

太阳影子定位

摘要

在视频数据分析中，视频拍摄的时间与地点信息具有极其重要的价值。本文根据相关的地理知识，以太阳影子的变化为突破口，在日期、时间与经纬度之间，建立了统一的数学模型。

针对问题一，使用统一公式，将具体值代入即可求得太阳影子长度的变化曲线，影长在 9 时至 12 时时间段内先减小再增大，在 12 时影长最短，为 3.8414 米。系统性的鲁棒分析过于复杂，本文使用了控制变量的思想，每次控制四个参数不变，通过作图的方式分析影长与另一参数之间的关系，研究讨论了影长与经纬度、北京时间与日期之间的关系。

针对问题二，采用化归的思想，将缺少的经纬度与杆长看做未知元，使用 *MATLAB* 用遍历的方法反解三元方程，得到若干可能的点，再使用最小二乘法对其进行精筛选并进行拟合验证。但考虑到地理学使用的公式大多为近似公式，理论最优点有可能并不是实际点，故进一步结合散点图，研究可能点的分布趋势，最终得出测量点最有可能位于海南省万宁市，较有可能位于云南省腾冲县与普洱县之间。

针对问题三，继续使用统一模型，分别对附件二、附件三中的数据使用遍历反解的方法，计算可能的地点、时间与日期。并由求出数据的散点图发现日期与纬度呈近似的正弦关系，得出附件二的测量点最有可能是 5 月 5 日与 7 月 27 日的新疆喀什地区，较有可能是 5 月 8 号的新疆和田地区；附件三的测量点最有可能是 9 月 28 日的宁夏银川，较有可能是 3 月 16 日的山西吕梁。

针对问题四，考虑到透视畸变、直杆倾斜与缺乏基准点难以进行反演与测量，故使用 *Photoshop* 近似的测取影长，使用统一模型反解，得出测量点最优可能是内蒙古锡林郭勒盟，较有可能是内蒙古乌兰察布；在日期未知的情况下，得出测量点最有可能是 5 月 27 日的内蒙古呼和浩特，较有可能是 5 月 6 日的张家口，次有可能的是 8 月 19 日的山东淄博。

本文还使用问题二的数据对建立的统一模型进行运算开销的分析，得到了本模型较适宜的遍历深度，结合问题一种的参数分析，较好的检验了本模型的合理参数与稳定性，较好的解决了问题。

关键词：太阳高度角 控制变量 最小二乘法 深度遍历

问题重述

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.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?

一. 问题分析

确定视频的拍摄地点与拍摄时间对视频数据的处理意义非凡,然而在实际处理数据的过程中,采集视频所能获得的信息维度是不确定的,往往会有所遗漏或赘余。本题共四问,前三问的任务就是在不同信息富余程度下,确定视频的拍摄地点与拍摄时间。第四问则是在前三问的基础上加入视频数据转化处理的环节。

问题一中,需要解决的是一个已知日期、时间、纬度、杆长、影长,求太阳在直杆上投影影长的问题。建立地平坐标系,根据地理学的基本公式,构造数学模型,再使用 *MATLAB* 将相关数据代入模型,就可以得到影长的变化曲线。

问题二中,已知条件有日期、时间差、影长。参考问题一的模型,建立二元方程,利用程序遍历可能解,再通过最小二乘法进行筛选,便可得若干个可能的地点;问题三中,已知数据继续减少,影长,时间差,在第二问的基础上,对缺失数据进行遍历。再通过最小二乘法筛选数据即可。

问题四中,最大的难点显然是讲视频数据转化为数值数据。本题使用的地理定位方法是光度测量方法,需要分析太阳的角速度与视频中的投影在太阳旋转平面的投影角度。由于透视畸变,拍摄镜头与拍摄目标间空间距离的变化会使等长

物体在视频中所占的像素块大小发生改变，同时视频中的直杆存在倾斜，投入运算之前需要进行矫正。

最后，通过加大遍历密度测算模型开销经济性与控制变量测量模型稳定性对建立的模型进行校验，并评价结果的正确性和方法的可靠性。

二. 基本假设

- (1) 假设数据反映真实情况，忽略测量带来的误差。
- (2) 假设太阳光线是平行光，忽略大气层对太阳高度角的影响。
- (3) 假设测量过程中，测量环境平稳，地面平整，湿度适中，粉尘较少。
- (4) 假设同一日期的赤纬角不变，忽略由于日地关系变化造成的赤纬角变化。
- (5) 假设地球是一个理想球体，忽略梨形地球在不同角度下的转速区别。

三. 符号系统

符号	含义
E	时差
t_0	真太阳时
t_B	北京时间
L_0	经度
φ	纬度
ω	时角
δ	赤纬角
n	日期序列
α	太阳高度角
H	杆长
Y	影长
β	阈值

四. 模型的建立

太阳影子定位技术的原理实质上与日晷是相似的，都是利用太阳位置来进行时间的计量，区别是太阳影子定位技术还可以结合不同时间下的日地关系与少量附加信息，更进一步的获得测量点的地理位置。换言之，太阳影子定位技术就是研究并利用日地运行规律的技术。

分别以空间与时间的角度对日地运行规律进行考察，可衍生出对应的空间坐标系系统与时间系统。具体而言，就是公转导致了四季更替，而想要考察日地间相对位置，则必须要建立合适的坐标系；时间是通过物质的运动形式来表达的，而自转产生的昼夜分割催生了时间计量系统。

为了说明本文用到的相关变量，需先建立坐标系与时间系统，为此先扼要介绍下天文学中广泛使用的天球坐标系与太阳时的概念。

5.1 模型的准备

天球坐标系：

中国古代对天文地舆的描述大体上可归纳为“天圆地方”，天空像一个巨大的空心半球罩在大地上，而大地是一切的中心，而在西方天文学中也有类似的表述：假定存在一个“天球”，以地球为坐标原点，无限长为天球半球，天空中的各种天体的投影错落分布在这个球的内表面上。根据运动的相对性原理，太阳好像在这个球面上周而复始的运动，本文中用到的天球坐标系包括赤道坐标系与地平坐标系。

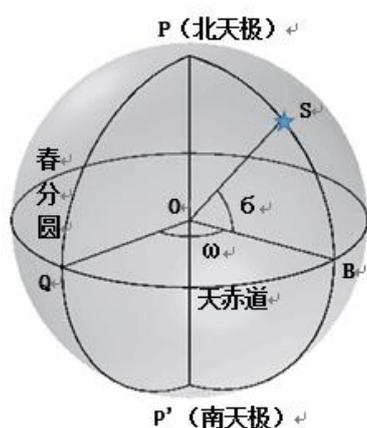


图 5-1：赤道坐标系

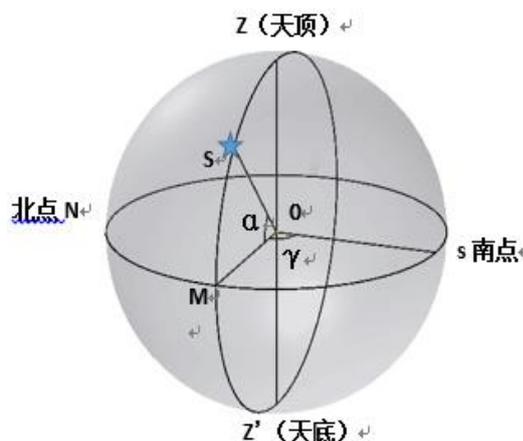


图 5-2：地平坐标系

赤道坐标系：

将地球两端无限延长，构成的这条直线成为天轴，天轴与天球球面相交于两点，为天极，与地球北极对应的一极为北天极 P，另一极为南天极 P'；将地球的赤道无限的扩大，与天球球面相交所得的大圆为天赤道 QQ'，以天赤道为基

本圈，以天赤道和天子午圈的交点 Q 为原点的天球坐标系就是赤道坐标系，如图 5-1 所示，其坐标值为时角 ω 和赤纬角 δ 。

地平坐标系：

观测者相对于地平面所在的铅垂线，延长后与天球面相交于两点，在测者头顶上的一点为天顶 Z，另一点为天底 Z'。通过球心 O 与 ZZ 相垂直的平面在天球上所截出的大圆叫做真地平，以真地平为基本圈，以南点 S 为原点的天球坐标系就是地平坐标系，如图 5-2 所示，其坐标值为高度角 α 和方位角 γ 。^[1]

太阳时：

以太阳的视圆面中心表示真太阳，以真太阳作为量时天体所计量的时间叫做真太阳时，真太阳时可以表示时角，但是由于黄赤交角的存在以及地球公转速度不均匀，真太阳时是不等长的。日常使用的时间，是以真太阳的平均角速度，在天赤道上自西向东运行的假想一个量时天体（平太阳），并以此授时的，所以真太阳时需要用平太阳时换算。

5.2 模型的建立

下面开始说明本文用到的变量及其计算公式。

时差 E：

钟表所指示的时间叫平太阳时，真太阳时与平太阳时之差叫做时差 E，即：

$$E = t_0 - t \quad [2] \quad (1)$$

其中 t_0 为真太阳时， t 为平太阳时，E 可通过查询中国天文年历获得。

真太阳时 t_0 ：

我国使用的平太阳时为“北京时间”，位处东八区，与格林威治时间相差 8 小时。由地球每小时自转 15。与北京时间，可推算任意经度地点的平太阳时。

$$t = t_B - 4 \times (120 - L_0) \quad (2)$$

式中 t_B 指北京时间， L_0 指地方时间的标准子午线经度，单位为度。

由式 (1) 和式 (2) 得任意地区真太阳时和北京时间之间关系为：

$$t_0 = t_B + E - 4 \times (120 - L_0) \quad (3)$$

时角 ω ：

时角是天体相对子午圈的方位和角距离，如图 5-1 中 $\angle QOB$ 所示，是过天体 M 的时圈平面与午圈平面之间的夹角，从天子午圈上 Q 点开始，向西顺时针度量，数学表达式为：

$$\omega = 15(t_0 - 12) \quad (4)$$

式中 t_0 单位为小时

赤纬角 δ :

赤纬是天体相对于天赤道的方向和角距离，如图 5-1 中 $\angle SOB$ 所示。是天体 S 与地心连线与天赤道平面之间的夹角，从天赤道开始向北天极方向度量为正，向南天极方向为负， δ 可由 Cooper 方程近似计算得到：

$$\delta = 23.45 \sin\left(\frac{2\pi(284+n)}{365}\right) \quad (5)$$

其中 n 为日期，例如，1 月 1 日为 $n=1$ ，9 月 13 日为 $n=256$ 。

太阳高度角 α :

太阳高度角是太阳相对于地平线的高度角，如图 5-2 中 $\angle SOM$ 所示，这是以太阳视盘面的几何中心和理想地平线所夹的角度。太阳高度角可以使用下面的算式，经由计算得到很好的近似值^[2]：

$$\sin \alpha = \sin \varphi \sin \delta + \cos \varphi \cos \delta \cos \omega \quad (6)$$

式中 α 为太阳高度角， ω 为时角， δ 为当时的太阳赤纬， φ 为当地的纬度。

直杆-影子系统:

本题的任务是根据直杆的影长获得测量点的时间与位置，实现这个过程的关键装置就是由直杆与其影子构成的系统。

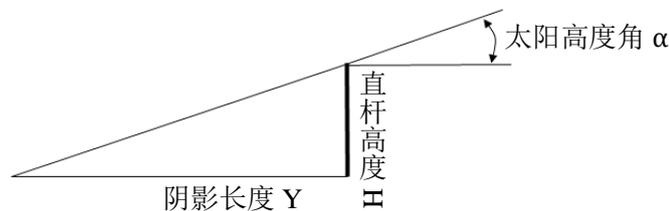


图 5-3: 太阳高度角原理图

如图 5-3，分别以影长与杆长为直角边，构建一个简单的直角三角形模型，太阳高度角 α 即为夹角 α 。换言之，得知任一点影长与杆长的比例关系，便可得到该点得太阳高度角，并借由上文提及的各个公式，得到该点得时间与位置信息。影长与杆长关系数学表达式为：

$$Y = \frac{H}{\tan \alpha} \quad (7)$$

式中 H 是直杆长度， Y 为阴影长度。

既此，我们可以找到太阳高度角、纬度、日期、北京时间、时差、经度之间的函数关系，太阳高度角是问题的关键，其求解流程如下：

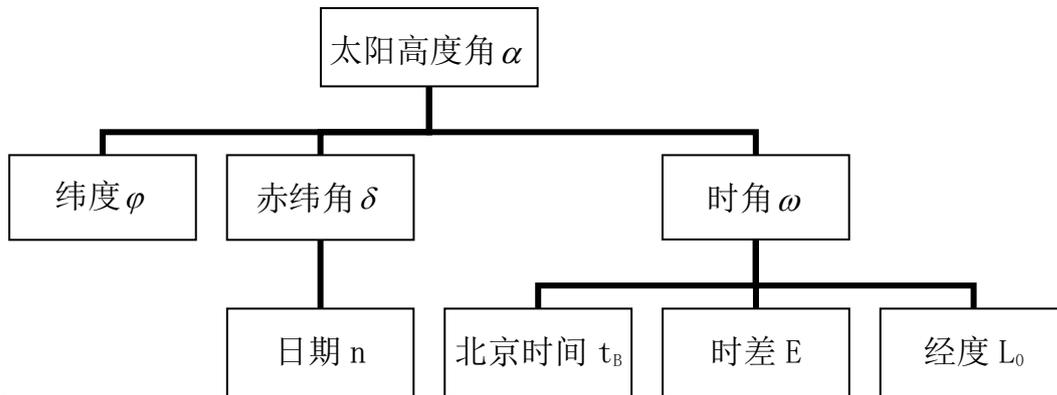


图 5-4：各因素数量关系图

由式 (3) (4) (5) (6) (7) 得到影长和北京时间、当地纬度、经度、杆长、日期五个独立变量有关，其统一表达式如下：

$$Y = \frac{H}{\tan\{\arcsin[\sin\varphi\sin\delta - \cos\varphi\cos\delta\cos(15t_0 - 180)]\}} \quad (8)$$

上式中 t_0 与 δ 可分别由式 (3)，式 (5) 求出。

本题四问都可以使用 (8) 式进行反解，区别仅在未知数数量与相应的遍历次数。第一问中数据充裕，可以获得唯一解；第二问、第三问与第四问由于数据的缺失，会得到一系列可能的时间或者地点。要得到符合现实条件的时空点，这就需要对初次计算后的取值进行筛选。(8) 式中的统一公式是以影长 Y 作为输出的，所以以影长 Y 为统一的筛选指标。

理论上与实际杆长最接近的时空点是最优解，但由于误差的存在，实际测量点处的时空点可能并不是理论上的最优点。所以最后无法得到确定的时空点，仅仅能获得一系列可能性较高的时空点。

本文使用最小二乘法，对求得的杆长与实际杆长进行最小二乘化处理，数学表达式如下：

$$\sum(Y_i - Y_0)^2 \leq \beta^{[3]} \quad (9)$$

其中， Y_i 为计算所得的杆长， Y_0 为实际杆长，阈值 β 设定为 0.05。

五. 模型的求解

6.1 求太阳影子长度的变化曲线

本问中给出了日期、经纬度、杆长、北京时间，根据（8）式

$$Y = \frac{H}{\tan\{\arcsin[\sin\varphi\sin\delta - \cos\varphi\cos\delta\cos(15t_0 - 180)]\}}$$

将 $n=295$, $L_0=116$ 度 23 分 29 秒, $\varphi=39$ 度 54 分 26 秒, 北京时间 9:00-15:00 代入, 可得影子长度的变化曲线, 如图 6-1 (a) 所示:

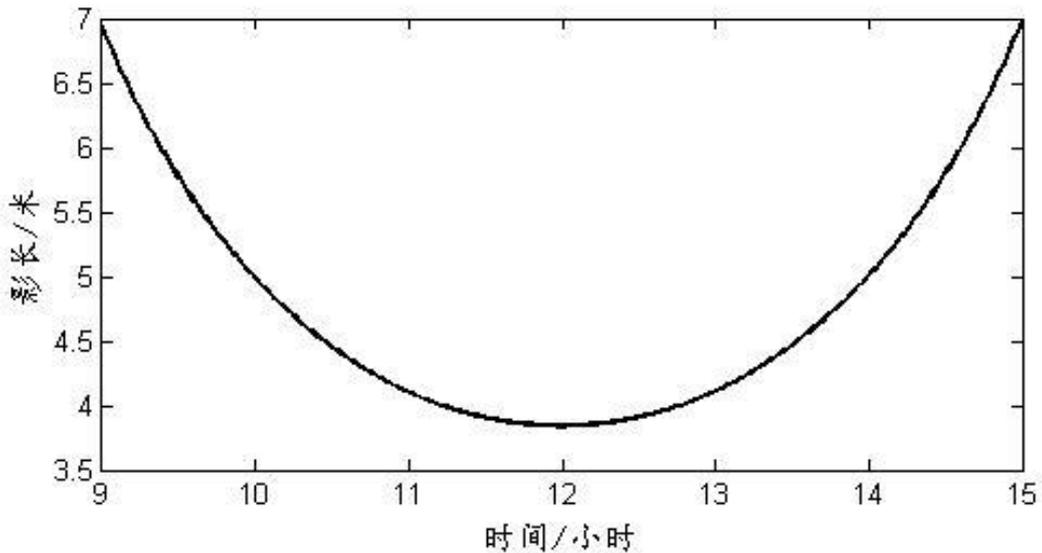


图6-1(a): 影子长度变化图

由图 6-1 (a) 可知, 北京天安门广场 2015 年 10 月 22 日那天在 9 时至 15 时内物体影长变化曲线近似为抛物线, 3 米高的直杆其影长在 9 时至 12 时时间段内不断减小, 随后又不断增大, 在 12 时影长达到最短, 为 3.8414 米。

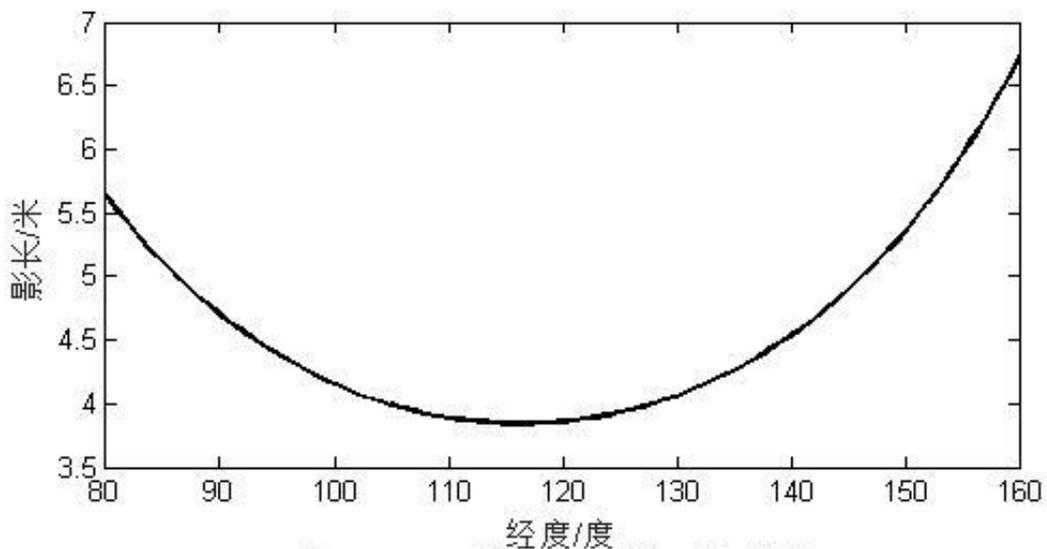


图6-1(b): 影子长度随经度变化图

为分析影子长度关于各个参数的变化规律，我们采用控制变量法的思想，每次控制四个参数不变，通过作图的方式分析影长与另一参数之间的关系。此法虽不能全面准确反映影长与某一因素之间关系，但直观方便。

1. 影子长度随北京时间变化规律如图 6-1 (a) 所示。

2. 影长随经度变化规律如图 6-1 (b) 所示，在东经 117 度之前，影长随经度的增加不断减小，后又开始增大，在东经 117 度达到最小。

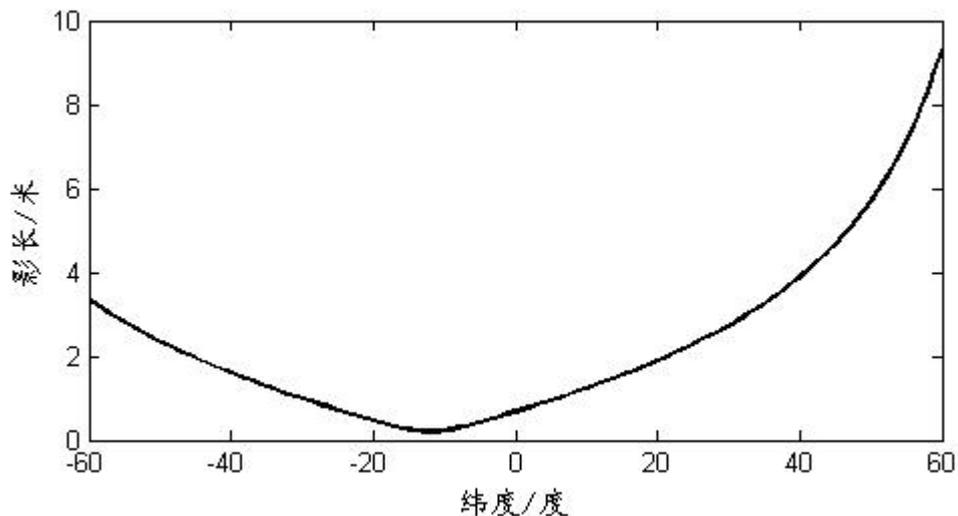


图6-1(c): 影子长度随纬度变化

3. 影长随纬度变化规律如图 6-1 (c) 所示，在南纬 60 度至南纬 11.7 度之间，长度随纬度减小而减小；在南纬 11.07 度至北纬 60 度之间，影长随纬度增加而增大。

4. 影长随日期变化规律如图 6-1 (d) 所示，影长从前一年 11 月 25 日至当年 5 月 22 日不断减小，后又不断增大直到 11 月 25 日达到最大。

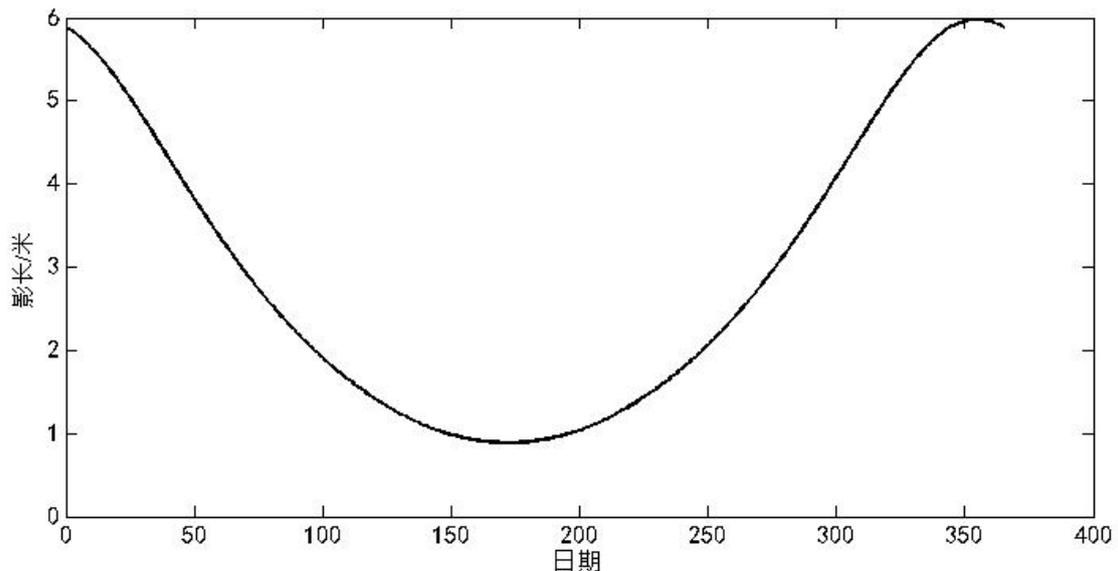


图6-1(d): 影子长度随日期变化图

6.2 根据太阳影子长度求可能的地点

第二问与第一问相比，缺少了经纬度、杆长，(8)式变成了一个三元函数。利用附件中影子的坐标可以计算出实际测定的影子长度，又已知测算时的北京时间，及测算时的日期，由之前的公式我们，知道影子与日期，经纬度，杆长，北京时间 5 个有关，因此我们可以穷举拟合的手段近似计算出一组理论影长与实际测算影子长度进行对比，取出其中较好的点作为可能的参照点。设定遍历深度，通过遍历所有的可能点，可以优先计算出首个点的影长与实测首个影长做差，确定一个较好的阈值，进行第一步的粗略筛选，这样可以大大减少消耗的时间，且可以得到较好的结果并进行下一步的筛选，算法流程如图 6-2 所示：

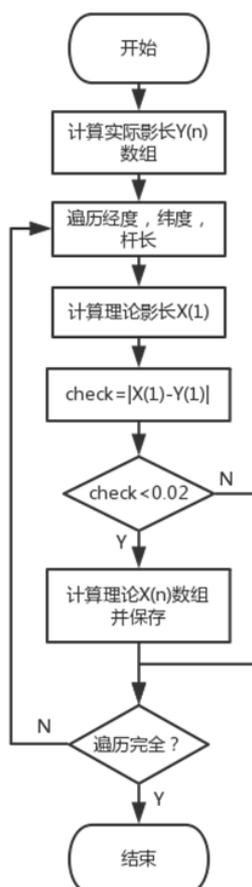


图 6-2: 算法流程图一

步骤一：

计算实测影子长度

步骤二：

顺序遍历经度、纬度、杆长，遍历深度分别为 1、0.1、0.1。

步骤三：

计算理论影长，对每组首个理论影长与实际影长进行对比，做差取绝对值，与阈值对比。

步骤四：

小于阈值 0.02，则保留数据；否则，跳至步骤二。

步骤五：

检验是否遍历完经度、纬度、杆长，若遍历完全，则结束；否则，重复步骤二。

通过遍历，可得到 27732 个点，使用最小二乘法对这些粗略点进行筛检，可得 19 个点，再去除位于国外或海上的坐标，其中理论上可能性最大的 6 个点坐标如表 6-1 所示：

表 6-1

纬度	经度	地点
24.7	98.5	云南保山市腾冲县
22.9	101.5	云南普洱宁洱哈尼族彝族自治县
23	101.5	云南普洱墨江哈尼族自治县
18.7	110.5	海南省万宁市
18.8	110.5	海南省万宁市
18.9	110.5	海南省万宁市

将这 6 个点进行拟合校验，其拟合曲线如图 x 所示：

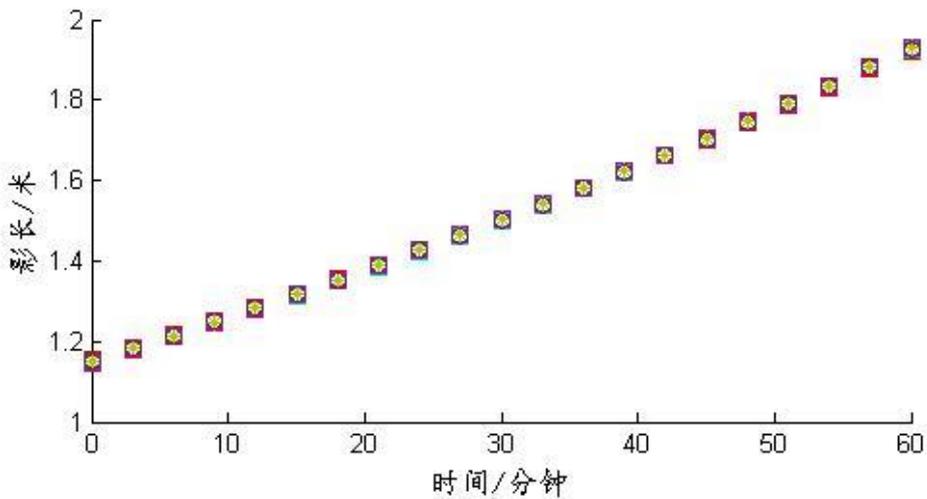


图 6-3：多点拟合图

由图可知，这 6 个点的拟合程度极佳。但由于拟合程度的较小差别，仍能引起经纬度的较大变化，且影长测量过程中不可避免地存在一定误差，所以现实测量点可能反而因为残差项较大或散落在国境陆地以外的地方而被排除。但测量点的散点图可以反映可能点的分布趋势，可以大致的确定可能点位置，故做 19 个点的散点图，这里只考虑国境以内或国境周边的点，其局部散点图 6-4 如下：

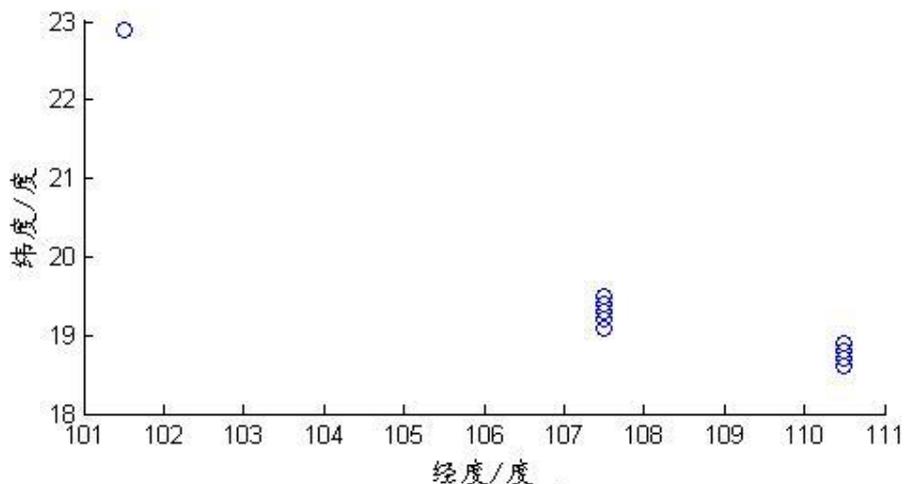


图6-4: 可能点数点图

分析散点图可知，可能点大致分布于东经 100 度北纬 22 度左右与东经 110 度北纬 19 度左右的区域，查询地图可知，这两块区域大致为海南西部的北部湾，与海南岛上东南部的万宁市。

综上，表 6-1 中的点均为可能点，测量点位于海南的三个可能点的可能性比云南境内的可能点大。

理论上，为消除误差，获得更精确的可能点，可以通过提高遍历深度来实现。若将遍历深度调高一个数量级，经度、纬度、杆长的遍历深度分别为 0.1、0.01、0.01，则时间复杂度增加 1000 倍。

将纬度的遍历深度提高 10 倍，同样取 19 个点进行拟合，将之与提高遍历深度之前的拟合图像进行比对，如图 6-5 所示：

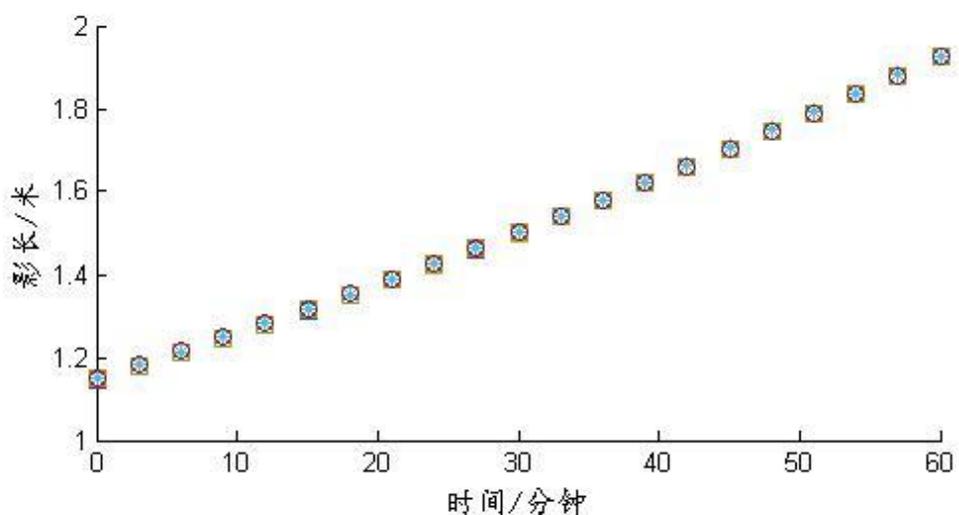


图6-5: 高遍历深度多点拟合图

其对应散点图 6-6 如下：

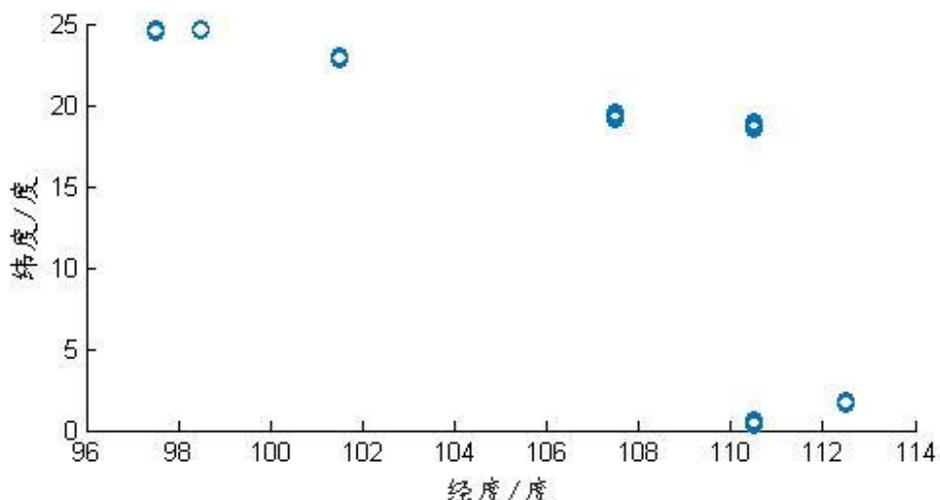


图 6-6: 高遍历深度可能点散点图

加大遍历深度后拟合曲线的拟合程度与散点图中对应可能点的分布趋势，与未加大遍历深度前相比，并无显著差异，仅仅在对应可能点处的聚集程度更加密集，且仍无法消除位于海上与国境外的可能点，且其时间开销为 4541 秒，大约为加深前的 51 倍，性价比低。

究其原因，在于地理公式均为近似公式，故一味提高遍历深度，并不能改善结果的精确程度，相反其效比较低，无益于问题的解决。本文使用的遍历深度可认为是一个合适值，运算与时间开销较为适宜，模型的改进点不在此。

6.3 根据太阳影子长度求可能的地点与日期

与第二问相比，问题三缺少了日期，(8) 式变成了一个四元函数。

通过观察数据可以发现，附件 2 中的影子是变短，而附件 3 中的影子是变长，由问题 1 可知影子在上午是随时间变短，下午随时间变长，假定太阳的赤纬角在同一天保持不变，则可将上午与下午的影子变化看做方向相反，路径相同的过程，从而考虑半天的变化便可知整天的变化，将计算量减少一半。

利用附件中给予的北京时间，以及影子坐标，我们同样可以计算出影长，由于没有给出当时测量的日期，而测量时间影响到了赤纬角及真太阳时的计算，于是我们在第二题的基础上遍历了所有日期，时间复杂度又一次加大。

由于时间复杂度增加，我们用穷举方法会遇到较大的难题，因此我们降低了纬度的遍历深度，同时提高数据的质量，要求每组最后一个理论点与实测点之间

也不能相差超过 0.02，因而减小了计算需要的时间，也得到较好的解。花费时间为第二问的 5-6 倍。算法流程如图 6-7 所示：

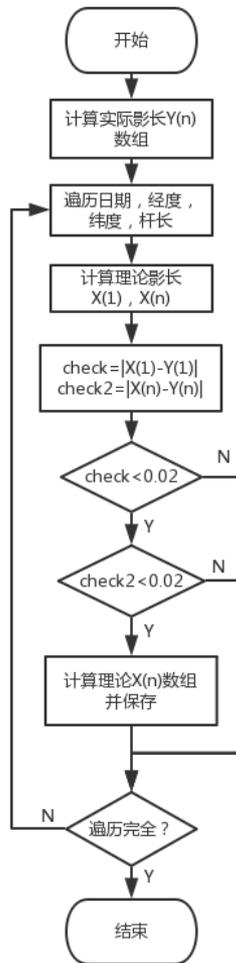


图 6-7: 算法流程图二

步骤一：

计算实测影子长度。

步骤二：

顺序遍历日期、经度、纬度、杆长，遍历深度分别为 1、1、1、0.1。

步骤三：

计算理论影长，对每组首位理论影长与实际影长进行对比，做差取绝对值，与阈值 0.02 对比。

步骤四：

计算理论影长，对每组末尾理论影长与实际影长进行对比，做差取绝对值，与阈值 0.02 对比。

步骤五：

小于阈值，则保留数据；否则，跳至步骤二。

步骤六：

检验是否遍历完经度、纬度、杆长，若遍历完全，则结束；否则，重复步骤二。

附件二与附件三的处理方法相同，下文的求解过程均以附件二为例。

通过遍历，并使用最小二乘法对这些粗略点进行筛检，可得 719 个点，其中理论上可能性最大的 8 个坐标如表 6-2 所示：

表 6-2

纬度	经度	日期	地点
35	63.75	107	西藏自治区阿里地区札达县
39	59.75	125	新疆喀什地区塔什库尔干塔吉克自治县
37	78.75	128	新疆喀什地区叶城县
39	58.75	129	新疆和田地区策勒县
42	52.75	148	新疆喀什地区巴楚县
39	81.75	206	新疆和田地区墨玉县
40	59.75	208	新疆喀什地区疏附县
39	58.75	223	新疆喀什地区塔什库尔干塔吉克自治县

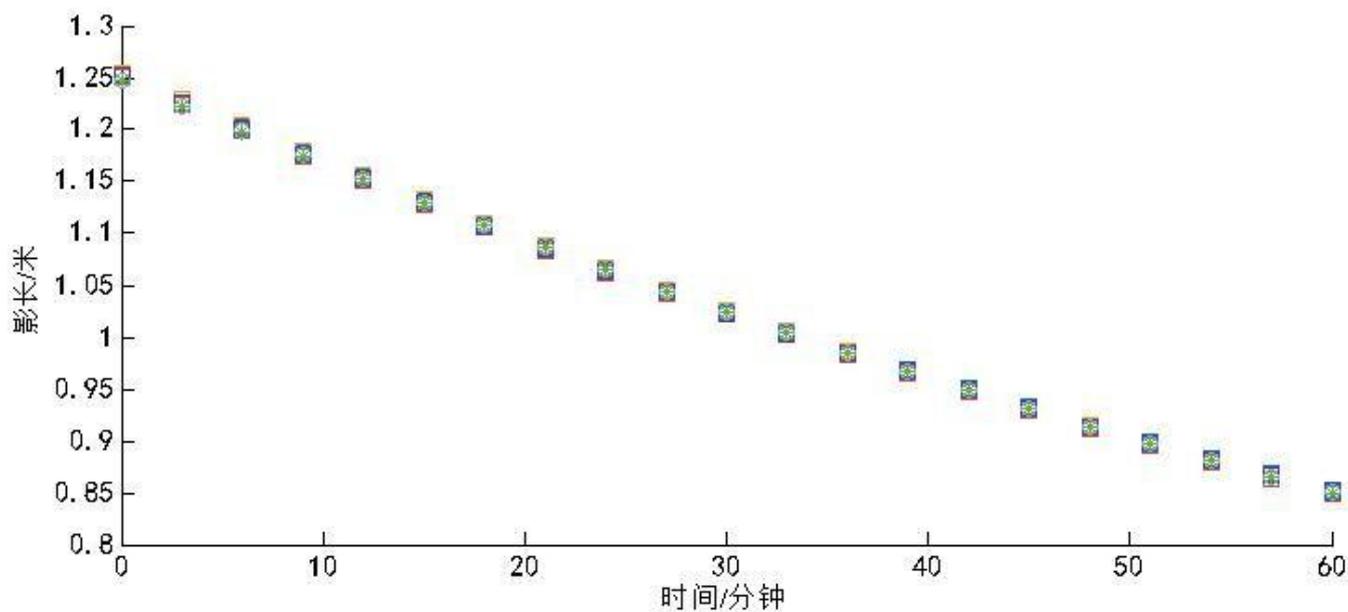


图 6-8：多点拟合图

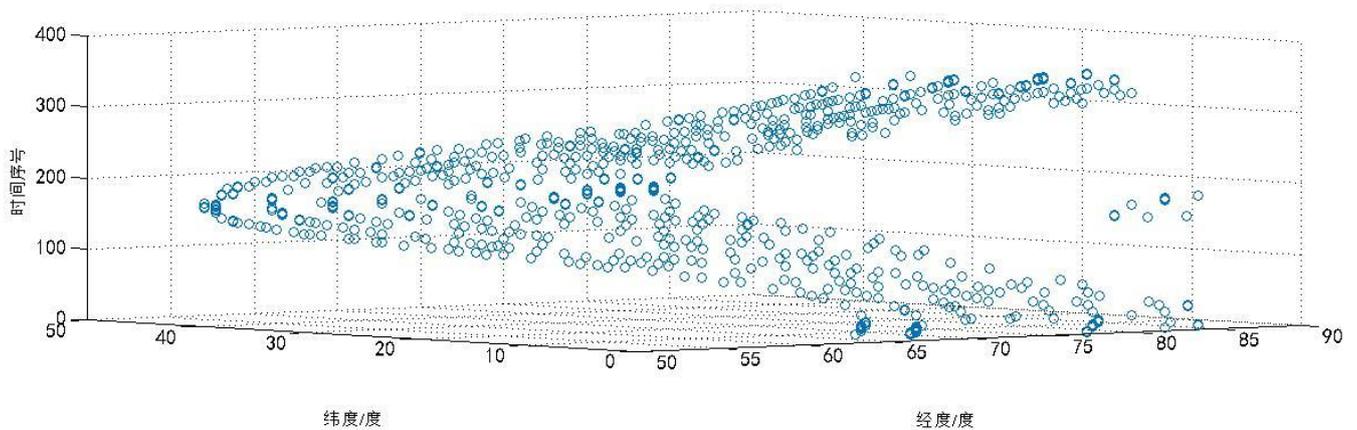


图 6-9 多点散点图

附件 3 中计算得到的可能的时空点，详见表 6-3：

表 6-3：附件 3 中可能的时空点

纬度	经度	日期	地方
36	110.75	70	山西省临汾市吉县
38	110.75	75	山西省吕梁市临县
42	109.75	90	内蒙古包头市达尔罕茂明安联合旗
42	107.75	250	内蒙古巴彦淖尔市乌拉特中旗
38	106.75	271	宁夏银川市灵武市
35	105.75	277	甘肃省天水市秦安县

6.4 根据太阳影子长度求视屏拍摄的地点与日期

本文采用的是通过影长与杆长的比例关系换算太阳高度角，从而获取测量点的时空信息的。由前三问可知，获取视频中影子的准确长度就可以确定若干个可能的时空点。第四问与前三问相比，计算位置与时间的原理没有差异，区别在于第四问需要将视频数据转化为数值数据，而视频中的影子的轴向变化不仅仅是太阳高度角造成的，视频本身存在的透视畸变也会导等长的影子在投影到二维图像上时出现距离的“缩短”，即第四问需要一个图形处理的步骤。

视频数据转化：

透视畸变是指图像使用二维的表现方式来表现将现实世界中的三维关系时，由于透视原理，两平行直线必在极远点相交，而两平行线是永不相交的，即图像出现了畸变，不能准确刻画物体的特征。

透视畸变在本题中的影响，是使视频中影子长度与其对应时刻的真实影子长度不一致。

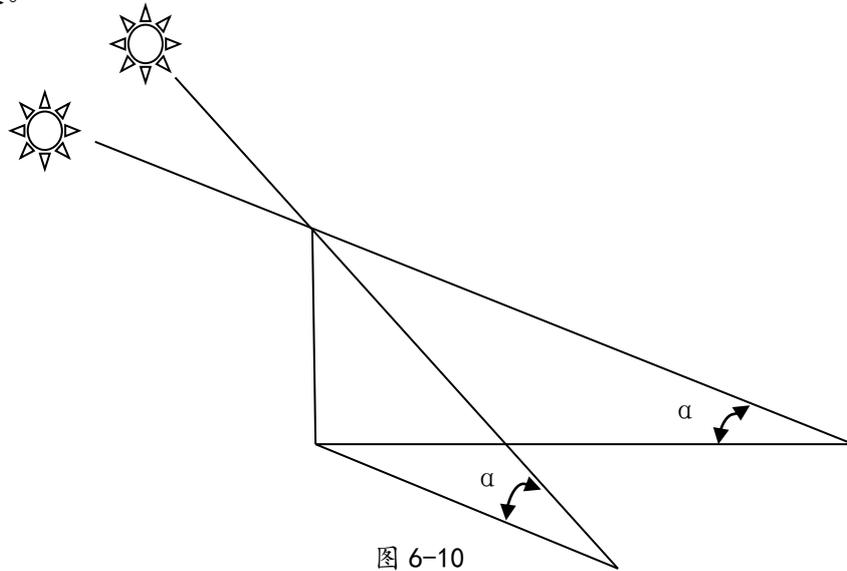


图 6-10

假设太阳高度角相同，则同一根直杆对应的影子长度应该相等，但由于透视畸变，如上图 x-x，转过不同角度时，影子在二维平面上的长度是不一样的。

为了能得到视频中影子的实际长度值，考虑通过逆透视变换法实现。逆透视变换指根据 2D 图像坐标来确定 3D 客观实物的坐标，或者说将一个图像点反过来映射回 3D 空间。但是查阅相关文献后发现，除非对映射到图像点的 3D 空间点有一些先验知识（如知道它的其中一个坐标值），否则不可能将一个 3D 空间的坐标从它的图像中完全恢复过来。或者说，要利用逆透视变换将 3D 空间点从其图像中恢复出来需要知道该点的至少一个真实坐标值^[4]，而题目及附件中并无相关数据。另外考虑到本文基于最小二乘原理建立的模型本身就存在一定的误差，过分追求高精度效果反而不太好，故可考虑用 2D 图像中影子的长度来近似代替影子的实际长度。

同时仔细观察视频，摄像机与直杆间还存在一定的倾斜角，即视频中的直杆与视频底框不是垂直的，大约逆时针倾斜了一定角度，在一定程度上，反向补偿了透视畸变造成的影子长度的缩短。

数据处理：

本文采用的方法是使用 *Photoshop* 截取特定帧的视频画面，使用 *Photoshop* 的“量尺”功能，选取直杆顶点像素块与直杆基座原点上的像素块，以这两点间的近似距离作为直杆的杆长。

由于图片中标杆的角度由于拍摄的时候存在倾斜，使用测距前先将整个截图逆时针旋转 1 度，可以减少人工估计的误差。测得标杆在图片中占用像素点约 684 个，测量时先测量影子像素，测得之后与标杆比较，按比例可以获得影子近似长度，以视频时间 8:55 作为开始时间，每 2 分钟测一次影子像素，共测 20 组，代入问题 2 的模型当中，可以计算出，当地纬度 41.9 度，经度 113.25 度。

第二小问则仅仅知道影子近似长度和测量时候的时差，未知当地时间，则可以利用第三问模型遍历 365 天的所有情况，以求得可能的地点和时间，结果如下：

表 6-4: 可能的时空点

纬度	经度	日期	地点
37	118.25	231	山东省淄博市
41	111.25	147	内蒙古自治区呼和浩特市
41	115.25	126	河北省张家口市
44	113.25	193	内蒙古自治区锡林郭勒盟
39	118.25	227	河北省唐山市

六. 模型的评价与改进

模型的优点：

(1) 模型简单实用，将最小二乘法与遍历算法相结合，通过巧妙设置阈值，增加了模型的求解速度。

(2) 先分析最直接影响影长的参数，后通过逆向思维层层推理得到几个影响影长并相互无关的参数，最后再正向思维列数学表达式建立模型，此种思维在解决看似复杂的问题方面值得推广。

模型的缺点与改进：

(1) 因测量方法等因素的影响，题目所给数据与实际影长会存在一定偏差，故直接利用题目所给数据而未进行数据预处理，模型求解结果会有一定误差。

(2) 问题四中直接利用 2D 图像中影长近似代替真实影长，估计对模型的求解正确率存在较大影响。因此可改进的方面有通过其他最新研究理论成果，或侧面处理等方法来得到视频中影长的真实长度并代入模型进行计算，以提高结果精确率。

七. 参考文献

- [1] 刘真，基于天文导航的无人机定位方法研究[D]，西安：西安电子科技大学，2009.9-17
- [2] 方荣生，太阳能应用技术，北京：中国农业机械出版社，1985
- [3] 司守奎,孙玺菁，数学建模算法与应用，北京：国防工业出版社，2011
- [4] 章毓晋，图像理解（下册），北京：清华大学出版社，2012

九. 附录

MATLAB 程序:

第一问程序:

```
dega=39;
degb=54;
degc=36;
height=3;
montharry=[31,28,31,30,31,30,31,31,30,31,30,31];
timemonth=10;
timeday=22;
timeorder=0;
timenow=9-4/60*(120-116-23/60-29/3600):1/60:15-4/60*(120-116-23/60-29/3600);
alpha=[];
for i=1:timemonth-1
timeorder=montharry(i)+timeorder;
end
timeorder=timeorder+timeday;
deg=(dega+degb/60+degc/3600)*pi/180;
sigma=23.45*sin(2*pi*(284+timeorder)/365)*pi/180;
timeomega=(15*(timenow+timesub(timeorder)/60-12))/180*pi;
alpha=asin((sin(deg)*sin(sigma)+cos(deg)*cos(sigma)*cos(timeomega)));
shadow=height./tan(alpha);
plot(9:1/60:15,shadow,'k');
xlabel('Ë±ä (h)');
ylabel('Ó³ (m)');
```

第二问程序:

```
tic;
Beijintime=14+42/60;
problem2length=sqrt(x.*x+y.*y);
problem2length=problem2length';
dega=39;
degb=54;
degc=36;
montharry=[31,28,31,30,31,30,31,31,30,31,30,31];
timemonth=4;
timeday=18;
timeorder=0;
```

```

for i=1:timemonth-1
timeorder=montharry(i)+timeorder;
end
timeorder=timeorder+timeday;
problem2shadowcollect=[];
problem2take=[];
for timecheck=12:4/60:18
timenow=timecheck:3/60:timecheck+3/60*20;
for height=0:0.1:8
for deg= (0:0.1:90)*pi/180
sigma=23.45*sin(2*pi*(284+timeorder)/365)*pi/180;
timeomega=(15*(timenow-12))/180*pi;
alpha=asin((sin(deg)*sin(sigma)+cos(deg)*cos(sigma)*cos(timeomega)));
if abs(height/tan(alpha(1))-problem2length(1))>0.02
    continue;
end
problem2shadowX=height./tan(alpha);
problem2shadowcollect=[problem2shadowcollect;problem2shadowX];
problem2take=[problem2take;deg/pi*180,height,timecheck,60*(timecheck-
Beijintime)/4+120,timeorder];
end
end
end
problem2check=inf;
findcheck=inf;
problem2nearbest=[];
for i=1:size(problem2shadowcollect,1)
problem2check=min(problem2check,sum(abs(problem2shadowcollect(i,:)-pr
oblem2length)));
if sum(abs(problem2shadowcollect(i,:)-problem2length))<=0.05
    problem2nearbest=[problem2nearbest;i];
end

if problem2check==sum(abs(problem2shadowcollect(i,:)-problem2length))
    findcheck=i;
end
end
problem2nearbestinfo=[];
for i=1:size(problem2nearbest,1)

problem2nearbestinfo=[problem2nearbestinfo;problem2take(problem2nearb
est(i),:)];
end

```

```

figure(1);
hold on;
for i=1:5
scatter(0:3:60,problem2shadowcollect(problem2nearbest(i),:),'s')
end
scatter(0:3:60,problem2shadowcollect(findcheck,:),'*');
scatter(0:3:60,problem2length);
figure(2);
scatter(problem2nearbestinfo(:,4),problem2nearbestinfo(:,1));
toc;

```

第三问程序:

```

tic;
Beijintime=13+9/60;
dega=39;
degb=54;
degc=36;
montharry=[31,28,31,30,31,30,31,31,30,31,30,31];
timemonth=4;
timeday=18;
timeorder=0;
problem2length=sqrt(x.*x+y.*y);
problem2length=problem2length';

problem2shadowcollect=[];
problem2take=[];
for timeorder =1:1:365
for timecheck=12:4/60:18
timenow=timecheck+9/60:3/60:timecheck+1+9/60;
timenow=timecheck:3/60:timecheck+3/60*20;
for height=4:0.1:6
for deg= (0:1:90)*pi/180
sigma=23.45*sin(2*pi*(284+timeorder)/365)*pi/180;
timeomega=(15*(timenow + timesub(timeorder)/60 - 12))/180*pi;
alpha=asin((sin(deg)*sin(sigma)+cos(deg)*cos(sigma)*cos(timeomega)));
if abs(height/tan(alpha(1))-problem2length(1))>0.02
continue;
end
if abs(height/tan(alpha(21))-problem2length(21))>0.02
continue;
end
problem2shadowX=height./tan(alpha);
problem2shadowcollect=[problem2shadowcollect;problem2shadowX];
problem2take=[problem2take;deg/pi*180,height,timecheck,60*(timecheck-

```

```

Beijintime)/4+120,timeorder];
end
end
end
end

problem2check=inf;
findcheck=inf;
problem2nearbest=[];
for i=1:size(problem2shadowcollect,1)
problem2check=min(problem2check,sum(abs(problem2shadowcollect(i,:)-problem2length)));
if sum(abs(problem2shadowcollect(i,:)-problem2length))<=0.05

problem2nearbest=[problem2nearbest;i,sum(abs(problem2shadowcollect(i,:)-problem2length))];
end
if problem2check==sum(abs(problem2shadowcollect(i,:)-problem2length))
    findcheck=i;
end
end
problem2nearbestinfo=[]
for i=1:size(problem2nearbest,1)

problem2nearbestinfo=[problem2nearbestinfo;problem2take(problem2nearbest(i,1,:),:),problem2nearbest(i,2)];
end
figure(1);
hold on;
for i=1:5
scatter(0:3:60,problem2shadowcollect(problem2nearbest(i,:),:),'s');
end
scatter(0:3:60,problem2shadowcollect(findcheck,:),'*');
scatter(0:3:60,problem2length);
figure(2);
scatter(problem2nearbestinfo(:,4),problem2nearbestinfo(:,1));
figure(3);

scatter3(problem2nearbestinfo(:,4),problem2nearbestinfo(:,1),problem2nearbestinfo(:,5))
toc;

```

第四问程序：

```
tic;
```

```

Beijintime=8+55/60;
dega=39;
degb=54;
degc=36;
%height=3;
montharry=[31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31];
timemonth=7;
timeday=13;
timeorder=0;
problem2length=xx;
problem2length=problem2length';
for i=1:timemonth-1
timeorder=montharry(i)+timeorder;
end
timeorder=timeorder+timeday;%           天数序号
%deg =(dega+degb/60+degc/3600)*pi/180; % 当地纬度
problem2shadowcollect=[];
problem2take=[];
%for timeorder =1:1:365
for timecheck=6:4/60:12
%timenow=timecheck+55/60+6/60:2/60:timecheck+1+33/6
0+6/60;

```

```

timenow=timecheck:2/60:timecheck+2/60*19;%%加上真太阳时 和 平太阳时时差
for height=684
for deg= (0:0.1:90)*pi/180
sigma=23.45*sin(2*pi*(284+timeorder)/365)*pi/180; %
赤纬角 -为南半球纬度
timeomega=(15*(timenow+timesub(timeorder)/60-12))/180*pi; % 时角
alpha=asin((sin(deg)*sin(sigma)+cos(deg)*cos(sigma)*cos(timeomega))); % 高度角
if
abs(height/tan(alpha(1))-problem2length(1))/342>0.2
    continue;
end
if
abs(height/tan(alpha(20))-problem2length(20))/342>0.2
    continue;
end

problem2shadowX=height./tan(alpha);
problem2shadowcollect=[problem2shadowcollect;proble

```

```

m2shadowX];
problem2take=[problem2take;deg/pi*180,height,timecheck,60*(timecheck-Beijintime)/4+120,timeorder];
%scatter(timenow,problem2shadowX);
%hold on
end
end
end
%end

```

```

problem2check=inf;
findcheck=inf;
problem2nearbest=[];
for i=1:size(problem2shadowcollect,1)
problem2check=min(problem2check,sum(abs(problem2shadowcollect(i,:)-problem2length))/342);

```

```

if
sum(abs(problem2shadowcollect(i,:)-problem2length)/342)<=0.25

```

```

problem2nearbest=[problem2nearbest;i,sum(abs(proble

```

```

m2shadowcollect(i, :)-problem2length))];
end

if
problem2check==sum(abs(problem2shadowcollect(i, :)-p
roblem2length)/342)
    findcheck=i;
end
end

problem2nearbestinfo=[];
for i=1:size(problem2nearbest, 1)

problem2nearbestinfo=[problem2nearbestinfo;problem2
take(problem2nearbest(i), :), problem2nearbest(i, 2)];
end

figure(1);
hold on
scatter(0:2:38, problem2shadowcollect(findcheck, :)/3
42, '*');
scatter(0:2:38, problem2length/342);

```

```
for i=1:4
scatter(0:2:38, problem2shadowcollect(problem2nearbest(i), :)/342, 's');
end
figure(2);
scatter(problem2nearbestinfo(:, 4), problem2nearbestinfo(:, 1));

toc;
```

未知日期情况:

```
tic;
Beijintime=8+55/60;
dega=39;
degb=54;
degc=36;
%height=3;
montharry=[31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31];
timemonth=7;
timeday=13;
timeorder=0;
```

```

problem2length=xx;
problem2length=problem2length';
for i=1:timemonth-1
timeorder=montharry(i)+timeorder;
end
timeorder=timeorder+timeday;%          天数序号
%deg =(dega+degb/60+degc/3600)*pi/180; % 当地纬度
problem2shadowcollect=[];
problem2take=[];
for timeorder =1:1:365
for timecheck=6:4/60:12
%timenow=timecheck+55/60+6/60:2/60:timecheck+1+33/6
0+6/60;
timenow=timecheck:2/60:timecheck+2/60*19;%%加上真太
阳时 和 平太阳时时差
for height=684
for deg= (0:1:90)*pi/180
sigma=23.45*sin(2*pi*(284+timeorder)/365)*pi/180; %
赤纬角 -为南半球纬度
timeomega=(15*(timenow+timesub(timeorder)/60-12))/1
80*pi;          % 时角
alpha=asin((sin(deg)*sin(sigma)+cos(deg)*cos(sigma)

```

```

*cos(timeomega)); % 高度角
if
abs(height/tan(alpha(1))-problem2length(1))/342>0.2
    continue;
end
if
abs(height/tan(alpha(20))-problem2length(20))/342>0
.2
    continue;
end

problem2shadowX=height./tan(alpha);
problem2shadowcollect=[problem2shadowcollect;problem
m2shadowX];
problem2take=[problem2take;deg/pi*180,height,timech
eck,60*(timecheck-Beijintime)/4+120,timeorder];
%scatter(timenow,problem2shadowX);
%hold on
end
end
end
end

```

```

problem2check=inf;
findcheck=inf;
problem2nearbest=[];
for i=1:size(problem2shadowcollect,1)
problem2check=min(problem2check, sum(abs(problem2sha
dowcollect(i, :)-problem2length))/342);

if
sum(abs(problem2shadowcollect(i, :)-problem2length)/
342)<=0.2

problem2nearbest=[problem2nearbest;i, sum(abs(proble
m2shadowcollect(i, :)-problem2length))];
end

if
problem2check==sum(abs(problem2shadowcollect(i, :)-p
roblem2length)/342)
    findcheck=i;
end
end

```

```

problem2nearbestinfo=[];
for i=1:size(problem2nearbest,1)

problem2nearbestinfo=[problem2nearbestinfo;problem2
take(problem2nearbest(i,:),problem2nearbest(i,2)];
end

figure(1);
hold on
scatter(0:2:38,problem2shadowcollect(findcheck,+)/3
42,'*');
scatter(0:2:38,problem2length/342);

figure(2);
scatter(problem2nearbestinfo(:,4),problem2nearbesti
nfo(:,1));
figure(3);

scatter3(problem2nearbestinfo(:,4),problem2nearbest
info(:,1),problem2nearbestinfo(:,5));
toc;

```

