laserCAD

He Zhuang

July 2023

Abstract

LaserCAD is an open-source, parametric software tool designed for optic simulation and visualization. Developed using the Python programming language, LaserCAD enables users to perform optical tracing and model the outcomes using FreeCAD. By incorporating fundamental principles such as the law of reflection, Snell's law, and the ABCD-Matrices , LaserCAD offers a versatile platform that supports various mutable optical elements, including lenses, mirrors, gratings, and adjustable light sources such as ray groups with diverse distributions and Gaussian beams.

The inclusion of peripheral fixation for optical elements in LaserCAD enhances the realism of the 3D modeling and brings it closer to the intricacies of a laboratory setup. Since conflicts in the settings can be found very clearly, LaserCAD provides researchers and practitioners with valuable insights and a reliable tool for experimental design.

1 Introduction

During the process of optical design, the availability of a suitable optic design software becomes imperative. However, existing optic design software often presents challenges in terms of complexity and the absence of a suitable application programming interface (API) capable of supporting scripting or parametric design. Consequently, users are compelled to invest substantial manual effort, leading to increased workload and unnecessary complications. Furthermore, the majority of optic design software lacks the provision of peripheral fixation, such as mounts and posts for optical elements, that accurately replicates real-world laboratory setups. Moreover, most available software involve precise but often computational expensive ray tracing algorithms, which are not required in most laser setups, since most lasers have a small divergence angle and small bandwidth. In addition, these software packages are often limited by specific operating systems and impose significant licensing costs, thereby increasing the entry threshold for potential users. Consequently, our research group recognizes the need to develop a new software tool that can simulate ray tracing and facilitate 3D model construction.

By addressing the aforementioned limitations, we aim to create a user-friendly platform that streamlines the optical design process and empowers researchers and practitioners with enhanced flexibility and efficiency. Since we are mainly dealing with narrow band low-divergence laser beams, our method uses the simpler and more efficient parabolic approximation.

2 Basic Objects of LaserCAD Design

The fundamental design concept behind LaserCAD revolves around the systematic classification of all essential optical objects. This is achieved by establishing a comprehensive overarching class called "Geom_Object" that serves as the primary entity for initializing and storing optical and non optical objects. The optical objects themselves can be categorized into two main groups: optical elements and light sources.

2.1 Geometric Object

The "Geom_Object" class serves as the fundamental building block for all objects within LaserCAD. Its primary purpose is to define and encapsulate the geometric information associated with each object. This includes crucial parameters such as the inner coordinate system, position, and normal vector.

Within the "Geom_Object" class, several characteristic functions have been defined to facilitate the retrieval and modification of geometric information. For instance, the "get_coordinate_system" function allows users to access the coordinate system associated with an object, providing valuable information for subsequent calculations or transformations. Similarly, the "rotate" function enables users to modify the geometric properties of an object.

2.2 Light sources

Within the realm of light sources, the fundamental constituent is the Ray, which forms the basic element. To represent more realistic beams, LaserCAD utilizes a group comprising multiple Rays. This approach allows for the flexible representation of various light source configurations.

2.2.1 Ray

The "Ray" class within LaserCAD serves to describe an 1D optical ray, encompassing important attributes such as length, position, normal vector, and wavelength. By capturing these parameters, the "Ray" class provides a comprehensive representation of the ray's characteristics. Since ray class is a basic 1D light source object. There is no width or cross section for this subject.

The 'intersection' function that Ray has will describe where it interacts with the optics. Based on this, all optical elements possess an 'next_ray' function that describes the effect they have on the incident ray. This function plays a crucial role in determining the interaction of the ray with optical elements it encounters along its path. This allows for accurate modeling of the ray's behavior as it traverses through the optical system.

2.2.2 Beam

Normally, optically designed light sources do not have a 1D beam without a radius, so a new class 'Beam' was created to simulate a more realistic 3D light source. 'Beam' is designed as a different distribution consisting of multiple rays. Taking the most basic cone distribution as an example, this beam consists of a central ray and one or more peripheral rays, with the central ray being used to define the direction of light propagation and the peripheral rays being used to define the radius and diffraction angle of the light.

At the same time, Gaussian beams will not be described in this way due to their special characteristics. The Gaussian beam will have an axis Ray describing the direction and position of its propagation, while its self-contained Q-parameter characterizes the fundamental parameters of the Gaussian beam as well as the interaction with the ABCD-matrix of the optical elements.

2.3 Optical Element

The class optical element is the basic foundation for all optical objects such as mirror and lens. The primary objective is to determine the behavior of rays after intersection. To achieve this, LaserCAD employs analytic ray tracing algorithms.

By evaluating the specific optical element that the ray intersects, the "next_ray" function calculates the resulting changes to the ray, such as refraction and reflection.

2.3.1 Mirror

For mirrors, the fundamental algorithm relies on the law of reflection. When considering 3-dimensional modeling, describing the reflection law using vectors provides a straightforward approach to connect geometric information with reflection. The relationship between the incident ray and the outgoing ray can be determined as follows: [2]

$$\mathbf{a}_2 = \mathbf{a}_1 - (\mathbf{a}_1 \cdot \mathbf{N})\mathbf{N} - sign(\mathbf{a}_1 \cdot \mathbf{N}) \times \sqrt{(\mathbf{a}_1 \cdot \mathbf{N})^2}\mathbf{N} = \mathbf{a}_1 - 2(\mathbf{a}_1 \cdot \mathbf{N})\mathbf{N}$$

In the function above, $\mathbf{a}_1$ is the incoming vector, $\mathbf{a}_2$ is the outgoing vector, and $\mathbf{N}$ is the normal vector of the mirror. The relationship between the three can be shown more clearly in the Figure 1 below.

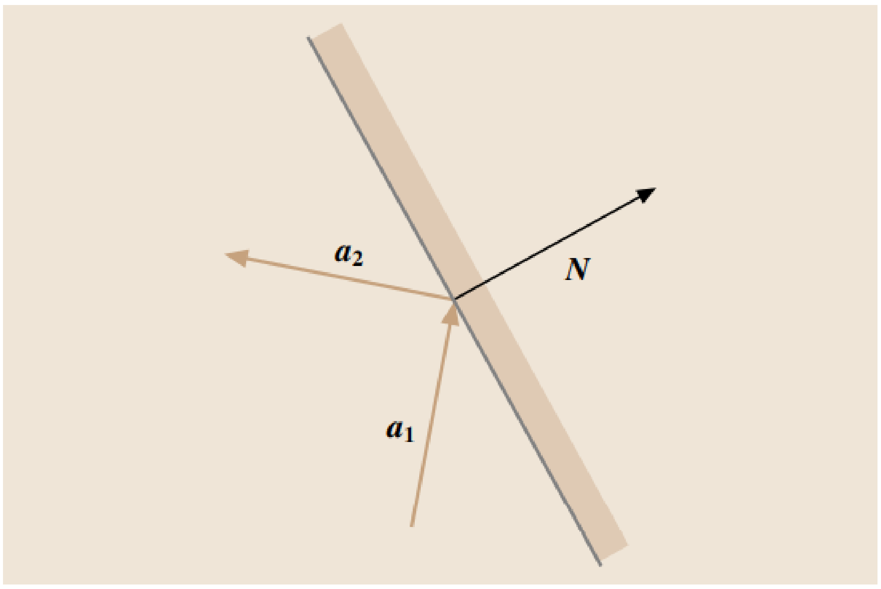

Figure 1: Parameters for reflection of a ray at a surface [2]

Additionally, LaserCAD incorporates specialized mirror types, such as curved mirrors or cylindrical mirrors. By employing geometric calculations, it becomes possible to obtain the intersection points and the equivalent normal vectors for light beams incident on mirrors with various shapes.

2.3.2 Lens

For Lenses, the well established ABCD-Matrix formalism is used to determine the refracted ray. To achieve this the 2 dimensional matrix law has to be extended to 3 dimensions. Since LaserCAD relies on the ABCD-matrix, the beam under consideration must fulfill the par-axial approximation, which are perfectly suitable for laser beams. Moreover, the ABCD-matrix proves to be a useful tool in laser design to quickly analyses optical setups. On the one hand, most lasers propagate on axes and have small radii, which qualify for the par-axial approximation. On the other hand, the ABCD-matrix helps a lot in calculating Gaussian beams. Through the ABCD-matrix of the optics, the following relationship exists for the incident Gaussian beam and the outgoing Gaussian beam:

$q_2 = \frac{Aq_1+B}{Cq_1+D}$ [1]

In the function above, A, B, C, D is the ABCD-matrix of optical elements, q_1 and q_2 is the q parameter of the input and output Gaussian beam.

2.3.3 Gratings

As for gratings, the relationship between the incident ray and the outgoing ray can be determined as follows: In the 2 dimensions case the action of a grating can be described by the grating equation

$\sin\varphi' = \sin\varphi + m\frac{\lambda}{\Lambda}$ [2]

From the equation, λ represents the wavelength of the input ray. Λ is the grating period, shows the inverse of the spatial frequency, φ' describes the angle of reflection, φ is the angle of incidence, and the integer m is the diffraction order of the grating. In the 3D case that is investigated in LaserCAD however the law has to be formulated as follows:

$\mathbf{n}_\mathrm{r} = \frac{\mathbf{k}_r}{\|\mathbf{k}_r\|} \quad \mathbf{n}_\mathrm{i} = \frac{\mathbf{k}_i}{\|\mathbf{k}_i\|}$

$\mathbf{k}_r = \mathbf{k}_\mathrm{p,out} + \|\mathbf{k}_\mathrm{n,r}\|\, \mathbf{n}_s$

$\|\mathbf{k}_{n,r}\| = \sqrt{(n_1 k_0)^2 - \|\mathbf{k}_{p,out}\|^2}$

$\mathbf{k}_{p,out} = \mathbf{k}_\mathrm{p} + m\mathbf{G} \quad \mathbf{k}_\mathrm{p} = \mathbf{n}_\mathrm{s} \times (\mathbf{k}_i \times \mathbf{n}_\mathrm{s})$

$\mathbf{G} = 2\pi\frac{\mathbf{T}_\mathrm{g}}{d} \quad \mathbf{T}_\mathrm{g} = \frac{\mathbf{n}_\mathrm{s}\times(\mathbf{T}_\mathrm{p}\times\mathbf{n}_\mathrm{s})}{\|\mathbf{n}_\mathrm{s}\times(\mathbf{T}_\mathrm{p}\times\mathbf{n}_\mathrm{s})\|}$

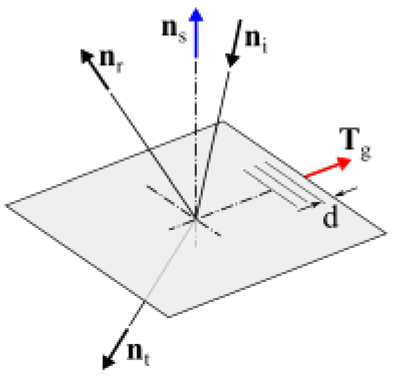

Figure 2: Parameters for reflection of gratings

In the above equation and Figure 2, $\mathbf{T}_g$ is the orientation of the raster arrangement, which is nicely demonstrated in the figure above. $\mathbf{n}_i$ is the incoming vector, $\mathbf{n}_r$ is the reflecting vector, $\mathbf{n}_s$ is the normal vector of the grating surface, $\mathbf{k}$ describes the according wave vector, and d is the grating consent.

2.3.4 Refractive planes

Furthermore, for refractive planes, LaserCAD defines the relative refractive index, as well as the position and direction of the plane. By applying Snell's law, the beam's behavior after passing through the plane can be calculated. The Snell's law can be described as:

$\frac{sin\theta_1}{sin\theta_2} = n_{21} = \frac{n_2}{n_1}$

From the equation, θ_1 is the angle of incidence, θ_2 if the angle of refraction, n_{21} is the relative refractive index, which equals to the ratio of refractive indices of the mediums on both sides of the refractive plane. And the vector expression for this law can be described as:

$\mathbf{N} \times (n_2\mathbf{a}_2 - n_1\mathbf{a}_1) = 0$ [2]

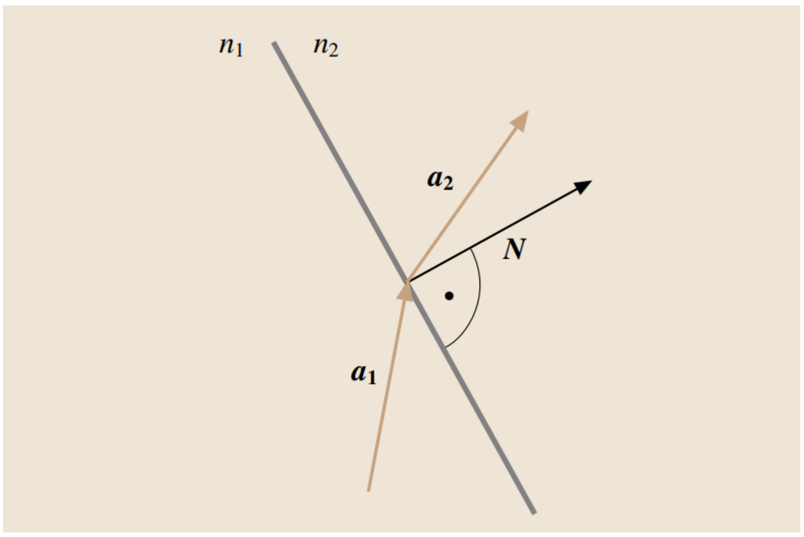

Figure 3: Parameters for refraction of a ray at a surface [2]

where n_1 and n_2 are the refractive indices of the two materials and $\mathbf{a}_1$ and $\mathbf{a}_2$ are the unit direction vectors of the incident and refracted ray, respectively. $\mathbf{N}$ is the local surface normal vector at the point of intersection of the incident ray with the surface. All the described quantities are depcited in Figure 3.

For the sake of tracing refracted rays. a_2 can be expressed in terms of $\mathbf{a}_1$ and N as:

$\mathbf{a}_2 = \frac{n_1}{n_2}\mathbf{a}_1 - \frac{n_1}{n_2}(\mathbf{a}_1 \cdot \mathbf{N})\mathbf{N} \pm \sqrt{1 - (\frac{n_1}{n_2})^2[1 - (\mathbf{a}_1 \cdot \mathbf{N})^2]}\mathbf{N}$ [2]

2.3.5 Intersection plane

The intersection plane is another important tool for imaging and analyzing results. By adding the intersection plane into the composition, the spot diagram of the output ray group can be shown.

2.4 Composition

Once the individual optical objects have been constructed within LaserCAD, it is possible to create a composition that serves as a container class and simulates different optical designs. The underlying design concept involves setting a light source as the starting point and subsequently adding various optical elements in specific arrangements. In the addition, the composition of optical elements is employed for ray tracing.

There are two primary methods for adding optical elements within LaserCAD. The first method is called "add_on_axis," which involves adding elements in succession following the optical axis which is computed and traded as a single ray. In par axial optics it corresponds to the (0,0) ray propagation. Using this function, optical elements can be added at the end of the previous ray, effectively placing them on the optical axis.

The second method is known as "add_fixed_elm," which allows for the addition of elements at user-defined positions. This method is particularly useful for incorporating optical elements that may not align perfectly with the optical axis but still need to be included in the simulation and for multi-pass systems.

Additionally, the "propagate" function is employed to characterize the propagation of the next beam. By utilizing this function, users can define the desired propagation length for the next beam, allowing for precise control over the simulated light path.

Another function "matrix" plays a key role in laser design, particularly for assessing the stability of a resonator. This function calculates the optical matrix of the entire optical system, providing valuable insights into the behavior and properties of the laser system.

By leveraging these functions and capabilities, LaserCAD enables the creation and analysis of complex optical systems, facilitating a comprehensive understanding of their performance and aiding in laser design and optimization.

2.5 Example Keppler Telescope

The following code describes the construction of a magnifying Keppler telescope to demonstrate the previous described concepts and the general workflow in LaserCAD, here is the code design:

```python
    teles = Composition(name="KepplerTelescope")
    liso = Beam(radius=3, angle=0)
    teles.set_light_source(liso)
    f1 = 100; f2 = 200
    le1 = Lens(f=f1); le2 = Lens(f=f2)
    teles.propagate(f1)
    teles.add_on_axis(le1)
    teles.propagate(f1+f2)
    teles.add_on_axis(le2)
    teles.propagate(f2)
    flip1 = Mirror(phi=65)
    teles.add_on_axis(flip1)
    teles.propagate(80)
    teles.draw()
```

Listing 1: Python code of Keppler Telescope

In the code, all length units are in mm and all angle units are in degree.

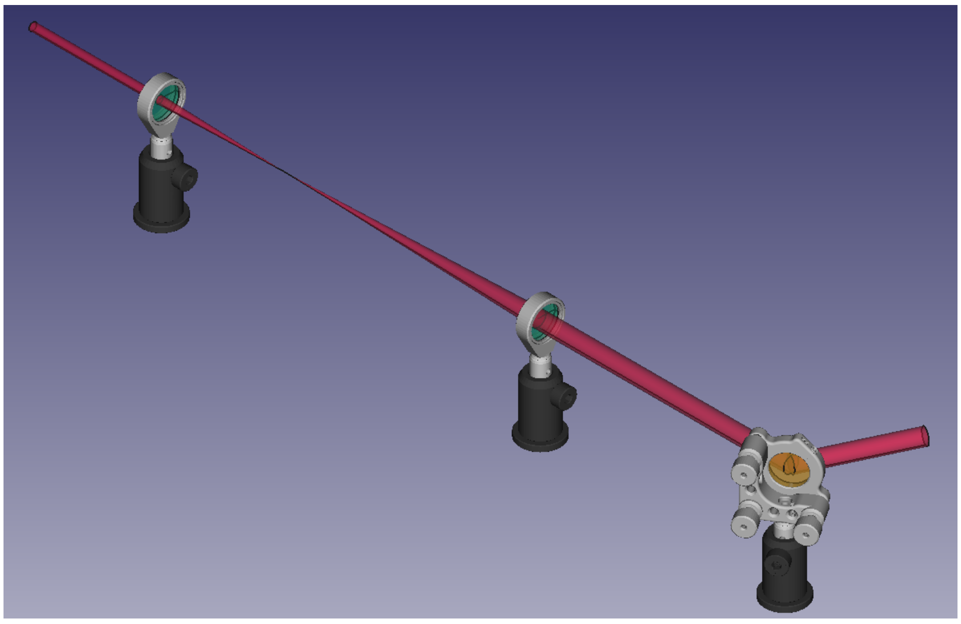
Figure 4: 3D model of the telescope example

After running the above code by FreeCAD, you can get a 3D model of the entire optical design. The Figure 2.5 is the simulation result from FreeCAD.

3 Model building

After successfully simulating the beam tracking, the results of the simulation have to be visualized. A 3D model of all the elements has to be constructed so that they can be built in FreeCAD.

3.1 The model of normal optical elements

Since FreeCAD design is based on python, it is possible to build a model through python code directly. As for rays and normal beams, they can be described as different types of cylinders and cones. As for normal beam, it is important to find out the position of its focal point. If the focal point is not in the propagation path, the model of the beam can be described as a cone or cylinder. Else, a composition of two cones to describe the focus is needed.

As for some simple elements, like plane mirrors and gratings, the modeling function can be replaced by drawing some basic structure like cylinders and cubes. However, for other elements, such as curved mirrors and lenses, using sketch to design complex elements plays a key role. Take the curved mirror as an example, a basic sketch is drawn to define the mirror's aperture and curvature. Then the revolution of the sketch constructs the model of the mirror. This can be seen in the Figure 5. Also, the colors and transparency of the models can be customized directly by code instead of setting it up in the user interface. All parameters of the drawing function are passed by a dictionary data structure called "draw_dict" of each individual element.

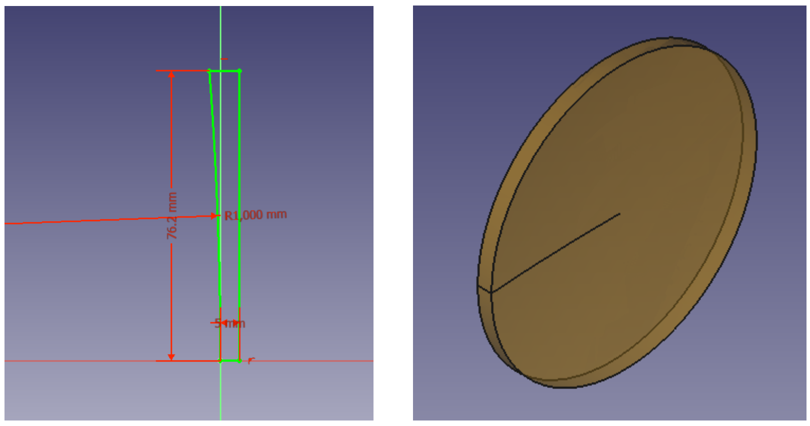

Figure 5: The construction of curved mirror. (1) The sketch of the mirror, constrained by mirror's aperture, curvature and thickness. (2) The construction of the mirror.

3.2 The model of peripheral fixation

When it comes to 3D model building software, LaserCAD leverages the capabilities of FreeCAD, which allows for easy import of various mesh and 3D files. To make the most of this feature, a database of different mounts has been created to facilitate the import of typical mount models. This is achieved through the use of the 'addObject()' and 'ImportGui.insert' functions from FreeCAD, enabling the direct import of downloaded mount models into the FreeCAD file.

For the peripheral structures like mounts and posts individual classes where defined to gather the 3D data as well as to automate the positioning of the elements.

For the Mont Class, it covers all the parameters needed to draw the Mount, including the inner diameter, model, element type and post type. Mount can be drawn in two ways, one is to enter aperture and keep 'model' as 'default'. This way Mount's function will automatically find a suitable model, or you can set your own model for Mount, so that Mount will import the defined model and ignore the aperture setting. In this case, the 'model' should be saved as the file name of the imported mount.

In most cases, small angle adjustments of the mirror in the tilt plane are allowed without rotating the mount itself. This is because the mirror can be adjusted using studs in the mount. Therefore, small angle adjustments in the z-direction are neglected.

When a new mount is imported from an external data base for the first time, it usually does not perfectly fit the mirror in terms of position and direction. Initially, all mounts need to be manually adjusted. The following Figure 6 shows the adjusting process. This process only need to be done once if a new mount was imported. After that, the adjusted mount should be saved in the correct document.

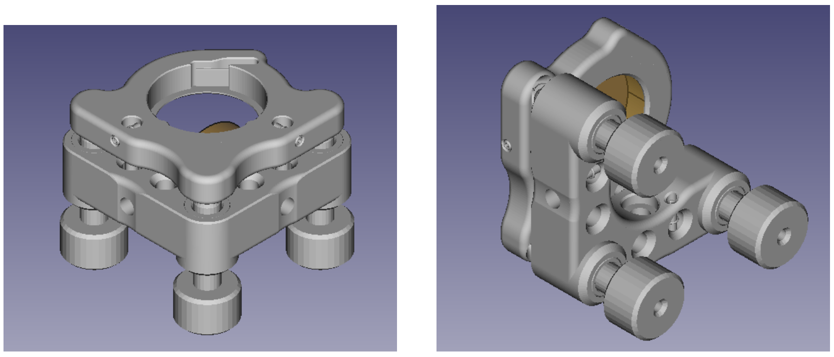

Figure 6: Comparison of mount adjustment. (1) Before adjustment. (2) After adjustment.

However, in complex compositions involving multiple mirrors, manually adjusting each mirror mount becomes impractical. To address this, a table is created to store the shift and rotation values from the original mount model to the adjusted mount model. This table can also store other parameters such as price, center height, and aperture of the mount. The 'set_by_table' function is written to load these parameters from a CSV file, which serves as the table. Once the mount is adjusted, the mesh can be saved as an adjusted mount for easier future reference.

After the mount's geometric information is settled, a new geometric object 'docking_obj' is defined to describe the docking position and docking direction of the mount. The docking_obj is used to describe the next connection place of the mount. For most cases, the docking object is used to describe where the mount and post are connected.

In some cases, special mounts, such as grating holders or rooftop mirror mounts, may be required during the optical design process. To accommodate these special mounts, a 'Special_mount' class is created to import them. After importing and adjusting, a new geometric information of docking position is defined to replace the old information, allowing for the connection between the special mount and the normal mount. With extensibility in mind, the 'load_stl' function can easily add the newly added mounts to the database as a new special mount if the mount's stl mesh file is saved in the specified location.

Typically, a mount needs to be supported by a post and post holder. To ensure proper placement, the combination of the post and post holder should be positioned on the $X-Y$ plane, which serves as the ground plane (where z=0). The selection of the post and post holder combination is determined by the height of the mirrors. The 'draw_post_part' function searches for the suitable combination and automatically draws the post in a similar manner as the mount. However, when drawing the post holder, the geometric information needs to be adjusted slightly by removing all z-direction information, as the post holder should be placed on the ground vertical plane. The combination of mirrors of different heights and their supports can be visualized very well in the Figure 7.

In some cases, a post base is required to support the post. An option is added to import the post base, and if it is needed, the post holder must be moved to ensure there is sufficient space for the base.

These functionalities within LaserCAD, utilizing FreeCAD's capabilities, enable the import and adjustment of mount models, automated drawing of posts, and the inclusion of post bases when necessary. Take the rooftop mirror as an example. This combination can be shown in the Figure 8. All perimeter fixed elements are grouped into a new part group 'mount, post and base'. From this point on it is easy to hide all elements that are not related to the beam propagation.

In summary, for optics with variable geometrical parameters, model construction requires sketching and 3D modeling from scratch using LaserCAD. For cases where the model already exists and is in use, we save the model and import it when needed.

LaserCAD is not only a software that can realize 3D modeling of op-

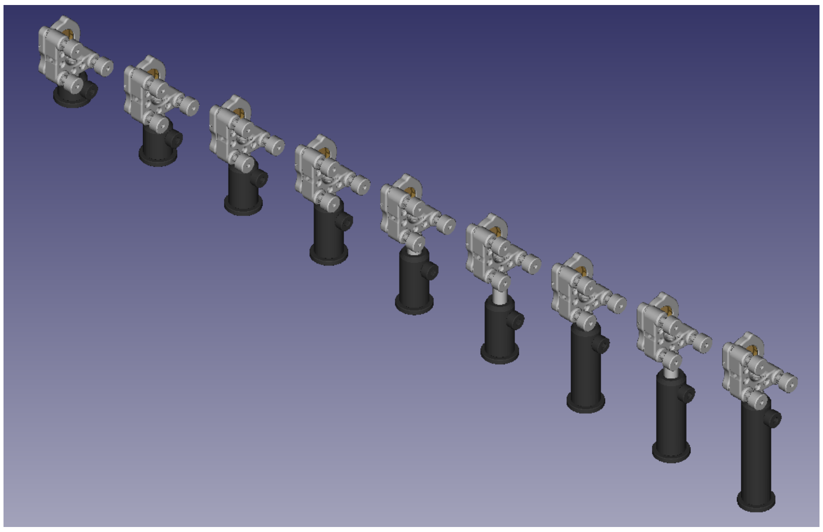

Figure 7: Fixed brackets for facing mirrors of different heights (automatic matching). The height is from 40mm to 120mm. The distance between two mirrors is 50mm. The mount, post and holder have different color, which can be characterised.

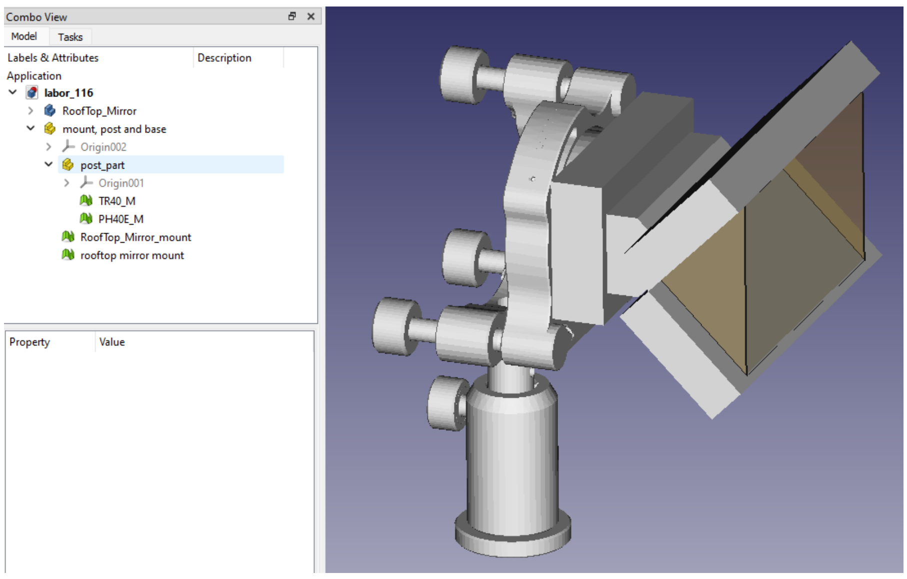

Figure 8: Rooftop mirror model. A new special mirror mount was added in front of the standard mirror mount. The special mount has its own docking position for the connection to the standard mirror mount.

tics, but also can realize ray tracing of known optical designs to a certain extent, so as to continuously optimize and improve the design. Besides, LaserCAD also retains the possibility of adding various models, which can be easily imported into other mesh files, and has good adaptability when dealing with complex and changeable optical designs. Based on this, the geometry parameters, colors and sizes of imported and drawn models can be easily read and changed. Moreover, LaserCAD's classification in different assemblies and parts makes it simple for users to high light specific objects in the FreeCAD output as well as to hide non relevant ones.

4 Limitation

While LaserCAD offers capabilities for achieving fundamental optical modeling, it is important to acknowledge certain limitations in its software design. These limitations come from specific aspects of the modeling approach employed.

Sine the LaserCAD is based on the geometric optics, all the diffraction and absorption was neglected. Besides, Laser can not show the non-linear optics result.

One limitation arises from the adoption of the paraxial approximation in ray tracing, which results in some inaccuracies when modeling lenses. The paraxial approximation assumes small angles and neglects higher-order aberrations, which can lead to deviations in the predicted behavior of rays passing through refractive optics.

Moreover, LaserCAD's implementation of optical modeling through programming may pose a challenge for users without programming skills. The absence of a dedicated user interface limits the accessibility and ease of use for individuals unfamiliar with programming concepts. As a result, it may require a certain level of programming proficiency to effectively utilize and navigate the software.

It is essential to recognize these limitations when utilizing LaserCAD, and users should consider these factors when interpreting the simulation results and assessing their applicability to real-world optical systems.

5 Outlook

The capabilities of LaserCAD, including its accurate ray tracing of mirrors, present opportunities for future developments and advancements in optical design. Here are some potential directions for further exploration:

Element expansion possibilities: because the structure of LaserCAD can add new elements easily, lots of new optics, such as cylindrical mirrors and lens, astigmatic beams and all kinds of polarization optics can be involved. At the time this article was written, some of these elements were already added.

Aberration Analysis: LaserCAD's ability to simulate and analyze aberrations caused by curved mirrors such as astigmatism allows for a deeper understanding of their impact on optical systems. Continued research and refinement in this area can lead to improved aberration correction techniques and enhanced system accuracy.

Integration with Other Software: With a comprehensive understanding of LaserCAD's underlying code, integrating and comparing it with other optical design software becomes a promising avenue for development. This interoperability can enable users to leverage the strengths of different software tools, facilitating enhanced design workflows and more robust analyses. As an example, LaserCAD can print the position and direction of all optics, which can be used as a starting point for more precise simulation tools like COMSOL.

Specialized Design: LaserCAD's flexibility is well suited to meet specific design requirements, such as those associated with fixed mounts. By extending the software's capabilities in design-specific configurations, users can address a wider range of optical design challenges and tailor solutions for specific applications.

Design Combination for Complex Systems: The ability to combine different designs in LaserCAD opens up the possibility of designing large optical systems. By seamlessly integrating and simulating interconnected components, users can gain insight into system-level behavior, optimize performance, and ensure overall design compatibility and functionality.

By pursuing these directions, LaserCAD can continue to evolve as a powerful tool in optical design, offering advanced analysis capabilities, interoperability with other software tools, support for specialized designs, and the ability to handle complex system-level simulations.

LaserCAD can construct some other alignment tools and save them as 3D models, which can be printed by 3D printers. Thus, it can help the user to align optics faster and automated.

References

[1] J. P. Taché. *Derivation of ABCD law for Laguerre-Gaussian beams.* Optical Society of America, 1987.

[2] Prof. Dr. Frank Träger. *Springer Handbook of Lasers and Optics.* Springer Dordrecht Heidelberg London New York, 2012.