

基于遗传算法的影子定位模型



姓名	学院	年级
庄楚斌	自动化学院	2013
张俊铭	材料工程学院	2013
吴 慧	数学与统计学院	2013

2015年9月11~14日

基于遗传算法的影子定位模型

摘要

本文针对影子定位问题，建立了影子长度变化模型，对物体的影子长度关于各个参数的变化规律进行了描述，并建立回归模型和非线性规划模型对问题进行求解。

针对问题一，要求建立影子长度的变化模型对直杆的影子变化过程加以描述。首先，我们构建了太阳高度角与方位角关于各参变量之间的函数关系式；接着，考虑到平年闰年的影响，为了保证精确度，我们对公式中的年份，经度和时刻等重要变量进行了修正并优化了模型；最后我们将数据代入修正之后的模型中，求解出题目给定时间下对应的直杆的影子长度，并绘制出影子长度随时间的变化曲线来加以描述，得到影子在 **12:20** 左右长度最小为 **3.7m**，在 **9:00** 左右长度最大为 **6.6m**。

针对问题二，要求根据某固定直杆的影子顶点坐标数据确定直杆所处的地点。首先，基于问题一建立的影子长度的变化模型，我们构建了二元非线性方程来描述影子长度关于各个参数的关系，其中时间作为自变量，影子长度作为方程的因变量，经度，纬度，杆长作为方程的系数；接着，针对模型的求解，一方面，我们使用 **1stopt(First Optimization)** 软件结合 **麦夸特法外加通用全局最优算法** 对数据进行了非线性拟合，精确预测出该地的经纬度为 **108.5° E, 19.2° N**；另一方面，我们采用最小二乘法，以实现计算的直杆影子长度和附件提供的影子长度的均方差最小为优化目标，建立了非线性规划模型，并使用 **遗传算法** 进行了优化求解，得到结果为 **108.83°E, 19.15°N** 和 **110.64°E, 0.59°N**，可能位置为 **海南省西部以及马来西亚西海岸**；最后将结果地点用 **Excel Power Map** 进行三维重现。

针对问题三，要求根据某固定直杆在水平地面上的太阳影子顶点坐标数据确定直杆所处的地点和日期。我们在问题二中建立的非线性规划模型的基础上同样构建了二元非线性方程，其中时间作为自变量，影子长度作为方程的因变量，经度，纬度，杆长，年份以及积日作为方程系数（决策变量），将问题的求解转化为一个实现目标函数值最小的优化问题，然后将数据代入非线性规划模型中用 **Matlab** 进行求解，结果显示：附件 2 中直杆所在的可能经纬度坐标为 **85.18E, 40.84N**，时间为：2015 年 6 月 30 日；附件 3 中直杆所在的可能经纬度坐标为 **110.12E, 22.33N** 和 **112.13E, 36.33N**，时间分别为 2015 年 1 月 7 号和 2015 年 1 月 4 号。

针对问题四，要求通过直杆在太阳下的影子变化的视频，确定视频的拍摄地点以及拍摄日期。首先，由于视频图像中像平面坐标系与路面坐标系存在映射关系，将其转化为 **单目摄像头二维成像平面的三维空间测距问题**，我们利用图像消失点的属性，标定出视频测速中像平面与路面坐标的映射关系。然后，根据图像射影几何原理，计算出画面的消失点坐标。并利用摄像机的各个参数与消失点的关系求解出像平面与路面坐标的映射关系，以此求解各个时刻影子的实际长度；最后，利用得到的影子长度结合第三问的模型与方法计算出：**已知时间情况下的经纬度为：110.09E, 29.01N；未知时间情况下为：110.94E; 29.17N。日期为：2015 年 7 月 23 日。**

综上所述，本文构建了直杆影子的定位模型，通过分析物体的影子变化来确定物体所处的地点和日期。该模型可广泛运用于各种场合，具有较高的使用价值。

关键字：太阳高度角麦夸特法遗传算法非线性规划模型射影几何坐标转换

1. 问题重述

1.1. 背景

如何确定视频的拍摄地点和拍摄日期是视频数据分析的重要方面，太阳影子定位技术就是通过分析视频中物体的太阳影子变化，确定视频拍摄的地点和日期的一种方法。

1.2. 问题

1) 建立影子长度变化的数学模型，分析影子长度关于各个参数的变化规律，并应用你们建立的模型画出 2015 年 10 月 22 日北京时间 9:00-15:00 之间天安门广场（北纬 39 度 54 分 26 秒,东经 116 度 23 分 29 秒）3 米高的直杆的太阳影子长度的变化曲线。

2) 根据某固定直杆在水平地面上的太阳影子顶点坐标数据，建立数学模型确定直杆所处的地点。将你们的模型应用于附件 1 的影子顶点坐标数据，给出若干个可能的地点。

3) 根据某固定直杆在水平地面上的太阳影子顶点坐标数据，建立数学模型确定直杆所处的地点和日期。将你们的模型分别应用于附件 2 和附件 3 的影子顶点坐标数据，给出若干个可能的地点与日期。

4) 附件 4 为一根直杆在太阳下的影子变化的视频，并且已通过某种方式估计出直杆的高度为 2 米。请建立确定视频拍摄地点的数学模型，并应用你们的模型给出若干个可能的拍摄地点。如果拍摄日期未知，你能否根据视频确定出拍摄地点与日期？

2. 模型的假设以及符号系统

2.1. 模型的假设

- 1) 将地球视为一个具有平坦表面的均匀球体，即不考虑山丘、沟壑、海洋等地形因素对直杆影长的影响；
- 2) 为便于计算影长，将问题四中由两个较小直杆组成的直杆视为一个标准的长为 2m 的直杆；
- 3) 不考虑光在空气中的漫反射现象，即所有的光在空气中都是沿直线传播的，不会发生角度偏移；
- 4) 因为太阳相对地球为无限远光源，所以将太阳光视为平行光

2.2. 符号系统

符号	符号意义
h	太阳相对被测点的高度角
A	太阳相对被测点的方位角
δ	太阳的赤纬角
S_{\odot}	真太阳时
S_d	地方时
F_{\odot}	真太阳分
E_t	时差
h_{\odot}	太阳相对被测点的高度角（修正值）
JD	经度
JF	经分
N	积日
$length_i$	长杆影子长度
f	摄像机有效焦距（ mm ）
h	摄像机距离地面高度（ m ）
$X_wY_wZ_w$	地面坐标系
$X_cY_cZ_c$	像平面坐标系
$X_{cs}Y_{cs}Z_{cs}$	倾斜坐标系
$X_{wh}Y_{wh}Z_{wh}$	$z_w=h$ 时的地面坐标系
L	相平面距离路面坐标系水平距离（ mm ）
K	内部矩阵
R	旋转矩阵
T	平移矩阵
O	摄像机光轴中心
P_x	视频帧横向像素（ $pixel$ ）
P_y	视频帧纵向像素（ $pixel$ ）
s	摄像机内部旋转角（ $^{\circ}$ ）
p	摄像机光轴在地面投影与道路面之间的夹角（ $^{\circ}$ ）
t	摄像机的光轴与与地面之间的夹角（ $^{\circ}$ ）

3. 问题一的建模求解

3.1. 问题的分析

该问要求建立影子长度变化的数学模型并分析影子长度关于各个参数的变化规律以及结合实际情况建立影子长度随时间变化的方程。因此首先需要确定各参数对影子长度的影响情况。

忽略空气中太阳光的散射以及地面对光的漫反射，根据三角形的相似定理以及光沿直线传播定理，影子的长度与实物长度的比值即为太阳的高度角的正切，而影子的方位与太阳的方位角密切相关，所以确定影子的长度以及方向的前提是得知太阳相对于实物的高度角以及方位角。

由于已知量为所处位置的经纬度、时间以及垂直物的长度，所以依据现有数值，我们可以推导出太阳的高度角以及方位角并最后得到影子的长度以及方位。

3.2. 高度角以及方位角

以待测物 O 点为球心，由图 3-1 可得：

太阳光光线入射方向与天顶方向的夹角为天顶角，太阳高度角 h 和太阳天顶角互为余角。

太阳方位角 A 即太阳所在的方位，指太阳光线在地平面上的投影与当地子午线的夹角（可近似地看作是竖立在地面上的直线在阳光下的阴影与正南方的夹角）。方位角以正南方向为零，由南向东向北为负，由南向西向北为正。

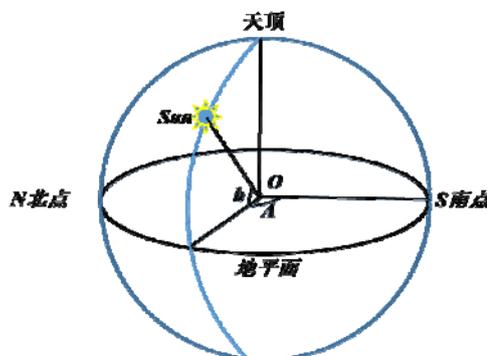


图 3-1.太阳的高度角以及方位角

3.3. 高度角以及方位角的计算公式^[1]

太阳高度角随着地方时和太阳的赤纬的变化而变化。太阳赤纬（与太阳直射点纬度相等）以 δ 表示，观测地地理纬度用 φ 表示，地方时（时角）以 τ 表示，有太阳高度角的计算公式：

$$\sin h = \sin \delta \sin \varphi + \cos \delta \cos \varphi \cos \tau \quad (3-1)$$

3.3.1. 太阳赤纬角

地球的自转轴（极轴）与地球绕太阳公转的轨道平面（黄道面）呈 66.5° 角并保持定值，日地中心的连线与赤道面间的夹角无时无刻不处在变化之中，这个角度就称是太阳赤纬角。（如图 3-2）

由于太阳赤纬角在周年运动中任何时刻的具体值都是严格已知的，所以它（ δ ）可用一个数学表达式表述：

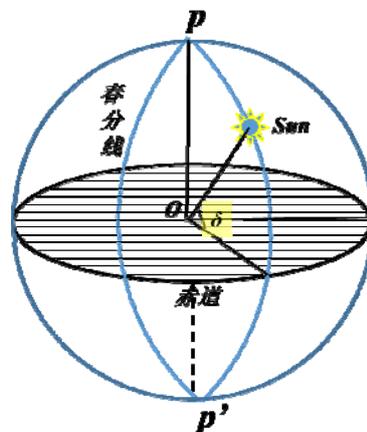


图 3-2.太阳的赤纬角

$$\delta = 0.3723 + 23.2567 \sin \theta + 0.1149 \sin 2\theta - 0.1712 \sin 3\theta - 0.758 \cos \theta + 0.3656 \cos 2\theta + 0.0201 \cos 3\theta \quad (3-2)$$

式中 θ 是日角，其计算公式为：

$$\theta = \frac{2\pi t}{365.2422} \quad (3-3)$$

这里 t 由两部分组成，即：

$$t = N - N_0 \quad (3-4)$$

式中 N 为积日，即日期在年内的顺序号。

其中 N_0 的计算各公式为：

$$N_0 = 79.6764 + 0.2422 \times (\text{year} - 1985) - \text{INT} \left[\frac{(\text{year} - 1985)}{4} \right] \quad (3-5)$$

其中 $\text{INT}[x]$ 为取整数部分， year 为当年年份。

3.3.2. 时差

太阳时 S 是基本均匀的时间计量系统，可以由平太阳时来求真太阳时。二者的差值就是时差，以 E_t 表示，即：

$$S_{\odot} = S + E_t \quad (3-6)$$

由于真太阳的周年运动不均匀，因此时差也随时都在变化着，时差的表达式为：

$$E_t = 0.0028 - 1.9857 \sin \theta + 9.9059 \sin 2\theta - 7.0924 \cos \theta - 0.6882 \cos 2\theta \quad (3-7)$$

3.3.3. 时角

天体的时角被定义为该天体的赤经与当地恒星时的差值。一个天体的时角表示该天体是否通过了当地的子午圈（中天）。其数值的物理含义为该天体与当地子午圈的角距离，以小时来计量：

$$\tau = (S_{\odot} + \frac{F_{\odot}}{60} - 12) \times 15^{\circ} \quad (3-8)$$

这里时 S_{\odot} 和分 F_{\odot} 表示的是真太阳时，我们从北京时间求出真太阳时，需要两个步骤：

Step1: 将北京时间换成地方时 S_d ：

$$S_d = S + \frac{\left\{ F - \left[120^{\circ} - \left(JD + \frac{JF}{60} \right) \right] \times 4 \right\}}{60} \quad (3-9)$$

其中 120° 是北京时间的标准经度，乘以 4 即将角度转化为时间，即每度相当于 4 分钟，除以 60 将分钟转化为小时。

Step2: 进行时差校正：

$$S_{\odot} = S_d + \frac{E_t}{60} \quad (3-10)$$

其中，时角是以太阳正午时刻为 0 点的，顺时针方向（下午）为正，反之为负。

3.4. 公式的误差修正

由于对公式存在以下问题：

- 1) 忽略了平年和闰年的差别，导致闰年影响的累计效应逐步增长；
 - 2) 我们所需要的数值，会因所在地点的地理经度以及具体时刻与表值有异而不同。
- 所以我们有必要进行如下 3 项订正：

3.4.1. 年度订正

由于一回归年的实际长度不是 365 日，而是 365.2422 日，但日历上只有整日，不可能有小数日，因此我们需要按照如下表格的方式进行日期修正

表 3-1. 年度误差订正的规则

年份	2000	2001	2002	2003	2004	1985
订正值	0	-0.2422	-0.4844	-0.7267	-0.9688	0.0312

假定我们选用的是 2000 年的表值，修正的方法为：2001 年使用时加上 -0.2（-0.2422）日的订正，2002 年改为 -0.51（-0.4844）日的订正，1983 年为 -0.7（-0.7266）日，1984 年为 -1.0（-0.9688）但此年为闰年，多了 1 日，实际订正应为（0.0312）日，1985 年为 0.0（-0.0312）日，依次类推。

3.4.2. 经度订正

即使我们查阅的是当年的天文年历，也需此项订正。在我国国土范围内：

表 3-2. 经度误差订正的规则

经度范围	≤90° E	90° E~128° E	≥128° E
订正值	-0.2	-0.3	-0.4

3.4.3. 时刻订正：

由于我国普遍采用的是北京时，与标准格林尼治时间差 8 小时，具体到我国情况下，订正规则为：

表 3-3. 北京时间的时刻误差订正的规则

时段	2:00—4:24	4:24—6:48	6:48—9:12	9:12—11:36
订正值	-0.2	-0.1	0	0.1
时段	11:36—14:00	14:00—16:24	16:24—18:48	18:48—21:12
订正值	0.2	0.3	0.4	0

3.5. 修正后的计算公式

太阳高度角的计算公式：

$$\sin h_{\odot} = \sin \delta \sin \varphi + \cos \delta \cos \varphi \cos \tau \quad (3-11)$$

其中 h_{\odot} 为修正之后的高度角。

3.6. 模型的求解

由于已知被测物的经纬度、长度以及测量时间，我们依据上述参数提炼出总的求解方程，并进行求解，公式为：

$$\left\{ \begin{aligned}
 N_0 &= 79.6764 + 0.2422 \times (\text{year} - 1985) - INT \left[\frac{(\text{year} - 1985)}{4} \right] \\
 t &= N - N_0 \\
 \theta &= \frac{2\pi t}{365.2422} \\
 \delta &= 0.3723 + 23.2567 \sin \theta + 0.1149 \sin 2\theta - 0.1712 \sin 3\theta - 0.758 \cos \theta \\
 &\quad + 0.3656 \cos 2\theta + 0.0201 \cos 3\theta \\
 E_t &= 0.0028 - 1.9857 \sin \theta + 9.9059 \sin 2\theta - 7.0924 \cos \theta - 0.6882 \cos 2\theta \\
 S_d &= S + \frac{\left\{ F - \left[120^\circ - \left(JD + \frac{JF}{60} \right) \right] \times 4 \right\}}{60} \\
 S_\odot &= S_d + \frac{E_t}{60} \\
 \tau &= \left(S_\odot + \frac{F_\odot}{60} - 12 \right) \times 15^\circ \\
 \sin h &= \sin \delta \sin \varphi + \cos \delta \cos \varphi \cos \tau
 \end{aligned} \right. \quad (3-12)$$

将北京地区各参数值代入得：影子长度随着时间的推移的变化如下图 3-3 所示：
（详见 q1_shadow_length.m）

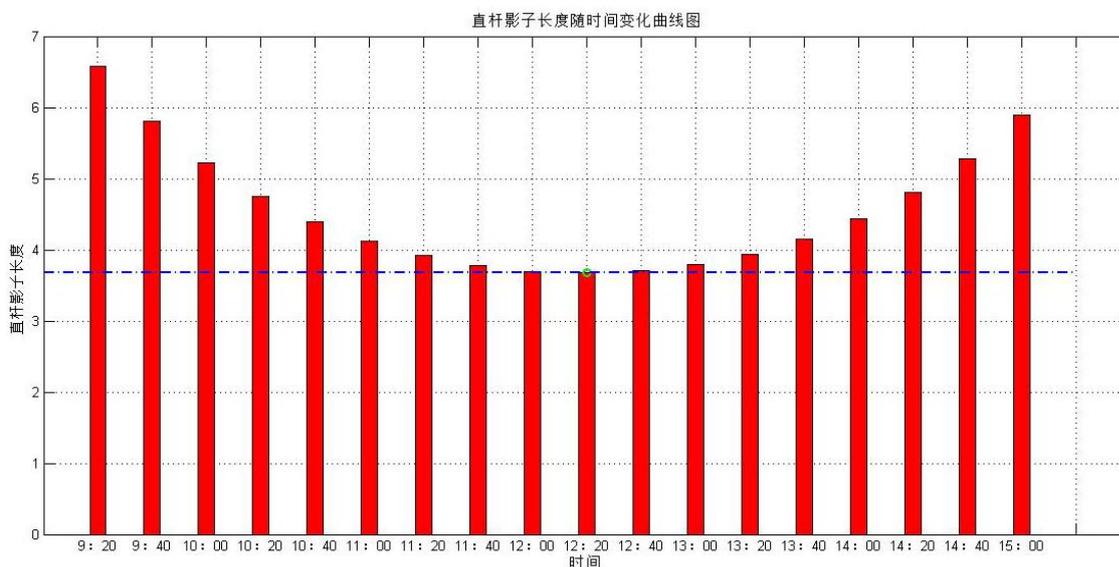


图 3-3. 直杆的影子长度随时间变化图示

由图 3-3 可知，直杆的影子长度经历了先减小后增大的过程。在 9 点到 12 点 20 分这段时间内，直杆的影子长度逐渐减小；在 12 点 20 分到 15 点这段时间内，直杆的影子长度逐渐增大。在 9 点时达到最大值 6.65m，12 点 20 分时达到最小值 3.67m，符合实际情况。

4. 问题二的求解

4.1. 问题的分析

该问要求根据某固定直杆在水平地面上的太阳影子顶点坐标数据，建立数学模型确定直杆所处的地点，并给出若干个可能的地点。

由于确定影子的长度需要至少四个参数：物体所处位置（经度，纬度）、直杆长度、以及当地的标准时间，但是该附件信息中只有直杆影子在直角坐标系上的投影（直角坐标系坐标轴的朝向未知）以及当地的年月日和每一时刻的时间，进行匹配之后只有直杆长度以及当地时间（时刻）可以作为有用变量，通过建立这些变量与经度，纬度，杆长的关系方程，就可以构成一个二元非线性方程组：其中时刻时钟数以及时刻分钟数作为自变量 x_1, x_2 ，影子长度作为方程的因变量 y ，经度，纬度，杆长作为方程系数 a, b, c ，我们将用 21 组实测数据去拟合出方程组的系数，使每一固定时刻杆长的实测值与计算值的差异性最小。

4.2. 数据预处理

首先对附件二 sheet 1 中的数据进行预处理，得到相应的观测值时刻的时钟数，分钟数以及影子长度，得到表 4-1.

表 4-1. 附件二 sheet1 数据预处理

组数	1	2	3	4	5	6	7
时	14	14	14	14	14	14	15
分	42	45	48	51	54	57	0
影子长度/m	1.149626	1.182199	1.215297	1.249051	1.283195	1.317993	1.353364
组数	8	9	10	11	12	13	14
时	15	15	15	15	15	15	15
分	3	6	9	12	15	18	21
影子长度/m	1.389387	1.426153	1.4634	1.501482	1.540232	1.579853	1.620145
组数	15	16	17	18	19	20	21
时	15	15	15	15	15	15	15
分	24	27	30	33	36	39	42
影子长度/m	1.661271	1.703291	1.746206	1.790051	1.835014	1.880875	1.927918

4.3. 二元非线性方程的构建

由公式 (3-13) 已知经纬度以及杆长与高度角的关系，但是由于在本问中将进行非线性拟合，采用常规拟合方法误差较大，很难定下初始值，因此我们将经分 JF 这个量剔除，使其以小数的形式出现在经度里面；又由于该问具体年月日已知，直接将其带入方程组，最后得到简化后计算的公式：

$$\begin{cases} S_d = S + \frac{\{F - [120^\circ - JD] \times 4\}}{60} \\ S_\odot = S_d + \frac{E_t}{60} \\ \tau = (S_\odot - 12) \times 15^\circ \\ \sin h = \sin \delta \sin \varphi + \cos \delta \cos \varphi \cos \tau \end{cases} \quad (4-1)$$

将已知量日期 2015 年 4 月 18 日 ($N=108$) 代入, 则其中的 S, E_t, δ 可通过上述公式求得具体值, 因此可由未知常量经度 JD , 纬度 φ 表示出高度角 h , 代入杆长度 $height$, 目标函数为:

$$\tan h = \frac{length}{y} \quad (4-2)$$

4.4. 通过 1stopt 软件进行非线性拟合

由于 Matlab 的非线性拟合函数 lsqcurvefit 等函数经度较差, 而且在本次求解过程总共有三个系数, 如果采用常规数学分析软件进行拟合需要设置较为良好的初值, 否则无法在短时间内甚至永远无法跳出局部最优解, 得到的结果不理想, 因此我们采用 **1stopt** (First Optimization) 软件进行建模:

采用方法: 麦夸特法(Lecenberg-Marduardt, LM)&通用全局最优算法(Universal Global Optimization)

拟合参数: 收敛判断指标: 1.00E-10, 最大迭代数 1000

(详见程序 q2_1stopt_nihe.mff)

4.5. 拟合结果及分析

拟合结果为: $a=108.5$, $b=0.337$, $c=2.033$,

即所在位置为: 东经 **108.5° E**, 北纬 **19.2° N**, 杆长 **2.033m**。

代入参数得到的拟合值与实测值关系如下表 4-2:

表 4-2. 拟合之后每组的实测值, 计算值及其差值

组数	1	2	3	4	5	6	7
实测值	1.149626	1.182199	1.215297	1.249051	1.283195	1.317993	1.353364
计算值	1.149645	1.182227	1.215334	1.248983	1.283194	1.317986	1.353381
差值	-1.9E-05	-2.8E-05	-3.7E-05	6.79E-05	1.54E-06	7.35E-06	-1.7E-05
组数	8	9	10	11	12	13	14
实测值	1.389387	1.426153	1.4634	1.501482	1.540232	1.579853	1.620145
计算值	1.389401	1.426072	1.463418	1.501467	1.540248	1.57979	1.620127
差值	-1.4E-05	8.08E-05	-1.8E-05	1.43E-05	-1.6E-05	6.3E-05	1.75E-05
组数	15	16	17	18	19	20	21
实测值	1.661271	1.703291	1.746206	1.790051	1.835014	1.880875	1.927918

计算值	1.661292	1.703321	1.746254	1.790129	1.834991	1.880884	1.927858
差值	-2.1E-05	-3.1E-05	-4.8E-05	-7.8E-05	2.38E-05	-9.1E-06	6.05E-05

我们将该关系通过 MATLAB 绘制出得图 4-1：其中蓝色为实测值，红色为计算值。
（程序详见 q2_guanxi.m）

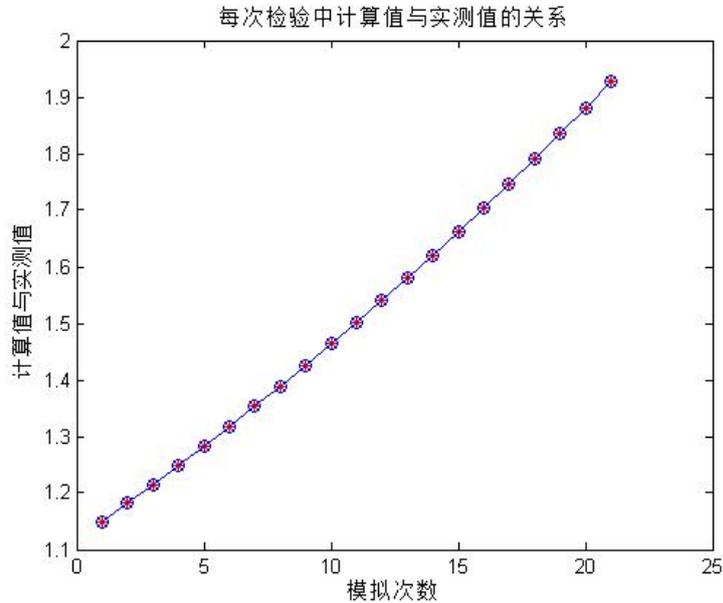


图 4-1. 每次测量中实测值与计算值区别

由上图可看出，用拟合出来的系数代入计算得到的 21 次计算值与实测值相差无几，说明该软件的非线性拟合速度非常快而且经度极高。

同时，绘制残差图得图 4-2：

（程序详见 q2_cancha.m）

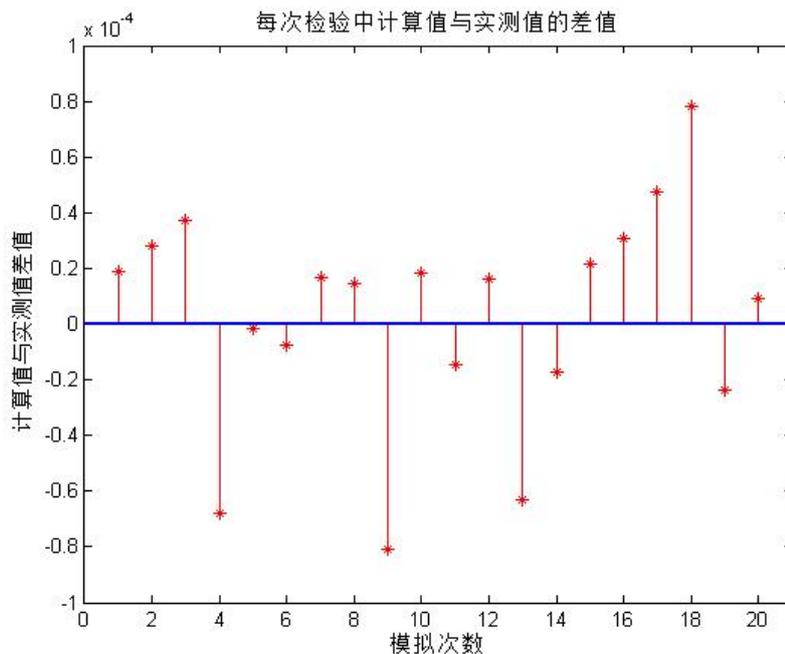


图 4-2. 每次检验中计算值与实测值的差值

通过做残差图：结合下表 4-2，我们可以得知，在 21 次实测中，残差均趋近于零而且 0 的上下均匀分布，说明此次拟合的效果极佳。

拟合的相关统计量见表 4-3：

表 4-3. 拟合结果的相关统计量

统计量	值
均方差(RMSE)	3.98220032704166E-5
残差平方和(SSE)	3.33016308338505E-8
相关系数(R)	0.999999985629292
卡方系数(CHI-SQUARE)	1.08793469910302E-8

4.6. 采用 Matlab 进行最小二乘法拟合

在上一次建模中，我们采用 1stopt 软件进行求解，为了证实该解的全局最优性质，我们采用 Matlab 进行了最小二乘法拟合，以拟合结果的影子长度计算值与实测值之差的平方和最小为优化拟合的目标。

题目已知直杆在水平地面上的太阳影子顶点坐标数据，因此可求解出给定时间下直杆的影子长度。针对该问题，直杆的长度为未知量，在此可将直杆所在的经度，纬度，以及直杆的长度作为决策变量，将其转化为一个带约束条件的非线性规划问题。

目标函数为实现计算得到的直杆影子长度和附件提供的影子长度的均方差最小，可描述为：

$$\min f(x) = \frac{1}{21} \sum_{i=1}^{21} (\text{length}_i - L_i)^2 \quad (4-3)$$

其中， L_i 为第 i 次测量时直杆的影子长度， length_i 为对应求解出的直杆的影子长度；其约束条件为：

1) 坐标经纬度的约束：

此处，关于坐标的经纬度，规定东经为正数，西经为负数；北纬为正数，南纬为负数；经纬度的取值满足如下约束条件：

$$\begin{cases} -180^\circ \leq JD \leq 180^\circ & \text{经度约束} \\ -90^\circ \leq WD \leq 90^\circ & \text{纬度约束} \end{cases} \quad (4-4)$$

2) 直杆长度约束：

直杆的长度必须为正数，且其不可能为无限长，为了减少计算工作量，加快求解速度，我们规定其长度不超过 5m，约束条件表示为：

$$0 < \text{len} \leq 5 \quad (4-5)$$

综上，该非线性规划模型可表示为：

$$\begin{aligned} \min f(x) &= \frac{1}{21} \sum_{i=1}^{21} (\text{length}_i - L_i)^2 \\ \text{s.t.} \quad &\begin{cases} -180^\circ \leq JD \leq 180^\circ \\ -90^\circ \leq WD \leq 90^\circ \\ 0 < \text{len} \leq 5 \end{cases} \end{aligned} \quad (4-6)$$

4.7. 最小二乘法拟合结果

上述模型是一个非线性组合优化模型，目标函数为非线性函数，在此处，我们分别采用穷举法和遗传算法来进行求解，之后可以通过对两种方法所得结果的分析比较，验证模型求解的准确性。

4.7.1. 使用穷举法对模型进行求解

由于数据量较大，直接求解将耗费大量的时间和资源，因此我们考虑先用穷举法对问题进行初步求解，先求解出一个局部较优解，结果如下表 4-3 所示：

(程序见附录 Solve_2_qiongju1.m)

表 4-3. 模型初求解结果

经度	纬度	直杆长度/m	均方差
105	21	2.1	1.14×10^{-4}

之后，再用穷举法遍历局部较优解邻域里的所有可能取值，选出其中均方差最小的数据作为全局最优解，结果如表 4-4 所示：

(程序见附录 Solve_2_qiongju2.m)

表 4-4. 模型最终求解结果

经度	纬度	直杆长度/m	均方差
109.1	19.2	2.0	1.14×10^{-4}

如表 2.2 所示，使用穷举法对模型进行求解之后，得到该直杆所在位置的经纬度坐标为(109.1°E,19.2°N)，对应的直杆长度为 2.0m，均方差为 $1.14 \times 10^{-4} m$ ，模型求解结果较好。

4.7.2. 使用遗传算法对模型进行求解

上述穷举法虽然能较为精确地求解出最优解，但其消耗的时间和资源过大，在数据量较大时求解较为困难。因此，我们又使用了一种快速的启发式算法——遗传算法来求解该问题，在多次运行程序之后，得到一组局部最优解如表 4-5 所示：

(程序见附录 Solve_2_GA.m)

表 4-5. 遗传算法求解的一组局部最优解

编号	经度	纬度	直杆长度/m	均方差
1	108.23	19.65	2.05	5.88×10^{-5}
2	111.53	0.95	1.75	5.15×10^{-4}
...
N-1	106.47	20.80	2.17	1.05×10^{-4}
N	110.59	0.97	1.81	4.78×10^{-4}

注：经度以东为正，以西为负；纬度以北为正，以南为负

观察得到的这组最优解，可以发现坐标的经纬度大致可以分为两类，因此我们针对这两类坐标点各自取 5 个纬度值，求解出平均值作为各自的最优解，如表 4-6 所示：

表 4-6. 选取的两类纬度值和各自的均值

类别	经度	纬度	直杆长度/m	均方差
1	111.53	0.95	1.75	5.15×10^{-4}
	110.59	0.97	1.81	4.78×10^{-4}
	110.94	0.57	1.78	4.76×10^{-4}
	109.91	0.23	1.84	4.06×10^{-4}
	110.22	0.22	1.82	4.25×10^{-4}
平均值	110.64	0.59	1.80	4.60×10^{-4}
类别	经度	纬度	直杆长度/m	均方差
2	108.23	19.65	2.05	5.88×10^{-5}
	109.13	18.86	2.00	1.05×10^{-5}
	108.69	19.29	2.03	3.83×10^{-5}
	108.59	19.19	2.03	4.29×10^{-5}
	108.92	19.26	2.06	6.07×10^{-5}
平均值	108.83	19.15	2.03	4.04×10^{-5}

注：经度以东为正，以西为负；纬度以北为正，以南为负

4.7.3. 通过 Excel Power Map 进行结果可视化

通过表 4-6 所得结果，我们将其在地球坐标系上绘制柱状图形针对地理位置用 -ExcelPowerMap 进行可视化描述。（图 4-4）以及（图 4-5）



图 4-4. 第一个可能点计算过程中的五次运行的经纬度结果展示

该程序第一个结果点的位置在**海南省西部**，柱状图的柱体高度代表以该经纬度带入计算得到的计算值杆长 $length$ 与实测值的方差，即杆长越长，方差越大（平均方差为 4.6×10^{-4} ）总的误差极小，可以接受。由该图可得，五个地理位置值相聚非常近，取平均值为 **108.83° E, 19.15° N**，具体处于海南省西部的昌江黎族自治县境内。

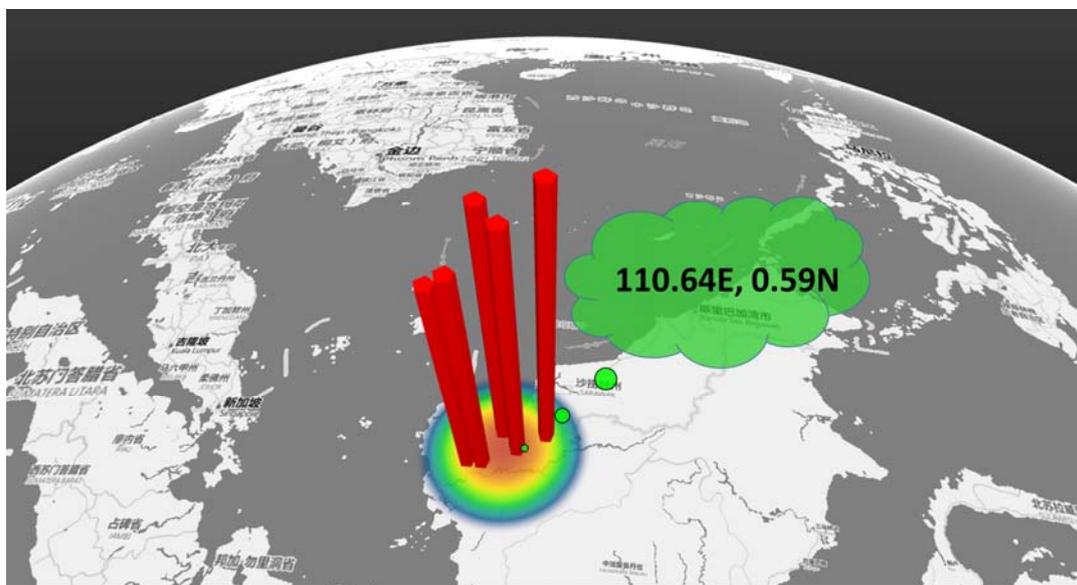


图 4-5.第二个可能点计算过程中的五次运行的经纬度结果展示

该程序第二个结果点的位置在**马来西亚西部**，柱状图的柱体高度代表以该经纬度带入计算得到的计算值杆长 $length$ 与实测值的方差，即杆长越长，方差越大（平均方差为 4.04×10^{-5} ）总的误差极小，可以接受。由该图可得，五个地理位置值相聚非常近，取平均值为 **110.64° E, 0.59° N**，具体处于**马来西亚西部**境内。

上图为两个最可能的地理位置点在地球上的立体图示，综上所述，用遗传算法求解得到直杆所处位置的可能取值点有两个：**海南省西部**的 **108.83° E, 19.15° N** 以及**马来西亚西部**的 **110.64° E, 0.59° N**，它们各自对应求解出的直杆影子长度的均方差为 $4.04 \times 10^{-5} m$ 和 $4.25 \times 10^{-4} m$ 。

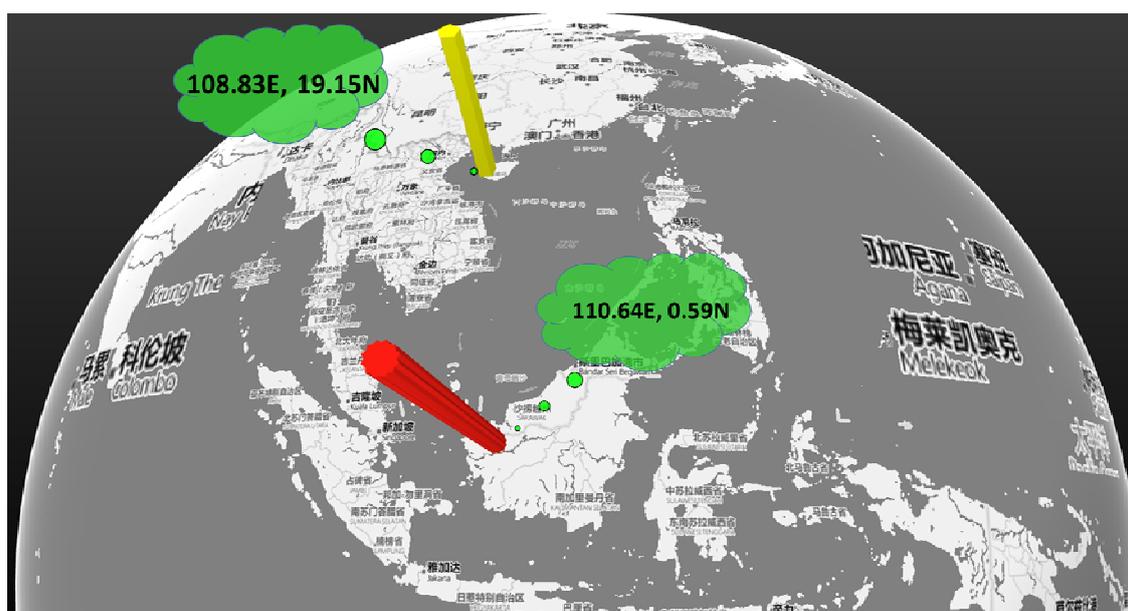


图 4-5. 两个可能点五次运行的平均经纬度结果展示

4.7.4. Matlab 两种算法的比较

通过比较两种不同方法求出的直杆纬度值可知，虽然采用的方法不同，但求出的值大致相同，且不同纬度值对应求解出的直杆影子长度与实际测量长度的均方差的值均较小，说明模型求出的解十分精确，具有可靠性与通用性。

5. 问题三的建模求解

5.1. 问题的分析

该问与第二问相类似，要求根据某固定直杆在水平地面上的太阳影子顶点坐标数据，建立数学模型确定直杆所处的地点，并给出若干个可能的地点。但是提供的数据量更少，只有影子的 x,y 坐标数据。我们采取的算法与上一问类似，首先通过建立这些变量与经度，纬度，杆长和当地日期（年月日）的关系方程，可以构成一个二元非线性方程组：其中时刻时钟数以及时刻分钟数作为自变量 x_1, x_2 ，影子长度作为方程的因变量 y ，经度，纬度，杆长以及年份和日期在的序号作为方程系数 a, b, c, d, f ，即决策变量，我们将用 21 组实测数据去拟合出方程组的系数，使每一固定时刻杆长的实测值与计算值的差异性最小。

5.2. 数据预处理：

对附件 2 和附件 3 中给出的数据进行预处理，求解出给定时间下所对应的直杆的影子长度，结果如表 5-1 和表 5-2 所示：

表 5-1. 附件 2 中直杆影子长度与时间的关系表

组数	1	2	3	4	5	6	7
时	12	12	12	12	12	12	12
分	41	44	47	50	53	56	59
影子长度/ M	1.247256	1.222795	1.198921	1.175429	1.15244	1.129917	1.107835
组数	8	9	10	11	12	13	14
时	13	13	13	13	13	13	13
分	59	5	8	11	14	17	20
影子长度/ M	1.086254	1.065081	1.044446	1.024264	1.00464	0.985491	0.96679
组数	15	16	17	18	19	20	21
时	13	13	13	13	13	13	13
分	23	26	29	32	35	38	41
影子长度/ M	0.948585	0.930928	0.913752	0.897109	0.880974	0.865492	0.850504

表 5-2. 附件 3 中直杆影子长度与时间的关系表

组数	1	2	3	4	5	6	7
时	12	12	12	12	12	12	12

分	41	44	47	50	53	56	59
影子长度/ <i>M</i>	3.533142	3.546768	3.561798	3.578101	3.595751	3.614934	3.635426
组数	8	9	10	11	12	13	14
时	13	13	13	13	13	13	13
分	59	5	8	11	14	17	20
影子长度/ <i>M</i>	3.657218	3.680541	3.705168	3.731278	3.758918	3.788088	3.818701
组数	15	16	17	18	19	20	21
时	13	13	13	13	13	13	13
分	23	26	29	32	35	38	41
影子长度/ <i>M</i>	3.85081	3.884585	3.919912	3.956876	3.995535	4.035751	4.077863

5.3. 非线性规划模型的建立与求解:

针对该问题, 在基于问题二的求解基础上, 再考虑时间这一变量, 把直杆所在的经度, 纬度, 直杆的长度, 年份和积年这五个变量作为决策变量, 将其转化为一个带约束条件的非线性规划问题。

目标函数为实现计算得到的直杆影子长度和附件提供的影子长度的均方差最小, 可描述为:

$$\min f(x) = \frac{1}{21} \sum_{i=1}^{21} (\text{length}_i - L_i)^2 \quad (5-1)$$

其中, L_i 为第 i 次测量时直杆的影子长度, length_i 为对应求解出的直杆的影子长度;

约束条件有:

(1) 坐标经纬度的约束:

此处, 关于坐标的经纬度, 规定东经为正数, 西经为负数; 北纬为正数, 南纬为负数; 经纬度的取值满足如下约束条件:

$$\begin{cases} -180^\circ \leq JD \leq 180^\circ & \text{经度约束} \\ -90^\circ \leq WD \leq 90^\circ & \text{纬度约束} \end{cases} \quad (5-2)$$

(2) 直杆长度约束:

直杆的长度必须为正数, 且其不可能为无限长, 为了减少计算工作量, 加快求解速度, 我们规定其长度不超过 $5m$, 约束条件表示为:

$$0 < \text{len} \leq 5 \quad (5-3)$$

(3) 时间约束:

测量时间所处年份的约束条件:

$$2000 \leq N \leq 2015 \quad (5-4)$$

为了简化求解, 不考虑闰年的影响, 将一年视为 365 天处理, 得到积日的约束条件为:

$$1 \leq n \leq 365 \quad (5-5)$$

综上，该非线性规划模型可表示为：

$$\begin{aligned} \min f(x) &= \frac{1}{21} \sum_{i=1}^{21} (\text{length}_i - L_i)^2 \\ \text{s.t.} \quad & \begin{cases} -180^\circ \leq JD \leq 180^\circ \\ -90^\circ \leq WD \leq 90^\circ \\ 0 < \text{len} \leq 5 \\ 2000 \leq N \leq 2015 \\ 1 \leq n \leq 365 \end{cases} \end{aligned} \quad (5-6)$$

根据上述模型，编程使用 MATLAB 自带的遗传算法工具箱对问题进行求解，设定种群规模为 80，进化代数数为 500 代，Stall Time 为 300 代（即最佳种群超过 300 代不进化则算法停止），使用 *gplotbestf* 函数绘制出遗传算法在进化过程中最佳个体适应度值和种群平均适应度值变化趋势图，如图 5-1 所示：

（程序见附录 Solve_3_GA.m）

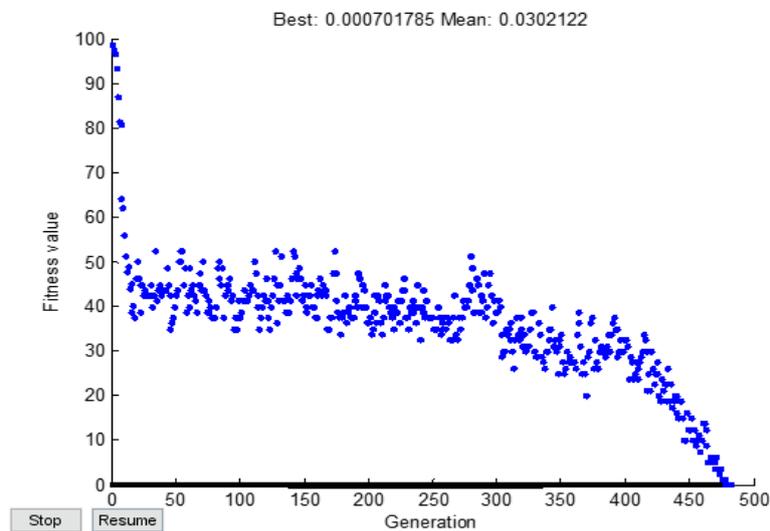


图 5-1. 适应度变化趋势图

如图 5-1 所示，种群在还没进化到 500 代就停止了，原因是最佳种群长时间不进化而使得算法自动停止运行。随着进化代数数的增加，种群平均适应度值（即种群的平均均方差）的值也相应地减小，最佳个体适应度值较小，模型求解结果良好。

重复上面的步骤，在多次运行程序之后，求解出一组局部最优解如表 5-3 所示：

表 5-3. 遗传算法求解的一组局部最优解

编号	经度	纬度	直杆长度/M	积日	均方差
1	86.21	40.70	2.50	177	6.26×10^{-4}
2	85.17	1.92	1.96	181	6.70×10^{-4}
...
N-1	85.13	40.93	2.39	180	4.39×10^{-4}
N	85.67	1.90	1.99	184	7.16×10^{-4}

注：经度以东为正，以西为负；纬度以北为正，以南为负

用 MATLAB 编程描出所有解的经纬度的值，如图 5-2 所示：
 (程序见附录 Solve_3_draw.m)

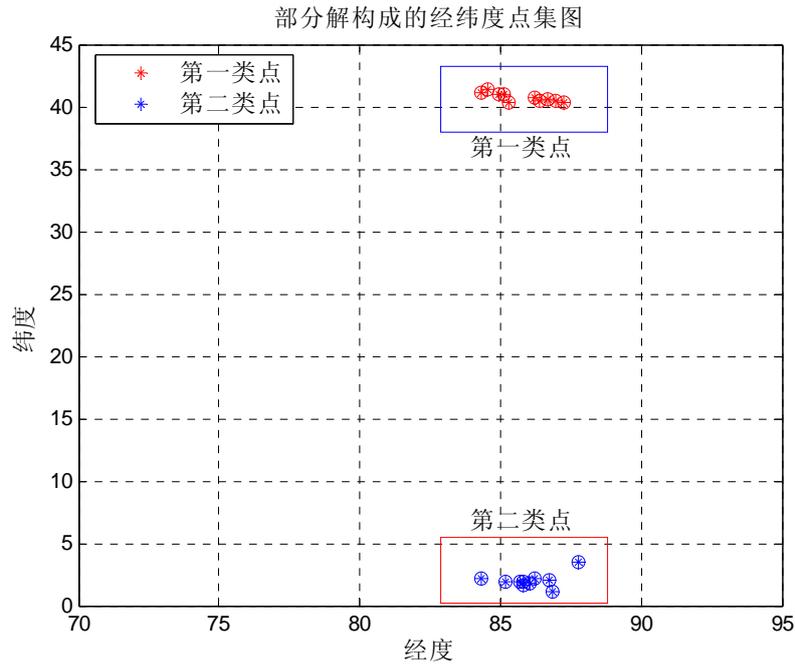


图 5-2. 部分解构成的经纬度点集图

观察得到的这一组解数据，可以看出虽然不同解对应的经纬度的取值不同，但如图 5-2 所示，这些由不同解的经纬度构成的点集大致可以分为两大类，且每一类别中的点的经纬度差异不大，因此我们考虑到这对应于直杆真实的位置坐标可能存在两个不同取值。因此我们针对这两类坐标点各自从这一组解数据中取出 5 个均方差最小的经纬度值，求解出它们的平均值作为各自的直杆位置坐标的经纬度值。结果如表 5-4 所示：

表 5-4. 附件二选取的两类经纬度值和各自的均值

类别	经度	纬度	直杆长度/m	积日	均方差
1	86.21	40.70	2.50	177	6.26×10^{-4}
	84.93	41.03	2.37	179	4.12×10^{-4}
	85.31	40.38	2.36	188	4.59×10^{-4}
	85.13	40.93	2.39	180	4.39×10^{-4}
	84.32	41.17	2.33	178	3.39×10^{-4}
平均值	85.18	40.84	2.39	180	4.55×10^{-4}
类别	经度	纬度	直杆长度/m	积日	均方差
2	85.17	1.92	1.96	181	4.55×10^{-4}
	85.67	1.90	1.99	184	6.70×10^{-4}
	84.33	2.18	2.06	185	7.16×10^{-4}
	86.02	1.84	2.01	186	8.22×10^{-4}
	86.21	2.15	2.04	183	7.50×10^{-4}
平均值	85.92	2.00	2.01	184	7.87×10^{-4}

注：经度以东为正，以西为负；纬度以北为正，以南为负

由上表可知，模型求解得到的直杆所处位置的可能取值点有两个，分别为(40.84°N,85.18°E)和(2.00°N,85.92°E)，时间为：6月30日以及7月4日。由于年份的影响因素太小，误差较大，无法求得，因此我们定为2015年。它们各自对应求解出的直杆影子长度的均方差为 4.55×10^{-4} 和 7.49×10^{-4} 。查阅地图之后发现，点(2.00°N,85.92°E)位于海面位置，不符合题目要求，应予以剔除，因此直杆所处位置的经纬度为(40.84°N,85.18°E)。

同样地，将附件三的数据代入上述模型进行求解，得到一组局部最优解如表5-5所示：

(程序见附录 Solve_3_GA1.m)

表 5-5. 遗传算法求解的一组局部最优解

编号	经度	纬度	直杆长度/m	均方差
1	112.33	35.13	2.18	7.86×10^{-5}
2	110.92	23.29	3.56	4.08×10^{-4}
...
N-1	111.96	36.51	2.04	1.18×10^{-4}
N	109.95	22.95	3.37	2.90×10^{-4}

注：经度以东为正，以西为负；纬度以北为正，以南为负

观察得到的这一组解数据，可以发现坐标的经纬度同样可以分为两大类，因此我们针对这两类坐标点各自从这一组解数据中取出5个均方差最小的经纬度值，求解出它们的平均值作为各自的直杆位置坐标的经纬度值，选取的数据和计算结果如表5-6所示：

表 5-6. 附件三选取的两类纬度值和各自的均值

类别	经度	纬度	直杆长度/M	积日	均方差
1	112.33	35.13	2.18	5.640436	7.86×10^{-5}
	111.96	36.51	2.04	1.847464	1.18×10^{-4}
	112.19	37.06	2.00	4.450944	1.21×10^{-4}
	112.22	36.28	2.08	3.927274	1.04×10^{-4}
	111.96	36.66	2.03	2.10311	1.19×10^{-4}
平均值	112.13	36.33	2.07	3.593846	1.08×10^{-4}
类别	经度	纬度	直杆长度/m	积日	均方差
2	110.92	23.29	3.56	15.53347	4.08×10^{-4}
	109.95	22.95	3.37	1.857441	2.90×10^{-4}
	110.11	21.53	3.67	10.19363	4.84×10^{-5}

	110.20	22.20	3.54	7.406303	3.95×10^{-4}
	109.42	21.69	3.68	3.462663	3.90×10^{-4}
平均值	110.12	22.33	3.56	7.690701	3.93×10^{-4}

注：经度以东为正，以西为负；纬度以北为正，以南为负

由上表可知，模型求解得到的直杆所处位置的可能取值点有两个，分别为(36.33°N,112.13°E)和(22.33°N,110.12°E)，时间分别为2015年1月7号和2015年1月4号。它们各自对应求解出的直杆影子长度的均方差为 1.08×10^{-4} 和 3.93×10^{-4} 。

5.4. 通过 Excel Power Map 进行结果可视化

通过表 5-4 所得结果，我们将其在地球坐标系上绘制图形进行地理位置用 EexcelPowerMap 进行可视化描述。图 5-3，5-4.

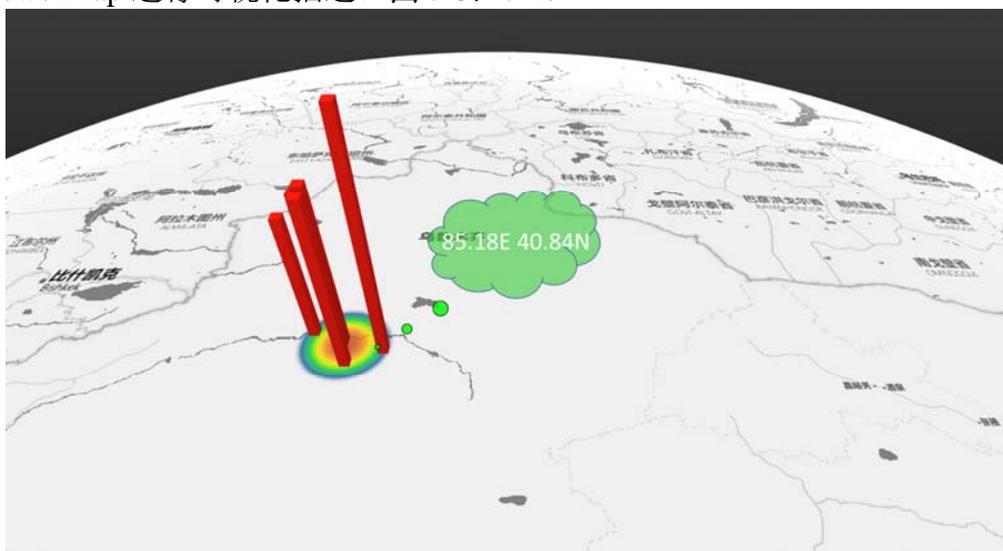


图 5-3 附件二的计算过程中的五次运行的经纬度结果及其均值展示

该程序第一个结果点的位置在我国新疆省西部，柱状图的柱体高度代表以该经纬度带入计算得到的计算值杆长 $length$ 与实测值的方差，即杆长越长，方差越大（平均方差为 4.55×10^{-4} ）总的误差极小，可以接受。由该图可得，五个地理位置值相距非常近，取平均值为 **85.18E, 40.84N**。

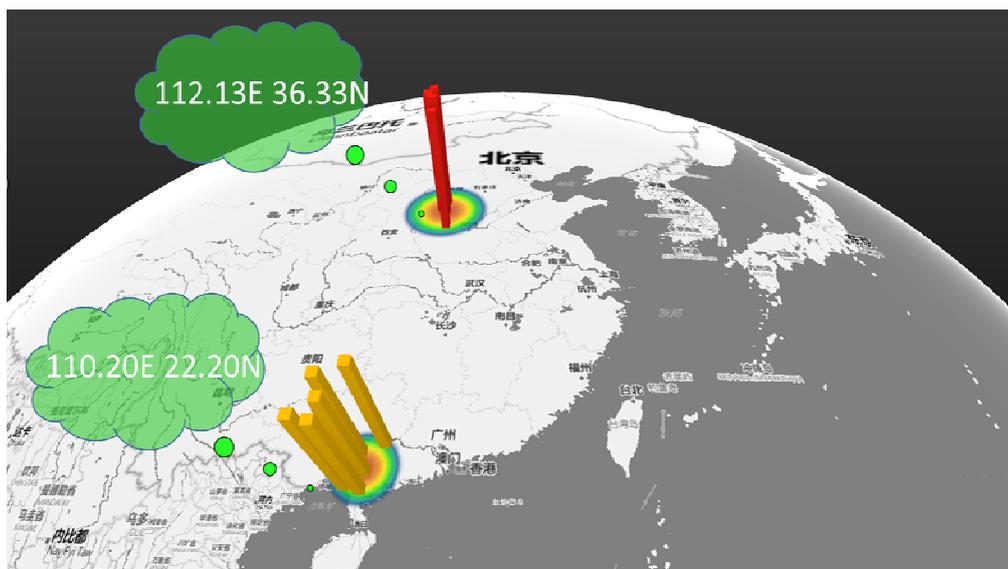


图 5-4. 附件三的计算过程中的五次运行的经纬度结果及其均值展示

该程序第一个结果点的位置在广东省西部，柱状图的柱体高度代表以该经纬度带入计算得到的计算值杆长 $length$ 与实测值的方差，即杆长越长，方差越大(平均方差为 43.95×10^{-4}) 总的误差极小，可以接受。由该图可得，五个地理位置值相距非常近，取平均值为 **110.12E, 22.33N**。

该附件第一个结果点的位置在河北省西南部，柱状图的柱体高度代表以该经纬度带入计算得到的计算值杆长 $length$ 与实测值的方差，即杆长越长，方差越大(平均方差为 1.08×10^{-4}) 总的误差极小，可以接受。由该图可得，五个地理位置值相距非常近，取平均值为 **112.13E, 36.33N**。

6. 问题四的求解

6.1. 问题的分析

在附件 4 中，我们仅仅拥有视频中的杆的长度信息以及在摄像机的相平面上的像素位置信息，为了找到拍摄地点的经纬度等位置信息，我们需要仅仅根据有限的视频资料求出类似第二问以及第三问中的数据：即在每一个时刻下，影子的长度（或者自定直角坐标系下的横纵坐标值）。

根据已有视频，我们需要采用射影几何以及坐标系转换原理将**单目静止摄像头的二维平面所成像进行三维空间的重构以及测距**：从视频中提取相应帧的照片，并进行分析，从而得到待测点在空间位置上的坐标。由于实际空间中的三维坐标在映射投影到摄像机成像的二维坐标系时，往往空间中的几何关系都会发生改变。因此我们需要找到某种方法，让摄像机的成像平面（像平面）坐标系与真实（路面）坐标系之间可以以某种方式任意转换，这样就能根据物体在视频中位置的变化，得到不同的绝对位置坐标值从而得到实际坐标系中物体运动的距离。

6.2. 视频预处理

通过视频处理软件，我们可以将视频截取成为一帧一帧的图像，并每间隔三分钟截取一次样本图片并利用上述方法进行测距，获取影子在地球坐标系中的长度。

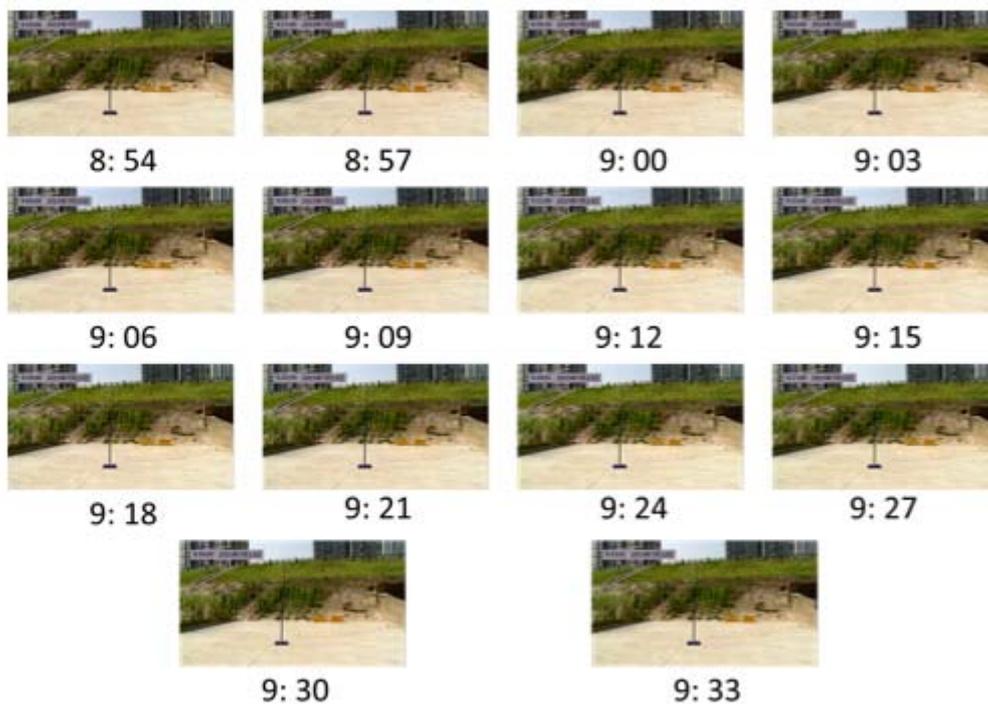


图 6-1. 在后续计算中所截取的视频帧

我们截取的样本帧图像如图 6-1:

6.3. 像平面坐标系与路面坐标系的转换

由上述帧画面可得，截图中图像除了发生摄影变换之外几乎没有发生镜头畸变，在成像过程中忽略因镜头扭曲造成的几何失真，因此可以用标准摄像机的小孔成像工作原理进行求解及其测距。

由于空间中的点的随意性，我们需要对摄像机形成像平面坐标系与实际空间中的坐标系进行转换，即：我们可以通过坐标系的矩阵变化由某一点在像平面坐标系中的位置确定该点在实际平面坐标系中的位置。像平面坐标系向地面坐标系的映射关系如图 6-2:

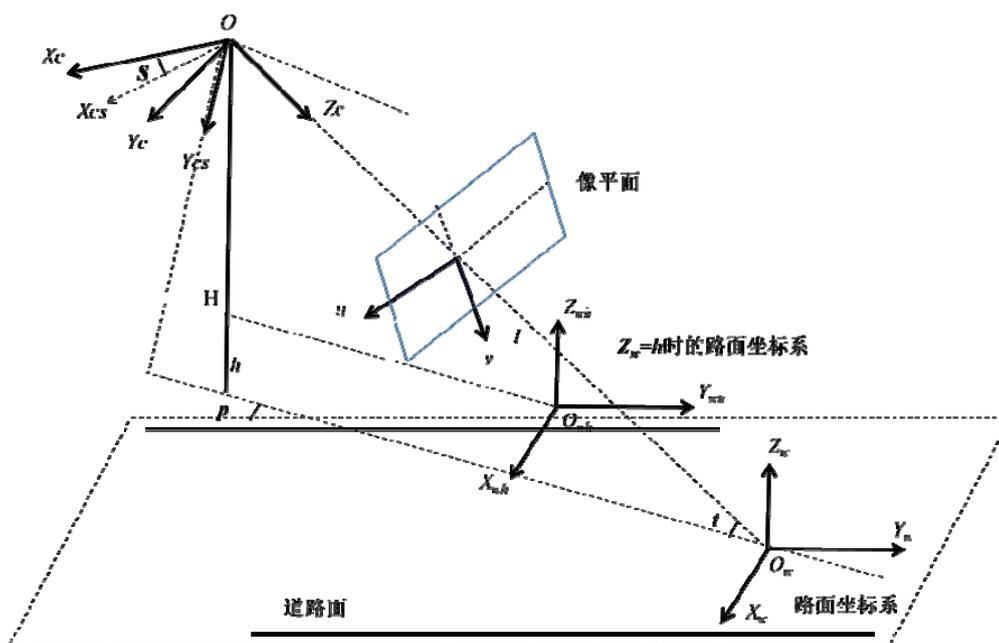


图 6-2. 像平面与地面坐标系的映射关系

图 6-2 中有三个角度关系，其中 p 是摄像机的光轴向地面投影以后与道路面之间的夹角， t 是摄像机的光轴与与地面之间的夹角， s 是摄像机内部的旋转角另外， H 为摄像机在支柱上的安装高度， l 是摄像机的相平面到路面坐标系的距离，这五个角度量都是未知量，将在后文中进行求解。

图中的四个坐标系分别为： $X_w Y_w Z_w$ ：路面坐标系； $X_c Y_c Z_c$ ：相机坐标系；

$X_{cs} Y_{cs} Z_{cs}$ ：相机坐标系统 Z_c 轴旋转 s 角度后得到的； $X_{wh} Y_{wh} Z_{wh}$ 是当 $Z_w = h$ 时的路面坐标系； uv 是相平面坐标系。在数学上，这四个坐标系之间有着相互关联的关系（图 6-2），经过一系列的变换，就能得到实际坐标系与相平面坐标系之间的关系。

这四个映射坐标系之间的数学关系为^[2]：

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{Z_c} \begin{bmatrix} f_x & 0 & u_0 & 0 \\ 0 & f_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \quad (6-1)$$

u 、 v 是像平面坐标系的坐标， X_c 、 Y_c 、 Z_c 为相机坐标系三个坐标值， f_x 、 f_y 分别为 u 轴和 v 轴上的归一化焦距， u_0 和 v_0 则表示的是光学中心，是摄像机光轴与图像平面的交点，通常位于图像中心处。其中， f_x 、 f_y 、 u_0 、 v_0 均是摄像机本身的参数，当摄像机的焦距大小为 f ， p_x 、 p_y 分别为摄像机画面的横向像素点个数和纵向像素点个数时，满足：

$$u_0 = \frac{1}{2} p_x, v_0 = \frac{1}{2} p_y \quad (6-2)$$

其中，为了便于化简，令：

$$[k \ 0] = \begin{bmatrix} f_x & 0 & u_0 & 0 \\ 0 & f_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (6-3)$$

$X_{cs} Y_{cs} Z_{cs}$ 可由相机坐标系统 Z_c 轴旋转 s 角度后得到，同时可得 $X_{cs} Y_{cs} Z_{cs}$ 倾斜坐标系与 uv 相平面坐标系之间的关系。

由于 $X_{cs} Y_{cs} Z_{cs}$ 倾斜坐标系可以看成是由 $X_w Y_w Z_w$ 路面坐标系经过一次 p 角度和 t 角度的旋转而得到的，并且它们之间的距离为 l ，则 $X_{cs} Y_{cs} Z_{cs}$ 倾斜坐标系和 $X_w Y_w Z_w$ 路面坐标系的关系为：

$$\begin{cases} X_{cs} = \cos(p) \times X_w - \sin(p) \times Y_w \\ Y_{cs} = -\sin(t) \sin(p) \times X_w - \sin(t) \cos(p) \times Y_w + \cos(t) \times Z_w \\ Z_{cs} = \cos(t) \sin(p) \times X_w - \cos(t) \cos(p) \times Y_w + \sin(t) \times Z_w + l \end{cases} \quad (6-4)$$

因此在公式(6-5)中加入这个变换即能得到 $X_w Y_w Z_w$ 路面坐标系与 uv 相平面坐标系之间的关系为

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{Z_c} \begin{bmatrix} k & 0 \end{bmatrix} \begin{bmatrix} -\cos(s) & \sin(s) & 0 & 0 \\ -\sin(s) & -\cos(s) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(p) & -\sin(p) & 0 & 0 \\ -\sin(t) \sin(p) & -\sin(t) \cos(p) & \cos(t) & 0 \\ \cos(t) \sin(p) & -\cos(t) \cos(p) & \sin(t) & l \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (6-5)$$

为了便于计算，我们将上述公式进行简化，即令：

$$\begin{aligned} M &= \begin{bmatrix} -\cos(s) & \sin(s) & 0 & 0 \\ -\sin(s) & -\cos(s) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(p) & -\sin(p) & 0 & 0 \\ -\sin(t) \sin(p) & -\sin(t) \cos(p) & \cos(t) & 0 \\ \cos(t) \sin(p) & -\cos(t) \cos(p) & \sin(t) & l \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \cos(p) \cos(s) - \sin(t) \sin(p) \sin(s) & -\sin(t) \cos(p) \sin(s) - \sin(p) \cos(s) & \cos(t) \sin(s) & 0 \\ \sin(t) \sin(p) \cos(s) + \cos(p) \sin(s) & \sin(p) \sin(s) + \sin(t) \cos(p) \cos(s) & -\cos(t) \cos(s) & 0 \\ -\cos(t) \sin(p) & \cos(t) \cos(p) & -\sin(t) & l \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (6-6)$$

综上所述，简化后的公式如下所示：

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{Z_c} \begin{bmatrix} k & 0 \end{bmatrix} M \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = \frac{1}{Z_c} \begin{bmatrix} k & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (6-7)$$

其中 k 、 R 、 T 的值分别为

$$k = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (6-8)$$

$$R = \begin{bmatrix} \cos(p) \cos(s) - \sin(t) \sin(p) \sin(s) & -\sin(t) \cos(p) \sin(s) - \sin(p) \cos(s) & \cos(t) \sin(s) \\ \sin(t) \sin(p) \cos(s) + \cos(p) \sin(s) & \sin(p) \sin(s) + \sin(t) \cos(p) \cos(s) & -\cos(t) \cos(s) \\ -\cos(t) \sin(p) & \cos(t) \cos(p) & -\sin(t) \end{bmatrix} \quad (6-9)$$

$$T = [0 \ 0 \ l] \quad (6-10)$$

在前面介绍过，由于矩阵 k 的数值只与摄像机的内部参数有关，因此矩阵 k 又被称为**内部矩阵**；矩阵 R 是通过两次坐标系变换，即经过 s 角度、 p 角度和 t 角度变换以后

得到的，因此矩阵 R 又被称为**旋转矩阵**；矩阵 T 是坐标系之间的平移， l 是摄像机的相平面平移到路面坐标系的距离，因此矩阵 T 又被称为**平移矩阵**。

对公式 (6-7) 进行反射射变换，就可以建立从像平面到路面坐标系($Z = 0$)的转换关系，具体公式如下^[3]：

$$X_w = \frac{l[u_s \cos(p) \sin(t) + v_s \sin(p)]}{v_s \cos(t) + f_x \sin(t)} \quad (6-11)$$

$$Y_w = \frac{l[-u_s \sin(p) \sin(t) + v_s \cos(p)]}{v_s \cos(t) + f_x \sin(t)} \quad (6-12)$$

其中， u_s 和 v_s 分别的值分别如下：

$$\begin{cases} u_s = (u - u_0) \cos(s) + (v - v_0) \sin(s) \\ v_s = (u - u_0) \sin(s) - (v - v_0) \cos(s) \end{cases} \quad (6-13)$$

经过上面对四个坐标系之间进行变换及部分主要公式的推导，就完成了摄像机的标定以及摄像机的相平面坐标系中的坐标点向实际路面坐标系的转换。但是由式 (6-7) 以及式 (6-12) 可知，要完成像平面到路面坐标系的反射射投影，需要标定出摄像机与路面的夹角 f 、偏角 p 、两坐标系之间的距离 l 这 3 个外部参数和像平面的中心 u_0, v_0 ，相机的内部偏角 s 、实际焦距 f 等 5 个内部参数。一言以蔽之，即需要求出公式 (6-7) 中的内部矩阵 K 、旋转矩阵 R 、以及平移向量 T 这三个矩阵以确定映射关系中的参数，完成两平面之间的转换。

6. 4. 基于消失点原理的实际空间坐标系建立方法

6. 4. 1. 消失点

有射影几何原理可知，当沿着教室墙壁或者公路两边的平行线向前看去时，两条平行线将不再平行，而是趋近于无限远处交于某一点，而该点在透视学中就被称为消失点，如同下图所示：

在原始的的地面空间坐标系中，直线 AB 与 CD 平行，但转为右图的人眼观看之后（相机坐标系）后 $A'B'$ 与 $C'D'$ 会相交于点 S ， S 则称为即消失点（灭点）。如图 6-2。

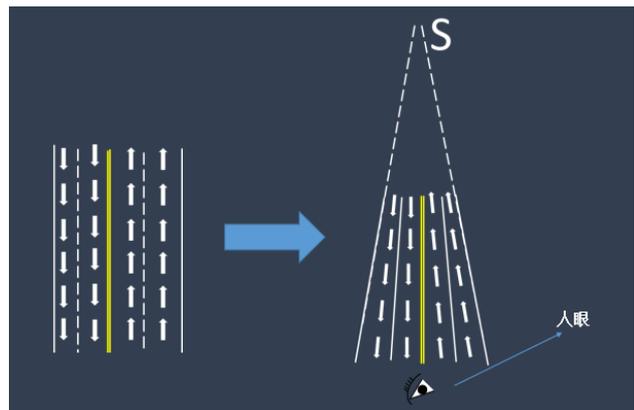


图 6-2. 射影几何原理

为了求解出 K, T, R 方程，也就是解出上述未知变量是 s, q, r, t ，我们应用消失点的部分属性来进行下述求解。

6. 4. 2. 建立像坐标系

建立像坐标系，标定所需点的像素坐标。

由于该处地面是由四块大的水泥砖构成，而根据水泥砖的矩形结构，我们很容易可

以得到在实际空间中两组互相垂直的平行线，如下图 6-4 所示：

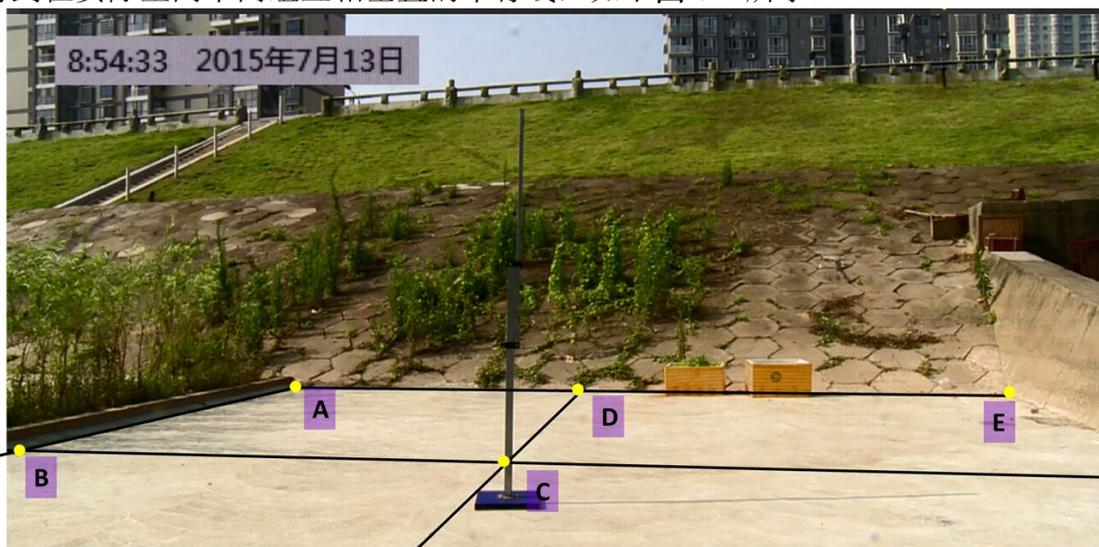


图 6-4.用于标定相坐标系坐标的两组平行线

其中在地面坐标系中满足 $AB \parallel CD$, $AD \parallel BC$, 在标定各点后，我们建立以左上角为坐标原点的二维平面相坐标系，其中横纵坐标的单位为像素值 (*pixel*)，而通过 Matlab 软件得到的各点的像素坐标为：

$$A(343,442) \quad B(28,515) \quad C(591,528) \quad D(676,446) \quad E(1175,451)$$

图片的分辨率为：1291ppi*636ppi(*pixel per inch*)。

而一般情况下的主点坐标 (u_0, v_0) 可近似为图像中心，即为 (645.5, 318)。

6.4.3. 求解相坐标系中的平行线在地面坐标系中的距离

由于我们已知的量只有长杆的高度，在同一平面上的相垂直的平行直线有①长杆的上下左右四条边，②地面上的大水泥地板两条边与江堤的底部直线，但是由于长杆四条边非常相近，如果用他们作为标称过程中的平行直线，会使得结果误差极大，因此我们采用地面上的大水泥地板两条边与江堤的底部直线构成的平面相交平行线进行标定。

直杆所处位置的具体三维重现如下图 6-5：

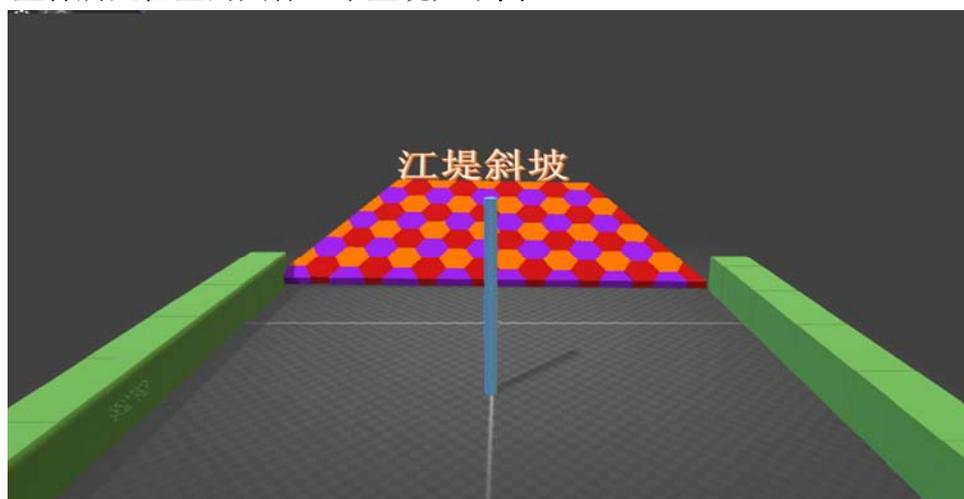


图 6-5. 直杆所处空间位置的三维重现

又因为这四条平行线中必须有一条线的一部分是真实长度已知的。由图 6-5 结合图 6-4 可发现，ADE 直线由江堤底部构成，上面均匀分布这江堤护坡上的六边形，通过查阅国家建造标准，得到在江河护坡以及水库等的堤坝护坡上常常采用的六边形是 C20 预制砼六角块^[4]，水泥板尺寸规格为：(图 6-6)

水平边长 300mm，即横向总长为 600mm。

由于 AE 段总共的护坡六角砖个数 (n) 可以通过视频清晰得出，即 $n=22.5$ ，又因为 AE 段长度 $L_{AE} = l \times n$ ，故我们得到 AE 段的长度为 6.75 米，因

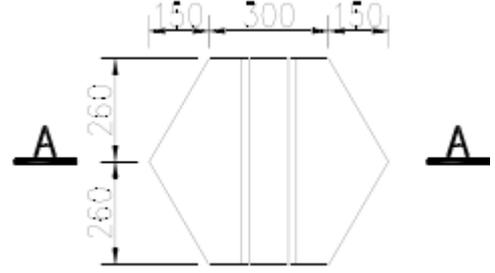


图 6-6. 砼六角块水泥板尺寸规格

此在以后的运算中，AE 的实际长度 L_{AE} 都作为已知量 6.75m 处理，同理， $L_{AD}=2.85m$ 。

6.4.4. 通过标定得到摄像机焦距

根据 A,B,C,D 的坐标，可以计算得到 AB,BC,CD,AD 的直线方程，进而可以得到 AD 与 BC 的交点（即水平向左的消失点）坐标为 $V_x(x_1, y_1) = (406.4303, 523.7382)$ ，以及 AB 与 CD 的交点（即垂直向上的消失点）坐标为 $V_y(x_2, y_2) = (463.4851, 414.0781)$

由射影定理的定义，在实际空间坐标系下，两两正交的三条直线投影到相机坐标系下组成的单位向量仍是两两正交的，故有三条直线 j, k, l 在像平面坐标系下对应的消失点的坐标为： $L(X_j, Y_j, -f)$ 、 $M(X_k, Y_k, -f)$ 、 $N(X_l, Y_l, -f)$ ，则有：

$$\begin{cases} X_j X_k + Y_j Y_k + f^2 = 0 \\ X_j X_l + Y_j Y_l + f^2 = 0 \\ X_k X_l + Y_k Y_l + f^2 = 0 \end{cases} \quad (6-14)$$

因为同处于一条摄像机光线上的点在像平面上的投影坐标都一样，即投影点的坐标变换与相机的安装距离无关系，只与安装的角度有关。若 \vec{u} 为消失点在路面坐标系下的坐标向量， \vec{u}' 为向量 \vec{u} 在相机坐标系下的坐标向量^[5]，则有：

$$\vec{u}' = R \times \vec{u} \quad (6-15)$$

当三条直线 l, j, k 为各个方向的单位坐标时，旋转矩阵可由以上 3 个消失点的坐标位置表示出来，则根据公式 (6-14) 在相机坐标系中 L, M, N 三个点的坐标及公式 (6-15) 的逆变换可得^[7]：

$$R = \begin{bmatrix} \frac{X_{cm}}{\sqrt{X_{cm}^2 + Y_{cm}^2 + f^2}} & \frac{X_{cl}}{\sqrt{X_{cl}^2 + Y_{cl}^2 + f^2}} & \frac{X_{cn}}{\sqrt{X_{cn}^2 + Y_{cn}^2 + f^2}} \\ \frac{Y_{cm}}{\sqrt{X_{cm}^2 + Y_{cm}^2 + f^2}} & \frac{Y_{cl}}{\sqrt{X_{cl}^2 + Y_{cl}^2 + f^2}} & \frac{Y_{cn}}{\sqrt{X_{cn}^2 + Y_{cn}^2 + f^2}} \\ -f & -f & -f \\ \frac{-f}{\sqrt{X_{cm}^2 + Y_{cm}^2 + f^2}} & \frac{-f}{\sqrt{X_{cl}^2 + Y_{cl}^2 + f^2}} & \frac{-f}{\sqrt{X_{cn}^2 + Y_{cn}^2 + f^2}} \end{bmatrix} \quad (6-16)$$

该等效于就是将 L 、 M 、 N 的三点的坐标进行单位化, 通过上述方法, 我们就能基本上确定了旋转 R 的计算方法。

6.4.5. 内部矩阵 K 的求解

由式 (6-14) 就能得到有效焦距 f 的值, 即为:

$$f = \begin{cases} \sqrt{-(X_{cl}X_{cm} + Y_{cl}Y_{cm})} \\ \sqrt{-(X_{cl}X_{cn} + Y_{cl}Y_{cn})} \\ \sqrt{-(X_{cm}X_{cn} + Y_{cm}Y_{cn})} \end{cases} \quad (6-17)$$

该公式说明只需要相机坐标系中的两个点就能计算出摄像机有效焦距 f 的值, 就能进行内部矩阵 K 的求解。

6.4.6. 平移矩阵 T 的求解

平移矩阵中, l 的值是相平面到实际路面坐标系的距离。消失点是实际路面坐标系中的两个平行直线映射到相平面以后所得到的点。

如图 6-7, O 为摄像机的光心, l_1 和 l_2 是空间坐标系下两条相互平行的直线, v 是直线 l_1 和 l_2 在像平面上的投影所形成的交点也就是消失点, 直线 Ov 是摄像机的光心 O 和消失点 v 的连线所组成的直线, 则直线 $Ov // l_1 // l_2$ 成立。

将上图通过俯视观察得到下图 6-8 所示, l 、 m 是实际路面坐标系中的两条直线, l 、 m 是直线上的两个点, L 、 M 是相平面上对应 l 、 m 的直线, L 、 M 为对应点, O 为摄像机的光心, s 为像平面中 L 、 M 直线的消失点, 它们的位置关系如下图所示。于是由射影几何的原理以及小孔成像就可以得到:

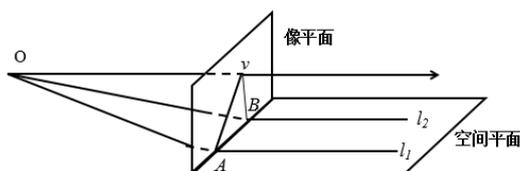


图 6-7. 消失点成像原理

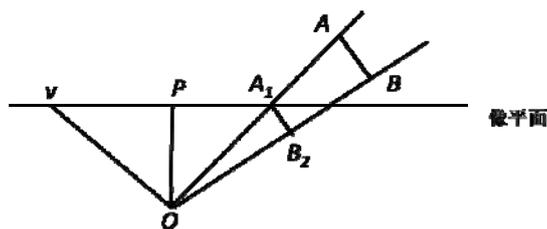


图 6-8. 平移向量求解原理图

$$\frac{|A_1B_2|}{|OV|} = \frac{|A_1B_1|}{|VB_1|} \Rightarrow |A_1B_2| = \frac{|A_1B_1| \cdot |OV|}{|VB_1|} \quad (6-18)$$

$$\frac{|OA|}{|OA_1|} = \frac{|AB|}{|A_1B_1|} \Rightarrow |OA| = \frac{|AB| \cdot |OA_1|}{|A_1B_1| \cdot |OV|} \cdot |VB_1|$$

由公式可得: 最后得到的得到的内部矩阵即为:

$$T = [0, 0, \frac{|AB| \cdot |OA_1|}{|A_1B_1| \cdot |OV|} \cdot |VB_1|] \quad (6-19)$$

6.5. 地面坐标系与像坐标系对应关系的建立结果

经过前面的大量推导之后, 我们已经可以利用已知的数据将单目摄像头二维像平面在地面坐标系上进行映射并得到最终的结果, 求解过程及如下:

已知标定点的坐标 $A(343, 442)$ $B(28, 515)$ $C(591, 528)$ $D(676, 446)$ $E(1175, 451)$

根据上述点的坐标，求出的 AB 、 BC 、 CD 、 DA 四条直线的方程分别为

$$\begin{cases} AB: y = -0.2317x + 521.4889 \\ BC: y = 0.1520x + 510.7440 \\ CD: y = -0.9797x + 1108.3 \\ AD: y = 0.0120x + 437.9 \end{cases} \quad (6-20)$$

通过公式 (6-20) 的求解及模型第三个消失点的求解方法，我们得到消失点的坐标分别如下表 6-1 所示：

表 6-1 两个消失点的坐标

项目	x 轴坐标	y 轴坐标
第一个消失点	-5.1176e+03	443.0688
第二个消失点	822.8118	378.8273

根据视频截取的画面可以得到： p_x 、 p_y 、 u_0 、 v_0 值如下表 6-2 所示

表 6-2. 视频画面的基本数值

项目名称	p_x	p_y	u_0	v_0
数值/像素	1366	768	683	384

在前面中，根据在相机坐标系中的三个消失点和相平面内对应三点的坐标关系，即有

将消失点带入即得 $f_x = f_y = f_1 = 158.5$ （像素焦距）。

根据上面所得到的数据，代入公式 (6-18)，(6-19)，(6-21) 可以得到内部矩阵 K 为：

$$k = \begin{bmatrix} 158.5 & 0 & 683.0 \\ 0 & 158.5 & 384.0 \\ 0 & 0 & 1.0 \end{bmatrix} \quad (6-21)$$

旋转矩阵 R 为

$$R = \begin{bmatrix} 0.6608 & -0.9996 & 0.0218 \\ -0.0244 & 0.0102 & 0.8836 \\ -0.7501 & -0.0273 & -0.4677 \end{bmatrix} \quad (6-22)$$

平移矩阵 T 为

$$T = \begin{bmatrix} 0 \\ 0 \\ 1.3559 \end{bmatrix} \quad (6-23)$$

相应的，根据在简化对 k 、 R 、 T 三个矩阵进行定义时，我们的目的是为了减少参数的复杂度，因此根据已经求得的矩阵，我们得到的最原始的三个不同坐标系的转换角度 s 、 p 、 t 的值如下表所示

图 6-3 三个坐标系转换角度大小

角度的名称	内部偏角 s	偏角 p	仰角 t
大小 (弧度)	-0.0247	1.6017	0.4867

最后我们通过转换关系输入像平面上影子在每一帧图像上的顶点坐标值并得到真实的长度值：具体结果如下表所示：
(详见程序 `Solve_dingbiao.m`)

表 6-4. 各时刻影子顶点坐标以及其真实长度

时间	顶点横坐标 (像坐标系)	顶点纵坐标 (像坐标系)	长度
8:54:00	1193	617	2.57778
8:57:00	1184	619	2.513147
9:00:00	1171	619	2.459785
9:03:00	1161	620	2.405689
9:06:00	1153	622	2.347886
9:09:00	1143	623	2.295156
9:12:00	1132	624	2.238864
9:15:00	1122	625	2.187108
9:18:00	1117	626	2.155938
9:21:00	1103	627	2.089022
9:24:00	1094	627	2.052847
9:27:00	1083	629	1.988746
9:30:00	1075	629	1.956764
9:33:00	1065	630	1.907389

6. 6. 结合影子长度得到的被测地经纬度 (已知时间)

由于与第三问方法完全相同，在此便不再复述，通过 Matlab 计算可得：
结果：**110.09E, 29.01N**
(详见程序 `Solve_4_GA.m`)

6. 7. 结合影子长度得到的被测地经纬度 (未知时间)

结果：**110.94E; 29.17N**
时间为：**2015 年 7 月 23 日**
(详见程序 `Solve_4_GA1.m`)

7. 模型的优化

7.1. 模型优点

- 1) 采用非线性拟合软件 1stopt 进行建模;
- 2) 非线性规划模型的求解过程中使用了遗传算法这一优化算法, 该算法具有良好的收敛性, 在保证了解模型求解精度的同时, 也大大减少了时间开支;
- 3) 问题一中在求解太阳高度角和方位角时, 对其中的参变量进行了修正, 使得模型的求解结果更加精准;
- 4) 使用摄像头内部参数标定对直杆的影子长度进行测量, 提高了结果的精准度;
- 5) 本模型仅需基于视频图像, 不必使用外界的相应参数, 尤其是摄像机的本身参数, 即可求解出视频的拍摄地点和拍摄日期, 对外界参数难以测定或无法测定时的情况, 有着很好的实用性。

7.2. 模型缺点

- 1) 非线性规划模型的求解过程基于传统的遗传算法, 该算法存在着“早熟”的缺点, 对新空间的探索能力不足, 容易收敛到局部最优解;
- 2) 在使用摄像机标定时, 因我们采用了一维标定的方法且视频数据较少, 无法找到与摄像头位于同一直线上的等距离标定物, 因而存在角度补偿问题, 得到的数据与实际有些许偏差。

8. 参考资料

- [1] 王炳忠.太阳辐射计算讲座第一讲太阳能中天文参数的计算[J].太阳能.1999(02)
- [2] 李焕, 基于视频的车速测量系统研究
- [3] 李勃, 董蓉, 陈启美.路况 PTZ 摄像机自动标定方法[J].北京邮电大学学报.2009(32) 24-29.
- [4] 张重德,张崇巍.一种提高视频车速检测精度的方法[J]. 上海交通大学学报. 2010(10)
- [4] 郭莉.预制砼六角块在大坝除险加固工程中的应用[J].中国西部科技.2013(06)12 卷第 06 期总第 287 期
- [5] B. Caprile,V. Torre.Using vanishing points for camera calibration[J]. International Journal of Computer Vision . 1990(2)
- [6] 张重德,张崇巍.一种提高视频车速检测精度的方法[J]. 上海交通大学学报. 2010(10)
- [7] 崔烂, 张国华. 一种基于单幅图像双消失点的摄像机标定方法[J].科学技术工程. 2012

9. 附录

9.1. 1stopt (First Optimization) 程序

程序名称	程序功能
q2_1stopt_nihe.mff	拟合出直杆最佳位置
Variables x, y, z; Parameters a[0], b[-1.58,1.58], c[0,5]; ConstStr mm=sin(b)*sin(0.1855)+cos(b)*cos(0.1855)*cos((x+(y-(120-a*100)*4)/60+0.5523/60-12)*0.2618); Function z=c/(mm/sqrt(1-mm^2)); Data; 14 42 1.149625826 14 45 1.182198976 14 48 1.215296955 14 51 1.249051052 14 54 1.28319534 14 57 1.317993149 15 0 1.353364049 15 3 1.389387091 15 6 1.426152856 15 9 1.463399853 15 12 1.501481622 15 15 1.540231817 15 18 1.579853316 15 21 1.620144515 15 24 1.661270613 15 27 1.703290633 15 30 1.74620591 15 33 1.790050915 15 36 1.835014272 15 39 1.880875001 15 42 1.927918447	

9.2. MATLAB 程序

程序名称	程序功能
q1_shadow_length.m	求出各个时刻的直杆影子长度
<pre>%q1_shadow_length.m %求解影子长度 clc; clear;</pre>	

```

JD=116;JF=23;JM=29;    %代入经度数据
WD=39;WF=54;WM=26;L=3; %代入纬度数据
S=9;F=0;fi=WD+WF/60+WM/3600;
N=295;    %N 为积年
N0=79.6764+0.2422*(2015-1985)-fix((2015-1985)/4);
t=N-N0;
theta=2*pi*t/365.2422;
Ed=0.3723+23.2567*sin(theta)+0.1149*sin(2*theta)-0.1712*sin(3*theta)-0.758*cos(theta)+0.
3656*cos(2*theta)+0.0201*cos(3*theta);% 太阳赤纬角
Et=0.0028-1.9857*sin(theta)+9.9059*sin(2*theta)-7.0924*cos(theta)-0.6882*cos(2*theta);
%求解时差
i=1;
for S=9:14
    for F=0:20:40
        Sd=S+(F-(120-(JD+JF/60+JM/3600))*4)/60;%求解地方时间
        SS=Sd+Et/60;
        t=(SS-12)*15;
        Fi=fi*2*pi/360;t=t*2*pi/360;
        ED=Ed*2*pi/360;%将角度转化为弧度
        h(i)=asind(sin(Fi)*sin(ED)+cos(Fi)*cos(ED)*cos(t)); %求解太阳高度角
        length(i)=L/tand(h(i));i=i+1;
    end
end
[min,ind]=min(length);
figure
bar(1:i-1,length,'r','barwidth',0.3);
grid on;
hold on;
plot(10,3.6744,'go','linewidth',2);
xlabel('时间');
ylabel('直杆影子长度');
title('直杆影子长度随时间变化曲线图');
line([0,19],[3.6744,3.6744],'LineStyle','-','LineWidth',2);
text(180-80,3.6744+0.15,'正午时刻, 太阳高度角最大');

```

程序名称	程序功能
q2_guanxi.m	绘制出 1stopt 拟合出的计算值与实测值的关系
<pre> %q2_guanxi.m x=1:21; figure y=[1.149625826 1.1496445 1.182198976 1.1822269 1.215296955 1.2153339 </pre>	

```

1.249051052    1.2489832
1.28319534    1.2831938
1.317993149    1.3179858
1.353364049    1.3533807
1.389387091    1.3894014
1.426152856    1.4260721
1.463399853    1.4634183
1.501481622    1.5014673
1.540231817    1.5402478
1.579853316    1.5797903
1.620144515    1.620127
1.661270613    1.661292
1.703290633    1.7033214
1.74620591     1.7462535
1.790050915    1.7901289
1.835014272    1.8349905
1.880875001    1.8808841
1.927918447    1.9278579
]
plot(x,y(:,1),'r*')
hold on
plot(x,y(:,2),'b-o')
xlabel('模拟次数');
ylabel('计算值与实测值');
title('每次检验中计算值与实测值的关系')

```

程序名称	程序功能
q2_cancha.m	绘制出 1stopt 拟合出的计算值与实测值的关系
<pre> %q2_cancha.m x=1:21; figure y=[1.149625826 1.1496445 1.182198976 1.1822269 1.215296955 1.2153339 1.249051052 1.2489832 1.28319534 1.2831938 1.317993149 1.3179858 1.353364049 1.3533807 1.389387091 1.3894014 1.426152856 1.4260721 1.463399853 1.4634183 1.501481622 1.5014673 1.540231817 1.5402478 </pre>	

```

1.579853316    1.5797903
1.620144515    1.620127
1.661270613    1.661292
1.703290633    1.7033214
1.74620591     1.7462535
1.790050915    1.7901289
1.835014272    1.8349905
1.880875001    1.8808841
1.927918447    1.9278579
]
stem(x,y(:,2)-y(:,1),'r*')
hold on
line([0,21],[0,0],'LineWidth',2);
axis([0 21 -0.0001 0.0001]);
xlabel('模拟次数');
ylabel('计算值与实测值差值');
title('每次检验中计算值与实测值的差值')

```

程序名称	程序功能
Solve_2_qiongju1.m	大范围穷举直杆可能地点
<pre> %q2_qingju_1.m clc; clear; load L.mat; % 已知时间 S,F, x, y 坐标 S=[14,14,14,14,14,14,15,15,15,15,15,15,15,15,15,15,15,15,15,15]; F=[42:3:57,0:3:42]; N=108;N0=79.6764+0.2422*(2015-1985)-fix((2015-1985)/4); t=N-N0; theta=2*pi*t/365.2422; Ed=0.3723+23.2567*sin(theta)+0.1149*sin(2*theta)-0.1712*sin(3*theta)-0.758*cos(theta)+0.3656*cos(2*theta)+0.0201*cos(3*theta);% 太阳赤纬角 Et=0.0028-1.9857*sin(theta)+9.9059*sin(2*theta)-7.0924*cos(theta)-0.6882*cos(2*theta); % 求解时差 k=1;E=10000; for len=0.1:0.1:4 for JD=50:180 for WD=0:90 for i=1:length(S) Sd=S(i)+(F(i)-(120-JD)*4)/60;% 求解地方时间 SS=Sd+Et/60; T=(SS-12)*15; h=asind(sind(WD)*sind(Ed)+cosd(WD)*cosd(Ed)*cosd(T)); % 求解太阳高度角 l(i)=len/tand(h); % 计算影子长度 end end end end </pre>	

```

        div(i)=(l(i)-L(i))^2;
    end
    D(k)=sqrt(sum(div));
    k=k+1;
end
end
end
[min,m]=min(D);
l_zhigan=fix(m/11921)/10;
de_J=fix((m-l_zhigan*10*11921)/91)+49;
de_W=m-l_zhigan*10*11921-91*(de_J-49)-1;

```

程序名称	程序功能
Solve_2_qiongju2.m	小范围穷举直杆可能地点
<pre> %q2_qingju_2.m clc; clear; load L.mat; % 已知时间 S,F, x, y 坐标 S=[14,14,14,14,14,14,14,15,15,15,15,15,15,15,15,15,15,15,15,15]; F=[42:3:57,0:3:42]; N=108;N0=79.6764+0.2422*(2015-1985)-fix((2015-1985)/4); t=N-N0; theta=2*pi*t/365.2422; Ed=0.3723+23.2567*sin(theta)+0.1149*sin(2*theta)-0.1712*sin(3*theta)-0.758*cos(theta)+0.3656*cos(2*theta)+0.0201*cos(3*theta);% 太阳赤纬角 Et=0.0028-1.9857*sin(theta)+9.9059*sin(2*theta)-7.0924*cos(theta)-0.6882*cos(2*theta); % 求解时差 k=1;E=10000; for len=1.8:0.1:2.4 for JD=100:0.1:110 for WD=18:0.1:24 for i=1:length(S) Sd=S(i)+(F(i)-(120-JD)*4)/60;% 求解地方时间 SS=Sd+Et/60; T=(SS-12)*15; h=asind(sind(WD)*sind(Ed)+cosd(WD)*cosd(Ed)*cosd(T)); % 求解太阳高度角 l(i)=len/tand(h); % 计算影子长度 div(i)=(l(i)-L(i))^2; end D(k)=sqrt(sum(div)); k=k+1; end end end </pre>	

```

end
end

%求解最佳的地点
[min,m]=min(D);
l_zhigan=fix(m/101/61)/10+1.8;
de_J=fix((m-(l_zhigan-1.8)*10*61*101)/61)*0.1+100;
de_W=(m-(l_zhigan-1.8)*10*101*61-61*(de_J-100)*10)*0.1+18;

```

程序名称	程序功能
Solve_2_GA.m	遗传算法求解
<pre> % Solve_2_GA.m clc; clear; options=gaoptimset('InitialPopulation',[2,110,20],'Generation',200,... 'PopulationSize',80,'StallTimeLimit',50,'StallGenLimit',80,'PlotFcns',@gaplotbestf); [x0,fval0,reason0,output,final_pop]=ga(@fitnesscon,3,options); options1=gaoptimset('InitialPopulation',final_pop,'Generation',100,... 'PopulationSize',80,'PlotFcns',@gaplotbestf); [x,fval,reason]=ga(@fitnesscon,3,options1); </pre>	

程序名称	程序功能
Solve_3_GA.m	遗传算法求解
<pre> % Solve_3_GA.m clc; clear; options=gaoptimset('InitialPopulation',[2,110,10,2006,180],'Generation',500,... 'PopulationSize',80,'StallTimeLimit',30,'StallGenLimit',500,'PlotFcns',@gaplotbestf); [x0,fval0,reason0,output,final_pop]=ga(@fitnessfuncon,5,options); options1=gaoptimset('InitialPopulation',final_pop,'Generation',500,... 'PopulationSize',80,'PlotFcns',@gaplotbestf); [x,fval,reason]=ga(@fitnessfuncon,5,options1); </pre>	

程序名称	程序功能
Solve_3_draw.m	绘制遗传算法结果
<pre> % Solve_3_draw.m clc; clear; JD=[86.67,87.27,86.37,84.56,86.96,86.21,84.93,85.31,85.13,84.32,85.17,85.67,84.33,86.02,8 6.21,86.87,85.80,87.75,86.75,85.80]; WD=[40.56,40.38,40.40,41.34,40.41,40.70,41.03,40.38,40.93,41.17,1.92,1.90,2.18,1.84,2.15, 1.12,1.69,3.49,2.04,1.87]; plot(JD(1:10),WD(1:10),'r*',JD(11:length(JD)),WD(11:length(JD)),'b*'); legend('第一类点','第二类点'); </pre>	

```

hold on;
plot(JD(1:10),WD(1:10),'ro',JD(11:length(JD)),WD(11:length(JD)),'bo');
grid on; axis([70,95,0,45]);
rectangle('position',[82.9,38,5.9,5.2],'edgecolor','b');
rectangle('position',[82.9,0.197,5.9,5.2],'edgecolor','r');
text(82.9+1,38.5-2,'第一类点');
text(82.9+1,0.197+6.5,'第二类点');
xlabel('经度');ylabel('纬度');
title('部分解构成的经纬度点集图');

```

程序名称	程序功能
Solve_3_GA1.m	遗传算法求解
<pre> %Solve_3_GA1.m clc; clear; options=gaoptimset('InitialPopulation',[3,110,40,2006,180],'Generation',500,... 'PopulationSize',80,'StallTimeLimit',300,'StallGenLimit',400,'PlotFcns',@gaplotbestf); [x0,fval0,reason0,output,final_pop]=ga(@fitnessfuncon1,5,options); options1=gaoptimset('InitialPopulation',final_pop,'Generation',500,... 'PopulationSize',80,'PlotFcns',@gaplotbestf); [x,fval,reason]=ga(@fitnessfuncon1,5,options1); </pre>	

程序名称	程序功能
Solve_dingbiao.m	相机参数的标定
<pre> %Solve_dingbiao.m clc; clear; %% 基础数据的录入 A=[355,487];B=[22,564];C=[617,570];D=[710,488];E=[710,488]; AE=3.0; G=[619,628]; mat=[1193 617 1184 619 1171 619 1161 620 1153 622 1143 623 1132 624 1122 625 1117 626 1103 627 1094 627 1083 629 </pre>	

```

1075    629
1065    630
];

%% 相关矩阵计算
px=1366;py=768;%像素比例
%dx=cx/px;dy=cy/py;
u0=px/2;v0=py/2;
%AD 和 BC 的消失点
%AD 直线方程: y-A(2)=(D(2)-A(2))/(D(1)-A(1))*(x-A(1)),
%BC 直线方程: y-B(2)=(C(2)-B(2))/(C(1)-B(1))*(x-B(1)),
%AD      与      BC      交      点      :
[x1,y1]=solve('y1-A(2)=(D(2)-A(2))/(D(1)-A(1))*(x1-A(1))','y1-B(2)=(C(2)-B(2))/(C(1)-B(1))*(x1-B(1))');
[x1,y1]=solve('y1-487=(494-487)/(1227-355)*(x1-355)','y1-564=(578-564)/(617-22)*(x1-22)');
);
x1=double(x1),y1=double(y1),
%AB 和 CD 的消失点
%AB 直线方程: y-B(2)=(A(2)-B(2))/(A(1)-B(1))*(x-B(1)),
%CD 直线方程: y-C(2)=(D(2)-C(2))/(D(1)-C(1))*(x-C(1)),
%AB      与      CD      交      点      :
[x2,y2]=solve('y2-B(2)=(A(2)-B(2))/(A(1)-B(1))*(x2-B(1))','y2-C(2)=(D(2)-C(2))/(D(1)-C(1))*(x2-C(1))');
[x2,y2]=solve('y2-564=(487-564)/(355-22)*(x2-22)','y2-578=(488-578)/(710-617)*(x2-617)');
);
x2=double(x2),y2=double(y2),
%像素焦距
f1=sqrt(-(x1-u0)*(x2-u0)-(y1-v0)*(y2-v0)),
%内部矩阵
K=[f1 0 u0;0 f1 v0;0 0 1],
%求解第三个消失点
%x3,y3      满      足      (x1-u0)*(x3-u0)+(y1-v0)*(y3-v0)+f1^2=0      ,
(x2-u0)*(x3-u0)+(y2-v0)*(y3-v0)+f1^2=0 方程
%[x3,y3]=solve('(x1-u0)*(x3-u0)+(y1-v0)*(y3-v0)+f1^2=0','(x2-u0)*(x3-u0)+(y2-v0)*(y3-v0)+f1^2=0'),
f=158.68;
M=[x1-u0,y1-v0;x2-u0,y2-v0];
N=[(x1-u0)*u0+(y1-v0)*v0-f*f;(x2-u0)*u0+(y1-v0)*v0-f*f];
M=inv(M)*N,
x3=M(1,1),
y3=M(2,1),
V1=sqrt((x1-u0)^2+(y1-v0)^2+f^2);V2=sqrt((x2-u0)^2+(y2-v0)^2+f^2);V3=sqrt((x3-u0)^2+(y3-v0)^2+f^2);
v1=[(x1-u0)/V1 (y1-v0)/V1 -f/V1]';v2=[(x2-u0)/V2 (y2-v0)/V2 -f/V2]';v3=[(x3-u0)/V3

```

```

(y3-v0)/V3 -f/V3]';
R=[v2 v1 v3],
%平移向量
oa=sqrt(f^2+(A(1)-u0)^2+(A(2)-v0)^2);
ae=sqrt((A(1)-E(1))^2+(A(2)-E(2))^2);
ov=sqrt(f^2+(x2-u0)^2+(y2-u0)^2);
ve=sqrt((E(1)-x2)^2+(E(2)-y2)^2);
l=AE*oa*ve/(ae*ov);
T=[0 0 l]',

%% 参数求解
%像素焦距
t=asin(-R(3,3)),%仰角
p=acos(R(3,2)/cos(t)),%偏角
s=asin(-R(1,3)/cos(t)),%内部偏角
%坐标映射
%fx=f;
%fy=f;
%% 路面坐标系建立结果
for i=1:14
%修正后相机坐标
H=mat(i,:);
us1=(G(1)-u0)*cos(s)+(G(2)-v0)*sin(s);
vs1=(G(1)-u0)*sin(s)-(G(2)-v0)*cos(s);
us2=(H(1)-u0)*cos(s)+(H(2)-v0)*sin(s);
vs2=(H(1)-u0)*sin(s)-(H(2)-v0)*cos(s);
%实际距离
X1=l*(us1*cos(p)*sin(t)+vs1*sin(p))/(vs1*cos(t)+f*sin(t)),
Y1=l*(-us1*sin(p)*sin(t)+vs1*cos(p))/(vs1*cos(t)+f*sin(t)),
X2=l*(us2*cos(p)*sin(t)+vs2*sin(p))/(vs2*cos(t)+f*sin(t)),
Y2=l*(-us2*sin(p)*sin(t)+vs2*cos(p))/(vs2*cos(t)+f*sin(t)),
d(i)=sqrt((X2-X1)^2+(Y2-Y1)^2),
end

```

程序名称	程序功能
Solve_4_GA.m	遗传算法求解
<pre> %Solve_4_GA.m clc; clear; options=gaoptimset('InitialPopulation',[110,50],'Generation',200,... 'PopulationSize',70,'StallTimeLimit',30,'StallGenLimit',80,'PlotFcns',@gaplotbestf); [x0,fval0,reason0,output,final_pop]=ga(@fitnessfuncon4,2,options); options1=gaoptimset('InitialPopulation',final_pop,'Generation',500,... 'PopulationSize',80,'PlotFcns',@gaplotbestf); </pre>	

```
[x,fval,reason]=ga(@fitnessfuncon4,2,options1);
```

程序名称	程序功能
Solve_4_GA1.m	遗传算法求解
<pre>%Solve_4_GA1.m clc; clear; options=gaoptimset('InitialPopulation',[110,40,2016,200],'Generation',500,... 'PopulationSize',80,'StallTimeLimit',30,'StallGenLimit',80,'PlotFcns',@gaplotbestf); [x0,fval0,reason0,output,final_pop]=ga(@fitnessfuncon4_1,4,options); options1=gaoptimset('InitialPopulation',final_pop,'Generation',500,... 'PopulationSize',80,'PlotFcns',@gaplotbestf); [x,fval,reason]=ga(@fitnessfuncon4_1,4,options1);</pre>	