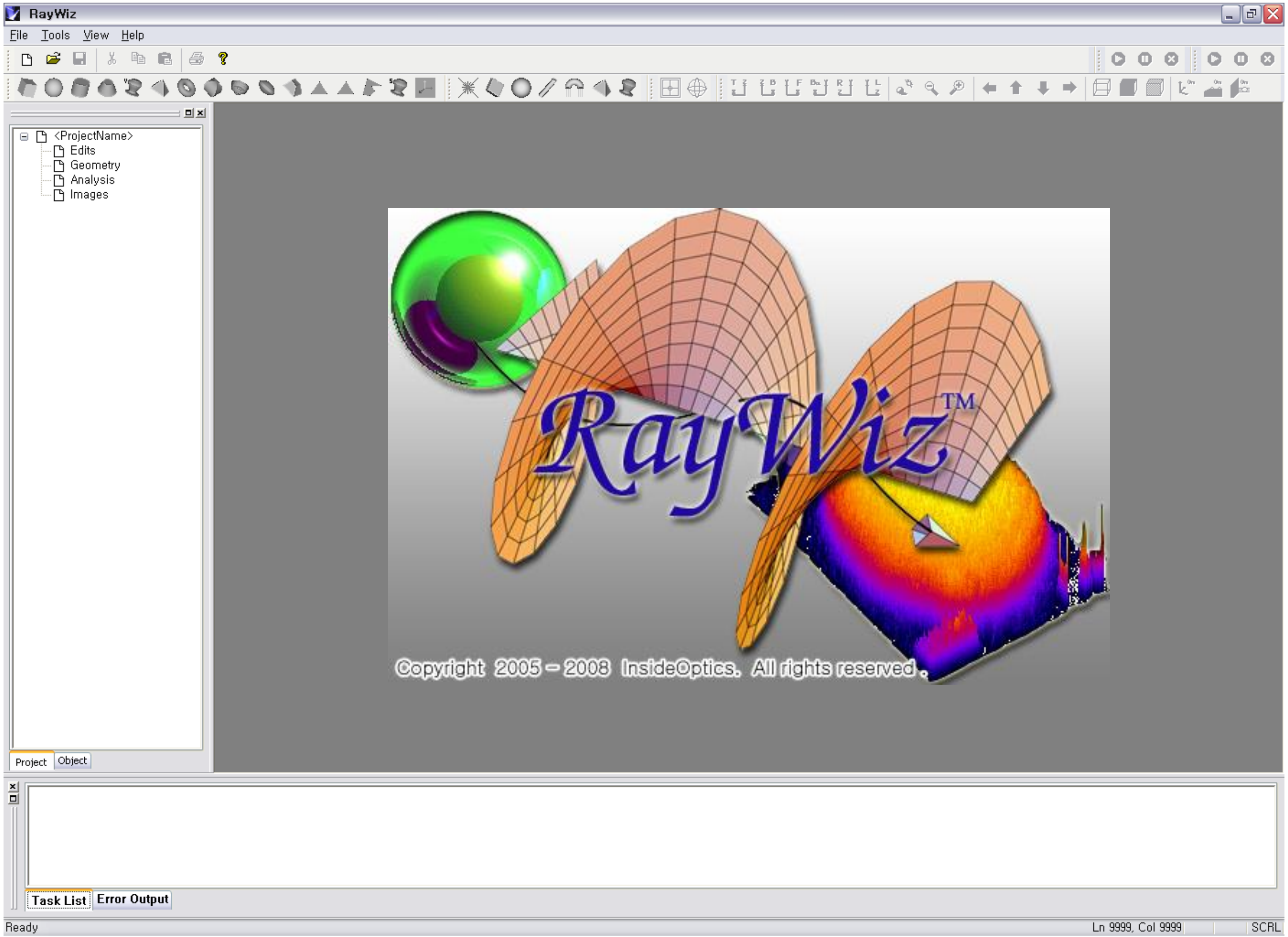
“RAYWIZ Basic & Display Application”

www.insideoptics.com

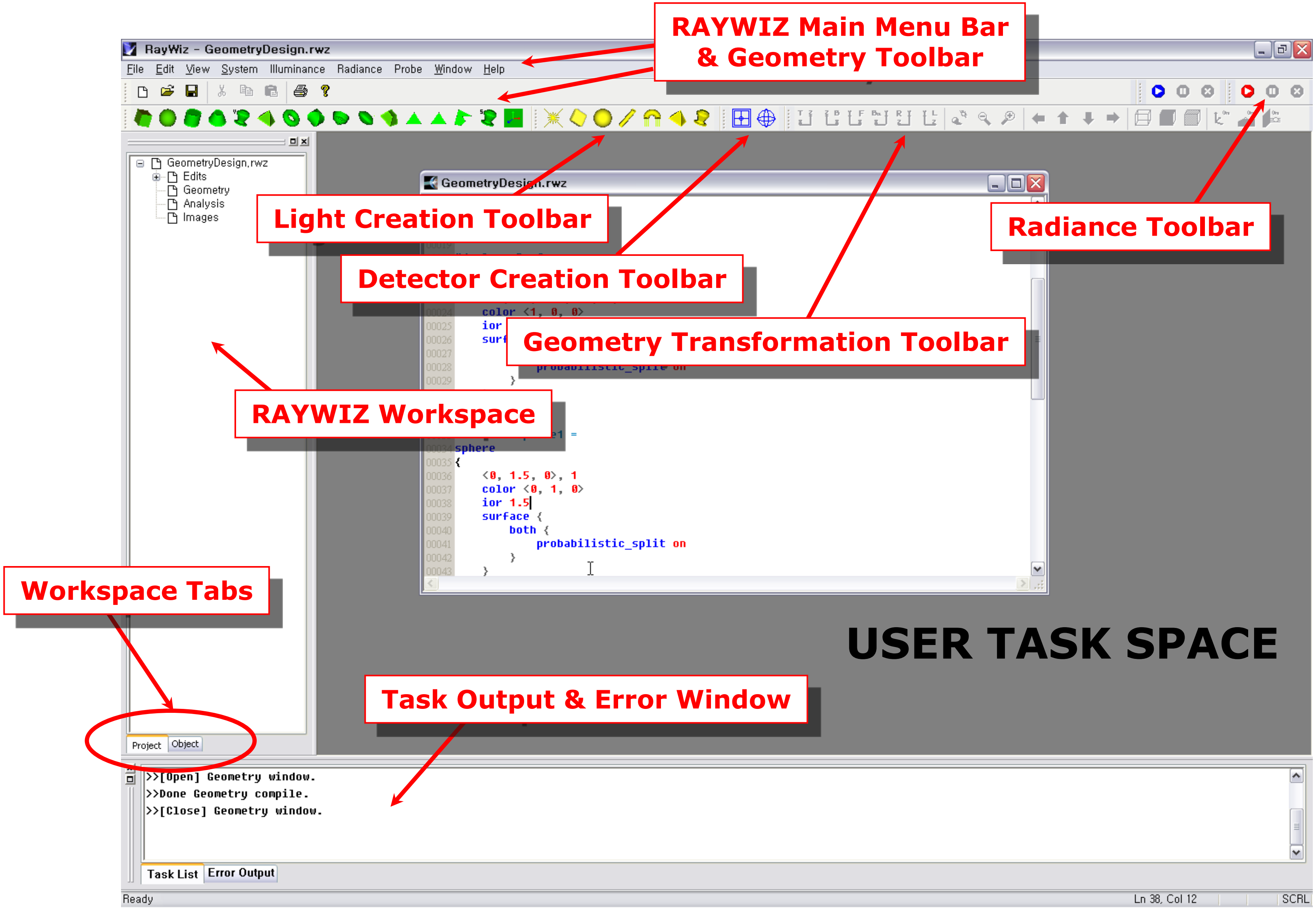
RAYWIZ BASIC

Basic: Starting RAYWIZ

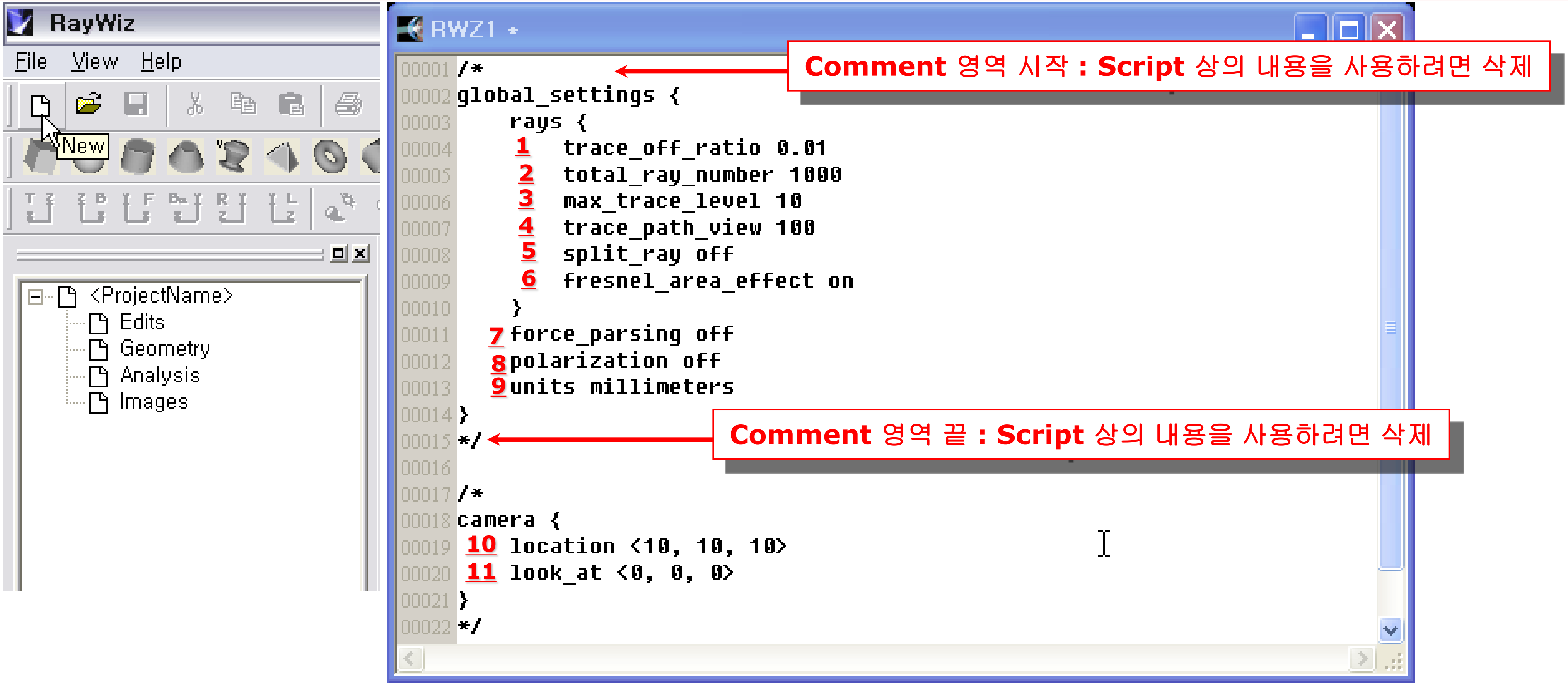
바탕화면의 **RAYWIZ** 아이콘을 **Double-Click** 하여 실행



Basic: The RAYWIZ Window



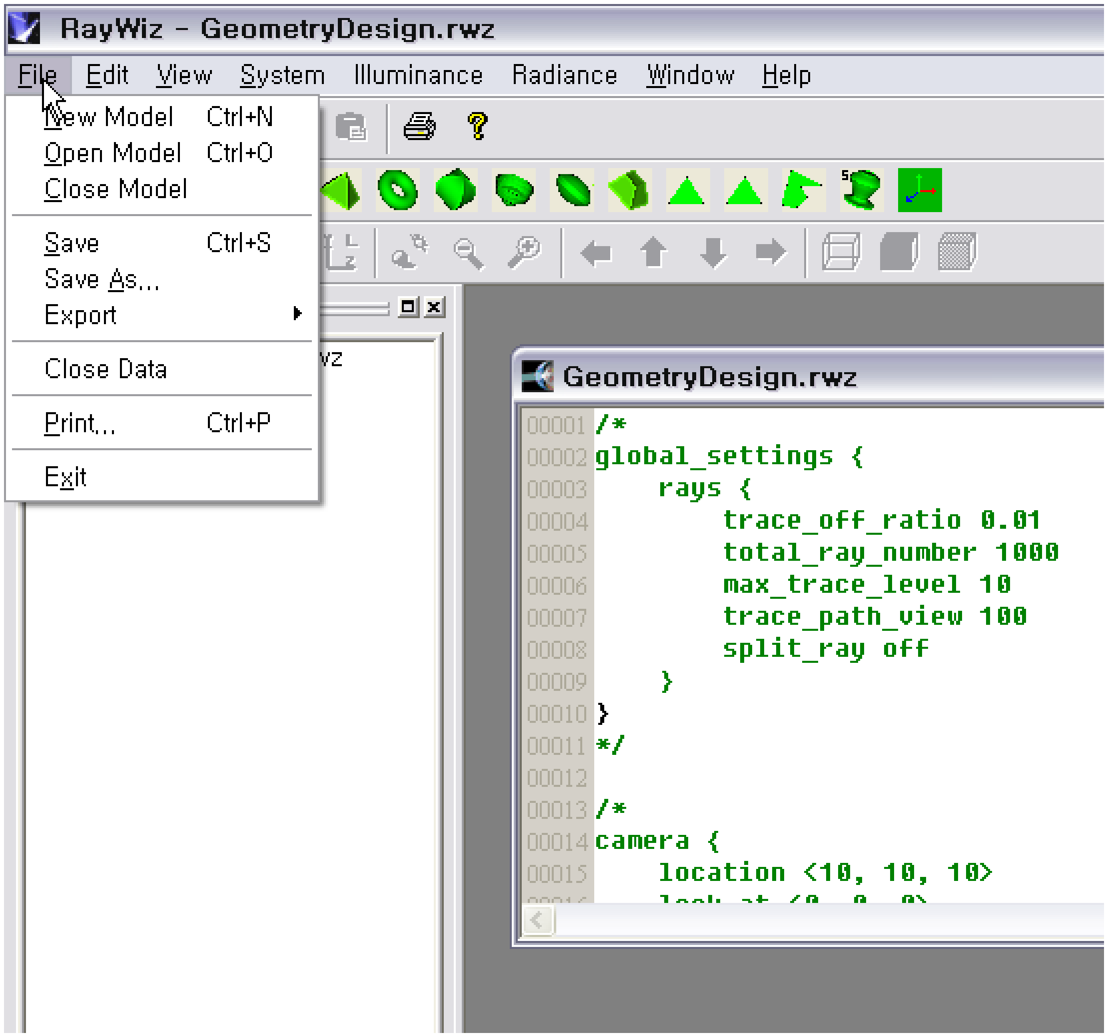
Basic: Project File



1. **Light Source** 에서 출발한 **Ray**가 처음에너지의 몇**%**에 도달할 때 까지 진행할지 설정
2. **Simulation**에 사용할 총 **Ray** 수 설정
3. **Object**의 각 **Surface**에 몇 회의 충돌을 허용 할지 설정
4. **Simulation**이 끝나면 **Geometry View** 창에 표시해줄 **Ray** 수 설정
5. **Ray**가 **Object**의 **Surface**와 만났을 때 매번 **Split** 시켜 두개의 **Ray**로 진행시킬지 설정
6. **Ray** 진행에 있어 **Frsnel Equation**의 면적효과를 적용 설정
7. **Uniformity Optimization**을 위한 강제 갱신 기능설정
8. **Surface**에 **Jones Matrix**를 부여한 경우 **Polarization**을 설정. **9.** 시스템에 적용할 기본 단위
10. **Radiance View**의 위치를 설정
11. **Radiance View**의 관찰점을 설정

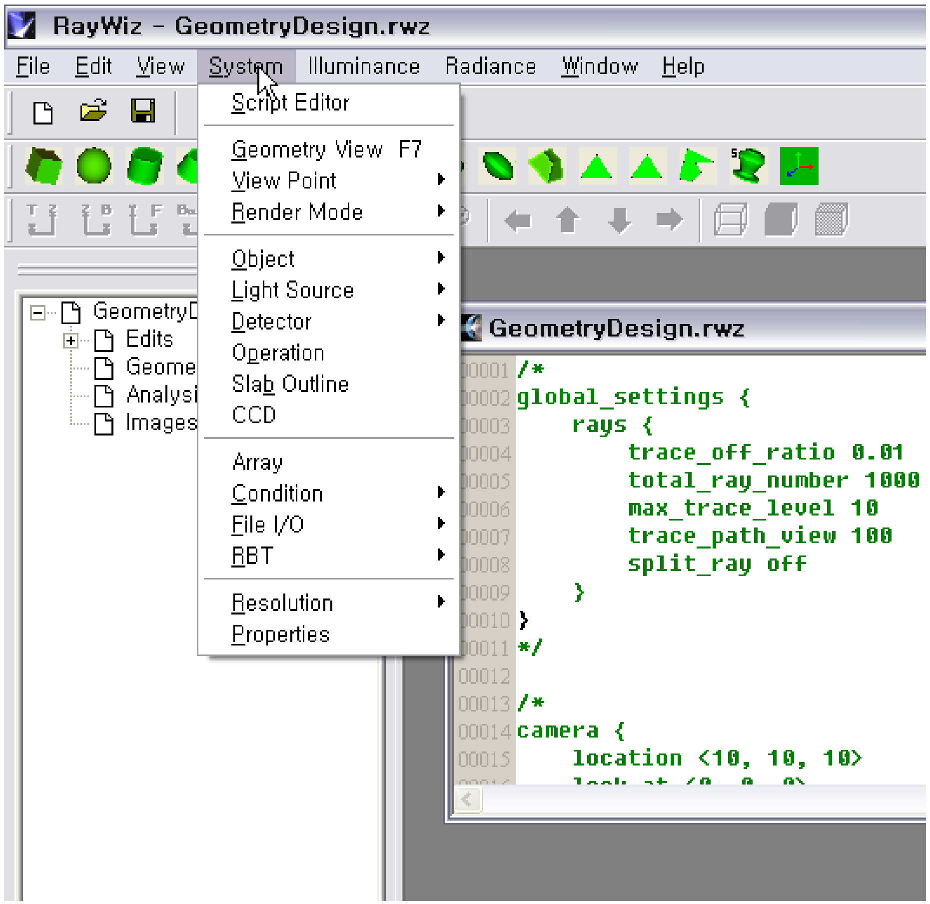
RAYWIZ BASIC
- MENU -

Basic: File Menu



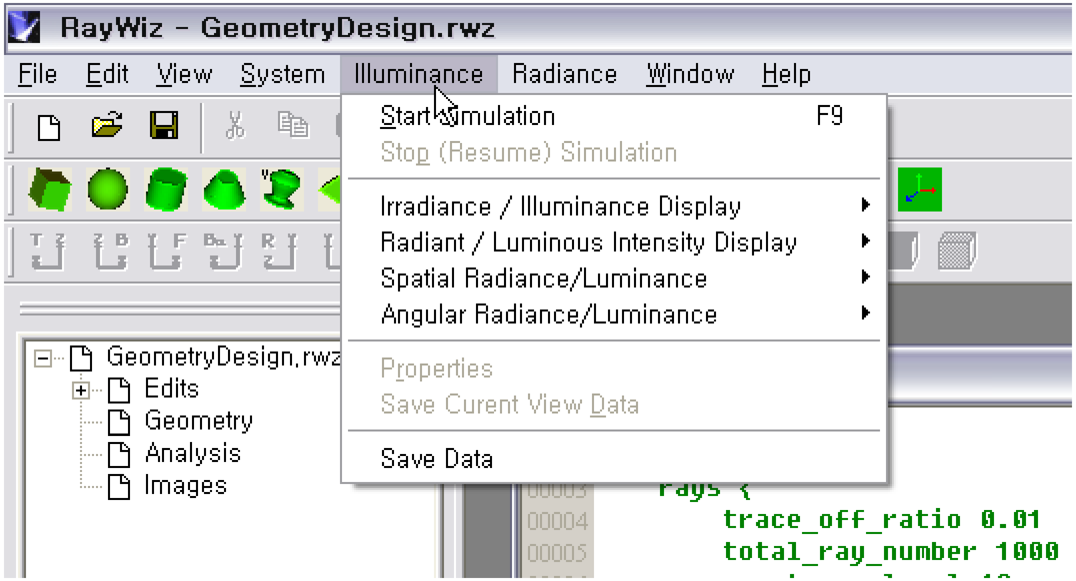
1. **File**
2. **Edit**
3. **View**
4. **System**
5. **Illuminance**
6. **Radiance**
7. **Window**
8. **Help**

Basic: System Menu



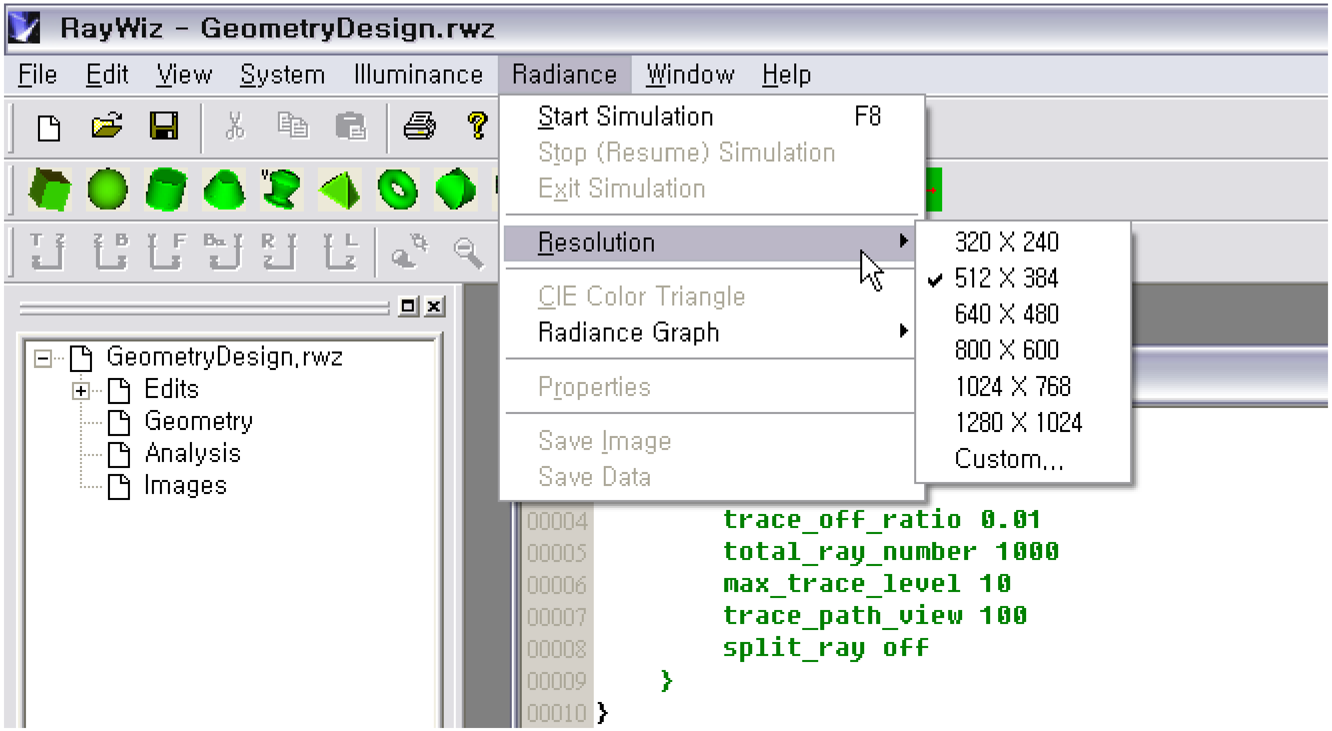
- **Script Editor :** 에디터 창으로 이동
- **Geometry View : Geometry** 화면 보기
- **View Point : Geometry** 화면에서 좌표계의 시점을 변경
- **Render Mode : Geometry** 화면을
  /Wireframe
  / Solid
  / Translucent 보기 변경
- **Object :** 에디터 창에 **Geometry Object**를 생성
- **Light Source :** 에디터 창에 **Light Source**를 생성
- **Detector :** 에디터 창에 **Detector**를 생성
- **Operation :** 에디터 창에
  /Union
  /Intersect
  /Subtract
  /Immerse
  /Group 실행
- **Slab Outline : Slab Object** 생성
- **CCD:** 시스템 스크립트 생성
- **Array:** 명령어 스크립트 생성
- **Condition:** 매크로 문법 스크립트 생성
- **File I/O: File Input/Output** 스크립트 생성
- **RBT: *.rbt** 파일을 위한 스크립트 생성
- **Resolution : Geometry** 해상도
- **Properties : Global Setting** 대화상자 실행

Basic: Illuminance Menu



- **Start Simulation :** 시뮬레이션 시작
- **Stop Simulation :** 시뮬레이션 중지
- **Irradiance/Illuminance Display : Illuminance** 계산
- **Radiant/Luminous Intensity Display : Radiance** 계산
- **Spatial Radiance/Luminance : Plane Detector** 가 있을 경우 **Luminance(**위치 별**)** 계산
- **Angular Radiance/Luminance : Plane Detector** 가 있을 경우 **Luminance(**각도 별**)** 계산
- **Properties :** 측정 **Bin** 의 크기 속성 변경 및 **Intensity Bar Range** 속성 변경
- **Save current View Data :** 개별의 결과분석 창의 **Data**를 저장
- **Save Data : Simulation**을 수행한 전체 **Ray Data**를 저장 ← 다시 불러들여 모든 분석창을 재실행할 수 있음.

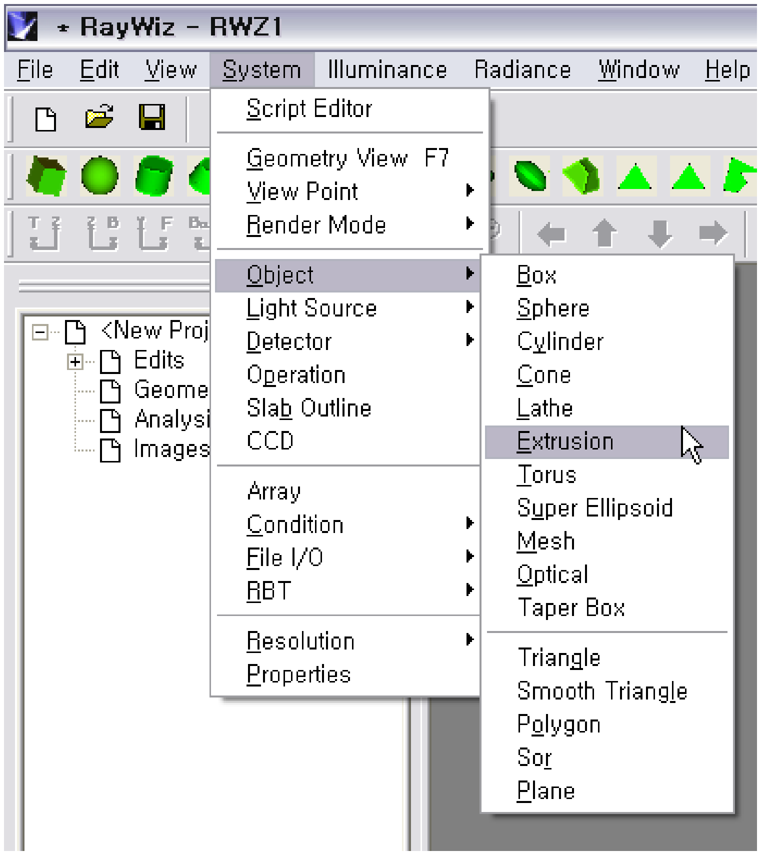
Basic: Radiance Menu



- **Start Simulation :** 시뮬레이션 시작
- **Stop Simulation :** 시뮬레이션 일시 정지
- **Exit Simulation :** 시뮬레이션 정지
- **Resolution :** 해상도 변경
- **CIE Color Triangle, Radiance Graph :** 정량 분석창
- **Save Image :** 결과이미지 및 변환 이미지 저장

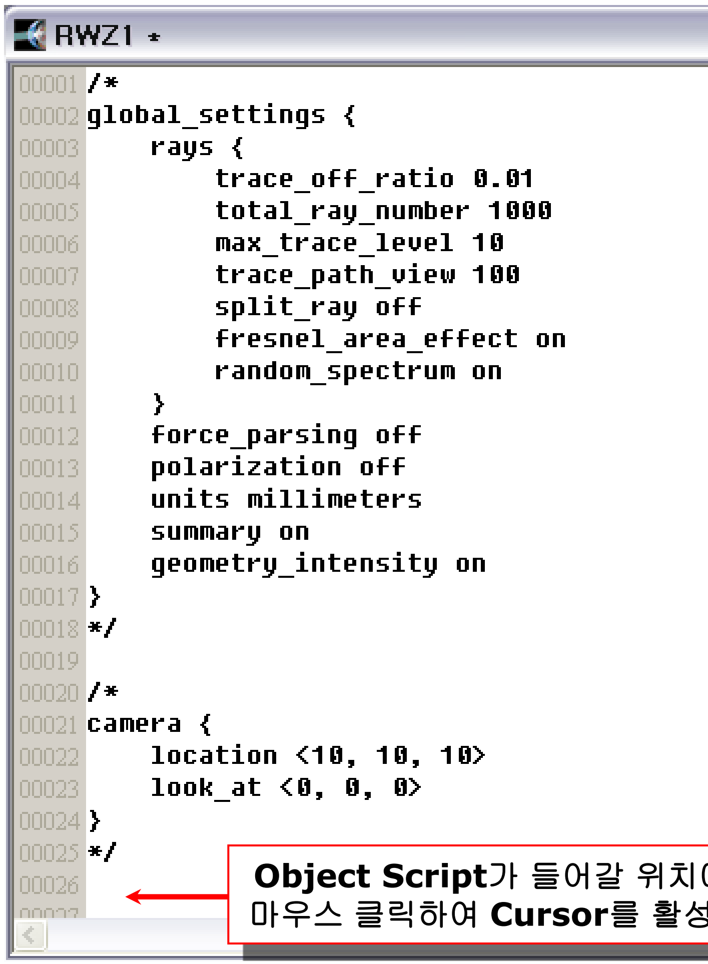
RAYWIZ BASIC
- Object Geometry I -

Geometry I: Primitive Object

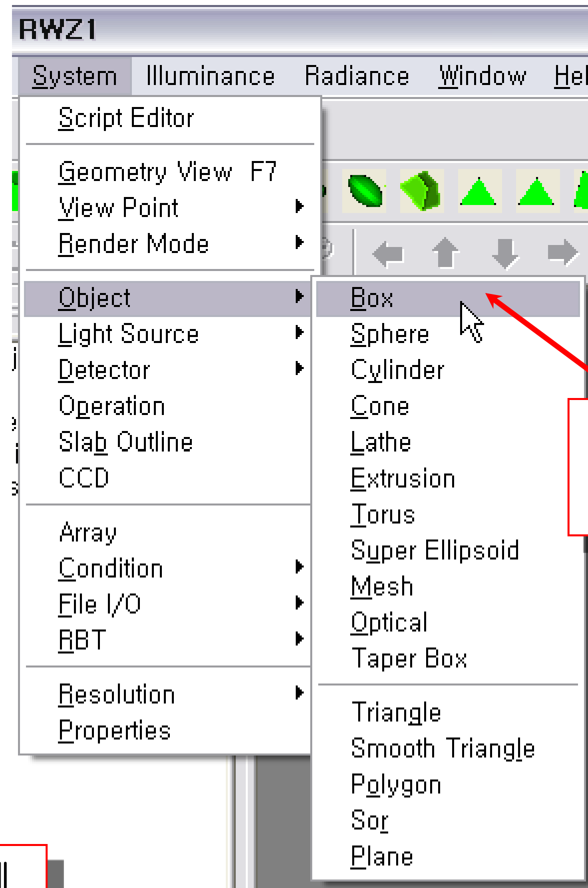


1. **Box**
2. **Sphere**
3. **Cylinder**
4. **Cone**
5. **Lathe**
6. **Extrusion**
7. **Torus**
8. **Super Ellipsoid**
9. **Mesh**
10. **Optical**
11. **Taper Box**
12. **Triangle**
13. **Smooth Triangle**
14. **Polygon**
15. **Sor**
16. **Plane**

Geometry I: Create Your First Object – Box(1)

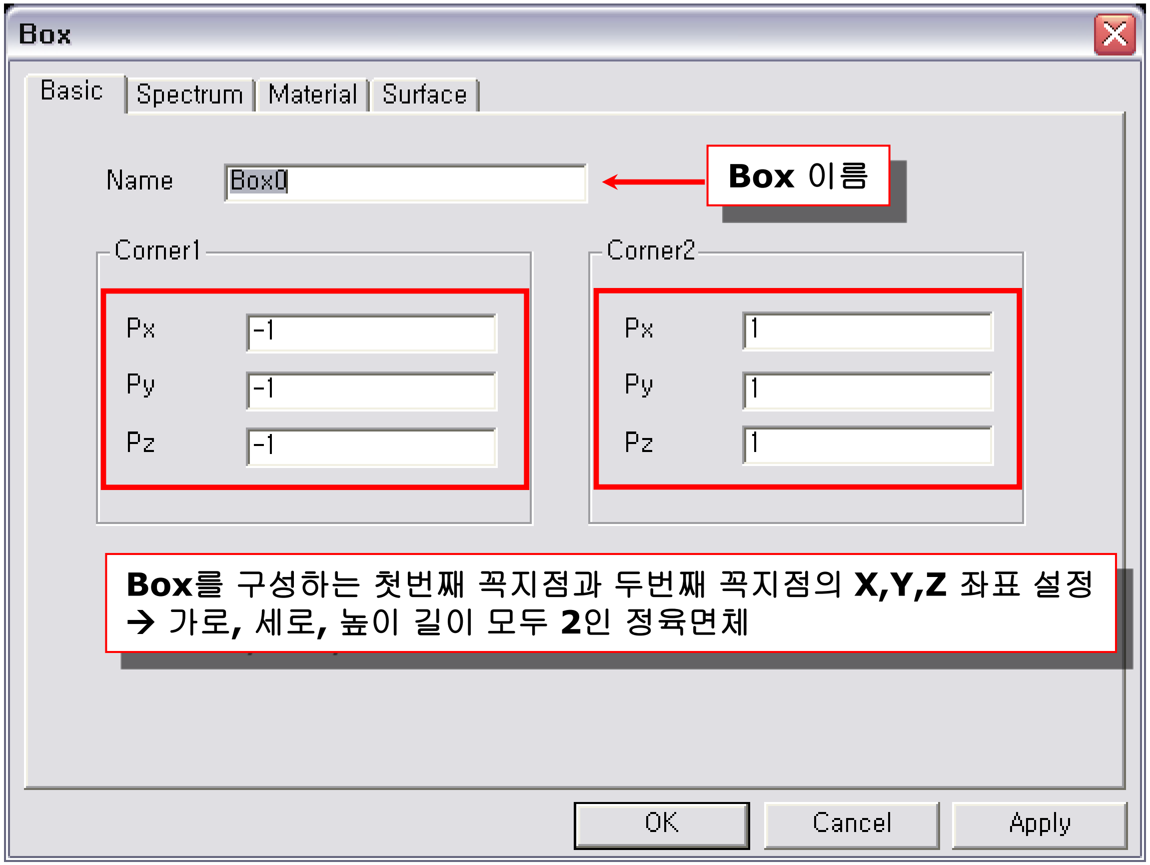


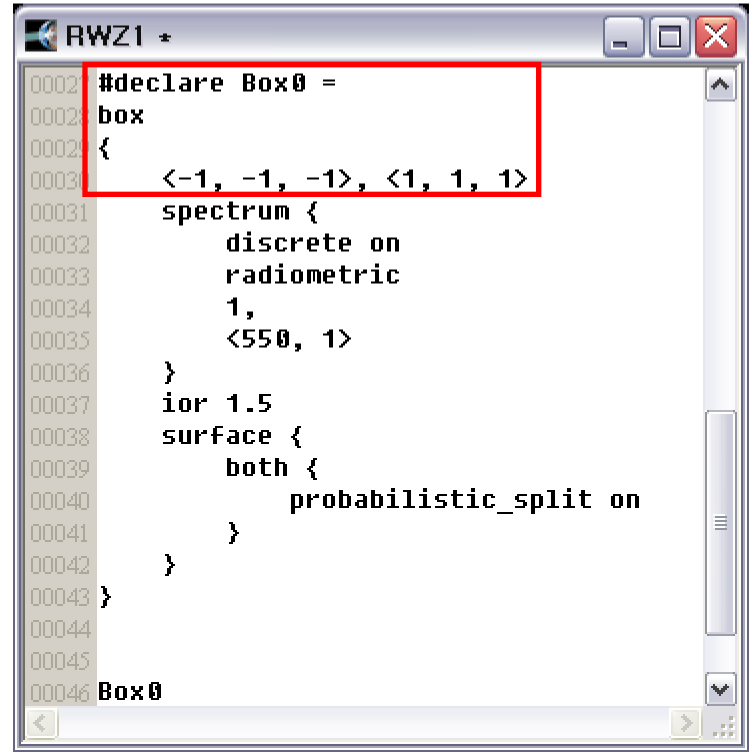
Object Script가 들어갈 위치에
마우스 클릭하여 **Cursor**를 활성화



System > Object > Box 선택
또는
Object Toolbar에서 **Box** 선택

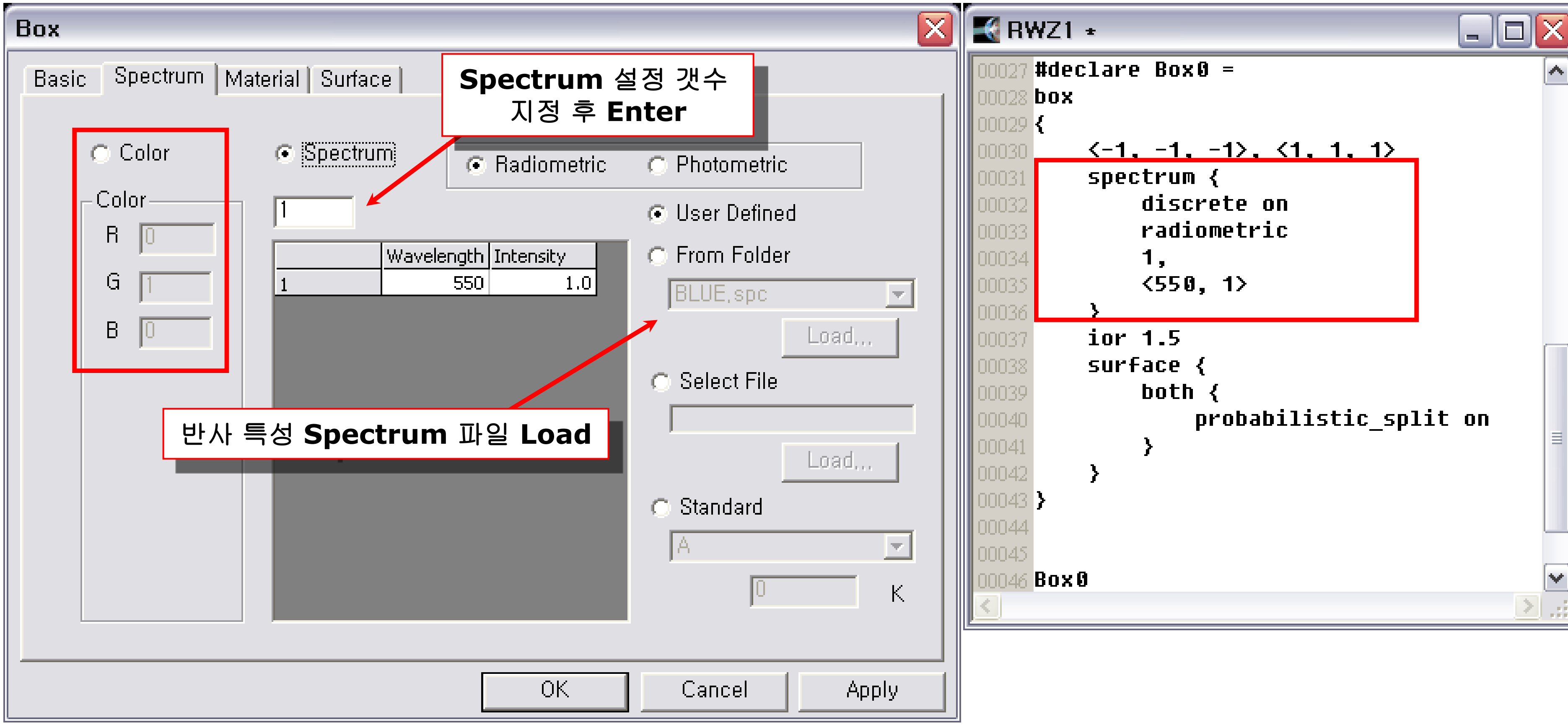
Geometry I: Create Your First Object – Box(2)





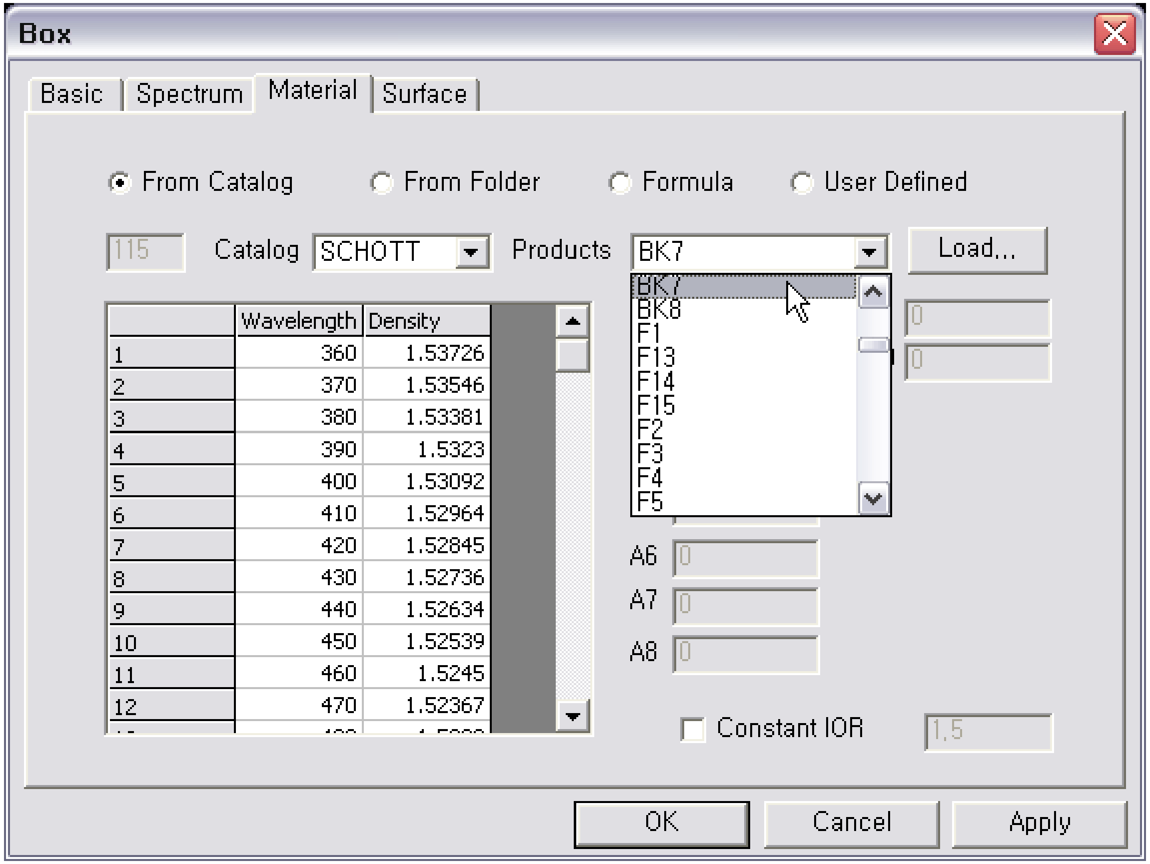
기본값 설정 그대로 확인을 누르면 **Script** 창에 **Box0**라는 **Object**가 선언된다.

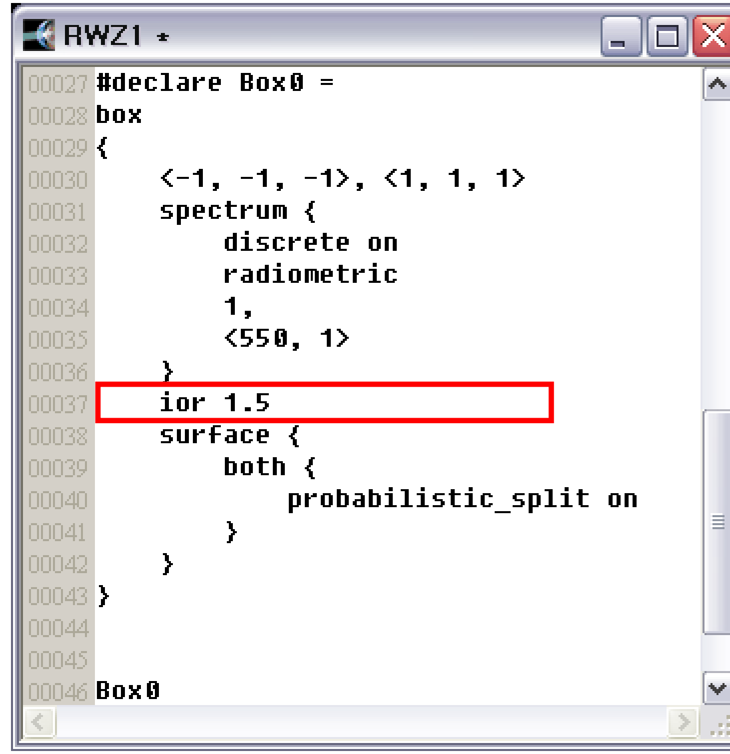
Geometry I: Create Your First Object – Box(3)



Color R,G,B 값으로 반사 특성을 정의하거나 **Spectrum**을 직접 입력하여 정의할 수 있다.

Geometry I: Create Your First Object – Box(4-1)

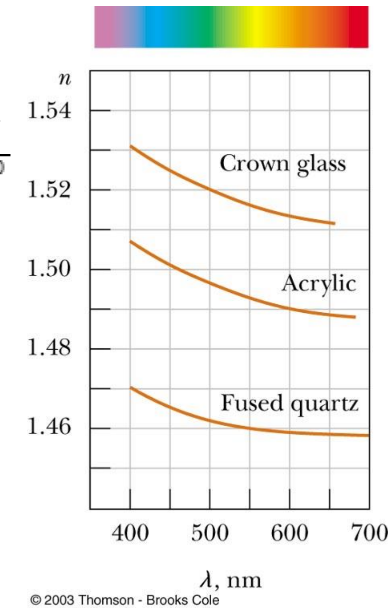




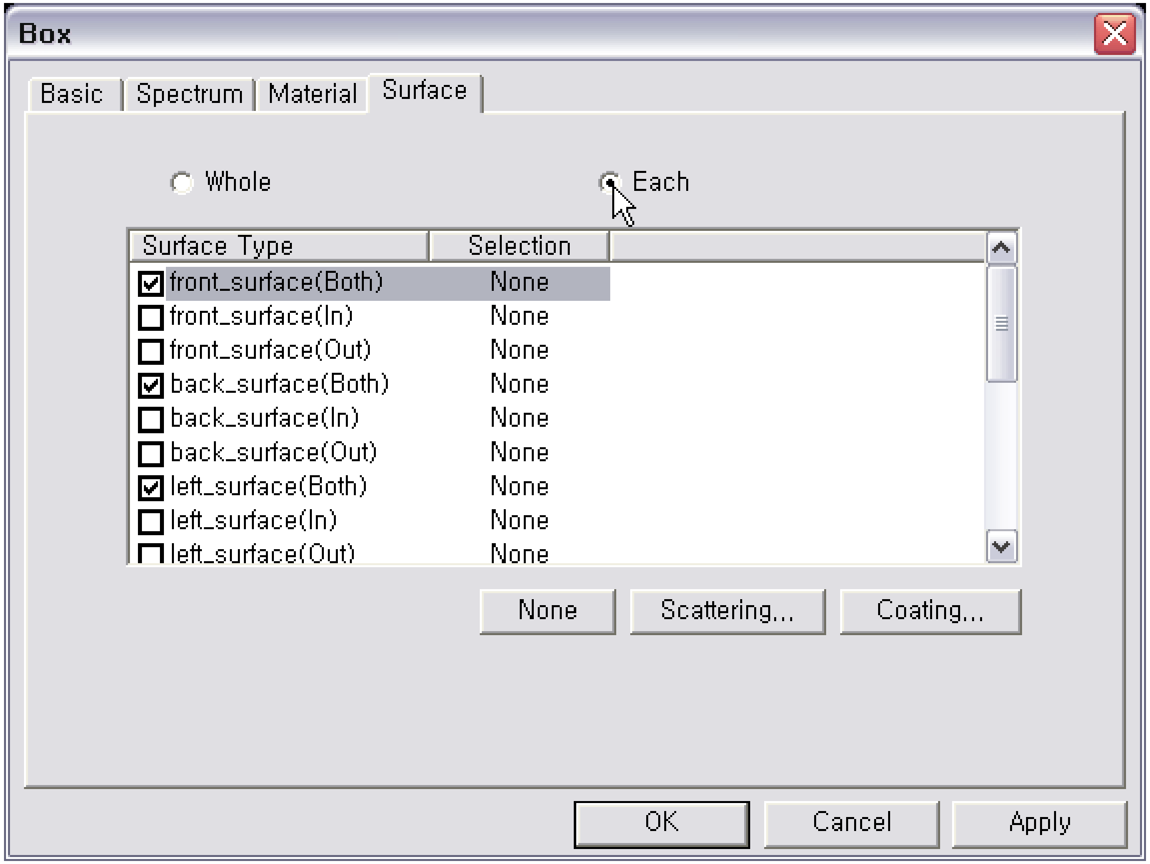
IOR (Index of Refraction) 지정을 위한 **Tab**으로
기본값인 **Constant IOR**를 해제하여 **Glass**나 **Formula**에 의한 설정을 할 수 있다.

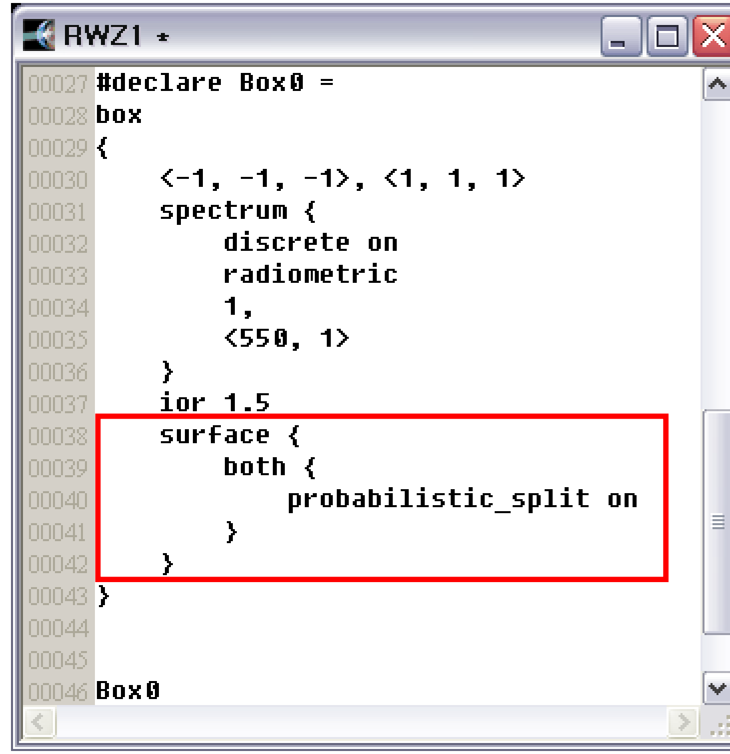
Geometry I: Create Your First Object – Box(4-2)

CONRADY	CONRADY1	$n(\lambda)=A+\frac{B}{\lambda}+\frac{C}{\lambda^{3.5}}$
EXTENDED-SCHOTT	SCHOTT5X5	$n(\lambda)^2=A+B\lambda^2+C\lambda^4+D\lambda^6+E\lambda^8+\frac{F}{\lambda^2}+\frac{G}{\lambda^4}+\frac{H}{\lambda^6}+\frac{J}{\lambda^8}+\frac{K}{\lambda^{10}}$
HERZBERG	HERZBRGR4X2	$n(\lambda)=A+B\lambda^2+C\lambda^4+D\lambda^6+\frac{E}{(\lambda^2-0.028)}+\frac{F}{(\lambda^2-0.028)^2}$
OPTICS1	HoO1	$n(\lambda)^2=A+B\lambda^2+\frac{C}{\lambda^2-D^2}$
OPTICS2	HoO2	$n(\lambda)^2=A+B\lambda^2+\frac{C\lambda^2}{\lambda^2-D^2}$
SCHOTT	SCHOTT2X4	$n(\lambda)^2=A+B\lambda^2+\frac{C}{\lambda^2}+\frac{D}{\lambda^4}+\frac{E}{\lambda^6}+\frac{F}{\lambda^8}$
SELLMEIER1	SELL3T	$n(\lambda)^2-1=\frac{A\lambda^2}{\lambda^2-B^2}+\frac{C\lambda^2}{\lambda^2-D^2}+\frac{E\lambda^2}{\lambda^2-F^2}$
SELLMEIER2	SELLMOD6	$n(\lambda)^2=A+\frac{B\lambda^2}{\lambda^2-C^2}+\frac{D}{\lambda^2-E^2}$
SELLMEIER3	SELL4T	$n(\lambda)^2-1=\frac{A\lambda^2}{\lambda^2-B^2}+\frac{C\lambda^2}{\lambda^2-D^2}+\frac{E\lambda^2}{\lambda^2-F^2}+\frac{G\lambda^2}{\lambda^2-H^2}$
SELLMEIER4	SELL2TA	$n(\lambda)^2=A+\frac{B\lambda^2}{\lambda^2-C^2}+\frac{D\lambda^2}{\lambda^2-E^2}$



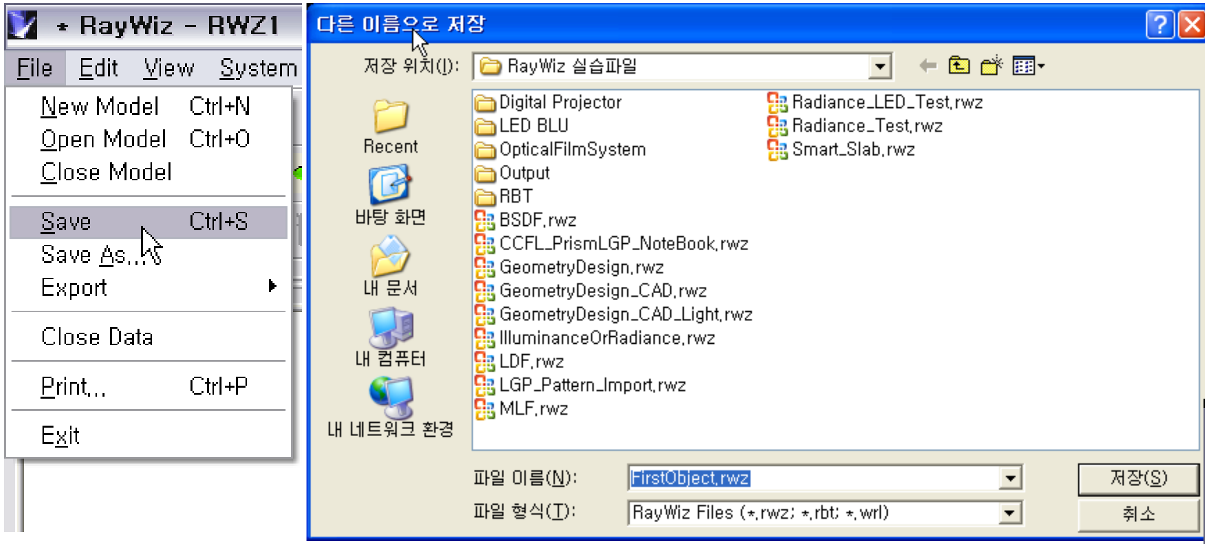
Geometry I: Create Your First Object – Box(5)

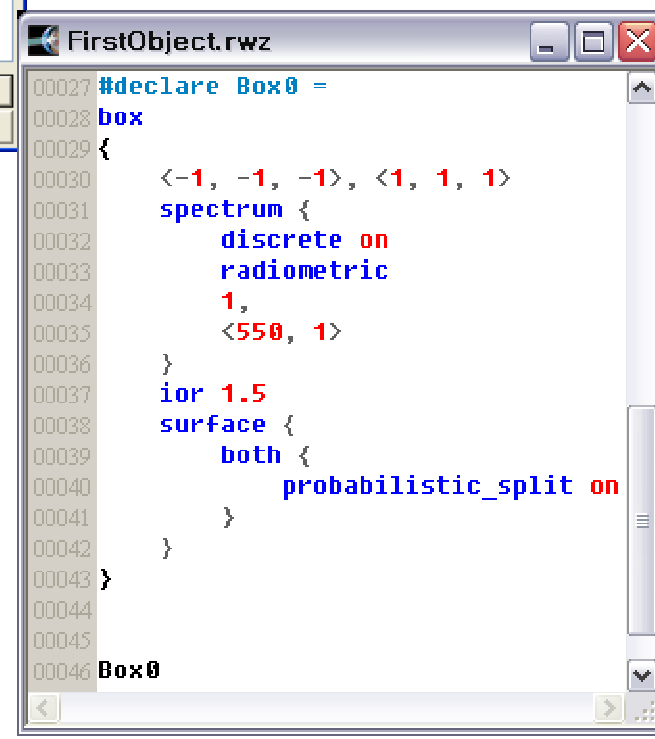




기본값은 **Object**의 모든 **Surface** 특성을 한번에 정의하는 **Whole**이 선택되어 있으나
Object의 각 **Surface**의 특성을 달리 설정하려면 **Each**를 선택하여 **Scattering**이나 **Coating** 설정을 한다.

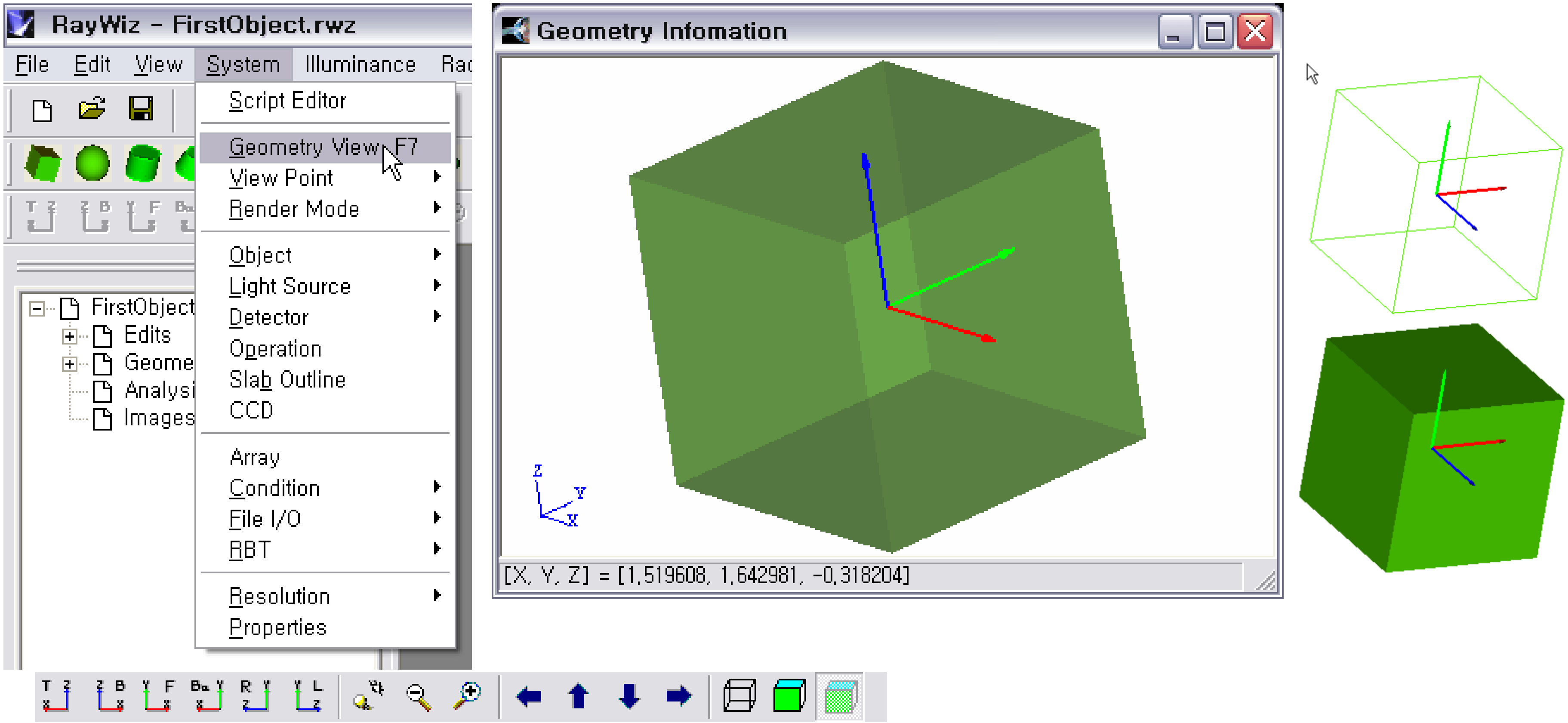
Geometry I: Create Your First Object – Box(6)





File>Save 를 선택하여 적절한 이름으로 **Project Script**를 저장한다. 그러면 **Script**창의 **Title**이 저장한 이름으로 변경되며 명령어들은 몇 가지 색깔로 변하게 된다.

Geometry I: Create Your First Object – Box(7)



기준축 정렬

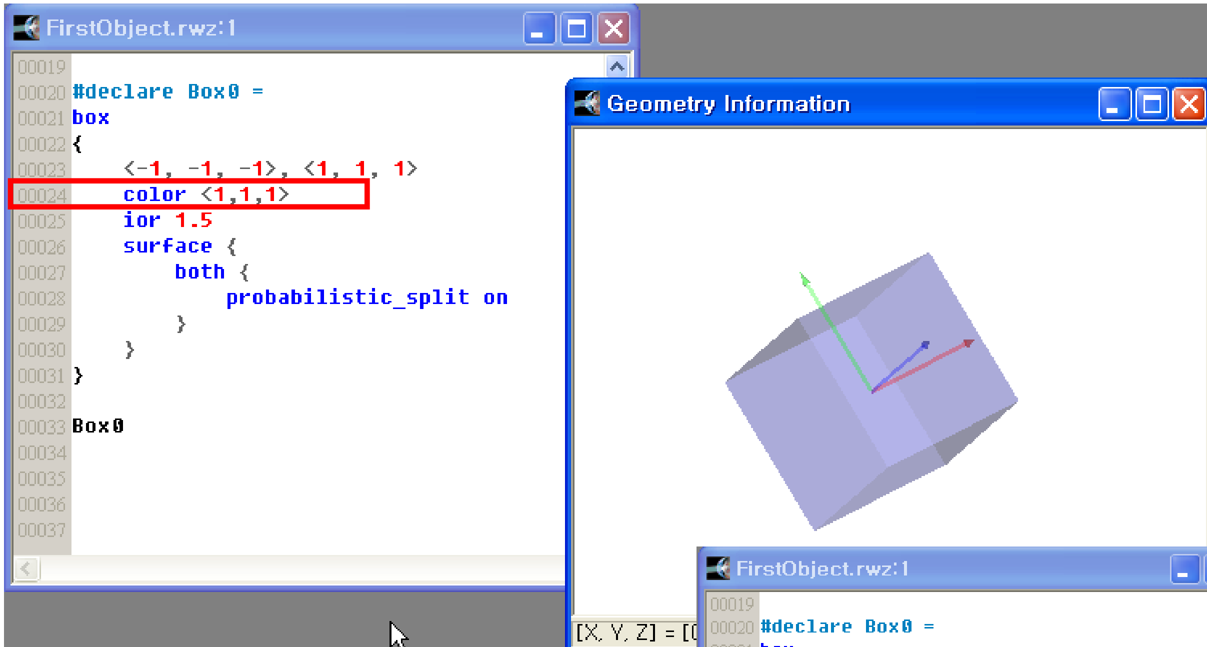
Perspective View 및 확대 축소

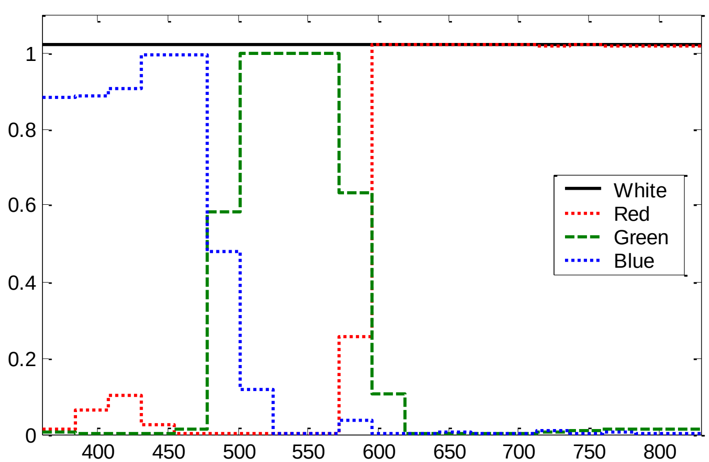
상하, 좌우 이동

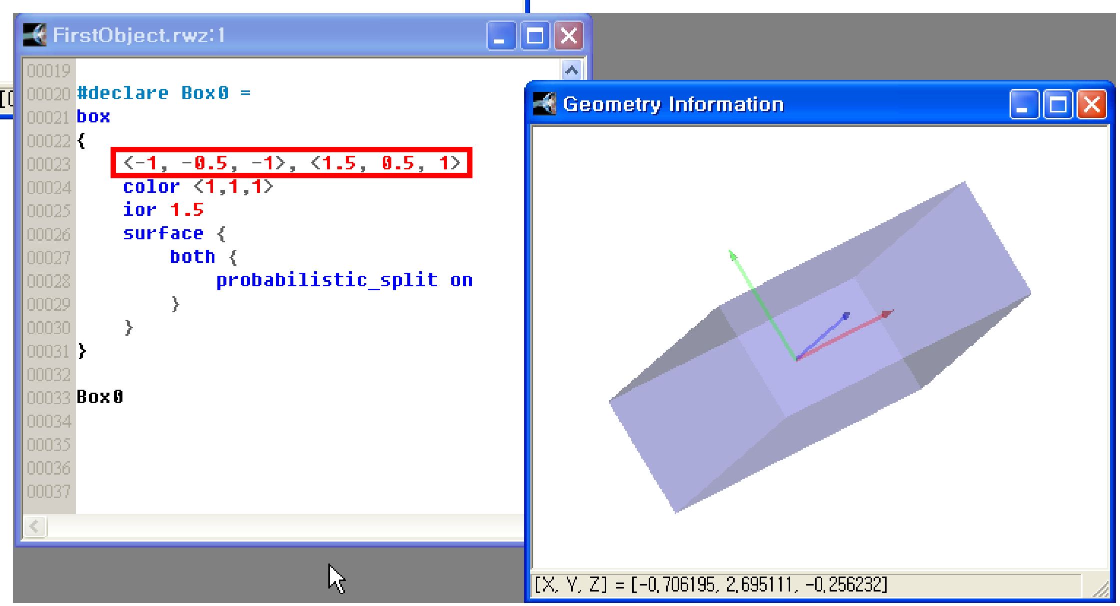
Wireframe/ Solid/ Translucent

- 마우스 왼쪽 버튼을 누르고 드래그 → 자유회전
- 스크롤 또는 오른쪽 버튼을 누르고 드래그 → 확대, 축소
- **Shift** 키 누르고 드래그 → **2D** 자유이동
- 마우스 오른쪽 버튼이용 영역선택 → 부분확대

Geometry I: Modify Your First Object – Spectrum & Scale

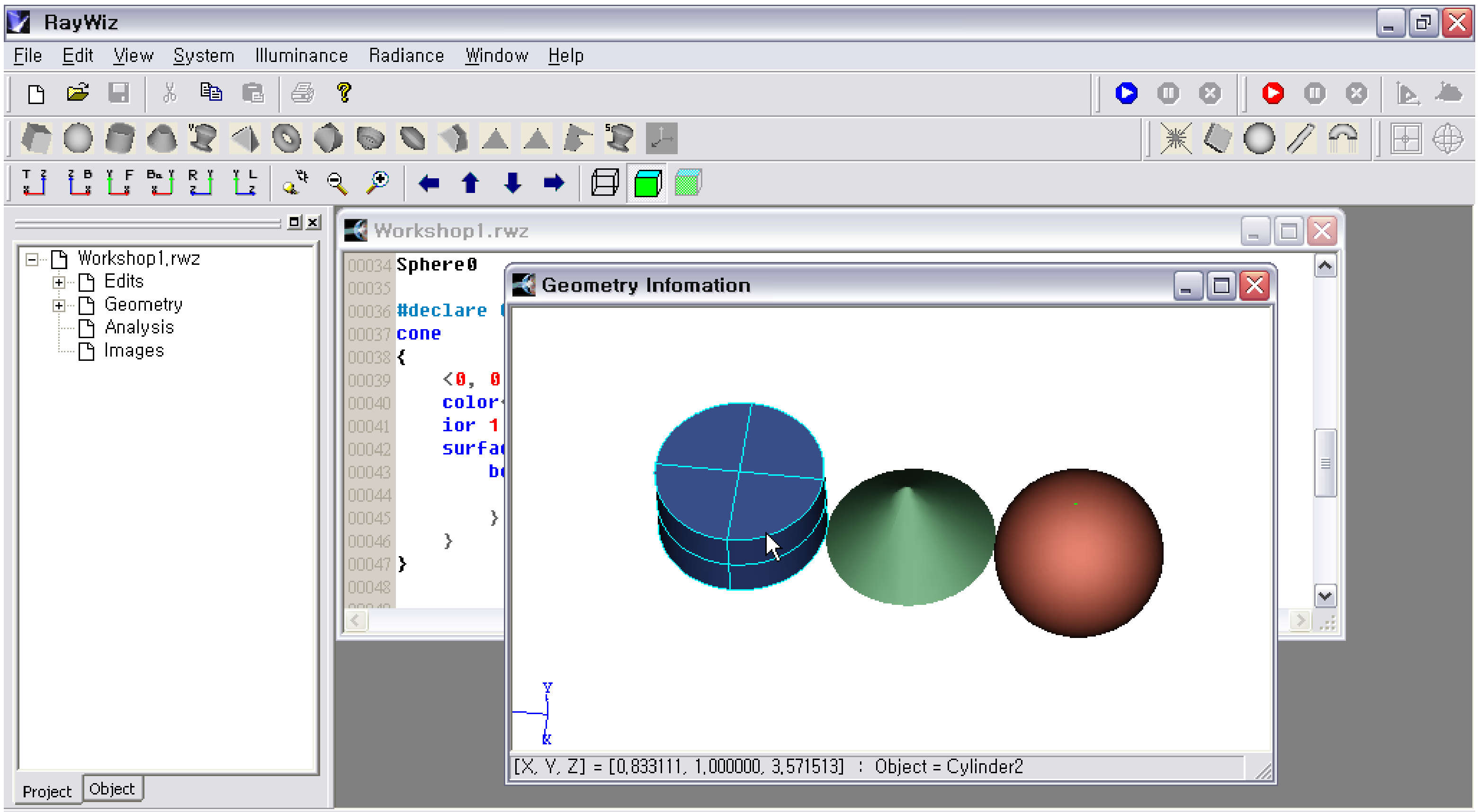




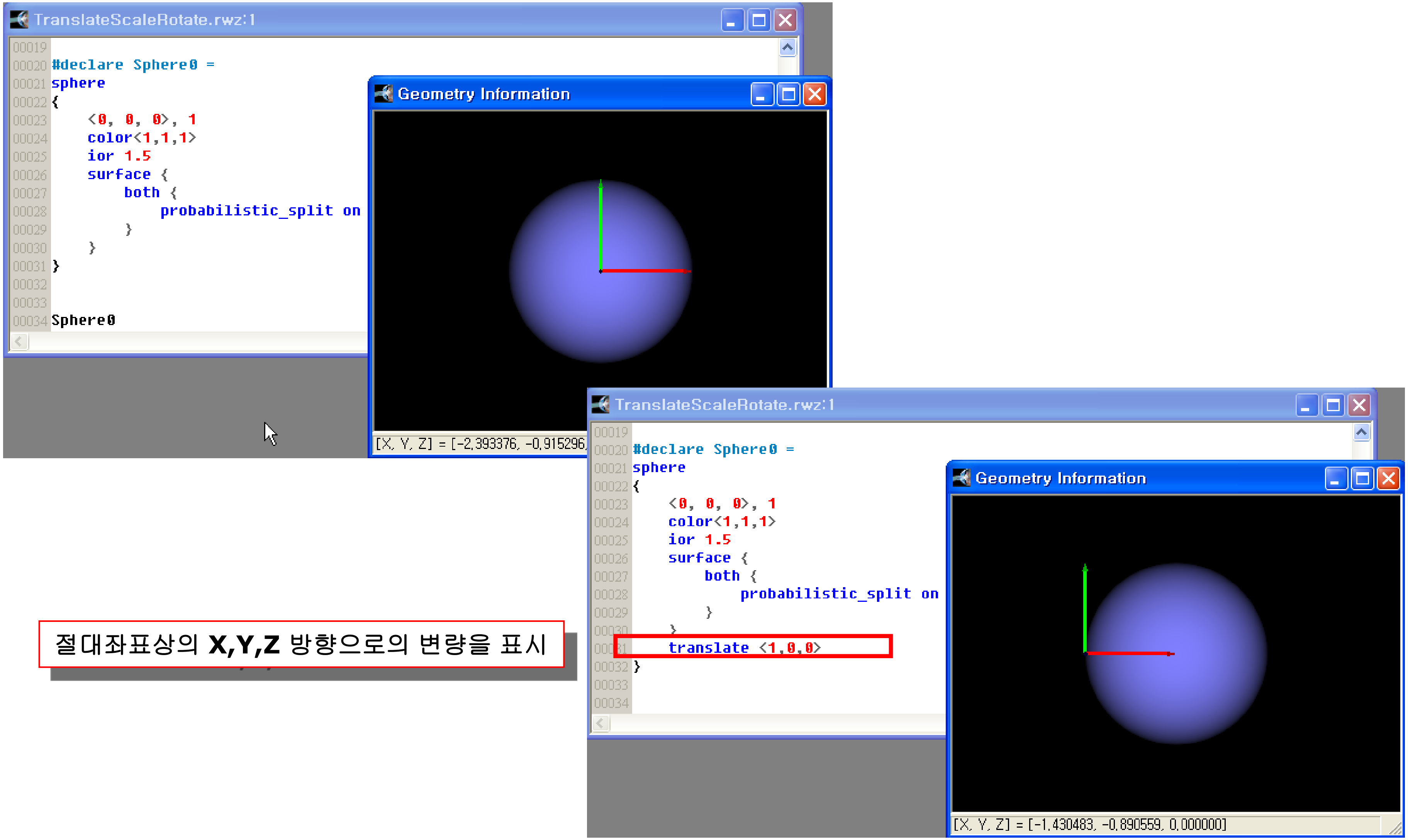


Workshop 1: Make Objects

1. **RayWiz**를 실행하고 새 파일을 만들어 작업 **Directory**에 **"Workshop1.rwz"**로 저장하시오.
2. **Menu**나 **Toolbar**를 이용하여 **Sphere**와 **Cone** 그리고 **Cylinder**를 작성해 보시오.
3. **Geometry View**를 실행하여 만약 **object**들이 만나는 영역이 존재한다면 **Script**를 적절히 수정하여 만나지 않도록 만드시오.
4. **Script**를 수정하여 **Sphere**는 붉은색, **Cone**은 초록색, **Cylinder**는 파란색으로 수정하시오.

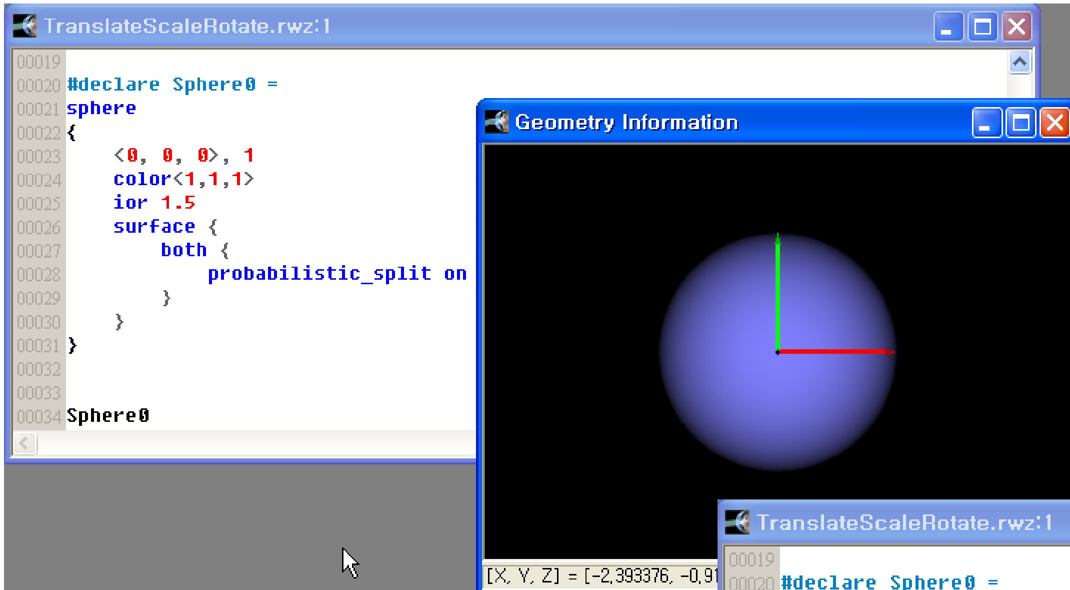


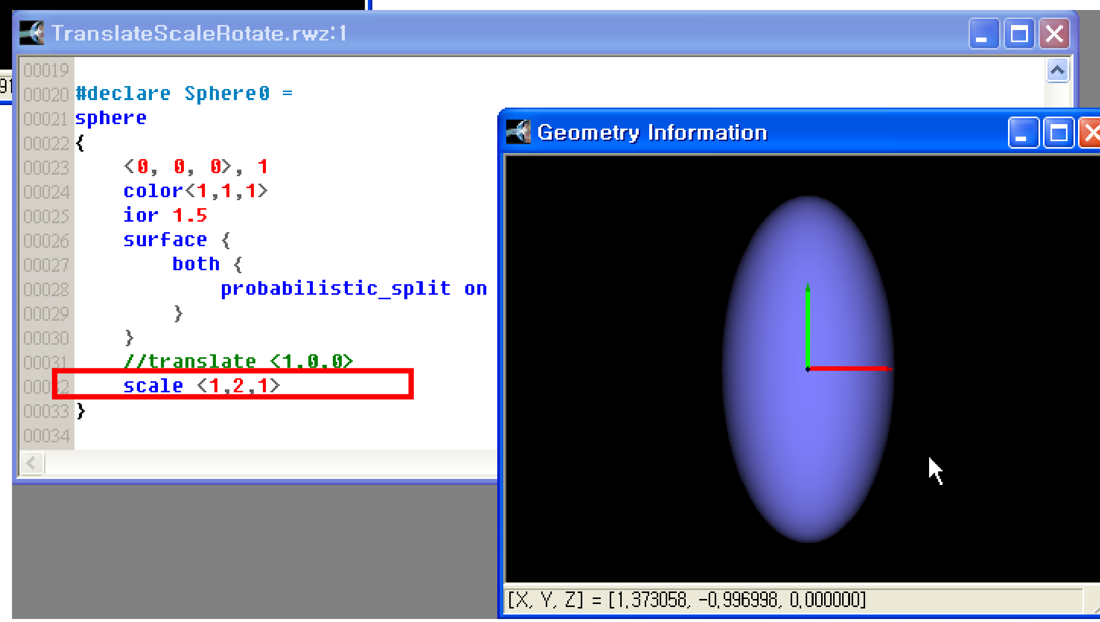
Geometry I: Translate



절대좌표상의 **X,Y,Z** 방향으로의 변량을 표시

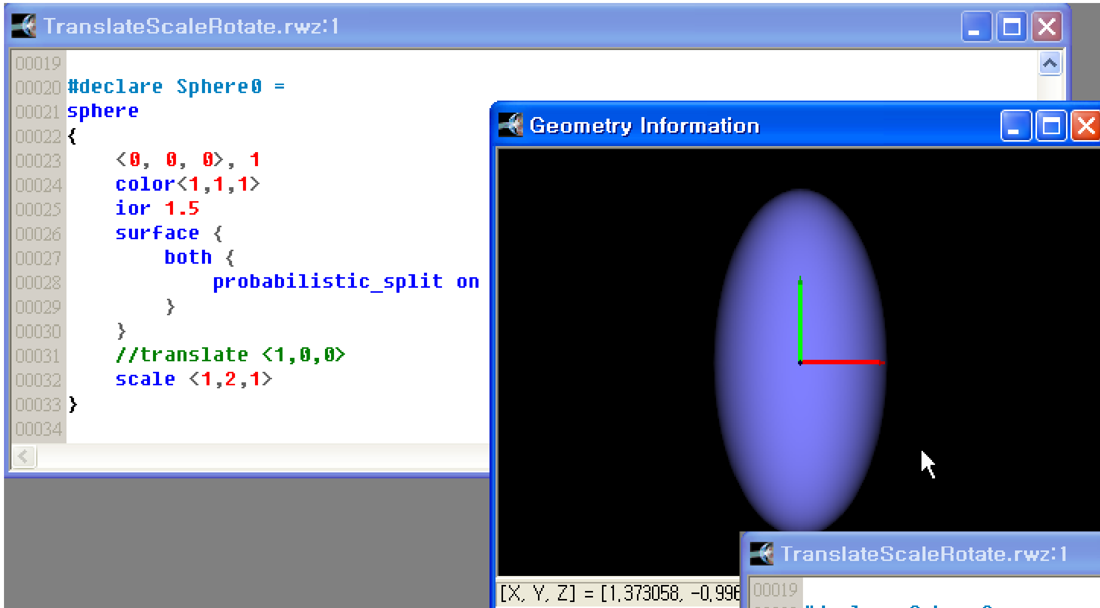
Geometry I: Scale

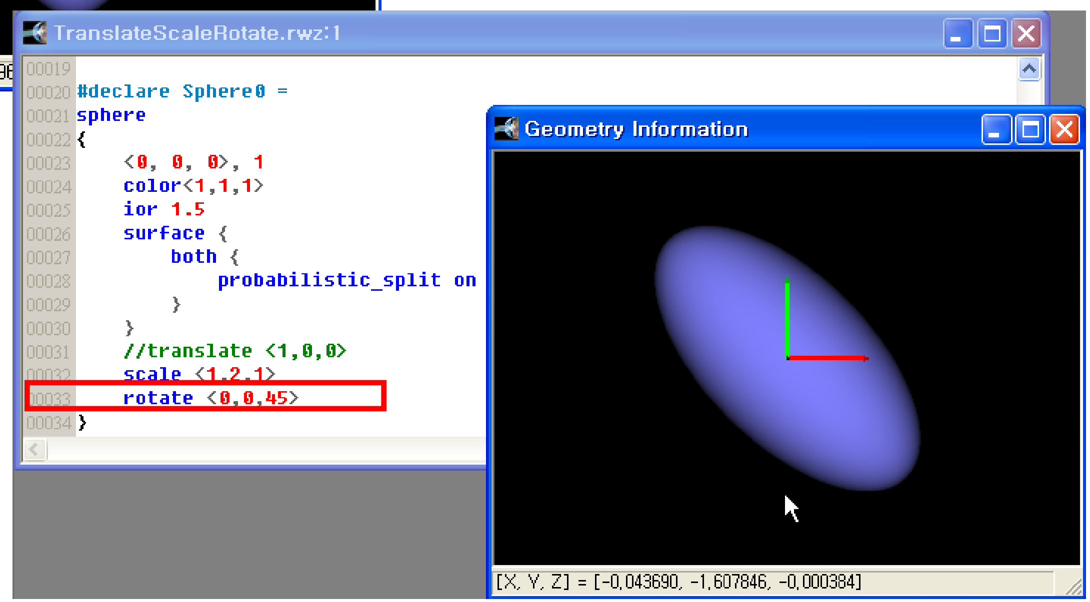




절대좌표상의 **X,Y,Z** 방향으로 **X**와 **Z**축 방향으로는 원래 크기대로 **Y**축 방향으론 **2**배로 변형하여 타원체 형성
← 원점기준

Geometry I: Rotate

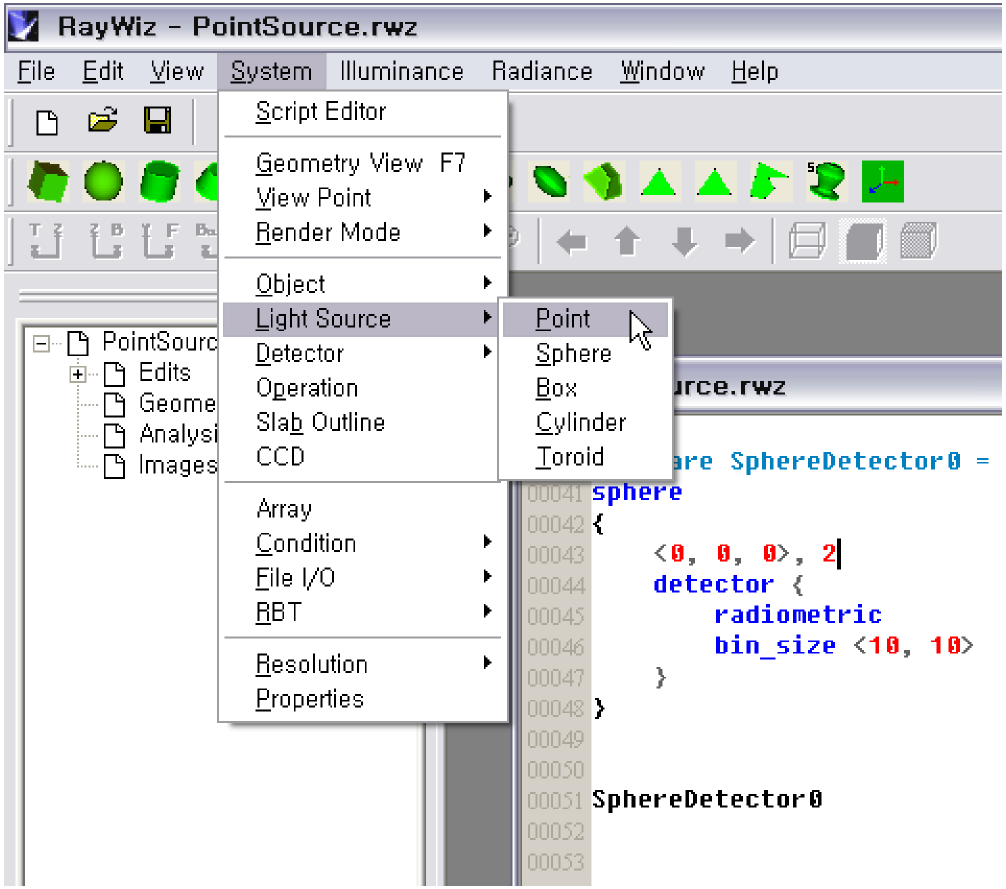


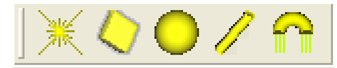


절대좌표상의 **X,Y,Z** 각 축을 잡고 원점을 기준으로 **object**를 회전 변환함.

RAYWIZ BASIC
- Light Source I -

Light Source I: Light Sources

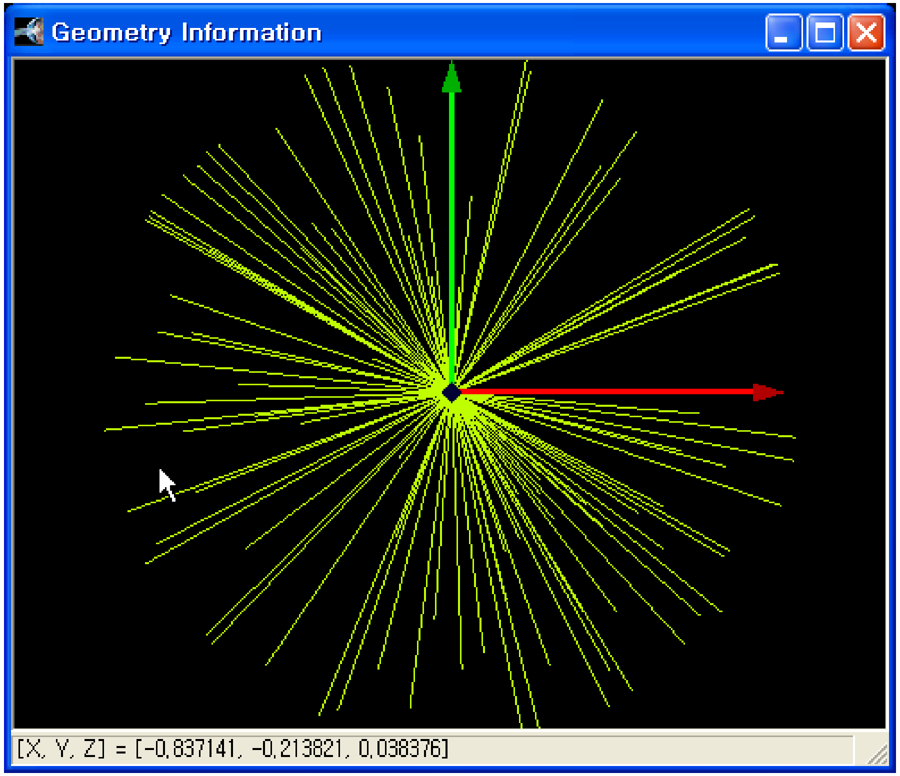


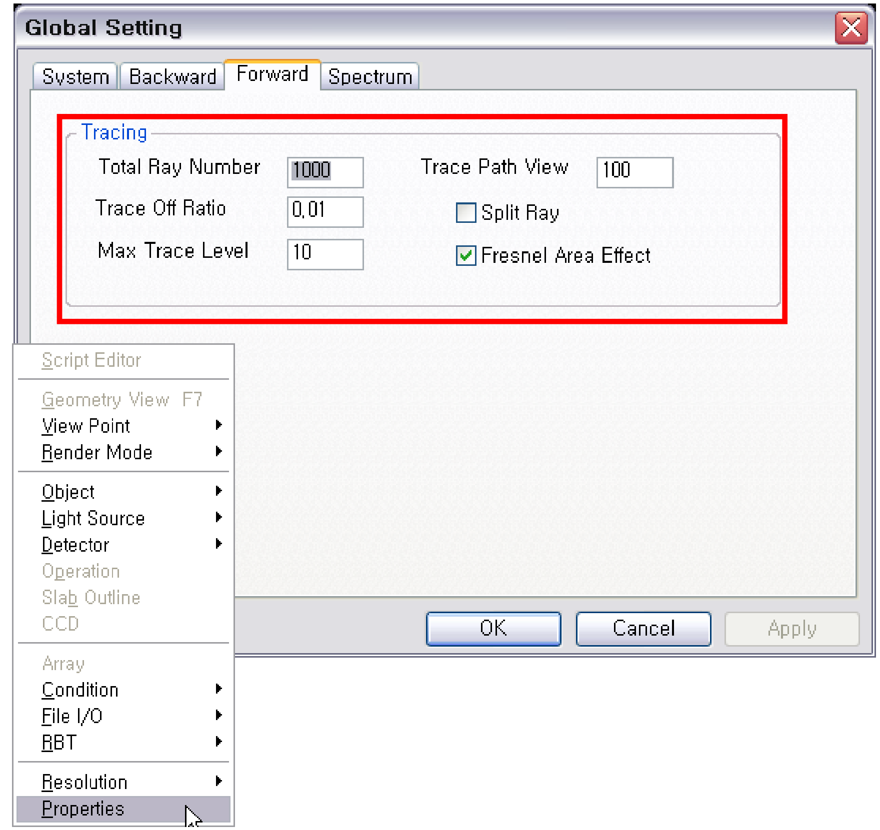


1. **Point Source**
2. **Sphere Source**
3. **Box Source**
4. **Cylinder Source**
5. **Toroid Source**
6. **CSG Source**

Light Source는 **Ray** 들의 집합체로 **Object**의 형태로 분포를 하는 것으로 **Light**의 실제 **Surface**란 존재하지 않는다.
또한 **System**의 어떠한 **Object**에도 위치 될 수 있으며 **LED Die**와 같이 **Material** 안에 위치 시킬 경우에도 **Light**에서 출발한 **Ray**는 특별한 설정 없이 물질 안에서 출발 했음을 전제로 **Simulation**을 수행한다.

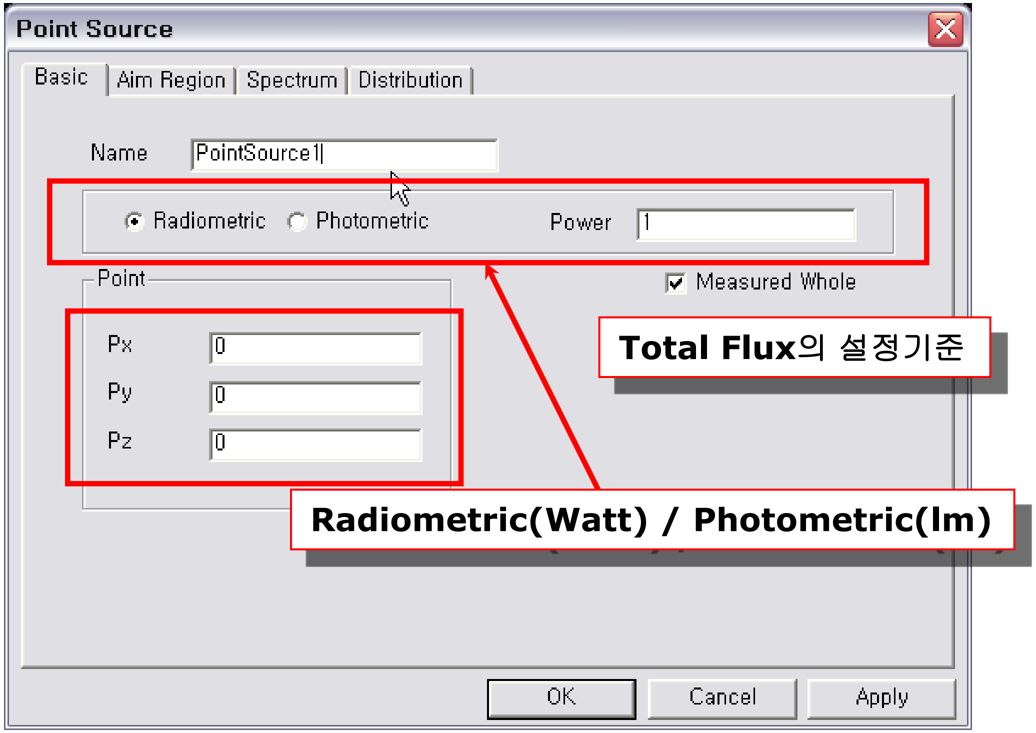
Light Source I: Point Source(1)

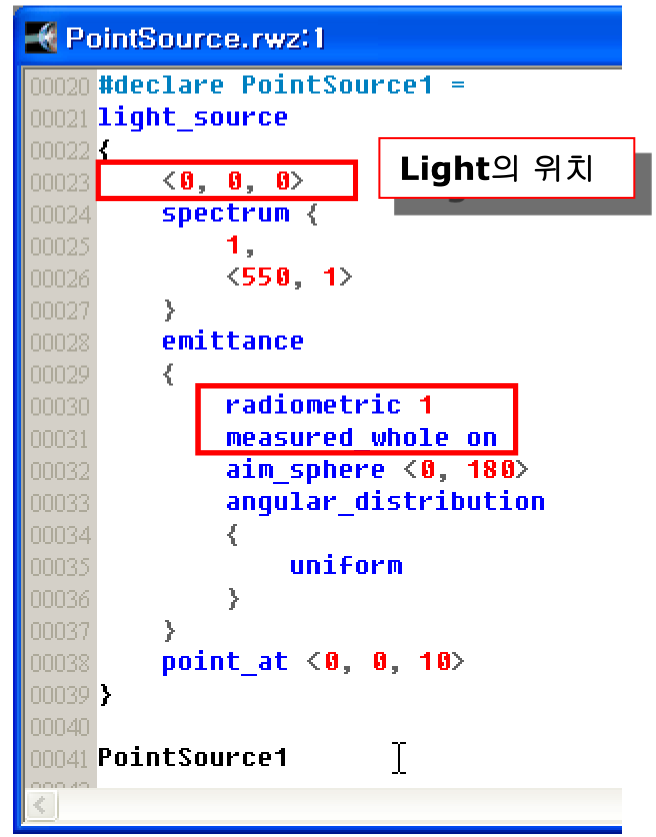




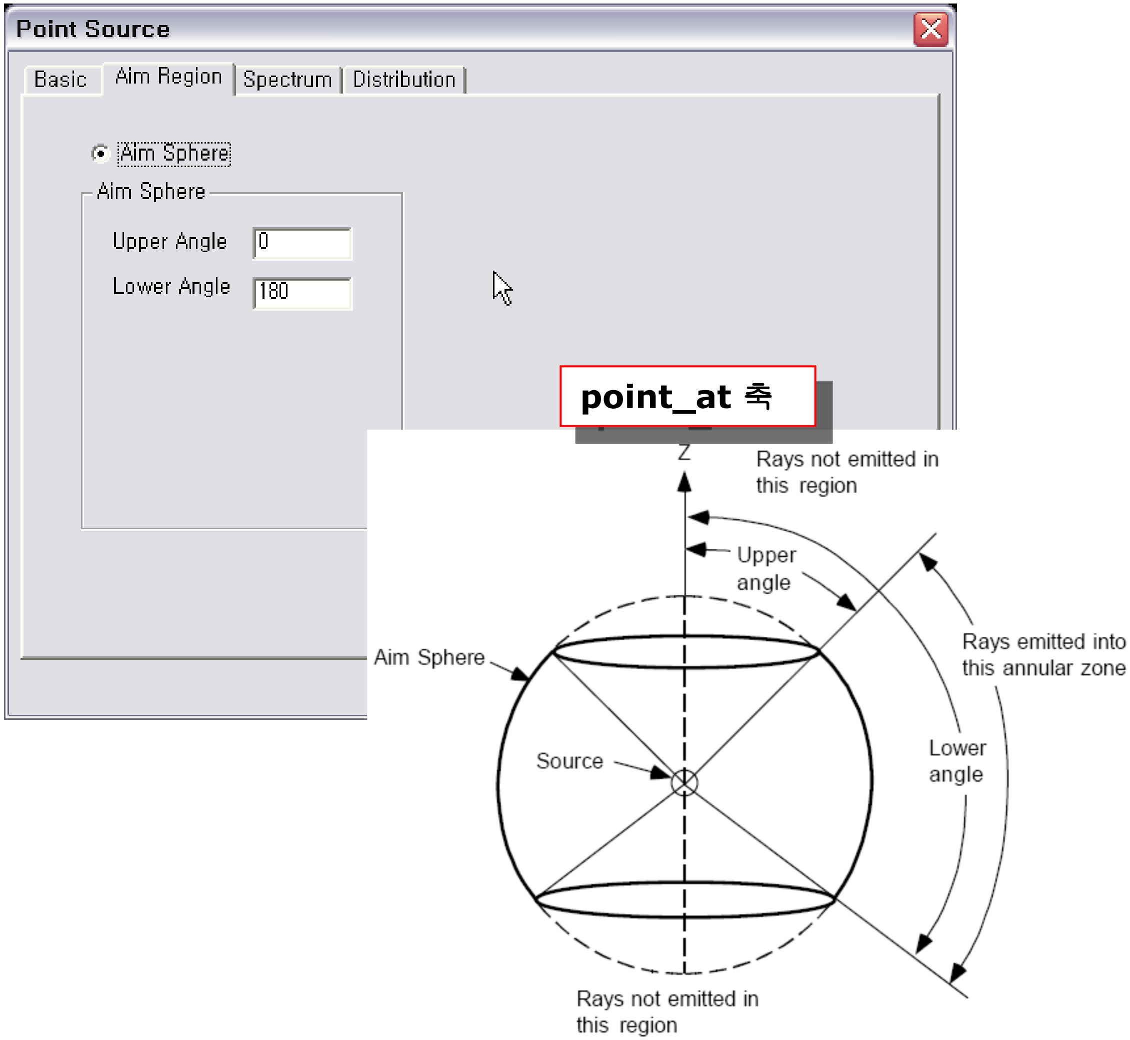
- 크기도 없고 보이지도 않는 일반적인 광원(볼륨이 없는 광원)
- 모든 방향을 향해 빛을 보낼 수도 있고 각도를 지정하여 제한영역을 주어 빛을 보내는 것도 가능하다.
- 사용자가 직접 광선의 방향과 세기를 지정할 수 있다.

Light Source I: Point Source(2)





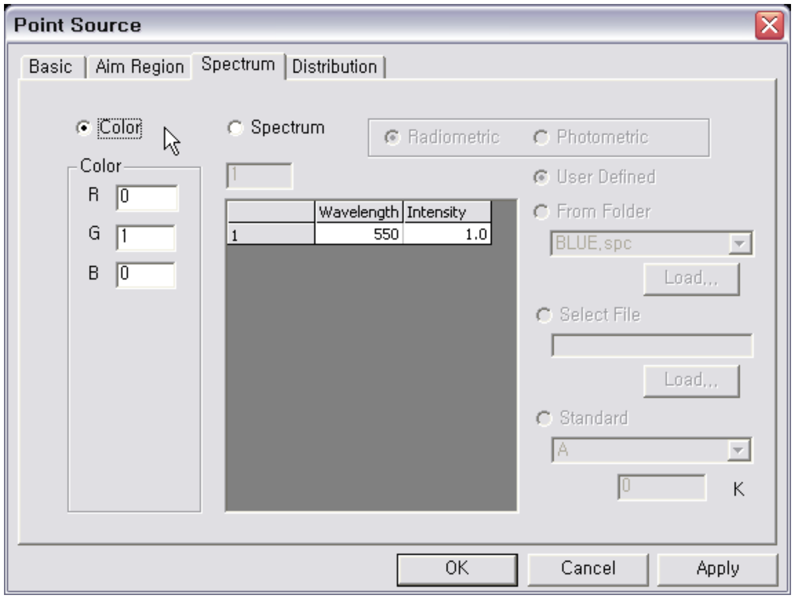
Light Source I: Point Source(3)

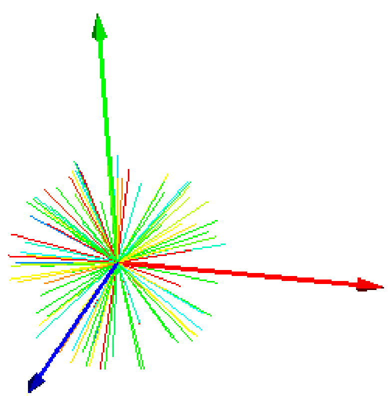


PointSource.rwz:1

```
#declare PointSource1 =
light_source
{
    <0, 0, 0>
    spectrum {
        1,
        <550, 1>
    }
    emittance
    {
        radiometric 1
        measured whole on
        aim sphere <0, 180>
        angular_distribution
        {
            uniform
        }
    }
    point_at <0, 0, 10>
}

PointSource1
```

Light Source I: Point Source(4)





Color <0, 1, 0>에 해당하는 **Spectrum**으로 변환

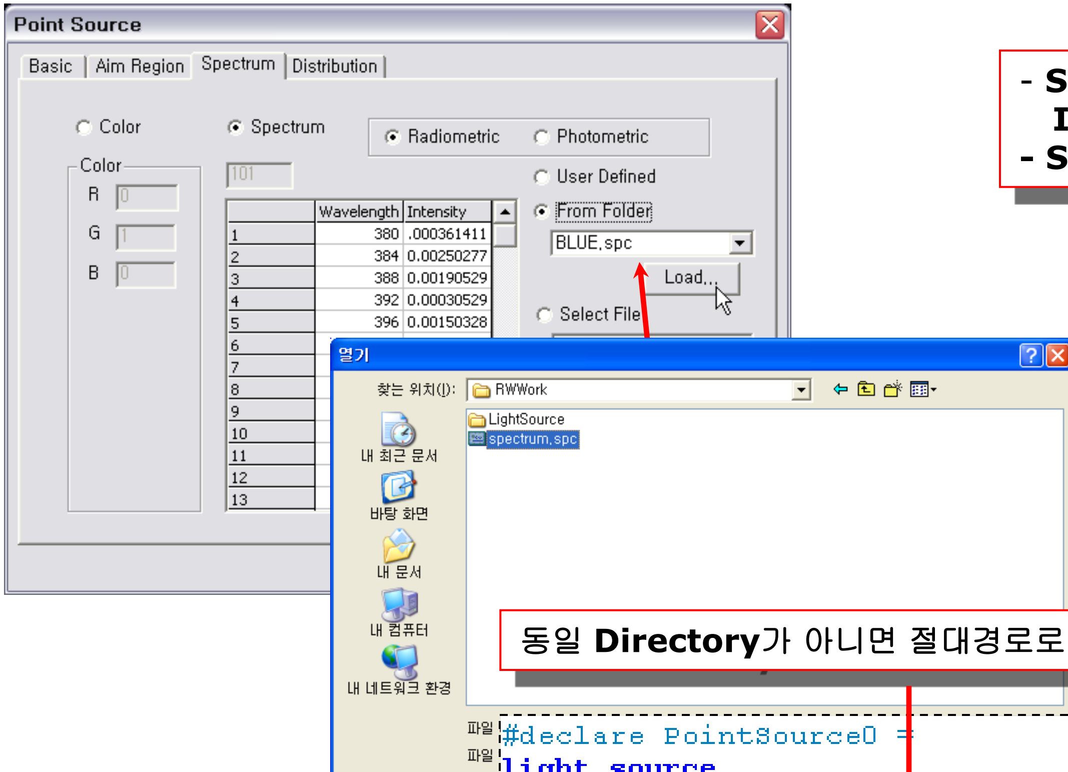
- **Ray의 Spectrum을 RGB로 입력가능**
- **Illuminance Ray Tracing 시, RGB Color는 Spectrum으로 변환.**

PointSource.rwz

```
}

#declare PointSource0 =
light_source
{
    <0, 0, 0>
    color <0, 1, 0>
    emittance
    {
        radiometric 1
        measured_whole on
        aim_sphere <0, 180>
        angular_distribution
        {
            lambertian
        }
    }
    point_at <0, 0, 10>
}


PointSource0
```

Light Source I: Point Source(5)

- **Spectrum**을 파일로 읽어서 광선의 파장 별 **Intensity**를 지정할 수 있다.
- **Spectrum** 을 **Grid**에 직접 입력 할 수도 있다.



동일 **Directory**가 아니면 절대경로로 표시

```
#declare PointSource0 =
light_source
{
    <0, 0, 0>
    spectrum {
        1,
        <550, 1.5>
    }
    emittance
    {
        radiometric 1
```

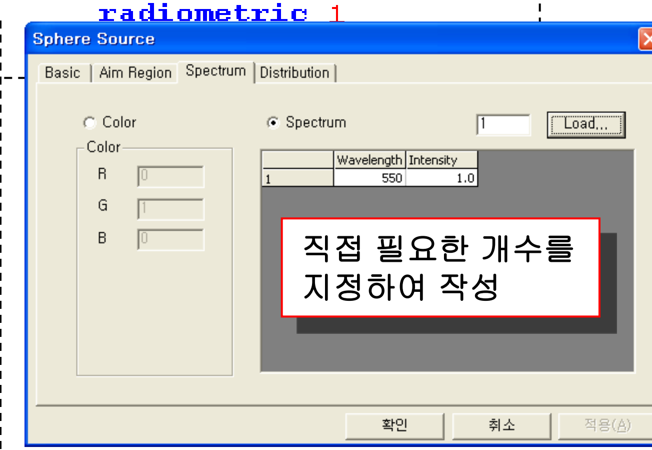
spectrum.spc

```
1 191
2 380 9.80E+00
3 382 1.02E+01
4 384 1.07E+01
5 386 1.11E+01
6 388 1.16E+01
```

데이터 수

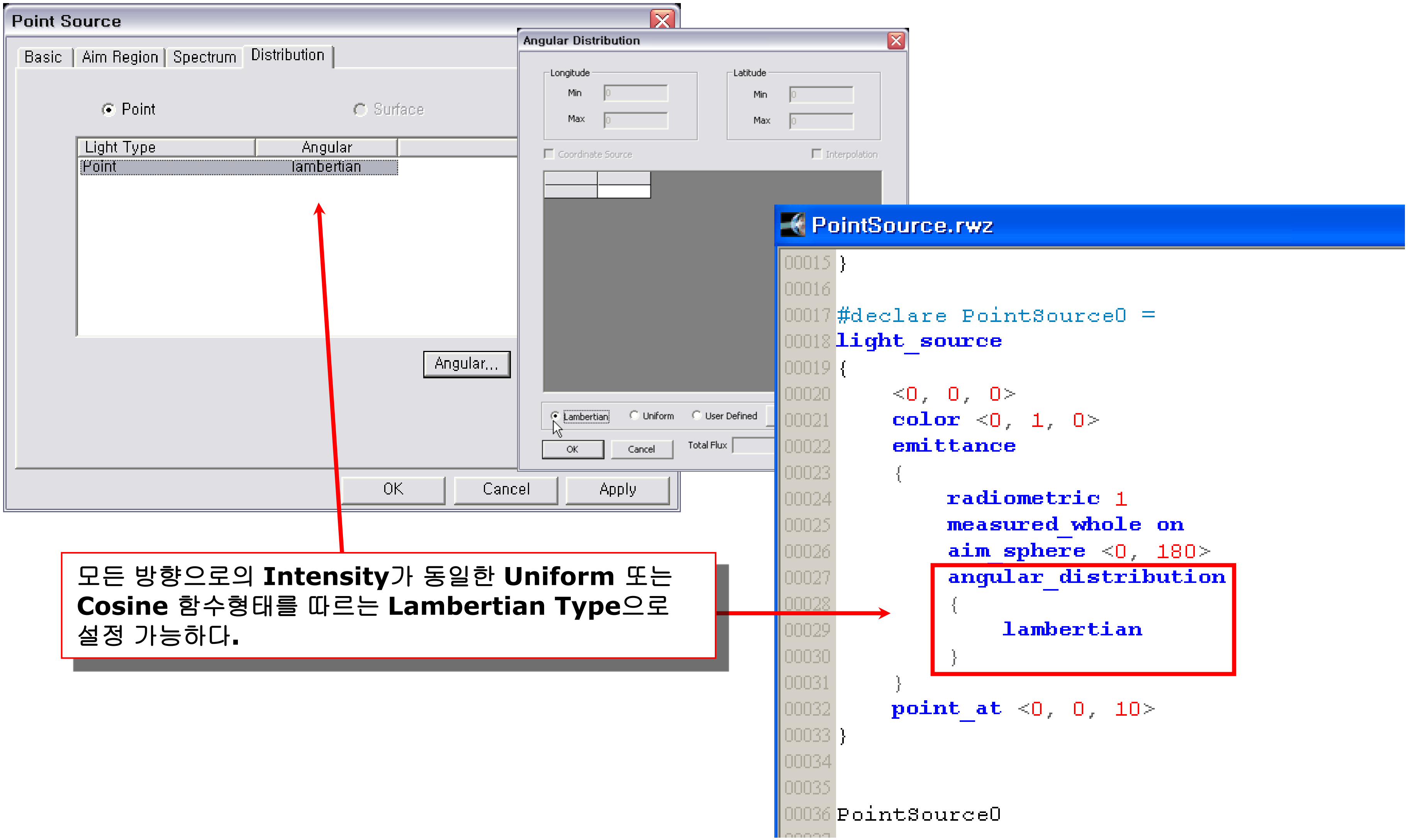
파장(nm) **Intensity**

```
#declare PointSource0 =
light_source
{
    <0, 0, 0>
    spectrum {
        "C:\RWWork\spectrum.spc"
    }
    emittance
    {
        radiometric 1
        measured_whole on
        aim_sphere <0, 180>
```



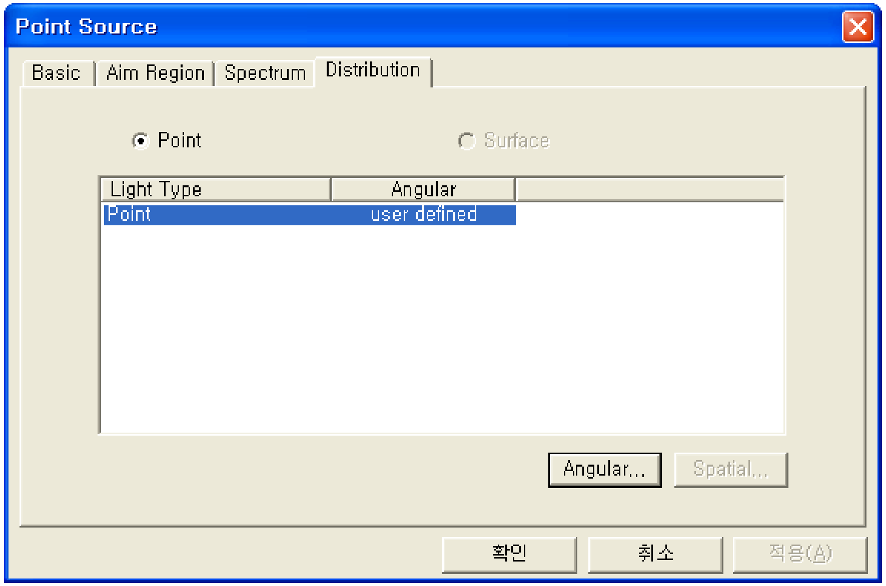
직접 필요한 개수를 지정하여 작성

Light Source I: Point Source(6)



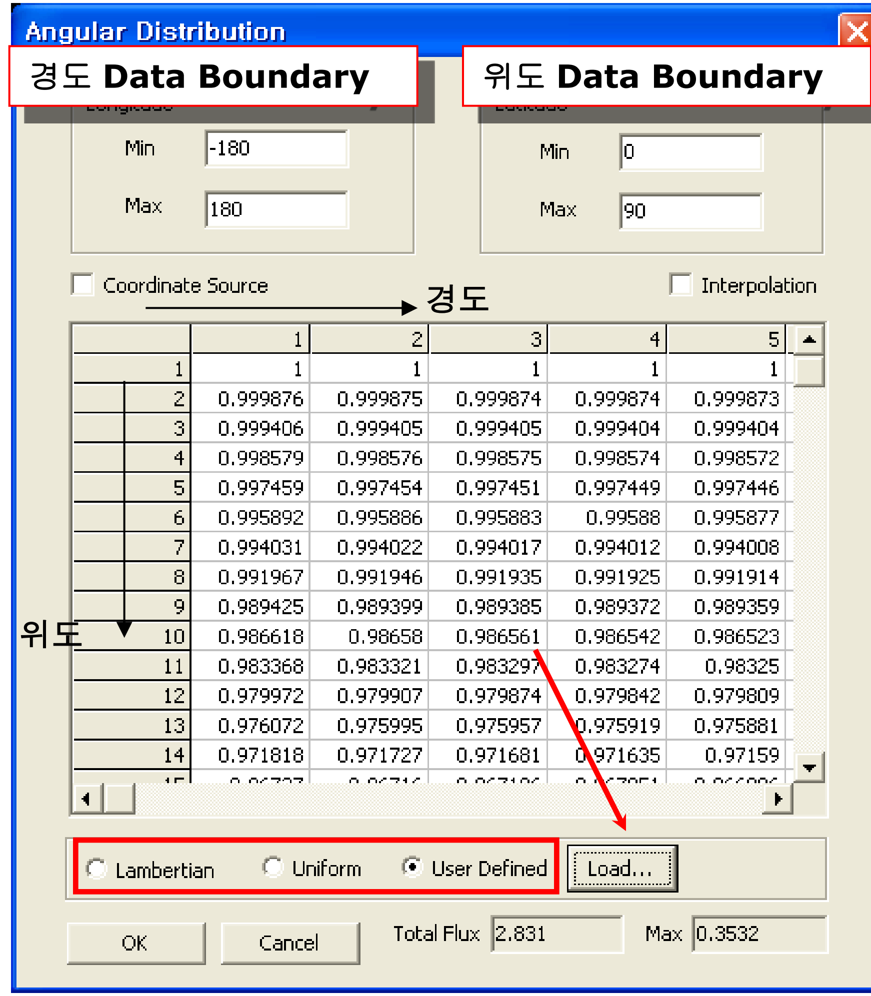
모든 방향으로의 **Intensity**가 동일한 **Uniform** 또는 **Cosine** 함수형태를 따르는 **Lambertian Type**으로 설정 가능하다.

Light Source I: Point Source(7)



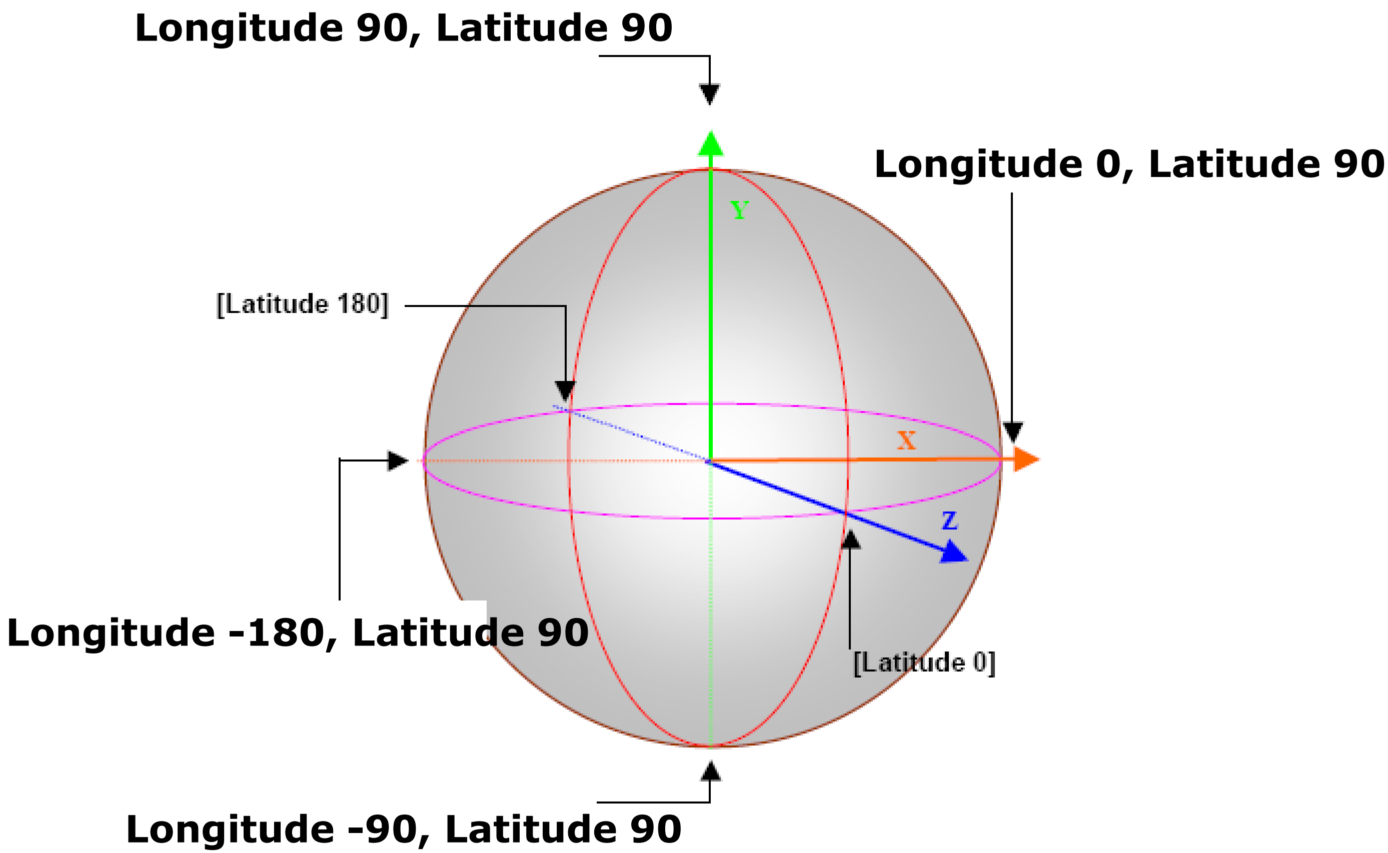

```
emittance
{
    radiometric 1
    measured_whole on
    aim_sphere <0, 180>
    angular_distribution
    {
        "C:\RWWork\angular_apodization.apo"
        interpolation off
        coordinate_source off
    }
}
point at <0, 0, 10>
```

Apodize 파일 경로 표시



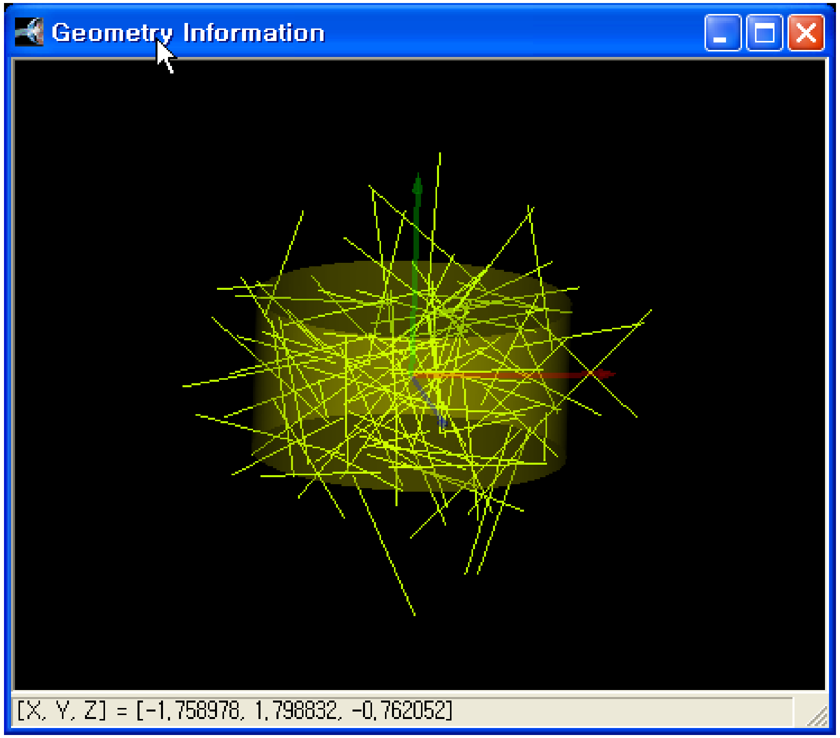
	1	2	3	4	5
1	1	1	1	1	1
2	0.999876	0.999875	0.999874	0.999874	0.999873
3	0.999406	0.999405	0.999405	0.999404	0.999404
4	0.998579	0.998576	0.998575	0.998574	0.998572
5	0.997459	0.997454	0.997451	0.997449	0.997446
6	0.995892	0.995886	0.995883	0.99588	0.995877
7	0.994031	0.994022	0.994017	0.994012	0.994008
8	0.991967	0.991946	0.991935	0.991925	0.991914
9	0.989425	0.989399	0.989385	0.989372	0.989359
10	0.986618	0.98658	0.986561	0.986542	0.986523
11	0.983368	0.983321	0.983297	0.983274	0.98325
12	0.979972	0.979907	0.979874	0.979842	0.979809
13	0.976072	0.975995	0.975957	0.975919	0.975881
14	0.971818	0.971727	0.971681	0.971635	0.97159

Light Source I: Point Source(8)



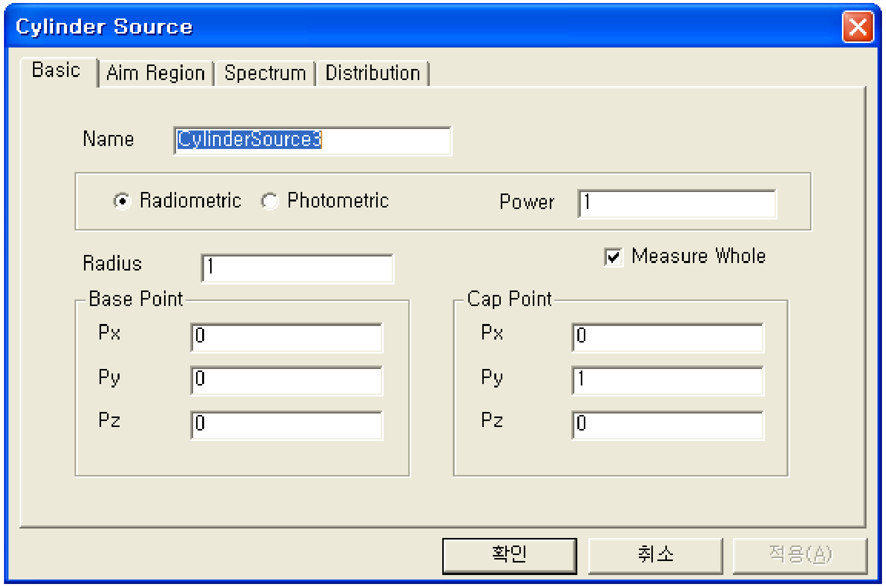
Point_at 축을 Z 기준으로 했을 때 경도는 –X에서 -180도로 시작하여 180 도로 설정되며, 위도는 Z축을 0도로 시작하여 –Z축을 180 도로 설정된다.

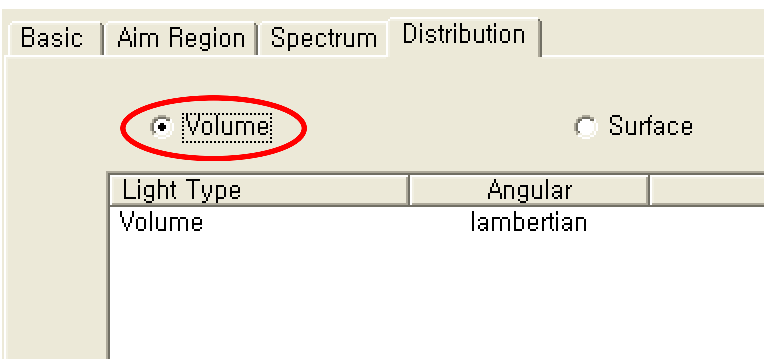
Light Source I: Cylinder Source - Volume(1)



- 원기둥의 크기를 가지는 볼륨 광원
- 원기둥의 내부영역에서부터 광선을 내보낸다.
- 모든 방향을 향해 빛을 보낼 수도 있고 각도를 지정하여 제한영역을 주어 빛을 보내는 것도 가능하다.
- 사용자가 직접 광선의 방향과 세기를 지정할 수 있다.

Light Source I: Cylinder Source - Volume(2)

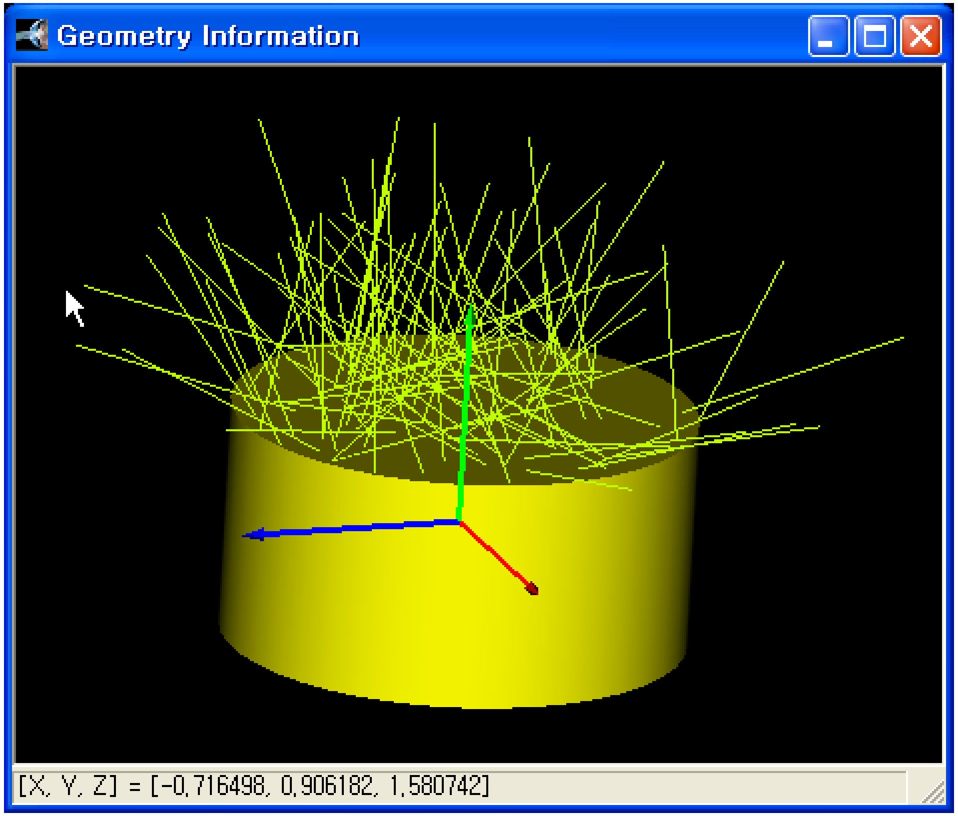





```
#declare CylinderSource0 =
light_source
{
    <0, 0, 0>
    spectrum {
        1,
        <550, 1>
    }
    object_light
    {
        volume
        cylinder
        {
            <0, 0, 0>, <0, 1, 0>, 1
        }
    }
    emittance
    {
        radiometric 1
        measured_whole on
        aim_sphere <0, 180>
        angular_distribution
        {
            uniform
        }
        volume_distribution
        {
            uniform
        }
    }
    point_at <0, 2, 0>
}

CylinderSource0
```

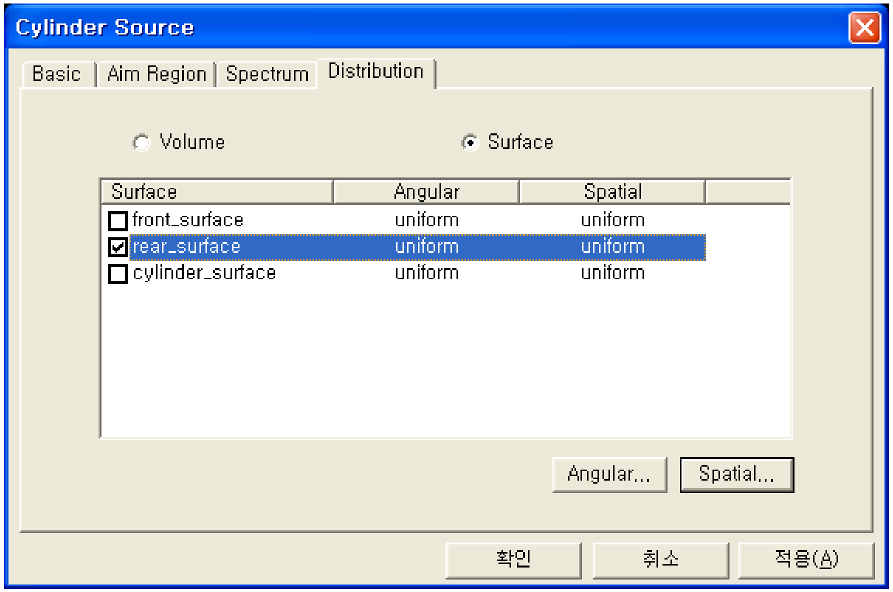
광원의 형태를 **Cylinder**로 정의한다
(아랫면/윗면의 꼭지점, 반지름**)**

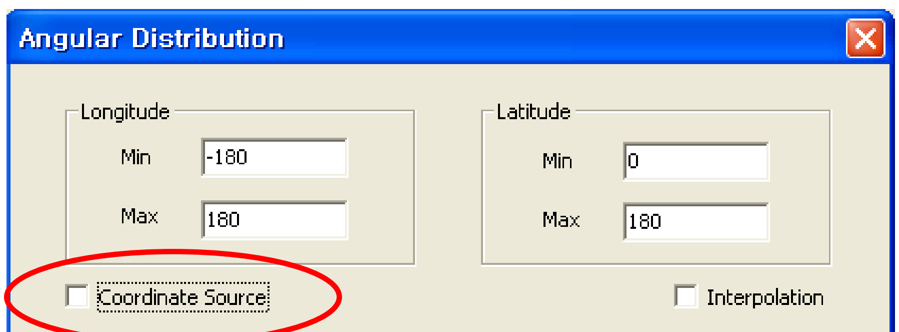
Light Source I: Cylinder Source - Surface(3)

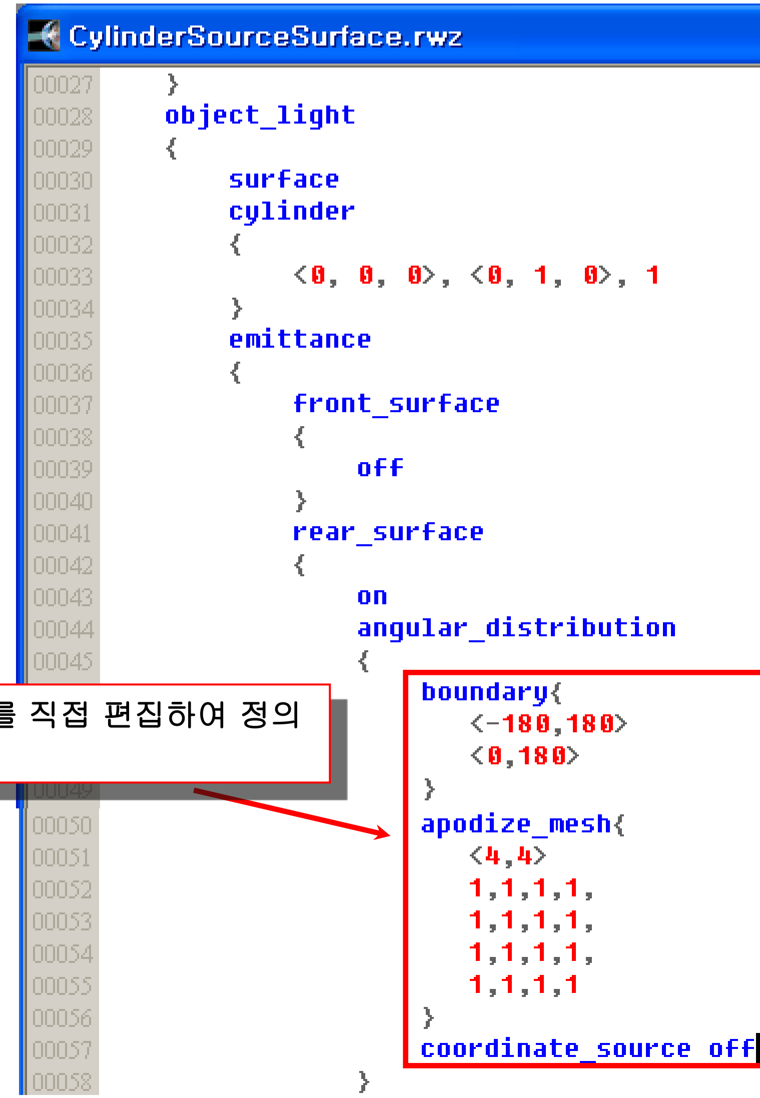


- 원기둥 면에서부터 빛이 나가는 광원
- **3**개의 면마다 빛이 나가거나 안 나가게 할 수 있다**.**
- **3**개의 면 각각에 광선의 방향과 세기를 지정할 수 있다**.**

Light Source I: Cylinder Source - Surface(4)





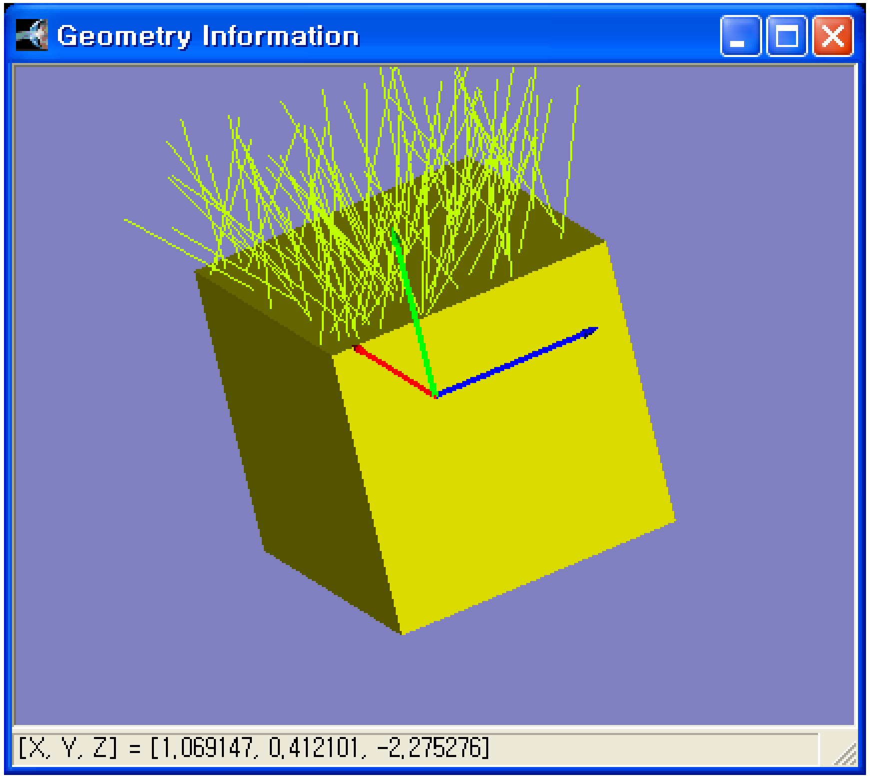


Script를 직접 편집하여 정의 가능

기본값은 선택되어 있지 않으면 각도 측정의 중심점을 **Surface**의 중심으로 하며 선택하게 되면 **Object**의 중심을 기준으로 한다.

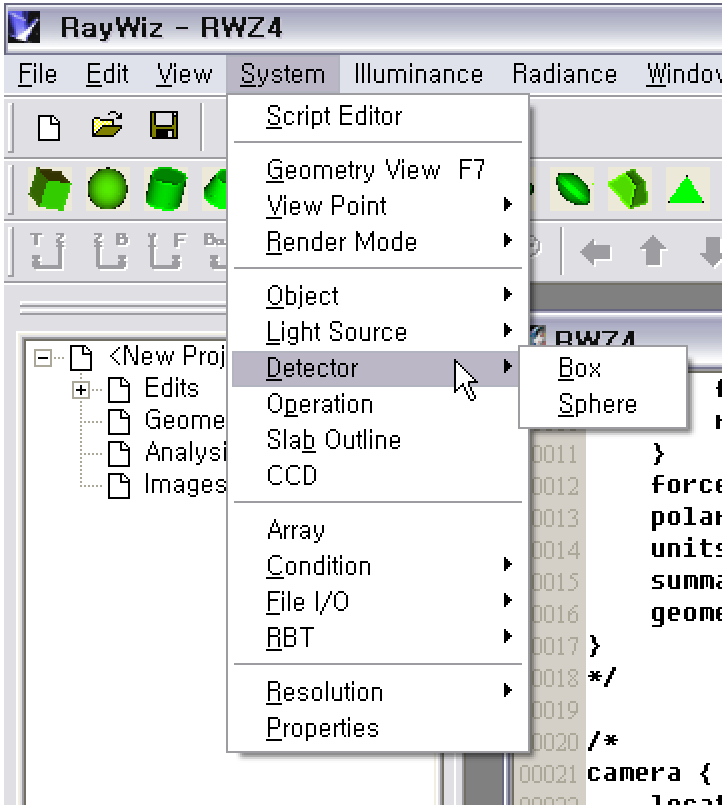
Workshop 2: Make Light

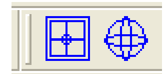
1. **RayWiz**를 실행하고 새 파일을 만들어 작업 **Directory**에 **"Workshop2.rwz"**로 저장하시오.
2. **Menu**나 **Toolbar**를 이용하여 **Surface Box Light**를 작성하되 **+Y(Top_Surface)**로만 나가도록 설정하며 **aim_angle**을 **0**도 **45**도로 제한을 주시오.
3. **Geometry View**를 실행하여 **Illuminance** 메뉴에 있는 **>Start Simulation**을 선택하여 **Ray**들의 진행 방향을 관찰하시오



RAYWIZ BASIC
- Detector I -

Detector: Detector Types





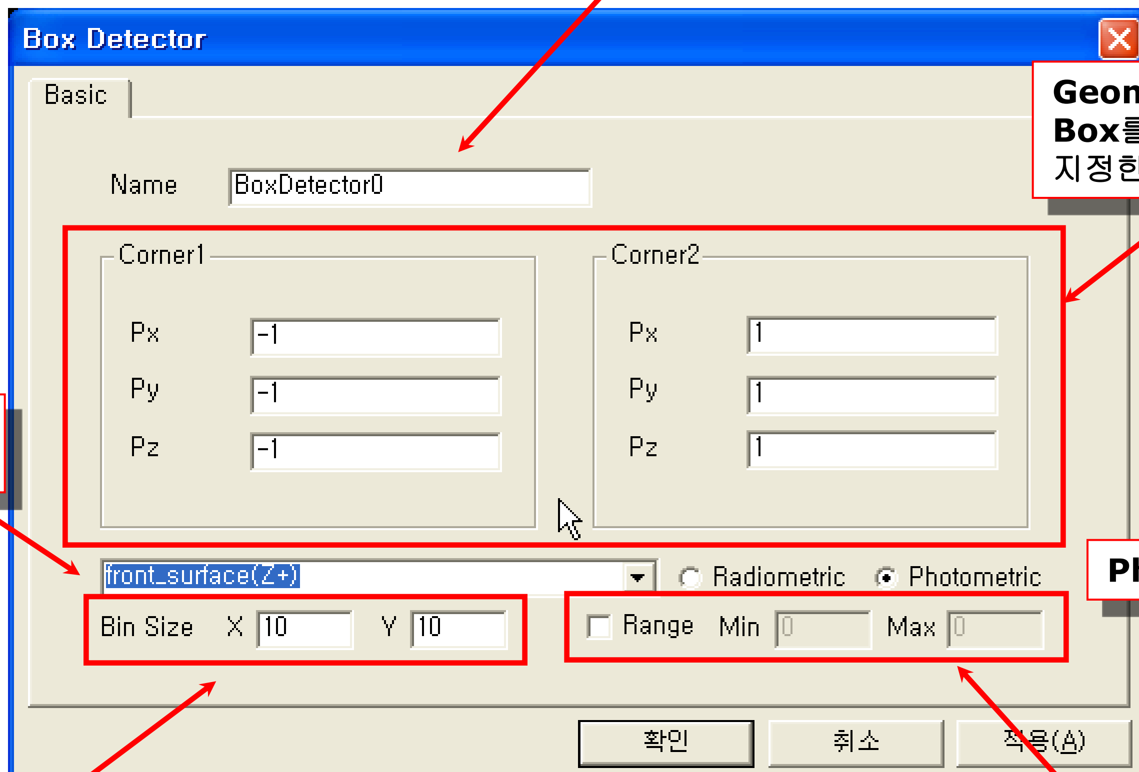
1. **Box Detector**
2. **Sphere Detector**

Total Flux를 기준으로 계산하려는 정량데이터의 종류에 단위면적이 기준이 되면 **Box Detector**를 입사각이 기준이면 **Sphere Detector**를 필요로 하게 된다.
RAYWIZ는 복수개의 **Detector**설정이 가능하다.

Detector: Box Detector(1)

Detector 이름

Geometry 설정부분으로 **Box**를 이루는 대각의 두 점을 지정한다.



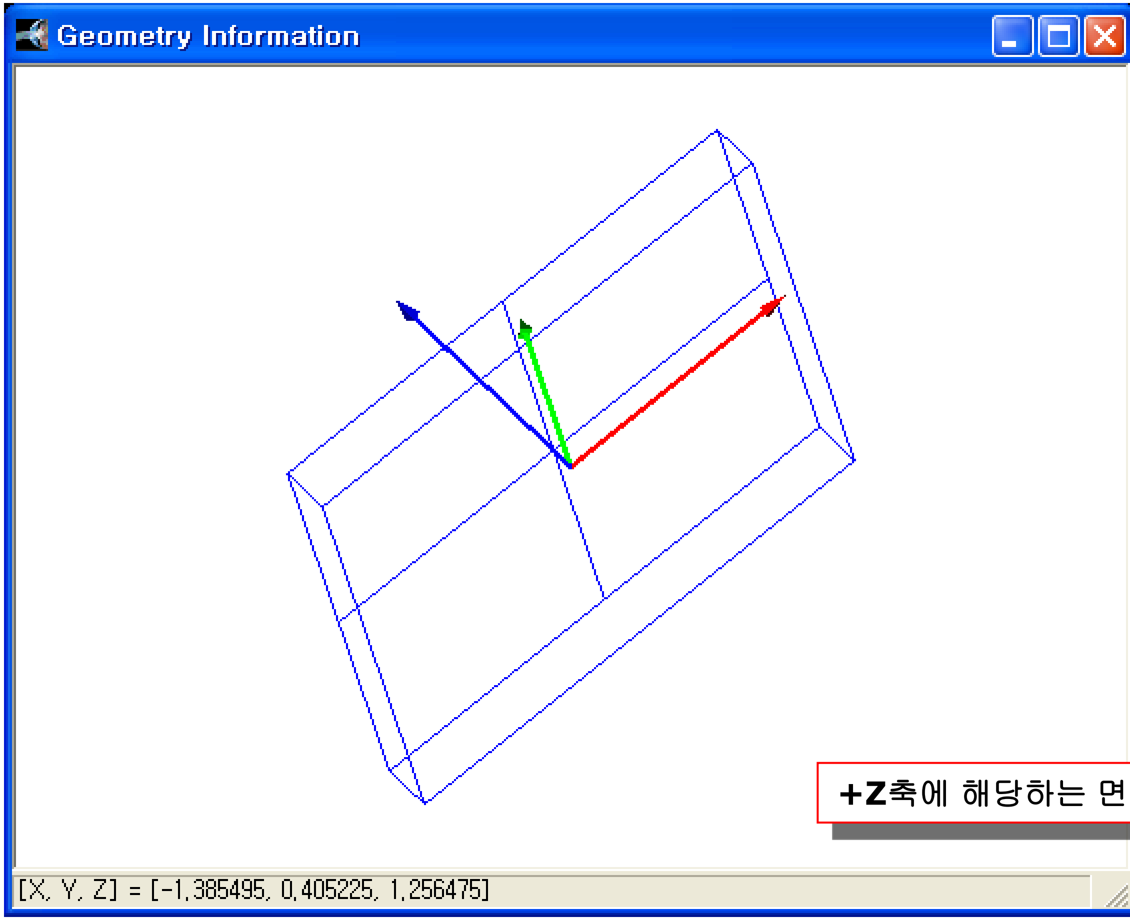
Detector로 사용될 **Surface**

Photometry 단위선택

Detecting 2D Cell 수

결과 값으로 표현할 **Data** 최대, 최소

Detector: Box Detector(2)

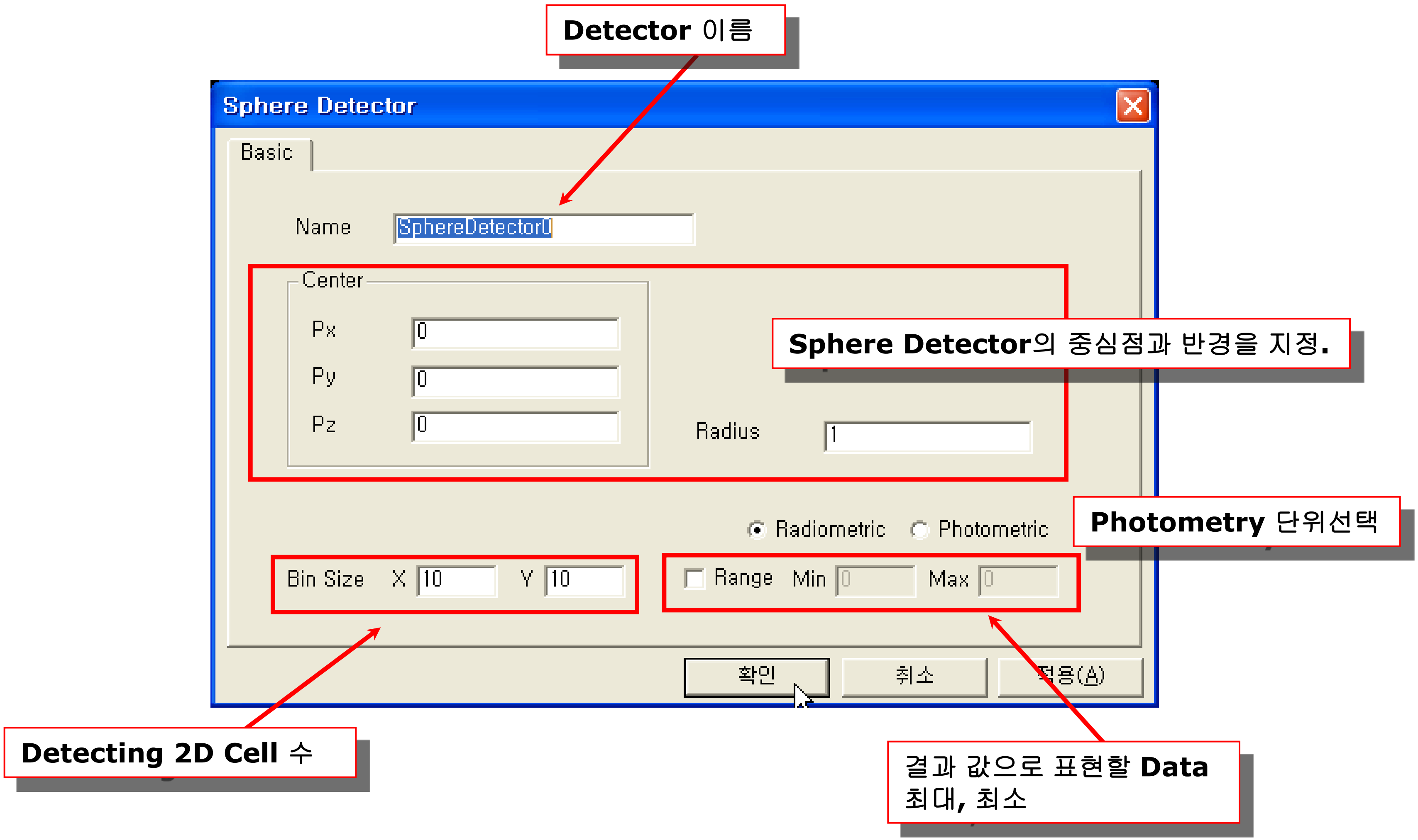


BoxDetector.rwz:1

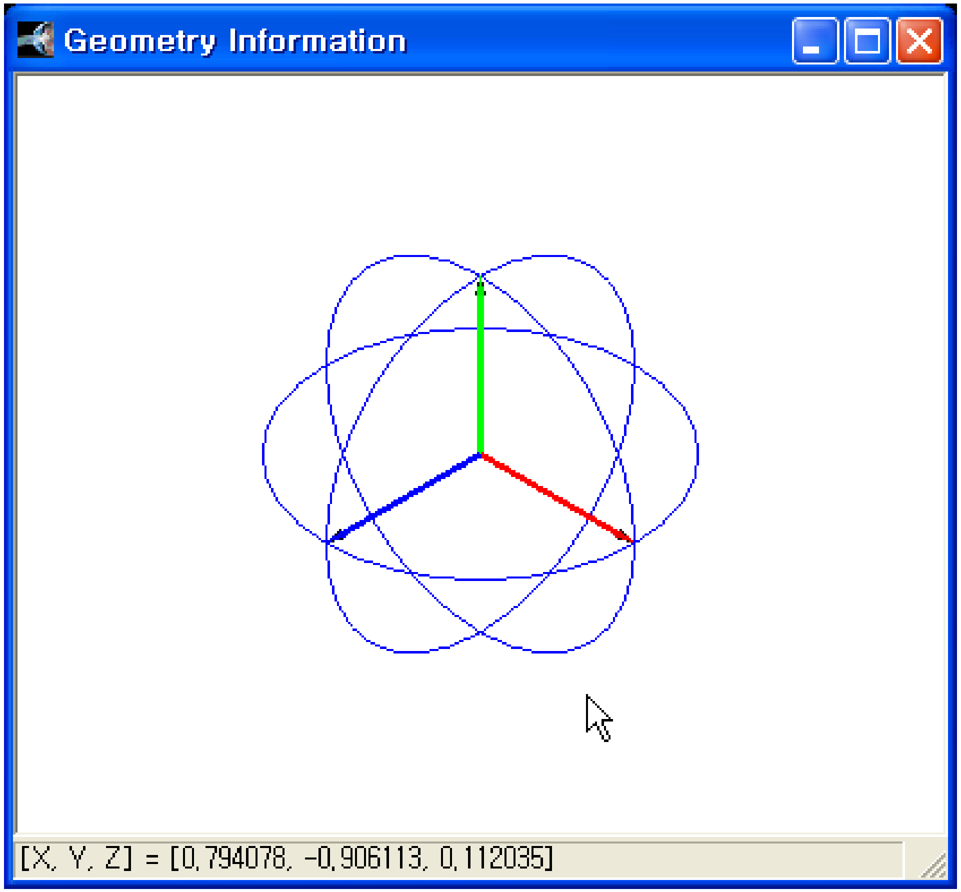
```
}
*/

#declare BoxDetector0 =
box
{
    <-1, -1, -0.1>, <1, 1, 0.1>
    detector {
        front_surface
        photometric
        bin_size <10, 10>
    }
}

BoxDetector0

```

+Z축에 해당하는 면에 십자 표시가 나타남.

Detector: Sphere Detector(1)

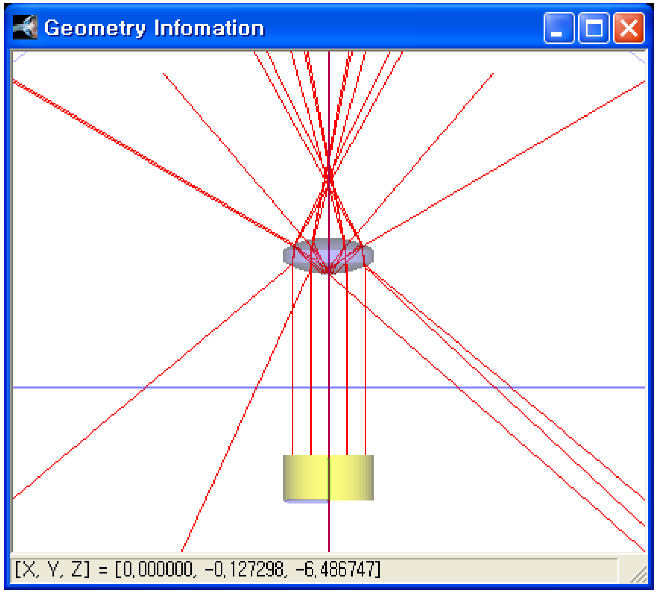


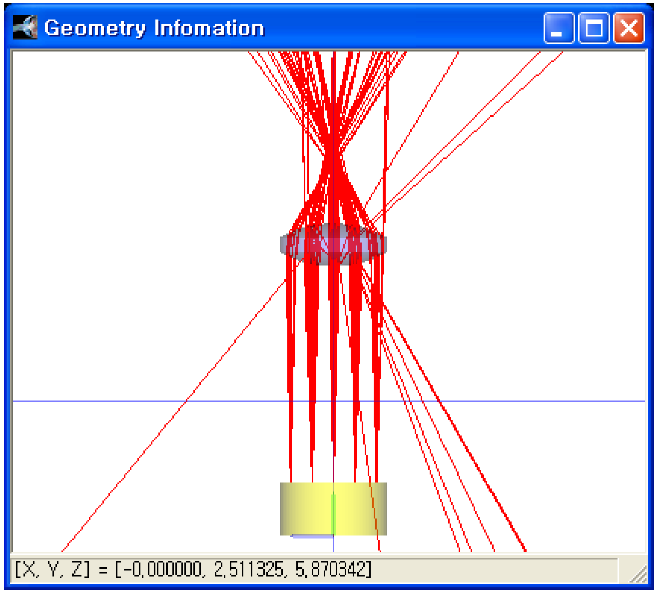
Detector: Sphere Detector(2)

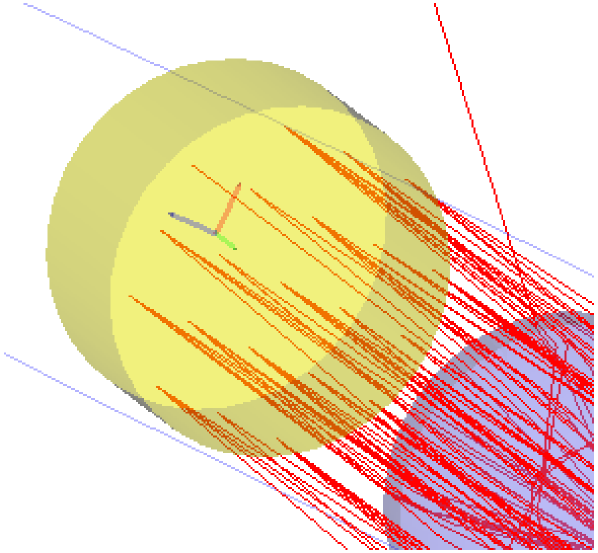



```
#declare SphereDetector0 =
sphere
{
    <0, 0, 0>, 1
    detector {
        radiometric
        bin_size <10, 10>
    }
}

SphereDetector0
```

Light Source II: Grid Light (1)







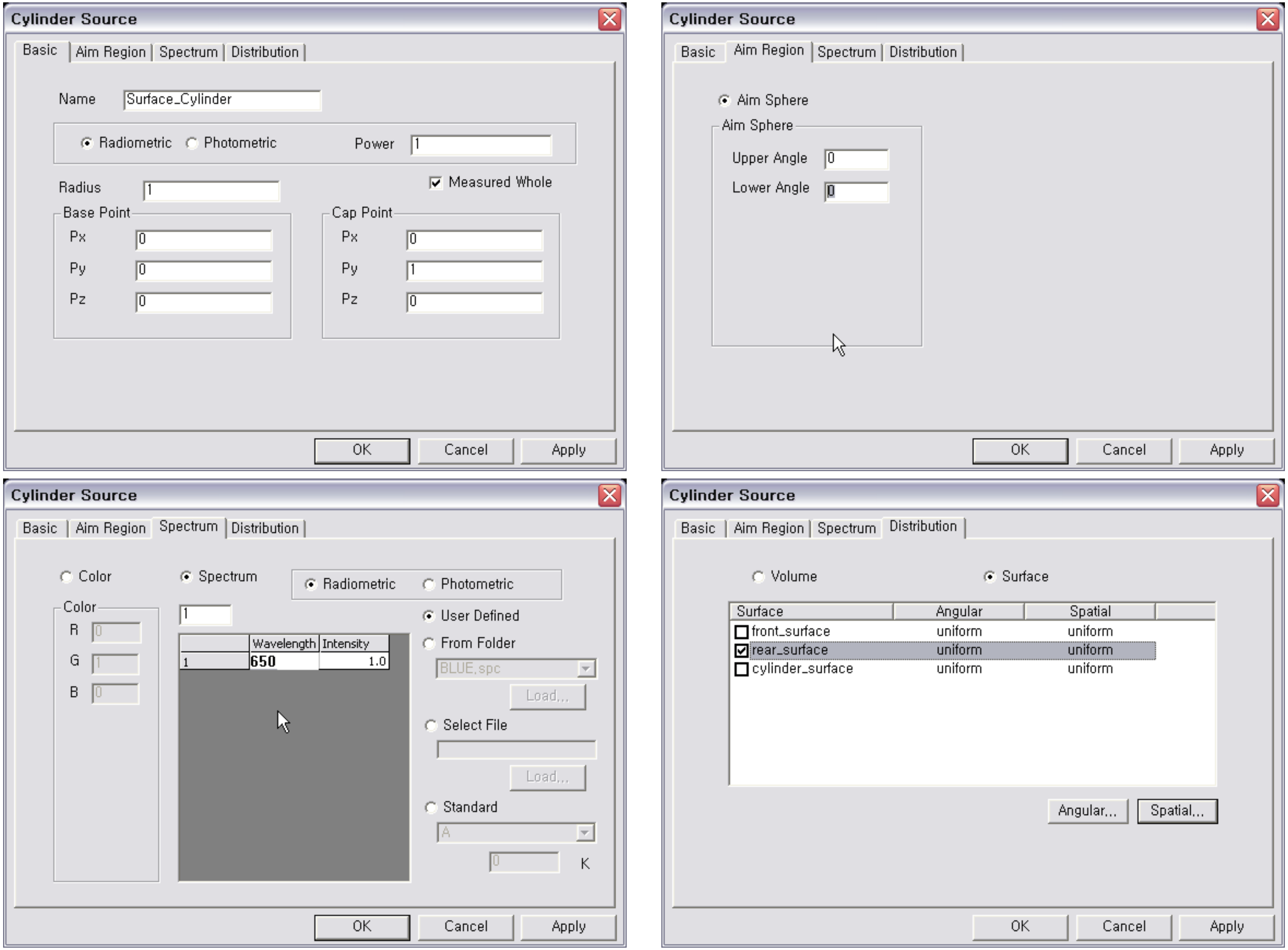
Light Source II: Grid Light (2)

Grid_Ray.rwz

```
global_settings {
    rays {
        trace_off_ratio 0.01
        total_ray_number 1000
        max_trace_level 10
        trace_path_view 100
        split_ray off
        fresnel_area_effect on
        random_spectrum on
    }
    force_parsing off
    polarization off
    units millimeters
    summary on
    geometry_intensity on
}

/*
camera {
    location <10, 10, 10>
    look_at <0, 0, 0>
}
*/
```

Comment 해제

Light Source II: Grid Light (3)



Light Source II: Grid Light (4)

Grid_Ray.rwz

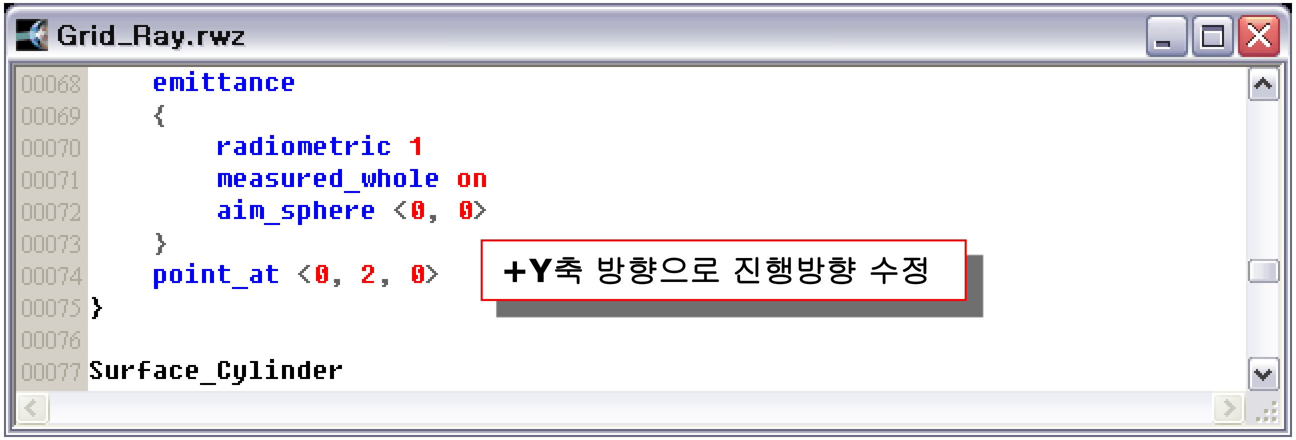
```
#declare Surface_Cylinder =
light_source
{
    <0, 0.5, 0>
    spectrum {
        discrete on
        1,
        <650, 1>
    }
    object_light
    {
        surface
        cylinder
        {
            <0, 0, 0>, <0, 1, 0>, 1
        }
```

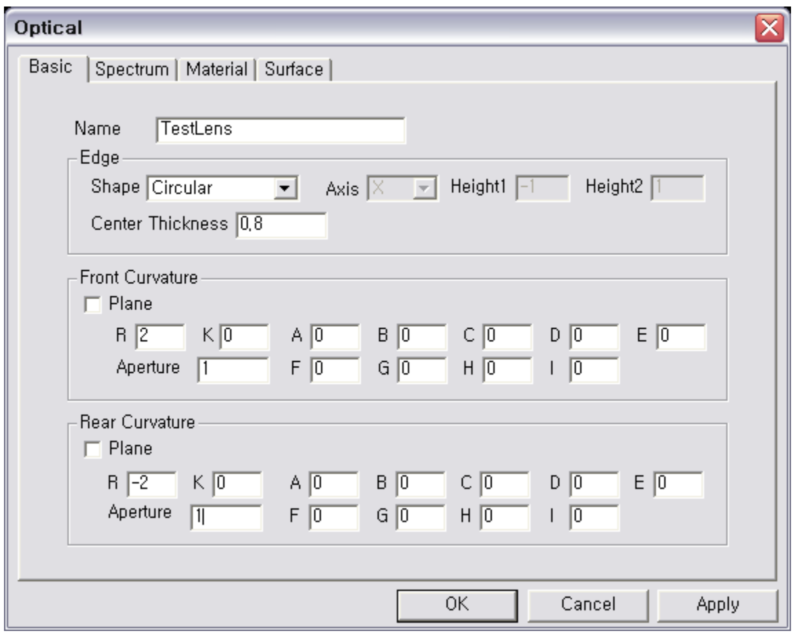
Grid_Ray.rwz

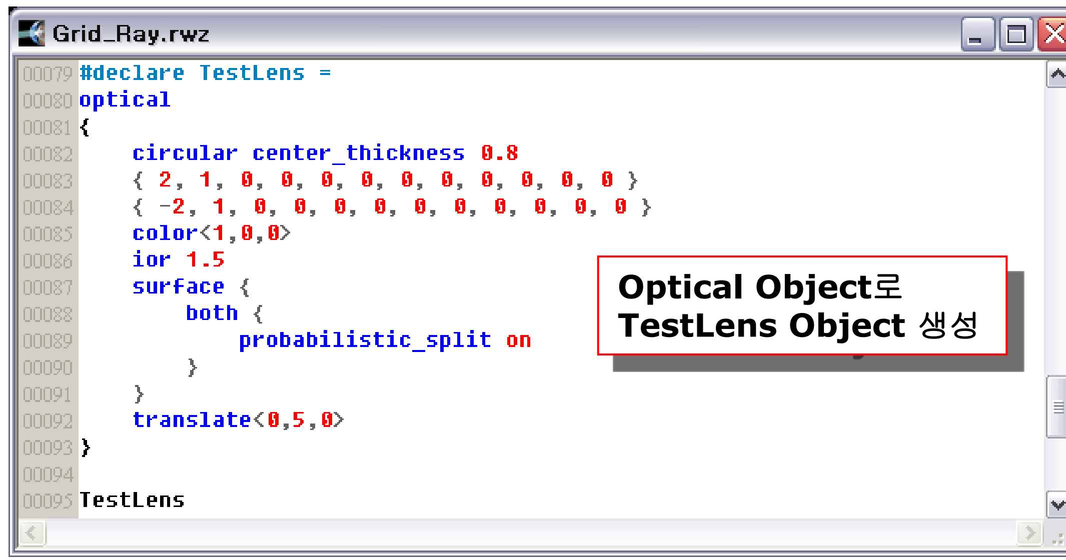
```
        emittance
        {
            front_surface
            {
                off
            }
            rear_surface
            {
                on
                angular_distribution
                {
                    uniform
                }
                spatial_distribution
                {
                    grid
                    boundary   {
                        5, 5
                    }
                }
            }
            cylinder_surface
            {
                off
            }
        }
    }
```

원하는 배열 차원을 입력

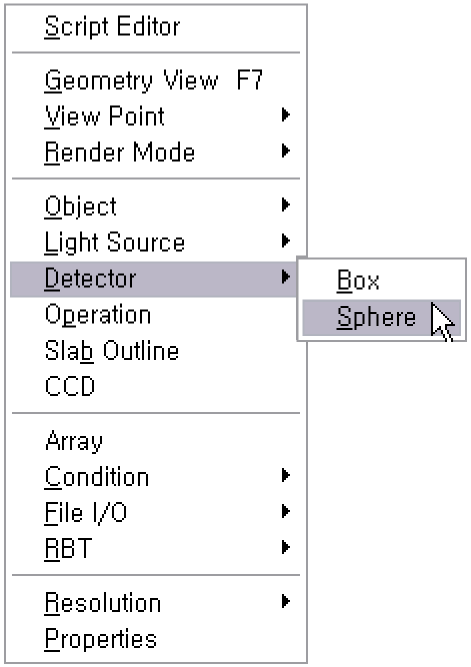
Light Source II: Grid Light (5)

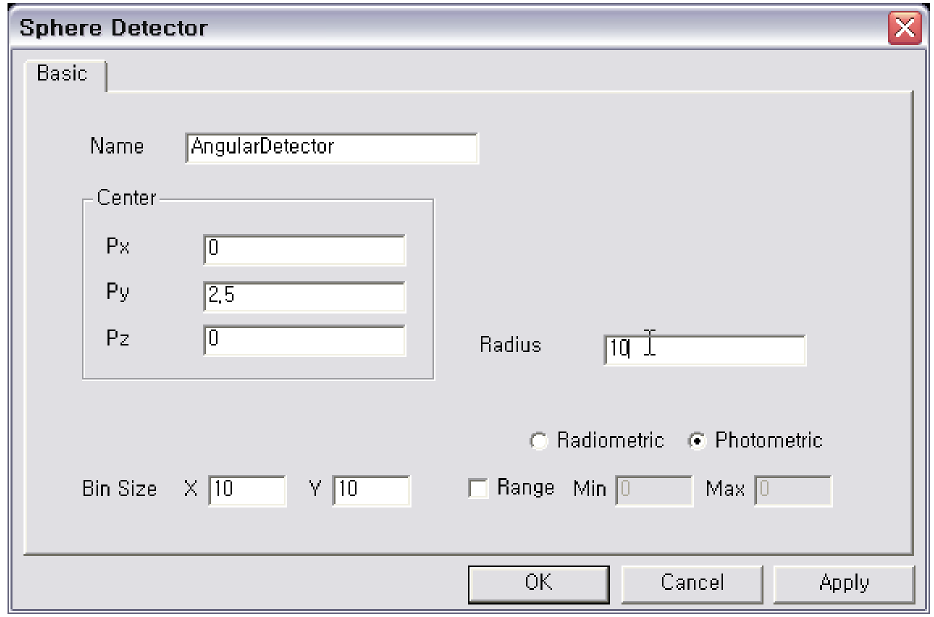


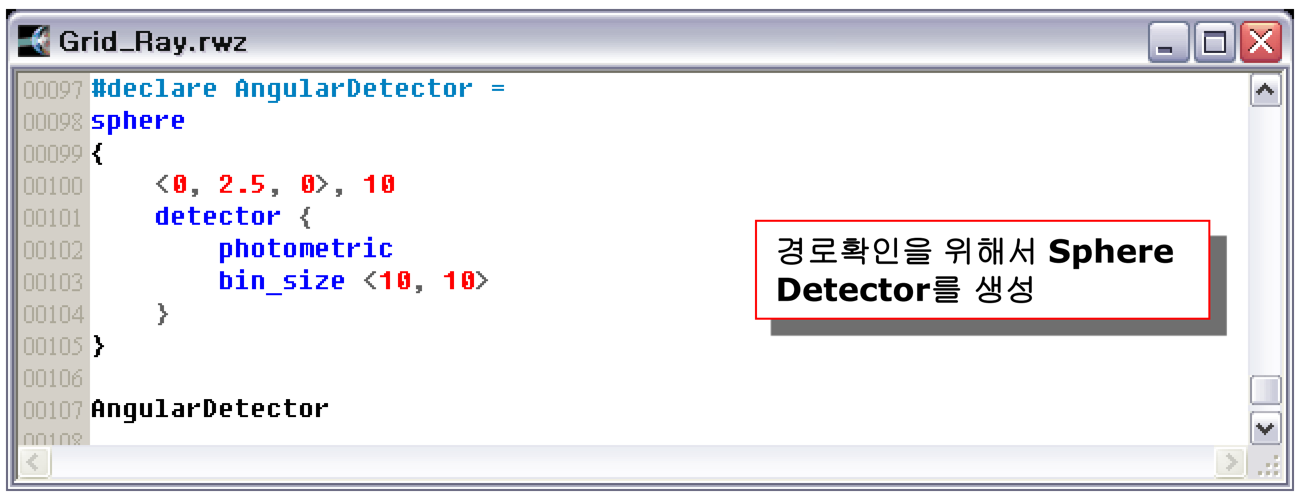




Light Source II: Grid Light (6)

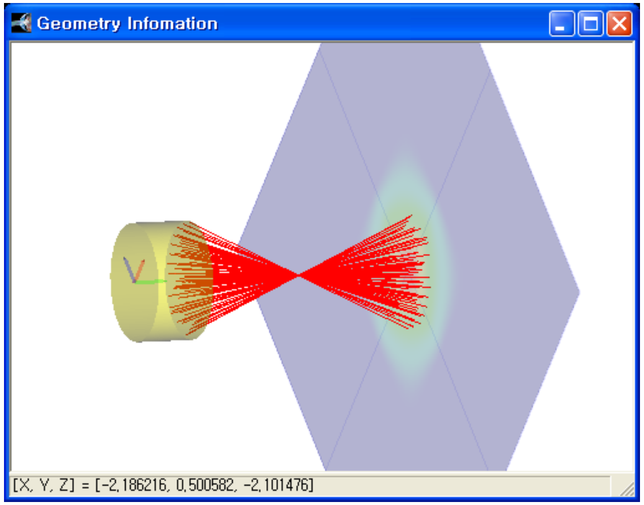




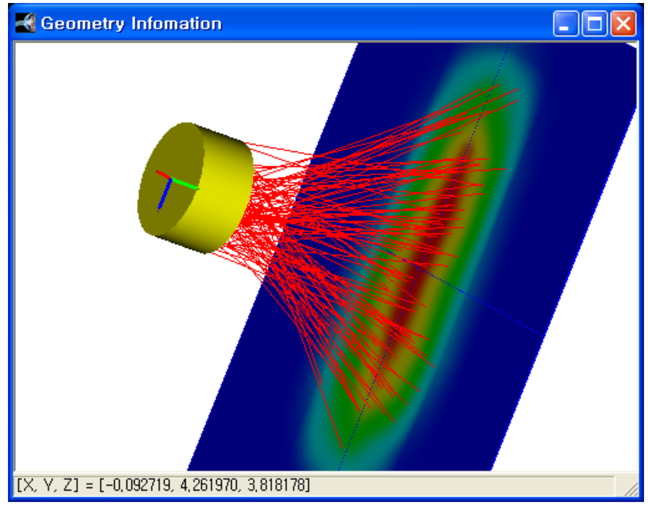


경로확인을 위해서 **Sphere Detector**를 생성

Light Source II: Aim_Sphere, Aim_Point, Aim_Area (1)



<Aim_Point>



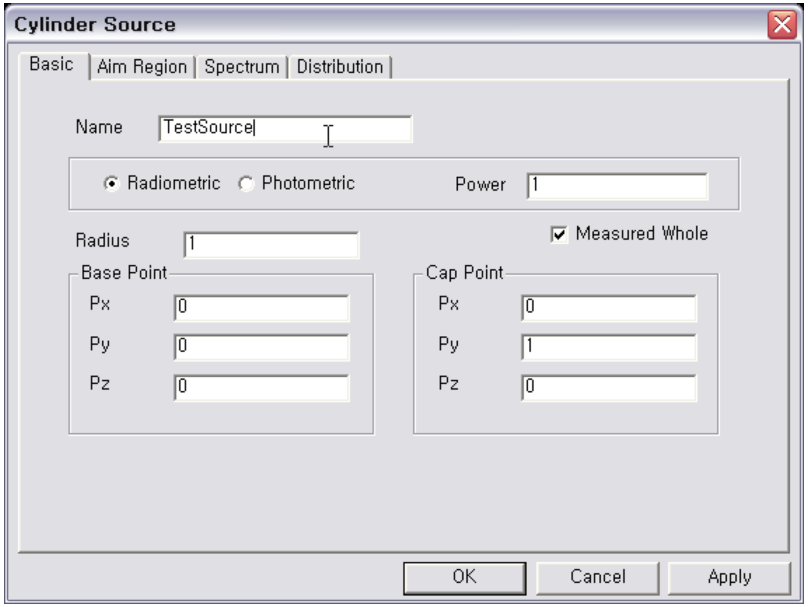
<Aim_Area>

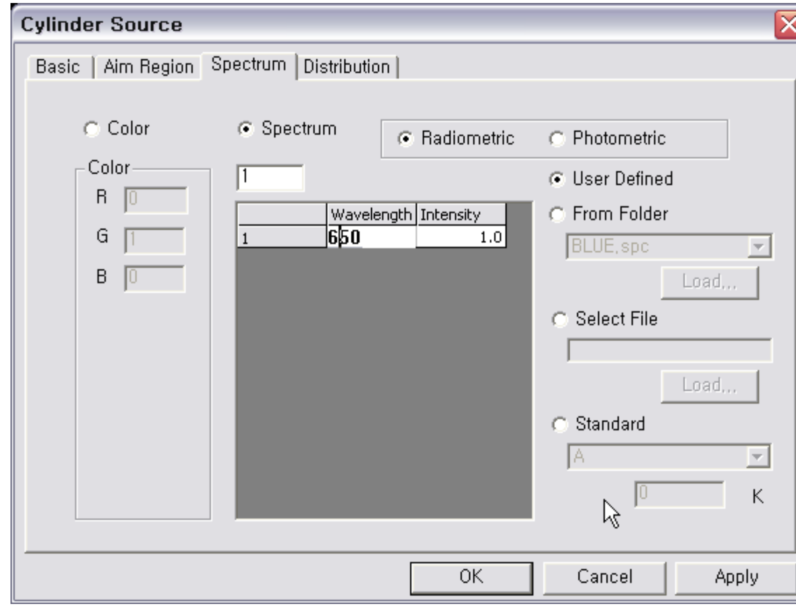
Aim_Sphere, Aim_Point, Aim_Area 중 하나를 설정

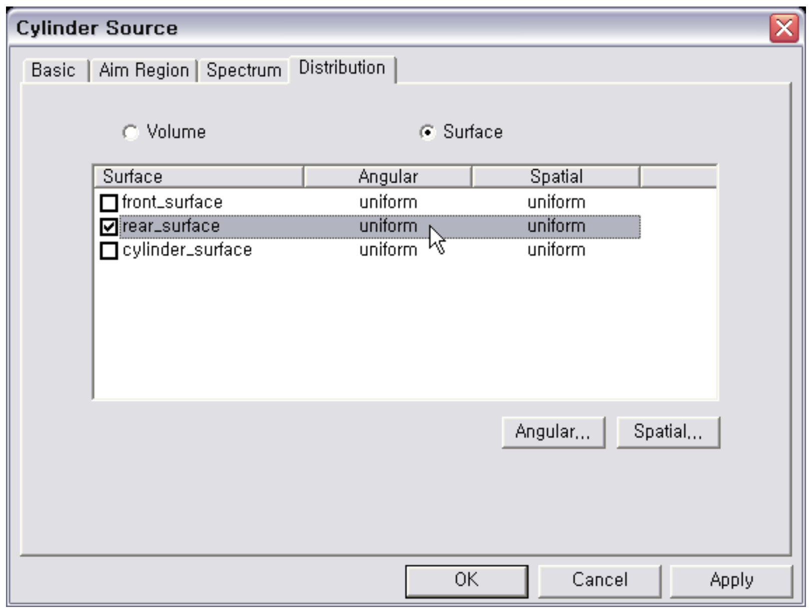
AimPoint_AimArea.rwz

```
        }
    }
    emittance
    {
        radiometric 1
        measured whole on
        //aim_sphere <0, 180>
        //aim_point
        aim_area {
          <0,3,0> <0,6,0>    //<center point> <direction point>
          //circular 0.1     // circular    [radius]
          rectangular 1, 3   // rectangular [width],[height]
        }
    }
    point_at <0, 3, 0>
}

TestSource

#declare TestDetector =
```

Light Source II: Aim_Sphere, Aim_Point, Aim_Area (2)







Light Source II: Aim_Sphere, Aim_Point, Aim_Area (3)

AimPoint_AimArea.rwz

```
#declare TestSource =
light_source
{
    <0, 0.5, 0>
    spectrum {
        discrete on
        1,
        <650, 1>
    }
    object_light
    {
        surface
        cylinder
        {
            <0, 0, 0>, <0, 1, 0>, 1
        }
```

AimPoint_AimArea.rwz

```
            emittance
            {
                front_surface
                {
                    off
                }
                rear_surface
                {
                    on
                    angular_distribution
                    {
                        uniform
                    }
                    spatial_distribution
                    {
                        uniform
                    }
                }
                cylinder_surface
                {
                    off
                }
            }
        }
```

Rear Surface 만 Emitting

Light Source II: Aim_Sphere, Aim_Point, Aim_Area (4)

AimPoint_AimArea.rwz

```
    emittance
    {
        radiometric 1
        measured_whole on
        //aim_sphere <0, 180>
        //aim_point
        aim_area {
          <0,3,0> <0,6,0>   //<center point> <direction point>
          //circular 0.1    // circular   [radius]
          rectangular 1, 3  // rectangular [width],[height]
        }
    }
    point_at <2, 3, 0>
}

TestSource
```

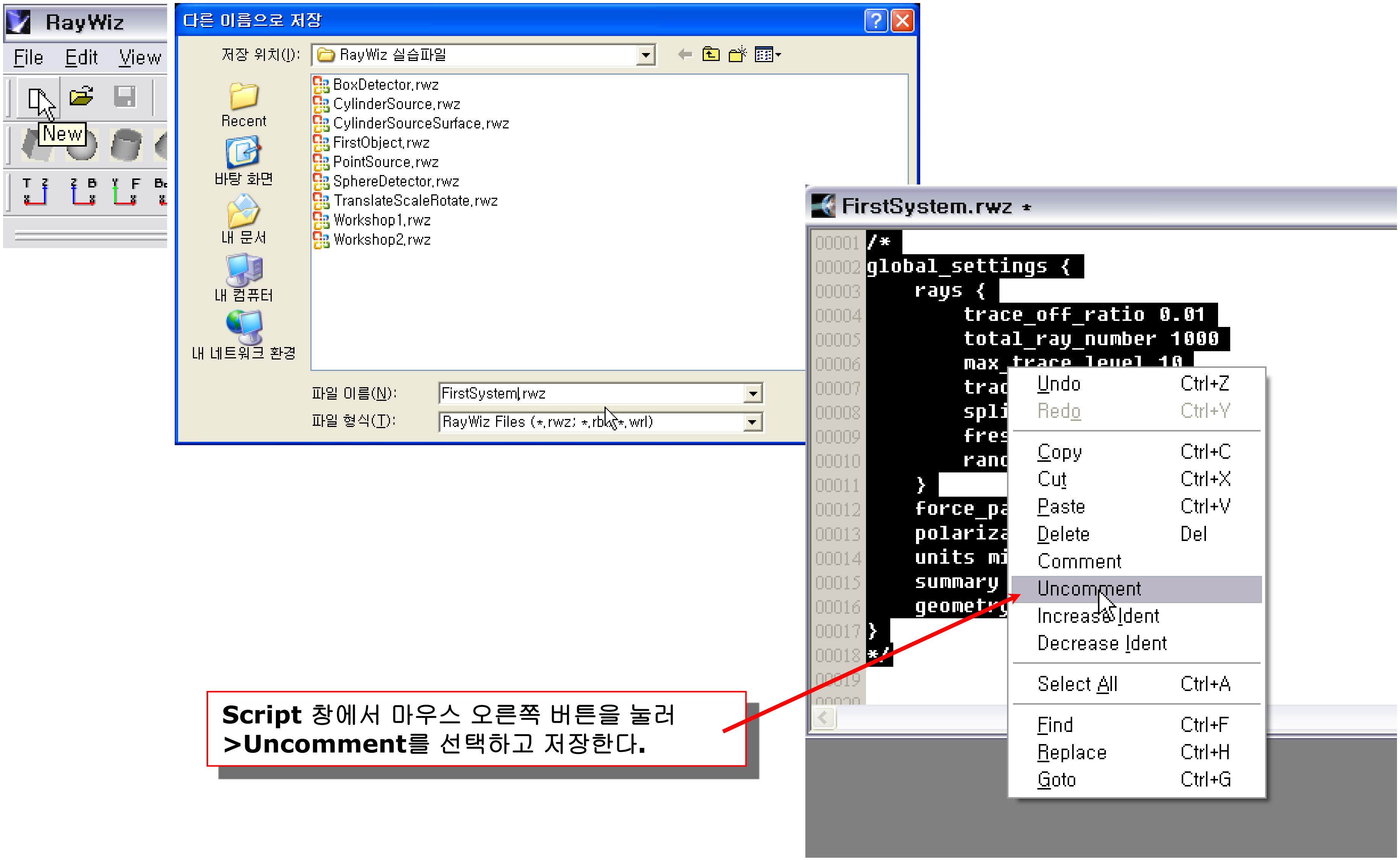
Aim_sphere 부분을 **comment** 처리하고 **aim_point** 나 **aim_area**를 입력한다.
-**Aim_point**인 경우는 **point_at** 의 좌표를 마치 이상적인 렌즈 초점처럼 **Ray**들을 집중 시킨다.
-**Aim_area**인 경우는 가상의 목표 영역의 중심점과 방향벡터 그리고 가로세로의 길이나 반경을 입력하여 정의하고 **Ray**들은 이 지정된 영역으로만 **Emitting** 하게된다.

AimPoint_AimArea.rwz

```

#declare TestDetector =
box
{
    <-5, 5, -5>, <5, 5.01, 5>
    detector {
        bottom_surface
        photometric
        bin_size <10, 10>
    }
}

TestDetector
```

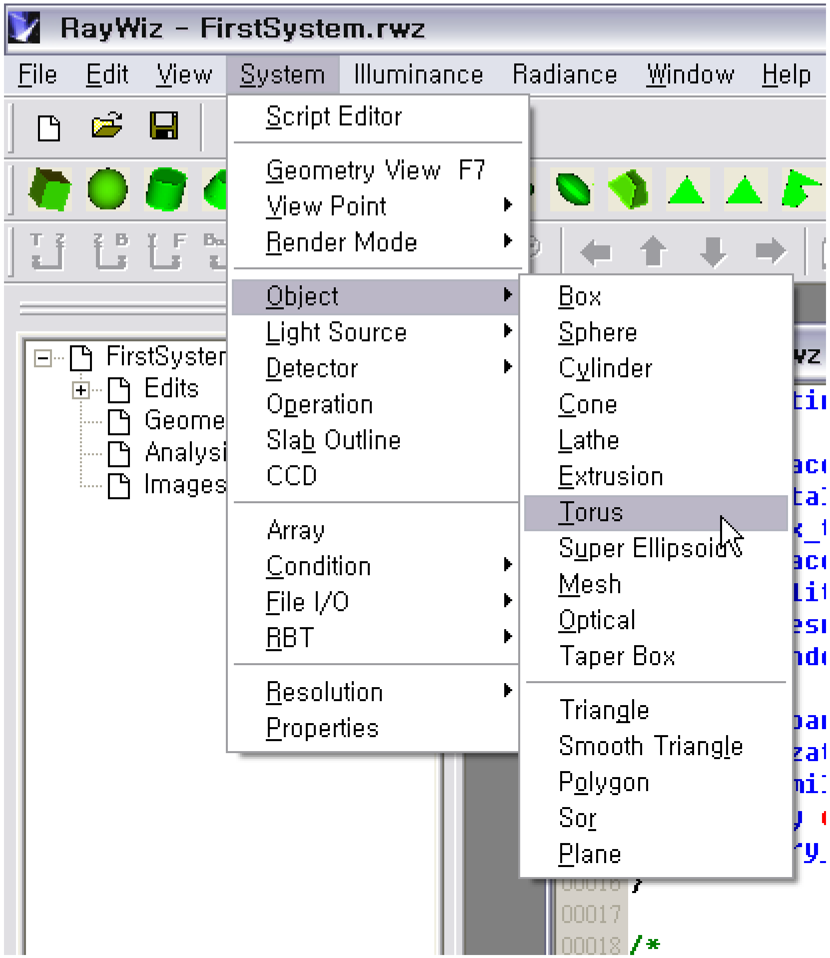
Box Detector를 정의하여 **Ray Path**를 관찰한다.

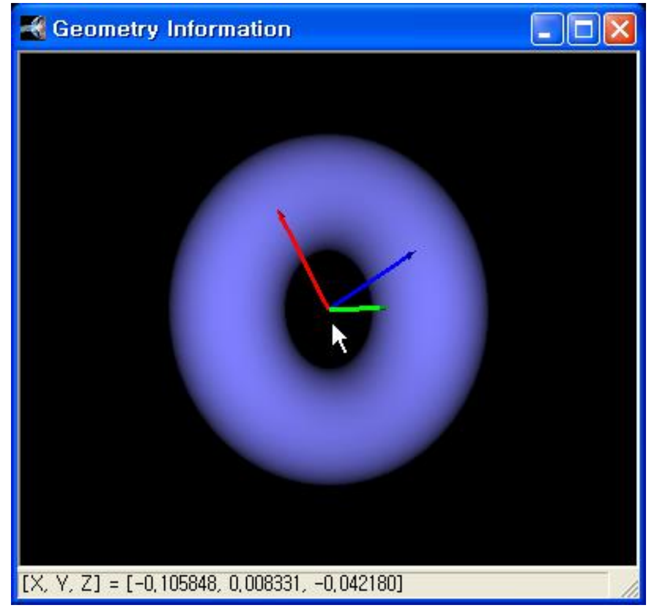
Detector: Create Your First System (1) – File Creation

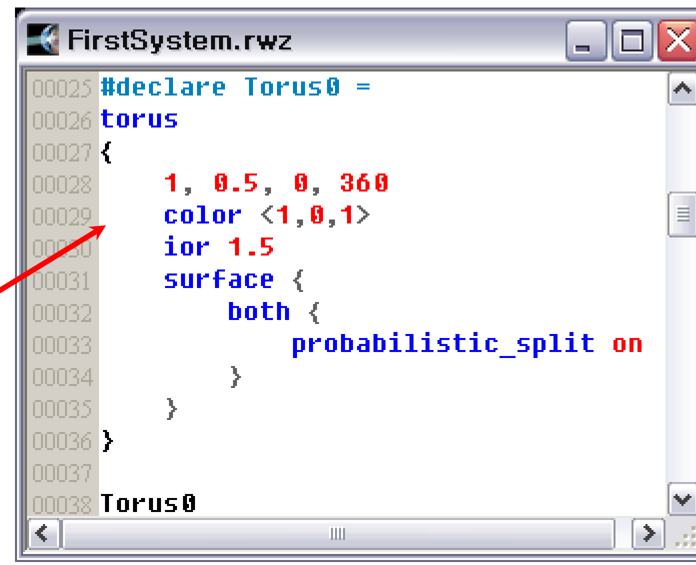


Script 창에서 마우스 오른쪽 버튼을 눌러
>Uncomment를 선택하고 저장한다.

Detector: Create Your First System (2) – Object Creation

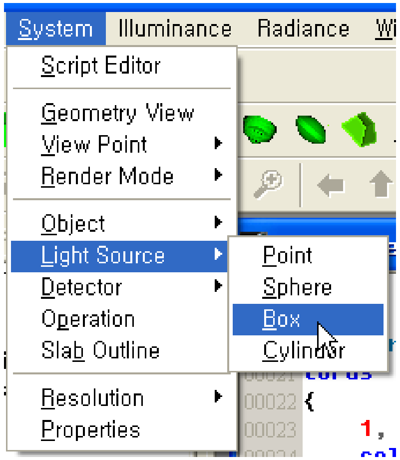


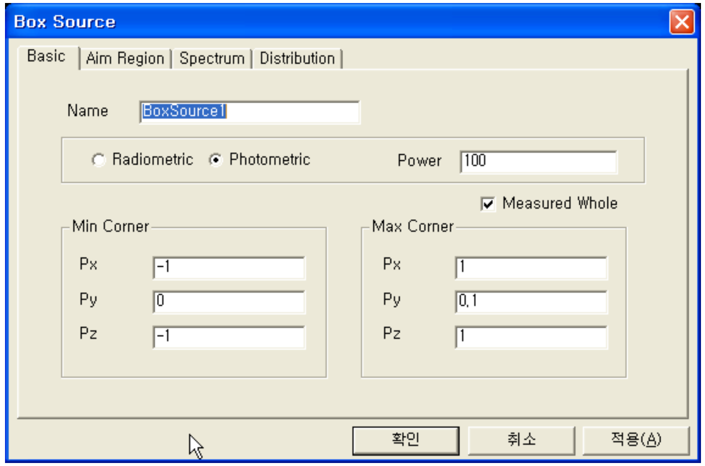


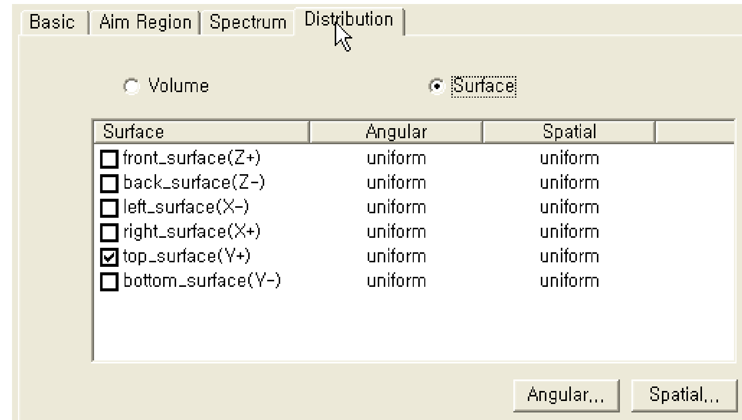


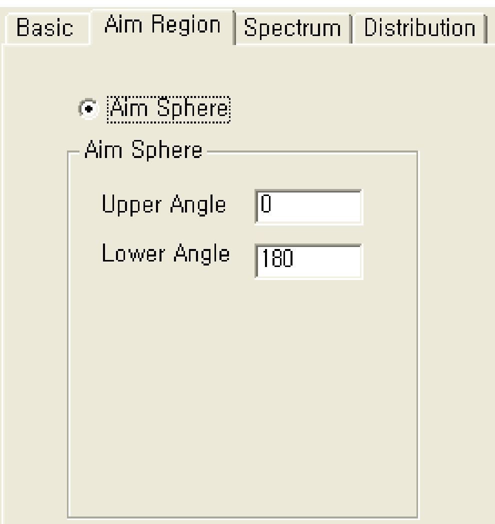
안쪽 반경이 **0.5**이고 바깥쪽 반경이 **1**인 **Torus**를 만들고 **E**광원 **Spectrum**에 해당하는 **RGB <1,0,1>**로 설정한다.

Detector: Create Your First System (3) – Source Creation







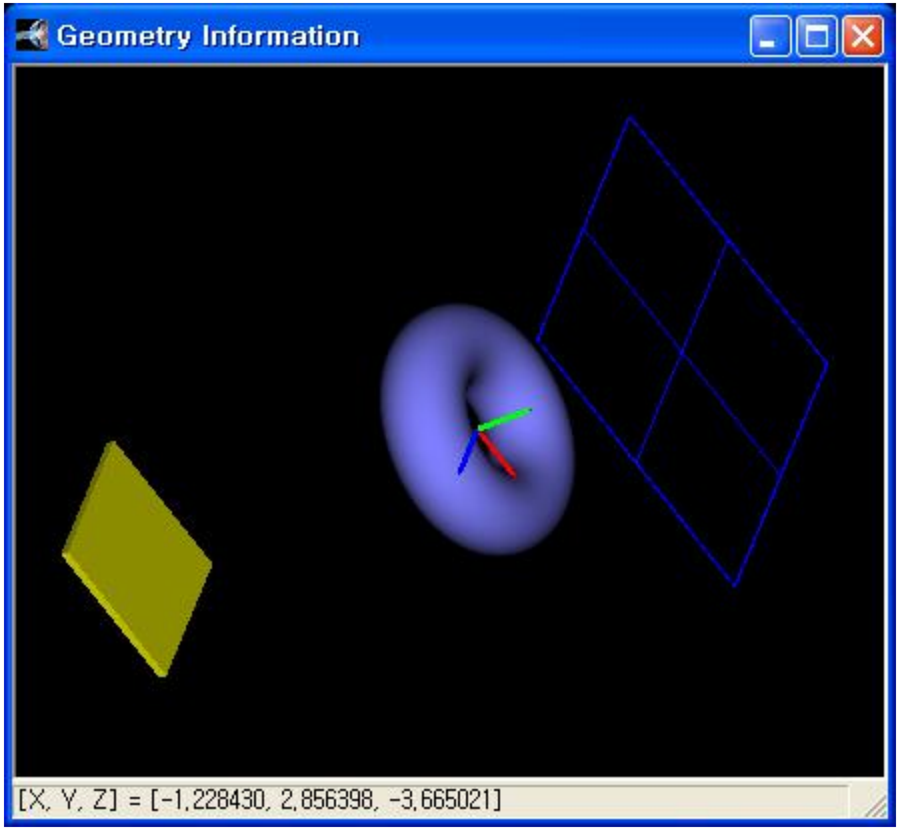


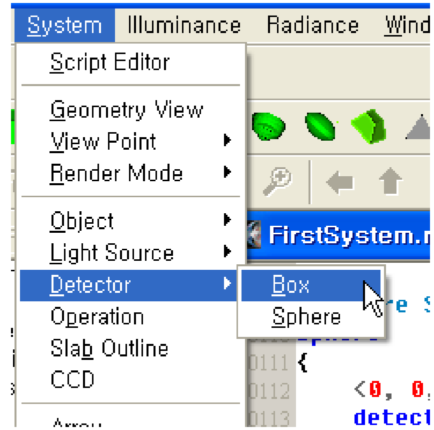

```
#declare BoxSource1 =
light_source
{
    <0, -5, 0>
    color <1, 1, 1>
    object_light
    {
        surface
        box
        {
            <-1, 0, -1>, <1, 0.1, 1>
        }
        emittance
        {
            photometric 100
            measured_whole on
            aim_sphere <0, 0>
        }
        point_at <0, 3, 0>
    }
```

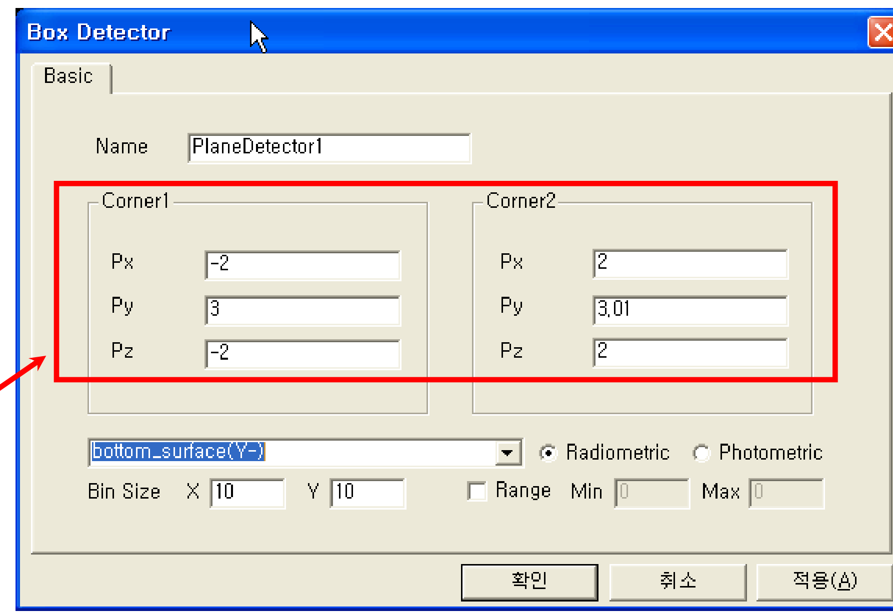
Light Source의 중점을 –Y축 방향으로 이동하도록 Script 수정

+Y방향을 향하도록 수정

Detector: Create Your First System (4)–Detector Creation

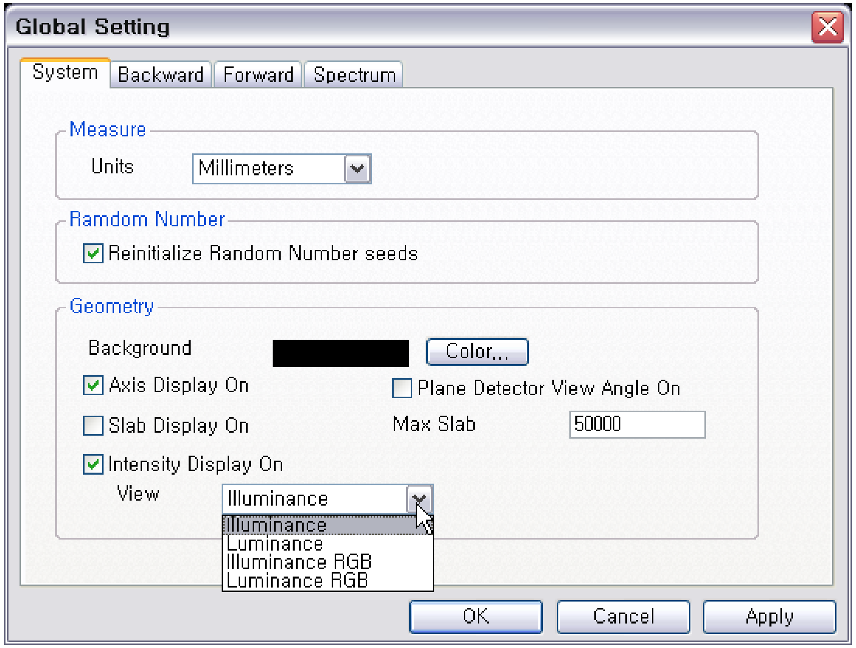


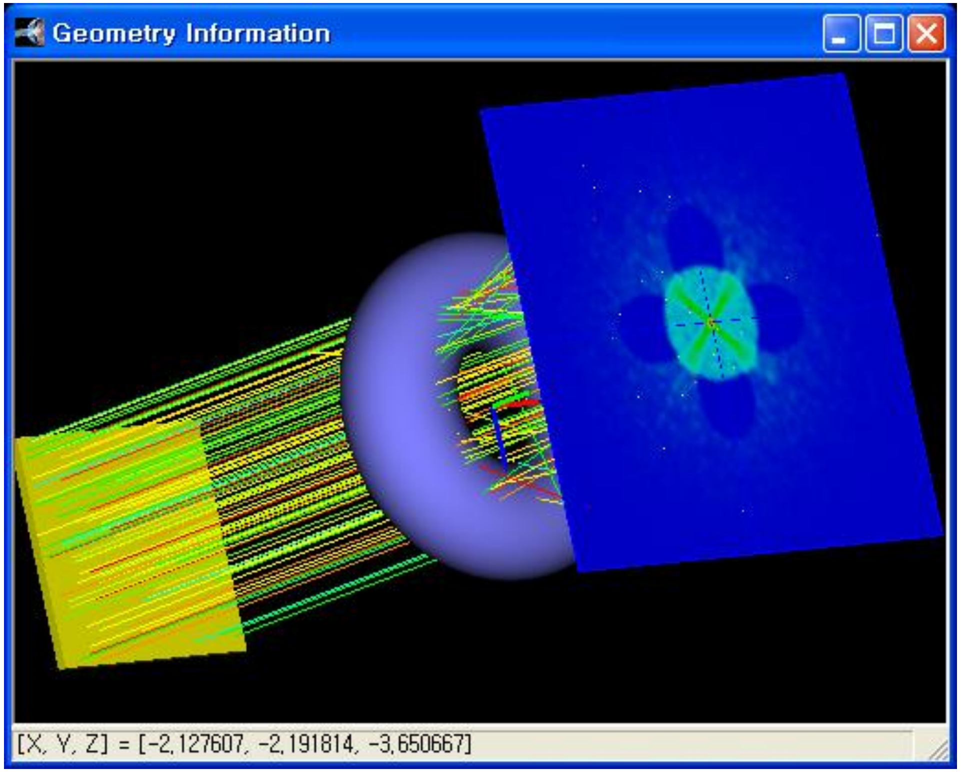


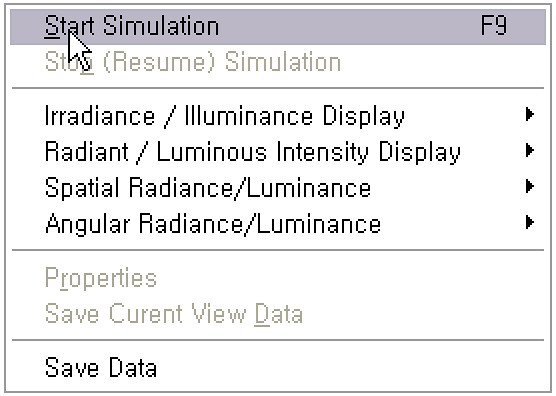


Y축 방향의 두께가 얇은 box detector를 정의하고
Bottom_surface (Y-)를 detecting 면으로 정의

Detector: Create Your First System (5) – Simulation

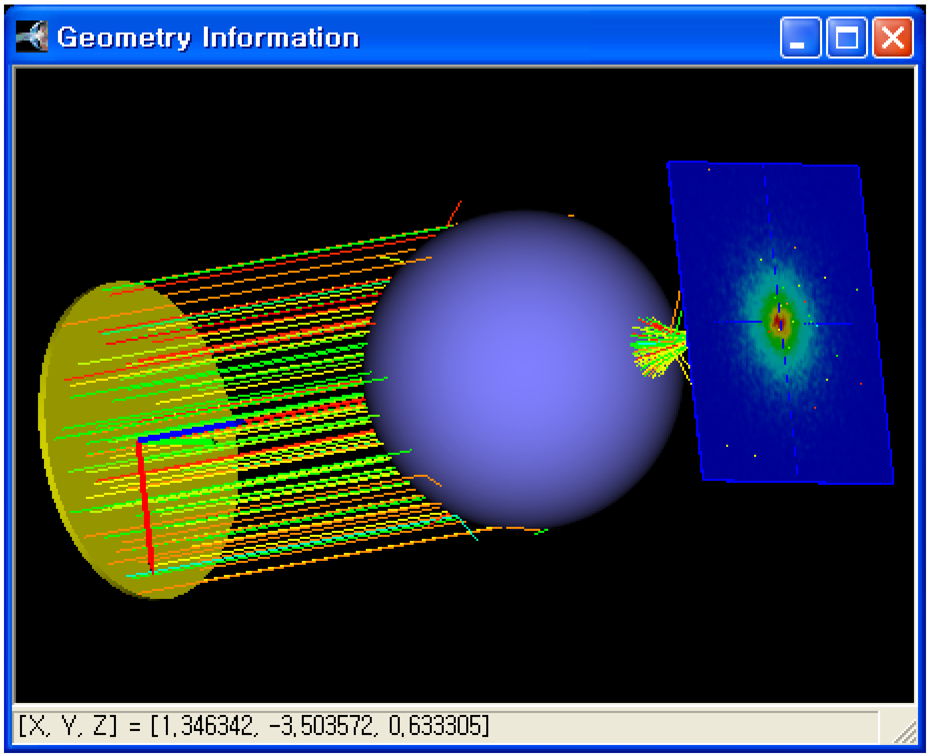






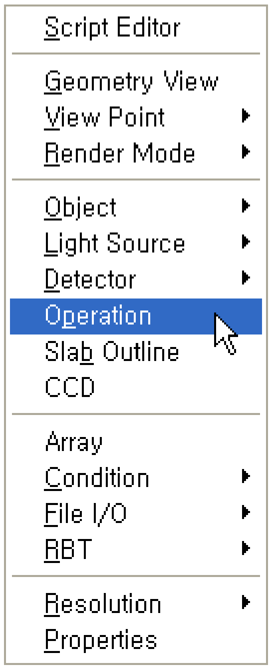
Workshop 3: Make System

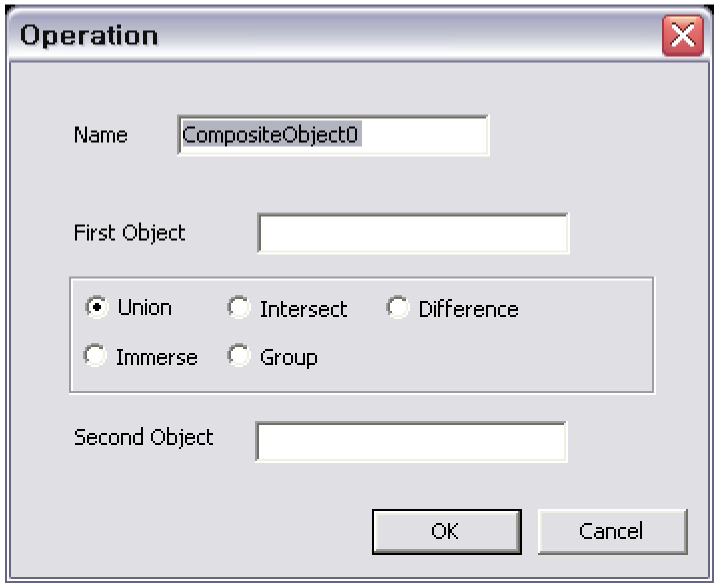
1. **RayWiz**를 실행하고 새 파일을 만들어 작업 **Directory**에 **"Workshop3.rwz"**로 저장하시오.
2. 반경이 **2mm**인 **Sphere object**를 작성하고 그 중점의 위치를 **<0,0,6>** 으로 설정하시오.
3. 반경이 **2mm**이고 높이가 **0.1mm**인 **Surface Cylinder Light**를 작성하고 **Rear Surface**에서만 **Ray**가 직진하도록 하고 **Light**의 위치를 **<0,0,0>**으로 설정하시오.
4. **+Z**축 방향의 두께가 **0.01mm**이고 가로,세로 길이가 각각 **4mm Box Detector** 하나를 생성하고 **back_surface(Z-)**를 **detecting** 면으로 지정하고 그 위치를 **<0,0,10>**으로 설정하시오.
5. **Ray number**를 **100000**으로 설정하고 **simulation**을 수행하시오.



RAYWIZ BASIC
Object Geometry II
- [Operation] -

Geometry II: Operation

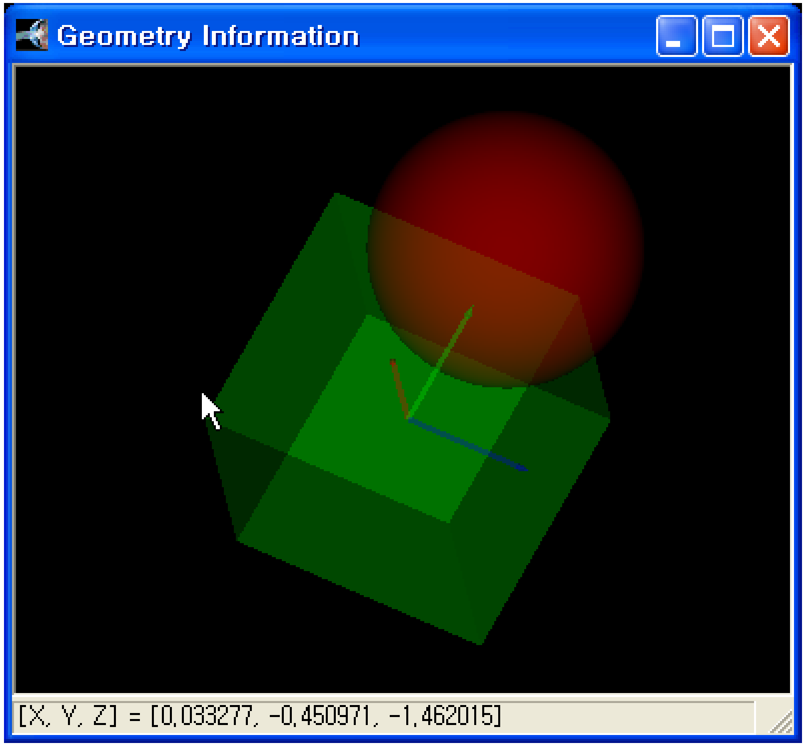




1. **Union**
2. **Intersect**
3. **Subtract**
4. **Immerse**
5. **Group**

물성이 다른 **Object**들의 묶음을 하나로 정의

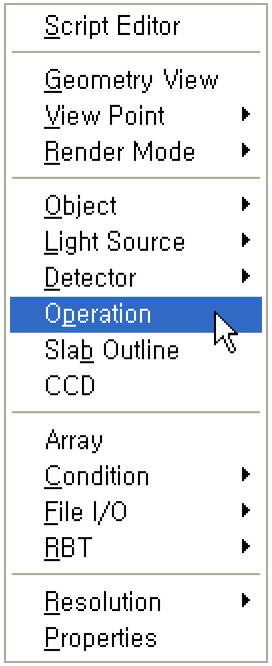
Geometry II: Declare

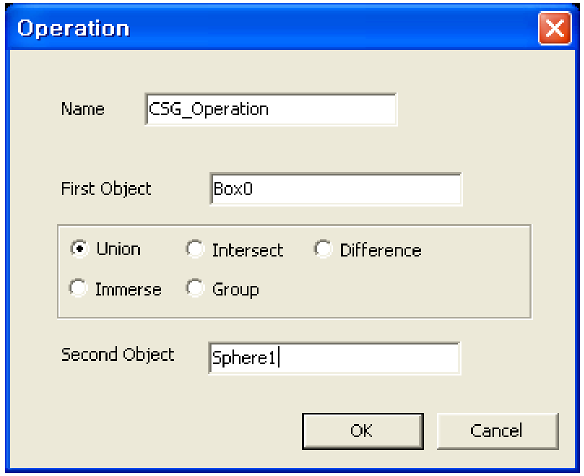


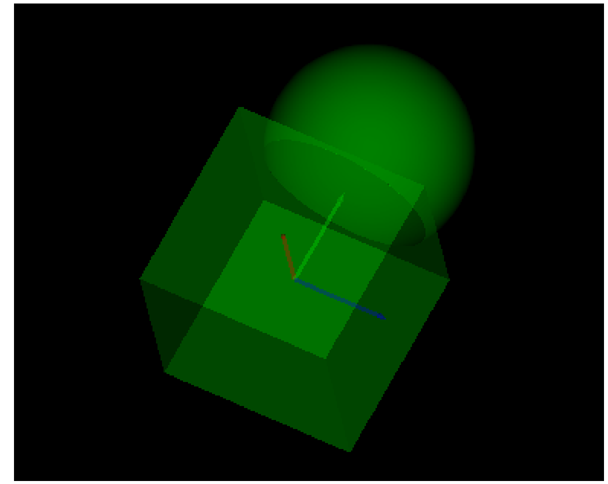

```
#declare Box0 =
box
{
    <-1, -1, -1>, <1, 1, 1>
    color <0, 1, 0>
    ior 1.5
    surface {
        both {
            probabilistic_split on
        }
    }
}

#declare Sphere1 =
sphere
{
    <0, 1.5, 0>, 1
    color <1, 0, 0>
    ior 1.5
    surface {
        both {
            probabilistic_split on
        }
    }
}
```

Operation에 사용될 **Object**들을 정의하고 부르지는 않는다.
초록색의 **Box**와 빨간색의 **Sphere !**

Geometry II: Union

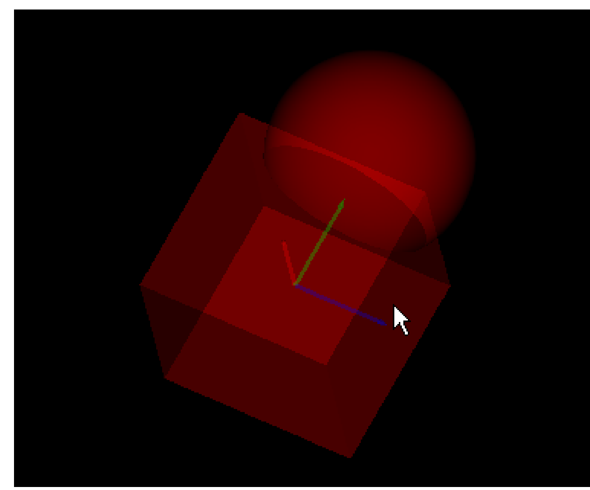




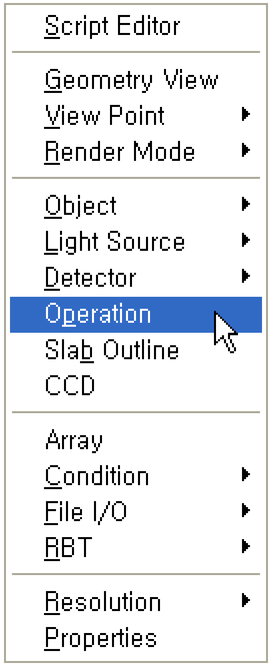


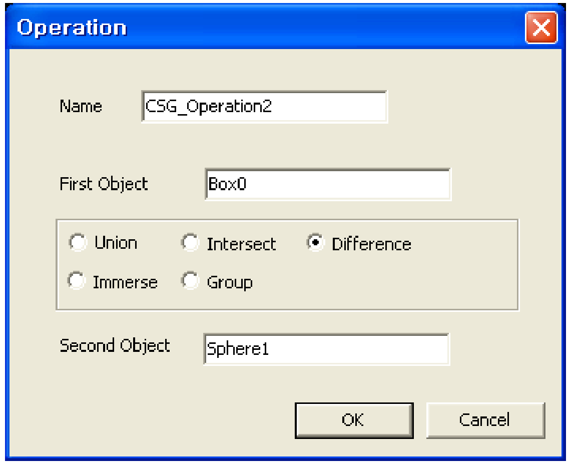
Union 된 Object는 First Object의 물성을 따른다.

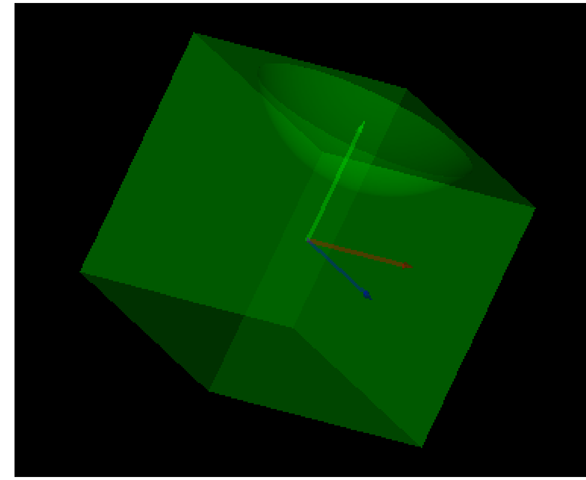
```
#declare CSG_Operation =
union
{
    object { Sphere1 }
    object { Box0 }
}

CSG_Operation
```



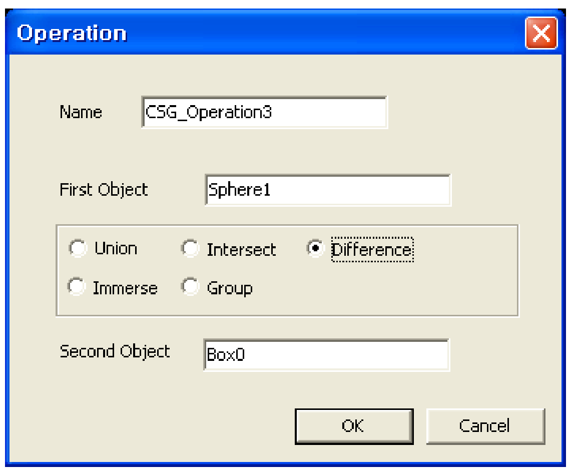
Geometry II: Difference

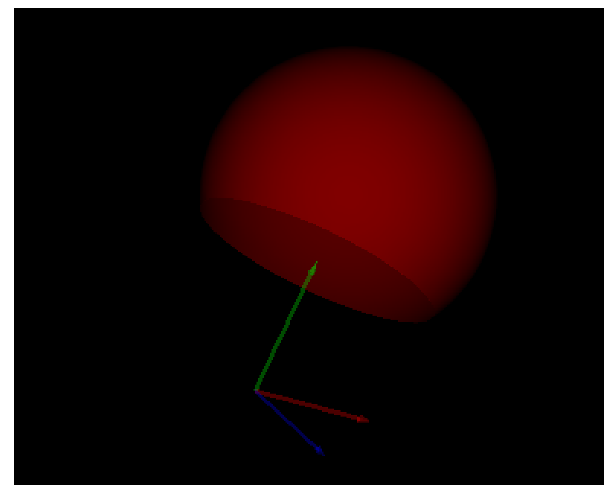




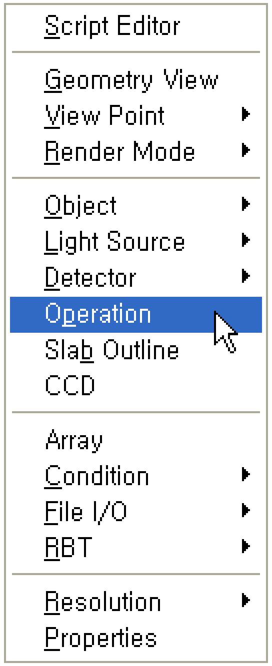


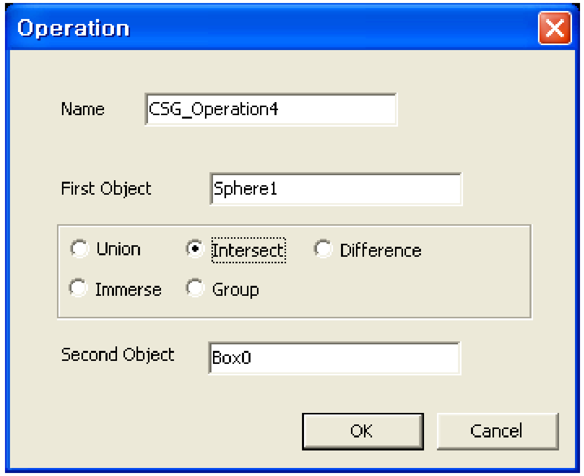
Difference 된 Object는 First Object의 물성을 따른다.

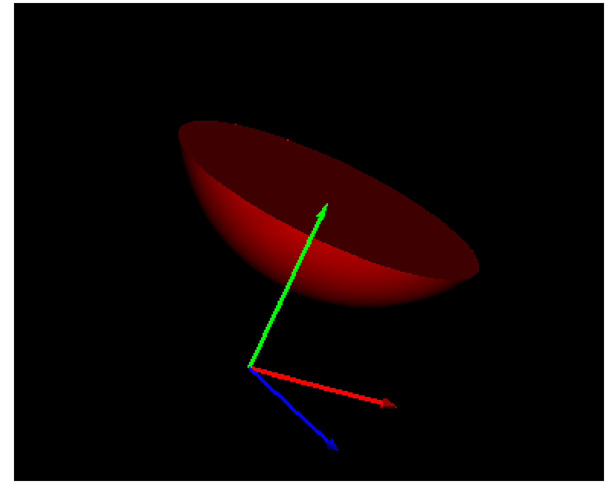




Geometry II: Intersection

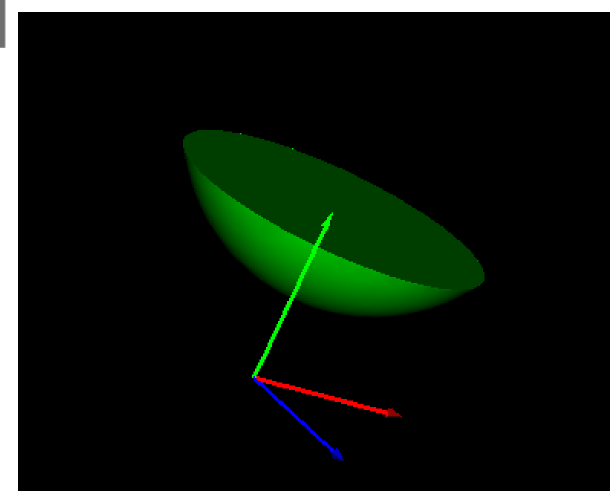




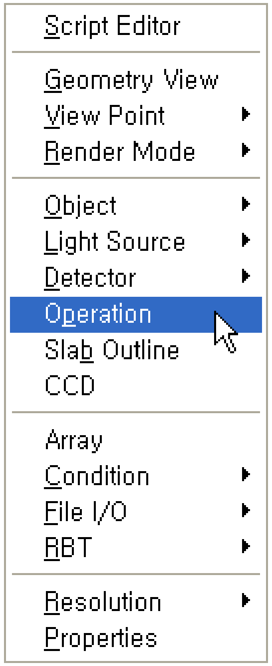


Intersection 된 Object는 First Object의 물성을 따른다.

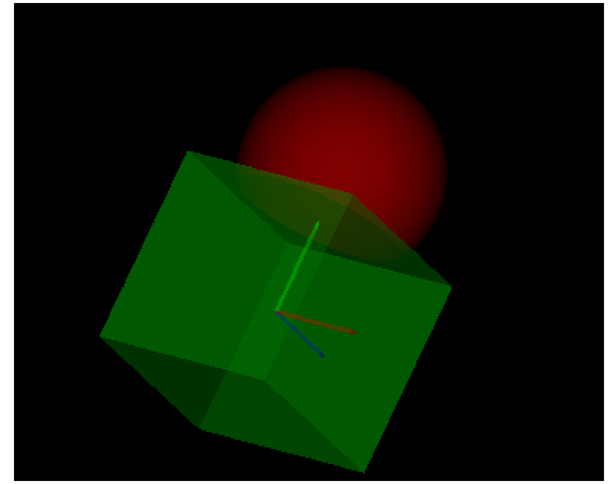
```
#declare CSG_Operation5 =
intersection
{
     object { Box0 }
     object { Sphere1 }
}

CSG_Operation5
```



Geometry II: Immerse

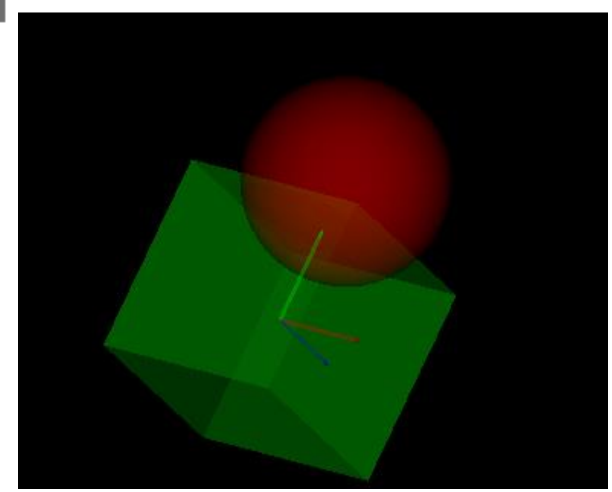



```
#declare CSG_Operation6 =
immerse
{
      object { Box0 }
      object { Sphere1 }
}

CSG_Operation6
```



First Object는 물성을 유지하며 **Second Object**에 공유 영역 만큼 들어가게 된다.

```
#declare CSG_Operation7 =
immerse
{
      object { Sphere1 }
      object { Box0 }
}

CSG_Operation7
```

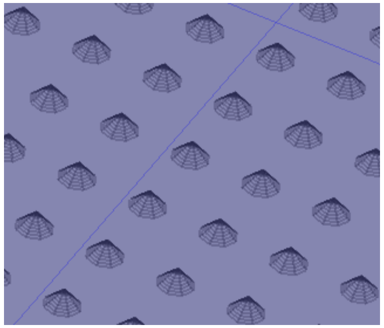


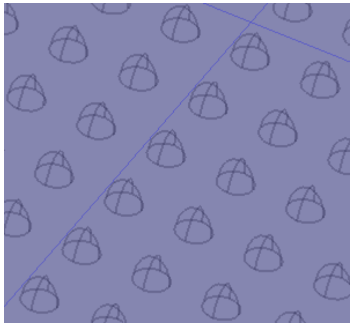
RAYWIZ BASIC
Object Geometry II
- [Slab Outline] -

Geometry II: What is Slab ?

BLU 의 **LGP**와 **BEF**와 같이 동일한 종류의 **Object** 가 반복적으로 그룹을 지어 단일 **Object** 표면에 돌출되어 있는 경우**(Bump)**나 **Object** 표면이 함몰되어있는 경우**(Hole),** 또는 **LCD Color Filter**나 **Panel**의 **Pixel** 와 같이 반복적인 형상을 가지며 그 물성이 다른 경우**(Immerse) Slab_Outline** 명령어를 사용합니다**.**

Geometry View 상에 **Slab Object**의 아웃라인만을 그리기 때문에 **Geometry** 표현시간을 줄일 수 있으며**,** 특수한 **Algorithm**이 적용되어 빠른 **Simulation** 이 가능하게 합니다**.**

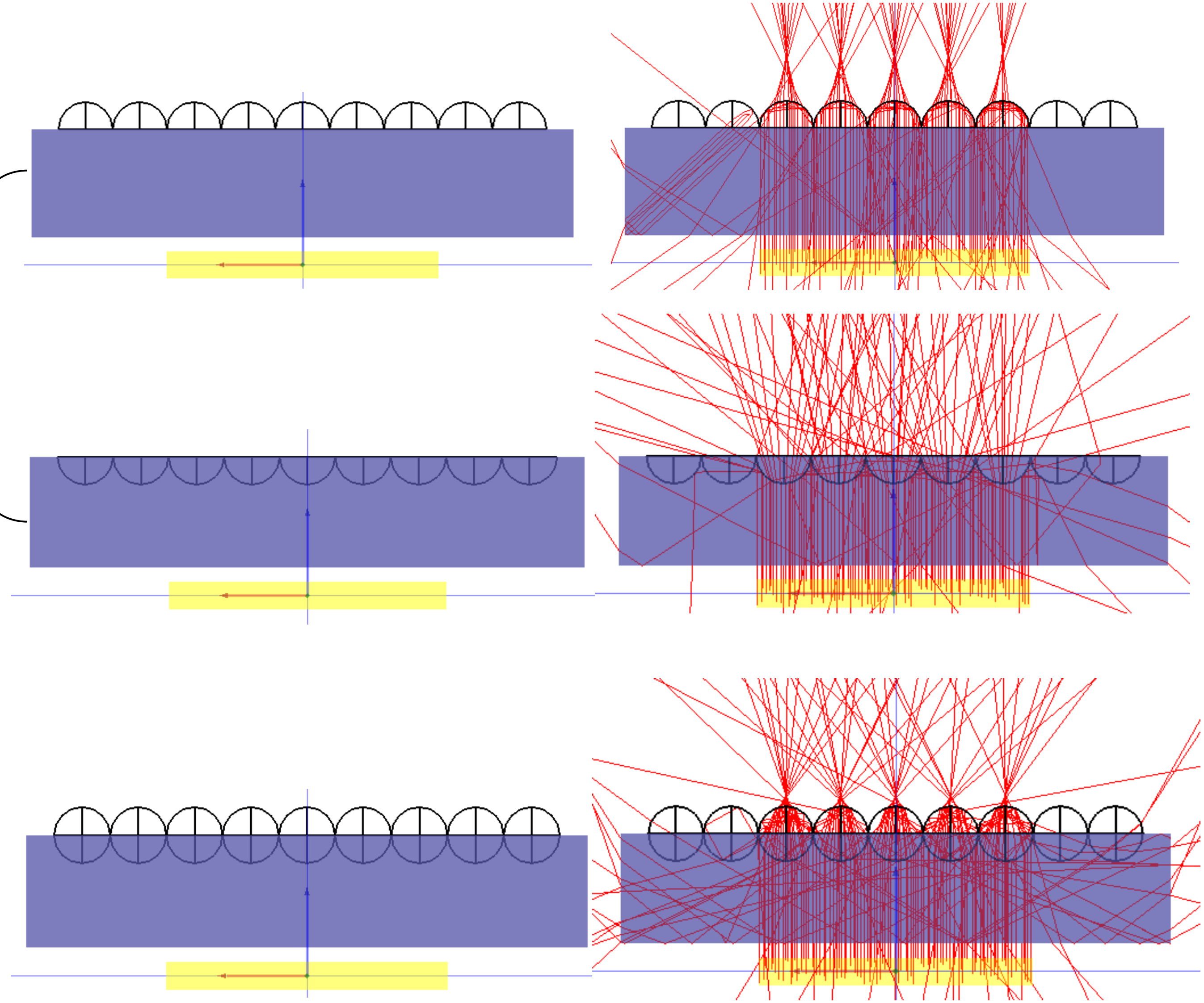




Geometry II: Smart Slab

Slab

3D Texture



Geometry II: Slab_Outline Grammar (1)

모든 **Object**는 **Slab**으로 사용되어 질 수 있으며 모든 **Object Surface**에 적용 가능하다. 또한 간단한 **macro** 작업에 의해서 어떠한 형태의 분포라도 형성이 가능하다. 자주 사용되어지는 **Cube Surface**에 대해서는 자동으로 몇 가지 미리 정해진 규칙에 의한 배열을 형성시켜준다.

```
SLAB_OUTLINE:
        slab_outline
        {
                SLAB_TYPE  CUBE_SURFACE_TYPE
                [normal]
                [SLAB_DETAILS...]
                base        { BASE_OBJECT }
                slab        { SLAB_OBJECT }
        }

SLAB_TYPE:
        bump | hole | immerse
CUBE_SURFACE_TYPE:
        left_surface | right_surface | front_surface | back_surface |
        top_surface | bottom_surface
```

Geometry II: Slab_Outline Grammar (2)

```
SLAB_DETAILS:
        SLAB_POSITION |SLAB_SIZE |SLAB_DEPTH |SLAB_SHIFT
SLAB_POSITION:
        position_x|position_y {
                Coefficient_N0(상수부), Coefficient_N1(1차),
                Coefficient_N2(2차), Coefficient_N3(3차),
                Coefficient_N4(4차), Coefficient_N5(5차)
        }
        |
        position { SLAB_POSITION_FILE_LOCATION }
```

Polynomial 5th 계수 입력

위치 값이 있는 파일지정

```
SLAB_SIZE:
        size_x|size_y {
                Coefficient_N0(상수부), Coefficient_N1(1차),
                Coefficient_N2(2차), Coefficient_N3(3차)
        }
        |
        size { SLAB_SIZE_FILE_LOCATION }
```

Polynomial 3th 계수 입력

크기 값이 있는 파일지정

Geometry II: Slab_Outline Grammar (3)

SLAB_DEPTH:
depth {
Coefficient_N0(상수부), Coefficient_N1(1차),
Coefficient_N2(2차), Coefficient_N3(3차)
}
|
depth { SLAB_DEPTH_FILE_LOCATION }

Polynomial 3th 계수 입력

깊이 값이 있는 파일지정

SLAB_SHIFT:
shift <X_Shift, Y_Shift>

지정된 면의 시작점을 변경

Geometry II: FirstSlab - Base Object & Slab Object

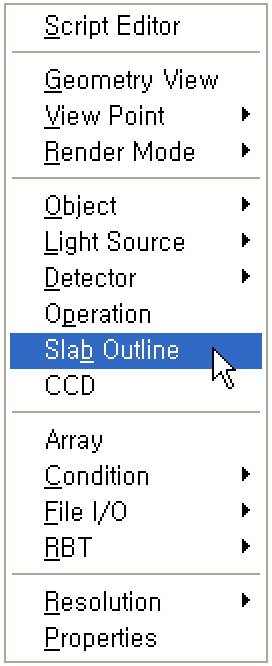
```
#declare Box0 =
box
{
    <-1, -1, 0.1>, <1, 1, 0.5>
    color <1,1,1>
    ior 1.5
    surface {
        both {
            probabilistic_split on
        }
    }
}
```

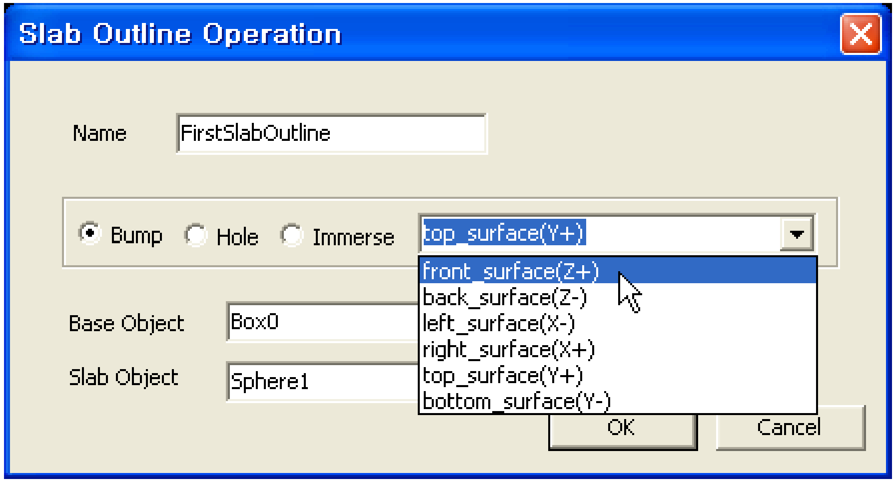
Base로 사용될 **Box** 선언

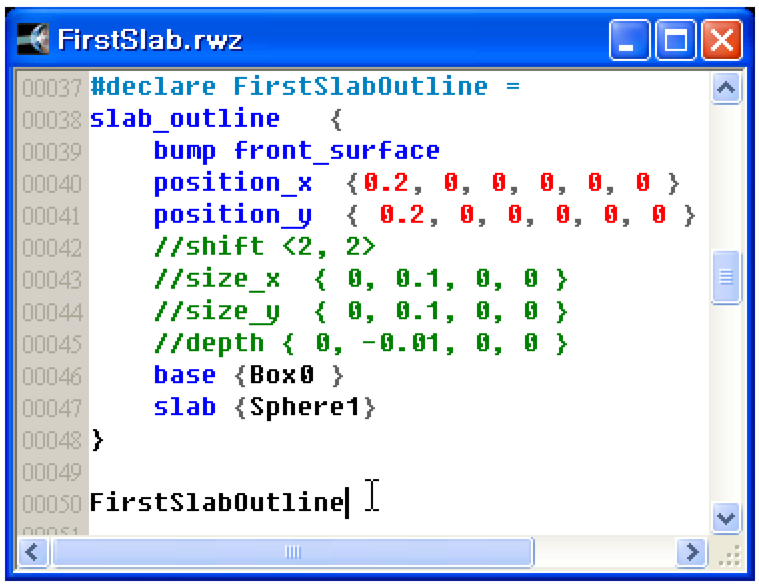
```
#declare Sphere1 =
sphere
{
    <0, 0, 0>, 0.1
    color <1,1,1>
    ior 2.0
    surface {
        both {
            probabilistic_split on
        }
    }
}
```

Slab으로 사용될 **Sphere** 선언

Geometry II: FirstSlab - Slab_Outline Operation

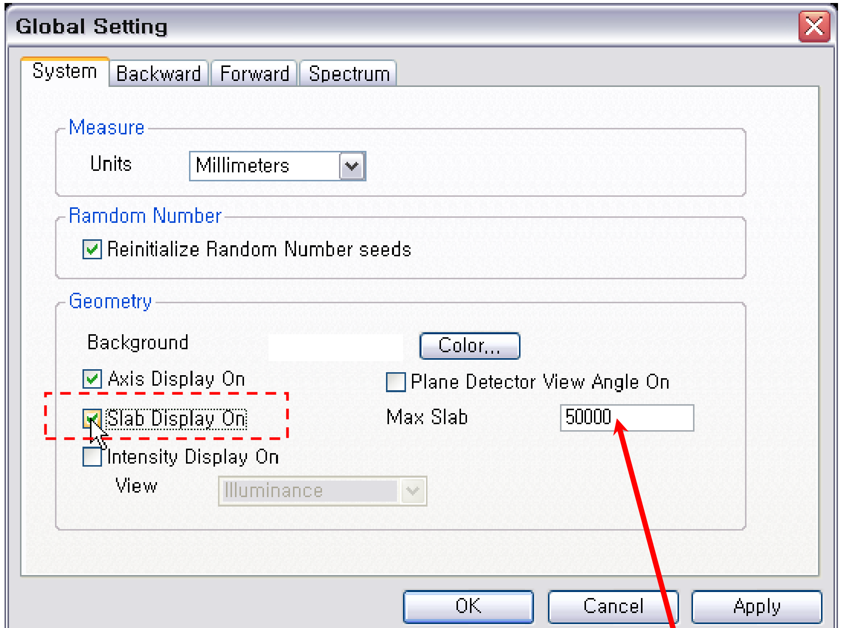


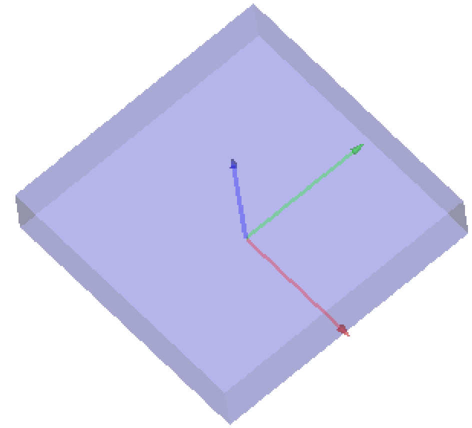




X-Y Pitch를 Slab으로 사용된 Sphere의 지름으로 설정하여 정방 배열로 만든다.

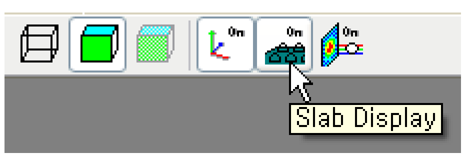
Geometry II: FirstSlab - Geometry View

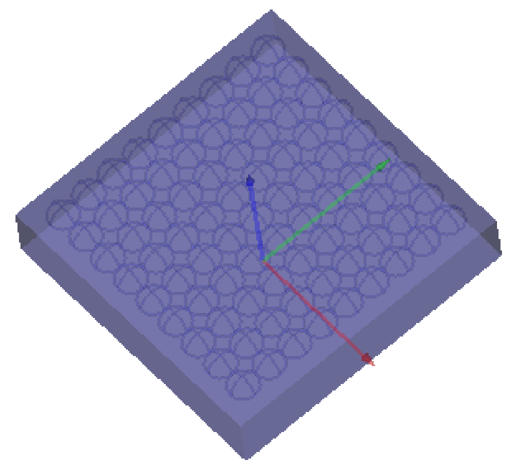




Global Option의
기본값은 **Slab**이 나타나지
표현되지 않도록 되어있다.

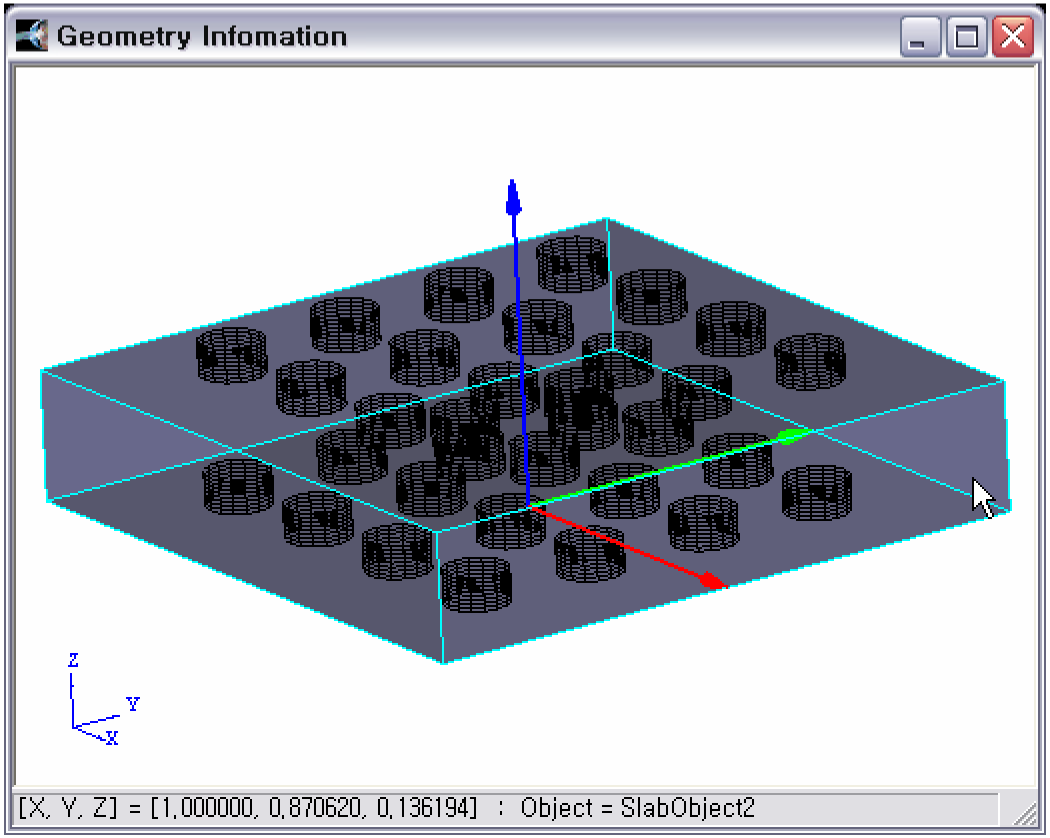
Slab이 보여지도록 설정되어 있을 경우 최대
몇 개 까지를 그릴 것인지에 대한 제한
→그래픽카드나 메모리의 양에 따라
적절히 조절한다.





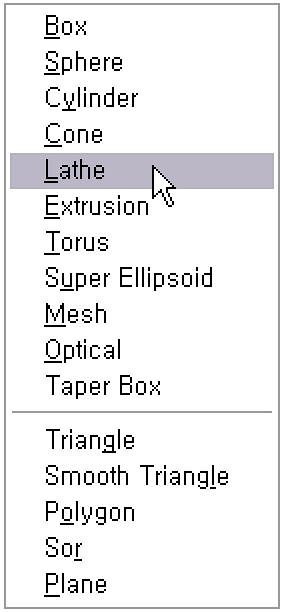
Workshop 4: Make Slab Object

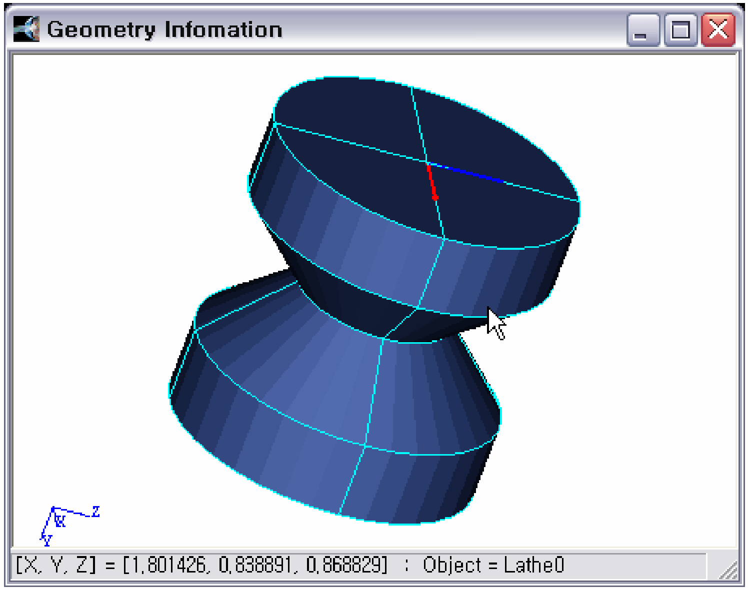
1. **RayWiz**를 실행하고 새 파일을 만들어 작업 **Directory**에 **"Workshop4.rwz"**로 저장하시오.
2. **Menu**나 **Toolbar**를 이용하여 **Base**로 사용될 **2mm x 2mm x 0.4mm** 인 **Box Object**를 선언하고 **Slab**으로 사용될 반지름이 **0.1mm** 이고 높이가 **0.1mm**인 **Cylinder** 선언하시오.
3. **Slab Operation** 메뉴를 **2**번 사용하여 피치가 **0.4mm**가 되도록 **Base Box**의 한쪽 면에는 정방 배열의 **Bump**를 반대쪽 면에는 정방배열의 **Hole**을 작성하시오.

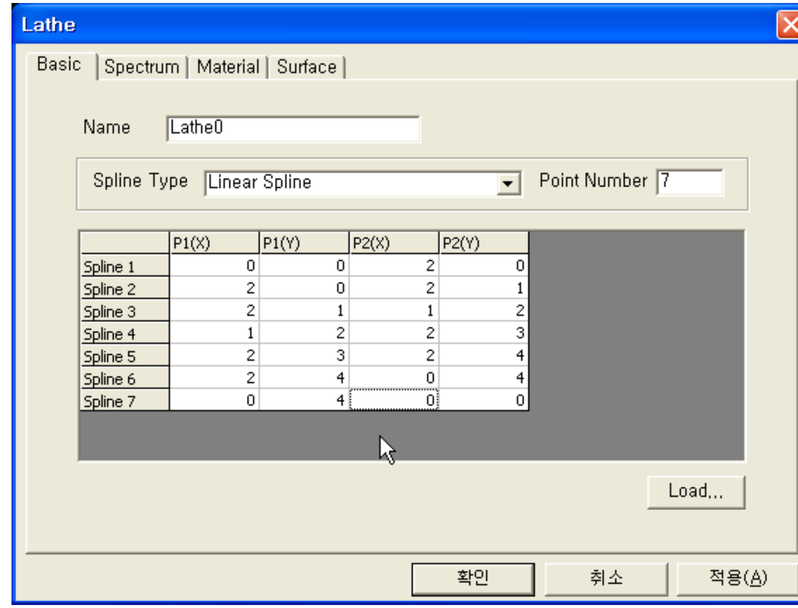


RAYWIZ BASIC
Object Geometry II
- [Complex Geometry] -

Geometry II: Lathe



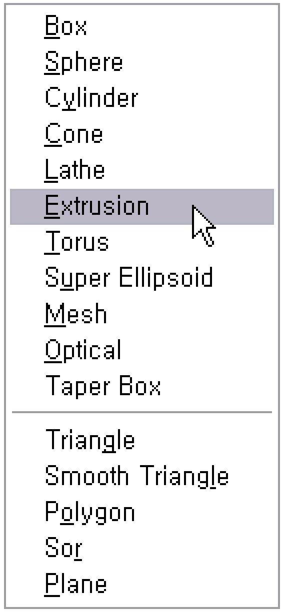


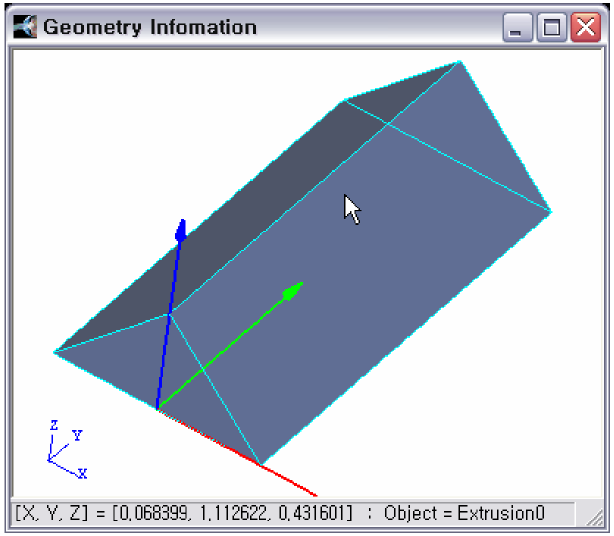


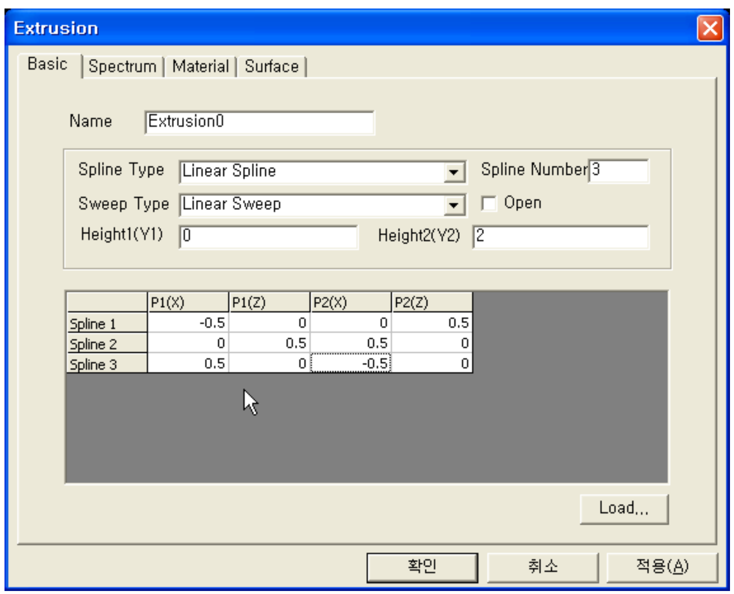

```
#declare Lathe0 =
lathe
{
    linear_spline 7,
    <0, 0>,
    <2, 0>,
    <2, 1>,
    <1, 2>,
    <2, 3>,
    <2, 4>,
    <0, 4>
    color <1, 1, 1>
    ior 1.5
    surface {
        both {
            probabilistic_split on
        }
    }
}

Lathe0
```

- 회전체
- **Y축으로 회전시킬 면을 이루는 꼭지점의 좌표를 입력한다.**
- 회전체를 생성하는 경우는 첫 번째 점과 마지막 점의 좌표를 동일하게 입력 해준다.

Geometry II: Extrusion



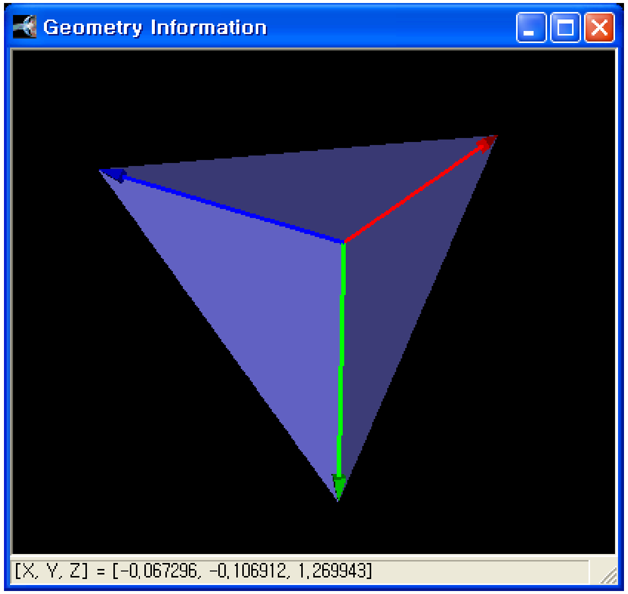


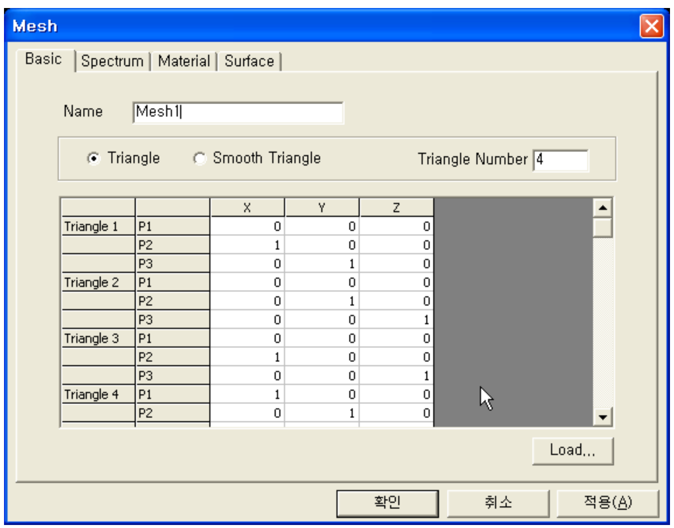



```
#declare Extrusion0 =
extrusion
{
    linear_spline linear_sweep
    0, 2,
    4,
    <-0.5, 0>,
    <0, 0.5>,
    <0.5, 0>,
    <-0.5,0>
    color <1,1,1>
    ior 1.5
    surface {
        both {
            probabilistic_split on
        }
    }
}

Extrusion0
```

- **XZ평면도형을 Y축으로 돌출시킨 입체**
- 선의 종류와 높이, **XZ** 평명 도형의 꼭지점의 좌표를 입력한다.

Geometry II: Mesh Object – Corner Cube (1)

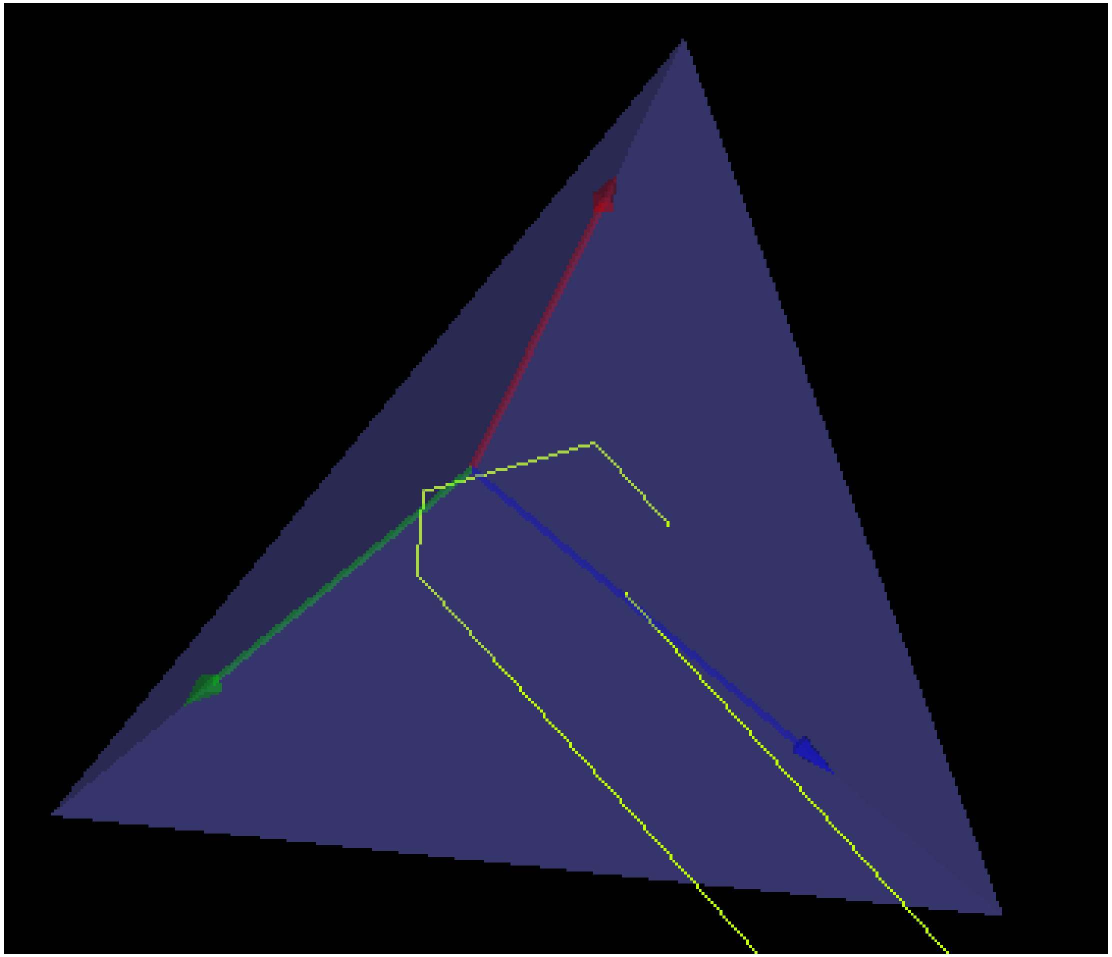




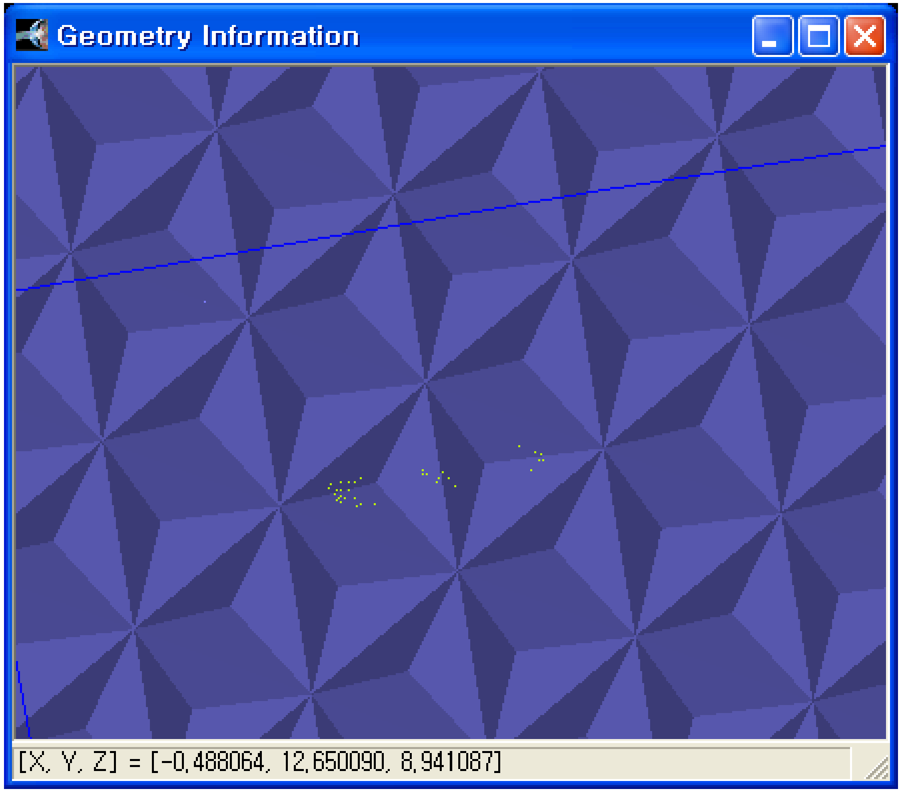

```
#declare Mesh1 =
mesh
{
    triangle
    {
        <0, 0, 0>, <1, 0, 0>, <0, 1, 0>
    }
    triangle
    {
        <0, 0, 0>, <1, 0, 0>, <0, 0, 1>
    }
    triangle
    {
        <0, 0, 0>, <0, 1, 0>, <0, 0, 1>
    }
    triangle
    {
        <1, 0, 0>, <0, 1, 0>, <0, 0, 1>
    }
    color <1, 1, 1>
    ior 1.5
    surface {
        both {
            probabilistic_split on
        }
    }
}

Mesh1
```

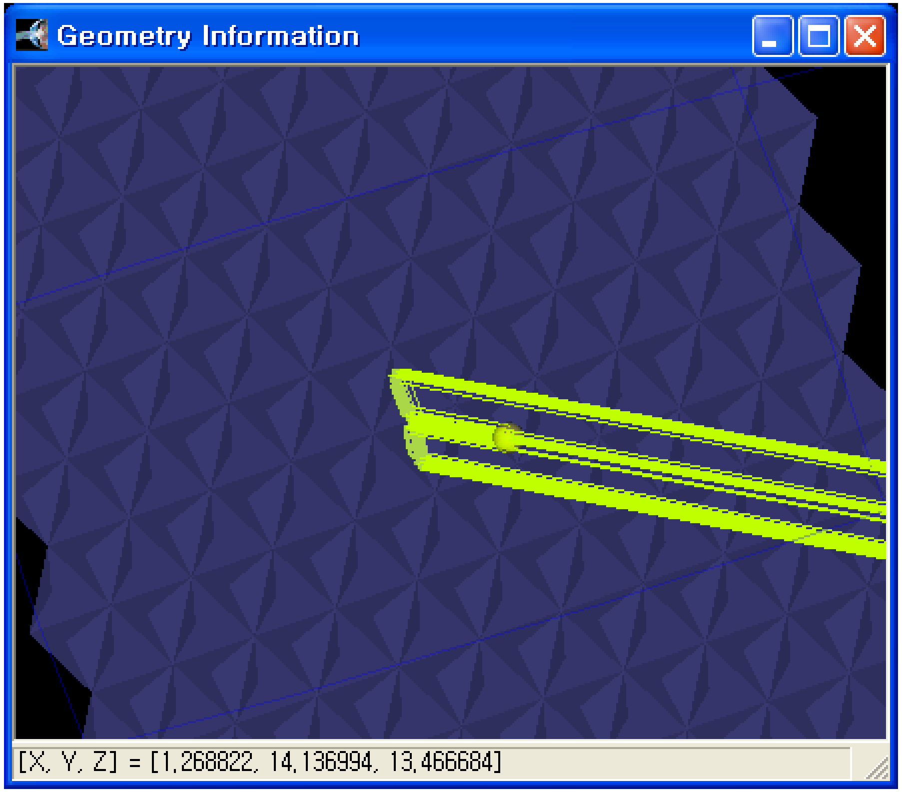
- 삼각형으로 이루어진 그물 모양의 면
- 다수의 삼각형의 좌표를 입력한다.
- 모서리를 부드럽게 표현하기 위해서는 **Smooth Triangle**을 선택하여 법선 벡터를 지정한다.

Geometry II: Mesh Object – Corner Cube (2)



Geometry II: Mesh Object – Corner Cube (3)



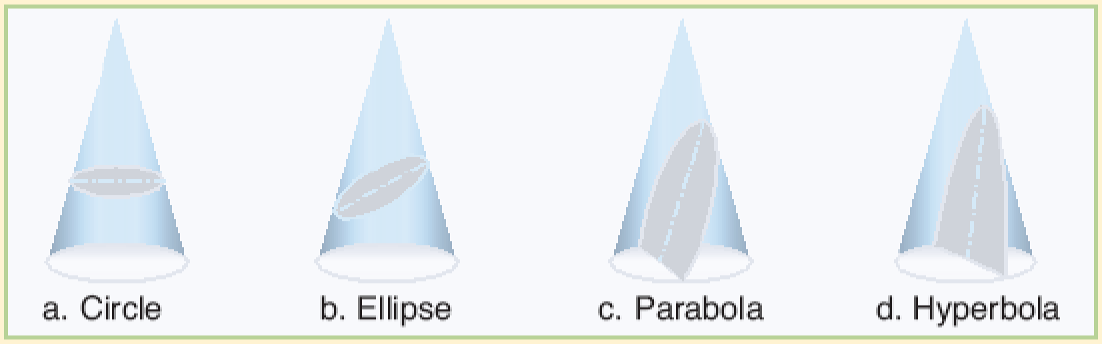


Geometry II: Lens Surface Type

Conic

$$z = \frac{cr^2}{1+\sqrt{1-(1+k)c^2r^2}} \quad \text{where} \quad r^2 = x^2 + y^2$$

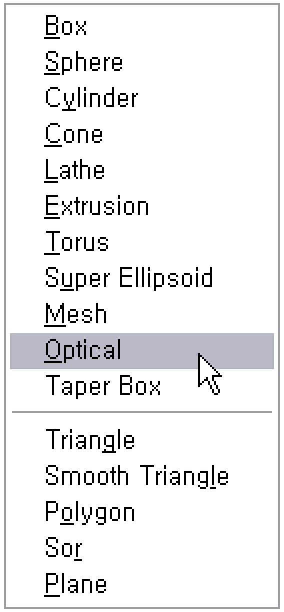
k = 0		sphere
–1 < k < 0		ellipse with major axis on the z axis
k = –1		parabola
k < –1		hyperbola
k > 0		oblate sphere (not a true conic section)

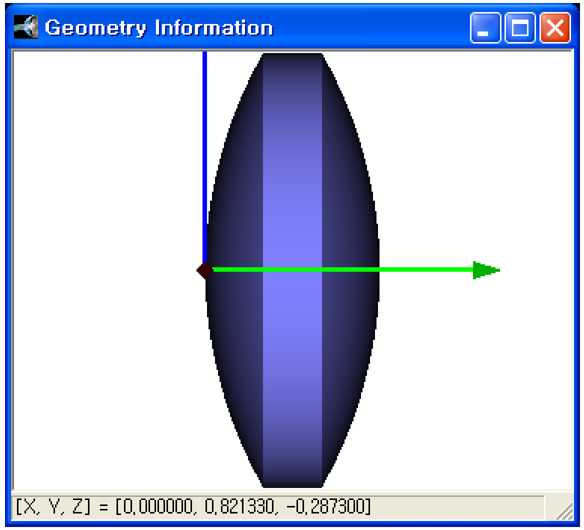


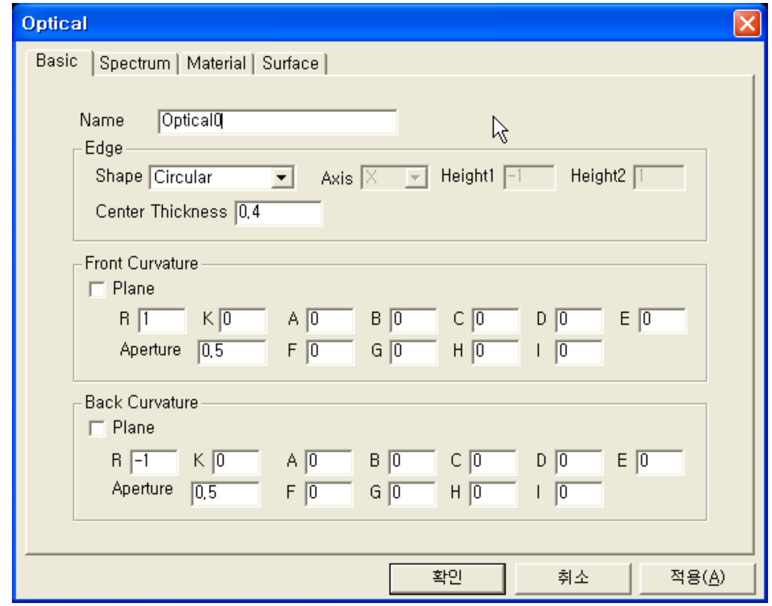
Polynomial Aspheric

$$z = \frac{cr^2}{1+\sqrt{1-(1+k)c^2r^2}} + \sum_{n=2}^{10} C_{2n} r^{2n} \quad \text{where} \quad r^2 = x^2 + y^2$$

Geometry II: Lens & Reflector (1)

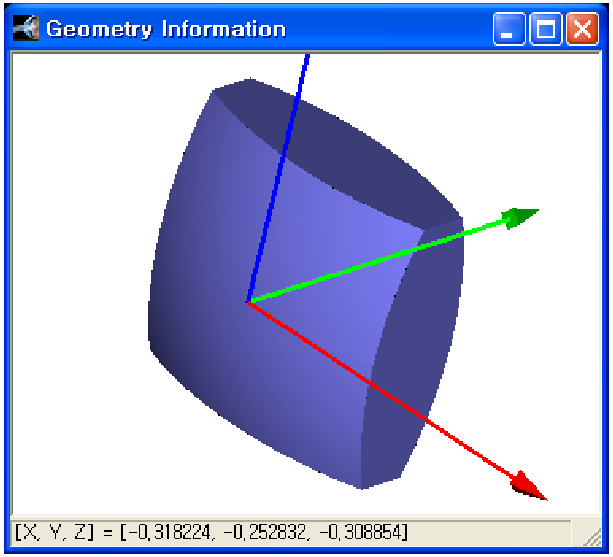


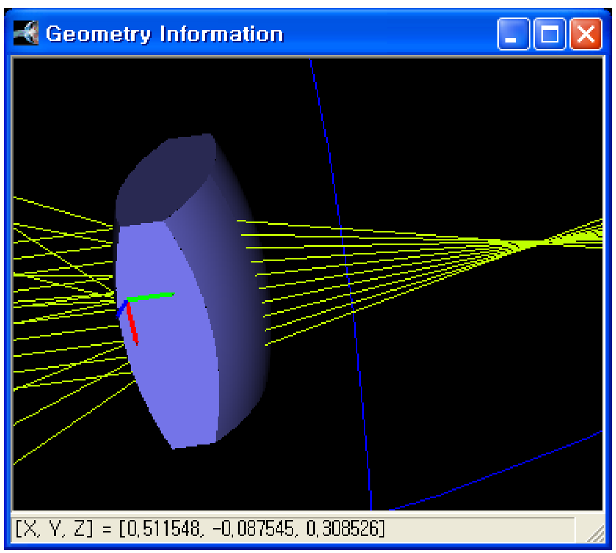





```
#declare Optical0 =
optical
{
    circular center_thickness 0.4
    { 1, 0.5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 }
    { -1, 0.5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 }
    color <1, 1, 1>
    ior 1.5
    surface {
        both {
            probabilistic_split on
        }
    }
}


Optical0
```

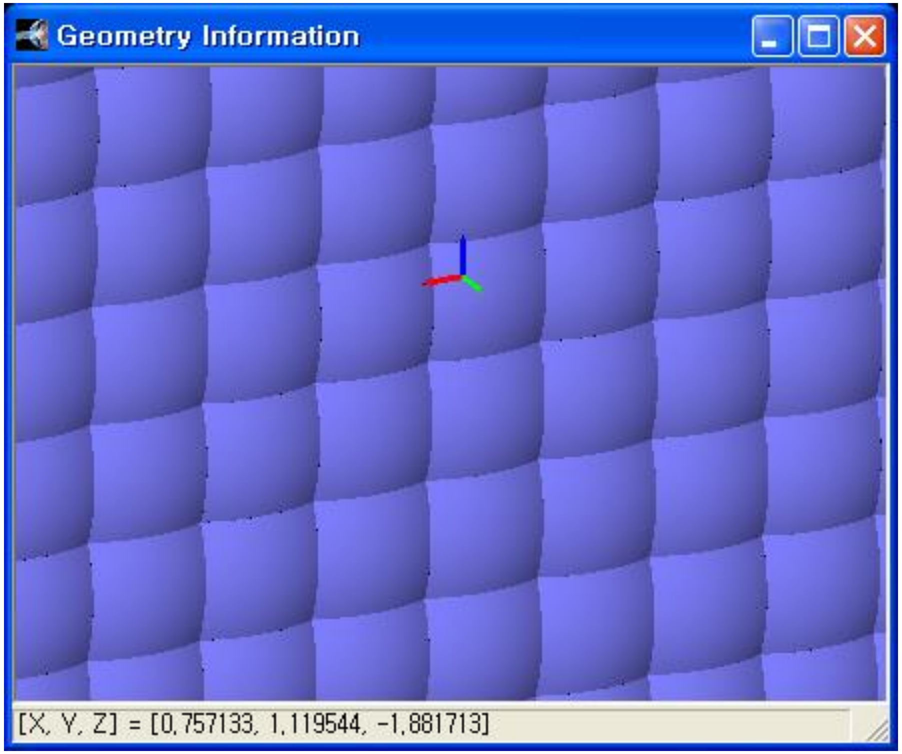
Geometry II: Lens & Reflector (2-1)

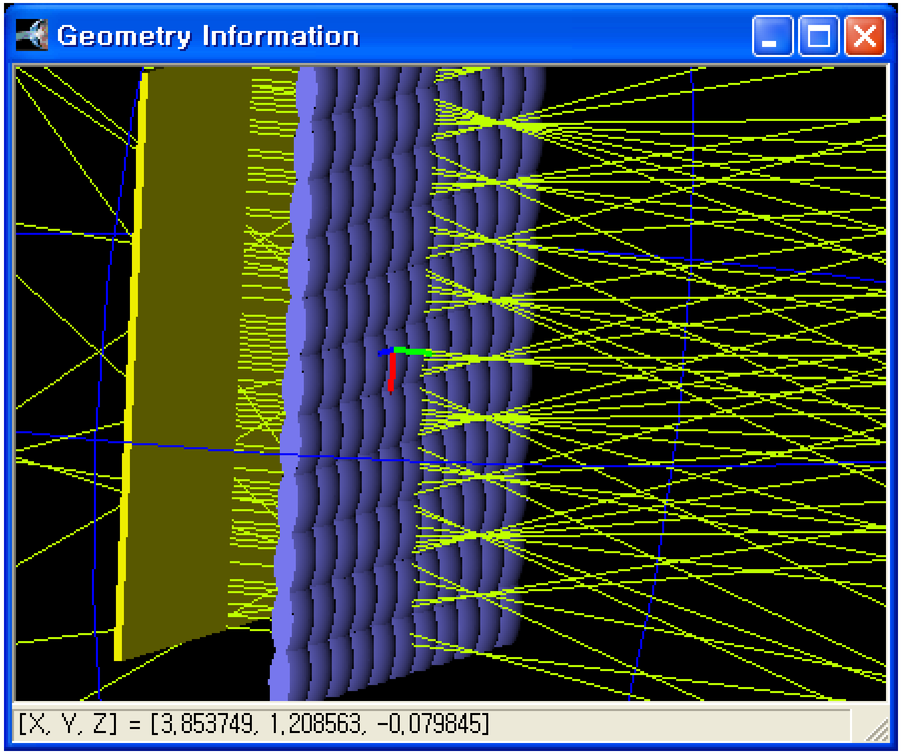


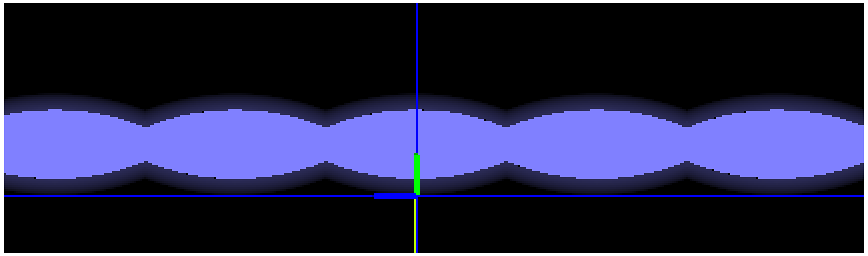



```
#declare Optical1 =
optical
{
    rectangular center_thickness 0.4
    { 1, 0.5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 }
    { -1, 0.5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 }
    color <1, 1, 1>
    ior 1.5
    surface {
        both {
            probabilistic_split on
        }
    }
}

Optical1
```

Geometry II: Lens & Reflector (2-2)

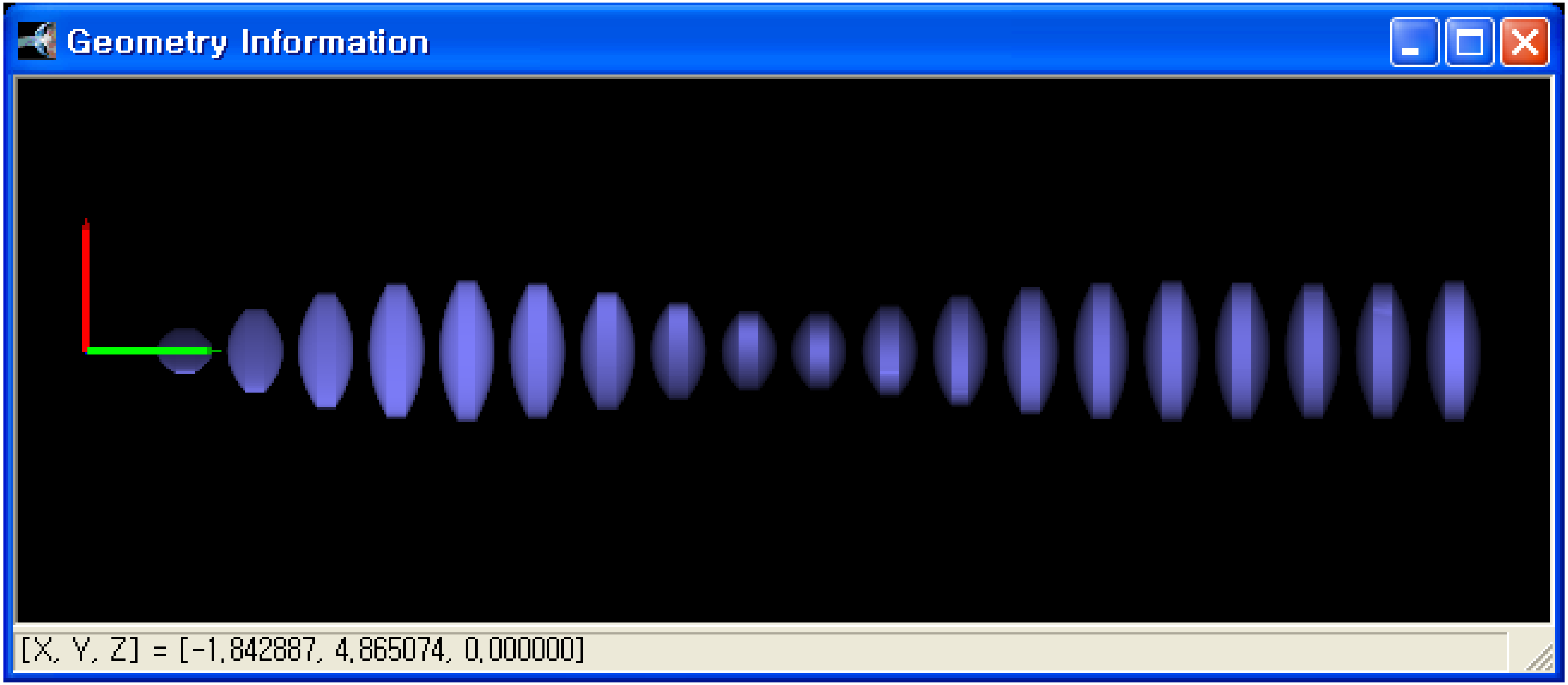






```
#macro FlyEyeLens()
    #declare ii=0;
    #while(ii<11)
        #declare jj=0;
        #while(jj<11)
        object{
                Optical1
                translate<(ii-5)*sqrt(2)/2,0,(jj-5)*sqrt(2)/2>
        }
        #declare jj=jj+1;
        #end
    #declare ii=ii+1;
    #end
#end
```

Workshop 5: #declare, #macro, #while (1)



그림과 같이 **20**개의 렌즈가 점진적으로 회전하고 크기가 바뀌는 배열을
#macro와 **#while**문을 사용하여 작성해 보시오.
[곡률반경**: 1, -1, Apetrue Size : 0.5]**

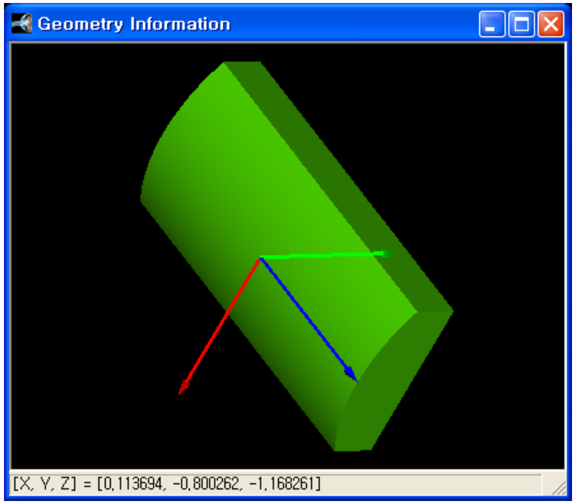
Workshop 5: #declare, #macro, #while (2)

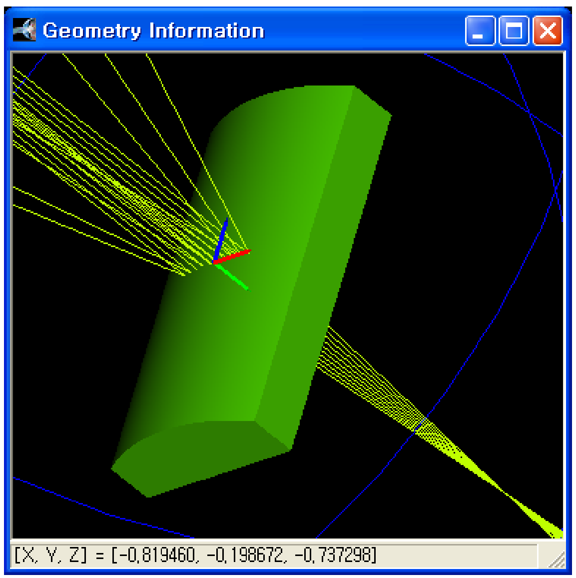
```
#declare NumUnit=20;
#declare NumScaleStep=NumUnit;
#declare Lens2Lens=0.5;
```

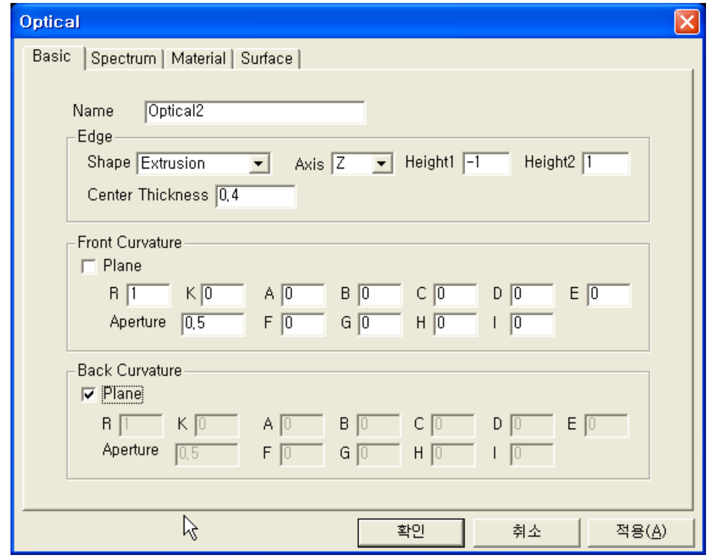
```
#declare LensUnit =
optical
{
    circular center_thickness 0.4
    { 1, 0.5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 }
    { -1, 0.5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 }
    color<1,1,1>
    ior 1.5
    surface {
        both {
            probabilistic_split on
        }
    }
}
```

```
#macro LensArray()
#declare ii=0;
    #while(ii < NumUnit)
    object{
        LensUnit
        translate<0,Lens2Lens*ii,0>
        scale<1/NumUnit*(ii+1),1,1>
        rotate<0,360/NumUnit*ii,0>
    }
    #declare ii=ii+1;
    #end
#end

LensArray()
```

Geometry II: Lens & Reflector (3)

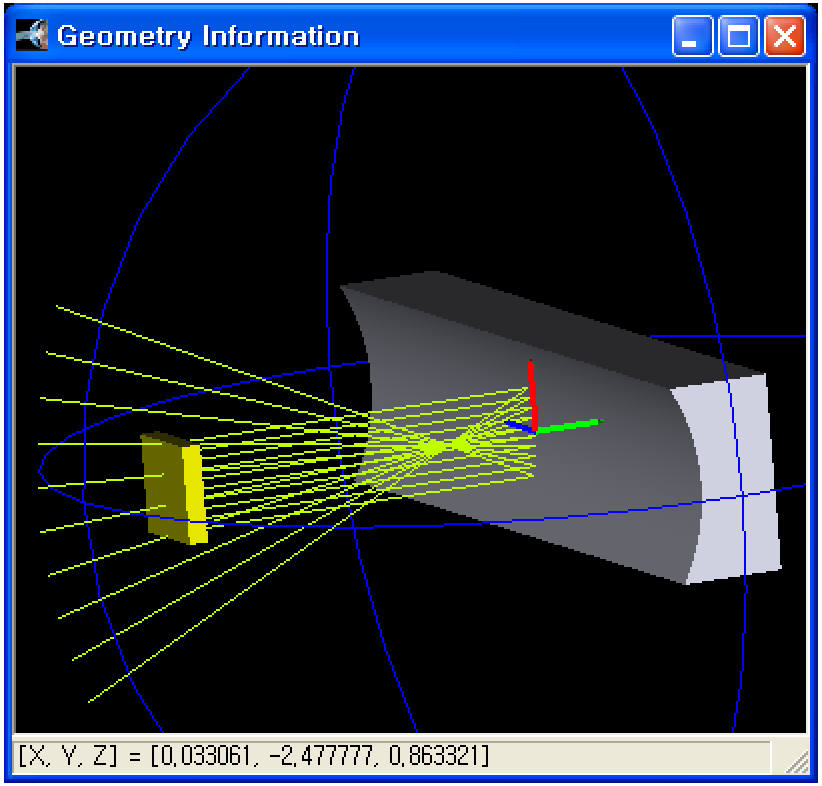







```
#declare Optical2 =
optical
{
    extrusion { z_axis, -1, 1 } center_thickness 0.4
    { 1, 0.5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 }
    { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 }
    spectrum {
        1,
        <550, 1>
    }
    ior 1.5
    surface {
        both {
            probabilistic_split on
        }
    }
}

Optical2
```

Geometry II: Lens & Reflector (4)



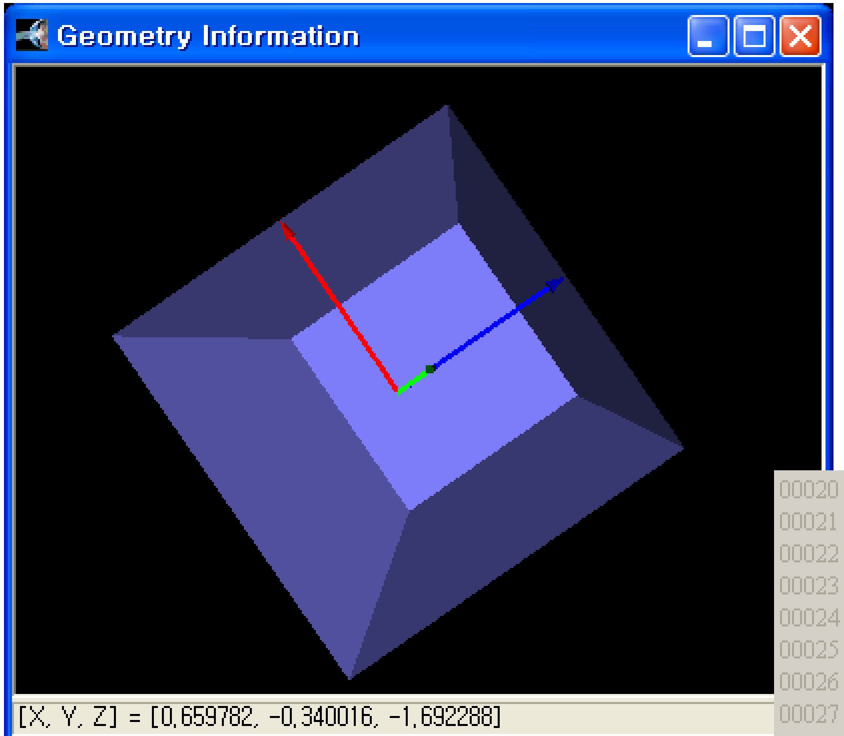

```
#declare Optical3 =
optical
{
    extrusion { z_axis, -2, 2 } center_thickness 0.4
    { -1, 0.5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 }
    { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 }
    color silver
    surface {
        both {
            reflectance{0.92}
            transmittance{0}
        }
    }
}

Optical3
```

- 불투명체로 **IOR**을 삭제하고 반사 투과율을 지정한다.

Geometry II: Taper Box

Box
Sphere
Cylinder
Cone
Lathe
Extrusion
Torus
Super Ellipsoid
Mesh
Optical
Taper Box

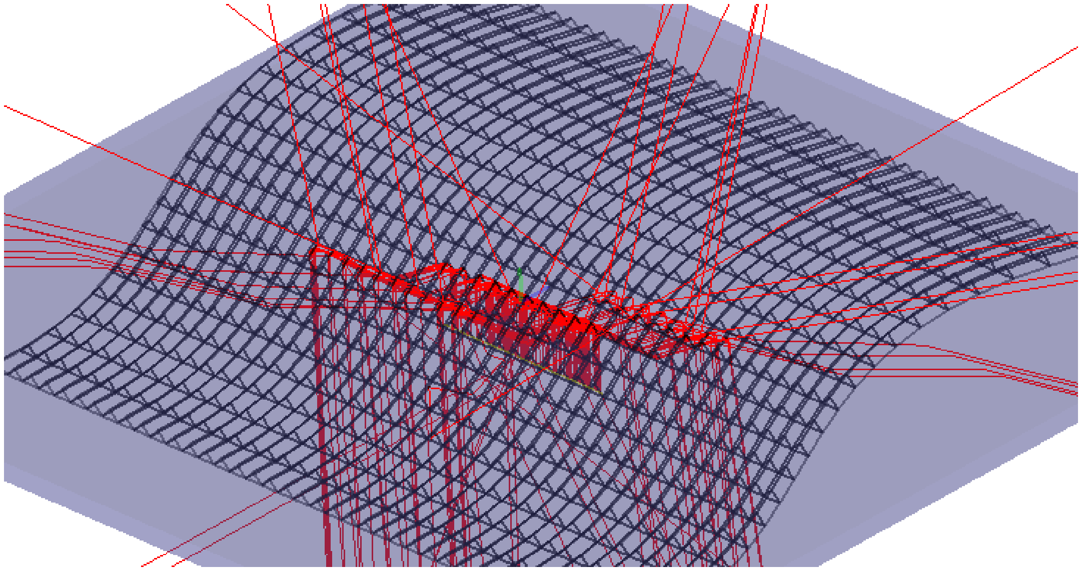
Triangle
Smooth Triangle
Polygon
Sor
Plane

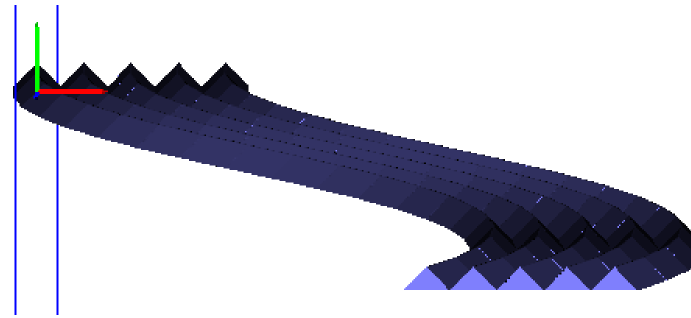


front_surface(Z+)
back_surface(Z-)
left_surface(X-)
right_surface(X+)
top_surface(Y+)
bottom_surface(Y-)

```
#declare TaperBox0 =
taper_box
{
    bottom { <-1, 0, -1>, <1, 0, 1> }
    top { <-0.5, 1, -0.5>, <0.5, 1, 0.5> }
    color <1, 1, 1>
    ior 1.5
    surface {
        both {
            probabilistic_split on
        }
    }
}

TaperBox0
```

Geometry II: Skew Object – BEF3



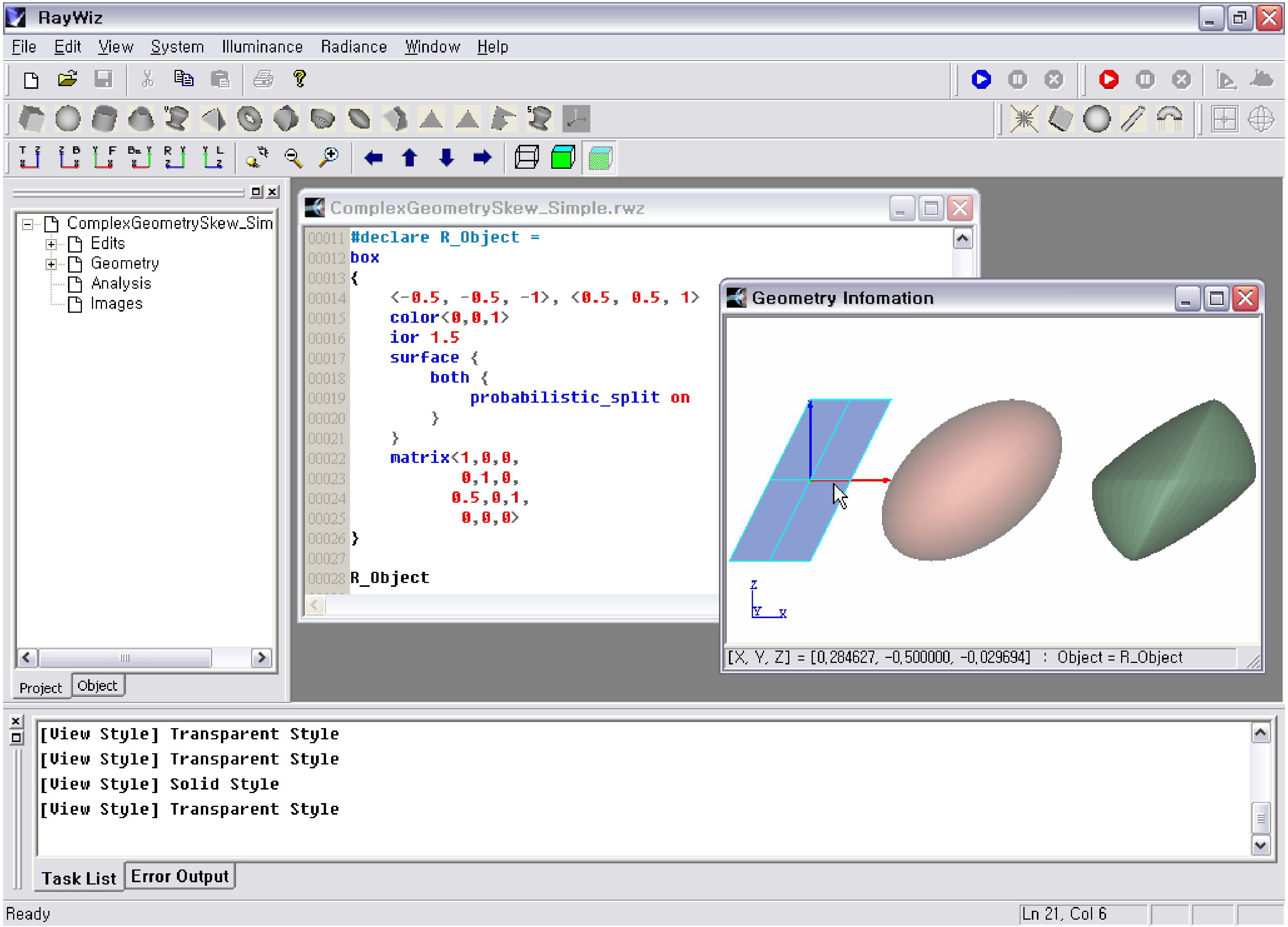


```
#declare SkewXX=Skew_val*sin(0.2*ii);
#declare XX=SkewXX;
matrix<1,0,0,
       0,1,0,
       SkewXX,0,1,
       XX,0,2*ii>
```

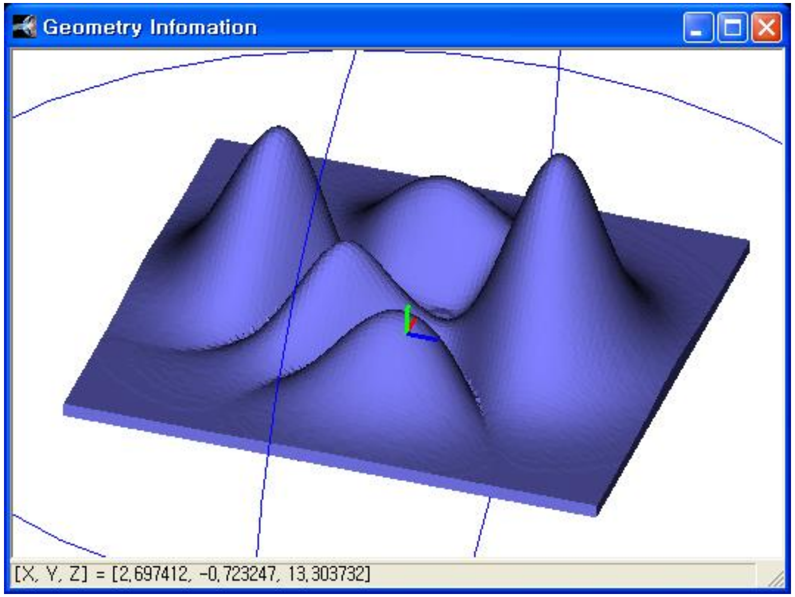
translate

X-Axis Skew

Geometry II: Skew Object

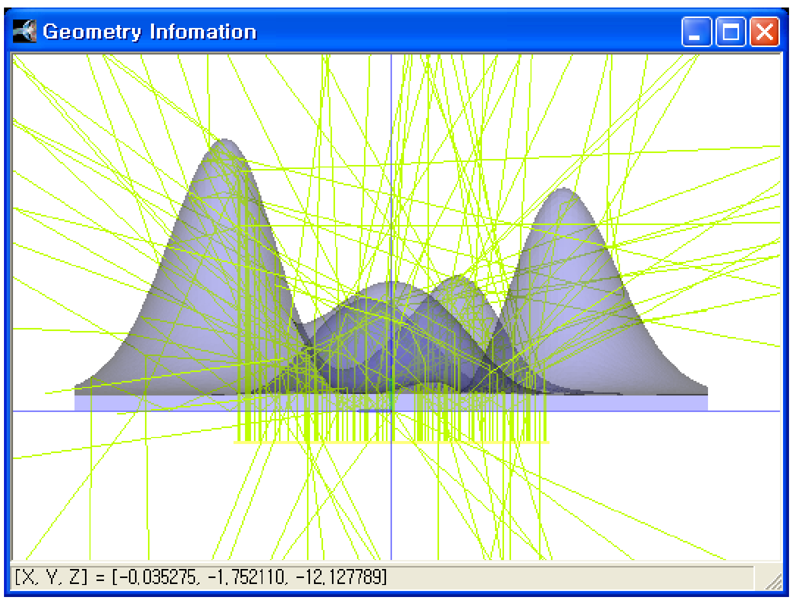


Geometry II: Matrix_Mesh (1)



MatrixMesh.rwz

```
//File을 읽어 배열에 할당
#fopen DataFile "exp.txt" read
#read(DataFile, NumR1, NumC1)
#declare NumR=NumR1;
#declare NumC=NumC1;
#declare DataArray = array[NumR][NumC]
#declare I = 0;
#while ( I < NumR )
    #declare J = 0;
    #while (J < NumC)
        #read (DataFile, MyData)
            #declare DataArray[I][J] = abs(MyData);
    #declare J = J+1;
    #end
#declare I = I+1;
#end
#fclose DataFile
```



MatrixMesh.rwz

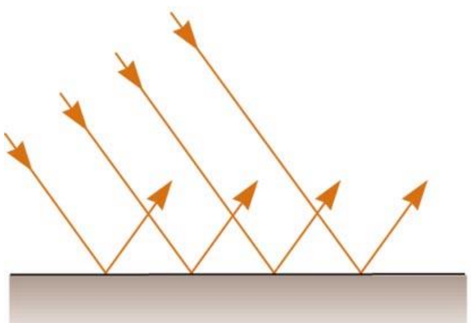
```
mesh_array  {
    boundary    {
        <-10, 10>
        <-10, 10>
    }
    thickness 0.5
    DataArray
    color<1,1,1>
    ior 1.5
    surface {
        both {
            probabilistic_split on
        }
    }
    //scale<1,1,1>
}
```

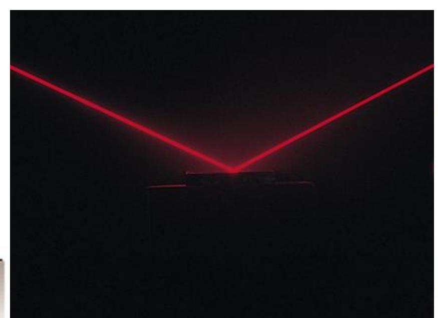
RAYWIZ BASIC
Optical Properties I

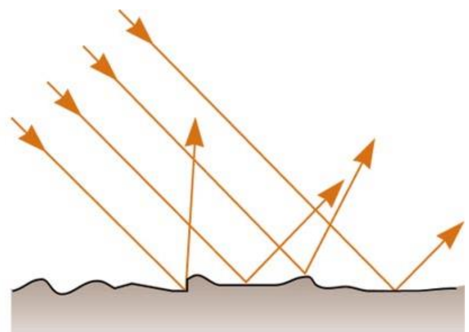
Optical Property I: Specular & Diffuse (Scatter)

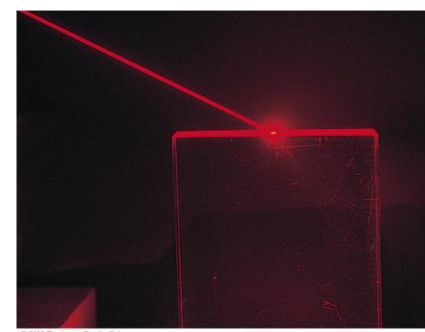
Specular

Diffuse



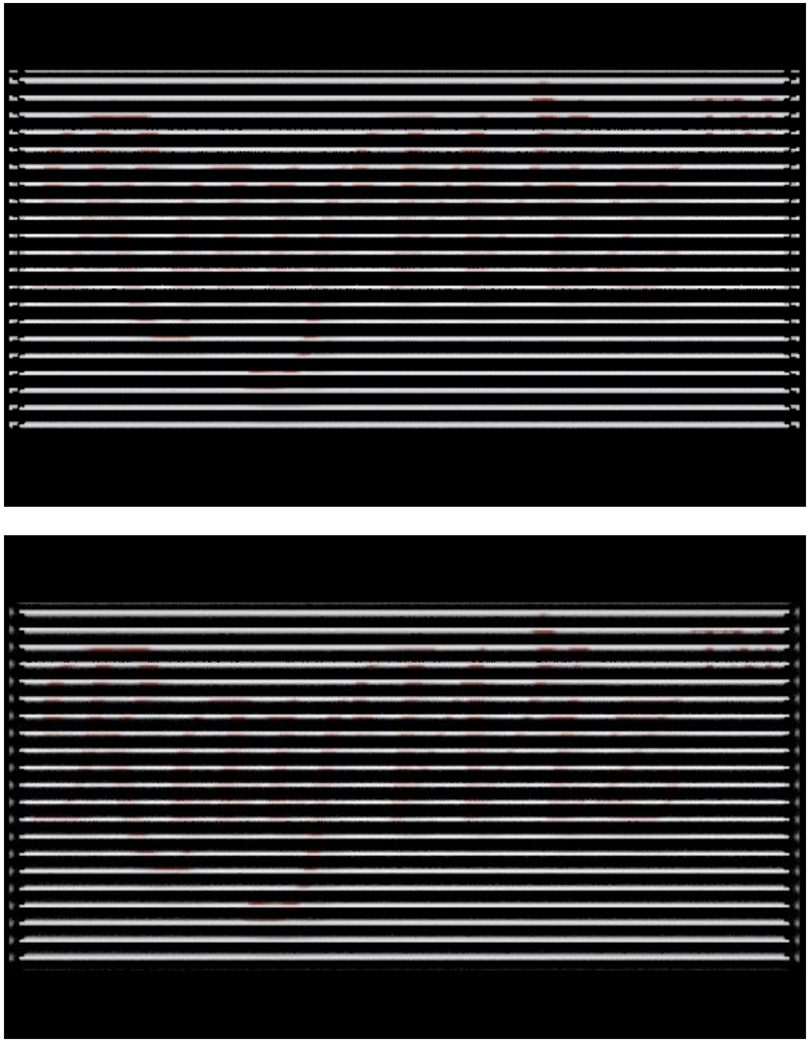




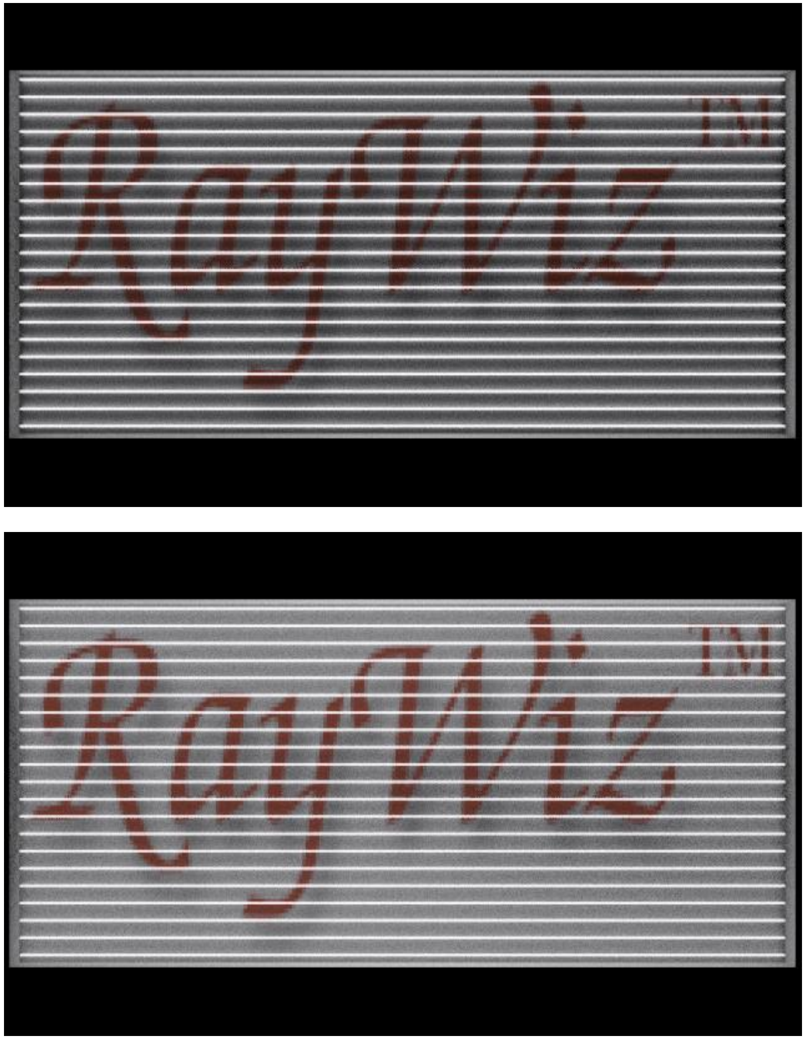


Optical Property I: Specular & Diffuse (Scatter)

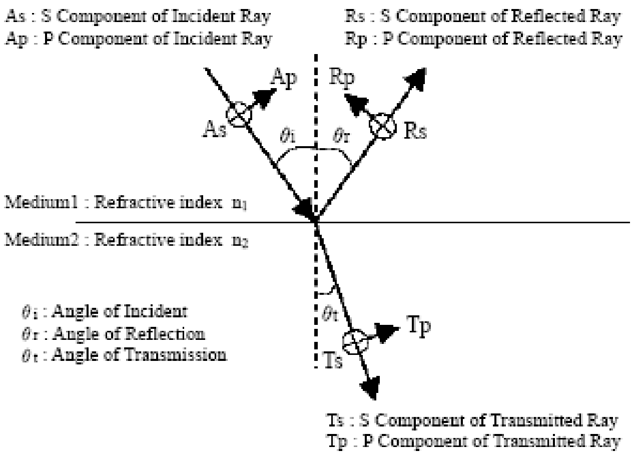
Specular

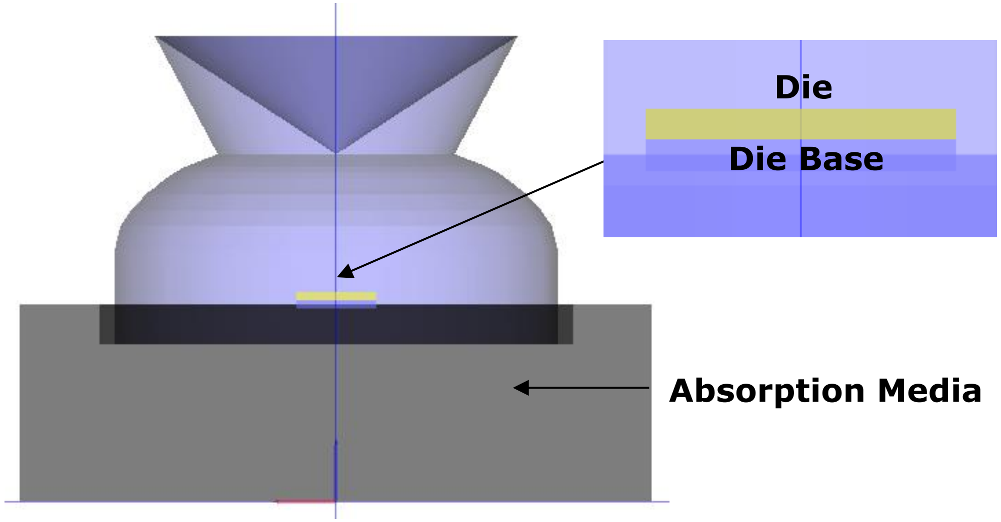


Diffuse

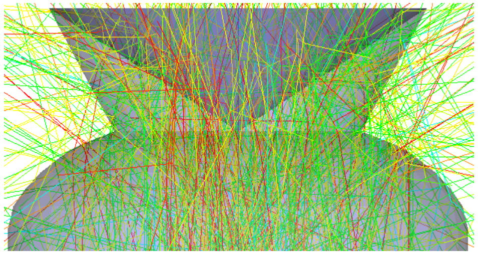


Optical Property I: Specular [Ray & Surface]





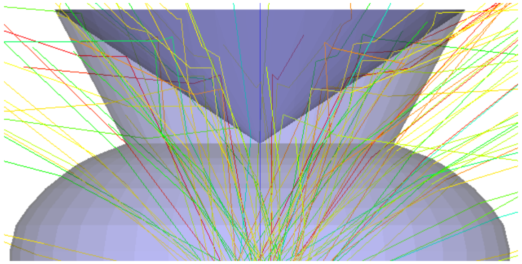
Input Energy =
Reflected Energy + Transmitted Energy



<Split_Ray On>

```
surface {
    both
    {
        Probabilistic_split on
    }
}
```

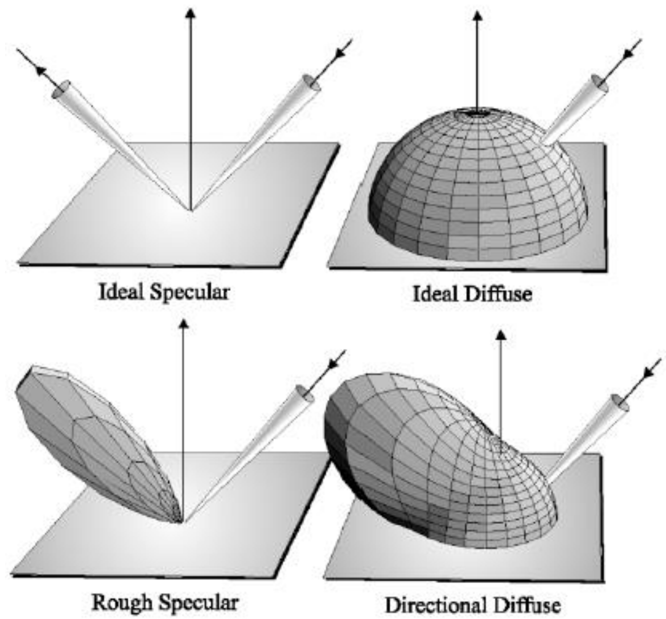
Input Energy =
Reflected Energy or Transmitted Energy



<Split_Ray Off>

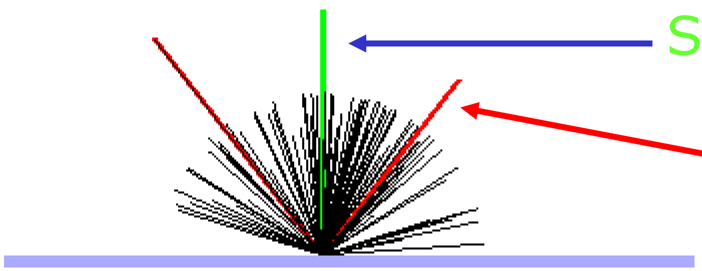
Optical Property I: Scatter Models

- Rays spread after reflection/transmission
- Ray Propagation Direction can be Transmitted (forward scatter), Reflected (back scatter) or both
- Specify scattering distribution
  - Most distributions centered about the specular (specular: direction of the ray that obeys Snell's law)



Optical Property I: Lambertian Scattering

- Random scattering surface (white paint)
- Each incident ray can scatter in any direction with the same probability
- Scattering occurs about the surface normal and *not* the incident angle
- Each scattered ray has the same energy



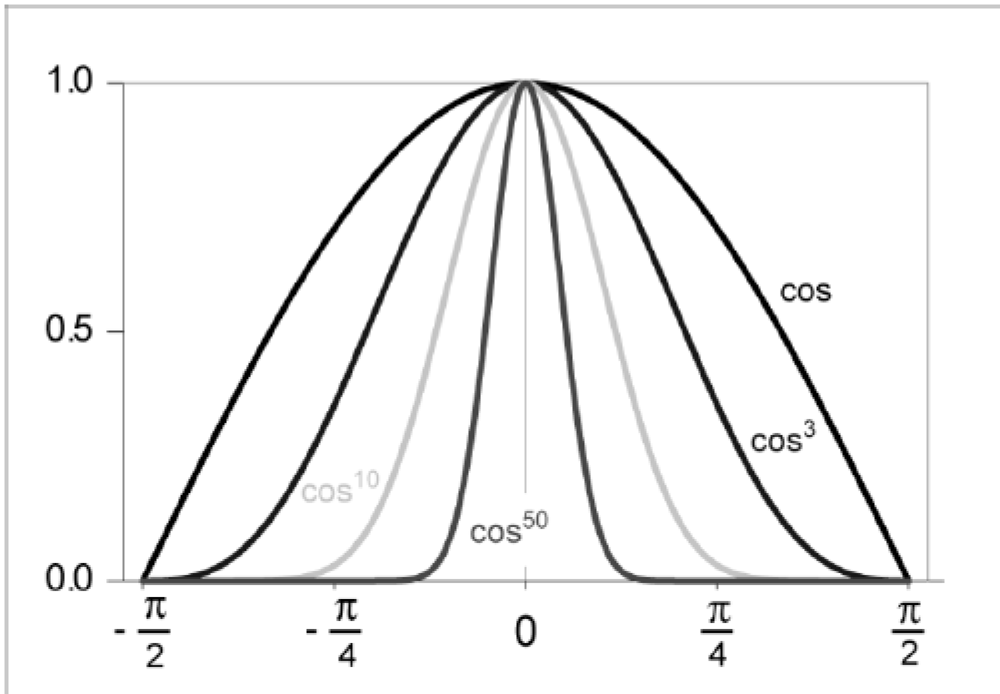
Optical Property I: Gaussian Scattering

- Each incident ray can scatter with the Gaussian distribution given by

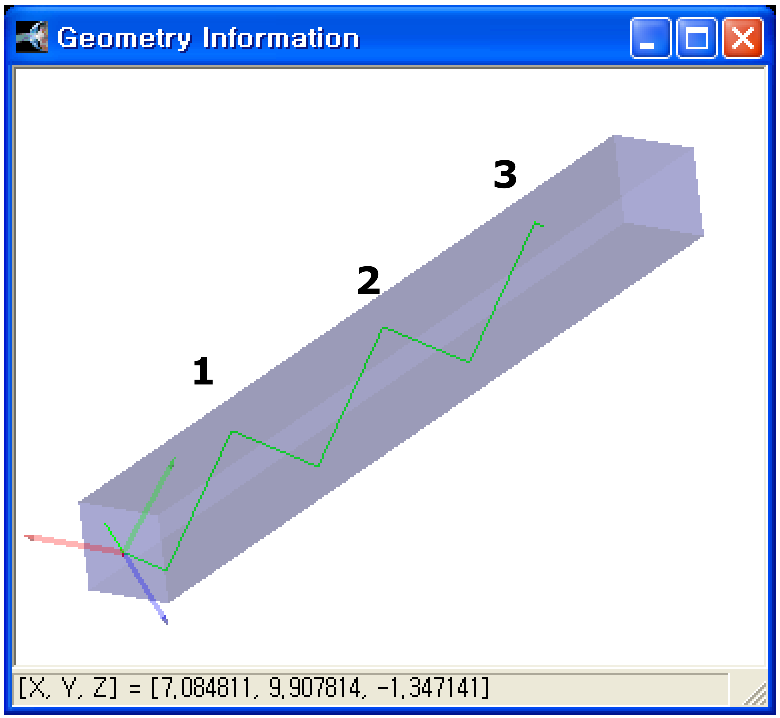
$$P(\theta) = P_o \exp\left[\left(-\frac{1}{2}\right)\left(\frac{\theta}{\sigma}\right)^2\right]$$

  where
  $P(\theta)$ = intensity or radiance in the θ direction
  P_o = intensity or radiance in the specular direction
  σ = standard deviation of the Gaussian distribution, in degrees
- Useful for near-specular or narrow distribution scatter

Optical Property I: Cosine Nth



Optical Property I: max_trace_level & trace_off_ratio



MaxTraceLevel.rwz:1

```
global_settings {
    rays {
        trace_off_ratio 0.01
        total_ray_number 1000
        max_trace_level 3
        trace_path_view 100
        split_ray off
    }
}
```

-**max_trace_level: Object**를 구성하는 각 **Surface**에 **Ray**가 몇 번 맞을 때 까지 **tracing**을 진행할 것인지를 결정.

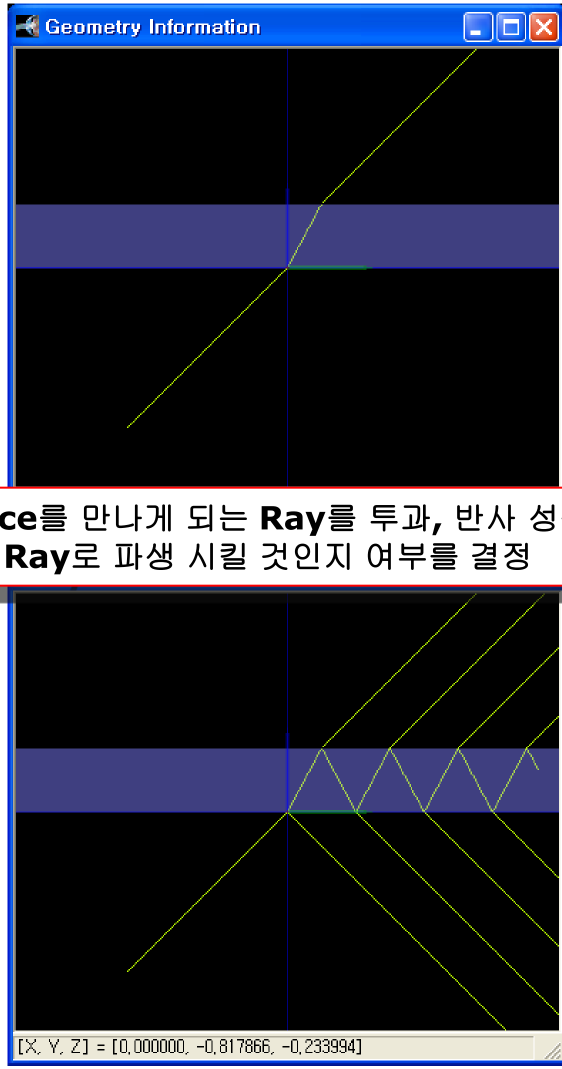
- **trace_off_ratio: Ray**가 처음 출발할 시점의 에너지 대비 얼마만큼의 비율로 감소될 시점까지 **tracing**을 진행할 것인지를 결정.

- 두 가지 설정 부분의 내용 중 먼저 만족되는 조건을 따름.

Optical Property I: split_ray [on, off]

split_ray off

split_ray on



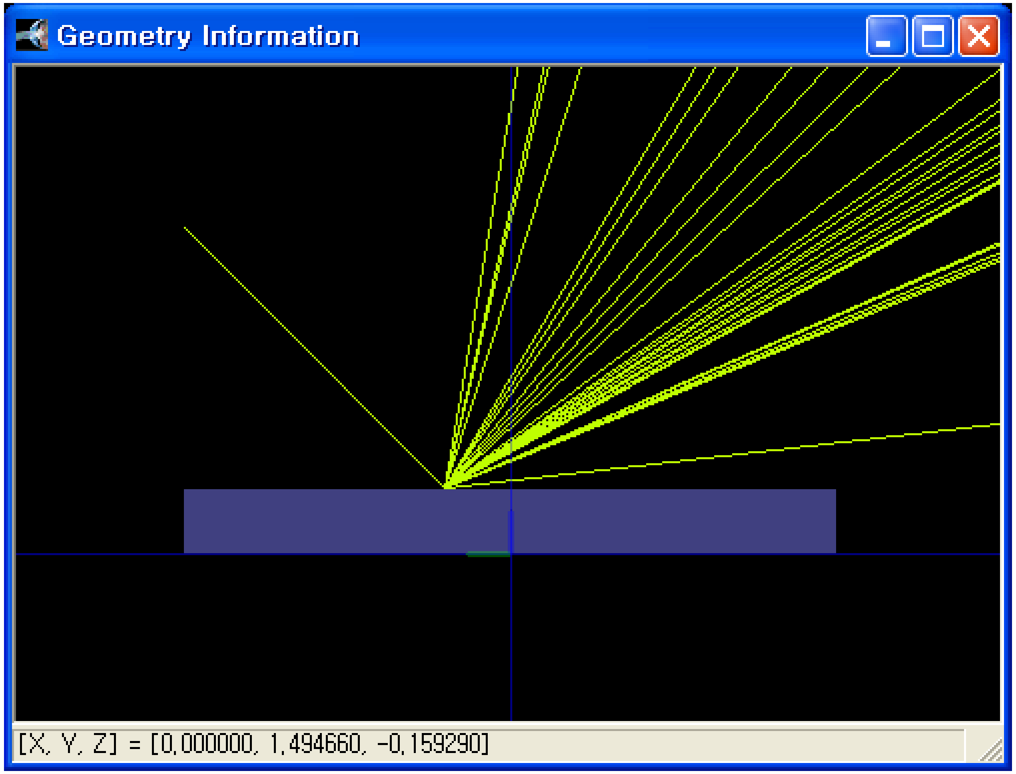
- **Surface**를 만나게 되는 **Ray**를 투과, 반사 성분으로 두개의 **Ray**로 파생 시킬 것인지 여부를 결정

```
#declare Box0 =
box
{
    <-1, -1, 0>, <1, 1, 0.2>
    color <1, 1, 1>
    ior 1.5
    surface {
        both {
            transmittance{0.6}
            reflectance{0.4}
        }
    }
}
```

```
global_settings {
    rays {
        trace_off_ratio 0.001
        total_ray_number 1
        max_trace_level 10
        trace_path_view 1
        split_ray off
    }
}
```

```
global_settings {
    rays {
        trace_off_ratio 0.001
        total_ray_number 1
        max_trace_level 10
        trace_path_view 1
        split_ray on
    }
}
```

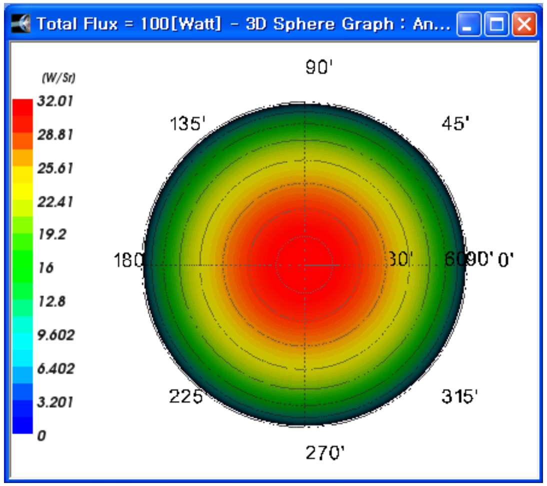
Optical Property I: Scattering(1)

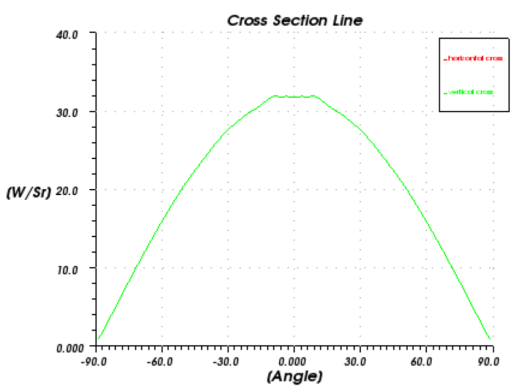



```
#declare Box0 =
box
{
    <-1, -1, 0>, <1, 1, 0.2>
    color <1, 1, 1>
    surface {
        both{
            reflectance{1.0}
        }
        front_surface {
            both {
                reflectance {
                    cosnth 10
                }
                scattering_num 30
            }
        }
    }
}
```

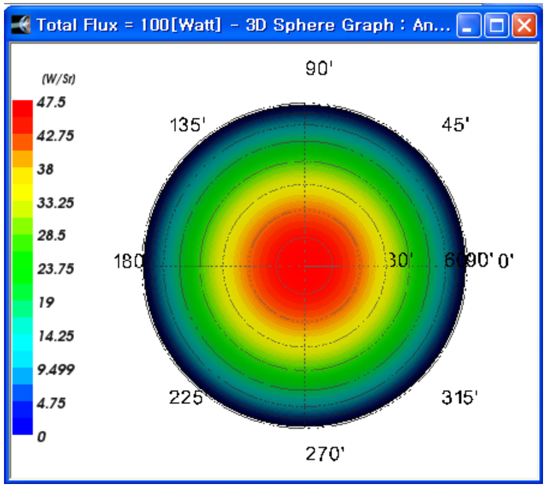
Optical Property I: Scattering(2) – Basic Function

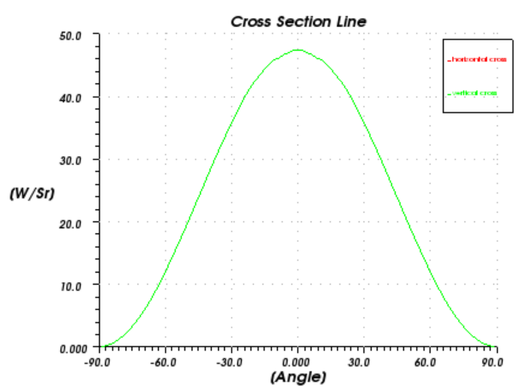
<Lambertian Type>



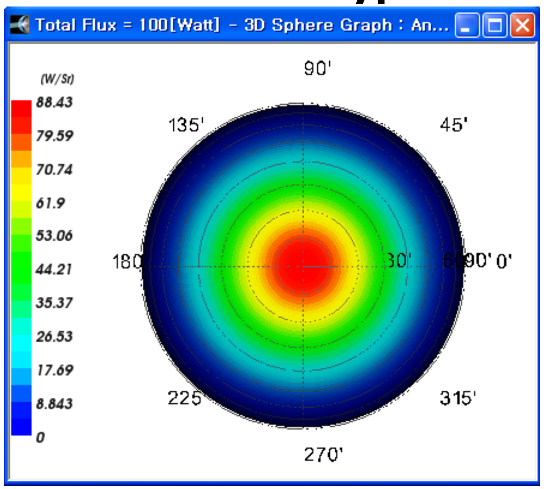


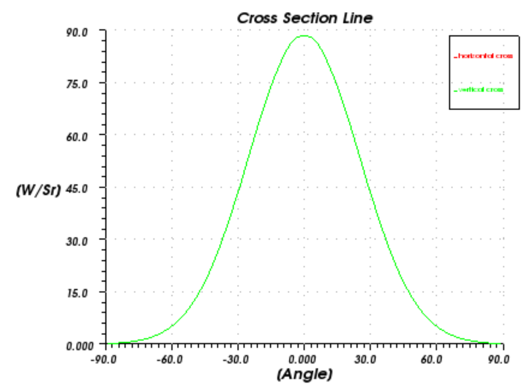
<CosNth Type>





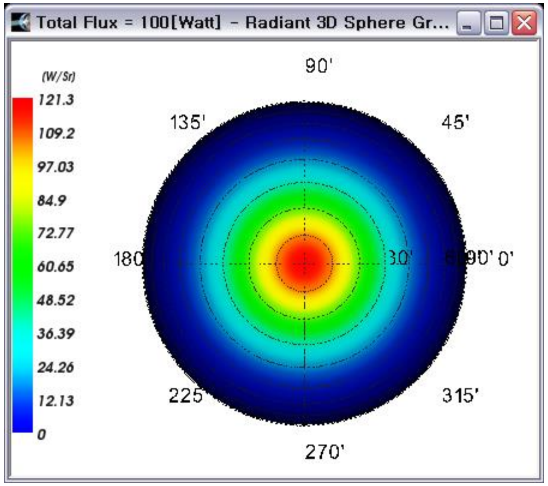
<Gaussian Type>

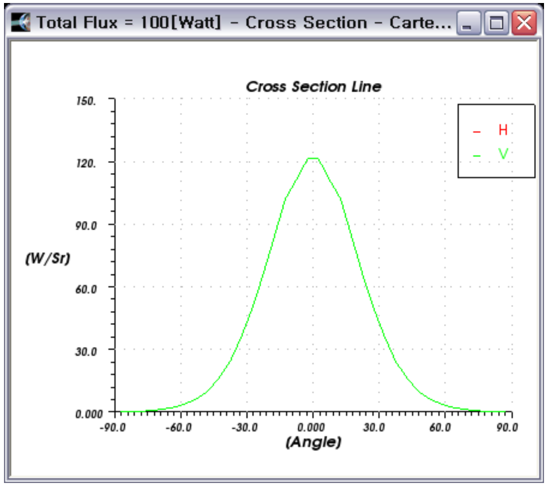




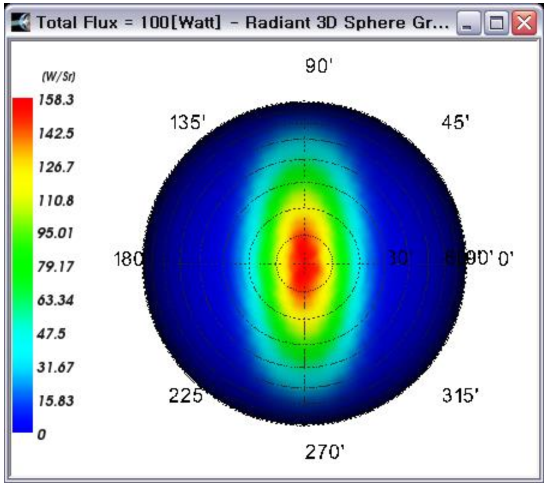
Optical Property I: Scattering(3) – Basic Function

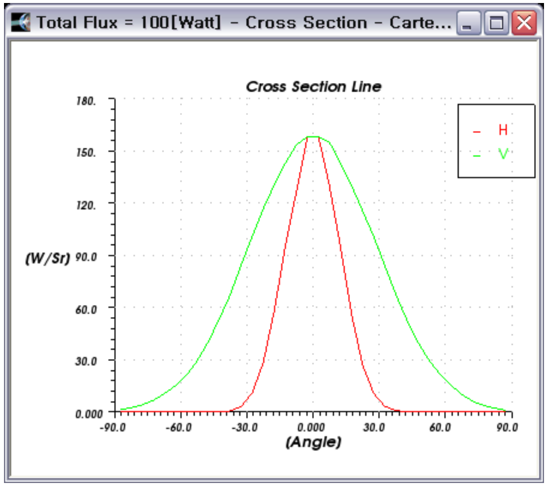
<Composite Type>





<Elliptical_Gaussian>



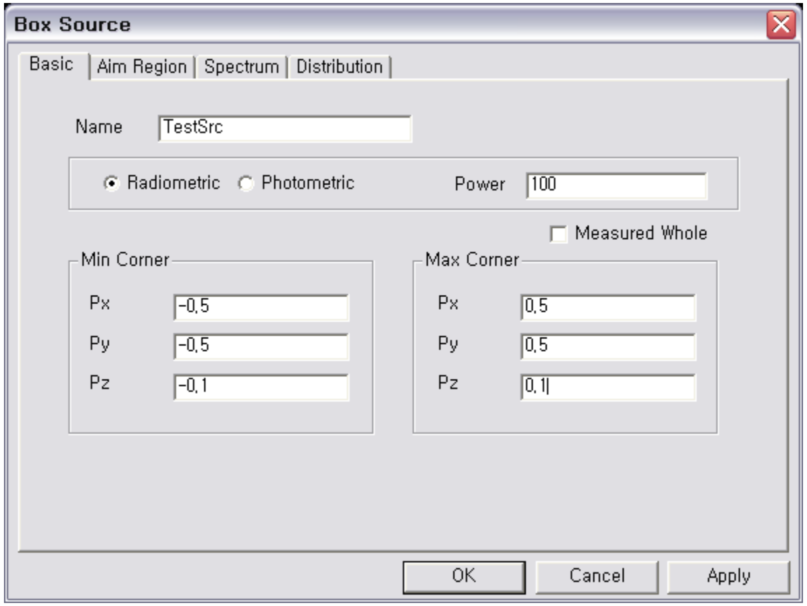


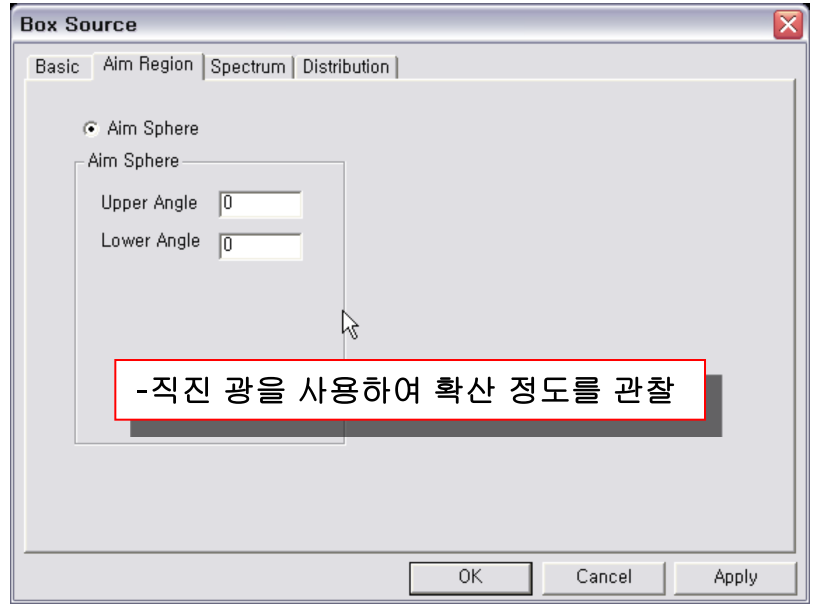
BSDF_Sample.rwz

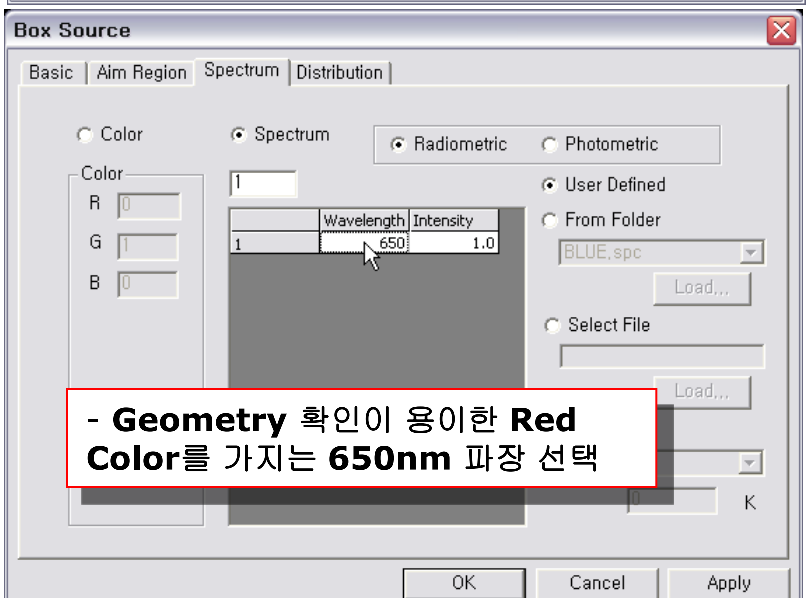
```
global_settings {
    rays {
        trace_off_ratio 0.01
        total_ray_number 100000
        max_trace_level 10
        trace_path_view 100
        split_ray off
        fresnel_area_effect on
        random_spectrum on
    }
    force_parsing off
    polarization off
    units millimeters
    summary on
    geometry_intensity on
}
```

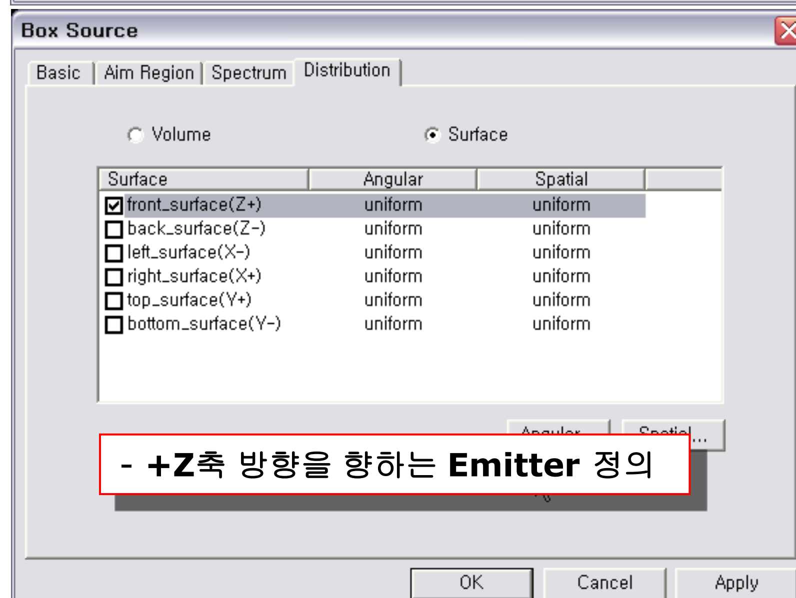
새로운 파일을 “**BSDF_Sample.rwz**”로 저장하고 **Global_Setting comment**를 해제하고 **Tracing Ray Number**를 **100000** 으로 수정한다**.**

Optical Property I: Scattering(5) – Basic Function









Optical Property I: Scattering(6) – Basic Function

BSDF_Sample.rwz

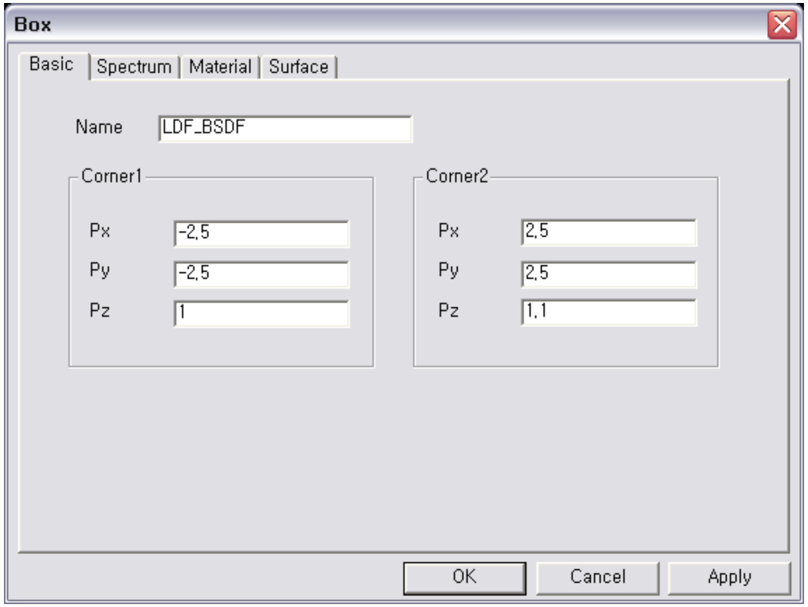
```
#declare TestSrc =
light_source
{
    <0, 0, 0>
    spectrum {
        1,
        <650, 1>
    }
    object_light
    {
        surface
        box
        {
            <-0.5, -0.5, -0.1>, <0.5, 0.5, 0.1>
        }
```

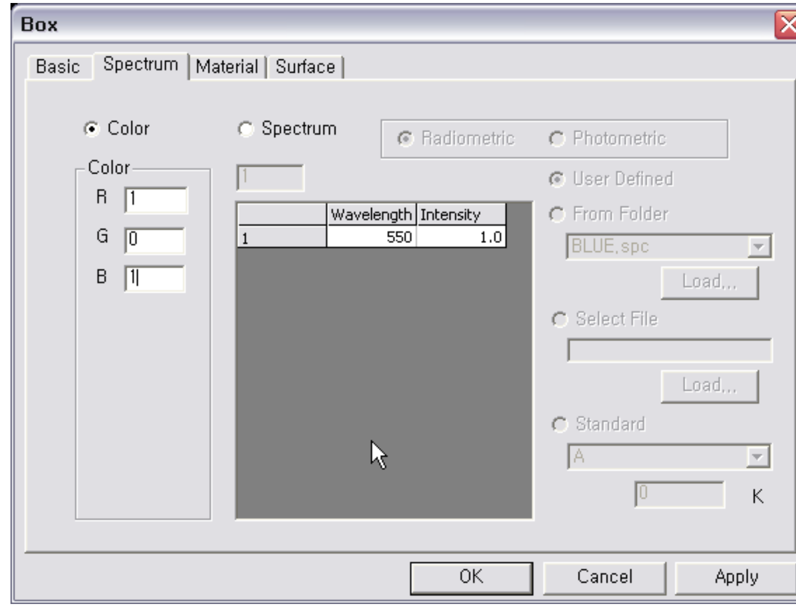
BSDF_Sample.rwz

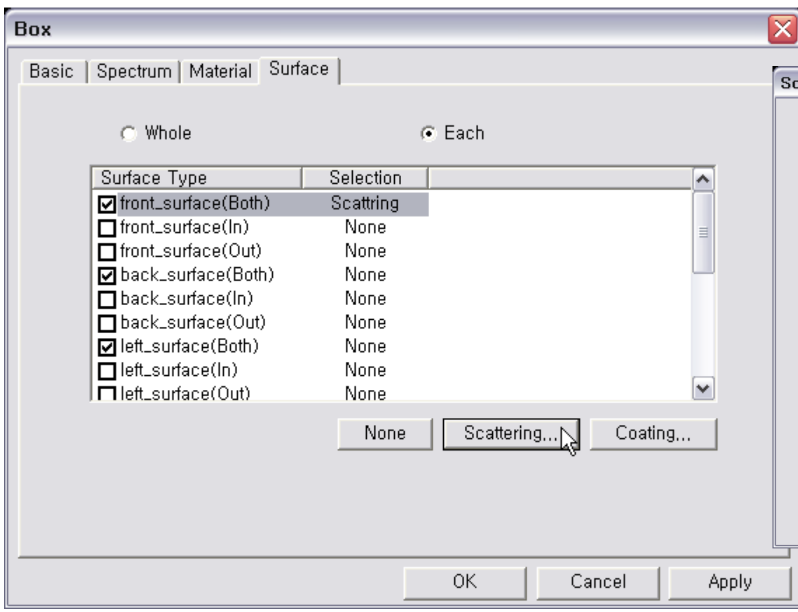
```
        emittance
        {
            front_surface
            {
                on
                angular_distribution
                {
                    uniform
                }
                spatial_distribution
                {
                    uniform
                }
            }
        }
    }
    emittance
    {
        radiometric 100
        measured_whole on
        aim_sphere <0, 0>
    }
    point_at <0, 0, 3>
}

TestSrc
```

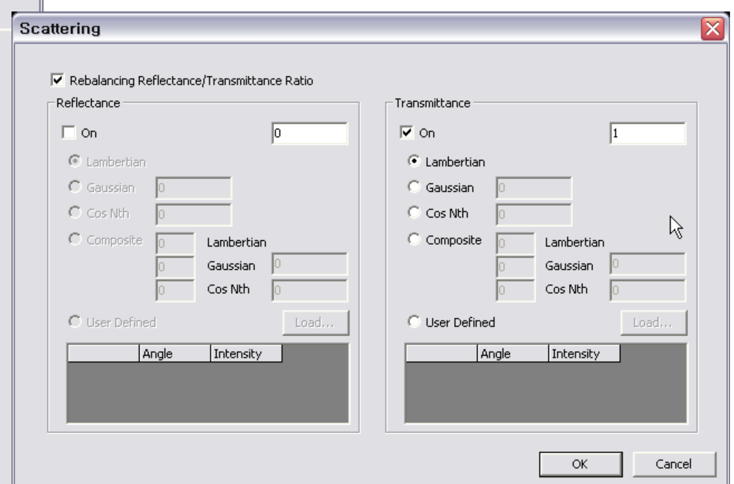
+Z 방향을 향하도록 **point_at** 좌표 지정

Optical Property I: Scattering(7) – Basic Function







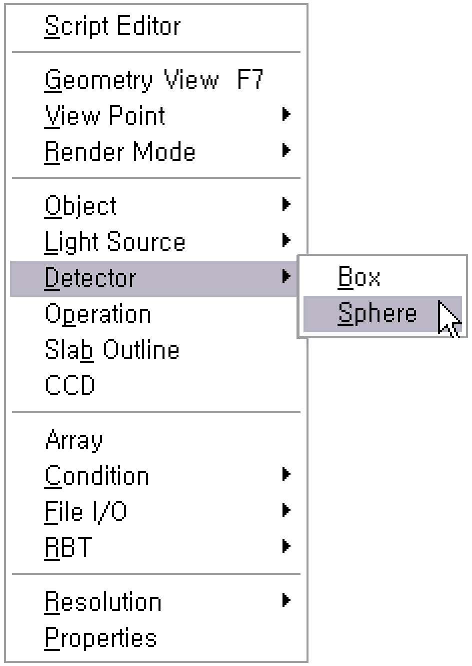


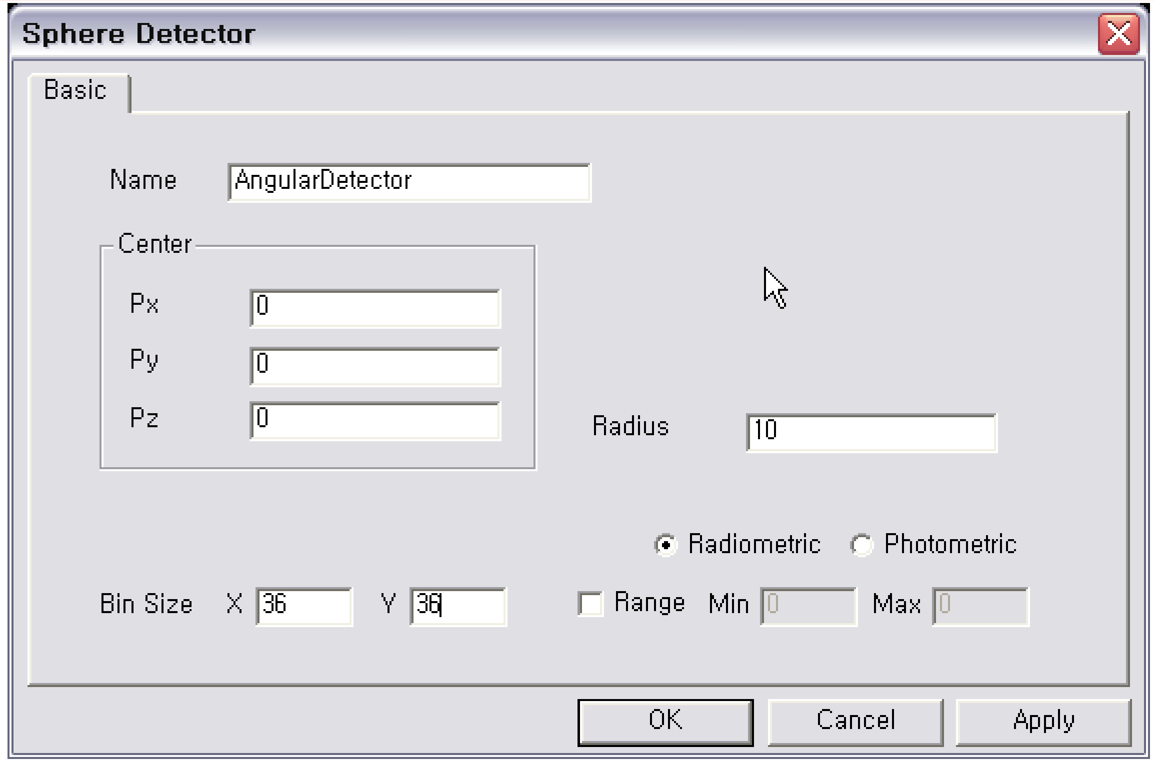
BSDF_Sample.rwz

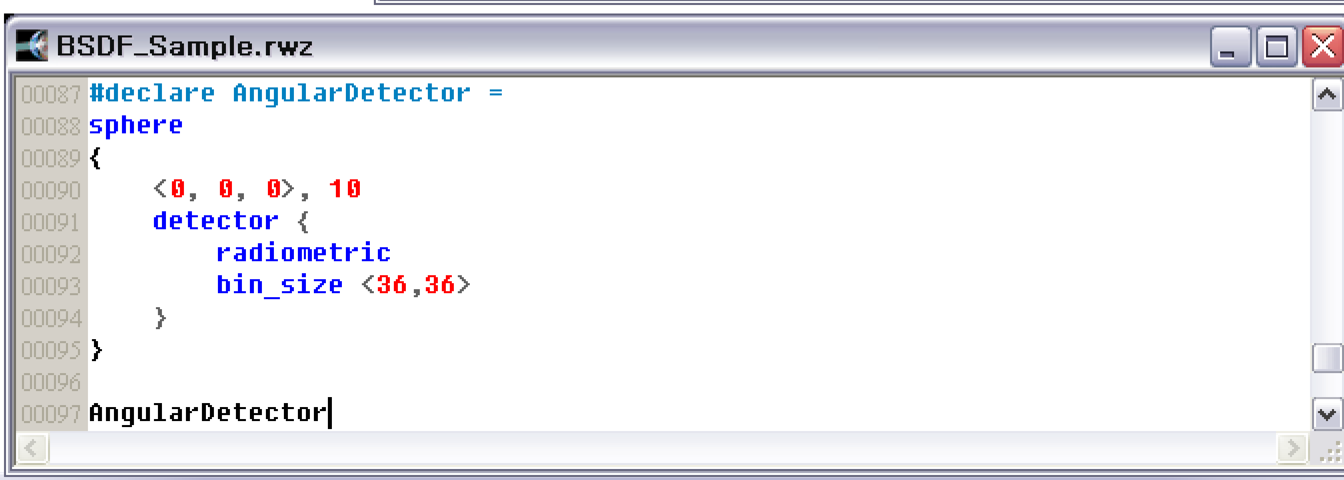
```
#declare LDF_BSDF =
box
{
    <-2.5, -2.5, 1>, <2.5, 2.5, 1.1>
    color<1,0,1>
    ior 1.5
    surface {
        both{
                probabilistic_split off
        }
        front_surface{
            both{
                transmittance{1
                  lambertian
                    //cosnth 4
                    //gaussian 25
                    //composite{0.2, 0.5, 0.3, 25, 10} //Le:Ge:Ce Sigma Nth*/
                    //elliptical_gaussian 10 30
                }
                reflectance{0}
            }
        }
    }
}

LDF_BSDF
```

확산특성은 투과 반사 비율 다음에 기록하거나 투과 반사 비율에 대한 항이 없으면 **Fresnel Equation**을 기준으로 **Simulation** 한다.

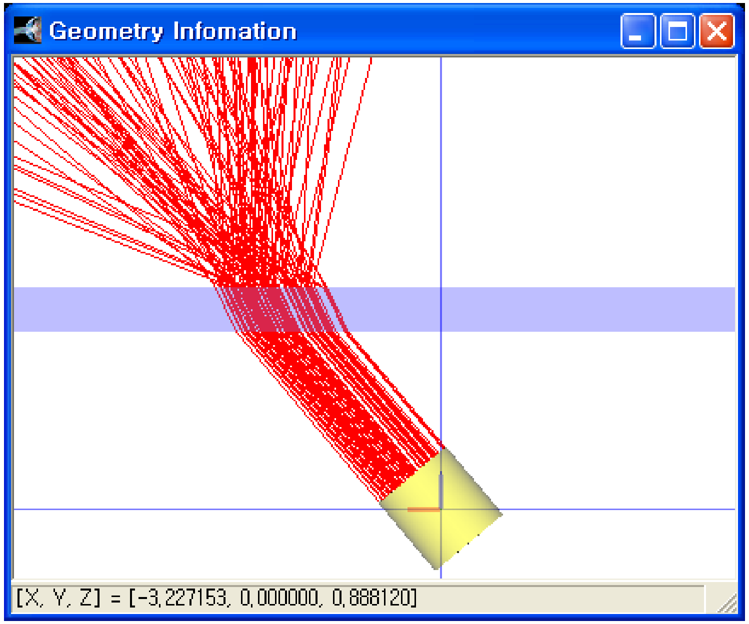
Optical Property I: Scattering(8) – Basic Function







Optical Property I: Scattering(9) – Angle_Intensity



Angle_Intensity_Angle2_Intensity.rwz

```
#declare TestBox1 =
box
{
    <-5, -5, 2>, <5, 5, 2.5>
    color<1,1,1>
    ior 1.5
    surface {
        both {
          probabilistic_split off
        }
        front_surface {
            both {
                transmittance {
                    angle_intensity {
                        //"angle_intensity.ain"
                        5, <0, 1>, <10, 0.8>, <20, 0.6>, <30, 0.4>, <40, 0.2>
                    }
                }
            }
        }
    }
}
```

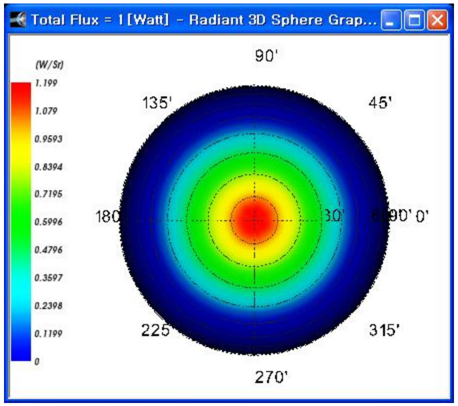
```
front_surface {
     both {
          transmittance {
               angle2_intensity {
                    //"angle2_intensity_data.ain"
                    3, 5,
                    { 0, <0, 1>, <10, 0.8>, <20, 0.6>, <30, 0.4>, <40, 0.2>}
                    { 15, <0, 0.8>, <10, 1>, <20, 0.8>, <30, 0.6>, <40, 0.4>}
                    { 30, <0, 0.4>, <10, 0.6>, <20, 0.8>, <30, 1>, <40, 0.8>}
               }
          }
     }
}
```

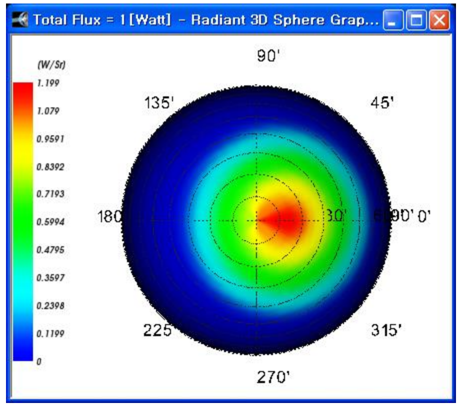
angle2_intensity_data.ain

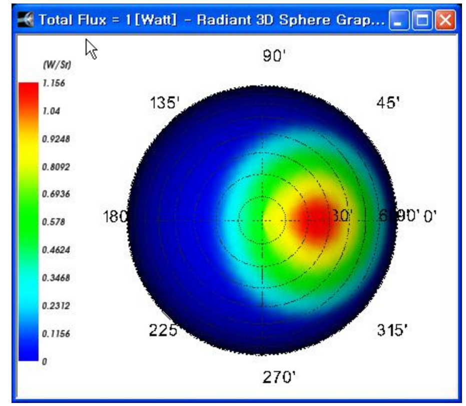
```
3 5
0
0 1.0
10 0.8
20 0.6
30 0.4
40 0.2
15
0 0.8
10 1.0
20 0.8
30 0.6
40 0.4
30
0 0.4
10 0.6
20 0.8
30 1.0
40 0.8
```

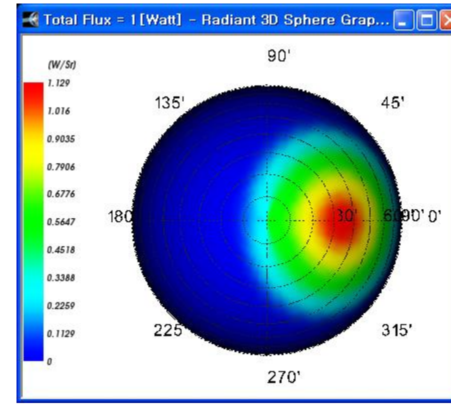
Optical Property I: Scattering(10) – Angle_Intensity

Angle_Intensity

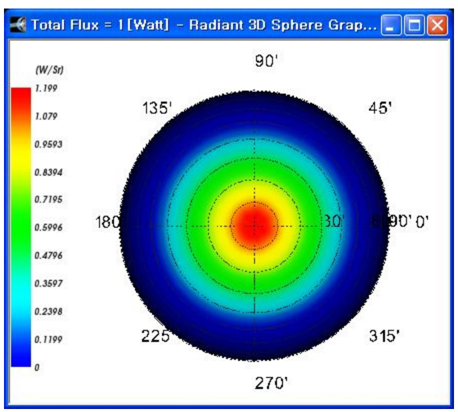


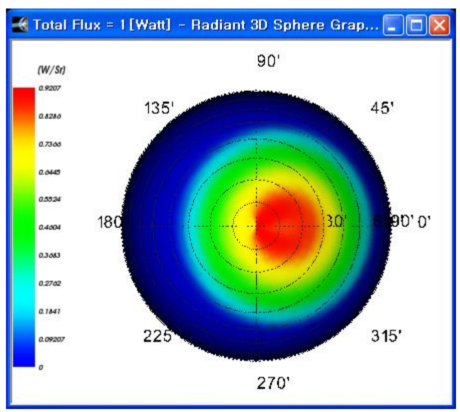


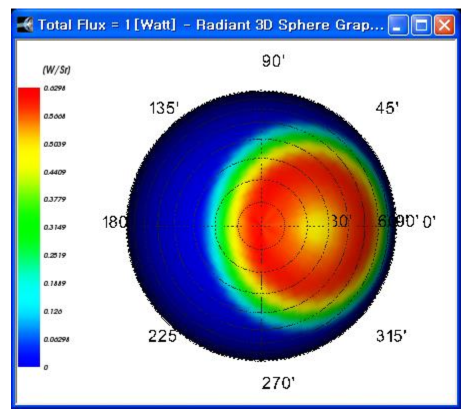


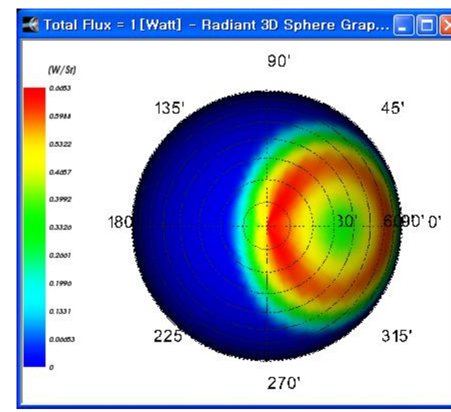


Angle2_Intensity









<0도 입사> **<20도 입사>** **<40도 입사>** **<60도 입사>**

Optical Property I: Coating(1)

```
<9, 3>
0,
<450, 0.0, 0.9>
<550, 0.1, 0.8>
<650, 0.2, 0.7>
10,
<450, 0.0, 0.9>
<550, 0.1, 0.8>
<650, 0.2, 0.7>
20,
<450, 0.0, 0.9>
<550, 0.1, 0.8>
<650, 0.2, 0.7>
30,
<450, 0.0, 0.9>
<550, 0.1, 0.8>
<650, 0.2, 0.7>
40,
<450, 0.0, 0.9>
<550, 0.1, 0.8>
<650, 0.2, 0.7>
50,
```

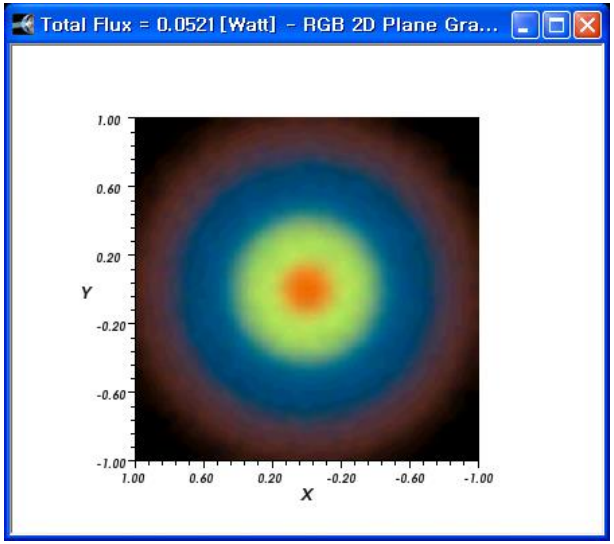
Script Expression

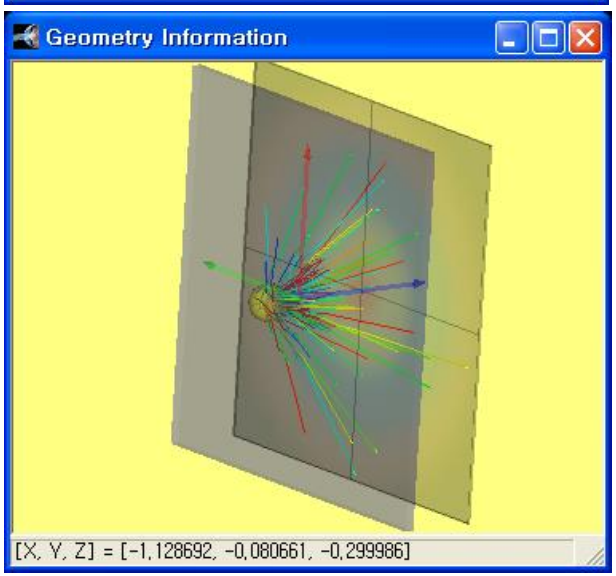
```
COATING:
        coating     {
                    <Angle_Of_Incidence_Num, Wavelength_Num>
                    Angle_Of_Incidence_1,
                    <Wavelength_1, Reflectance, Transmittance>
                    <Wavelength_2, Reflectance, Transmittance>
                    ...
                    Angle_Of_Incidence_N,
                    <Wavelength_1, Reflectance, Transmittance>
                    <Wavelength_2, Reflectance, Transmittance>
                    ...
                    <Wavelength_N, Reflectance, Transmittance>
        } |
        coating     {
                    COATING_FILE_LOCATION
        }
```

```
9 3
0
450 0.0 0.9
550 0.1 0.8
650 0.2 0.7
10
450 0.0 0.9
550 0.1 0.8
650 0.2 0.7
20
450 0.0 0.9
550 0.1 0.8
650 0.2 0.7
30
450 0.0 0.9
550 0.1 0.8
650 0.2 0.7
40
450 0.0 0.9
550 0.1 0.8
650 0.2 0.7
50
```

File Expression

Optical Property I: Coating(2)





coating_data.txt

```
9 3
0
450 0.0 0
550 0.0 0
650 0.0 1
10
450 0.0 0
550 0.0 0
650 0.0 1
20
450 0.0 0
550 0.0 1
650 0.0 0
30
450 0.0 0
550 0.0 1
650 0.0 0
40
450 0.0 1
550 0.0 0
650 0.0 0
50
450 0.0 1
550 0.0 0
650 0.0 0
60
```

```
#declare Box1 =
box
{
    <-1, -1, 0>, <1, 1, 0.05>
    color <1, 1, 1>
    ior 1.5
    surface {
        both {
            probabilistic_split on
        }
        back_surface{
          out{
              coating{ "coating_data.txt" }
          }
        }
    }
}

Box1
```

```
emittance
{
    radiometric 1
    measured_whole on
    aim_sphere <0, 60>
    angular_distribution
    {
        uniform
    }
    volume_distribution
    {
        uniform
    }
}
point_at <0, 0, 1>
```

Optical Property II: Bitmap Surface(1)

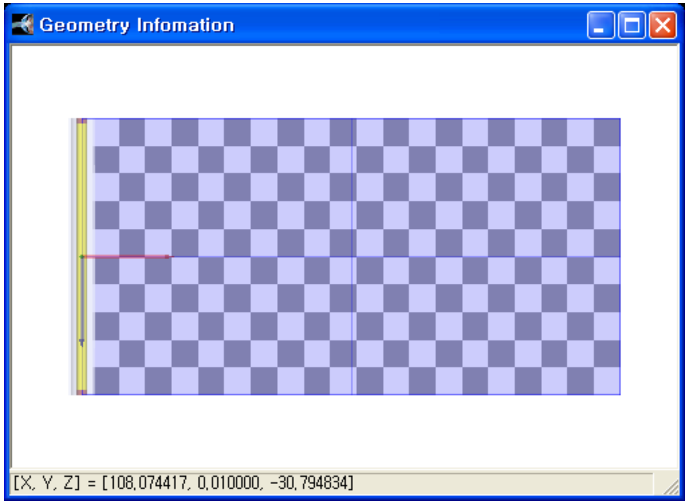
```
surface {
        both {
            reflectance {
                1
                lambertian
            }
            transmittance {
                0
            }
            zone {
                "1bit.bmp"
             }
            //probabilistic_split off
            //scattering_num N
        }
    }
```

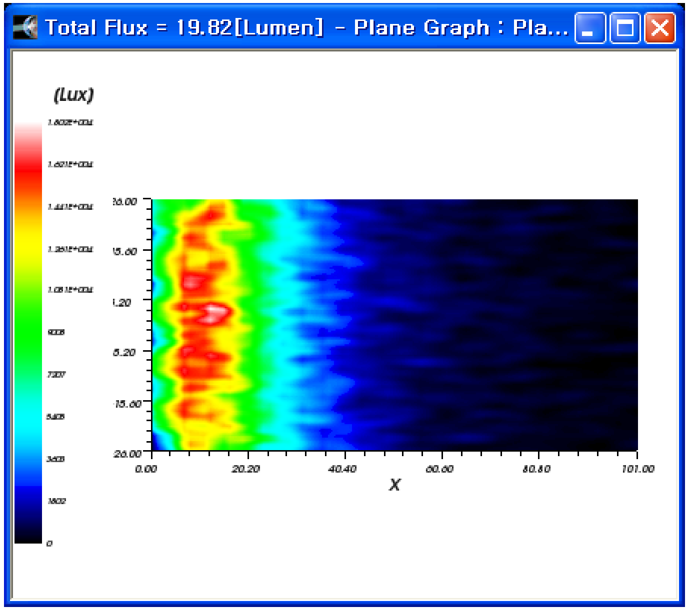
Zone 에 따라 **Transmittance** 와 **Reflectance** 조건이 설정되어 있으면 전체 면 특성 무시

```
zone {  <10,10>,        //비트맵사이즈
        1,              //비트맵 칼라 비트 수
        0,1,0,1,0, ....
        1,0,0,1, ....   //비트맵 데이터
   }
```

Optical Property II: Bitmap Surface(2-1)

```
#declare LGP =
box
{
    <2, -2.25, -26>, <101, 2.25, 26>
    color<1,1,1>
    ior {"PMMA.mat"}
    surface {
        both {
            probabilistic_split on
        }
        bottom_surface{
          both{
          reflectance {0.92 lambertian}
          transmittance {0}

          zone{
              <10,20>, //행,열
              1,
              1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,
              0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,
              1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,
              0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,
              1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,
              0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,
              1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,
              0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,
              1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,
              0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1
              invert on
              }
          }
        }
    }
}
```





Optical Property II: Bitmap Surface(2-2)

ZonePatchBitmap.rwz

```
#declare Check_Array = array[10][20]
#local ii=0;
#while (ii<10)
    #local jj=0;
    #while(jj<20)
        #declare Check_Array[ii][jj]=mod((ii+jj),2);
    #local jj=jj+1;
    #end
    #local ii=ii+1;
#end
```

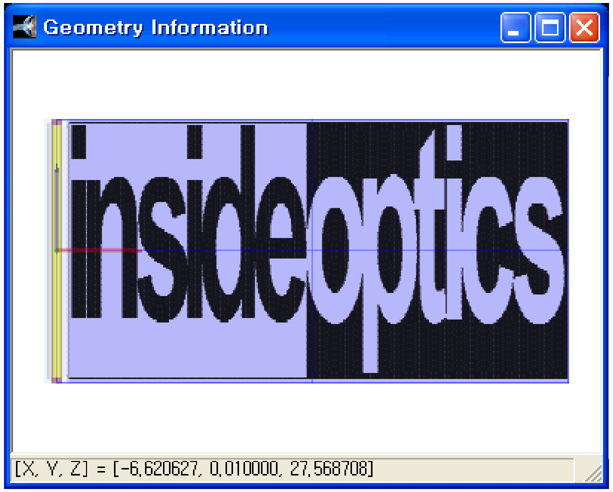
Bitmap Type으로 사용할 **Array** 변수 정의

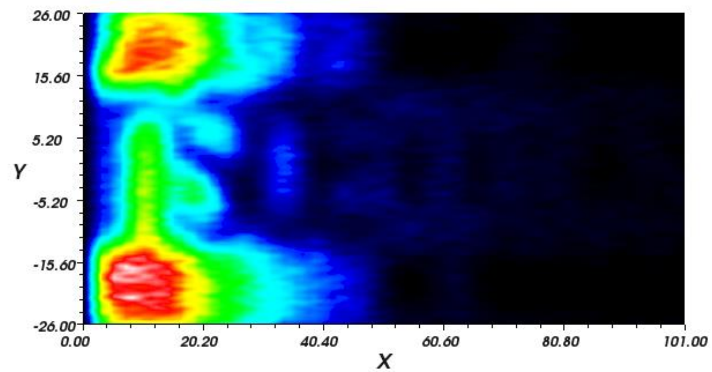
ZonePatchBitmap.rwz

```
#declare LGP =
box
{
    <2, -2.25, -26>, <101, 2.25, 26>
    color<1,0,0>
    ior {"PMMA.mat"}
    surface {
        both {
            probabilistic_split on
        }
        bottom_surface{
          both{
              reflectance {0.92 lambertian}
              transmittance {0}
              zone{
                  Check_Array
                  //transpose
                  invert on
                  }
            }
        }
    }
}
```

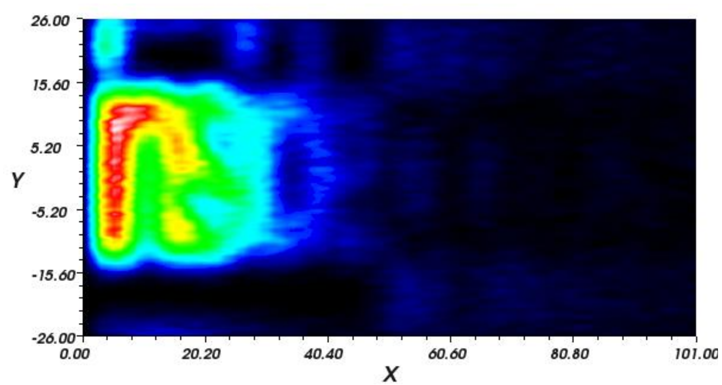
Array 변수 Call

LGP: Bitmap Surface(3)



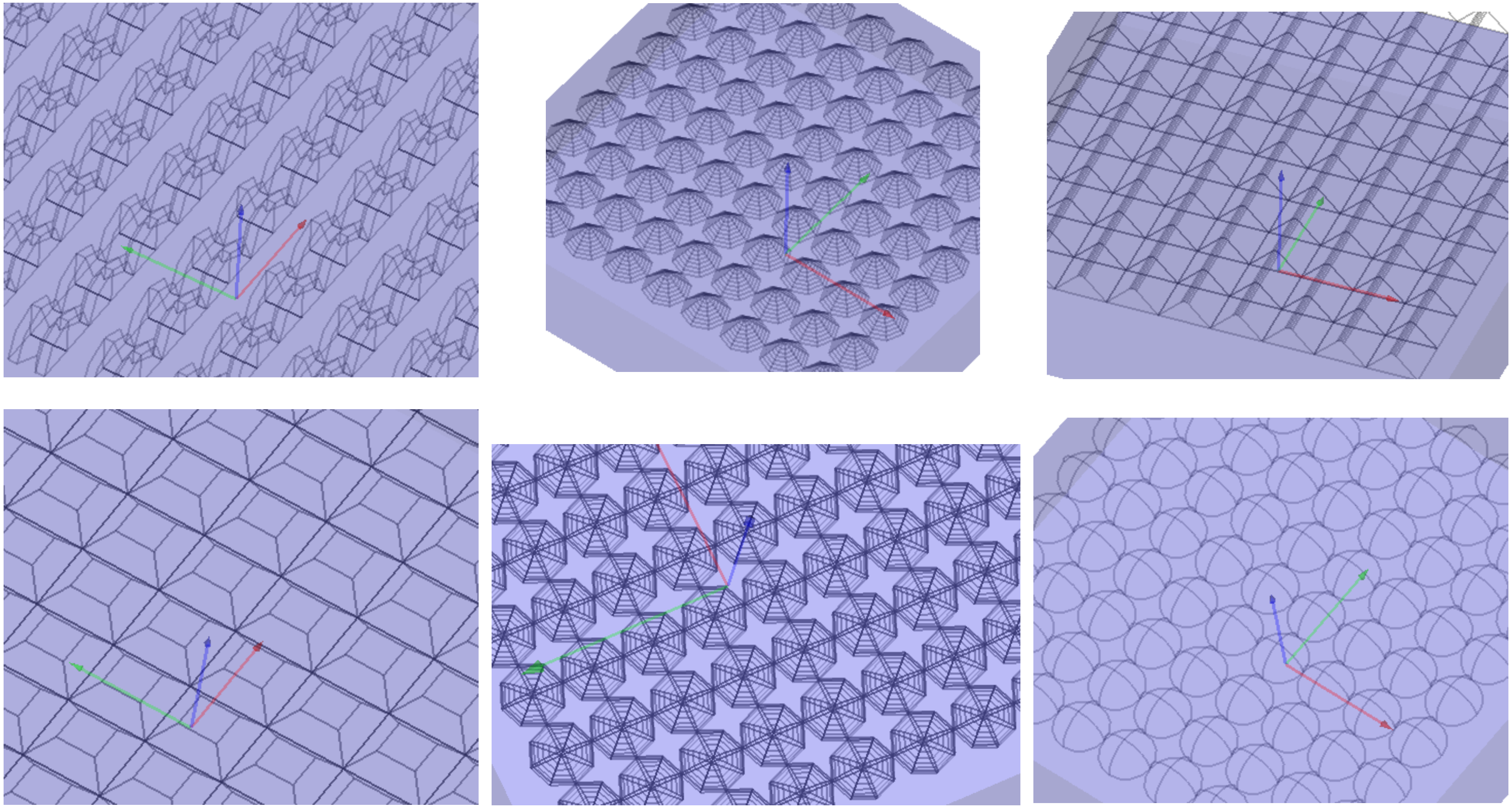



```
surface {
    both {
        probabilistic_split on
    }
    bottom_surface{
      both{
      reflectance {0.92 lambertian}
      transmittance {0}
      zone{
          "InsideOptics.bmp"
          invert on
          }
      }
    }
}
```



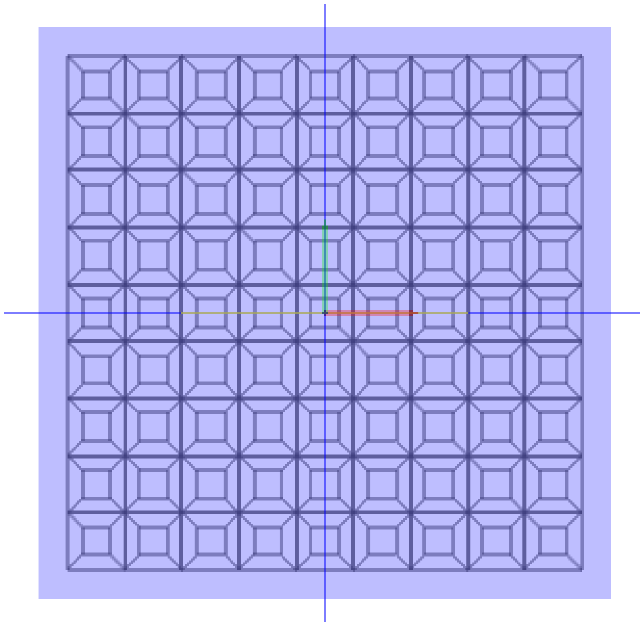

```
surface {
    both {
        probabilistic_split on
    }
    bottom_surface{
      both{
      reflectance {0.92 lambertian}
      transmittance {0}
      zone{
          "InsideOptics.bmp"
          invert off
          }
      }
    }
}
```

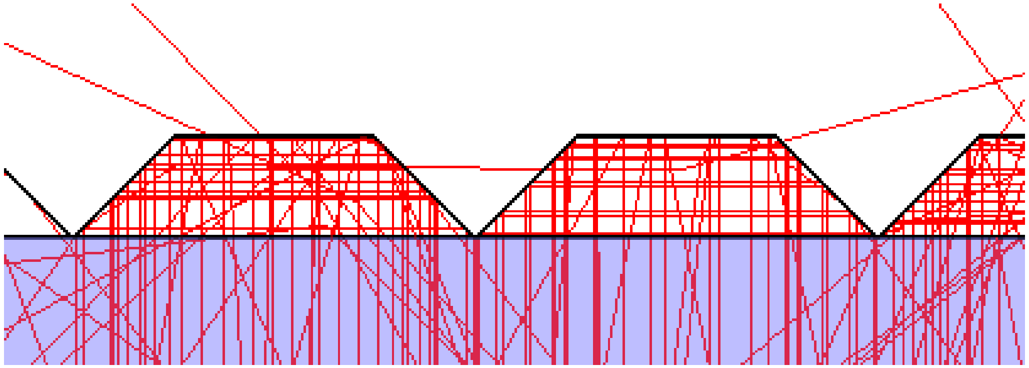
LGP: Slab Object - Geometry

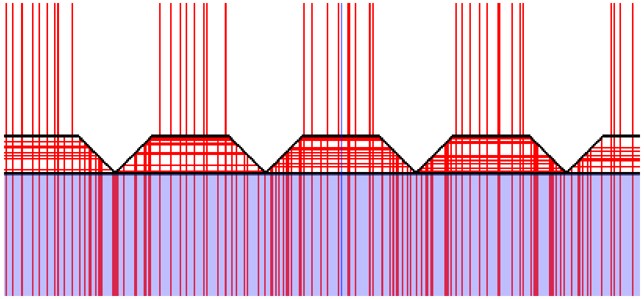


모든 **Geometry** 지원**:** 기본 **Object,**
CSG (Constructive Solid Geometry),
CAD Imported Geometry

LGP: Slab Object - Property







```
#declare TaperBox1 =
taper_box
{
    back { <-0.1, -0.1, 0>, <0.1, 0.1, 0> }
    front { <-0.05, -0.05, 0.05>, <0.05, 0.05, 0.05> }
    color<1,1,1>
    ior 1.5
    surface {
        both {
            probabilistic_split on
        }
        front_surface{
          in{
              reflectance{0.92 lambertian}
              transmittance{0}
          }
        }
    }
}
```

Different IOR & Different BSDF
← Immerse Slab

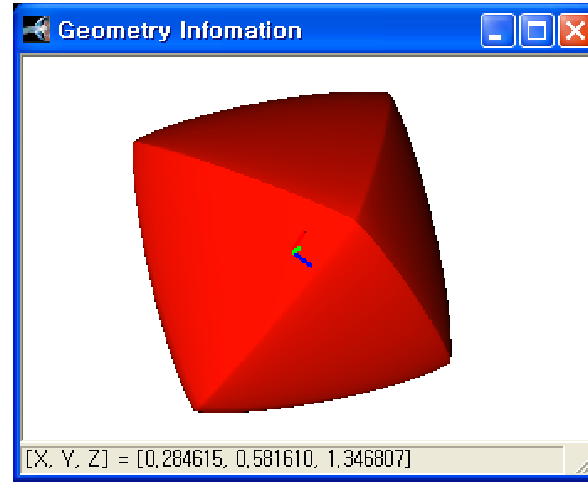
Macro Grammar I
- Basic -

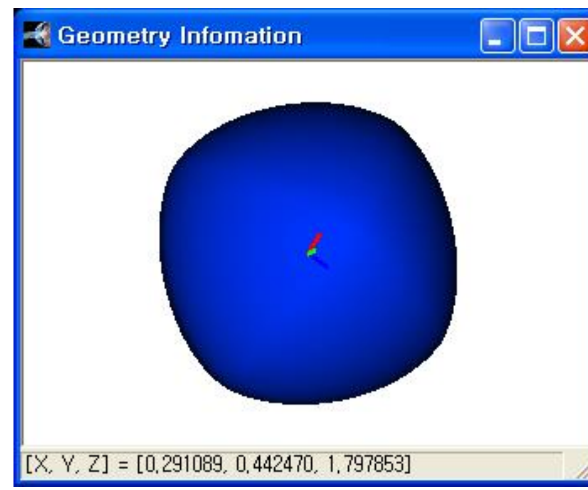
Macro Basic: #declare & #local

DECLARATION: #declare IDENTIFIER = VALUE
#local IDENTIFIER = VALUE

declare_local.rwz

```
#declare A=1.2;
#declare B=<650,1>;
#declare C=<450,1>;

#declare SuperEllipsoid0 =
superellipsoid
{
    <A, A>
    spectrum {
        discrete on
        1,
        B//<550, 1>
    }
    ior 1.5
    surface {
        both {
            probabilistic_split on
        }
    }
}

SuperEllipsoid0
```





$$f(x,y,z) = \left(|x|^{\left(\frac{2}{e}\right)} + |y|^{\left(\frac{2}{e}\right)}\right)^{\left(\frac{e}{n}\right)} + |z|^{\left(\frac{2}{n}\right)} - 1 = 0$$

Macro Basic: Math Functions & Pre-defined Math Values

Pi
True| False
degrees(FLOAT) | radians(FLOAT) |
val(STRING) |
abs(FLOAT) | int(FLOAT) |
pow(FLOAT , FLOAT) | sqrt(FLOAT) |
ceil(FLOAT) | floor(FLOAT) |
sin(FLOAT) | sinh(FLOAT) |
asin(FLOAT) | asinh(FLOAT) |
cos(FLOAT) | cosh(FLOAT) |
acos(FLOAT) | acosh(FLOAT) |
tan(FLOAT) | tanh(FLOAT) |
atan(FLOAT) | atanh(FLOAT) | atan2(FLOAT , FLOAT) |
exp(FLOAT) | log(FLOAT) | ln(FLOAT) |
max(FLOAT , FLOAT, …) | min(FLOAT , FLOAT, …) | mod(FLOAT , FLOAT) |
seed(FLOAT) | rand(FLOAT) |
strcmp(STRING , STRING) | strlen(STRING)

Expression: -2.0 -4 34 3.4e6 2e-5 .3 0.6

Macro Basic: String Functions

STRING:
STRING_FUNCTION | STRING_IDENTIFIER |
STRING_LITERAL STRING_LITERAL:
"up to 256 ASCII characters"
STRING_FUNCTION:
str(FLOAT , INT , INT) |
chr(INT) |
concat(STRING , STRING , [STRING ,...]) |

```
str(123.456, 0, 2) "123.46"
str(123.456, 0, 0) "123"
str(123.456, 5, 0) " 123"
str(123.000, 7, 2) " 123.00"

#declare A=34.567;
concat("Result is ", str(A,3,1), " lm")
→ "Result is 34.5 lm"
```

Macro Basic: #debug, #warning

#debug STRING | #error STRING | #warning STRING

debug_warnimg.rwz

```
#declare A=1;
#declare B=2;
#declare C="Macro I - Basic";
#declare CC="Output is ";

#debug C
#debug str(A+B,3,0)
#debug concat(CC, str(sqrt(A/B),3,1))

#warning C
#warning str(A+B,3,0)
#warning concat(CC, str(sqrt(A/B),3,1))
```

```
  3
Output is 0.7
Macro I - Basic
  3
Output is 0.7
```

Task List | Error Output

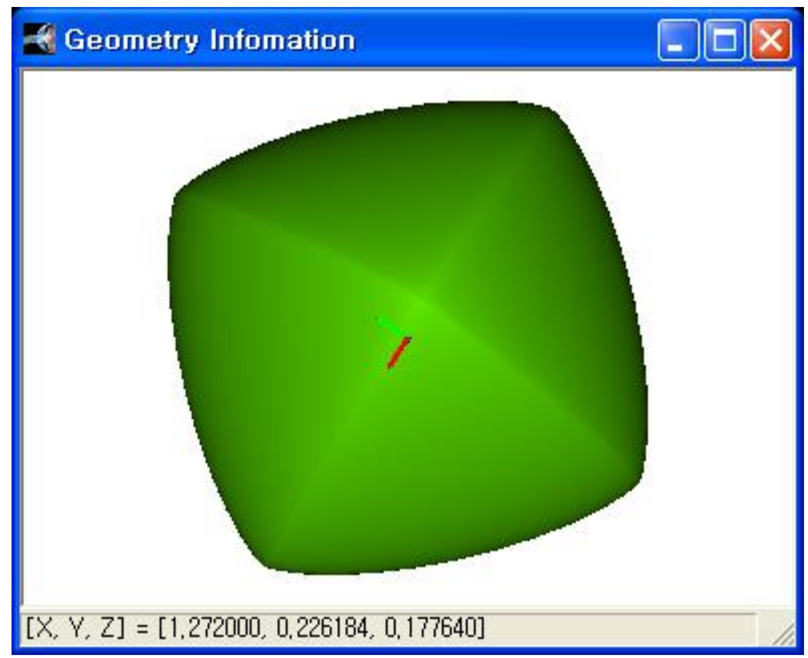
Ready

Macro Basic: #if, #else, #end

#if (Cond) TOKENS... [#else TOKENS...] #end

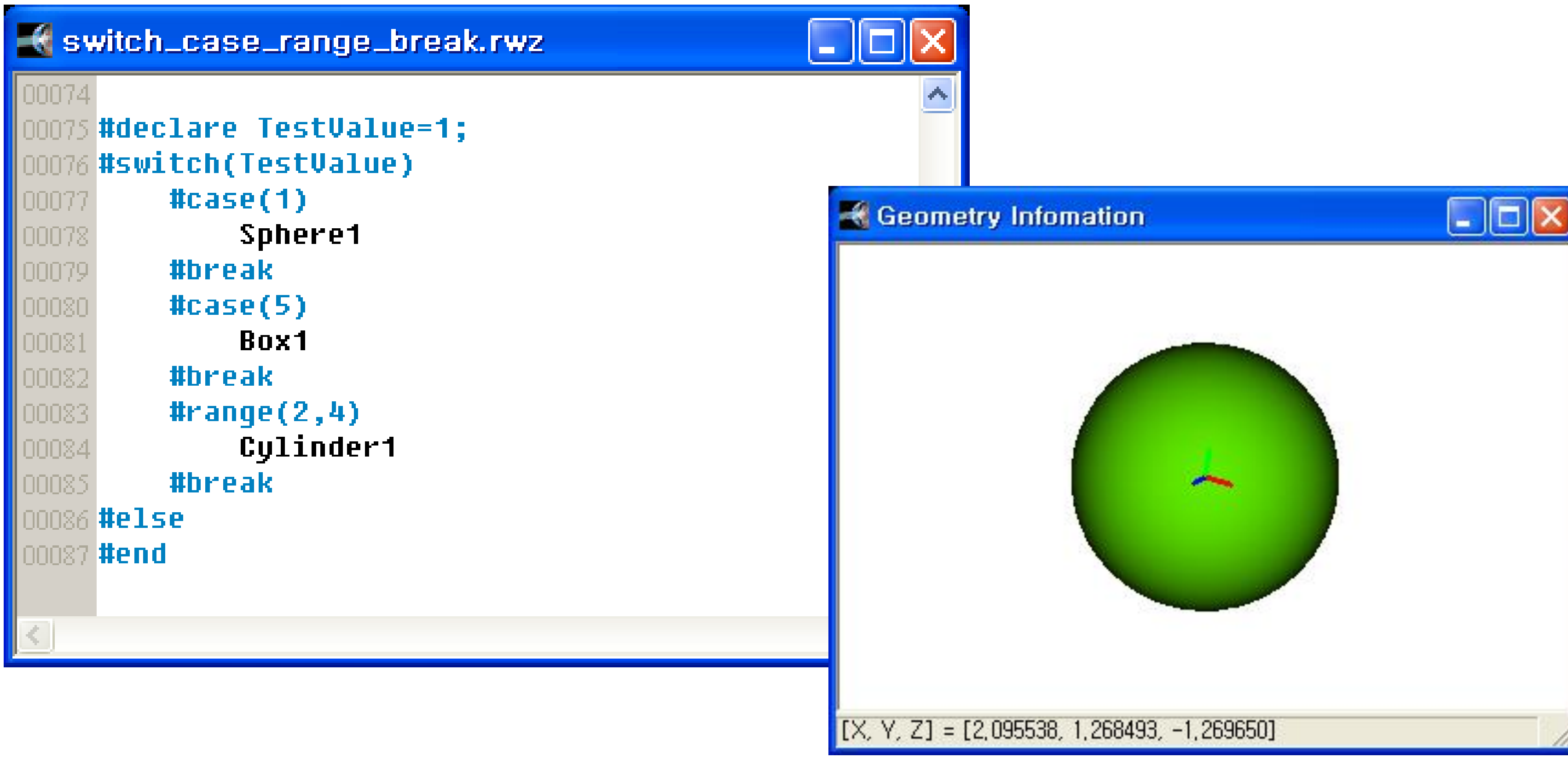
if_else_end.rwz

```
#declare factor=0; //1 or 0, true or false
#if (factor)
    box1
#else
    SuperEllipsoid1
#end
```



Macro Basic: #switch, #case, #range, #break

SWITCH_DIRECTIVE:
#switch (Switch_Value) SWITCH_CLAUSE... [#else TOKENS...] #end
SWITCH_CLAUSE:
#case(Case_Value) TOKENS... [#break] |
#range(Low_Value , High_Value) TOKENS... [#break]

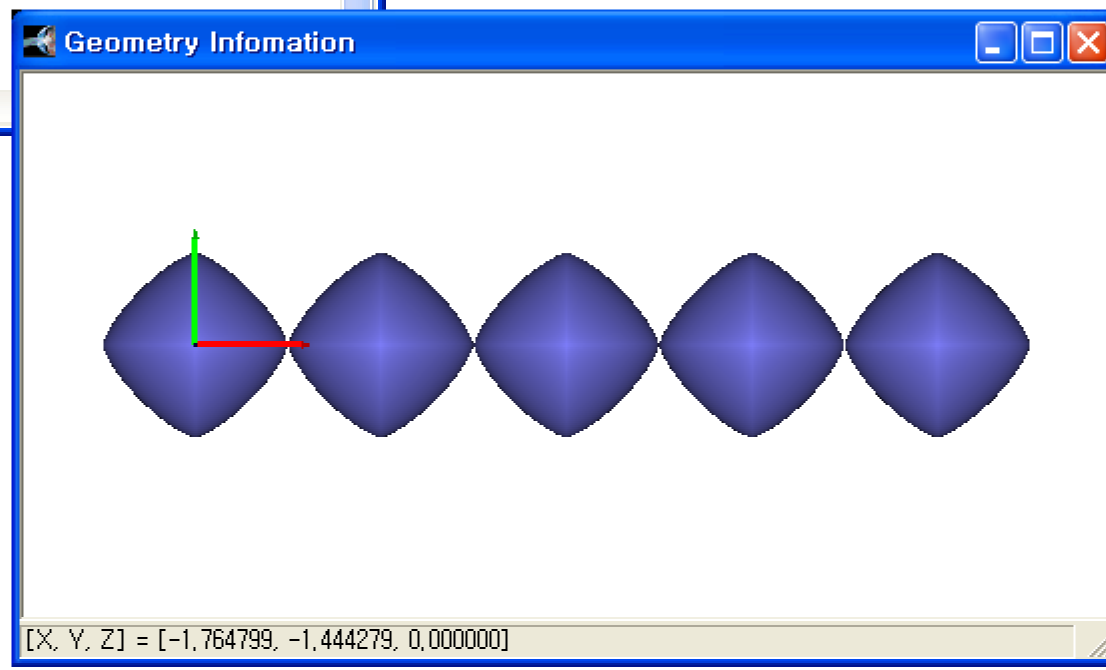


Macro Basic: #while, #end (1)

#while (Cond) TOKENS… #end

while_end.rwz

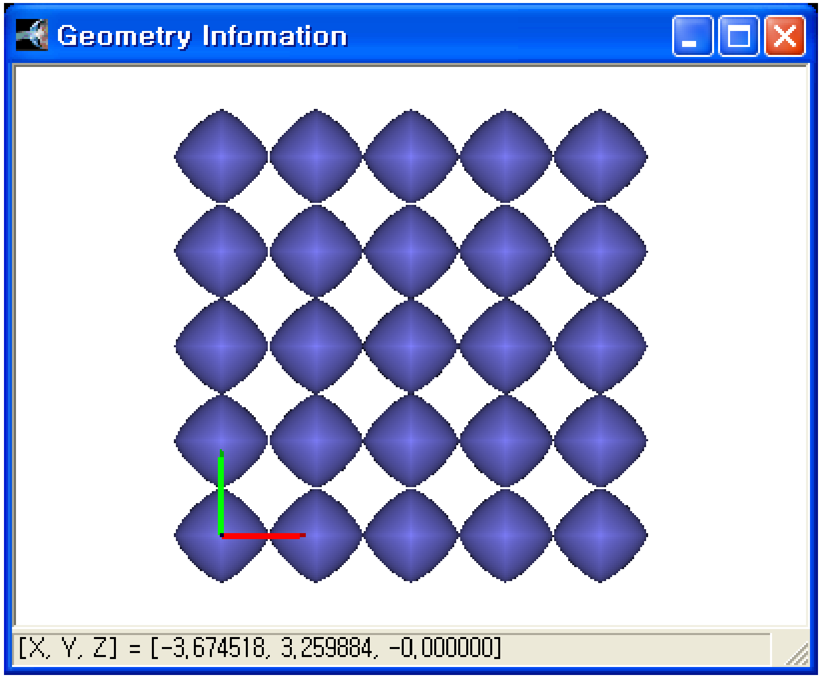
```
#declare ii=0;
#while (ii < 5)
    object{
            MyObject
            translate <2*ii,0,0>
          }
#declare ii=ii+1;
#end
```



Macro Basic: #while, #end (2)

while_end_2.rwz

```
#declare ii=0;
#while (ii < 5)
    #declare jj=0;
    #while (jj < 5)
     object{
            MyObject
            translate <2*ii,2*jj,0>
          }
    #declare jj=jj+1;
    #end
#declare ii=ii+1;
#end
```



Macro Basic: function command & Pre-defined Macro

#local FUNCTION_IDENTIFIER = function(IDENT_LIST) { FLOAT }
#declare FUNCTION_IDENTIFIER = function(IDENT_LIST) { FLOAT }

function_pre_macro.rwz

```
#declare AA = function(x) {prod(ii,1,x,pow(2,ii))}
#declare BB = function(x) {sum(i,1,x,i)}

#declare MyFunction = function(x,y,z) {sqrt(x)+pow(y,4)+log(z)}
#declare CC = MyFunction(5,5,10);

#debug str(AA(5),3,3)
#debug str(BB(10),3,3)
#debug str(CC,3,3)
```

```
Progress parsing...
32768.000
55.000
628.236
No objects in scene.
```

Task List | Error Output

Ready

$$\prod_{i=a}^{n} F$$

Prod (i, a, n, F)

$$\sum_{i=a}^{n} F$$

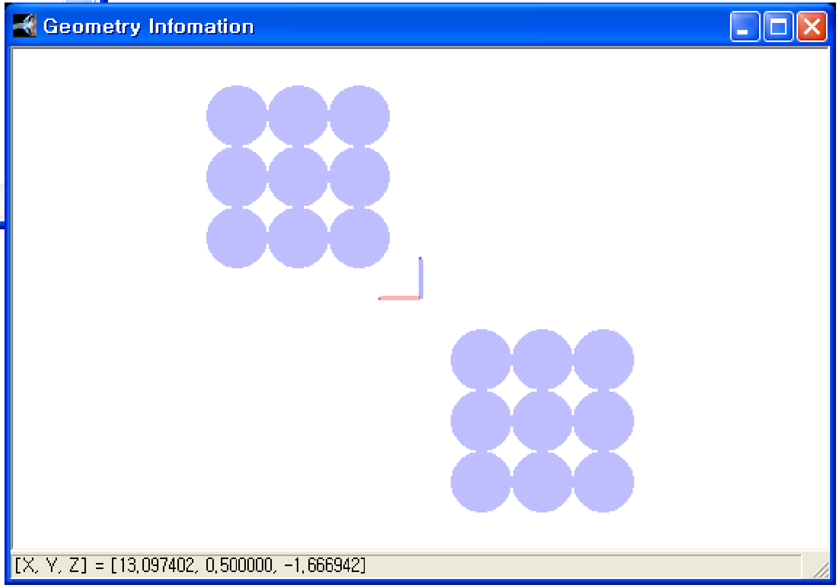
Sum (i, a, n, F)

Macro Basic: #macro, #end (1)

#macro IDENTIFIER ([PARAM_IDENT] [, PARAM_IDENT]...) TOKENS... #end

macro_end.rwz

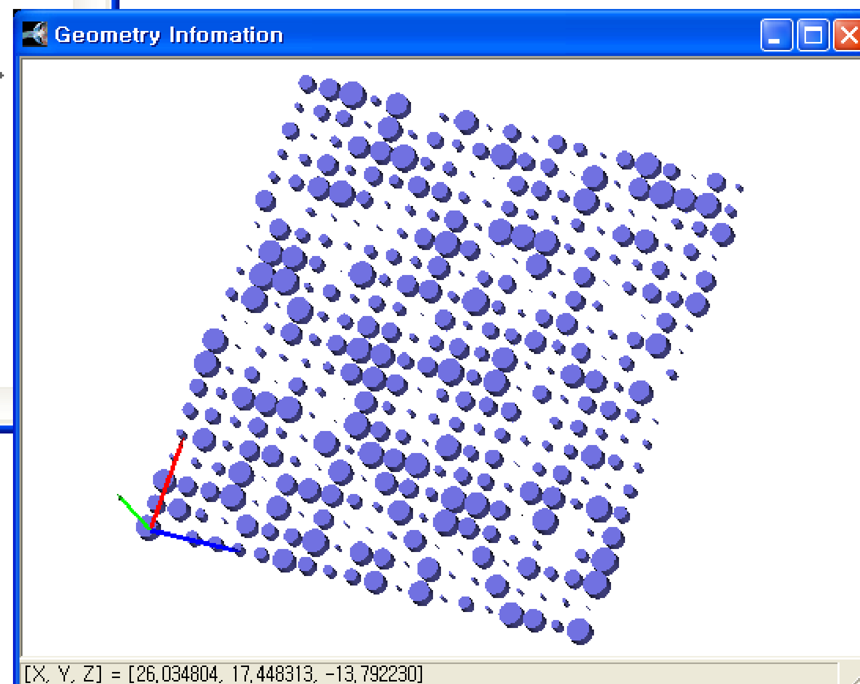
```
#macro MyMacro(PosX,PosZ,NumX,NumY)
    #local ii=0;
    #while(ii<NumX)
        #local jj=0;
        #while(jj<NumY)
            object{
                    UnitObject
                    translate <PosX+2*ii,0,PosZ+2*jj>
                }
            #local jj=jj+1;
        #end
        #local ii=ii+1;
    #end
#end

MyMacro(2,2,3,3)
MyMacro(-6,-6,3,3)
```



Macro Basic: #macro, #end (2)

macro_end_2.rwz

```
#declare R1 = seed(10000);
#macro MyMacro(PosX,PosZ,NumX,NumY)
    #local ii=0;
    #while(ii<NumX)
        #local jj=0;
        #while(jj<NumY)
            object{
                    UnitObject
                    #local RVal=rand(R1);
                    #if(0.5<=RVal<=1)
                    #else
                        #local RVal=1;
                    #end
                    scale<RVal,1,RVal>
                    translate <PosX+2*ii,0,PosZ+2*jj>
            }
            #local jj=jj+1;
        #end
        #local ii=ii+1;
    #end
#end

MyMacro(0,0,20,20)
```



Macro Basic: #declare, #local array

ARRAY_DECLARATION:

#declare IDENTIFIER = array[INT][[INT]]..[ARRAY_INITIALIZER] |

#local IDENTIFIER = array[INT][[INT]]..[ARRAY_INITIALIZER]

ARRAY_INITIALIZER:

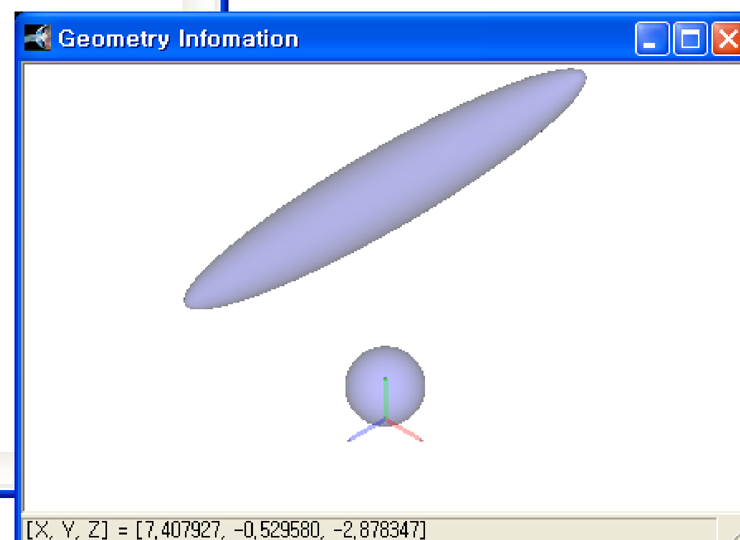
{ARRAY_ITEM, [ARRAY_ITEM,]... }

ARRAY_ITEM:

RVALUE | ARRAY_INITIALIZER

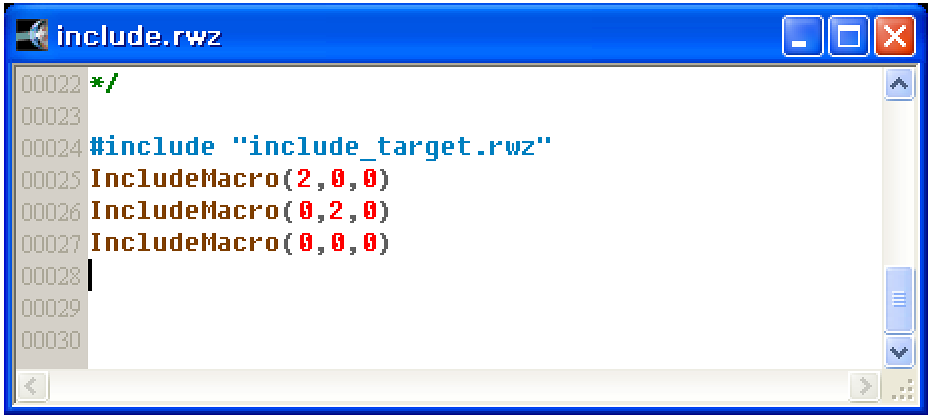
array.rwz

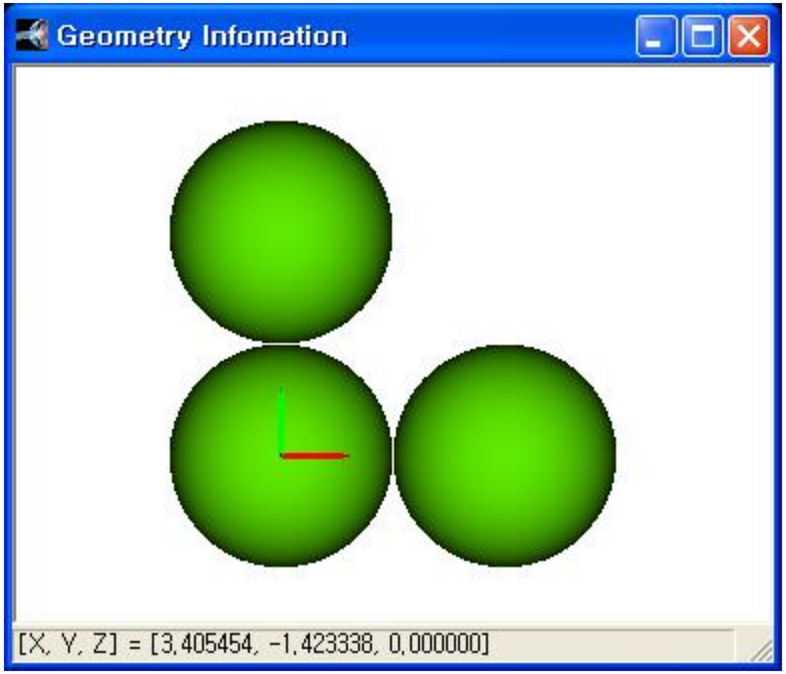
```
#declare LocationArray = array[4][10] //4행(0 ~ 3) 10열(0 ~ 9)
{
  {7,6,7,0,2,1,6,5,5,0},
  {1,2,3,4,5,6,7,8,9,0},
  {0,9,8,7,6,5,4,3,2,1},
  {1,1,2,2,3,3,4,4,5,5}
}

#macro SphereArray(A,B)
object{
        Sphere0
        translate<0,LocationArray[A][B],0>
        scale<1,1,LocationArray[A][B]>
}
#end

SphereArray(2,3)
SphereArray(3,1)
```



Macro Basic: #include

#include "filename"





Macro Basic: #fopen, #fclose

```
FOPEN_DIRECTIVE:
        #fopen IDENTIFIER "filename" OPEN_TYPE
OPEN_TYPE:
        read | write | append
FCLOSE_DIRECTIVE:
        #fclose FILE_HANDLE_IDENTIFIER
READ_DIRECTIVE:
        #read (FILE_HANDLE_IDENTIFIER, DATA_IDENTIFIER[,DATA_IDENTIFIER]..)
DATA_IDENTIFIER:
        UNDECLARED_IDENTIFIER | FLOAT_IDENTIFIER | VECTOR_IDENTIFIER |
        STRING_IDENTIFIER
WRITE_DIRECTIVE:
        #write( FILE_HANDLE_IDENTIFIER, DATA_ITEM[,DATA_ITEM]...)
DATA_ITEM:
        FLOAT | VECTOR | STRING
```

Macro Basic: #fopen, #fclose - #write

fopen_fclose_write.rwz

```
#declare DataArray=array[4][3]
{
  {1,2,3},
  {4,5,6},
  {7,8,9},
  {10,11,12}
}
//File을 생성하고 배열을 저장
#fopen DataFile "data_write.txt" write
#declare NumR=4;
#declare NumC=3;
#write (DataFile, NumR, ",", NumC, ",")
#write (DataFile, "₩n")
#declare I = 0;
#while ( I < NumR )
    #declare J = 0;
    #while (J < NumC)
        #if(I+J < NumR+NumC-2)
            #write (DataFile, DataArray[I][J], ",")
        #else
            #write (DataFile, DataArray[I][J])
        #end
    #declare J = J+1;
    #end
    #write (DataFile, "₩n")
#declare I = I+1;
#end
#fclose DataFile
```

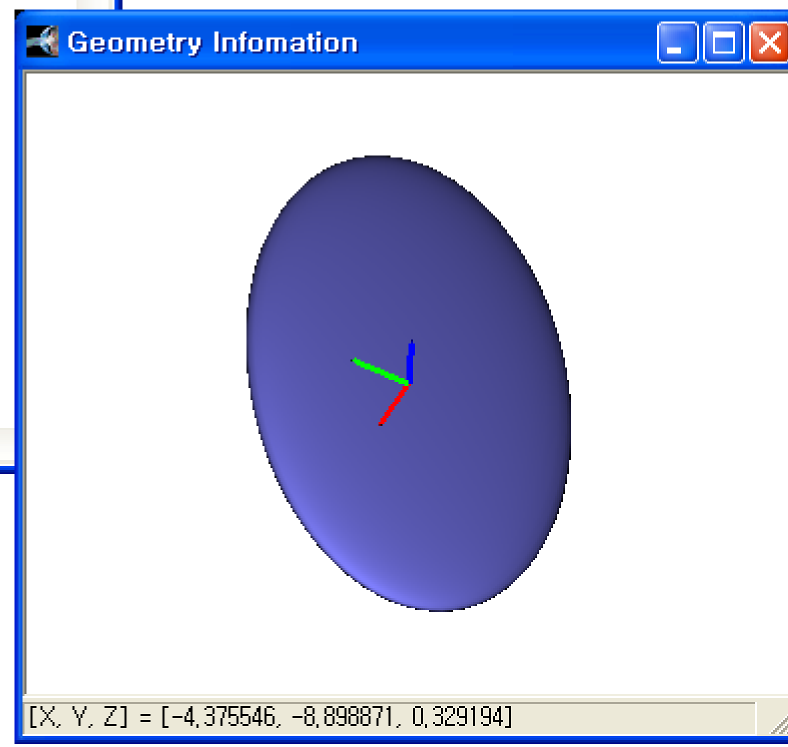
data_write.txt

```
4,3,
1,2,3,
4,5,6,
7,8,9,
10,11,12
```

Macro Basic: #fopen, #fclose - #read

fopen_fclose_read.rwz

```
//File을 읽어 배열에 할당
#fopen DataFile "data_write.txt" read
#read(DataFile, NumR1, NumC1)
#declare NumR=NumR1;
#declare NumC=NumC1;
#declare DataArray = array[NumR][NumC]
#declare I = 0;
#while ( I < NumR )
    #declare J = 0;
    #while (J < NumC)
        #read (DataFile, MyData)
        #declare DataArray[I][J] = MyData;
    #declare J = J+1;
    #end
#declare I = I+1;
#end
#fclose DataFile

object{
    SphereUnit
    scale<DataArray[0][0],DataArray[1][1],DataArray[2][2]>
}
```



Macro Grammar II
- RBT: [Sequence, Loop] –

RBT: Syntax (1)

```
RBT :
        [RBT_TYPE] ,
        [RBT_STATEMENT]
```

```
RBT_TYPE:
        [ SEQUENCE | LOOP | OPTIMIZE ]
```

```
RBT_STATEMENT :

SEQUENCE :
        Number of files,
        { "FileName" , [OUTPUT TYPE]}
        { "FileName" , [OUTPUT TYPE]}
        ...

LOOP :
        Number of Loop Variable1, Number of Loop Variable2,…, Number of Loop VariableN,
        FILE    { "FileName" , [OUTPUT TYPE] }
        SCRIPT  string
```

RBT: Syntax (2)

```
OPTIMIZE :
        LIMIT                   number
        SCRIPT_FILE             "FileName"
        OBJECTIVE_FUNCTION      { [OBJECTIVE FUNCTION]}
```

```
        TARGET_VALUE            { [INTENSITY_VALUE number | CIE_VALUE <val, val>] }
        DELTA                   number
        VARIABLES
        {
                Number of variables,
                <Lower bound, Upper bound>,
                <Lower bound, Upper bound>,
                ...
        }
```

RBT: Syntax (3)

```
OUTPUT TYPE :
        RAW_DATA | DETECTOR_DATA | RASTER_DATA | SPECTRUM_DATA | IMAGE_DATA [null | ALL]
OBJECTIVE FUNCTION :
        DETECTOR_NUM
        {
         "Detector Name 1 ",
        [VALUE TYPE] [CONDITION] [COMPARE] [VALUE]
        }
        ..
        {
         "Detector Name N ",
        [VALUE TYPE] [CONDITION] [COMPARE] [VALUE]
        }
```

RBT: Syntax (4)

VALUE TYPE :

[Illuminance | Spatial_Luminance | Angular_Luminance | Illuminance_CIExy | Luminance_CIExy | Intensity | Intensity_CIExy | Total_Flux]

CONDITION :

[**TOTAL** | **FUNCTION**]

TOTAL : Total

FUNCTION :

Function_Express " [**OPERATOR**] [**Bin Index**] + [*Number*] "

COMPARE :

= | < | > | <= | >=

VALUE : Number

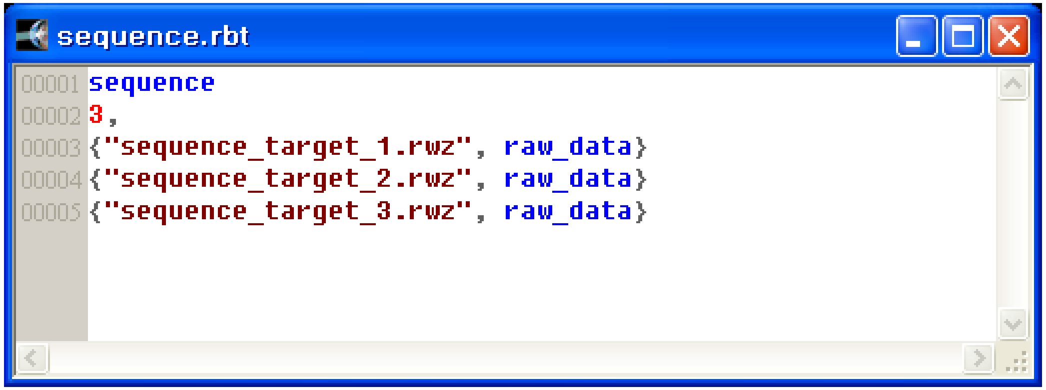
OPERATOR :

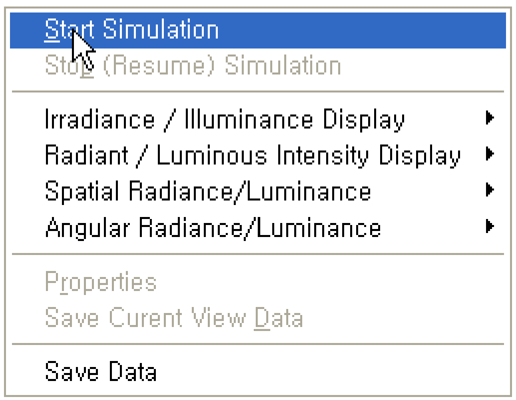
[(|) | + | - | * | / | sqrt | sqr | exp | pow | sin | cos | tan]

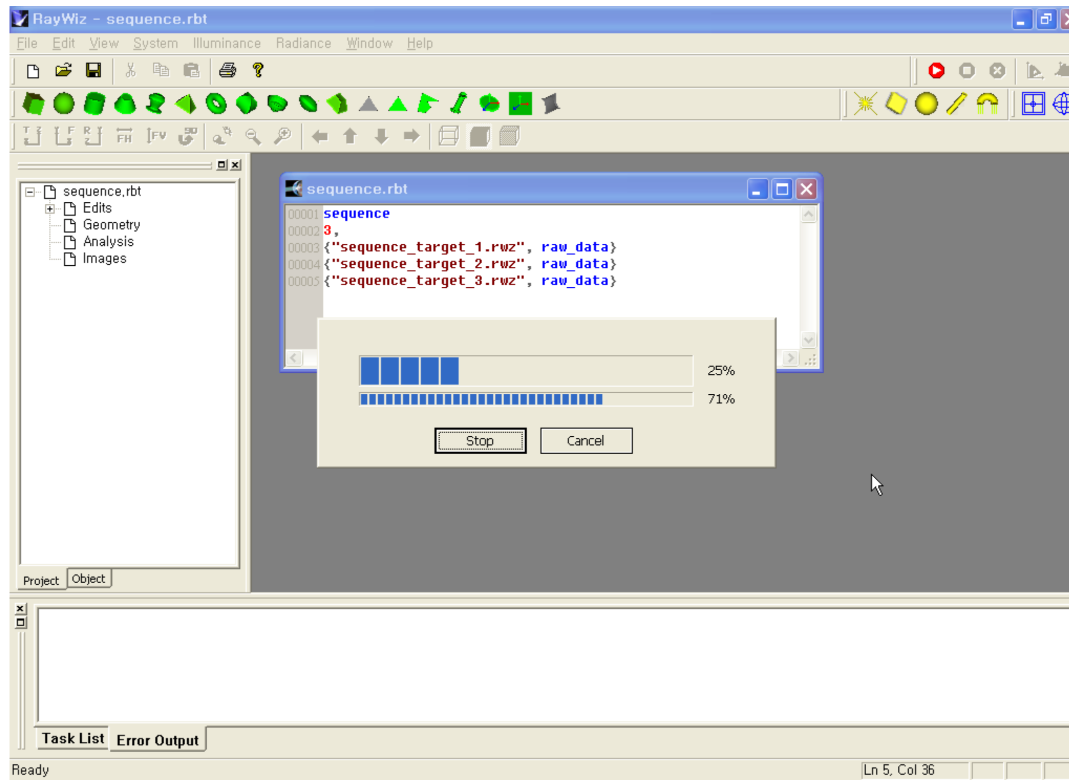
Bin Index : bin[i, j] | bin[i1 ~ i2, j1 ~ j2] | maxbin[i1 ~ i2, j1 ~ j2] | minbin[i1 ~ i2, j1 ~ j2] | meanbin[i1 ~ i2, j1 ~ j2] | totalMax | totalMin | totalMean

i,j : 0 .. N

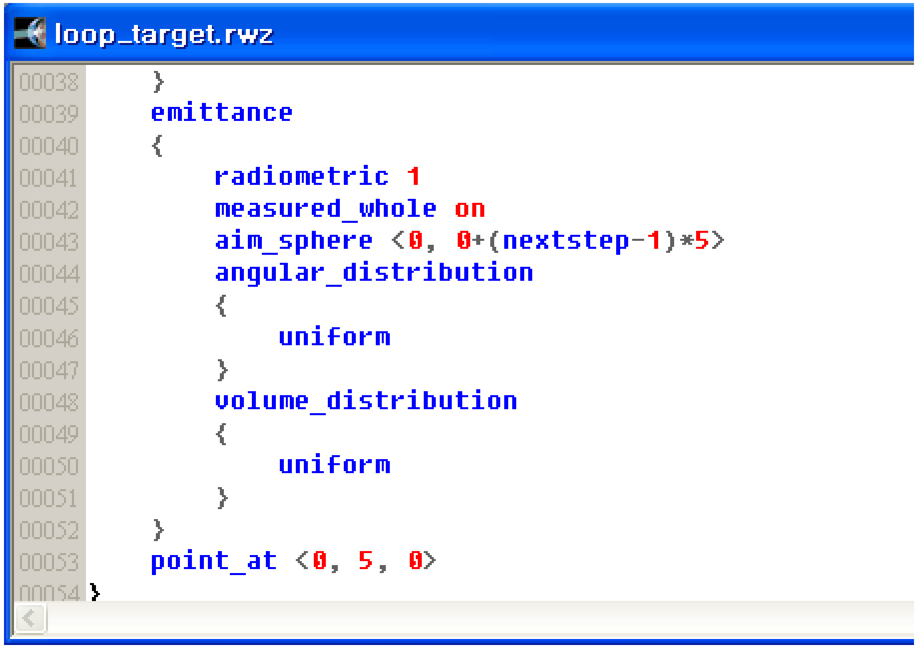
RBT: Sequence

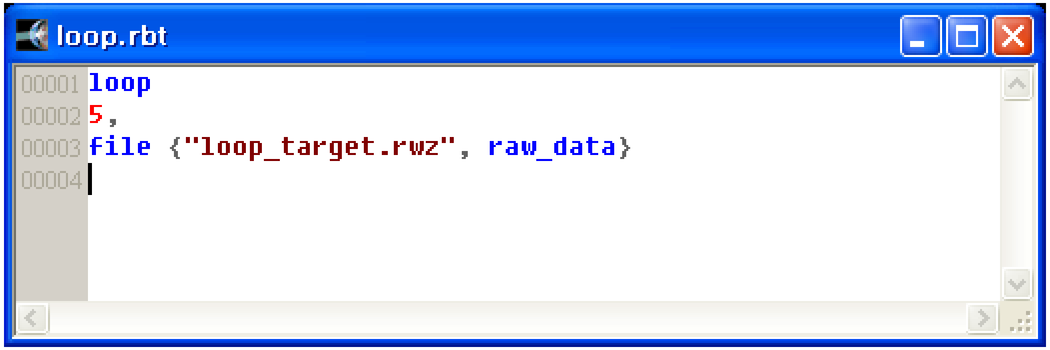


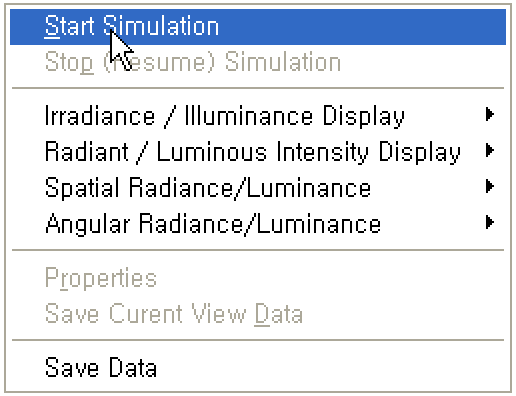




RBT: Loop (1)


loop_target.rwz
}
emittance
{
radiometric 1
measured_whole on
aim_sphere <0, 0+(nextstep-1)*5>
angular_distribution
{
uniform
}
volume_distribution
{
uniform
}
}
point_at <0, 5, 0>
}


loop.rbt
loop
5,
file {"loop_target.rwz", raw_data}


Start Simulation
Stop (Resume) Simulation
Irradiance / Illuminance Display
Radiant / Luminous Intensity Display
Spatial Radiance/Luminance
Angular Radiance/Luminance
Properties
Save Curent View Data
Save Data

RBT: Loop (2)

loop_multi_target.rwz

```
#declare SuperEllipsoid1 =
superellipsoid
{
    <1.0+$x0*0.1, 1.0+$x1*0.1>
    color<1,1,1>
    ior 1.5
    surface {
        both {
            probabilistic_split on
        }
    }
    translate<0,2,0>
}
```

loop_multi.rbt

```
loop
3,3,
file {"loop_multi_target.rwz", raw_data}
```

Microsoft Excel - loop_multi.csv

파일(F) 편집(E) 보기(V) 삽입(I) 서식(O) 도구(T)

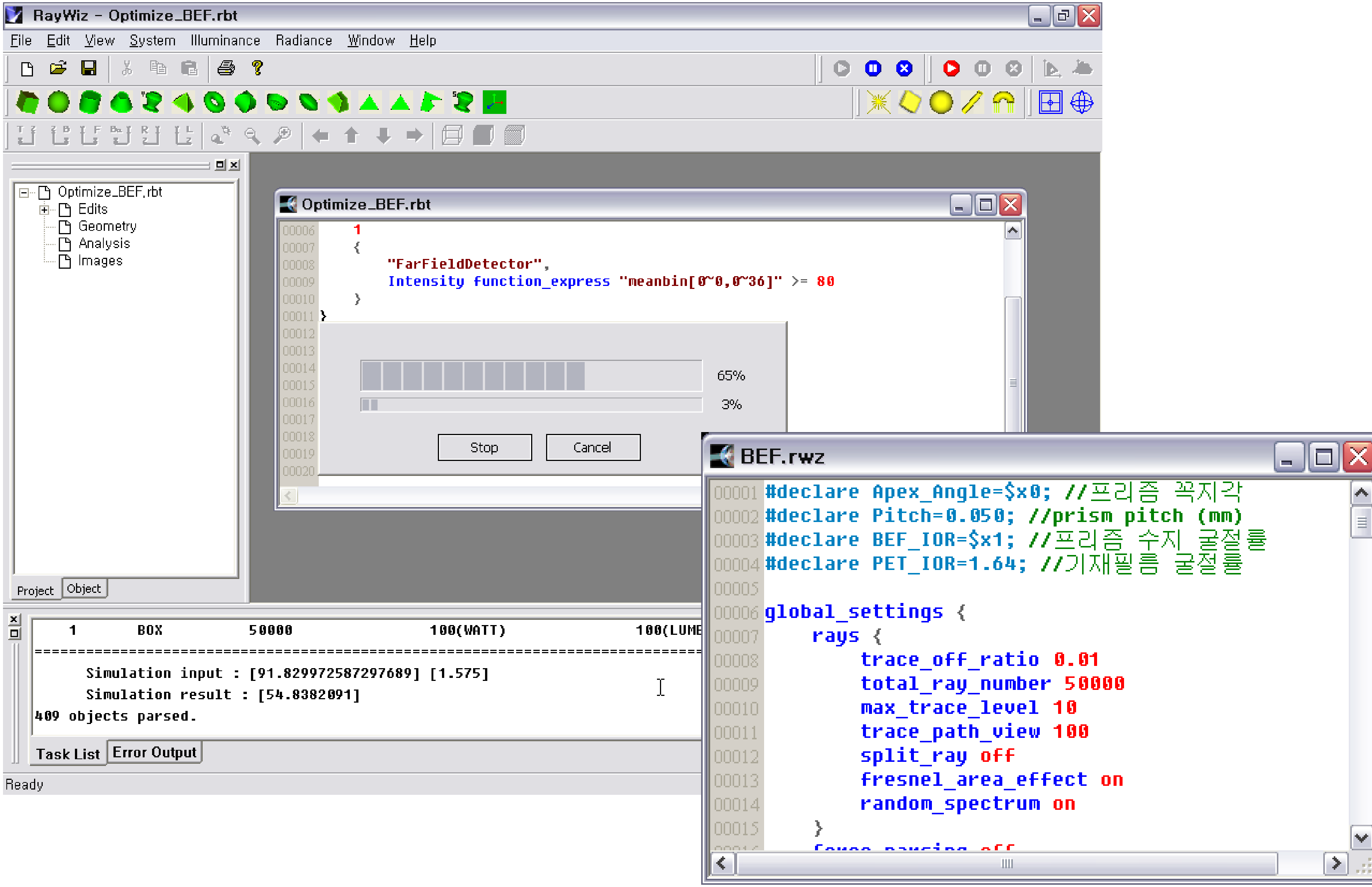
A15

	A	B	C	D
1	Index	$x0	$x1	
2	1	1	1	
3	2	2	1	
4	3	3	1	
5	4	1	2	
6	5	2	2	
7	6	3	2	
8	7	1	3	
9	8	2	3	
10	9	3	3	
11				

Macro Grammar III
- RBT: [Optimization] –

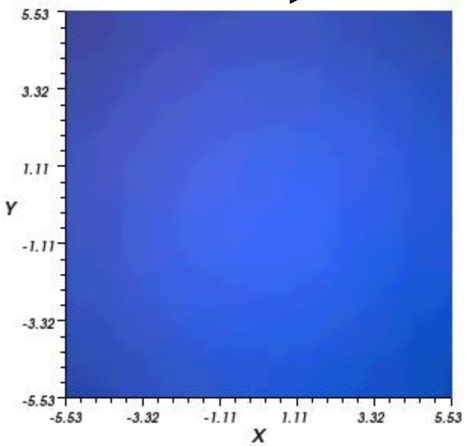
Optimization in RAYWIZ

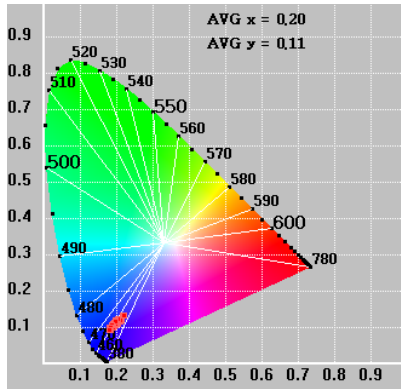
Optimization: Center Intensity Maximization

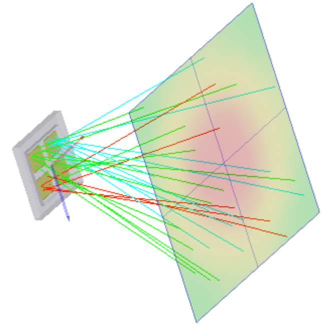


Optimization: Color Balance

Initial







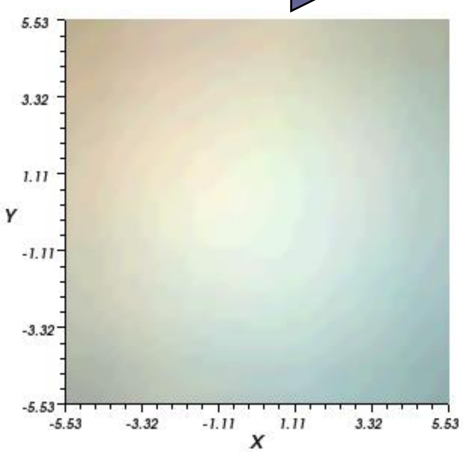
RGGB4in1.rwz:1 *

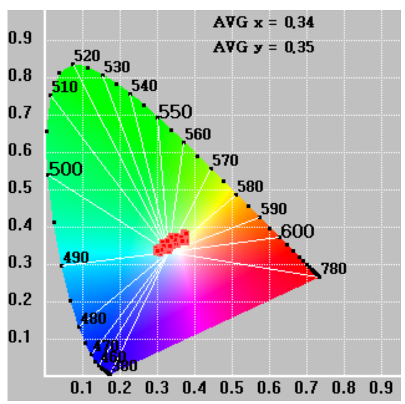
```
#declare CupBaseIn_Y = 2*Pixel_Y;
#declare CupBaseOut_X = CupBaseIn_X+1;
#declare CupBaseOut_Y = CupBaseIn_Y+1;
#declare CupDepth = 0.2;
#declare Power = 100;
#declare Ratio_R = $x0+1;//ex)lumen 28:25:5
#declare Ratio_G = $x1+1;
#declare Ratio_B = $x2+1;
#declare Power_R = Power*Ratio_R;
#declare Power_G = Power*Ratio_G;
#declare Power_B = Power*Ratio_B;
#declare LED_Space_X = 10; //10mm
#declare LED_Space_Y = 10; //65mm
#declare Display_X = 2*25.4*4/18.3576; //NumLED_X*LED_Space_X;
#declare Display_Y = 2*25.4*4/18.3576; //NumLED_Y*LED_Space_Y;
```

Optimize_RGGB4in1.rbt

```
optimize
limit 100
script_file "RGGB4in1.rwz"
objective_function
{
    "PlaneDetector",
    Illuminance_CIExy function_express
    "(Bin[4,4]+Bin[5,5]+Bin[4,5]+Bin[5,4])/4"
}
target_value { CIE_Value <0.31,0.31> }
delta 0.01
variable
{
    3,
    <1,3
    <1,3
    <1,3
}
```

Final



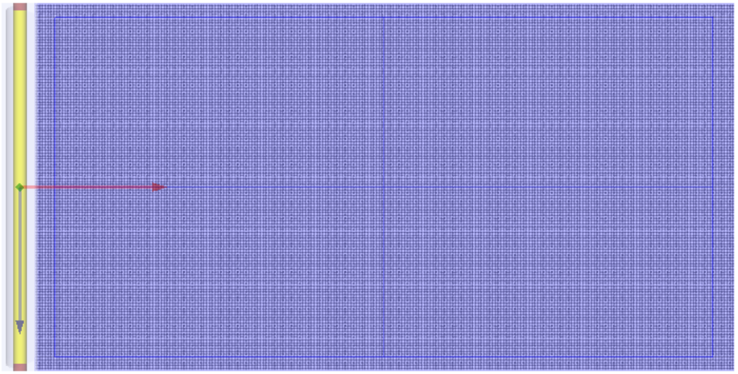


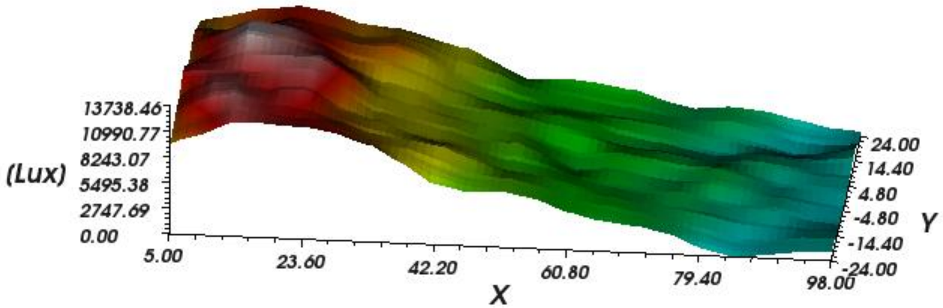
Microsoft Excel - Optimize_RGGB4in1_opt20070324123018.csv

	A	B	C	D	E	F
1	Calculation	Variable0	Variable1	Variable2	Objective Value1	Objective Value2
2	Temporal	17	17	17	0,198596253	0,109218354
3	Temporal	17,17	17	17	0,199062179	0,109393266
4	Temporal	17	17,17	17	0,198561995	0,10967629
5	Temporal	17	17	17,17	0,198173283	0,108593443
6	Temporal	17	17	17	0,198596253	0,109218354
7	Temporal	17,17	17	17	0,199062179	0,109393266
8	Temporal	17	17,17	17	0,198561995	0,10967629
9	Temporal	17	17	17,17	0,198173283	0,108593443
10	Temporal	25,54718441	26,79011467	1	0,377244996	0,384158987
11	Step	25,54718441	26,79011467	1	0,198596253	0,109218354
12	Temporal	25,54718441	26,79011467	1	0,377244996	0,384158987
13	Temporal	25,80265625	26,79011467	1	0,378562708	0,383809361
14	Temporal	25,54718441	27,05801582	1	0,376524715	0,385432241
15	Temporal	25,54718441	26,79011467	1,01	0,376931381	0,383680425
16	Temporal	24,12463981	28,02872676	2,73590881	0,323967017	0,322682265
17	Step	24,12463981	28,02872676	2,73590881	0,377244996	0,384158987
18	Final	24,12463981	28,02872676	2,73590881	0,377244996	0,384158987

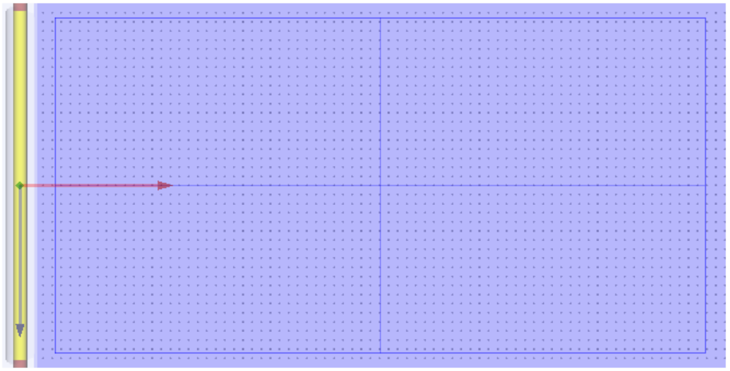
Optimization: Uniformity

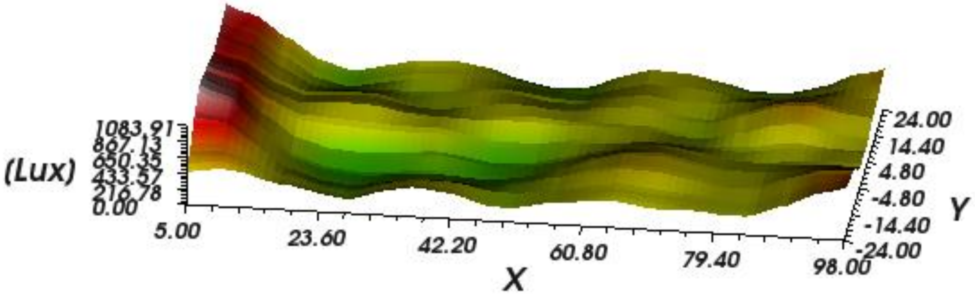
Initial



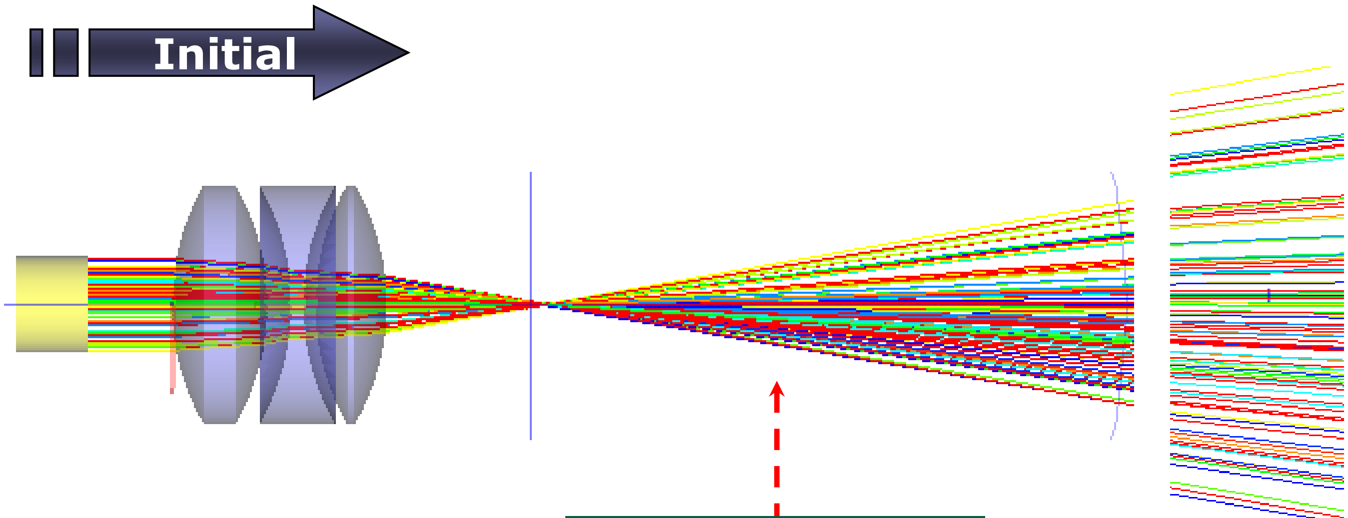


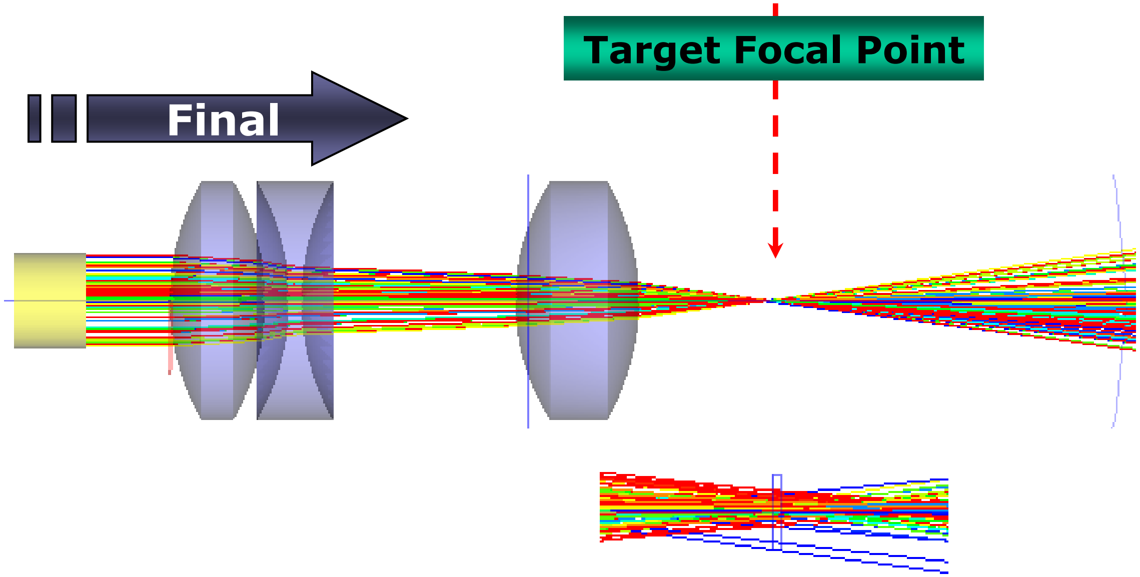
Final

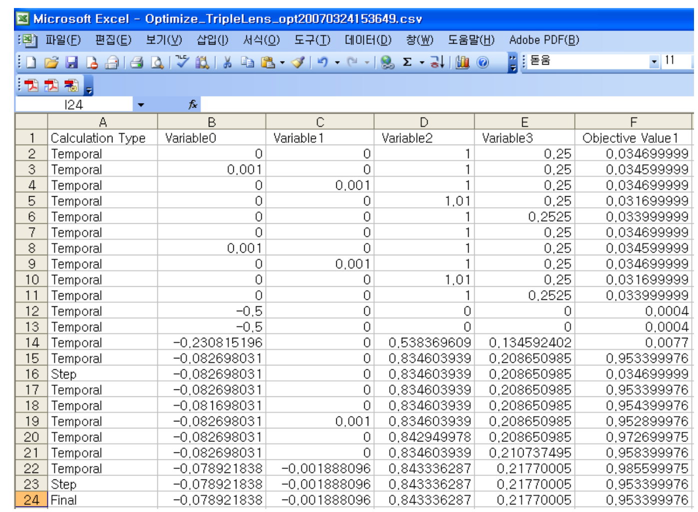




Optimization: Lens System

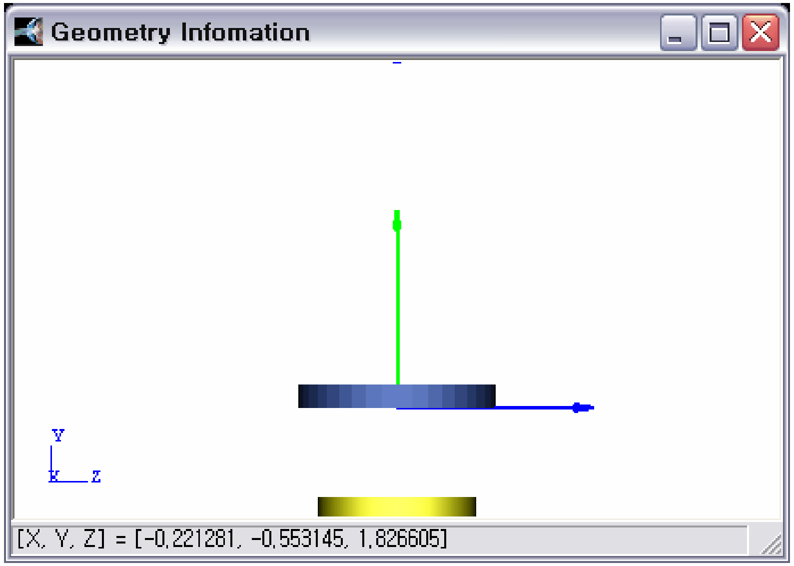


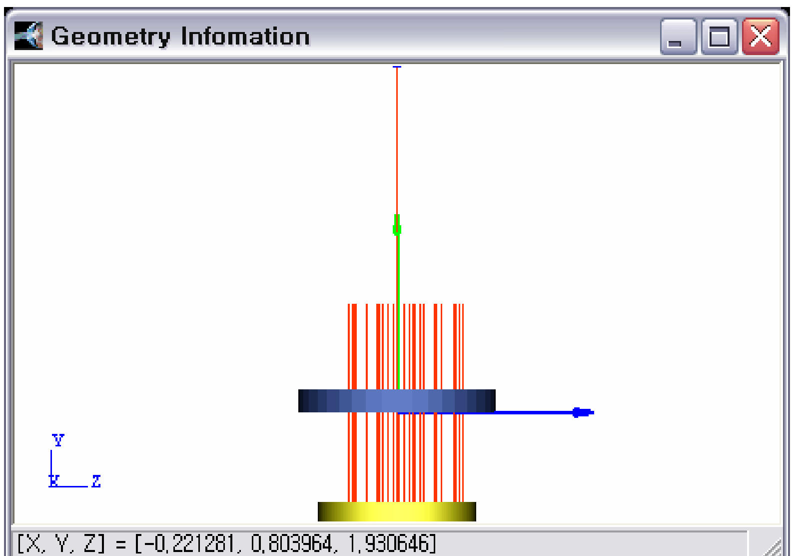




	A	B	C	D	E	F
1	Calculation Type	Variable0	Variable1	Variable2	Variable3	Objective Value1
2	Temporal	0	0	1	0,25	0,034699999
3	Temporal	0,001	0	1	0,25	0,034599999
4	Temporal	0	0,001	1	0,25	0,034699999
5	Temporal	0	0	1,01	0,25	0,031699999
6	Temporal	0	0	1	0,2525	0,033999999
7	Temporal	0	0	1	0,25	0,034699999
8	Temporal	0,001	0	1	0,25	0,034599999
9	Temporal	0	0,001	1	0,25	0,034699999
10	Temporal	0	0	1,01	0,25	0,031699999
11	Temporal	0	0	1	0,2525	0,033999999
12	Temporal	-0,5	0	0	0	0,0004
13	Temporal	-0,5	0	0	0	0,0004
14	Temporal	-0,230815196	0	0,538369609	0,134592402	0,0077
15	Temporal	-0,082698031	0	0,834603939	0,208650985	0,953399976
16	Step	-0,082698031	0	0,834603939	0,208650985	0,034699999
17	Temporal	-0,082698031	0	0,834603939	0,208650985	0,953399976
18	Temporal	-0,081698031	0	0,834603939	0,208650985	0,954399976
19	Temporal	-0,082698031	0,001	0,834603939	0,208650985	0,952899976
20	Temporal	-0,082698031	0	0,842949978	0,208650985	0,972699975
21	Temporal	-0,082698031	0	0,834603939	0,210737495	0,958399976
22	Temporal	-0,078921838	-0,001888096	0,843336287	0,21770005	0,985599975
23	Step	-0,078921838	-0,001888096	0,843336287	0,21770005	0,953399976
24	Final	-0,078921838	-0,001888096	0,843336287	0,21770005	0,953399976

RBT Optimization: Focusing (1)





SingleLens_Target.rwz

```
#declare LightSrc =
light_source
{
    <0, -0.5, 0>
    spectrum {
        discrete on
        1,
        <650, 1>
    }
    object_light
    {
        surface
        cylinder
        {
            <0, 0, 0>, <0, 0.1, 0>, 0.40
        }
```

SingleLens_Target.rwz

```
        emittance
        {
            front_surface
            {
                off
            }
            rear_surface
            {
                on
                angular_distribution
                {
                    uniform
                }
                spatial_distribution
                {
                    uniform
                }
            }
            cylinder_surface
            {
                off
            }
        }
    }
```

RBT Optimization: Focusing (2)

SingleLens_Target.rwz

```
    emittance
    {
        photometric 100
        measured_whole on
        aim_sphere <0, 0>
    }
    point_at <0, 2, 0>
}

LightSrc
```

SingleLens_Target.rwz

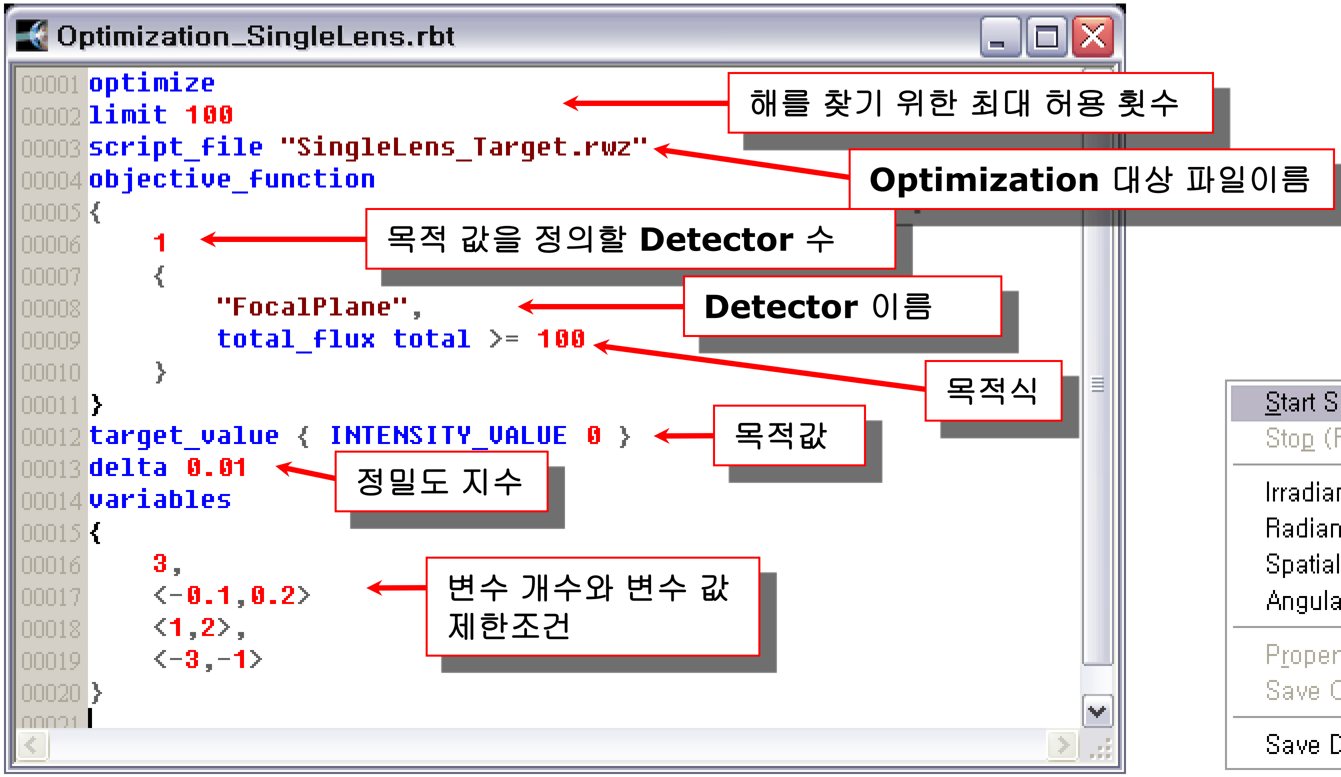
```
#declare FocalPlane =
box
{
    <-0.025, 1.75, -0.025>, <0.025, 1.751, 0.025>
    detector {
        bottom_surface
        photometric
        bin_size <21, 21>
    }
}

FocalPlane
```

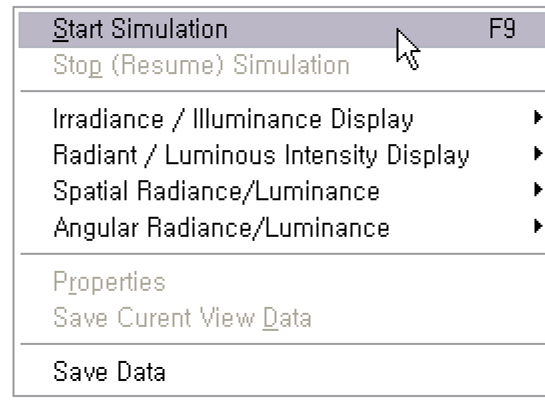
SingleLens_Target.rwz

```
#declare SingleLens =
optical
{
    circular center_thickness 0.12+$x0
    { $x1, 0.5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 }
    { $x2, 0.5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 }
    color<0,0,1>
    ior 1.5
    surface {
        both {
            probabilistic_split on
        }
    }
}

SingleLens
```

Optimization 대상 변수[$x0~$xn]
처음 초기값은 **0**

RBT Optimization: Focusing (3)



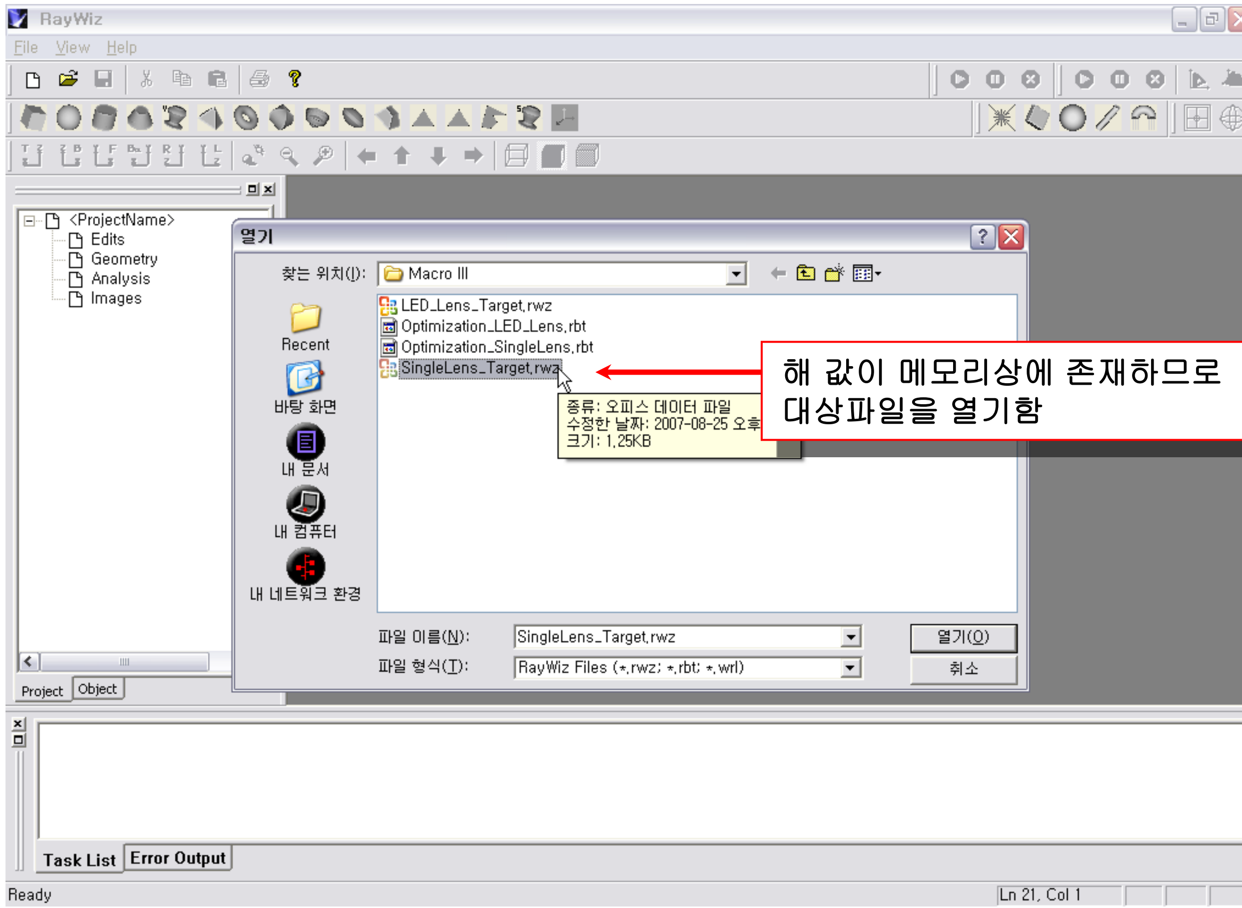


RBT Optimization: Focusing (4)

```
X0 = [0.1375383]
X1 = [1.124533]
X2 = [-3]
Optimize result data saved at 'Optimization_SingleLens_opt20070826212419.csv'
Done Simulation.
```

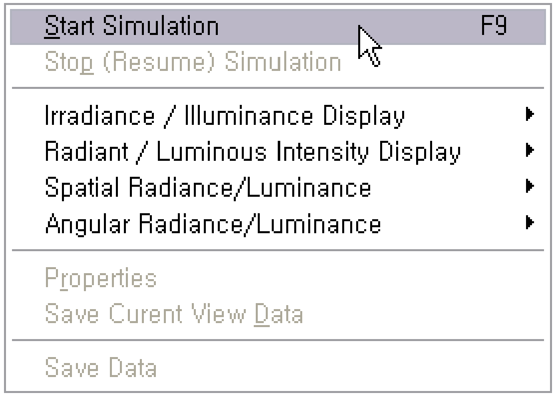
Task List | Error Output

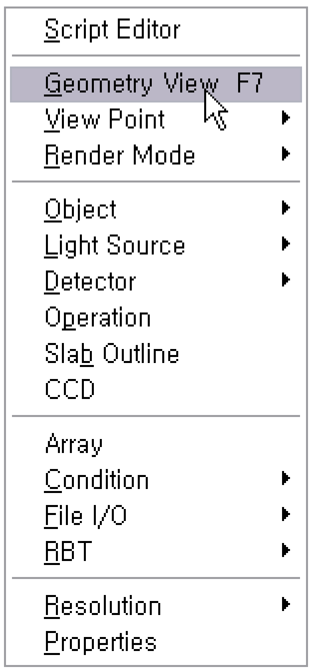
결과값 및 저장파일명

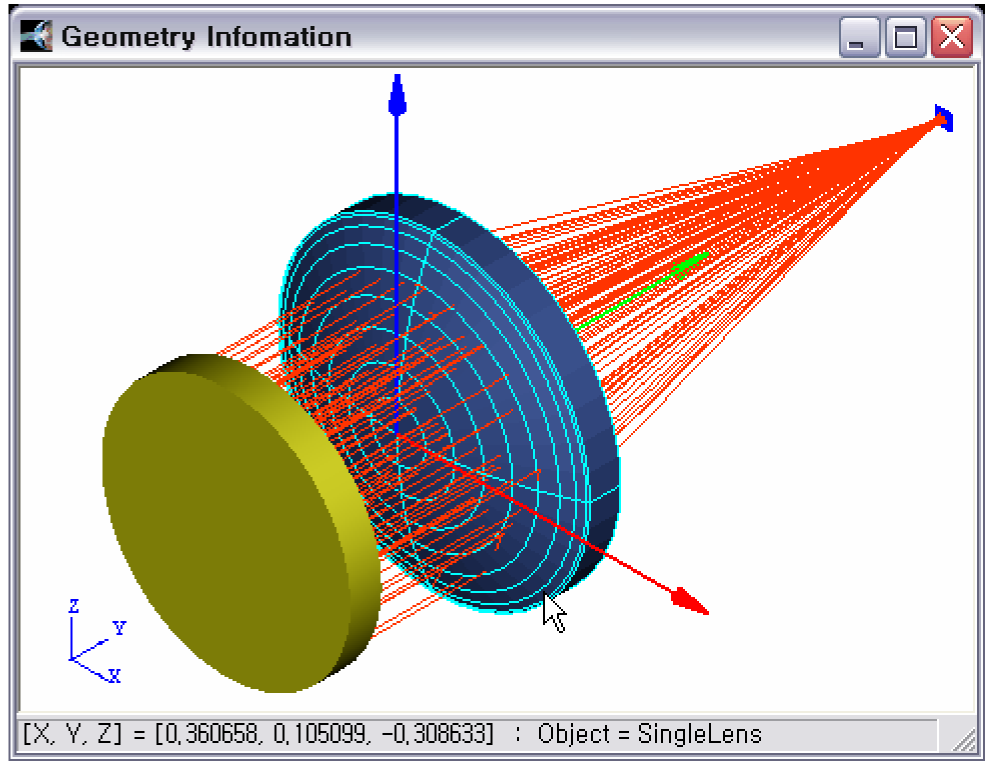


해 값이 메모리상에 존재하므로
대상파일을 열기함

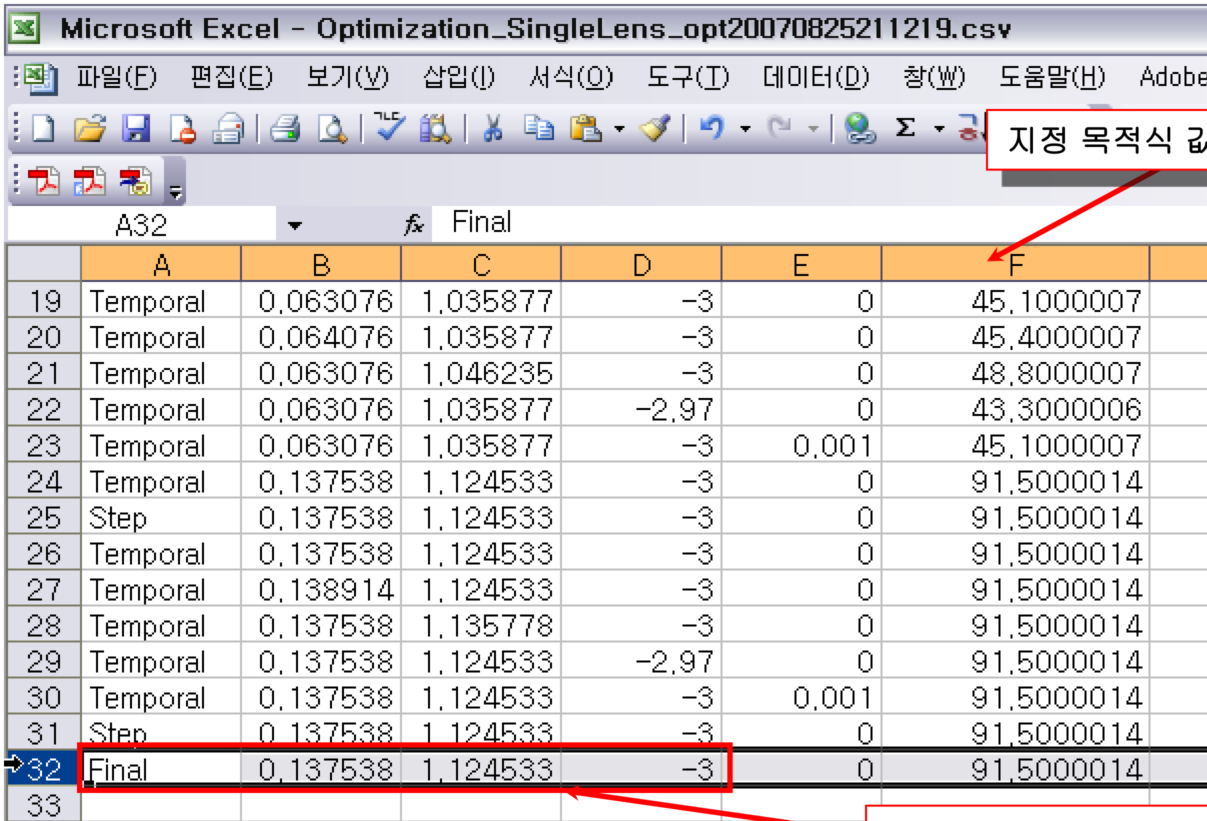
RBT Optimization: Focusing (5)







RBT Optimization: Focusing (6)

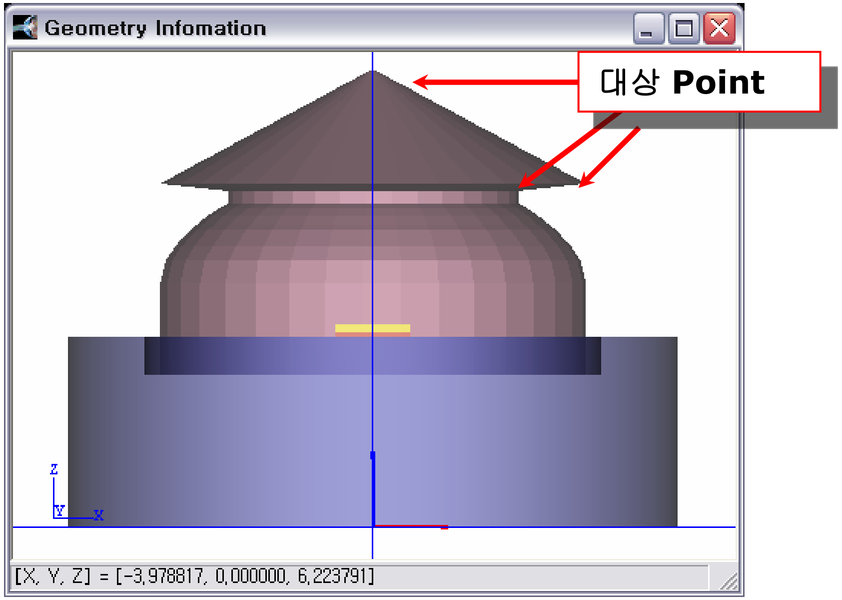


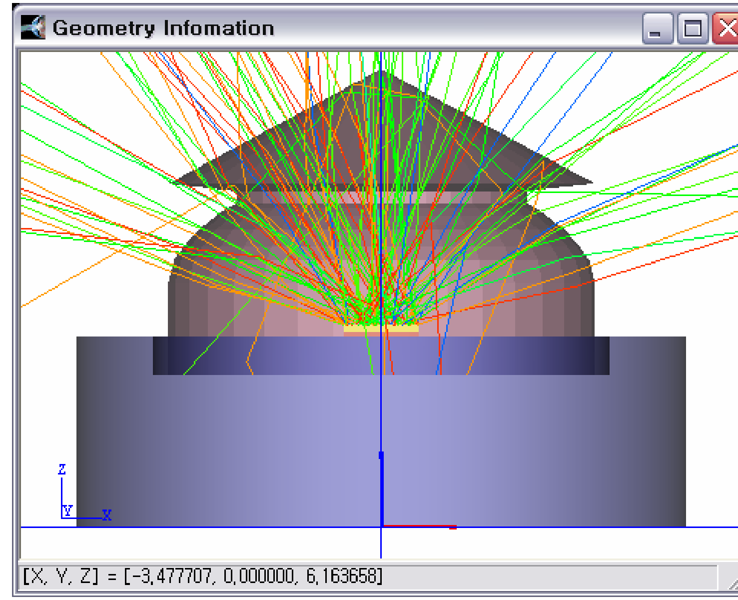
	A	B	C	D	E	F
19	Temporal	0.063076	1.035877	-3	0	45.1000007
20	Temporal	0.064076	1.035877	-3	0	45.4000007
21	Temporal	0.063076	1.046235	-3	0	48.8000007
22	Temporal	0.063076	1.035877	-2.97	0	43.3000006
23	Temporal	0.063076	1.035877	-3	0.001	45.1000007
24	Temporal	0.137538	1.124533	-3	0	91.5000014
25	Step	0.137538	1.124533	-3	0	91.5000014
26	Temporal	0.137538	1.124533	-3	0	91.5000014
27	Temporal	0.138914	1.124533	-3	0	91.5000014
28	Temporal	0.137538	1.135778	-3	0	91.5000014
29	Temporal	0.137538	1.124533	-2.97	0	91.5000014
30	Temporal	0.137538	1.124533	-3	0.001	91.5000014
31	Step	0.137538	1.124533	-3	0	91.5000014
32	Final	0.137538	1.124533	-3	0	91.5000014
33						

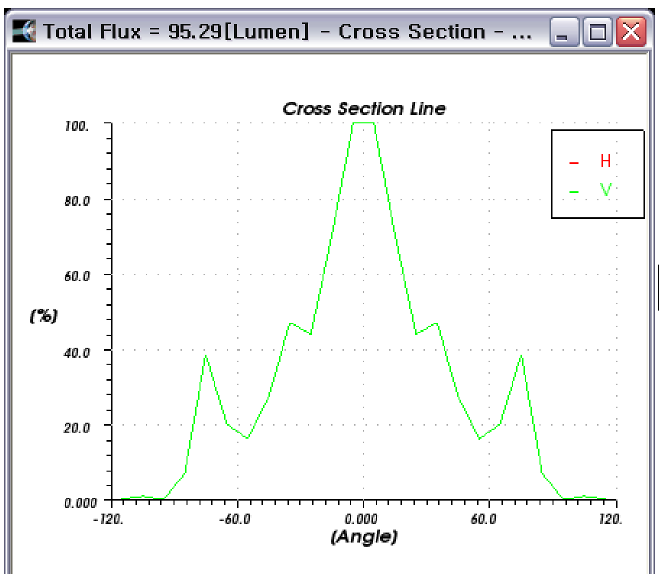
지정 목적식 값의 변화

대상 변수 해의 값의 변화

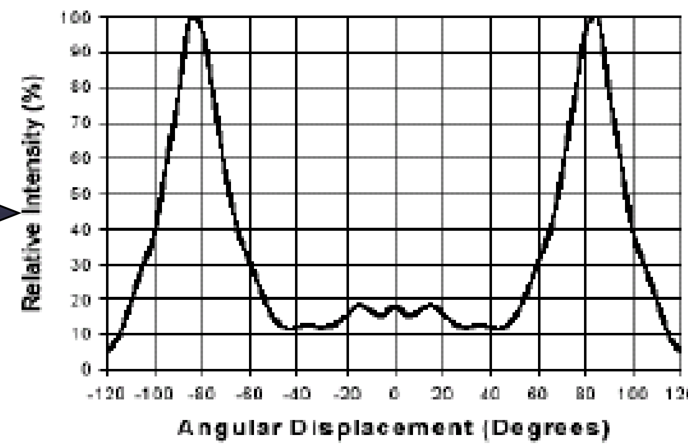
RBT Optimization: Angular Profile (1)







Target Profile



RBT Optimization: Angular Profile (2)

LED_Lens_Target.rwz

```
global_settings {
    rays {
        trace_off_ratio 0.000001
        total_ray_number 50000
        max_trace_level 10
        trace_path_view 100
        split_ray off
        fresnel_area_effect on
        random_spectrum on
    }
    force_parsing off
    polarization off
    units millimeters
    summary on
    geometry_intensity on
}
```

LED_Lens_Target.rwz

```
#declare DieBase =
box
{
    <-0.5, -0.5, 0>, <0.5, 0.5, 0.1>
    color <1,0,0>
    surface {
        both {
            reflectance {1 gaussian 5}
            transmittance{0}
        }
    }
    translate<0,0,2.4499>
}
```

반사체 이므로 **IOR** 항목 삭제

RBT Optimization: Angular Profile (3)

LED_Lens_Target.rwz

```
//**** BaseCup ***
#declare Cup=
difference{
        cylinder{<0,0,0>, <0,0,2.5>, 4}
        cylinder{<0,0,2.0>, <0,0,3>, 3}
        color <0,0,0>
        surface{
            both
            {
                transmittance{0}
                reflectance {0}
            }
        }
}
```

Cylinder 를 이용한 CSG Object

LED_Lens_Target.rwz

```
//*** Lens ***
#declare Lens=
lathe{
        linear_spline
        14,
        <0,0>,<2.8,0>,<2.8,1.2>, //base vertical
        <2.73,1.49>,<2.6,1.68>,<2.4,1.9>,<2.32,1.96>,
        <2.17,2.08>,<2.04,2.17>,<1.91,2.24>,
        <1.91+$x0,2.4>,//<1.5,2.4>, //1st variable
        <2.8+$x1,2.5+$x2>,//<2.3,3.9>, //2nd variable
        <0,4+$x3>//<0,2.4>, //3rd variable
        <0,0>
        ior 1.49
        color <1,0,1>
        surface {
            both
            {
                Probabilistic_split on
            }
        }
        rotate <90,0,0>
        translate <0,0,2.001>
}
```

초기 형상정보 정의 및 변수 설정

RBT Optimization: Angular Profile (4)

LED_Lens_Target.rwz

```
#declare DieLight =
light_source
{
    <0,0,0>
    color <1,1,1>
    object_light
    {
        surface
        box
        {
            <-0.5, -0.5, 0>, <0.5, 0.5, 0.1>
        }
```

Box Surface Light를 정의하고
Spectrum은 **E**광원으로 설정

LED_Lens_Target.rwz

```
            emittance
            {
                front_surface
                {
                    on
                    angular_distribution
                    {
                        lambertian
                    }
                    spatial_distribution
                    {
                        uniform
                    }
                }
            }
        }
        emittance
        {
            photometric 100
            measured_whole off
            aim_sphere <0, 90>
        }
        point_at <0, 0, 1>
        translate <0,0,2.6>
}
```

RBT Optimization: Angular Profile (5)

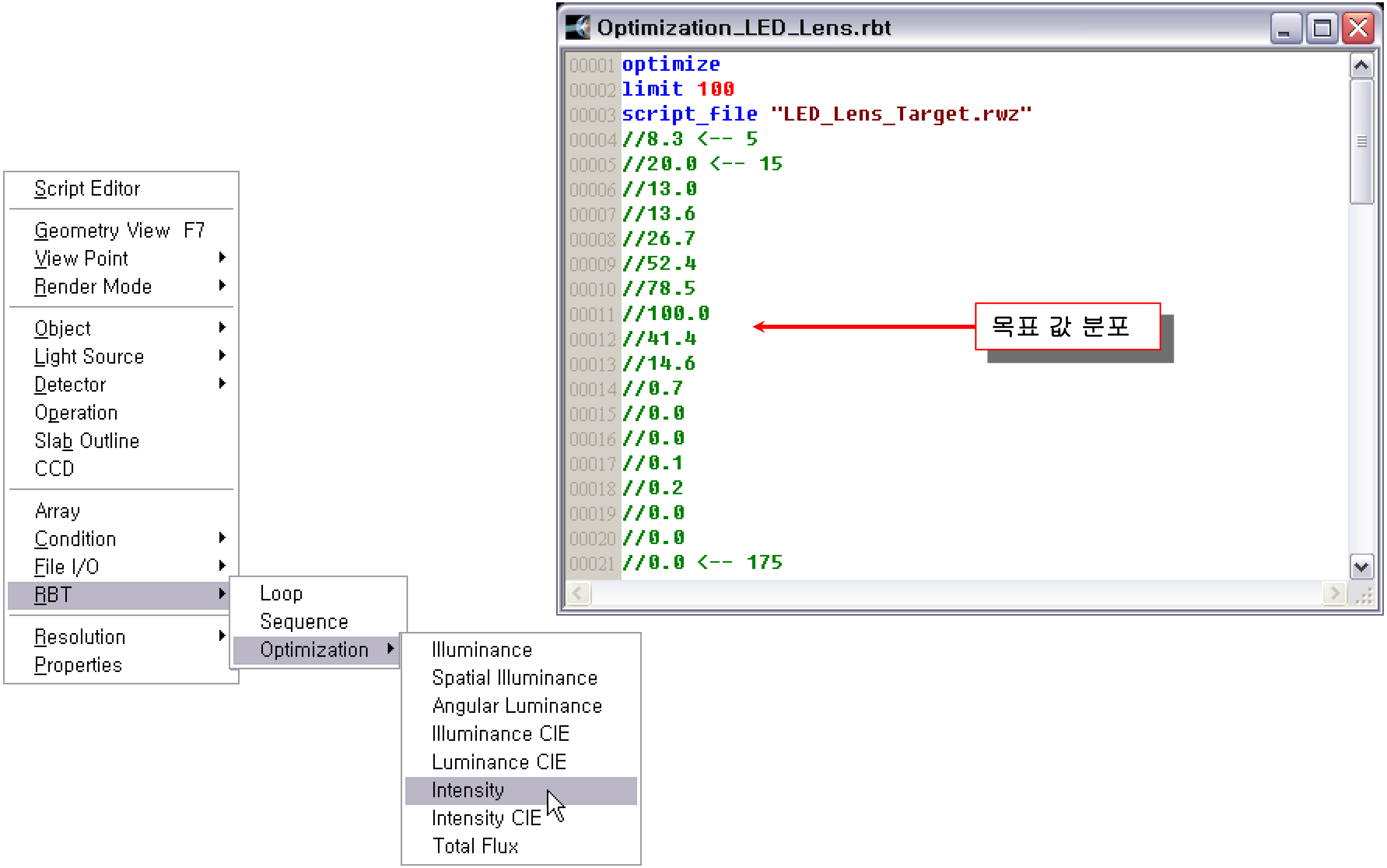
LED_Lens_Target.rwz

```
#declare AngularDetector =
sphere
{
     <0, 0, 0>, 25
     detector{
        photometric
        symmetry on
        bin_size <18,18>
     }
}

Cup
DieBase
DieLight
Lens
AngularDetector
```

회전 대칭으로 설정

앞서 정의한
Object 들을 Call

RBT Optimization: Angular Profile (6)



RBT Optimization: Angular Profile (7)

Optimization_LED_Lens.rbt

```
objective_function
{
    6
    {
        "AngularDetector",
        Intensity function_express
        "pow(abs(bin[1,0]/totalMax-8.3),2)+
         pow(abs(bin[2,0]/totalMax-20.0),2)+
         pow(abs(bin[3,0]/totalMax-13.0),2)" = 0
    }
    {
        "AngularDetector",
        Intensity function_express
        "pow(abs(bin[4,0]/totalMax-13.6),2)+
         pow(abs(bin[5,0]/totalMax-26.7),2)+
         pow(abs(bin[6,0]/totalMax-52.4),2)" = 0
    }
    {
        "AngularDetector",
        Intensity function_express
        "pow(abs(bin[7,0]/totalMax-78.5),2)+
         pow(abs(bin[8,0]/totalMax-100),2)+
         pow(abs(bin[9,0]/totalMax-41.4),2)" = 0
    }
```

Optimization_LED_Lens.rbt

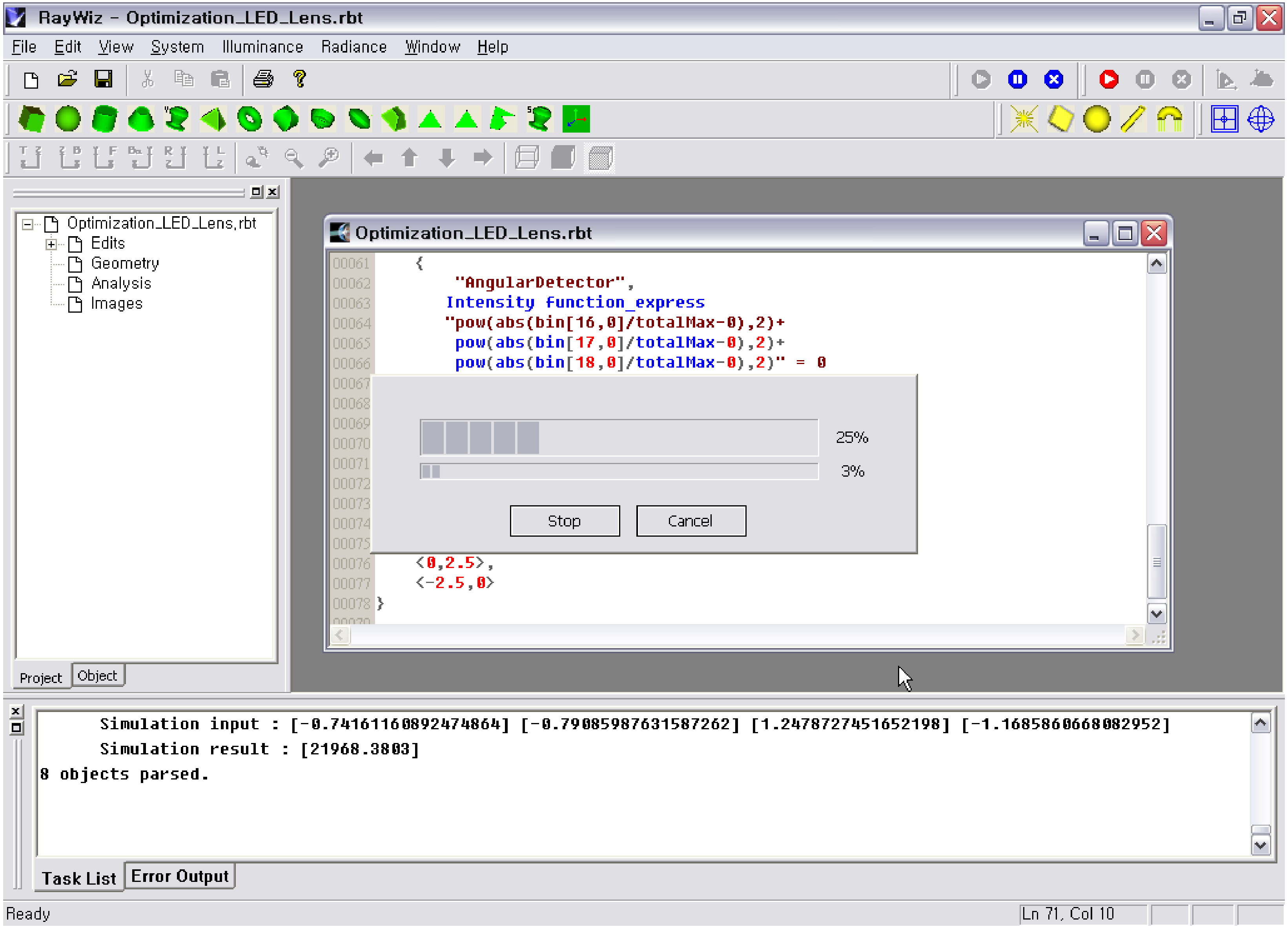
```
    {
        "AngularDetector",
        Intensity function_express
        "pow(abs(bin[10,0]/totalMax-14.6),2)+
         pow(abs(bin[11,0]/totalMax-0.7),2)+
         pow(abs(bin[12,0]/totalMax-0),2)" = 0
    }
    {
        "AngularDetector",
        Intensity function_express
        "pow(abs(bin[13,0]/totalMax-0),2)+
         pow(abs(bin[14,0]/totalMax-0.1),2)+
         pow(abs(bin[15,0]/totalMax-0.2),2)" = 0
    }
    {
        "AngularDetector",
        Intensity function_express
        "pow(abs(bin[16,0]/totalMax-0),2)+
         pow(abs(bin[17,0]/totalMax-0),2)+
         pow(abs(bin[18,0]/totalMax-0),2)" = 0
    }
}
```

Optimization_LED_Lens.rbt

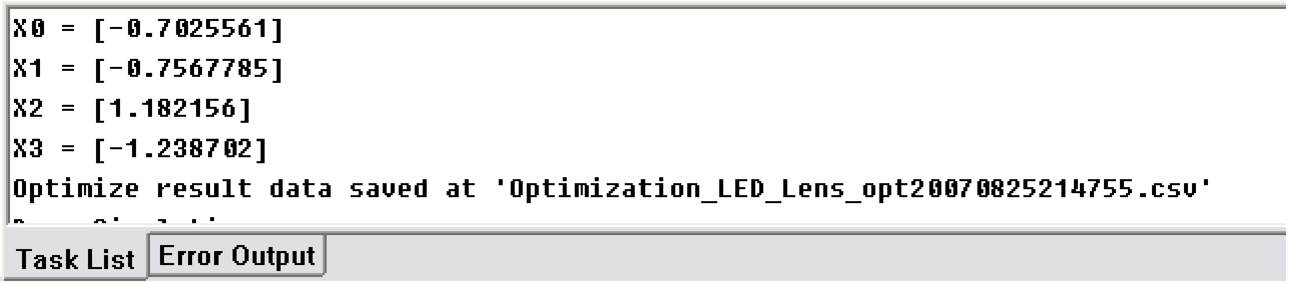
```
target_value { INTENSITY_VALUE 0 }
delta 0.01
variables
{
    4,
    <-1.5,0>,
    <-1.5,0>,
    <0,2.5>,
    <-2.5,0>
}
```

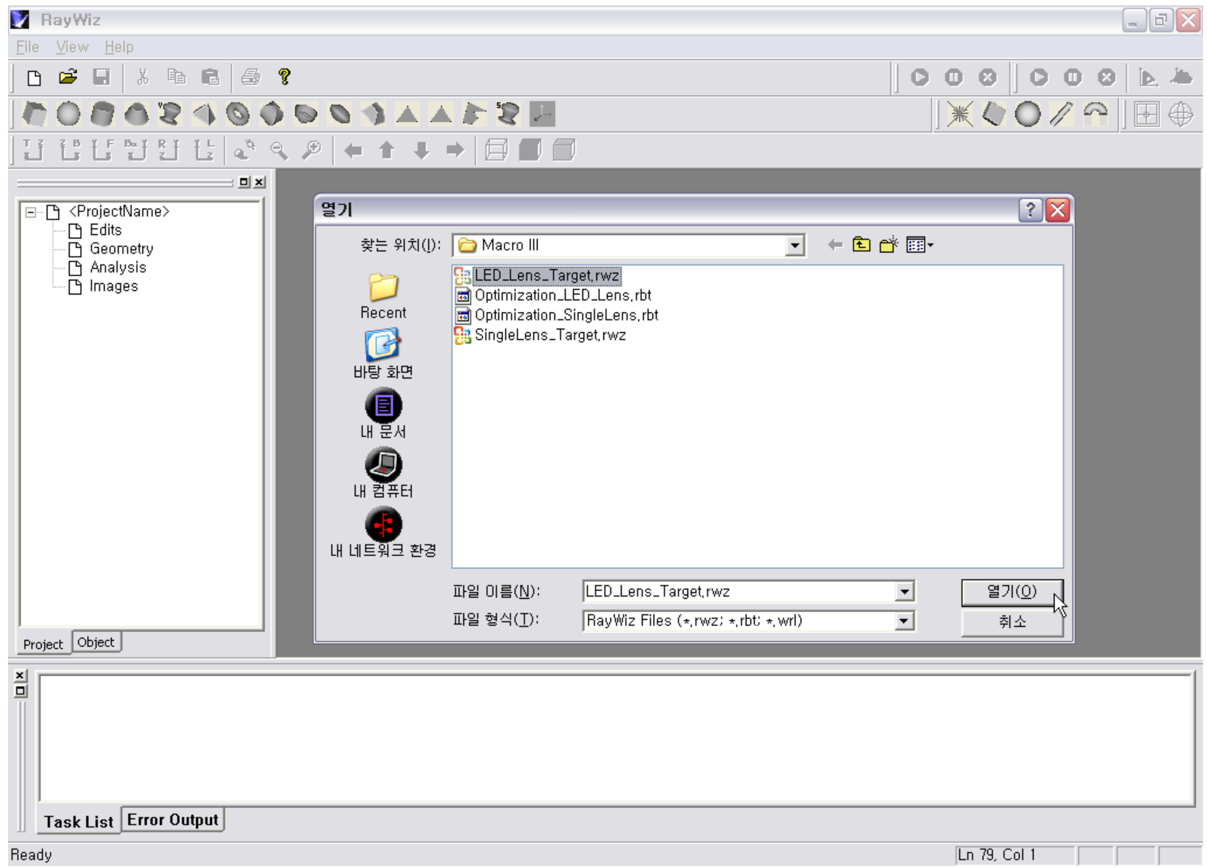
변수 값 범위

RBT Optimization: Angular Profile (8)

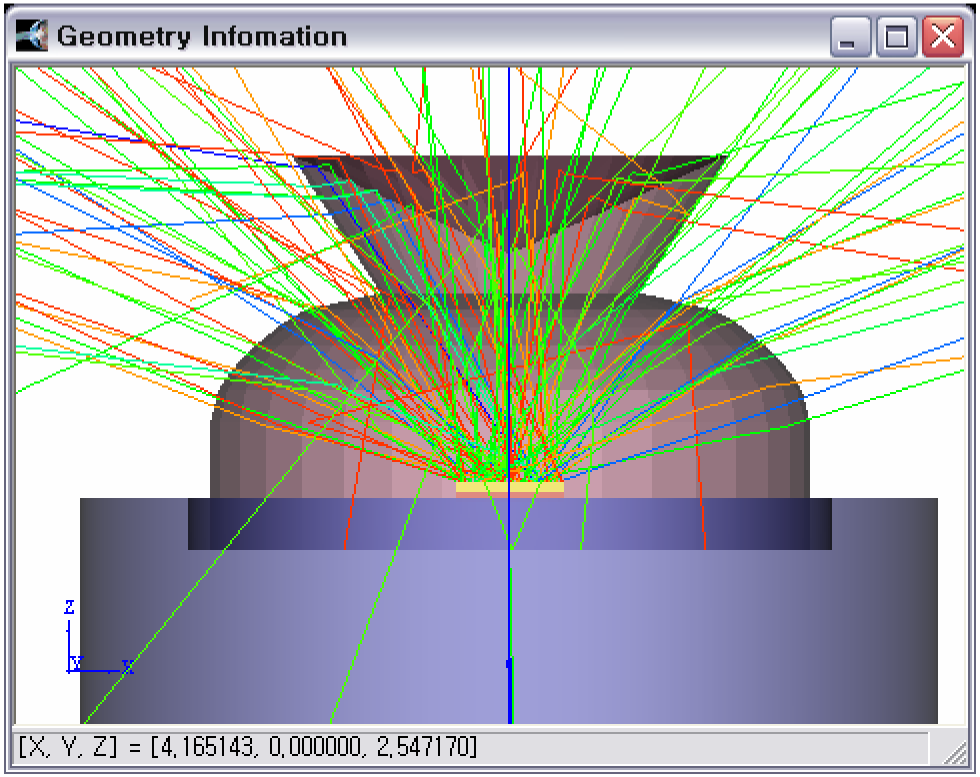


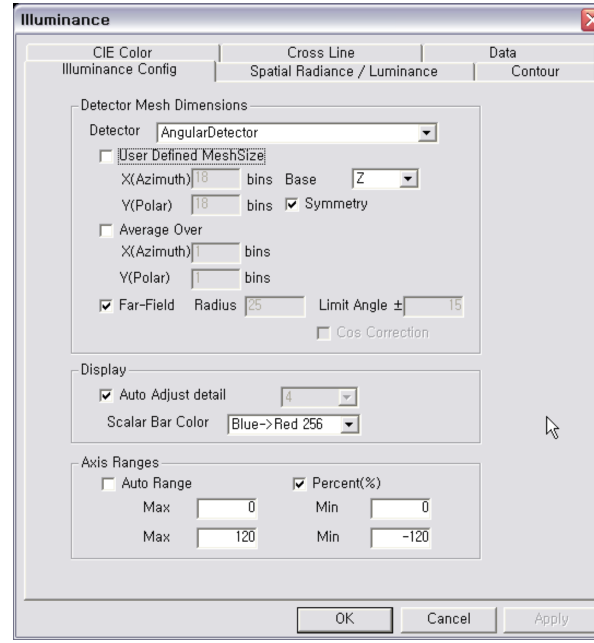
RBT Optimization: Angular Profile (9)

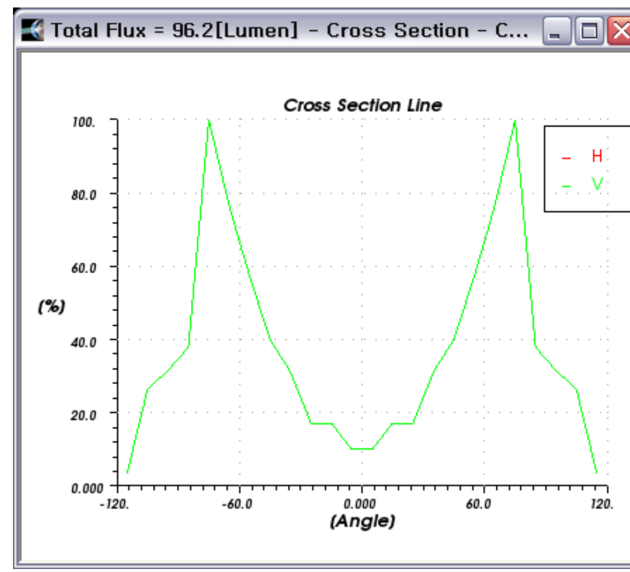




RBT Optimization: Angular Profile (10)







RBT Optimization: Angular Profile (11)

Microsoft Excel - Optimization_LED_Lens_opt20070822185117.csv

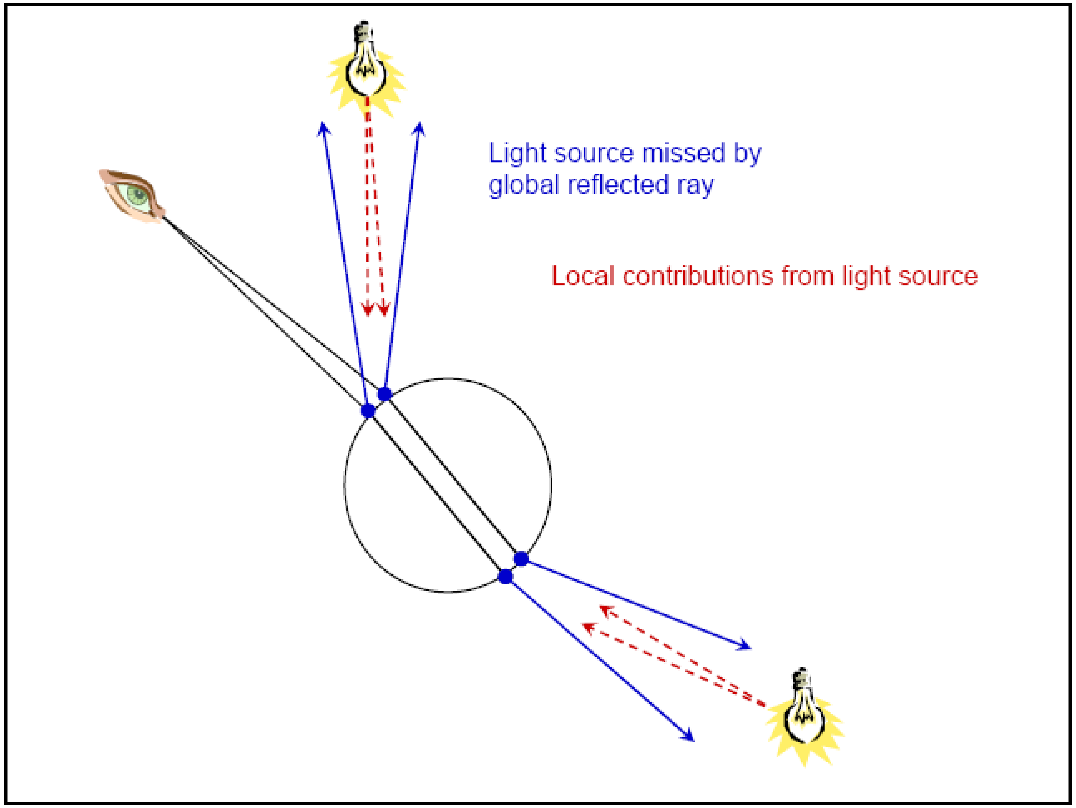
파일(F) 편집(E) 보기(V) 삽입(I) 서식(O) 도구(T) 데이터(D) 창(W) 도움말(H) Adobe PDF(B)

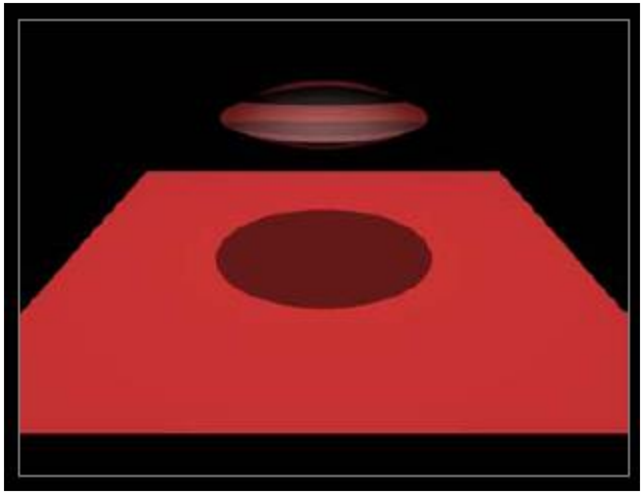
A1 | Calculation Type

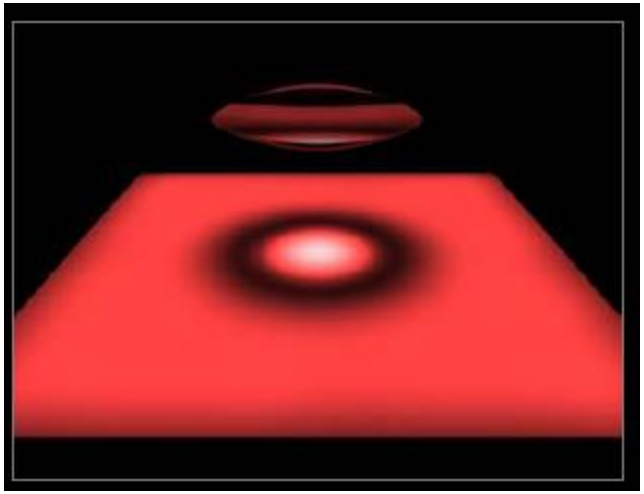
	A	B	C	D	E	F	G	H	I	J	K	L
79	Temporal	-0.4733	-0.50983	0.796406	-1.65028	0	0	0	0	0	0	21986.15
80	Temporal	-0.64158	-0.69109	1.079549	-1.34818	0	0	0	0	0	0	21973.35
81	Temporal	-0.70256	-0.75678	1.182156	-1.2387	0	0	0	0	0	0	21967.81
82	Step	-0.70256	-0.75678	1.182156	-1.2387	0	0	0	0	0	0	21967.81
83	Temporal	-0.70256	-0.75678	1.182156	-1.2387	0	0	0	0	0	0	21967.81
84	Temporal	-0.69553	-0.75678	1.182156	-1.2387	0	0	0	0	0	0	21968.22
85	Temporal	-0.70256	-0.74921	1.182156	-1.2387	0	0	0	0	0	0	21969.71
86	Temporal	-0.70256	-0.75678	1.193977	-1.2387	0	0	0	0	0	0	21970.98
87	Temporal	-0.70256	-0.75678	1.182156	-1.22632	0	0	0	0	0	0	21969.47
88	Temporal	-0.70256	-0.75678	1.182156	-1.2387	0.001	0	0	0	0	0	21967.81
89	Temporal	-0.70256	-0.75678	1.182156	-1.2387	0	0.001	0	0	0	0	21967.81
90	Temporal	-0.70256	-0.75678	1.182156	-1.2387	0	0	0.001	0	0	0	21967.81
91	Temporal	-0.70256	-0.75678	1.182156	-1.2387	0	0	0	0.001	0	0	21967.81
92	Temporal	-0.70256	-0.75678	1.182156	-1.2387	0	0	0	0	0.001	0	21967.81
93	Temporal	-0.70256	-0.75678	1.182156	-1.2387	0	0	0	0	0	0.001	21967.81
94	Temporal	-1.5	-1.5	0	-2.5	0	0	0	0	0	0	22048.96
95	Temporal	-1.5	-1.5	0	-2.5	0	0	0	0	0	0	22048.96
96	Temporal	-1.0607	-1.09057	0.651231	-1.80517	0	0	0	0	0	0	22001.84
97	Temporal	-0.83836	-0.88335	0.980834	-1.4535	0	0	0	0	0	0	21989.88
98	Temporal	-0.7512	-0.80211	1.110045	-1.31564	0	0	0	0	0	0	21978.44
99	Temporal	-0.71914	-0.77224	1.157566	-1.26494	0	0	0	0	0	0	21971.67
100	Temporal	-0.7082	-0.76204	1.173792	-1.24763	0	0	0	0	0	0	21971.13
101	Temporal	-0.70393	-0.75806	1.180119	-1.24088	0	0	0	0	0	0	21969.4
102	Temporal	-0.70281	-0.75701	1.181782	-1.2391	0	0	0	0	0	0	21967.95
103	Temporal	-0.70263	-0.75685	1.182049	-1.23882	0	0	0	0	0	0	21967.77
104	Step	-0.70256	-0.75678	1.182156	-1.2387	0	0	0	0	0	0	21967.77
105	Final	-0.70256	-0.75678	1.182156	-1.2387	0	0	0	0	0	0	21967.77

Radiance

Radiance: Basic Concept (1)

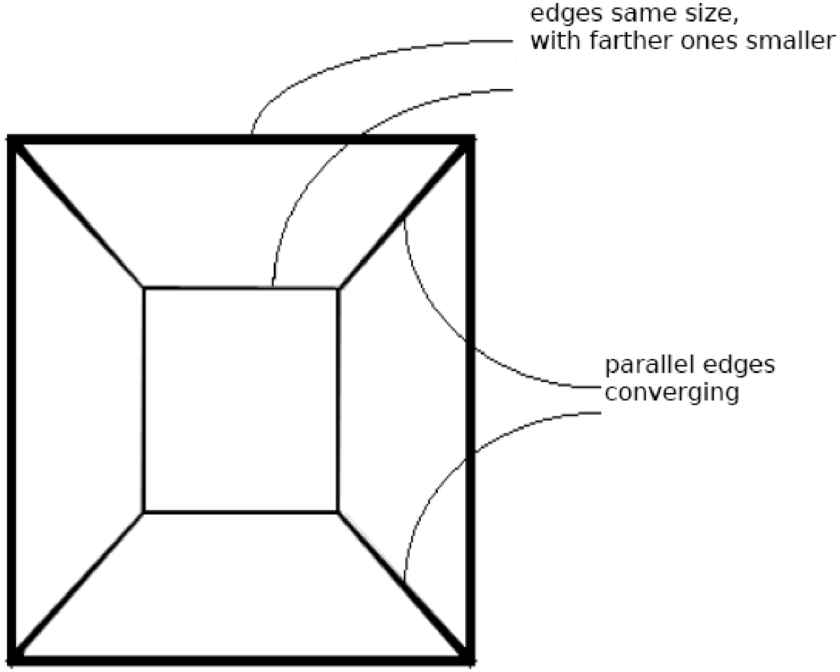


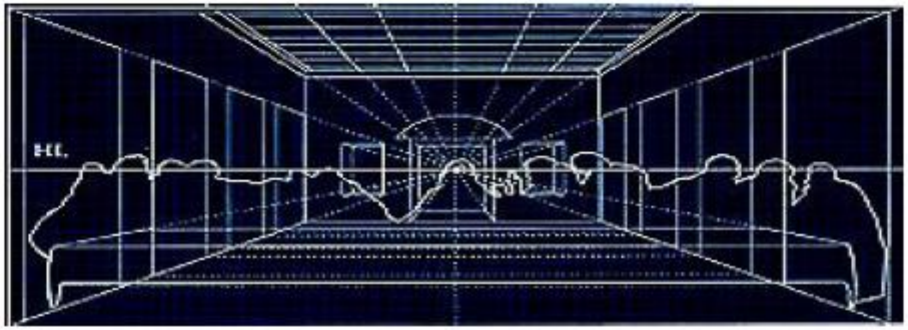




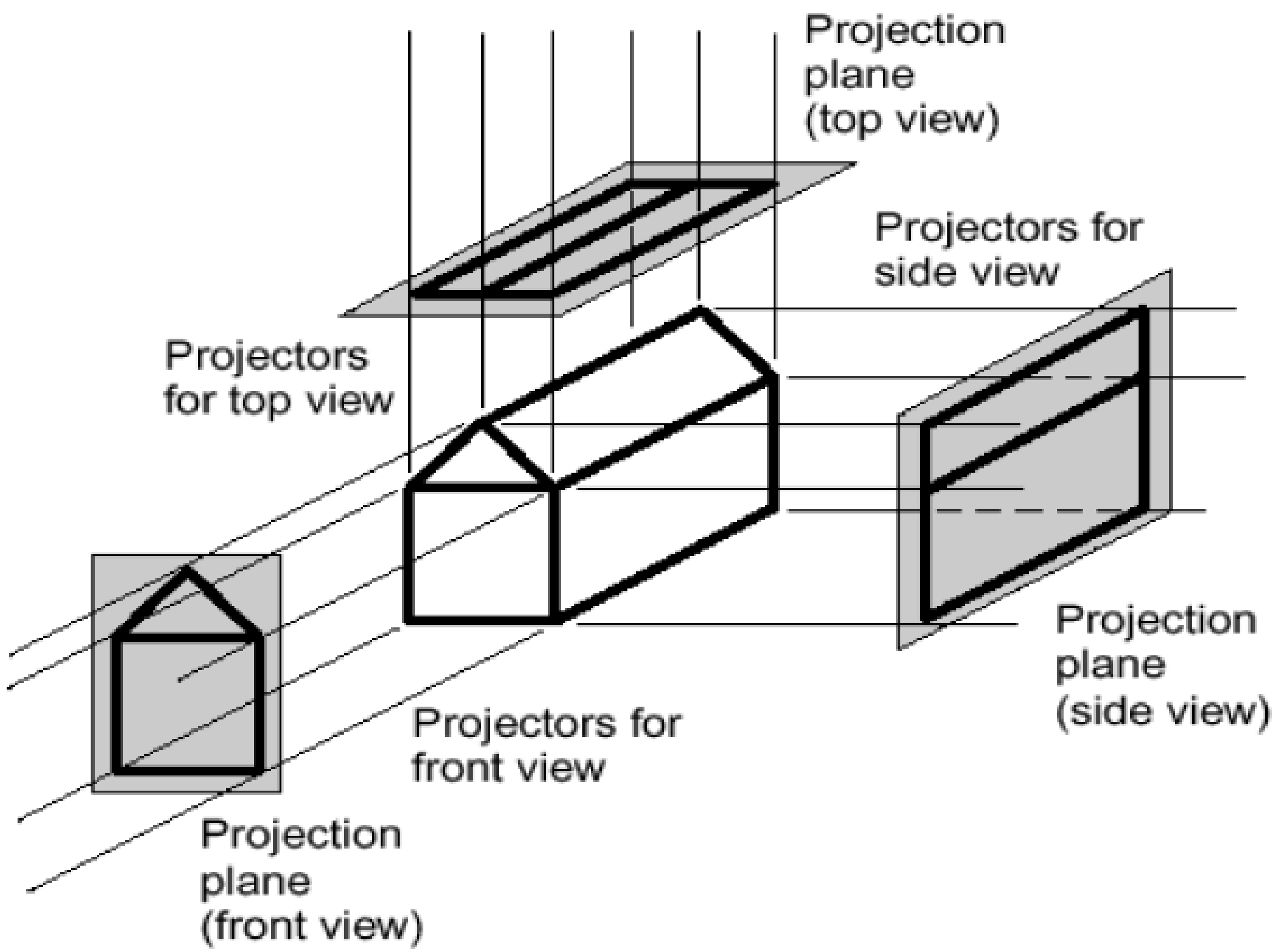
Bi-Directional Ray

Radiance: Basic Concept (2) – Perspective View

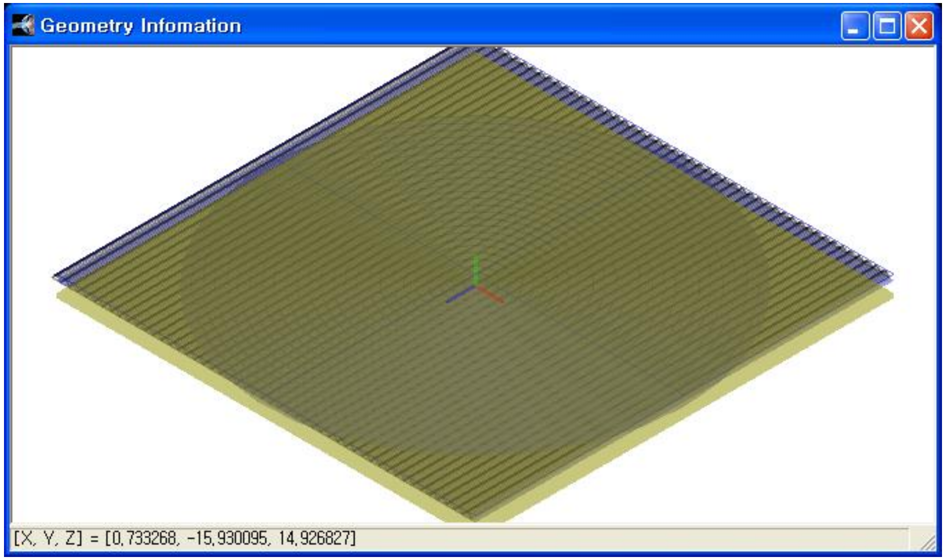


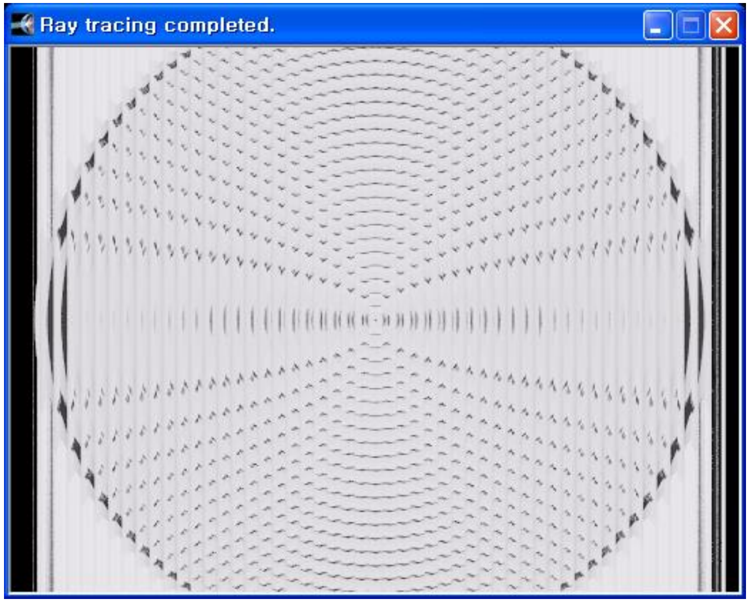


Radiance: Basic Concept (3) – Orthographic View

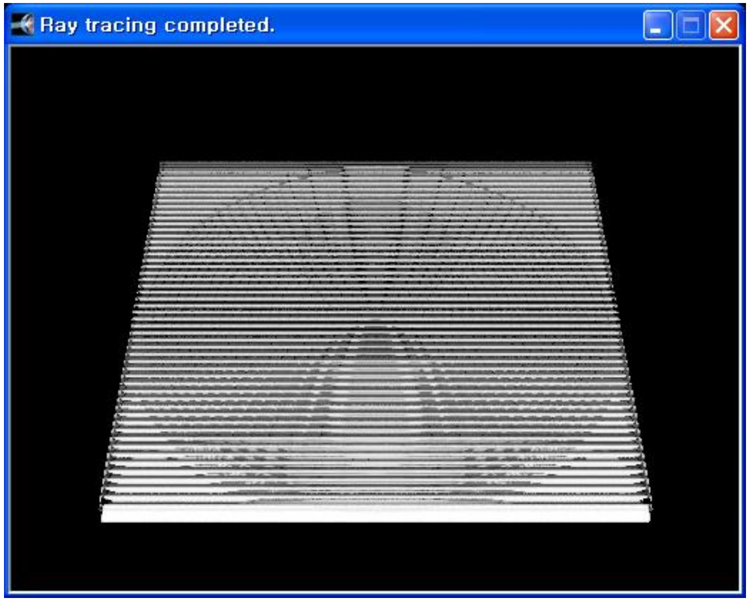


Radiance: Camera (1)






```
Camera
{
        Perspective | orthographic
        Location < X,Y,Z >
        Direction < X,Y,Z >
        Look_at < X,Y,Z >
        Up < X,Y,Z >
        Right < X,Y,Z >
        Angle degree(각도)
}
```



Radiance: Camera (2)

Moire.rwz

```
#declare SampleSize=50;
#declare FresnelPitch=1;
#declare LentiPitch=1;

global_settings {
    rays {
        trace_off_ratio 0.0001
        total_ray_number 100000
        max_trace_level 100
        trace_path_view 100
        split_ray off
    }
}

camera {
    location <0, 100, 0>
    look_at <0, 0, 0>
    angle 30
}
```

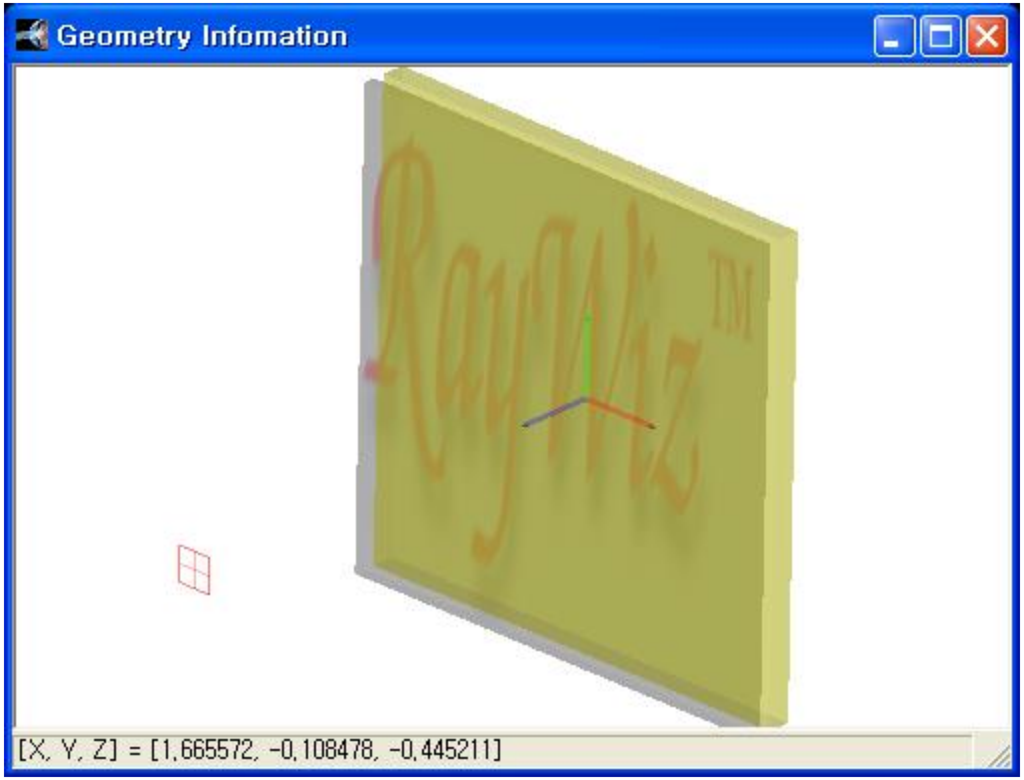
```
#declare FresnelMaxHeight=FresnelPitch/2;
#declare Fresnel_Lens =
lathe
{
    linear_spline 3*int(SampleSize/FresnelPitch/2)+1,
    #declare i=0;
    #while (i < int(SampleSize/FresnelPitch/2))
        <i*FresnelPitch, 0>,
        <i*FresnelPitch+FresnelPitch*0.05, FresnelMaxHeight>,
        <i*FresnelPitch+FresnelPitch, 0>,
    #declare i=i+1;
    #end
    <0,0>
    color <1, 1, 1>
    ior 1.5
    surface{
       both{
               probabilistic_split on
           }
    }
}
```

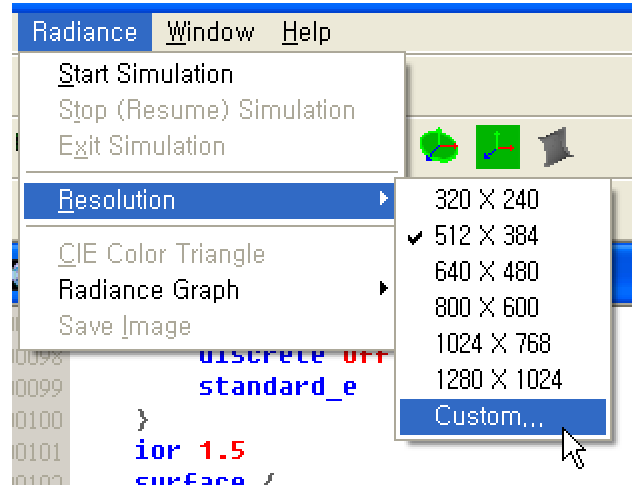
```
#declare NumLentiPitch= int(SampleSize/LentiPitch);
#declare LentiUnit =
extrusion
{
    linear_spline linear_sweep
    0, SampleSize,
    12,
    <-LentiPitch/2, 0>,
    <-LentiPitch/2+LentiPitch/10, 1.5*LentiPitch/10>,
    <-LentiPitch/2+2*LentiPitch/10, 2.7*LentiPitch/10>,
    <-LentiPitch/2+3*LentiPitch/10, 3.7*LentiPitch/10>,
    <-LentiPitch/2+4*LentiPitch/10, 4.5*LentiPitch/10>,
    <-LentiPitch/2+5*LentiPitch/10, 4.7*LentiPitch/10>,
    <-LentiPitch/2+6*LentiPitch/10, 4.5*LentiPitch/10>,
    <-LentiPitch/2+7*LentiPitch/10, 3.7*LentiPitch/10>,
    <-LentiPitch/2+8*LentiPitch/10, 2.7*LentiPitch/10>,
    <-LentiPitch/2+9*LentiPitch/10, 1.5*LentiPitch/10>,
    <-LentiPitch/2+LentiPitch, 0>,
    <-LentiPitch/2,0>
    rotate <-90,0,0>
    color <1, 1, 1>
    ior 1.5
    slab
}
```

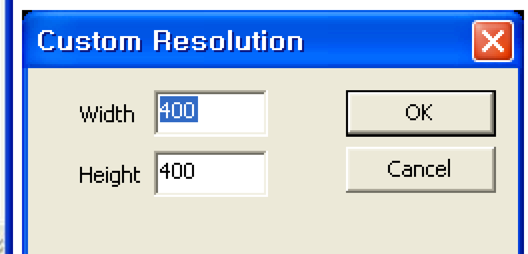
Radiance: Camera (3)

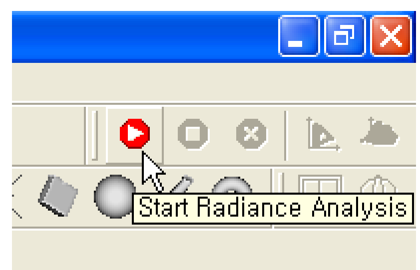
```
#macro LentiArray()
    #declare Count_i=0;
    #while (Count_i < NumLentiPitch)
            object{
                LentiUnit
                translate <-SampleSize/2+Count_i*LentiPitch,FresnelMaxHeight+LentiPitch/10+0.01,SampleSize/2>
            }
        #declare Count_i=Count_i+1;
    #end
#end

#declare LentiScreen=
union
{
    slab_outline on
    object{LentiBase}
    union{
        slab_outline on
        LentiArray()
    }
}
```

Radiance: CCD (1)

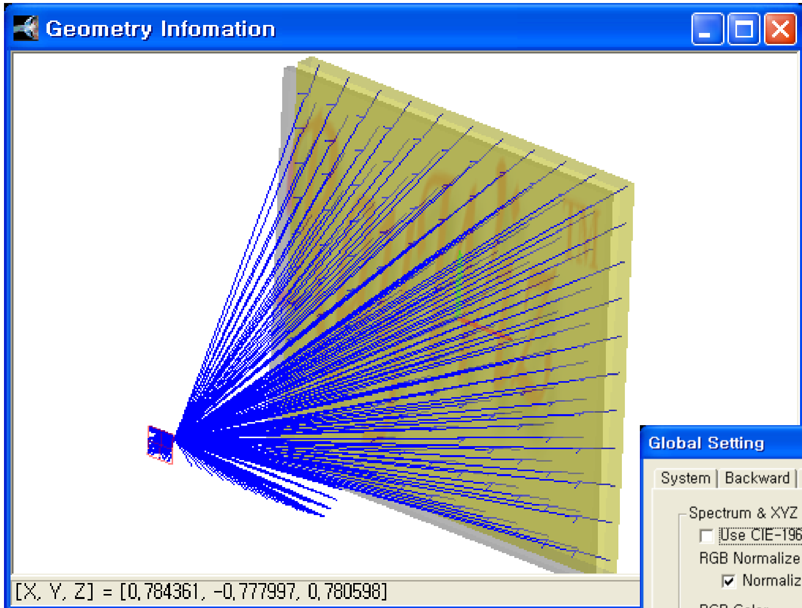


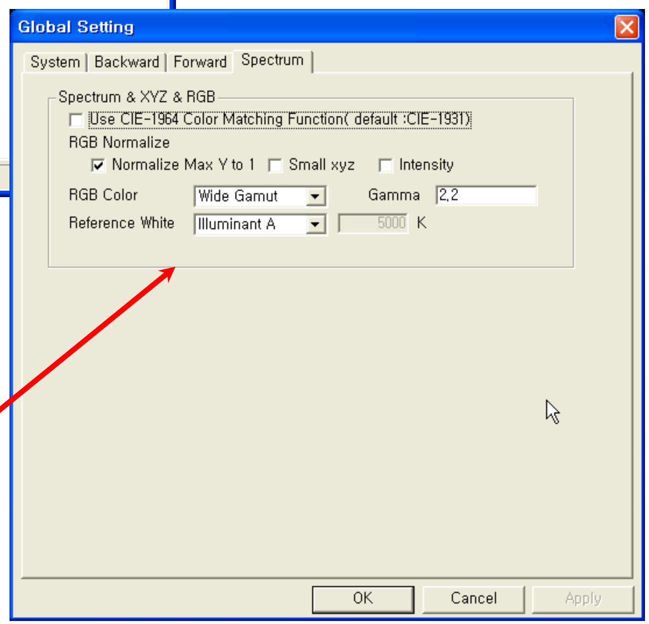






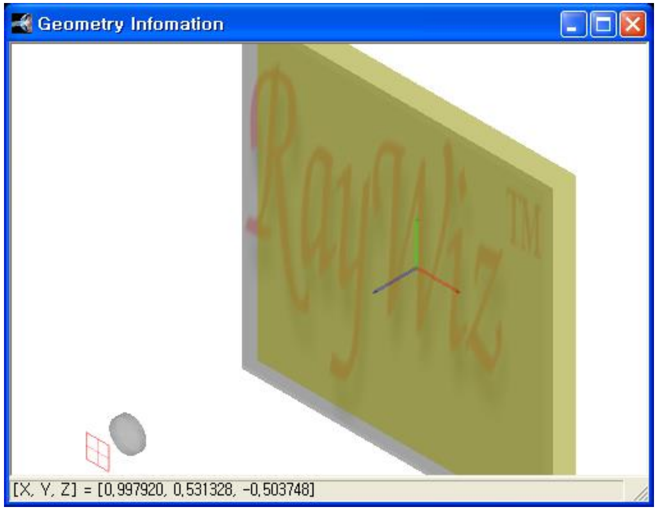
Radiance: CCD (2)



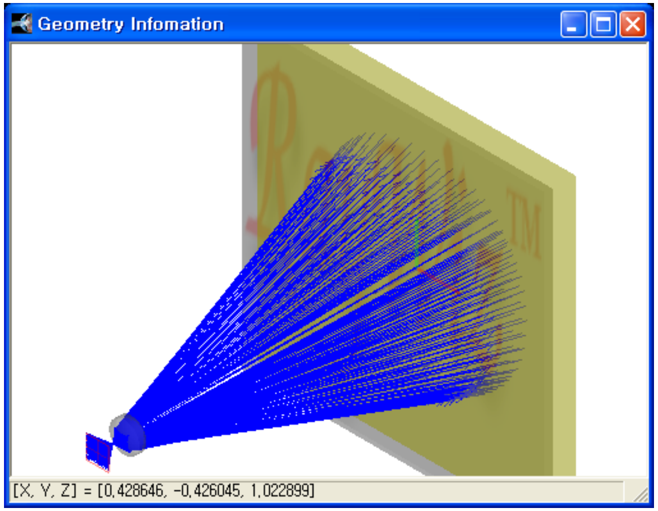


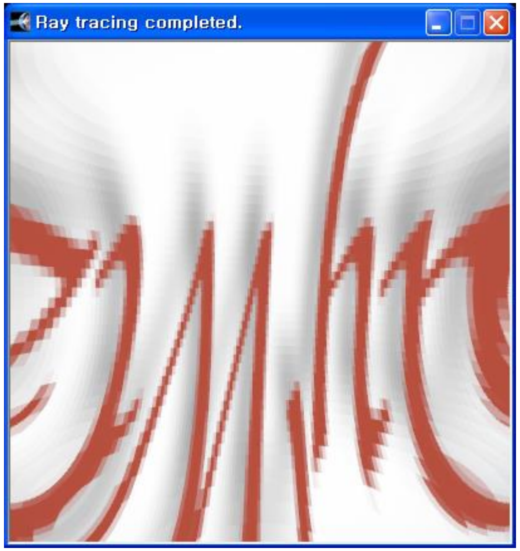
Color Matrix,
Gamma,
Reference White에
따른 변환 조건 선택

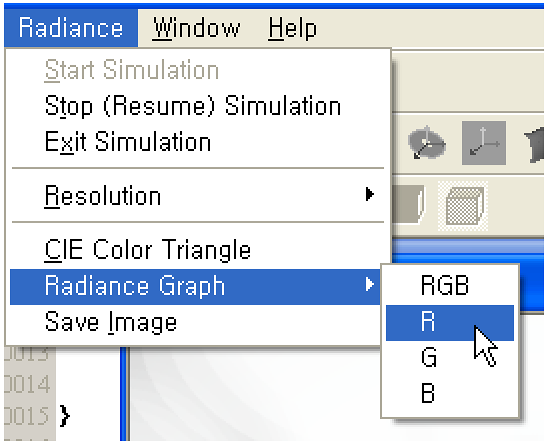
Radiance: CCD (3)

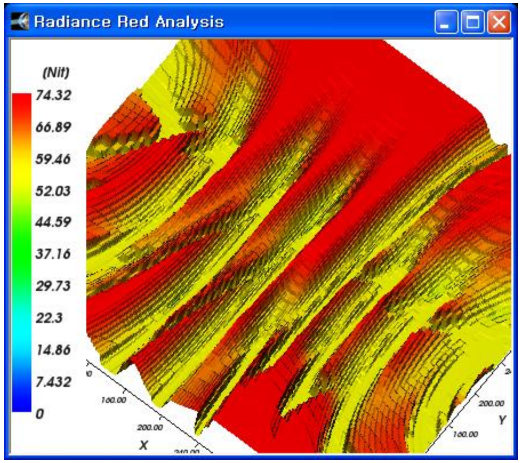


Lens 특성이 반영









Radiance: CCD (4)

CCD-Image.rwz

```
#declare Lens_Flag=1;

global_settings {
    rays {
        trace_off_ratio 0.01
        total_ray_number 100000
        max_trace_level 10
        trace_path_view 100
        split_ray off
        fresnel_area_effect on
    }
    force_parsing off
    polarization off
    units millimeters
}
```

```
camera {
    ccd
    width 0.15
    height 0.15
    aim_area {
        <0, 0, 2> // Center Point
        <0, 0, -1> // Direction Point
        circular 0.0001 // Radius
        //rectangular Width_Value, Height_Value
        }
    location <0, 0, 2.1>
    look_at <0, 0, 0>
    up <0, 0, 1>
    right <1, 0, 0>
}
```

```
#declare ImgBox =
box
{
    <-1, -1, 0.1>, <1, 1, 0.15>
    spectrum {
        discrete off
        standard_e
    }
    ior 1.5
    surface {
        both {
            probabilistic_split on
        }
        front_surface{
          both{
              zone{
                      transmittance{1.0}
                      reflectance{0}
                      "RayWiz.bmp"
                  }
            }
        }
    }
}
```

Radiance: CCD (5)

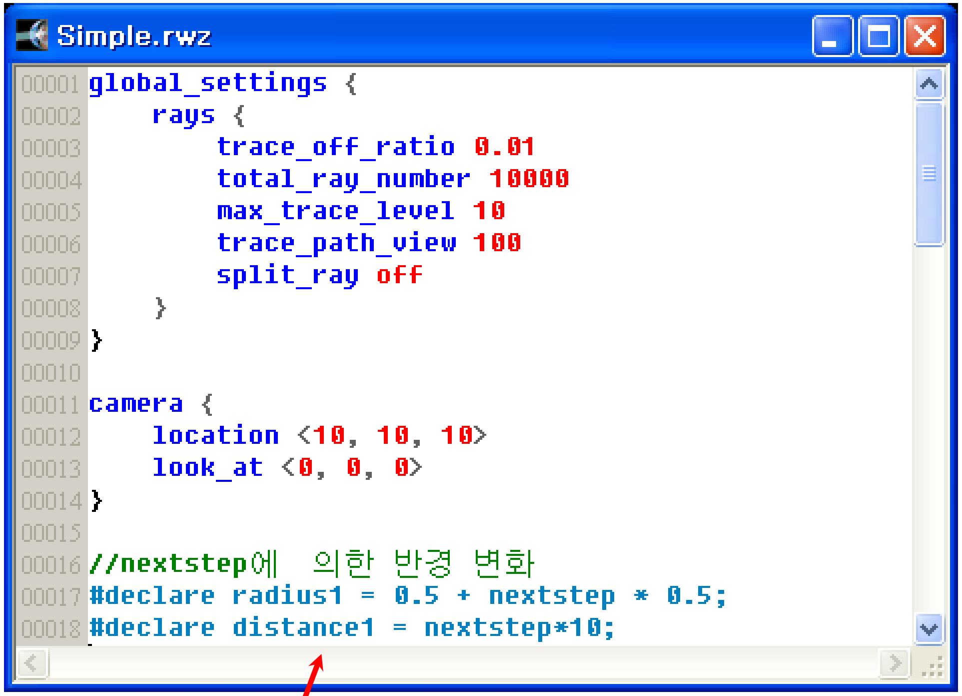
```
#declare BL =
light_source
{
    <0, 0, 0>
    spectrum {
        discrete off
        standard_e
    }
    object_light
    {
        surface
        box
        {
            <-1, -1, 0>, <1, 1, 0.1>
        }
        emittance
        {
            front_surface
            {
                on
                angular_distribution
                {
                    lambertian
                }
                spatial_distribution
                {
                    uniform
                }
            }
            back_surface
            {
                off
            }
            left_surface
            {
                off
            }
            right_surface
            {
                off
            }
            top_surface
            {
                off
            }
            bottom_surface
            {
                off
            }
        }
    }
    emittance
    {
        radiometric 1
        measured_whole on
        aim_sphere <0, 90>
    }
    point_at <0, 0, 3>
}
```

Radiance: CCD (6)

```
#declare TestLens =
optical
{
    circular center_thickness 0.1
    { 0.2, 0.1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 }
    { -0.2, 0.1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 }
    spectrum {
        discrete off
        standard_e
    }
    ior 1.5
    surface {
        both {
            probabilistic_split on
        }
    }
    rotate<90,0,0>
    translate<0,0,1.85>
}

BL
ImgBox
#if(Lens_flag)
    TestLens
#else
#end
```

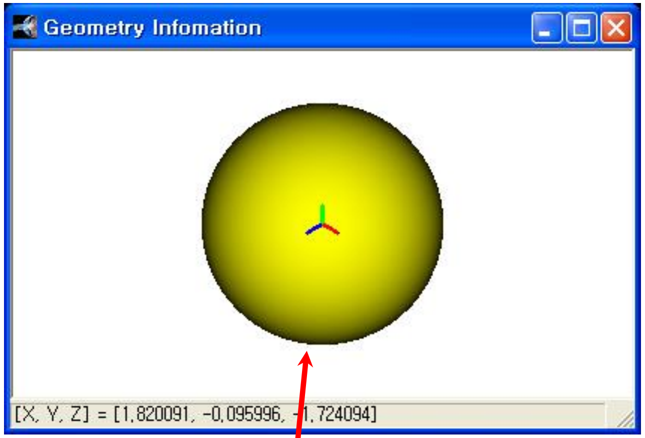
Image RBT

Image RBT: Target System (1)



Simple.rwz

```
global_settings {
    rays {
        trace_off_ratio 0.01
        total_ray_number 10000
        max_trace_level 10
        trace_path_view 100
        split_ray off
    }
}

camera {
    location <10, 10, 10>
    look_at <0, 0, 0>
}

//nextstep에  의한 반경 변화
#declare radius1 = 0.5 + nextstep * 0.5;
#declare distance1 = nextstep*10;
```



Loop Index 증가에 따라
반경
Spectrum
Flux 변화

Loop Index 증가에 따라
반경 증가 **nextstep = 1 ~ n**

Image RBT: Target System (2)

```
#declare SphereSource0 =
light_source
{
    <0, 0, 0>
    // 루프에 따라 스펙트럼 변화
    spectrum {
        3,
        <400, $x0 -1 >,
        <550, $x1 -1 >,
        <650, $x2 -1 >
    }
```

Loop Index 증가에 따라
spectrum 세기 분포 변화
$x0 = 1 ~ n
$x1 = 1 ~ n
$x2 = 1 ~ n

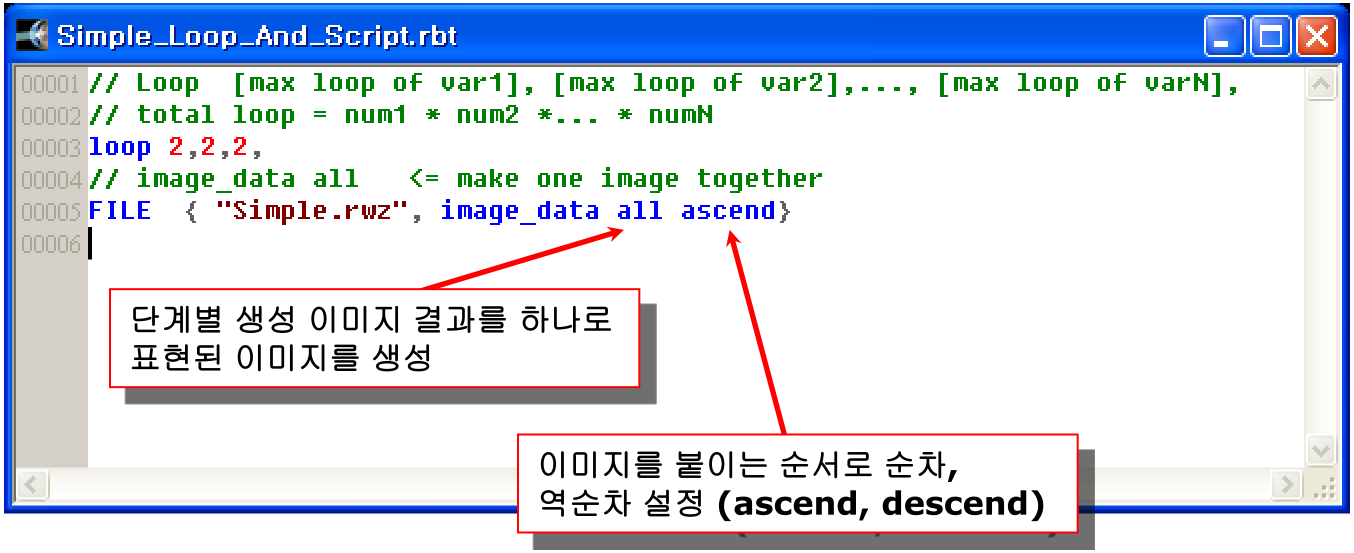
```
object_light
{
    surface
    sphere
    {
        <0, 0, 0>, radius1
    }
    emittance
    {
        angular_distribution
        {
            lambertian
        }
        spatial_distribution
        {
             uniform
        }
    }
}
```

Image RBT: Target System (3)

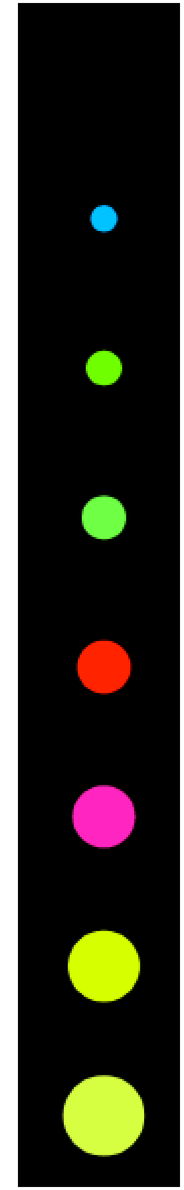
```
    emittance
    {
        // 파워 변화
        radiometric $x0 + $x1 * 10 + $x2 * 100
        measured_whole on
        aim_sphere <0, 180>
    }
    point_at <0, 0, 2>
}

SphereSource0
```

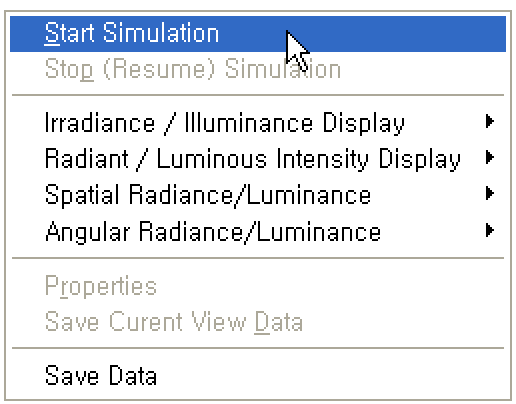
Loop Index 증가에 따라
Flux 세기 변화
$x0 = 1 ~ n
$x1 = 1 ~ n
$x2 = 1 ~ n

Image RBT:

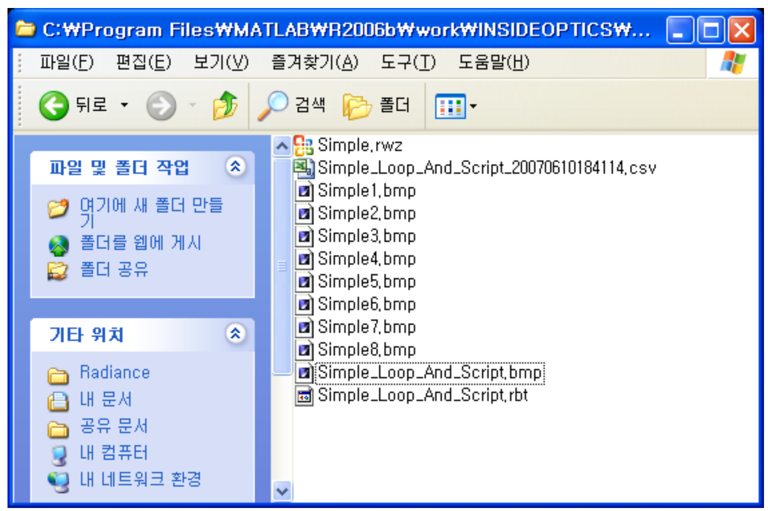


Simple_Loop_And_Script.rbt
// Loop [max loop of var1], [max loop of var2],..., [max loop of varN],
// total loop = num1 * num2 *... * numN
loop 2,2,2,
// image_data all <= make one image together
FILE { "Simple.rwz", image_data all ascend}
단계별 생성 이미지 결과를 하나로
표현된 이미지를 생성
이미지를 붙이는 순서로 순차,
역순차 설정 (ascend, descend)





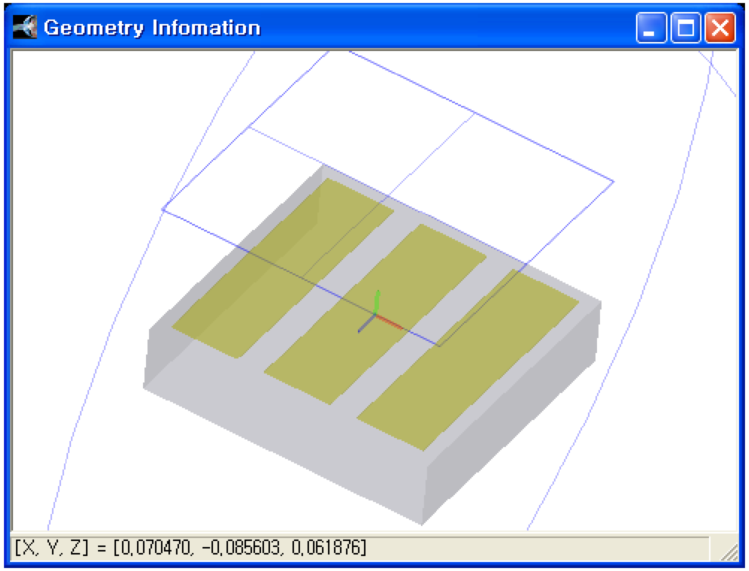
Start Simulation
Stop (Resume) Simulation
Irradiance / Illuminance Display
Radiant / Luminous Intensity Display
Spatial Radiance/Luminance
Angular Radiance/Luminance
Properties
Save Curent View Data
Save Data

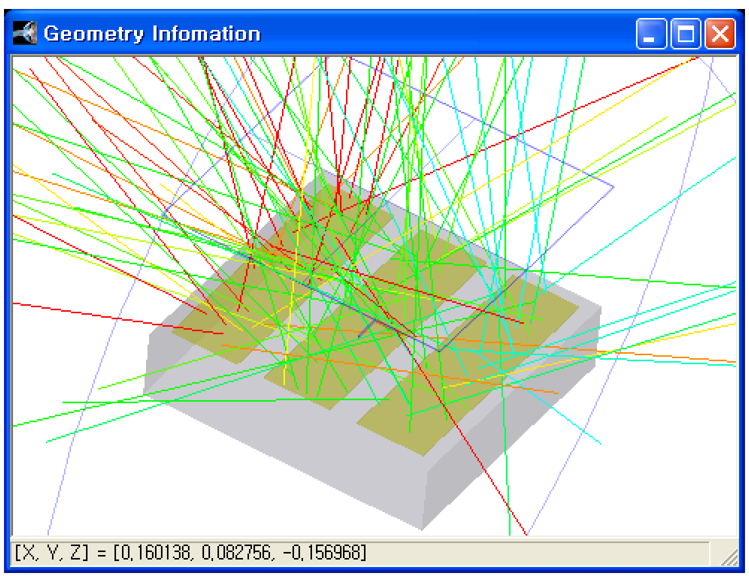


C:₩Program Files₩MATLAB₩R2006b₩work₩INSIDEOPTICS₩...
파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(T) 도움말(H)
뒤로
검색
폴더
파일 및 폴더 작업
여기에 새 폴더 만들기
폴더를 웹에 게시
폴더 공유
기타 위치
Radiance
내 문서
공유 문서
내 컴퓨터
내 네트워크 환경
Simple.rwz
Simple_Loop_And_Script_20070610184114.csv
Simple1.bmp
Simple2.bmp
Simple3.bmp
Simple4.bmp
Simple5.bmp
Simple6.bmp
Simple7.bmp
Simple8.bmp
Simple_Loop_And_Script.bmp
Simple_Loop_And_Script.rbt

Color Mix

Color Mix: Pixel – Lambertian R,G,B (1)



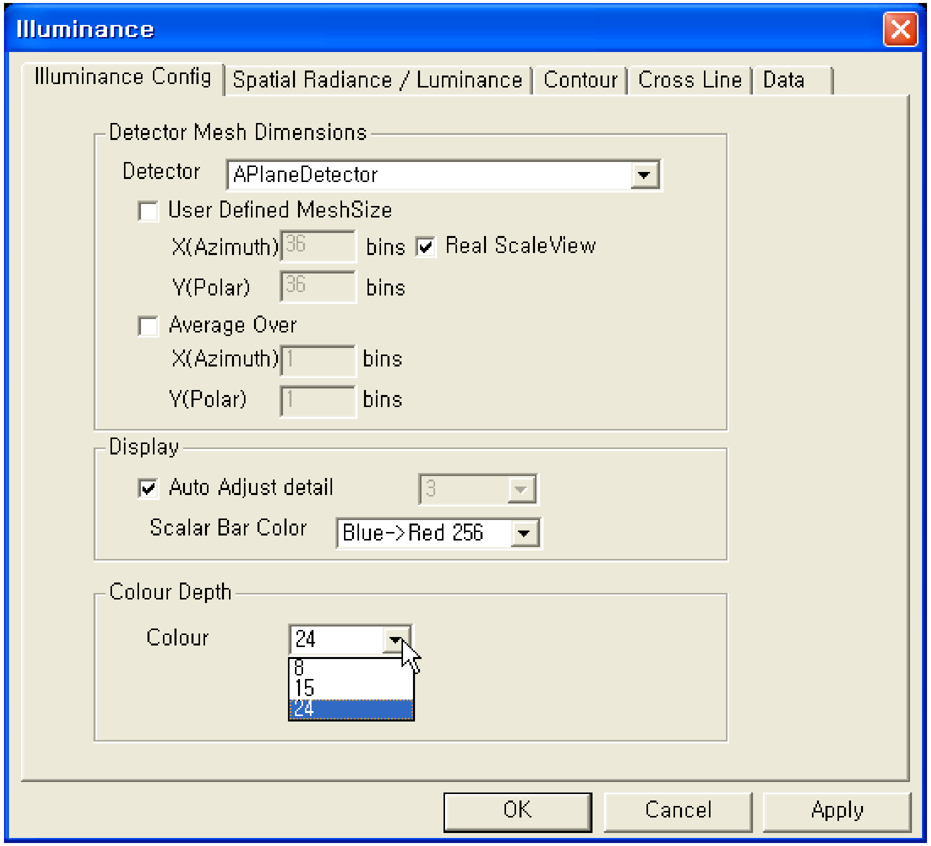


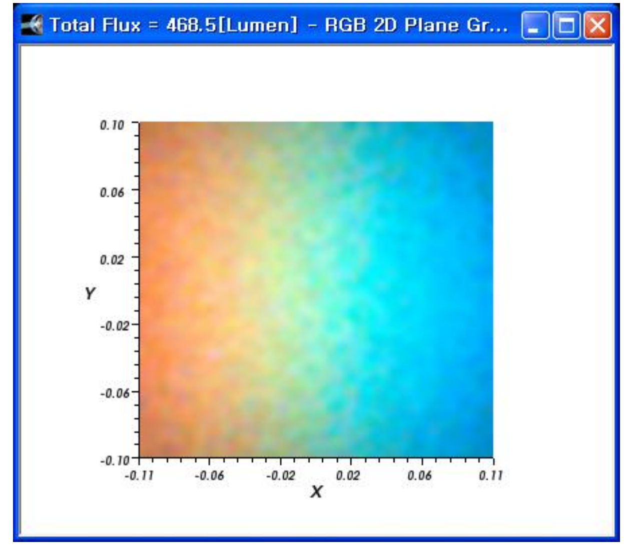
Single_Pixel.rwz

```
#declare TestObject_Flag = 0;
#declare AimAngle_End = 90;
#declare PlaneDetector_Pos = 0.01;
#declare SubPixel_W = 0.050;
#declare SubPixel_H = 0.180;
#declare SubPixel_T = 0.001;
#declare SpaceW = 0.020;
#declare Pixel_W = SubPixel_W*3+SpaceW*3;
#declare Pixel_H = SubPixel_H+SpaceW;
#declare NumW = 1;
#declare NumH = 1;
#declare Display_W = NumW*Pixel_W;
#declare Display_H = NumH*Pixel_H;
#declare Power = 1000;
#declare Ratio_R = 26;
#declare Ratio_G = 60;
#declare Ratio_B = 14;
#declare Power_R = Power/(NumW*NumH)*(Ratio_R/(Ratio_R+Ratio_G+Ratio_B));
#declare Power_G = Power/(NumW*NumH)*(Ratio_G/(Ratio_R+Ratio_G+Ratio_B));
#declare Power_B = Power/(NumW*NumH)*(Ratio_B/(Ratio_R+Ratio_G+Ratio_B));
#declare Substrate_T = 0.05;
```

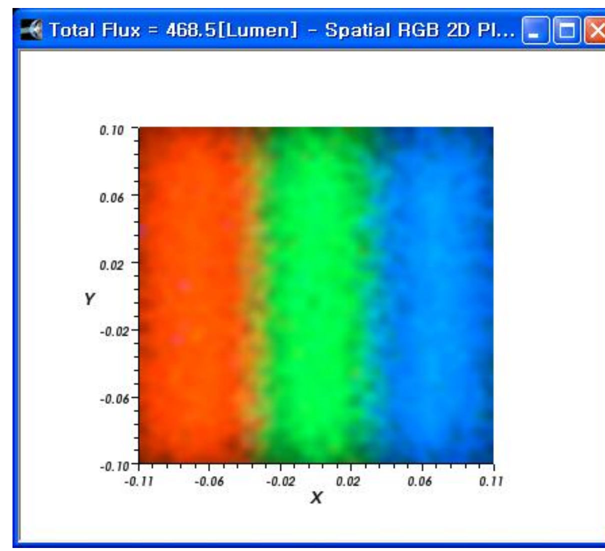
```
global_settings {
    rays {
        trace_off_ratio 0.000001
        total_ray_number 1000000
        max_trace_level 100
        trace_path_view 100
        split_ray off
    }
}
```

Color Mix: Pixel – Lambertian R,G,B (2)– Plane Color



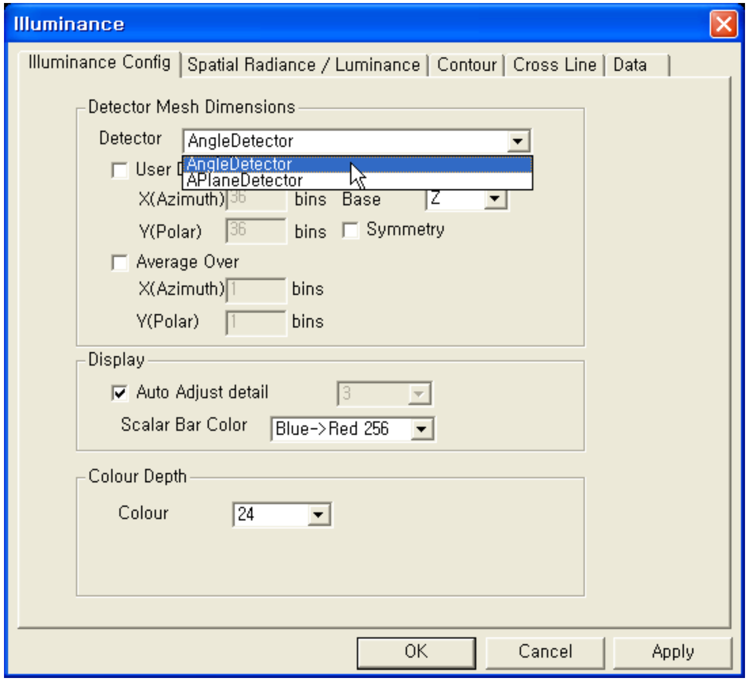


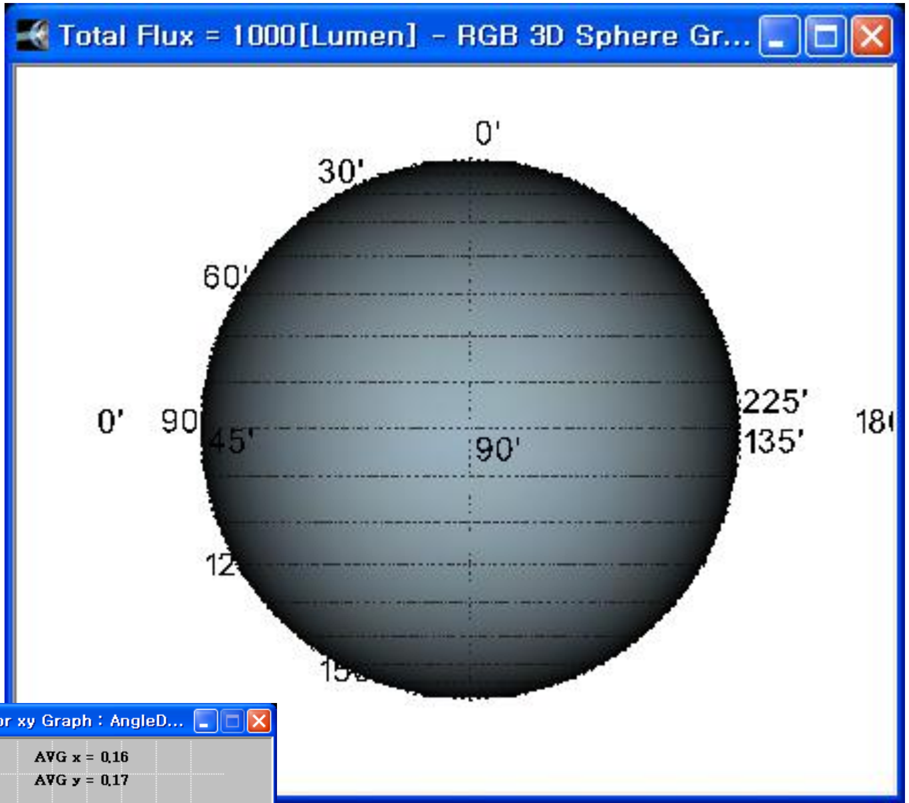
<Illuminance RGB>

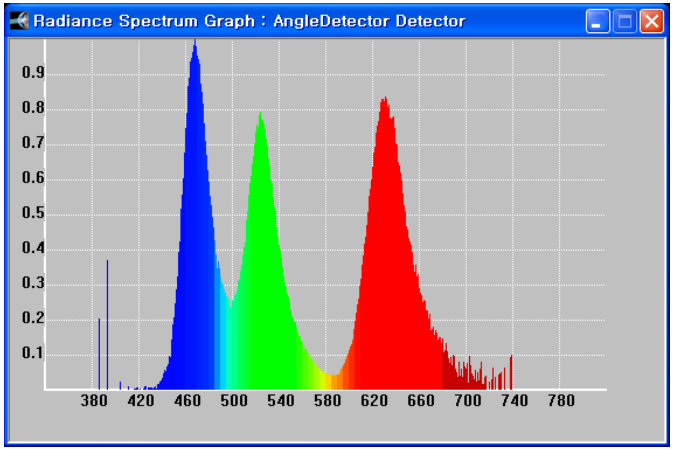


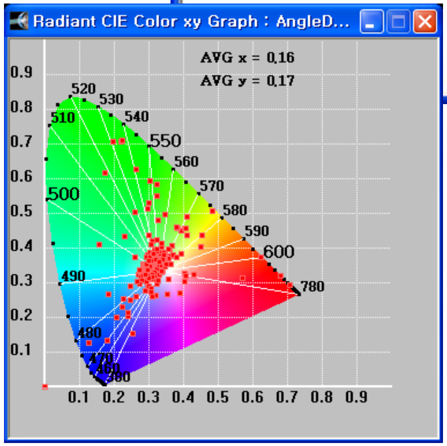
<Luminance RGB>

Color Mix: Pixel – Lambertian R,G,B (3) – Angular Color

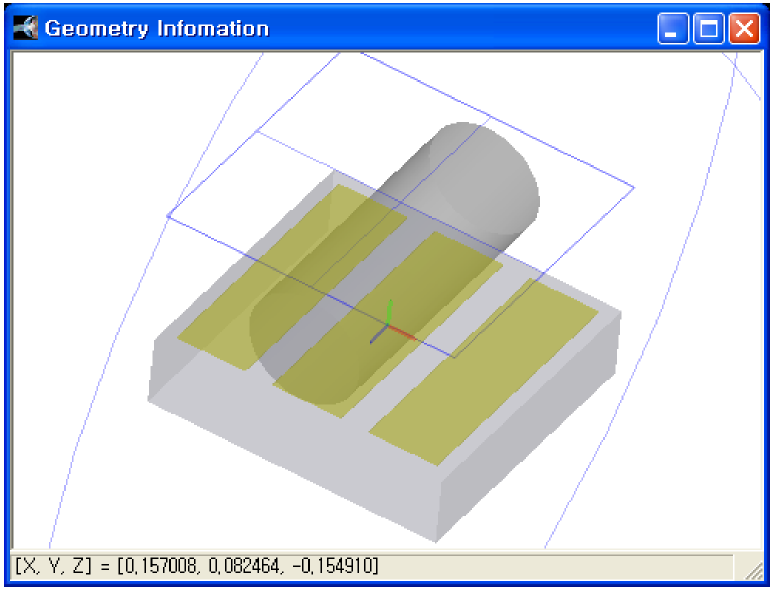


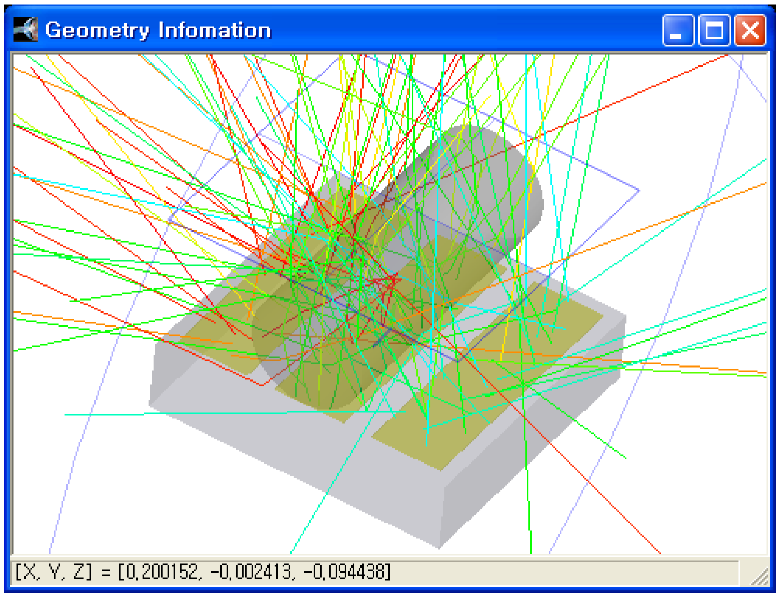






Color Mix: Pixel – Lambertian R,G,B (4)

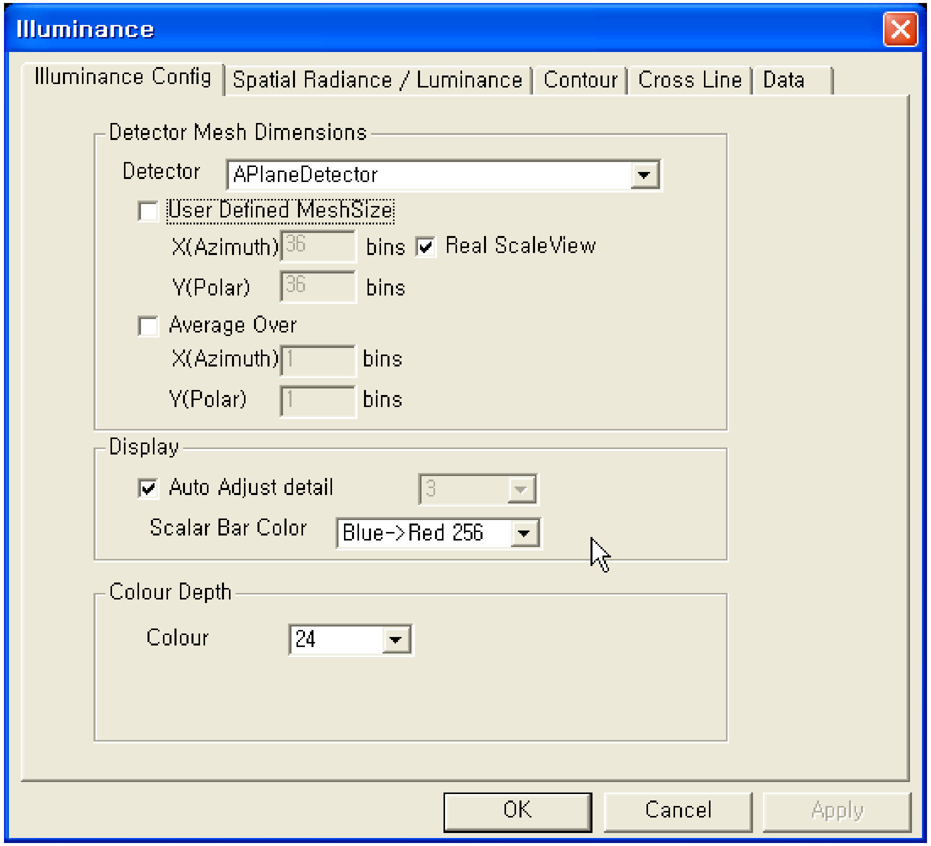


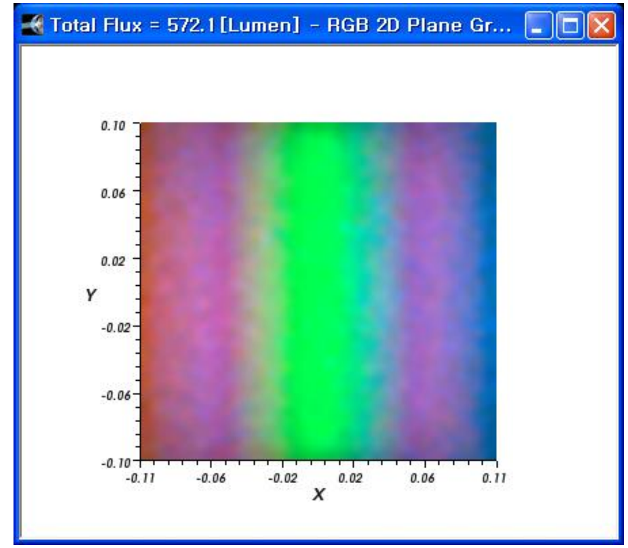


Single_Pixel.rwz

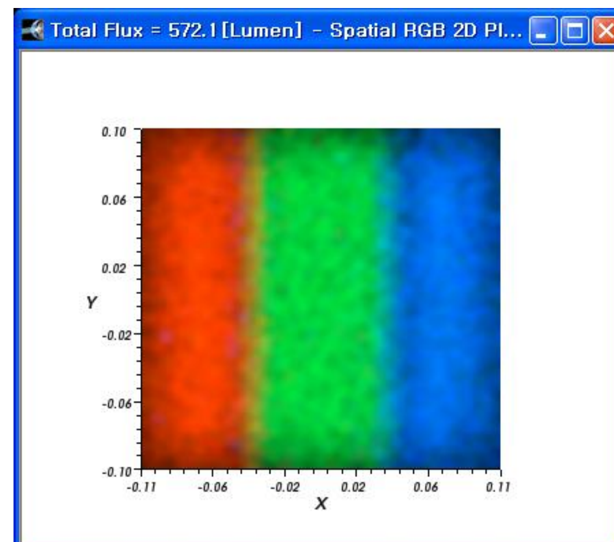
```
#declare TestObject_Flag = 1;
#declare AimAngle_End = 90;
#declare PlaneDetector_Pos = 0.01;
#declare SubPixel_W = 0.050;
#declare SubPixel_H = 0.180;
#declare SubPixel_T = 0.001;
#declare SpaceW = 0.020;
#declare Pixel_W = SubPixel_W*3+SpaceW*3;
#declare Pixel_H = SubPixel_H+SpaceW;
#declare NumW = 1;
#declare NumH = 1;
#declare Display_W = NumW*Pixel_W;
#declare Display_H = NumH*Pixel_H;
#declare Power = 1000;
#declare Ratio_R = 26;
#declare Ratio_G = 60;
#declare Ratio_B = 14;
#declare Power_R = Power/(NumW*NumH)*(Ratio_R/(Ratio_R+Ratio_G+Ratio_B));
#declare Power_G = Power/(NumW*NumH)*(Ratio_G/(Ratio_R+Ratio_G+Ratio_B));
#declare Power_B = Power/(NumW*NumH)*(Ratio_B/(Ratio_R+Ratio_G+Ratio_B));
#declare Substrate_T = 0.05;
```

Color Mix: Pixel – Lambertian R,G,B (4) – Plane Color



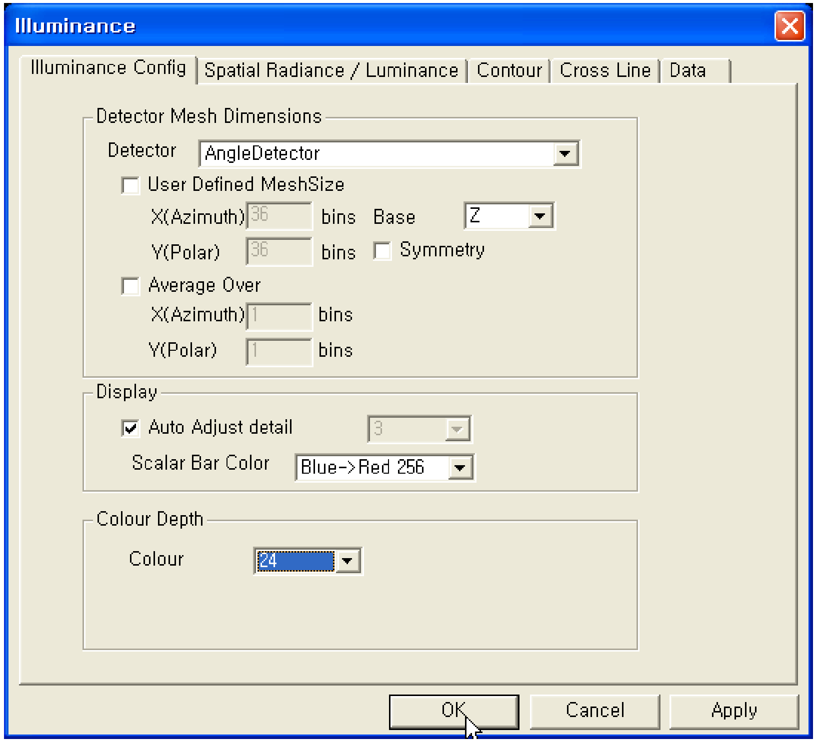


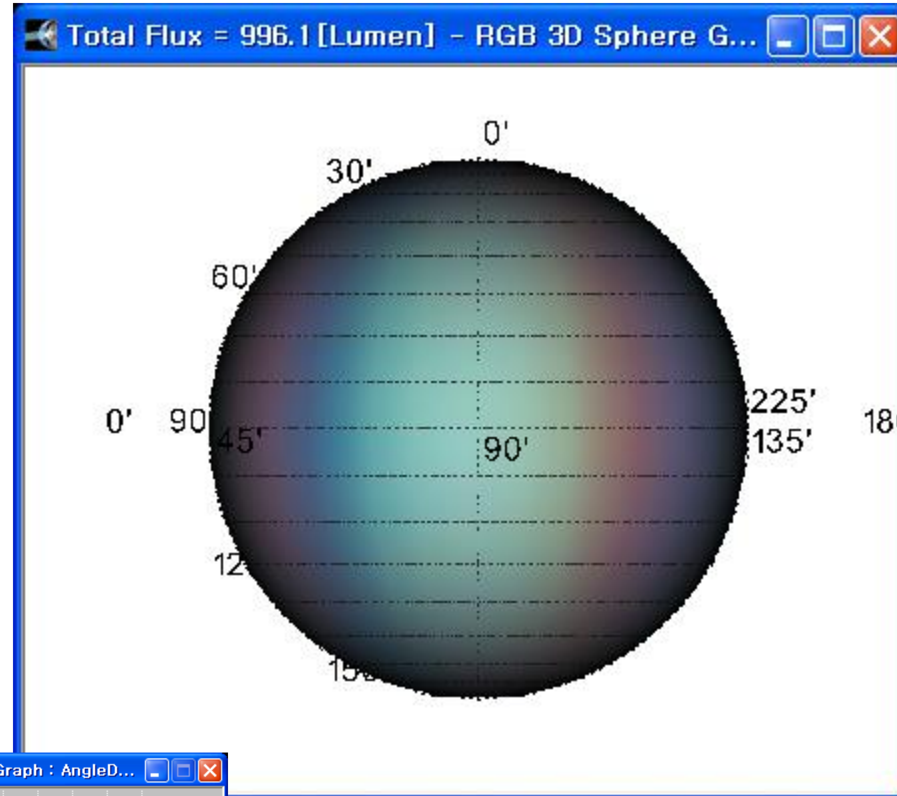
<Illuminance RGB>

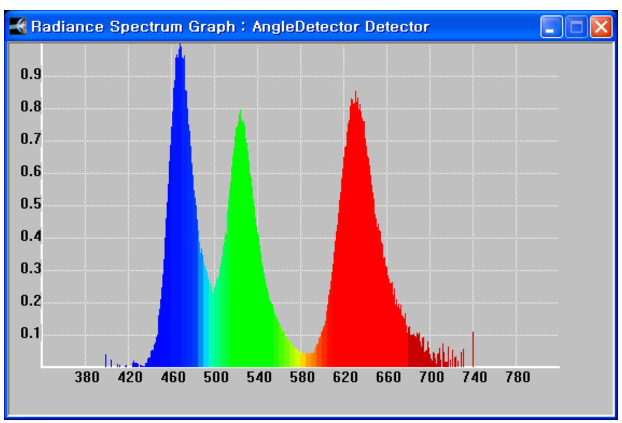


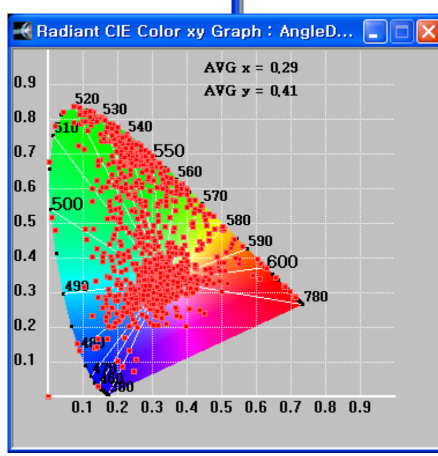
<Luminance RGB>

Color Mix: Pixel – Lambertian R,G,B (5) – Plane Color









Color Mix: Pixel – Lambertian R,G,B (6)

```
#declare SubPixel_R =
light_source
{
    <0, SubPixel_T/2, 0>
    spectrum {
       discrete off
        "RED.spc"
    }
    object_light
    {
        surface
        box
        {
            <-SubPixel_W/2, 0, -SubPixel_H/2>, <SubPixel_W/2, SubPixel_T, SubPixel_H/2>
        }
        emittance
        {
            front_surface
            {
                off
            }
            back_surface
            {
                off
            }

            ...

    emittance
    {
        photometric Power_B
        measured_whole off
        aim_sphere <0, AimAngle_End>
    }
    point_at <0, 1, 0>
}
```

Color Mix: Pixel – Lambertian R,G,B (7)

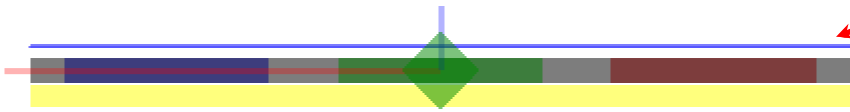
```
#macro ArraySubPixel_R()
#local Count_i=0;
#while (Count_i < NumW)
    #local Count_j=0;
    #while (Count_j < NumH)
        object{
               SubPixel_R
               translate <-Display_W/2+SpaceW/2+SubPixel_W/2+Pixel_W*Count_i,0,0>
               translate <0,0,-Display_H/2+SpaceW/2+SubPixel_H/2+Pixel_H*Count_j>
              }
        #local Count_j=Count_j+1;
    #end
    #local Count_i=Count_i+1;
#end
#end
```

```
Al_Sub
ArraySubPixel_R()
ArraySubPixel_G()
ArraySubPixel_B()
```

```
#declare TestObject =
cylinder
{
    <0, 0, -0.1>, <0, 0, 0.1>, 0.04
    spectrum {
        discrete off
        standard_e
    }
    ior 1.5
    surface {
        both {
            probabilistic_split on
        }
    }
    translate<0,0.05,0>
}

#if(TestObject_Flag)
    TestObject
#else
#end
```

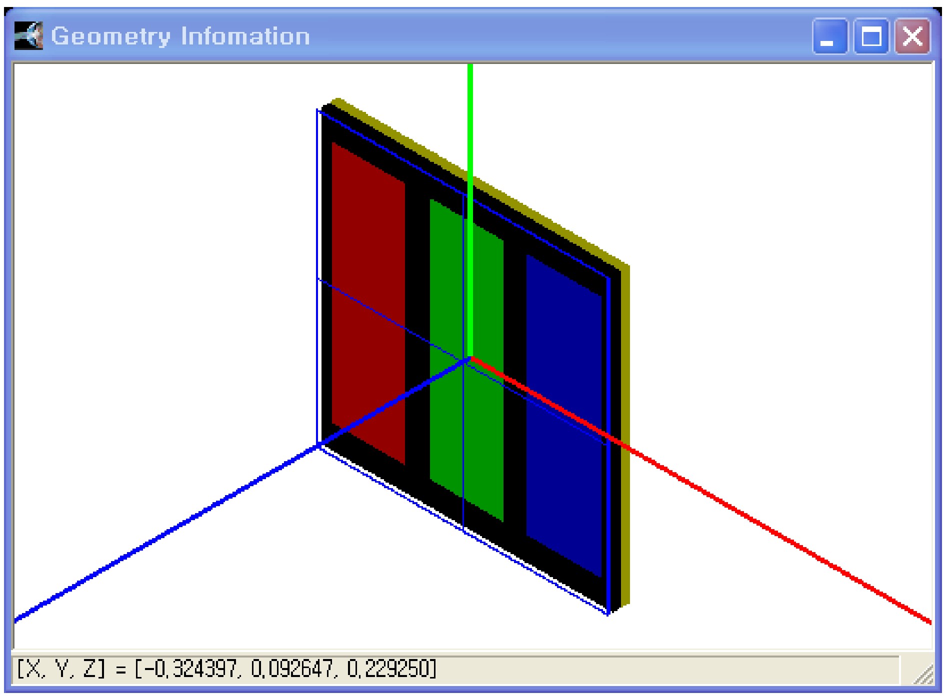
Color Filter

Color Mix: Geometry



Plane Detector :
필요에 따라 위치를 바꿔서 테스트

Lambertian White Light



Color Mix: Script (1)

```
#declare Pixel_W=0.360;
#declare Pixel_H=0.360;
#declare Pixel_T=0.010;
#declare SubPixel_W=0.090;
#declare SubPixel_H=0.300;
#declare SubPixel_T=0.011;
#declare Space_W=(Pixel_W-SubPixel_W*3)/3;
#declare Space_H=(Pixel_H-SubPixel_H)/2;
```

Pixel 과 SubPixel의 치수 정의

```
    emittance
    {
        radiometric 1
        measured_whole on
        aim_sphere <0, 90>
    }
    point_at <0, 0, 3>
}
```

Emitting 각도제한

Color Mix: Script (2)

```
#declare Pixel_W=0.360;
#declare Pixel_H=0.360;
#declare Pixel_T=0.010;
#declare SubPixel_W=0.090;
#declare SubPixel_H=0.300;
#declare SubPixel_T=0.011;
#declare Space_W=(Pixel_W-SubPixel_W*3)/3;
#declare Space_H=(Pixel_H-SubPixel_H)/2;
```

Pixel 과 SubPixel의 치수 정의

```
    spectrum {
        discrete off
        standard_d65
    }
```

표준광원 **D65 white**

```
    emittance
    {
        radiometric 1
        measured_whole on
        aim_sphere <0, 90>
    }
    point_at <0, 0, 3>
}
```

Emitting 각도제한

Color Mix: Script (3)

```
#declare SubPixel_R =
box
{
    <-SubPixel_W/2, -SubPixel_H/2, -SubPixel_T/2>,
    color <1, 0, 0>
    ior 1.5
    surface {
        both {
            transmittance{1}
            reflectance{0}
        }
        front_surface{
          in{
              coating{ "Red_Filter.coa" }
          }
        }
    }
    translate<-SubPixel_W-Space_W,0,0>
}
```

Surface Spectrum 파일

Color Mix: Coating File Format

Red_Filter.coa

```
9          17
0
375        0          0.1945
400        0          0.0688
425        0          0
450        0          0
475        0          0
500        0          0
525        0          0
550        0          0
575        0          0
600        0          0.5924
625        0          0.9073
650        0          0.8856
675        0          0.9514
700        0          0.9065
725        0          0.9364
750        0          0.9491
775        0          0.9245
10
375        0          0.1945
400        0          0.0688
425        0          0
450        0          0
475        0          0
500        0          0
525        0          0
550        0          0
575        0          0
600        0          0.5924
625        0          0.9073
650        0          0.8856
675        0          0.9514
700        0          0.9065
725        0          0.9364
750        0          0.9491
775        0          0.9245
```

입사각도 갯수, 파장 갯수

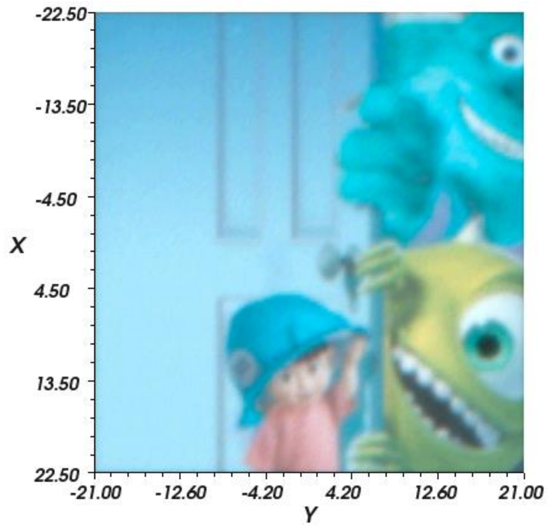
입사각도 **0**도

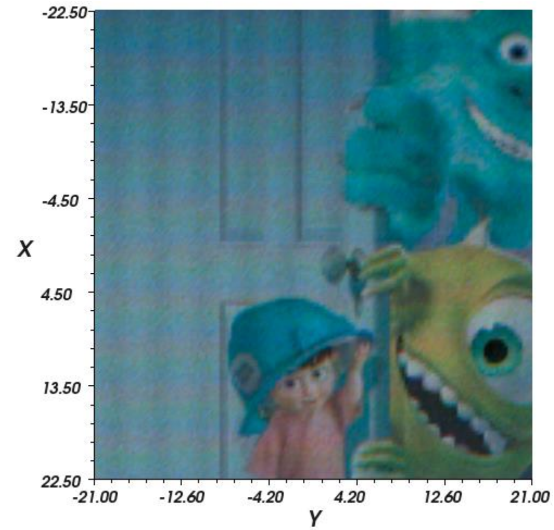
파장, 반사, 투과

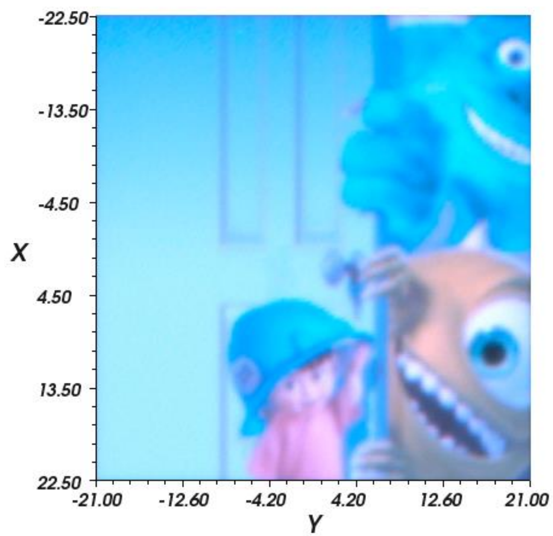
입사각도 **10**도

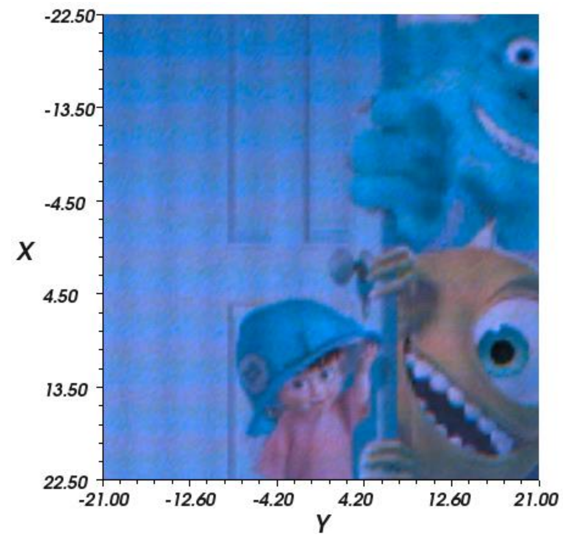
Display Image Simulation

Display Simulation: [Open 50%, R:G:B=2:4:1, 1:1:1]

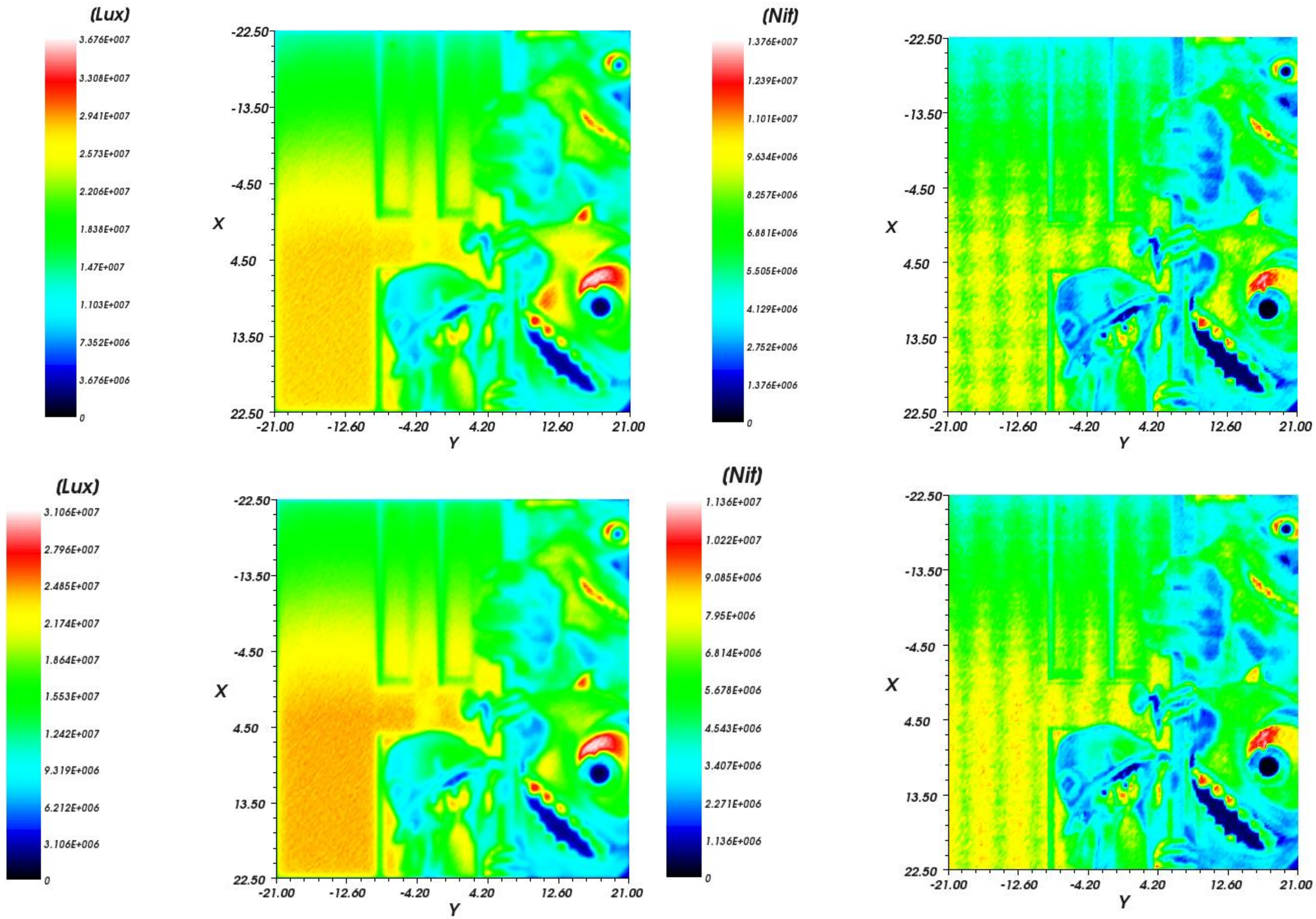






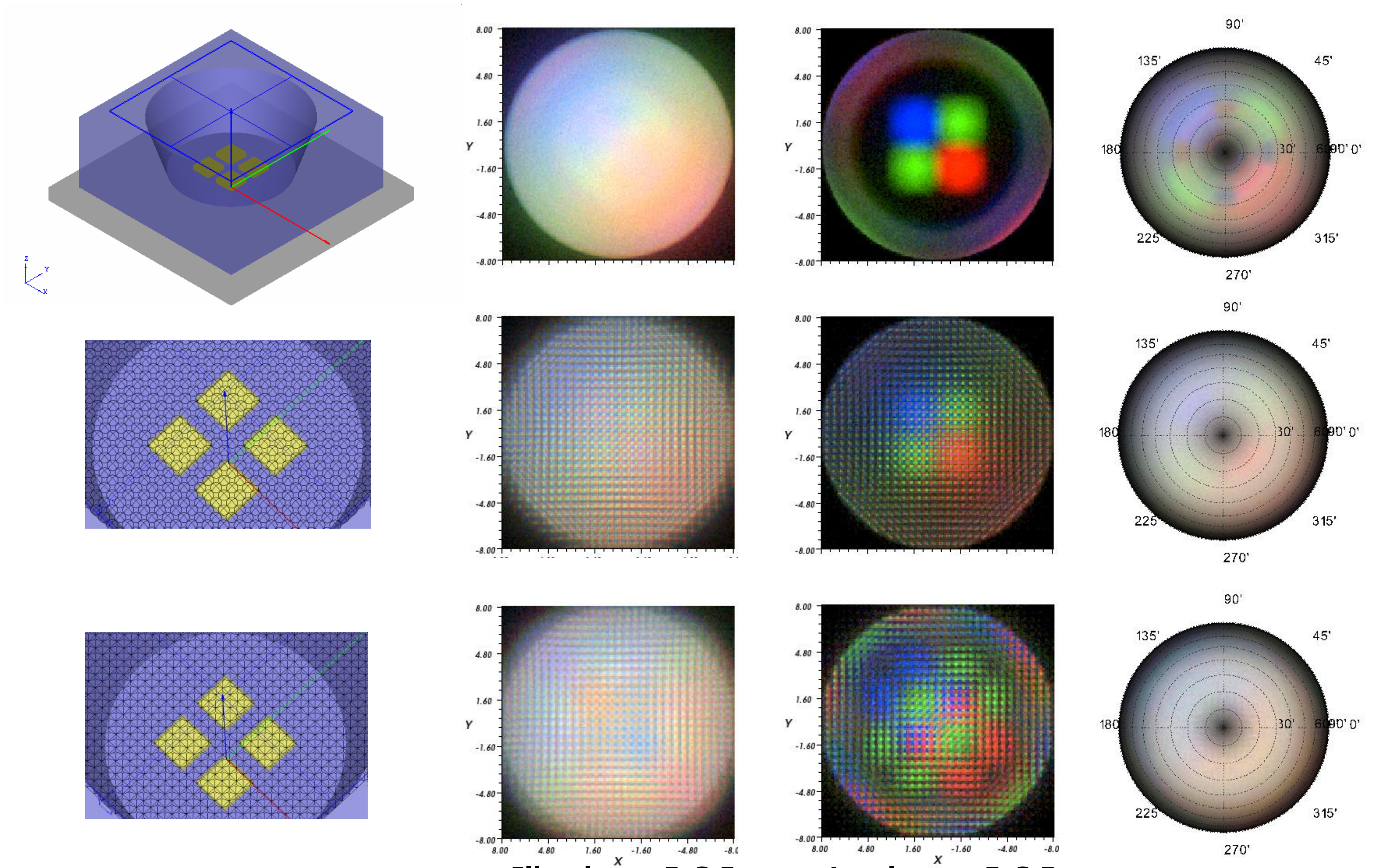


Display Simulation: [Open 50%, R:G:B=2:4:1, 1:1:1]



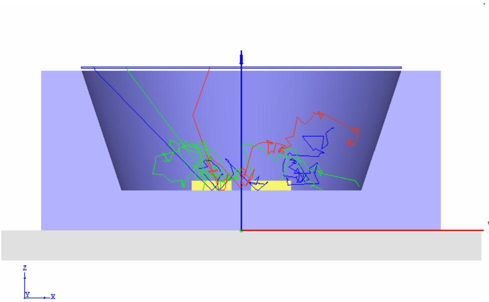
LED Packaging

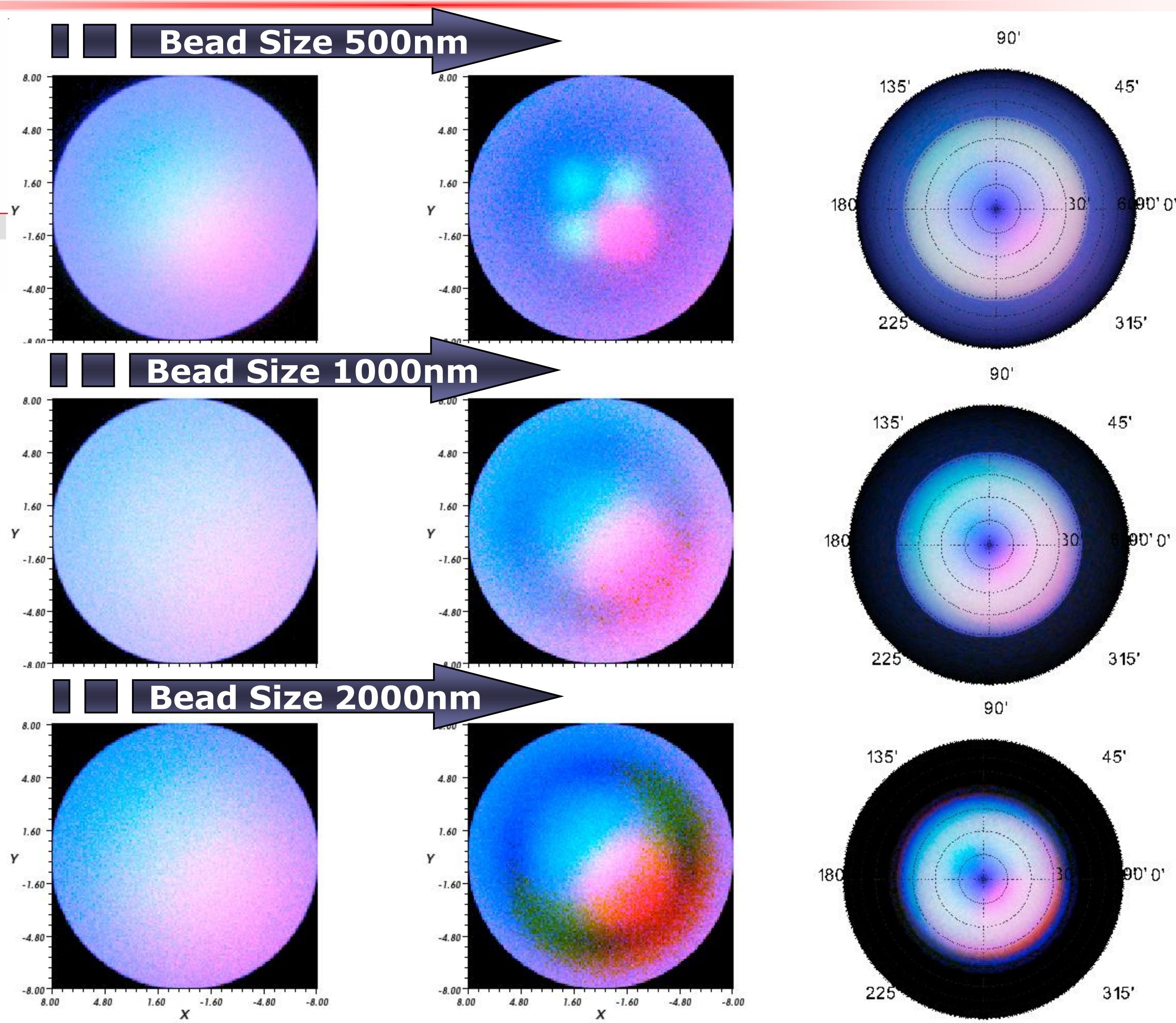
LED BLU Design: Emitting Unit Design (2)



<Illuminace R,G,B> <Luminance R,G,B>

LED BLU Design: Emitting Unit Design (3)





<Illuminace R,G,B>   <Luminance R,G,B>