太阳影子定位

摘要

当今社会摄像技术越来越普及，视频成为一种信息数据载体，这使得视频数据分析尤为重要。以视频中太阳影子长度变化为背景提出了四个问题，本文运用地理相关知识、网格搜素算法等成功地解决了这四个问题，得到了影子长度的变化规律图以及在给定条件下如何得到直杆的地点和日期。

针对问题一，根据太阳成影图（见图 1），确定直杆影子长度与太阳高度角的关系。接着，根据太阳高度角与日期、时刻、经度、纬度的关系，建立了影长的数学模型。接着用 VC6.0 和 MATLAB 软件进行编程，控制变量绘制影子长度关于各个参数的变化规律图（见图 2-1 至图 2-6）。然后利用所给模型绘制出 3 米高直杆在 2015 年 10 月 22 日北京时间 9:00-15:00 天安门广场的太阳影子长度的的变化曲线（见图 3）。

针对问题二，建立直杆影长与太阳高度角、日期、时间和经纬度的数学模型。在日期、时间和影子顶点坐标已知情况下，考虑直杆所处的地点。首先，根据顶点的坐标我们可以计算出影长。然后，取定一定的杆高范围，采用全局和局部网格搜索，我们得到了可能地点（东经109.27°，北纬18.88°）（即海南内部偏西南）（见表 1）。

对于问题三，与问题二不同的是，问题三在时间和影子顶点坐标已知情况下，考虑直杆所处的地点和日期。由于问题三中日期未知，我们采用类似问题二的网格搜索算法，同时添加日期的搜索，逐步筛选出来可能地点（附件 2:东经80.31°，北纬 41.6°，新疆阿克苏以北，附件 3: 东经110.63°，北纬31.81°，湖北十堰；东经110.35°，北纬 29.05°，湖南张家界），（见表 2，表 3）。

针对问题四，在杆高和视频拍摄年份、日期、时间已知情况下，确定直杆所处的地点。首先，我们从 40 分钟的视频中等间隔地截取 20 张图片，利用网格划分建立坐标系确定各个时刻影子顶点的坐标，然后计算图片中影长，最后利用成像原理得到实际影长。最后，采用全局和局部网格搜索，逐步筛选出可能地点（东经111.26°，北纬 40.71°），即可能的地点（呼和浩特）。如果拍摄日期未知，我们也可以根据视频确定出拍摄地点与日期（结果见 15 页）。

本文综合地理相关知识、网格搜索算法，结合 VC6.0 和 MATLAB 软件，对太阳影子定位进行了多角度分析。为验证模型的稳定性，以附录 1 中的数据为例，对 x 值、y 值（即初值数据）进行不同程度的扰动，求出在扰动下的地点并与问题二的答案进行对比，观察最终结果中经纬度和精度的差异，定量说明模型的稳定性。在文章的最后，对所建立的模型和算法的优缺点给出了客观的评价，同时给出需要改进的方法。

关键词：影子定位 太阳高度角 网格搜索 均方误差

一、 问题重述

1.1 引言

视频信息的处理、图像的定位是信息挖掘的难点，但具有很重要的实际意义。确定视频的拍摄地点与日期可以通过太阳下影子长度变化来进行分析。通过考察太阳影子长度关于一些参数的变化规律，建立一个相应的数学模型，然后根据所测视频中太阳影子的长度，利用所建太阳影子长度数学模型，确定视频的拍摄地点和日期。视频的拍摄地点和日期的估计精度高低与所测影子的长度精准度有直接的关系。

1.2 问题的提出

围绕影子长度的变化规律，建立影子长度变化数学模型，本文依次提出如下问题：

（1）给出影子长度关于各个参数的变化规律，要求建立关于影子长度变化的数学模型。给定 3 米高的直杆在具体日期（2015 年 10 月 22 日）、时间段（北京时间 9:00-15:00）、地点（天安门广场，北纬：39 度 54 分 26 秒,东经：116 度 23 分 29 秒）。要求根据所建模型画出此直杆的太阳影子长度变化图。

（2）对于给定的太阳影子顶点坐标，建立数学模型讨论固定直杆所在经纬度，即地点。对于给定附件 1，通过其所给影子顶点坐标和所建立模型确定固定直杆可能的经纬度。

（3）给定直杆太阳影子顶点坐标，给出数学模型讨论其固定直杆的拍摄日期和地点。对于给定附件 2 和附件 3，利用其所建模型，给出固定直杆的拍摄日期和地点。

（4）给定直杆高度和其直杆的太阳影子的变化视频（附件 4）。讨论其视频的拍摄地点的数学模型。根据所建数学模型给出一些可能的拍摄地点。假设拍摄日期不知道，讨论视频拍摄的日期和地点。

二、模型假设

（1）假设太阳光为平行光线
（2）假设测量结果无误差
（3）杆不随温度、湿度等因素发生形变
（4）假设地面平整
（5）不考虑闰平年
（6）不考虑大气层、温度、湿度等因素对光线传播产生的影响

三、符号说明

l	影子长度
h	杆长
α	太阳高度角
δ	太阳赤纬
φ	测点纬度
ϖ	太阳时角
θ	日角
N	积日
Y	年份
p	经度
T	日期
TT	真太阳时
Δt	时差
t	北京时间

四、问题解答

4.1 问题一

4.1.1 问题分析

问题一要求给出物体的太阳影子长度变化的数学模型，并根据数学模型确定各个参数对影子长度的影响，另外利用所建数学模型对一个物体（给定时间、地点和物体高度）画出其影子长度变化的变化图。我们可以从时间，日期，经纬度，物体的高度等参数对来建立影子长度变化的数学模型。通过控制部分参数来研究某一参数对影子长度的影响。最后，将问题所给相关数据代入所建数学模型求出影子长度的变化图。

4.1.2 模型建立

为了帮助理解，我们做直杆成影图示意图[1]，如图 1 所示，其中 A 点为杆顶成影点，规定西向东方向为正方向，东西走向为 x 轴，南北方向为正方向，南北走向为 y 轴。直杆的顶点投影为点 $B(x,y)$，C 点为直杆的顶点。α 为太阳高度角，标杆长度为 $|AB|$，令 $|AB|=h$。

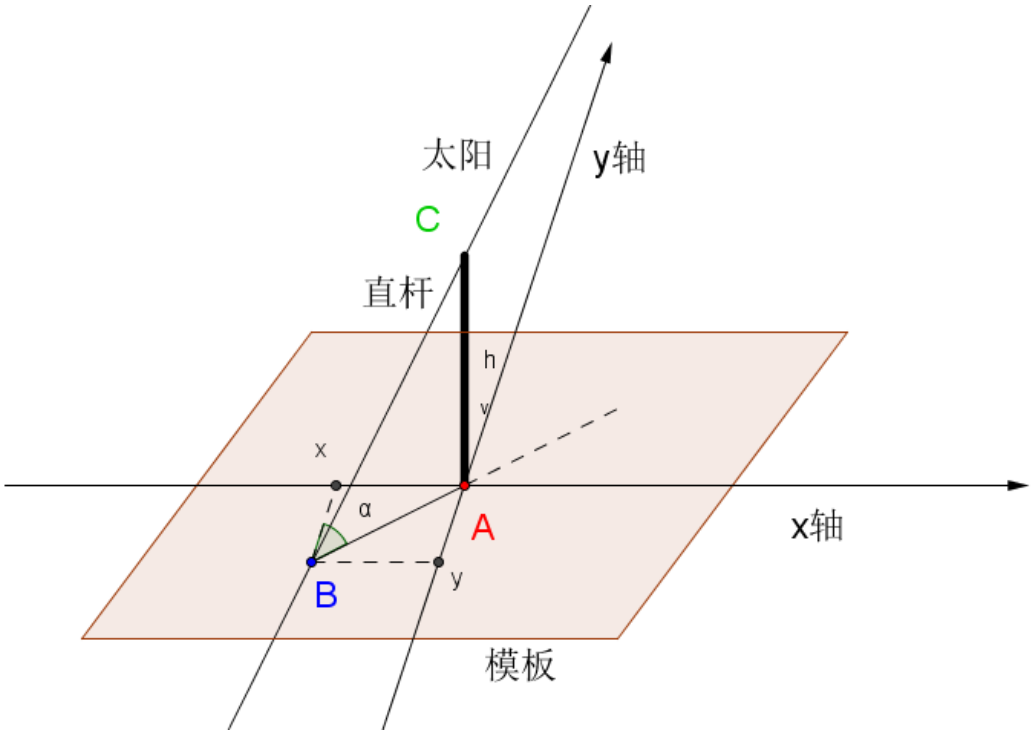


图 1 直杆成影图

从几何关系上我们有：

$$l=\frac{h}{\tan\alpha} \tag{1}$$

根据参考文献[2,3]，太阳高度角 α 的计算公式为：

$$\sin\alpha=\sin\delta\sin\varphi+\cos\delta\cos\varphi\cos\varpi \tag{2}$$

其中 δ 为太阳赤纬，φ 为测点纬度，ϖ 为太阳时角。而太阳赤纬 δ [4,5] 计算公式如下：

$$\begin{aligned}\delta=&0.3723+23.2567\sin\theta+0.1149\sin2\theta-0.1712\sin3\theta\\&-0.7580\cos\theta+0.3656\cos2\theta+0.0201\cos3\theta\end{aligned} \tag{3}$$

由文献[6]，日角 θ 公式为：

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

其中，积日 N 为日数，自该年的 1 月 1 日开始计算。若年份为 Y，则 $N_0 = 79.6764 + 0.2422*(Y-1985) - [\frac{Y-1985}{4}]$，其中 $[\bullet]$ 代表取整。

由文献[5]，太阳时角 ϖ 为：

$$\varpi = 15*(TT-12) \tag{5}$$

其中 TT 为真太阳时，在中国地区，北京时间为 t，时差 Δt，经度 p 东经为正，西经为负，则真太阳时的换算公式为：

$$TT = t + \Delta t, \qquad \Delta t = \frac{p-120^\circ}{15^\circ} \tag{6}$$

4.1.3 问题解答

（1）直杆影子长度变化的数学模型

综合以上分析，直杆影长变化的数学模型如下：

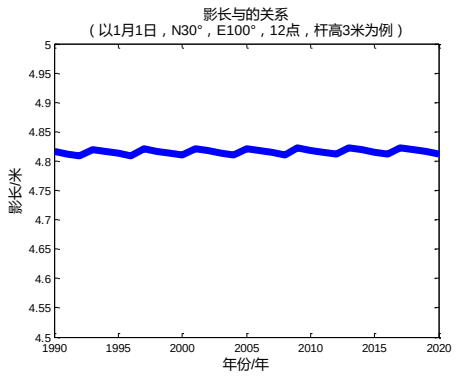
$$\begin{aligned} l &= \frac{h}{\tan\alpha} \\ &= h\cot\left(\arcsin(\sin\delta\sin\varphi + \cos\delta\cos\varphi\cos\varpi)\right) \end{aligned} \tag{7}$$

其中 $h,\alpha,\delta,\varphi,\varpi$ 分别表示直杆的长度、太阳高度角、太阳赤纬、测点纬度和太阳时角。

（2）直杆影子长度关于各个参数的变化规律

为描述直杆影长关于各个参数的变化规律，根据以上建立的模型（见式(7)），运用 MATLAB 软件（程序参见附录 problem1-3.m），分别绘制影长关于各变量示意图。

影长与年份的关系（见图 2-1）。从图 2-1 可知，在 2000 年至 2100 年共 100 年的变化时间中，影子的长度在大约在 13.37m 至 13.13m 之间变化，可视为在足够长的时间中只发生了细微的变化，即不同年份的同一天同一时刻同一直杆在相同地点所获得的影长是近似相等的，这也与我们的实际生活体验相符合。



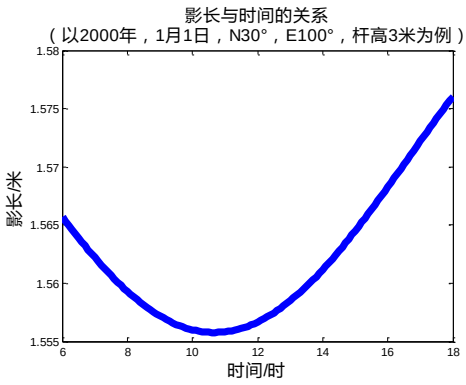
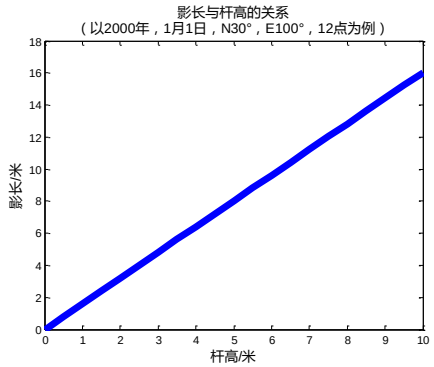


图 2-1 影长与年份关系图　　**图 2-2 影长与日期关系图**

影长与日期的变化（见图 2-2）。从图 2-2 可知，在 2000 年不同日期的影子长度变化曲线在 1 月 1 日长度较长，该日期假设点为冬季，随日期的增加，影长逐渐变短，在 150 天至 200 天区间内获得最短值，此时日期在 5 月只 7 月之间，该假设点为夏季。

影长与杆高的变化关系（见图 2-3）。从图 2-3 可知，影长与杆高成正比例关系，当杆长的长度越长时获得的影长的长度越长。



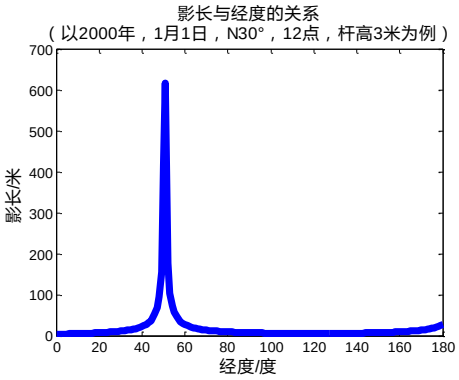


图 2-3 影长与杆长关系图　　**图 2-4 影长与经度关系图**

影长与经度的变化关系（见图 2-4）。从图 2-4 可知，峰值点的横坐标是当地地方时为 12 时位置的经度值，从该位置向东西方向改变经度值，影长均呈减小状态，当向左右方向扩展超过一定程度时影长为 0，即此时的位置点为黑夜。

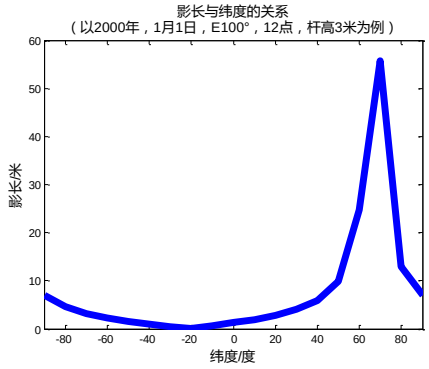


图 2-5 影长与纬度关系图

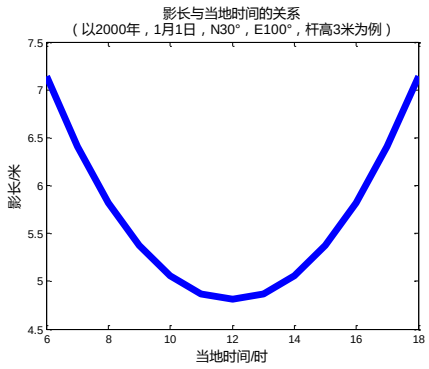


图 2-6 影长与时间关系图

影长与纬度的变化关系（见 2-5）。从图 2-5 可知，当其他条件一定是，纬度为 0 的地点即赤道地区影长较短，随着纬度的增高影长的长度逐渐增大，达到峰值后又逐渐减小。

影长与时间的变化关系（见图 2-6）。从图 2-6 可知，对某一地点，直杆的影长值在当地时间为 12 时时获得最小值，时间到达当地时间 12 时之前影长随时间的增大而减小，在当地时间 12 时之后影长随时间的增大而增大。

③直杆影子长度的变化规律

根据我们建立的模型（见公式（7）），将日期T：2015 年 10 月 22 日，时间t，t的取值范围为 9:00 至 15:00，北纬 39 度 54 分 26 秒,东经 116 度 23 分 29 秒，杆长$h=3m$代入公式(7)，运用 MATLAB 软件，编写程序（详见附 problem1-3.cpp），得到影子长度的变化规律，如图 3 所示

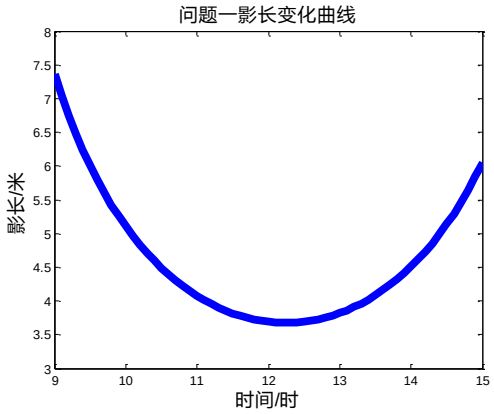


图 3 影长变化曲线图

从上图我们可以看到影子长度在 12 点至 13 点之间的某时刻t_0取得最小值，从 9:00 到t_0影子的长度逐渐减小，从t_0时刻到 15:00 影子的长度逐渐增大。影长的取值范围在 3.5m 至 7.5m 之间。

4.2 问题二

4.2.1 问题分析

问题二要求我们根据已知直杆太阳影子顶点坐标数据，给出直杆所处地点的数学模型，并根据数学模型给出一个物体（已知影子长度、时间、日期）的可能地点。我们可以利用所给影长的数学模型和网格搜索算法，求出直杆可能的地点。

4.2.2 模型建立

根据直杆物体太阳影子顶点坐标(x,y)，建立数学模型如下：

$$\begin{aligned}\sqrt{x^2+y^2} &= \frac{h}{\tan\alpha} \\ &= h\cot\left(\arcsin(\sin\delta\sin\varphi+\cos\delta\cos\varphi\cos\varpi)\right)\end{aligned} \tag{8}$$

其中$h,\alpha,\delta,\varphi,\varpi$分别表示直杆的长度、太阳高度角、太阳赤纬、测点纬度和太阳时角。

4.2.3 模型的解决

（1）解决方法

下面利用网格搜索算法[7,8]，寻找直杆可能的地点。将整个地球表面按照经纬网划分，将东西经方向按某一适当步长b_0划分，南北纬方向同理划分，取网格交点进行直杆影长的计算，如图 4 所示。

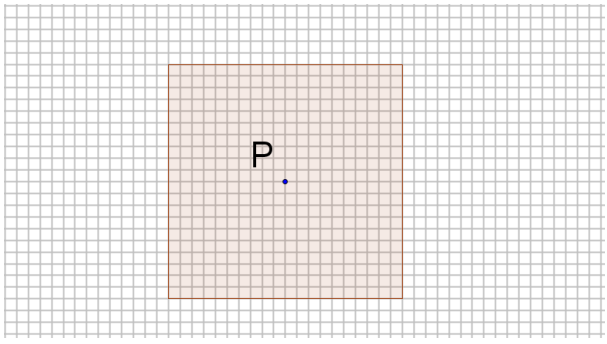


图 4 地表经纬网格划分图

由于杆长未知，此处根据实际情况假设杆长h的长度范围为[1,6]m，设定的最小长度为h_s，最大长度为h_e，杆长增量$\Delta h=0.1m$。

由附件 1 可得直杆的太阳影长的测量时刻t的取值范围为 14:42 至 15:42，起始时间为t_s，结束时间为t_e，以$\Delta t=3\min$为增量，共有 21 个时刻点。

对P点，影长在t时刻，杆长为h时的影长为l_{tph}，影长的测量日期T为 2015 年 4 月 18 日。

首次网格划分步长$b_0=0.2°$，二次划分步长$b_1=0.01°$。首次筛选出的网格交

点数为$n_0=20$，二次筛选的网格交点数为$n_1=3$。

取网格交点P，对所有的网格交点依次完成如下搜索步骤：

步骤 1：设定初始计算杆长$h=h_s$，初始计算时间$t=t_s$，n_0组均方差Δl初始值为无穷大。

步骤 2：利用问题一的影子长度计算模型计算出P点在日期T，时刻t,杆长为h的影长值l_{tph}。

步骤 3：当$t<t_e$时，$t=t+\Delta t$，进行步骤 2。

步骤 4：根据附件 1 所给的影子坐标(x,y)易得此时所需要确定地点的实际影长值为l_{t0}。利用公式（9）计算实际影长与P点计算所得的影长值l_{tph}之间的方差和Δl：

$$\Delta l=\sum_{t=t_s}^{t_e}(l_{tph}-l_{t0})^2 \qquad (9)$$

步骤 5：将P点计算的方差和Δl与n_0组较小方差和进行比较，若小于其中任一值则进行替换。

步骤 6：当$h<h_e$时，$h=h+\Delta h$，转到步骤 2。

步骤 7：筛选出n_0组方差和最小的网格交点的经纬度、杆长h以及此时的方差和Δl。

对所有的网格交点进行以上计算步骤后，再从筛选出的n_0组方差和最小的网格交点选取P'，东西南北方向各扩展步长b_0，获得以P'为中心点的正方形，以步长b_1再次划分网格，杆长h固定为已筛选出的数值，再次进行相似的搜索，筛选出n_1个可能地点。

（2）问题解答

根据以上建立的数学模型（见公式（8）），利用网格搜索算法，运用 C 语言编写程序（详见附录 problem2.cpp），我们可以得到直杆可能所在地点的经纬度和杆长，如表 1 所示。

由于我们以均方差作为该点是否为正确地点的判断标准，且均方差可以计算到的最小数量级为10^{-7}，此时对应的可能地点为海南，其坐标为北纬$18.88°$,东

经109.27°，杆长为 2m，其均方误差为$5.39035\times10^{-7}m^2$。

北纬/°	东经/°	杆长/m	均方差/m^2	可能的地点
18.88	109.27	2	5.39035×10^{-7}	海南内部西南
21.1	106.08	2.2	6.89253×10^{-6}	越南
24.79	97.71	2.8	1.03396×10^{-5}	云南边境
-3.14	103.56	2.2	1.1392×10^{-5}	印尼
22.04	104.54	2.3	1.22212×10^{-5}	越南

表 1 附件 1 最有可能的地点

对附件 1 测量地点不同时刻的影长值与上面的 2 个地点做曲线拟合如下图 5 所示：

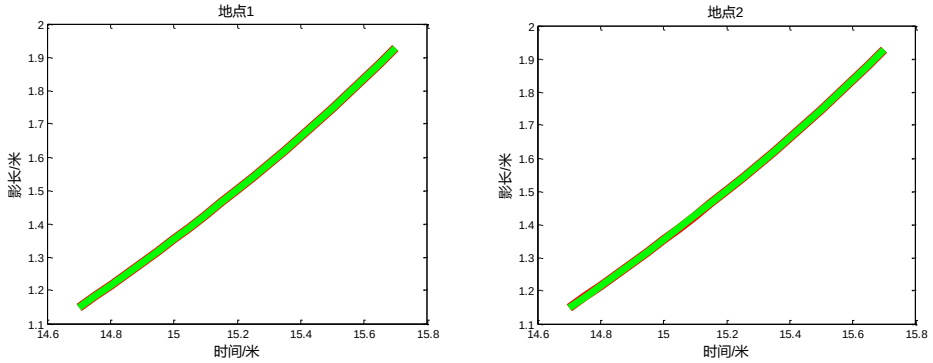


图 5　不同时刻影长值与计算地点曲线拟合

注：红色曲线为实际影长值，绿色曲线为计算得到的地点影长值的拟合曲线。

根据图 5 我们可以看到实际曲线与计算曲线的拟合程度极高，在一定程度上显示了计算结果的误差很小，精度很高。

4.3 问题三

4.3.1 问题分析

问题三要求我们建立已知直杆太阳影子顶点坐标数据，求物体可能所处的地点和测量日期的数学模型，利用数学模型求一个物体（给定影子长度、时间）可能的地点和测量日期。我们可以将日期作为新的搜索参量，改进并利用问题二建立的模型得到可能的地点和测量日期。

4.3.2 模型的解决

（1）解决方法

如问题二建立数学模型（见公式（8）），按图 4 所示，建立相同的网格，杆长、测量时刻，影长、网格步长的假设条件及参数保持不变。

首次筛选出的网格交点数为$n_0=10$，二次筛选的网格交点数为$n_1=3$。

影长的测量日期为T，T的取值范围从Ts：1 月 1 日第 1 天起，至T_e:12 月 31 日 365 天止，以$\Delta T=1d$为增量。

取网格交点P，对所有的网格交点依次执行如下步骤：

步骤 1：参照问题二建立的模型，设定初始测量日期$T=T_s$。

步骤 2：依次完成问题二中步骤 1 至 6。

步骤 3：当$T < T_e$时，$T = T + \Delta T$，进行步骤 2。

步骤 4：筛选出n_0组方差和最小的网格交点的经纬度、杆长h以及此时的方差和Δl。

同问题二一致，从筛选出的n_0组均方差最小的网格交点选取P'，东西南北方向各扩展步长b_0，获得以P'为中心点的正方形，以步长b_1再次划分网格，再次进行相似的搜索，但此时不同的是杆长h固定为已筛选出的数值，筛选出n_1个可能地点。

（2）问题解答

根据建立的模型，利用网格搜索算法，参照附件 2 数据，运用 C 语言编写程序（详见附件 problem3-1.cpp, problem3-2.cpp），我们可以得到直杆所在地的经度、纬度、杆长、日期，如表 2 所示。

北纬/°	东经/°	杆长/m	日期/d	均方差/m^2	可能地点
41.6	80.31	2.1	182	9.40126×10^{-8}	新疆阿克苏北
38.22	79.01	1.9	212	1.46879×10^{-7}	新疆和田西北
36.15	78.36	1.8	123	3.07599×10^{-7}	新疆西

表 2　附件 2 最有可能的地点

由于我们采用均方误差作为精确度的判断标准，当均方误差值的数量级越小时，获得的结果越精确，表格 2 中列出的数据中，最小数量级为10^{-7}，另外两组计算得到的数量级均大于该值，则最小均方差对应的地点最有可能为正确地点，我们可以判断最有可能的地点为新疆，其经纬度为北纬$41.6°$，东经$80.31°$，杆长为 2.1m，测量日期为第 182 天。

根据附件 3 数据，利用建立的模型进行网格搜索计算，得到可能地点的相应数据如下表 3 所示：

北纬/°	东经/°	杆长/m	日期/d	均方差/m^2	可能地点
31.81	110.63	2.7	20	4.95806×10^{-7}	湖北十堰
29.05	110.35	2.8	337	1.14501×10^{-6}	张家界
36.95	109.73	3.6	62	1.29161×10^{-6}	延安
33.52	111	2.3	344	1.95678×10^{-6}	河南陕西边界

表 3 附件 3 最有可能的地点

根据最小均方误差值我们可以判断此时最有可能的地点为湖北十堰，具体坐标为北纬$31.81°$，东经$110.63°$，杆的长度为 2.7m，测量的日期为第 20 天。

4.4 问题四

4.4.1 问题分析

问题四要求给出已知物体影子变化规律、测量时间、测量日期、物体长度情况下，该物体可能的地点的数学模型。我们可以先通过影子变化规律求的影子长度，利用并改进问题二建立的数学模型来实现求解可能的地点以及可能的日期。

4.4.2 模型建立

问题四建立的模型与问题二相同（见公式（8）），从视频中获取直杆的实际影长值后，再次利用网格搜索算法，求出可能的地点和日期。

4.4.3 模型的解决

（1）解决方法

①影长的计算

首先我们对视频进行抓帧处理，在 40 分钟的视频中每隔 2 分钟截取一张图片，一共选取 20 张图片作为样本。对于每一张图片，我们对其进行网格划分，在画图软件中得出影子顶点的像素坐标。

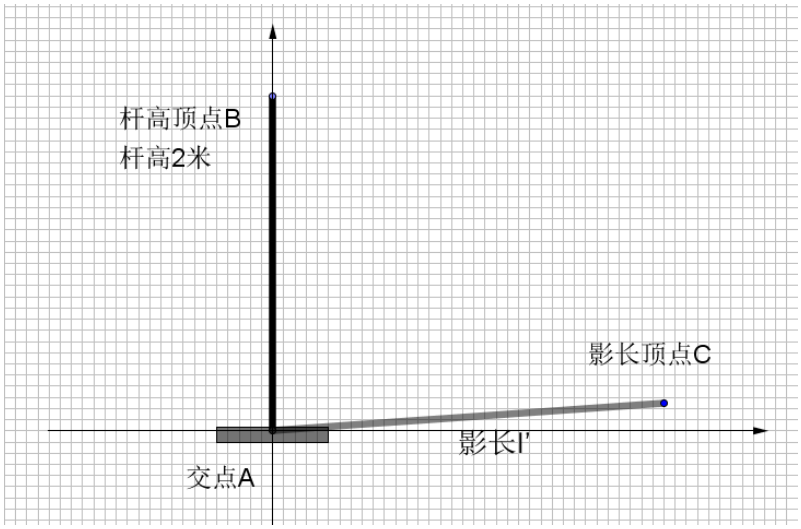


图 6　视频截图拟图 1

由图 6 所示，通过顶点影长 C 与交点 A 的坐标我们可以求出影长 l'。在实际图像中，由于存在底座导致交点 A 并不明显，准确选取几何意义上的交点 A 是准确衡量影长的关键之一。我们选取图片，在影长图像上手动地选取 9 个点，得出坐标数据，利用 MATLAB 中的 polyfit 函数拟合出一条直线，使之与竖直的杆相交于一点，视这个点即为图例中的交点 A。

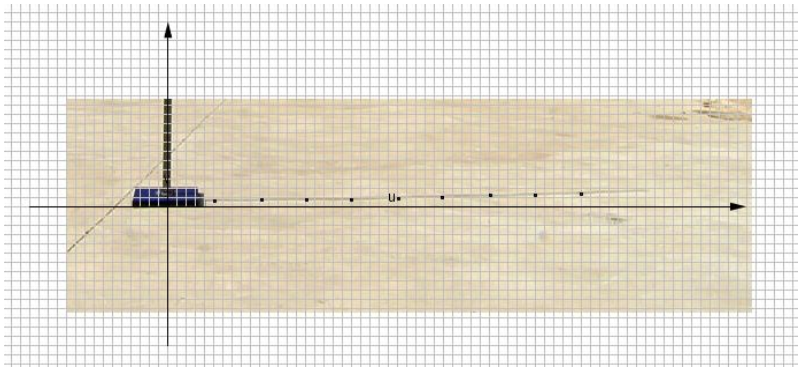


图 7 视频截图

这样在二维坐标系中我们就可以通过杆高 2 米确定比例尺，再通过图 7 中的影长计算出实际影长。进一步考虑到视角，我们发现实际上的影长是存在于三维视图中的，如图 8 所示：

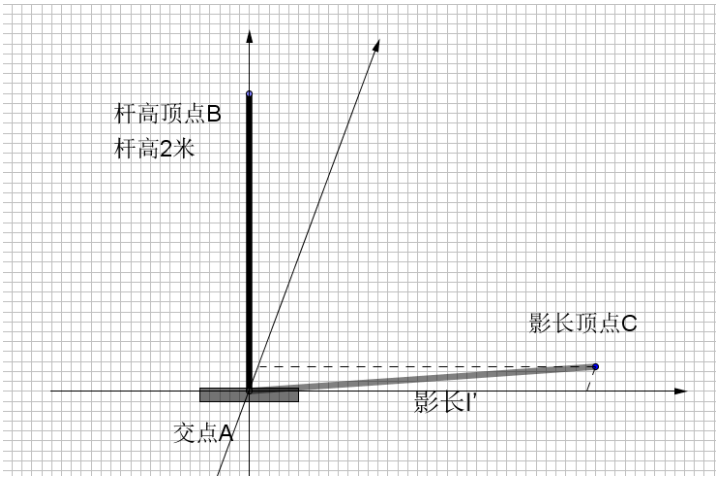


图 8 视频截图三维拟图

假设摄像机镜头所在的平面 γ，t 时刻直杆与影子所决定的平面 β 两个平面之间的夹角为 i，则实际影长 l 与截屏图片上的影长 l' 关于夹角 i 存在关系 f，即 $l = af(i)l'$，其中 a 为实际杆长与截屏图片杆长之间的比例系数。

当已知夹角 i 时我们可以对实际影长值做偏差修正，在本问题当中，如图根据视频截图 6 我们猜测夹角 i 很小，因此在计算的过程当中忽略夹角 i 对实际影长计算效果的影响。计算出的影长值如下表 4 所示：

时间/h:m	影长 l /m	时间/h:m	影长 l /m
8：55	2.34194	9：15	2.06635
8：57	2.32438	9：17	2.05461
8：59	2.29792	9：19	2.01068
9：1	2.2686	9：21	1.99602
9：3	2.23349	9：23	1.95797
9：5	2.20996	9：25	1.93456
9：7	2.18357	9：27	1.8994
9：9	2.15134	9：29	1.88184
9：11	2.125	9：31	1.86135
9：13	2.09858	9：33	1.83207

表 4 附件 4 中所测影长

②参数的假设

如问题二图 4 所示，建立相同的网格，测量时刻，影长、网格步长的假设条件及参数保持不变。

首次筛选出的网格交点数为 $n_0 = 10$，二次筛选的网格交点数为 $n_1 = 3$。

此时T为2015年7月13日，杆长已知通过某种方式估计出其长度为2米，即$h=2m$。

当视频拍摄的日期确定时

取网格交点P，对所有的网格交点执行以下步骤：

步骤1：首先假设初始计算时间$t=t_s$，n_0组均方误差Δl的初始值为无穷大。

步骤2：执行问题二模型中的步骤2至5，筛选出n_0组均方差最小的网格交点。

再从筛选出的n_0组方差和最小的网格交点选取P'，东西南北方向各扩展步长b_0，获得以P'为中心点的正方形，以步长b_1再次划分网格，再次进行上述搜索，筛选出n_1个可能地点。

当视频拍摄日期未知时

影长的测量日期为T，T的取值范围从Ts：1月1日第1天起，至T_e:12月31日365天止，以$\Delta T=1d$为增量。

取网格交点P，对所有的网格交点依次完成如下搜索步骤：

步骤1：假设初始计算时间$t=t_s$，n_0组较小方差和Δl初始值为无穷大。

步骤2：执行问题二模型中的步骤2至5。

步骤3：当$T<T_e$时，$T=T+\Delta T$，进行步骤2。

步骤4：筛选出n_0组均方差和最小的网格交点的经纬度、日期以及此时的均方差。

同问题二一致，从筛选出的n_0组均方差最小的网格交点选取P'，东西南北方向各扩展步长b_0，获得以P'为中心点的正方形，以步长b_1再次划分网格，再次进行上述搜索，筛选出n_1个可能地点。

（2）问题解答

①视频拍摄日期已知时

利用以上建立模型，知道日期时视频中影长测量有可能地点的相应数，给出的可能地点为东经$111.26°$，北纬$40.71°$，查找地图上相应的位置，我们可以判断可能的地点为呼和浩特。

②视频拍摄时间未知时

当日期未知时，可能的地点以及可能的拍摄日期如下表所示：

北纬/°	东经/°	均方差/m^2	日期/d	可能的地点
41.04	110.12	0.000863483	173	包头
40.91	110.64	0.000868156	187	呼和浩特

这里根据均方差的大小得到最有可能的地点为：包头，该结果与已知日期时获得的地点位置基本相同，且在以上有可能的地点时，计算得到的日期值为 173 天时，测量日期为 6 月 22 日，当日期值为 187 天时，测量日期为 7 月 6 日，在误差一定的范围内，可表明计算结果的准确性。

五、模型稳定性分析

在问题一中我们研究了影长与各个因素的关系，其中影长与经纬度关系图呈现出一种指数级的变化历程，建立在这种基础上的模型是否稳定与问题二结果的精确度相互关联。研究模型在初值扰动的情况下的稳定性，有利于验证模型的准确性，也是后续问题得以解决的基础。

针对于问题二的附件 1，我们对其初值进行不同方式、不同程度的扰动，观察所得的地点坐标与之前所得问题二的答案之间的差别。

在初值扰动范围 1%，扰动个数 2 的情况下，结果如下所示：

扰动前			扰动后		
精度	纬度	经度	精度	纬度	经度
5.39035e-007	18.88	109.27	0.000116219	20.4	107.6
6.89253e-006	21.1	106.08	0.000120071	19.4	109.2
1.03396e-005	24.79	97.71	0.000134687	23	103
1.1392e-005	-3.14	103.56	0.000135735	-4.4	100.4

在初值扰动范围 1%，扰动个数 4 的情况下，结果如下所示：

扰动前			扰动后		
精度	纬度	经度	精度	纬度	经度
5.39035e-007	18.88	109.27	0.000275984	20.4	107.6
6.89253e-006	21.1	106.08	0.000281875	-4.4	100.4
1.03396e-005	24.79	97.71	0.000282675	19.4	109.2
1.1392e-005	-3.14	103.56	0.000286296	21.4	106

我们可以看到扰动之后，所得结果的误差相比于之前大幅上升，但是仍在可接受的范围内，而且经纬度的变化也在较小范围内。在初值扰动范围 1%，扰动个数由 2 变为 4，两个扰动后的值基本一致，说明在扰动程度很小时，个数对结果的影响不大。

在初值扰动范围 5%，扰动个数 4 的情况下，结果如下所示：

扰动前			扰动后		
精度	纬度	经度	精度	纬度	经度
5.39035e-007	18.88	109.27	0.000471178	20.4	107.6
6.89253e-006	21.1	106.08	0.000477151	21.4	106
1.03396e-005	24.79	97.71	0.000478869	23	103
1.1392e-005	-3.14	103.56	0.000483139	19.4	109.2

在初值扰动范围 5%，扰动个数 6 的情况下，结果如下所示：

扰动前			扰动后		
精度	纬度	经度	精度	纬度	经度
5.39035e-007	18.88	109.27	0.00119668	23	103
6.89253e-006	21.1	106.08	0.00120079	23.6	101.6
1.03396e-005	24.79	97.71	0.00121571	24.2	100.2
1.1392e-005	-3.14	103.56	0.00122667	22.6	104.4

我们可以看到扰动之后，精度仍然可以得到 3 位小数的保证，所得的经纬度也在一定范围内正常波动。可以认为，在精度要求不很高的时候，模型是具有稳定性的，但如果精度要求过高，那么初值的正确选取是精确定位的前提。

六、模型评价及改进

6.1 优点

（1）模型框架简单

所构建的模型利用地理学的相关知识和简单的几何知识，应用文献中简化后的公式，得出计算影子长度的方法。模型的核心不涉及复杂的数学推理或换算，十分简洁易于理解。

（（2）程序优化程度高

本文我们采用分步网格搜索的方式来确定可能的地点和日期。这种分步网格搜索应用于编程，有很多可以优化的地方，具体如下：程序代码中，我们通过计算在每个数据上的影长与实际影长的方差和来比较精度，对于每个点的数据，根据所提供的数据的自变量（时间）计算得出因变量（影长），与所提供的因变量（影长）作差求平方，累加得到方差和。计算根据方差和的单调递增性，当前 i 组数据的方差和已经大于某一个值时就不再算后面的数据，使得程序效率大大提高，避免了无效计算。

6.2 缺点

（1）在所构建的计算影子长度的公式中，我们需要计算太阳赤纬，而这个数据没有精确的计算公式，只能估算，可能造成较大误差。

（2）对于问题二和问题三，我们不能利用题目所给的数据计算出杆长或者得到杆长的范围，而是认为得假定杆长在某一个区间里面，有失严谨性。对于问题三和问题四，我们没有考虑年份对太阳赤纬的影响。

（3）对于问题二、问题三和问题四，我们采用网格搜索的方式遍历地球上各个地点及日期、杆长，不能充分利用所给的数据缩小范围，因而需要进行大量

计算，效率较低。

（4）对于问题四，我们认为直杆的影子与我们的视线是垂直的，但是，影子始终处于运动状态，不能保证垂直关系，使得我们的测量有一定的误差；另外在计算视频中影子长度时，我们需要在杆的两端取点得到两点坐标以确定杆长，在这个过程中两个端点是通过手动来选取，具有测量误差。对于问题四，没有考虑年份对太阳赤纬的影响

6.3 改进

（1）对于问题四，为了减小测量误差，我们假设摄像机镜头所在的平面γ，t时刻直杆与影子所决定的平面β两个平面之间的夹角为i，实际影长l与截屏图片上的影长l'关于夹角i存在关系f，$l=af(i)l'$，其中a为实际杆长与截屏图片杆长之间的比例系数。当已知夹角i时，我们可以对实际影长值做偏差修正，以获得更加准确的实际影长值。

（2）大气圈对太阳光线的折射率也会产生一定的影响，但考虑到如图 2 所示：根据查找到的文献资料，空气折射率的计算公式如公式（13）：

其中气压p,温度t,湿度f,空气折射率n，ITS-90 同时使用国际开尔文温度（T90）和国际实用摄氏温度（t90）,单位分别为K和$°C$，换算公式为 t90=T90-273.15。

$$(n-1)_s*10^8=8342.54+\frac{2406147}{130-\sigma^2}+\frac{15998}{38.9-\sigma^2} \tag{10}$$

$$(n-1)_{tp}=\left(\frac{P(n-1)_s}{96095.43}\right)*\frac{[1+10^{-8}(0.601-0.00972t_{90})P]}{1+0.00361t_{90}} \tag{11}$$

$$n_{tpf}-n_{tp}=-f\left(3.7345-0.0401\sigma^2\right)*10^{-10} \tag{12}$$

将数据带入上式后,可计算出含有水汽压的空气折射率。大气中修正后的折射为：

$$\alpha=1+n_{tpf} \tag{13}$$

尽管在考虑了大气圈对太阳光纤的折射率影响时可以提高影长的计算精度,但由于在折射率的计算过程中需要考虑气压、温度、湿度等条件，这些条件在一直日期的情况下可以参考往年同期的数据预测出可能值，但此处存在误差，且当日期未知时则无法进行预估。

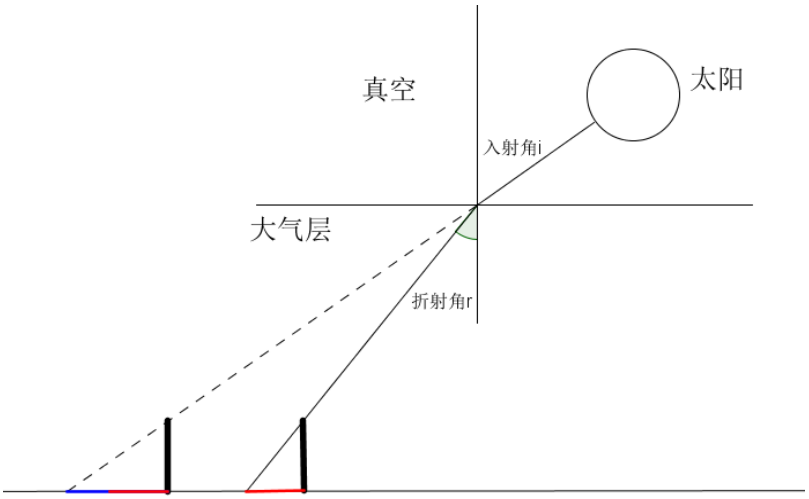


图 2

七、参考文献

[1] 屈名，王征兵，王德麾，基于较比不变性的太阳定位算法的研究，SILICON VALLEY，2013 年第 19 期（总第 139 期）：53，2013
[2] 王安国，米鸿涛，邓天宏，李亚男，李兰霞，太阳高度角和日出日落时刻太阳方位角一年变化范围的计算，气象与环境科学，第 30 卷：163，2007
[3] 谈小生，葛成辉，太阳角的计算方法及其在遥感中的应用，国土资源遥感，1995 年第 2 期（总第 24 期）：54，195
[4] 张文华，司德亮，徐淑通，祁东婷，太阳影子倍率的计算方法及其对光伏阵列布局的影响，太阳能技术与产品，2011 年第 9 卷：29.30，2011
[5] 张瑜，路博，一种高精度的太阳跟踪方法，可再生能源，第 30 卷第 2 期：104.105，2012
[6] 举子，太阳高度角的计算——用于建筑设计计算日照时间，http://blog.sina.com.cn/s/blog_4bc3b68d0100fxya.html，2015.9.11
[7] 温蓉胜，吴衍智，峰值网格搜索法，河北煤炭建铭工程学院学报，1993 年 01 期：1.2，2013
[8] 温浩宇，任小龙，徐国华，制造网格的搜索算法研究，中国机械工程，第 15 卷第 22 期：2014.2015，2004

八、附录

MATLAB 程序：

problem1-3.m

第一问__第二小问

```
x=9:0.1:15
y=[7.36574   7.0442   6.75076   6.48211   6.23545   6.00843   5.79904
5.60558   5.42656  5.26072   5.10696   4.96432   4.83197   4.70919   4.59533
```

```
4.48986   4.39227   4.30214  4.21909   4.1428   4.07298   4.00938   3.95178
3.89998   3.85383   3.81319   3.77792   3.74793   3.72315   3.70349   3.6889
3.67936   3.67483   3.6753   3.68078   3.69127  3.7068   3.72742   3.75319
3.78416    3.82043    3.86211    3.90931    3.96219    4.02091    4.08566
4.15668    4.23422    4.31858    4.41008    4.50912    4.61614    4.73163
4.85617    4.9904    5.13506    5.29103    5.45926    5.6409    5.83725
6.04982];
plot(x,y)
```

第四问__影长__确定交点
确定原点

```
x=[951  990  1045   1117    1249  1400   1471   1561   1639];
y=[885  884  882    882     880   887    875    872    870];
p=polyfit(x,y,1)

p =

   -0.0180  902.5041

x=869;
y=-0.018*x+902.5041

y =

  886.8621
```

C 语言程序：
problem1-3.cpp

```
#include <iostream>
#include <string.h>
#include <stdio.h>
#include <fstream>
#include <cmath>
using namespace std;


const double pi=acos(-1);
const double t_y=2015;
const double t_m=10;
const double t_d=22;//年月日
const double pw_d=39;
const double pw_m=54;
const double pw_s=26;//纬度
```

```
const double height=3;//直杆高度
//
double chiwei;
double weidu;
double shijiao;
double rijiao;
double time_10;
int cal_N(int a,int b)
{
    int n=0;
    switch(a)
    {
        case 12: n+=30;
        case 11: n+=31;
        case 10: n+=30;
        case 9: n+=31;
        case 8: n+=31;
        case 7: n+=30;
        case 6: n+=31;
        case 5: n+=30;
        case 4: n+=31;
        case 3: n+=28;
        case 2: n+=31;
        case 1: ;
    }
    n+=b;
    return n;
}

void cal_rijiao()
{
    double N=cal_N(10,22);
    double N0=79.6764+0.2422*(t_y-1985)-(int)((t_y-1985)/4);
    double t=N-N0;
    rijiao=2*pi*t/365.2422;
}

void cal_shijiao()
{
    time_10=time_10+(116+23/60.0+29/3600.0-120)/15.0;
    double ans=(time_10-12)*15;
    shijiao=ans*pi/180.0;
}
```

```
void cal_chiwei()
{
    cal_rijiao();
    chiwei=(0.3723+23.2567*sin(rijiao)+0.1149*sin(2*rijiao)-0.1712*si
n(3*rijiao)-0.758*cos(rijiao)+0.3656*cos(2*rijiao)+0.0201*cos(3*rijia
o))*pi/180.0;
}

void cal_weidu()
{
    weidu=(pw_d+pw_m/60.0+pw_s/3600.0)*pi/180.0;
}

double cal_x()
{
    cal_shijiao();
    double
sin_hs=sin(chiwei)*sin(weidu)+cos(chiwei)*cos(weidu)*cos(shijiao);
    double cos_hs=sqrt(1-sin_hs*sin_hs);
    double tan_hs=fabs(sin_hs/cos_hs);
    return height/tan_hs;
}

int main()
{
    cal_chiwei();
    cal_weidu();
    double a=9,b=15;
    double dx=0.1;//每两个点之间横坐标的差值，可以自己改
    double arrx[2000];
    double arry[2000];
    int cnt=0;
    double x=a;
    double y;
     while(x<=b)
     {
        time_10=x;
        y=cal_x();
        arrx[cnt]=x;
        arry[cnt++]=y;
        //cout<<x<<"  "<<y<<endl;
        x+=dx;
     }
    int i;
```

```
    for(i=0;i<cnt;i++)   cout<<arrx[i]<<" ";
    cout<<endl;
     for(i=0;i<cnt;i++)   cout<<arry[i]<<" ";
    return 0;
}
```

problem2.cpp

```
#include <iostream>
#include <string.h>
#include <stdio.h>
#include <fstream>
#include <cmath>
#include <time.h>
#include <algorithm>
using namespace std;
const double pi=acos(-1);
const double t_y=2015;
const double t_m=4;
const double t_d=18;//年月日

//
double chiwei;
double weidu;
double jingdu;
double shijiao;
double rijiao;
double time_10;
double height;//直杆高度
//
double
x[25]={1.0365,1.0699,1.1038,1.1383,1.1732,1.2087,1.2448,1.2815,1.3189
,1.3568,1.3955,1.4349,

1.4751,1.5160,1.5577,1.6003,1.6438,1.6882,1.7337,1.7801,1.8277};
double
y[25]={0.4973,0.5029,0.5085,0.5142,0.5198,0.5255,0.5311,0.5368,0.5426
,0.5483,0.5541,0.5598,0.5657,

0.5715,0.5774,0.5833,0.5892,0.5952,0.6013,0.6074,0.6135};
double len[25];
double
timm[25]={14,14,14,14,14,14,15,15,15,15,15,15,15,15,15,15,15,15,15,15
,15};
double
```

```
tims[25]={42,45,48,51,54,57,0,3,6,9,12,15,18,21,24,27,30,33,36,39,42}
;
double tim[25];
struct node
{
    double s2;
    double weidu;
     double jingdu;
    double height;//直杆高度
}pos[105],pos2[105];

int n=20;
////////
void init();
int cal_N(int a,int b);
void cal_rijiao();
void cal_chiwei();
void cal_shijiao();
double cal_x();
bool cmp(node x,node y)
{
    return x.s2<y.s2;
}
/////
int main()
{
    init();
    cal_chiwei();
    double i,j,k;
    int l,m;
    double span_weidu=0.2;
    double span_jingdu=0.2;
    double span_height=0.1;
    for(i=-180;i<180;i=i+span_jingdu)//jingdu
    {
        jingdu=i;
        for(j=-90;j<=90;j=j+span_weidu)//weidu
        {
            weidu=j;
            for(k=1;k<=6;k=k+span_height)//gangao
            {
                height=k;
                double s2=0;
                int flag=1;
```

```
                for(l=0;l<21;l++)
                {
                    time_10=tim[l];
                    double xt=cal_x();
                    s2+=(xt-len[l])*(xt-len[l]);

                    if(s2>pos[n-1].s2)
                    {
                        flag=0;
                        break;
                    }

                }
                if(flag==0)   continue;
                for(m=n-1;m>=0;m--)
                {
                    if(s2<pos[m].s2)
                    {
                        pos[m+1].s2=pos[m].s2;
                        pos[m+1].weidu=pos[m].weidu;
                        pos[m+1].jingdu=pos[m].jingdu;
                        pos[m+1].height=pos[m].height;

                    }
                    else break;
                }
                if(m==n-1)  continue;
                pos[m+1].s2=s2;
                pos[m+1].weidu=weidu;
                pos[m+1].jingdu=jingdu;
                pos[m+1].height=height;
            }
        }
    }
    int ii;
    for(ii=0;ii<n;ii++)
    {
        cout<<pos[ii].s2<<"  "<<pos[ii].weidu<<"  "<<pos[ii].jingdu<<"
"<<pos[ii].height<<endl;
    }
    for(ii=0;ii<5;ii++)  cout<<endl;
    double span_jingdu2=0.01;
    double span_weidu2=0.01;
    for(l=0;l<n;l++)//n 个点
```

```
    {
        height=pos[l].height;
        double l1=pos[l].jingdu-span_jingdu*3;
        double r1=pos[l].jingdu+span_jingdu*3;
        for(i=l1;i<=r1;i=i+span_jingdu2)
        {
            jingdu=i;
            double l2=pos[l].weidu-span_weidu*3;
             double r2=pos[l].weidu+span_weidu*3;
            for(j=l2;j<=r2;j=j+span_weidu2)
            {
                weidu=j;
                double s2=0;
                for(m=0;m<21;m++)
                {
                    time_10=tim[m];
                    double xt=cal_x();
                    s2+=(xt-len[m])*(xt-len[m]);
                }
                if(s2<pos2[l].s2)
                {
                    pos2[l].s2=s2;
                    pos2[l].weidu=weidu;
                    pos2[l].jingdu=jingdu;
                    pos2[l].height=height;
                     //cout<<s2<<"        "<<weidu<<"        "<<jingdu<<"
"<<height<<endl;
                }

            }
        }
    }
    sort(pos2,pos2+n,cmp);
    for(ii=0;ii<n;ii++)
    {

    //printf("%.15lf    %.3lf    %.3lf     %.1lf\n",pos2[ii].s2,pos2[ii]
.weidu*180/pi,pos2[ii].jingdu,pos2[ii].height);
        cout<<pos2[ii].s2<<"                          "<<pos2[ii].weidu<<"
"<<pos2[ii].jingdu<<"  "<<pos2[ii].height<<endl;
    }
     return 0;
}
```

```
///
void init()
{
    int i;
    for(i=0;i<21;i++)
    {
        len[i]=sqrt(x[i]*x[i]+y[i]*y[i]);
    }
    for(i=0;i<n;i++)
    {
        pos[i].s2=999999999;
        pos2[i].s2=999999999;
    }
    for(i=0;i<21;i++)
    {
        tim[i]=timm[i]+tims[i]/60.0;
    }

}

int cal_N(int a,int b)
{
    int n=0;
    switch(a)
    {
        case 12: n+=30;
        case 11: n+=31;
        case 10: n+=30;
        case 9: n+=31;
        case 8: n+=31;
        case 7: n+=30;
        case 6: n+=31;
        case 5: n+=30;
        case 4: n+=31;
        case 3: n+=28;
        case 2: n+=31;
        case 1: ;
    }
    n+=b;
    return n;
}

void cal_rijiao()
{
```

```
    double N=cal_N(t_m,t_d);
    double N0=79.6764+0.2422*(t_y-1985)-(int)((t_y-1985)/4);
    double t=N-N0;
     rijiao=2*pi*t/365.2422;
}

void cal_chiwei()
{
    cal_rijiao();
    chiwei=(0.3723+23.2567*sin(rijiao)+0.1149*sin(2*rijiao)-0.1712*si
n(3*rijiao)-0.758*cos(rijiao)+0.3656*cos(2*rijiao)+0.0201*cos(3*rijia
o))*pi/180.0;
}

void cal_shijiao()
{
    time_10=time_10+(jingdu-120)/15.0;
    double ans=(time_10-12)*15;
    shijiao=ans*pi/180.0;
}

double cal_x()
{
    cal_shijiao();
    double weidu_pi=weidu*pi/180.0;
    double
sin_hs=sin(chiwei)*sin(weidu_pi)+cos(chiwei)*cos(weidu_pi)*cos(shijia
o);
    double cos_hs=sqrt(1-sin_hs*sin_hs);
    double tan_hs=sin_hs/cos_hs;
    return height/tan_hs;
}
/*


*/
```

problem3-1.cpp

```
#include <iostream>
#include <string.h>
#include <stdio.h>
#include <fstream>
#include <cmath>
#include <algorithm>
using namespace std;
```

```
const double pi=acos(-1);
const double t_y=2015;
const double t_m=4;
const double t_d=18;//年月日

//
int date_order;
double chiwei;
double weidu;
double jingdu;
double shijiao;
double rijiao;
double time_10;
double height;//直杆高度

//////////////////////////////////////

double
x[25]={-1.2352,-1.2081,-1.1813,-1.1546,-1.1281,-1.1018,-1.0756,-1.049
6,-1.0237,-0.998,-0.9724,

-0.947,-0.9217,-0.8965,-0.8714,-0.8464,-0.8215,-0.7967,-0.7719,-0.747
3,-0.7227};

double
y[25]={0.173,0.189,0.2048,0.2203,0.2356,0.2505,0.2653,0.2798,0.294,0.
308,0.3218,0.3354,0.3488,

0.3619,0.3748,0.3876,0.4001,0.4124,0.4246,0.4366,0.4484};
double len[25];
double
timm[25]={12,12,12,12,12,12,12,13,13,13,13,13,13,13,13,13,13,13,13,13
,13};
double
tims[25]={41,44,47,50,53,56,59,2,5,8,11,14,17,20,23,26,29,32,35,38,41
};

////////////////////////////


/*
double
x[25]={1.1637,1.2212,1.2791,1.3373,1.396,1.4552,1.5148,1.575,1.6357,1
.697,1.7589,
```

```
1.8215,1.8848,1.9488,2.0136,2.0792,2.1457,2.2131,2.2815,2.3508,2.4213
};
double
y[25]={3.336,3.3299,3.3242,3.3188,3.3137,3.3091,3.3048,3.3007,3.2971,
3.2937,

3.2907,3.2881,3.2859,3.284,3.2824,3.2813,3.2805,3.2801,3.2801,3.2804,
3.2812};
double len[25];
double
timm[25]={13,13,13,13,13,13,13,13,13,13,13,13,13,13,13,13,13,14,14,14
,14};
double
tims[25]={9,12,15,18,21,24,27,30,33,36,39,42,45,48,51,54,57,0,3,6,9};
*/

double tim[25];
struct node
{
    double s2;
    double weidu;
     double jingdu;
    double height;//直杆高度
    int day_order;
}pos[105],pos2[105];

int n=40;
////////
void init();
int cal_N(int a,int b);
void cal_rijiao();
void cal_chiwei();
void cal_shijiao();
double cal_x();
int day_to_date();
bool cmp(node x,node y)
{
    return x.s2<y.s2;
}
/////
int main()
{
    init();
```

```
double i,j,k;
int l,m,o;
double span_weidu=0.2;
double span_jingdu=0.2;
double span_height=0.1;
double span_date=3;
for(o=span_date;o<=360;o+=span_date)
{
    //
    date_order=o;
    cal_chiwei();
    for(i=-180;i<180;i=i+span_jingdu)//jingdu
    {
        jingdu=i;
        for(j=-90;j<=90;j=j+span_weidu)//weidu
        {
            weidu=j;
            for(k=0.2;k<=4;k=k+span_height)//gangao
            {
                height=k;
                double s2=0;
                int flag=1;
                for(l=0;l<21;l++)
                {
                    time_10=tim[l];
                    double xt=cal_x();
                    s2+=(xt-len[l])*(xt-len[l]);
                    if(s2>pos[n-1].s2)
                    {
                        flag=0;
                        break;
                    }
                }
                if(flag==0)   continue;
                for(m=n-1;m>=0;m--)
                {
                    if(s2<pos[m].s2)
                    {
                        pos[m+1].s2=pos[m].s2;
                        pos[m+1].weidu=pos[m].weidu;
                        pos[m+1].jingdu=pos[m].jingdu;
                        pos[m+1].height=pos[m].height;
                        pos[m+1].day_order=pos[m].day_order;
```

```
                    }
                    else break;
                }
                if(m==n-1)  continue;
                pos[m+1].s2=s2;
                pos[m+1].weidu=weidu;
                pos[m+1].jingdu=jingdu;
                pos[m+1].height=height;
                pos[m+1].day_order=date_order;
            }
        }
    }
}
int ii;
for(ii=0;ii<n;ii++)
{
    cout<<pos[ii].s2<<"  "<<pos[ii].weidu<<"  "<<pos[ii].jingdu<<"
"<<pos[ii].height<<"  "<<pos[ii].day_order<<endl;
}
for(ii=0;ii<5;ii++)  cout<<endl;
double span_jingdu2=0.01;
double span_weidu2=0.01;
double span_date2=1;
for(l=0;l<n;l++)//n 个点
{
    height=pos[l].height;
    int l3=pos[l].day_order-span_date*2;
    int r3=pos[l].day_order+span_date*2;
    for(o=l3;o<=r3&&o<=365;o+=span_date2)
    {
        date_order=o;
        cal_chiwei();
        double l1=pos[l].jingdu-span_jingdu*4;
        double r1=pos[l].jingdu+span_jingdu*4;
        for(i=l1;i<=r1;i+=span_jingdu2)
        {
            jingdu=i;
            double l2=pos[l].weidu-span_weidu*4;
             double r2=pos[l].weidu+span_weidu*4;
            for(j=l2;j<=r2;j+=span_weidu2)
            {
                weidu=j;
                double s2=0;
```

```
                    for(m=0;m<21;m++)
                    {
                        time_10=tim[m];
                        double xt=cal_x();
                        s2+=(xt-len[m])*(xt-len[m]);
                    }
                    if(s2<pos2[l].s2)
                    {
                        pos2[l].s2=s2;
                        pos2[l].weidu=weidu;
                        pos2[l].jingdu=jingdu;
                        pos2[l].height=height;
                        pos2[l].day_order=date_order;
                    }

                }
            }
        }
    }
    sort(pos2,pos2+n,cmp);
    for(ii=0;ii<n;ii++)
    {

    //printf("%.15lf    %.3lf    %.3lf     %.1lf\n",pos2[ii].s2,pos2[ii]
.weidu*180/pi,pos2[ii].jingdu,pos2[ii].height);
        cout<<pos2[ii].s2<<"                          "<<pos2[ii].weidu<<"
"<<pos2[ii].jingdu<<"                          "<<pos2[ii].height<<"
"<<pos2[ii].day_order<<endl;
    }
    return 0;
}

///
void init()
{
    int i;
    for(i=0;i<21;i++)
    {
        len[i]=sqrt(x[i]*x[i]+y[i]*y[i]);
    }
    for(i=0;i<n;i++)
    {
        pos[i].s2=999999999;
        pos2[i].s2=999999999;
```

```
    }
    for(i=0;i<21;i++)
    {
        tim[i]=timm[i]+tims[i]/60.0;
    }

}

int cal_N(int a,int b)
{
    int n=0;
    switch(a)
    {
        case 12: n+=30;
        case 11: n+=31;
        case 10: n+=30;
        case 9: n+=31;
        case 8: n+=31;
        case 7: n+=30;
        case 6: n+=31;
        case 5: n+=30;
        case 4: n+=31;
        case 3: n+=28;
        case 2: n+=31;
        case 1: ;
    }
    n+=b;
    return n;
}

void cal_rijiao()
{
    double N=date_order;
    double N0=79.6764+0.2422*(t_y-1985)-(int)((t_y-1985)/4);
    double t=N-N0;
    rijiao=2*pi*t/365.2422;
}

void cal_chiwei()
{
    cal_rijiao();
    chiwei=(0.3723+23.2567*sin(rijiao)+0.1149*sin(2*rijiao)-0.1712*si
n(3*rijiao)-0.758*cos(rijiao)+0.3656*cos(2*rijiao)+0.0201*cos(3*rijia
o))*pi/180.0;
```

```
}

void cal_shijiao()
{
    time_10=time_10+(jingdu-120)/15.0;
    double ans=(time_10-12)*15;
    shijiao=ans*pi/180.0;
}

double cal_x()
{
    cal_shijiao();
    double weidu_pi=weidu*pi/180.0;
    double
sin_hs=sin(chiwei)*sin(weidu_pi)+cos(chiwei)*cos(weidu_pi)*cos(shijia
o);
    double cos_hs=sqrt(1-sin_hs*sin_hs);
    double tan_hs=sin_hs/cos_hs;
    return height/tan_hs;
}

int day_to_date()
{

    return 1;
}
/*


*/
```

problem3-2.cpp

```
#include <iostream>
#include <string.h>
#include <stdio.h>
#include <fstream>
#include <cmath>
#include <algorithm>
using namespace std;
const double pi=acos(-1);
const double t_y=2015;
const double t_m=4;
const double t_d=18;//年月日

//
```

```
int date_order;
double chiwei;
double weidu;
double jingdu;
double shijiao;
double rijiao;
double time_10;
double height;//直杆高度

//////////////////////////////////////
/*
double
x[25]={-1.2352,-1.2081,-1.1813,-1.1546,-1.1281,-1.1018,-1.0756,-1.049
6,-1.0237,-0.998,-0.9724,

-0.947,-0.9217,-0.8965,-0.8714,-0.8464,-0.8215,-0.7967,-0.7719,-0.747
3,-0.7227};

double
y[25]={0.173,0.189,0.2048,0.2203,0.2356,0.2505,0.2653,0.2798,0.294,0.
308,0.3218,0.3354,0.3488,

0.3619,0.3748,0.3876,0.4001,0.4124,0.4246,0.4366,0.4484};
double len[25];
double
timm[25]={12,12,12,12,12,12,12,13,13,13,13,13,13,13,13,13,13,13,13,13
,13};
double
tims[25]={41,44,47,50,53,56,59,2,5,8,11,14,17,20,23,26,29,32,35,38,41
};
*/
//////////////////////////////



double
x[25]={1.1637,1.2212,1.2791,1.3373,1.396,1.4552,1.5148,1.575,1.6357,1
.697,1.7589,

1.8215,1.8848,1.9488,2.0136,2.0792,2.1457,2.2131,2.2815,2.3508,2.4213
};
double
y[25]={3.336,3.3299,3.3242,3.3188,3.3137,3.3091,3.3048,3.3007,3.2971,
3.2937,
```

```
3.2907,3.2881,3.2859,3.284,3.2824,3.2813,3.2805,3.2801,3.2801,3.2804,
3.2812};
double len[25];
double
timm[25]={13,13,13,13,13,13,13,13,13,13,13,13,13,13,13,13,13,14,14,14
,14};
double
tims[25]={9,12,15,18,21,24,27,30,33,36,39,42,45,48,51,54,57,0,3,6,9};


double tim[25];
struct node
{
    double s2;
    double weidu;
     double jingdu;
    double height;//直杆高度
    int day_order;
}pos[105],pos2[105];

int n=40;
////////
void init();
int cal_N(int a,int b);
void cal_rijiao();
void cal_chiwei();
void cal_shijiao();
double cal_x();
int day_to_date();
bool cmp(node x,node y)
{
    return x.s2<y.s2;
}
/////
int main()
{
    init();

    double i,j,k;
    int l,m,o;
    double span_weidu=0.2;
    double span_jingdu=0.2;
    double span_height=0.1;
```

```
double span_date=3;
 for(o=span_date;o<=360;o+=span_date)
{
    //
     date_order=o;
    cal_chiwei();
     for(i=-180;i<180;i=i+span_jingdu)//jingdu
    {
        jingdu=i;
        for(j=-90;j<=90;j=j+span_weidu)//weidu
        {
           weidu=j;
            for(k=1;k<=5;k=k+span_height)//gangao
            {
                height=k;
               double s2=0;
               int flag=1;
                for(l=0;l<21;l++)
               {
                   time_10=tim[l];
                   double xt=cal_x();
                   s2+=(xt-len[l])*(xt-len[l]);
                   if(s2>pos[n-1].s2)
                   {
                       flag=0;
                       break;
                   }
               }
                if(flag==0)   continue;
               for(m=n-1;m>=0;m--)
               {
                   if(s2<pos[m].s2)
                   {
                       pos[m+1].s2=pos[m].s2;
                        pos[m+1].weidu=pos[m].weidu;
                       pos[m+1].jingdu=pos[m].jingdu;
                        pos[m+1].height=pos[m].height;
                       pos[m+1].day_order=pos[m].day_order;

                   }
                   else break;
               }
               if(m==n-1)  continue;
                pos[m+1].s2=s2;
```

```
                    pos[m+1].weidu=weidu;
                    pos[m+1].jingdu=jingdu;
                    pos[m+1].height=height;
                   pos[m+1].day_order=date_order;
                }
            }
        }
    }
    int ii;
    for(ii=0;ii<n;ii++)
    {
        cout<<pos[ii].s2<<"  "<<pos[ii].weidu<<"  "<<pos[ii].jingdu<<"
"<<pos[ii].height<<"  "<<pos[ii].day_order<<endl;
    }
    for(ii=0;ii<5;ii++)  cout<<endl;
    double span_jingdu2=0.01;
    double span_weidu2=0.01;
    double span_date2=1;
    for(l=0;l<n;l++)//n 个点
    {
        height=pos[l].height;
        int l3=pos[l].day_order-span_date*2;
        int r3=pos[l].day_order+span_date*2;
        for(o=l3;o<=r3&&o<=365;o+=span_date2)
        {
            date_order=o;
            cal_chiwei();
             double l1=pos[l].jingdu-span_jingdu*4;
             double r1=pos[l].jingdu+span_jingdu*4;
             for(i=l1;i<=r1;i+=span_jingdu2)
             {
                jingdu=i;
                double l2=pos[l].weidu-span_weidu*4;
                 double r2=pos[l].weidu+span_weidu*4;
                for(j=l2;j<=r2;j+=span_weidu2)
                {
                    weidu=j;
                    double s2=0;
                    for(m=0;m<21;m++)
                    {
                        time_10=tim[m];
                        double xt=cal_x();
                        s2+=(xt-len[m])*(xt-len[m]);
                    }
```

```
                    if(s2<pos2[l].s2)
                    {
                        pos2[l].s2=s2;
                        pos2[l].weidu=weidu;
                        pos2[l].jingdu=jingdu;
                        pos2[l].height=height;
                        pos2[l].day_order=date_order;
                    }

                }
            }
        }
    }
    sort(pos2,pos2+n,cmp);
    for(ii=0;ii<n;ii++)
    {

    //printf("%.15lf    %.3lf    %.3lf     %.1lf\n",pos2[ii].s2,pos2[ii]
.weidu*180/pi,pos2[ii].jingdu,pos2[ii].height);
        cout<<pos2[ii].s2<<"                          "<<pos2[ii].weidu<<"
"<<pos2[ii].jingdu<<"                          "<<pos2[ii].height<<"
"<<pos2[ii].day_order<<endl;
    }
    return 0;
}

///
void init()
{
    int i;
    for(i=0;i<21;i++)
    {
        len[i]=sqrt(x[i]*x[i]+y[i]*y[i]);
    }
    for(i=0;i<n;i++)
    {
        pos[i].s2=999999999;
        pos2[i].s2=999999999;
    }
    for(i=0;i<21;i++)
    {
        tim[i]=timm[i]+tims[i]/60.0;
    }
```

```
}

int cal_N(int a,int b)
{
    int n=0;
    switch(a)
    {
        case 12: n+=30;
        case 11: n+=31;
        case 10: n+=30;
        case 9: n+=31;
        case 8: n+=31;
        case 7: n+=30;
        case 6: n+=31;
        case 5: n+=30;
        case 4: n+=31;
        case 3: n+=28;
        case 2: n+=31;
        case 1: ;
    }
    n+=b;
    return n;
}

void cal_rijiao()
{
    double N=date_order;
    double N0=79.6764+0.2422*(t_y-1985)-(int)((t_y-1985)/4);
    double t=N-N0;
    rijiao=2*pi*t/365.2422;
}

void cal_chiwei()
{
    cal_rijiao();
    chiwei=(0.3723+23.2567*sin(rijiao)+0.1149*sin(2*rijiao)-0.1712*si
n(3*rijiao)-0.758*cos(rijiao)+0.3656*cos(2*rijiao)+0.0201*cos(3*rijia
o))*pi/180.0;
}

void cal_shijiao()
{
    time_10=time_10+(jingdu-120)/15.0;
    double ans=(time_10-12)*15;
```

```
    shijiao=ans*pi/180.0;
}

double cal_x()
{
    cal_shijiao();
    double weidu_pi=weidu*pi/180.0;
    double
sin_hs=sin(chiwei)*sin(weidu_pi)+cos(chiwei)*cos(weidu_pi)*cos(shijia
o);
    double cos_hs=sqrt(1-sin_hs*sin_hs);
    double tan_hs=sin_hs/cos_hs;
    return height/tan_hs;
}

int day_to_date()
{

    return 1;
}
/*


*/
```

problem4-1.cpp

```
#include <iostream>
#include <string.h>
#include <stdio.h>
#include <fstream>
#include <cmath>
#include <algorithm>
#include <iomanip>
using namespace std;
const double pi=acos(-1);
const double t_y=2015;
const double t_m=7;
const double t_d=13;//年月日
const double 0x=869;
const double 0y=887;
const double x_ganding=892;
const double y_ganding=204;

const int num=20;//数据个数
```

```
double
height_p=sqrt((0x-x_ganding)*(0x-x_ganding)+(0y-y_ganding)*(0y-y_gand
ing));
//

double chiwei;
double weidu;
double jingdu;
double shijiao;
double rijiao;
double time_10;
const double height=2.0;//直杆高度
//
double x[25];
double y[25];
double len_p[25];
double len[25];
double timm[25];
double tims[25];
double tim[25];
struct node
{
    double s2;
    double weidu;
     double jingdu;
}pos[105],pos2[105];

int n=20;
////////
void init();
int cal_N(int a,int b);
void cal_rijiao();
void cal_chiwei();
void cal_shijiao();
double cal_x();
bool cmp(node x,node y)
{
    return x.s2<y.s2;
}
/////
int main()
{
    init();
    cal_chiwei();
```

```
double i,j,k;
int l,m;
double span_weidu=0.2;
double span_jingdu=0.2;
for(i=-180;i<180;i=i+span_jingdu)//jingdu
{
    jingdu=i;
    for(j=-90;j<=90;j=j+span_weidu)//weidu
    {
        weidu=j;

        double s2=0;
        int flag=1;
        for(l=0;l<num;l++)
        {
            time_10=tim[l];
            double xt=cal_x();
            s2+=(xt-len[l])*(xt-len[l]);

            if(s2>pos[n-1].s2)
            {
                flag=0;
                break;
            }

        }
        if(flag==0)   continue;
        for(m=n-1;m>=0;m--)
        {
            if(s2<pos[m].s2)
            {
                pos[m+1].s2=pos[m].s2;
                pos[m+1].weidu=pos[m].weidu;
                pos[m+1].jingdu=pos[m].jingdu;

            }
            else break;
        }
        if(m==n-1)  continue;
        pos[m+1].s2=s2;
        pos[m+1].weidu=weidu;
        pos[m+1].jingdu=jingdu;
    }
}
```

```
    int ii;
    for(ii=0;ii<n;ii++)
    {
       //cout<<pos[ii].s2<<"                          "<<pos[ii].weidu<<"
"<<pos[ii].jingdu<<endl;

    printf("%.15lf    %.2lf    %.2lf\n",pos[ii].s2,pos[ii].weidu,pos[ii
].jingdu);
    }
    for(ii=0;ii<5;ii++)  cout<<endl;
    double span_jingdu2=0.01;
    double span_weidu2=0.01;
    for(l=0;l<n;l++)//n 个点
    {
       double l1=pos[l].jingdu-span_jingdu*3;
       double r1=pos[l].jingdu+span_jingdu*3;
       for(i=l1;i<=r1;i=i+span_jingdu2)
       {
          jingdu=i;
          double l2=pos[l].weidu-span_weidu*3;
           double r2=pos[l].weidu+span_weidu*3;
          for(j=l2;j<=r2;j=j+span_weidu2)
          {
             weidu=j;
             double s2=0;
             for(m=0;m<num;m++)
             {
                time_10=tim[m];
                double xt=cal_x();
                s2+=(xt-len[m])*(xt-len[m]);
             }
             if(s2<pos2[l].s2)
             {
                pos2[l].s2=s2;
                pos2[l].weidu=weidu;
                pos2[l].jingdu=jingdu;
                 //cout<<s2<<"       "<<weidu<<"       "<<jingdu<<"
"<<height<<endl;
             }

          }
       }
    }
    sort(pos2,pos2+n,cmp);
```

```
    for(ii=0;ii<n;ii++)
    {

    printf("%.15lf    %.2lf    %.2lf\n",pos2[ii].s2,pos2[ii].weidu,pos2
[ii].jingdu);

    //cout<<setw(20)<<pos2[ii].s2<<setw(10)<<pos2[ii].weidu<<setw(10)
<<pos2[ii].jingdu<<endl;
    }
    return 0;
}

///
void init()
{
    int i;
    for(i=0;i<num;i++)   cin>>timm[i];
    for(i=0;i<num;i++)   cin>>tims[i];
    for(i=0;i<num;i++)   cin>>x[i];
    for(i=0;i<num;i++)   cin>>y[i];
    for(i=0;i<num;i++)
len_p[i]=sqrt((x[i]-0x)*(x[i]-0x)+(y[i]-0y)*(y[i]-0y));
    for(i=0;i<num;i++)   tim[i]=timm[i]+tims[i]/60.0;
    double k=height/height_p;
    for(i=0;i<num;i++)   len[i]=len_p[i]*k;
    for(i=0;i<n;i++)
    {
        pos[i].s2=999999999;
        pos2[i].s2=999999999;
    }
    for(i=0;i<num;i++)   cout<<len[i]<<"  ";
    cout<<endl;
}

int cal_N(int a,int b)
{
    int n=0;
    switch(a)
    {
        case 12: n+=30;
        case 11: n+=31;
        case 10: n+=30;
        case 9: n+=31;
        case 8: n+=31;
```

```
        case 7: n+=30;
        case 6: n+=31;
        case 5: n+=30;
        case 4: n+=31;
        case 3: n+=28;
        case 2: n+=31;
        case 1: ;
    }
    n+=b;
    return n;
}

void cal_rijiao()
{
    double N=cal_N(t_m,t_d);
    double N0=79.6764+0.2422*(t_y-1985)-(int)((t_y-1985)/4);
    double t=N-N0;
    rijiao=2*pi*t/365.2422;
}

void cal_chiwei()
{
    cal_rijiao();
    chiwei=(0.3723+23.2567*sin(rijiao)+0.1149*sin(2*rijiao)-0.1712*si
n(3*rijiao)-0.758*cos(rijiao)+0.3656*cos(2*rijiao)+0.0201*cos(3*rijia
o))*pi/180.0;
}

void cal_shijiao()
{
    time_10=time_10+(jingdu-120)/15.0;
    double ans=(time_10-12)*15;
    shijiao=ans*pi/180.0;
}

double cal_x()
{
    cal_shijiao();
    double weidu_pi=weidu*pi/180.0;
    double
sin_hs=sin(chiwei)*sin(weidu_pi)+cos(chiwei)*cos(weidu_pi)*cos(shijia
o);
    double cos_hs=sqrt(1-sin_hs*sin_hs);
    double tan_hs=sin_hs/cos_hs;
```

```
    return height/tan_hs;
}
/*
8     8     8     9     9     9     9     9     9     9     9
9     9     9     9     9     9     9     9     9
55    57    59    1     3     5     7     9     11    13    15
17    19    21    23    25    27    29    31    33
1669  1663  1654  1644  1632  1624  1615  1604  1595  1586  1575
1571  1556  1551  1538  1530  1518  1512  1505  1495
868   868   870   871   871   873   874   875   875   877   878
879   880   881   881   881   883   883   883   884

*/
```

problem4-2.cpp

```
#include <iostream>
#include <string.h>
#include <stdio.h>
#include <fstream>
#include <cmath>
#include <algorithm>
using namespace std;
//////////////////////////
const double pi=acos(-1);
const double t_y=2015;
const double t_m=7;
const double t_d=13;//年月日
const double 0x=869;
const double 0y=887;
const double x_ganding=892;
const double y_ganding=204;

const int num=20;//数据个数

double
height_p=sqrt((0x-x_ganding)*(0x-x_ganding)+(0y-y_ganding)*(0y-y_gand
ing));
//

int date_order;
double chiwei;
double weidu;
double jingdu;
double shijiao;
```

```
double rijiao;
double time_10;
const double height=2.0;//直杆高度
//
double x[25];
double y[25];
double len_p[25];
double len[25];
double timm[25];
double tims[25];
double tim[25];
/////////////////////////////

struct node
{
    double s2;
    double weidu;
     double jingdu;
    int day_order;
}pos[105],pos2[105];

int n=60;
////////
void init();
int cal_N(int a,int b);
void cal_rijiao();
void cal_chiwei();
void cal_shijiao();
double cal_x();
bool cmp(node x,node y)
{
    return x.s2<y.s2;
}
/////
int main()
{
    init();

    double i,j,k;
    int l,m,o;
    double span_weidu=0.1;
    double span_jingdu=0.1;
    double span_date=1;
     for(o=span_date;o<=360;o+=span_date)
```

```
{
    //
     date_order=o;
    cal_chiwei();
     for(i=-180;i<180;i=i+span_jingdu)//jingdu
    {
        jingdu=i;
        for(j=-90;j<=90;j=j+span_weidu)//weidu
        {
           weidu=j;
            double s2=0;
            int flag=1;
           for(l=0;l<num;l++)
           {
               time_10=tim[l];
               double xt=cal_x();
               s2+=(xt-len[l])*(xt-len[l]);
              if(s2>pos[n-1].s2)
               {
                  flag=0;
                  break;
               }
           }
           if(flag==0)   continue;
            for(m=n-1;m>=0;m--)
           {
               if(s2<pos[m].s2)
               {
                    pos[m+1].s2=pos[m].s2;
                   pos[m+1].weidu=pos[m].weidu;
                    pos[m+1].jingdu=pos[m].jingdu;
                    pos[m+1].day_order=pos[m].day_order;

               }
               else break;
           }
            if(m==n-1)  continue;
           pos[m+1].s2=s2;
           pos[m+1].weidu=weidu;
           pos[m+1].jingdu=jingdu;
            pos[m+1].day_order=date_order;
       }
    }
}
```

```
    int ii;
    for(ii=0;ii<n;ii++)
    {
       cout<<pos[ii].s2<<"  "<<pos[ii].weidu<<"  "<<pos[ii].jingdu<<"
"<<pos[ii].day_order<<endl;
    }
    for(ii=0;ii<5;ii++)  cout<<endl;
    double span_jingdu2=0.01;
    double span_weidu2=0.01;
    double span_date2=1;
    for(l=0;l<n;l++)//n 个点
    {
       int l3=pos[l].day_order-span_date*2;
       int r3=pos[l].day_order+span_date*2;
       for(o=l3;o<=r3&&o<=365;o+=span_date2)
       {
          date_order=o;
          cal_chiwei();
           double l1=pos[l].jingdu-span_jingdu*3;
           double r1=pos[l].jingdu+span_jingdu*3;
           for(i=l1;i<=r1;i+=span_jingdu2)
           {
              jingdu=i;
              double l2=pos[l].weidu-span_weidu*3;
               double r2=pos[l].weidu+span_weidu*3;
              for(j=l2;j<=r2;j+=span_weidu2)
              {
                 weidu=j;
                 double s2=0;
                 for(m=0;m<num;m++)
                 {
                    time_10=tim[m];
                    double xt=cal_x();
                    s2+=(xt-len[m])*(xt-len[m]);
                 }
                 if(s2<pos2[l].s2)
                 {
                    pos2[l].s2=s2;
                    pos2[l].weidu=weidu;
                    pos2[l].jingdu=jingdu;
                    pos2[l].day_order=date_order;
                 }

              }
```

```
            }
        }
    }
    sort(pos2,pos2+n,cmp);
    for(ii=0;ii<n;ii++)
    {

    //printf("%.15lf    %.3lf    %.3lf     %.1lf\n",pos2[ii].s2,pos2[ii]
.weidu*180/pi,pos2[ii].jingdu,pos2[ii].height);
        cout<<pos2[ii].s2<<"                          "<<pos2[ii].weidu<<"
"<<pos2[ii].jingdu<<"  "<<pos2[ii].day_order<<endl;
    }
    return 0;
}

///
void init()
{
    int i;
    for(i=0;i<num;i++)   cin>>timm[i];
    for(i=0;i<num;i++)   cin>>tims[i];
    for(i=0;i<num;i++)   cin>>x[i];
    for(i=0;i<num;i++)   cin>>y[i];
    for(i=0;i<num;i++)
len_p[i]=sqrt((x[i]-0x)*(x[i]-0x)+(y[i]-0y)*(y[i]-0y));
    for(i=0;i<num;i++)   tim[i]=timm[i]+tims[i]/60.0;
    double k=height/height_p;
    for(i=0;i<num;i++)   len[i]=len_p[i]*k;
    for(i=0;i<n;i++)
    {
        pos[i].s2=999999999;
        pos2[i].s2=999999999;
    }
    for(i=0;i<num;i++)   cout<<len[i]<<"  ";
    cout<<endl;
}

int cal_N(int a,int b)
{
    int n=0;
    switch(a)
    {
        case 12: n+=30;
        case 11: n+=31;
```

```
        case 10: n+=30;
        case 9: n+=31;
        case 8: n+=31;
        case 7: n+=30;
        case 6: n+=31;
        case 5: n+=30;
        case 4: n+=31;
        case 3: n+=28;
        case 2: n+=31;
        case 1: ;
    }
    n+=b;
    return n;
}

void cal_rijiao()
{
    double N=date_order;
    double N0=79.6764+0.2422*(t_y-1985)-(int)((t_y-1985)/4);
    double t=N-N0;
    rijiao=2*pi*t/365.2422;
}

void cal_chiwei()
{
    cal_rijiao();
    chiwei=(0.3723+23.2567*sin(rijiao)+0.1149*sin(2*rijiao)-0.1712*si
n(3*rijiao)-0.758*cos(rijiao)+0.3656*cos(2*rijiao)+0.0201*cos(3*rijia
o))*pi/180.0;
}

void cal_shijiao()
{
    time_10=time_10+(jingdu-120)/15.0;
    double ans=(time_10-12)*15;
    shijiao=ans*pi/180.0;
}

double cal_x()
{
    cal_shijiao();
    double weidu_pi=weidu*pi/180.0;
    double
sin_hs=sin(chiwei)*sin(weidu_pi)+cos(chiwei)*cos(weidu_pi)*cos(shijia
```

```
o);
    double cos_hs=sqrt(1-sin_hs*sin_hs);
    double tan_hs=sin_hs/cos_hs;
    return height/tan_hs;
}
/*


8     8     8     9     9     9     9     9     9     9     9
9     9     9     9     9     9     9     9     9
55    57    59    1     3     5     7     9     11    13    15
17    19    21    23    25    27    29    31    33
1669  1663  1654  1644  1632  1624  1615  1604  1595  1586  1575
1571  1556  1551  1538  1530  1518  1512  1505  1495
868   868   870   871   871   873   874   875   875   877   878
879   880   881   881   881   883   883   883   884

*/
```