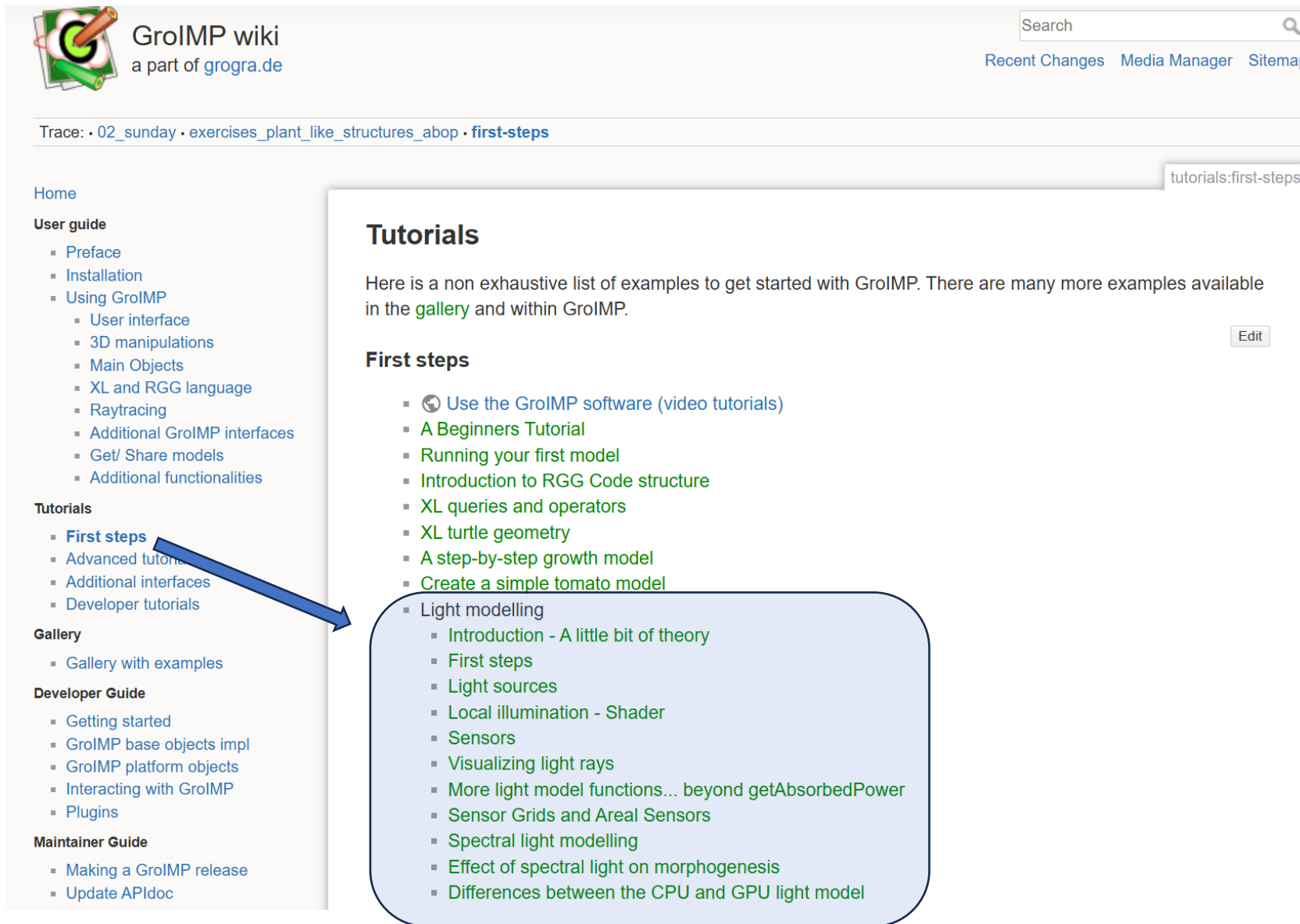


Introduction to light modelling and a glimpse of photosynthesis modelling

Michael Henke

Yazhouwan National Laboratory, China
崖州湾国家实验室

PIFI Summer Workshop on Systems Modelling of Plant Factory for
Excellent Young Scientist
中欧植物工厂系统建模暑期学校
Shanghai, 2025 Oct. 29



GrolIMP wiki
a part of grogra.de

Search

[Recent Changes](#) [Media Manager](#) [Sitemap](#)

Trace: • [02_sunday](#) • [exercises_plant_like_structures_abop](#) • **[first-steps](#)**

tutorials:first-steps

Home

User guide

- [Preface](#)
- [Installation](#)
- [Using GrolIMP](#)
 - [User interface](#)
 - [3D manipulations](#)
 - [Main Objects](#)
 - [XL and RGG language](#)
 - [Raytracing](#)
 - [Additional GrolIMP interfaces](#)
 - [Get/ Share models](#)
 - [Additional functionalities](#)

Tutorials

- **[First steps](#)**
- [Advanced tutorials](#)
- [Additional interfaces](#)
- [Developer tutorials](#)

Gallery

- [Gallery with examples](#)

Developer Guide

- [Getting started](#)
- [GrolIMP base objects impl](#)
- [GrolIMP platform objects](#)
- [Interacting with GrolIMP](#)
- [Plugins](#)

Maintainer Guide

- [Making a GrolIMP release](#)
- [Update API doc](#)

Tutorials

Here is a non exhaustive list of examples to get started with GrolIMP. There are many more examples available in the [gallery](#) and within GrolIMP.

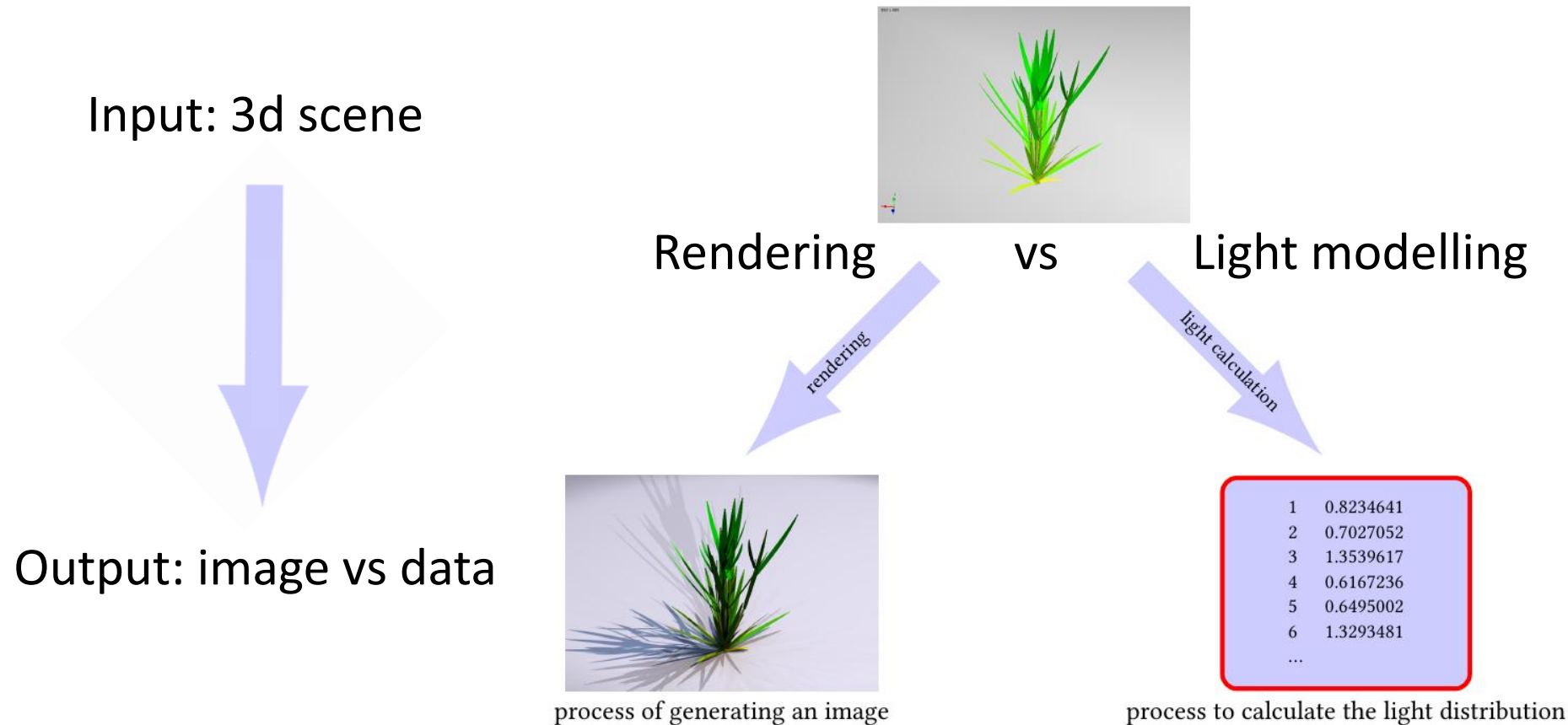
[Edit](#)

First steps

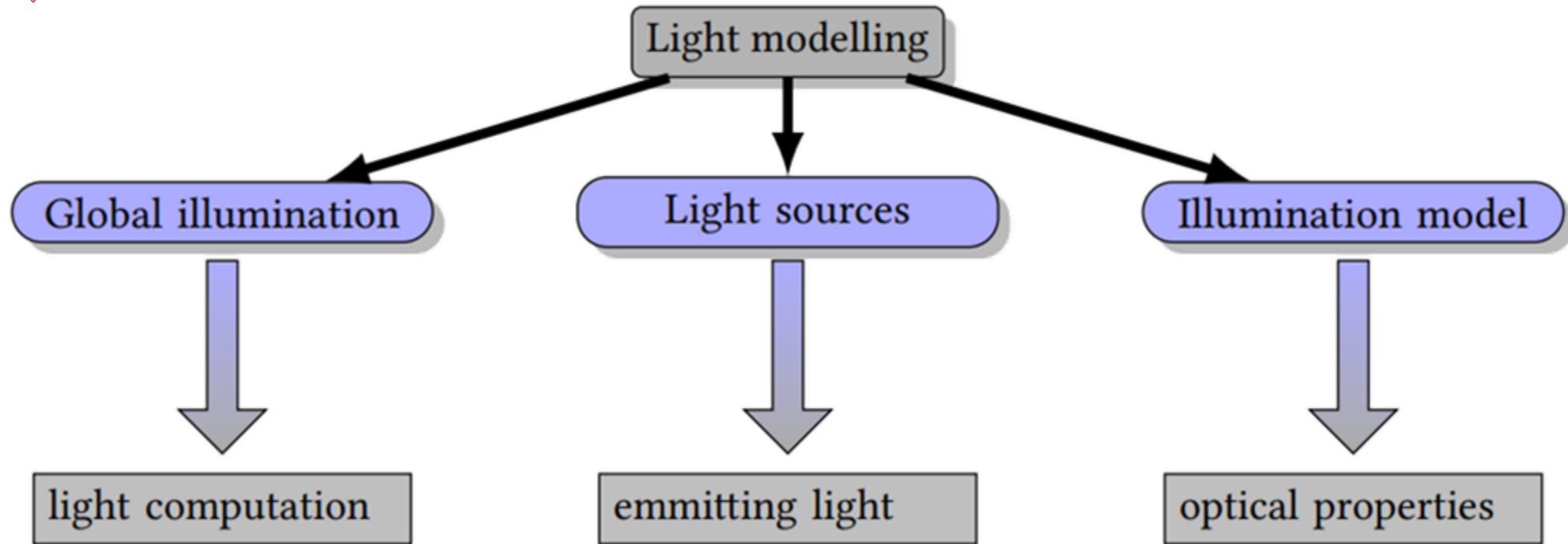
- [🌐 Use the GrolIMP software \(video tutorials\)](#)
- [A Beginners Tutorial](#)
- [Running your first model](#)
- [Introduction to RGG Code structure](#)
- [XL queries and operators](#)
- [XL turtle geometry](#)
- [A step-by-step growth model](#)
- [Create a simple tomato model](#)
- [Light modelling](#)
 - [Introduction - A little bit of theory](#)
 - [First steps](#)
 - [Light sources](#)
 - [Local illumination - Shader](#)
 - [Sensors](#)
 - [Visualizing light rays](#)
 - [More light model functions... beyond getAbsorbedPower](#)
 - [Sensor Grids and Areal Sensors](#)
 - [Spectral light modelling](#)
 - [Effect of spectral light on morphogenesis](#)
 - [Differences between the CPU and GPU light model](#)

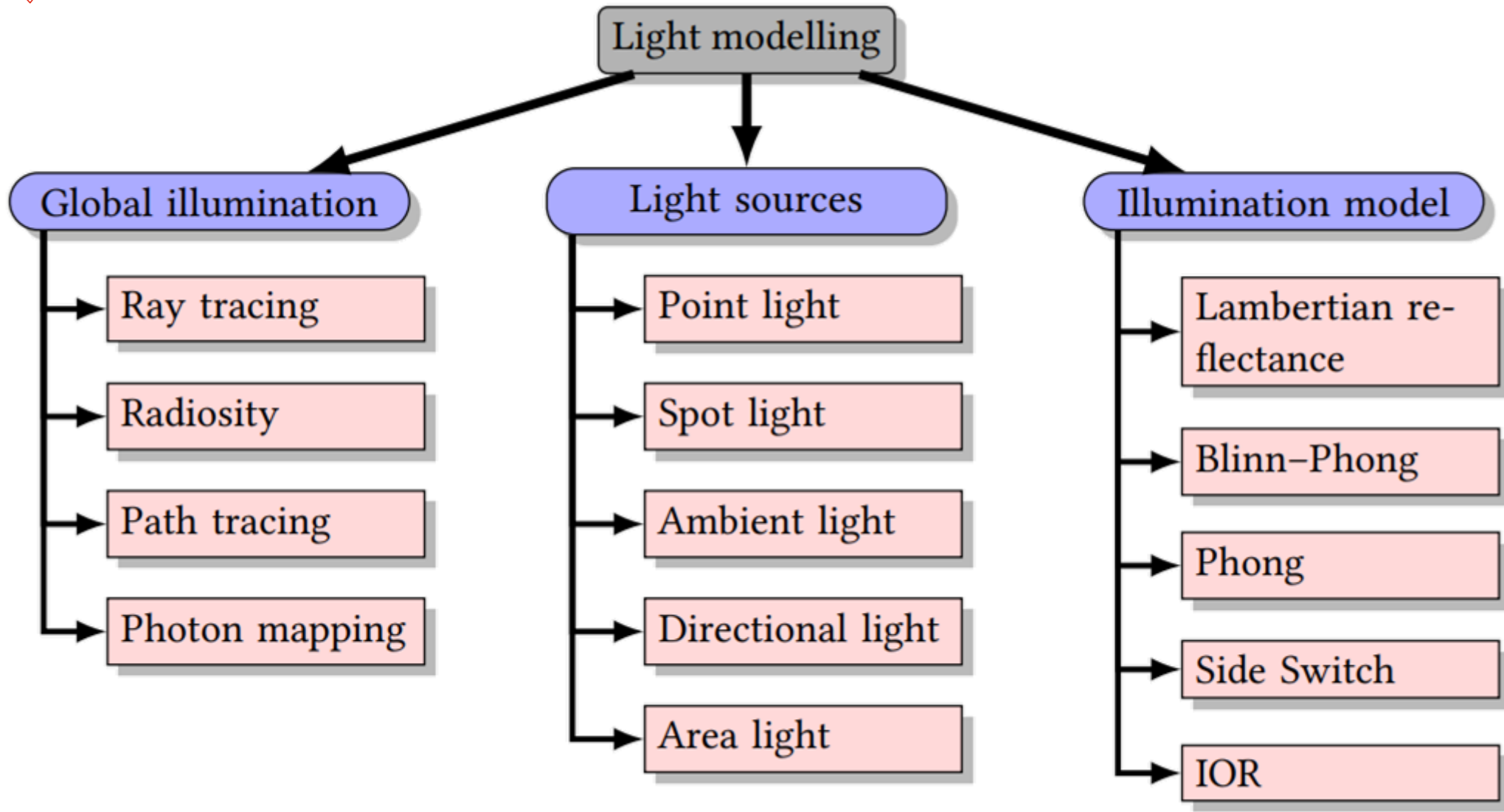
Wiki: <https://wiki.grogra.de/doku.php?id=tutorials:first-steps>

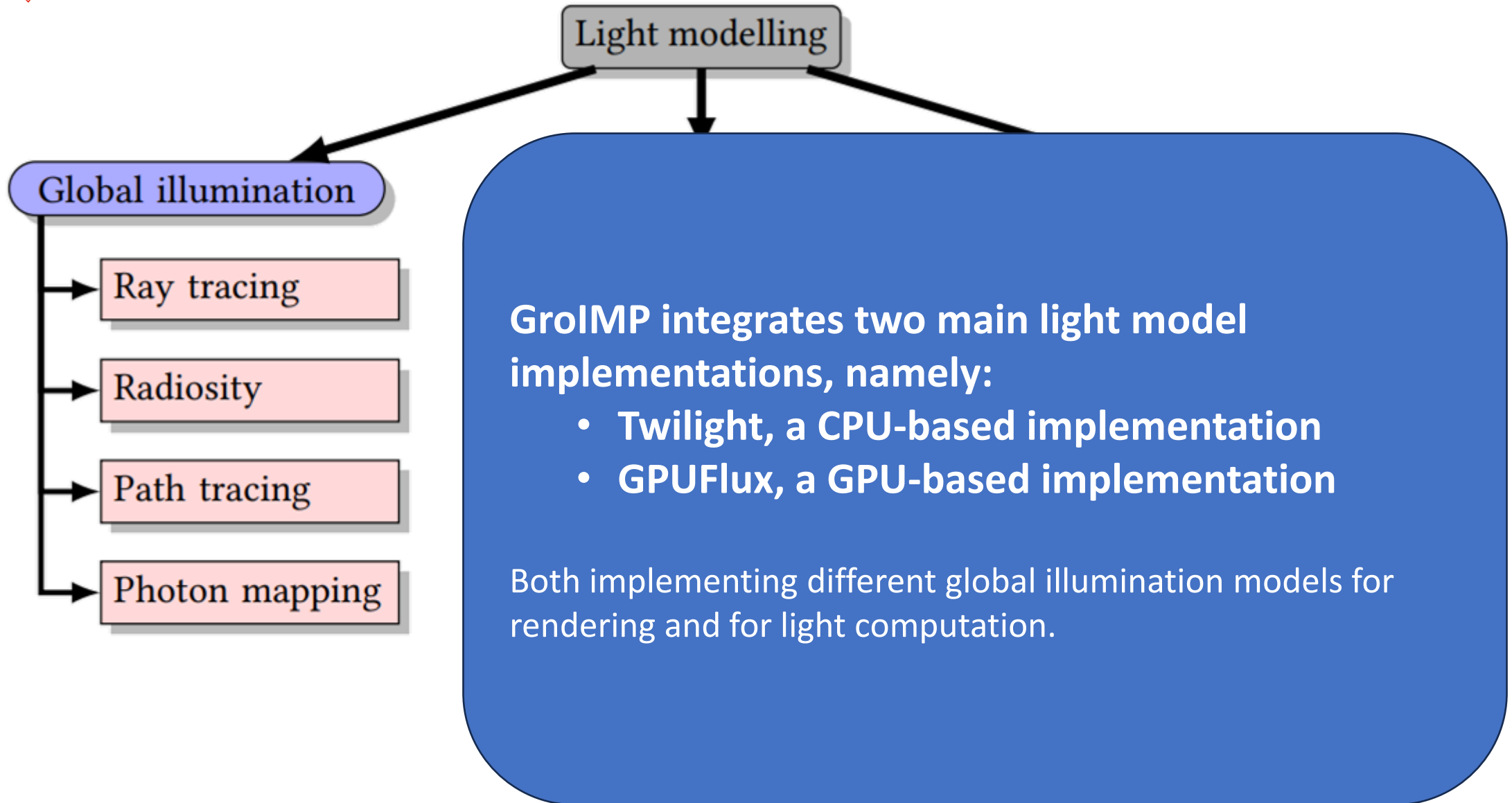
Light modelling is the numerical solution of how electromagnetic energy is **emitted**, **transported**, **scattered**, **absorbed**, and **measured** in a scene. Practically, it predicts **radiance** at objects from known **emitters**, **geometry**, and **materials**, across wavelength and time.



Wiki: <https://wiki.grogra.de/doku.php?id=tutorials:light-modeling-introduction>

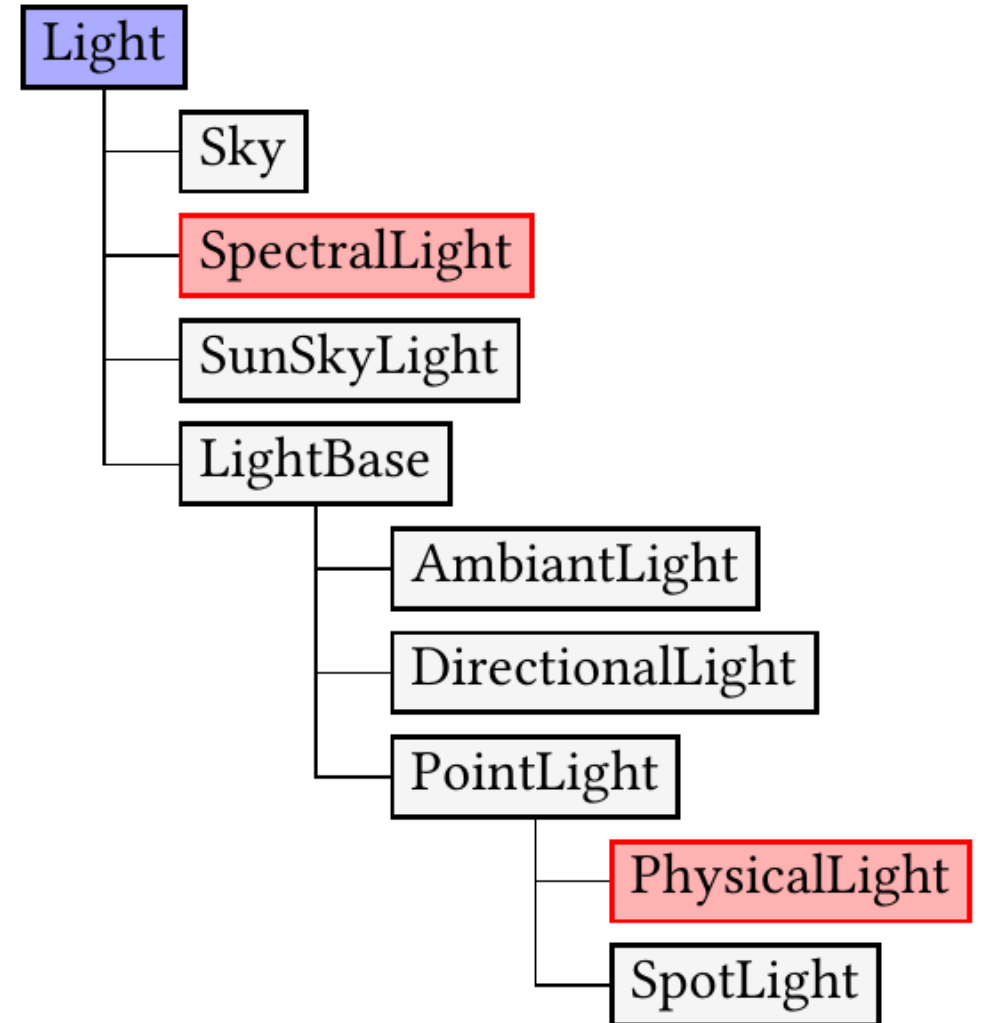


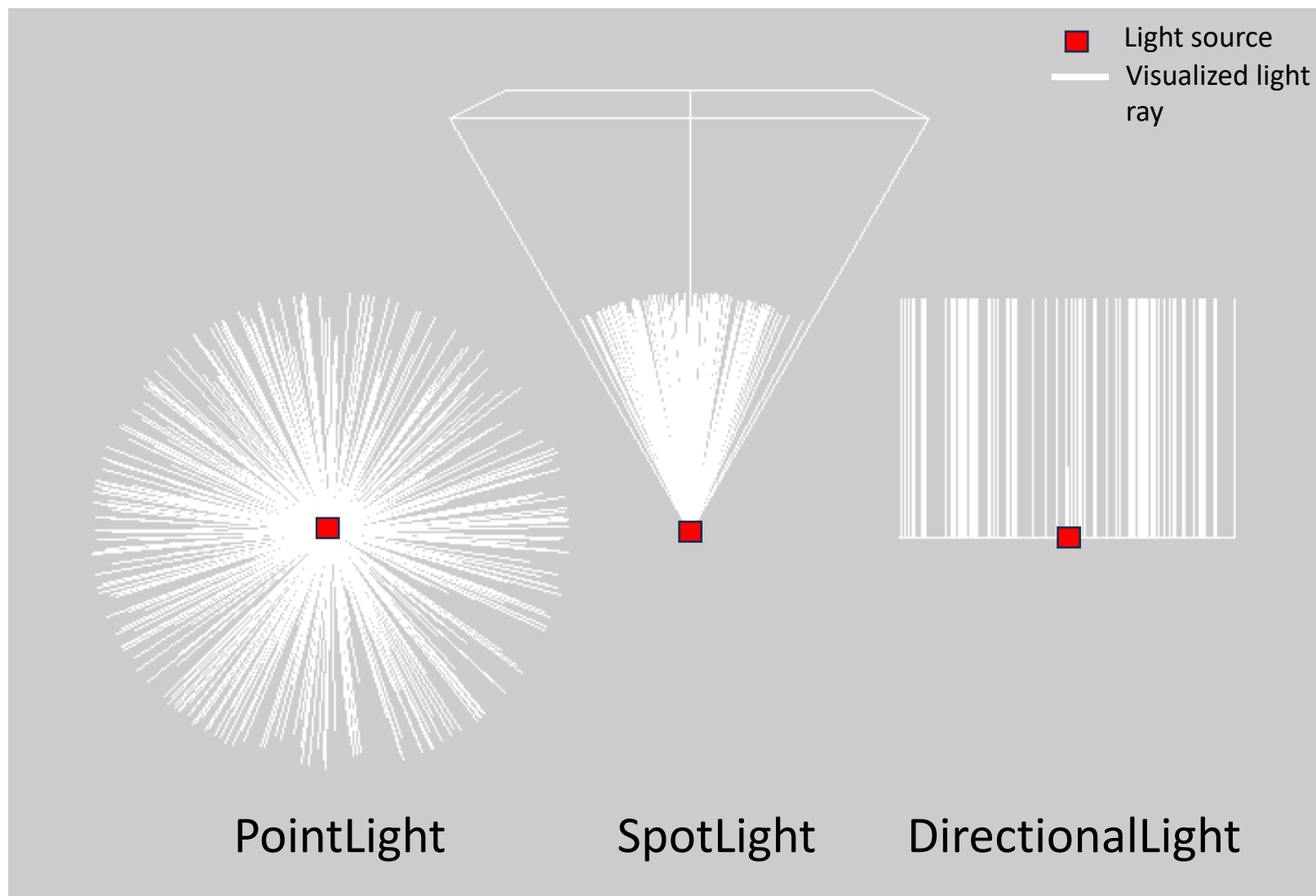




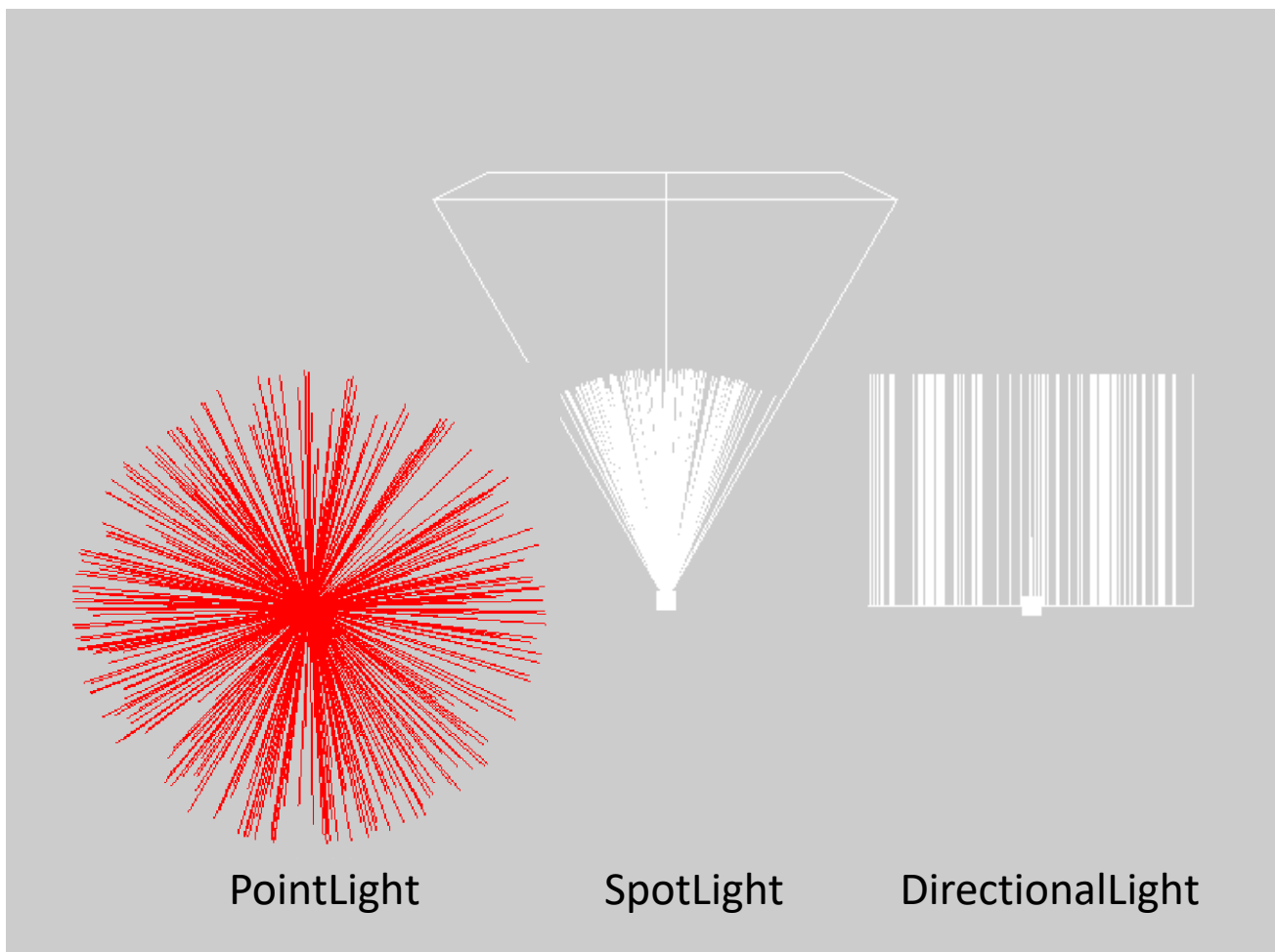
```
21 // initialize the light model
22 LightModel CPU_LM = new LightModel(RAYS, DEPTH);
23
24 CPU_LM.compute(); // run the light model
25
26 //check the scene objects for their light absorption
27 Spectrum ms;
28 [
29     x:Box::> { ms = CPU_LM.getAbsorbedPower(x); }
30 ]
31 print("absorbed = "+ms);println(" = "+ms.integrate(), 0xff0000);
```

- GroIMP provides a complete set of possible implementations
- All implement the *Light* and *LightBase* interfaces, which makes them easy to handle and exchange.





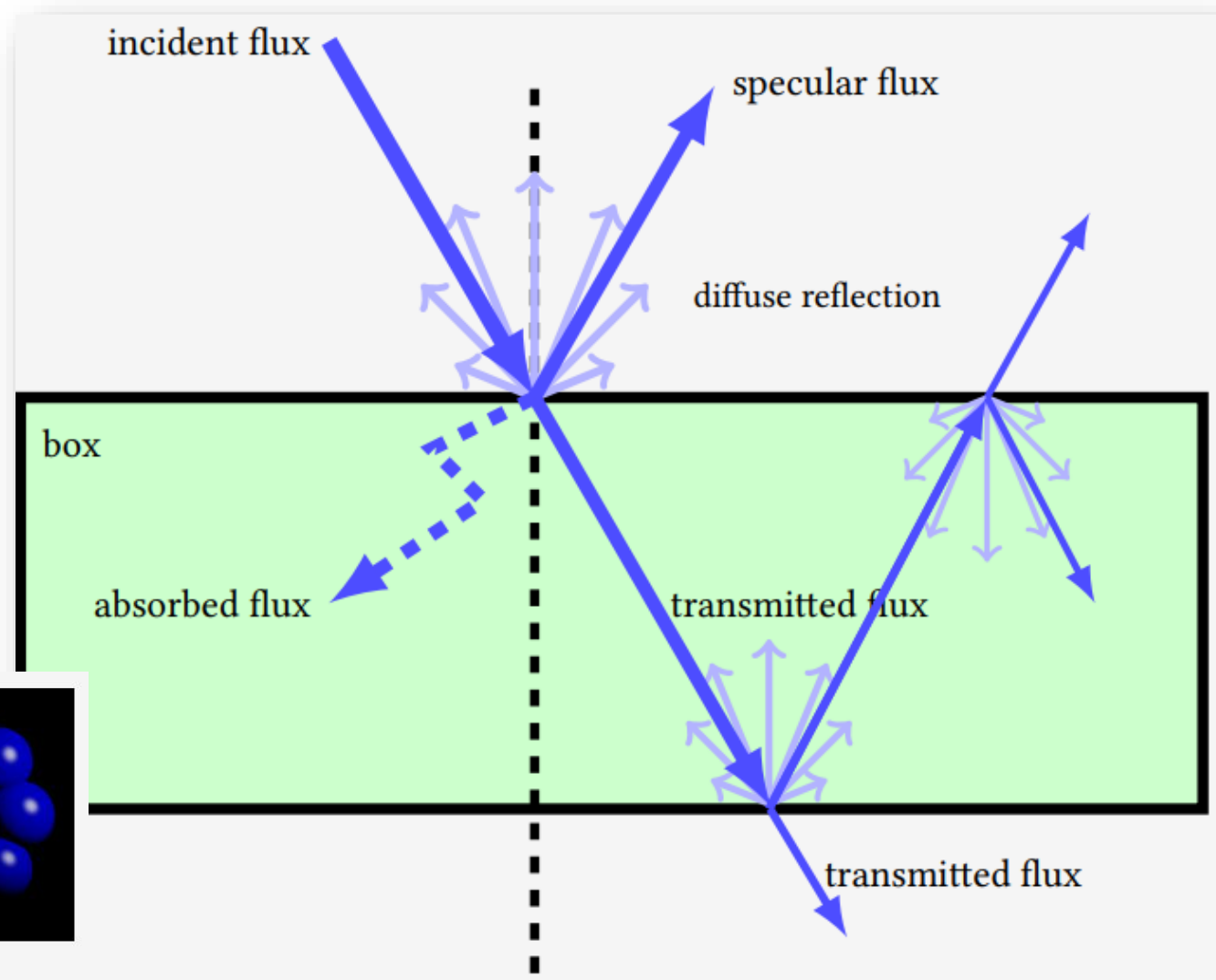
Wiki: <https://wiki.grogra.de/doku.php?id=tutorials:light-modeling-light-sources>



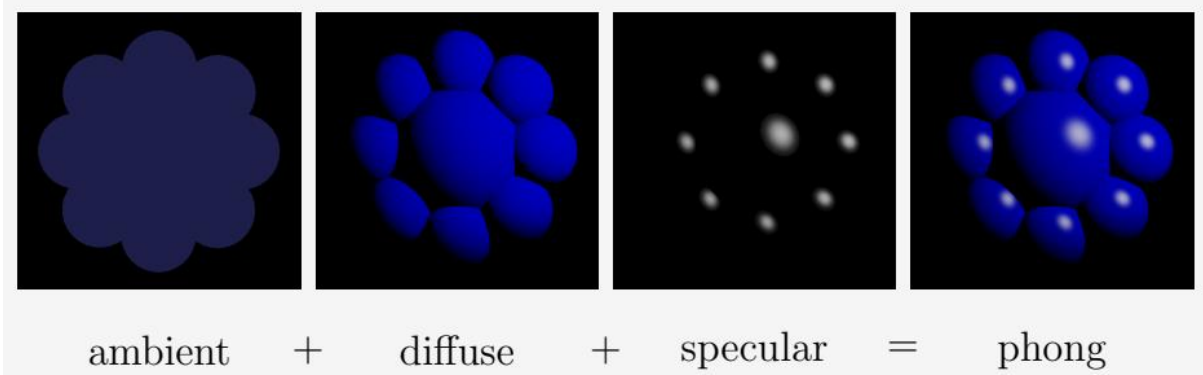
```
1  
2 protected void init() [  
3     Axiom ==>  
4     LightNode.(setLight(new PointLight().(  
5         setPower(100),  
6         setVisualize(true),  
7         setNumberofrays(400),  
8         setRaylength(0.5),  
9         setColor(new RGBColor(1,0,0))  
10    ));  
11 ]
```

Light modelling – Aspects: Local illumination models

- Definition of optical properties of objects
- In computer graphics local illumination model are called shaders
- Shaders define the amount of absorption, reflection and transmission and how the light rays are scattered



Phong reflection model, B.T. Phong, 1973



Wiki: <https://wiki.grogra.de/doku.php?id=tutorials:light-modeling-light-shader>

1. Define a Phong shader and set some properties
2. Apply the shader to a Box object
3. Note: **absorbed = received – diffuse(reflected) – transmitted**

```
static Phong myShader = new Phong();
static {
    myShader.setDiffuse(new RGBColor(0.1, 0.9, 0.0));
    myShader.setTransparency(new Graytone(0.1));
    myShader.setSpecular(new Graytone(0.1));
    myShader.setShininess(new Graytone(0.05));
}

protected void init () [
    Axiom ==> Box(0.001,1,1).(setShader(myShader));
]
```




GroIMP wiki
a part of [grogra.de](#)



[Recent Changes](#) [Media Manager](#) [Sitemap](#)

Trace: • [02_sunday](#) • [exercises_plant_like_structures_abop](#) • **[first-steps](#)**

tutorials:first-steps

First steps on light modelling

<https://wiki.grogra.de/doku.php?id=tutorials:light-modeling-first-steps>

Tutorials

Here is a non exhaustive list of examples to get started with GroIMP. There are many more examples available in the [gallery](#) and within GroIMP.

Edit

First steps

- [Use the GroIMP software \(video tutorials\)](#)
- [A Beginners Tutorial](#)
- [Running your first model](#)
- [Introduction to RGG Code structure](#)
- [XL queries and operators](#)
- [XL turtle geometry](#)
- [A step-by-step growth model](#)
- [Create a simple tomato model](#)
- [Light modelling](#)
 - [Introduction - A little bit of theory](#)
 - **[First steps](#)**
 - [Light sources](#)
 - [Local illumination - Shader](#)
 - [Sensors](#)
 - [Visualizing light rays](#)
 - [More light model functions... beyond getAbsorbedPower](#)
 - [Sensor Grids and Areal Sensors](#)
 - [Spectral light modelling](#)
 - [Effect of spectral light on morphogenesis](#)
 - [Differences between the CPU and GPU light model](#)

Tutorials

- [First steps](#)
- [Advanced tutorials](#)
- [Additional interfaces](#)
- [Developer tutorials](#)

Gallery

- [Gallery with examples](#)

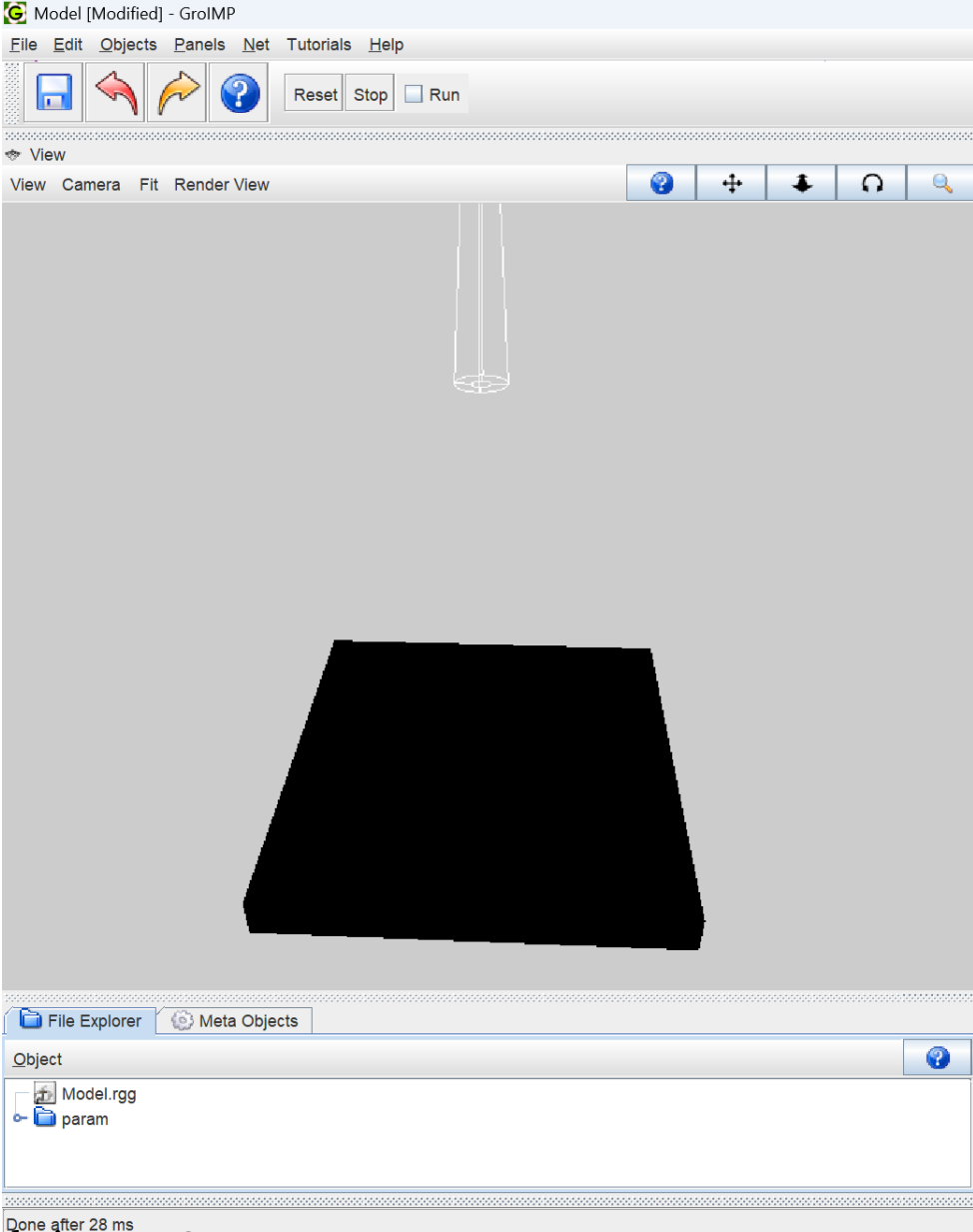
Developer Guide

- [Getting started](#)
- [GroIMP base objects impl](#)
- [GroIMP platform objects](#)
- [Interacting with GroIMP](#)
- [Plugins](#)

Maintainer Guide

- [Making a GroIMP release](#)
- [Update API doc](#)

Light modelling – Hands-on



The screenshot displays a 3D modeling environment. The main viewport shows a wireframe light source (a cylinder) and a rendered black object (a trapezoid). The interface includes a menu bar (File, Edit, Objects, Panels, Net, Tutorials, Help), a toolbar with icons for saving, undo, redo, and help, and a status bar at the bottom indicating 'Done after 28 ms'.

The right panel shows the 'jEdit - Model.rgg' editor with the following code:

```
//initialize the scene
protected void init() {
    //create the actual 3D scene
    [
        Axiom ==> Box(0.1,1,1).setShader(BLACK) M(2) RL(180)
        LightNode.(setLight(new SpotLight().setPower(100),setInnerAngle(0.02),setOuterAngle(0.055)))
    ]
    //make sure the changes on the graph are applied...
    {derive();}
    //so that we directly can continue and work on the graph

    // initialize the light model
    LightModel CPU_LM = new LightModel(RAYS, DEPTH);
    CPU_LM.compute(); // run the light model

    //check the scene objects for their light absorption
    Spectrum ms;
    [
        x:Box::> { ms = CPU_LM.getAbsorbedPower(x); }
    ]
    print("absorbed = "+ms);println(" = "+ms.integrate(), 0xff0000);
}
```

The bottom panel shows the 'Messages' and 'XL Console' tabs. The 'XL Console' displays the following output:

```
> ((* Model.MyLeaf *)).absorbedRadiation
> max((* Model.MyLeaf *)).absorbedRadiation
> Model.DiffLightSky ==> ;
> [Model.DiffLightSky ==> ;]
> [Model.Sky ==> ;]
> [Sky ==> ;]
> max((* Model.MyLeaf *)).absorbedRadiation
```

Manipulate the model

- Change the colour of the box object to, e.g., red, and see how the absorption changes

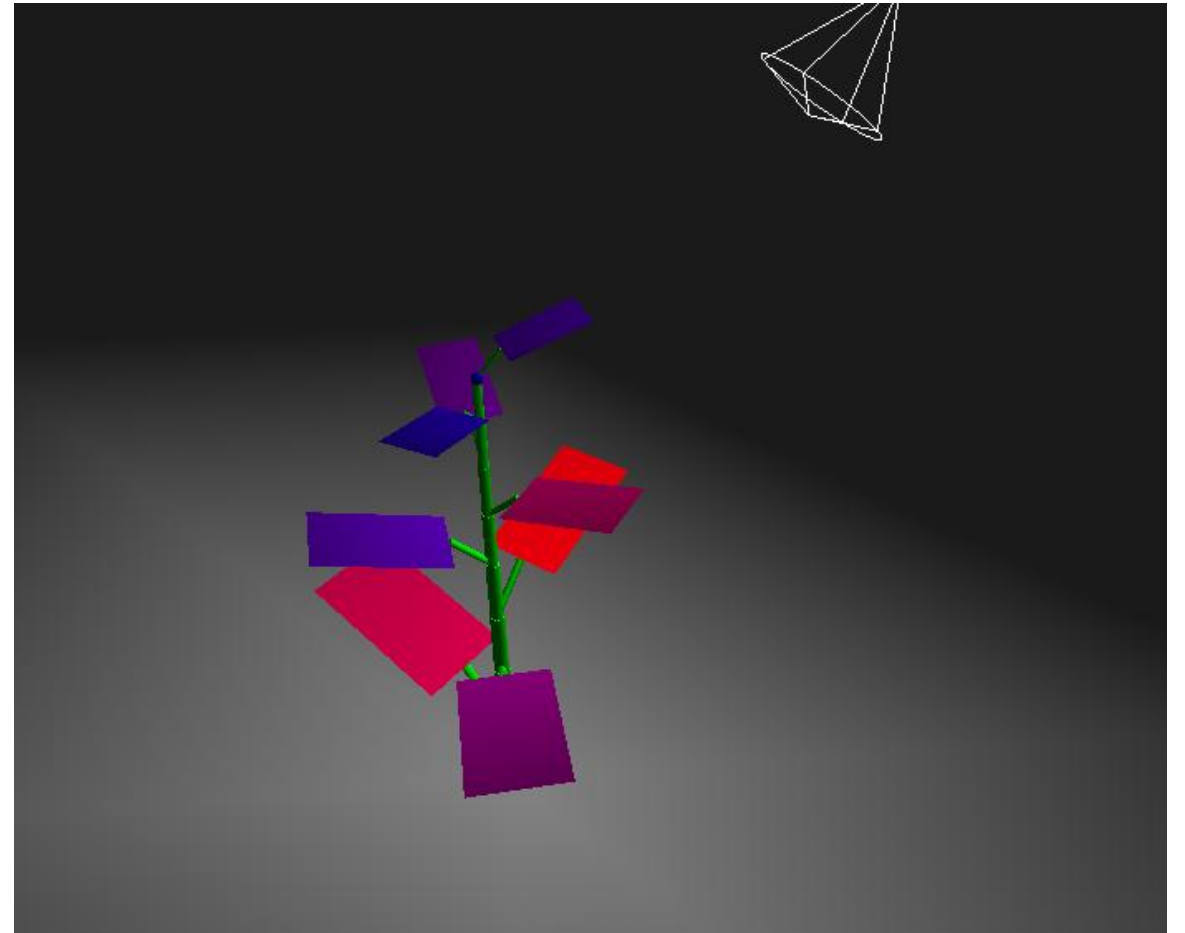
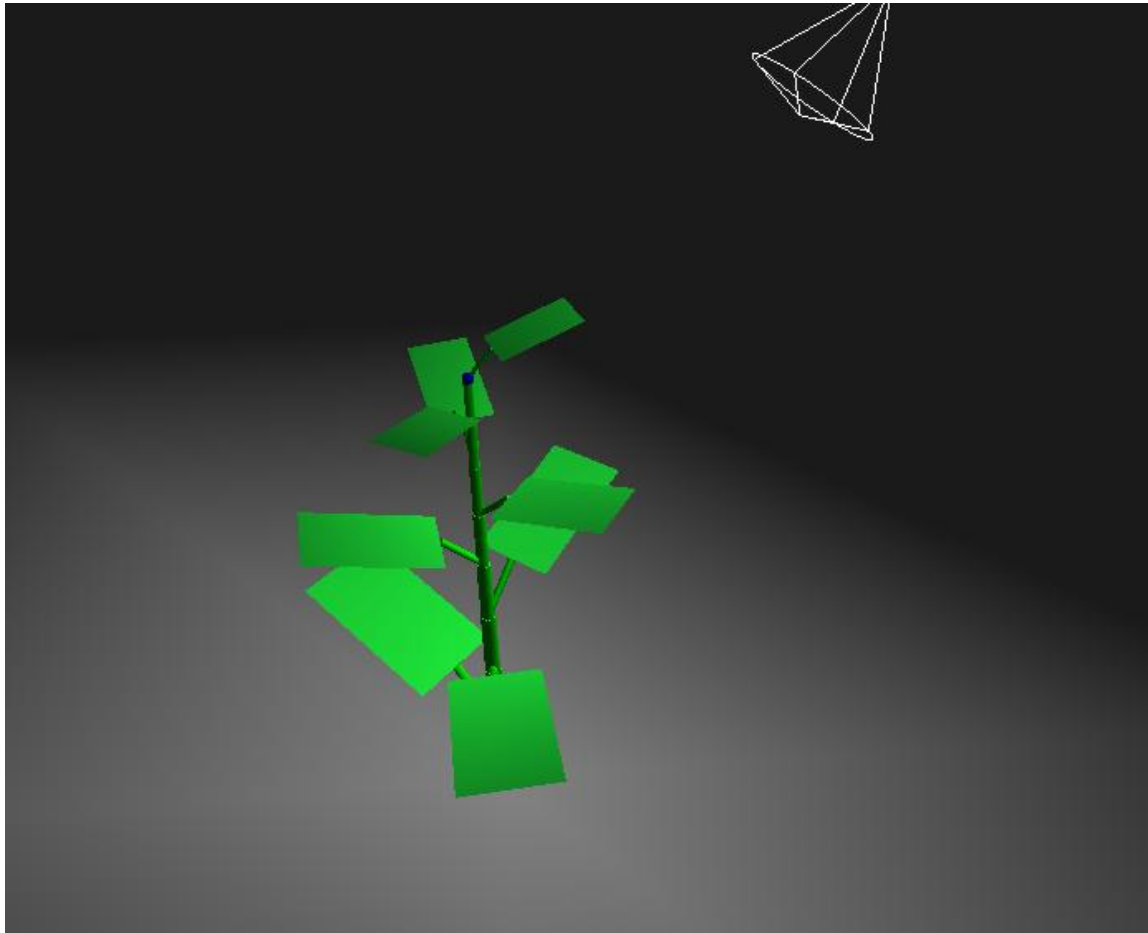
```
11 [
12   Axiom ==> Box(0.1,1,1).(setShader(RED)) M(2) RL(180)
13   LightNode.(setLight(new SpotLight().(setPower(100),setInnerAngle(0.02),
14 ]
```

- Change the colour of the box to some arbitrary colour and try to understand output.

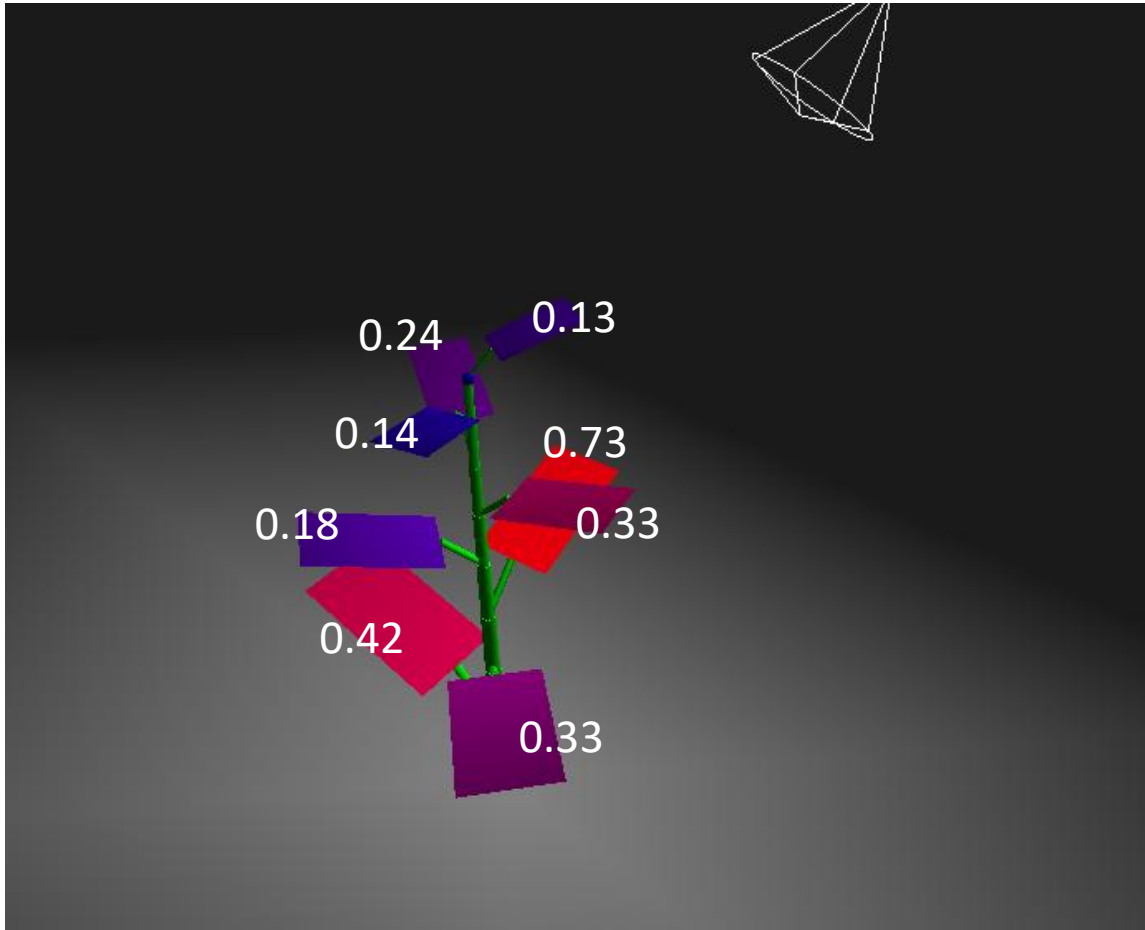
```
11 [
12   Axiom ==> Box(0.1,1,1).(setShader(new RGBAShader(0.1, 0.5, 0.3))) M(2)
13   LightNode.(setLight(new SpotLight().(setPower(100),setInnerAngle(0.02),
14 ]
```

```
absorbed = (29.99999995033074, 16.666666666665247, 23.33333293596735) = 69.99999955296333
```

Simulating light absorption at organ level



Using the output of the light modelling as input for photosynthesis estimation



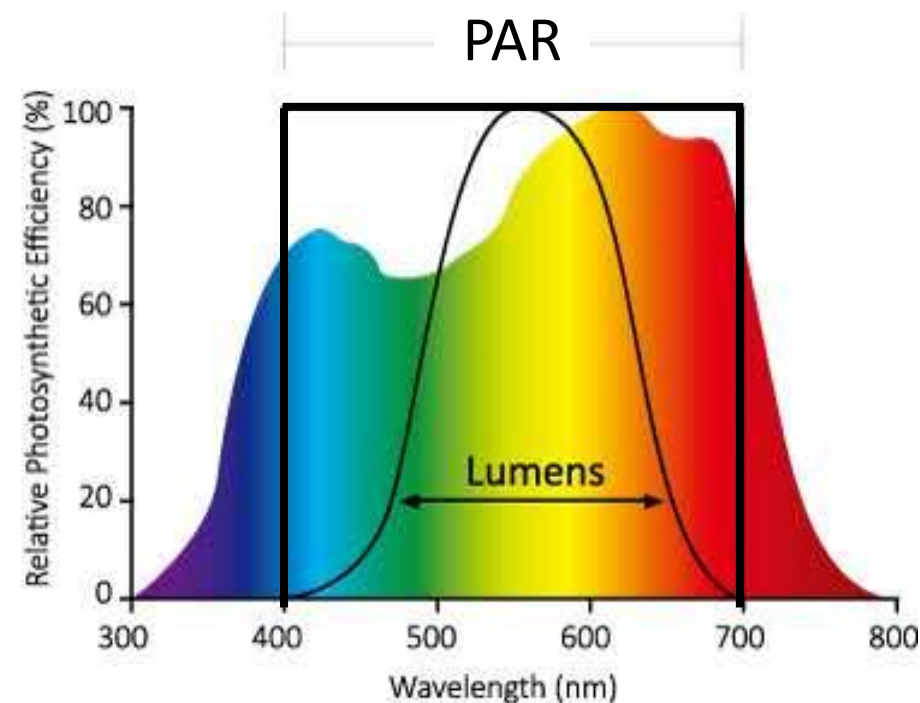
- Individual absorption values [Watt]
- Converting to PAR/PPFD

```
netPhotosynthesis = PSModel.calculatePhotosynthesis(  
    airTemp,  
    relativeHumidity,  
    CO2_ppm,  
    leafAge,  
    PPFD ← here comes the light absorption in  
);
```

PSModel ... Photosynthesis model, e.g., any FvCB implementation or simple light response curves, e.g., *KimLieth*, *Thornley*, *Baldocchi*, *Host*, *Marshall & Biscoe*, *LiehtPasian*, *JonsonThornley*, *Mueller & Echenroerder*, etc.

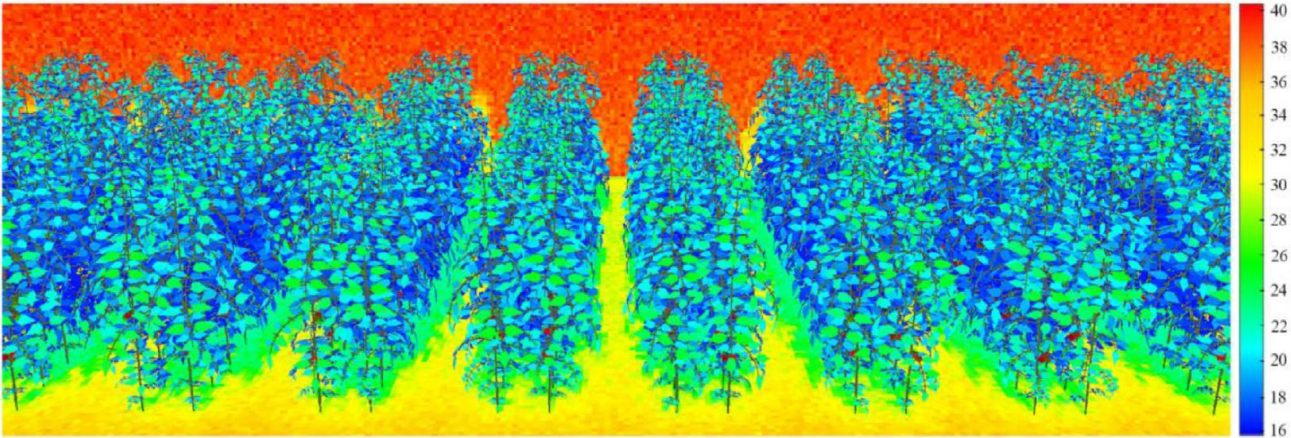
PAR, Photosynthetic Active Radiation, defines a range of wavelengths/light of the spectrum of (visible) light that plants can use for photosynthesis. PAR is NOT a measurement or “metric” like centimeter, or kilo. Typically, PAR is defined as the range of 400 to 700nm.

PPFD, Photosynthetic Photon Flux Density, is the measurement of the amount of light (how many PAR photons), actually hit a specific surface area every second. PPFD is expressed in $\mu\text{mol}/\text{m}^2/\text{s}$

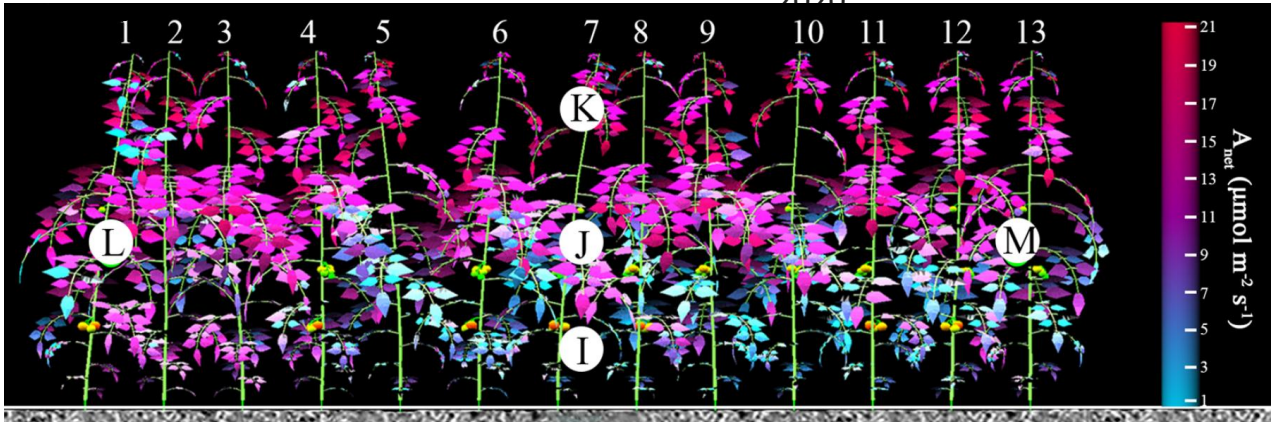


PAR is the “what kind of light”
PPFD is the “how much of it”

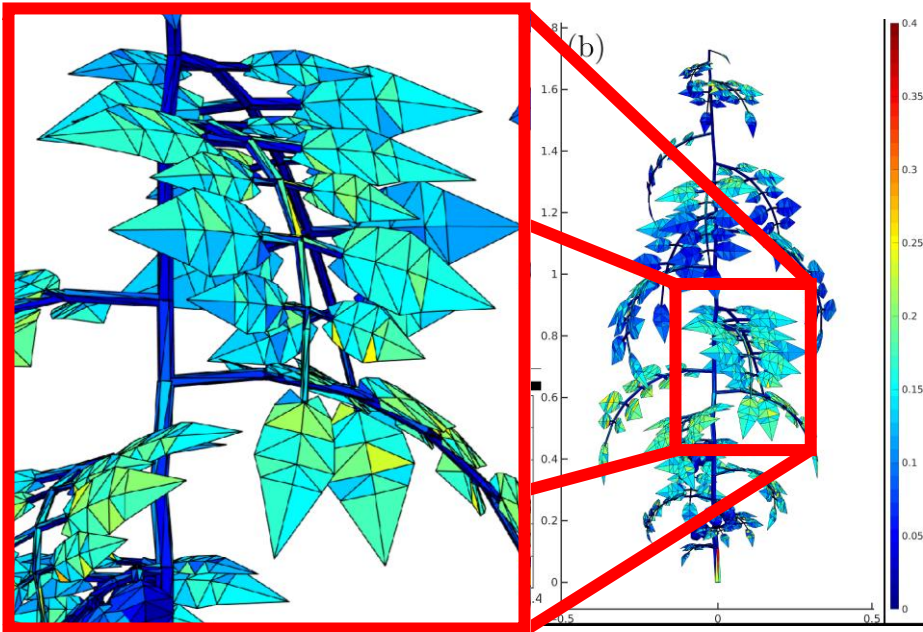
Detailed calculation of light microclimate at organ level (... and even below if needed)



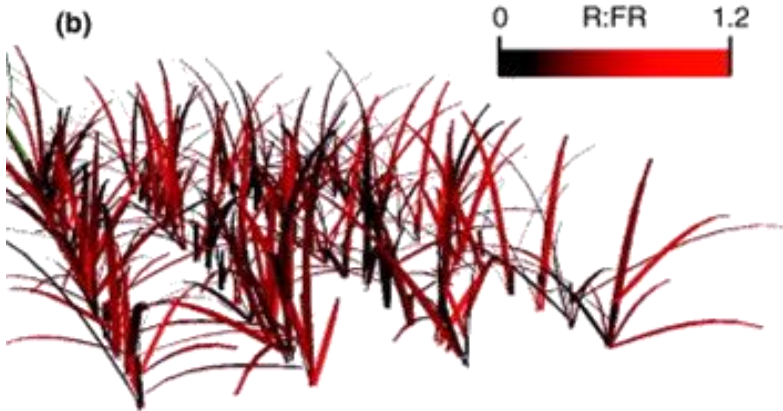
Zhang et al., Renewable Energy, 2020



Zhang et al., Agricultural and Forest Meteorology, 2021



Henke & Gladilin, Remote Sens., 2022



Evers et al., Trends in Plant Science, 2011

Series of tutorials on ‘Light interception to PS’ in the **Advanced tutorials** on the wiki pages

The screenshot displays the GrolIMP wiki interface. On the left is a sidebar with navigation links: Home, User guide (with sub-links: Preface, Installation, Using GrolIMP, User interface, 3D manipulations, Main Objects, XL and RGG language, Raytracing, Additional GrolIMP interfaces, Get/ Share models, Additional functionalities), Tutorials (with sub-links: First steps, Advanced tutorials, Additional interfaces, Developer tutorials), Gallery (with sub-link: Gallery with examples), and Developer Guide (with sub-link: Getting started). The main content area is titled 'Advanced tutorials' and contains a paragraph stating that these tutorials cover more specific features of GrolIMP, with examples available in the gallery and within GrolIMP. Below this is a list of tutorial topics. A blue rounded rectangle highlights a subset of these topics, starting from 'From light interception to photosynthesis' and ending with 'Dynamic crop model optimizing light interception and fruit yield'. The list includes: Basic to advanced XL queries, Virtual Laser Scanner for GrolIMP, How to use the ODE framework, How to create a video of a simulation, From light interception to photosynthesis (with sub-items: How to use the radiation model within a crop model: Introducing light interception of leaves, How to use the radiation model within a crop model: Plotting the absorbed light in a chart, Light interception in a dynamic crop model: Introducing leaf growth, Light interception in a dynamic crop model: Introducing absorbing tiles, Primitive photosynthesis in a dynamic crop model: RUE, Nonlinear photosynthesis in a dynamic crop model: light response curve, Dynamic crop model: flower aging and fruit formation, Dynamic crop model optimizing light interception and fruit yield), and Model fitting.

Home

User guide

- Preface
- Installation
- Using GrolIMP
 - User interface
 - 3D manipulations
 - Main Objects
 - XL and RGG language
 - Raytracing
 - Additional GrolIMP interfaces
 - Get/ Share models
 - Additional functionalities

Tutorials

- First steps
- **Advanced tutorials**
- Additional interfaces
- Developer tutorials

Gallery

- Gallery with examples

Developer Guide

- Getting started

Advanced tutorials

This tutorials cover more specific features of GrolIMP. There are many more example available in the [gallery](#) and within GrolIMP.

- Basic to advanced XL queries
- Virtual Laser Scanner for GrolIMP
- How to use the ODE framework
- How to create a video of a simulation
- From light interception to photosynthesis:
 - How to use the radiation model within a crop model: Introducing light interception of leaves
 - How to use the radiation model within a crop model: Plotting the absorbed light in a chart
 - Light interception in a dynamic crop model: Introducing leaf growth
 - Light interception in a dynamic crop model: Introducing absorbing tiles
 - Primitive photosynthesis in a dynamic crop model: RUE
 - Nonlinear photosynthesis in a dynamic crop model: light response curve
 - Dynamic crop model: flower aging and fruit formation
 - Dynamic crop model optimizing light interception and fruit yield
- Model fitting

Spectral light modelling

<https://wiki.grogra.de/doku.php?id=tutorials:basic-spectral-light-modeling>

The screenshot shows the GroIMP wiki interface. At the top, there's a logo and the text 'GroIMP wiki a part of grogra.de'. A search bar is on the right. Below the header, a breadcrumb trail reads 'Trace: • 02_sunday • exercises_plant_like_structures_abop • first-steps'. The main content area is titled 'Tutorials' and contains a list of links. A blue arrow points from the URL in the text box to the 'Spectral light modelling' link in the list. The link is highlighted with a red box.

GroIMP wiki
a part of grogra.de

Search

Recent Changes Media Manager Sitemap

Trace: • 02_sunday • exercises_plant_like_structures_abop • first-steps

tutorials:first-steps

Tutorials

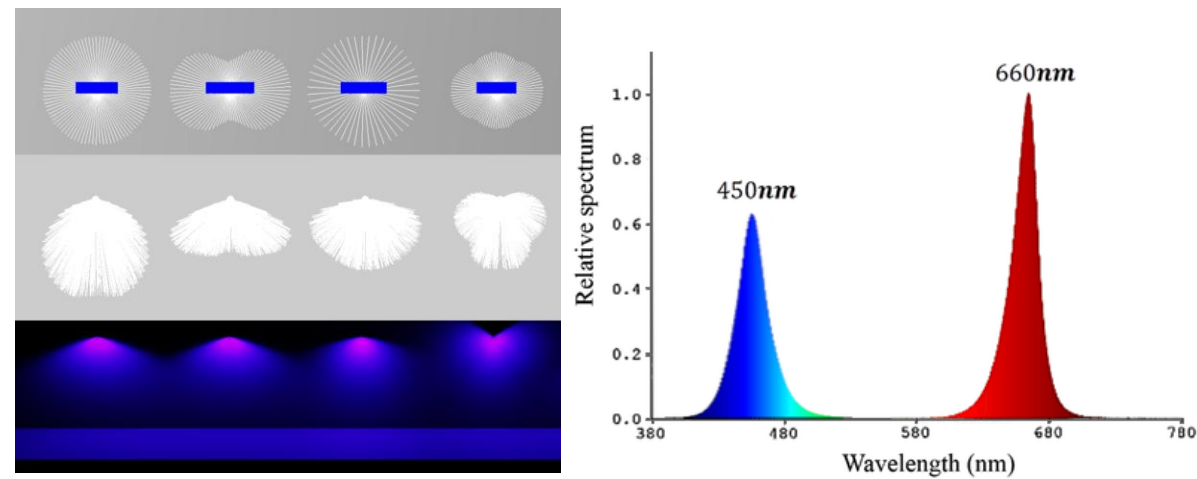
Here is a non exhaustive list of examples to get started with GroIMP. There are many more examples available in the [gallery](#) and within GroIMP.

Edit

First steps

- [Use the GroIMP software \(video tutorials\)](#)
- [A Beginners Tutorial](#)
- [Running your first model](#)
- [Introduction to RGG Code structure](#)
- [XL queries and operators](#)
- [XL turtle geometry](#)
- [A step-by-step growth model](#)
- [Create a simple tomato model](#)
- [Light modelling](#)
 - [Introduction - A little bit of theory](#)
 - [First steps](#)
 - [Light sources](#)
 - [Local illumination - Shader](#)
 - [Sensors](#)
 - [Visualizing light rays](#)
 - [More light model functions... beyond getAbsorbedPower](#)
 - [Sensor Grids and Areal Sensors](#)
 - [Spectral light modelling](#)
 - [Effect of spectral light on morphogenesis](#)
 - [Differences between the CPU and GPU light model](#)

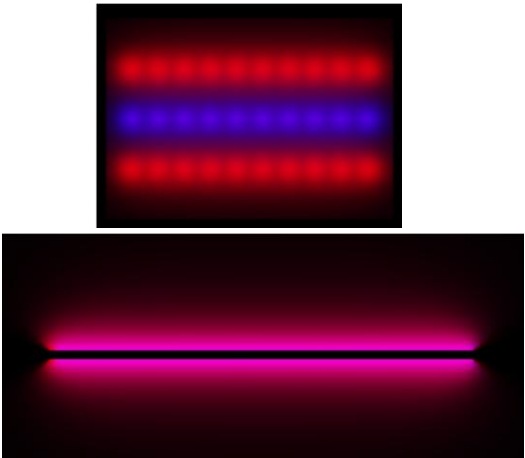
Single LED model (光珠)



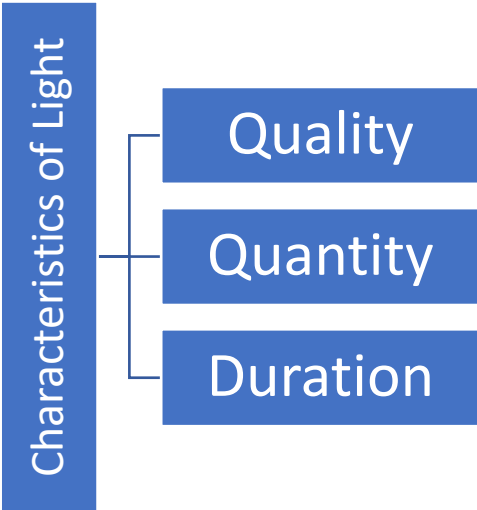
Physical Light Distribution
物理光分布

Spectral Power Distribution
光谱功率分布

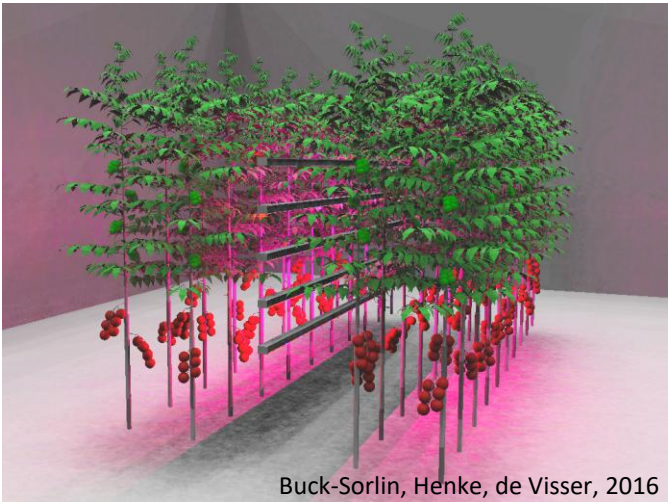
Simulated double-string LED stripes (灯条)



Simulated double-
striped LED panel



真实光照系统
LED模型



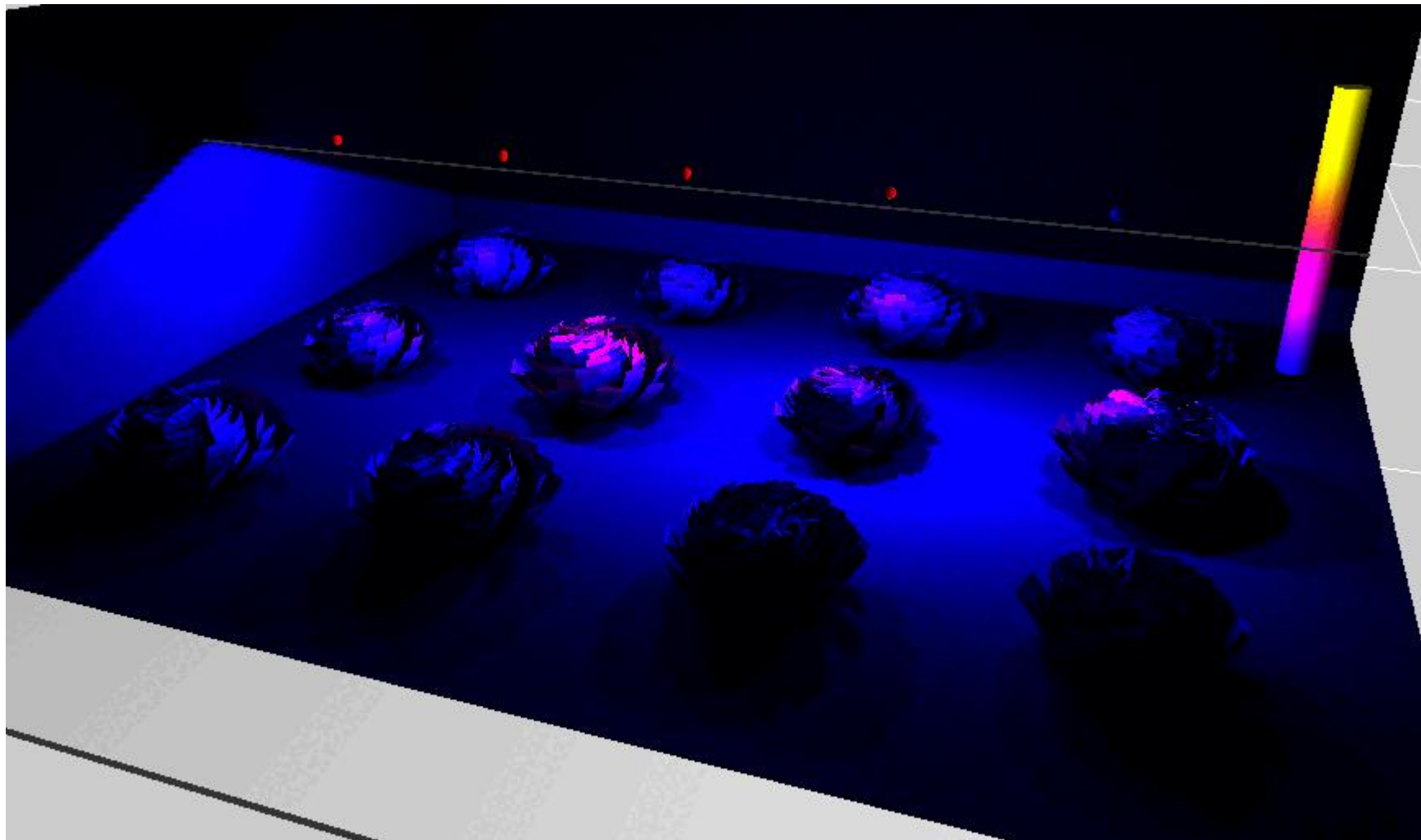
Buck-Sorlin, Henke, de Visser, 2016

Simulations of

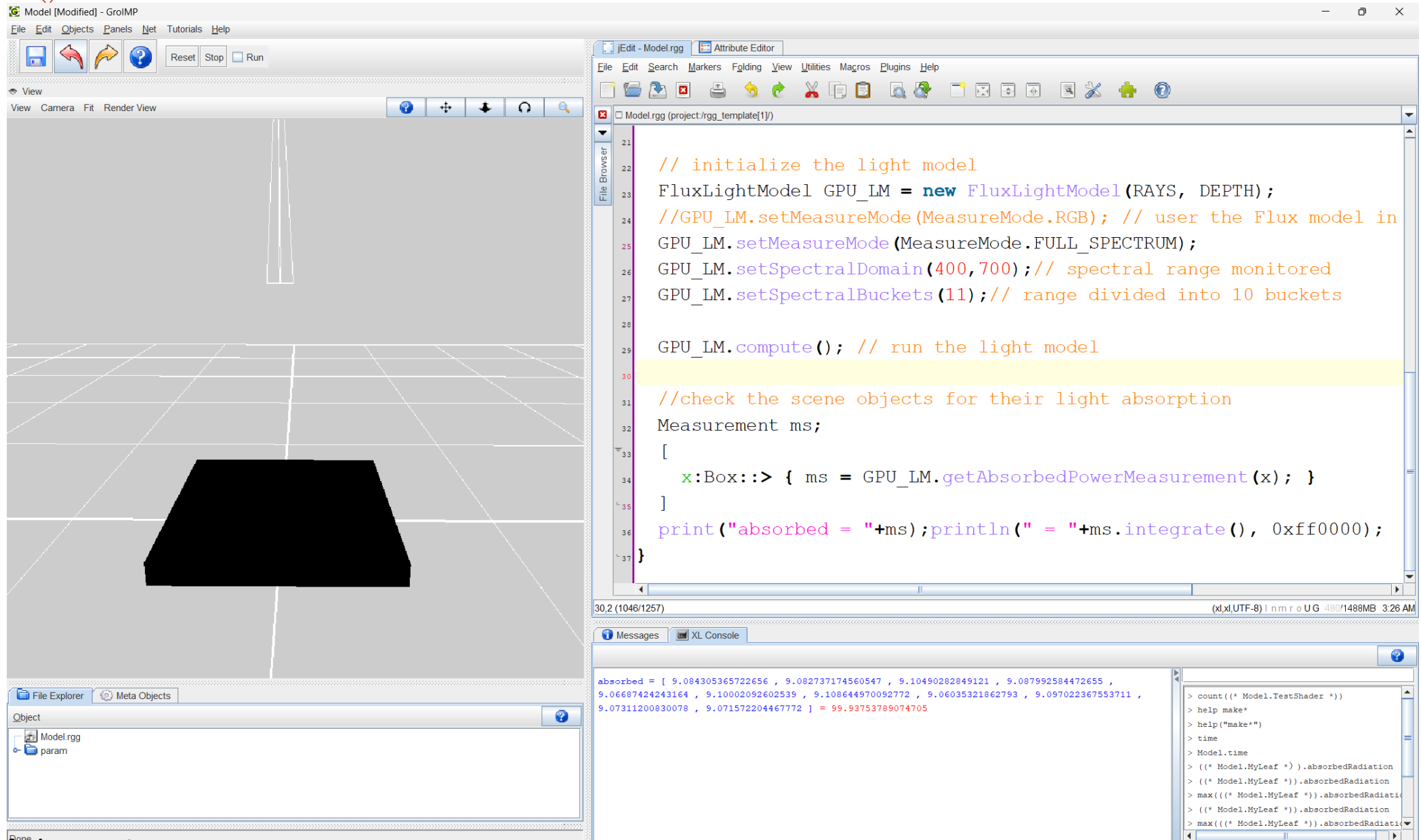
- Numbers of LEDs
灯珠数量
- Numbers of LED stripes
灯条数量
- Different PLDs
物理光分布
- Different SPDs
光谱功率分布
- Different arrangements

Obtain

- Detailed information
about light-micro climate
详细的光照微环境



Light modelling – Hands-on – spectral!



“Invisible” nodes to sense radiation without interfering with the scene

- Retuning absorption per square meter

```
import de.grogra.gpuflux.scene.experiment.Measurement;

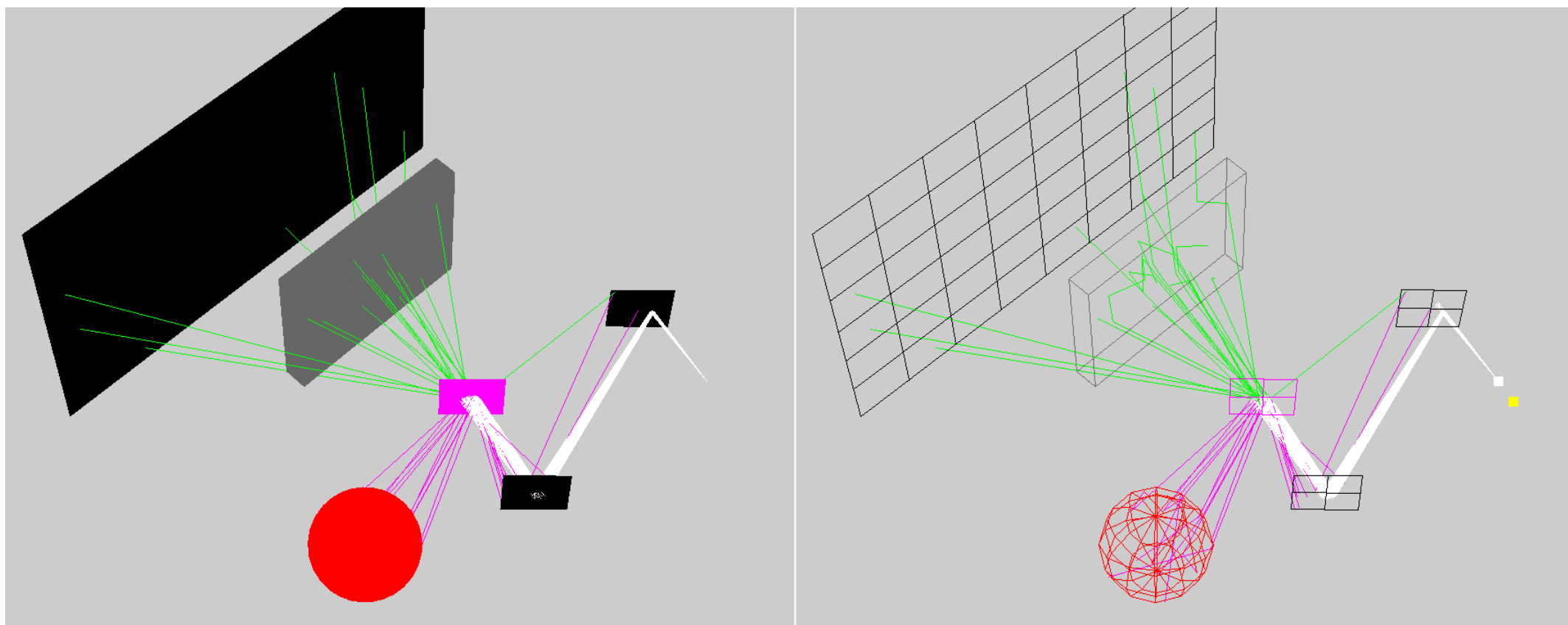
// create a 5cm, white sensor node
Axiom ==> SensorNode().(setRadius(0.05), setColor(1, 1, 1));

const FluxLightModel GPU_LM = new FluxLightModel(RAYS, DEPTH);
// enable dispersion
GPU_LM.setDispersion(true);

//check what the sensor node has sensed
x:SensorNode ::> {
    Measurement spectrum = GPU_LM.getSensedIrradianceMeasurement(x);
    float absorbedPower = spectrum.integrate();
    ...
}
```

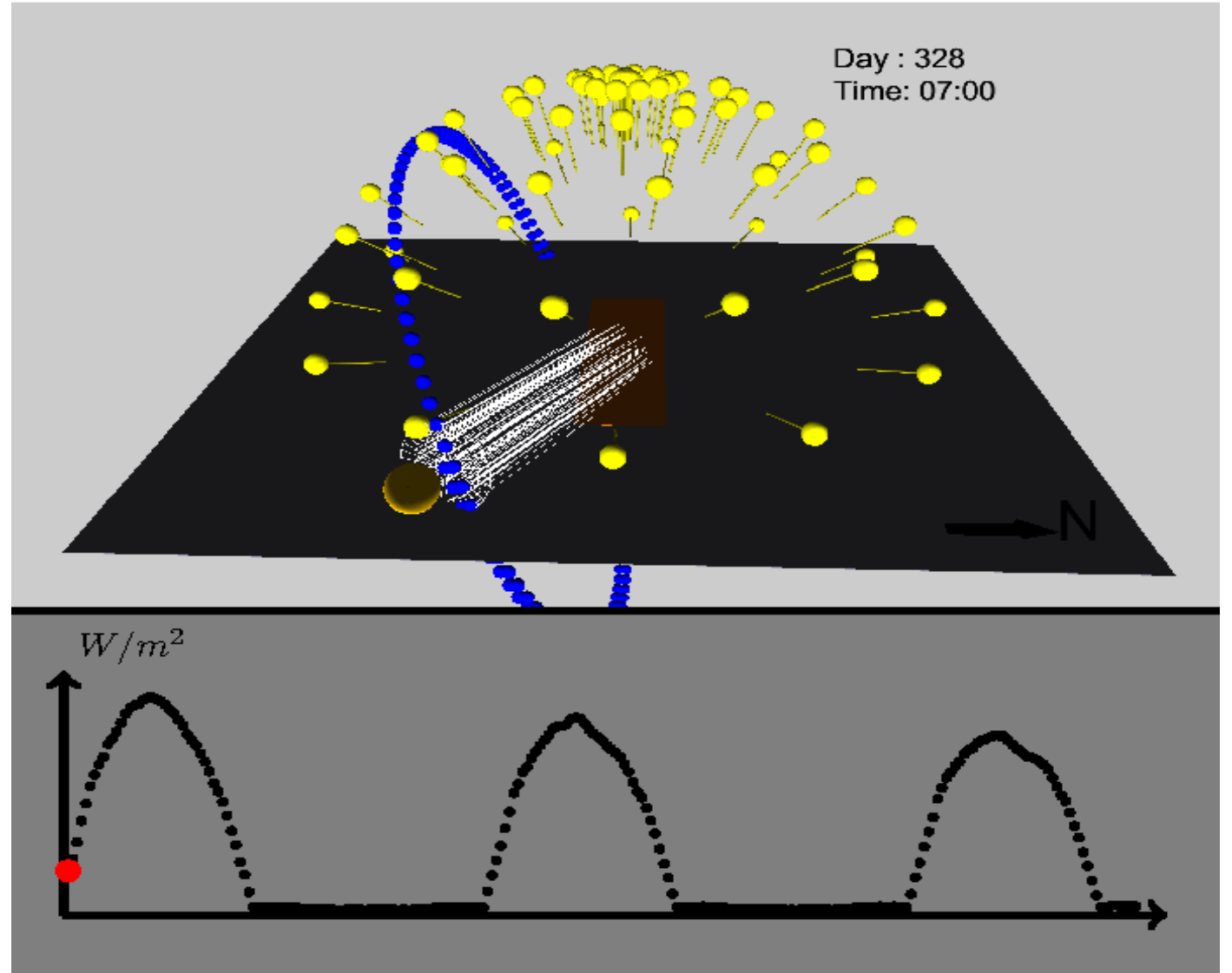
Wiki: <https://wiki.grogra.de/doku.php?id=tutorials:light-modeling-sensors>

```
protected void init () [  
    Axiom ==> LightModelVisualizer(175, 10);  
]
```



Direct and diffuse light dome

- Direct sun light
(one direct light node)
- Diffuse sky light
(hemisphere with 72 direct light nodes)



感谢观看 Thank you!

GroIMP WeChat Group



Valid until 11/5