算法库.so文件位于demo下的app\libs目录下

例子中算法库的调用接口实现在native-lib.cpp中

**extern "C" JNIEXPORT int JNICALL**Java\_cn\_tureal\_trlzdemo1\_MainActivity\_Test2

Java里面看到的原型为

**public native int** Test2(**byte** []ZiTieStream, **int**[] UserStrokeStream, **int**[] StrokeScore);

输入：

ZiTieStream zt文件的数据流数组

UserStrokeStream 用户书写轨迹的数据流

格式如下：

总笔画数 该笔画点个数 点1X 点1Y 点2X 点2Y… 该笔画点个数 点1X 点1Y 点2X 点2Y… 该笔画点个数 点1X 点1Y 点2X 点2Y…

StrokeScore 用于接收评分的数组，评分个数为用户笔画数+1，分别对用户的每一笔和整体给出了评分，满分100分

输出：

返回值： 1 成功计算了参数 0 当前书写和字帖不符（笔画偏差太大）

如果返回值为1的话，函数还会将计算好的匹配参数保存于调用类的成员变量中，这些成员变量有：

**public int m\_nTurnCenterX** = 0;  
**public int m\_nTurnCenterY** = 0;  
**public int m\_nOffsetX** = 0;  
**public int m\_nOffsetY** = 0;  
**public float m\_fScale** = 0.001f;  
**public float m\_fTurnAngle** = 0.0f;

显示的时候根据用户书写点的原始坐标和匹配参数，通过一系列计算，就可以算出显示的新坐标，按照新坐标来进行显示，就可以将用户书写的内容和原字帖叠加到一起

计算公式如下：

用户书写原始点为pt, 字帖图片左上角坐标(0,0)，右下角坐标(fMyImgWidth, fMyImgHeight)，显示的点计算后新的坐标为(xnew, ynew)

**double** fCosA = Math.*cos*((**double**)**m\_fTurnAngle**);  
**double** fSinA = Math.*sin*((**double**)**m\_fTurnAngle**);

**double** x = (pt.**x** - **m\_nTurnCenterX**)\*fCosA - (pt.**y** - **m\_nTurnCenterY**)\*fSinA + **m\_nTurnCenterX**;  
**double** y = (pt.**x** - **m\_nTurnCenterX**)\*fSinA + (pt.**y** - **m\_nTurnCenterY**)\*fCosA + **m\_nTurnCenterY**;  
  
**int** xnew = (**int**) ((x \* **m\_fScale** + **m\_nOffsetX**) \* fMyImgWidth/416.0);  
**int** ynew = (**int**) ((y \* **m\_fScale** + **m\_nOffsetY**) \* fMyImgHeight/416.0);