第15卷 第3期 华侨大学学报(自然科学版) Vol. 15 No. 3
1994年7月 Journal of Huaqiao University (Natural Science) Jul. 1994

服装CAD的排料处理*

张全伙 房蓉晖 范慧琳

(华侨大学计算机科学系,泉州 362011)

摘要 阐述服装计算机辅助设计(CAD)中排料处理的基本思想,指出实现自动排料的关键,给出排料处理算法及其设计的主要步骤,并用C语言加以实现.

关键词 计算机辅助设计,服装,排料,算法,多边形

分类号 TP 391.72 : TS 941

服装生产自动化是服装工业的发展趋势,服装CAD是服装生产自动化的重要环节,而排料处理又是服装CAD的关键所在. 所谓排料处理就是衣片在布料上如何合理放置,使得所耗用的布料长度最短,且被放置的各个衣片之间不存在任何覆盖现象. 以前这种排料工作都是由手工操作的,其操作过程是:先由排料放样人员在纸板上设计并剪下衣片纸样,然后尽可能紧凑、合理地排放在布料上,最后根据放样结果进行成批裁剪. 这种排料方法完全靠排料人员的经验,效率很低. 本文研究的目的是使排料人员免于裁剪大量纸样,而直接借助于我们设计的排料处理系统,在计算机屏幕上进行交互式排料. 我们的方法主要是通过移动或转动各个衣片,让其处于合适的位置,从而使所需的布料长度最短,并确保各衣片在布料上的经纬度和花纹走向正确. 为了获得比较满意的排料方案,本子系统还提供当前排料图的布料利用率.

1 排料处理

本子系统用C语言在ALR(80386)机上实现. 选用高分辨率的图形显示卡VGA,其分辨率为640×480,恰好与屏幕宽高比相同,因而图形不会变形,且效果较好. 下面给出排料处理子系统的模块结构,并着重讨论该子系统的算法实现.

1.1 子系统的模块结构

本子系统的总体结构如图1所示,可完成从给定排料布直到排料成功后存盘的一系列工作. 该子系统人机界面友好,具有较好的交互性能.

1.2 排料算法及其实现

算法定义了以下两个主要的数据结构形式,即

Struct pix

* 本文1993-11-31收到

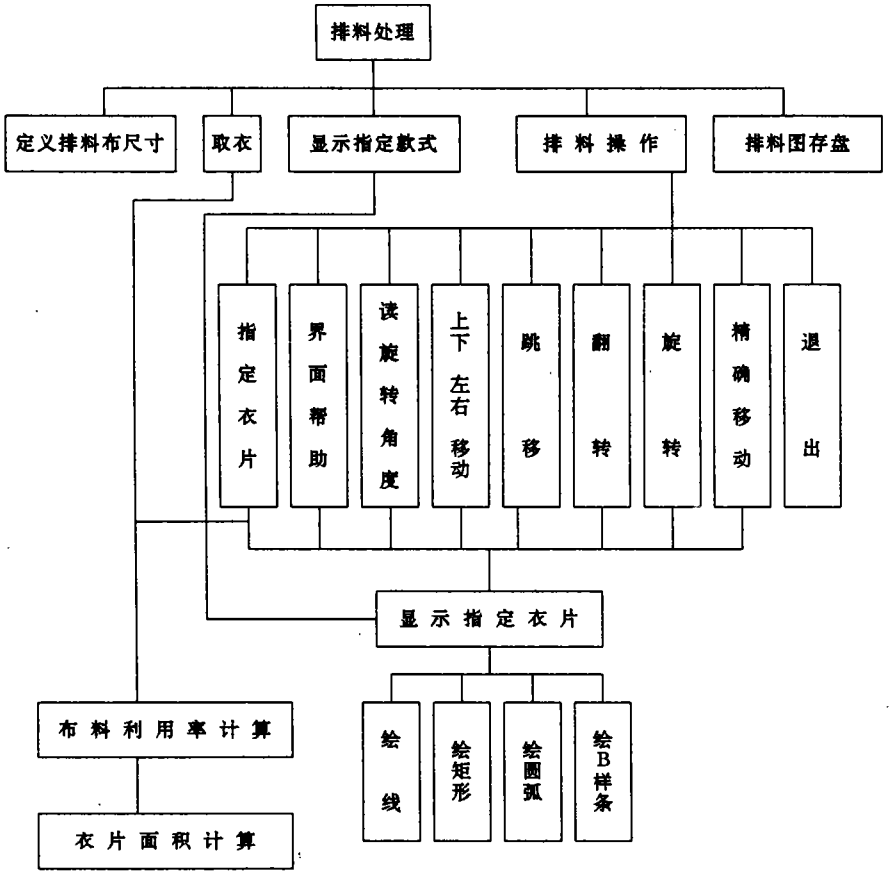

图1　排料子系统模块结构

{int x;　　int y;　　int arc;}

其中 x,y 是存放衣片存贮点的屏幕坐标,arc 是绘制类型,当它取值 1,2,3,4 时,分别表示绘制矩形、线段、圆弧、三次 B 样条;取值 0 和 −1 则分别表示绘制的分界点和衣片的结束点.

Struct pix point [$nn1$][$nn2$];

这是一个静态二维数组,用以存放指定款式中各个衣片的数据,这里 #define $nn1$ 7, #define $nn2$ 100. point[0][$nn2$], point[1][$nn2$], …, point[5][$nn2$], 是分别表示指定前衣片、后衣片、前袖片、后袖片、领片和口袋片. 下面我们给出排料模块中几个主要子算法的结构.

1.2.1　排料排版　在屏幕上显示一级菜单,按↑,↓键选择菜单项. 其算法结构为

```
P-process( )                          if (flag=0){display( );P−ption( );}}
{if"选中衣片排料"则                    if "排料图存盘"则 save( );否则退出函数,
  {define ( ); flag = get-coat        返回上层菜单}
( );
```

1.2.2　取衣　在屏幕上显示款式选择菜单,按→,←键选择菜单项,按回车键选中款式. 其算法结构为

get-coat( ) 否则 {Per=Cal-Per( )
{if"选中款式"则{读选中款式的各衣片文件. if (Per>1.0)提示"定义尺寸出错"并返
若读取不成功,则置取衣不成功标志 flag=0, 回 flag=0}}}

1.2.3 *排料操作* 根据读入功能键,调用相应的函数,完成相应的功能. 其算法结构为

P-ption( ) do{kk · v-key=bioskey(0);
{do{rr=get-pattern( ); 调用相应函数;
if (rr! =−1) }while(int(kk · v-key[1]! =ESC)}
{help( ); }while(rr! =−1);}

1.2.4 *指定欲操作衣片* 首先将指定衣片置成线红色,然后根据读入按键进行相应操作,其算法结构为

```
get-patter( )
  {draw-P(point[ss],LIGHTRED);
  do {kk · v-key=bioskey(0);
    若为"←",则{xx=ss;选择 xx 前面的一衣片并改变 ss 值;
              draw(point[xx],WHITE);
              draw(point[ss],LIGHTRED);}
    若为"→",则{xx=ss;选择 xx 后面的一衣片并改变 ss 值;
              draw-P(point[xx],WHITE);
              draw-P(point[ss],LIGHTRED);}
    若为"↓",则 draw-P(point[ss],LIGHTRED);
    若为"e",则{计算出排料图中所有衣片所处的最右坐标 erig 和最下坐标 ebot;计算
              所剩布料是否可再利用,若是,则剪下该布料并标上"可再利用"标志,
              重新计算布匹利用率,置退出循环标志,并返回"指定衣片不成功";}
    While(退出循环标志未置位)}}
```

1.2.5 *平移* 以每次移动 8 个象素距离,分为上、下、左、右四个方向移动,分别由功能键↑,↓,←,→完成.

1.2.6 *跳移* 可实现任意角度,任意步长的移动,分为左上、左下、右上、右下跳移,分别由功能键 F_1,F_2,F_3,F_4 完成.

1.2.7 *翻转* 分为上、下、左、右四个方向翻转,分别由功能键 F_5,F_6,F_7,F_8 完成.

1.2.8 *旋转* 根据给定的旋转角度 deter,绕布料中心点旋转,并判断是否超界[1].

下面我们对算法中用到的几个函数加以说明:(1)define( ),定义并调整排料布料尺寸;(2)display( ),显示排料图中各衣片;(3)save( ),将当前排料图存盘;(4)cal-per( ),计算布匹利用率.

1.3 布料利用率的计算

所谓布料利用率是指在一块布料上所有被利用的布料占该布料的百分率. 要计算布料利用率只需将各衣片的面积之和除以布料总面积即得. 我们假定布料是长方形的,长乘以宽即得其面积,因此关键是如何计算各个衣片的面积. 由于衣片的形状是不规则的,虽然我们已将衣片的曲线用直线段进行拟合,但简化后的衣片仍然存在凸多边形和凹多边形的情况,用计算

多边形面积的方法是不大适用的．考虑到象素点不同于几何点,象素点是有大小的,可计算其面积．这样,我们只要在多边形填充算法中增加一个计数器,对位于一个衣片内部的所有象素进行计数,就很容易计算出衣片的面积．设各衣片的面积总和为 sum,布料面积为 bs,则布料利用率为 sum/bs．目前已有多种行之有效的多边形填充算法,我们采用的是类似于多边形扫描转换中的 yx 算法．首先定义数据结构形式如下:

struct ed {int *y-top*,*y-bot*,*xa*;floot k;}edge〔*nn*2 + 1〕;

int *xa*〔*nn*2〕

其中 *y-top*,*y-bot* 分别为多边形中边的上端点和下端点 *y* 坐标;*xa*,*xa*〔*nn*2〕分别为存放多边形边和扫描线交点的 *x* 坐标以及经按升序排序的交点 *x* 坐标．

计算衣片面积的算法可简单描述如下:

```
area(  )                                  up date(end,& start,sca);
{sum=load-P(n1,s3,& n2);                  成对提取边和扫描线的交点,
if (n2<2)则提示"出错!"并退出此函数        点亮交点对内的所有象素并进行计数．
else{扫描线 sca=edge〔1〕. y-top+1;        sca++;
start=1;end=1;                            }while(此多边形中边未被全部扫描过)}
do{indude(& end,sca,n2);                  返回此衣片的象素点总数}
```

下面对算法中用到的几个函数作简要说明．

(1) load-P(　),用以将指定衣片的点信息转化为边信息,存入数组 *edge* 中,并按 *y-top* 从小到大对数组中 *y* 坐标进行排序,返回多边形中非水平边的数目 n_2．(2) indude(　),检查该扫描线是否存在不需要考虑的边,计算剩余边与扫描线交点,并将交点 *x* 坐标按升序排序配对存于数组 *xa* 中．

这里有一点需要指出,当扫描线恰交于衣片多边形的极值点(局部最高或最低的顶点),交点需记录两次．如图 2 所示,顶点 1,2,3,5 为极值点,而 4,6 则不是．

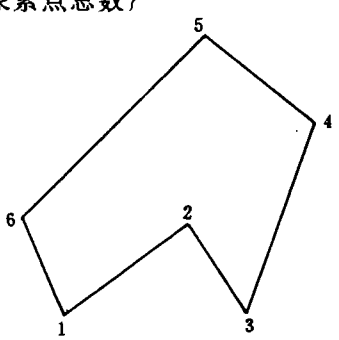

图 2　极值点示意图

2　自动排料

2.1　自动排料的基本思想

前面所述的排料处理方法比传统的手工排料先进许多,但它仍需排料人员在屏幕上进行交互式排料．其操作过程类似于手工排料．为摆脱排料过程中过多的人为干预,我们提出了实现计算机自动排料的改进方案．排料人员只需给定排料布尺寸及各个衣片数据,计算机便可自动进行处理,并输出排料图．我们知道,一个衣片实际上是一个多边形,由于自动排料时,处理过程对排料人员是不透明的,无法防止各个衣片之间可能出现的相交复盖等情况．因此,实现自动排料的关键是求出任意两个多边形的相交多边形．

2.2　自动排料算法设计的考虑和步骤[2]

由上可知,求解任意两个多边形的交集是实现自动排料的关键．由于两个相交多边形其凹凸性不能确定,甚至两个多边形的交集还可能形成若干个离散的多边形,它们的凹凸性也无法确定,如图 3 所示．对于两个凸多边形的相交多边形,可用平面—扫描算法求解．但大多数

的衣片模型都是凹多边形,已有的算法不大适用.我们可用如下方法来求解任意两个多边形的相交多边形.

2.2.1 *分析点的类型* 作为结果多边形中的顶点有3种不同类型:(1)点a是P_1的顶点,同时也在P_2中;(2)点b是P_2的顶点,同时也在P_1中.(3)点c是P_1中的一条边形P_2中的一条边的交点.其它一些点是一个多边形的顶点但不在另一个多边形中,它们不属于任何相交部分:点d是P_1的顶点但不在P_2中;点e是P_2的顶点但不在P_1中.

在图3所示的例子中的顶点具有如下类型:a类点:1,4,8,9;b类点:18,21;c类点:2,3,5,7,10,13;d类点:6,11,12;e类点:14,15,16,17,19,20.

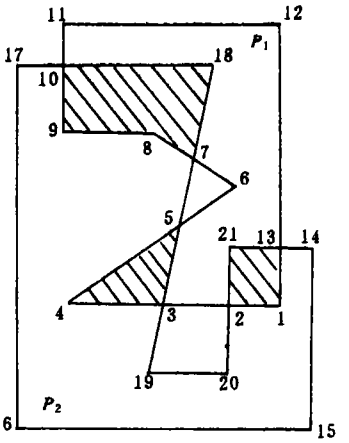

图3 两个多边形的交集

2.2.2 *标识点相对于多边形的位置* 为了将这些点分类,需要标识出点(x_0,y_0)是在多边形P的内部还是外部(多边形边界上的点认为在其内部).这个问题在文[1]中已经清楚地说明并解决了.

2.2.3 *形成数据结构* 在分出点类型a,b,c后,整个问题就可以变成将这些点按一致的方法进行排序,以建立多边形P_1和P_2交集的数据结构.

2.2.4 *计算相交多边形面积* 当所有结果多边形的数据结构已建立在计算机主存时,可用下面公式计算含有n个顶点$((x_i,y_i))$的相交多边形面积,即

$$0.5 * \mathrm{abs}(\sum_{i=0}^{n-1}(x_{i-1}-x_i)*(y_{i-1}+y_i))$$

$$= 0.5 * \mathrm{abs}(\sum_{i=0}^{n-1}x_i*(y_{i-1}-y_{i+1})).$$

当然,在算法设计和实现时还有许多细节需要考虑和处理,这里就不再深入讨论了.

参 考 文 献

1 罗杰斯D F著.计算机图形学的算法基础.梁友栋等译.北京:科学出版社,1987.73~101

2 Mangen A,Lasudry N. Search for the intersection polygon of any two polygons:application to the garment industry. Computer Graphics Forum,1991,(10):195~208

Material Layout in Dress CAD

Zhang Quanhuo Fang Ronghui Fan Huilin

(Dept. of Computer Science,Huaqiao Univ.,362011,Quanzhou)

Abstract With respect to the layout of polygonal dress materials in computer-aided design of dress,the authors set forth its basic idea;and gives the algorithm which can be implemented by C language;and put forward the key to realize automatic layout of dress material and main steps to design the algorithm.

Keywords computer aided design,garments,layout of dress material,algorithms,polygons