

# 基于最小二乘法拟合直线的研究

张艳莉

(广东农工商职业技术学院 机电系, 广东 广州 510507)

摘要: 文章针对二值图像中直线方向的检测问题, 提出了基于最小二乘法的检测算法。给出了最小二乘法的基本原理, 并以 Visual C++ 为开发平台, 用 C / C++ 编程实现了原始图像的输入和拟合直线图像的输出。

关键词: 二值图像; 最小二乘法; 拟合直线

中图分类号: O212

文献标识码: A

文章编号: 1673-1131(2014)11-0044-02

## The research of Fitting Straight-line least square method

Zhang Yanli

(Guangdong AIB Polytechnic College, Guangdong, Guangzhou, 510507)

Abstract: In this paper, a straight-line binary image in the direction of the detection problem, a least squares method based on the detection algorithm. Given the basic principle of least squares, and C / C++ programming to achieve the original image of the input and output image of a fitting straight line.

Keywords: binary image; least square method; Fitting straight line

### 0 引言

直线检测是图像处理中的一个重要内容, 为图像理解提供了直接依据。现有的直线检测算法主要有两大类: 通过对图像的处理得到目标的边界点集合, 然后利用 Hough 变换提取目标边界上的直线; 在对图像预处理后, 直接获取目标的边界线集合, 然后在该集合中进行直线段识别。

本文给出了基于最小二乘法拟合直线的详细过程: 先把二值图像中曲线点的坐标读出, 再用最小二乘法拟合成直线输出。本文的结构如下: 第二部分讨论了最小二乘法的原理; 第三部分介绍了对于给定图像实现直线拟合的具体算法; 第四部分是全文小结。

### 1 最小二乘法

在物理实验中经常要观测两个有函数关系的物理量。根据两个量的许多组观测数据来确定它们的函数曲线, 这就是实验数据处理中的曲线拟合问题。这类问题通常有两种情况: 一种是两个观测值  $x$  与  $y$  之间的函数形式已知, 但一些参数未知, 需要确定未知参数的最佳估计值; 另一种是  $x$  与  $y$  之间的函数形式还不知道, 需要找出它们之间的经验公式。后一

种情况常假设  $x$  与  $y$  之间的关系是一个待定的多项式, 多项式系数就是待定的未知参数, 从而可采用类似于前一种情况的处理方法。

#### 1.1 最小二乘法原理

在两个观测量中, 往往总有一个量精度比另一个高得多, 为简单起见把精度较高的观测量看作没有误差, 并把这个观测量选作  $x$ , 而把所有的误差只认为是  $y$  的误差。设  $x$  和  $y$  的函数关系由理论公式

$$y = f(x; c_1, c_2, \dots, c_m) \quad (1)$$

给出, 其中  $c_1, c_2, \dots, c_m$  是  $m$  个要通过实验确定的参数。对于每组观测数据  $(x_i, y_i), i = 1, 2, \dots, N$ , 都对应于  $xy$  平面上一个点。若不存在测量误差, 则这些数据点都准确落在理论曲线上。只要选取  $m$  组测量值代入式 (2-1-1), 便得到方程组

$$y_i = f(x_i; c_1, c_2, \dots, c_m) \quad (2)$$

式中  $i = 1, 2, \dots, m$ , 求  $m$  个方程的联立解即得  $m$  个参数的数值。显然  $N < m$  时, 参数不能确定。

在  $N > m$  的情况下, 式 (2) 成为矛盾方程组, 不能直接用解方程的方法求得  $m$  个参数值, 只能用曲线拟合的方法来处理。

### 3 实验结果分析

以 UUV 协调算法为例, 将传统移动通信算法与分层移动通信优化算法进行对比, 并将通信的误码率

$$Q_s = R \left[ \sqrt{\frac{1}{(K/QP)\cos^2\eta_k + (Q_0/2QV_f)}} \right] \quad (7)$$

作为具体的对比依据。通过计算可知, 对于在通信路径中移动速度较慢的节点而言, 其在上述两种算法下的误码率基本一致, 而对于移动速度较快的节点, 以 UUV 协调算法为基础的传统移动通信算法的误码率要远高于分层移动通信优化算法的误码率, 充分显示了分层移动通信优化算法处理节点误码率的优越性。

### 4 结语

本文通过对移动通信的基本原理进行阐述, 进而从通信

节点的分层处理以及移动通信分层实现等方面对基于分层移动通信的优化算法展开了详细分析。通过实验结果可知, 利用分层移动的优化算法对通信节点进行优化处理, 可以有效降低误码率并提高移动通信的质量和效率, 从而满足用户对通信的实际需求。

参考文献:

- [1] 魏焱. 蜂窝移动通信系统物理小区识别自组织技术研究[D]. 北京邮电大学, 2013
- [2] 张竹. IP/LEO 卫星网络中的移动性管理技术研究[D]. 哈尔滨工业大学, 2013
- [3] 王筱萍. 基于分层 Copula 函数的分布估计算法研究[D]. 兰州理工大学, 2013

设测量中不存在系统误差,或者说已经修正,则  $y$  的观测值  $y_i$  围绕着期望值  $\langle f(x; c_1, c_2, \dots, c_m) \rangle$  摆动,其分布为正态分布,则  $y_i$  的概率密度为

$$p(y_i) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left\{-\frac{[y_i - f(x_i; c_1, c_2, \dots, c_m)]^2}{2\sigma_i^2}\right\} \quad (3)$$

式中  $\sigma_i$  是分布的标准误差。为简便起见,下面用  $C$  代表  $(c_1, c_2, \dots, c_m)$ 。考虑各次测量是相互独立的,故观测值  $(y_1, y_2, \dots, y_N)$  的似然函数

$$L = \frac{1}{(\sqrt{2\pi})^N \sigma_1 \sigma_2 \dots \sigma_N} \exp\left\{-\frac{1}{2} \sum_{i=1}^N \frac{[y_i - f(x_i; C)]^2}{\sigma_i^2}\right\} \quad (4)$$

取似然函数  $L$  最大来估计参数  $C$ , 应使

$$\sum_{i=1}^N \frac{1}{\sigma_i^2} [y_i - f(x_i; C)]^2 = \min \quad (5)$$

取最小值:对于  $y$  的分布不限于正态分布来说,式(5)称为最小二乘法准则。若为正态分布的情况,则最大似然法与最小二乘法是一致的。因权重因子  $\omega_i = 1/\sigma_i^2$ ,故式(5)表明,用最小二乘法来估计参数,要求各测量值  $y_i$  的偏差的加权平方和为最小。

根据式(5)的要求,应有方程组

$$\sum_{i=1}^N \frac{1}{\sigma_i^2} [y_i - f(x_i; C)] \frac{\partial f(x_i; C)}{\partial C_k} \Big|_{C=\hat{C}} = 0 \quad (k=1, 2, \dots, m) \quad (6)$$

解方程组(6),即得  $m$  个参数的估计值  $\hat{c}_1, \hat{c}_2, \dots, \hat{c}_m$ , 从而得到拟合的曲线方程  $f(x; \hat{c}_1, \hat{c}_2, \dots, \hat{c}_m)$ 。

## 1.2 直线的最小二乘拟合原理

曲线拟合中最基本和最常用的是直线拟合。设  $x$  和  $y$  之间的函数关系由直线方程

$$y = a_0 + a_1 x \quad (7)$$

给出。式中有两个待定参数,  $a_0$  代表截距,  $a_1$  代表斜率。对于等精度测量所得到的  $N$  组数据  $(x_i, y_i) \quad i = 1, 2, \dots, N$ ,  $x_i$  值被认为是准确的,所有的误差只联系着  $y_i$ 。下面利用最小二乘法把观测数据拟合为直线。

前面指出,用最小二乘法估计参数时,要求观测值  $y_i$  的偏差的加权平方和为最小。对于等精度观测值的直线拟合来说,由式(5)可使

$$\sum_{i=1}^N [y_i - (a_0 + a_1 x_i)]^2 \Big|_{a=\hat{a}} \quad (8)$$

最小,即对参数  $a$  (代表  $a_0, a_1$ ) 最佳估计,要求观测值  $y_i$  的偏差的平方和为最小。

根据式(8)的要求,整理后得到正规方程组

$$\begin{cases} \hat{a}_0 N + \hat{a}_1 \sum x_i = \sum y_i, \\ \hat{a}_0 \sum x_i + \hat{a}_1 \sum x_i^2 = \sum x_i y_i. \end{cases} \quad (9)$$

解正规方程组便可求得直线参数  $a_0$  和  $a_1$  的最佳估计值  $\hat{a}_0$  和  $\hat{a}_1$ , 即

$$\hat{a}_0 = \frac{(\sum x_i^2)(\sum y_i) - (\sum x_i)(\sum x_i y_i)}{N(\sum x_i^2) - (\sum x_i)^2} \quad (10)$$

$$\hat{a}_1 = \frac{N(\sum x_i y_i) - (\sum x_i)(\sum y_i)}{N(\sum x_i^2) - (\sum x_i)^2} \quad (11)$$

## 2 基于最小二乘法的直线拟合方法

本文的算法是基于给定的二值图像的,若不是二值图像,则首先应把图像二值化,再应用本算法。算法步骤如下:

(1)调用 readBmp 函数,读取给定的二值图像信息,存入设定的数组中。这里应特别注意,由于二值图像是每行以 4 的倍数存储图像信息的,而老师给的图像是  $408 \times 321$ ,即每行只有 51 个字节,所以存储时计算机自动每隔 51 个字节补一个字节的 00。根据最小二乘法原理,我们只需要把图像中的白色点的坐标取出即可,所以存储时不仅要把图像的位图信息去除,而且要把每行的 00 字节去掉,因为白色点的像素值即是 0,若不去除,则直接影响拟合结果。

(2)调用数组,利用最小二乘法的原理把有用的点拟合成直线。弄清了最小二乘法的原理后,程序中直接以公式代入有用点的坐标即可,最后求得拟合直线的斜率是 309.132272,截距是 -0.634538。

(3)调用 saveBmp 函数,把拟合结果以图像形式重新输出。这里有一点需要注意,输出图像时首先要把原图像的位图头文件信息输出,再输出拟合直线的信息。

程序运行结果如下:

```
bfSize=16754 ,bfOffBits=62
lineByte=52 ,bmpHeight=321
width=408 ,height=321 ,biBitCount=1
a0=309.132272 ,a1=-0.634538
Press any key to continue
```

其中 bfSize 是文件的大小,以字节为单位, bfOffBits 是从文件头开始到实际的图像数据之间的字节偏移量,这个参数非常有用,因为位图信息头和调色板的长度会根据不同情况而变化,我们可以用这个偏移值迅速地从文件中读取到位数据, lineByte 是每行存储的字节数, bmpHeight 是图像的高度, biBitCount 是比特数/像素, a0 是拟合直线的斜率, a1 是截距。

## 3 结语

本文调用 VC++ 源程序中的 readBmp 函数和 saveBmp 函数,实现了对给定的二值图像中的曲线运用最小二乘法进行直线拟合,并输出了精确的拟合图像。

参考文献:

- [1] 宋坤,刘锐宁,李伟明. Visual C++ 开发技术大全[J]. 人民邮电出版社, 2007(3): 360-363
- [2] 何斌,马天予,王运坚,等. Visual C++ 数字图像处理[J]. 人民邮电出版社, 2002(12): 12-36
- [3] 谭浩强. C 程序设计(第二版)[J]. 清华大学出版社, 1999(6)
- [4] 谢凤英,赵丹培. Visual C++ 数字图像处理[J]. 电子工业出版社, 2008(5)

作者简介: 张艳莉(1983-),女,河南鹤壁人,讲师,硕士,研究方向为通信与信息系统。