Ray tracing

N. Fressengeas

Laboratoire Matériaux Optiques, Photonique et Systèmes
Unité de Recherche commune à l'Université de Lorraine et à CentraleSupélec

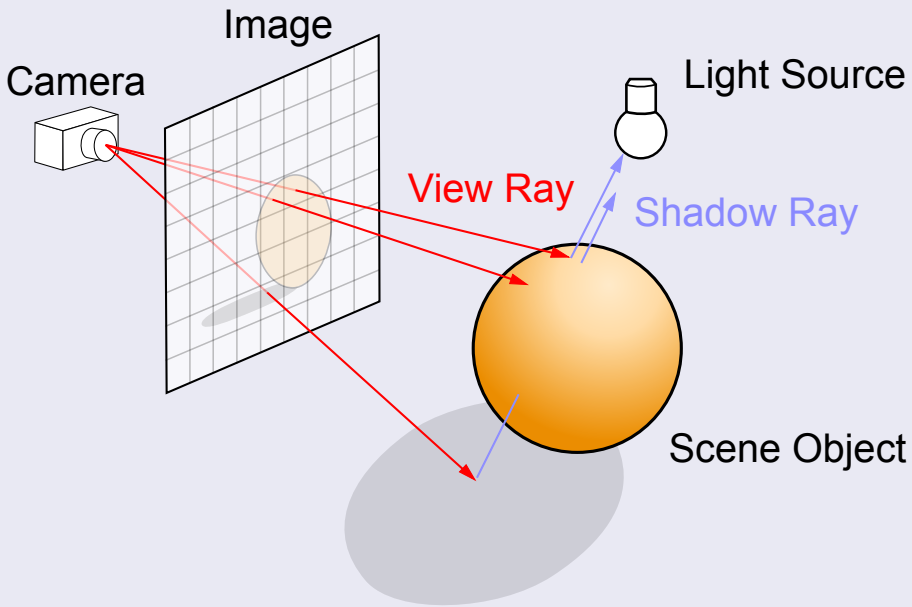
Teasing...

Ray tracing : the past...and the future ?

https://youtu.be/7VrPjVSPmKw

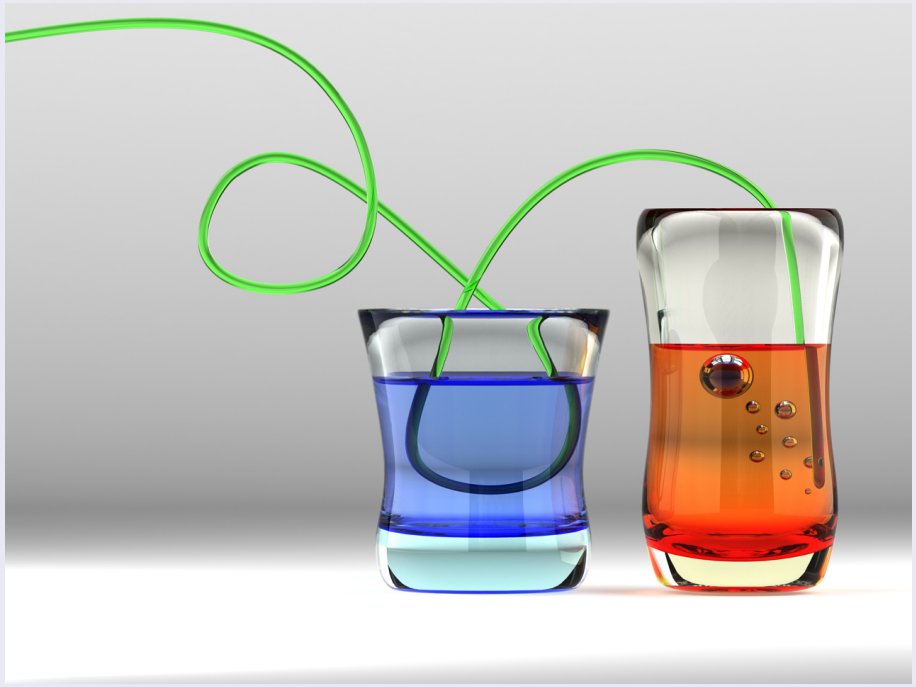
Light rays for rendering

The ray tracing algorithm builds an image by extending rays into a scene.



by Enrik on Wikimedia Commons

Ray tracing



by Mehran Moghtadai on Wikimedia Commons

Ray tracing algorithm
Lauching rays backwards

Algorithm

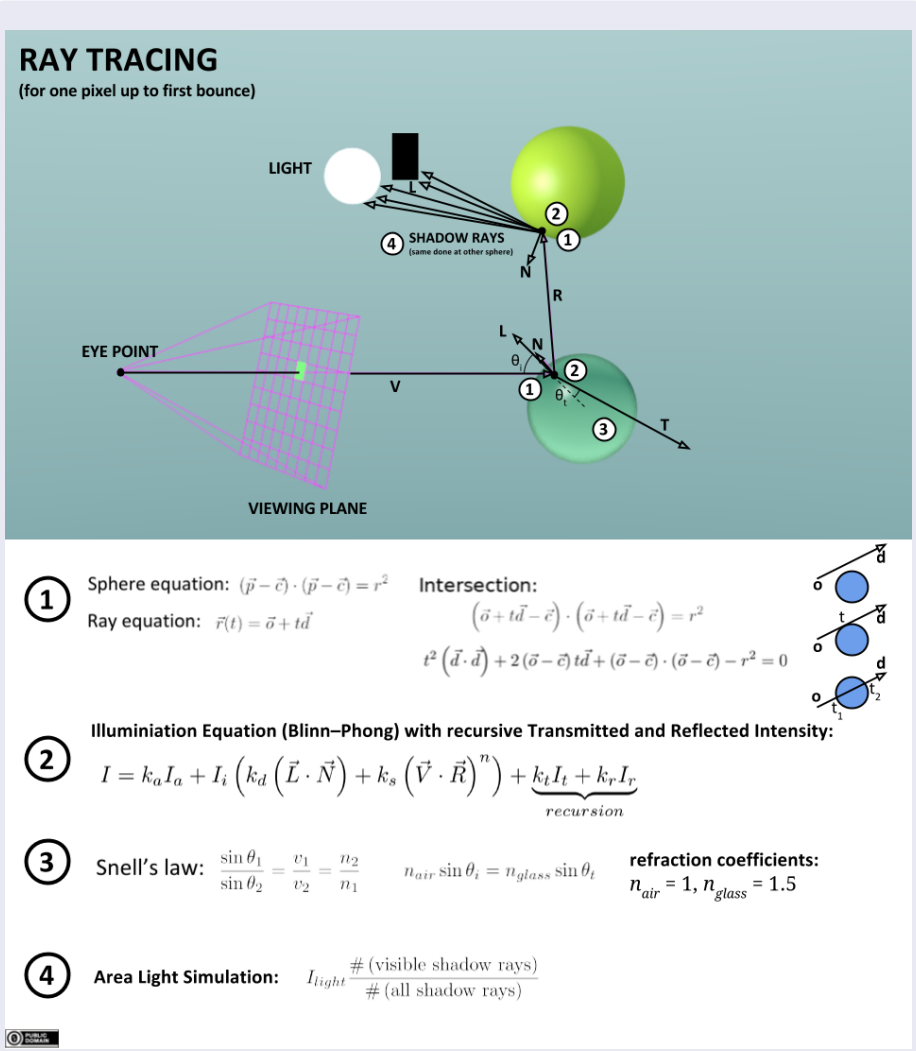
1. Launch ray from eye
2. Identify pixel in image plane
3. Identify luminosity and color from the ray path
4. Be recursive on the reflections

Are rays traveling backwards ?

No, just a computational trick

Ray tracing advantages

- Realistic, especially for shadows
- Suitable for parallel computing



Ray tracing algorithm
Lauching rays backwards

Algorithm

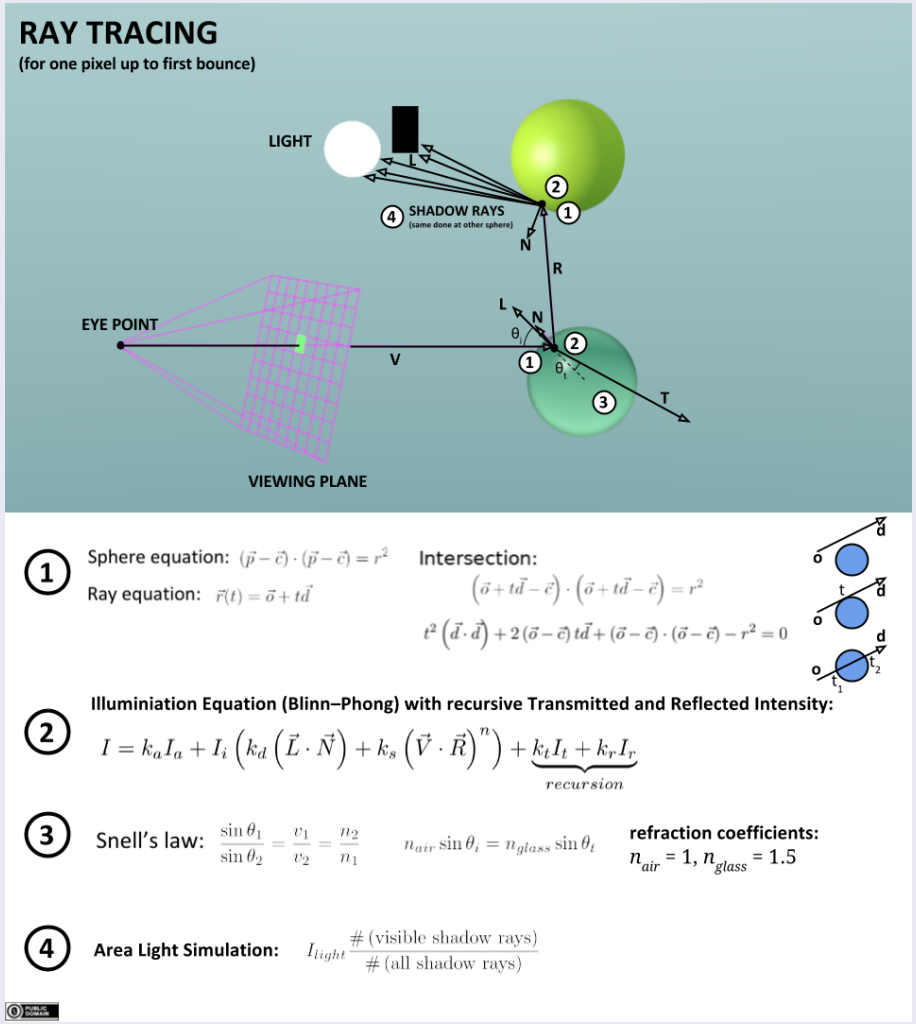
1. Launch ray from eye
2. Identify pixel in image plane
3. Identify luminosity and color from the ray path
4. Be recursive on the reflections

Are rays traveling backwards ?

No, just a computational trick

Ray tracing advantages

- Realistic, especially for shadows
- Suitable for parallel computing



Ray tracing algorithm
Lauching rays backwards

Algorithm

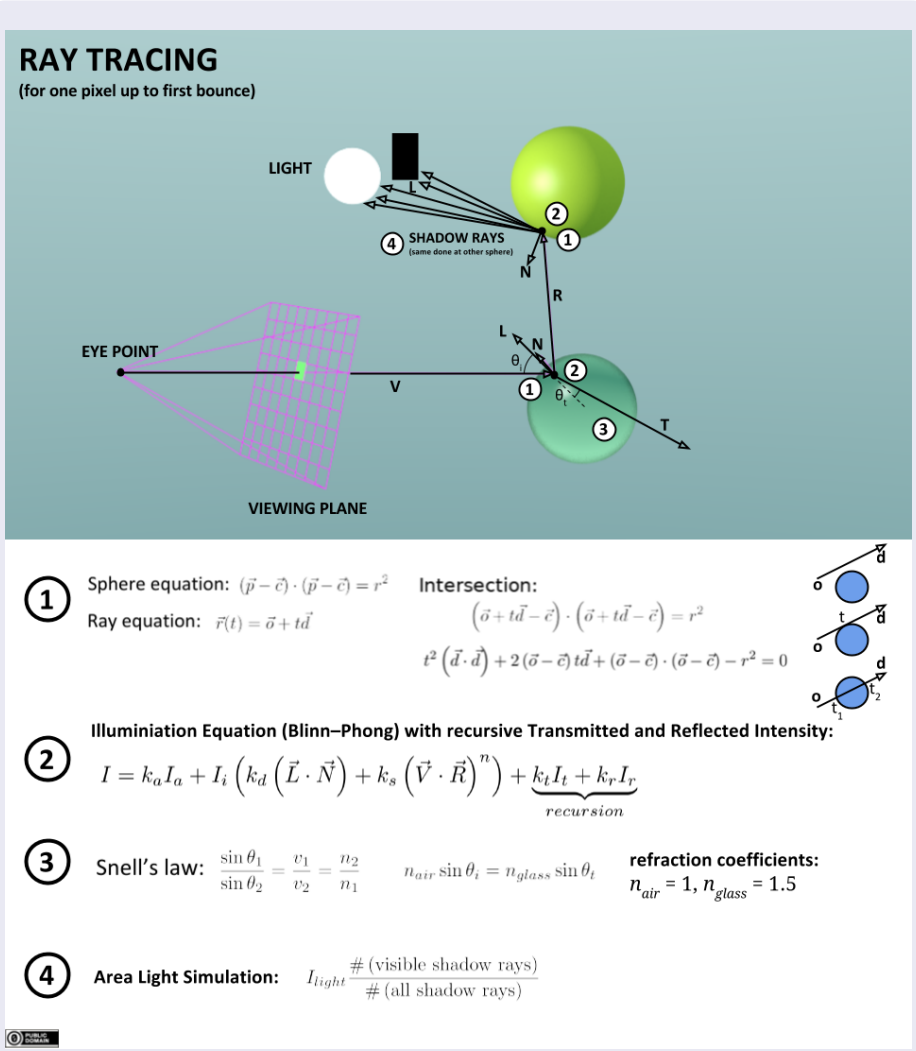
1. Launch ray from eye
2. Identify pixel in image plane
3. Identify luminosity and color from the ray path
4. Be recursive on the reflections

Are rays traveling backwards ?

No, just a computational trick

Ray tracing advantages

- Realistic, especially for shadows
- Suitable for parallel computing



Ray tracing in practice

https://en.wikipedia.org/wiki/List_of_ray_tracing_software

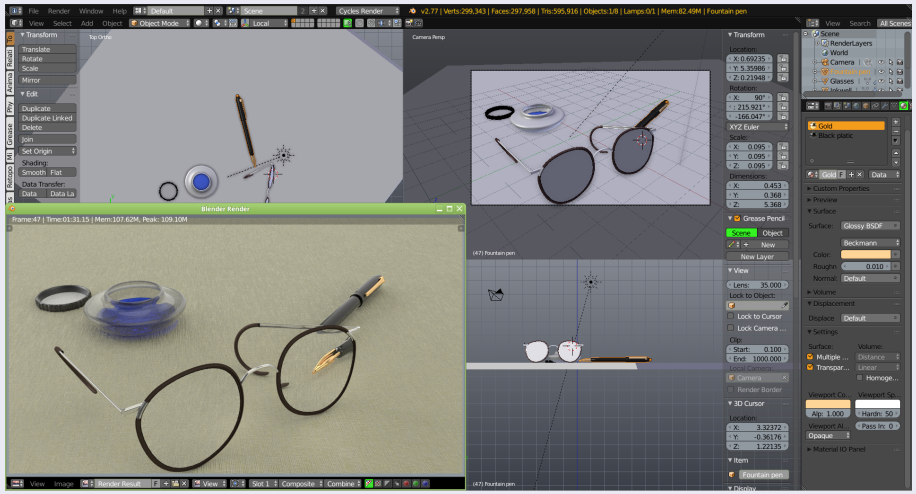
Comprehensive software list

- From geek level to professional
- From free to out of reach

To get a simple Python idea of the process

- Very simple ray tracing : raytracing.py on GitHub
- Faster : raytrace
- Comprehensive, in development : pyRT

Blender



by Asav on Wikimedia Commons

materalia
Pôle de Compétitivité Matériaux

ENLIGHT ME TZ

UNIVERSITÉ DE LORRAINE
CentraleSupélec

Ray tracing in practice

https://en.wikipedia.org/wiki/List_of_ray_tracing_software

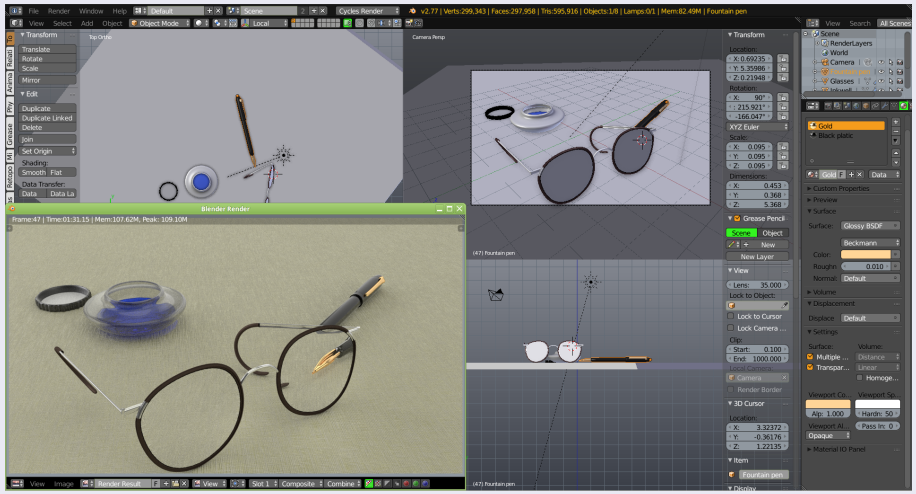
Comprehensive software list

- From geek level to professional
- From free to out of reach

To get a simple Python idea of the process

- Very simple ray tracing : raytracing.py on GitHub
- Faster : raytrace
- Comprehensive, in development : pyRT

Blender



by Asav on Wikimedia Commons

materalia
Pôle de Compétitivité Matériaux

ENLIGHT ME TZ

UNIVERSITÉ DE LORRAINE
CentraleSupélec

Path tracing is $\mathrm{XXI}^{\mathrm{st}}$ century ray tracing

Using a Monte Carlo method to solve the *Rendering equation*

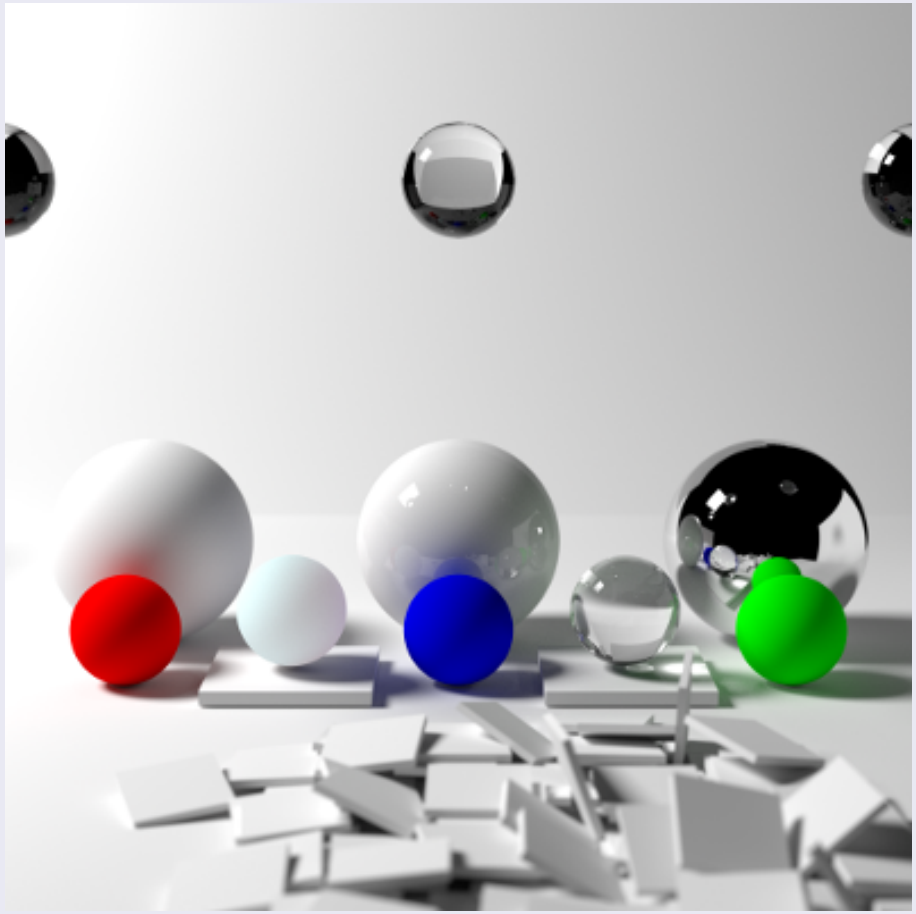
Ray tracing limitations

- Indirect lighting not taken into account
- Depth of field unavailable
- Solution : integrate all the light and calculate emission

Monte Carlo integration

- Many random rays
- Integration of the rendering equation
- Surface reflectance function

Path tracing



by Qutorial on Wikimedia Commons

Path tracing is XXIst century ray tracing

Using a Monte Carlo method to solve the *Rendering equation*

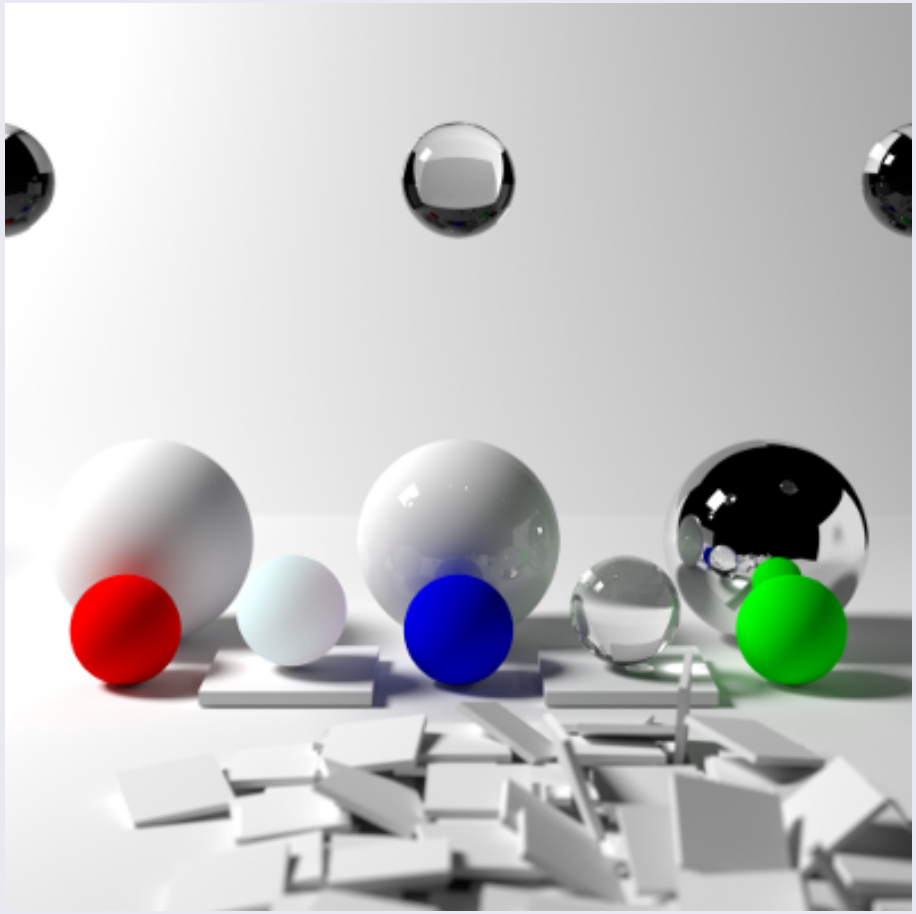
Ray tracing limitations

- Indirect lighting not taken into account
- Depth of field unavailable
- Solution : integrate all the light and calculate emission

Monte Carlo integration

- Many random rays
- Integration of the rendering equation
- Surface reflectance function

Path tracing



by Qutorial on Wikimedia Commons

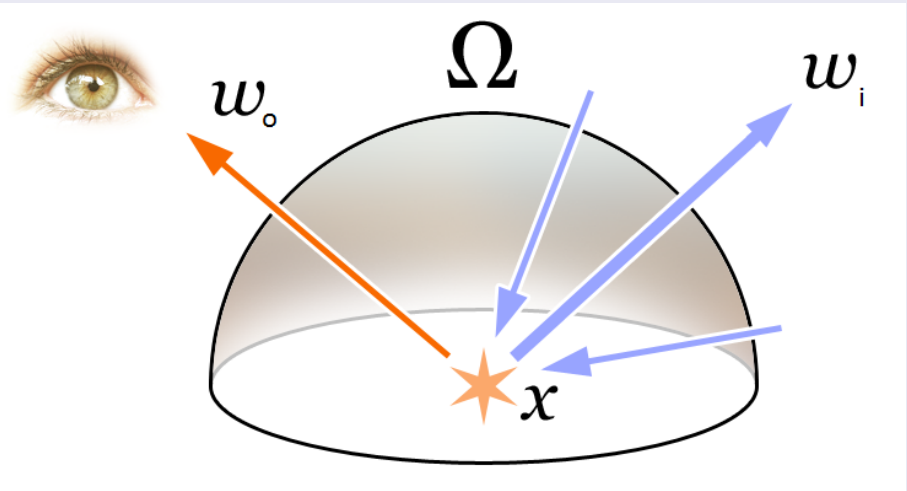
Illuminance on each scene point: a Monte Carlo integration of the rendering equation

The rendering equation — *Jim Kajiya, 1986*

$$L_o(\mathbf{x}, \omega_o, \lambda, t) = L_e(\mathbf{x}, \omega_o, \lambda, t) + \int_\Omega f_r(\mathbf{x}, \omega_i, \omega_o, \lambda, t)\, L_i(\mathbf{x}, \omega_i, \lambda, t)\, (\omega_i \cdot \mathbf{n})\, \mathrm{d}\,\omega_i$$

Monte Carlo numerical method

- Many random rays from the camera
- Bi-directionnal variant : rays from the sources as well
- Adaptive variant : details sensitive zones



by Timrb on Wikimedia Commons

- L : total spectral irradiance
- f_r : reflectance distribution

Path tracing in practice

https://en.wikipedia.org/wiki/List_of_ray_tracing_software

No particular software

High level ray tracing software include path tracing options

Example : Blender

https://youtu.be/Y8iXjf7XZ9Q
http://blender.org

€ Huge market $

Used anywhere 3D images are present

Path tracing in practice

https://en.wikipedia.org/wiki/List_of_ray_tracing_software

No particular software

High level ray tracing software include path tracing options

Example : Blender

https://youtu.be/Y8iXjf7XZ9Q
http://blender.org

€ Huge market $

Used anywhere 3D images are present

Optical design : Zemax, Code V, OSLO...

Tracing rays is good enough modeling for many applications

`https://youtu.be/3vgGMGU_iA8`

Industry standard software €

- ZEMAX – OpticStudio
- Code V
- OSLO

Free software

- OptGeo (2D)

Online software

- LightMachinery

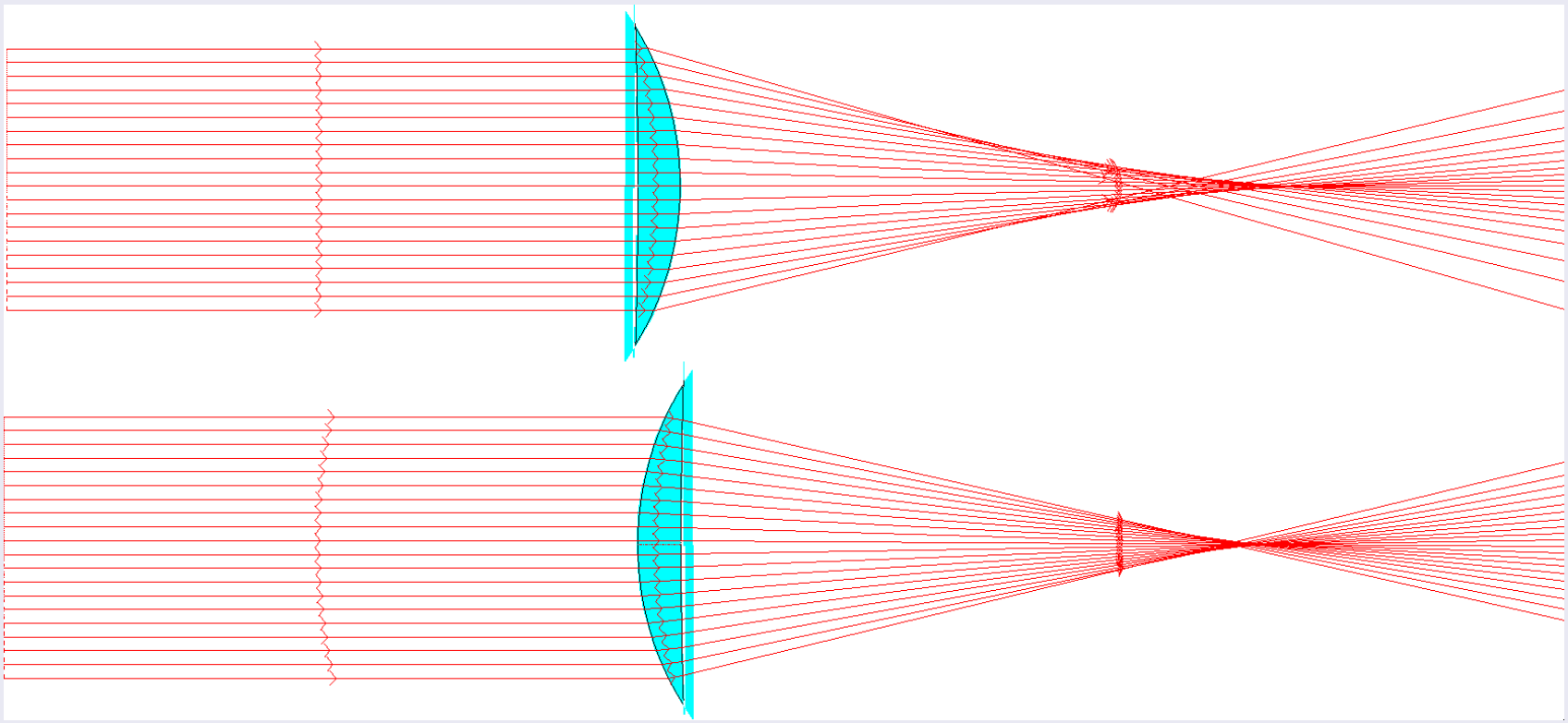
Software libraries to use inside your own code

- GNU Optical Design C++
- Opticspy Python

OptGeo
Efficient though simple

GNU

Example of a plano-convex lens that changes its behavior when reversed



Opticspy

Free, python based, useable everywhere and anyhow

by Xing Fan

Opticspy is

- implementing ray tracing among others
- free to use and modify
- compatible with any library with a python interface
- still in development
- documented through python comments
- available on all platforms
- yearning for your contribution

Opticspy is not

- clickable (though it can accept GUI's)
- as powerful as expensive software
- complete (but progressing)

RSITÉ
RRAINE

Opticspy

Free, python based, useable everywhere and anyhow

by Xing Fan

Opticspy is

- implementing ray tracing among others
- free to use and modify
- compatible with any library with a python interface
- still in development
- documented through python comments
- available on all platforms
- yearning for your contribution

Opticspy is not

- clickable (though it can accept GUI's)
- as powerful as expensive software
- complete (but progressing)

RSITÉ
RRAINE

Opticspy : an introduction

Introduction to the basic functions

http://sterncat.github.io/files/Real_Ray_Tracing.html

Import opticspy

```
%matplotlib inline
from opticspy.ray_tracing import *
```

Define lens

```
New_Lens = lens.Lens(lens_name='Triplet',creator='X
New_Lens.FNO = 5 #FNO=f/D is the f-number
```

Opticspy : an introduction

Introduction to the basic functions

http://sterncat.github.io/files/Real_Ray_Tracing.html

Import opticspy

```
%matplotlib inline
from opticspy.ray_tracing import *
```

Define lens

```
New_Lens = lens.Lens(lens_name='Triplet',creator='X
New_Lens.FNO = 5 #FNO=f/D is the f-number
```

Opticspy : define an optical system

Optical system definition

```
#Define wavelengths
New_Lens.add_wavelength(wl = 656.30) #add more
#Define start angles
New_Lens.add_field_YAN(angle=0) #Initial angles
#Define surfaces
New_Lens.add_surface(number=1,radius=10000000,thick
New_Lens.add_surface(number=2,radius=41.15909,thick
```

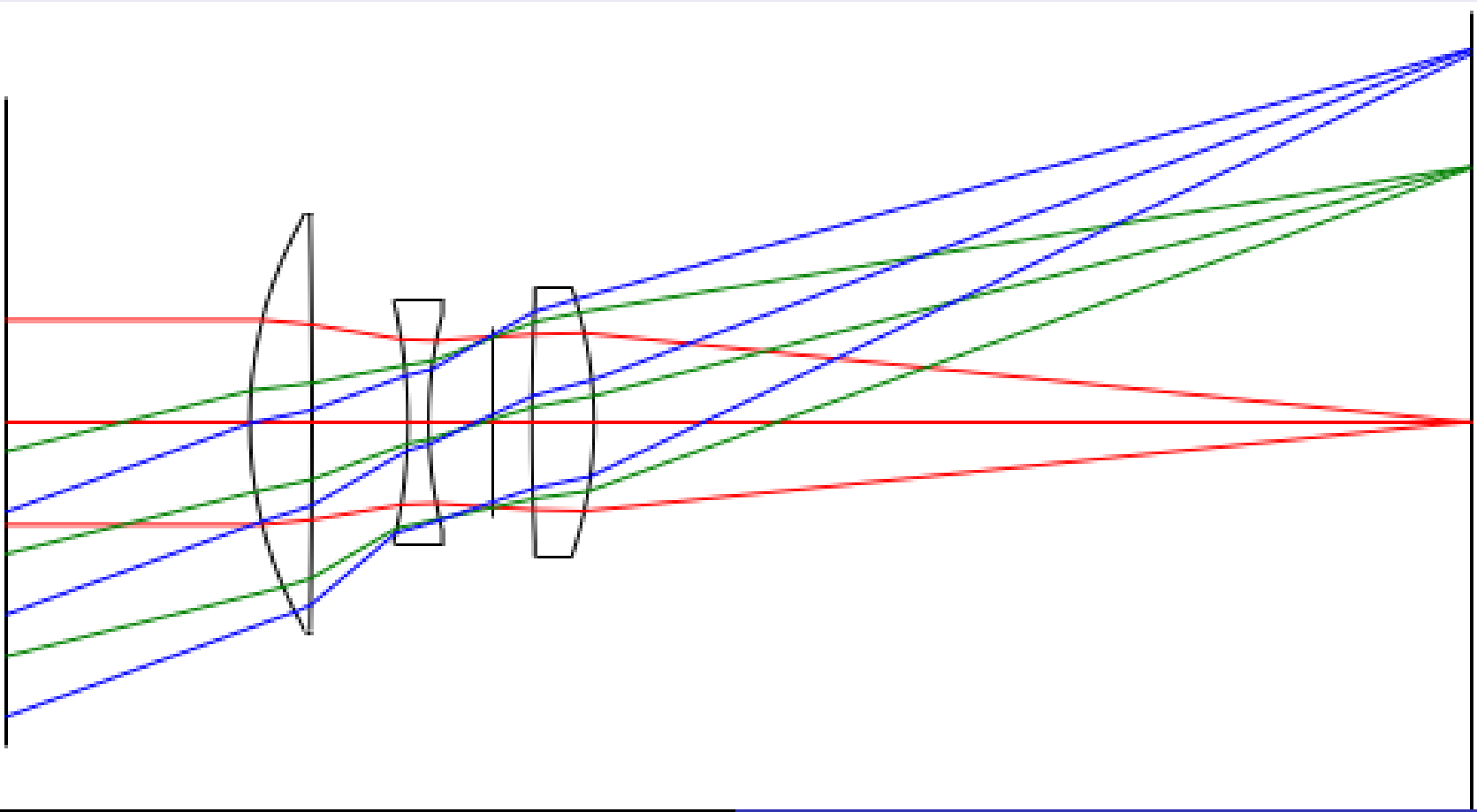
Optical system calculation

```
New_Lens.refresh_paraxial()
```

Opticspy : optical system analysis

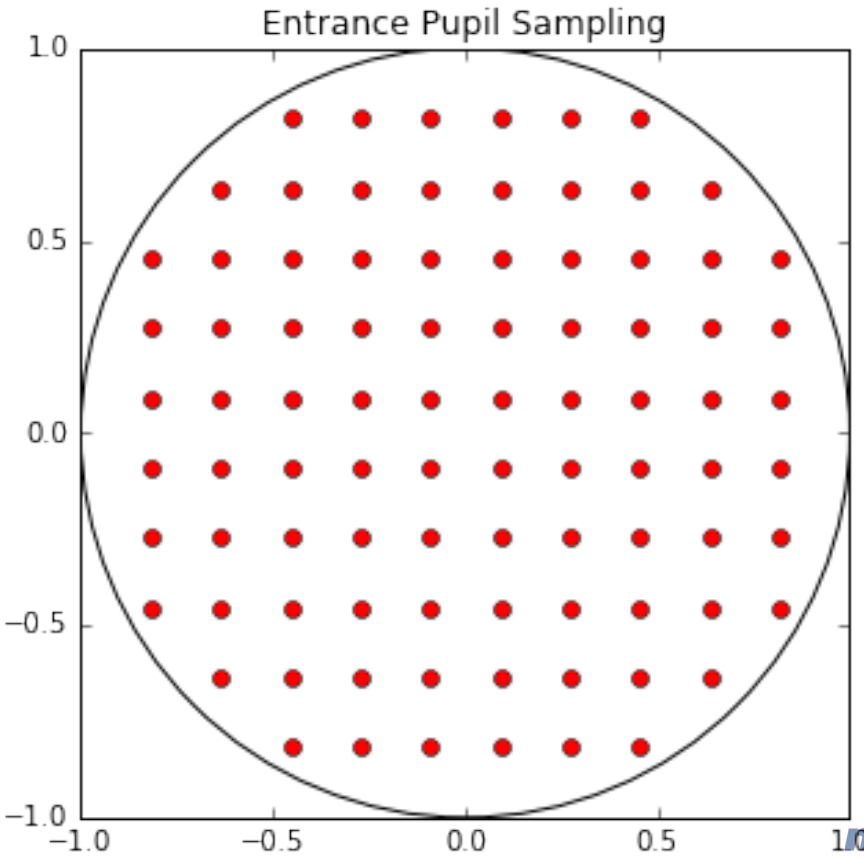
Trace rays

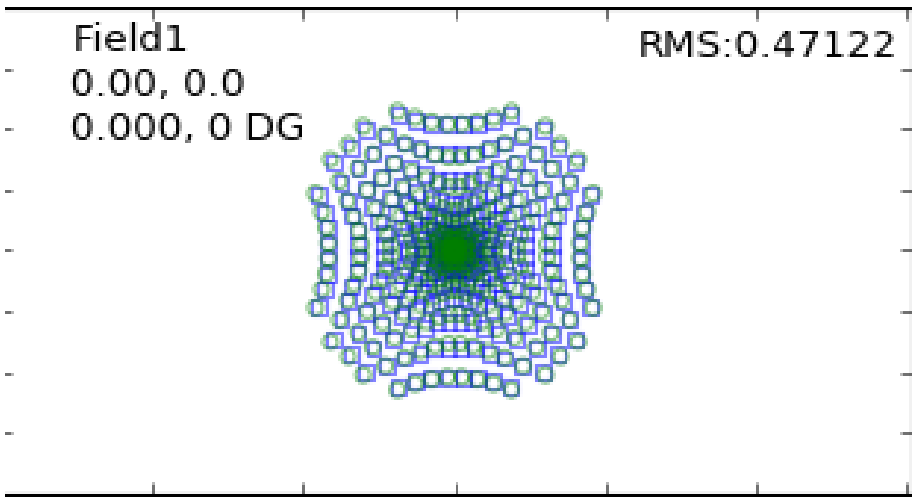
```
draw.draw_system(New_Lens)
```



Opticspy : Spot Diagram

```
analysis.spotdiagram(New_Lens,[1,2,3],[1,2,3],n=20,
```



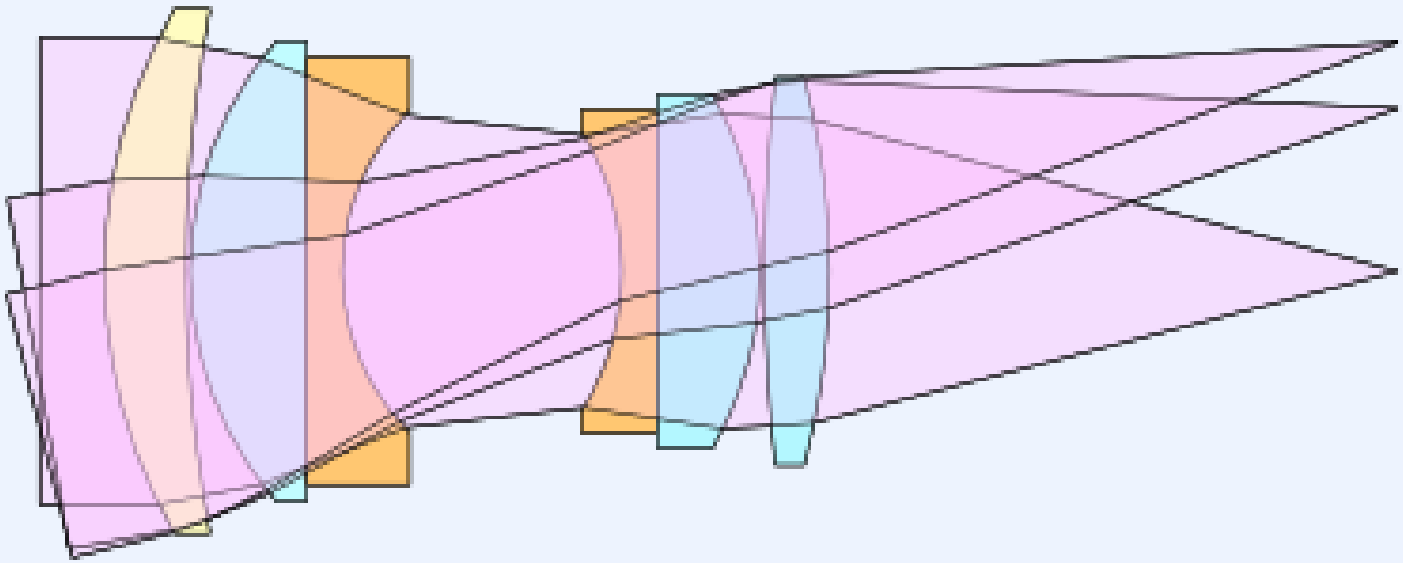


Opticspy : imagination is the limit

And much more

rayoptics python module

https://pypi.org/project/rayoptics/



Read the manual

https://ray-optics.readthedocs.io/en/latest/