置的中点作为所要求的阈值.
- 本文将阈值取为 127.5.

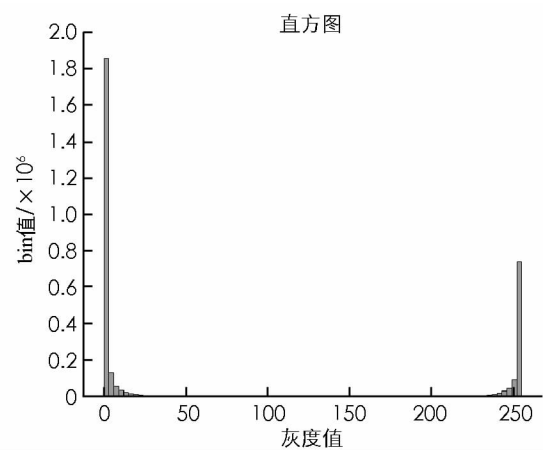


图 1 图像直方图

2 圆心搜索

虽然干涉圆环的中心不在画面中心, 但整段视频中, 只要确保仪器在实验过程中未被移动, 干涉圆环中心的位置就不会改变, 所以只需要确定视频中任意一帧干涉圆环中心的位置, 就能确定所有帧中圆环圆心的位置(图 2).

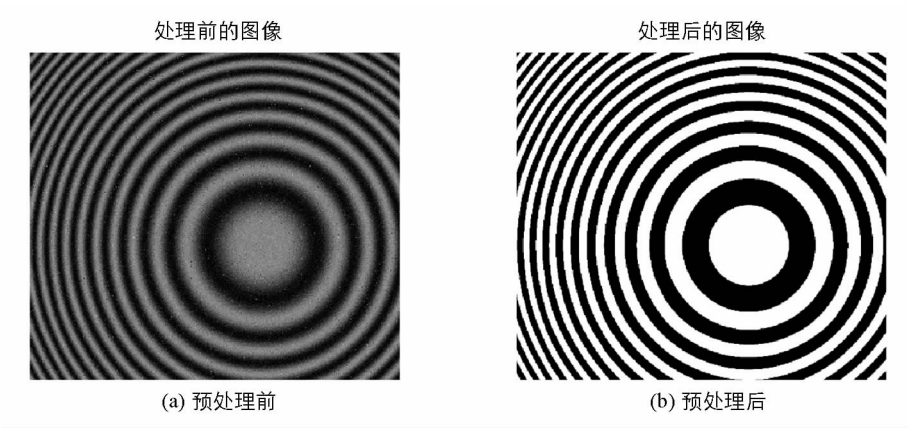


图 2 预处理前、后对比

2.1 方法概述

通常使用霍夫(Hough)变换^[6]在图像中搜索圆或直线, 但对于大量同心圆环, Hough 变换的效率较低. 鉴于本问题具有一定的特殊性, 可简化算法以提高效率.

在干涉图像中寻找圆心的思想为:
设视频帧的分辨率为 $w \times h$, 在已经过预处理的某一视频帧中, 取一适当大小的探测圆 C , 保证圆 C 能与干涉圆环图像相交. 将圆周上等间距的 n 个点的像素值进行对比, 若这 n 个点的像素值全为 0 或全为 255, 则说明此时圆 C 与干涉图样的某个亮环或某个暗环重合, 圆 C 的圆心在误差允许范围内可以认为是干涉图样的中心.

若圆周上 n 个点的像素值不完全相同, 则圆 C 未与干涉圆环重合, 此时应多次微调探测圆位置, 重复检测圆周上 n 个点的像素值, 直到它们全部相同为止. 微调的步长应取一个小值, 避免出现探测圆 C 越过干涉圆环导致搜索失败的情形. 但如果干涉圆环宽度过大, 采取小步长会导致探测圆搜索到的位置与干涉圆环相切, 导致搜索到的圆心与实际圆心偏离较大的情况出现. 对于这种情况, 可以让探测圆分别从左上

角和右下角开始搜索两次,取两次搜索到的圆心位置的平均值以消除误差.

若探测圆已遍历图像,还是没有使圆周上 24 点的像素值相同,说明探测圆的半径过大,此时应减小探测圆半径,重复上述步骤,直到找到为止.

2.2 具体步骤

2.2.1 确定探测圆圆心位置

设视频帧的宽为 w , 高为 h , 圆心移动在 x 方向或 y 方向移动的次数为 loop , 探测圆的圆心位置为 $O(o_{x_i}, o_{y_i})$. 在本问题中对圆半径的要求并不严苛, 只要不太大或太小都不影响圆心的搜索, 可取一适中大小, 这里取为 w 与 h 中较短的那个的 $1/4$, 即

$$r = \frac{\min(w, h)}{4} \quad (2)$$

由于探测圆的圆周最多与图片边缘相切, 所以圆心移动的范围位于图像中一大大小为 $(w-2r) \times (h-2r)$ 的矩形之中. 而因为移动次数为 loop , 所以圆心在 x 方向每次移动的距离为 $(w-2r)/\text{loop}$, 在 y 方向每次移动的距离为 $(h-2r)/\text{loop}$, 所以圆心位置可表示为:

$$\begin{aligned} o_{x_i} &= r + (i_x - 1) \frac{(w - 2r)}{\text{loop}} & i_x &= 1, 2, \dots, \text{loop} \\ o_{y_i} &= r + (i_y - 1) \frac{(h - 2r)}{\text{loop}} & i_y &= 1, 2, \dots, \text{loop} \end{aligned} \quad (3)$$

其中 i_x 与 i_y 分别为在 x 方向上的迭代次数.

2.2.2 确定对应某个圆心 $O(o_{x_i}, o_{y_i})$, 圆周上 n 个等间距的点的位置 (x_k, y_k) .

采用极坐标, 先取 n 个离散的等角间距的角度值为

$$\theta_k = k \frac{2\pi}{n} \quad k = 1, 2, \dots, n \quad (4)$$

则圆周上 n 个点的坐标可以表示为

$$\begin{aligned} x_k &= [o_{x_i} + r * \cos(\theta_k)] \\ y_k &= [o_{y_i} + r * \sin(\theta_k)] \end{aligned} \quad k = 1, 2, \dots, n \quad (5)$$

其中 $f(x) = [x]$ 为取整函数, 因为二值化后的图像在 MATLAB 中是一个矩阵, 若要把 x_k 与 y_k 作为矩阵索引, 则它们必须是整数.

2.2.3 比较 2.2.2 中 n 个点上的像素值是否一致.

设经过预处理后的视频帧对应一个大小为 $w \times h$ 的矩阵 I , 则可取到 n 个像素值

$$I_k = I(x_k, y_k) \quad k = 1, 2, \dots, n \quad (6)$$

若对于某个 $O(o_{x_i}, o_{y_i})$, 有 $I_1 = I_2 = \dots = I_n$, 说明这 n 个点处于某个亮环或某个暗环上, 此 $O(o_{x_i}, o_{y_i})$ 即为同心圆的圆心. 圆心搜索结果见图 3, 圆心搜索函数源代码为:

```
function[center_x, center_y]=SearchCircle(pct)
%输入预处理后的图片数据, 输出同心圆圆心坐标
%% 参数设定
[h, w]=size(pct); %获得图像尺寸
r=min(w/4, h/4); %探测圆初始半径设定
cx=zeros(1); cy=zeros(1); cx(1)=r; cy(1)=r;
%[cx, cy]为探测圆圆心坐标
loop_num=100; n=0; theta=0: pi / 36 : 2 * pi;
temp=zeros(1, length(theta)); temp(1)=1; %
temp 为用于储存遍历中被探测点上颜色信息的数组
%% 探测圆遍历
```

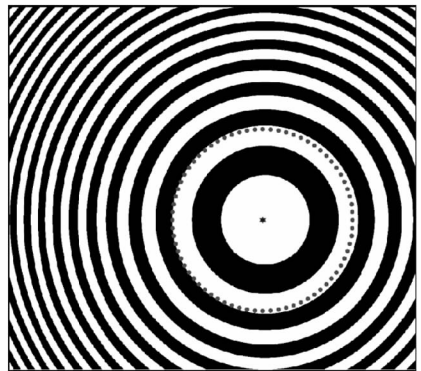


图 3 圆心搜索的结果

```

while sum(temp) / length(temp) ~= temp(1)
    for nx=1: loop_num
        for ny=1 : loop_num
            cx(nx)=r + (nx - 1) * (w - 2 * r) / loop_num;
            cy(ny)=r + (ny - 1) * (h - 2 * r) / loop_num; %根据式(5)计算出探测圆圆心的位置
            sx=round(cx(nx) + r * sin(theta));
            sy=round(cy(ny) + r * cos(theta)); %计算相应圆周上的各点的位置
            sx(sx == 0)=1; sy(sy == 0)=1;
            %如果 temp 中的所有元素均为 1 或均为 0, 则说明找到了圆环, 跳出循环. 反之则继续
            for i=1: length(theta)
                temp(i)=pct(sy(i), sx(i));
            end
            if sum(temp) / length(temp) == temp(1)
                break;
            end
        end
        if sum(temp) / length(temp) == temp(1)
            break;
        end
    end
    n=n + 1; r=r - 2; cx(1)=r; cy(1)=r;
end
end

```

3 计算 Δk 的值

找到同心圆环圆心后, 记录视频中每一帧圆心处的像素值并存入一个数组中, 可得到长度为总帧数的一维数组, 其元素均为 0 或 255. 将数组内相邻元素进行对比, 若相邻元素相等, 则对应帧圆环中心的颜色没有改变; 若相邻元素不等, 则对应帧圆环中心颜色改变. 每两次改变对应一个干涉圆环的“冒出”或“缩进”, 遍历数组, 获取改变次数 N , 则

$$\Delta k = \frac{N}{2}$$

得到 Δk 后, 即可根据光程差公式, 计算出单色光波长、未知介质折射率等数据.

4 结 语

基于 MATLAB 的干涉视频处理方法, 若结合 CCD 和 LabVIEW 等硬件相关的部分, 可实现迈克尔逊干涉仪相关实验的高度自动化. 本文通过圆心搜索算法确定计数区域, 并通过计数区域内像素值的变化得到条纹级数的变化, 避免繁琐的实际操作. 用多个不同的干涉视频进行测试后, 与手动计数所得的准确值进行对比, 可发现本方法具有高精度度. 且因图像处理的步骤不复杂, 较其他的自动计数方法而言本方法具有更快的运行速度, 可用于快速测量材料的物理性能(比如树脂玻璃板的折射率或金属棒的热膨胀系数等)等领域的研究工作, 在教学科研领域均具有一定的实践价值.

参考文献:

- [1] 陈晓莉, 王培吉. 普通物理实验: 下 [M]. 西南师范大学出版社, 2012.
- [2] 唐娅琴. 几种图像平滑去噪方法的比较 [J]. 西南大学学报(自然科学版), 2009, 31(11): 125-128.
- [3] CHEN T, MA K K, CHEN L H. Tri-state Median Filter for Image Denoising [J]. IEEE Transactions on Image Processing, 1999, 8(12): 1834-1838.
- [4] SAUVOLA J, PIETIKÄINEN M. Adaptive Document Image Binarization [J]. Pattern Recognition, 2000, 33(2): 225-236.
- [5] 梁华为. 直接从双峰直方图确定二值化阈值 [J]. 模式识别与人工智能, 2002, 15(2): 253-256.
- [6] 林金龙, 石青云. 用点 Hough 变换实现圆检测的方法 [J]. 计算机工程, 2003, 29(11): 17-18.
- [7] 邓 巍, 丁为民, 张 浩. MATLAB 在图像处理和分析中的应用 [J]. 农机化研究, 2006(6): 194-198.
- [8] 张德丰. MATLAB 数字图像处理 [M]. 北京: 机械工业出版社, 2012.

Interference Pattern Processing of Michelson Interferometer Based on MATLAB

ZOU Zhi-xing¹, HU Wei-dong¹,
WANG Zi-han², TAN Xing-wen¹, TANG Xi¹

1. School of Physical Science and Technology, Southwest University, Chongqing 400715, China;

2. Chongqing Chaoyang Middle School, Chongqing 400700, China

Abstract: In the Michelson interferometer experiment, we usually need to observe the variation of interference fringes quantity for further calculation. After using CCD to capture the fringe change video information, further processing is essential to obtain the interference fringe quantity. This paper is based on the image processing function of MATLAB, using the center search algorithm to complete the automatic identification of stripe quantity. The complication of observing the fringe variation of Michelson interferometer has been solved in this paper, the counting automation realized with certain practical value in the field of teaching and scientific research.

Key words: MATLAB; search algorithm; image processing; Michelson interferometer

责任编辑 潘春燕