



OPTOTEK

ZEMAX非序列照明设计实例教程

ZEMAX Examples for Non-Sequential Illumination Design



南京光科信息技术有限公司

NANJING OPTOTEK TECHNOLOGY CO., LTD.

目录

第一章 非序列简介	2
1、非序列物体坐标参考的应用	2
2、非序列相对坐标与全局坐标的转换	9
3、ZEMAX 完整非序列照明系统	16
4、序列、非序列与混合序列的区别与联系	25
第二章 非序列物体建模	31
5、非序列基本物体参考介绍	31
6、ZEMAX 中 CAD 物体的导入与导出	37
7、ZEMAX 布尔运算创建复杂模型	41
8、多面体物体 POB 建模方法	52
9、非序列分段小面反光杯 TOB 建模方法	61
10、非序列中旋转扫描物体建模方法	72
11、非序列中挤压物体的建模方法	76
12、菲涅尔透镜的建模及优化方法	83
13、DMD 投影数字微光片 MEMS 的建模方法	93
第三章 光源模型	98
14、ZEMAX 中如何输入光源配光曲线	98
15、ZEMAX 中如何导入 IES 光源和 LDT 光源	101
16、LED 芯片朗伯型发光简易光源的创建方法	105
17、激光二极管阵列光源的创建方法	109
18、体光源如环形直线或螺旋型灯管创建方法	112
第四章 非序列系统设计及优化	117
19、非序列元件膜层和散射特性的添加方法	117
20、ZEMAX 中分光/滤光镜的模拟方法	124
21、非序列中干涉系统模型方法	137
22、微透镜复眼阵列超均匀光斑设计	147
23、非序列中验证镜头成像效果	153
24、ZEMAX 非序列照明的颜色探测方法	164
25、如何在非序列模式下设计离轴反射系统	171
第五章 LED 照明系统设计	185
26、ZEMAX 中 LED 准直透镜的优化-实现方形光斑	185
27、LED CPC 射灯设计方法	197
28、ZEMAX 中 LED 反光杯的优化设计	203
29、LED 近场光源与远场光源区别	209

目录

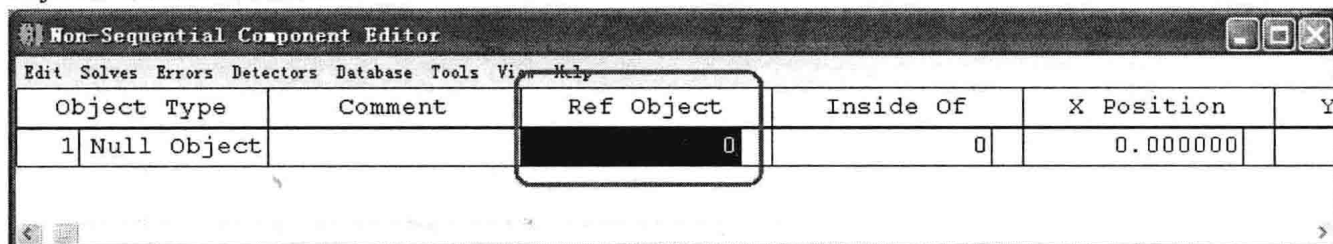
第一章 非序列简介	2
1、非序列物体坐标参考的应用	2
2、非序列相对坐标与全局坐标的转换	9
3、ZEMAX 完整非序列照明系统	16
4、序列、非序列与混合序列的区别与联系	25
第二章 非序列物体建模	31
5、非序列基本物体参考介绍	31
6、ZEMAX 中 CAD 物体的导入与导出	37
7、ZEMAX 布尔运算创建复杂模型	41
8、多面体物体 POB 建模方法	52
9、非序列分段小面反光杯 TOB 建模方法	61
10、非序列中旋转扫描物体建模方法	72
11、非序列中挤压物体的建模方法	76
12、菲涅尔透镜的建模及优化方法	83
13、DMD 投影数字微光片 MEMS 的建模方法	93
第三章 光源模型	98
14、ZEMAX 中如何输入光源配光曲线	98
15、ZEMAX 中如何导入 IES 光源和 LDT 光源	101
16、LED 芯片朗伯型发光简易光源的创建方法	105
17、激光二极管阵列光源的创建方法	109
18、体光源如环形直线或螺旋型灯管创建方法	112
第四章 非序列系统设计及优化	117
19、非序列元件膜层和散射特性的添加方法	117
20、ZEMAX 中分光/滤光镜的模拟方法	124
21、非序列中干涉系统模型方法	137
22、微透镜复眼阵列超均匀光斑设计	147
23、非序列中验证镜头成像效果	153
24、ZEMAX 非序列照明的颜色探测方法	164
25、如何在非序列模式下设计离轴反射系统	171
第五章 LED 照明系统设计	185
26、ZEMAX 中 LED 准直透镜的优化-实现方形光斑	185
27、LED CPC 射灯设计方法	197
28、ZEMAX 中 LED 反光杯的优化设计	203
29、LED 近场光源与远场光源区别	209

第一章 非序列简介

1、非序列物体坐标参考的应用

前言

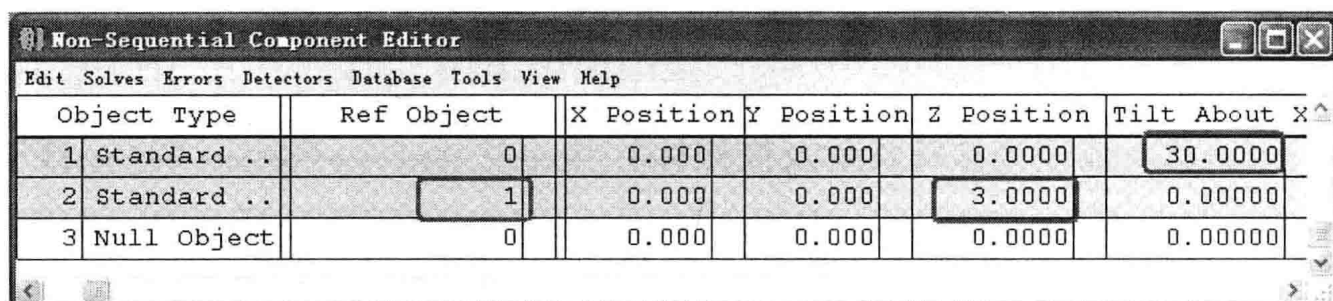
我们在非序列离轴系统设计的例子中已经初步讨论过几种坐标的使用方法，完全掌握 ZEMAX 非序列坐标系统及坐标参考方法，有助于我们快速完成元件建模或系统元件摆放。本篇文章我们主要讲解 Ref Object 参考物体的使用方法。



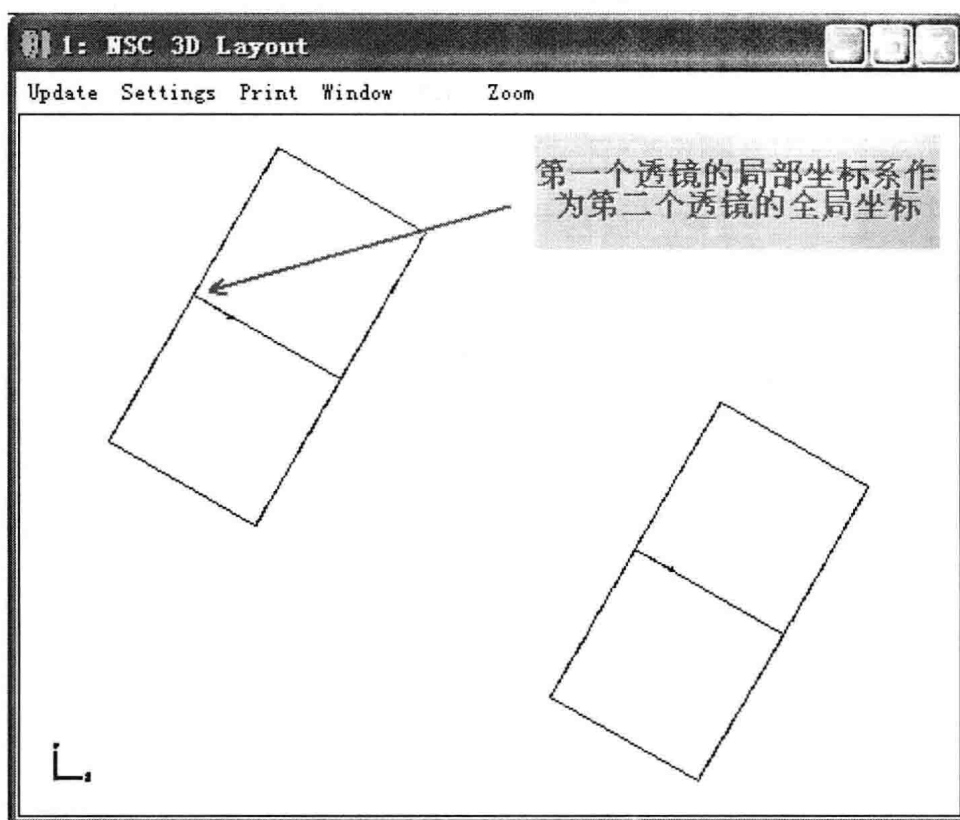
Object Type	Comment	Ref Object	Inside Of	X Position	Y
1	Null Object	0	0	0.000000	

Ref Object 称为参考物体，它确定了非序列全局坐标与局部坐标的转换，在使用时，被参考的物体在编辑器中永远位于参考物体之后。在参考时它有两种用法：绝对参考和相对参考。

绝对参考：即在 Ref Object 参数栏内直接输入要参考物体的物体序号即可（即在编辑器上的物体编号），这样被参考物体便以参考物体的局部坐标中心为全局坐标中心，如下图中两个透镜，第一个透镜旋转了 30 度，第二个透镜参考第一个透镜，虽然没有旋转，但仍以第一个透镜旋转之后的坐标作为全局坐标：



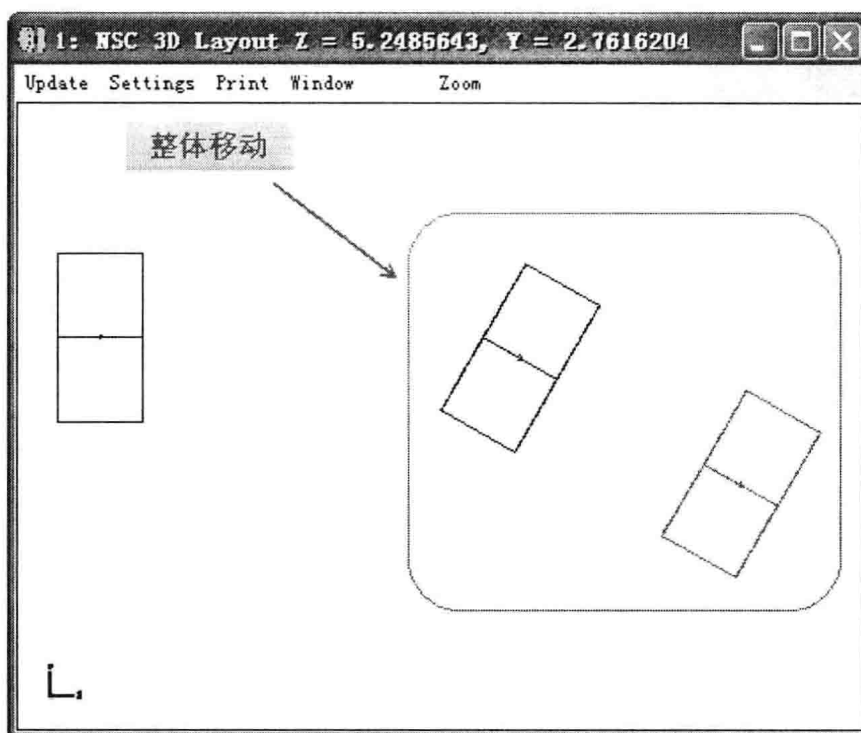
Object Type	Ref Object	X Position	Y Position	Z Position	Tilt About X
1 Standard ..	0	0.000	0.000	0.0000	30.0000
2 Standard ..	1	0.000	0.000	3.0000	0.00000
3 Null Object	0	0.000	0.000	0.0000	0.00000



有了这个功能，我们在设计元件较多的复杂系统时，只需知道某些元件之间的关联性，就可以轻松摆放它们，而不需要在整个坐标系统中去计算。通常可使用参考物体特性来连接某组元件，如三片透镜组成的镜头组，这样在移动时便可保证这组元件的整体移动。

如下图对之前的两个透镜移动，只需移动参考物体：

Object	Type	Ref Object	X Position	Y Position	Z Position	Tilt About X
1	Standard ..	0	0.000	0.000	0.0000	0.00000
2	Standard ..	0	0.000	0.000	5.0000	30.0000
3	Standard ..	2	0.000	0.000	3.0000	0.00000
4	...	0	0.000	0.000	0.0000	0.00000

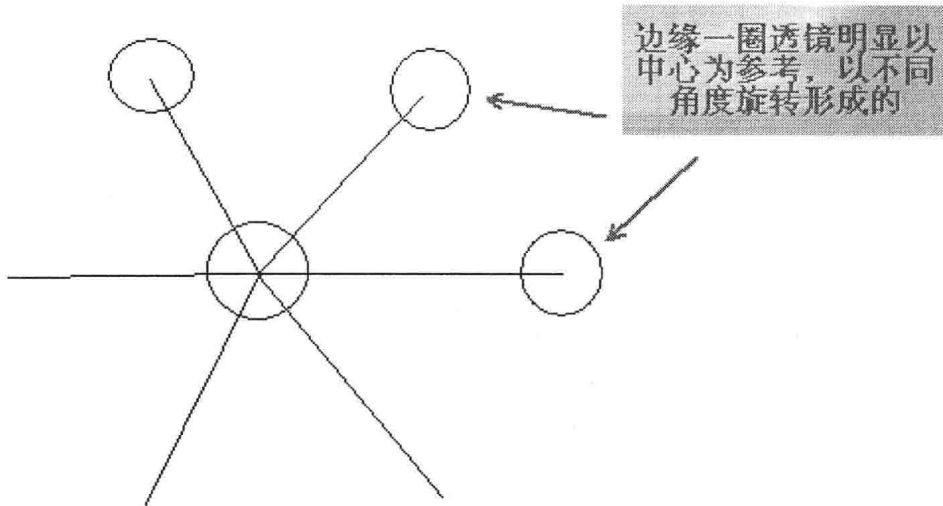


如果我们对上图中的这两个透镜组进行复制，同样让他们保持一个整体，此时问题就来了。由于使用的是绝对参考，复制后所有参考物体的序号和复制前的一模一样，这就违背了我们的想法。当然可以手动去修改参考序号，但如果复制的这组物体很多很复杂时，手动修改却消耗大量时间。这时相对参考便会体现出它的优势：

Object Type	Ref Object	X Position	Y Position	Z Position	Tilt About X
1 Standard ..	0	0.000	0.000	0.0000	0.00000
2 Standard ..	0			5.0000	30.0000
3 Standard ..	2			3.0000	0.00000
4 Null Object	0			0.0000	0.00000
5 Standard ..	0	0.000	0.000	5.0000	30.0000
6 Standard ..	2	0.000	0.000	3.0000	0.00000
7 Null Object	0	0.000	0.000	0.0000	0.00000

相对参考：即在 Ref Object 栏中输入一个负数，这个负数需小于当前物体序号，并且负数与当前序号之和即为所参考的物体号。在上图中，只需在物体 6 的 Ref 栏内输入 -1 即表示参考物体 5。相对参考比绝对参考更灵活，可适用于所有参考情况，但绝对参考较为直观，对于初学者来说更好理解。

到这里相信你对坐标参考有了更深层次的了解，接下来我们做个实例练习，假设我们要做一个透镜的六边形环阵列，六边形外接圆半径为 5mm，一共七个透镜（中心一个边缘六个）。设计简图如下：



打开 ZEMAX 并切至非序列模式，首先定义中心的透镜，其它透镜由中心透镜复制产生。材料为 BK7，半口径为 2mm，外边缘透镜由于是参考中心进行的旋转，相当于在参考物体上就已经发生了旋转，然后透镜以旋转后的坐标系作为全局坐标系统进行放置。很显然中心的参考点只是虚拟的，并没有实际物体，通常我们使用空物体（Null Object）来做参考物体：

Non-Sequential Component Editor							
Edit Solves Errors Detectors Database Tools View Help							
Object Type	Comment	Ref Object	X Position	Y Position	Z Posit..	Tilt About Z	
1 Standard ..	中心透镜	0	0.0000	0.000	0.00	0.000000	
2 Null Object		0	0.0000	0.000	0.00	0.000000	
3 Standard ..		-1	5.0000	0.000	0.00	0.000000	
4 Null Object		0	0	0	0.00	0.000000	
5 Null Object		0	0	0	0.00	0.000000	

作为参考物体的空物体
由于需要复制，使用相对参考

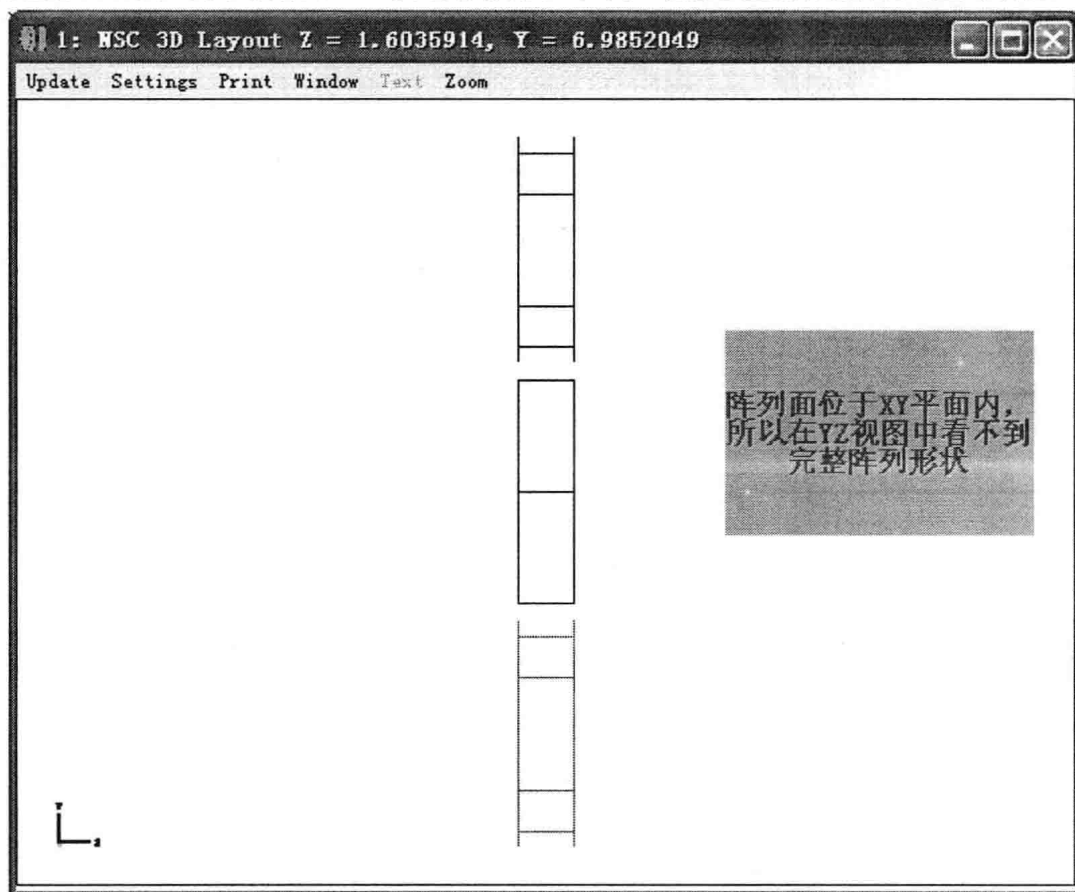
由于阵列面位于 XY 平面，则边缘透镜通过以中心参考绕 Z 轴旋转完成，6 个透镜的旋转角度分别为 (0, 60, 120, -60, -120, 180)

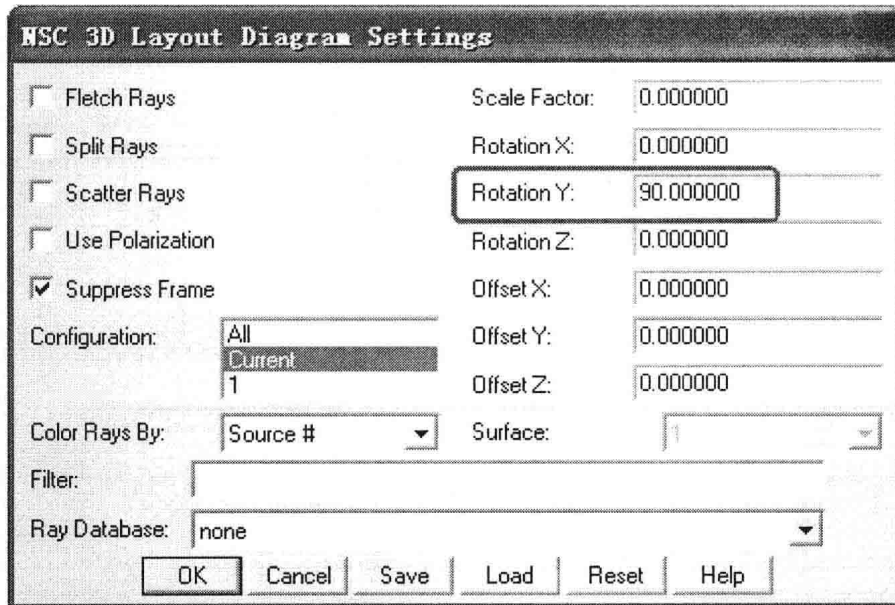
Non-Sequential Component Editor							
Edit Solves Errors Detectors Database Tools View Help							
Object Type	Comment	Ref Object	X Position	Y Position	Z Posit..	Tilt About Z	
1 Standard ..	中心透镜	0	0.0000	0.000	0.00	0.000000	
2 Null Object		0	0.0000	0.000	0.00	0.000000	
3 Standard ..		-1	5.0000	0.000	0.00	0.000000	
4 Null Object		0	0.0000	0.000	0.00	0.000000	
5 Null Object		0	0.0000	0.000	0.00	0.000000	

选中物体2和3，复制5次即可

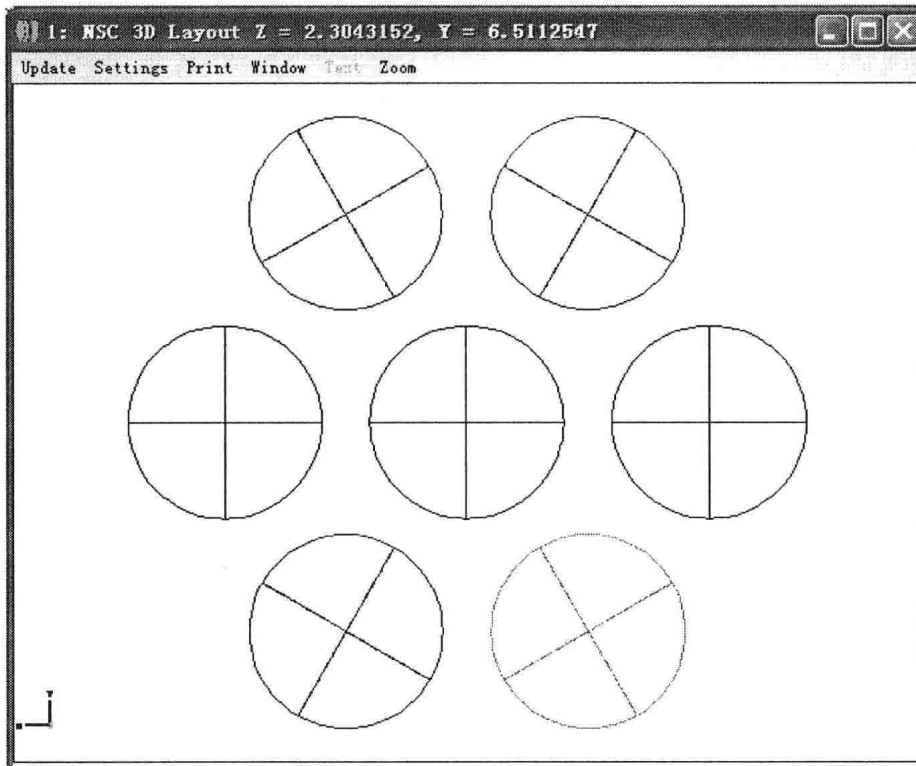
Non-Sequential Component Editor							
Edit Solves Errors Detectors Database Tools View Help							
Object Type	Comment	Ref Object	X Position	Y Position	Z Posit..	Tilt About	
1 Standard ..	中心透镜	0	0.0000	0.000	0.00	0.000000	
2 Null Object		0	0.0000	0.000	0.00	0.000000	
3 Standard ..		-1	5.0000	0.000	0.00	0.000000	
4 Null Object		0	0.0000	0.000	0.00	60.000000	
5 Standard ..		-1	5.0000	0.000	0.00	0.000000	
6 Null Object		0	0.0000	0.000	0.00	120.000000	
7 Standard ..		-1	5.0000	0.000	0.00	0.000000	
8 Null Object		0	0.0000	0.000	0.00	180.000000	
9 Standard ..		-1	5.0000	0.000	0.00	0.000000	
10 Null Object		0	0.0000	0.000	0.00	-60.000000	
11 Standard ..		-1	5.0000	0.000	0.00	0.000000	
12 Null Object		0	0.0000	0.000	0.00	-120.000000	
13 Standard ..		-1	5.0000	0.000	0.00	0.000000	

此时查看 L3n 图，为了能看到 XY 平面内的阵列，在图上任意位置点右键打开设置对话框：

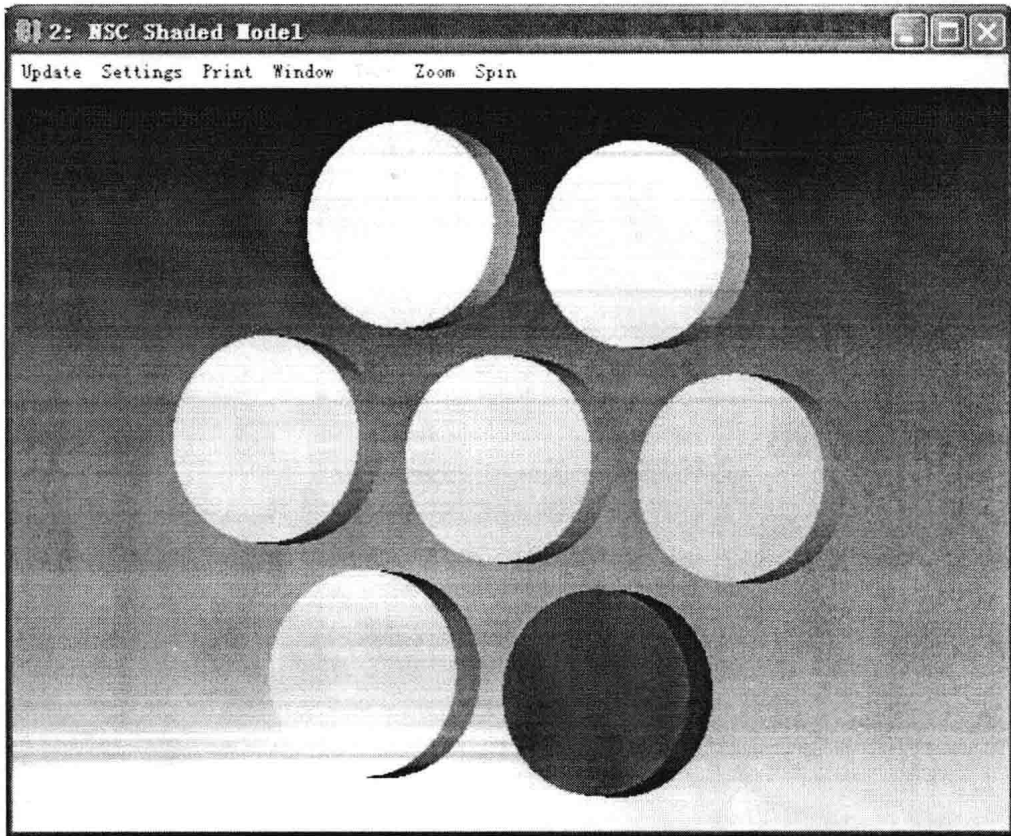




视图绕 Y 轴旋转 90 度，将转换到 XY 平面，看到阵列形状：



查看 3D 阴影模型：



总结：

本篇文章讲解了参考物体的两种使用方法，相对参考与绝对参考。通过对这两种参考的实例练习，我们看到它对于非序列物体建模的灵活性和实用性。希望在今后的成像及照明设计中能够不断发掘它们的强大用处，发挥它们对坐标系统修改的潜力。

2、非序列相对坐标与全局坐标的转换

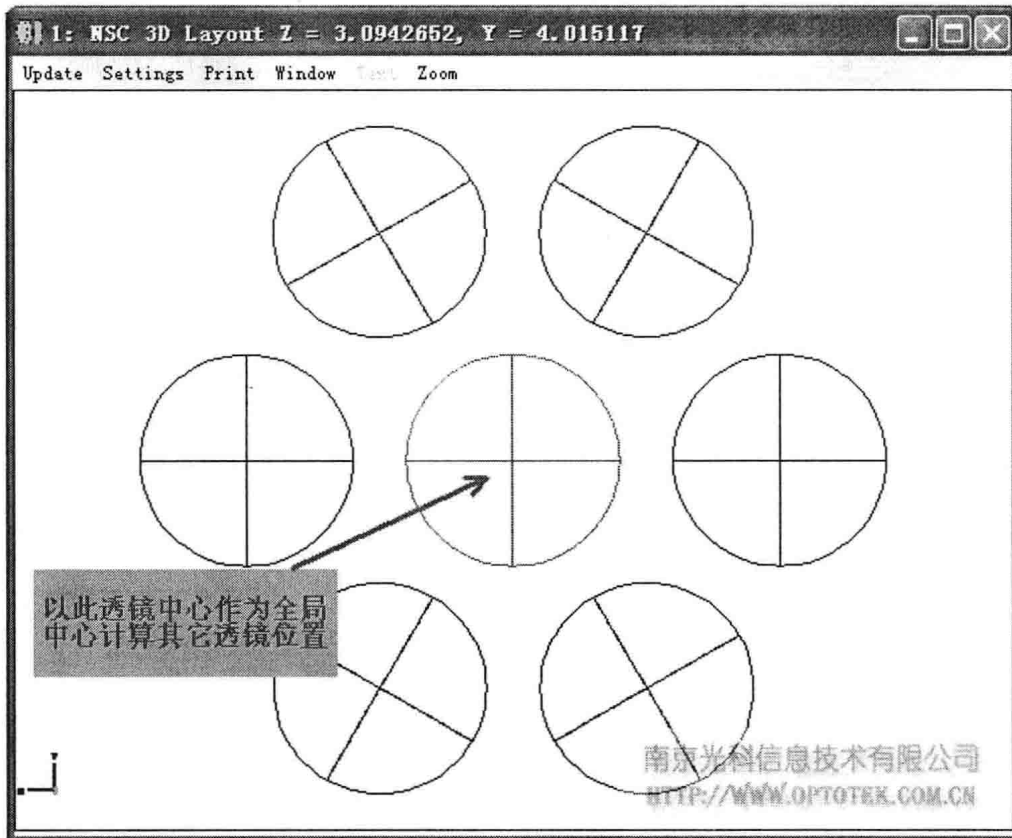
前言

在非序列建模应用中，有些系统元件摆放可能在空间位置上相对复杂，设计者在设计时一般都会使用相对坐标系统，根据某些元件间的相对位置坐标关系可以快速实现空间位置摆放。但在实际装配中，往往需要确定一个坐标原点，即以坐标点作为全局元件的坐标原点来计算出其它所有元件相对这个点的位置和角度。ZEMAX 则具备此功能，帮我们轻松在全局与局部坐标之间任意转换。

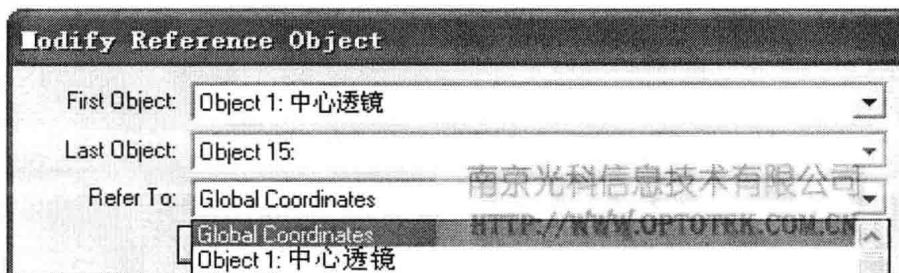
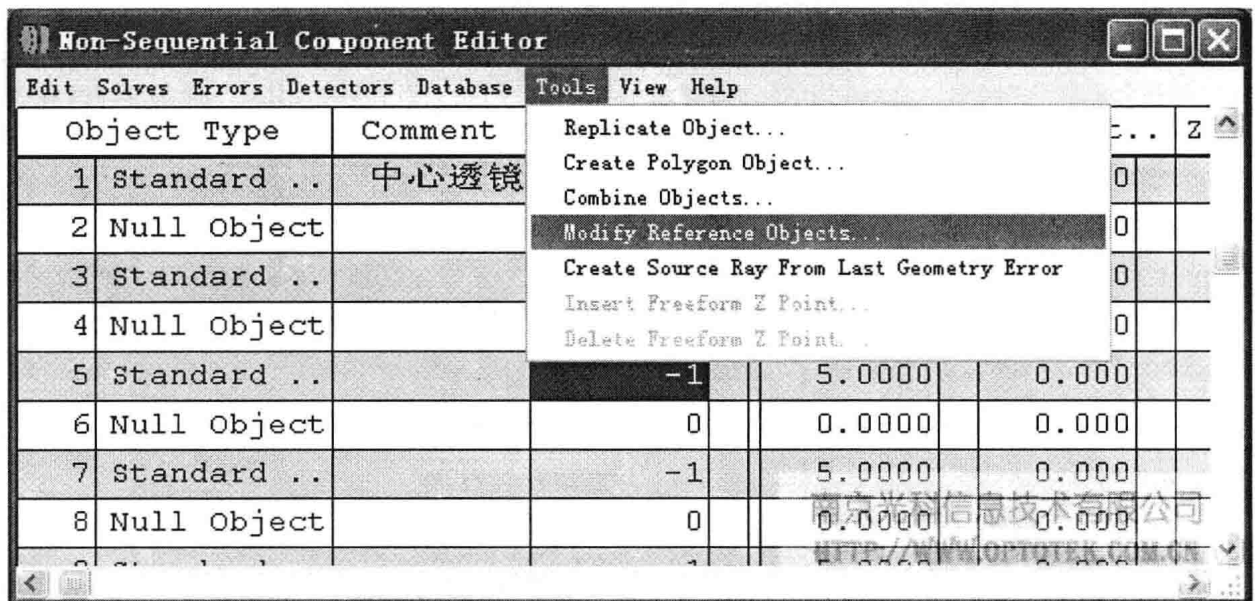
正如我们在上一节“非序列物体参考坐标应用”中看到的例子，（注意：学习本章内容的前提是将上一篇文章中的环形阵列例子完成）环形阵列使用相对坐标参考实现：

Object Type	Comment	Ref Object	X Position	Y Position	Z Position	Tilt About Z
1 Standard ..	中心透镜	0	0.0000	0.0000	0.00	0.000000
2 Null Object		0	0.0000	0.0000	0.00	0.000000
3 Standard ..		-1	5.0000	0.0000	0.00	0.000000
4 Null Object		0	0.0000	0.0000	0.00	60.000000
5 Standard ..		-1	5.0000	0.0000	0.00	0.000000
6 Null Object		0	0.0000	0.0000	0.00	120.000000
7 Standard ..		-1	5.0000	0.0000	0.00	0.000000
8 Null Object		0	0.0000	0.0000	0.00	180.000000

在上图中看到在 Y Position 下的所有值均为 0，而 Tilt About z 的值相应变化，这是典型的参考坐标系。假如我们想以中心透镜的中心点作为全局坐标原点来计算其它 6 个透镜坐标：



当然我们不必手动一个个计算，只需使用非序列编辑器上的坐标转换工具：Modify Reference Objects...



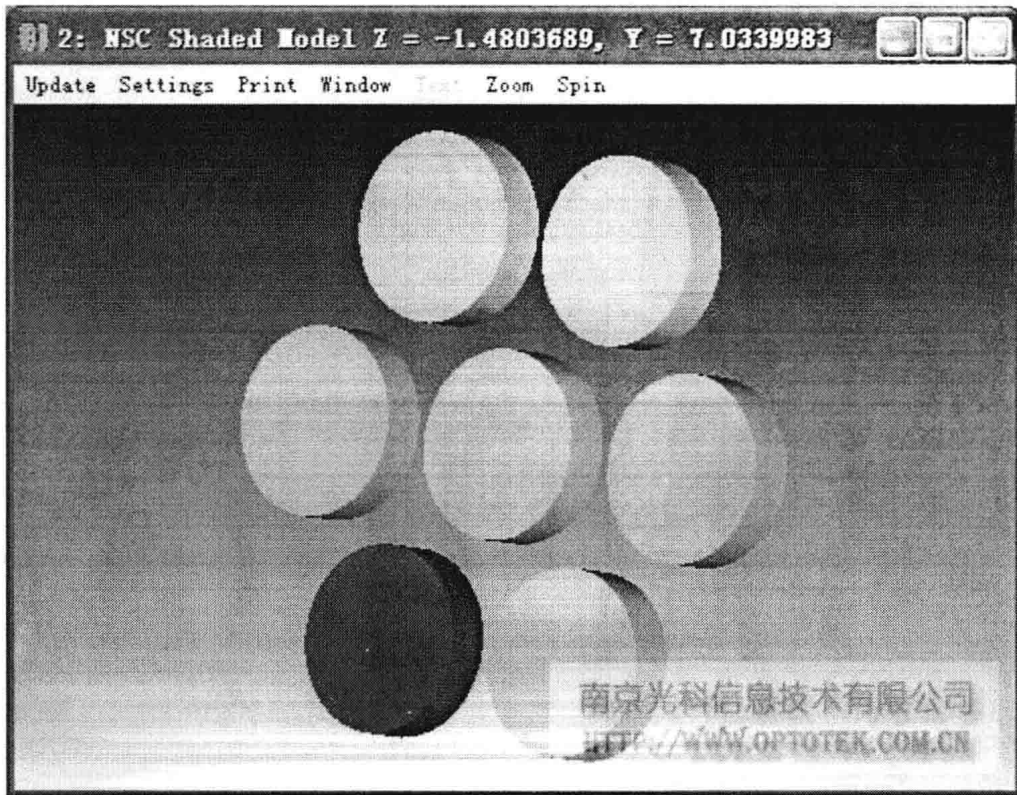
上图中选择指定的物体行, 在 Refer To 中选择要参考的对像, 这里我们以 Global Coordinates 全局参考, 点击 OK 以后全看到, 所有透镜上的参考物体变为 0, Y 坐标实际计算出来, 而旋转角度全为 0. 如下图:

Object	Type	Comment	Ref Object	X Position	Y Posi..	Z Posi..
1	Standard ..	中心透镜	0	0.0000	0.0	0.00
2	Null Object		0	0.0000	0.0	0.00
3	Standard ..		0	5.0000	0.0	0.00
4	Null Object		0	0.0000	0.0	0.00
5	Standard ..		0	2.5000	4.3	0.00
6	Null Object		0	0.0000	0.0	0.00
7	Standard ..		0	-2.500	4.3	0.00
8	Null Object		0	0.0000	0.0	0.00
9	Standard ..		0	-5.000	0.0	0.00
10	Null Object		0	0.0000	0.0	0.00
11	Standard ..		0	2.5000	*	0.00

这时整个系统所有元件便是全局坐标, 各个物体间都是单独存在互不影响, 所以我们可以把所有空白物体 Null Object 直接删除。系统会看起来非常简洁:

Object	Type	Comment	Ref Object	X Position	Y Posi..	Z Posi..
1	Standard ..	中心透镜	0	0.0000	0.0	0.00
2	Standard ..		0	5.0000	0.0	0.00
3	Standard ..		0	2.5000	4.3	0.00
4	Standard ..		0	-2.500	4.3	0.00
5	Standard ..		0	-5.000	0.0	0.00
6	Standard ..		0	2.5000	*	0.00
7	Standard ..		0	-2.500		0.00

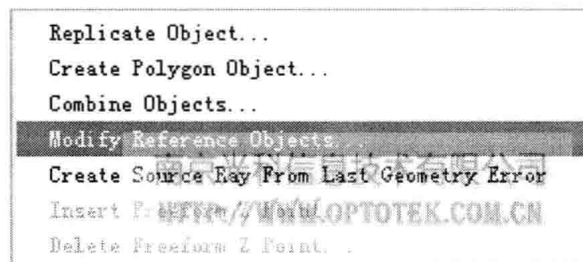
系统没发生任何变化:



此时若我们想生成两组这样的阵列系统，又该如何快速实现呢？同样，使用坐标的强大转换工具，先将系统转换为相对坐标参考，因为只有相对坐标参考下，我们才能保证某组元件的相对位置不变从而进行整体移动。操作方法如下：

在非序列编辑器上第一个物体行点击 Insert 插入一个新的 Null Object, 它将作为整组透镜的虚拟参考点，所有物体以它为参考。在移动时只需设置此 Null Object 的坐标即可。(这只是一种通用的方法，当然在这里我们可以直接使用第一个透镜作为参考，若如此作，这一步可以忽略)

点击非序列物体编辑栏上的 Tools/Modify Reference Objects 工具：



选择：



我们看到，系统又重新回到了局部参考坐标系，而且所有物体均参考物体 1，您可以使用这种方法来修改系统中任意元件间的相对参考坐标。

Non-Sequential Component Editor						
Edit Solves Errors Detectors Database Tools View Help						
Object Type	Comment	Ref Object	X Position	Y Posi..	Z Posi..	
1	Null Object	0	0.0000	0.0	0.00	
2	Standard .. 中心透镜	1	0.0000	0.0	0.00	
3	Standard ..	1	5.0000	0.0	0.00	
4	Standard ..	1	2.5000	4.3	0.00	
5	Standard ..	1	-2.500	4.3	0.00	
6	Standard ..	1	-5.000	0.0	0.00	
7	Standard ..	1	2.5000	*	0.00	
8	Standard ..	1	-2.500	*	0.00	

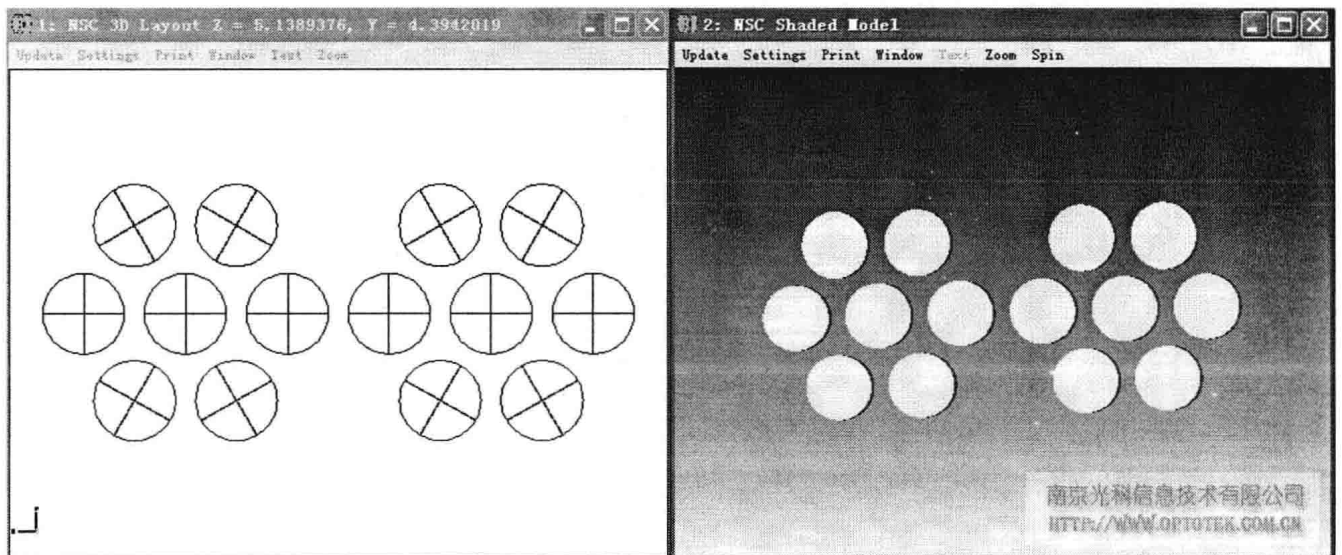
此时我们再来生成另一组阵列，就显得轻而易举了，选中 1 到 8 物体 Ctrl+C 进行复制：

Non-Sequential Component Editor						
Edit Solves Errors Detectors Database Tools View Help						
Object Type	Comment	Ref Object	X Position	Y Posi..	Z Posi..	
1	Null Object	0	0.0000	0.0	0.00	
2	Standard .. 中心透镜	1	0.0000	0.0	0.00	
3	Standard ..	1	5.0000	0.0	0.00	
4	Standard ..	1	2.5000	4.3	0.00	
5	Standard ..	1	-2.500	4.3	0.00	
6	Standard ..	1	-5.000	0.0	0.00	
7	Standard ..	1	2.5000	*	0.00	
8	Standard ..	1	-2.500	*	0.00	

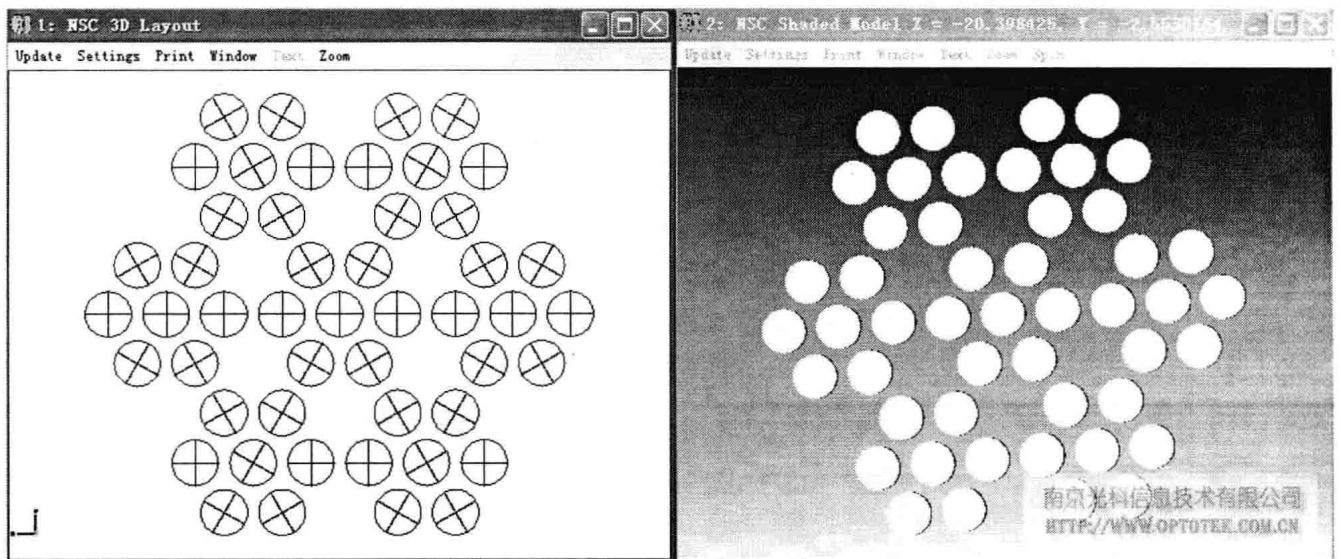
点击物体 1，Ctrl+V 粘贴，并修改第一个物体 X 坐标为 15。如下图，注意两组阵列的参考物体：

Non-Sequential Component Editor						
Edit Solves Errors Detectors Database Tools View Help						
Object Type	Comment	Ref Object	X Position	Y Posi..	Z Posi..	
1	Null Object	0	15.000	0.0	0.00	
2	Standard ..	1	0.0000	0.0	0.00	
3	Standard ..	1	5.0000	0.0	0.00	
4	Standard ..	1	2.5000	4.3	0.00	
5	Standard ..	1	-2.500	4.3	0.00	
6	Standard ..	1	-5.000	0.0	0.00	
7	Standard ..	1	2.5000	*	0.00	
8	Standard ..	1	-2.500	*	0.00	
9	Null Object	0	0.0000	0.0	0.00	
10	Standard ..	9	0.0000	0.0	0.00	
11	Standard ..	9	5.0000	0.0	0.00	

最后阵列视图如下：



同样使用上一篇文章所讲述的方法，我们可以很快将这一组透镜生成一个较大的六边形阵列：



总结:

本文继续上篇文章的相对坐标与局部坐标的讨论，深入讲解了全局坐标与局部坐标间的转换。在今后的建模任务中，可以轻松处理类似的情况，达到事半功倍的效果。