

# 全国第六届研究生数学建模竞赛



## 题 目 枪弹头痕迹自动比对方法的研究

---

### 摘 要：

我们完整的解决命题人提出的四个问题.

对于问题一，我们利用数据本身，给出一个计算并消除两组划痕数据之间平移误差和转动误差的精确算法（3. 1. 3 节）. 特别是在确定圆心坐标时我们给出了一个二元二次方程组（3. 1 式）的解析表达式（附录一），既提高了运算速度又极大地提高了精度. 在找出一对对应点以确定转动误差时，给出了一个精确的算法. 试验表明计算出两种误差值都在题目所给范围之内. 通过改变算法的一些参数，我们的算法可以适用各种型号的子弹比对（附录二）.

对于问题二，针对不同噪声和误差，我们采用了 2 种滤波方案. 其中中值滤波+低通滤波+高斯滤波的方案的效果较好.

对于问题三，通过对过滤后的数据进行分析，提取了一维特征向量并提出了比对特征相关系数作为检验弹头相似程度的标准.

对于问题四，使用问题三的比对方法，我们得到了 22 颗子弹相互之间的相似度，并利用这个相似度矩阵，得到了每个子弹的 5 个相似度较高的弹头编号.

关键词：枪弹头痕迹 自动比对 精确解析算法

## 1. 问题的提出

本题提供的数据中涉及的枪支与枪弹主要是 77 制式手枪及与之匹配的枪弹。

枪弹发射后，弹头上留下了枪管膛线（参见示意图 1）的擦痕痕迹。枪管有 4 条凸膛线共 8 个棱，分为 4 个主棱和 4 个次棱。所以在弹头上留下的痕迹分为 4 片主棱线的痕迹和 4 片次棱线的痕迹。由于枪弹通过枪管时只旋转了一个很小的角度，擦痕分布成斜线状（与圆柱母线有一个夹角），每一片痕迹的主要部分都显示为不同大小和不同深浅的线条（如图 2）。

在公安实践中，要根据弹头上的痕迹来判断两个弹头是否为同一支枪发射的。传统方法是通过显微镜肉眼观察，比对两个弹头上的线条型痕迹，看这些线条的粗细分布是否吻合。根据实践经验，对同一支枪发射的两个弹头，4 个主棱中通常有一至二个吻合得比较好；而 4 个次棱中通常可以有 2 至 3 个吻合比较好。

传统做法有两个弱点：一是效率极低，因为很多情况是“似是而非”的，所以对多个弹头（例如几十、几百个）的比对几乎无法进行；二是弹头样本不易保存，容易发生锈蚀损坏等情况。

现代高精度数据采集设备为自动比对方法创造了条件。自动比对方法的过程分为两步：

第一步，通过光学设备（如图 3，图 4 示意）采集弹头上 8 片痕迹的 3 维数据，保存为 8 个文件。测量的基准平面取为固定在测量设备上的空间直角坐标系的  $xoy$  平面，沿  $x$  轴方向和沿  $y$  轴方向的测量步长均为 2.75 微米， $z$  的测量精度为 1 微米，数据的单位是毫米。由于数据量很大，本题只提供对应 4 条次棱的数据，其顺序统一按照一个方向排序，例如按照从弹头的底部向头部看去为逆时针方向排序。这些数据文件可以用 Matlab 的数据导入打开，也可以用写字板打开。每个文件的数据有 40 多万行，每行有 3 列。每一行的三个数据对应弹头表面上一个点的空间坐标  $(x, y, z)$ ，第 1 列对应  $x$  坐标；第 2 列对应  $y$  坐标；第 3 列对应  $z$  坐标。基准平面在弹头痕迹的附近，测量时应调节弹头的姿态使得：（1）弹头圆柱中心线尽量平行于基准平面；（2） $y$  轴尽量平行于擦痕的走向。弹头的姿态靠人工调节，所以上述的两个平行都不是准确的平行。此外，这 4 条次棱的数据是分 4 次测量的，所以它们只能在 4 个独立的空间坐标系中分别显示它们的图像，没有一个参照系能使它们组合成一个整体图像在一个坐标系中显示。

第二步，采用适当的方法，通过电脑比对，判别一个弹头与样本弹头的相似程度，以便确认发射该弹头的枪支。这个问题很实用，但要求达到较高的准确性就很困难。希望研究生们研究下面几个问题：

- 1、在光学设备上采集弹头上痕迹的 3 维数据时，用于采集数据的光源和数据采集仪器是固定不动的，光源在被测弹头的侧上方，感光器在弹头的正上方；

弹头由人工固定在支架上，可以由人工调节该弹头的位置和姿态（姿态是指弹头在空间的姿势或状态，通常与转动有关）。人工调节很难使两个弹头在同样位置和同样姿态情况下被测量，从而会造成测量误差。通常会造成 0.03mm 左右的平移误差和  $0.2^\circ$  的转动误差。第一个问题是：如何处理由位置和姿态造成的测量误差，使得两个弹头可以尽量在相同位置和相同姿态的情况下进行比对。（这里假设弹头的直径为 7.90mm，长度约为 12mm）

- 2、弹头表面的损伤、锈迹、油斑、杂质物及痕迹生成的随机性，会造成数据误差和噪声。第二个问题是：采取怎样的方法去消除这些误差和噪声。
- 3、文件名以 77 开头的 12 个文件分别是 6 支枪发射的 12 个弹头（每支枪发射 2 个弹头）的次棱部分的测量数据，每个文件包含有 4 个以 c 开头的次棱数据子文件。数据文件名中的 t1 和 t2 分别表示对应同一支枪的 2 个弹头，其他数字是枪支的编号；子文件名中的 c1, c2, c3, c4 分别为同一个弹头的 4 个次棱按固定顺序的编号。第三个问题是：
  - (1) 依据这些数据，你们认为怎样的特征可以用于比对；并给出提取这些特征的方法。
  - (2) 依据这些数据，你们认为采用数据的哪一部分用作比对，其效果比较好。
  - (3) 给出完整的比对方案、算法，并在电子版附件中给出程序。
- 4、文件名以 t 开头的 22 个文件分别是另外 11 支枪发射的 22 个弹头（每支枪发射 2 个弹头）次棱部分的测量数据，每个文件包含有 4 个以 c 开头的次棱数据子文件，子文件名中的 c1, c2, c3, c4 的意义同上。第四个问题是：请你们，
  - (1) 用解决第三个问题的方法给出这 22 个弹头痕迹两两之间的相似程度，并列表表示；
  - (2) 根据 (1) 的结果，用列表的方式给出每个弹头按相似度由高到低给出与之相似的前 5 位的弹头文件号。

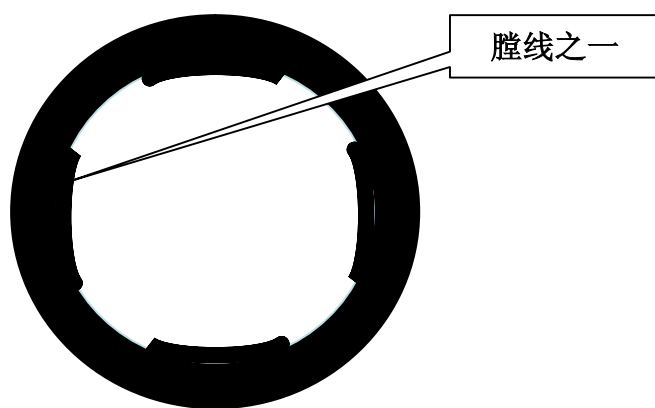


图 1：枪管截面的膛线示意图



图 2：某一次棱的痕迹放大照片

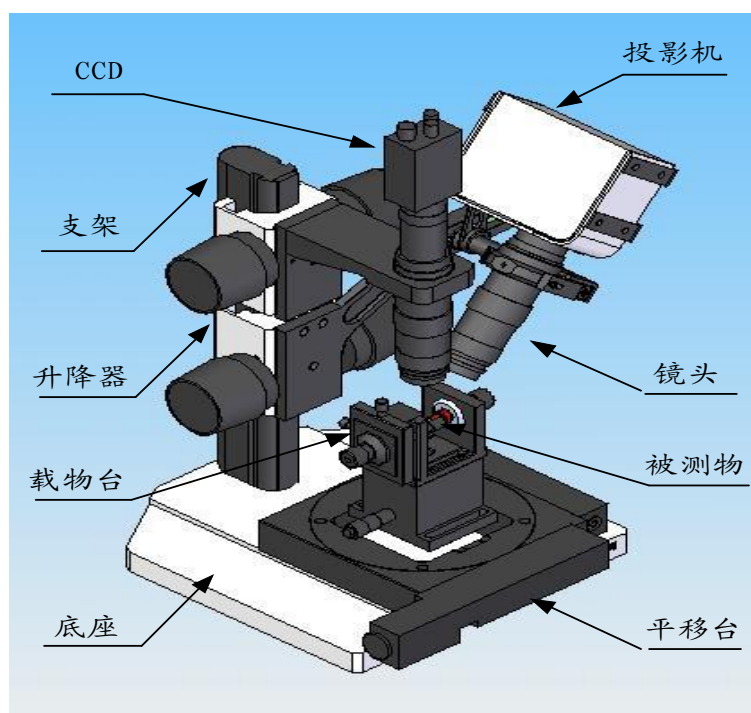


图 3：数据采集的机械与光学装置部分示意图



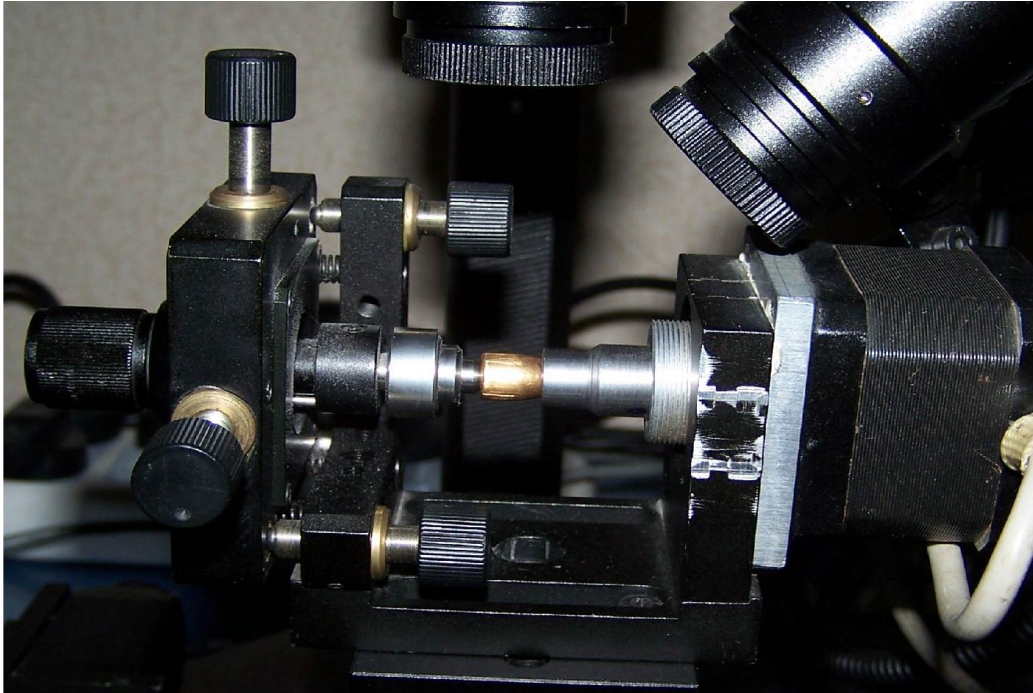


图 4：测量台部分的手动调节装置

## 2. 模型假设与约定

1. 子弹的形状为圆柱体.；
2. 子弹发射后未产生形变；
3. 子弹的次棱划痕在被扫描时，平移误差只产生于沿  $x$  轴和  $z$  轴的平移；
4. 子弹的次棱划痕在被扫描时，不存在倾斜所产生的误差；
5. 一组数据仅含一条次棱划痕上点的坐标；
6. 忽略由  $y$  轴与子弹旋转轴之间夹角

## 3. 问题的解答

### 3. 1 问题一

#### 3. 1. 1 问题一中的符号说明

$\Delta x$ ：对于两次扫描得到的坐标数据，两者之间  $x$  轴方向的平移误差

$\Delta z$ ：对于两次扫描得到的坐标数据，两者之间  $z$  轴方向的平移误差

#### 3. 1. 2 问题一的分析：

问题 1 的目的是要消除仪器扫描时子弹放置的平移误差和旋转误差照成的

对应点的误差. 对于两次扫描得到的坐标数据, 两者之间的平移误差 (包括  $x$  轴方向的平移误差  $\Delta x$  和  $z$  轴方向的平移误差  $\Delta z$ ) 和转动误差  $\Delta \theta$  是未知的. 所以我们要从两次数据本身计算出  $\Delta x$ ,  $\Delta z$  和  $\Delta \theta$ . 然后根据坐标变换公式即可将其之中的坐标值变换到另外一次扫描的坐标系下的坐标值. 这样就可以在相同位置和相同姿态的情况下进行比对.

首先,  $y$  轴并不平行于子弹的中轴, 所以平行于平面  $xoz$  的子弹切面一般不是圆形. 但是在子弹正上方表面的小块区域里, 切面的边缘和圆弧非常接近 (图 3. 1. 1). 故可以假设切面的边缘是圆弧.

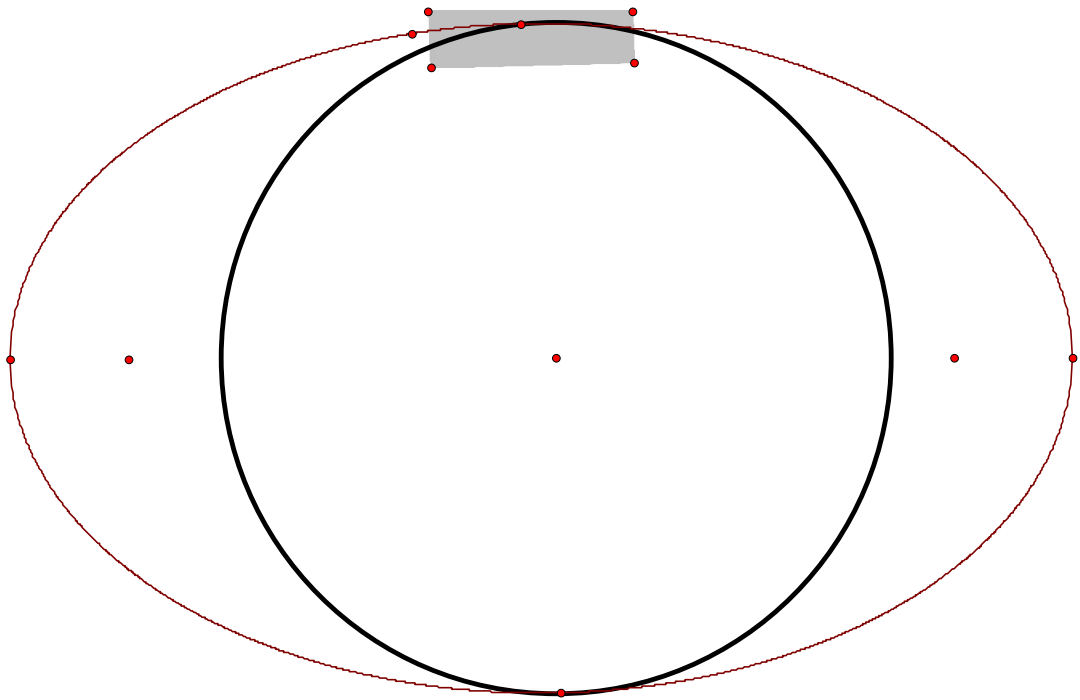


图 3. 1. 1

另外对于周长为  $7900\pi = 24818.6$  微米的子弹的次棱划痕的深度相对于圆周的影响是非常小的. 故我们就可以考虑利用统计和解析的方法求出切面圆心的坐标 (图 3. 1. 2). 两次扫描的数据的球心皆可求出,  $\Delta x$  和  $\Delta z$  就能得到.

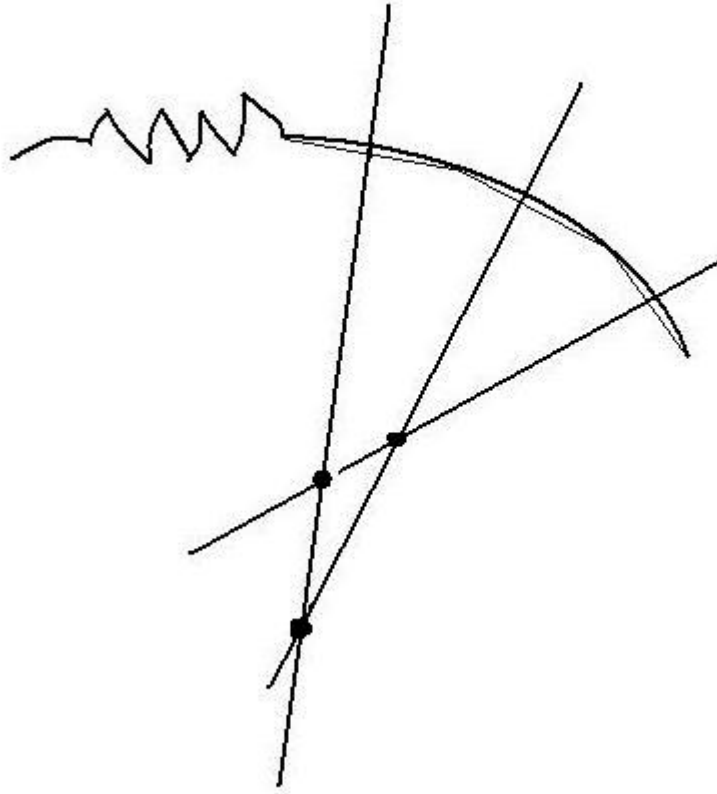


图 3. 1. 2

然后，对其中之一次扫描的数据进行平移变换得到新的坐标。在平行于  $xoz$  平面的切面上，运用一个判别算法得到满足一定条件的两组数据的分界点。此二点之间的球面距离（也是弧长）对应的圆心角即为  $\Delta z$ （图 6）。

最后进行旋转变换。将两次扫描数据沿  $x$  轴的公共部分保留，其余部分剔除，得到两组用于比对的新数据。

### 3. 1. 3 问题一的数据处理算法

根据 3. 1. 2 中的分析，下面给出具体的数据处理算法：

两次扫描得到的数据分别为  $(x_n, y_n, z_n)$ ， $n=1, 2, 3, \dots, m$  和  $(x'_n, y'_n, z'_n)$ ， $n=1, 2, 3, \dots, m$ 。为了叙述方便，我们令  $(x_n, y_n, z_n)$  数据分段方法与问题中的数据分段方法一致。即：

$$m = 426384 ;$$

$$x_n = [n(\bmod 564)] \times 0.00275 ;$$

$$y_n = 0.00275 \cdot \left\lfloor \frac{n-1}{756} \right\rfloor ;$$

其中  $n \bmod(564)$  表示  $n$  除以 564 的余数； $\lfloor \rfloor$  表示向下取整.

第一步：计算  $\Delta x$  和  $\Delta z$

依靠经验或者观察图形，总可以选取一个正整数  $\lambda < 564$ ，使得对于满足  $[n \bmod(564)] + 1 \geq \lambda$  的任意  $n$ ，令

$$\overline{x}_i = (i-1) \times 0.00275 ;$$

$$\overline{z}_i = \frac{\sum_{j=1}^{756} z_{i+564 \times (j-1)}}{756} ;$$

$$i = \lambda, \lambda+1, \dots, 564 .$$

我们将平行于平面  $xoz$ ，子弹的切面图形看做圆形进行处理，切面图形的边界最上面部分是和圆弧最接近的地方，我们这种近似处理的方法是有意义的.

确定圆心的方法可以有很多种，比如拟合圆弧法和垂直平分线交点法. 由于子弹半径已知，为了提高精确度和稳定性我们使用如下算法：

1)  $a = \lambda$ ， $b = 564$ ， $i = 1$ ， $t = 0$ ， $s = 0$ ，转 2)

2) 对二元一次方程组

$$\begin{cases} (\overline{x}_a - x)^2 + (\overline{z}_a - z)^2 = 3.95^2 \\ (\overline{x}_b - x)^2 + (\overline{z}_b - z)^2 = 3.95^2 \end{cases} \quad (3.1)$$

取满足  $x < \overline{x}_b$  和  $z < \overline{z}_b$  的那组解. 转 3)

3)  $t = t + x$ ， $s = s + z$  转 4)

4) 若  $b - a \geq \mu$  转 5); 若  $b - a < \mu$  转 6)

5)  $a = a + 1$ ， $b = b - 1$ ； $i = i + 1$  转 2)

6) 圆心坐标  $(x_{\text{圆心}}, z_{\text{圆心}}) = (\frac{t}{i}, \frac{s}{i})$

注意：1. 其中  $\mu$  的选取需要依靠经验和反复实验得到.

2. 如果采用最小二乘法等数值方法求解二元一次方程组不仅会降低程序



运行的速度，而且会降低精度。所以我们采用此二元一次方程组的解析解。用 Mathematica 7.03 软件求解的解析表达式见附录 1。

以相同的  $\mu$ ，按照上面的算法可以得到另一组数据的圆心坐标  $x'_{\text{圆心}}$  和  $z'_{\text{圆心}}$ 。

故  $\Delta x = x'_{\text{圆心}} - x_{\text{圆心}}$ ， $\Delta z = z'_{\text{圆心}} - z_{\text{圆心}}$ 。

第二步：将第一组数据，经过平移变换

$$\begin{cases} x^{\text{平移}}_n = x_n + \Delta x \\ z^{\text{平移}}_n = z_n + \Delta z \end{cases}$$

得到新的一组数据  $(x^{\text{平移}}_n, y_n, z^{\text{平移}}_n)$ 。

第三步：计算  $\Delta\theta$ 。

按照下面的算法确定平行于平面  $xoz$  的子弹切面上的损坏点与未损坏点的分界点的坐标：令

$$\overline{x^{\text{平移}}_i} = (i-1) \times 0.00275;$$

$$\overline{z^{\text{平移}}_i} = \frac{\sum_{j=1}^{756} z^{\text{平移}}_{i+564 \times (j-1)}}{756}.$$

1) 令  $h = \lambda$ , 转 2)

2) 若  $\frac{\left| \sqrt{(x'_{\text{圆心}} - \overline{x^{\text{平移}}_h})^2 + (z'_{\text{圆心}} - \overline{z^{\text{平移}}_h})^2} - 3.95 \right|}{3.95} < \beta$ , 则转 3);

若  $\frac{\left| \sqrt{(x'_{\text{圆心}} - \overline{x^{\text{平移}}_h})^2 + (z'_{\text{圆心}} - \overline{z^{\text{平移}}_h})^2} - 3.95 \right|}{3.95} \geq \beta$ , 则转 4);

3)  $h = h-1$ , 转 2);

4) 数据  $(x^{\text{平移}}_n, y_n, z^{\text{平移}}_n)$  分界点的坐标为  $(\overline{x^{\text{平移}}_h}, \overline{z^{\text{平移}}_h})$ 。

使用相同的  $\beta$ ，同理可以得到数据  $(x'_n, y'_n, z'_n)$  的分界点坐标为  $(\overline{x'_{h'}}, \overline{z'_{h'}})$ 。我们得到：

$$\Delta\theta = \arctan\left(\frac{\overline{z'_{h'}} - z'_{\text{圆心}}}{x'_{h'} - x'_{\text{圆心}}}\right) - \arctan\left(\frac{\overline{z^{\text{平移}}_h} - z'_{\text{圆心}}}{x^{\text{平移}}_h - x'_{\text{圆心}}}\right)$$

这里  $\beta$  的选取需要依靠经验和枪支弹药的类型。

第四步：对数据  $(x_n^{\text{平移}}, y_n, z_n^{\text{平移}})$  进行如下旋转变换

$$\begin{cases} x_n^{\text{新}} = x'_{\text{圆心}} + \cos(\arctan \frac{z_n^{\text{平移}} - z'_{\text{圆心}}}{x_n^{\text{平移}} - x'_{\text{圆心}}} + \Delta\theta) \times \sqrt{(z_n^{\text{平移}} - z'_{\text{圆心}})^2 + (x_n^{\text{平移}} - x'_{\text{圆心}})^2} \\ z_n^{\text{新}} = z'_{\text{圆心}} + \sin(\arctan \frac{z_n^{\text{平移}} - z'_{\text{圆心}}}{x_n^{\text{平移}} - x'_{\text{圆心}}} + \Delta\theta) \times \sqrt{(z_n^{\text{平移}} - z'_{\text{圆心}})^2 + (x_n^{\text{平移}} - x'_{\text{圆心}})^2} \end{cases}$$

得到新的数据  $(x_n^{\text{新}}, y_n, z_n^{\text{新}})$

第五步：对两组数据在  $x$  轴方向进行“裁剪”。

若  $x_1^{\text{新}} < x'_1 = 0$ ，则对  $(x_n^{\text{新}}, y_n, z_n^{\text{新}})$  做如下裁剪

- 1)  $q = 2$  转 2)
- 2) 若  $x_q^{\text{新}} < x'_1 = 0$ ，转 3)；若  $x_q^{\text{新}} \geq x'_1 = 0$ ，转 4)
- 3)  $q = q + 1$  转 2)
- 4) 对于数据  $(x_n^{\text{新}}, y_n, z_n^{\text{新}})$  每 564 行剔除前  $q-1$  行，得到用于下一步除噪的数据  $(x_n^{1\text{终}}, y_n^{1\text{终}}, z_n^{1\text{终}})$ ；对于数据  $(x'_n, y'_n, z'_n)$  每 564 行剔除倒数  $q-1$  行，得到用于下一步除噪的数据  $(x_n^{2\text{终}}, y_n^{2\text{终}}, z_n^{2\text{终}})$

若  $x_1^{\text{新}} > x'_1 = 0$ ，则对  $(x'_n, y'_n, z'_n)$  做如下裁剪

- 1)  $q = 2$  转 2)
- 2) 若  $x_q^{\text{新}} > x'_1 = 0$ ，转 3)；若  $x_q^{\text{新}} \leq x'_1 = 0$ ，转 4)
- 3)  $q = q + 1$ ，转 2)
- 4) 对于数据  $(x'_n, y'_n, z'_n)$  每 564 行剔除前  $q-1$  行，得到用于下一步除噪的数据  $(x_n^{2\text{终}}, y_n^{2\text{终}}, z_n^{2\text{终}})$ ；对于数据  $(x_n^{\text{新}}, y_n, z_n^{\text{新}})$  每 564 行剔除倒数  $q-1$  行，得到用于下一步除噪的数据  $(x_n^{1\text{终}}, y_n^{1\text{终}}, z_n^{1\text{终}})$

以上算法的 MATLAB 程序实现，见附录 2。

### 3. 1. 4 数据试验确定合适的 $\lambda$ ， $\mu$ ， $\beta$

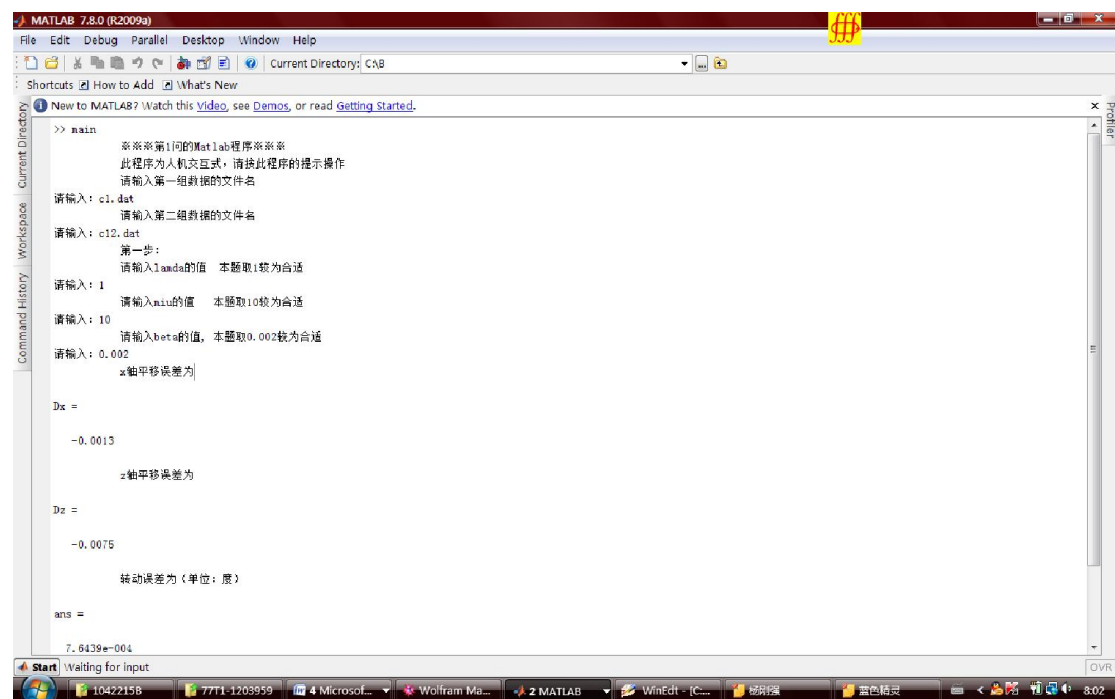
选取 77T1-1203959 中的一个次棱的数据文件 c1. dat 与 77T1-1203959 中一个次棱的数据文件 C2. Dat（重命名为 c12. dat）。用附录 2 中的交互程序反复手动实验比较 100 多次，得到

$$\lambda=1;$$

$$\mu=10;$$

$$\beta=0.002$$

时能够得十分理想的结果（图 3. 1. 3）



结果：

x 轴平移误差为

$$D_x =$$

$$-0.0013$$

z 轴平移误差为

$$D_z =$$

$$-0.0075$$

转动误差为（单位：度）

ans =

7.6439e-004

结果非常好.

### 3. 1. 5 算法的优点

1. 可以直接利用数据本身的来确定误差. 两种误差值是不可能预知的.
2. 由于在关键值的计算方面采用了取平均的方法, 算法的稳定性较高, 有一定的抗噪性.
3. 使用了解析表达式, 提高了精度了计算速度.

### 3. 1. 6 算法的缺点和改进

1. 一些参数的确定使用反复手动试验得到, 有一定的随意性.
2. 将切面看做圆面导致计算必然存在误差. 但是如果将切面看做椭圆形处理, 已知量子弹半径的将不能被使用,  $y$  轴与子弹中轴夹角未知, 只能用拟合和匹配等非解析方法来做, 仍然会有误差.

## 3.2 问题二

### 3.2.1 问题二的分析

噪声可以理解“妨碍人们感觉器官对所接收信息理解的因素”, 噪声在理论上可以定义为“不可预测, 只能用概率统计方法认识的随机误差”. 描述噪声完全可以借用随机过程及其概率分布函数和概率密度函数, 通常是用其数字特征, 即均值方差、相关函数等进行处理.

图像噪声一般分为外部噪声和内部噪声, 外部噪声主要有外部的作用产生的, 内部噪声器件内部本身的性质产生的. 对于弹头表面的损伤、锈迹、油斑、杂质物及痕迹生成的随机性, 会造成数据误差和噪声, 我们认为主要产生如下几种噪声: 1. 椒盐噪声 2. 高频杂波分量 3. 高斯噪声.

解决问题的两种方案:

- ①中值滤波+ 低通滤波+高斯滤波
- ②小波滤波

#### 3.2.1.1 两种方案的优缺点比较

方案一 优点: 采用多种滤波方法同时对Z轴划痕深度信号进行滤波, 能够针对性的滤除各种随机噪声, 如中值滤波可以很好的去除椒盐噪声, 低通滤波器可以很好的去除高频杂波分量, 高斯滤波器可以很好的滤除高斯噪声, 因此会达到很好的滤波效果. 缺点: 多个滤波器在一起实现比较困难, 容易引入新的杂

波分量，而且运行时间长。

方案二 优点：实现简单，运行时间短。 缺点：小波去噪一般对高斯噪声比较敏感，对其它的噪声滤除效果不好，后面有图进行对比。

经过综合考虑我们选择方案一，后面的实验也证明方案一的效果比方案二好。

### 3.2.2 方案一的具体实施

说明：为了分析问题的方便性与透彻性，我们选取枪编号为77T1-1203959 的第一个子弹的数据c1. dat。

首先我们选取子弹次棱的中间的三分之一作为有效数据分析的对象，具体原因参考[1]问题分析的简单性，即只分析怎样滤除噪声，不考虑特征向量的提取，只选取第250行的y轴数据进行分析. 当y固定时即可只考虑xoz平面上的截面。

#### 3.2.2.1 .y 固定时 XoZ 截面的初始值的截面

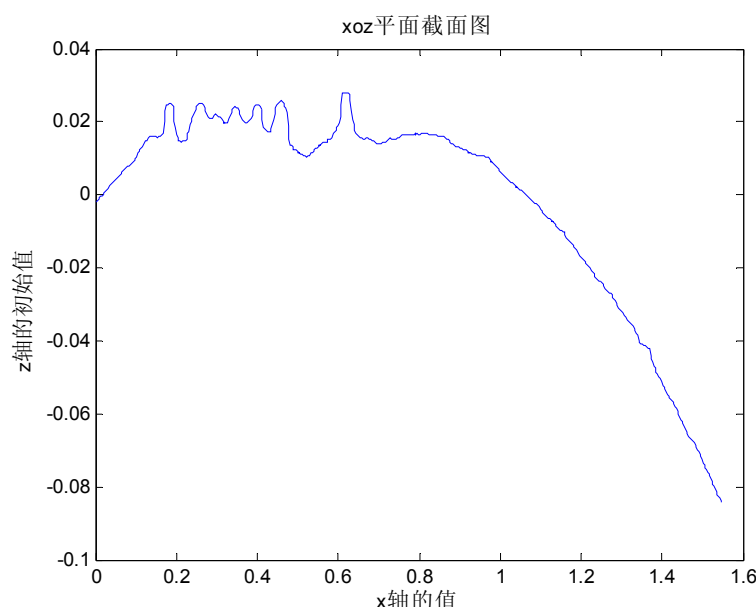


图3.2.2.1

由上图可见在X为[0 0.6]的范围内划痕的深度清晰可见，但是特征还不是很明显，是由于有误差和噪声叠加的缘故。

#### 3. 2.2.2 中值滤波

中值滤波是一种非线性滤波，它只能用数字方式实现，即只能在数字计算装置上实现。中值滤波对滤除脉冲干扰特别有用。在图像增强处理中用来成功的消除随机噪声和脉冲干扰以及图像中的扫描噪声等。中值滤波的实方式如下：

- (1)通过从图像数据的某个固定的采样窗口中取出奇数个数据进行排序。
- (2)用排序后的中值取代要处理的数据即可。

中值滤波对消除椒盐噪声非常有效，在光学测量条纹图像中作用非常大。

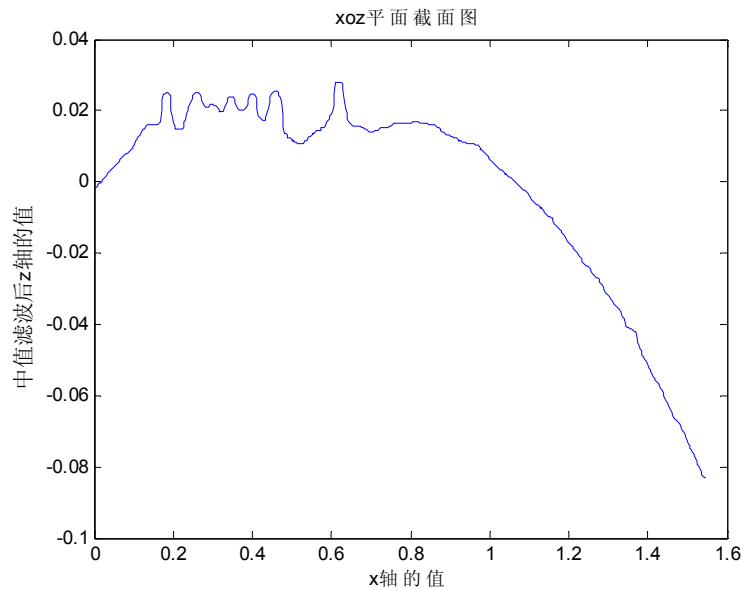


图3.2.2.2

上图为中值滤波出中值滤波消除了一些奇异的点，但是图像的整体效果变化不大，线条更加平滑。

### 3.2.2.3 低通滤波

一般来说图像的能量主要集中在低频部分，噪声所在的频段主要在高频段，有Z轴数据的频谱图可以看出（3.2.2.6），因此设计一个低通滤波器相当重要，本文采用butterworth型低通滤波器，其采样频率363HZ，截止频率100HZ，可滤除信号中参杂的高频分量。

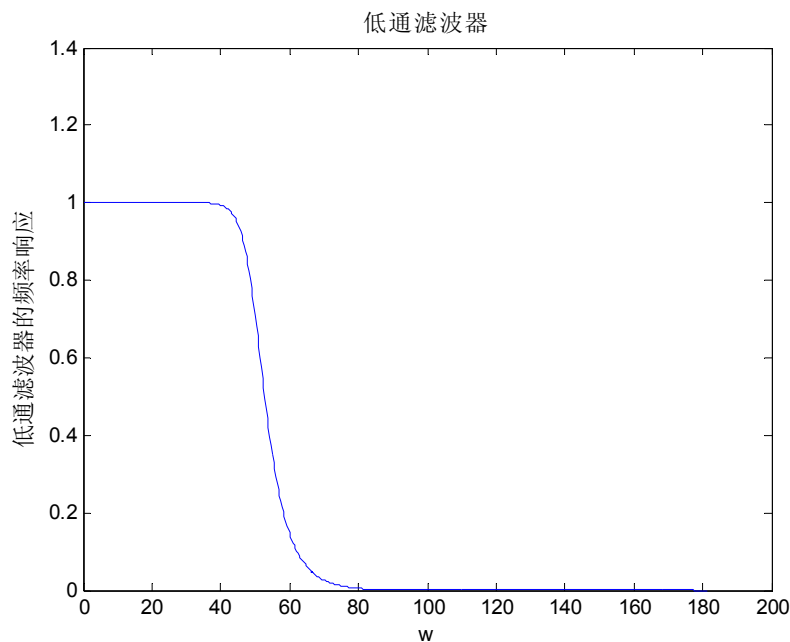


图3.2.2.3-1

上图为设计的Butterworth型低通滤波器



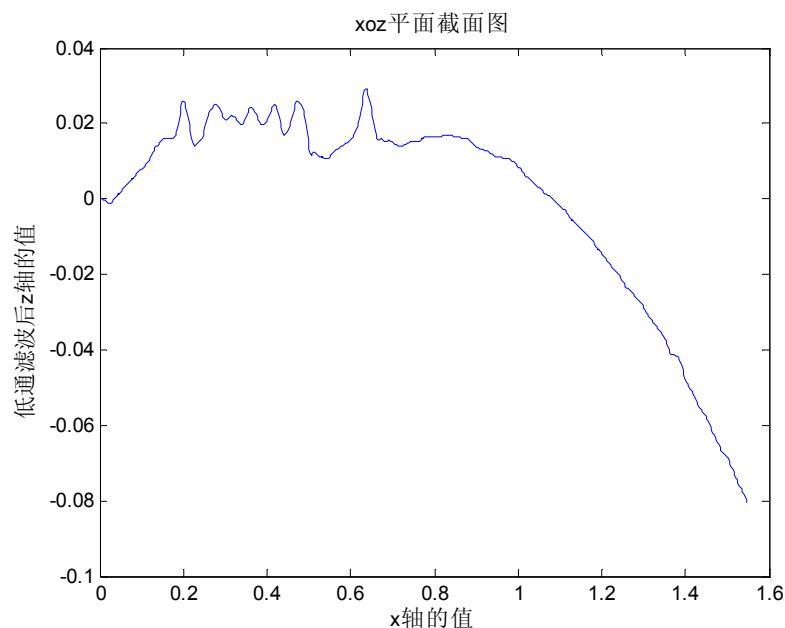


图3.2.2.3-2

上图为经过低通滤波器后的信号的波形图，由上可以看出划痕的峰值变得尖锐，特征变得更加明显，更加有利于后面的子弹的比对。这是由于滤除了高频分量的缘故。

但是不足的是在X为0的附近引入了新的误差，因此有必要进行进一步的裁剪对误差进行滤除。后面将会对此进行更加详细的分析。

#### 3.2.2.4 高斯滤波

由于随机噪声的随机性，以及子弹本身的特性，高斯噪声的存在是显而易见的，因此有必要在以上几种滤波的基础上进行高斯滤波，来滤除高斯噪声。相关的程序见附件。

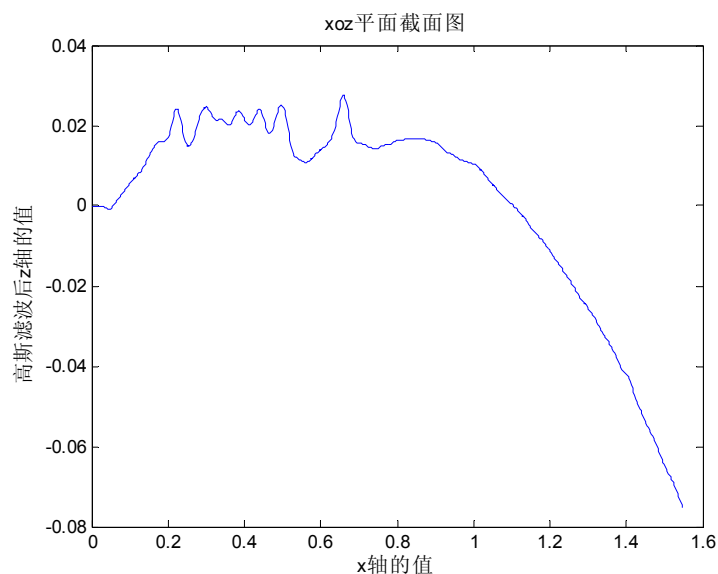


图3.2.2.4

由图可知经过高斯滤波后图像更加平滑，纹理更加突出，更能体现划痕的特征，但是在X为0的附近的噪声没有去除，下面我们将采用一种裁剪的方法来去除在X为0处的噪声。

### 3.2.2.5 去除 X 为 0 处的误差

采取裁剪的方法可以很好的去除此误差，观察可知在X为0处的有效信息并不是很多，因此可以采取去除X轴上初始的几个点的方法进行去除，取出后的效果如下：

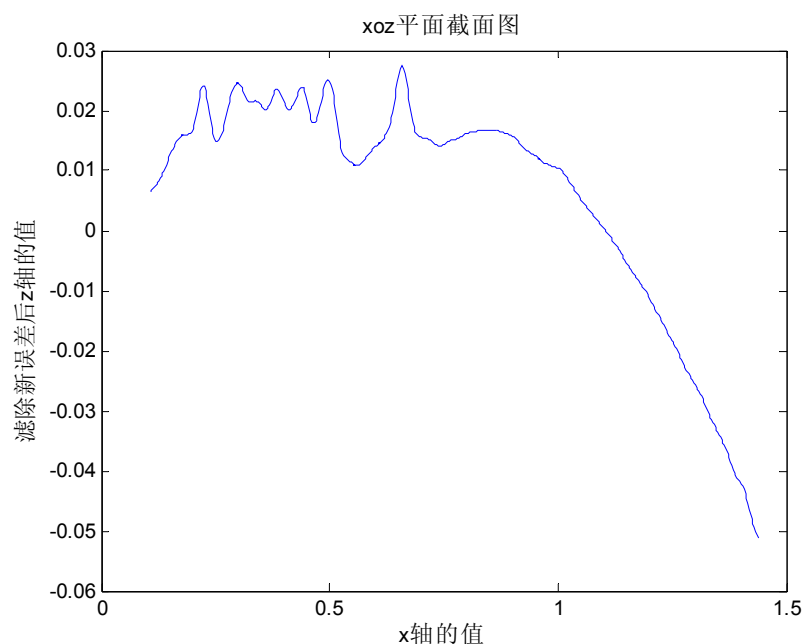


图3.2.2.5

滤除有误差的几个之后即可得到整个系统滤波后清晰明确的图像，可以为第三问的数据处理，比对提供一个更加准确的数据。

### 3.2.2.6 滤波前后信号的频谱

把滤波前后的信号频谱进行比对可知信号的有用信息得以保留，而噪声的频谱被滤除。

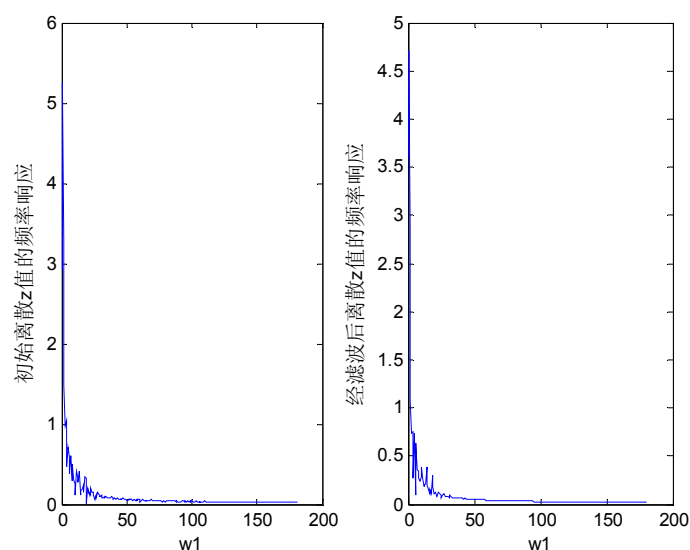


图3.2.2.6

可以看出信号的频率主要集中在50HZ以内，经过整个的滤波后效果比较好。

### 3.2.3 方案二的实施

采用小波变换进行分析，小波去噪也是一种非常有用的工具，在图像处理中，小波也发挥着很重要的作用，但对于此问题来讲，滤波后的效果本身不是很理想。

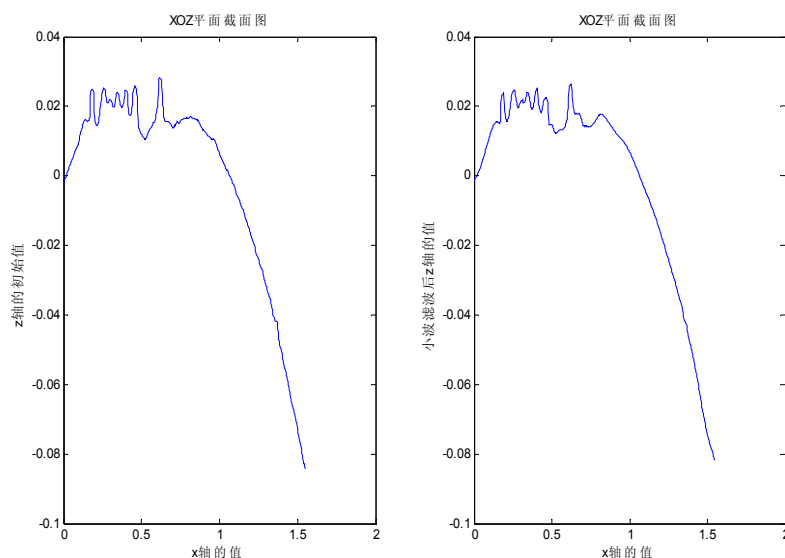


图3.2.3

由滤波后的效果图我们可以看出划痕的峰值有所减弱，没有方案一的效果好，但并不是说明此种方案不好，对于本题而言，在允许的误差范围内，此种滤波完全可以满足题目中的要求。鉴于我们提出了两种方案，我们优先选择滤波效果好的方案一。

### 3.2.4 方案一的算法实现(具体的matlab程序见附件3\第二问程序\2.1)

第一步：首先产生一个XOZ平面，选取250行的XOZ截面为研究对象。

第二步：将信号通过中值滤波器，采用自编的medianfun函数进行滤波，该中值滤波器的采样窗口为5，输入为初始的Z值，输出为经过中值滤波后的Z值。

第三步：设计butterworth低通滤波器，该滤波器的为9阶，采样频率为363HZ，截止频率为100HZ。通过filter函数对第二步的信号值进行滤波，得到经过低通滤波器后的Z的值。

第四步：通过函数gaussfiltcoef产生高斯滤波器的系数，然后通过filter函数完成第三步信号值的高斯滤波。

第五步：信号的剪除。通过去除前面的X轴上的四十个点，可以去除油低通滤波器带来的误差，得到更加清晰，明确的Z值。

此Z值即可用于第三问数据特征的提取。

### 3.2.5 方案二的算法实现 (附件 3\第二问程序\2.2)

第一步：产生一个XOZ截面（同方案一）

第二步：采用matlab中的wden函数可以进行小波去噪，其参数分别为heursure，s，min，sym8。由此即可得到小波去噪后的z值。

## 3.3 问题三

### 3.3.1 问题三中的符号说明

$c_1, c_2, c_3, c_4$  为一个弹头的 4 个次棱按固定顺序的编号。

$c_{11}, c_{12}, c_{13}, c_{14}$  为同一只枪发射出来的另一个弹头的 4 个次棱按固定顺序的编号。

### 3.3.2 问题三的分析

根据第一问和第二问的处理方法可以将初始数据进行优化处理, 可以大体消除由于人工测量产生测量误差和弹头表面的损伤、锈迹、油斑、杂质物及痕迹造成的数据误差和噪声。题目中给出了 6 只枪发射的 12 个弹头的各个次棱的观测数据, 下面要根据这些数据从中提取一些可以用于比对的特征。

首先分析一下枪号为 1203959 的手枪发出的两个弹头的数据, 试着从中找出一些有用的特征。

对 77T1-1203959 中的次棱  $c_1, c_2, c_3, c_4$  的数据进行分析。对于次棱  $c_1$ ，对其数

据进行完处理之后，画出其在  $xoz$  平面上的投影，见图 3.3.1.

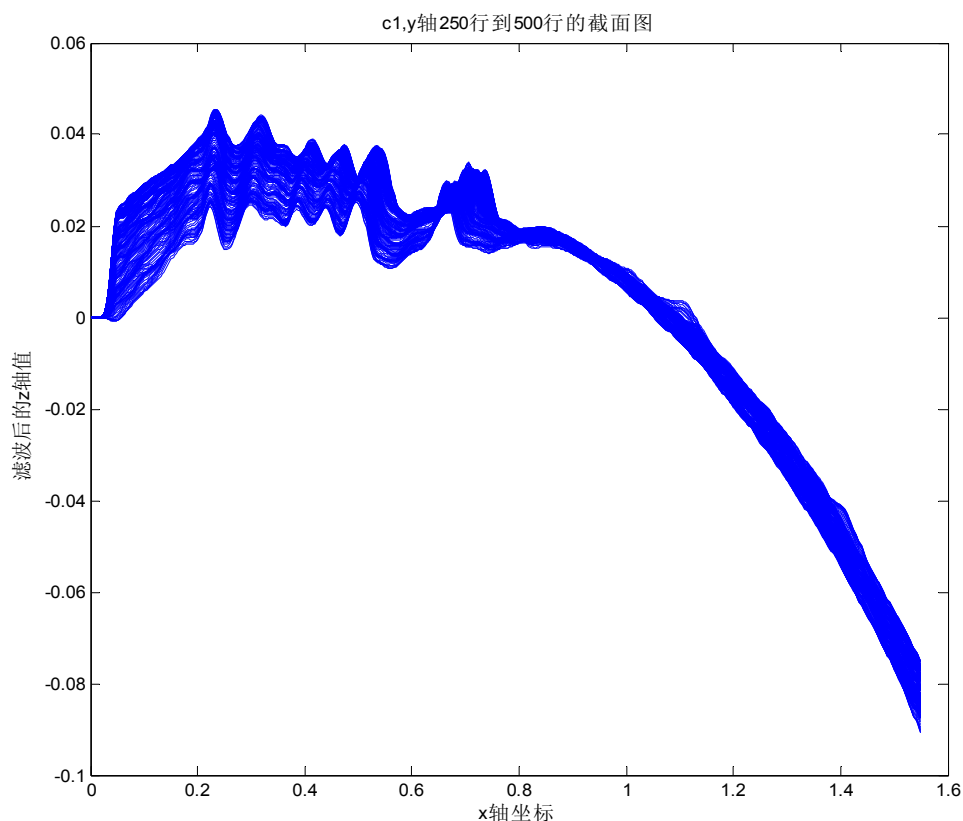


图 3.3.1

在这个图中,取了  $y$  轴第250行到第500行的数据进行投影,也就是取乐  $y$  轴数据中间的大约1/3进行的投影,这个原理根据文献[1]及实际情况,可知,子弹的次棱在子弹底部的边缘处误差较大,而根据我们的认为子弹为圆柱型的假设,以及查找文献的结果。综合考虑以上的因素,我们得到一种处理数据的方法,即选取756行 $y$ 轴值的中间1/3,250行到500行的数据进行分析,此段数据的SNR(信噪比)较高。

可以看出对  $y$  固定的任意一个截面在平面  $xoz$  上的投影形状大体相同。由于这个投影保留了这个次棱的大部分信息,可以认为若次棱投影的相似程度越高,则次棱的相似程度越高。

下面将次棱  $c1, c2, c3, c4$  的数据进行处理后在  $xoz$  平面上进行投影, 见图 3.3.2。

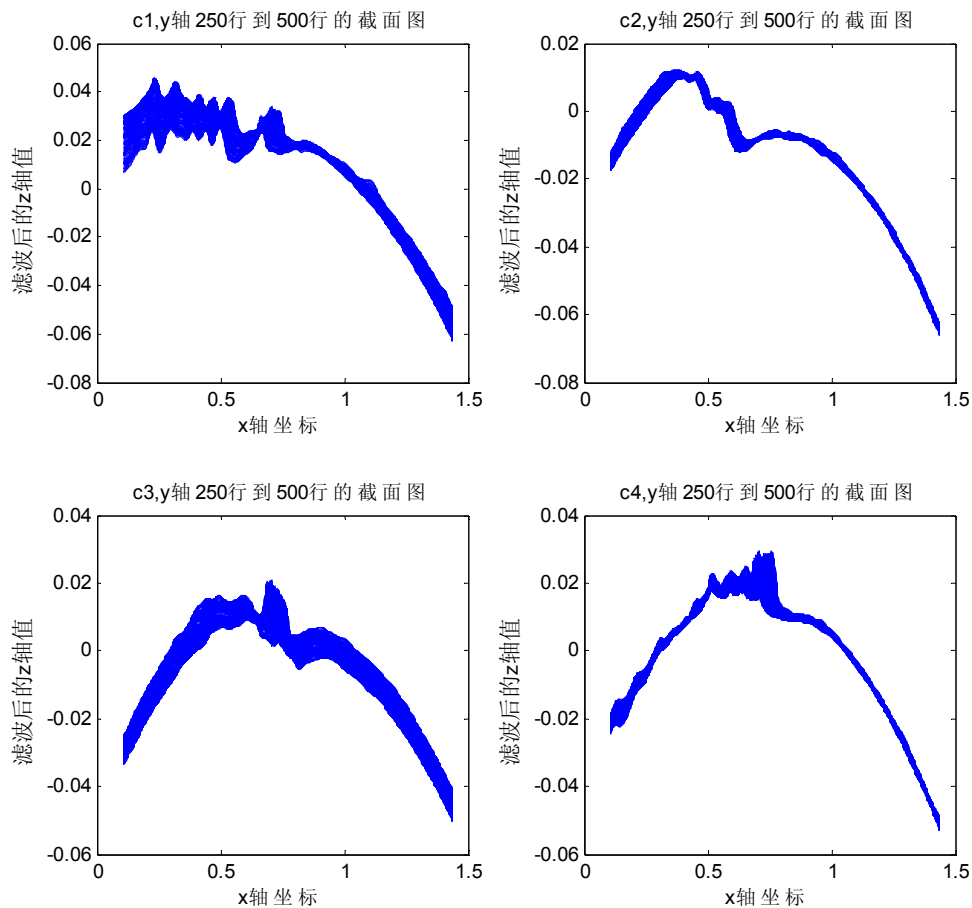


图 3.3.2

从图中可以看出对于其他次棱，它们在不同的  $y$  值情况下在平面  $xoz$  上也有相似的投影。这个也可以这样理解，由于每个次棱是由于膛线在子弹上留下的痕迹，而膛线的形状是一定的，所以每个次棱的形状应该与膛线的形状相对应，虽然深度（ $z$  坐标）可能不同，但在平面  $xoz$  上的投影的形状应该基本一致的。

基于上面的分析，一个次棱在不同  $y$  值情况下在平面  $xoz$  上投影形状基本一致，可以用一条曲线来表示这个投影的形状。这个曲线可以这样取，即在每一个  $x$  值上取所对应所有  $z$  值的均值，这样可以产生一个一维的  $z$  值的向量，因此可以产生一个可以代表次棱在平面  $xoz$  上投影形状的曲线，见图 3.3.3。



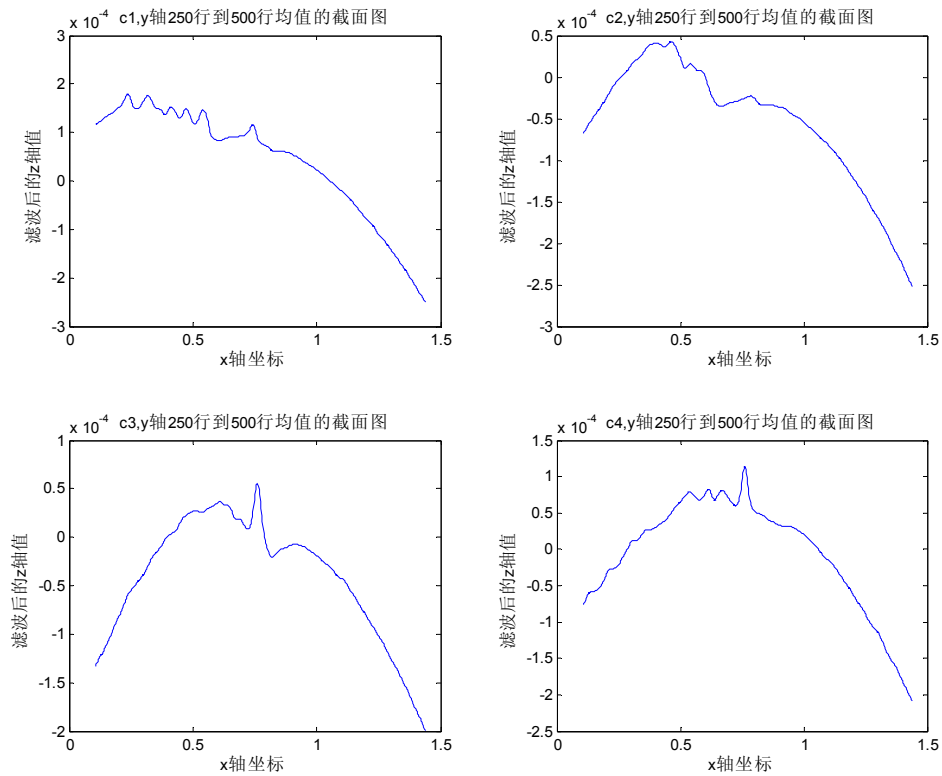


图 3. 3

比较图 3.3.2 和图 3.3.3, 可以看出按照上述方法所取的曲线可以基本代表次棱在平面  $xoz$  上投影的形状。将上面产生的曲线记做代表这个次棱的**特征曲线**, 将上面产生一维  $z$  值向量记做代表这个次棱的**一维特征向量**。下面利用 77T2-1203959 中的次棱  $c11, c12, c13, c14$  的数据来验证一下这个特征。

首先将次棱  $c11, c12, c13, c14$  的数据按照问题一和问题二所提的方法进行处理, 然后将处理后的次棱数据在平面  $xoz$  上进行投影, 观察投影结果, 如图 3.3.4。

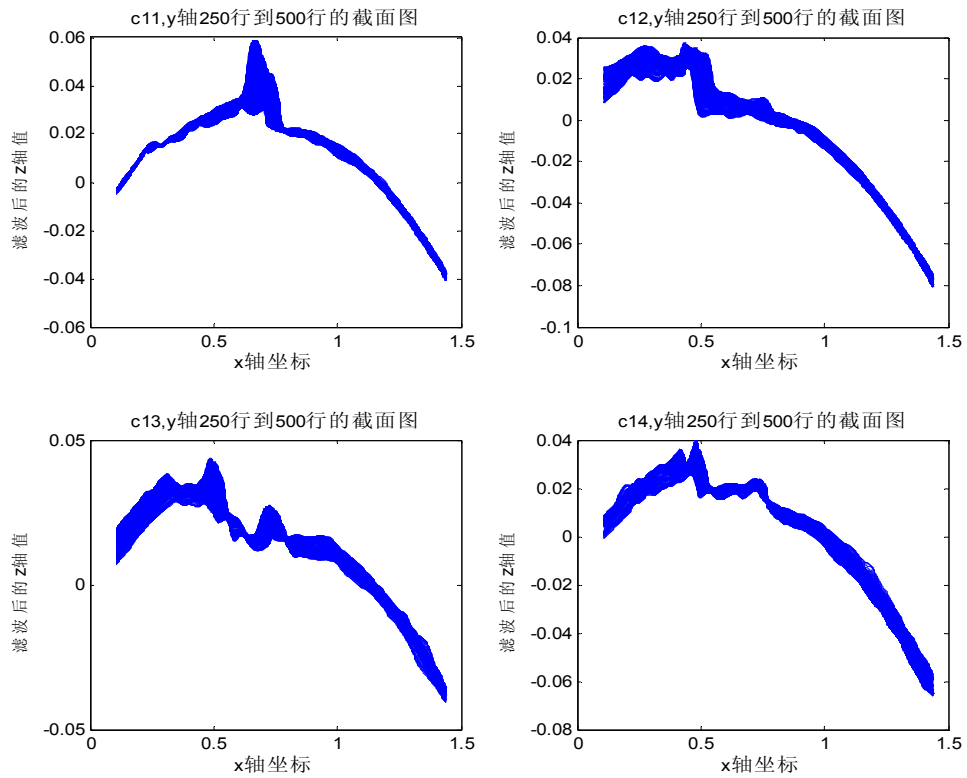


图 3.3.4

将图 3.3.2 和图 3.3.4 比较可以看出次棱  $c1$  与次棱  $c12$ , 次棱  $c4$  和次棱  $c11$  很相似, 而次棱  $c2$  与次棱  $c13$ , 次棱  $c3$  与次棱  $c14$  基本相似, 可以初步断定次棱  $c1, c2, c3, c4$  与次棱  $c11, c12, c13, c14$  相对应。

求出次棱  $c11, c12, c13, c14$  的一维特征向量 (即  $z$  值的平均值), 画出相应的特征曲线, 如图 3.3.5。

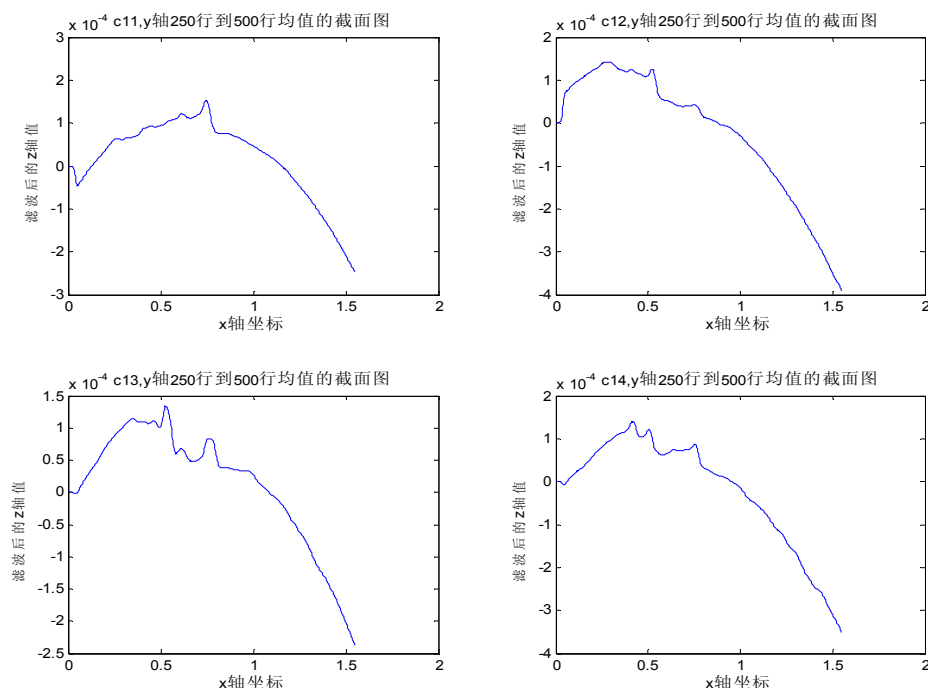


图 3.3.5

比较图 3.3.3 和图 3.3.5 可以得出同样的结论：次棱  $c1, c2, c3, c4$  与次棱  $c12, c13, c14, c11$  相对应。也就是说可以通过比较一维特征向量和特征曲线来判别次棱的对应情况，进而可以判断二个子弹是否是一只枪射出的。

通过对题目中给出了其他 5 只枪发射的 10 个弹头的各个次棱进行同样的分析，可以看出选取上面的一维特征向量（也可以认为特征曲线，因为特征曲线是以一维特征向量为纵坐标产生的）可以很好的作为判断两个次棱相似程度的标准。

可以这样认为，如果两个次棱所得出的特征曲线基本相似，则这两个次棱相匹配。怎样判断特征曲线的相似程度呢，根据统计学观点，可以通过比较这两个次棱的一维特征向量的**相关系数**来进行判别。若相关系数较高，则一维特征向量的数据有较好的线形相关性，从而以  $x$  值为横坐标，一维特征向量为纵坐标所得到的特征曲线相似程度较高，因而这两个次棱相似程度越高。具体的相关系数的**临界值**（即大于这个临界值，则认为进行比对的两个次棱是同一个枪发射的两个子弹的同一个位置的次棱）可以通过对题目已给的 6 只枪发射的 12 个弹头的各个次棱的观测数据进行分析得出。

上面的方法只是比较了两个次棱的相似程度，并没有给出进行比较的两个弹头的相似程度。由于本题提供对应 4 条次棱的数据，其顺序统一按照一个方向排序。所以次棱  $c1, c2, c3, c4$  和次棱  $c11, c12, c13, c14$ （它们是同一个枪射出的两个子弹的次棱）只有以下 4 种对应关系：

第一种对应关系： $c1-c11, c2-c12, c3-c13, c4-c14$ 。

第二种对应关系： $c1-c12, c2-c13, c3-c14, c4-c11$ 。

第三种对应关系:  $c1-c13, c2-c14, c3-c11, c4-c12$  。

第四种对应关系:  $c1-c14, c2-c11, c3-c12, c4-c13$  。

根据题目的背景知识: 对同一支枪发射的两个弹头, 4 个次棱中通常可以有 2 至 3 个吻合比较好。若上面的某一种对应关系是正确的, 则在这个对应关系中有 2 至 3 个对应次棱的相关系数较高, 那么这个对应关系中所有对应次棱的相关系数之和较高, 也可以说这个对应关系中所有对应次棱的相关系数的均值较高, 记相关系数的均值为**特征相关系数**, 可以利用这个来判断选取哪种对应关系。四种对应关系中的最大者记为**比对特征相关系数**。

注: 特征相关系数和比对特征相关系数的取值范围为 $[-1,1]$ , 值为 1 说明两个次棱的形状完全一样, 值为 0 说明两个次棱没有任何关系, 值为-1 说明两个次棱形状正好相反。

若进行比对的两个弹头不是同一只枪射出的, 则按照上面方法所得出的比对特征相关系数不会太高, 具体的临界值可以通过对题目已给的 6 只枪发射的 12 个弹头的各个次棱的观测数据进行分析得出。

### 3.3.3 问题三的具体对比方案和算法

#### 3.3.3.1 对比方案:

假设要比对两个弹头, 这两个弹头的次棱分别为:  $d1, d2, d3, d4$  和

$d11, d12, d13, d14$ 。

首先将第一个弹头的每个次棱和第二个弹头的每个次棱进行比对, 求出相应的一维特征向量, 进而求出该一维特征向量的相关系数。然后可以求出上面四种对应关系里每个对应关系的特征相关系数, 进而求出比对特征相关系数。最后将比对特征相关系数与临界值比较, 可以判断出这两个弹头是否是同一只枪发出的。

#### 3.3.3.2 对比算法

第一步:

输入要进行对比的两个弹头的各个次棱的数据信息, 这些数据信息经过了第一问和第二问所提方法的处理。

将次棱  $d1, d2, d3, d4$  的  $z$  轴的值作为列赋值到矩阵  $a$  中, 同样将次棱

$d11, d12, d13, d14$  的  $z$  轴的值作为列赋值到矩阵  $b$  中。

第二步：

对于次棱  $di$  和次棱  $d1j$ ，求出他们的一维特征向量。

- 1) 将  $di$  的  $z$  轴数据  $a(i)$  转化成以 2.75 微米为步长的  $x, y$  数据所对应的矩阵  $r$ 。 $r$  的每一行代表一个固定  $y$  值上次棱  $di$  上所对应  $z$  轴的数据。同样将  $d1j$  的  $z$  轴数据  $b(j)$  转化成以 2.75 微米为步长的  $x, y$  数据所对应的矩阵  $z1$ 。
- 2) 将矩阵  $r$  和  $z1$  中取  $y$  轴中间 1/3 的数据，计算每个  $x$  值上所有数据的均值，将得出的均值数据赋值到向量  $d$  和  $d1$  中。
- 3) 向量  $d$  和  $d1$  即为次棱  $di$  和  $d1j$  所对应的一维特征向量。

第三步：

求出一维特征向量  $d$  和  $d1$  所对应的相关系数  $f_{ij}$ 。

$$f_{ij} = \text{corrcoef}(d, d1) = \frac{d^T * d1}{\sqrt{d^T * d} * \sqrt{d1^T * d1}}$$

第四步：

重复上面算法求出每个  $di(i=1,2,3,4)$  和每个  $d1j(j=1,2,3,4)$  的相关系数  $f_{ij}$ ，并赋值到相关系数矩阵  $h$  中，其中  $h(i, j) = f_{ij}$ 。

第五步：

求出 4 种对应关系下的特征相关系数：

- 1) 第一种对应关系：

$$r(1) = (h(1,1) + h(2,2) + h(3,3) + h(4,4)) / 4$$

- 2) 第二种对应关系：

$$r(2) = (h(1,2) + h(2,3) + h(3,4) + h(4,1)) / 4$$

- 3) 第三种对应关系：

$$r(3) = (h(1,3) + h(2,4) + h(3,1) + h(4,2)) / 4$$

- 4) 第四种对应关系：

$$r(4) = (h(1,4) + h(2,1) + h(3,2) + h(4,3)) / 4$$

在求出比对特征相关系数  $r1 = \max(r)$

第六步：判断这两个弹头是否为同一手枪射出并求出是第几种对应关系

- 1) 若  $r1 > r_{\text{临界值}}$ , 转 3)
- 2) 若  $r1 < r_{\text{临界值}}$ , 这两个弹头不是同一手枪射出
- 3) 分别将  $r(1), r(2), r(3), r(4)$  与  $r1$  比较,

若  $r(i) = r1$ , 输出  $i$ , 这两个弹头是同一手枪射出并且是第  $i$  种对应关系

根据上面的算法, 编写了一个 test.m 的程序, 可以实现这个算法。

### 3.3.4 根据已有数据计算比对特征相关系数的临界值

首先, 根据对于前三问的分析编写了一个程序 disanwen.m, 这个程序的基本功能是通过输入要进行对比的两个弹头的各个次棱的数据信息可以自动的进行数据处理, 并且根据 3.3.3 中的对比方案可以自动的产生比对特征相关系数, 以及符合第几种对应关系。程序中有详细的注释。

将题目中每只枪射出的两个弹头的各个次棱的观测数据分别代入程序, 可以得到如下表格:

枪号	比对特征相关系数	第一个弹头次棱 c1 对应的第二个弹头的次棱编号
1203959	0.9434	c2
1504519	0.5767	c1
1811345	0.9918	c4
1812492	0.9481	c4
1923252	0.9521	c1
1928033	0.9531	c1

通过表格可以看出除了枪号为 1504519 的枪射出两个弹头的比对特征相关系数值比较低之外, 其他的比对特征相关系数都比较高。说明将比对特征相关系数作为比对弹头相似程度的特征效果是比较好的。

一般来说如果比对特征相关系数的值大于 0.95 (即比对特征相关系数临界值为 0.9), 我们可以认为这两个弹头来自同一个枪。



## 3.4 问题 4

### 3.4.1 问题分析

用解决第三问的方法解决问题四，要求给出弹头痕迹两两之间的相似程度。我们在第三问中给出了一维特征向量，并自编了character函数，对每一个次棱进行一维特征向量的计算。

给出了对一维特征向量进行比对的方法，并返回出了每个次棱两两之间的相似度，以及该子弹与另外一个子弹相似度的方式，如：c1-c11, 或c1-c12, 或c1-c13, 或c1-14。因此可以用第三问的方式进行两两比对。对于这一步我们自编了charactermatch函数，可用于自动比对。

### 3.4.2 问题实现

具体算法：

第一步：将22个子弹的22\*4个次棱读入matlab中。

第二步：对两两子弹进行对比采用for循环for i=1:22, j=1:22

当i=1时，y依次与其相比，得到相应的相似度矩阵，相似度等。

第三步：即可得到第二步返回的两两相似度矩阵H，相似度R，以及相似度对应的方式K。

算法说明：采用我们编写的按以上步骤的matlab程序，计算机运行很慢，且出现了一个很大的问题，即matlab的内存不能保存所有的数据，在还没有读完所有的数据的时候就已经出现了Out of memory的提示。因此对我们利用此种方法解决问题带来了很大的不便。参见程序附件3\第四问程序\disiwen

我们的其它的解决办法，利用计算机将每个子弹的特征函数，得到22个特征函数，利用这22个函数并利用charactermatch进行两两对比，即可得到两两相似度矩阵H，相似度R，以及相似度对应的方式K。具体可参见附件3中第三问程序\siwen。

### 3.4.3 问题结果

采用这种方法我们得到的结果如下：

两两相似度矩阵为：由于数据太多，我们附页表示：见附页4(EXCEL)

由于我们取的点较多，所以出现的精确度较高，但此时与实际情况有些不符，有待进一步的检验，由于时间原因，我们只能做到这一步。

4.2 根据（1）的结果，用列表的方式给出每个弹头按相似度由高到低给出与之相似的前5位的弹头文件号

此问比较简单，只要我们知道了子弹两两之间的相似度，我们即可按行求出每个子弹按相似度又高到底给出的与之相似的弹头文件号。进过比较，可以由上问得到此问的结果如下：（见附件5(EXCEL)）

附件4，附件5电子版在程序压缩包内，纸板附在后面。

	1	2	3	4	5
01	20	16	15	11	03
02	16	15	20	01	09
03	22	20	15	14	05
04	13	14	04	10	22
05	22	03	07	06	20
06	05	07	22	08	20
07	08	13	04	22	14
08	07	21	03	04	20
09	19	15	01	20	22
10	14	13	04	07	17
11	20	01	15	03	12
12	15	17	03	22	14
13	14	02	07	10	22
14	13	04	10	22	17
15	20	16	22	12	03
16	15	20	01	22	17
17	14	13	12	15	16
18	16	01	15	02	20
19	9	11	12	15	20
20	15	03	01	22	16
21	08	07	20	09	03
22	03	14	05	20	15

## 4. 本文总结与展望

### 4. 1 程序运行时间的分析：

1. 本次的运算数据量很大，运行时间很长，容易出现 outofmemory 的情况
2. 结合本次的程序，对每一步的运行时间进行总结：

每 load 一次文件大约需要 30s，对数据每进行一次滤波大约需要 30s，提取特征向量大约需要 30s，调用算法对特征向量进行比对大约需要 1min，即进行一个完整的过程大约需要 2-3min，而且如果画图的话，程序的运行时间会成倍的增加，所需时间会增加很多。对于第四问 22\*22 次的比对，大约需要 1452min=24h，若没有大型计算机的话会很难完成。这是在不考虑 matlab 内存 outofmemory 的情况。当然在程序的运行过程中出现了 outofmemory 的情况，再此情况下，我们需要采用其他的方法来完成，见理论叙述。

## 4. 2 本文展望

对于枪炮子弹痕迹对比系统可以采取更加细致、复杂的特征函数，并进行更为精确的比对，会得到更好的区分度，我们会在赛后继续这方面的研究。

### 参考文献

- [1] Paul J. Besl, Member, IEEE, and Neil D. McKay, A Method for Registration of 3-D Shapes, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL.. 14: pp239~256, FEBRUARY 1992。
- [2] Jan De Kinder, Monica BOnfanti Automated comparisons of bullet striations based on 3D topography Forensic Science International 101(1999)85-93 ,1999

## 附录 1

### Mathematica 7.03 求解二元一次方程组的结果:

`Solve[{(x1 - x)^2 + (z1 - z)^2 == 3.95^2, (x2 - x)^2 + (z2 - z)^2 == 3.95^2}, {x, z}]`

```
{x -> (1/(2. x1 - 2. x2)) (x1^2 - 1. x2^2 + z1^2 - 1. z2^2 - (400. x1^2 z1^2)/(400.
x1^2 - 800. x1 x2 + 400. x2^2 + 400. z1^2 - 800. z1 z2 + 400. z2^2) + (800. x1 x2
z1^2)/(400. x1^2 - 800. x1 x2 + 400. x2^2 + 400. z1^2 - 800. z1 z2 + 400. z2^2) - (400.
x2^2 z1^2)/(400. x1^2 - 800. x1 x2 + 400. x2^2 + 400. z1^2 - 800. z1 z2 + 400. z2^2) - (400.
z1^4)/(400. x1^2 - 800. x1 x2 + 400. x2^2 + 400. z1^2 - 800. z1 z2 + 400. z2^2) + (800.
z1^3 z2)/(400. x1^2 - 800. x1 x2 + 400. x2^2 + 400. z1^2 - 800. z1 z2 + 400. z2^2) + (400.
x1^2 z2^2)/(400. x1^2 - 800. x1 x2 + 400. x2^2 + 400. z1^2 - 800. z1 z2 + 400. z2^2) - (800.
x1 x2 z2^2)/(400. x1^2 - 800. x1 x2 + 400. x2^2 + 400. z1^2 - 800. z1 z2 + 400.
z2^2) + (400. x2^2 z2^2)/(400. x1^2 - 800. x1 x2 + 400. x2^2 + 400. z1^2 - 800. z1 z2 + 400.
z2^2) - (800. z1 z2^3)/(400. x1^2 - 800. x1 x2 + 400. x2^2 + 400. z1^2 - 800. z1 z2 + 400.
z2^2) + (400. z2^4)/(400. x1^2 - 800. x1 x2 + 400. x2^2 + 400. z1^2 - 800. z1 z2 + 400.
z2^2) + (1. z1 \[Sqrt](9.9856*10^6 x1^4 - 160000. x1^6 - 3.99424*10^7 x1^3
x2 + 960000. x1^5 x2 + 5.99136*10^7 x1^2 x2^2 - 2.4*10^6 x1^4 x2^2 - 3.99424*10^7 x1
x2^3 + 3.2*10^6 x1^3 x2^3 + 9.9856*10^6 x2^4 - 2.4*10^6 x1^2 x2^4 + 960000. x1 x2^5 - 160000.
x2^6 + 9.9856*10^6 x1^2 z1^2 - 320000. x1^4 z1^2 - 1.99712*10^7 x1 x2 z1^2 + 1.28*10^6
x1^3 x2 z1^2 + 9.9856*10^6 x2^2 z1^2 - 1.92*10^6 x1^2 x2^2 z1^2 + 1.28*10^6 x1 x2^3
z1^2 - 320000. x2^4 z1^2 - 160000. x1^2 z1^4 + 320000. x1 x2 z1^4 - 160000. x2^2 z1^4 + 0.
z1^6 - 1.99712*10^7 x1^2 z1 z2 + 640000. x1^4 z1 z2 + 3.99424*10^7 x1 x2 z1
z2 - 2.56*10^6 x1^3 x2 z1 z2 - 1.99712*10^7 x2^2 z1 z2 + 3.84*10^6 x1^2 x2^2 z1
z2 - 2.56*10^6 x1 x2^3 z1 z2 + 640000. x2^4 z1 z2 + 640000. x1^2 z1^3 z2 - 1.28*10^6
x1 x2 z1^3 z2 + 640000. x2^2 z1^3 z2 + 0. z1^5 z2 + 9.9856*10^6 x1^2 z2^2 - 320000. x1^4
z2^2 - 1.99712*10^7 x1 x2 z2^2 + 1.28*10^6 x1^3 x2 z2^2 + 9.9856*10^6 x2^2 z2^2 - 1.92*10^6
x1^2 x2^2 z2^2 + 1.28*10^6 x1 x2^3 z2^2 - 320000. x2^4 z2^2 - 960000. x1^2 z1^2
z2^2 + 1.92*10^6 x1 x2 z1^2 z2^2 - 960000. x2^2 z1^2 z2^2 + 0. z1^4 z2^2 + 640000. x1^2 z1
z2^3 - 1.28*10^6 x1 x2 z1 z2^3 + 640000. x2^2 z1 z2^3 + 0. z1^3 z2^3 - 160000. x1^2
z2^4 + 320000. x1 x2 z2^4 - 160000. x2^2 z2^4 + 0. z1^2 z2^4 + 0. z1 z2^5 + 0. z2^6))/(400.
x1^2 - 800. x1 x2 + 400. x2^2 + 400. z1^2 - 800. z1 z2 + 400. z2^2) - (1. z2
\[Sqrt](9.9856*10^6 x1^4 - 160000. x1^6 - 3.99424*10^7 x1^3 x2 + 960000. x1^5
x2 + 5.99136*10^7 x1^2 x2^2 - 2.4*10^6 x1^4 x2^2 - 3.99424*10^7 x1 x2^3 + 3.2*10^6 x1^3
x2^3 + 9.9856*10^6 x2^4 - 2.4*10^6 x1^2 x2^4 + 960000. x1 x2^5 - 160000. x2^6 + 9.9856*10^6
x1^2 z1^2 - 320000. x1^4 z1^2 - 1.99712*10^7 x1 x2 z1^2 + 1.28*10^6 x1^3 x2
z1^2 + 9.9856*10^6 x2^2 z1^2 - 1.92*10^6 x1^2 x2^2 z1^2 + 1.28*10^6 x1 x2^3 z1^2 - 320000.
x2^4 z1^2 - 160000. x1^2 z1^4 + 320000. x1 x2 z1^4 - 160000. x2^2 z1^4 + 0.
z1^6 - 1.99712*10^7 x1^2 z1 z2 + 640000. x1^4 z1 z2 + 3.99424*10^7 x1 x2 z1
z2 - 2.56*10^6 x1^3 x2 z1 z2 - 1.99712*10^7 x2^2 z1 z2 + 3.84*10^6 x1^2 x2^2 z1
z2 - 2.56*10^6 x1 x2^3 z1 z2 + 640000. x2^4 z1 z2 + 640000. x1^2 z1^3 z2 - 1.28*10^6
x1 x2 z1^3 z2 + 640000. x2^2 z1^3 z2 + 0. z1^5 z2 + 9.9856*10^6 x1^2 z2^2 - 320000. x1^4
z2^2 - 1.99712*10^7 x1 x2 z2^2 + 1.28*10^6 x1^3 x2 z2^2 + 9.9856*10^6 x2^2 z2^2 - 1.92*10^6
```

$$\frac{x_1^2 x_2^2 z^2 + 1.28 \times 10^6 x_1 x_2^3 z^2 - 320000 x_2^4 z^2 - 960000 x_1^2 z^2 + 1.92 \times 10^6 x_1 x_2 z^2 - 960000 x_2^2 z^2 + 0 z^4 + 640000 x_1^2 z^2 - 1.28 \times 10^6 x_1 x_2 z^3 + 640000 x_2^2 z^3 + 0 z^1^3 - 160000 x_1^2 z^4 + 320000 x_1 x_2 z^4 - 160000 x_2^2 z^4 + 0 z^1^2 z^4 + 0 z^1 z^5 + 0 z^2^6)}{(400 x_1^2 - 800 x_1 x_2 + 400 x_2^2 + 400 z^2 - 800 z^1 + 400 z^2^2)}, z \rightarrow (0.5 (400 x_1^2 z^1 - 800 x_1 x_2 z^1 + 400 x_2^2 z^1 + 400 z^1^3 + 400 x_1^2 z^2 - 800 x_1 x_2 z^2 + 400 x_2^2 z^2 - 400 z^1^2 z^2 - 400 z^1 z^2^2 + 400 z^2^3 - 1. \sqrt{(-400 x_1^2 z^1 + 800 x_1 x_2 z^1 - 400 x_2^2 z^1 - 400 z^1^3 - 400 x_1^2 z^2 + 800 x_1 x_2 z^2 - 400 x_2^2 z^2 + 400 z^1^2 z^2 - 400 z^1 z^2^2 + 400 z^2^3)^2 - 4} (400 x_1^2 - 800 x_1 x_2 + 400 x_2^2 + 400 z^2 - 800 z^1 + 400 z^2^2) (-6241 x_1^2 + 100 x_1^4 + 12482 x_1 x_2 - 400 x_1^3 x_2 - 6241 x_2^2 + 600 x_1^2 x_2^2 - 400 x_1 x_2^3 + 100 x_2^4 + 200 x_1^2 z^1^2 - 400 x_1 x_2 z^1^2 + 200 x_2^2 z^1^2 + 100 z^1^4 + 200 x_1^2 z^2^2 - 400 x_1 x_2 z^2^2 + 200 x_2^2 z^2^2 - 200 z^1^2 z^2^2 + 100 z^2^4)))) / (400 x_1^2 - 800 x_1 x_2 + 400 x_2^2 + 400 z^2 - 800 z^1 + 400 z^2^2)}$$

## 附录 2

※※※2009 年研究生数学建模竞赛 B 题的 Matlab 程序※※※

%第 1 问的 Matlab 程序

%将本程序和所有用到的数据放在 C:\B 目录下

%此程序为人机交互式，请按此程序的提示操作

function main()

while true

disp(' ※※※第 1 问的 Matlab 程序※※※');

disp(' 此程序为人机交互式，请按此程序的提示操作 ');

disp(' 请输入第一组数据的文件名 ');

FirstName=input('请输入: ','s');

DirPath='C:\B\';

DataPath=strcat(DirPath,FirstName);

FirstData=load(DataPath);

disp(' 请输入第二组数据的文件名 ');

SecondName=input('请输入: ','s');

DirPath='C:\B\';

DataPath=strcat(DirPath,SecondName);

SecondData=load(DataPath);

disp(' 第一步: ');

disp(' 请输入 lamda 的值 本题取 1 较为合适 ');%凭经验和图像观察选取

lamda=input('请输入: ');

%for i=lamda\*0.00275:0.00275:564\*0.00275

for i=1:564

t(i)=0;

r(i)=0;

end

for i=lamda:564

```

x1(i)=(i-1)*0.00275;
x2(i)=(i-1)*0.00275;
for j=1:756
    t(i)=FirstData(i+500*(j-1),3)+t(i);
    r(i)=SecondData(i+500*(j-1),3)+r(i)
end
z1(i)=t(i)/756;
z2(i)=r(i)/756;
end
%x1(lamda)
%z1(lamda)
%x1(564)
%z1(564)
%x1
%x2
disp('                请输入 miu 的值    本题取 10 较为合适 ');%凭经验和图像观
察选取
miu=input('请输入: ');
t1=0;
r1=0;
t2=0;
r2=0;
for i=lamda:564
    j=564-(i-lamda);

t1=t1+(1/(2*x1(i)-2*x1(j)))*(x1(i)^2-1*x1(j)^2+z1(i)^2-1*z1(j)^2-(400*x1(i)^2*z1(i)
^2)/(400*x1(i)^2-800*x1(i)*x1(j)+400*x1(j)^2+400*z1(i)^2-800*z1(i)*z1(j)+
400*z1(j)^2)+(800*x1(i)*x1(j)*z1(i)^2)/(400*x1(i)^2-800*x1(i)*x1(j)+400*x1(j)^2+
400*z1(i)^2-800*z1(i)*z1(j)+
400*z1(j)^2)-(400*x1(j)^2*z1(i)^2)/(400*x1(i)^2-800*x1(i)*x1(j)+400*x1(j)^2+400
*z1(i)^2-800*z1(i)*z1(j)+
400*z1(j)^2)-(400*z1(i)^4)/(400*x1(i)^2-800*x1(i)*x1(j)+400*x1(j)^2+400*z1(i)^2-
800*z1(i)*z1(j)+
400*z1(j)^2)+(800*z1(i)^3*z1(j))/(400*x1(i)^2-800*x1(i)*x1(j)+400*x1(j)^2+400*z
1(i)^2-800*z1(i)*z1(j)+
400*z1(j)^2)+(400*x1(i)^2*z1(j)^2)/(400*x1(i)^2-800*x1(i)*x1(j)+400*x1(j)^2+400
*z1(i)^2-800*z1(i)*z1(j)+
400*z1(j)^2)-(800*x1(i)*x1(j)*z1(j)^2)/(400*x1(i)^2-800*x1(i)*x1(j)+400*x1(j)^2+
400*z1(i)^2-800*z1(i)*z1(j)+
400*z1(j)^2)+(400*x1(j)^2*z1(j)^2)/(400*x1(i)^2-800*x1(i)*x1(j)+400*x1(j)^2+400
*z1(i)^2-800*z1(i)*z1(j)+
400*z1(j)^2)-(800*z1(i)*z1(j)^3)/(400*x1(i)^2-800*x1(i)*x1(j)+400*x1(j)^2+400*z
1(i)^2-800*z1(i)*z1(j)+
400*z1(j)^2)+(400*z1(j)^4)/(400*x1(i)^2-800*x1(i)*x1(j)+400*x1(j)^2+400*z1(i)^2

```



$$\begin{aligned}
& -800*z1(i)*z1(j)+ \\
& 400*z1(j)^2+(1*z1(i)*\sqrt{9.9856*10^6*x1(i)^4-160000*x1(i)^6-} \\
& 3.99424*10^7*x1(i)^3*x1(j)+960000*x1(i)^5*x1(j)+ \\
& 5.99136*10^7*x1(i)^2*x1(j)^2-2.4*10^6*x1(i)^4*x1(j)^2- \\
& 3.99424*10^7*x1(i)*x1(j)^3+3.2*10^6*x1(i)^3*x1(j)^3+ \\
& 9.9856*10^6*x1(j)^4-2.4*10^6*x1(i)^2*x1(j)^4+960000*x1(i)*x1(j)^5- \\
& 160000*x1(j)^6+9.9856*10^6*x1(i)^2*z1(i)^2-320000*x1(i)^4*z1(i)^2- \\
& 1.99712*10^7*x1(i)*x1(j)*z1(i)^2+1.28*10^6*x1(i)^3*x1(j)*z1(i)^2+ \\
& 9.9856*10^6*x1(j)^2*z1(i)^2-1.92*10^6*x1(i)^2*x1(j)^2*z1(i)^2+ \\
& 1.28*10^6*x1(i)*x1(j)^3*z1(i)^2-320000*x1(j)^4*z1(i)^2- \\
& 160000*x1(i)^2*z1(i)^4+320000*x1(i)*x1(j)*z1(i)^4-160000*x1(j)^2*z1(i)^4+ \\
& 0*z1(i)^6-1.99712*10^7*x1(i)^2*z1(i)*z1(j)+640000*x1(i)^4*z1(i)*z1(j)+ \\
& 3.99424*10^7*x1(i)*x1(j)*z1(i)*z1(j)-2.56*10^6*x1(i)^3*x1(j)*z1(i)*z1(j)- \\
& 1.99712*10^7*x1(j)^2*z1(i)*z1(j)+3.84*10^6*x1(i)^2*x1(j)^2*z1(i)*z1(j)- \\
& 2.56*10^6*x1(i)*x1(j)^3*z1(i)*z1(j)+640000*x1(j)^4*z1(i)*z1(j)+ \\
& 640000*x1(i)^2*z1(i)^3*z1(j)-1.28*10^6*x1(i)*x1(j)*z1(i)^3*z1(j)+ \\
& 640000*x1(j)^2*z1(i)^3*z1(j)+0*z1(i)^5*z1(j)+9.9856*10^6*x1(i)^2*z1(j)^2- \\
& 320000*x1(i)^4*z1(j)^2-1.99712*10^7*x1(i)*x1(j)*z1(j)^2+ \\
& 1.28*10^6*x1(i)^3*x1(j)*z1(j)^2+9.9856*10^6*x1(j)^2*z1(j)^2- \\
& 1.92*10^6*x1(i)^2*x1(j)^2*z1(j)^2+1.28*10^6*x1(i)*x1(j)^3*z1(j)^2- \\
& 320000*x1(j)^4*z1(j)^2-960000*x1(i)^2*z1(i)^2*z1(j)^2+ \\
& 1.92*10^6*x1(i)*x1(j)*z1(i)^2*z1(j)^2-960000*x1(j)^2*z1(i)^2*z1(j)^2+ \\
& 0*z1(i)^4*z1(j)^2+640000*x1(i)^2*z1(i)*z1(j)^3- \\
& 1.28*10^6*x1(i)*x1(j)*z1(i)*z1(j)^3+640000*x1(j)^2*z1(i)*z1(j)^3+ \\
& 0*z1(i)^3*z1(j)^3-160000*x1(i)^2*z1(j)^4+320000*x1(i)*x1(j)*z1(j)^4- \\
& 160000*x1(j)^2*z1(j)^4+0*z1(i)^2*z1(j)^4+0*z1(i)*z1(j)^5+ \\
& 0*z1(j)^6)/(400*x1(i)^2-800*x1(i)*x1(j)+400*x1(j)^2+400*z1(i)^2- \\
& 800*z1(i)*z1(j)+ \\
& 400*z1(j)^2)-(1*z1(j)*\sqrt{9.9856*10^6*x1(i)^4-160000*x1(i)^6-} \\
& 3.99424*10^7*x1(i)^3*x1(j)+960000*x1(i)^5*x1(j)+ \\
& 5.99136*10^7*x1(i)^2*x1(j)^2-2.4*10^6*x1(i)^4*x1(j)^2- \\
& 3.99424*10^7*x1(i)*x1(j)^3+3.2*10^6*x1(i)^3*x1(j)^3+ \\
& 9.9856*10^6*x1(j)^4-2.4*10^6*x1(i)^2*x1(j)^4+960000*x1(i)*x1(j)^5- \\
& 160000*x1(j)^6+9.9856*10^6*x1(i)^2*z1(i)^2-320000*x1(i)^4*z1(i)^2- \\
& 1.99712*10^7*x1(i)*x1(j)*z1(i)^2+1.28*10^6*x1(i)^3*x1(j)*z1(i)^2+ \\
& 9.9856*10^6*x1(j)^2*z1(i)^2-1.92*10^6*x1(i)^2*x1(j)^2*z1(i)^2+ \\
& 1.28*10^6*x1(i)*x1(j)^3*z1(i)^2-320000*x1(j)^4*z1(i)^2- \\
& 160000*x1(i)^2*z1(i)^4+320000*x1(i)*x1(j)*z1(i)^4-160000*x1(j)^2*z1(i)^4+ \\
& 0*z1(i)^6-1.99712*10^7*x1(i)^2*z1(i)*z1(j)+640000*x1(i)^4*z1(i)*z1(j)+ \\
& 3.99424*10^7*x1(i)*x1(j)*z1(i)*z1(j)-2.56*10^6*x1(i)^3*x1(j)*z1(i)*z1(j)- \\
& 1.99712*10^7*x1(j)^2*z1(i)*z1(j)+3.84*10^6*x1(i)^2*x1(j)^2*z1(i)*z1(j)- \\
& 2.56*10^6*x1(i)*x1(j)^3*z1(i)*z1(j)+640000*x1(j)^4*z1(i)*z1(j)+ \\
& 640000*x1(i)^2*z1(i)^3*z1(j)-1.28*10^6*x1(i)*x1(j)*z1(i)^3*z1(j)+ \\
& 640000*x1(j)^2*z1(i)^3*z1(j)+0*z1(i)^5*z1(j)+9.9856*10^6*x1(i)^2*z1(j)^2-
\end{aligned}$$

$$\begin{aligned}
& 320000*x1(i)^4*z1(j)^2-1.99712*10^7*x1(i)*x1(j)*z1(j)^2+ \\
& 1.28*10^6*x1(i)^3*x1(j)*z1(j)^2+9.9856*10^6*x1(j)^2*z1(j)^2- \\
& 1.92*10^6*x1(i)^2*x1(j)^2*z1(j)^2+1.28*10^6*x1(i)*x1(j)^3*z1(j)^2- \\
& 320000*x1(j)^4*z1(j)^2-960000*x1(i)^2*z1(i)^2*z1(j)^2+ \\
& 1.92*10^6*x1(i)*x1(j)*z1(i)^2*z1(j)^2-960000*x1(j)^2*z1(i)^2*z1(j)^2+ \\
& 0*z1(i)^4*z1(j)^2+640000*x1(i)^2*z1(i)*z1(j)^3- \\
& 1.28*10^6*x1(i)*x1(j)*z1(i)*z1(j)^3+640000*x1(j)^2*z1(i)*z1(j)^3+ \\
& 0*z1(i)^3*z1(j)^3-160000*x1(i)^2*z1(j)^4+320000*x1(i)*x1(j)*z1(j)^4- \\
& 160000*x1(j)^2*z1(j)^4+0*z1(i)^2*z1(j)^4+0*z1(i)*z1(j)^5+ \\
& 0*z1(j)^6)/(400*x1(i)^2-800*x1(i)*x1(j)+400*x1(j)^2+400*z1(i)^2- \\
& 800*z1(i)*z1(j)+400*z1(j)^2));
\end{aligned}$$

$$\begin{aligned}
t2=t2+(1/(2*x2(i)-2*x2(j)))*(x2(i)^2-1*x2(j)^2+z2(i)^2-1*z2(j)^2-(400*x2(i)^2*z2(i) \\
^2)/(400*x2(i)^2-800*x2(i)*x2(j)+400*x2(j)^2+400*z2(i)^2-800*z2(i)*z2(j)+ \\
400*z2(j)^2)+(800*x2(i)*x2(j)*z2(i)^2)/(400*x2(i)^2-800*x2(i)*x2(j)+400*x2(j)^2+ \\
400*z2(i)^2-800*z2(i)*z2(j)+ \\
400*z2(j)^2)-(400*x2(j)^2*z2(i)^2)/(400*x2(i)^2-800*x2(i)*x2(j)+400*x2(j)^2+400 \\
*z2(i)^2-800*z2(i)*z2(j)+ \\
400*z2(j)^2)-(400*z2(i)^4)/(400*x2(i)^2-800*x2(i)*x2(j)+400*x2(j)^2+400*z2(i)^2- \\
800*z2(i)*z2(j)+ \\
400*z2(j)^2)+(800*z2(i)^3*z2(j))/(400*x2(i)^2-800*x2(i)*x2(j)+400*x2(j)^2+400*z2(i)^2- \\
800*z2(i)*z2(j)+ \\
400*z2(j)^2)+(400*x2(i)^2*z2(j)^2)/(400*x2(i)^2-800*x2(i)*x2(j)+400*x2(j)^2+400 \\
*z2(i)^2-800*z2(i)*z2(j)+ \\
400*z2(j)^2)-(800*x2(i)*x2(j)*z2(j)^2)/(400*x2(i)^2-800*x2(i)*x2(j)+400*x2(j)^2+ \\
400*z2(i)^2-800*z2(i)*z2(j)+ \\
400*z2(j)^2)+(400*x2(j)^2*z2(j)^2)/(400*x2(i)^2-800*x2(i)*x2(j)+400*x2(j)^2+400 \\
*z2(i)^2-800*z2(i)*z2(j)+ \\
400*z2(j)^2)-(800*z2(i)*z2(j)^3)/(400*x2(i)^2-800*x2(i)*x2(j)+400*x2(j)^2+400*z2(i)^2- \\
800*z2(i)*z2(j)+ \\
400*z2(j)^2)+(400*z2(j)^4)/(400*x2(i)^2-800*x2(i)*x2(j)+400*x2(j)^2+400*z2(i)^2- \\
800*z2(i)*z2(j)+ \\
400*z2(j)^2)+(1*z2(i)*sqrt(9.9856*10^6*x2(i)^4-160000*x2(i)^6- \\
3.99424*10^7*x2(i)^3*x2(j)+960000*x2(i)^5*x2(j)+ \\
5.99136*10^7*x2(i)^2*x2(j)^2-2.4*10^6*x2(i)^4*x2(j)^2- \\
3.99424*10^7*x2(i)*x2(j)^3+3.2*10^6*x2(i)^3*x2(j)^3+ \\
9.9856*10^6*x2(j)^4-2.4*10^6*x2(i)^2*x2(j)^4+960000*x2(i)*x2(j)^5- \\
160000*x2(j)^6+9.9856*10^6*x2(i)^2*z2(i)^2-320000*x2(i)^4*z2(i)^2- \\
1.99712*10^7*x2(i)*x2(j)*z2(i)^2+1.28*10^6*x2(i)^3*x2(j)*z2(i)^2+ \\
9.9856*10^6*x2(j)^2*z2(i)^2-1.92*10^6*x2(i)^2*x2(j)^2*z2(i)^2+ \\
1.28*10^6*x2(i)*x2(j)^3*z2(i)^2-320000*x2(j)^4*z2(i)^2- \\
160000*x2(i)^2*z2(i)^4+320000*x2(i)*x2(j)*z2(i)^4-160000*x2(j)^2*z2(i)^4+ \\
0*z2(i)^6-1.99712*10^7*x2(i)^2*z2(i)*z2(j)+640000*x2(i)^4*z2(i)*z2(j)+ \\
3.99424*10^7*x2(i)*x2(j)*z2(i)*z2(j)-2.56*10^6*x2(i)^3*x2(j)*z2(i)*z2(j)-
\end{aligned}$$

$$\begin{aligned}
& 1.99712*10^7*x2(j)^2*z2(i)*z2(j)+3.84*10^6*x2(i)^2*x2(j)^2*z2(i)*z2(j)- \\
& 2.56*10^6*x2(i)*x2(j)^3*z2(i)*z2(j)+640000*x2(j)^4*z2(i)*z2(j)+ \\
& 640000*x2(i)^2*z2(i)^3*z2(j)-1.28*10^6*x2(i)*x2(j)*z2(i)^3*z2(j)+ \\
& 640000*x2(j)^2*z2(i)^3*z2(j)+0*z2(i)^5*z2(j)+9.9856*10^6*x2(i)^2*z2(j)^2- \\
& 320000*x2(i)^4*z2(j)^2-1.99712*10^7*x2(i)*x2(j)*z2(j)^2+ \\
& 1.28*10^6*x2(i)^3*x2(j)*z2(j)^2+9.9856*10^6*x2(j)^2*z2(j)^2- \\
& 1.92*10^6*x2(i)^2*x2(j)^2*z2(j)^2+1.28*10^6*x2(i)*x2(j)^3*z2(j)^2- \\
& 320000*x2(j)^4*z2(j)^2-960000*x2(i)^2*z2(i)^2*z2(j)^2+ \\
& 1.92*10^6*x2(i)*x2(j)*z2(i)^2*z2(j)^2-960000*x2(j)^2*z2(i)^2*z2(j)^2+ \\
& 0*z2(i)^4*z2(j)^2+640000*x2(i)^2*z2(i)*z2(j)^3- \\
& 1.28*10^6*x2(i)*x2(j)*z2(i)*z2(j)^3+640000*x2(j)^2*z2(i)*z2(j)^3+ \\
& 0*z2(i)^3*z2(j)^3-160000*x2(i)^2*z2(j)^4+320000*x2(i)*x2(j)*z2(j)^4- \\
& 160000*x2(j)^2*z2(j)^4+0*z2(i)^2*z2(j)^4+0*z2(i)*z2(j)^5+ \\
& 0*z2(j)^6)/(400*x2(i)^2-800*x2(i)*x2(j)+400*x2(j)^2+400*z2(i)^2- \\
& 800*z2(i)*z2(j)+ \\
& 400*z2(j)^2)-(1*z2(j)*sqrt(9.9856*10^6*x2(i)^4-160000*x2(i)^6- \\
& 3.99424*10^7*x2(i)^3*x2(j)+960000*x2(i)^5*x2(j)+ \\
& 5.99136*10^7*x2(i)^2*x2(j)^2-2.4*10^6*x2(i)^4*x2(j)^2- \\
& 3.99424*10^7*x2(i)*x2(j)^3+3.2*10^6*x2(i)^3*x2(j)^3+ \\
& 9.9856*10^6*x2(j)^4-2.4*10^6*x2(i)^2*x2(j)^4+960000*x2(i)*x2(j)^5- \\
& 160000*x2(j)^6+9.9856*10^6*x2(i)^2*z2(i)^2-320000*x2(i)^4*z2(i)^2- \\
& 1.99712*10^7*x2(i)*x2(j)*z2(i)^2+1.28*10^6*x2(i)^3*x2(j)*z2(i)^2+ \\
& 9.9856*10^6*x2(j)^2*z2(i)^2-1.92*10^6*x2(i)^2*x2(j)^2*z2(i)^2+ \\
& 1.28*10^6*x2(i)*x2(j)^3*z2(i)^2-320000*x2(j)^4*z2(i)^2- \\
& 160000*x2(i)^2*z2(i)^4+320000*x2(i)*x2(j)*z2(i)^4-160000*x2(j)^2*z2(i)^4+ \\
& 0*z2(i)^6-1.99712*10^7*x2(i)^2*z2(i)*z2(j)+640000*x2(i)^4*z2(i)*z2(j)+ \\
& 3.99424*10^7*x2(i)*x2(j)*z2(i)*z2(j)-2.56*10^6*x2(i)^3*x2(j)*z2(i)*z2(j)- \\
& 1.99712*10^7*x2(j)^2*z2(i)*z2(j)+3.84*10^6*x2(i)^2*x2(j)^2*z2(i)*z2(j)- \\
& 2.56*10^6*x2(i)*x2(j)^3*z2(i)*z2(j)+640000*x2(j)^4*z2(i)*z2(j)+ \\
& 640000*x2(i)^2*z2(i)^3*z2(j)-1.28*10^6*x2(i)*x2(j)*z2(i)^3*z2(j)+ \\
& 640000*x2(j)^2*z2(i)^3*z2(j)+0*z2(i)^5*z2(j)+9.9856*10^6*x2(i)^2*z2(j)^2- \\
& 320000*x2(i)^4*z2(j)^2-1.99712*10^7*x2(i)*x2(j)*z2(j)^2+ \\
& 1.28*10^6*x2(i)^3*x2(j)*z2(j)^2+9.9856*10^6*x2(j)^2*z2(j)^2- \\
& 1.92*10^6*x2(i)^2*x2(j)^2*z2(j)^2+1.28*10^6*x2(i)*x2(j)^3*z2(j)^2- \\
& 320000*x2(j)^4*z2(j)^2-960000*x2(i)^2*z2(i)^2*z2(j)^2+ \\
& 1.92*10^6*x2(i)*x2(j)*z2(i)^2*z2(j)^2-960000*x2(j)^2*z2(i)^2*z2(j)^2+ \\
& 0*z2(i)^4*z2(j)^2+640000*x2(i)^2*z2(i)*z2(j)^3- \\
& 1.28*10^6*x2(i)*x2(j)*z2(i)*z2(j)^3+640000*x2(j)^2*z2(i)*z2(j)^3+ \\
& 0*z2(i)^3*z2(j)^3-160000*x2(i)^2*z2(j)^4+320000*x2(i)*x2(j)*z2(j)^4- \\
& 160000*x2(j)^2*z2(j)^4+0*z2(i)^2*z2(j)^4+0*z2(i)*z2(j)^5+ \\
& 0*z2(j)^6)/(400*x2(i)^2-800*x2(i)*x2(j)+400*x2(j)^2+400*z2(i)^2- \\
& 800*z2(i)*z2(j)+400*z2(j)^2));
\end{aligned}$$

$$x10=t1/(i-lamda+1);$$

$$x20=t2/(i-lamda+1);$$

```

r1=r1+(0.5*(400*x1(i)^2*z1(i)-800*x1(i)*x1(j)*z1(i)+400*x1(j)^2*z1(i)+400*z1(i)^3+400*x1(i)^2*z1(j)-800*x1(i)*x1(j)*z1(j)+400*x1(j)^2*z1(j)-400*z1(i)^2*z1(j)-400*z1(i)*z1(j)^2+400*z1(j)^3-1*sqrt((-400*x1(i)^2*z1(i)+800*x1(i)*x1(j)*z1(i)-400*x1(j)^2*z1(i)-400*z1(i)^3-400*x1(i)^2*z1(j)+800*x1(i)*x1(j)*z1(j)-400*x1(j)^2*z1(j)+400*z1(i)^2*z1(j)+400*z1(i)*z1(j)^2-400*z1(j)^3)^2-4*(400*x1(i)^2-800*x1(i)*x1(j)+400*x1(j)^2+400*z1(i)^2-800*z1(i)*z1(j)+400*z1(j)^2)*(-6241*x1(i)^2+100*x1(i)^4+12482*x1(i)*x1(j)-400*x1(i)^3*x1(j)-6241*x1(j)^2+600*x1(i)^2*x1(j)^2-400*x1(i)*x1(j)^3+100*x1(j)^4+200*x1(i)^2*z1(i)^2-400*x1(i)*x1(j)*z1(i)^2+200*x1(j)^2*z1(i)^2+100*z1(i)^4+200*x1(i)^2*z1(j)^2-400*x1(i)*x1(j)*z1(j)^2+200*x1(j)^2*z1(j)^2-200*z1(i)^2*z1(j)^2+100*z1(j)^4))))/(400*x1(i)^2-800*x1(i)*x1(j)+400*x1(j)^2+400*z1(i)^2-800*z1(i)*z1(j)+400*z1(j)^2);

```

```

r2=r2+(0.5*(400*x2(i)^2*z2(i)-800*x2(i)*x2(j)*z2(i)+400*x2(j)^2*z2(i)+400*z2(i)^3+400*x2(i)^2*z2(j)-800*x2(i)*x2(j)*z2(j)+400*x2(j)^2*z2(j)-400*z2(i)^2*z2(j)-400*z2(i)*z2(j)^2+400*z2(j)^3-1*sqrt((-400*x2(i)^2*z2(i)+800*x2(i)*x2(j)*z2(i)-400*x2(j)^2*z2(i)-400*z2(i)^3-400*x2(i)^2*z2(j)+800*x2(i)*x2(j)*z2(j)-400*x2(j)^2*z2(j)+400*z2(i)^2*z2(j)+400*z2(i)*z2(j)^2-400*z2(j)^3)^2-4*(400*x2(i)^2-800*x2(i)*x2(j)+400*x2(j)^2+400*z2(i)^2-800*z2(i)*z2(j)+400*z2(j)^2)*(-6241*x2(i)^2+100*x2(i)^4+12482*x2(i)*x2(j)-400*x2(i)^3*x2(j)-6241*x2(j)^2+600*x2(i)^2*x2(j)^2-400*x2(i)*x2(j)^3+100*x2(j)^4+200*x2(i)^2*z2(i)^2-400*x2(i)*x2(j)*z2(i)^2+200*x2(j)^2*z2(i)^2+100*z2(i)^4+200*x2(i)^2*z2(j)^2-400*x2(i)*x2(j)*z2(j)^2+200*x2(j)^2*z2(j)^2-200*z2(i)^2*z2(j)^2+100*z2(j)^4))))/(400*x2(i)^2-800*x2(i)*x2(j)+400*x2(j)^2+400*z2(i)^2-800*z2(i)*z2(j)+400*z2(j)^2);

```

```

z10=r1/(i-lamda+1);

```

```

z20=r2/(i-lamda+1);

```

```

if j-i<miu-1

```

```

    break;

```

```

end

```

```

%fsolve([(x1(i)-x)^2+(z1(i)-z)^2-3.95^2;(x1(j)-x)^2+(z1(j)-z)^2-3.95^2],[-100,x1(j)],optimset('fsolve'))];solve([(x1(i)-x)^2+(z1(i)-z)^2-3.95^2;(x1(j)-x)^2+(z1(j)-z)^2-3.95^2])

```

```

end

```

```

%x10

```

```

%x20

```

```

Dx=x20-x10;

```

```

Dz=z20-z10;

```

```

FirstData(:,1)=FirstData(:,1)-Dx;

```

```

FirstData(:,3)=FirstData(:,3)-Dz;

```

```

%x20

```

```

for i=1:564

```

```

    t(i)=0;

```

```

    r(i)=0;

```

```

end
for i=1:564
    x1(i)=(i-1)*0.00275-Dx;
    %x1(i)
    x2(i)=(i-1)*0.00275;
    for j=1:756
        t(i)=FirstData(i+564*(j-1),3)+t(i);
        r(i)=SecondData(i+564*(j-1),3)+r(i);
    end
    z1(i)=t(i)/756;
    z2(i)=r(i)/756;
end
% for i=546:-1:1
%     % x20-x1(i)
%     % z20-z1(i)
%
%     a(i)=abs(sqrt((x20-x1(i))^2+(z20-z1(i))^2)-3.95)/3.95
% end
% plot(a)
%
disp('          请输入 beta 的值, 本题取 0.002 较为合适          ');%凭经验和图像
观察选取
beta=input('请输入: ');
for i=1:564
    %     x1(i)
    %     z1(i)
    %     abs(sqrt((x20-x1(i))^2+(z20-z1(i))^2)-3.95)/3.95
    if abs(sqrt((x20-x1(i))^2+(z20-z1(i))^2)-3.95)/3.95<beta
        FenJieX1=x1(i);
        FenJieZ1=z1(i);
        break;
    end
end
end
for i=1:564
    if abs(sqrt((x20-x2(i))^2+(z20-z1(i))^2)-3.95)/3.95<beta
        FenJieX2=x2(i);
        FenJieZ2=z2(i);
        break;
    end
end
Dtheta=atan((FenJieZ2-z20)/(FenJieX2-x20))-atan((FenJieZ1-z10)/(FenJieX1-x10));
%360*(Dtheta/(2*pi))
for i=1:564

```

```

TempFirstDataX(i)=x20+cos(atan((FirstData(i,3)-z20)/(FirstData(i,1)-x20))+Dtheta)*
sqrt((FirstData(i,3)-z20)^2+(FirstData(i,1)-x20)^2);
TempFirstDataZ(i)=z20+sin(atan((FirstData(i,3)-z20)/(FirstData(i,1)-x20))+Dtheta)*s
qrt((FirstData(i,3)-z20)^2+(FirstData(i,1)-x20)^2);
end
for i=1:564
FirstData(i,1)=TempFirstDataX(i);
    FirstData(i,3)=TempFirstDataZ(i);
end
%FirstData(1,1)
if FirstData(1,1)<SecondData(1,1)
    q=floor(abs(Dx)/0.00275);
    if q>0
        for i=1:756
            FirstData(1+(i-1)*564:q+(i-1)*564,:)=[];
        end
    else
        q=floor(abs(Dx)/0.00275);

        if q>0
            for i=1:756

NewSecond(1+(i-1)*(564-q+1):i*(564-q+1),:)=SecondData(q+(i-1)*564:i*564,:);
                %size(SecondData)
                %A=SecondData(q,564)
            end
        end
    end
end
end
%FirstData=reshape(FirstData,3,3);
%FirstData=reshape(FirstData,100,3)
%=FirstName
%FirstData
disp('          x 轴平移误差为');
Dx
disp('          z 轴平移误差为');
Dz
disp('          转动误差为（单位：度）');
360*(Dtheta/(2*pi))
End

```

### 附录 3

```

*****
*****

```

## 第二问程序

```
*****
*****
*****
```

### 2.1 方案一 中值滤波+低通滤波+高斯滤波

```
*****
%%%%%%%%%%产生一个 xz 轴平面
load F:\建模\题\6x2 次棱数据\77T1-1203959\c1.dat
%size(c1)
i=1:564;      %x 轴点的个数。
x=zeros(1,564);
x(i)=0:0.00275:1.54825;
for i=1:564
    z(i)=c1((564*249)+i,3);
end
figure(2)
plot(x,z)
xlabel('x 轴的值');ylabel('z 轴的初始值');title('xoz 平面截面图');
%%%%%%%%%%
%%%进行中值滤波
zz=medianfun(z,5);
figure(3)
plot(x,zz)
xlabel('x 轴的值');ylabel('中值滤波后 z 轴的值');title('xoz 平面截面图');
%%%%%%%%%%
%%%%%%%%%%设计低通滤波器
Fs=363;
[b,a]=butter(9,100/363,'low');
[H,w]=freqz(b,a,512);
figure(4)
plot(w*Fs/(2*pi),abs(H));
xlabel('w');ylabel('低通滤波器的频率响应');title('低通滤波器');
%%%%%%%%%%
%%%%%%%%%%
%信号进行滤波
sf=filter(b,a,zz);
figure(5)
plot(x,sf);
xlabel('x 轴的值');ylabel('低通滤波后 z 轴的值');title('xoz 平面截面图');
%%%%%%%%%%
%%%%%%%%%%进行高斯滤波
yy=filter(gaussfiltcoef(363,20),1,sf);
figure(6)
```

```

plot(x,yy);
xlabel('x 轴的值');ylabel('高斯滤波后 z 轴的值');title('xoz 平面截面图');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(7) %对于滤波后加入的新的误差
x1=x(40:524);
yyy=yy(40:524);
plot(x1,yyy);
xlabel('x 轴的值');ylabel('滤除新误差后 z 轴的值');title('xoz 平面截面图');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%看前后频谱图
S=fft(z,512);
SF=fft(yy,512);
w1=(0:255)/256*(Fs/2);
figure(8)
subplot(1,2,1)
plot(w1,abs(S(1:256)));
xlabel('w1');ylabel('初始离散 z 值的频率响应');
subplot(1,2,2)
plot(w1,abs(SF(1:256)));
xlabel('w1');ylabel('经滤波后离散 z 值的频率响应');

```

\*\*\*\*\*

## 2.2 中值滤波函数

\*\*\*\*\*

%中值滤波函数

%实现方法:

% 1: 通过从图像中的某个采样窗口取出奇数个数据进行排序

% 2: 用排序后的中值取代要处理的数据即可

function [y]=median(x,a) %a=5 x 输入为行向量

```

N=length(x);
b=(a-1)/2;
y=zeros(N,1);
temp=zeros(N+4);
x1=zeros(N);
x2=zeros(N);
x1(1:2)=0;
x1(N+3:N+4)=0;
x1(3:N+2)=x;
for i=3:N+2
    temp=x1(i-b:i+b);
    tempsort=sort(temp);
    x2(i)=tempsort(b+1);
end

```



```
y(1:N)=x2(3:N+2);
y=y';
```

\*\*\*\*\*

### 2.3 高斯滤波系数函数

\*\*\*\*\*

```
function b=gaussfiltcoef(SR,fco)
%GAUSSFILTCOEFF Return coefficients of Gaussian lowpass filter.
% SR=sampling rate, fco=cutoff (-3dB) freq, both in Hz.
% Coeffs for FIR filter of length L (L always odd) are computed.
% This symmetric FIR filter of length L=2N+1 has delay N/SR seconds.
% Examples of use
% Compute Gaussian filter frequency response for SR=1000, fco=50 Hz:
% freqz(gaussfiltcoef(1000,50),1,256,1000);
% Filter signal X sampled at 5kHz with Gaussian filter with fco=500:
% y=filter(gaussfiltcoef(5000,500),1,X);
% SR, fco are not sanity-checked. WCR 2006-10-11.
```

```
b=0;
a=3.011*fco;
N=ceil(0.398*SR/fco); %filter half-width, excluding midpoint
%Width N corresponds to at least +-3 sigma which captures at least 99.75%
%of area under Normal density function. sigma=1/(a*sqrt(2pi)).
L=2*N+1; %full length of FIR filter
for k=-N:N
    b(k+N+1)=3.011*(fco/SR)*exp(-pi*(a*k/SR)^2);
end;
%b(k) coeffs computed above will add to almost exactly unity, but not
%quite exact due to finite sampling and truncation at +- 3 sigma.
%Next line adjusts to make coeffs b(k) sum to exactly unity.
b=b/sum(b);
```

\*\*\*\*\*

### 2.4 自编的滤波器函数 (lvbo)

\*\*\*\*\*

```
%%%%%%%%%%%%%%
% 输入为 y 固定时, xoz 平面的 z 值, 输出为经过中值滤波, 低通滤波, 高斯滤波后的 z 值。
function zzz=lvbo(z)
%%%%%%%%%%%%%%
%%%进行中值滤波
zz=medianfun(z,5);
%figure(3)
%plot(x,zz)
```

```

%低通滤波器的设计
%采用 BUtterworth 型
%采样率 363HZ 截止频率 100HZ
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%设计低通滤波器
Fs=363;
[b,a]=butter(9,100/363,'low');
[H,w]=freqz(b,a,512);
%figure(4)
%plot(w*Fs/(2*pi),abs(H));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%
%信号进行滤波
sf=filter(b,a,zz);
%figure(5)
%plot(x,sf);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%进行高斯滤波
yy=filter(gaussfiltcoef(363,20),1,sf);
%figure(6)
%plot(x,yy);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%
%看前后频谱图
S=fft(z,512);
SF=fft(sf,512);
w1=(0:255)/256*(Fs/2);
%figure(7)
%plot(w1,abs(S(1:256)));
%figure(8)
%plot(w1,abs(SF(1:256)));
zzz=yy;
*****

2.5 方案二 小波去噪
*****

load F:\建模\题\6x2 次棱数据\77T1-1203959\c1.dat
size(c1)
i=1:564; %x 轴点的个数。
x(i)=0:0.00275:1.54825;
for i=1:564
    z(i)=c1((564*249)+i,3);
end
%figure(1)
%plot(x,z)

```

```

%y=filter(gaussfiltcoef(363,20),1,z);
%figure(2)
%plot(x,y)
y=wden(z,'heursure','s','mln',5,'sym8'); %小波变换 作用范围对高斯噪声起的效果较大
figure(1)
subplot(1,2,1)
plot(x,z)
xlabel('x 轴的值');ylabel('z 轴的初始值');title('XOZ 平面截面图');
subplot(1,2,2)
plot(x,y)
xlabel('x 轴的值');ylabel('小波滤波后 z 轴的值');title('XOZ 平面截面图');

```

```

*****
*****

```

### 第三问

```

*****
*****

```

3.1 特征提取的程序（此程序用到了第二问中的 medianfun lvbo gaussfiltcoef 函数）

```

*****

```

#### 3.1.1 y 轴 250-500 行的截面图

```

*****

```

%此程序的作用是画出 200-500 行的 XZ 轴的截面图

```

%%%%%%%%%%

```

%%%%%%%%产生一个 xz 轴平面

```

load F:\建模\题6x2 次棱数据\77T1-1203959\c1.dat

```

```

load F:\建模\题6x2 次棱数据\77T1-1203959\c2.dat

```

```

load F:\建模\题6x2 次棱数据\77T1-1203959\c3.dat

```

```

load F:\建模\题6x2 次棱数据\77T1-1203959\c4.dat

```

```

%load c1.dat

```

```

%size(c1)

```

```

for j=250:500;

```

```

i=1:564; %x 轴点的个数。

```

```

x(i)=0:0.00275:1.54825;

```

```

for i=1:564

```

```

    z1(i)=c1((564*(j-1))+i,3);

```

```

    z2(i)=c2((564*(j-1))+i,3);

```

```

    z3(i)=c3((564*(j-1))+i,3);

```

```

    z4(i)=c4((564*(j-1))+i,3);

```

```

end

```

```

zzz=zeros(1,564);

```

```

zzz1=lvbo(z1);

```

```

figure(2)

```

```

plot(x,zzz1)
xlabel('x 轴坐标'); ylabel('滤波后的 z 轴值');title('c1,y 轴 250 行到 500 行的截面图');
hold on
zzz2=lvbo(z2);
zzz3=lvbo(z3);
zzz4=lvbo(z4);
figure(3)
subplot(2,2,1)
plot(x,zzz1);
xlabel('x 轴坐标'); ylabel('滤波后的 z 轴值');title('c1,y 轴 250 行到 500 行的截面图');
hold on
subplot(2,2,2)
plot(x,zzz2);
xlabel('x 轴坐标'); ylabel('滤波后的 z 轴值');title('c2,y 轴 250 行到 500 行的截面图');
hold on
subplot(2,2,3)
plot(x,zzz3);
xlabel('x 轴坐标'); ylabel('滤波后的 z 轴值');title('c3,y 轴 250 行到 500 行的截面图');
hold on
subplot(2,2,4)
plot(x,zzz4);
xlabel('x 轴坐标'); ylabel('滤波后的 z 轴值');title('c4,y 轴 250 行到 500 行的截面图');
hold on
figure(4)
%%%%%%%%%%%%去掉前面的几个由误差的点后的截面
Z1=zzz1(40:524);
Z2=zzz2(40:524);
Z3=zzz3(40:524);
Z4=zzz4(40:524);
x1=x(40:524);
subplot(2,2,1)
plot(x1,Z1);
xlabel('x 轴坐标'); ylabel('滤波后的 z 轴值');title('c1,y 轴 250 行到 500 行的截面图');
hold on
subplot(2,2,2)
plot(x1,Z2);
xlabel('x 轴坐标'); ylabel('滤波后的 z 轴值');title('c2,y 轴 250 行到 500 行的截面图');

```

```

');
hold on
subplot(2,2,3)
plot(x1,Z3);
xlabel('x 轴坐标'); ylabel('滤波后的 z 轴值');title('c3,y 轴 250 行到 500 行的截面图');
hold on
subplot(2,2,4)
plot(x1,Z4);
xlabel('x 轴坐标'); ylabel('滤波后的 z 轴值');title('c4,y 轴 250 行到 500 行的截面图');
hold on
end

```

\*\*\*\*\*

### 3.1.2 y 轴 250-500 行均值的截面图

\*\*\*\*\*

%此程序的作用是画出 200-500 行均值的 XZ 轴的截面图

%%%%%%%%%

%%%%%%%%%产生一个 xz 轴平面

load F:\建模\题\6x2 次棱数据\77T1-1203959\c1.dat

load F:\建模\题\6x2 次棱数据\77T1-1203959\c2.dat

load F:\建模\题\6x2 次棱数据\77T1-1203959\c3.dat

load F:\建模\题\6x2 次棱数据\77T1-1203959\c4.dat

%load c1.dat

%size(c1)

for j=250:500;

i=1:564; %x 轴点的个数。

x(i)=0:0.00275:1.54825;

for i=1:564

z1(i)=c1((564\*(j-1))+i,3);

z2(i)=c2((564\*(j-1))+i,3);

z3(i)=c3((564\*(j-1))+i,3);

z4(i)=c4((564\*(j-1))+i,3);

end

zzz=zeros(1,564);

zzz1=lvbo(z1);

figure(1)

plot(x,zzz1)

xlabel('x 轴坐标'); ylabel('滤波后的 z 轴值');title('c1,y 轴 250 行到 500 行的截面图');

);

hold on

zzz2=lvbo(z2);

zzz3=lvbo(z3);

zzz4=lvbo(z4);

```

zzz10=0;
zzz20=0;
zzz30=0;
zzz40=0;
zzz10=zzz10+zzz1;
zzz20=zzz20+zzz2;
zzz30=zzz30+zzz3;
zzz40=zzz40+zzz4;
%%%%%%%%%%去掉前面的几个由误差的点后的截面
Z1=zzz1(40:524);
Z2=zzz2(40:524);
Z3=zzz3(40:524);
Z4=zzz4(40:524);
Z10=0;
Z20=0;
Z30=0;
Z40=0;
Z10=Z10+Z1;
Z20=Z20+Z2;
Z30=Z30+Z3;
Z40=Z40+Z4;
x1=x(40:524);
end
figure(2)
subplot(2,2,1)
plot(x,zzz10/251);
xlabel('x 轴坐标'); ylabel('滤波后的 z 轴值');title('c1,y 轴 250 行到 500 行均值的截面图');
%hold on
subplot(2,2,2)
plot(x,zzz20/251);
xlabel('x 轴坐标'); ylabel('滤波后的 z 轴值');title('c2,y 轴 250 行到 500 行均值的截面图');
%hold on
subplot(2,2,3)
plot(x,zzz30/251);
xlabel('x 轴坐标'); ylabel('滤波后的 z 轴值');title('c3,y 轴 250 行到 500 行均值的截面图');
%hold on
subplot(2,2,4)
plot(x,zzz40/251);
xlabel('x 轴坐标'); ylabel('滤波后的 z 轴值');title('c4,y 轴 250 行到 500 行均值的截面图');

```

```

%hold on
figure(3)
subplot(2,2,1)
plot(x1,Z10/251);
xlabel('x 轴坐标'); ylabel('滤波后的 z 轴值');title('c1,y 轴 250 行到 500 行均值的截面图');
%hold on
subplot(2,2,2)
plot(x1,Z20/251);
xlabel('x 轴坐标'); ylabel('滤波后的 z 轴值');title('c2,y 轴 250 行到 500 行均值的截面图');
%hold on
subplot(2,2,3)
plot(x1,Z30/251);
xlabel('x 轴坐标'); ylabel('滤波后的 z 轴值');title('c3,y 轴 250 行到 500 行均值的截面图');
%hold on
subplot(2,2,4)
plot(x1,Z40/251);
xlabel('x 轴坐标'); ylabel('滤波后的 z 轴值');title('c4,y 轴 250 行到 500 行均值的截面图');
%hold on

```

\*\*\*\*\*

### 3.2 自编的 character 函数（以后可直接调用）

\*\*\*\*\*

%%%%%%%%%%

%函数作用：提取一个子弹的特征向量

    % 输入：同一幅图的次棱信息

    %输出：一个子弹次棱信息的特征向量

function B=character(c1,c2,c3,c4,a)    %c1,c2,c3,c4 为同一幅图的四个次棱，a 为 x 轴上数

据点的个数

        %默认时 a=564

```

for j=250:500;
i=1:a;        %x 轴点的个数。
x(i)=0:0.00275:1.54825;
for i=1:a
    z1(i)=c1((a*(j-1))+i,3);
    z2(i)=c2((a*(j-1))+i,3);
    z3(i)=c3((a*(j-1))+i,3);
    z4(i)=c4((a*(j-1))+i,3);
end
zzz=zeros(1,a);

```

```

zzz1=lvbo(z1);
zzz2=lvbo(z2);
zzz3=lvbo(z3);
zzz4=lvbo(z4);
zzz10=0;
zzz20=0;
zzz30=0;
zzz40=0;
zzz10=zzz10+zzz1;
zzz20=zzz20+zzz2;
zzz30=zzz30+zzz3;
zzz40=zzz40+zzz4;
%%%%%%%%%%%%去掉前面的几个由误差的点后的截面
Z1=zzz1(40:a-40);
Z2=zzz2(40:a-40);
Z3=zzz3(40:a-40);
Z4=zzz4(40:a-40);
Z10=0;
Z20=0;
Z30=0;
Z40=0;
Z10=Z10+Z1;
Z20=Z20+Z2;
Z30=Z30+Z3;
Z40=Z40+Z4;
x1=x(40:a-40);
end
B=[Z10;Z20;Z30;Z40]';
size(B);

```

\*\*\*\*\*

### 3.3 test.m

比较两个弹头是否匹配的程序,并找出最佳的匹配及相应的相关系数.

输入两个弹头各个次棱的数据

输出各个次棱的相关系数矩阵,最佳的次棱匹配

\*\*\*\*\*

%比较两个次棱是否匹配的程序,并找出最佳的匹配及相应的相关系数.

load c1.dat c2.dat c3.dat c4.dat c11.dat c12.dat c13.dat c14.dat

%c1.dat,c2.dat,c3.dat,c4.dat 表示要被匹配的弹头的 4 个次棱的处理后数

据,c11.dat,c12.dat,c13.dat,c14.dat 表示和已有的弹头进行匹配的另一个弹头各个次棱的数



据.

```

%x=0:0.00275:1.54825;%表示 x 轴以步长 2.75 微米的数组
%y=0:0.00275:2.07625;%表示 y 轴以步长 2.75 微米的数组
a=[c1(:,3) c2(:,3) c3(:,3) c4(:,3)];
%将要被匹配弹头的各个次棱的深度的数据整合成一个矩阵.
b=[c11(:,3) c12(:,3) c13(:,3) c14(:,3)];
%将要与已有的弹头进行匹配的另一个弹头的各个次棱的深度的数据整合成一个矩阵.
%h=zeros(4,4);
format long
m=1:4;
k=0;
for m1=m(1):1:m(end);
    k=k+1;
    n=1:4;
    k1=0;
    for n1=n(1):1:n(end);
        k1=k1+1;
z=zeros(756,564);
z1=zeros(756,564);
for j=1:564;
    for i=1:756;
        z(i,j)=a((i-1)*564+j,k);
    end
end
%将 cm.dat(若 m=1,则为 c1.dat,以此类推)中 z 轴的数据变成和以 2.75 微米为步长的 x,y 数据所对
应的矩阵 z.
for j=1:564;
    for i=1:756;
        z1(i,j)=b((i-1)*564+j,k1);
    end
end
%将 c1n.dat 中 z 轴的数据变成和以 2.75 微米为步长的 x,y 数据所对应的矩阵 z.
d=zeros(1,564);
for j=1:564;
    e=z(250:500,j);
    d(1,j)=mean(e);
end
%将矩阵 z 中取 y 轴中间的 1/3 的数据,计算 250-500 行的均值,并赋值到新的数组 d 中
d1=zeros(1,564);
for j=1:564;

```

```

        e=z1(250:500,j);
        d1(1,j)=mean(e);
    end
    %将矩阵 z1 中取 y 轴中间的 1/3 的数据,计算 250-500 行的均值,并赋值到新的数组 d1 中
    g=corrcoef(d,d1);
    f=g(2,1);
    %求出数组 d 和 d1 的相关系数,用 f 表示
    h(k,k1)=f;
    %将数据 cm 和 c1n 按照上述方法所得到的相关系数的均值赋值到矩阵 h 中
    end
end
h
%输出按照上述方法得到的相关系数矩阵,例如 h(i,j)表示 ci 和 c1j 之间的相关系数.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%下面找出最佳的次棱匹配
r(1)=(h(1,1)+h(2,2)+h(3,3)+h(4,4))/4;%当 c1.dat 与 c11.dat 匹配时各个匹配的次棱的相关系数的均值.
r(2)=(h(1,2)+h(2,3)+h(3,4)+h(4,1))/4;%当 c1.dat 与 c12.dat 匹配时各个匹配的次棱的相关系数的均值.
r(3)=(h(1,3)+h(2,4)+h(3,1)+h(4,2))/4;%当 c1.dat 与 c13.dat 匹配时各个匹配的次棱的相关系数的均值.
r(4)=(h(1,4)+h(2,1)+h(3,2)+h(4,3))/4;%当 c1.dat 与 c14.dat 匹配时各个匹配的次棱的相关系数的均值.
r1=max(r)
k=1;
while r(k)~=r1;
    k=k+1;
end
k %求出与 c1.dat 匹配的 c1k.dat
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
3.4 自编的弹头匹配函数
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%该函数的作用是对两颗子弹进行匹配
%输入: a 第一颗子弹的特征向量 b 第二颗子弹的特征向量
%输出: h 次棱两两的相似度 r1 a, b 之间的相似度, k 已那种方式进行匹配, 共有 4 种方式, 见论文, K=1, 2, 3, 4
function [h,r1,k]=charactermatch(a,b) %a 第一颗子弹 % b 为第二颗子弹

```

```

%h=zeros(4,4);
format long
m=1:4;
k=0;
for m1=m(1):1:m(end);
    k=k+1;
    n=1:4;
    k1=0;
    for n1=n(1):1:n(end);
        k1=k1+1;
        g=corrcoef(a(:,k),b(:,k1));
        f=g(2,1);
        %求出数组 d 和 d1 的相关系数,用 f 表示
        h(k,k1)=f;
        %将数据 cm 和 c1n 按照上述方法所得到的相关系数的均值赋值到矩阵 h 中
    end
end
h;
%输出按照上述方法得到的相关系数矩阵,例如 h(i,j)表示 ci 和 c1j 之间的相关系数.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%下面找出最佳的次棱匹配
r(1)=(h(1,1)+h(2,2)+h(3,3)+h(4,4))/4;%当 c1.dat 与 c11.dat 匹配时各个匹配的次棱
的相关系
数的均值.
r(2)=(h(1,2)+h(2,3)+h(3,4)+h(4,1))/4;%当 c1.dat 与 c12.dat 匹配时各个匹配的次棱
的相关系
数的均值.
r(3)=(h(1,3)+h(2,4)+h(3,1)+h(4,2))/4;%当 c1.dat 与 c13.dat 匹配时各个匹配的次棱
的相关系
数的均值.
r(4)=(h(1,4)+h(2,1)+h(3,2)+h(4,3))/4;%当 c1.dat 与 c14.dat 匹配时各个匹配的次棱
的相关系
数的均值.
r1=max(r);
k=1;
while r(k)~=r1;
    k=k+1;
end
k;%求出与 c1.dat 匹配的 c1k.dat
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

                第二问和第三问联合的程序

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```
*****
*****
```

2\_3

%此程序要用到以上的如下函数 character charactermatch median lvbo gaussfiltcoef  
%此程序的作用得到两幅图次棱的两两相似度，两个子弹的相似度，以及匹配的方式

%%%%%%%%产生一个 xz 轴平面

```
load F:\建模\题\6x2 次棱数据\77T1-1504519\c1.dat
load F:\建模\题\6x2 次棱数据\77T1-1504519\c2.dat
load F:\建模\题\6x2 次棱数据\77T1-1504519\c3.dat
load F:\建模\题\6x2 次棱数据\77T1-1504519\c4.dat
load F:\建模\题\6x2 次棱数据\77T2-1504519\c11.dat
load F:\建模\题\6x2 次棱数据\77T2-1504519\c12.dat
load F:\建模\题\6x2 次棱数据\77T2-1504519\c13.dat
load F:\建模\题\6x2 次棱数据\77T2-1504519\c14.dat
```

```
a1=564;
```

```
b1=character(c1,c2,c3,c4,a1);
```

```
b2=character(c11,c12,c13,c14,a1);
```

```
[h,r1,k]=charactermatch(b1,b2);
```

```
h
```

```
r1
```

```
k
```

```
*****
*****
```

#### 第四问程序

```
*****
*****
*****
```

##### 4.1 最优化的第四问程序 disiwen.m

```
*****
```

%此程序的作用 disiwen 直接将 88 个数据导入，然后进行特征提取，在进行特征比对，

但是 pc 机无法运行，outofmemory

%%%%%%%%产生一个 xz 轴平面

```
A(:,1)=load('F:\建模\题\第四问 1\第四问题的数据之一\t01-T1-1504519\c1.dat');
A(:,2)=load('F:\建模\题\第四问 1\第四问题的数据之一\t01-T1-1504519\c2.dat');
A(:,3)=load('F:\建模\题\第四问 1\第四问题的数据之一\t01-T1-1504519\c3.dat');
A(:,4)=load('F:\建模\题\第四问 1\第四问题的数据之一\t01-T1-1504519\c4.dat');
A(:,5)=load('F:\建模\题\第四问 1\第四问题的数据之一\t02-t1-1812492\c1.dat');
A(:,6)=load('F:\建模\题\第四问 1\第四问题的数据之一\t02-t1-1812492\c2.dat');
A(:,7)=load('F:\建模\题\第四问 1\第四问题的数据之一\t02-t1-1812492\c3.dat');
```

[illegible]

```

A(:,49)=load('F:\建模\题第四问 2\第四问题的数据之二\t13-t1-1811345\c1.dat');
A(:,50)=load('F:\建模\题第四问 2\第四问题的数据之二\t13-t1-1811345\c2.dat');
A(:,51)=load('F:\建模\题第四问 2\第四问题的数据之二\t13-t1-1811345\c3.dat');
A(:,52)=load('F:\建模\题第四问 2\第四问题的数据之二\t13-t1-1811345\c4.dat');
A(:,53)=load('F:\建模\题第四问 2\第四问题的数据之二\t14-t1-1814117\c1.dat');
A(:,54)=load('F:\建模\题第四问 2\第四问题的数据之二\t14-t1-1814117\c2.dat');
A(:,55)=load('F:\建模\题第四问 2\第四问题的数据之二\t14-t1-1814117\c3.dat');
A(:,56)=load('F:\建模\题第四问 2\第四问题的数据之二\t14-t1-1814117\c4.dat');
A(:,57)=load('F:\建模\题\第四问 2\第四问题的数据之二\t15-T1-1923252\c1.dat');
A(:,58)=load('F:\建模\题\第四问 2\第四问题的数据之二\t15-T1-1923252\c2.dat');
A(:,59)=load('F:\建模\题\第四问 2\第四问题的数据之二\t15-T1-1923252\c3.dat');
A(:,60)=load('F:\建模\题\第四问 2\第四问题的数据之二\t15-T1-1923252\c4.dat');
A(:,61)=load('F:\建模\题\第四问 2\第四问题的数据之二\t16-T1-1930832\c1.dat');
A(:,62)=load('F:\建模\题\第四问 2\第四问题的数据之二\t16-T1-1930832\c2.dat');
A(:,63)=load('F:\建模\题\第四问 2\第四问题的数据之二\t16-T1-1930832\c3.dat');
A(:,64)=load('F:\建模\题\第四问 2\第四问题的数据之二\t16-T1-1930832\c4.dat');
A(:,65)=load('F:\建模\题\第四问 2\第四问题的数据之二\t17-T2-1202999\c1.dat');
A(:,66)=load('F:\建模\题\第四问 2\第四问题的数据之二\t17-T2-1202999\c2.dat');
A(:,67)=load('F:\建模\题第四问 2\第四问题的数据之二\t17-T2-1202999\c3.dat');
A(:,68)=load('F:\建模\题\第四问 2\第四问题的数据之二\t17-T2-1202999\c4.dat');
A(:,69)=load('F:\建模\题\第四问 2\第四问题的数据之二\t18-T2-1504519\c1.dat');
A(:,70)=load('F:\建模\题\第四问 2\第四问题的数据之二\t18-T2-1504519\c2.dat');
A(:,71)=load('F:\建模\题\第四问 2\第四问题的数据之二\t18-T2-1504519\c3.dat');
A(:,72)=load('F:\建模\题\第四问 2\第四问题的数据之二\t18-T2-1504519\c4.dat');
A(:,73)=load('F:\建模\题第四问 2\第四问题的数据之二\t19-t2-1812492\c1.dat');
A(:,74)=load('F:\建模\题第四问 2\第四问题的数据之二\t19-t2-1812492\c2.dat');
A(:,75)=load('F:\建模\题第四问 2\第四问题的数据之二\t19-t2-1812492\c3.dat');
A(:,76)=load('F:\建模\题第四问 2\第四问题的数据之二\t19-t2-1812492\c4.dat');
A(:,77)=load('F:\建模\题\第四问 2\第四问题的数据之二

```

```

\t20-T2-1814688\c1.dat');
A(:,78)=load('F:\建模\题\第四问 2\第四问题的数据之二\t20-T2-1814688\c2.dat');
A(:,79)=load('F:\建模\题\第四问 2\第四问题的数据之二\t20-T2-1814688\c3.dat');
A(:,80)=load('F:\建模\题\第四问 2\第四问题的数据之二\t20-T2-1814688\c4.dat');
A(:,81)=load('F:\建模\题\第四问 2\第四问题的数据之二\t21-T2-1928033\c1.dat');
A(:,82)=load('F:\建模\题\第四问 2\第四问题的数据之二\t21-T2-1928033\c2.dat');
A(:,83)=load('F:\建模\题\第四问 2\第四问题的数据之二\t21-T2-1928033\c3.dat');
A(:,84)=load('F:\建模\题\第四问 2\第四问题的数据之二\t21-T2-1928033\c4.dat');
A(:,85)=load('F:\建模\题\第四问 2\第四问题的数据之二\t22-T2-1931817\c1.dat');
A(:,86)=load('F:\建模\题\第四问 2\第四问题的数据之二\t22-T2-1931817\c2.dat');
A(:,87)=load('F:\建模\题\第四问 2\第四问题的数据之二\t22-T2-1931817\c3.dat');
A(:,88)=load('F:\建模\题\第四问 2\第四问题的数据之二\t22-T2-1931817\c4.dat');
a1=564;
for i=1:22
    for j=1:22
        b1(i)=character(A(:,1+4*(i-1)),A(:,1+4*(i-1)+1),A(:,1+4*(i-1)+2),A
(:,1+4*(i-1)+3),a1);
        b2(j)=character(A(:,1+(j-1)*4),A(:,1+1+(j-1)*4),A(:,1+2+(j-1)*4),A
(:,1+3+(j-1)*4),a1);
        [h,r1,k]=charactermatch(b1,b2);
        R(i,j)=r1;
        K(i,j)=k;
    end
end
R
K

```

\*\*\*\*\*

#### 4.2 方法二:

利用较笨重的方法得到每个子弹的特征向量,然后利用下列函数 siwen 得到两两之间的相似度

```

*****
load d.mat
for i=1:22;
    for j=1:22;
        b1=d(:,i);
        b2=d(:,j);
        [h,r1,k]=charactermatch(b1,b2);
        % H(:,4*(i-1)+j)=h;
        R(i,j)=r1;
        K(i,j)=k;
    end
end
%H
sqrt(R)
%K

*****
*****
                                END
*****
*****

```