

# Visual Basic.net을 이용한 구면수차 시뮬레이션 교육 프로그램 개발

## 서정의

대구보건대학 안경광학과  
(2007년 8월 10일 받음, 2007년 9월 1일 수정본 받음)

본 논문에서는 구면과 비구면에서 발생하는 구면수차를 광선추적에 의해서 시뮬레이션 하고 그 수차량을 계산해 낼 수 있는 프로그램을 개발하였다. 이를 통하여 비구면인 타원, 포물선, 쌍곡선면을 광선추적 하였을 때 발생하는 구면수차의 변화를 확인할 수 있었고 이를 구면과 비교함으로써 비구면렌즈 설계의 원리와 필요성을 이해할 수 있었다. 또한 포물선 거울면에 대한 광선추적 프로그램을 개발하여 시뮬레이션에 대한 정확도를 확인 할 수 있도록 하였다.

주제어: 비구면, 시뮬레이션, 구면수차

## 서 론

렌즈에서 발생하는 구면수차는 렌즈의 특성을 좌우하는 중요한 요소로 작용하고 있다. 최근 이러한 구면수차를 제거하기 위해 비구면으로 렌즈를 설계하여 구면수차를 최소화 시킨 렌즈가 널리 사용되고 있다. 본 논문에서는 렌즈 구면의 곡률반경의 변화를 알아보고자 한다. 구면수차량을 구하는 것은 복잡한 광학식을 이용하여야 하므로 곡률반경에 따른 수차량의 변화와 비구면의 정도에 따른 수차량을 구하여 비교해 보는 것은 매우 힘든 일이며 많은 시간을 요하는 어려운 작업이다. 더욱이 현재 비구면 렌즈의 수요가 많이 늘어나고 상황에서 그 중요성은 높아지고 있다. 안경사를 준비하는 학생이 이러한 비구면 렌즈에 대한 이해를 돕기 위해서 Visual Basic.net을 이용하여 구면수차를 재현하는 프로그램을 개발하였다.

## 본 문

### 1. 비구면의 구현

비구면은 타원면, 포물면, 쌍곡선면으로 나뉜다. 이것은 이심률(eccentricity)에 의해서 분류할 수 있다<sup>[1]</sup>. 그림 1에 서와 같이 (x, y)가 위치하였다면 다음과 같은 식으로 표현 된다.

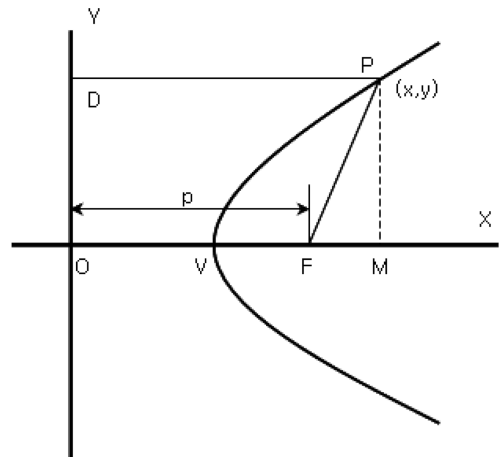


Fig. 1. Asphere and eccentricity.

$$e(\text{eccentricity}) = \frac{\sqrt{(x+p)^2 + y^2}}{x} \quad (1)$$

e값에 따라서 분류하면 다음과 같다.

- e = 0 : circle
- e < 1 : ellipse
- e = 1 : parabola
- e > 1 : hyperbola

포물선의 경우인 e=1을 식 (1)에 대입하여 정리하면

$$y^2 = 2xp - p^2 \quad (2)$$

주 저자 연락처: 서정의, 702-722 대구광역시 북구 태전동 산 7번지 대구보건대학 안경광학과  
TEL: 053-320-1381, E-mail: mk\_lucifer@lycos.co.kr

```

a = CSng(Me.Inpt_a.Text)
b = CSng(Me.Inpt_b.Text)
n = CSng(Me.Inpt_n.Text)
x0 = 200 + a
y0 = 150
yold = 109
xold = -Sqrt(a * a - (a * a + yold + yold) / (b * b))
'렌즈면 그리기
For ynew = 109 To -109 Step -1
    xnew = -Sqrt(Abs(a * a - (a * a + ynew + ynew) / (b * b)))
    g.DrawLine(Pens.Black, xold + x0, yold + y0, xnew + x0, ynew + y0)
    xold = xnew
    yold = ynew
Next
us = 0.002
xx = 0.01
'x = -x0
ys = 105
For ys = 105 To 0 Step -30

```

Fig. 2. Programing of asphere.

으로 표현되고, 여기에  $p^2$ 을 소거하기 위해 포물면이  $p/2$ 만큼  $x$ 축으로 이동한 값인  $x=x-p/2$ 를 대입하여 정리하면 다음과 같이 된다.

$$y^2=2px=4fx \tag{3}$$

$$y^2-4fx=0 \tag{4}$$

과 같이 되고, 초점(F)는  $p/2$ 에 위치하게 된다.

같은 방법으로 정리하면 타원면은

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \tag{5}$$

쌍곡선면은

$$\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1 \tag{6}$$

로 정리된다. 이 결과를 Visula basic.net으로 구현하기 위한 코드는 그림 2와 같이 표현된다<sup>[2]</sup>.

### 2. 비구면에서 광선추적

광선추적을 위해서는 무한 원방에서 출발한 광선의 렌즈면까지 이동거리 계산, 렌즈면에서 입사각과 출사각의 계산, 렌즈 내에서의 광선의 이동에 관한 계산이 필요하

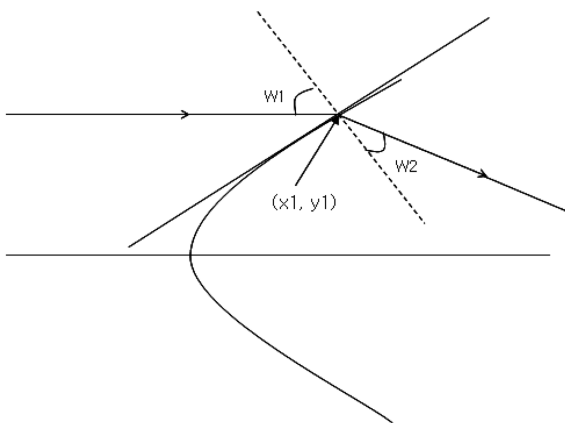


Fig. 3. Incidence and refraction angle on asphere.

다. 먼저 렌즈면까지 광선의 이동에 대한 계산은 광선의 높이를 결정하는  $y$ 축 값과 광선의 진행을 결정하는  $x$ 축 값으로 나뉜다. 무한 원방에서 출발하는 평행광선이므로  $y$ 값은 고정값이 되고,  $x$ 값이 변함에 따라 광선이 렌즈면에 도착했는지 판단 할 수 있다<sup>[3]</sup>. 따라서 각  $x$ 와  $y$ 값을 식 (4), 식 (5), 식 (6)에 대입하여 등식이 성립할 때가 광선이 렌즈면에 도착 했을 때이다. 이를 코드화 시키면 그림 3과 같은 결과가 된다.

렌즈면에서의 입사각과 출사각을 계산하기 위해서는 snell의 공식을 이용하여 계산할 수 있다<sup>[4]</sup>. 구면 위의 한 점인  $(x_1, y_1)$ 에서의 접선 방정식의 기울기는

$$a = \frac{dx}{dy} = -\frac{b^2x}{a^2y} \tag{7}$$

이고, 법선과 입사광선과의 각인 입사각은

$$w_1 = \frac{\pi}{2} - \tan^{-1}\left(-\frac{b^2x}{a^2y}\right) \tag{8}$$

이 된다.

snell의 법칙을 이용해서 출사각을 계산하면

$$w_2 = \sin^{-1}\frac{w_1}{n} \tag{9}$$

이 된다. 포물선, 쌍곡선에서도 같은 방법으로 계산하게 되면 출사각을 얻을 수 있다. 이를 Visual Basic.net으로 코드화 하면 그림 4와 같이 나타난다<sup>[2]</sup>.

### 3. 포물선 거울면에서 광선추적

포물선 거울면에서의 광선추적은 지금까지 구현한 시뮬레이션 프로그램의 정확성을 검증하기 위한 단계로서, 평행하게 입사한 모든 빛은 초점에 모인다는 원리를 이용한 것이다<sup>[1]</sup>. 이를 위해서 광선추적의 방식은 빛의 진행방향

```

'렌즈면까지 광선의 이동
yup = ys
x = -x0
Do While ((x * x) / (a * a) + (yup + yup) / (b * b)) > 1
    x = x + ds 'x축으로 ds만큼씩 이동
    'MessageBox.Show(x & ", " & yup)

    g.DrawLine(Pens.Red, x + x0, 150 - yup, x + xx + x0, 150 - yup + xx)
    g.DrawLine(Pens.Red, x + x0, 150 + yup, x + xx + x0, 150 + yup + xx)
Loop

'렌즈면에 입사각과 출사각 계산
Dim w1, w2, w3 As Single
xs = Sqrt((a * a) - (a * a + ys + ys) / (b * b))
'입사각 계산
w1 = (90 * Pi) / 180 - Atan((b * b + x) / (a * a + yup))
'출사각 계산
w2 = Asin((Sin(w1)) / n)
'출사각을 수평각으로 전환
w3 = w1 - w2

'x,y 좌표의 이동량 계산
Dim dx, dy As Single
dx = ds
dy = dx * Tan(w3)

```

Fig. 4. Programing of incidence and refraction angle.

이 반대이고, 공기에서 공기로 진행하는 의미를 가지는 굴절률 '-1'로 고정한 것 이외에는 모두 동일하게 적용하여 구현하였다.

**결과 및 고찰**

**1. 포물선 거울면에서의 광선추적 결과**

포물선 거울면에서의 광선추적의 결과는 평행하게 입사한 광선이 모두 초점을 향해서 모여들어야 한다. 이는 앞에서 언급한 것과 같이 비구면 렌즈에서의 광선추적의 정확성을 검증하기 위한 단계이다. 결과는 그림 6에서 보는 것과 같이 모든 광선이 초점에 모여들었다.

```

'렌즈면에 입사각과 출사각 계산
Dim w1, w2, w3 As Single
w1 = (90 * PI) / 180 - Atan((4 * p) / (2 * yup))
w2 = Asin(Sin(w1))
w3 = w1 + w2

Dim dx, dy, dss As Single
dss = -0.001
If w3 > (PI / 2) Then
    dss = -dss
End If
dx = dss
dy = -dx + Tan(w3)
If dy > 0 Then
    dy = -dy
End If
    
```

Fig. 5. Programing of parabolic mirror.

**2. 구면 및 비구면에서의 광선추적 결과**

곡률반경을 150 mm로 했을 때 그림 7에서 보는 것과 같이 가장 바깥쪽에서 입사한 광선과 안쪽에 입사한 광선의 초점거리 차이는 -13.30072 mm이며, 이를 굴절력으로 계산해보면 -0.16598D로 나타났다.

a=201.3 mm, b=150 mm로 했을 때의 그림 8에서 보는 것과 같이 타원에서는 초점거리 차이는 0.01089 mm이며, 굴절력 차이는 0.00015D로 나타났다. 이는 주변부의 곡률을 약간 작게 해주는 타원에 의한 결과로서 구면수차가 거의 줄어드는 결과를 보여주고, 이를 시뮬레이션 해 봄으로써 학생들은 비구면렌즈의 설계에 대한 원리를 직접 경험해 볼 수 있는 기회를 제공하여 교육적인 효과가 클 것이라 예상된다.

Fig. 9와 Fig. 10은 쌍곡선면과 포물선면에서의 광선추적의 결과를 보여준다.

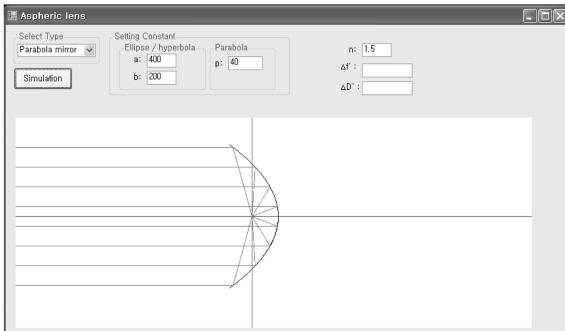


Fig. 6. Ray tracing on parabolic mirror.

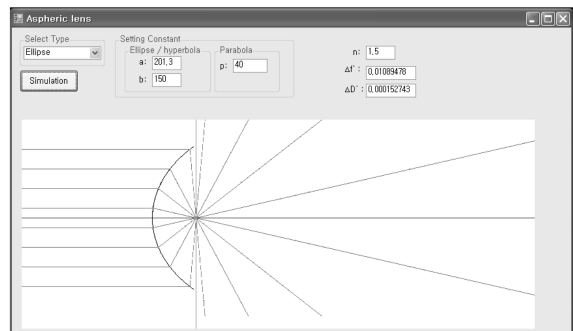


Fig. 8. Ray tracing on ellipsoid.

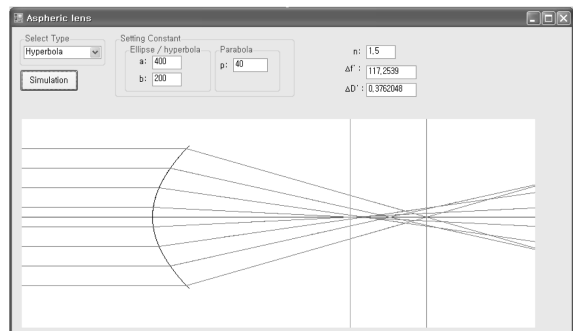


Fig. 9. Ray tracing on hyperboloid.

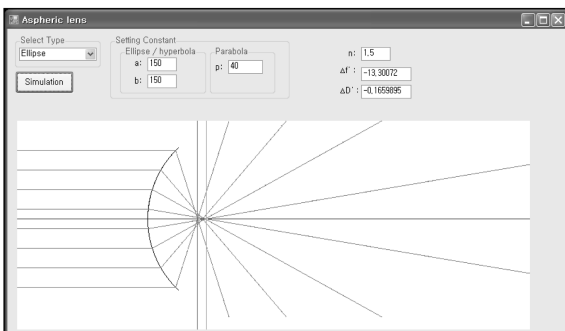


Fig. 7. Ray tracing on sphere.

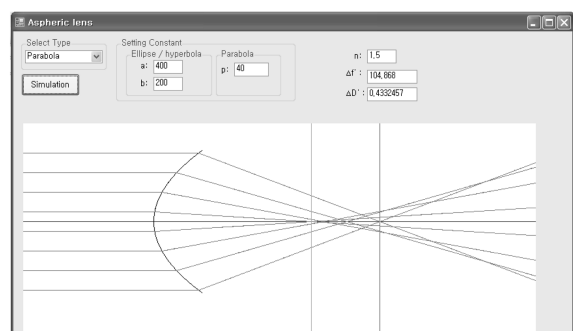


Fig. 10. Ray tracing on paraboloid.

쌍곡선과 포물선에 의한 결과로 학생들은 주변부를 타원보다 곡률을 더 작게 했을 때의 구면수차의 반대 방향으로 더 크게 나타남을 알 수 있다.

## 결 론

포물선 거울면에서 광선추적을 통해서 프로그램의 신뢰성을 확보하였고, 구면에서 발생하는 구면수차량을 확인하였다. 구면을 사용했을 때 발생했던  $-0.16598D$ 의 구면수차가 타원면을 사용함으로써  $0.00015D$ 로 감소함을 확인 할 수 있었다. 이를 통해서 학생들이 비구면의 종류에

따른 구면수차량의 변화를 알 수 있었고, 비구면 렌즈 설계에 대해 이해를 도울 수 있다.

## 참고문헌

- [1] Bijana Tadic, "Optical System Design", SPIE, U.S.A.. pp. 87-89(2000).
- [2] Bill Evjen, "Visual Basic.net Bible", Hungry Minds Inc., U.S.A., pp. 640-647(2001).
- [3] D. Malacara and Z. Maracara, "Handbook of Lens Design", Marcel Dekker Inc., U.S.A., pp. 341-350(1994).
- [4] 조재홍, "광학", 두성사, 서울, pp. 128(2002).

# Development of Spherical Aberration Simulation Program for Education using Visual Basic.net

Jeung-Ik Seo

Department of Ophthalmic Optics, Daegu Health College

(Received August 10, 2007: Revised manuscript received September 1, 2007)

In this study, a program has been developed to enable a simulation of spherical aberration - originated from a spherical surface and an aspheric surface - via ray tracing and calculation of the degree of spherical aberration. This process has been verified for the variation of spherical aberration generated from ray tracing for aspheric surfaces such as an oval, a parabola, and a hyperbola. By comparing it with the spherical surface, the principle and necessity of the aspheric surface lens design have been understood. In addition, it allows confirming the accuracy of the simulation by developing a ray tracing program in relation to the mirror plane of a parabola.

**Key words:** asphere, simulation, spherical aberration