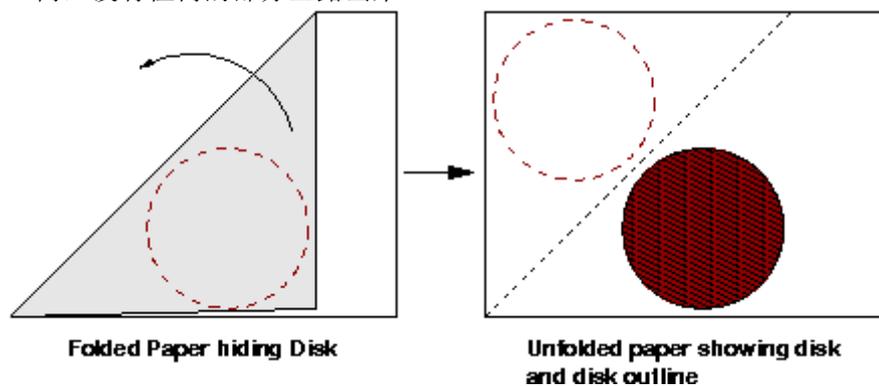


计算几何项目报告

夏东林 慕巍 见英

问题背景:

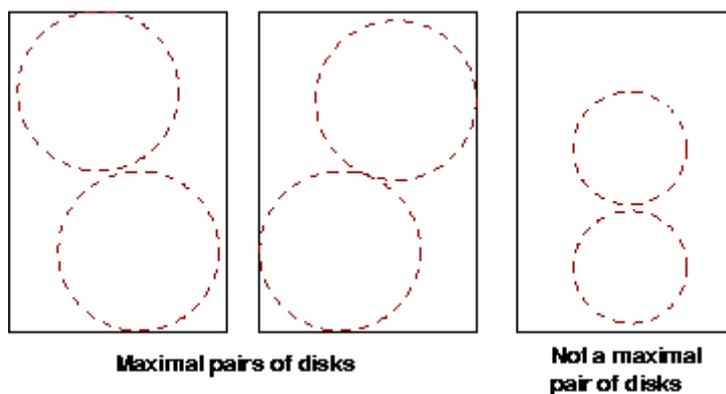
包扎问题是计算几何中的一类经典难题，我们研究的是这一问题的简化版本——One Single Fold。任意给定一个简单多边形 P 和一个圆盘 D ，仅仅折叠一次 P ，使得 D 的表面全部包围在 P 内，没有任何的部分显露出来。



那么，任意给定的简单多边形必然会存在满足一次折叠覆盖条件的最大 D 。我们的问题就是求出这样最大的 D ，并且给出折叠方案。

算法及原理:

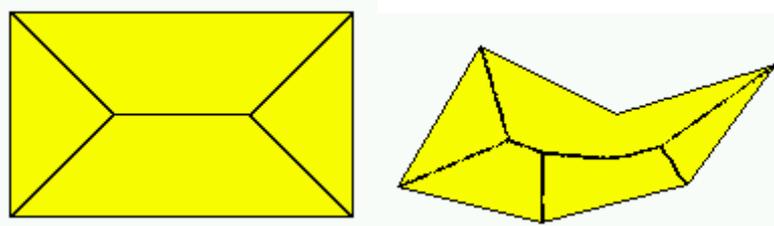
显而易见，这个问题等价于在 P 中求两个半径相同且不相重叠的最大圆，然后过这两个圆的圆心做一条垂直平分线，便是题目所求得包扎方案。



如图所示是在多边形中的相同大圆的一些例子。经过观察可知，如果给出的圆是最大圆的话，必然会满足一些必要条件。比如，必然与边界相切，而且不能仅与一条边界相切，换言之，两个大圆中至少要有一个与边界有两个以上的切点。由于圆的对称性，很容易知道其中一个圆心在多边形的中轴上。由此，Demaine 证明了如下的引理：

引理 1: 任意简单多边形中一对最大的半径相等且不重叠圆，必然可以通过移动使得他

们的圆心全部在多边形的中轴上。



Medial Axis of two polygons

这就保证了我们在寻找问题的解答时，只需要将圆心限定在多边形的中轴之上。进一步，Demaine 证明了两个大圆如果不相切，必然可以将其中的一个移动到多边形中轴的某个顶点上。

由于多边形中轴线可以分为直线段和抛物线段两类，所以有如下的结果：

中轴线上最大圆半径	
边 类 型	最大半径计算表达式
e_1, e_2 均在 直线 中轴 上	$2r = d(a_1 + \frac{r}{\sin\theta_1}\vec{x}_1, a_2 + \frac{r}{\sin\theta_2}\vec{x}_2)$ $p_1 = a_1 + \frac{r}{\sin\theta_1}\vec{x}_1 \qquad p_2 = a_2 + \frac{r}{\sin\theta_2}\vec{x}_2$ 圆心坐标
e_1 在直 线中 轴 上， e_2 在抛 物线 中轴 上	$4r^2 = (d(a_1 + \frac{r}{\sin\theta}\vec{x}_1, a_2 + r\vec{y} + (u_x + \sqrt{u_y(2r - u_y)})\vec{x}_2))^2$ $p_1 = a_1 + \frac{r}{\sin\theta_1}\vec{x}_1$ $p_2 = a_2 + r\vec{y} + (u_x + \sqrt{u_y(2r - u_y)})\vec{x}_2$
e_1, e_2 全部 在抛 物线 上	问题的方程是一个 8 次的方程，由于比较繁琐，暂不在这里给出。

根据如上的原理，我们设计的算法步骤如下：

1. 计算多边形 P 的中轴 M
2. for 中轴的每条边 e1, do:
 - (a) for 中轴的每个顶点 v, do:

找到圆心为 v 的在 多边形 P 内部的最大圆，设半径为 r
在边 e_1 上发现点 p ，使得圆心为 p 半径为 r 的圆在多边形 P 内，且 $d(p,v) \geq 2*r$
若找到则将其加入圆对队列

(b) for M 的每条边 e_2 do:

找到点 p_1 在边 e_1 上，点 p_2 在边 e_2 上， $r = d(p_1, p_2) / 2$ ，以点 p_1, p_2
为圆心， r 为半径分别作圆，且都在多边形内。

若找到则将其加入圆对队列

3. 在所有的圆对中发现半径最大的，即为结果。

系统设计

系统采用了以 VC 编程为基本框架，嵌入多种功能模块的方式。其中包括两类主要模块，第一个是经过 java 整理的多边形中轴计算模块，第二个是采用 Matlab 进行编写的方程求解模块。VC 编写的框架负责处理界面，数据的输入输出和中间两次的数据转换，数据合法性的验证。

过程中遇到的问题及解决对策

(1) 抛物线的多解问题

由于抛物线是二次曲线，解方程会有多根。我们将抛物线由顶点分割为两部分。

(2) 方程求解问题

由于线段边和抛物线边的方程是 4 次方的，抛物线边和抛物线边的方程是 8 次方的，因此求解时要利用牛顿插值。我们借用了 matlab 工具来解。

(3) 在 VC 中画抛物线

由于一般性的抛物线方程较为复杂，而且一般 x,y 坐标不是一一对应，所以要用参数表示。我们采用抛物线标准方程，用坐标变换进行对应。

(4) 简单多边形的判断

由于问题要求多边形为简单多边形，因此在输入时要加以判断。我们采用了多边形生成时逐点判断。

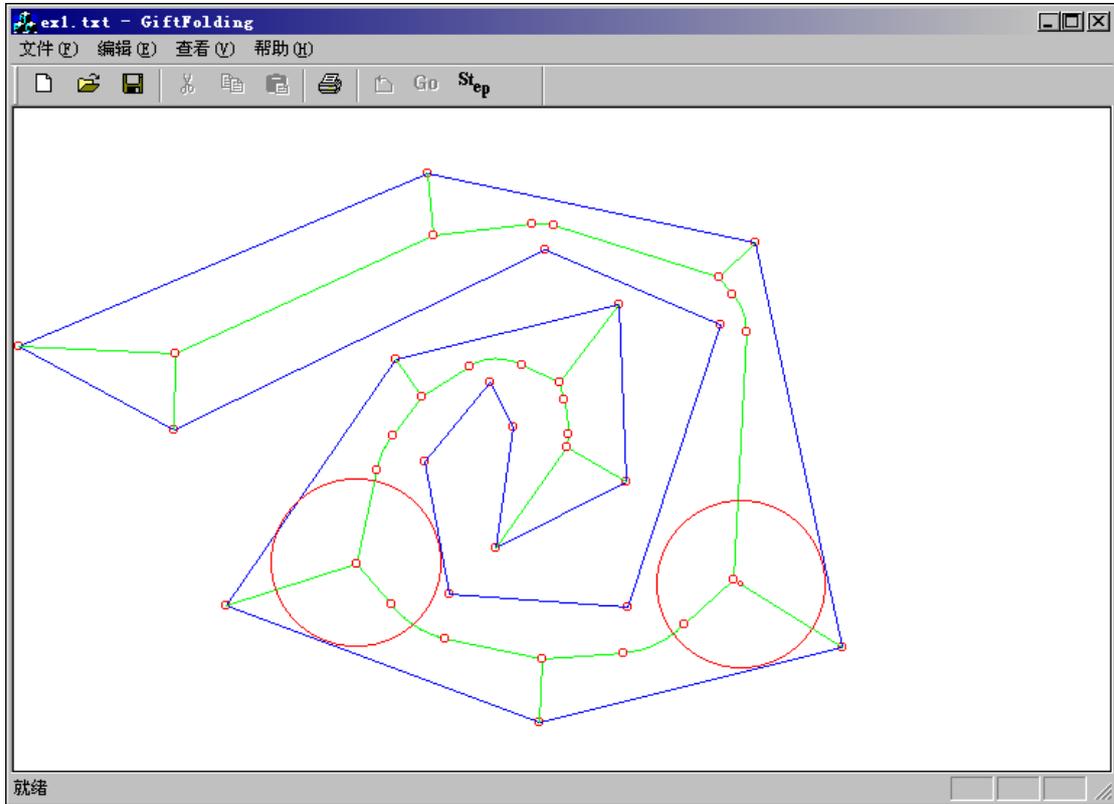
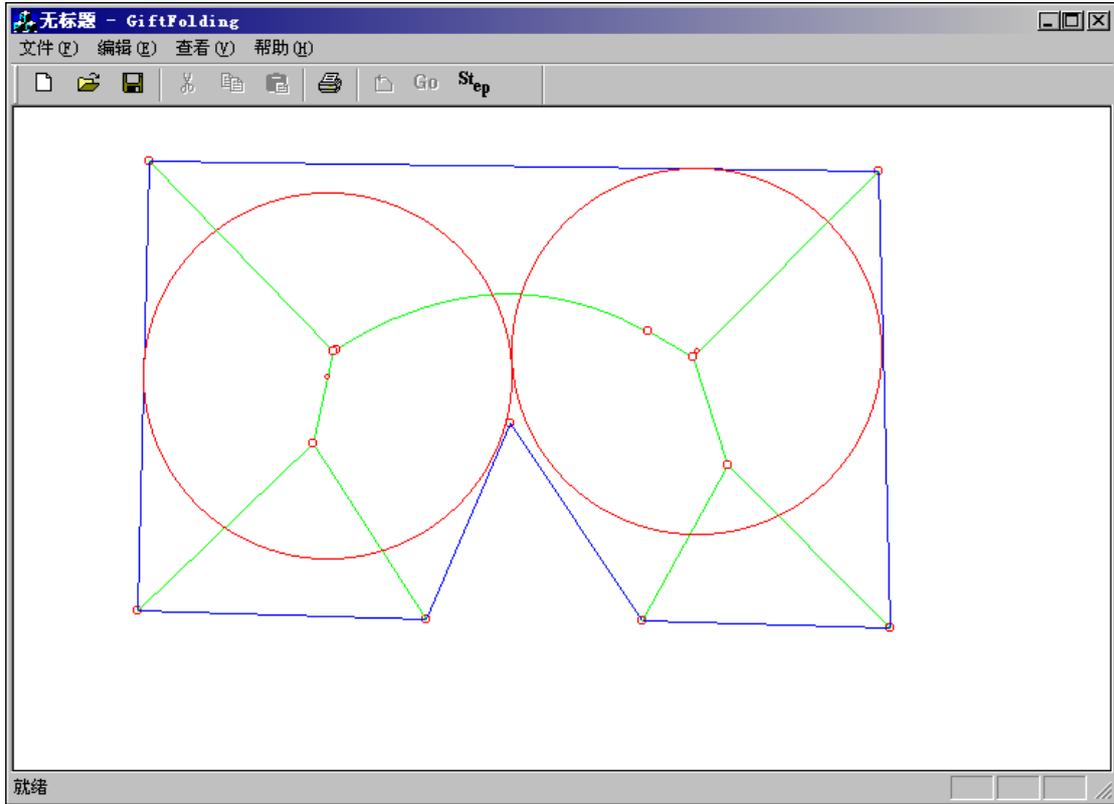
(5) Java, Matlab 接口

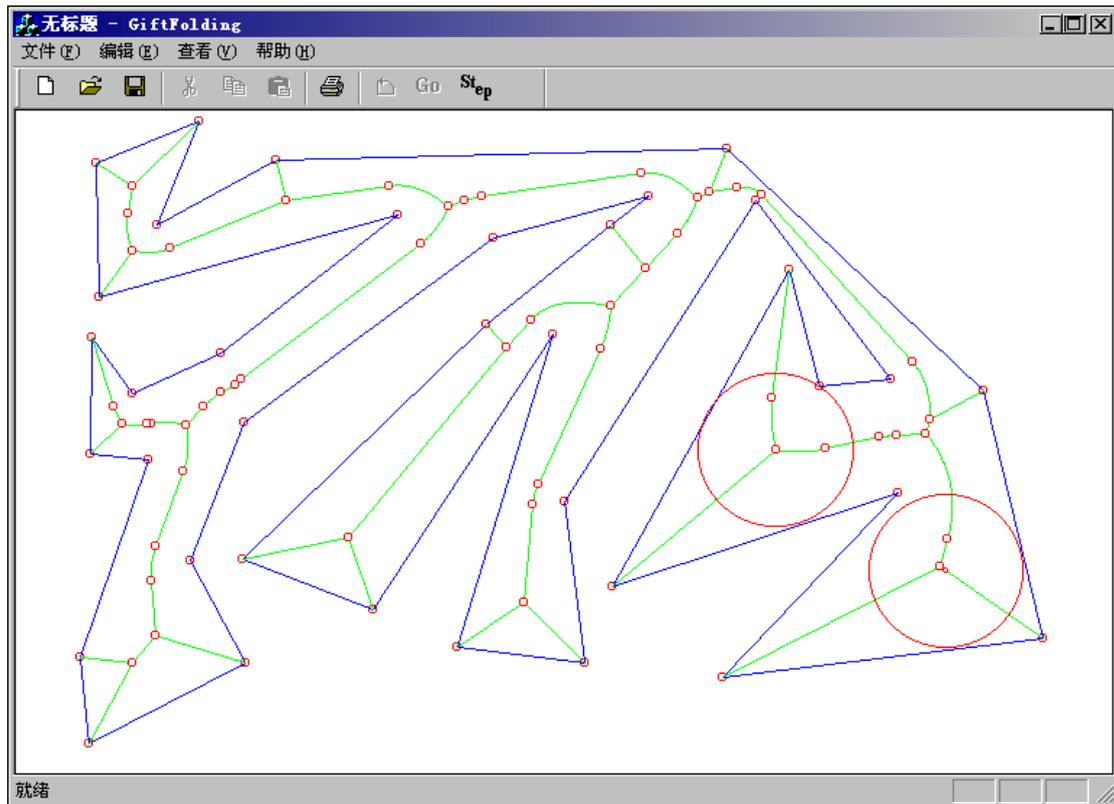
Java 调用采用了 VC 中的系统调用，用文件实现了参数传递。关于 Matlab 我们采用了 Matlab 引擎。

(6) 链表数据结构

对应边，点，我们定义了类，然后用链表模板加以解决。

测试结果





未解决问题

一、由于使用了多个模块集成的方式，增加了程序之间切换的系统调用，而且难以完全集成到一个单一的模式中，限制了其适用的范围，最遗憾的是不能在网页上方便的共享出来。

二、目前算法的计算复杂度为 N^2 ，其中 N 为中轴线段的段数，在非常复杂的情况下，仍然显得计算量很大。

有关文献

Therese C_ Biedl_ Erik D_ Demainez Martin L_ Demainey, "Hiding Disks in Folded Polygons", June 30 1998