
naivert

Release 0.0.1

Minghao Gou

Nov 15, 2020

CONTENTS

1	About naivert	1
2	Examples	3
3	Python API	9
4	Indices and tables	11

ABOUT NAIVERT

naivert is an implementation of ray tracing algorithm using Phong illumination model.

For geometry calculation, naivert uses the library of [Geometry3D](#).

You may refer to the examples and documentation of Geometry3D to get to know how to use naivert.

1.1 Core Features

- Simple Object like Cubic, Sphere, Cylinder, Cone, Rectangle, Parallelepiped, Parallogram and Circle.
- Point light sources and ambient light sources.
- Basic materials.
- Ray tracing algorithm rendering.
- Phong illumination model.

1.2 Resources

- Documentations
- PDF_Documentations
- Code: <https://github.com/GouMinghao/naivert>

EXAMPLES

Here are some examples. This might be very slow, you should change the resolution of the image to reduce the time cost.

2.1 Example 1

Simply run the code below after installation:

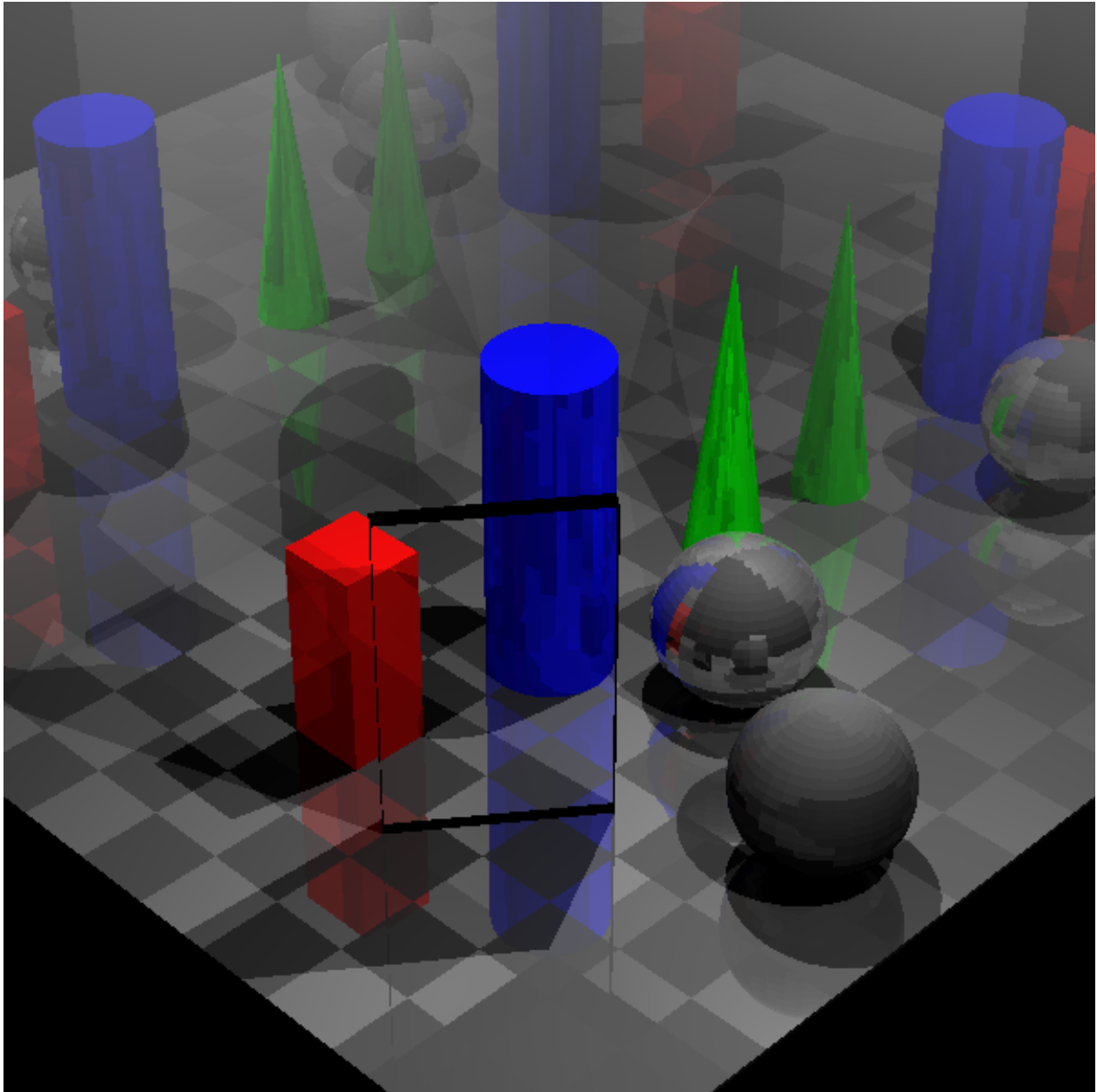
```
>>> import naivert
>>> from Geometry3D import *
>>> from math import sqrt
>>> main_scene = naivert.Scene()
>>> main_camera = naivert.Camera(Point(400,-300,400),Point(385,-285,385),10 * Vector(-
↪1 /sqrt(6),1/sqrt(6),>>> 2/sqrt(6)),10*Vector(1/sqrt(2),1/sqrt(2),0),'main_camera.
↪png',resolution=(600,600))
>>> point_light = naivert.PointLight(Point(50,100,200),[5.0,5.0,5.0])
>>> point_light2 = naivert.PointLight(Point(200,50,200),[4.0,4.0,4.0])
>>> ambient_light = naivert.AmbientLight([3.0,3.0,3.0])
>>> main_scene.add_camera(main_camera)
>>> main_scene.add_light(point_light)
>>> main_scene.add_light(point_light2)
>>> main_scene.add_light(ambient_light)
>>> main_scene.add_cph(Parallelepiped(Point(-51,-50,0),1*x_unit_vector(),200*y_unit_
↪vector(),>>> 200*z_unit_vector()),naivert.Material.SpecularMaterial_White_1(),
↪reverse_normal=False)
>>> main_scene.add_cph(Parallelepiped(Point(-50,150,0),200*x_unit_vector(),1*y_unit_
↪vector(),>>> 200*z_unit_vector()),naivert.Material.SpecularMaterial_White_1(),
↪reverse_normal=False)
>>> main_scene.add_cph(Parallelepiped(Point(50,0,0),30*x_unit_vector()+40*y_unit_
↪vector(),-4*x_unit_vector()>>> +3*y_unit_vector(), 80*z_unit_vector()),naivert.
↪Material.Glass(),reverse_normal=False)
>>> main_scene.add_cph(Parallelepiped(Point(10,10,0),20*x_unit_vector(),20*y_unit_
↪vector(), 50 >>> *z_unit_vector()),naivert.Material.DiffusionMaterial_Red_1(),
↪reverse_normal=False)
>>> main_scene.add_cph(Sphere(Point(70,90,20),20,30,15),naivert.Material.
↪SpecularMaterial_White_1(),>>> reverse_normal=False)
>>> main_scene.add_cph(Sphere(Point(120,60,20),20,30,15),naivert.Material.
↪DiffusionMaterial_White_1(),>>> reverse_normal=False)
>>> main_scene.add_cph(Cone(Point(30,130,0),10,80*z_unit_vector(),n=30),
↪material=naivert.Material.>>> DiffusionMaterial_Green_1(),reverse_normal=False)
>>> main_scene.add_cph(Cylinder(Point(30,70,0),15,80*z_unit_vector(),n=30),
↪material=naivert.Material.>>> DiffusionMaterial_Blue_1(),reverse_normal=False)
>>> r = Render()

```

(continues on next page)

(continued from previous page)

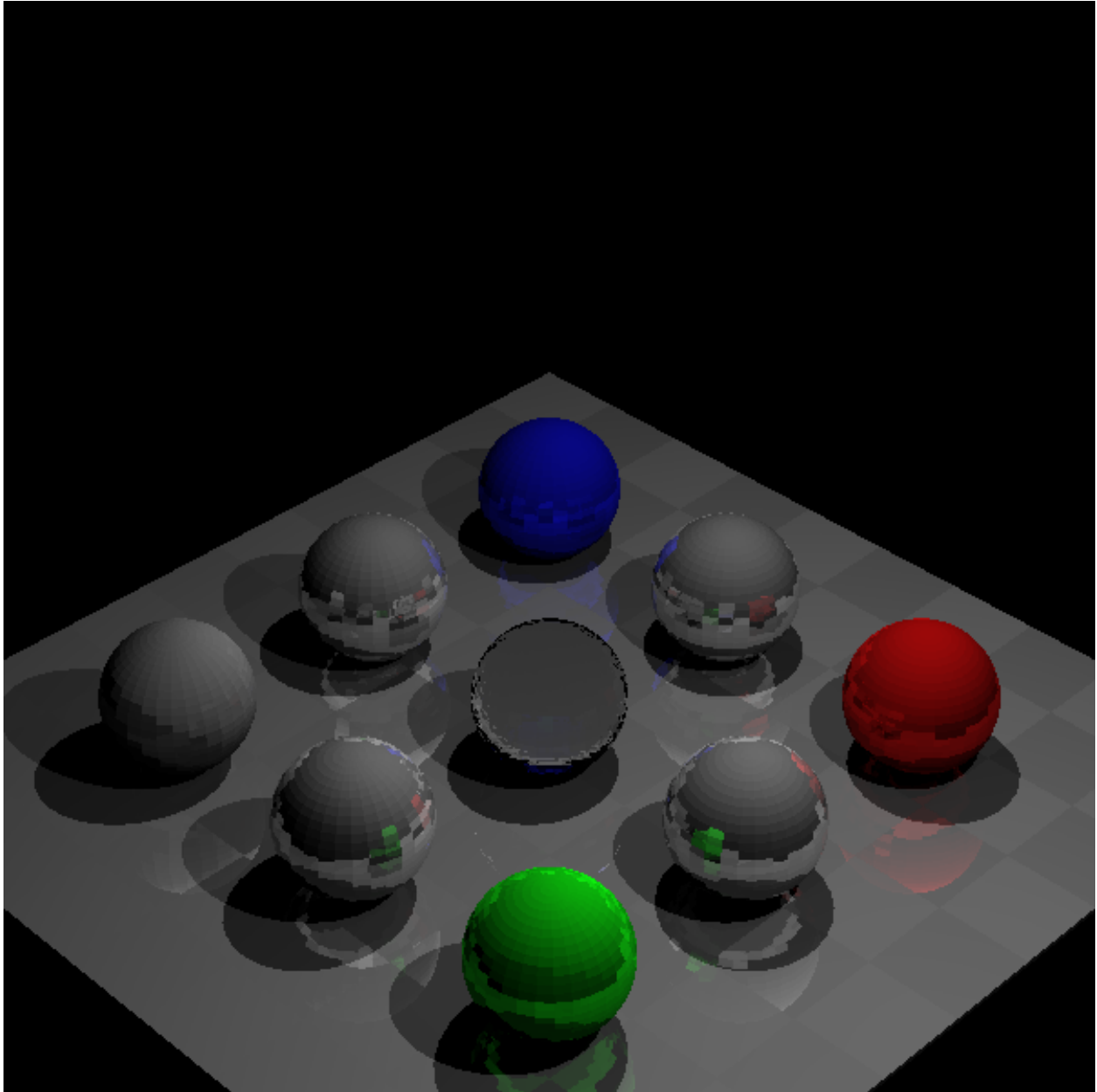
```
>>> for face in main_scene.face_list:  
...     r.add((face.cpg, 'r', 1))  
>>> r.show()  
>>> main_scene.render_scene(32)  
>>> main_scene.write_scene()
```



2.2 Example 2

Simply run the code below after installation:

```
>>> import naivert
>>> from Geometry3D import *
>>> from math import sqrt
>>>
>>> main_scene = naivert.Scene()
>>>
>>> main_camera = naivert.Camera(Point(350,-250,350),Point(335,-235,335),10 * Vector(-
↳ 1 /sqrt(6),1/sqrt(6),2/sqrt(6)),10*Vector(1/sqrt(2),1/sqrt(2),0),'main_camera.png',
↳ resolution=(600,600))
>>> point_light2 = naivert.PointLight(Point(200,50,200),[4.0,4.0,4.0])
>>> point_light = naivert.PointLight(Point(50,100,200),[5.0,5.0,5.0])
>>> main_scene.add_camera(main_camera)
>>> main_scene.add_light(point_light)
>>> main_scene.add_light(point_light2)
>>>
>>> main_scene.add_floor(-50,200,-50,200)
>>>
>>> main_scene.add_cph(Sphere(Point(0,0,15),15,30,15),naivert.Material.
↳ DiffusionMaterial_White_1(),reverse_normal=False)
>>> main_scene.add_cph(Sphere(Point(100,0,15),15,30,15),naivert.Material.
↳ DiffusionMaterial_Green_1(),reverse_normal=False)
>>> main_scene.add_cph(Sphere(Point(100,100,15),15,30,15),naivert.Material.
↳ DiffusionMaterial_Red_1(),reverse_normal=False)
>>> main_scene.add_cph(Sphere(Point(0,100,15),15,30,15),naivert.Material.
↳ DiffusionMaterial_Blue_1(),reverse_normal=False)
>>> main_scene.add_cph(Sphere(Point(50,0,15),15,30,15),naivert.Material.
↳ SpecularMaterial_White_1(),reverse_normal=False)
>>> main_scene.add_cph(Sphere(Point(0,50,15),15,30,15),naivert.Material.
↳ SpecularMaterial_White_1(),reverse_normal=False)
>>> main_scene.add_cph(Sphere(Point(50,100,15),15,30,15),naivert.Material.
↳ SpecularMaterial_White_1(),reverse_normal=False)
>>> main_scene.add_cph(Sphere(Point(100,50,15),15,30,15),naivert.Material.
↳ SpecularMaterial_White_1(),reverse_normal=False)
>>> main_scene.add_cph(Sphere(Point(50,50,15),15,30,15),naivert.Material.Glass(),
↳ reverse_normal=False)
>>>
>>> main_scene.render_scene(56)
>>>
>>> main_scene.write_scene()
```



2.3 Example 3

Simply run the code below after installation:

```
>>> import naivert
>>> from Geometry3D import *
>>> from math import sqrt
>>> import copy
>>>
>>> main_scene = naivert.Scene()
>>> main_camera = naivert.Camera(Point(400, -300, 400), Point(385, -285, 385), 10 * Vector(-
↪ 1 / sqrt(6), 1 / sqrt(6), 2 / sqrt(6)), 10 * Vector(1 / sqrt(2), 1 / sqrt(2), 0), 'main_camera.png',
↪ resolution=(15, 15))
```

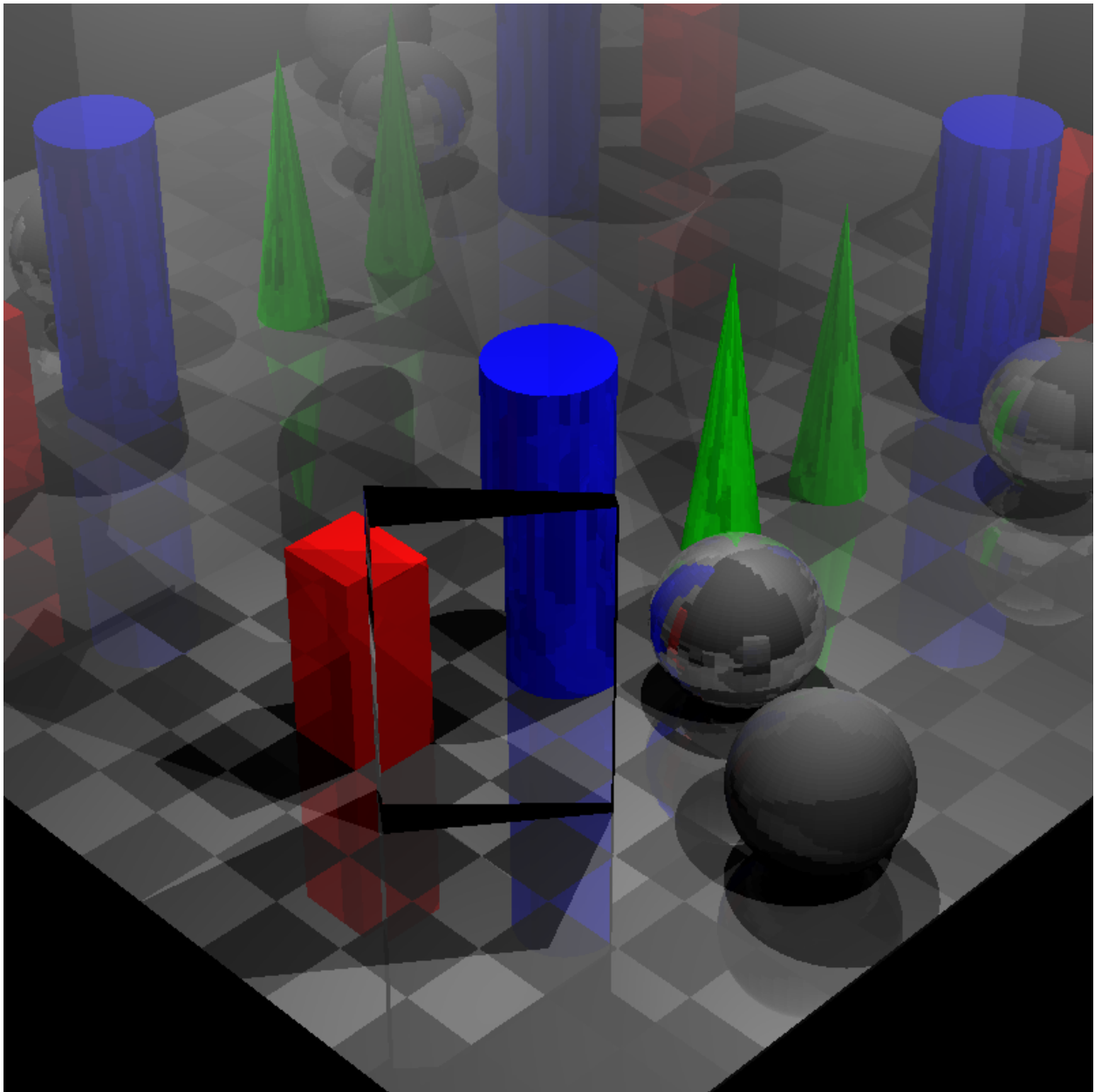
(continues on next page)

(continued from previous page)

```

>>> point_light = naivert.PointLight(Point(50,100,200),[5.0,5.0,5.0])
>>> point_light2 = naivert.PointLight(Point(200,50,200),[4.0,4.0,4.0])
>>> ambient_light = naivert.AmbientLight([3.0,3.0,3.0])
>>> main_scene.add_camera(main_camera)
>>> main_scene.add_light(point_light)
>>> main_scene.add_light(point_light2)
>>> main_scene.add_light(ambient_light)
>>> main_scene.add_floor(-50,200,-50,200)
>>> main_scene.add_cph(Parallelepiped(Point(-51,-50,0),1*x_unit_vector(),200*y_unit_
↳vector(),200*z_unit_vector()),naivert.Material.SpecularMaterial_White_1(),reverse_
↳normal=False)
>>>
>>> main_scene.add_cph(Parallelepiped(Point(-50,150,0),200*x_unit_vector(),1*y_unit_
↳vector(),200*z_unit_vector()),naivert.Material.SpecularMaterial_White_1(),reverse_
↳normal=False)
>>>
>>> a = Point(50,0,0)
>>> b = Point(80,40,0)
>>> c = Point(76,43,0)
>>> d = Point(38,9,0)
>>> a1 = Point(50,0,80)
>>> b1 = Point(80,40,80)
>>> c1 = Point(76,43,80)
>>> d1 = Point(38,9,80)
>>> cp1 = ConvexPolygon((a,b,c,d))
>>> cp2 = ConvexPolygon((a1,b1,c1,d1))
>>> cp3 = ConvexPolygon((a,b,b1,a1))
>>> cp4 = ConvexPolygon((b,c,c1,b1))
>>> cp5 = ConvexPolygon((c,d,d1,c1))
>>> cp6 = ConvexPolygon((d,a,a1,d1))
>>> cph = ConvexPolyhedron((cp1,cp2,cp3,cp4,cp5,cp6))
>>> main_scene.add_cph(cph,naivert.Material.Glass(),reverse_normal=False)
>>>
>>> main_scene.add_cph(Parallelepiped(Point(10,10,0),20*x_unit_vector(),20*y_unit_
↳vector(), 50 *z_unit_vector()),naivert.Material.DiffusionMaterial_Red_1(),reverse_
↳normal=False)
>>> main_scene.add_cph(Sphere(Point(70,90,20),20,30,15),naivert.Material.
↳SpecularMaterial_White_1(),reverse_normal=False)
>>> main_scene.add_cph(Sphere(Point(120,60,20),20,30,15),naivert.Material.
↳DiffusionMaterial_White_1(),reverse_normal=False)
>>> main_scene.add_cph(Cone(Point(30,130,0),10,80*z_unit_vector(),n=30),
↳material=naivert.Material.DiffusionMaterial_Green_1(),reverse_normal=False)
>>> main_scene.add_cph(Cylinder(Point(30,70,0),15,80*z_unit_vector(),n=30),
↳material=naivert.Material.DiffusionMaterial_Blue_1(),reverse_normal=False)
>>>
>>> main_scene.render_scene(48)
>>> main_scene.write_scene()

```



PYTHON API

3.1 naivert.camera package

3.1.1 Submodules

3.1.2 naivert.camera.camera module

3.1.3 Module contents

3.2 naivert.geometry package

3.2.1 Submodules

3.2.2 naivert.geometry.face module

3.2.3 naivert.geometry.get_reflection_halfline module

3.2.4 naivert.geometry.get_refraction_halfline module

3.2.5 naivert.geometry.get_tangential_vector module

3.2.6 naivert.geometry.inter_halfline_face_list module

3.2.7 Module contents

3.3 naivert.light package

3.3.1 Submodules

3.3.2 naivert.light.light module

3.3.3 Module contents

3.4 naivert.material package

3.4.1 Submodules

3.4.2 naivert.material.material module

3.4.3 Module contents

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`