

# 计算机辅助光学作图系统中光路计算模型<sup>\*</sup>

牛彦敏

( 重庆师范大学 物理学与信息技术学院, 重庆 400047 )

**摘 要** 利用计算机来辅助光学作图具有现实意义, 其核心在于如何实现光路自动计算。该文利用面向对象分析方法构造出对象关联模型, 建立了一种光路计算的关键算法模型, 它由基本算法模型和多光学零件的算法模型组成。在此基础上进而研究开发了计算机辅助光学作图系统。文中还以单球面镜为例对该算法进行了较为详细的阐述。

**关键词** 光路计算; 算法; 对象关联模型; 计算机辅助光学作图; 单球面镜

中图分类号: TP921

文献标识码: A

文章编号: 1672-6693( 2005 )02-0034-05

## Calculation Model of Ray in Computer Aided Optical Chart System

NIU Yan-min

( College of Physics and Information , Chongqing Normal University , Chongqing 400047 , China )

**Abstract** The most important thing is how to calculate light path automatically in computer aided optical chart system. The article firstly constructs the calculation model by object-oriented method, and then presents basic part and multiple optical part algorithms. They are demonstrated based on single spherical lens. On this basis a computer aided optical chart system is developed.

**Key words** ray calculation; algorithm; object-oriented method; computer aided optical chart system; single spherical lens

在光学镜头设计中, 尽管设计者也假设简单的数学模型来计算通过镜头的能量, 从而追踪某些光线的路径, 但主要还是通过几何方法计算通过镜头的光线。对于近轴的光线, 高斯成像是相当精确的<sup>[1]</sup>, 其计算公式简单, 但常常要求精确到小数点后 6~8 位。对于一个有 6 片镜片的镜头设计, 每个镜片的表面需要计算 200 条光路, 整个镜头的计算量达到了 3 000 条光路, 在个人手工计算的条件下, 即便不出错误也至少需要 3 个月才能完成全部计算, 而哪怕有一个错误, 也容易导致前功尽弃。如果能够借助计算机代替人工进行这些枯燥的计算, 不仅可以提高效率, 杜绝错误, 而且能够直观地展示镜片设计的效果, 便于修改。对于学习光学的人来说, 缺乏形象生动的光路示例是导致学习厌倦效率低下的重要原因之一。因此利用计算机辅助进行光学作图具有切实的重要意义。基于这种考虑, 本文利用面向对象分析方法构造出对象关联模型, 建立了一

种光路计算的关键算法模型, 在此基础上进而研究开发了计算机辅助光学作图系统。

### 1 系统需求分析与对象模型

计算机辅助光学作图系统作为一种特殊的绘图软件系统, 主要依据用户构造的光学元件和入射光线, 然后应用几何光学的基本原理, 计算出每条入射光线的出射光线, 最后以一定方式直观地显示在计算机或其它输出设备上。它对用户交互性输入的要求并不高, 但是对图形显示有如下要求。(1)能够自动计算光路, 结果与相关理论相吻合;(2)体现光路特点, 能够以图形显示, 遍布整个幅面, 直观形象;(3)支持缩放功能, 用户既可查找局部交点位置, 又可全局观看。

其中要求(2)、(3)可通过合理选择开发工具来满足要求(1)则需要设计算法。经过分析, 系统选用 Visual C++ 6.0 作为开发平台, 这是目前最为复

\* 收稿日期 2004-11-16 修回日期 2005-02-22

作者简介: 牛彦敏(1975-), 女, 河南许昌人, 讲师, 硕士, 主要研究方向为 PCI 数据采集卡、数字图像。

杂的开发平台,它通过使用 MFC 类库<sup>[2]</sup>,将数据和数据的显示分离,可以方便地进行图形的可视化显示缩放等操作。而它所采用的 C++ 编译器则可保证算法计算的快速性。但 VC 是基于对象来解决问题

的,为此利用面向对象的分析设计方法<sup>[3,4]</sup>设计了系统的对象模型(如图 1)。图中将所有光学零件的父类抽象为 COptPart,方便了系统的扩充。

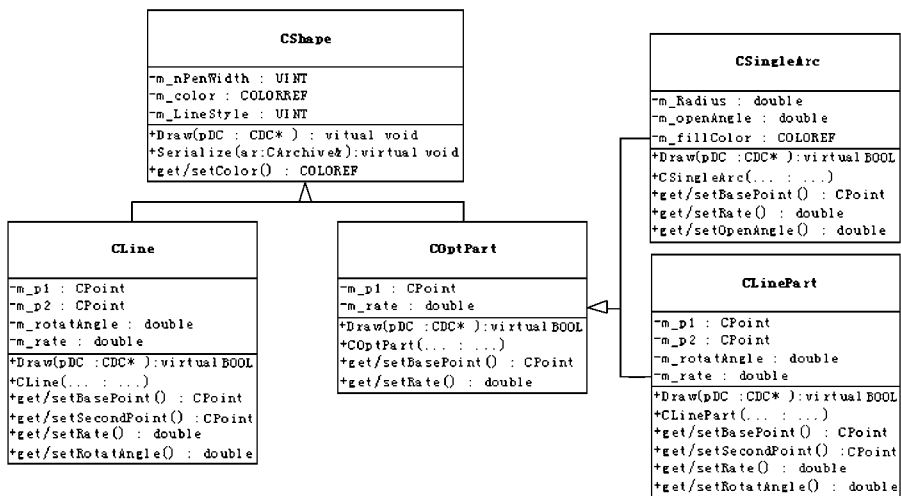


图 1 系统对象模型

## 2 光路计算算法

### 2.1 基本算法模型

在几何光学中,常用的光学零件有反射镜、透镜、单球面镜、棱镜、平行平板等,经过组合之后对光路的求解变得非常复杂,对于共轴光学系统可以逐步求解,而对非共轴光学系统有时甚至只能够手工计算。但是无论光学零件组有多么复杂,都能分解为基本模型的组合。该基本模型可描述为:一条入射光线经过一个光学零件之后,确定出一条出射光线(折射、反射)。图 2 示意了抽象的光路计算基本模型。本文以单球面镜进行示例说明。实际进行绘图算法设计时,要将光线转换为可绘图的线段来表示,根据入射光线光源位置、光线折射与否、折射程度,需考虑图 3 所示的 9 种情况分别进行判定。对于入射角、折射角、沿轴线段等符号的约定非常重要,应注意保持一致。此外还需要利用 MFC 来设置坐标系,确定映射模式,以方便屏幕绘图尺寸和物理尺寸的吻合。

基本算法的算法名称为 LineSingleArcJ,算法说明:根据输入光线和光学零件的关系,自适应地确定是否折射、折射类型、折射交点和折射角并完善输入光线,输入参数为入射光指针 pLine 及光学零件(单球面镜)pSa,输出参数为折射交点 interPoint 和折射角 outUp,函数返回形式为 BOOL 类型。

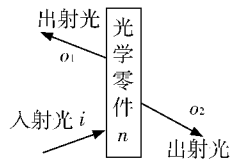
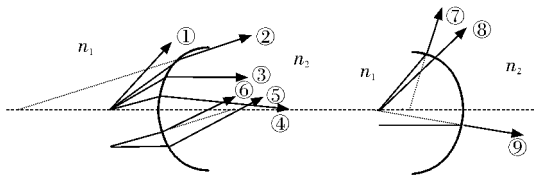


图 2 光路计算基本模型



注:若入射光原点不在子午面要求与子午面的交点

图 3 光线通过单球面镜后折射情况分类

算法的 VC 代码描述如下。

```
BOOL LineSingleArcJ( CLine * pLine ,CSingleArc
* pSa ,CPoint& interPoint ,double& outUp ,... ){
    if( fabs( oy0 - cy0 ) > delta && fabs
( U ) < delta ) {
        //如果入射光点不在子午面上,
        且不是平行光,要转成子午面上
        ox0 = ox0 + ( cy0 - oy0 )/tan( U );
        oy0 = cy0 ;}
    L = ( - 1 ) * ( cx0 - R - ox0 ) //保证
    L < 0
    if( fabs( U ) > fabs( maxInterAngle +
delta ) ){
```

AfxMessageBox(“ 入射角大于单

```
球面镜近轴临界角 ,不折射!") ;return FALSE ; }  
    if( fabs( U ) < delta && fabs( oy0 -  
cy0 ) > fabs( R * cos( openAngle * 0.5 ) + delta )) {  
        AfxMessageBox( " 平行光不经过  
单球面镜 ,不折射 !" ) ;return FALSE ;}  
    if( fabs( U ) = < delta ) { //平行光  
        sinI =( oy0 - cy0 ) / R ;I = asin( si-  
nI ) ;sinIp = sinI * N / Np ;Ip = asin( sinIp ) ;  
        fai = I ;Up = Ip - I ;Lp = R * ( 1 +  
sinIp / sin( Up ) ) ;  
        interX = cx0 - R * cos( I ) ;interY  
= oy0 ;X =( int )interX ;Y =( int )interY ;  
        if( fabs( interX - X ) > = 0.5 ) X  
+ + ;  
        pLine - > setSecondPoint( CPoint  
( X , Y ) ) ; //修改入射光第二点  
    } else {  
        sinI =( L - R ) * sin( U ) / R ;sinIp  
= N * sinI / Np ;Ip = asin( sinIp ) ;  
        I = asin( sinI ) ;fai = U + I ;Up = U  
+ I - Ip ;
```

```
        if( fabs( Up ) < delta ) Lp = R + R  
* 100000 //折射光平行子午面  
        else Lp = R + R * Ip / Up ;  
        interX = cx0 - R * cos( fai ) ;interY  
= cy0 - R * sin( fai ) ;X =( int )interX ;Y =( int )interY ;  
    } //返回值  
    interPoint = CPoint( X , Y ) ;inI = I ;out-  
Ip = Ip ;outFai = fai ,outUp = Up ,outLp = Lp ;return  
TRUE ; }
```

2.2 多光学零件的算法模型

图 4 所示的是一种“多镜多线”的情形。系统中有多个单球面镜 ,多条入射光线和多条折射光线 ,经过分析 ,它有 4 个基本特征<sup>[5]</sup> ( 1 )单球面镜、入射线的数目无法预先规定 ,但当两者数目被决定后 ,折射光线数目被唯一确定 ( 2 )前一个单球面镜的折射光线即成为下一个单球面镜的入射光线 ( 3 )折射光线依附入射光线而存在 ,可以用基本算法确定 ; ( 4 )入射光线、单球面镜、折射光线三者严格顺序匹配。

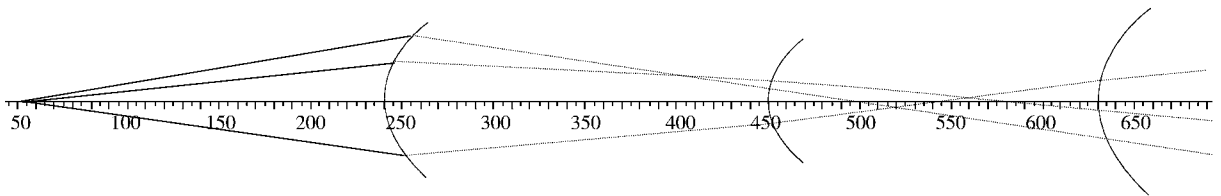


图 4 多条入射光线通过多个单球面镜

这类模型可统一描述为 ,每一条入射光线经过一个单球面镜处理产生一条折射光线 ,该折射光线依次经过所有单球面镜处理产生该入射光线的所有折射光线 ,入射光线、单球面镜由输入决定。显然 ,利用数组等数据结构来存储这 3 类对象是不可行

的 ,可以考虑分别建立 3 个对象链表来组织。但是折射光线依附于每条入射光线 ,如果将其独立出来 ,就很难计算链表中的每个对象了。经过分析 ,发现可以将折射光线链表的首指针放在入射光线链表的每个结点中 ,从而形成了图 5 所示的算法模型。

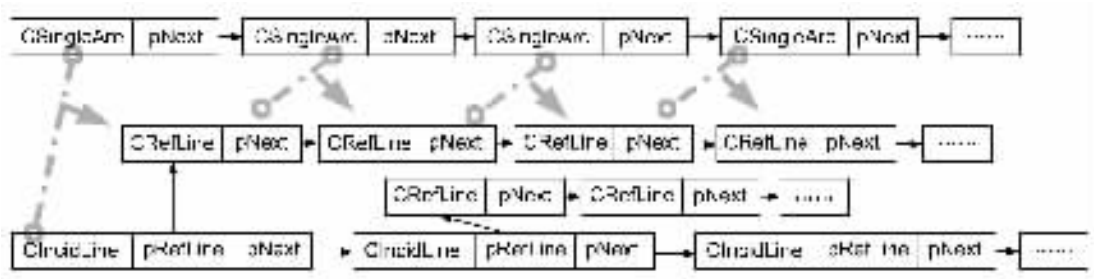


图 5 多单球面镜系统光路算法模型

图 5 中示意了一条入射光线经过单球面镜后的折射光线计算过程 ,三向虚线箭头可理解为圆圈两

端的对象作为参数送入基本模型算法 LineSingleArcJ 中 ,得到出射光线 ,根据出射光线建立新结

点。对每条入射光线遍历该过程,就可以自动计算出所有光路。如果光学零件光路性质有所改变,只需修改 LineSingleArcJ 算法即可,而无须改变本算法和其它部分,这样系统就能够适用于其它光学零件的光路计算,因此图5所建立的算法模型具有很强的通用性。

入射光线链表、单球面镜链表应采用如下定义。

```
CTypedPtrList < CObList ,CSingleArc * > m_Sin-
gleArcList ;
```

```
CTypedPtrList < CObList ,CLine * > m_In-
cidList ;
```

其中入射光线类定义中,要包含折射光线链表。

```
class CLine :public CShape{
```

```
...
```

```
CLine( );
```

```
CLine( CPoint p1 ,CPoint p2 ,UINT line-
width ,UINT linestyle ,UINT linecolor ,double rate );
```

```
CLine( CPoint p1 ,double rotatAngle ,UINT
lintwidth ,UINT linestyle ,UINT linecolor ,double rate );
```

```
CTypedPtrList < CObList ,CLine * > m_Re-
frList ;
```

```
...};
```

关于多光学零件的自动光路计算算法模型的描述如下。

算法名称为 Lines2SingleArcsJ;算法说明是根据多条输入光线和多个光学零件的关系,对于每一条入射光线自适应地确定是否折射、折射类型、所有的折射交点和折射角并完善该入射光线,构造该输入光线的所有折射光线;输入参数为入射光指针链表 m\_IncidList;光学零件链表(单球面镜)m\_SingleArcList;输出参数为所有的折射线链表 m\_RefrList(逐一包括了折射交点 interPoint 和折射角 outUp);函数返回形式为 BOOL 类型。

算法的 VC 代码描述如下。

```
posIncid = m_IncidList. GetHeadPosition( );
```

```
while( posIncid ){//入射线不为空
```

```
IncidLine = m_IncidList. GetNext( posIn-
cid );
```

```
curIncidLine = IncidLine ;
```

```
posSArc = m_SingleArcList. GetHeadPosition
( );
```

```
while( posSArc ){//单球面镜不为空
```

```
SArc = m_SingleArcList. GetNext( pos-
SArc );
```

```
LineSingleArcJ( curIncidLine ,SArc ,in-
terPoint ,inI ,outIp ,outFai ,outUp ) ;//调用基本模型
算法
```

```
curRefr = new CLine( interPoint ,outUp *
180/PI ,//构造折射线
```

```
0.5 ,NULL ,curIncidLine - >
getColor( ) ,SArc - > getRate( ) );
```

```
Xp = interPoint. x + 2000 * cos( outUp );
```

```
XXp =( int )Xp ;
```

```
Yp = interPoint. y + 2000 * sin( outUp );
```

```
YYp =( int )Yp ;
```

```
curRefr - > setSecondPoint ( CPoint
( XXp ,YYp ) ) ;//设置第二点
```

```
if( IncidLine - > m_RefrList. IsEmpty( ) )
IncidLine - > setSecondPoint( inter-
Point );
```

```
else{
```

```
CLine * pPrevious = IncidLine - > m_
RefrList. GetTail( );
```

```
pPrevious - > setSecondPoint( cur-
Refr - > getBasePoint( ) ) ;//修改前一节点的第二点
}
```

```
IncidLine - > m_RefrList. AddTail( cur-
Refr ) ;//加入折射线到链表中
```

```
curIncidLine = curRefr ;//将折射线改
为入射线,继续遍历计算
```

```
}
```

需要注意的是,由于入射光线经过单球面镜后,是一个由直线到线段再到直线的变化过程,而为了便于计算,程序中的入射光线和折射光线都是采用点斜式的直线表示法,为此要注意在适当的时候修改第二点使之成为线段,否则难以绘图。此外系统的角度和长度符号必须统一规定成矢量。图6给出了该算法在系统中的使用情况。

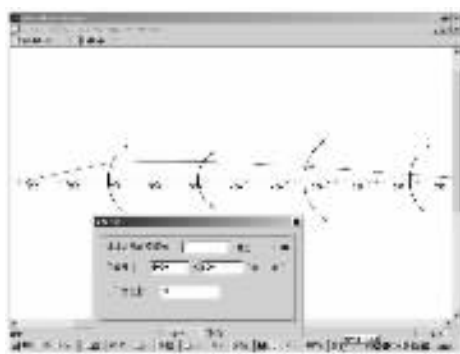


图6 计算机辅助几何光学作图平台主界面

3 结 论

本文结合几何光学的原理 ,利用面向对象分析方法建立了计算机辅助光学作图系统的对象关联模型 ,通过分析光路特点 ,对其中的关键问题——光路计算提出了通用的算法。该算法由基本算法模型和多零件的算法模型组成 ,文中还以单球面镜为例进行了对该算法较为详细的阐述。利用本文算法所设计的系统已经在重庆师范大学重点课程《工程光学》的教学演示及结果校核中发挥了积极作用。

参考文献 :

[ 1 ] 郁道银. 工程光学[ M ]. 北京 :机械工业出版社 ,1999.

[ 2 ] 付俊明 ,严东明 ,张双民. Visual C + + 6. 0MFC 类库参考手册[ M ]. 北京 :人民邮电出版社 ,2002.

[ 3 ] LEE R C ,TEPFENHAR W M. UML and C + + :A practical Guide to Object-Oriented Development[ M ]. 北京 :机械工业出版社 ,2002.

[ 4 ] WAMPLER B E. Java 与 UML 面向对象程序设计[ M ]. 北京 :人民邮电出版社 ,2002.

[ 5 ] 梁一平 ,戴特力 ,熊玲玲. 双曲线和椭圆焦点的光学意义 [ J ]. 重庆师范大学学报( 自然科学版 ) ,2004 ,21( 3 ) : 26-28.

( 责任编辑 黄 颖 )