

基于三角函数拟合与遗传算法的太阳影子定位分析

摘 要

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。依据影子长度进行地理定位的精度提高以及提升对视频中进行图像处理的精度成为太阳影子定位技术的两大核心任务。

针对问题一,本文基于地理、天文学知识给出了计算影子长度的数学模型。经分析,太阳影子长度受日期、经纬度、杆长、北京时间的影响。其中,年份对太阳影子长度的影响可以忽略;积日对影长的影响呈夏短冬长(南半球则相反);相等的影长在经纬度上呈环状分布,并且纬度越高,影子越长。随着纬度的增加,影长增加的越来越快;最后求出的影子长度变化曲线在日出和日落时均存在影长突变的情况。

针对问题二,基于问题一的影长函数结构,本文自定义了一个能够高度近似原影长函数的三角拟合函数,通过拟合求解出当地经度为 109.6575° 并由拟合出的系数给出了杆长与纬度的预测值 2 米、 22.5° ,相当于预测了一个地理位置。然后基于最小二乘法给出了适应度函数并使用遗传算法进行求解。遗传算法搜索出两个较优解 18.41° , -0.13° 。可见其中一个与预测效果相差不大。附件一提供的数据为 4 月 18 日,太阳直射北半球。可见搜索出的两个最优解基本关于直射点对称。故而推测最可能的地点为:海南岛三亚和西加里曼丹。

针对问题三,相当于在问题二的基础上增加了积日这个变量。为了保证自定义的拟合函数拟合效果,针对积日的变动对拟合函数进行了改进并求解出经度为 79.6857° 。随后采用与问题二相同的方法求解。对于附件二,遗传算法搜索结果中纬度为 39.5341° 、杆长为 2.0403 米、积日为 203 天的适应度函数最低且与预测值 39.8433° 、1.99099 米、141 天十分相近。故而推测最可能位于新疆西部地区,时属 4 月份或 7 月份;同理可推知附件三中最可能位于湖南张家界地区,时属 10 月份。

针对问题四,我们先对视频按一分钟每帧截取了 41 张图片,通过透视变换,将截取的原始图片的影子投影变换为光线垂直于地面的正投影。然后对变换后图片进行一系列处理,通过引入水平辅助线,计算出真实影子长度。结合问题二、三,有日期时视频拍摄地的可能地点位于桂林而没有日期时视频拍摄地的可能位于陕西并时属 6、7 月份。最后对模型求解过程可能存在的误差以及对经度变化对求解结果的影响进行了分析。

关键词: 太阳影子定位 三角函数拟合 遗传算法 透视变换

一、 问题重述

1.1 问题背景

根据美国能源信息署 EIA 发布的国际能源展望,世界能源市场消耗量 2005 年到 2030 年预计增加 50%,CO₂ 的排放量将由 281 亿吨增至 423 亿吨,而 CO₂ 的排放中,有四分之一来自于汽车尾气。目前,世界各国的交通工具主要以燃油为主,且燃油汽车数量还在不断的快速增长之中,燃油汽车在消耗大量能源的前提下,还造成了严重的环境问题。

新能源汽车的研发与应用对日益紧张的能源形势和环境问题的巨大缓解作用已经引起了世界多数国家的重视,被认为是 21 世纪汽车工业改造和发展的主要方向。2015 年 5 月,我国在国务院正式印发的《中国制造 2025》中,明确定义节能与新能源汽车为 10 大重点领域之一。工信部部长苗圩在接受央视专访时透露,国家确定了节能与新能源汽车 2019 年要占到 8%,2020 年要占到 10% 的奋斗目标,取消燃油车的时间表正在研究之中。荷兰、挪威、法国等欧洲国家更是已经发布了燃油汽车禁售时间表。

电动汽车作为新能源汽车的代表,近年来发展迅速,据 EV Sales 公布的数据显示,2017 年全年全球电动车销量超过 1,223,000 辆,对比 2016 年增长 58%,进而促使全球电动汽车销量在全球汽车销量当中的占比超过 1%。作为电动汽车重要的服务设施,充电站的位置、类型、数量等对电动汽车未来的发展起着至关重要的作用,对消费者购买电动汽车意愿有着重大影响。

1.2 问题提出

假设我国的内部、外部政策环境、经济增长速度、主要基础设施等保持不变,充电站仍为两种类型,即快速充电类型(超级充电或直流充电)和普通充电类型(目的地充电或交流充电),完成下述问题:

1. 建立数学模型,探讨电动车数量与充电站数目、类型以及位置之间的关系规律。
2. 应用你所建立的数学模型针对某个国家或某个地区进行计算,通过与该地区实际的充电站数目、类型以及位置进行对比,对模型进行验证、评价。(提示:充电站相关数据可以参考 tesla 公司网站,国内大城市,例如北京,亦有充电站位置分布网站或(APP))
3. 以陕西省为例,假设至 2050 年,陕西省内所有出租车和私家车全部转换为电动车,试计算所需的充电站数量、类型,位置,以及为建设这些充电站所投入的资金数量。

二、 问题分析

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,而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。影子长度的变化需要以地理学、天文学为基础进行推导。当影子长度确定后,要实现地理定位便需要进行搜索。于此同时,对已知数据的拟合效果直接决定了搜索的效果

2.1 问题一的分析

问题一关键在与建立地理、天文学变量之间的关系。但是对一些天文学变量往往很难计算其准确值。因此可以选取一些误差非常小的近似公式作为代替推导出影子长度与经纬度、积日、杆长、时间之间的关系。最后进行单一变量分析并给出指定条件下的影子长度变化曲线。

2.2 问题二的分析

问题二关键在于给出可能性高的衡量标准。为了提高搜索的精度,可以先用拟合的方法求出经度。由于问题一中给出了函数结构,故可以自定义一个拟合函数,使其不仅能求出经度,也能提供纬度、杆长的预测值。然后可以基于最小二乘法的思想,利用遗传算法中每一代的变量值计算出附件一中每一时刻对应的函数值,并与提供的数值作差并平方,衡量每一个样本点的误差。将 21 个样本的误差和作为适应度函数进行搜索求解即可。

2.3 问题三的分析

问题三是在问题二的基础上增加了积日这一个变量,但是却给搜索带来了精度上的挑战。为使得拟合函数能够近似表征原有函数,需要对问题二中的拟合函数进行修改。进而采用三个变量的遗传算法对问题三进行求解。

2.4 问题四的分析

问题四的关键在于如何将视频中影子的中心投影转换为平行投影,从而计算出影子的实际长度。所以首先需要对视频图像进行截取,然后对图像进行透视变换,转化为平行投影之后,经过对图像的一系列处理,采用添加辅助线的方式,得出影子的实际长度。再结合问题二与问题三模型,分别给出有无日期情况下的可能求解结果。

三、 模型假设

1. 假设地球是一个正球体
2. 假设太阳光为平行光且大气对太阳光的折射可以忽略
3. 假设地面可以看成一个水平面且被测直杆垂直于地面

四、 符号说明

符号	意义
L	直杆长度
l	影子长度
δ	太阳赤纬角
α	太阳高度角
ψ	纬度
ϕ	经度
ω	时角
N	积日
t	当地北京时间
T_c	地方时

五、 模型的建立与求解

5.1 问题一影子长度变化模型的建立与分析

5.1.1 影子长度变化模型的建立

基于地理学与天文学知识,本文给出了以经纬度、时间、杆长为自变量的影子长度变化的数学模型。

在已知杆长 L 的前提下,只需要一个太阳高度角 α 便可以求出影子长度。具体物理情形如图 1。太阳高度角是指太阳光的入射方向和地平面之间的夹角,专业上讲太阳高度角是指某地太阳光线与通过该地与地心相连的地表切面的夹角。

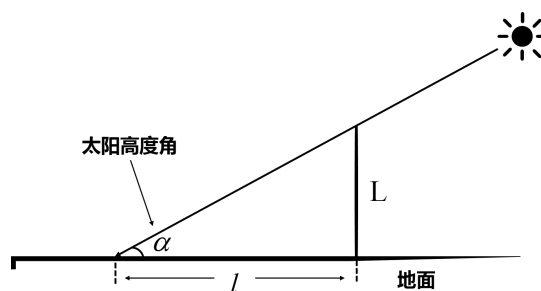


图 1: 太阳高度角

由图 1 易知有

$$\tan\alpha = \frac{L}{l} \quad (1)$$

其中 L 为杆长, l 为影子长度, α 为太阳高度角。查阅相关资料, 给出以下太阳高度角计算方法:

设 ψ 为纬度, ϕ 为经度, ω 为时角, δ 为太阳赤纬角。其几何意义如图 2 所示 (ω_0 表示日出时角)。

$$\sin\alpha = \sin\psi\sin\delta + \cos\omega\cos\psi\cos\delta \quad (2)$$

利用球面三角函数公式, 可以得到

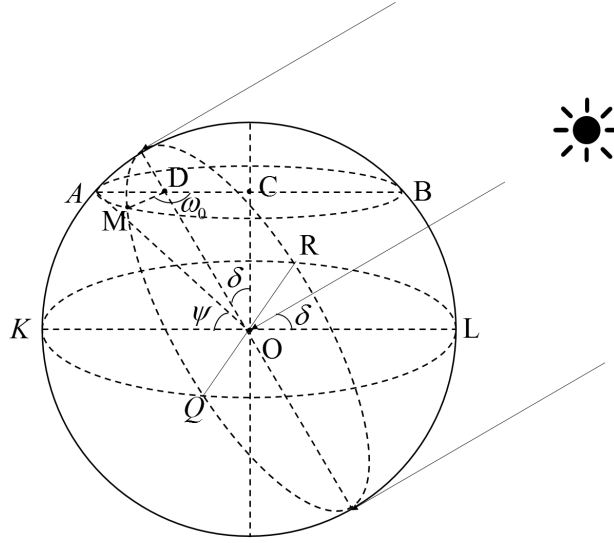


图 2: 太阳相关角度

下面给出太阳赤纬角 δ 和太阳时角 ω 的计算方法:

1. 太阳赤纬角亦称黄赤交角 (又名: 黄道交角) 是地球公转轨道面 (黄道面) 与赤道面 (天赤道面) 的交角, 也称为黄赤大距。目前地球的太阳赤纬角为约为 $23^\circ 26'$ 。查阅资料^[5] 知其计算公式满足以下公式:

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

$$\theta = \frac{2\pi(N - N_0)}{365.2422} \quad (4)$$

$$N_0 = 79.6764 + 0.2422 \times (\text{year} - 1985) - \left\lfloor \frac{\text{year} - 1985}{4} \right\rfloor \quad (5)$$

其中, θ 为日角 (单位: 度), 用于衡量地球的公转过程。 N 为积日, 其值等于测试影长的日期到 1 月 1 日的时间, 平时时积日的最大值为 365 天, 闰年最大值为 366 天。 ΔN 表示积日修正日, N_0 实际上是对平年、闰年的一种修正; year 为年份, 问题一中为 2015 年, 从而 $N_0 = 79.9424$; $\lfloor \cdot \rfloor$ 为向下取整运算, 相当于 MATLAB 中的 floor , 如 $\lfloor 3.1 \rfloor = 3$; 从上式中可以看出太阳赤纬角仅与和积日、年份有关。

2. 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)。其数值则表示了该天体与当地子午圈的角距离, 并借用时间的单位, 可以用小时来计量 ($1\text{HA} = 15^\circ$)。例如, 一个天体的时角是 2.5HA , 就表示他已经在 2.5 个小时之前通过当地的子午圈, 并且在当地子午圈的西方 37.5° 的距离上。负数则表示在多少小时之后将通过当地的子午圈。从文献中可查到其表达式 (单位: rad):

$$\omega = \pi \left(\frac{T_c - 12}{12} \right) \quad (6)$$

$$T_c = \frac{\phi - 120^\circ}{15} + t_c \quad (7)$$

$$t_c = t + \frac{\Delta t}{60} \quad (8)$$

$$\Delta t = 0.0028 - 1.9857 \sin \theta + 9.9059 \sin 2\theta - 7.0924 \cos \theta - 0.6822 \cos 2\theta \quad (9)$$

上式中, T_c 为当地地方时, t_c 为真太阳时。并且: 真太阳时 = 平太阳时 + 真平太阳时差: $t_c = t + \Delta t$, 其中 Δt 为平太阳时 (北京时间 t) 和真太阳时时差 (只与积日有关)。 Δt (单位: 分) 为平太阳时 (北京时间 t) 和真太阳时时差。从上式中可以看出时角与经度、真太阳时 (修正的北京时间) 有关。

3. 影子长度和杆长显然呈正比关系 (其他变量保持一定时), 此处不再详细讨论。

联立以上公式, 可以得出影子长度 l 的总表达式: ($\alpha \in (0, \frac{\pi}{2})$)

$$l = \frac{L \sqrt{1 - (\sin \alpha)^2}}{\sin \alpha} \quad (10)$$

$$\sin \alpha = \sin \psi \sin \delta + \cos \omega \cos \psi \cos \delta$$

$$\omega = \pi \left(\frac{T_c}{12} - 1 \right)$$

$$\delta = \frac{469 \pi \sin \left(\frac{2\pi(N+284)}{365} \right)}{3600} \quad (11)$$

5.1.2 影子长度变化的因素分析

由上述分析可知太阳影子长度与日期、经纬度、杆长、当天北京时间有关。而日期又包括年份与积日两个部分。下面分别考虑这些因素对太阳影子长度的影响:

1. 年份对太阳影子长度的影响: 画出北京市 1988 到 2016 年之间, 10 月 22 日上午 9 时整的影子长度变化曲线如图 3 所示。图中有明显的周期与分层现象, 这种现象归结于闰年、平年的周期变换, 故而图像中层次与周期均为 4。我们知道, 地球的公转过程是有误差的, 天文学中也有很多的矫正公式, 而我们上述模型中使用的也仅是近似公式, 故而出现了影子长度每四年均上升一定长度的现象。但是由于这种增长速度相对来说及其缓慢, 这些公式往往可以用于平时的计算过程中。由图 3 可以计

算,在这 29 年内,最大的影子长度平均年变化率为

$$p = \frac{6.72 - 6.70}{6.70 \times 28} \times 100\% = 0.016\%$$

可见由年份造成的误差几乎可以忽略,因此,本文在后续讨论中均不考虑年份的影响。

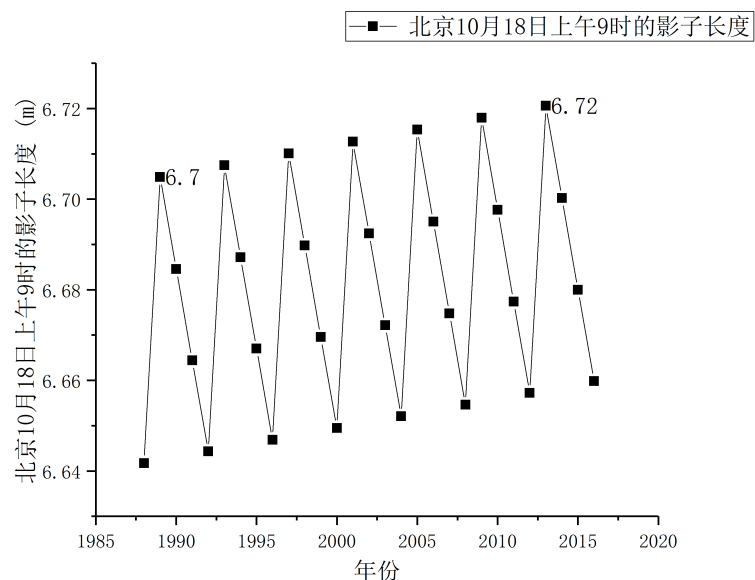


图 3: 年份对太阳影子长度的影响

2. 积日对太阳影子长度的影响: 画出 2015 年北京时间 9 点全年的太阳影子长度变化图如 4 所示。图中可以明显反映出北京四季的变化。夏季影子短,影子长度处于图中低谷阶段;冬季影子长,影子长度处于图中峰值阶段。

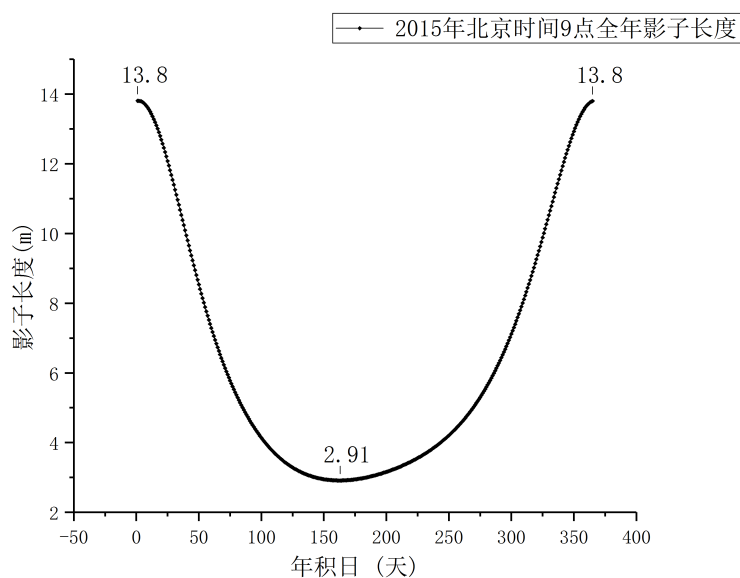


图 4: 积日对太阳影子长度的影响

3. 经纬度对太阳影子长度的影响: 经初步验算, 2015 年 10 月 22 日北京时间上午九点太阳应该直射南半球且影子长度为零的点应该在东经 120° 以东。故而本文选取南纬 $[-30^\circ, -10^\circ]$, 东经 $[150^\circ, 170^\circ]$ 为 XY 面绘制出影子长度关于经纬度的三维图像如图 5 所示。图中曲线上的黑色弧线为等高线, 可见以太阳直射点为中心, 太阳向四周辐射的相等影长呈环状分布 (同一时间, 相同杆长)。观察东经 150° 可以发现以太阳直射点以南, 纬度越高, 影子越长。而且, 随着纬度的增加影子长度增加的越来越快。对经度也有类似的结论。

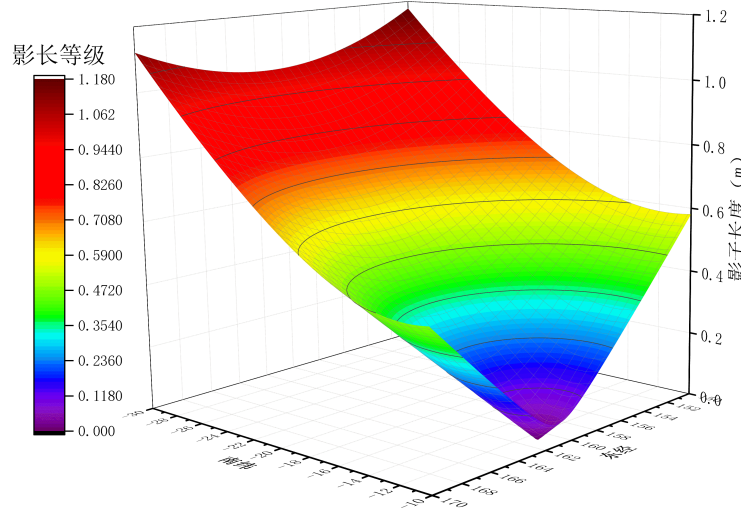


图 5: 经纬度对太阳影子长度的影响

5.1.3 影子长度变化模型的求解与分析

将问题一中具体参数: $year = 2015, N = 295$ 天 (10 月 22 日), $t \in [9, 15]$ (小时), 以及

$$L = 3, \psi = \frac{39.9072222\pi}{180}, \phi = \frac{116.391389\pi}{180}$$

代入上述影子长度变化模型可得太阳影子长度随当地北京时间 t 的变化曲线如图 6 所示。从图中可以看出影长变化的趋势为先减后增, 在北京时间 12.2 时达到正中午, 影长最短。为了看出一整天中影子变化规律, 我们将横坐标范围加大, 绘制出图 7。图中两个峰值分别对应着日出和日落时影长会发生突变的现象, 而峰值的不同说明早上和下午是有区别的, 图 6 中影长关于极小值点所在纵轴并不对称 (说明函数不是二次函数) 亦说明了这一点。我们默认太阳高度角 α 是大于 0 的, 对于图 7 中位于 0 以下的曲线部分实际上并无意义。

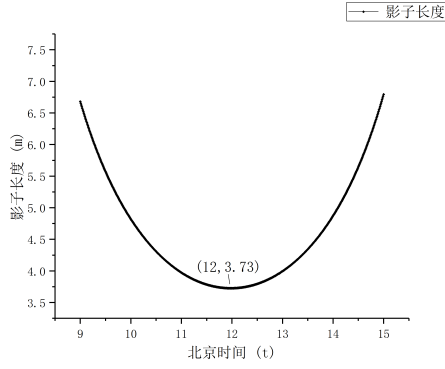


图 6: 太阳影子长度变化曲线

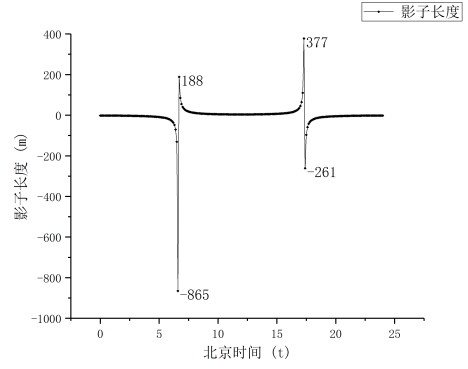


图 7: 全天影子长度变化曲线

5.2 问题二直杆地点反演模型的建立与求解

5.2.1 直杆地点反演模型的建立

分析问题二可知, 此处的未知量有经度 ϕ 、纬度 ψ 、杆长 L 、北京时间 t 、影子长度 l 。而拍摄地的地理位置由经纬度确定, 由问题一中模型经纬度可以表示为:

$$(\phi, \psi) = f(L, t, l) \quad (12)$$

由地理知识易知当某地的真太阳时等于 12 时, 影子长度达到最小。经度 $\phi_0 = 120^\circ$ 的地方在北京时间 $t_0 = 12$ 点时影子长度最小。从而, 如若知道某地白天影子最短时的北京时间 t' , 便可以通过下式求出当地经度。

$$\frac{\phi - \phi_0}{15^\circ} = t' - t_0 \quad (13)$$

为确定当地白天的影子长度变化关系, 利用附件一中的数据可以得到不同时刻的 21 个影长 ($l_i, i = 1, 2, \dots, 21$), 在此基础上, 结合问题一中的函数关系并利用三角函数拟合 (见 5.2.2) 可得影长与北京时间的变化关系 $l(t)$ 。利用 MATLAB 可求得 $l(t)$ 的极小值与极值点 (白天内只有一个极值), 从而求得经度 ϕ 。此时, 只需要确定纬度即可, 纬度的函数为:

$$\psi = f(L, t, l) \quad (14)$$

为求得最可能的地点, 亦即找出影子长度与附件一中 21 组数据最为接近的地方。为此, 基于最小二乘法的思想建立目标函数:

$$\min \Delta L = \sum_{i=1}^{21} (l_i - l_{t_i})^2 \quad (15)$$

其中, l_{t_i} 表示在任意给定一组 (ψ, L) 情况下, 当地北京时间为 t_i 时的的影长计算值。计算式为 $l_{t_i} = g(\psi, L, t_i)$, 函数 g 的表达式由公式14确定。

5.2.2 拟合三角函数的确定

由问题一中可知影子长度 l 与北京时间 t 的关系满足:

$$l(t) = \frac{L\sqrt{1 - (\sin\alpha)^2}}{\sin\alpha}$$

$$\sin\alpha = \sin\psi\sin\delta + \cos\omega\cos\psi\cos\delta$$

本问中拍摄日期 (4 月 18 日) 已经给定, 亦即积日 $N = 108$ 天已经给定, 由公式11可知太阳赤纬角 δ 是定值。而对于给定的某地、给定的直杆, 纬度 ψ 、杆长 L 亦应该为一定值。从而决定 l 的只有 ω , 而

$$\omega = \pi(\frac{T_c}{12} - 1), T_c = t + \Delta t$$

可知 ω 与北京时间 t 为一次函数关系。从而不难待定

$$\sin\alpha = A_2\cos(\pi(\frac{t}{12} - b_2)) + b_1$$

其中, $A_2 \in [-1, 1], b_1 \in [-1, 1]$ 。基于以上分析, 利用待定系数法可以给定拟合函数

$$l_1 = \frac{A_1\sqrt{1 - (A_2\cos(\pi(\frac{t}{12} - b_2)) + b_1)^2}}{A_2\cos(\pi(\frac{t}{12} - b_2)) + b_1} \quad (16)$$

其中, A_1, A_2, b_1, b_2 为需要拟合的参数。并且它们满足以下几式:

$$|A_1| = L \quad (17)$$

$$|b_1| = |\sin\psi\sin\delta| \quad (18)$$

可通过 Origin 自定义拟合函数拟合非线性曲线求解。

5.2.3 直杆地点反演模型的求解

5.2.3.1 拟合函数的求解

在 Origin 中自定义好拟合函数后对附件一中的影长进行拟合, 参数设置如表 1。利

表 1: Origin 非线性拟合参数设置

迭代算法	容差	加权方式
正交距离回归	1×10^{-14}	不加权

用 Origin 的迭代算法拟合时发现正交距离回归效果较好, 文献中给出了其与最小二乘法的误差对比, 正交回归更具优势。容差是给定数据和当前拟合函数之间加权距离相对变化的最小约束。如果拟合过程中的相对变化低于预先设置的容差, 拟合过程将停止。加权方式是指对所给数据中各个数据的偏重程度, 不加权相当于认为附件一所给 21 组数据地位平等。

经迭代收敛后得到报告如图 8、表2所示。

从表2中可以看出看出拟合效果非常好。四个参数的误差范围均十分小, 最大的波动仅达 0.011; Reduced Chi-Sqr 指加权卡方检验系数, 表示数据点和拟合函数相应点差的平方和, 这个值越接近于零越好; R^2 表示决定系数, 在 $[0, 1]$ 之间变化, 越接近于 1 越好。

我们知道, 一般的函数拟合 (通常不知道原函数的结构) 仅在给定数据区间具有较好的近似效果。但是本文的拟合函数是根据影子长度的具体函数结构构造而来, 可以比较

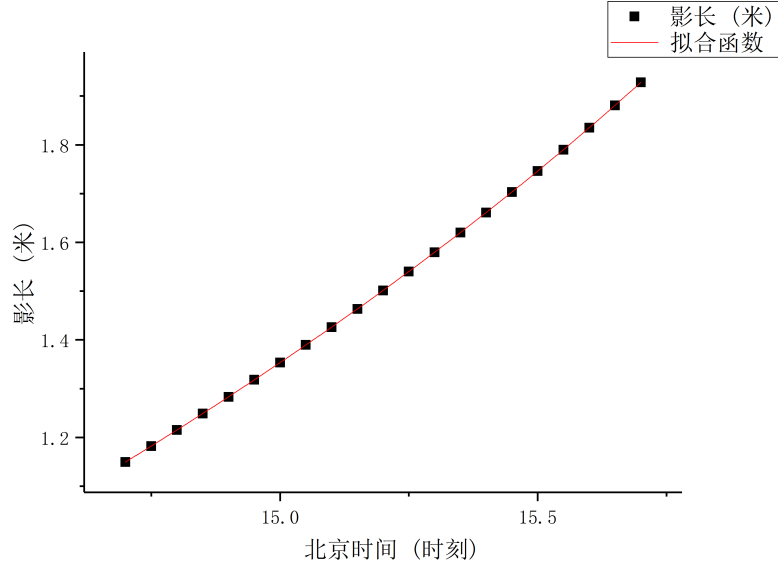


图 8: 拟合结果报告

表 2: 函数拟合参数报告	
参数	值
A1	-2 ± 0.01123
A2	-0.92184 ± 0.00204
b1	-0.06992 ± 0.00315
b2	1.05746 ± 0.00178
Reduced Chi-Sqr	1.11508×10^{-9}
R 平方 (COD)	1

好的近似整个区间上函数值。为此,我们画出由上述参数确定的拟合函数 l_1 在中午区间与一整天 24 小时区间的函数图像如图 9、10 所示 对比图 6、7 发现,此拟合函数 l_1 与问题一求解出的函数趋势、形状几乎一模一样,唯一不同点仅在于拟合的函数在日出、日落的峰值大小不一样。

基于此拟合函数的高度近似,我们完全可以不仅仅局限于用它来求解经度。容易计算 $\sin \delta = 0.1824$, 而 $b_1 = -0.06992$ 。由拟合函数的系数满足的公式18可得 $\psi = 22.5382^\circ$ 且 $L = 2$ 。故而可以推测这两个值应该是下文的一个理想解。亦即下文求出来的部分杆长解应该在 2 米左右,部分纬度解应该在 22.5382° 左右。

5.2.3.2 直杆地点反演模型的求解与分析

由上述拟合方法已经确定了拟合函数 $l(t)$ 。利用 MATLAB 可以求得白天中影子最短的北京时间为 12.6895,最短影长为 0.2583 米。将最短时间代入公式13,可得 $\phi = 109.6575^\circ$ 。

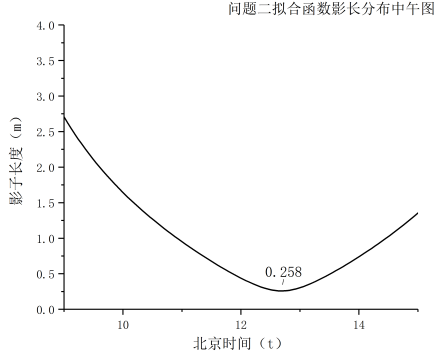


图 9: 拟合函数中午影长曲线图

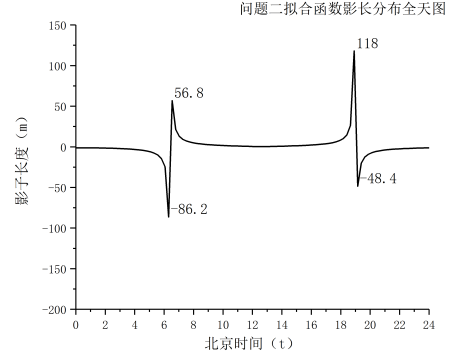


图 10: 拟合函数全天影长曲线图

从而我们的目标变成了以下的一个两个自变量的单目标优化问题:

$$\begin{aligned} \min \quad & \Delta L = \sum_{i=1}^{21} (l_i - l_{t_i})^2 \\ & l_{t_i} = g(\psi, L, t_i), i = 1, 2, \dots, 21 \\ & L \in (0, +\infty) \\ & \psi \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right) \end{aligned}$$

本文采用遗传算法求解,经尝试后调节好参数。程序单独运行 30 次为一组,一共运行十组,结果如表 3 所示。

表 3: 遗传算法所得结果		
纬度	杆长	适应度函数数量级
18.4120	1.9671	-6
18.4144	1.9671	-6
-0.1331	1.8479	-6
-0.1299	1.8480	-6
-0.1313	1.8479	-6
18.4137	1.9671	-6
-0.1302	1.8480	-6
18.4150	1.9671	-6
18.4148	1.9671	-6
-0.1319	1.8479	-6

从表中可以看出基本只有两个较优纬度: 18.41° 和 -0.13° 。并且杆长与预测值 2 米十分接近,而纬度 18.41° 与 22.5382° 亦相差不大。预测的如此准确归根结底是因为附件一所提供的数据十分可靠,使得根据问题一中函数结构拟合的函数极高的还原了原来函数的特质。附件一所提供数据为 4 月 18 日,太阳直射北半球,遗传算法求出的两个较优纬度基本关于直射纬度对称。故而可以推测最可能的地点为:海南岛三亚和西加里曼丹。

5.3 问题三直杆地点反演模型的建立与求解

5.3.1 直杆地点反演模型的建立

分析问题三可知,此问关键在于在第二问的基础上增加了积日 N 这一个未知量。由上述公式可知与 N 有关的公式有

$$\delta = \frac{469\pi \sin\left(\frac{2\pi(N+284)}{365}\right)}{3600}$$

即积日变化会引起太阳赤纬角 δ 的变化。因此,为了保证拟合的高度近似,需要对问题二中的拟合函数进行改进。

当引入一个新的变量 N 后,问题二中的纬度函数 l_{t_i} 的函数变为:

$$l_{t_i} = \varphi(\psi, L, t_i, N) \quad (19)$$

基于问题二中同样的思想,在求得经度 ϕ 的前提下。我们建立目标函数:

$$\min \Delta L = \sum_{i=1}^{21} (l_i - l_{t_i})^2$$

其中, l_{t_i} 表示在任意一组 (ψ, L, N) 情况下当地北京时间为 t_i 时的影长计算值。

5.3.2 拟合三角函数的改进

显然对拟合函数的改进关键在于公式

$$\sin\alpha = \sin\psi\sin\delta + \cos\omega\cos\psi\cos\delta$$

的待定。在充分考虑上式中的变量以及公式11可以给出如下待定:

$$\sin\alpha = A_2\cos(k_1\sin(b_3))\cos(k_2t - b_1) + b_2\sin(k_2\sin(b_3)) \quad (20)$$

从而最终确定的拟合函数为

$$l_1 = \frac{A_1\sqrt{1 - (A_2\cos(k_1\sin(b_3))\cos(k_2t - b_1) + b_2\sin(k_2\sin(b_3)))^2}}{A_2\cos(k_1\sin(b_3))\cos(k_2t - b_1) + b_2\sin(k_2\sin(b_3))} \quad (21)$$

上式中, A_1, A_2, b_1, b_2, b_3 为需要拟合的参数, $k_1 = 0.4093, k_2 = 0.2618$ 。同问题二,这些参数满足以下几式:

$$|A_1| = L \quad (22)$$

$$|A_2| = |\cos\psi| \quad (23)$$

$$b_3 = \frac{2\pi(N + 284)}{365} \pm 2k\pi, k \in Z \quad (24)$$

并且 $b_2 \in [-1, 1], A_2 \in [-1, 1]$ 。此拟合不仅可以求出经度 ϕ , 由此 3 式还可以分别给出杆长、纬度、积日的预测值。

5.3.3 直杆地点反演模型的求解与分析

5.3.3.1 附件二数据对应模型的求解与分析

Origin 拟合报告如表 4。可见拟合效果仍旧非常好，最高参数误差仅达 4.7 且 $R^2 = 1$ 。在 MATLAB 中求出此函数在北京时间 14.6876 时达影长最小值 0.6934 米。从而计算得到经度 $\phi = 79.6857^\circ$ 。为了确保在此拟合下是否仍能准确预测杆长、纬度、甚至

表 4: 附件二函数拟合报告	
参数	值
A1	-1.99099 ± 0.01463
A2	0.7678 ± 0.27691
b1	0.70362 ± 0.00157
b2	-1 ± 2.72981
b3	1.04044 ± 4.74445
Reduced Chi-Sqr	9.24525×10^{-10}
R 平方 (COD)	1

积日，我们画出拟合出的函数影子长度变化曲线如图11、12与问题一中的图9、10进行对比。

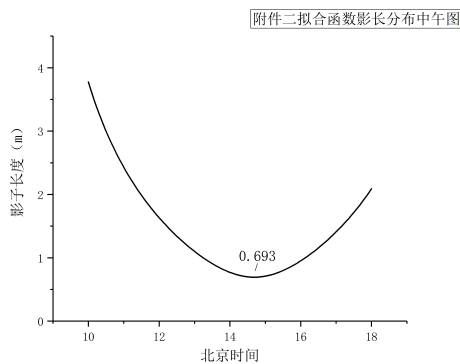


图 11: 拟合函数中午影长曲线图

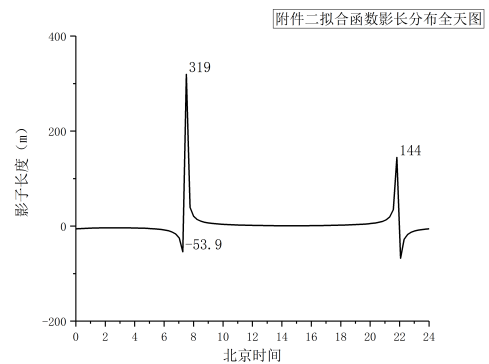


图 12: 拟合函数全天影长曲线图

进而发现所拟合的函数仍与问题一中函数高度相似，可以继续用于下面的变量预测：
 $L = |A_1| = 1.99099$, $\psi = \arccos(A_2) = 39.8433^\circ$, 而 $N = 141.4411$ 。

我们同样采用遗传算法求解 (三个变量), 得到结果如表 9。

表 5: 遗传算法所得结果

纬度	杆长	积日	适应度函数量级
35.1382	1.6540	116.6	-7
33.9814	1.8642	231.3	-7
—	—	—	—
39.5341	2.0403	203.3	-7
-37.8581	1.8939	316.6	-7
—	—	—	—
40.9046	1.8845	151.8	-7

由于此处遗传算法求出的较优解不是很集中,表格中仅筛选了满足适应度函数量级达到-7 以上的较优解。可见纬度预测值 39.8433° 、杆长预测值两米均十分可靠。而积日 N 虽然偏差较大,但也可以接受。注意到纬度基本对称分布,推测太阳直射点在赤道附近,选取积日为 151.8 的与 231.3 的解作为最可能日期。故而可以推测最可能处于新疆西部地区。时属 4 月份或七月份。

5.3.3.2 附件三数据对应模型的求解与分析

Origin 拟合报告如表 6。拟合效果较为乐观, $R^2 = 1$ 。在 MATLAB 中求出此函数在北京时间 12.6504 时达影长最小值 3.4685 米。从而计算得到经度 $\phi = 110.2440^\circ$ 。同

表 6: 附件三函数拟合报告

参数	值
A1	3.03565 ± 0.02534
A2	0.88044 ± 0.0078
b1	$3.31186 \pm 4.32682 \times 10^{-4}$
b2	0.57619 ± 0.03194
b3	17.27732 ± 34.97774
Reduced Chi-Sqr	1.3983×10^{-9}
R 平方 (COD)	1

样为了确保在此拟合下是否仍能准确预测杆长、纬度、甚至积日,我们画出拟合出的函数影子长度变化曲线如图13、14与问题一中的图9、10进行对比。进而发现所拟合的函数仍与问题一中函数高度相似,可以继续用于下面的变量预测: $L = |A_1| = 3.03565$, $\psi = \arccos(A_2) = 28.3045^\circ$, 而 $N = 354.66$ 。

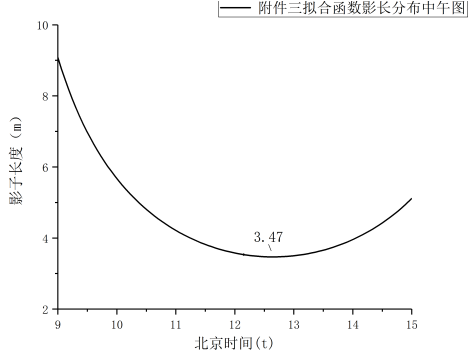


图 13: 拟合函数中午影长曲线图

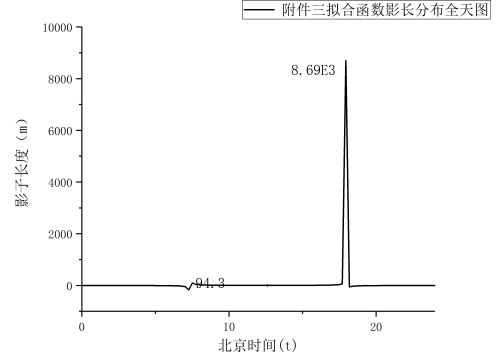


图 14: 拟合函数全天影长曲线图

本文同样采用遗传算法求解 (三个变量), 最终可以得到如下结果: 纬度主要分布在 $[29.44, 29.85]$ 与 $[50.87, 54.72]$ 之间; 杆长主要分布在 $[1.06, 1.35]$ 与 $[3.14, 3.51]$ 之间; 而积日则主要分布在 $[305.9, 329.2]$ 之间。亦即纬度与杆长预测较为准确, 而积日的预测值偏小。原因可能有二, 一是附件所给数据存在误差, 二是遗传算法过早陷入了局部最优解。综合来说二的可能性更大, 故而可以推测: 最可能位于湖南张家界地区, 时属 10 月份。

5.4 问题四基于图像处理的反演模型的建立与求解

本问题由于相机位于杆的前方, 影子的成像实际为真实影子的中心投影, 图像中直接测量得影子长度并非真实影子长度, 需要利用透视变换将图像转换为光线垂直于地面的正投影来计算影子长度。

5.4.1 模型的准备

5.4.1.1 透视变换

透视变换 (Perspective Transformation) 是指利用透视中心、像点、目标点三点共线的条件, 按透视旋转定律使承影面 (透视面) 绕迹线 (透视轴) 旋转某一角度, 破坏原有的投影光线束, 仍能保持承影面上投影几何图形不变的变换。其本质是将图像投影到一个新的视平面, 由资料中可知通用变换公式为

$$[x', y', w'] = [u, v, w] \times T \quad (25)$$

$$\begin{cases} x = \frac{x'}{w'} \\ y = \frac{y'}{w'} \end{cases} \quad (26)$$

其中, (u, v) 为原始图像的像素坐标, (x, y) 为变换后图像的像素坐标, T 为透视变换矩阵

$$\begin{bmatrix} a_{11}, a_{12}, a_{13} \\ a_{21}, a_{22}, a_{23} \\ a_{31}, a_{32}, a_{33} \end{bmatrix} \quad (27)$$

经过计算,透视变换的数学表达为:

$$\begin{aligned} x &= \frac{x'}{w'} = \frac{a_{11}u + a_{21}v + a_{31}}{a_{13}u + a_{23}v + a_{33}} \\ y &= \frac{y'}{w'} = \frac{a_{12}u + a_{22}v + a_{32}}{a_{13}u + a_{23}v + a_{33}} \end{aligned} \quad (28)$$

所以,对于给定的变换前后的 4 组对应图像像素坐标,即可求出上式中的各个参数。

5.4.1.2 视频与图像处理

按以下步骤进行操作:

1. 将视频载入 matlab,每隔一分钟取一帧,共得到 41 张图片。
2. 调用 opencv 图像处理函数库编写程序,载入所有图片,对图片进行透视变换,变换前取点如图15所示(假设真实世界中 AB, CD 两条直线相互平行)。



图 15: 变换前取点

原图像选取的点依次为 $A(274, 738), B(508, 683), C(1727, 683), D(1871, 738)$ 。变换后点依次为 $A'(1, 368), B'(1, 1), C'(1000, 1), D'(1000, 368)$ 。

3. 参考资料对变换后的图像进行预处理如图16

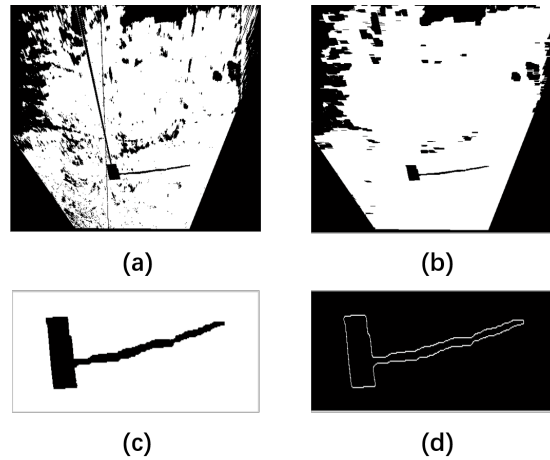


图 16: 变换后的处理

- (a) 对图像进行灰度化, 二值化
- (b) 以线性结构元素对图像进行闭运算, 填充图像空洞
- (c) 截取影子部分
- (d) 对截取后的图像中所有连通区域进行标号, 仅保留联通区域面积最大的部分 (即影子部分), 利用 canny 算子进行边缘检测

5.4.2 计算影子长度模型的建立

通过计算得出, 变换矩阵

$$T = \begin{bmatrix} -0.0867 & 0 & 0 \\ -0.3703 & -0.9261 & -0.0015 \\ 296.8916 & 632.4676 & 1 \end{bmatrix} \quad (29)$$

发现其中 $a_{12} = 0, a_{13} = 0$, 结合公式28可知, 变换后的 y 只和变换前的 v 有关, 即如果变换前若干点的像素纵坐标相同, 变换后的像素纵坐标也相同。所以考虑在图像的水平方向上添加辅助线来计算影长的实际距离。

对于截取的初始图像中, 假设以杆与地面的交点像素坐标 $O(x_0, y_0)$, 杆的真实长度为 Z , 杆的顶端像素坐标 (x_1, y_1) , 这里近似认为 $x_0 = x_1$ 。即杆为严格垂直于地面的。此时杆在图片中的长度

$$Z^* = |y_0 - y_1| \quad (30)$$

以 O 为顶点, 沿图像 x 轴正方向做辅助线, 真实长度为 $K = 0.5 \times Z$, 在初始图像长度上为 $K^* = 0.5 \times Z^*$, 参见图 17。对于变换后的图像, 由上述理论, 并结合图 18, 在此时的视平面中, 影子在图片中的长 S^* 与影子的真实长度 S 有以下关系

$$\frac{K^{**}}{K} = \frac{S^*}{S} \quad (31)$$

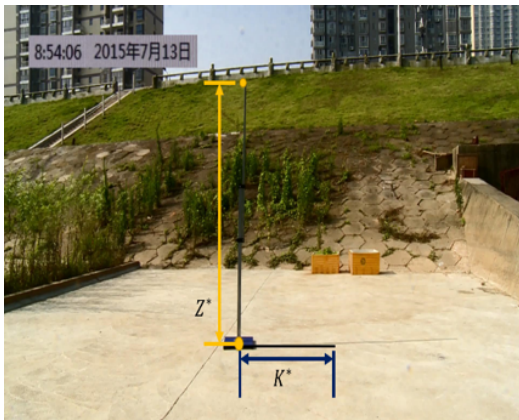


图 17: 透视变换前

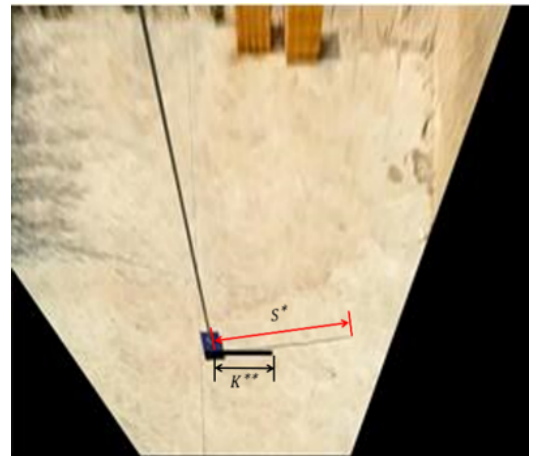


图 18: 透视变换后

其中, K^{**} 为辅助线在变换后图片中的长度。对于 S^* 的计算, 在上述对初始图像进行处理后, 得到影子的白色轮廓, 灰度值为 1, 对图片矩阵由右向左进行逐个监测, 检测到白色轮廓时记下此时像素点坐标 (x_1^*, y_1^*) , 假设给定处理后图片上的杆与地面交点像素坐标 (x_0^*, y_0^*) , 则:

$$S^* = \sqrt{(x_1^* - x_0^*)^2 + (y_1^* - y_0^*)^2} \quad (32)$$

5.4.3 直杆地点反演模型的求解与分析

对截取的所有 41 张图片利用上述模型求解影子的实际长度。部分结果如表 7 所示。不过经尝试发现, 由于这次拟合所用的数据是自己经过图像识别采集的, 并不符合问题一中的函数结构, 因此拟合效果非常差, 即便是用来求经度, 也十分的不合理。最终我们采用了二次拟合的方法计算出经度为 111° 。

表 7: 图像处理所得影子长度

北京时间 (t)	影子长度 (m)
8.90	2.61
8.92	2.59
8.93	2.58
—	—
9.50	2.01
9.52	2.00
9.53	1.99
9.55	1.98
9.57	1.97

5.4.3.1 拍摄日期已知 (7 月 13 日)

此问题化归成了问题二。采用遗传算法求得纬度基本都为 24.88° 。故而推测地点在广西桂林。

5.4.3.2 拍摄日期未知

如若拍摄日期也未知实际上便是问题三的求解。采用同样的算法可得如表 9 所示结果。

表 8: 遗传算法所得结果

纬度	积日	可能地点
35.6709	201	陕西渭南市
39.7608	198	内蒙古鄂尔多斯市
-35.365	13	-
36.9887	129	陕西榆林市
-28.3574	291	-
-35.3651	13	-
-32.7052	295	-
35.4806	201	陕西渭南市
34.0168	126	陕西省商洛市
34.1508	126	陕西省商洛市
-35.392	12	-

故而可以推测最优可能位于陕西省商洛、渭南一带。日期可能处于一月初或者六七月份。

5.4.4 问题四的灵敏度分析

本问题中的经度最终由二次函数拟合,所以在经度的计算上存在一定误差。表9给出了有日期时和无日期时经度变化对求解结果的影响。

表 9: 有拍摄日期

经度变化值	纬度变化值	纬度变化率
-2	34.8067	39.89%
-1	38.799	55.94%
0	24.8808	0.00%
1	18.0161	-27.59%
2	14.4676	-41.85%

由表9可以看出,在有日期时,对由我们拟合出的一天中影长随时间变化曲线来说,计算得出的经度发生微小的变化就会很大程度上影响最终的地理位置,说明拍摄地地理位置受经度变化影响较为灵敏,因此需要改进经度计算方法,提高经度的准确性;

表 10: 无拍摄日期

经度变化值	纬度变化值	积日变化值	纬度变化率	日期变化率
-2	35.3693	186	-4.38%	44.19%
-1	36.318	132	-1.81%	2.33%
0	36.9887	129	0.00%	0.00%
1	38.2441	124	3.39%	-3.88%
2	33.7315	221	-8.81%	71.32%

由表10可以看出,在日期未知的情况下,经度在一定范围内变化时,纬度变化率非常小,说明纬度受经度变化影响较小,而日期的变化在初始计算出的经度基础上小范围(± 1 度)内变化时,变化率也非常小,而在较大范围(± 2 度)内变化时,变化率很大,说明日期未知时,纬度受经度变化影响较小,日期在一定程度上受经度变化影响较小。综上所述,为了更加准确的确定视频拍摄地的地点与日期,在计算经度的过程中需要不断改进与完善模型,使计算得出的经度更加准确,从而减小误差,使结果更加精确。

六、 模型的评价

6.1 模型的优点

1. 在问题一中考虑了真太阳时,对平太阳时进行了修正。
2. 拟合采用依据问题一函数结构自定义的拟合函数,高度近似了原影长函数并可以给出准确的预测值。
3. 问题四中图像处理能够减少透视投影引起的影子长度的误差。

6.2 模型的缺点

1. 对图像的透视变换及一系列的处理过程中会在一定程度上对图像造成失真,导致测量结果不准确.
2. 问题四中的二次拟合会造成一定误差.
3. 利用遗传算法得出的解可能为局部最优解,与真实解存在一定差距。

6.3 模型的推广

本文中的基于函数结构的自定义非线性拟合可以推广到很多种实际问题中,不仅可以拟合参数,也可以提供预测^[6]。^[3] ^[1;4] ^[2]

参考文献

- [1] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [2] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- [3] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.
- [4] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [5] Fei Wu and Ramteen Sioshansi. A stochastic flow-capturing model to optimize the location of fast-charging stations with uncertain electric vehicle flows. *Transportation Research Part D Transport & Environment*, 53:354–376, 2017.

- [6] 胡毅华, 杨旭龙, and 刘媛萍. 太阳影子定位模型的构建. 洛阳师范学院学报, (11):13–18, 2015.

A 附录

1.1 问题一源程序

代码描述: **Input matlab source:**

```
function [path_celue,numbers,status]=Find_path(origin,destination,PATH,
    numbers,allowed,Strategy,steps,duian_allow)
%% Success
    if origin==destination
        path_celue=PATH;
        status=1;
        numbers=numbers+1;
        steps=steps;
        return
    end
    for dire=1:length(Strategy)
    % Next status
        origin_now=origin+Strategy(dire, :)*(-1)^steps;
        if (origin_now>=0)==ones(1,6) & (origin_now<=1)==ones(1,6)
            duian=ones(1,6)-origin_now;
            [origin_flag,ia,ib] =
                intersect(origin_now, allowed, 'rows');
            duian_flag = intersect(duian, duian_allow, 'rows');
            if origin_flag==origin_now & duian_flag==duian
                temp=allowed(ib, :);
                allowed(ib, :)=[];
                PATH=[PATH;dire];
                steps=steps+1;
                [path_celue,numbers,status]=
                    Find_path(origin_now,zeros(1,6),
                        PATH,numbers,allowed,Strategy,steps,duian_allow);
                if status==1
                    return
                else if status==0
                    allowed=[allowed;temp];
                    steps=steps-1;
                    PATH(length(PATH), :)=[];
                end
            end
        end
    end
    end
    path_celue=[];
```



```
status=0;
numbers=0;
steps=0;
```

end

1.2 问题二源程序

代码描述:**Input C++ source:**

```
//=====
// Name      : Sudoku.cpp
// Author    : wzlf11
// Version   : a.0
// Copyright  : Your copyright notice
// Description : Sudoku in C++.
//=====

#include <iostream>
#include <cstdlib>
#include <ctime>
using namespace std;
int table[9][9];
int main() {
    for(int i = 0; i < 9; i++){
        table[0][i] = i + 1;
    }
    srand((unsigned int)time(NULL));
    shuffle((int *)&table[0], 9);
    while(!put_line(1))
    {
        shuffle((int *)&table[0], 9);
    }
    for(int x = 0; x < 9; x++){
        for(int y = 0; y < 9; y++){
            cout << table[x][y] << " ";
        }
        cout << endl;
    }
    return 0;
}
```

代码描述:**Input matlab source:**

```
clc
clear
data = normrnd(0,1,100,1);%mu,sigma,row,column
```

```

%绘制频率直方图，纵坐标为频率
[counts,centers] = hist(data,10);
bar(centers,counts / sum(counts));
%核密度估计
[f,xi] = ksdensity(data);
hold on;
plot(xi,f);
%正态分布拟合
[mu,sigma] = normfit(data);
xj = -5:0.01:5;
yj = normpdf(xj,mu,sigma);
hold on;
plot(xj,yj);
%美化图片
%legend('原始数据','核密度估计','正态拟合');
legend('initial data','ksdensity','normfit');
%fig2plotly(gcf,'offline',true);

```

1.3 问题三源程序

代码描述:**Input Python source:**

```

# coding='utf-8'
"""# 一个对s曲线数据集上进行各种降维的说明。"""
from time import time

import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from matplotlib.ticker import NullFormatter

from sklearn import manifold, datasets

# # Next line to silence pyflakes. This import is needed.
# Axes3D

n_points = 1000
# x是一个(1000, 3)的2维数据，color是一个(1000,)的1维数据
X, color = datasets.samples_generator.make_s_curve(n_points,
    random_state=0)
n_neighbors = 10
n_components = 2

fig = plt.figure(figsize=(8, 8))
# 创建了一个figure，标题为"Manifold Learning with 1000 points, 10

```

```

    neighbors"
plt.suptitle("Manifold Learning with %i points, %i neighbors"
            % (1000, n_neighbors), fontsize=14)

''' 绘制S曲线的3D图像'''
ax = fig.add_subplot(211, projection='3d')
ax.scatter(X[:, 0], X[:, 1], X[:, 2], c=color, cmap=plt.cm.Spectral)
ax.view_init(4, -72) # 初始化视角

'''t-SNE'''
t0 = time()
tsne = manifold.TSNE(n_components=n_components, init='pca',
                    random_state=0)
Y = tsne.fit_transform(X) # 转换后的输出
t1 = time()
print("t-SNE: %.2g sec" % (t1 - t0)) # 算法用时
ax = fig.add_subplot(2, 1, 2)
plt.scatter(Y[:, 0], Y[:, 1], c=color, cmap=plt.cm.Spectral)
plt.title("t-SNE (%.2g sec)" % (t1 - t0))
ax.xaxis.set_major_formatter(NullFormatter()) # 设置标签显示格式为空
ax.yaxis.set_major_formatter(NullFormatter())
# plt.axis('tight')

plt.show()

fig.savefig('sample.png', dpi=600)

```

1.4 问题四源程序

代码描述: **Input R source:**

```

## 设置路径
setwd("C:/Users/16053/Documents/R/18guosai/降维/")
shouxin<-read.csv(file="tsne_train.csv",header = TRUE);shouxin

## Not run:
library(tsne)

vec=550:670
# 定义绘图函数
ecb = function(x,y){ plot(x,type='p',label='',pch=17,col=colors[
    shouxin$result[vec]]) }

```

```

# tsne 降维
tsne_iris = tsne(shouxin[vec,3:30], epoch_callback = ecb, perplexity
    =50)
# compare to PCA

dev.new()
pca_iris = princomp(shouxin[vec,3:30])$scores[,1:2]
plot(pca_iris, type = 'p', lable='', pch=17, col=colors[shouxin$result[vec
    ]])
## End(Not run)

## MDS 降维
## ggplot_mds
df <- scale(shouxin[,3:30]) ;df
d.mds<- dist(df)
mds=cmdscale(d.mds,k=2,eig=T)
mds_1 = mds$points[,1]
mds_2 = mds$points[,2]
library(ggplot2)
library('grid')

p=ggplot(data.frame(mds_1,mds_2),aes(mds_1,mds_2))
Default<-factor(shouxin$class)
main_ploy<-p+geom_point(size=3,alpha=0.8,aes(colour=Default))+
    theme(legend.title = element_text(size=12,face = "bold"))+
    annotate("segment", x = -1.7, xend = -0.8, y = 2.8, yend = 6.8, size
        = 1.2,colour = "blue",
        arrow = arrow(ends = "both", angle = 30, length = unit(0.2, "
            cm"))))

subplot<-p+geom_point(size=3,alpha=0.8,aes(colour=Default))+coord_
    cartesian(xlim=c(-2.8,-1),ylim = c(-2,2))+
    theme(panel.grid.major = element_blank(),panel.grid.minor = element_
        blank(),
        legend.position = "none",panel.grid =element_blank(),panel.
            border = element_blank(),
            axis.text = element_blank(),axis.ticks = element_blank(),axis.
                title = element_blank())
main_ploy
subvp_1 <- viewport(width=0.5,height=0.4,x=0.35,y=0.75)
print(subplot, vp=subvp_1)

```
