image analysis script,

The code that presented in this document, might not have the exact same parameters as used in the report.

kode:

In [2]:

```
%matplotlib widget

import numpy as np
import matplotlib.pyplot as plt
import cv2
import mplcursors
from mpl_toolkits.axes_grid1 import make_axes_locatable
from PIL import Image
from skimage.draw import polygon
from matplotlib.lines import Line2D
import os
```

click handler og neutral axis

In [3]:

```
# Initialize a list to store the coordinates of the four points
testing = False
if testing:
    clicked_points = [(159, 412), (1080, 519), (1086, 739), (176, 950)]
    print("testmode")
else:
    clicked_points = []


# Function to handle mouse clicks
def click_handler(event, x, y, flags, param):
    if event == cv2.EVENT_LBUTTONDOWN and len(clicked_points) < 4:
        clicked_points.append((x, y))
        cv2.circle(param, (x, y), 5, (0, 255, 0), -1)
        cv2.putText(param, str(len(clicked_points)), (x + 5, y), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 1)
        cv2.imshow("Image", param)
```

```
def find_horizontal_neutral_axis(image):
    if len(image.shape) == 3:
        gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    else:
        gray_image = image.copy()

    row_sums = np.sum(gray_image, axis=1)
    cumulative_sums = np.cumsum(row_sums)
    neutral_axis_index = np.argmin(np.abs(cumulative_sums - cumulative_sums[-1] / 2))

    clicked_points = []

    return neutral_axis_index
```

perspective transform with mouseclicks

In [4]:

```
def perspective_transform_with_mouse_clicks(image_path, width_mm, height_mm):
    image = cv2.imread(image_path, cv2.IMREAD_UNCHANGED)
    #gamma factor to increase light, to be able to choose the crosshairs precisly
    gamma = 0.2

    # Keep a copy of the original image
    original_image = image.copy()

    # Adjust gamma
    gamma_corrected = np.power(image / 255.0, gamma) * 255.0
    image = np.uint8(gamma_corrected)

    cv2.namedWindow("Image")
    cv2.setMouseCallback("Image", click_handler, image)

    while len(clicked_points) < 4:
        cv2.imshow("Image", image)
        cv2.waitKey(1)

    # Now revert back to the original image
    image = original_image.copy()

    cv2.destroyAllWindows()
    print(f"Clicked coordinates: {clicked_points}, Testing: {testing}")


    #pixel extension to the right in the image
```

```
extend_pixels = 400

# Copy clicked_points
clicked_points_extended = clicked_points.copy()

# Extend top right point
top_vector = np.array(clicked_points[1]) - np.array(clicked_points[0])
top_vector = top_vector / np.linalg.norm(top_vector)
clicked_points_extended[1] = clicked_points[1] + extend_pixels * top_vector

# Extend bottom right point
bottom_vector = np.array(clicked_points[2]) - np.array(clicked_points[3])
bottom_vector = bottom_vector / np.linalg.norm(bottom_vector)
clicked_points_extended[2] = clicked_points[2] + extend_pixels * bottom_vector

# Calculate the reference lengths in pixels
width_ref_length_pixels = np.linalg.norm(np.array(clicked_points_extended[0]) - np.array(clicked_points_extended[1]))
height_ref_length_pixels = np.linalg.norm(np.array(clicked_points_extended[1]) - np.array(clicked_points_extended[2]))


#calculate original length of top vector
width_original_length_pixels = np.linalg.norm(np.array(clicked_points[0]) - np.array(clicked_points[1]))
#calculate original length of right vector
height_original_length_pixels = np.linalg.norm(np.array(clicked_points[1]) - np.array(clicked_points[2]))



# Calculate the DPI for both width and height
width_dpi = width_ref_length_pixels * 25.4 / width_mm
height_dpi = height_ref_length_pixels * 25.4 / height_mm

# Compute the average DPI
avg_dpi = (width_dpi + height_dpi) / 2

#factor width is extended
width_extension_factor = width_ref_length_pixels/width_original_length_pixels
#factor height is inadvertenly extended
height_extension_factor = height_ref_length_pixels/height_original_length_pixels
#print(f"{width_extension_factor}\n{width_ref_length_pixels}\n{width_original_length_pixels}")

 # Convert width and height in millimeters to pixels
width_px = int(width_extension_factor * width_mm * avg_dpi / 25.4)
height_px = int(height_extension_factor * height_mm * avg_dpi / 25.4)


# Define the destination points for the transformed image
```

```
    destination_points = np.float32([[0, 0], [width_px, 0], [width_px, height_px], [0, height_px]])


    # Calculate the perspective transform matrix and apply it
    transform_matrix = cv2.getPerspectiveTransform(np.float32(clicked_points_extended), destination_points)
    transformed_image = cv2.warpPerspective(image, transform_matrix, (width_px, height_px))

    return image, transformed_image
```

In [53]:

```
```

vizualize images

In [277…

```
def vizualize_images(input_image, transformed_image,save_path):
    '''
    takes the input image and the transformed images and plots them
    '''

    #lager en linje midt i bildet
    neutral_axis = find_horizontal_neutral_axis(transformed_image)
    visualized_image = transformed_image.copy()
    cv2.line(visualized_image, (0, neutral_axis), (visualized_image.shape[1], neutral_axis), (0, 255, 0), 2)
    # Display the images side by side using matplotlib
    fig, (ax2, ax1) = plt.subplots(1, 2, figsize=(10, 5))


    ax1.imshow(input_image, aspect="equal", cmap='gray')#, norm="linear", vmin=0,vmax=160)

    #ax1.set_title(save_path.split("/")[-1].replace("_"," ").replace("deg","\u00b0")+' input image with points')
    ax1.axis('off')


    ax2.imshow(visualized_image,aspect='equal',cmap='gray')#, norm="linear", vmin=0,vmax=160)
    #ax2.set_title(save_path.split("/")[-1].replace("_"," ").replace("deg","\u00b0")+' transformed with neutral axis')
    ax2.axis('off')

    plt.gcf().savefig(save_path+".png")

    plt.show()
```

remove background

In [6]:

```
def remove_background(image, threshold=40):
    # Set all values below threshold to zero
    image_no_background = np.where(image < threshold, 0, image)
    # Subtract the threshold from the remaining pixel intensities
    image_no_background = np.where(image_no_background > 0, image_no_background - threshold, 0)

    return image_no_background
```

Out[6]:

```
'\ndef remove_background(image, treshold = 40):\n    #sets all values below treshold to zero\n    removedBackroundImage = np.where(image < 
treshold, 0, image)\n    return removedBackroundImage\n'
```

plot intensity

In [7]:

```
def plot_intensity(image,neutral_axis):
    '''
    takes a image, and creates a intesity plot beside it
    '''



    #checks for black/white, and transforms it if not
    if len(image.shape) == 3 and image.shape[2] == 3:
        gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    else:
        gray_image = image

    accumulated_intensity = np.sum(gray_image, axis=1)

    # Normalize the accumulated intensity to the range [0, 255]
    normalized_intensity = (accumulated_intensity - np.min(accumulated_intensity)) / (np.max(accumulated_intensity) - np.min(accumulated_in
    normalized_intensity = normalized_intensity.astype(np.uint8)

    # Create an empty image with the same height as the original image and a width equal to the maximum intensity value
    intensity_image = np.zeros((image.shape[0], np.max(normalized_intensity)), dtype=np.uint8)

    # Set the intensity values along the x-axis for each row
    for row, value in enumerate(normalized_intensity):
        intensity_image[row, :value] = 255

    # Combine the original image and the intensity image
    combined_image = np.concatenate((gray_image, intensity_image), axis=1)

    return combined_image
```

```
def show_intensity_plot(combined_image,neutral_axis,save_path):
    '''
    shows a image with a given horizontal line
    '''
    # Show the combined image
    plt.figure()
     #adds a horizontal line where the intensity is the same at both sides
    plt.axhline(y=neutral_axis,linewidth=1)
    plt.imshow(combined_image, cmap='gray', aspect='auto')
    plt.xlabel("Row and Intensity")
    plt.ylabel("Column")
    plt.title(save_path.split("/")[-1].replace("_"," ").replace("deg","\u00b0")+f" intensity plot with neutral axis:{neutral_axis}")
    plt.gca().set_aspect('equal')

    plt.gcf().savefig(save_path+"_intensity.png")

    plt.show()
```

show focal energy

In [289…

```
def show_focal_energy(image, neutral_axis,image_height_mm, focal_height_mm, save_path):#,intensity_threshold = 40):
    #sum of intensyties of each row in a picture
    row_sums = [sum(row) for row in image] #list of each rows intensity
    #copy of that sum
    start_sums = row_sums.copy()
    #values for y-axis,  corresponding to the mm size of the square on the paper
    y_heights = np.linspace(image_height_mm,0,len(row_sums))
    #total intensity of the whole image
    total_intensity = sum(row_sums)
    original_intensity = sum(row_sums)

    if (image_height_mm != focal_height_mm):
        #creating the min/max indexes of the intensity list points for the focal tube
        indexes_to_keep = (len(row_sums) * (focal_height_mm/image_height_mm))
        index_to_keep_max = int(neutral_axis + indexes_to_keep//2)
        index_to_keep_min = int(neutral_axis - indexes_to_keep//2)

        #setting all values outside of interest(focal tube) to zero and deleting the values from total intensity
        for i in range(len(row_sums[0:index_to_keep_min])):
            total_intensity -= row_sums[i]
            row_sums[i] = 0
        for j in range(len(row_sums[index_to_keep_max:])):
            total_intensity -= row_sums[j+index_to_keep_max]
            row_sums[j+index_to_keep_max] = 0
```

```
     # Plotting y_heights and row_sums
    plt.figure()
    plt.plot(row_sums, y_heights,)
    plt.plot(start_sums, y_heights)

    #hline for focal size
    try:
        plt.axhline(y=y_heights[index_to_keep_min],linewidth=1,color="tab:red")
    except Exception as e1:
        print(f"An error occurred trying to create min_hline, in 'show focal energy'-function: {e1} \n the hline was set to the first index
        plt.axhline(y=y_heights[0],linewidth=1,color="tab:red")
    try:
        plt.axhline(y=y_heights[index_to_keep_max],linewidth=1,color="tab:red")
    except Exception as e2:
        print(f"An error occurred trying to create max_hline, in 'show focal energy'-function: {e2} \n the hline was set to the last index
        plt.axhline(y=y_heights[-1],linewidth=1,color="tab:red")



    plt.gca().set_aspect('auto')

    plt.ylabel('Y Heights (mm)')
    plt.xlabel('Intensity')
    plt.title(save_path.split("/")[-1].replace("_"," ").replace("deg","\u00b0")+' intensity profile along Y heights')
    custom_lines = [Line2D([0], [0], color='tab:red', lw=2),
                Line2D([0], [0], color='blue', lw=2),
                Line2D([0], [0], color='orange', lw=2)]
    plt.legend(custom_lines, [f'Focal boundaries, height: {focal_height_mm}mm', f'Altered intensity: {total_intensity:.2f}', f'Original int

    plt.gcf().savefig(save_path+"_focal.png")

    plt.show()

    return total_intensity
```

In [8]:

```
def show_focal_energy_multiple(images, neutral_axes, image_height_mm, focal_heights_mm, labels, save_path):
    plotted_focal_lower_tube = False
    plotted_focal_upper_tube = False

    # Making sure that lists have the same length
    assert len(images) == len(neutral_axes) == len(focal_heights_mm) == len(labels),  'Input lists must be of the same length.'

    # Create a figure
```

```
plt.figure()

# Define colors for different plots
colors = ['blue', 'green', 'orange', 'purple']

for (image, neutral_axis, focal_height_mm, label, color) in zip(images, neutral_axes, focal_heights_mm, labels, colors):

    row_sums = [sum(row) for row in image]
    start_sums = row_sums.copy()
    y_heights = np.linspace(image_height_mm, 0, len(row_sums))
    total_intensity = sum(row_sums)
    original_intensity = sum(row_sums)

    if image_height_mm != focal_height_mm:
        indexes_to_keep = (len(row_sums) * (focal_height_mm / image_height_mm))
        index_to_keep_max = int(neutral_axis + indexes_to_keep // 2)
        index_to_keep_min = int(neutral_axis - indexes_to_keep // 2)

        for i in range(len(row_sums[0:index_to_keep_min])):
            total_intensity -= row_sums[i]
            row_sums[i] = 0
        for j in range(len(row_sums[index_to_keep_max:])):
            total_intensity -= row_sums[j + index_to_keep_max]
            row_sums[j + index_to_keep_max] = 0

    # Plot without creating a new figure
    plt.plot(start_sums,y_heights,linestyle='dotted')
    plt.plot(row_sums, y_heights, label=f'{label}: intensity {total_intensity:.2f}')


    # Plot horizontal lines for focal boundaries
    if not plotted_focal_lower_tube:
        try:
            plt.axhline(y=y_heights[index_to_keep_min], linewidth=1, color=color, linestyle='dashed',label="Focal tube")
            plotted_focal_lower_tube = True
        except Exception as e1:
            plt.axhline(y=y_heights[0], linewidth=1, color=color, linestyle='dashed', label="Focal tube")
            plotted_focal_lower_tube = True
    if not plotted_focal_upper_tube:
        try:
            plt.axhline(y=y_heights[index_to_keep_max], linewidth=1, color=color, linestyle='dashed', label="Focal tube")
            plotted_focal_upper_tube = True
        except Exception as e2:
            plt.axhline(y=y_heights[-1], linewidth=1, color=color, linestyle='dashed', label="Focal tube")
            plotted_focal_upper_tube = True
plt.gca().set_aspect('auto')

plt.ylabel('Y Heights (mm)')
plt.xlabel('Intensity')
plt.title('Intensity profiles along Y heights')
plt.legend()

# Save the figure
plt.gcf().savefig(save_path + "_focal.png")

# Show the figure
plt.show()
```

get focal energy

In [9]:

```
def get_focal_energy(image, neutral_axis,image_height_mm, focal_height_mm):#,intensity_threshold = 40):
    #sum of intensyties of each row in a picture
    row_sums = [sum(row) for row in image] #list of each rows intensity
    #copy of that sum
    start_sums = row_sums.copy()
    #values for y-axis,  corresponding to the mm size of the square on the paper
    y_heights = np.linspace(image_height_mm,0,len(row_sums))
    #total intensity of the whole image
    total_intensity = sum(row_sums)
    original_intensity = sum(row_sums)

    if (image_height_mm != focal_height_mm):
        #creating the min/max indexes of the intensity list points for the focal tube
        indexes_to_keep = (len(row_sums) * (focal_height_mm/image_height_mm))
        index_to_keep_max = int(neutral_axis + indexes_to_keep//2)
        index_to_keep_min = int(neutral_axis - indexes_to_keep//2)

        #setting all values outside of interest(focal tube) to zero and deleting the values from total intensity
        for i in range(len(row_sums[0:index_to_keep_min])):
            total_intensity -= row_sums[i]
            row_sums[i] = 0
        for j in range(len(row_sums[index_to_keep_max:])):
            total_intensity -= row_sums[j+index_to_keep_max]
            row_sums[j+index_to_keep_max] = 0

    return total_intensity, original_intensity
```

flux width plot

In [70]:

```python
def flux_width_plot(images, neutral_axes, legend_names, focal_widths, image_height_mm, reference_Flux, save_string):
    plotted_vline = False
    fig, ax = plt.subplots()
    # Loop through each image, neutral axis, and focal width
    for image, neutral_axis, legend_name, focal_width in zip(images, neutral_axes, legend_names, focal_widths):
        # Sum of intensities of each row in the image
        row_sums = np.array([np.sum(row) for row in image])
        # Calculate height of each row in mm
        row_Height_mm = image_height_mm / len(row_sums)

        # Initialize lists to store flux and width values
        flux = []
        width = []
        # Start at the neutral axis
        middle_index = neutral_axis

        flux.append(row_sums[neutral_axis])
        width.append(row_Height_mm)

        for i in range(1, min(neutral_axis, len(row_sums) - neutral_axis)):
            # Adding sum of corresponding row intensities from the neutral axis
            flux.append(flux[i-1] + row_sums[neutral_axis-i] + row_sums[neutral_axis+i])
            width.append(width[i-1] + 2 * row_Height_mm)

        flux_normalized = 100 * (np.array(flux) / reference_Flux)

        # Plot data on the subplot
        ax.plot(width, flux_normalized, label=legend_name)

        # Plot vertical line for focal width
        if not plotted_vline:
            ax.axvline(x=focal_width, color="tab:red", linestyle='dashed')
            plotted_vline = True
        else:
            ax.axvline(x=focal_width, color="tab:red", linestyle='dashed')

    ax.set_xlabel('Width (mm)')
    ax.set_ylabel('Flux % of reference')
    ax.set_title(save_string.split("/")[-1].replace("_", " ").replace("deg", "\u00b0").replace("percent", "%") + ', Flux vs. Width')
    ax.grid(True)
    ax.legend(loc='lower right')
    #fig.savefig(save_string+"_flux.png")
    for axes in axes_list:
        for line, label in zip(axes.lines, axes.get_legend_handles_labels()[1]):
            ax.plot(line.get_xdata(), line.get_ydata(), label=label)

    plt.show()
    return ax
```

plot multiple axes

In [73]:

```
ax_to_plot = {}
```

In [237…

```
#"name":ax
def add_ax_to_plot(key,ax):
    ax_to_plot[key] = ax
    print("dictionary: ",ax_to_plot)

def plot_all_axis():
    label_list = []

    fig,ax =plt.subplots(figsize=(9,7))
    for key, value in ax_to_plot.items():
        for line, label in zip(value.lines, value.get_legend_handles_labels()[1]):
            linestyle = "solid"
            if key in ["simulation","ref"]:
                linestyle = "dashed"
            if label in label_list:
                ax.plot(line.get_xdata(), line.get_ydata())
            else:
                ax.plot(line.get_xdata(), line.get_ydata(), label=label,linestyle=linestyle)
            label_list.append(label)

    #plot focaltube
    ax.axvline(x=12.45, color="tab:red", linestyle='dotted', label="Focal tube")

    #plot settings
    ax.set_xlabel('Width (mm)')
    ax.set_ylabel('Flux % of reference')
    ax.set_title("Flux / Width, Intensity plot")
    ax.grid(True)

    #sort legend
    handles, labels = ax.get_legend_handles_labels()
    labels, handles = zip(*sorted(zip(labels, handles), key=lambda t: t[0]))
    ax.legend(handles, labels,loc='center left', bbox_to_anchor=(0.6, 0.2))
    fig.savefig("ImageAnalysis/transmission_gathered")
    ax.grid(True)
    plt.show()
```

output comparison

In [291…

```
loss_factor_simulated = 0.79

def output_comparison(etendue_path,reference_path,save_path,background_etendue,background_reference,legend_name="etendue"):
    squareWidth = 96 #mm
    squareHeight = 60 #mm
    focalTube = 12.45

    etendue_save_path = save_path+"_Etendue"
    reference_save_path = save_path+"_Reference"

    #transformerer bildet
    reference_image, transformed_reference_image = perspective_transform_with_mouse_clicks(reference_path, squareWidth, squareHeight)
    etendue_image, transformed_etendue_image = perspective_transform_with_mouse_clicks(etendue_path, squareWidth, squareHeight)


    #viser input bildet og transformert bilde med nøytralakse
    vizualize_images(etendue_image,transformed_etendue_image,etendue_save_path)
    vizualize_images(reference_image,transformed_reference_image, reference_save_path)


    #fjerner bakgrunn, kan legge in threshpold som 2. argument
    etendue_removed_background = remove_background(transformed_etendue_image,background_etendue)
    reference_removed_background = remove_background(transformed_reference_image,background_reference)

    #neutral axis:
    etendue_neutral_axis = find_horizontal_neutral_axis(etendue_removed_background)
    reference_neutral_axis = find_horizontal_neutral_axis(reference_removed_background)

    # Plot the accumulated intensity of each row in the transformed image
    etendue_intensityplot = plot_intensity(etendue_removed_background, etendue_neutral_axis)
    reference_intensityplot = plot_intensity(reference_removed_background, reference_neutral_axis)

    #neutral axis:
    etendue_neutral_axis = find_horizontal_neutral_axis(etendue_removed_background)
    reference_neutral_axis = find_horizontal_neutral_axis(reference_removed_background)

    #viser intentsitetsplottet
    show_intensity_plot(etendue_intensityplot,etendue_neutral_axis,etendue_save_path)
    show_intensity_plot(reference_intensityplot,reference_neutral_axis,reference_save_path)

    #calulating flux
    etendue_flux, original_etendue_flux = get_focal_energy(etendue_removed_background, etendue_neutral_axis, squareHeight, focalTube)#show_
    reference_flux, original_reference_flux = get_focal_energy(reference_removed_background, reference_neutral_axis, squareHeight, squareHe

    reference_flux = show_focal_energy(reference_removed_background, reference_neutral_axis, squareHeight, squareHeight,reference_save_path

    #showing focal energy multiple plots
    #show_focal_energy_multiple([etendue_removed_background,reference_removed_background], [etendue_neutral_axis,reference_neutral_axis], s
    #def show_focal_energy_multiple(images, neutral_axes, image_height_mm, focal_heights_mm,labels, save_path):

    transmission = (etendue_flux/reference_flux)*100
    print(f"referece flux = {reference_flux}\n etendue flux = {etendue_flux}\n  transmission = {int(transmission)}%" )

    reference_with_loss = [[x*loss_factor_simulated for x in y] for y in reference_removed_background]



    #flux_width_plot([etendue_removed_background,reference_removed_background,reference_with_loss],[etendue_neutral_axis,reference_neutral_

    #create two ax objects to add to ax:to_plot dict
    '''
    ax = flux_width_plot([etendue_removed_background],[etendue_neutral_axis],[legend_name+", "+str(int(transmission))+"%"], [focalTube], sq
    ax_ref = flux_width_plot([reference_removed_background],[reference_neutral_axis],["Reference"], [focalTube], squareHeight, reference_fl
    ax_ref_loss = flux_width_plot([reference_with_loss],[reference_neutral_axis],["reference fresnel reduced"], [focalTube], squareHeight,
    '''
    #add the  two plots to the dictionary
    #add_ax_to_plot(legend_name,ax)
    #add_ax_to_plot("ref",ax_ref)
```

Virtual model

In [167…

```
def get_x_within_interval(coordinates, y_min, y_max):
    x_values = []
    x_values_total = []
    for coord in coordinates:
        x, y = coord[0], coord[1]
        x_values_total.append(x)
        if y_min <= y <= y_max:
            x_values.append(x)
    focal_intensity = sum(x_values)
    total_intensity = sum(x_values_total)
    return focal_intensity, total_intensity
```

In [168…

```
def extract_data(path):
    data = []
    with open(path, 'r') as file:
        for line in file:
            line = line.strip()  # Remove leading/trailing whitespace
            if line:  # Skip empty lines
                values = line.split()  # Split the line by whitespace
                data.append([float(value) for value in values])  # Convert values to floats and append to the data list

    # Extract x and y values from the data
    y = [row[0] for row in data]
    x = [row[1] for row in data]

    swapped_data = [[coord[1], coord[0]] for coord in data]

    return x, y, swapped_data
```

In [233…

```
def nice_string(s):
    parts = s.split()

    if len(parts) >= 3:
        s0 = parts[0].ljust(11)
        s1 = parts[1].ljust(6)
        s2 = parts[2].ljust(3)

        formatted_string = s0 + s1 + s2
        return (formatted_string)
    else:
        print("Input string does not contain enough parts.")
        return s
```

In [231…

```
def flux_width_plot_virtual(images, neutral_axes, legend_names, focal_widths, image_height_mm, reference_Flux):
    plotted_vline = False
    fig, ax = plt.subplots()

    # Loop through each image, neutral axis, and focal width
    for image, neutral_axis, legend_name, focal_width in zip(images, neutral_axes, legend_names, focal_widths):
        # Sum of intensities of each row in the image
        row_sums = np.array([c[0] for c in image])
        # Calculate height of each row in mm
        row_Height_mm = image_height_mm / len(row_sums)

        # Initialize lists to store flux and width values
        flux = []
        width = []
        # Start at the neutral axis
        middle_index = neutral_axis

        flux.append(row_sums[neutral_axis])
        width.append(row_Height_mm)
        print(f"neutral axis: {neutral_axis} len row_sums: {len(row_sums)}")

        for i in range(1, neutral_axis-1):
            # Adding sum of corresponding row intensities from the neutral axis
            flux.append(flux[i-1] + row_sums[neutral_axis-i] + row_sums[neutral_axis+i])
            width.append(width[i-1] + 2 * row_Height_mm)


        flux_normalized = 100 * (np.array(flux) / reference_Flux)

        # Plot data on the subplot
        ax.plot(width, flux_normalized, label=legend_name,linestyle='dotted')

        '''
        # Plot vertical line for focal width
        if not plotted_vline:
            ax.axvline(x=focal_width, color="tab:red", linestyle='dashed', label="Focal tube")
            plotted_vline = True
        else:
            ax.axvline(x=focal_width, color="tab:red", linestyle='dashed')
        '''

    ax.set_xlabel('Width (mm)')
    ax.set_ylabel('Flux % of reference')
    ax.grid(True)
    ax.legend(loc='lower right')
    ax.set_title("0\u00b0 Simulation, transmission 79%, Flux vs. Width")
    plt.savefig(os.getcwd()+"\\"+"zemax_0deg_transmission.png")
    plt.show()
    return ax
```

In [284…

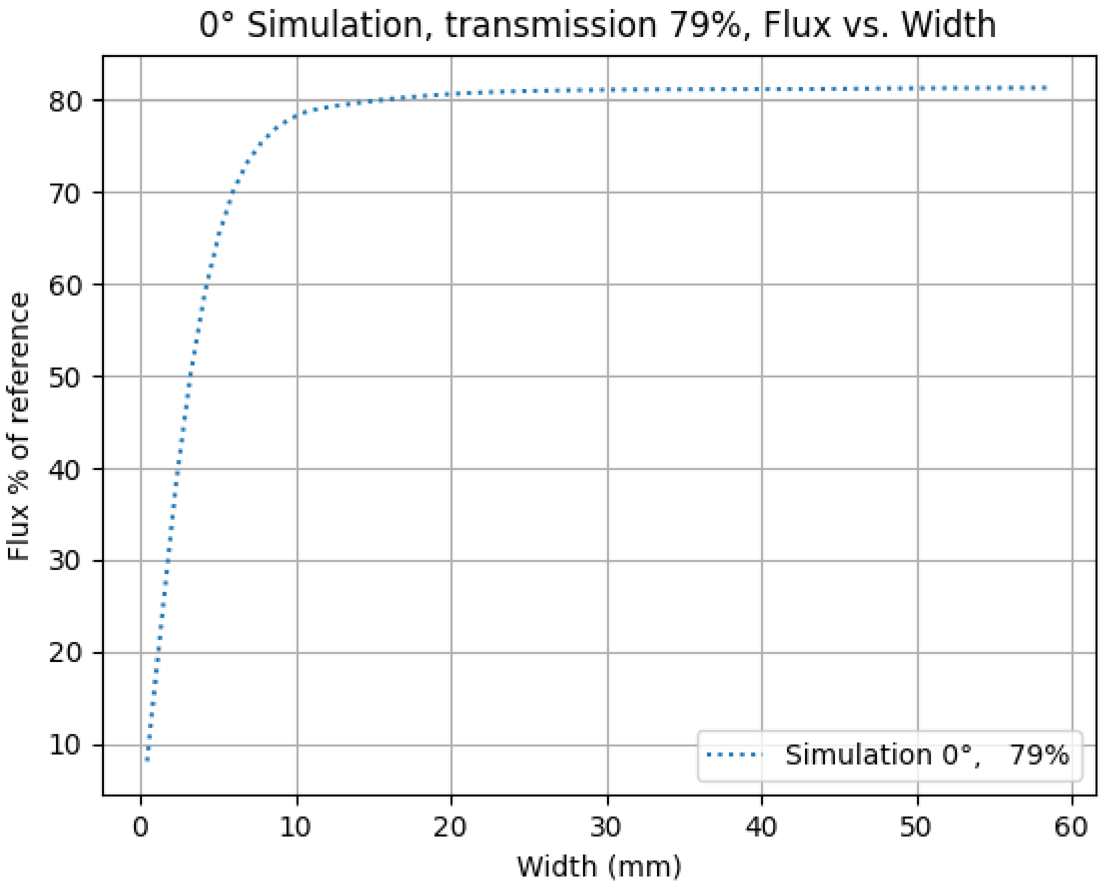
```
focal_height = 12.45
```

```python
#file names
etendue_name = "zemax/etendue_histogram.txt"
reference_name = "zemax/reference_histogram.txt"

##paths
etendue_path = os.getcwd()+"\\"+etendue_name
reference_path = os.getcwd()+"\\"+reference_name

x_ete, y_ete, ete_image = extract_data(etendue_path)
x_ref, y_ref, ref_image = extract_data(reference_path)

focal_intensity, total_intensity  = get_x_within_interval(ete_image, -focal_height/2,focal_height/2)
focal_intensity_ref, total_intensity_ref  = get_x_within_interval(ref_image, -focal_height/2,focal_height/2)
print(total_intensity)
print(focal_intensity)
factor = (focal_intensity/total_intensity)*100
print(f"focal flux = {focal_intensity}\n total flux = {total_intensity}\n  transmission = {int(factor)}%" )

# Create the plot
plt.plot(x_ete, y_ete, label = f"Etendue: {int(factor)}% intensity")
plt.plot(x_ref, y_ref, label='Reference')

#focal tube
plt.axhline(focal_height/2, color='b', linestyle='dotted', label='Focal tube')
plt.axhline(-focal_height/2, color='b', linestyle='dotted')

# Add labels and title to the plot
plt.xlabel('relative intensity')
plt.ylabel('Y height [mm]')
plt.title('Intensity profile along Y heights')
plt.legend()

plt.savefig(os.getcwd()+"\\"+"zemax_0deg_focal.png")

# Show the plot
#plt.show()

images = [np.array(ete_image),np.array(ref_image)]
neutral_axes = [len(ete_image)//2,len(ref_image)//2]
legend_names = ["Simulation 0\u00b0, 79%","reference"]
focal_widths = [focal_height,focal_height]
image_height_mm = 60
reference_Flux = total_intensity_ref


#flux_width_plot_virtual(images, neutral_axes, legend_names, focal_widths, image_height_mm, reference_Flux)
```

```
ax = flux_width_plot_virtual([images[0]], [neutral_axes[0]], [legend_names[0]], [focal_widths[0]], image_height_mm, reference_Flux)
add_ax_to_plot("simulation",ax)
```

```
1.4669119230699996
1.42954524
focal flux = 1.42954524
 total flux = 1.4669119230699996
  transmission = 97%
neutral axis: 65 len row_sums: 130
```

Figure



```
dictionary:  {'3D Print 0°': <Axes: title={'center': '0° Etendue, transmission 52%, Flux vs. Width'}, xlabel='Width (mm)', ylabel='Flux % o
f reference'>, '15i 0°': <Axes: title={'center': '0° Etendue, transmission 73%, Flux vs. Width'}, xlabel='Width (mm)', ylabel='Flux % of re
ference'>, '15i 11.75°': <Axes: title={'center': '11.75° Etendue, transmission 67%, Flux vs. Width'}, xlabel='Width (mm)', ylabel='Flux % o
f reference'>, '15i 23.5°': <Axes: title={'center': '23.5° Etendue, transmission 49%, Flux vs. Width'}, xlabel='Width (mm)', ylabel='Flux %
of reference'>, '3D Print 23.5°': <Axes: title={'center': '23.5° Etendue, transmission 39%, Flux vs. Width'}, xlabel='Width (mm)', ylabel
='Flux % of reference'>, '3D Print 11.75°': <Axes: title={'center': '11.75° Etendue, transmission 48%, Flux vs. Width'}, xlabel='Width (m
m)', ylabel='Flux % of reference'>, 'simulation': <Axes: title={'center': '0° Simulation, transmission 79%, Flux vs. Width'}, xlabel='Width
(mm)', ylabel='Flux % of reference'>, 'ref': <Axes: title={'center': ', Flux vs. Width'}, xlabel='Width (mm)', ylabel='Flux % of referenc
e'>}
```

In [203…

```
#del ax_to_plot["ref_loss"]
```

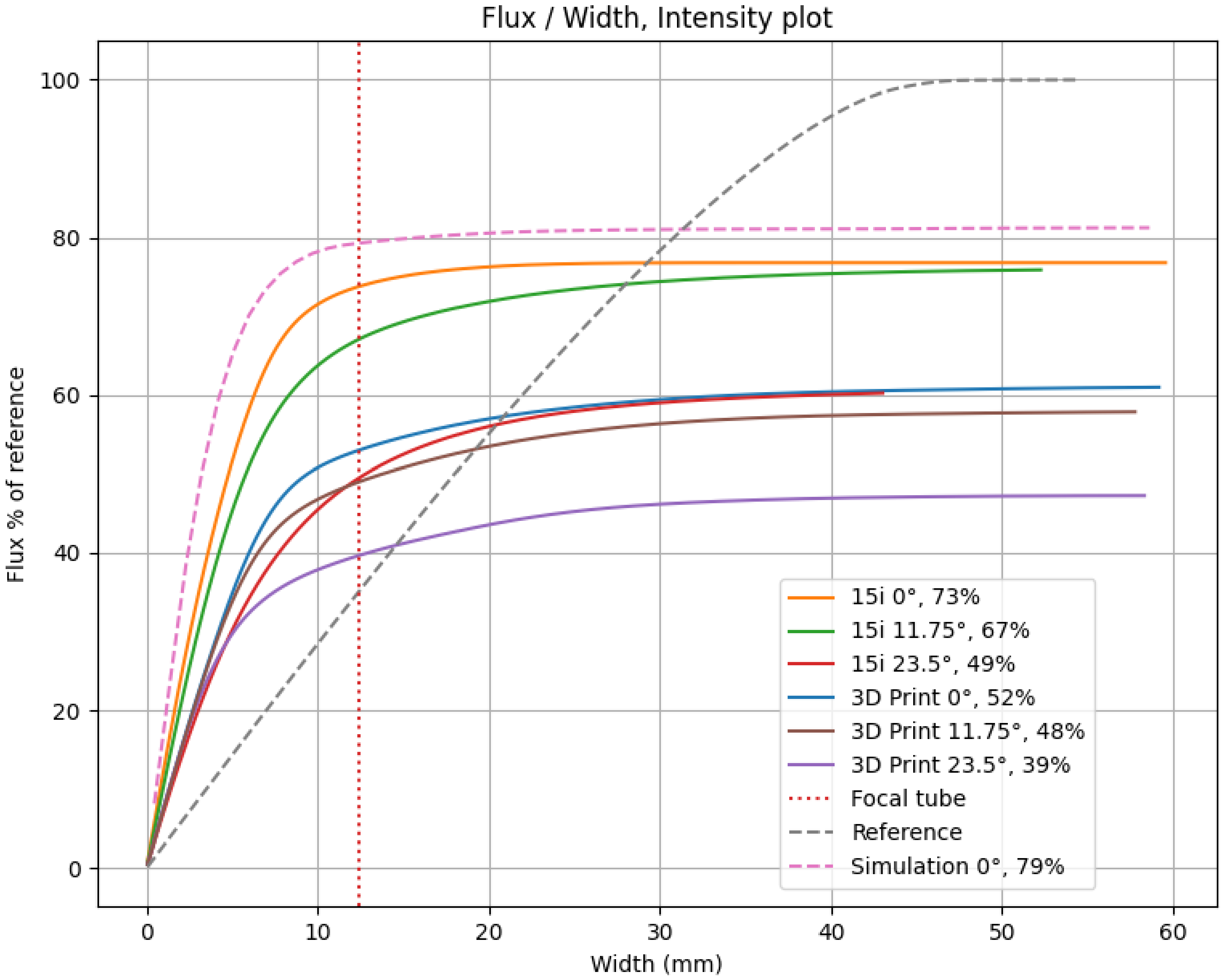
resultater:

In [238…

```
for key in ax_to_plot:
    print(key)
plot_all_axis()
```

```
3D Print 0°
15i 0°
15i 11.75°
15i 23.5°
3D Print 23.5°
3D Print 11.75°
simulation
ref
```

Figure



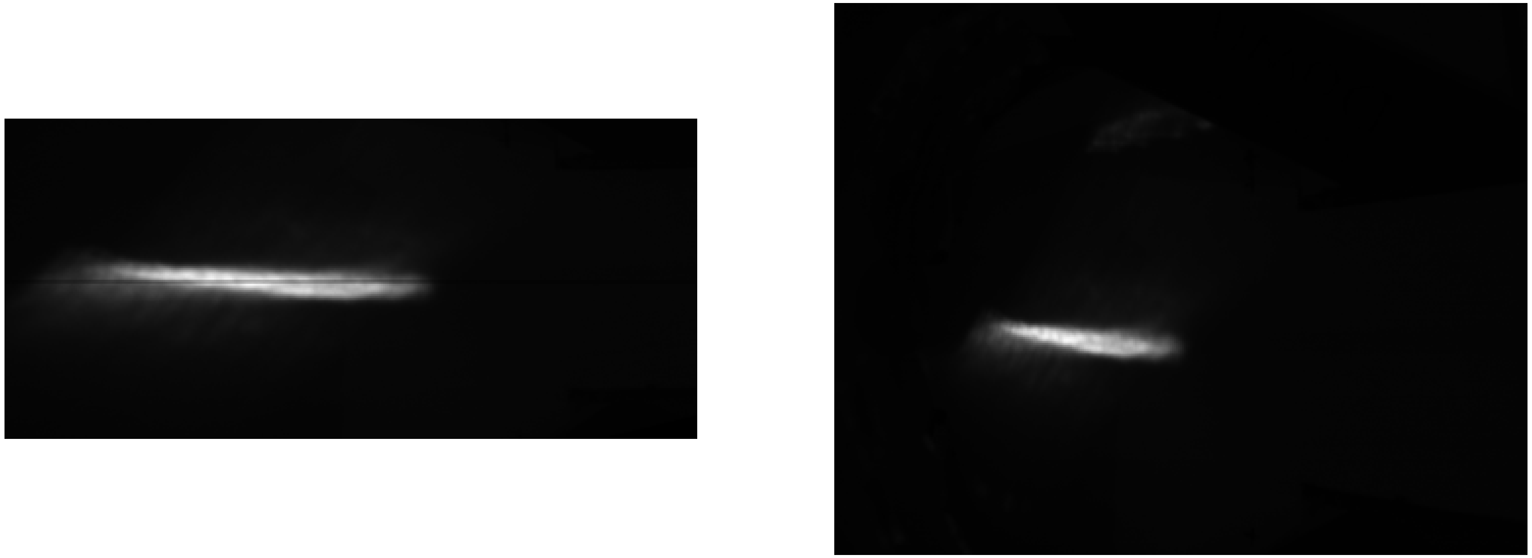
3dprint

In [293…

```
plt.close("all")
clicked_points = [(238, 398), (791, 290), (778, 984), (217, 828)]
output_comparison("ImageAnalysis/3dprint/0deg.tif","ImageAnalysis/3dprint/0deg_ref.tif","ImageAnalysis/3dprint/plots/0deg",5,5,"3D Print 0\
```

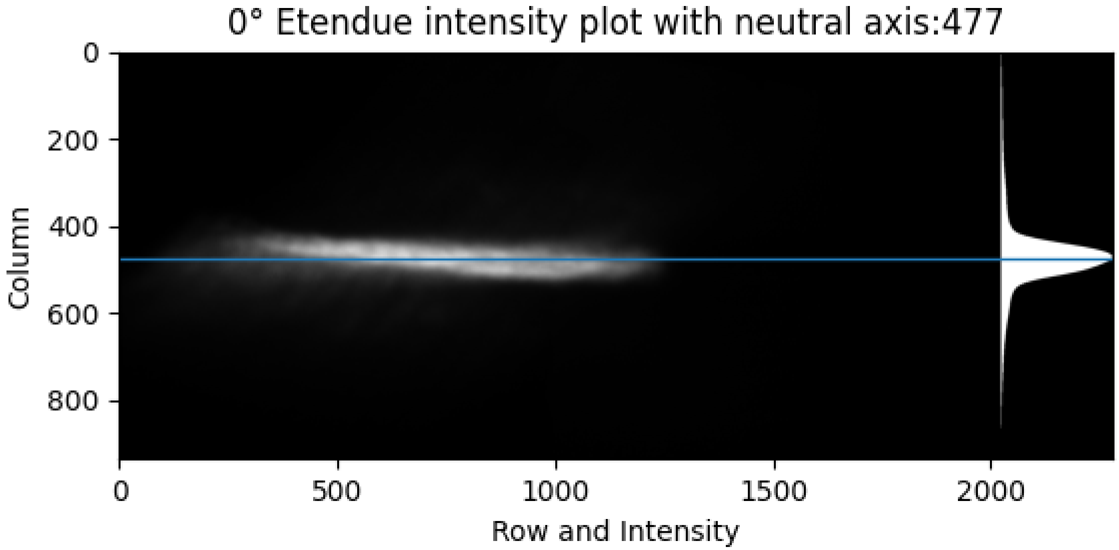
```
Clicked coordinates: [(238, 398), (791, 290), (778, 984), (217, 828)], Testing: False
Clicked coordinates: [(238, 398), (791, 290), (778, 984), (217, 828)], Testing: False
```

Figure

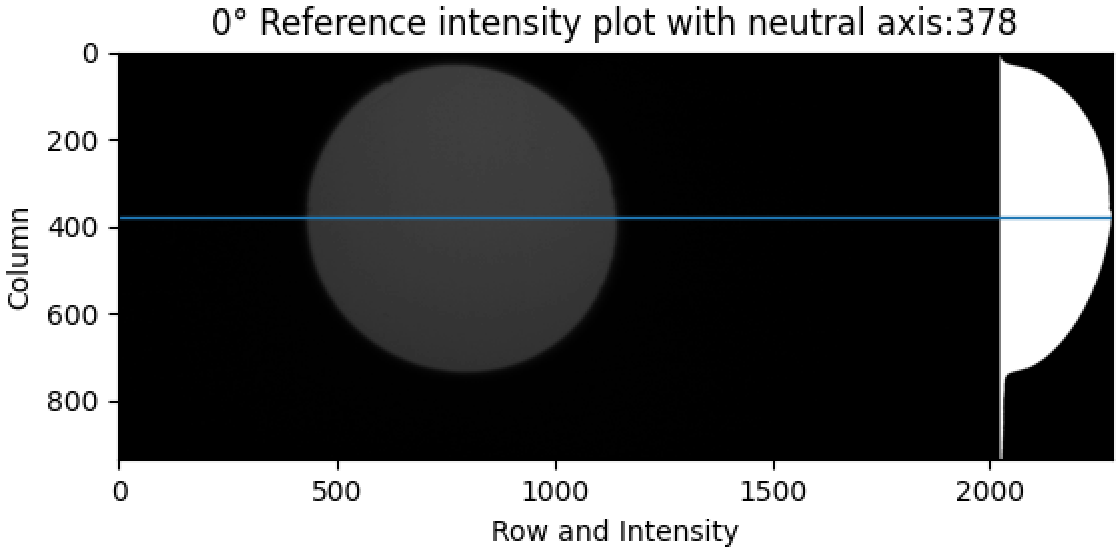


Figure

Figure

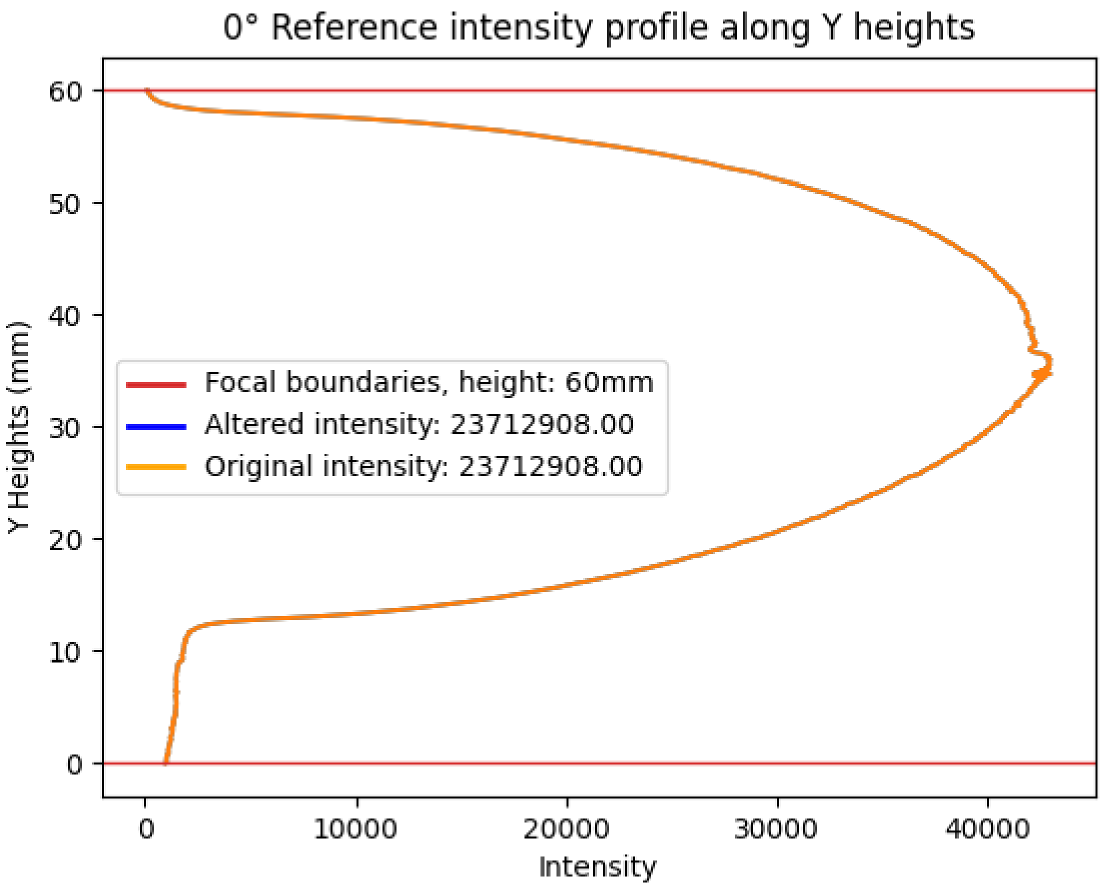

0° Etendue intensity plot with neutral axis:477
Column
0
200
400
600
800
0
500
1000
1500
2000
Row and Intensity

Figure



```
An error occurred trying to create min_hline, in 'show focal energy'-function: local variable 'index_to_keep_min' referenced before assignm
ent
 the hline was set to the first index of the list
An error occurred trying to create max_hline, in 'show focal energy'-function: local variable 'index_to_keep_max' referenced before assignm
ent
 the hline was set to the last index of the list
```

Figure



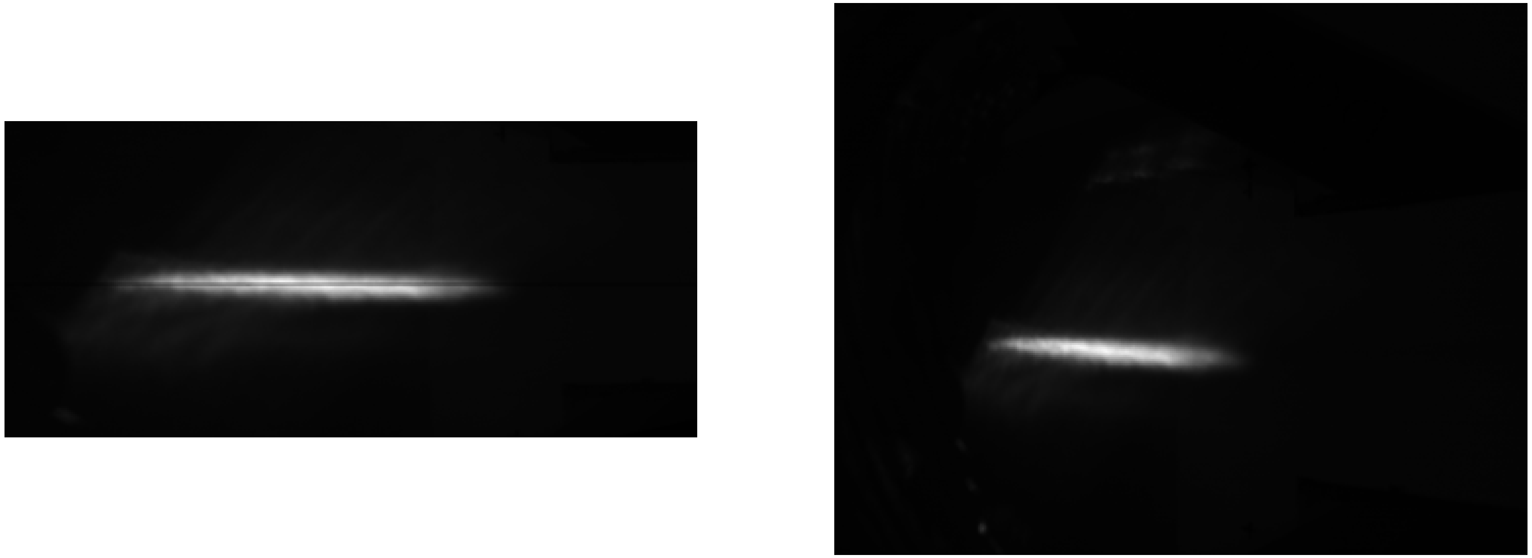
```
referece flux = 23712908
 etendue flux = 11712631
  transmission = 49%
```

In [294…

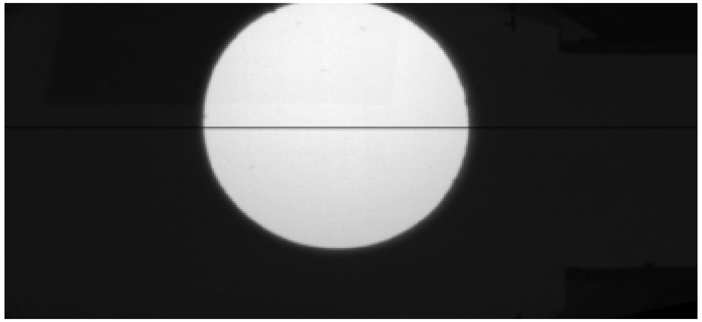
```
clicked_points = [(209, 410), (785, 318), (772, 980), (184, 840)]
output_comparison("ImageAnalysis/3dprint/11.75.tif","ImageAnalysis/3dprint/11.75ref.tif","ImageAnalysis/3dprint/plots/11.75deg",7,7,"3D Pri
```

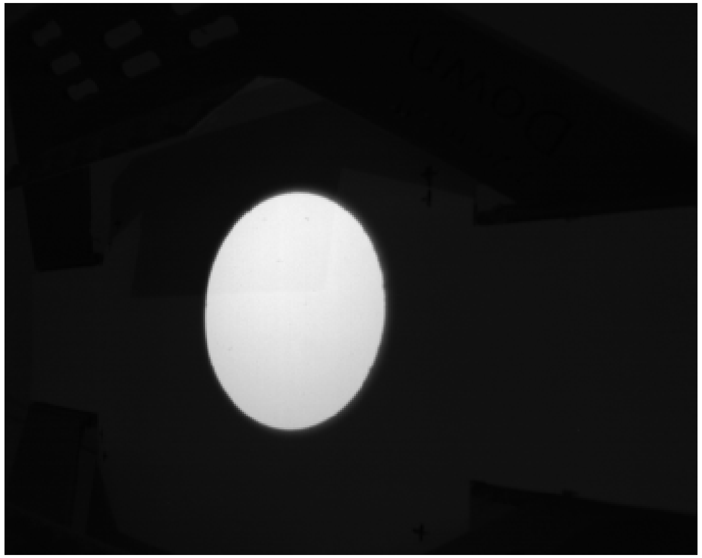
```
Clicked coordinates: [(209, 410), (785, 318), (772, 980), (184, 840)], Testing: False
Clicked coordinates: [(209, 410), (785, 318), (772, 980), (184, 840)], Testing: False
```

Figure

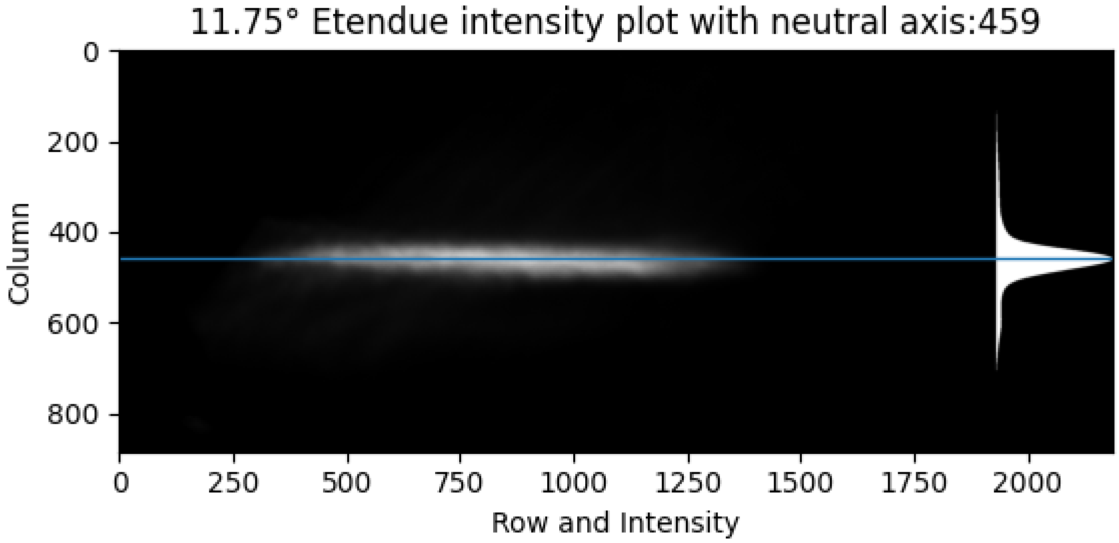


Figure

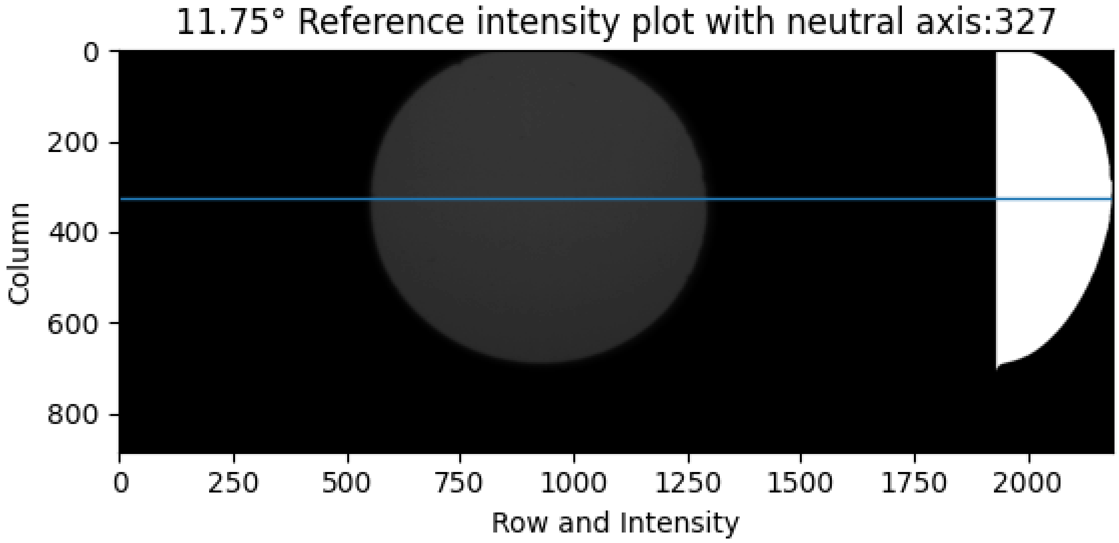




Figure

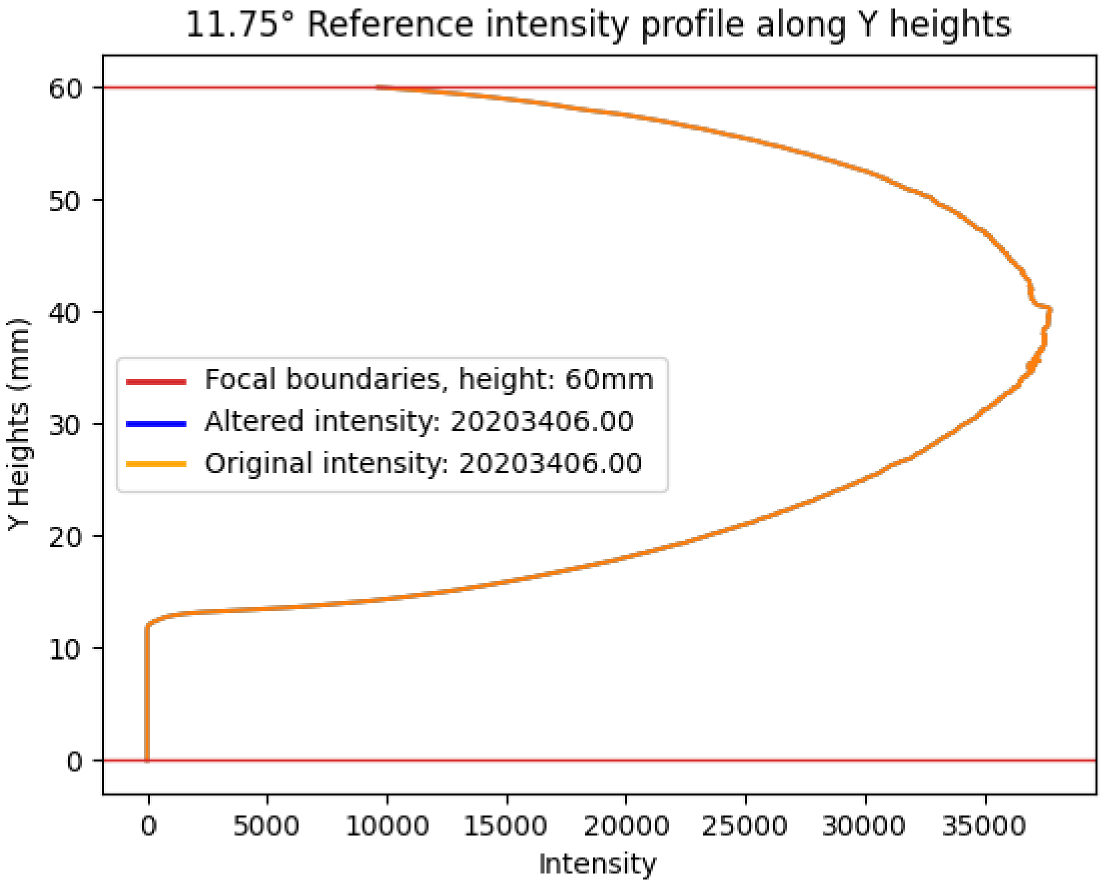

11.75° Etendue intensity plot with neutral axis:459
Column
0
200
400
600
800
0
250
500
750
1000
1250
1500
1750
2000
Row and Intensity

Figure



```
An error occurred trying to create min_hline, in 'show focal energy'-function: local variable 'index_to_keep_min' referenced before assignm
ent
 the hline was set to the first index of the list
An error occurred trying to create max_hline, in 'show focal energy'-function: local variable 'index_to_keep_max' referenced before assignm
ent
 the hline was set to the last index of the list
```

Figure



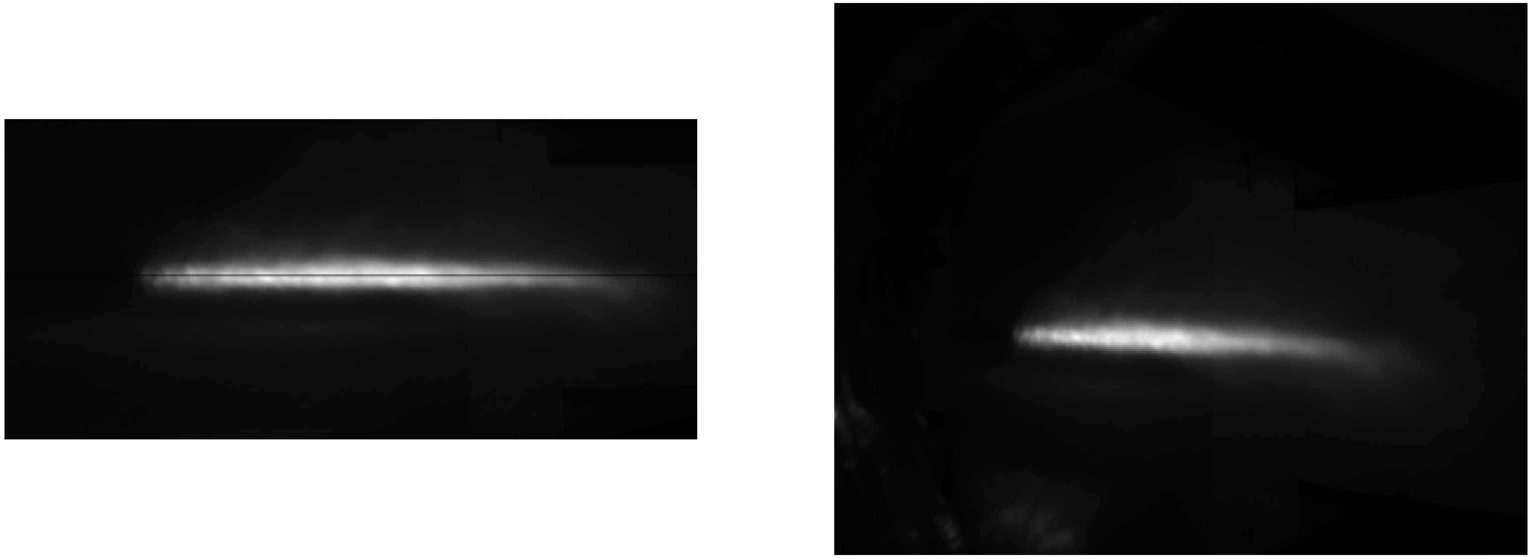
```
referece flux = 20203406
 etendue flux = 8982894
  transmission = 44%
```

In [295…

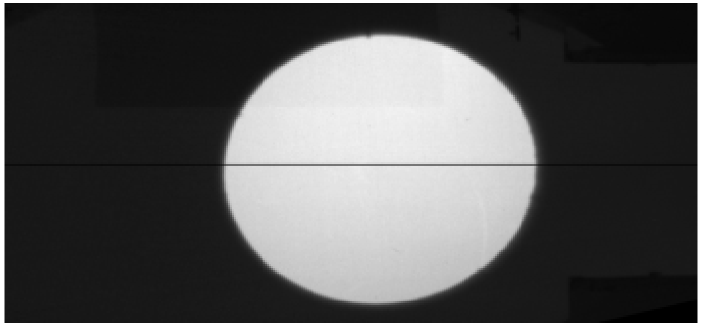
```
plt.close("all")
clicked_points = [(238, 403), (795, 298), (779, 985), (215, 833)]
output_comparison("ImageAnalysis/3dprint/23.5.tif","ImageAnalysis/3dprint/23.5ref.tif","ImageAnalysis/3dprint/plots/23.5deg",6,6,"3D Print
```

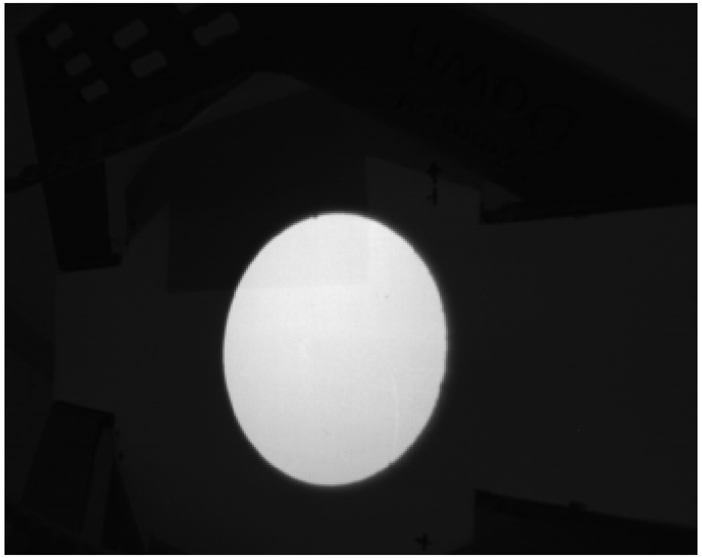
```
Clicked coordinates: [(238, 403), (795, 298), (779, 985), (215, 833)], Testing: False
Clicked coordinates: [(238, 403), (795, 298), (779, 985), (215, 833)], Testing: False
```

Figure

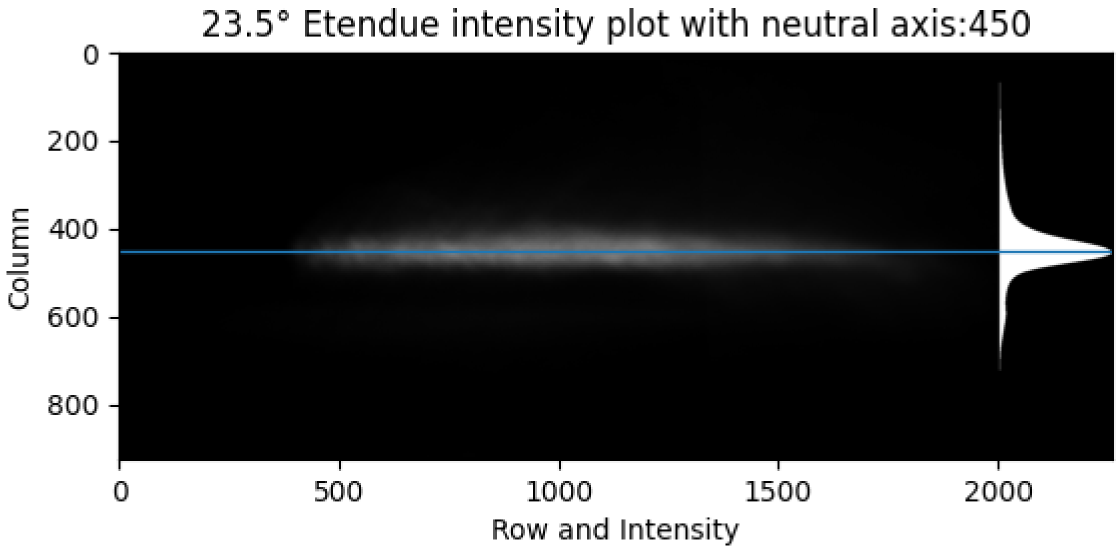


Figure

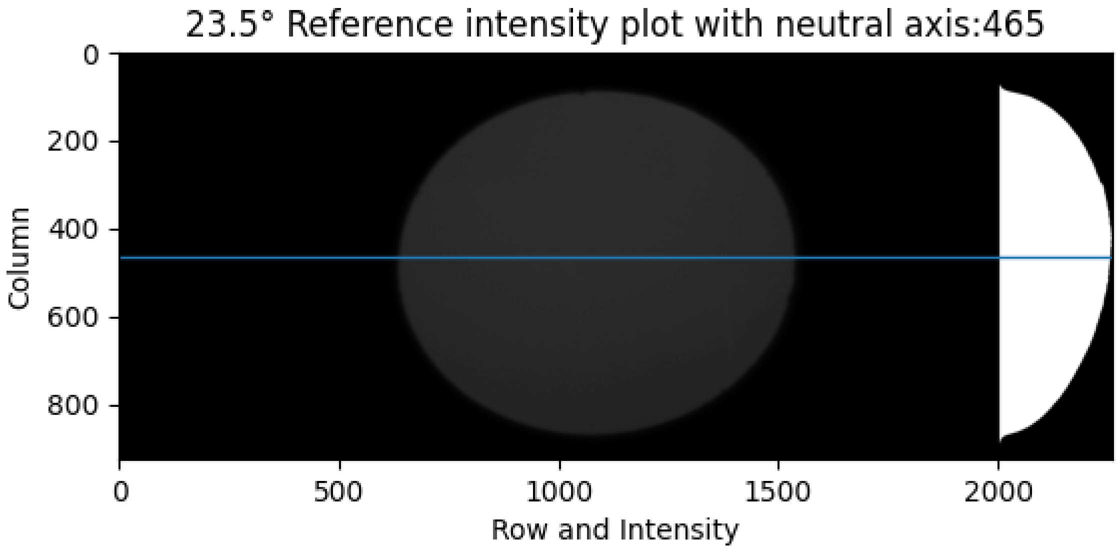




Figure

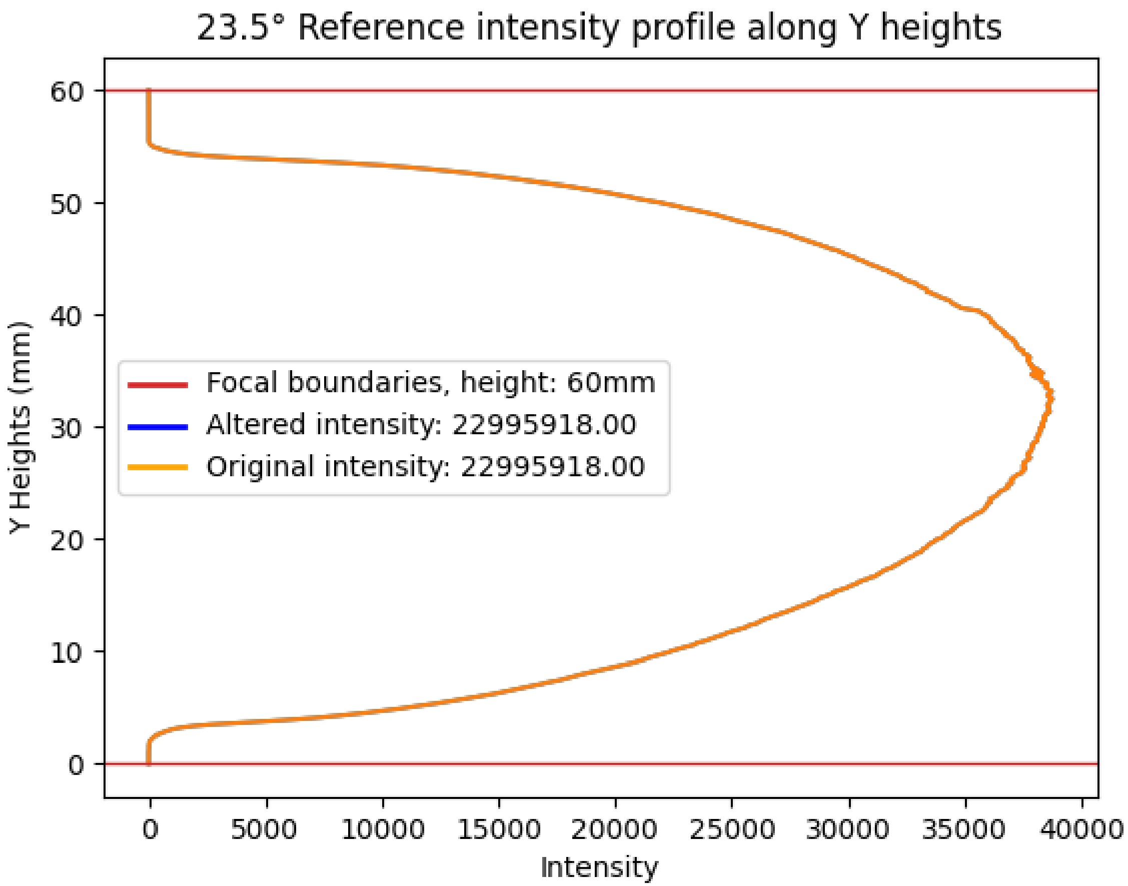

23.5° Etendue intensity plot with neutral axis:450
0
200
400
600
800
Column
0
500
1000
1500
2000
Row and Intensity

Figure



```
An error occurred trying to create min_hline, in 'show focal energy'-function: local variable 'index_to_keep_min' referenced before assignm
ent
 the hline was set to the first index of the list
An error occurred trying to create max_hline, in 'show focal energy'-function: local variable 'index_to_keep_max' referenced before assignm
ent
 the hline was set to the last index of the list
```

Figure



```
referece flux = 22995918
 etendue flux = 9109988
  transmission = 39%
```

3d_2

In [296…

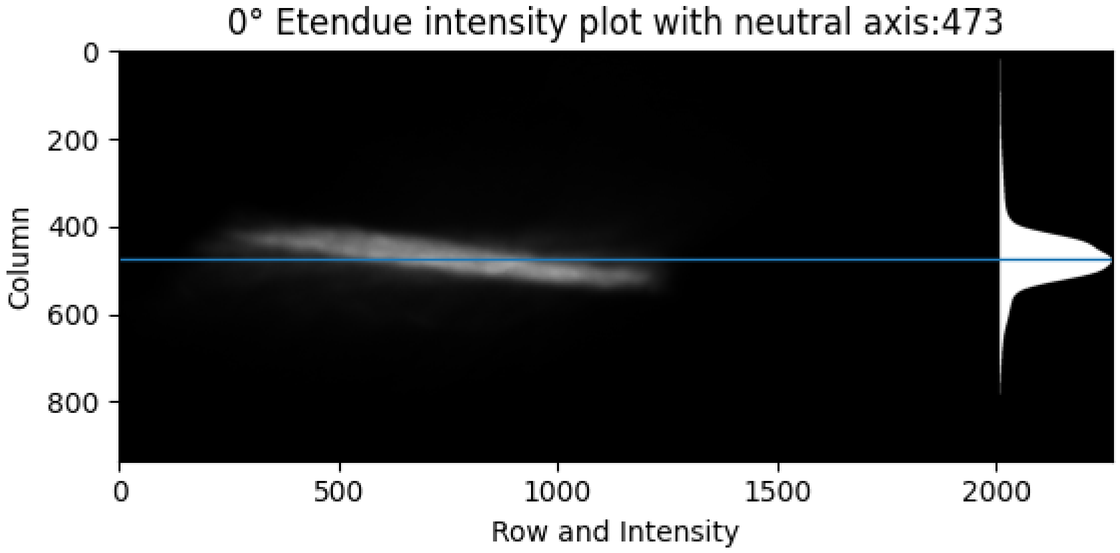
```
clicked_points = [(314, 411), (875, 302), (863, 991), (294, 835)]
output_comparison("ImageAnalysis/3dprint/3d_2/0deg.tif","ImageAnalysis/3dprint/3d_2/0deg_ref.tif","ImageAnalysis/3dprint/3d_2/plots/0deg",5
```

```
Clicked coordinates: [(314, 411), (875, 302), (863, 991), (294, 835)], Testing: False
Clicked coordinates: [(314, 411), (875, 302), (863, 991), (294, 835)], Testing: False
```

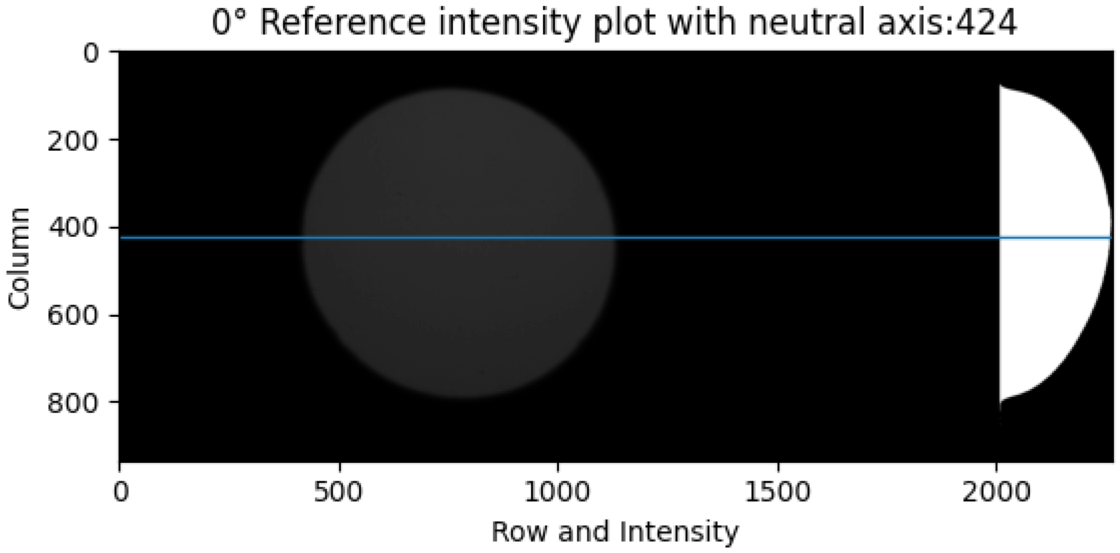
Figure

Figure

Figure

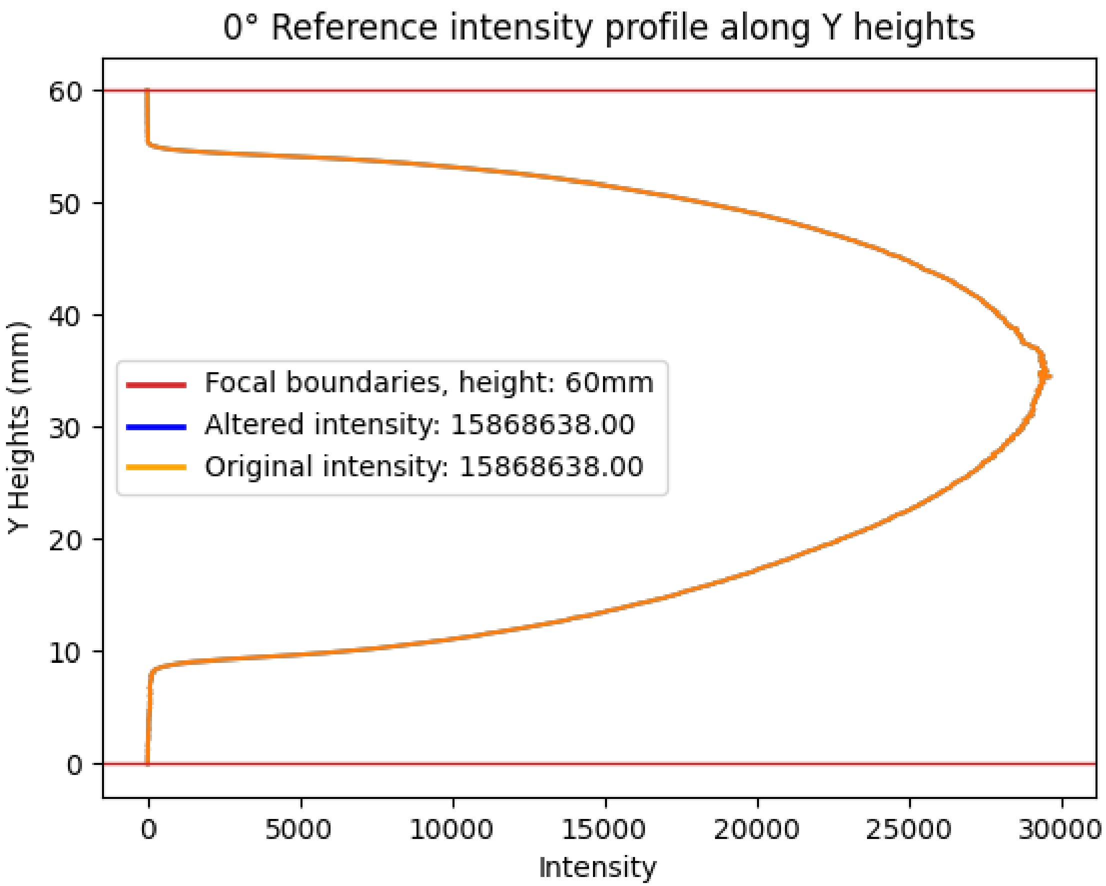

0° Etendue intensity plot with neutral axis:473
0
200
400
600
800
Column
0
500
1000
1500
2000
Row and Intensity

Figure



```
An error occurred trying to create min_hline, in 'show focal energy'-function: local variable 'index_to_keep_min' referenced before assignm
ent
 the hline was set to the first index of the list
An error occurred trying to create max_hline, in 'show focal energy'-function: local variable 'index_to_keep_max' referenced before assignm
ent
 the hline was set to the last index of the list
```

Figure



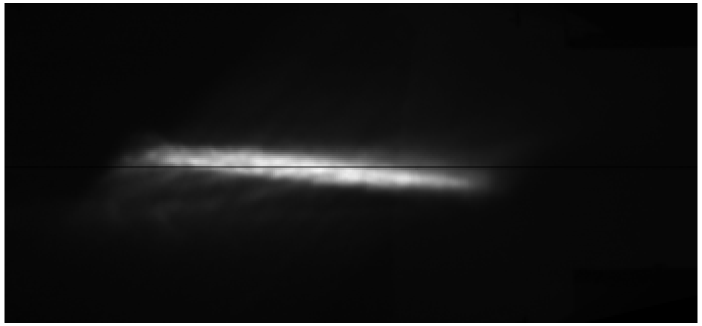
```
referece flux = 15868638
 etendue flux = 8399003
  transmission = 52%
```

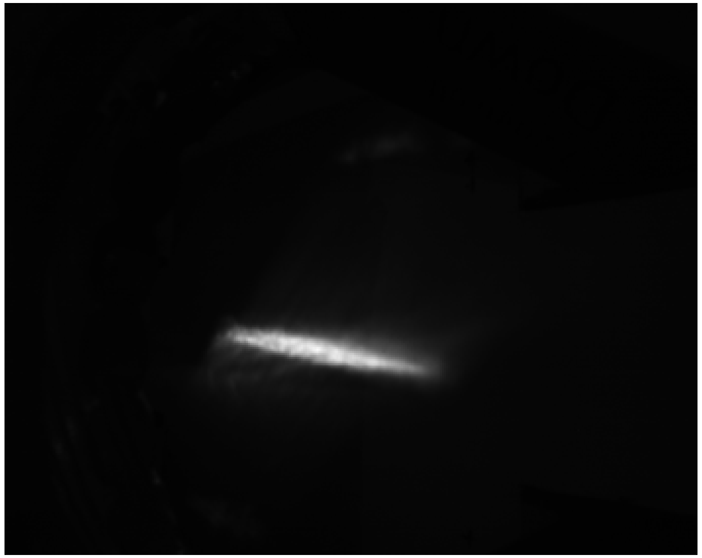
In [297…

```
clicked_points = [(314, 410), (872, 300), (862, 993), (296, 834)]
output_comparison("ImageAnalysis/3dprint/3d_2/11.75deg.tif","ImageAnalysis/3dprint/3d_2/11.75_ref.tif","ImageAnalysis/3dprint/3d_2/plots/11
```

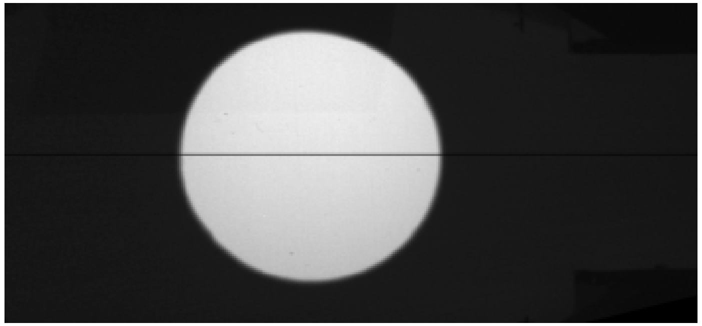
```
Clicked coordinates: [(314, 410), (872, 300), (862, 993), (296, 834)], Testing: False
Clicked coordinates: [(314, 410), (872, 300), (862, 993), (296, 834)], Testing: False
```

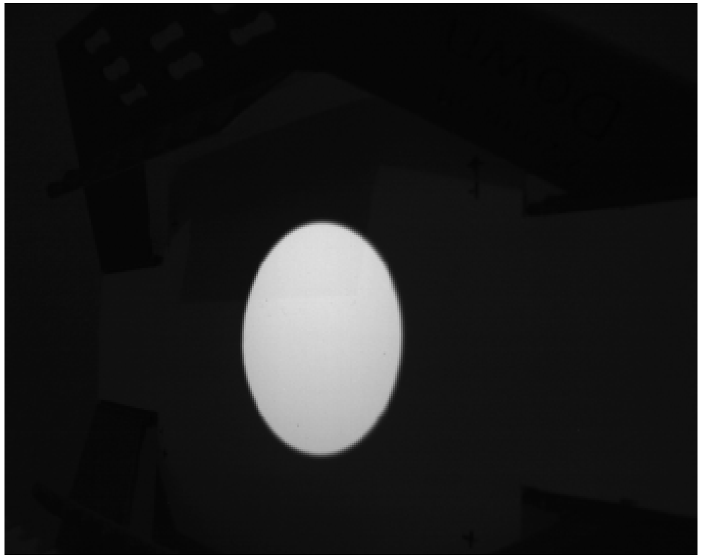
Figure



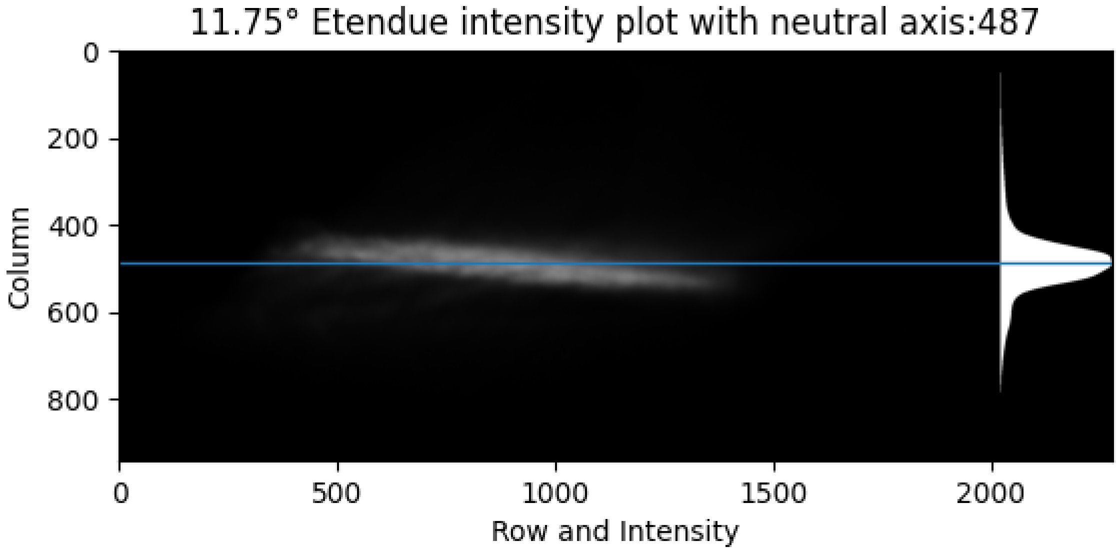


Figure

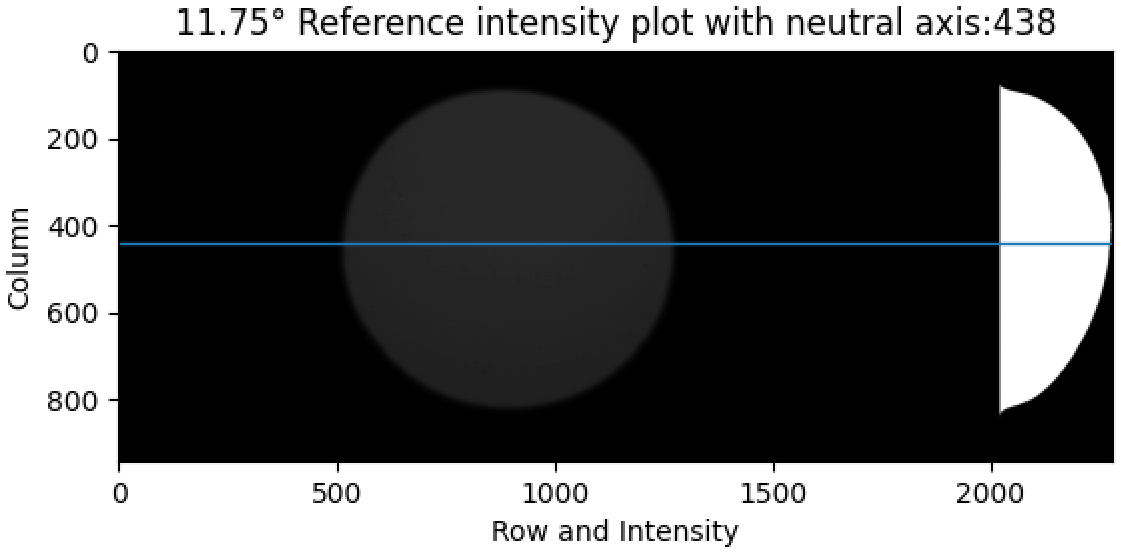




Figure

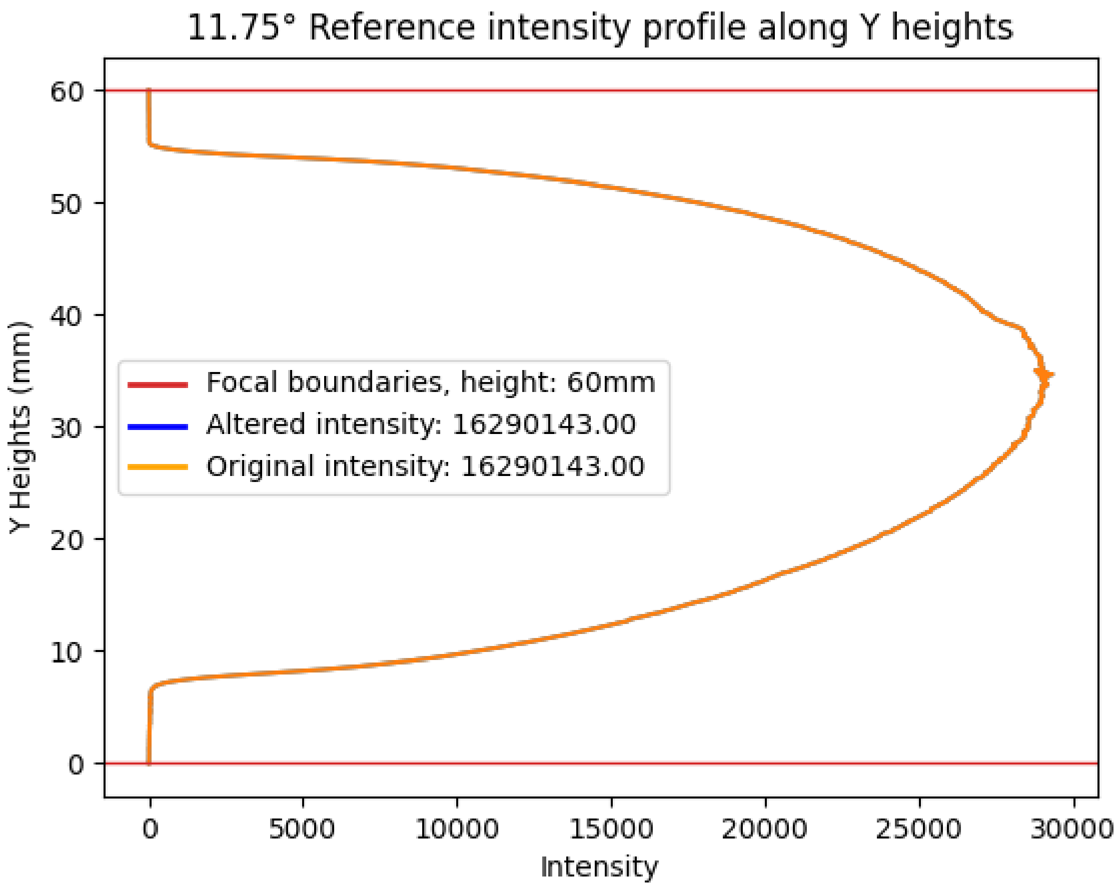

11.75° Etendue intensity plot with neutral axis:487
0
200
400
600
800
Column
0
500
1000
1500
2000
Row and Intensity

Figure



```
An error occurred trying to create min_hline, in 'show focal energy'-function: local variable 'index_to_keep_min' referenced before assignm
ent
 the hline was set to the first index of the list
An error occurred trying to create max_hline, in 'show focal energy'-function: local variable 'index_to_keep_max' referenced before assignm
ent
 the hline was set to the last index of the list
```

Figure



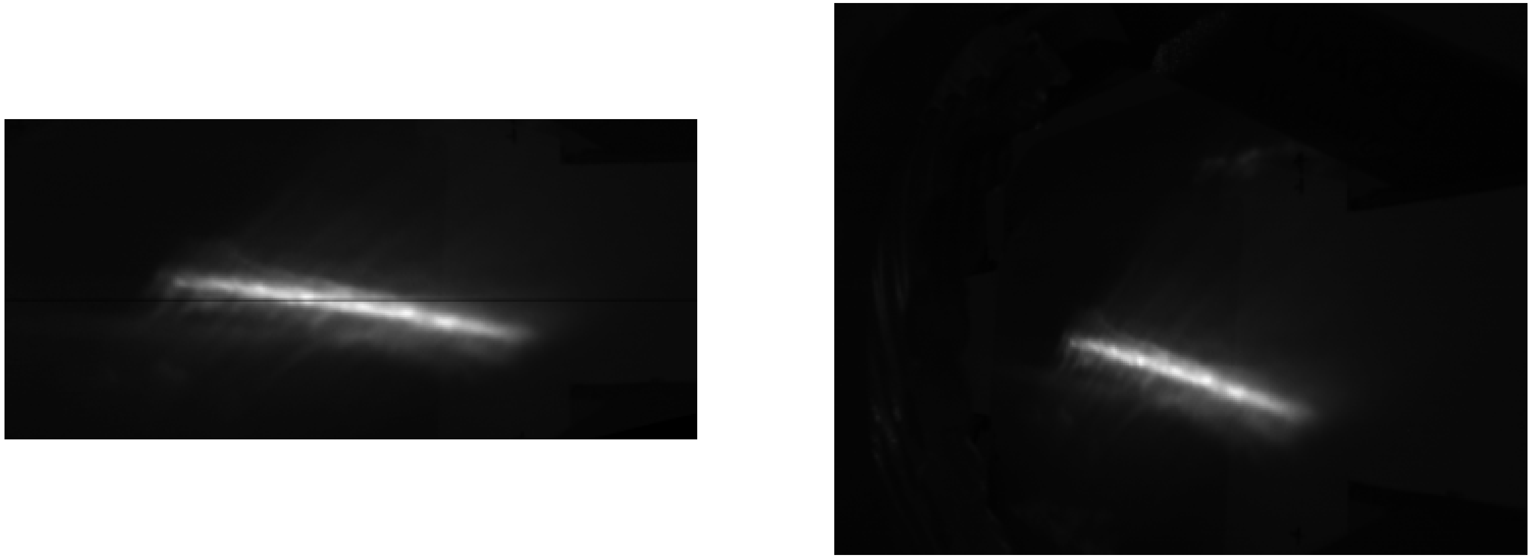
```
referece flux = 16290143
 etendue flux = 7978506
  transmission = 48%
```

In [298…

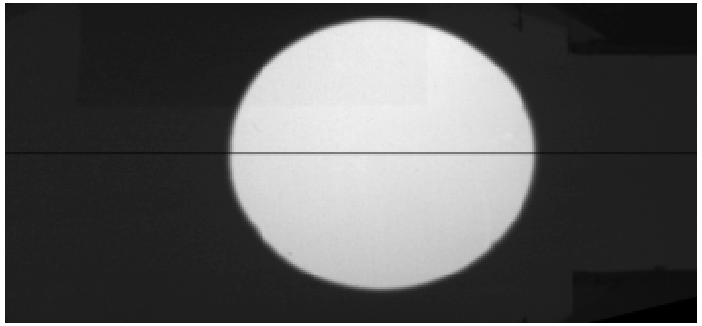
```
clicked_points = [(315, 410), (873, 301), (864, 991), (295, 835)]
output_comparison("ImageAnalysis/3dprint/3d_2/23.5.tif","ImageAnalysis/3dprint/3d_2/23.5ref.tif","ImageAnalysis/3dprint/3d_2/plots/23.5deg"
```

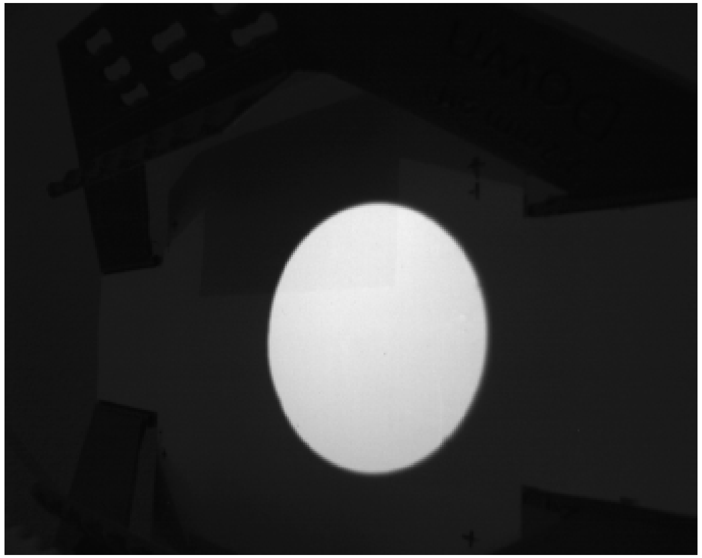
```
Clicked coordinates: [(315, 410), (873, 301), (864, 991), (295, 835)], Testing: False
Clicked coordinates: [(315, 410), (873, 301), (864, 991), (295, 835)], Testing: False
```

Figure

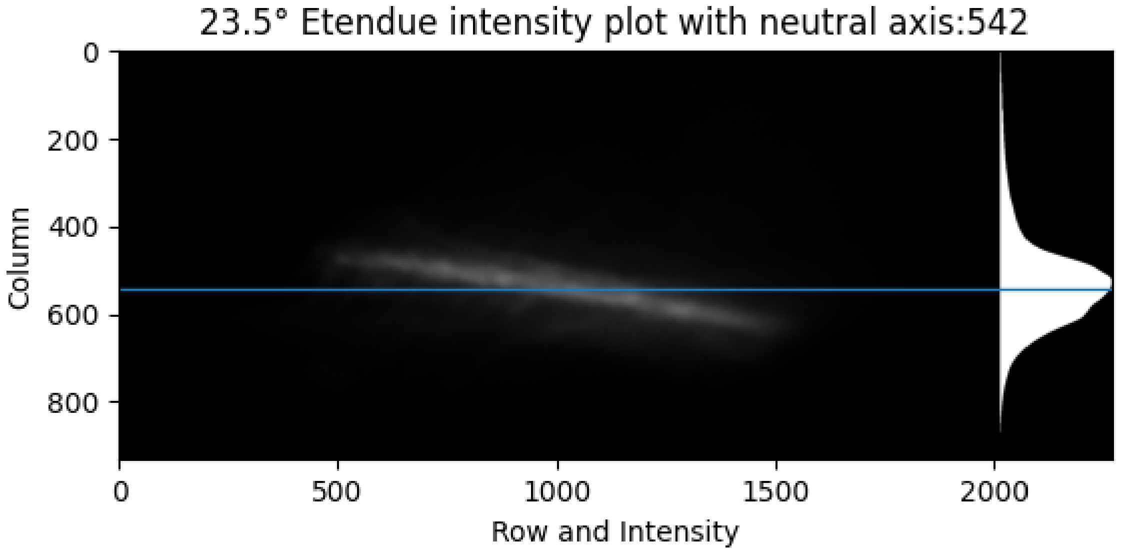


Figure

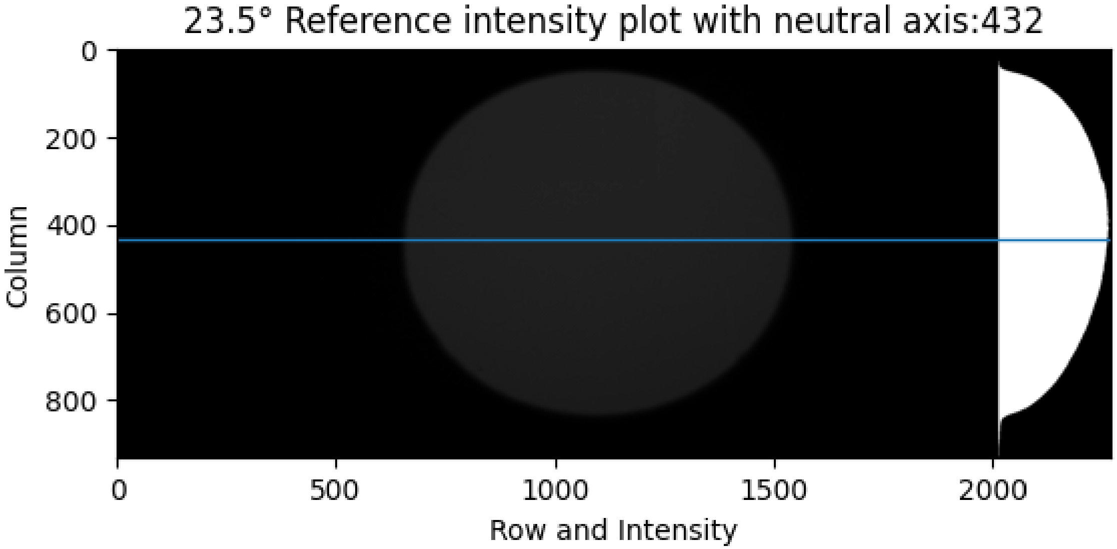




Figure

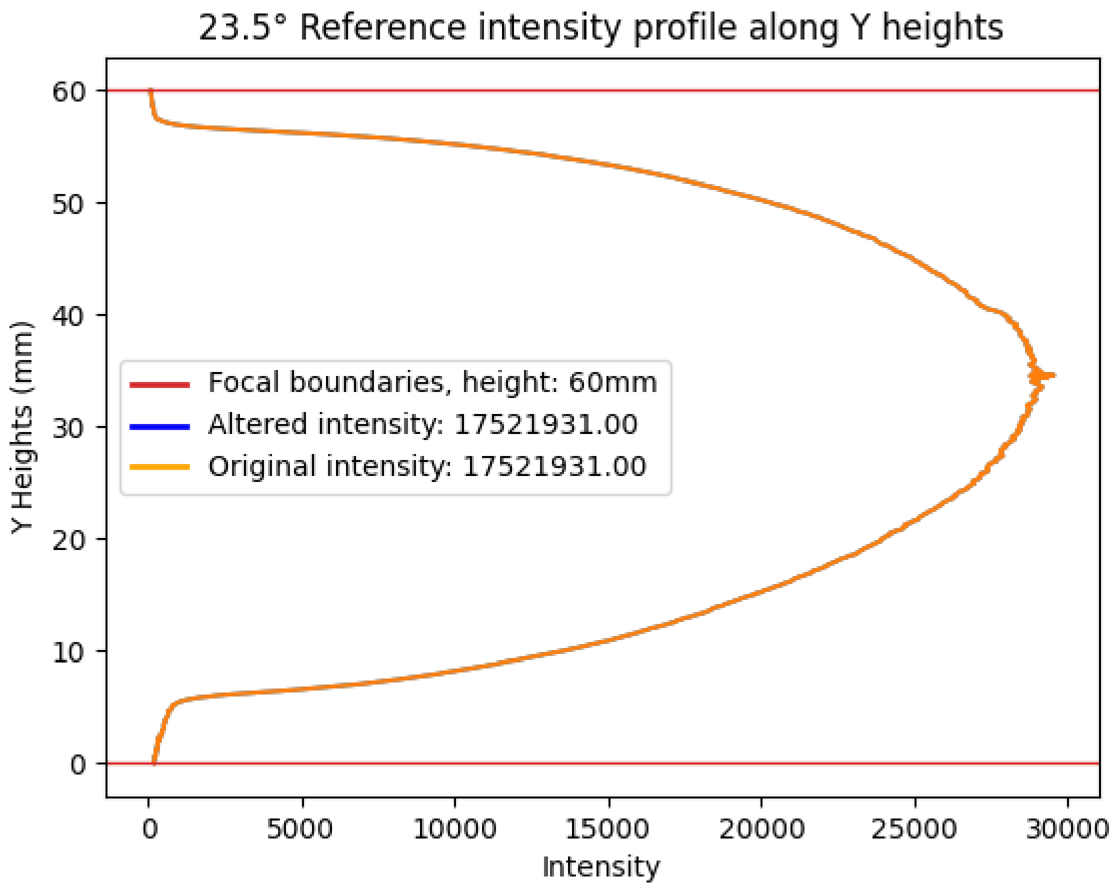

23.5° Etendue intensity plot with neutral axis:542
Column
0
200
400
600
800
0
500
1000
1500
2000
Row and Intensity

Figure



```
An error occurred trying to create min_hline, in 'show focal energy'-function: local variable 'index_to_keep_min' referenced before assignm
ent
 the hline was set to the first index of the list
An error occurred trying to create max_hline, in 'show focal energy'-function: local variable 'index_to_keep_max' referenced before assignm
ent
 the hline was set to the last index of the list
```

Figure



```
referece flux = 17521931
 etendue flux = 6519176
  transmission = 37%
```

15iv

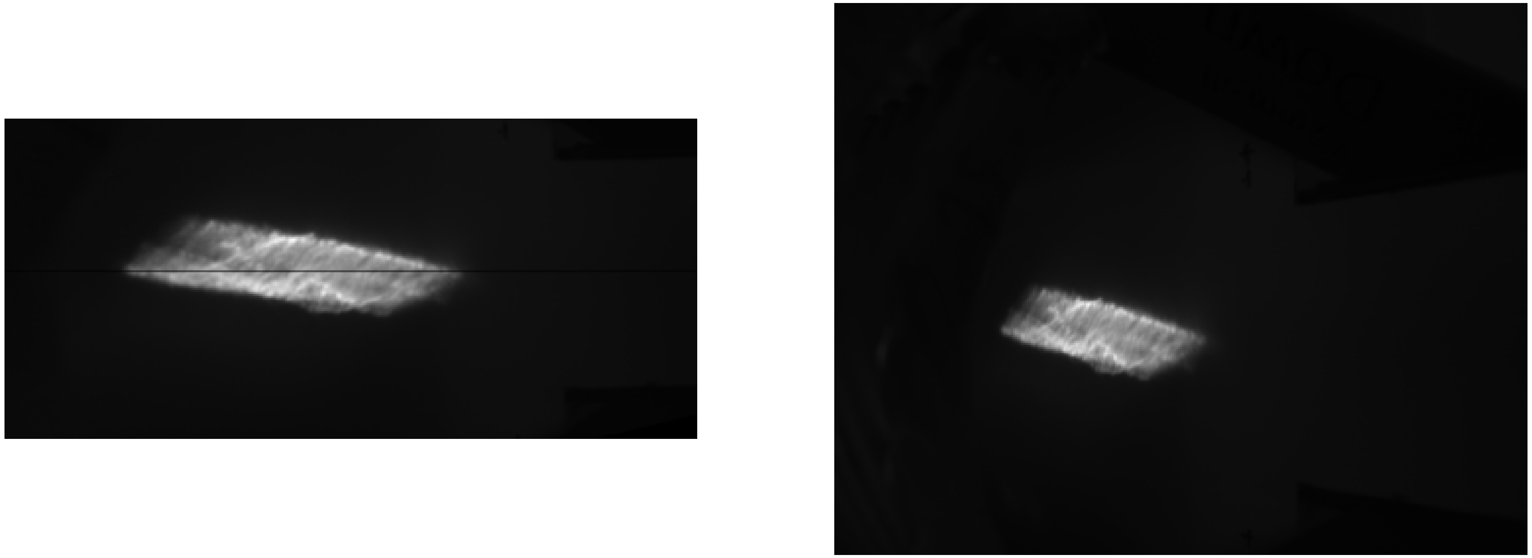
In [299…

```
clicked_points = [(225, 403), (783, 301), (774, 987), (206, 834)]
output_comparison("ImageAnalysis/15iv/0deg.tif","ImageAnalysis/15iv/0ref.tif","ImageAnalysis/15iv/plots/0deg",13,13)
```

```
Clicked coordinates: [(225, 403), (783, 301), (774, 987), (206, 834)], Testing: False
Clicked coordinates: [(225, 403), (783, 301), (774, 987), (206, 834)], Testing: False
```

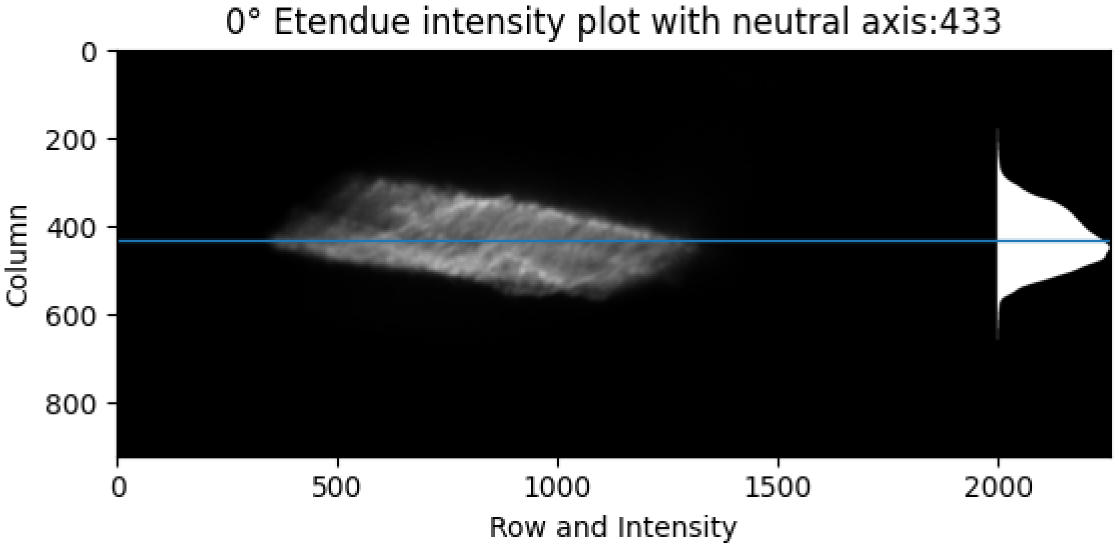
```
C:\Users\bjorn\AppData\Local\Temp\ipykernel_19732\4001352899.py:11: RuntimeWarning: More than 20 figures have been opened. Figures created
through the pyplot interface (`matplotlib.pyplot.figure`) are retained until explicitly closed and may consume too much memory. (To control
this warning, see the rcParam `figure.max_open_warning`). Consider using `matplotlib.pyplot.close()`.
  fig, (ax2, ax1) = plt.subplots(1, 2, figsize=(10, 5))
```

Figure

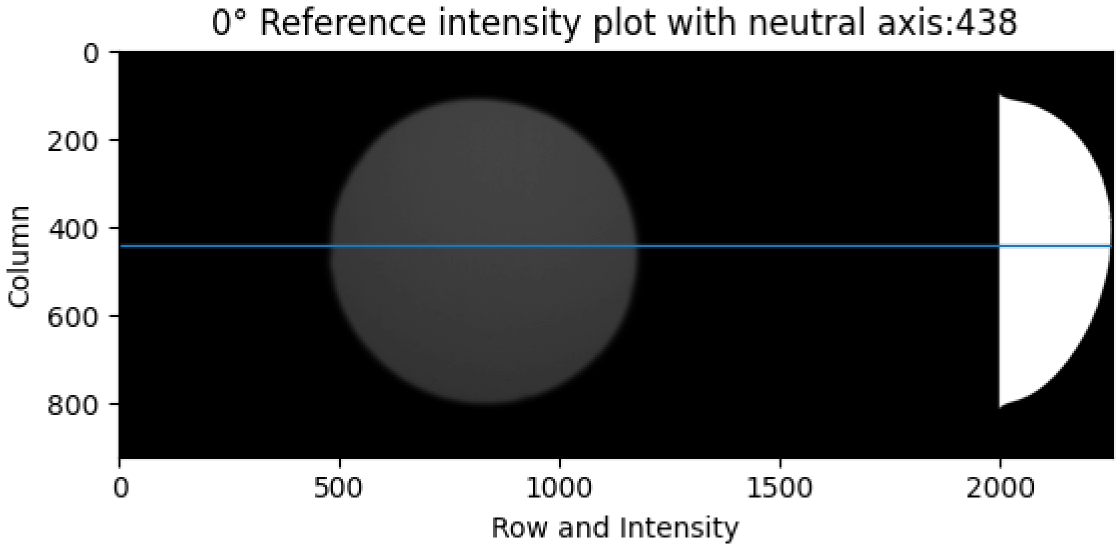


Figure

Figure

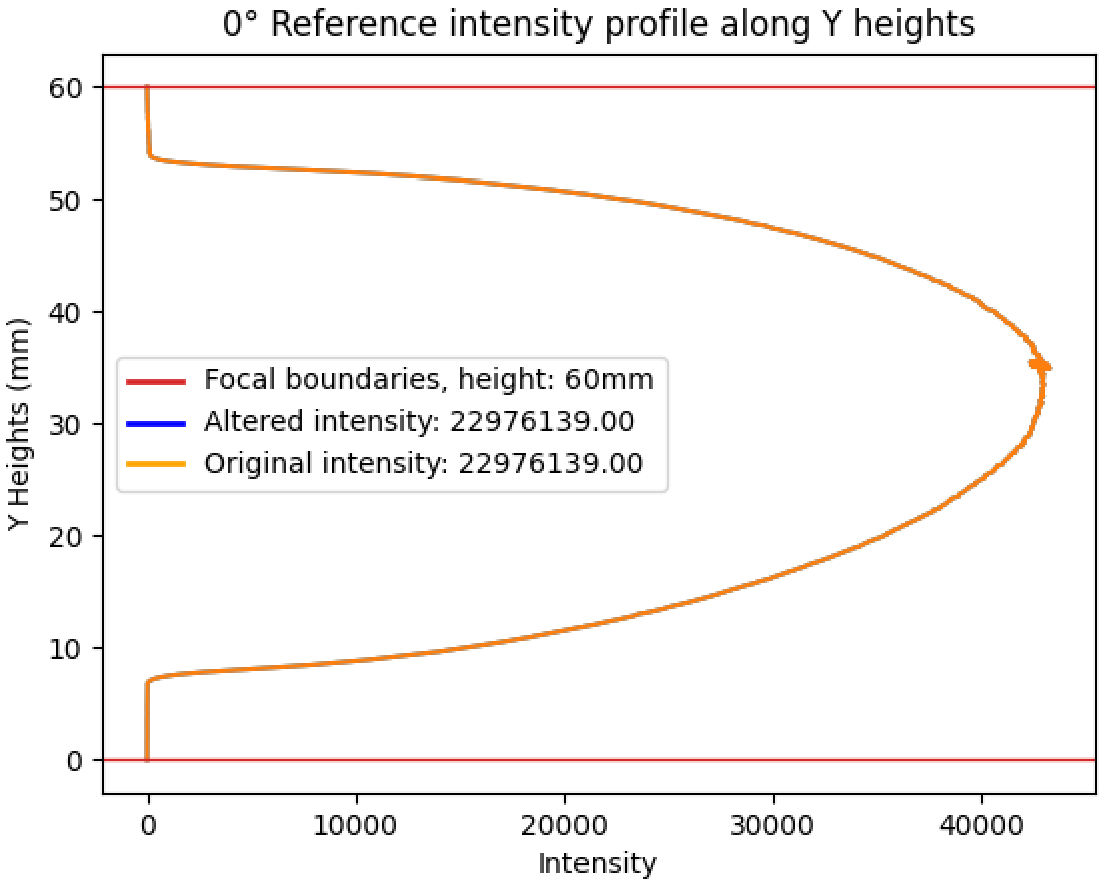

0° Etendue intensity plot with neutral axis:433
0
200
400
600
800
Column
0
500
1000
1500
2000
Row and Intensity

Figure



```
An error occurred trying to create min_hline, in 'show focal energy'-function: local variable 'index_to_keep_min' referenced before assignm
ent
 the hline was set to the first index of the list
An error occurred trying to create max_hline, in 'show focal energy'-function: local variable 'index_to_keep_max' referenced before assignm
ent
 the hline was set to the last index of the list
```

Figure


0° Reference intensity profile along Y heights

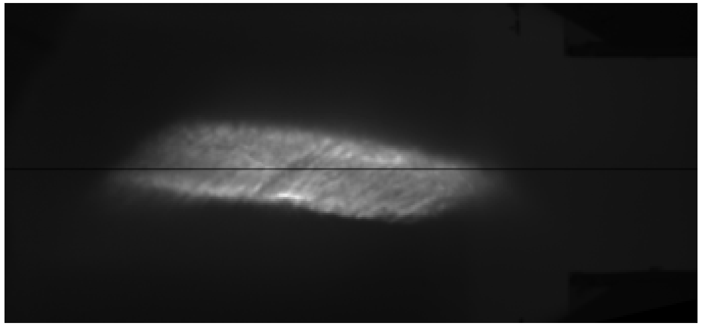
```
referece flux = 22976139
 etendue flux = 14470345
  transmission = 62%
```

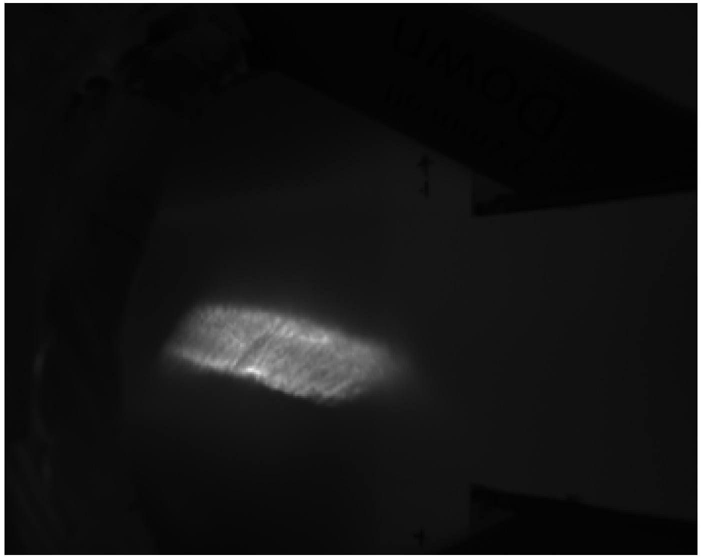
In [300…

```python
clicked_points = [(225, 404), (778, 296), (772, 987), (209, 836)]
output_comparison("ImageAnalysis/15iv/11.75.tif","ImageAnalysis/15iv/11.75ref.tif","ImageAnalysis/15iv/plots/11.75deg",14,14)
```

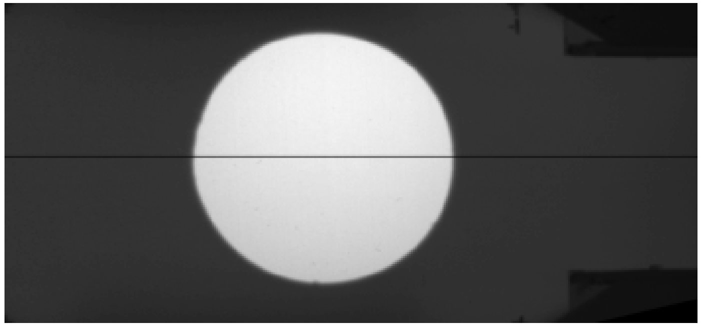
```
Clicked coordinates: [(225, 404), (778, 296), (772, 987), (209, 836)], Testing: False
Clicked coordinates: [(225, 404), (778, 296), (772, 987), (209, 836)], Testing: False
```

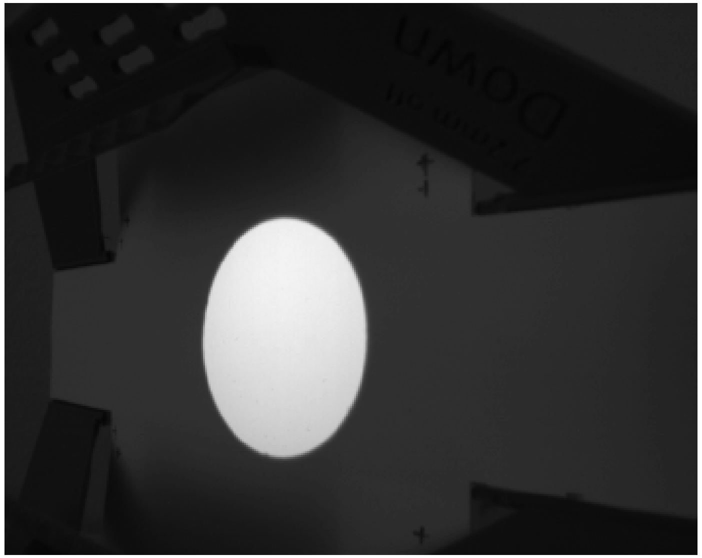
Figure



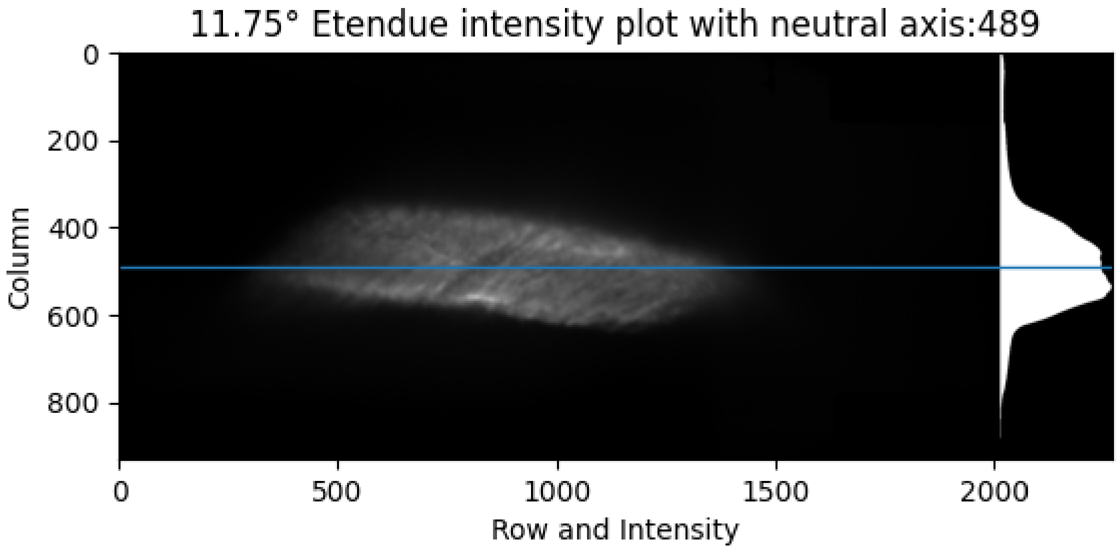


Figure

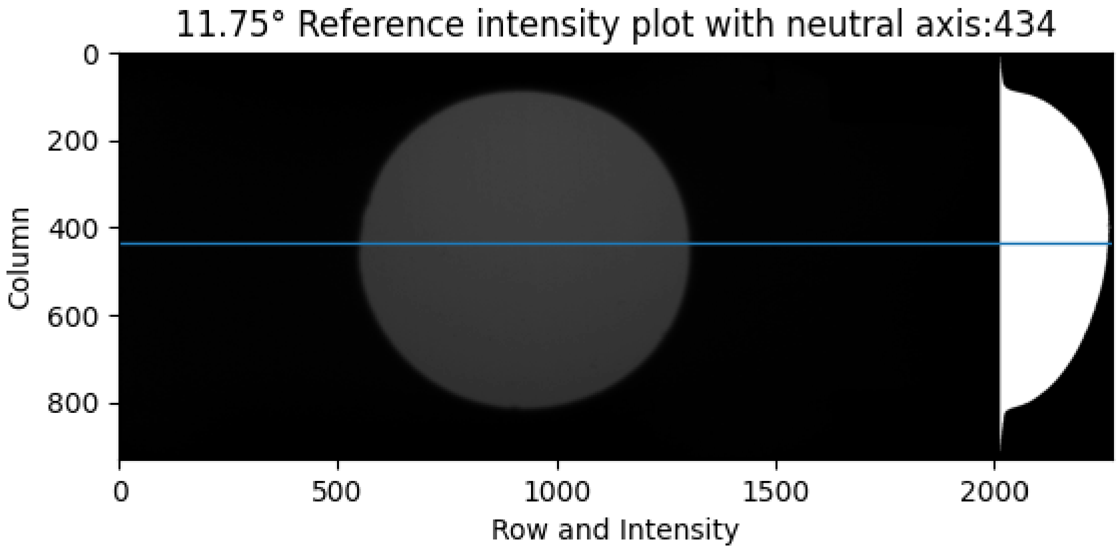




Figure

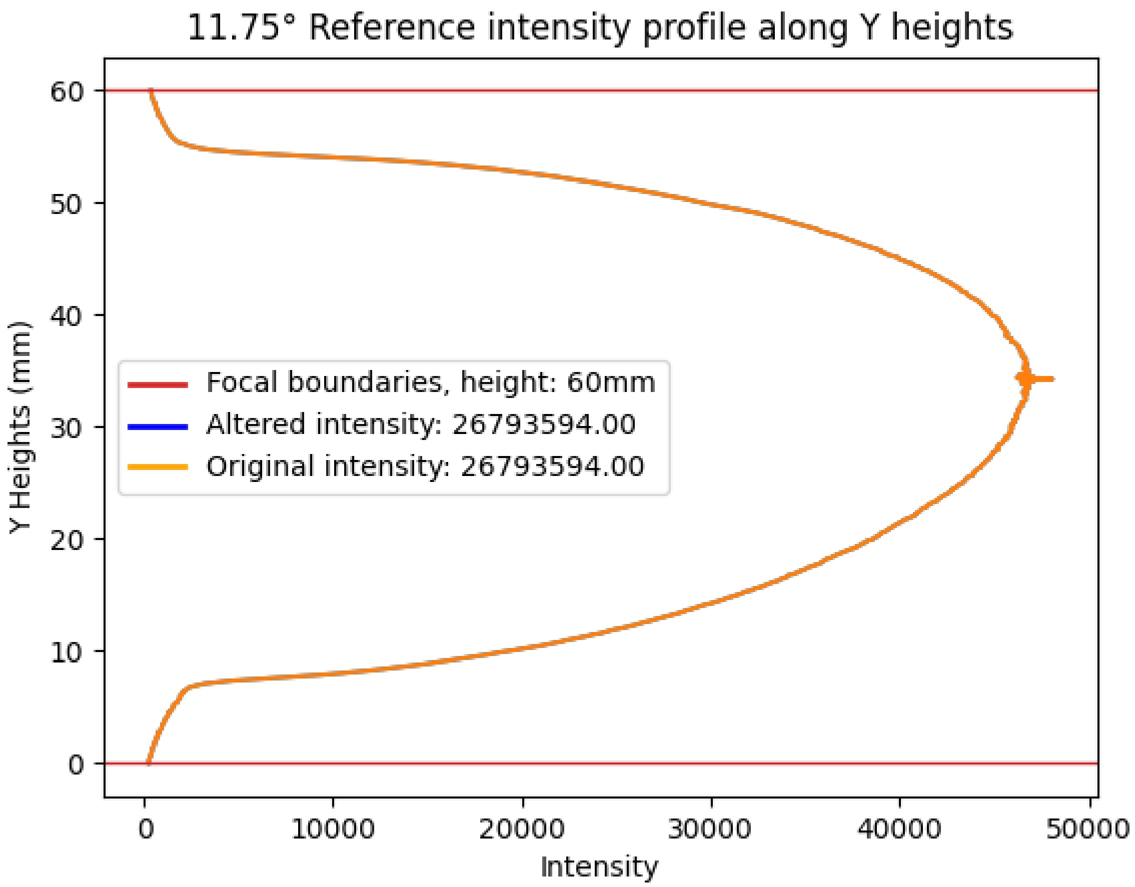

11.75° Etendue intensity plot with neutral axis:489
Column
0
200
400
600
800
0
500
1000
1500
2000
Row and Intensity

Figure



```
An error occurred trying to create min_hline, in 'show focal energy'-function: local variable 'index_to_keep_min' referenced before assignm
ent
 the hline was set to the first index of the list
An error occurred trying to create max_hline, in 'show focal energy'-function: local variable 'index_to_keep_max' referenced before assignm
ent
 the hline was set to the last index of the list
```

Figure



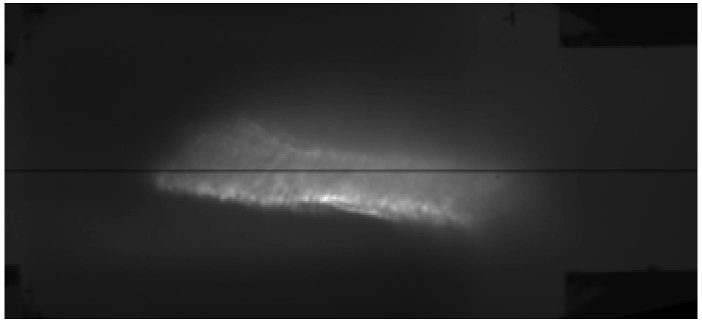
```
referece flux = 26793594
 etendue flux = 14369583
  transmission = 53%
```

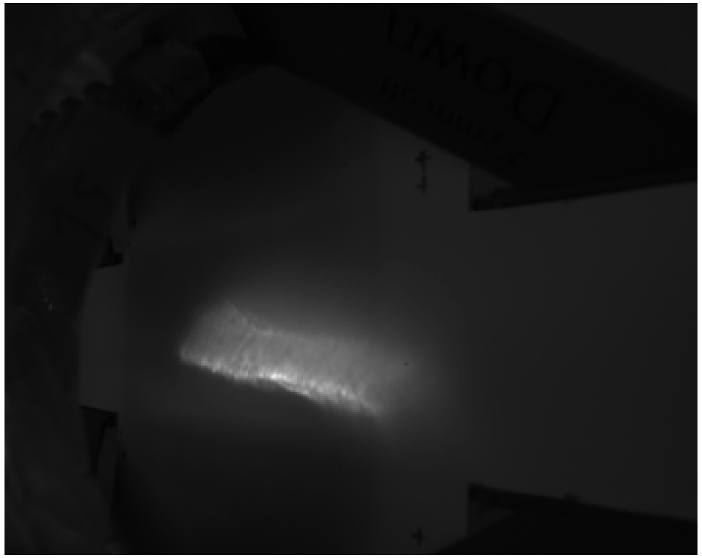
In [301…

```
clicked_points = [(215, 401), (777, 303), (770, 977), (195, 832)]
output_comparison("ImageAnalysis/15iv/23.5.tif","ImageAnalysis/15iv/23.5ref.tif","ImageAnalysis/15iv/plots/23.5deg",27,27)
```

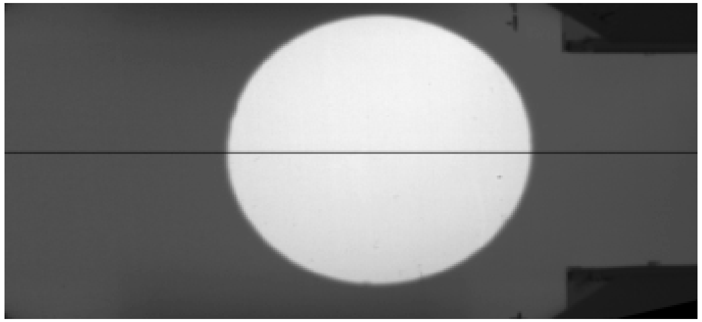
```
Clicked coordinates: [(215, 401), (777, 303), (770, 977), (195, 832)], Testing: False
Clicked coordinates: [(215, 401), (777, 303), (770, 977), (195, 832)], Testing: False
```

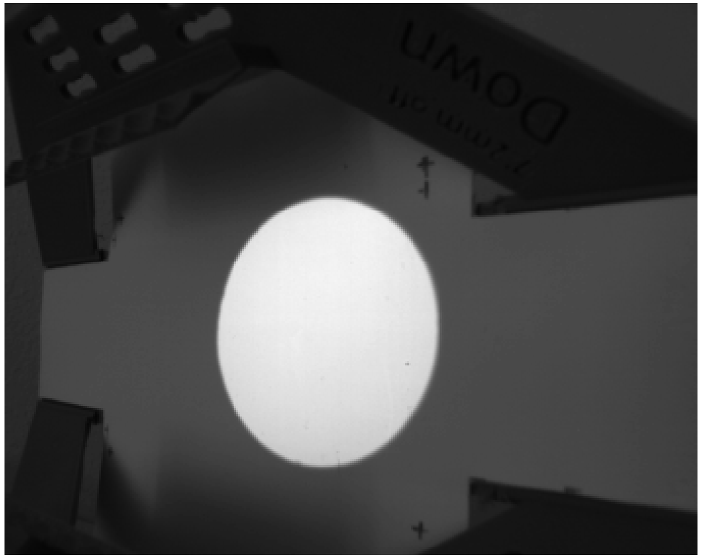
Figure



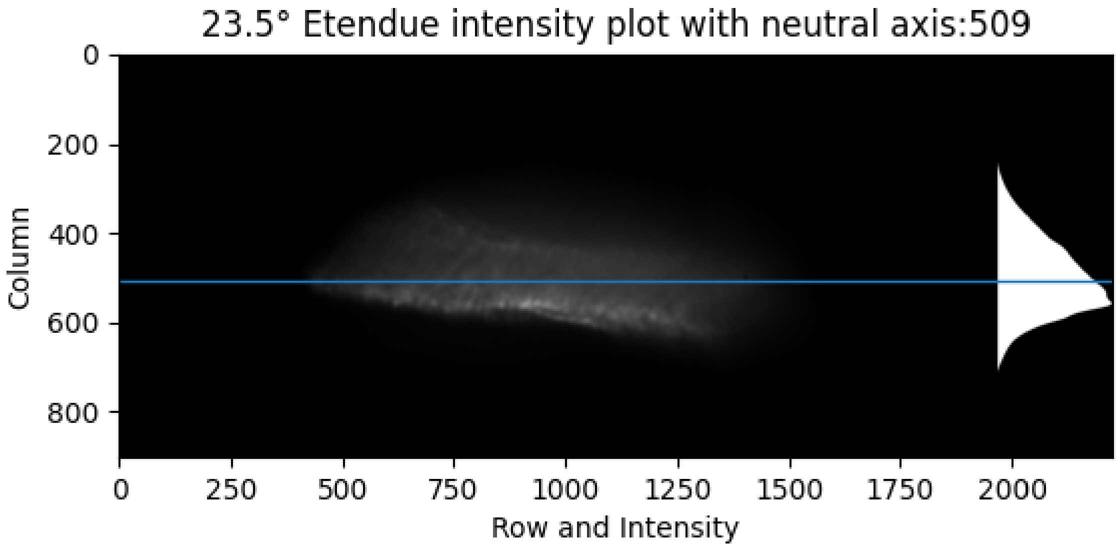


Figure

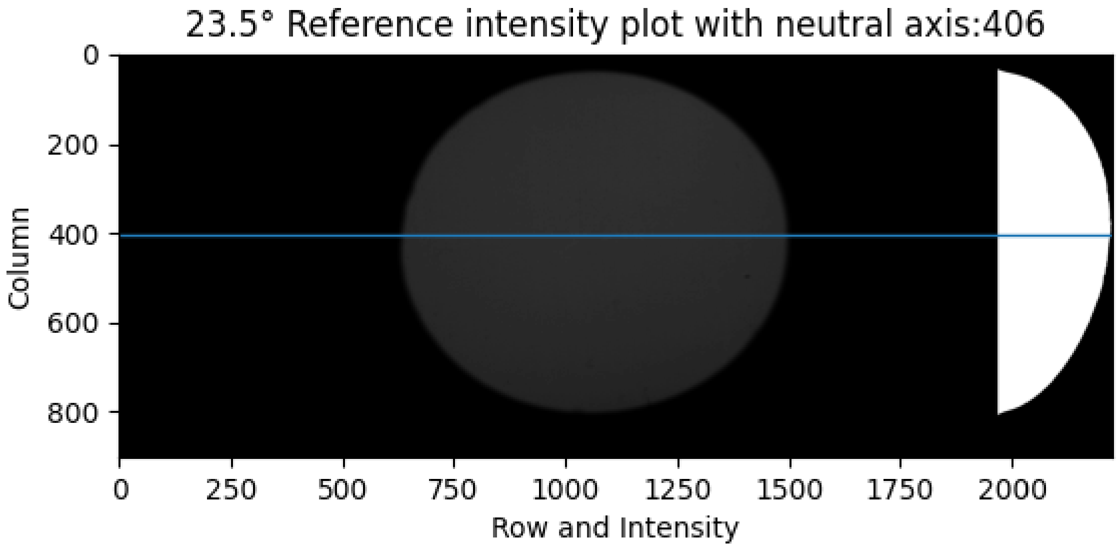




Figure

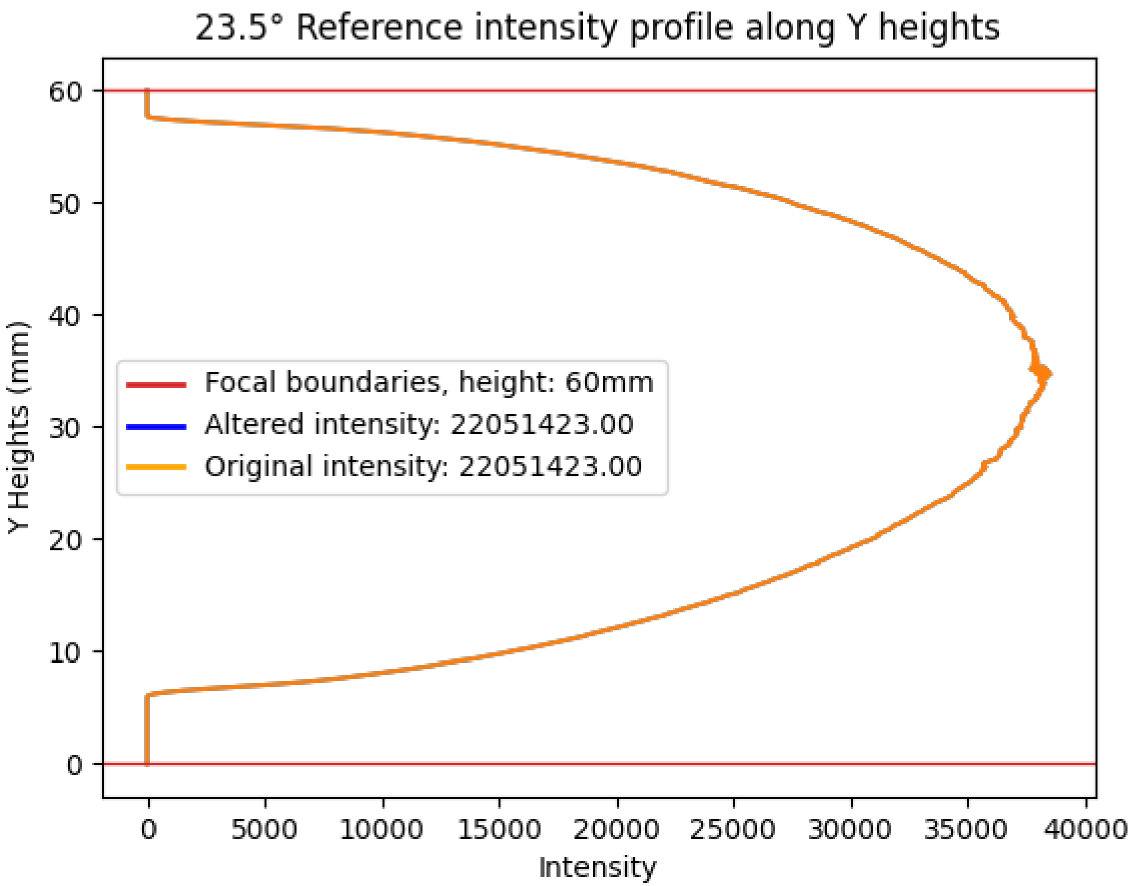

23.5° Etendue intensity plot with neutral axis:509
Column
0
200
400
600
800
0
250
500
750
1000
1250
1500
1750
2000
Row and Intensity

Figure



```
An error occurred trying to create min_hline, in 'show focal energy'-function: local variable 'index_to_keep_min' referenced before assignm
ent
 the hline was set to the first index of the list
An error occurred trying to create max_hline, in 'show focal energy'-function: local variable 'index_to_keep_max' referenced before assignm
ent
 the hline was set to the last index of the list
```

Figure



```
referece flux = 22051423
 etendue flux = 8306431
  transmission = 37%
```

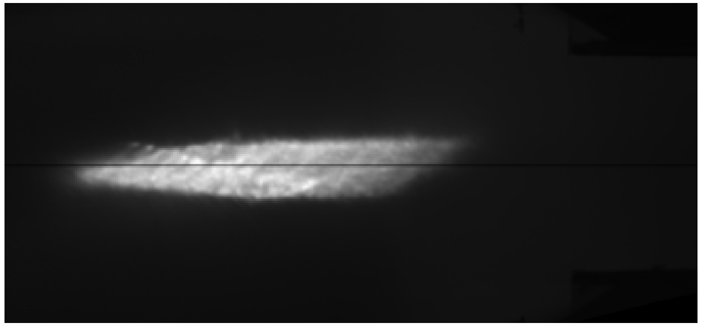
15vii

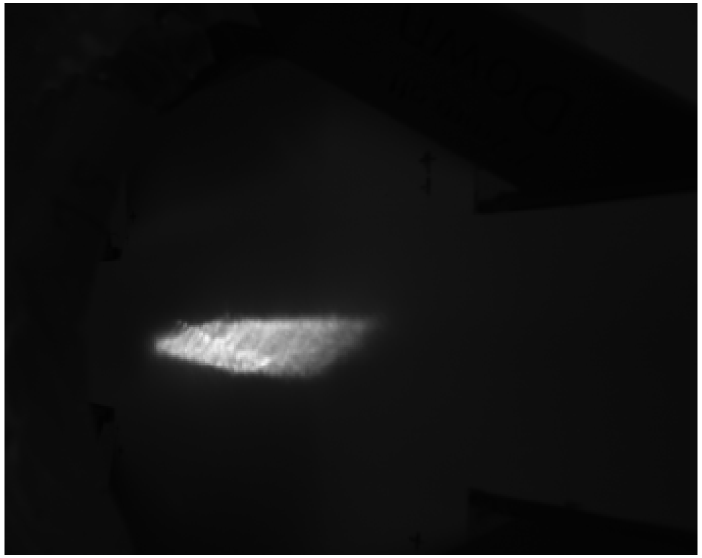
In [302…

```
clicked_points = [(237, 403), (777, 295), (765, 997), (218, 833)]
output_comparison("ImageAnalysis/15vii/0deg.tif","ImageAnalysis/15vii/0degref.tif","ImageAnalysis/15vii/plots/0deg",13,13)
```

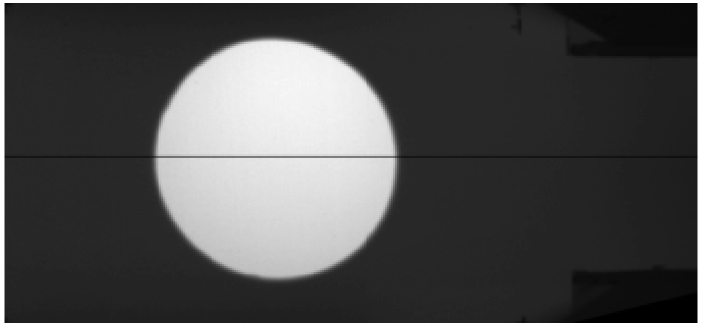
```
Clicked coordinates: [(237, 403), (777, 295), (765, 997), (218, 833)], Testing: False
Clicked coordinates: [(237, 403), (777, 295), (765, 997), (218, 833)], Testing: False
```

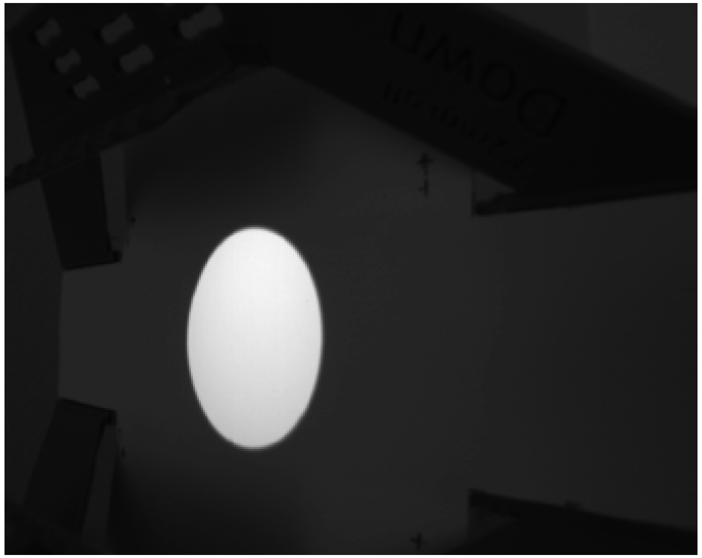
Figure



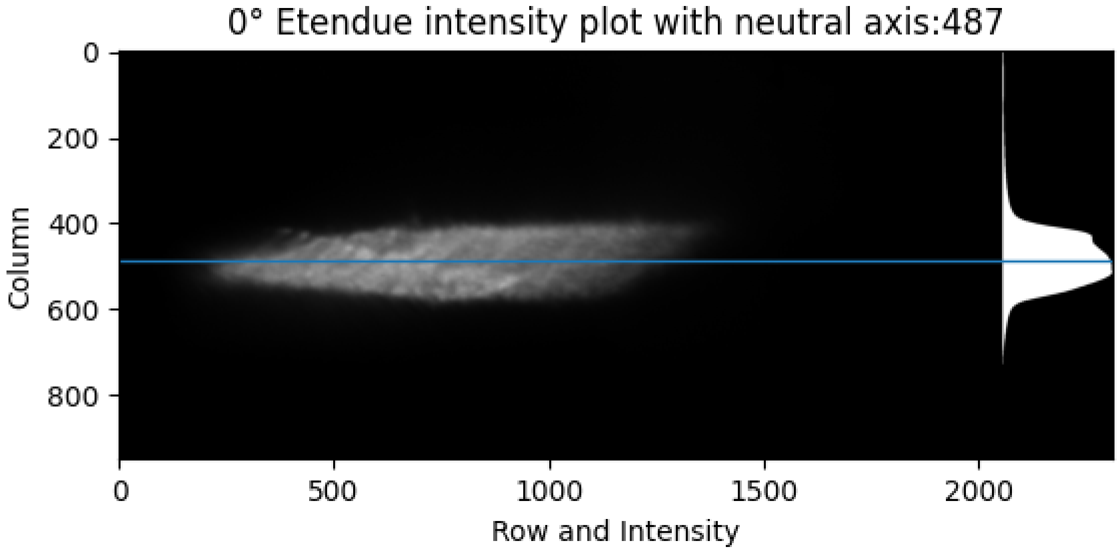


Figure

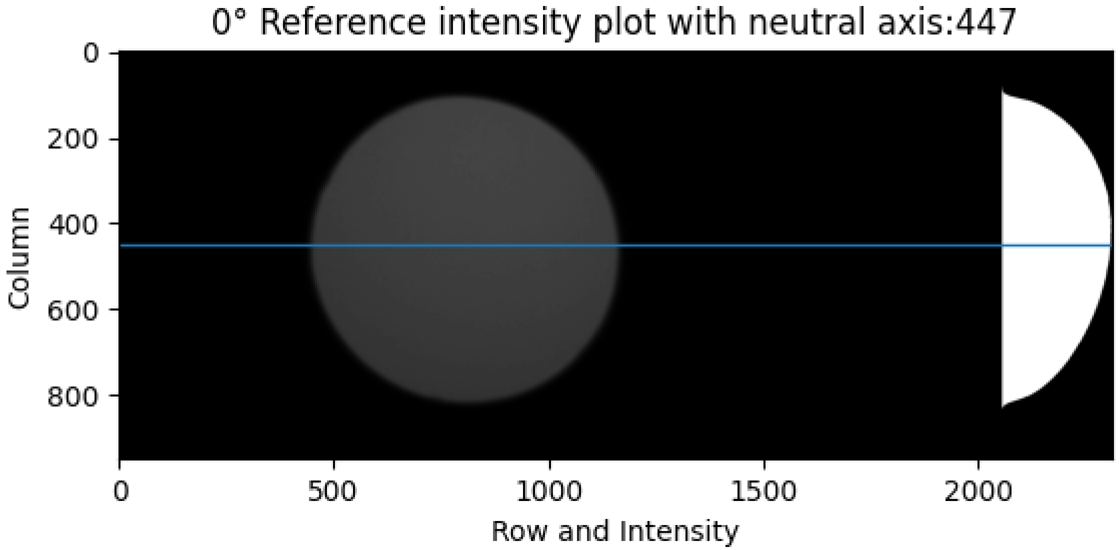




Figure

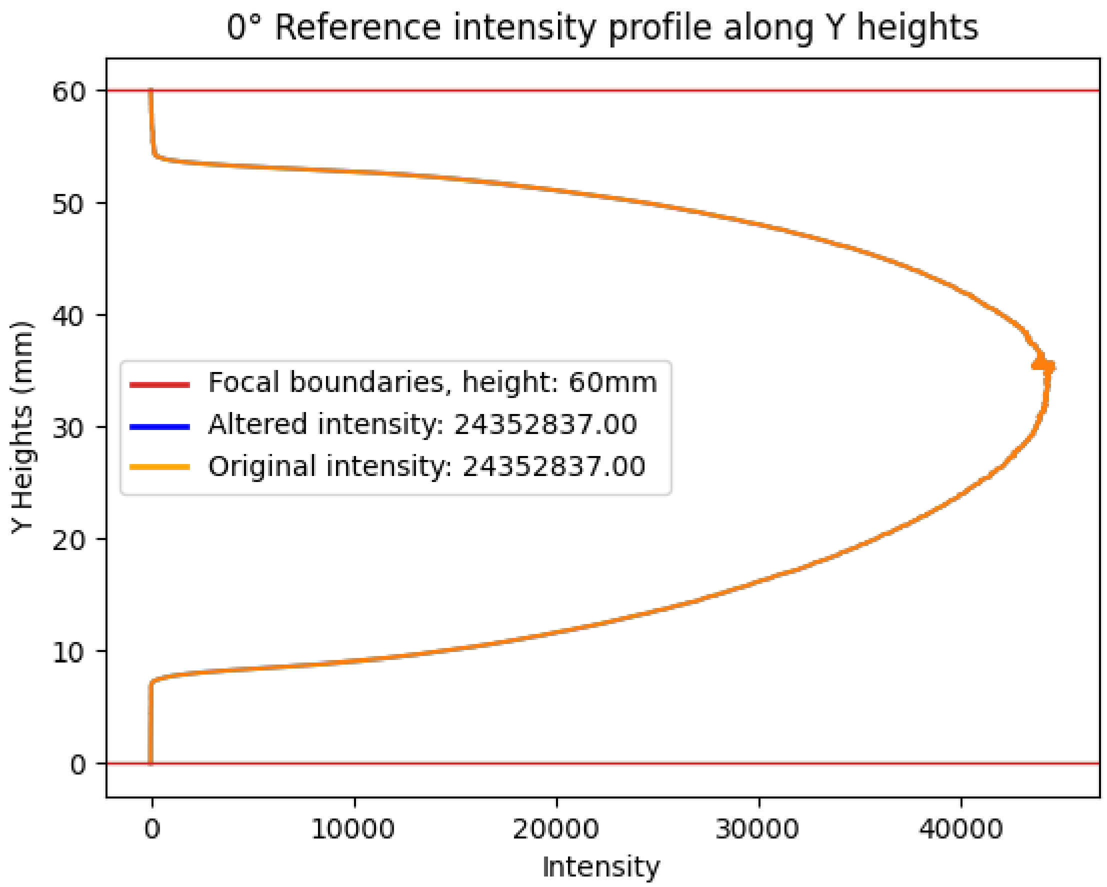

0° Etendue intensity plot with neutral axis:487
Column
0
200
400
600
800
0
500
1000
1500
2000
Row and Intensity

Figure



```
An error occurred trying to create min_hline, in 'show focal energy'-function: local variable 'index_to_keep_min' referenced before assignm
ent
 the hline was set to the first index of the list
An error occurred trying to create max_hline, in 'show focal energy'-function: local variable 'index_to_keep_max' referenced before assignm
ent
 the hline was set to the last index of the list
```

Figure



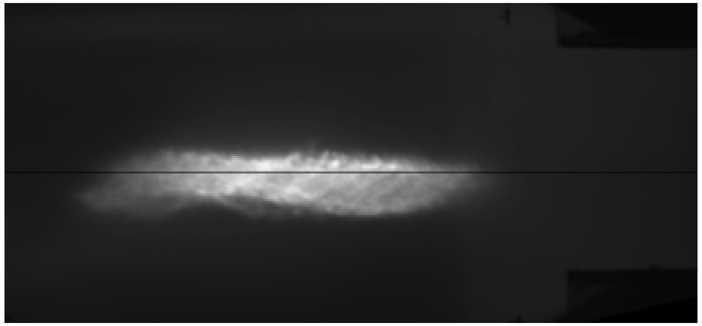
```
referece flux = 24352837
 etendue flux = 17039808
  transmission = 69%
```

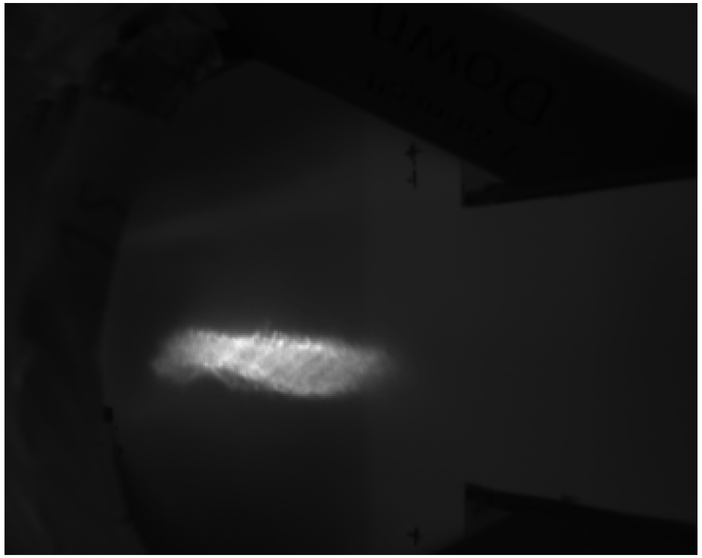
In [304…

```
clicked_points = [(227, 405), (773, 296), (765, 991), (213, 836)]
output_comparison("ImageAnalysis/15vii/11.75.tif","ImageAnalysis/15vii/11.75ref.tif","ImageAnalysis/15vii/plots/11.75deg",0,0)
```

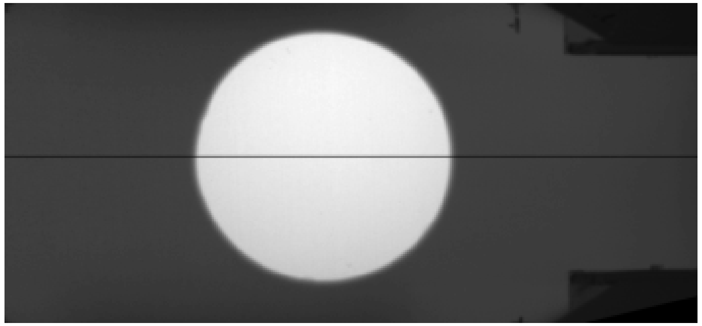
```
Clicked coordinates: [(227, 405), (773, 296), (765, 991), (213, 836)], Testing: False
Clicked coordinates: [(227, 405), (773, 296), (765, 991), (213, 836)], Testing: False
```

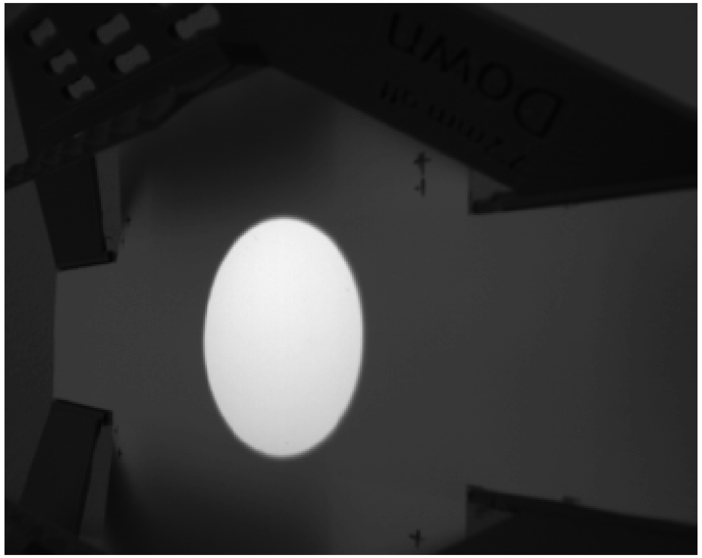
Figure



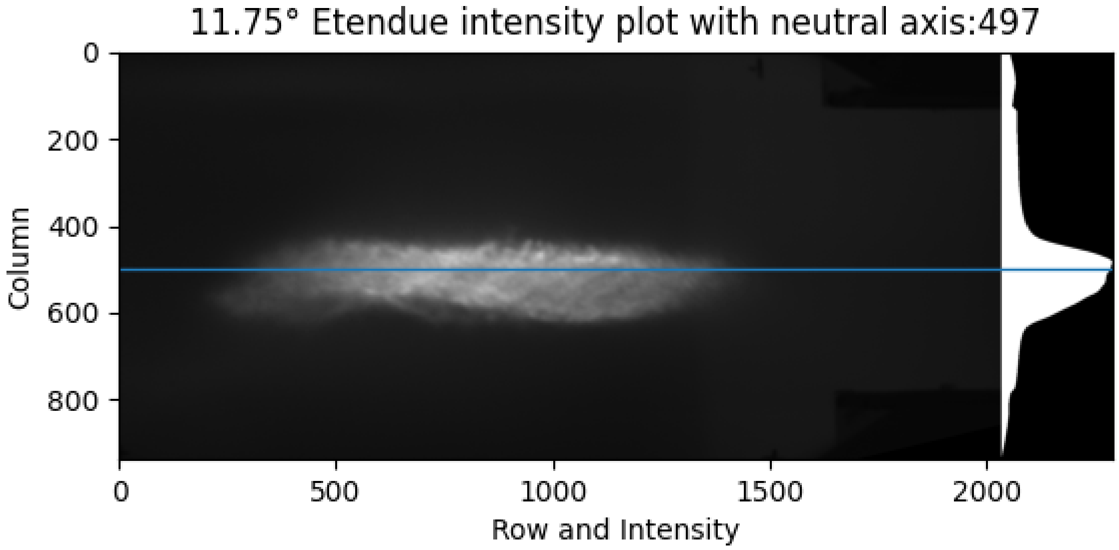


Figure

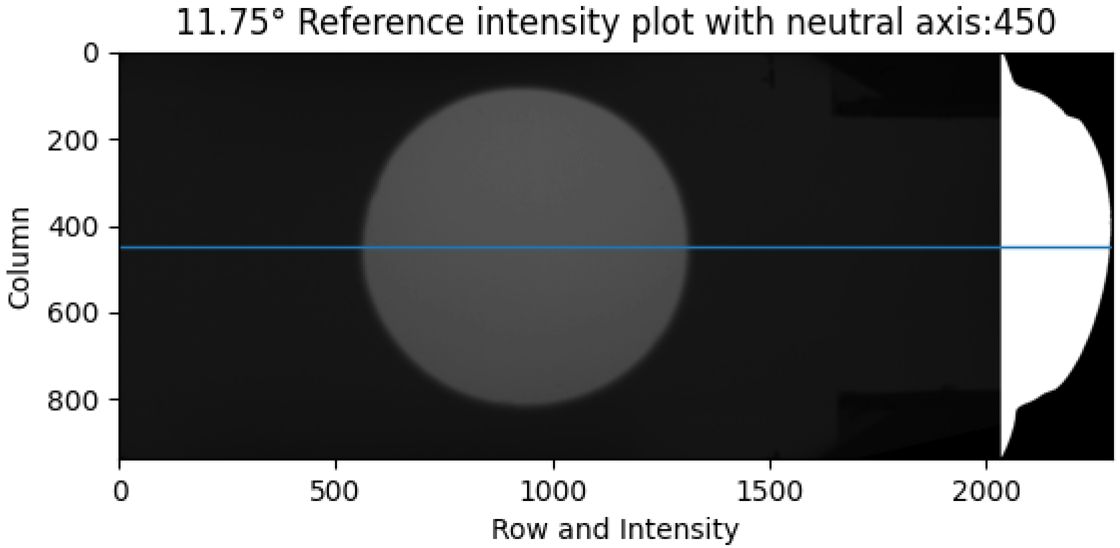




Figure

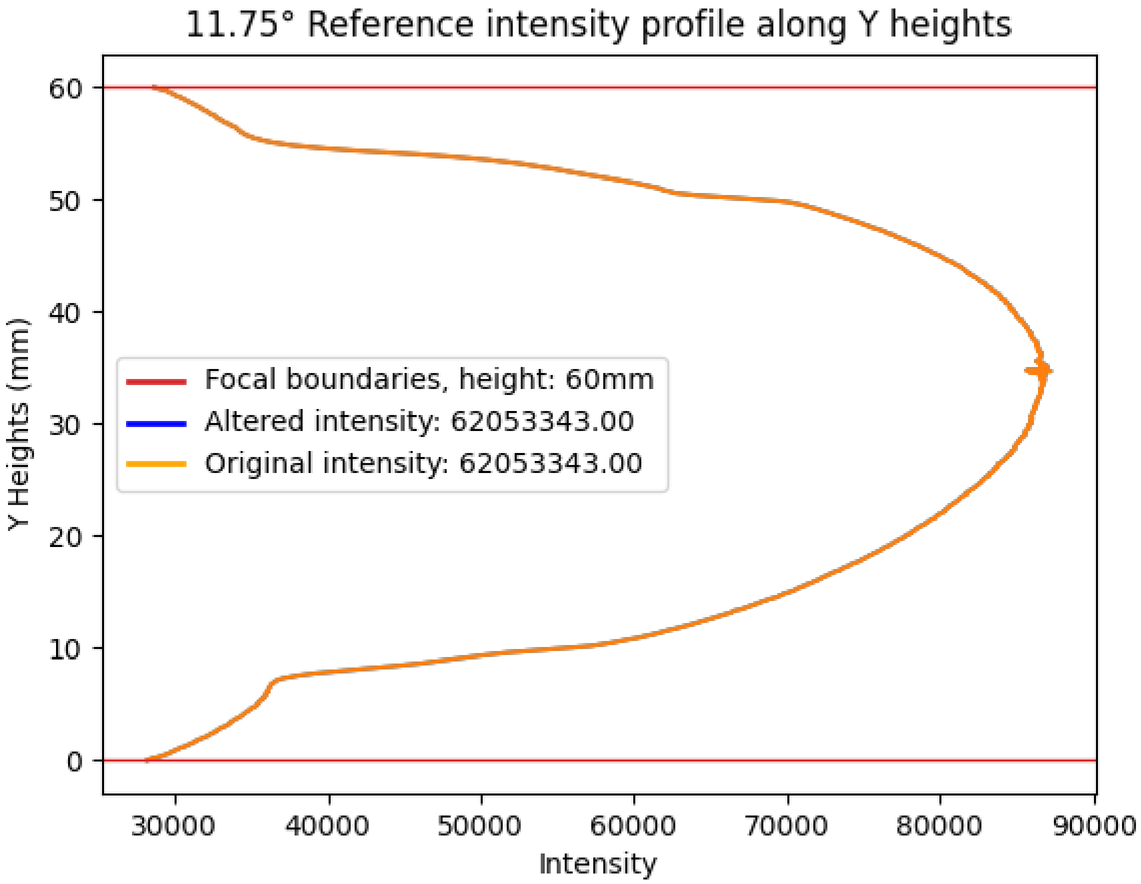

11.75° Etendue intensity plot with neutral axis:497
0
200
400
600
800
Column
0
500
1000
1500
2000
Row and Intensity

Figure



```
An error occurred trying to create min_hline, in 'show focal energy'-function: local variable 'index_to_keep_min' referenced before assignm
ent
 the hline was set to the first index of the list
An error occurred trying to create max_hline, in 'show focal energy'-function: local variable 'index_to_keep_max' referenced before assignm
ent
 the hline was set to the last index of the list
```

Figure



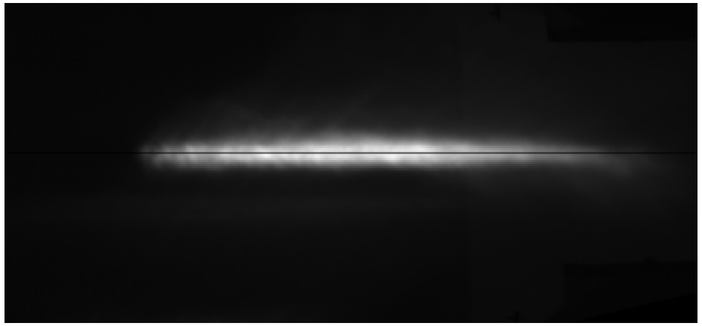
```
referece flux = 62053343
 etendue flux = 23964998
  transmission = 38%
```

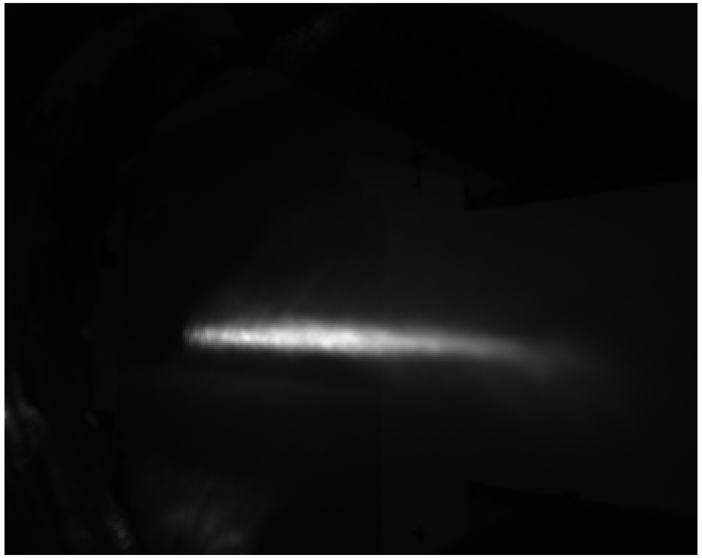
In [305…

```
clicked_points = [(238, 410), (798, 313), (776, 999), (215, 840)]
output_comparison("ImageAnalysis/3dprint/23.5.tif","ImageAnalysis/3dprint/23.5ref.tif","ImageAnalysis/15vii/plots/23.5deg",0,0)
```

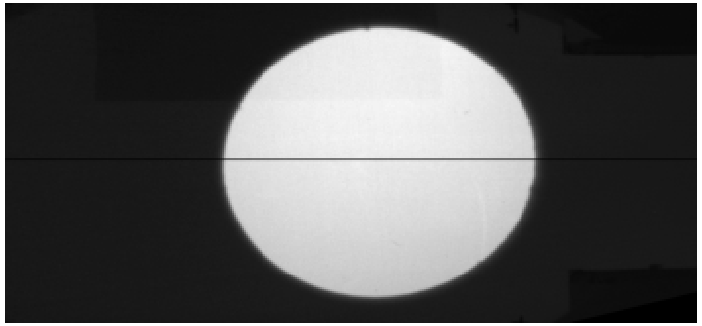
```
Clicked coordinates: [(238, 410), (798, 313), (776, 999), (215, 840)], Testing: False
Clicked coordinates: [(238, 410), (798, 313), (776, 999), (215, 840)], Testing: False
```

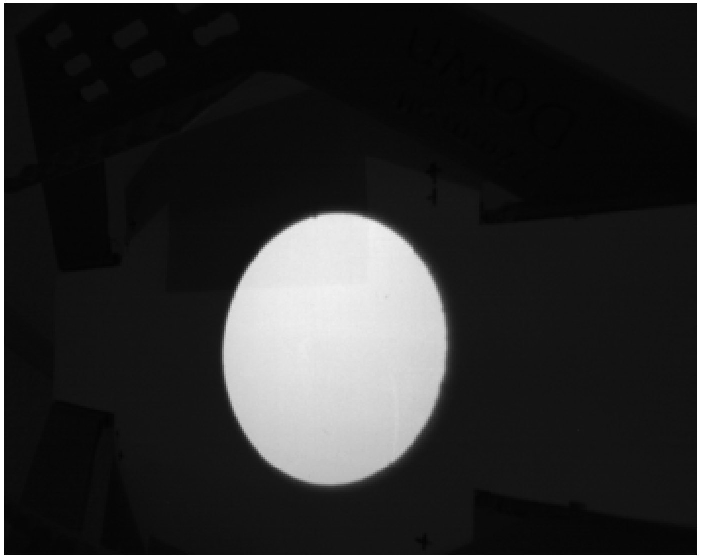
Figure



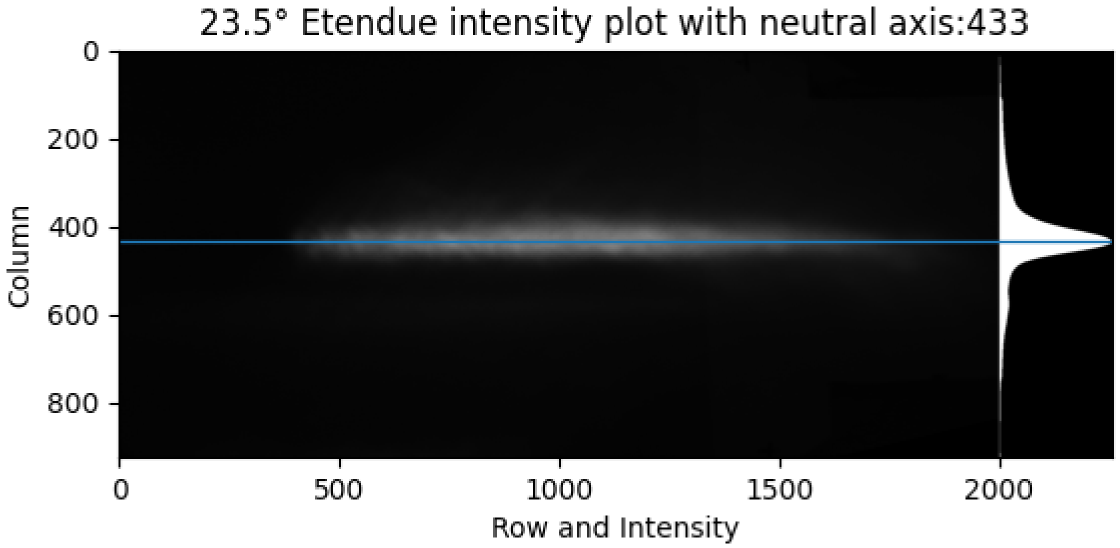


Figure

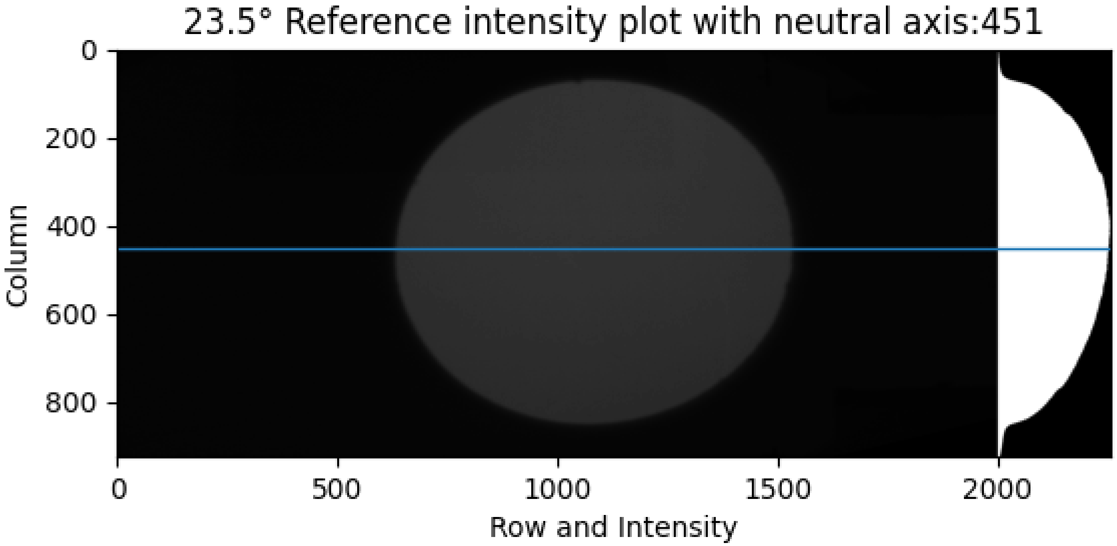




Figure

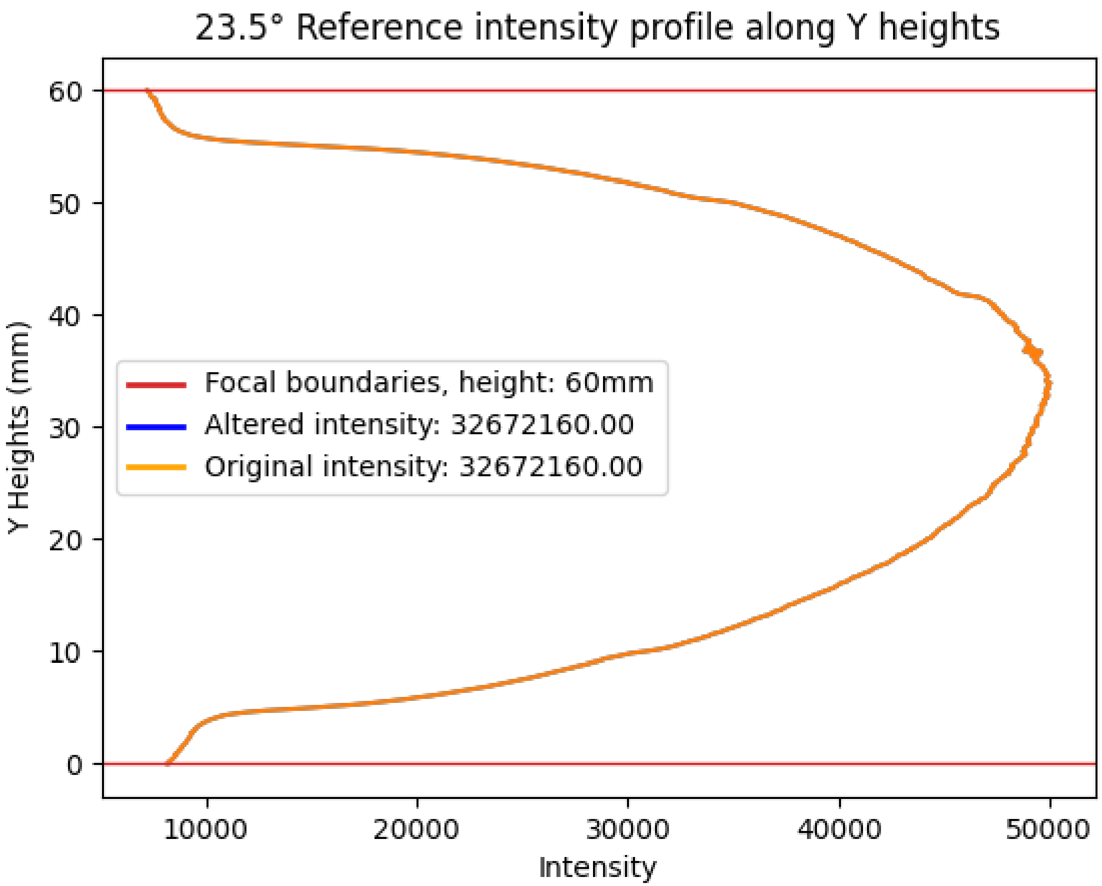

23.5° Etendue intensity plot with neutral axis:433
Column
0
200
400
600
800
0
500
1000
1500
2000
Row and Intensity

Figure



```
An error occurred trying to create min_hline, in 'show focal energy'-function: local variable 'index_to_keep_min' referenced before assignm
ent
 the hline was set to the first index of the list
An error occurred trying to create max_hline, in 'show focal energy'-function: local variable 'index_to_keep_max' referenced before assignm
ent
 the hline was set to the last index of the list
```

Figure



```
referece flux = 32672160
 etendue flux = 11257618
  transmission = 34%
```

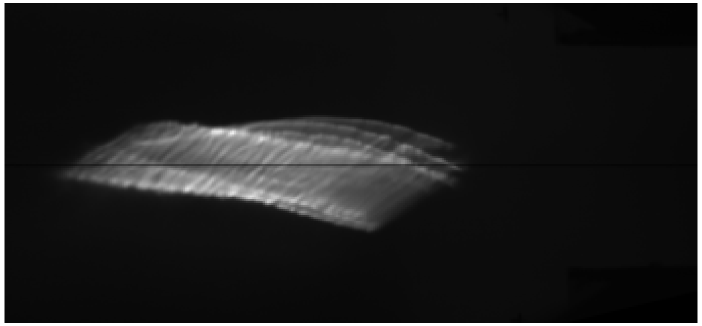
7iv

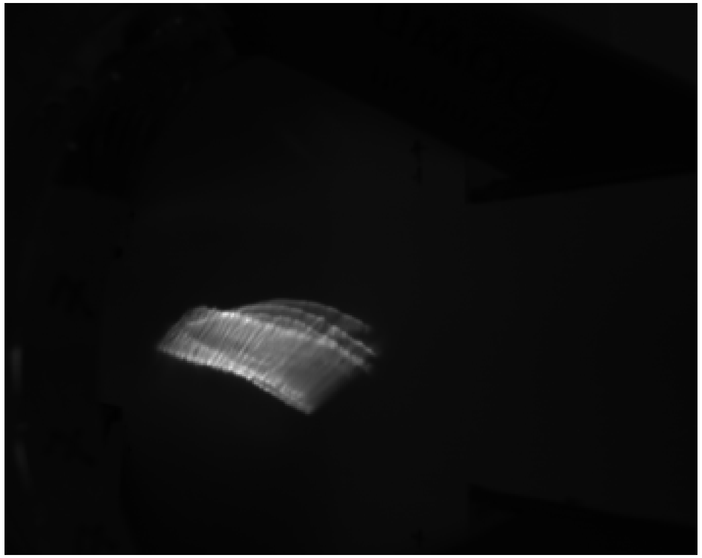
In [292…

```
clicked_points = [(254, 406), (789, 290), (773, 1000), (235, 837)]
output_comparison("ImageAnalysis/7iv/0deg.tif","ImageAnalysis/7iv/0degref.tif","ImageAnalysis/7iv/plots/0deg",0,0)
```

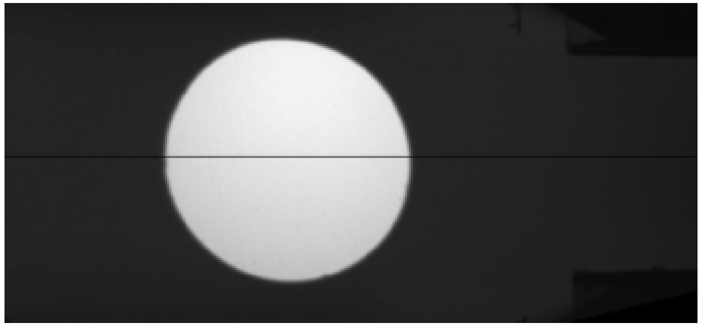
```
Clicked coordinates: [(254, 406), (789, 290), (773, 1000), (235, 837)], Testing: False
Clicked coordinates: [(254, 406), (789, 290), (773, 1000), (235, 837)], Testing: False
```

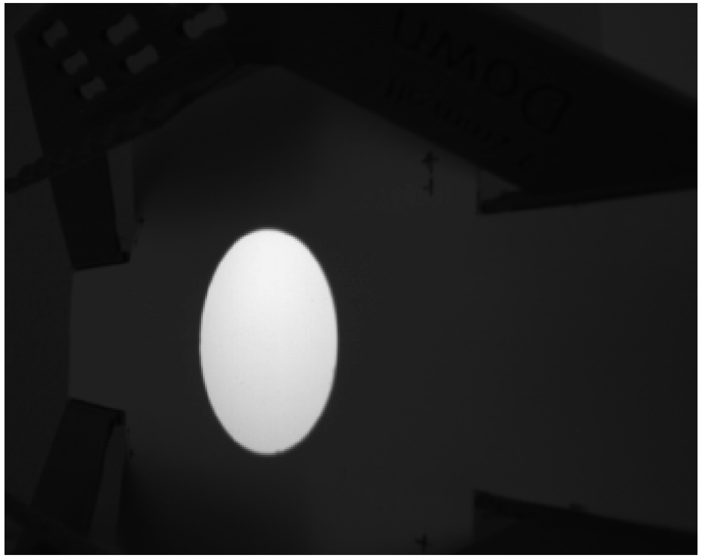
Figure



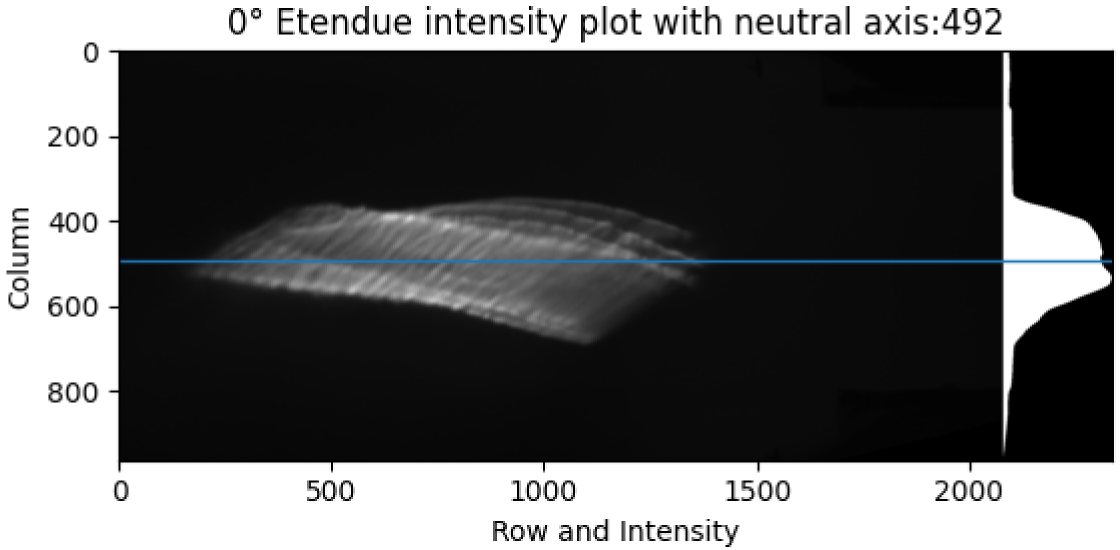


Figure

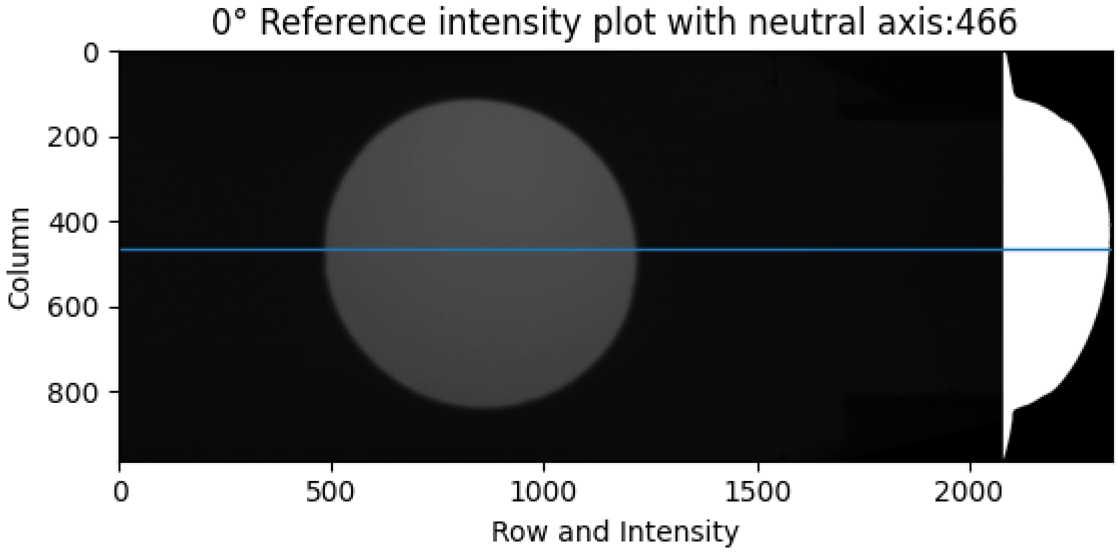




Figure

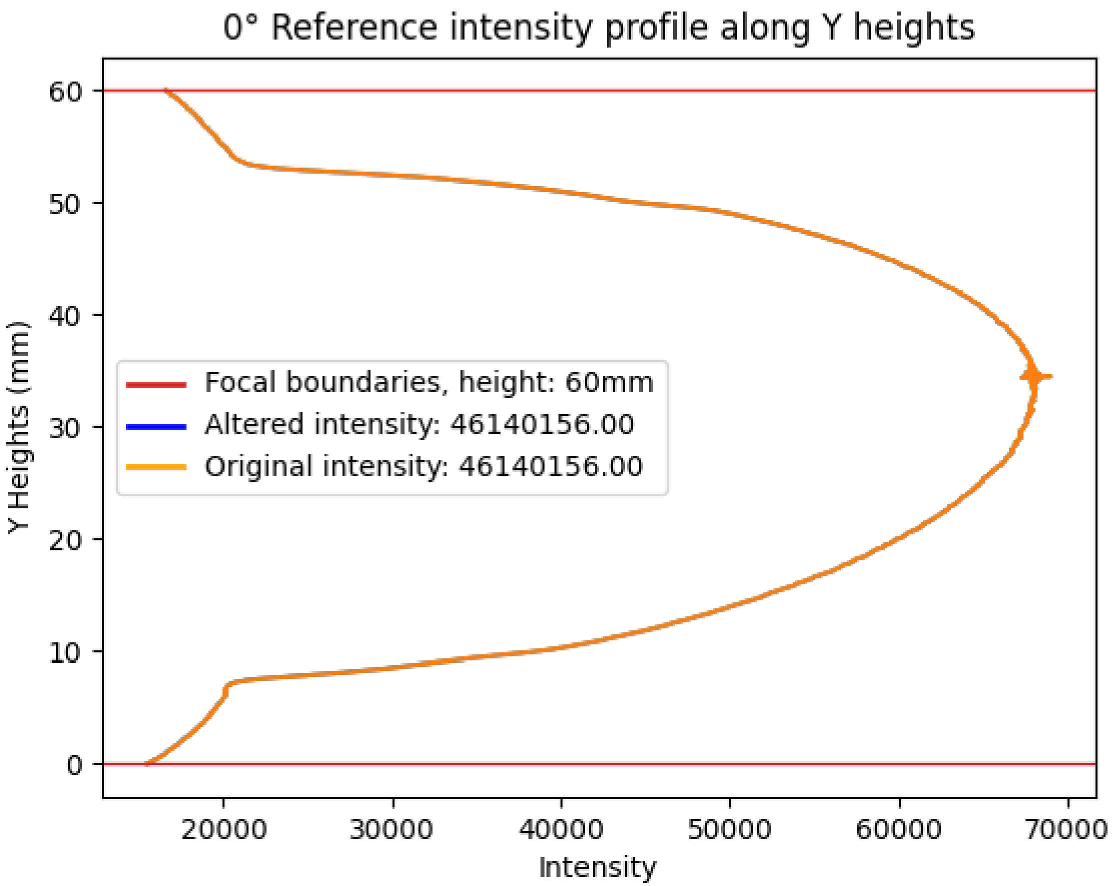

0° Etendue intensity plot with neutral axis:492
0
200
400
600
800
Column
0
500
1000
1500
2000
Row and Intensity

Figure



```
An error occurred trying to create min_hline, in 'show focal energy'-function: local variable 'index_to_keep_min' referenced before assignm
ent
 the hline was set to the first index of the list
An error occurred trying to create max_hline, in 'show focal energy'-function: local variable 'index_to_keep_max' referenced before assignm
ent
 the hline was set to the last index of the list
```

Figure



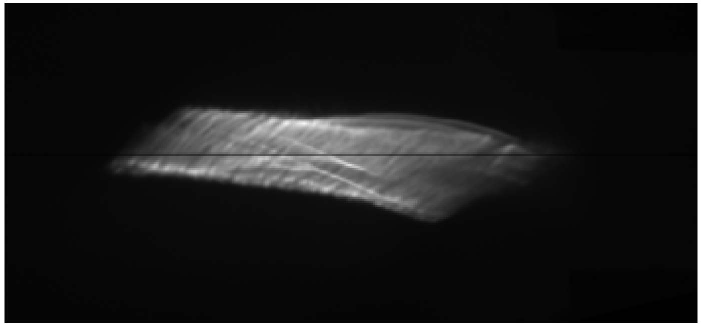
```
referece flux = 46140156
 etendue flux = 21111505
  transmission = 45%
```

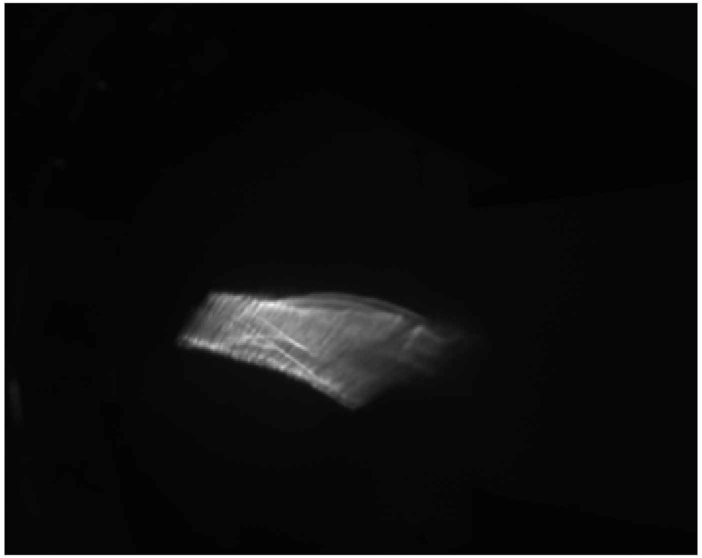
In [308…

```
clicked_points = [(254, 405), (788, 292), (776, 999), (235, 835)]
output_comparison("ImageAnalysis/7iv/11.75.tif","ImageAnalysis/7iv/11.75ref.tif","ImageAnalysis/7iv/plots/11.75deg",0,0)
```

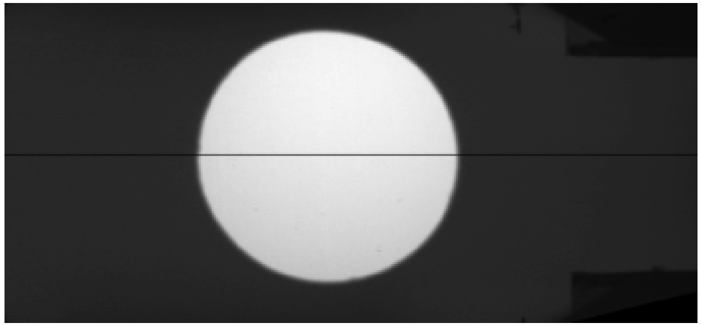
```
Clicked coordinates: [(254, 405), (788, 292), (776, 999), (235, 835)], Testing: False
Clicked coordinates: [(254, 405), (788, 292), (776, 999), (235, 835)], Testing: False
```

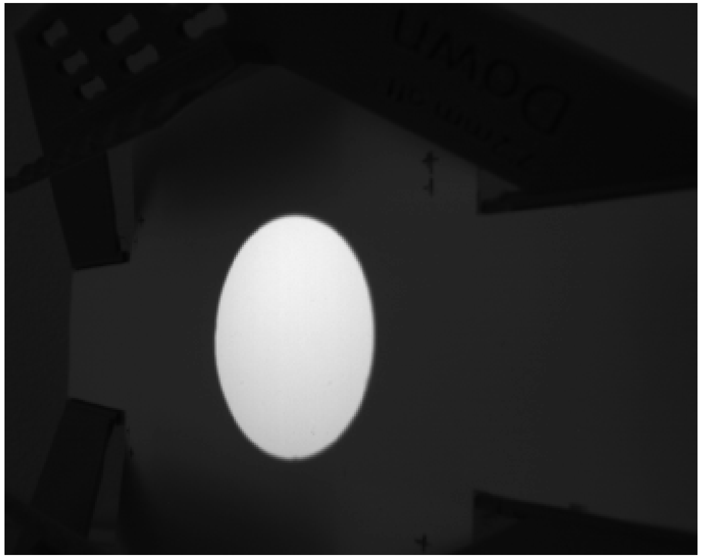
Figure



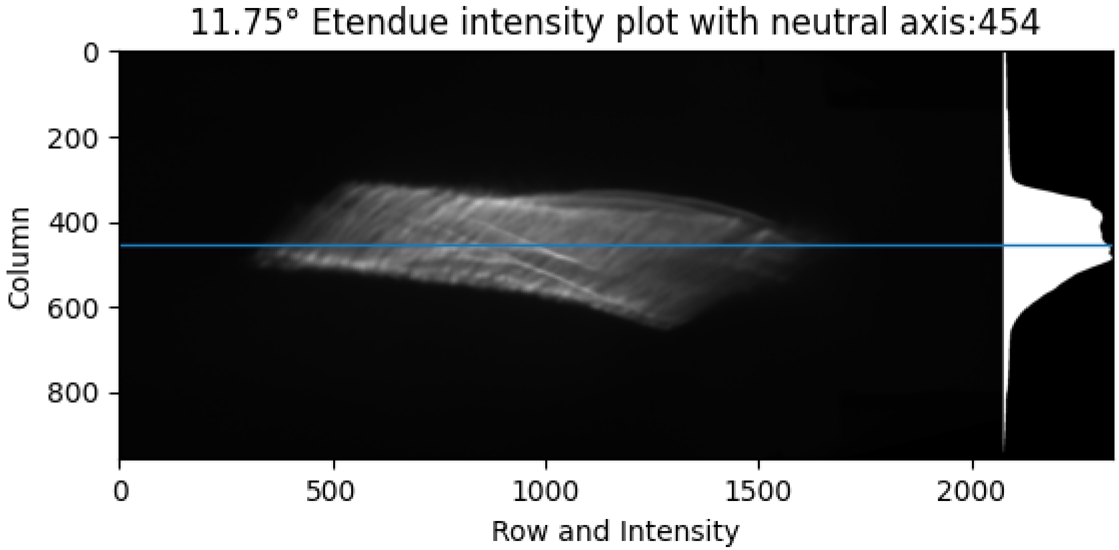


Figure

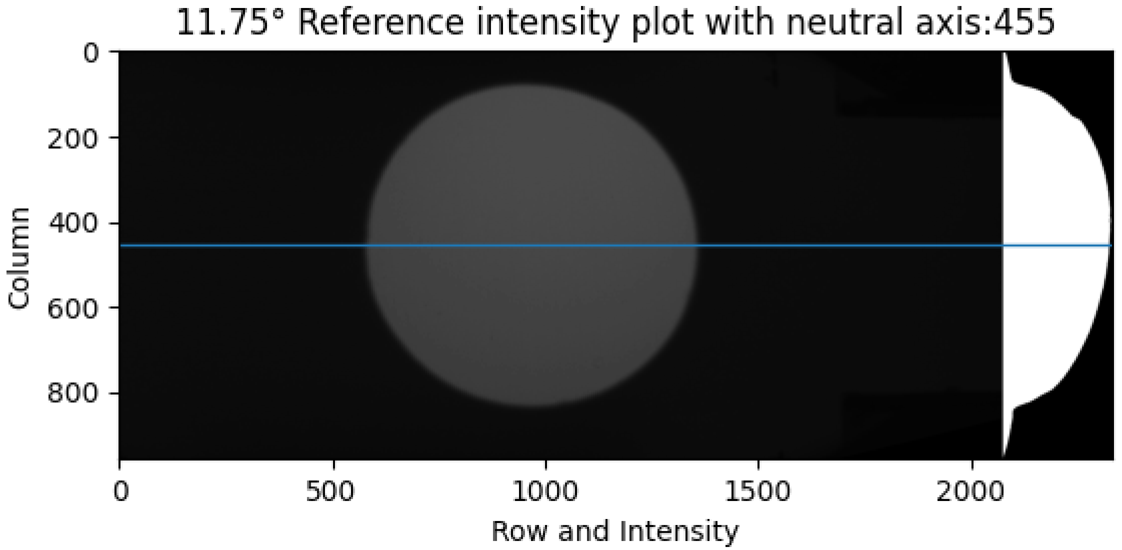




Figure

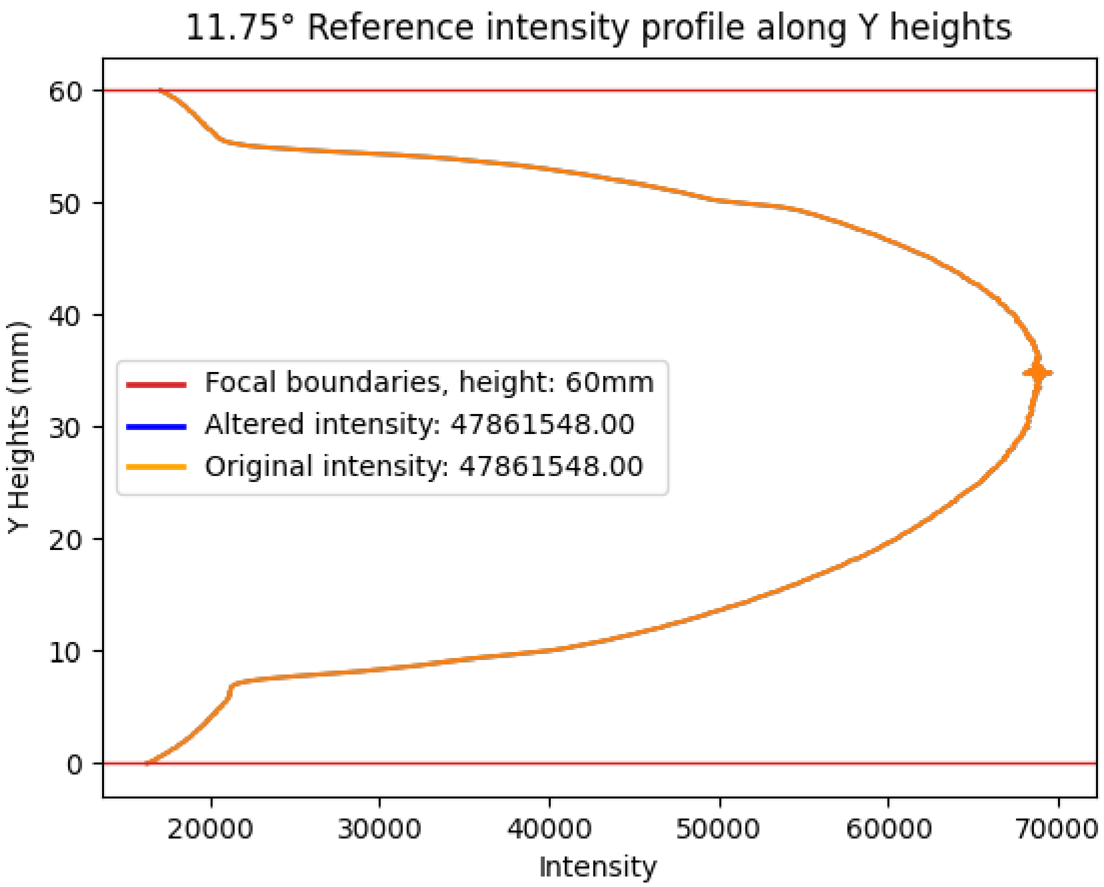

11.75° Etendue intensity plot with neutral axis:454
0
200
400
600
800
Column
0
500
1000
1500
2000
Row and Intensity

Figure



```
An error occurred trying to create min_hline, in 'show focal energy'-function: local variable 'index_to_keep_min' referenced before assignm
ent
 the hline was set to the first index of the list
An error occurred trying to create max_hline, in 'show focal energy'-function: local variable 'index_to_keep_max' referenced before assignm
ent
 the hline was set to the last index of the list
```

Figure



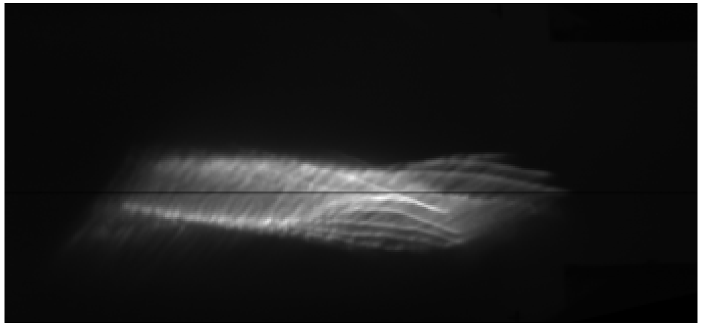
```
referece flux = 47861548
 etendue flux = 19556457
  transmission = 40%
```

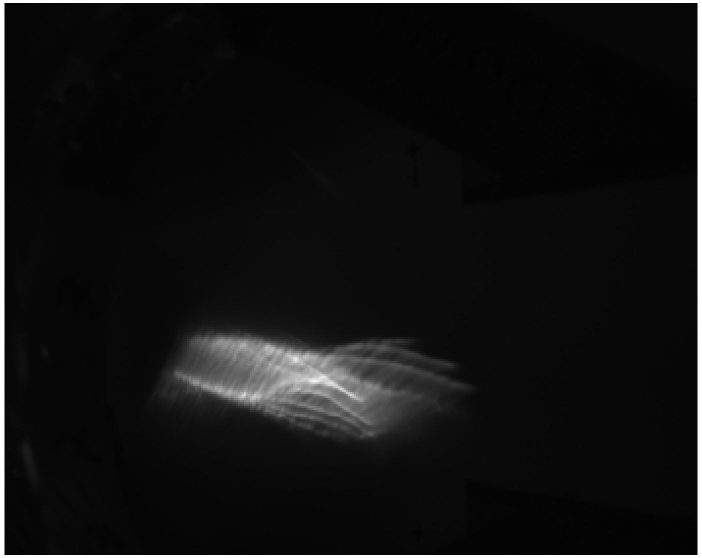
In [311…

```
clicked_points = [(240, 406), (788, 302), (775, 998), (219, 836)]
output_comparison("ImageAnalysis/7iv/23.5.tif","ImageAnalysis/7iv/23.5ref.tif","ImageAnalysis/7iv/plots/23.5deg",0,0)
```

```
Clicked coordinates: [(240, 406), (788, 302), (775, 998), (219, 836)], Testing: False
Clicked coordinates: [(240, 406), (788, 302), (775, 998), (219, 836)], Testing: False
```

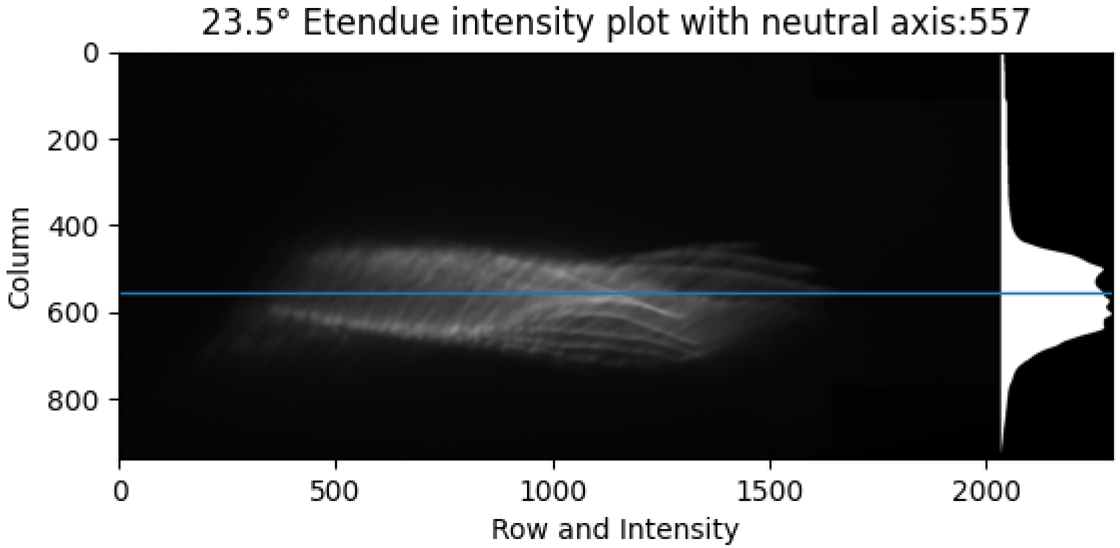
Figure



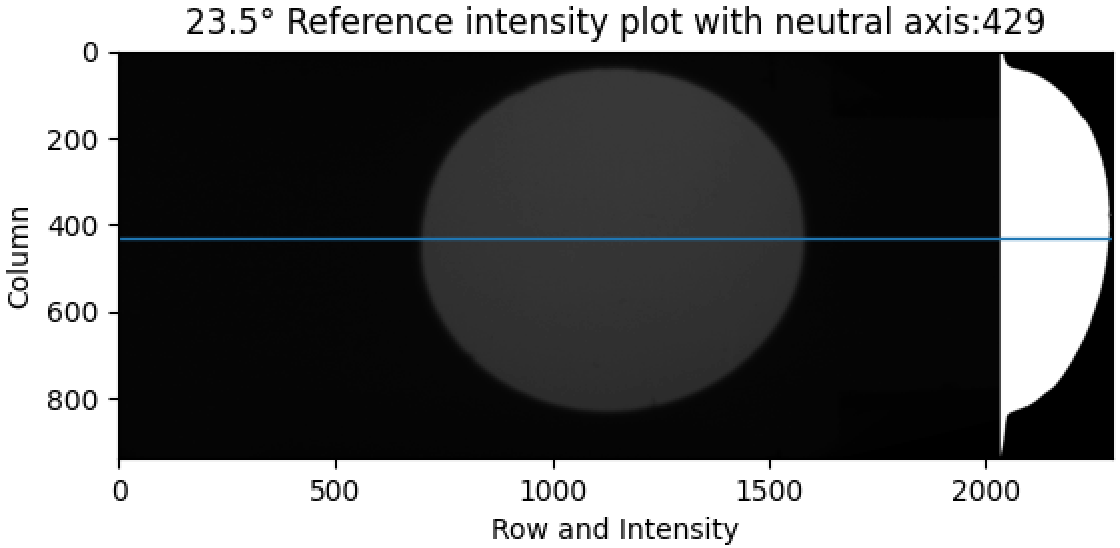


Figure

Figure

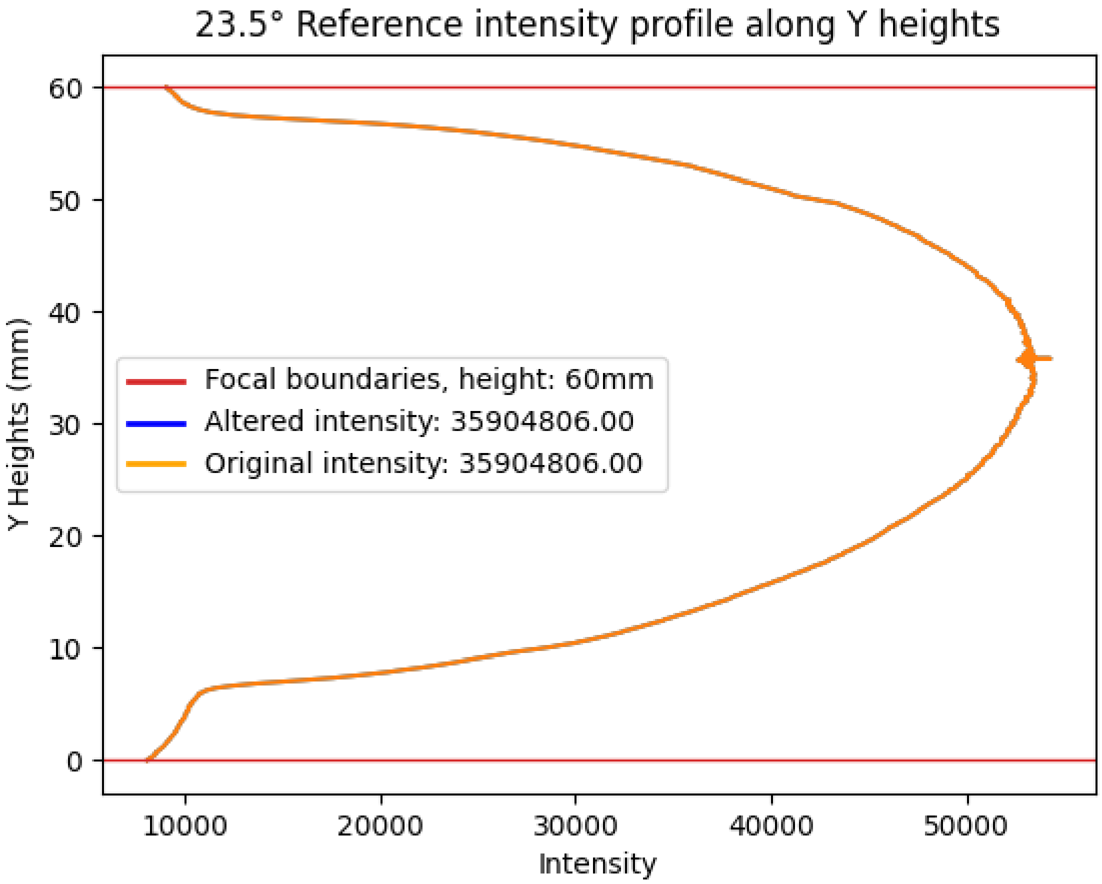

23.5° Etendue intensity plot with neutral axis:557
Column
0
200
400
600
800
0
500
1000
1500
2000
Row and Intensity

Figure



```
An error occurred trying to create min_hline, in 'show focal energy'-function: local variable 'index_to_keep_min' referenced before assignm
ent
 the hline was set to the first index of the list
An error occurred trying to create max_hline, in 'show focal energy'-function: local variable 'index_to_keep_max' referenced before assignm
ent
 the hline was set to the last index of the list
```

Figure



```
referece flux = 35904806
 etendue flux = 16392789
  transmission = 45%
```

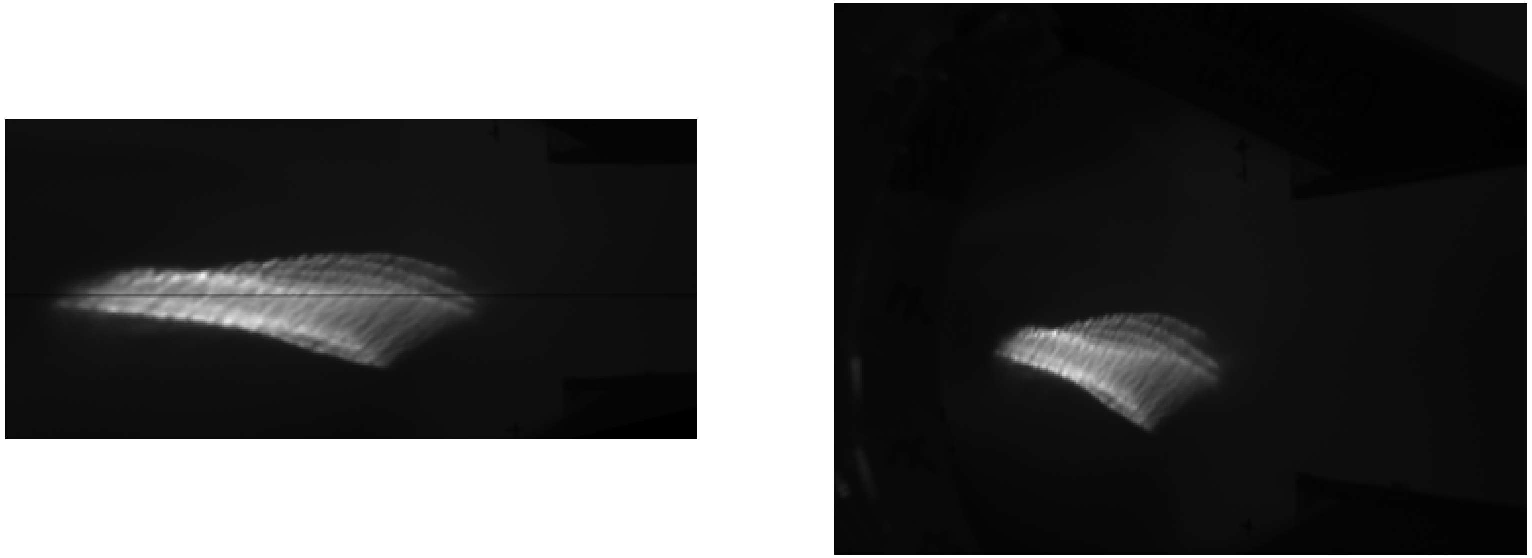
7vii

In [312…

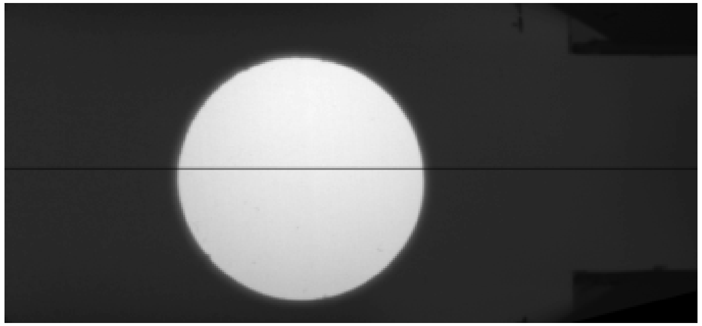
```
plt.close()
clicked_points = [(277, 399), (795, 276), (782, 998), (255, 828)]
output_comparison("ImageAnalysis/7vii/0deg.tif","ImageAnalysis/7vii/0degref.tif","ImageAnalysis/7vii/plots",0,0)
```

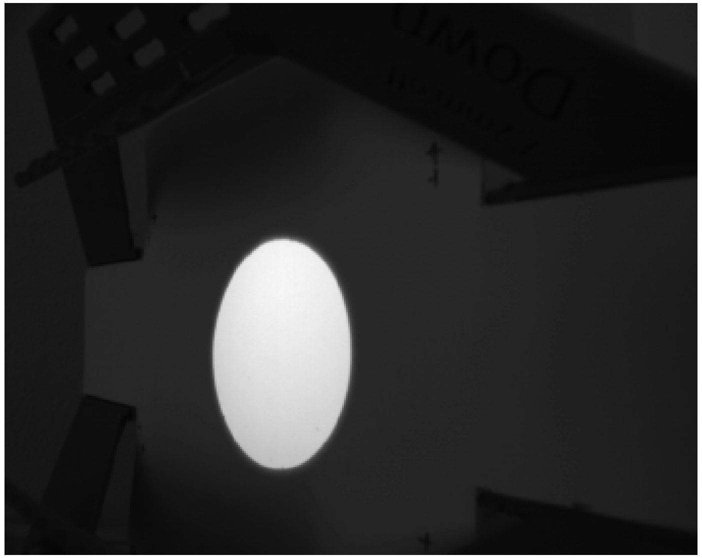
```
Clicked coordinates: [(277, 399), (795, 276), (782, 998), (255, 828)], Testing: False
Clicked coordinates: [(277, 399), (795, 276), (782, 998), (255, 828)], Testing: False
```

Figure

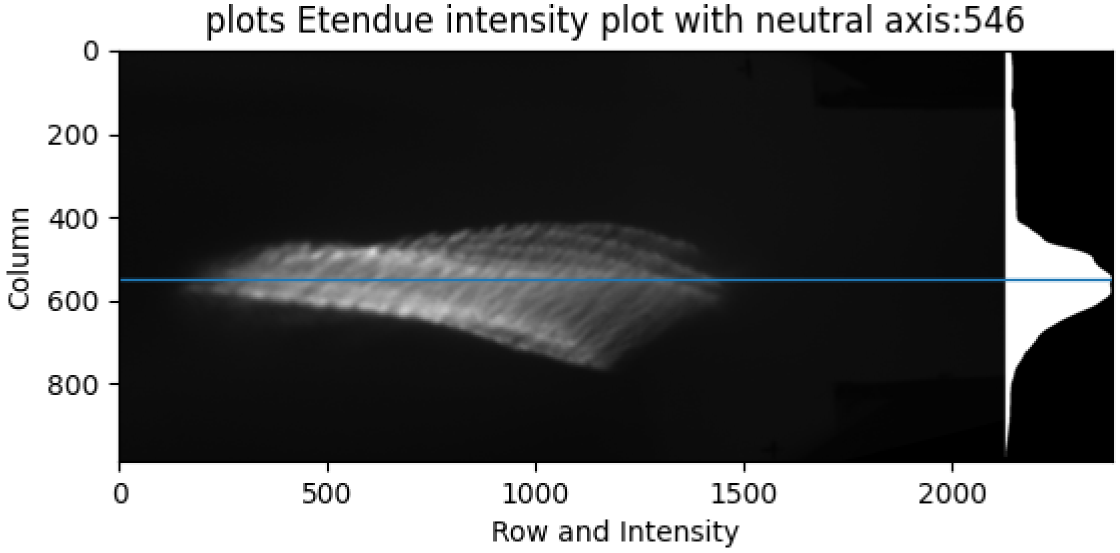


Figure

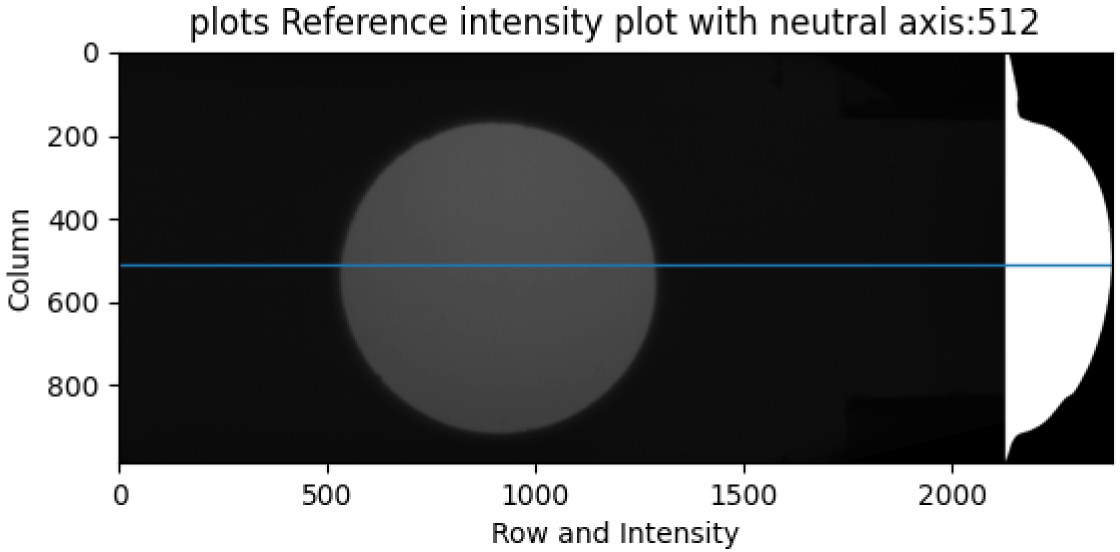




Figure

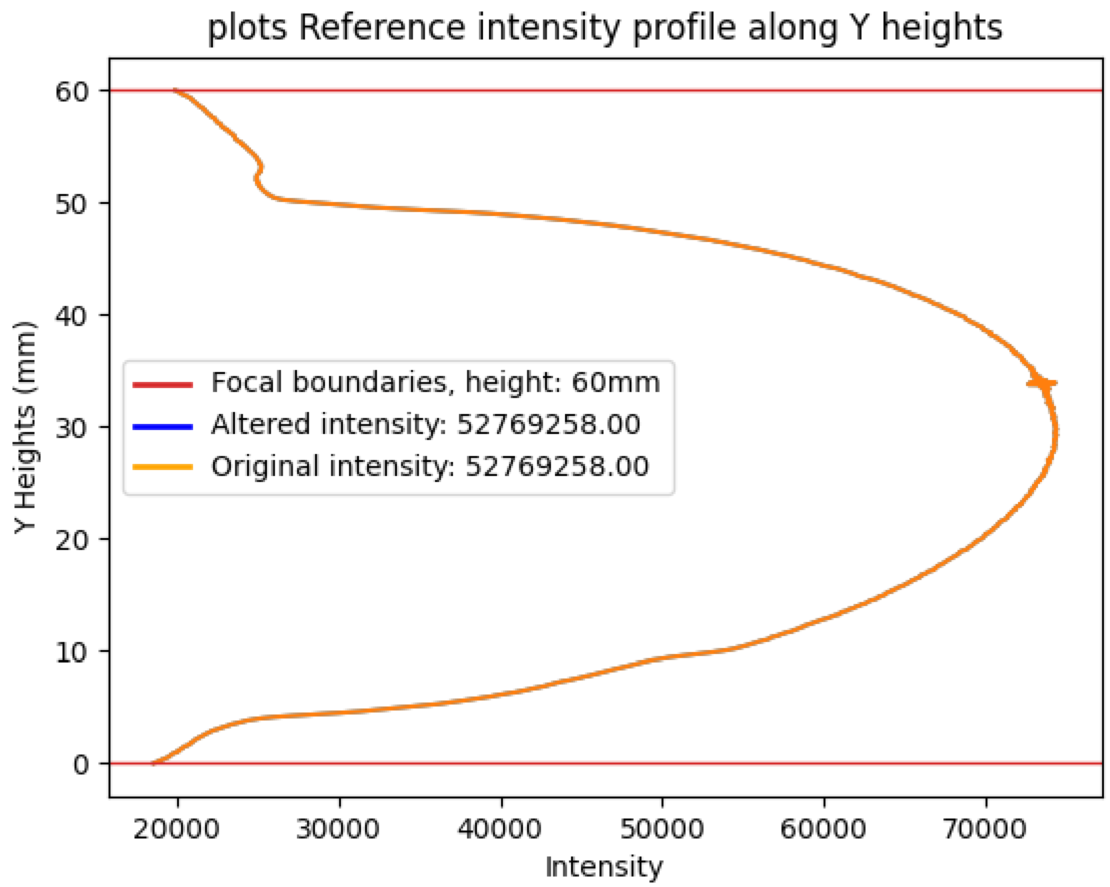

plots Etendue intensity plot with neutral axis:546
0
200
400
600
800
Column
0
500
1000
1500
2000
Row and Intensity

Figure



```
An error occurred trying to create min_hline, in 'show focal energy'-function: local variable 'index_to_keep_min' referenced before assignm
ent
 the hline was set to the first index of the list
An error occurred trying to create max_hline, in 'show focal energy'-function: local variable 'index_to_keep_max' referenced before assignm
ent
 the hline was set to the last index of the list
```

Figure



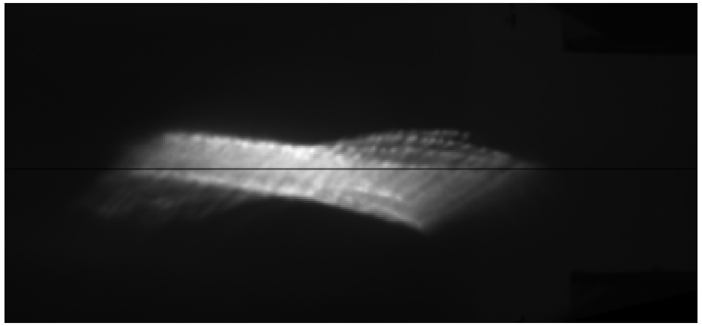
```
referece flux = 52769258
 etendue flux = 25067450
  transmission = 47%
```

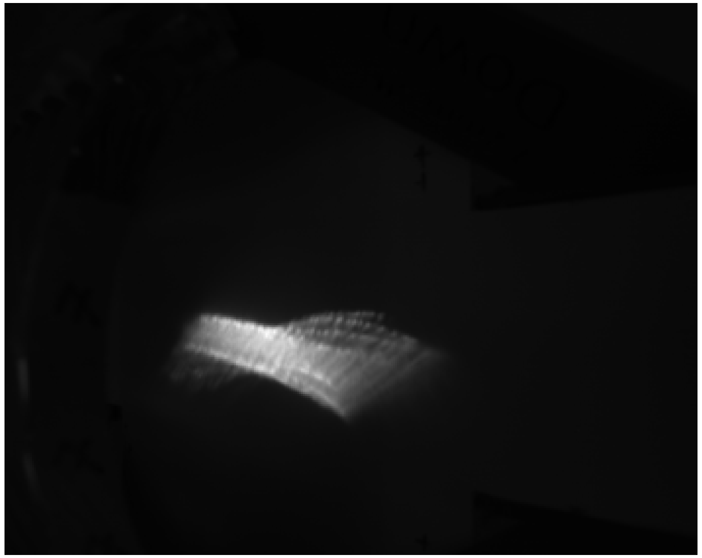
In [314…

```
plt.close()
clicked_points = [(252, 407), (784, 293), (779, 1002), (239, 836)]
output_comparison("ImageAnalysis/7vii/11.75.tif","ImageAnalysis/7vii/11.75ref.tif","ImageAnalysis/7vii/plots/11.75deg",0,0)
```

```
Clicked coordinates: [(252, 407), (784, 293), (779, 1002), (239, 836)], Testing: False
Clicked coordinates: [(252, 407), (784, 293), (779, 1002), (239, 836)], Testing: False
```

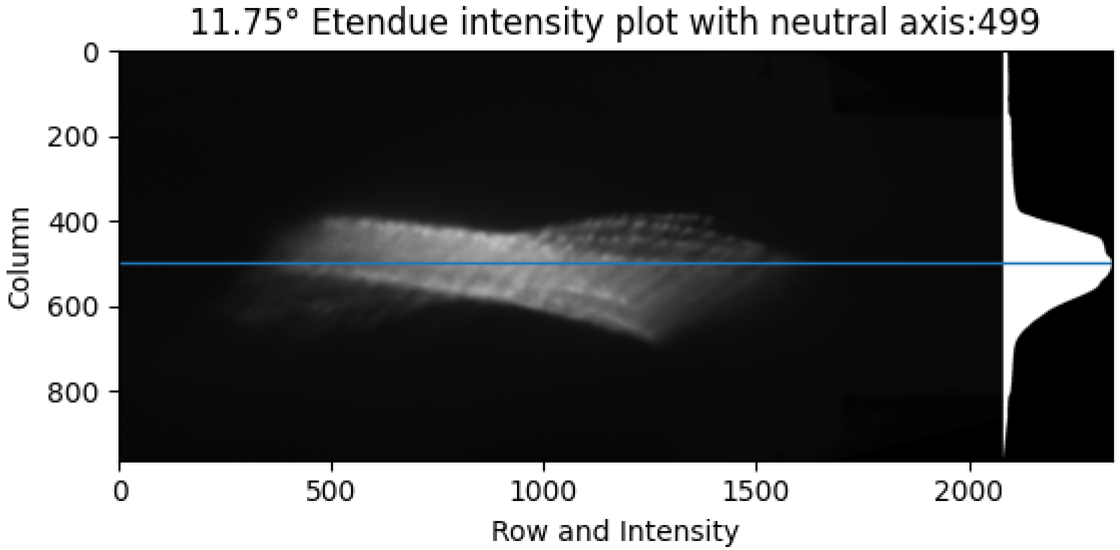
Figure



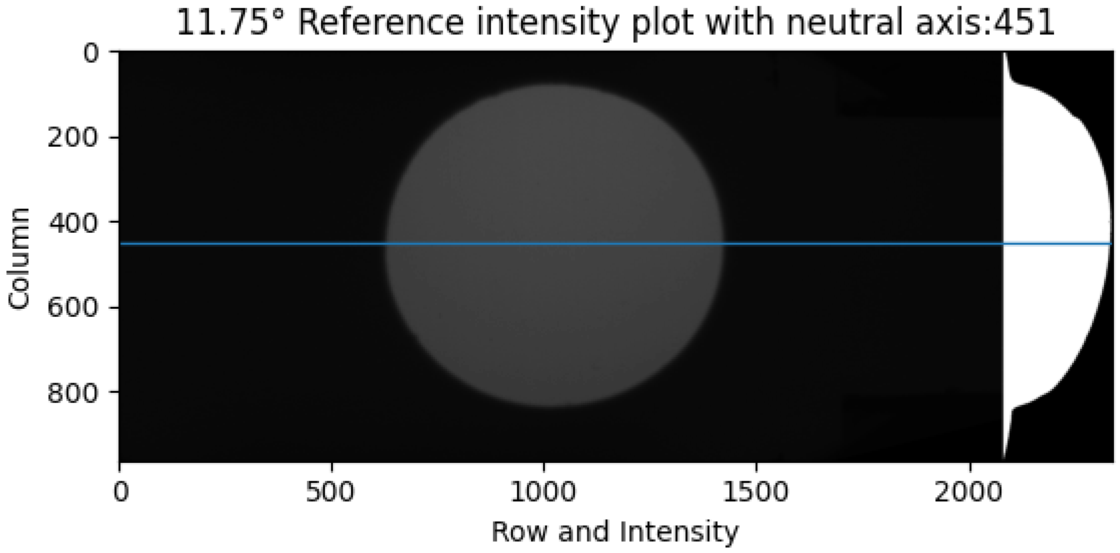


Figure

Figure

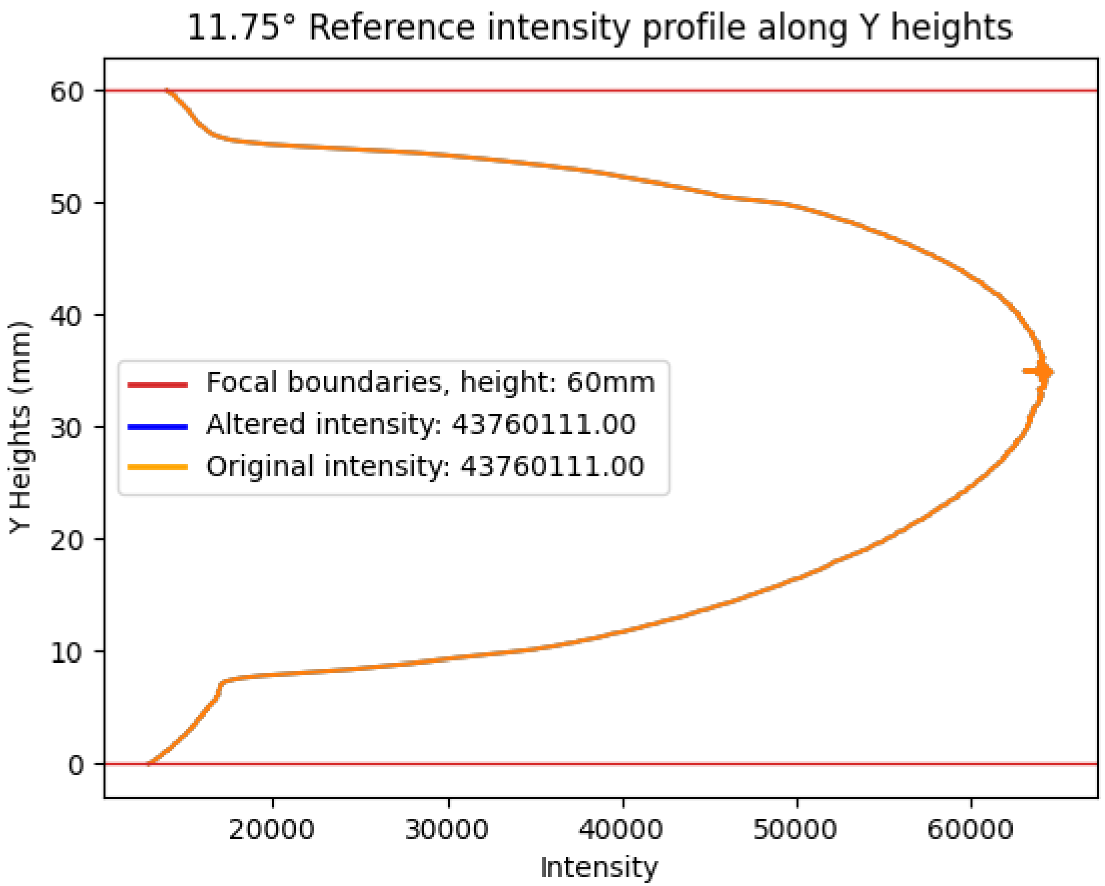

11.75° Etendue intensity plot with neutral axis:499
0
200
400
600
800
Column
0
500
1000
1500
2000
Row and Intensity

Figure



```
An error occurred trying to create min_hline, in 'show focal energy'-function: local variable 'index_to_keep_min' referenced before assignm
ent
 the hline was set to the first index of the list
An error occurred trying to create max_hline, in 'show focal energy'-function: local variable 'index_to_keep_max' referenced before assignm
ent
 the hline was set to the last index of the list
```

Figure



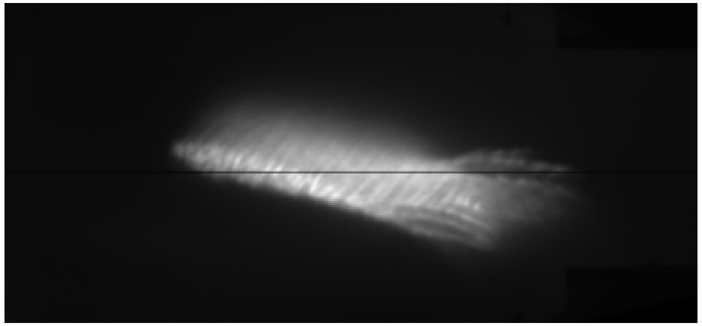
```
referece flux = 43760111
 etendue flux = 21417433
  transmission = 48%
```

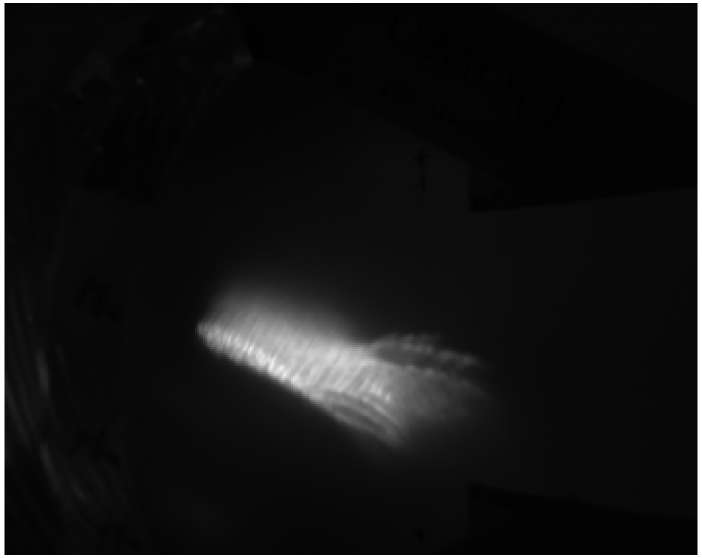
In [315…

```
plt.close()
clicked_points = [(234, 406), (789, 303), (775, 994), (216, 836)]
output_comparison("ImageAnalysis/7vii/23.5.tif","ImageAnalysis/7vii/23.5ref.tif","ImageAnalysis/7vii/plots/23.5deg",0,0)
```

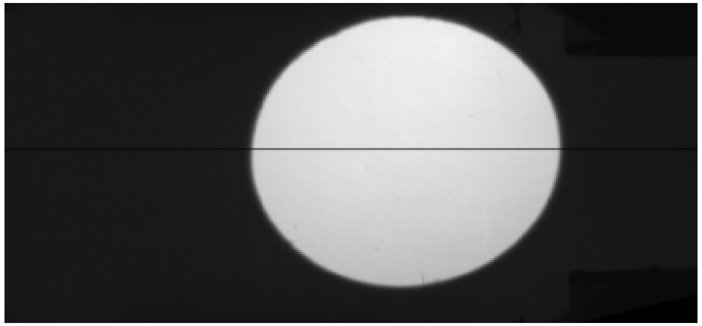
```
Clicked coordinates: [(234, 406), (789, 303), (775, 994), (216, 836)], Testing: False
Clicked coordinates: [(234, 406), (789, 303), (775, 994), (216, 836)], Testing: False
```

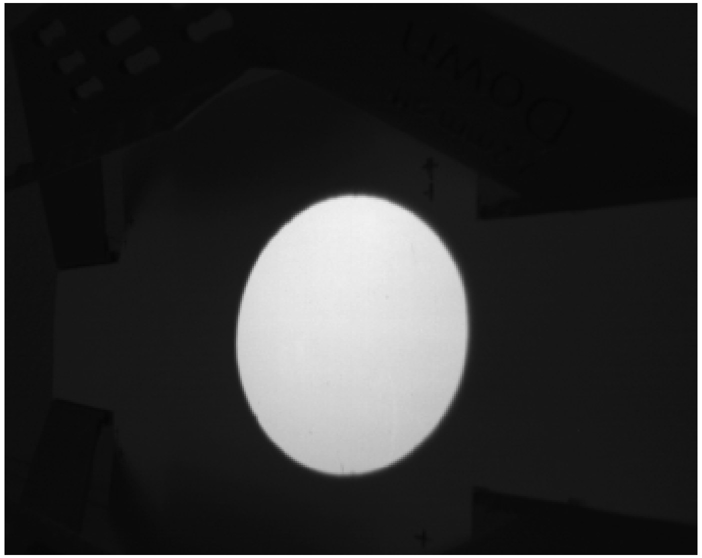
Figure



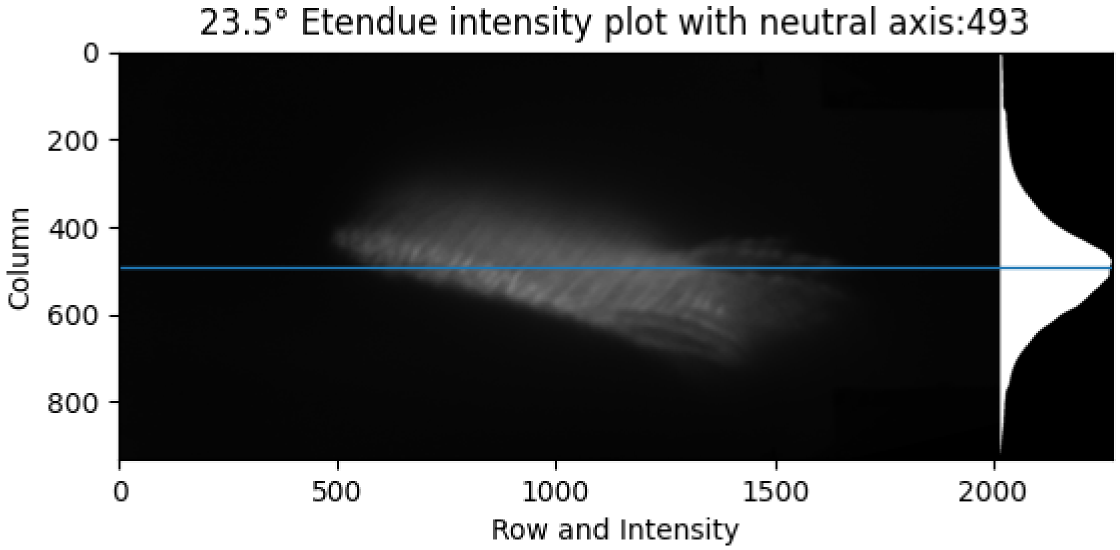


Figure

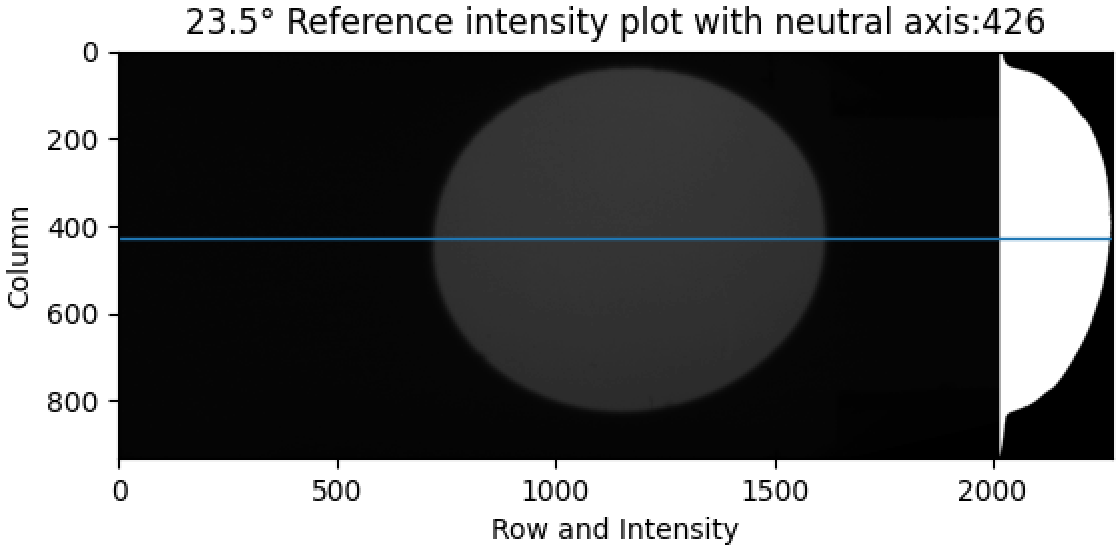




Figure

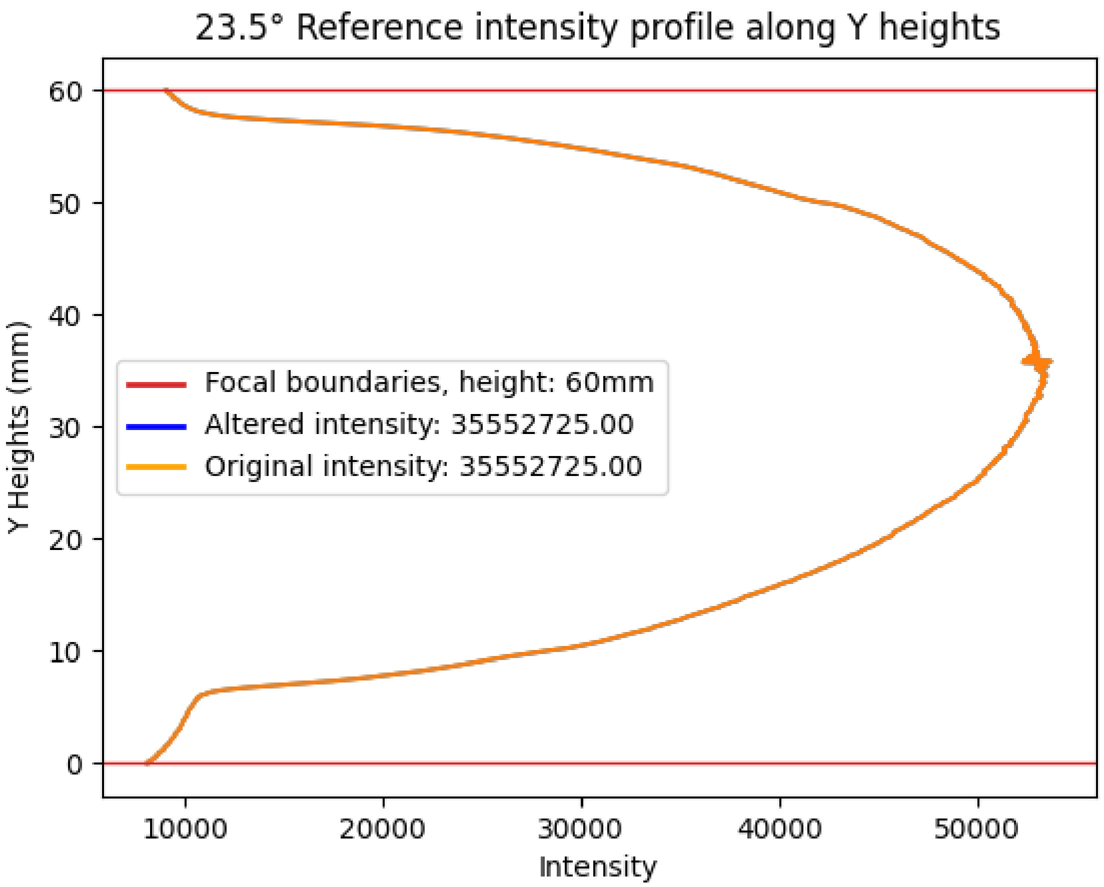

23.5° Etendue intensity plot with neutral axis:493
Column
0
200
400
600
800
0
500
1000
1500
2000
Row and Intensity

Figure



```
An error occurred trying to create min_hline, in 'show focal energy'-function: local variable 'index_to_keep_min' referenced before assignm
ent
 the hline was set to the first index of the list
An error occurred trying to create max_hline, in 'show focal energy'-function: local variable 'index_to_keep_max' referenced before assignm
ent
 the hline was set to the last index of the list
```

Figure



```
referece flux = 35552725
 etendue flux = 13418202
  transmission = 37%
```

In [ ]:

```

```

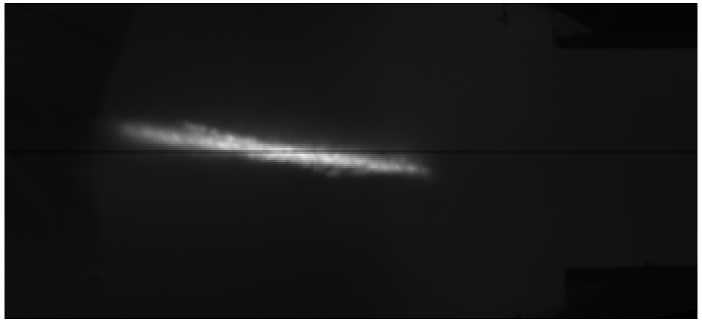
15i

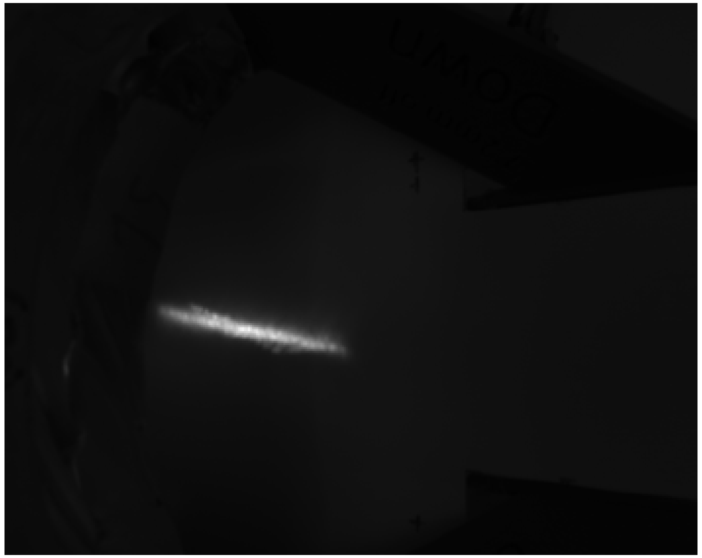
In [316…

```
clicked_points = [(201, 401), (777, 302), (766, 963), (177, 830)]
output_comparison("ImageAnalysis/15i/0deg.tif","ImageAnalysis/15i/0degref.tif","ImageAnalysis/15i/plots/0deg",17,15)
```

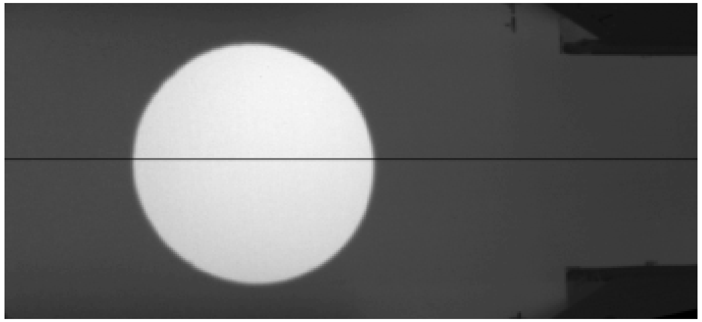
```
Clicked coordinates: [(201, 401), (777, 302), (766, 963), (177, 830)], Testing: False
Clicked coordinates: [(201, 401), (777, 302), (766, 963), (177, 830)], Testing: False
```

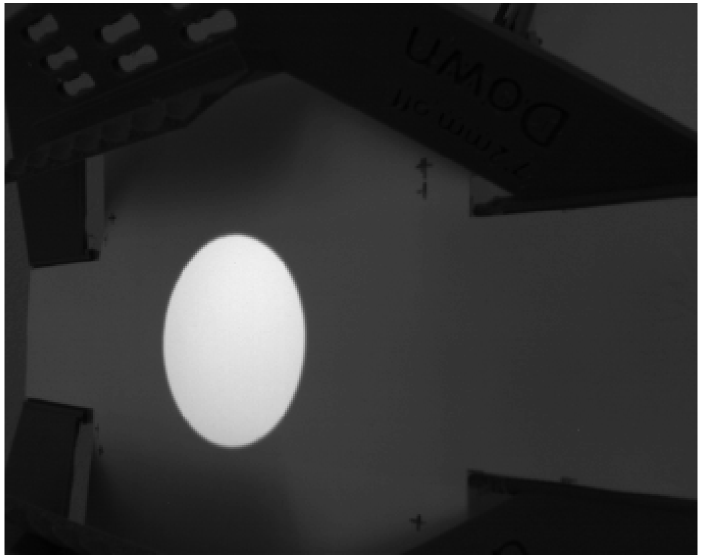
Figure



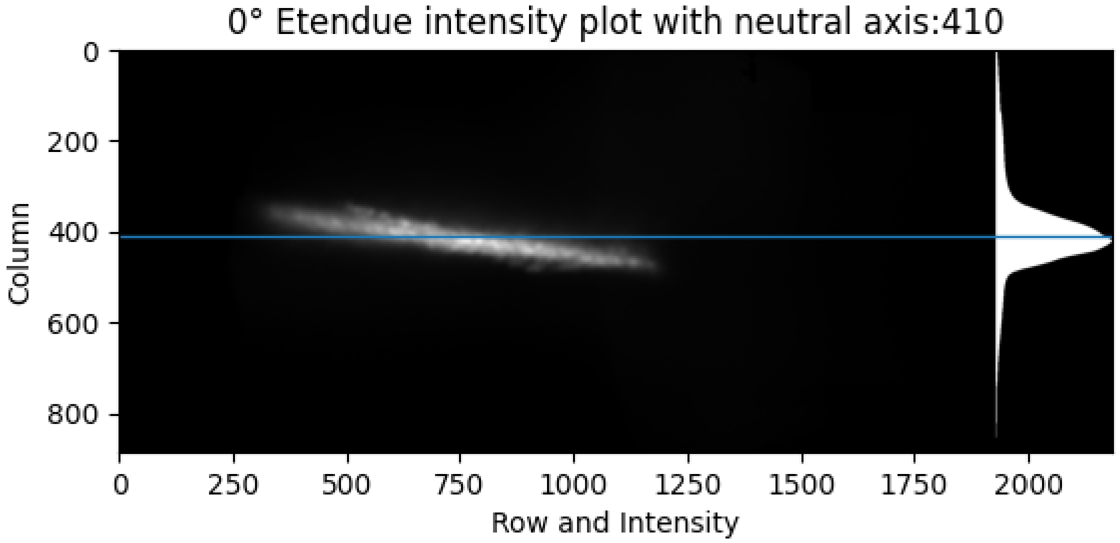


Figure

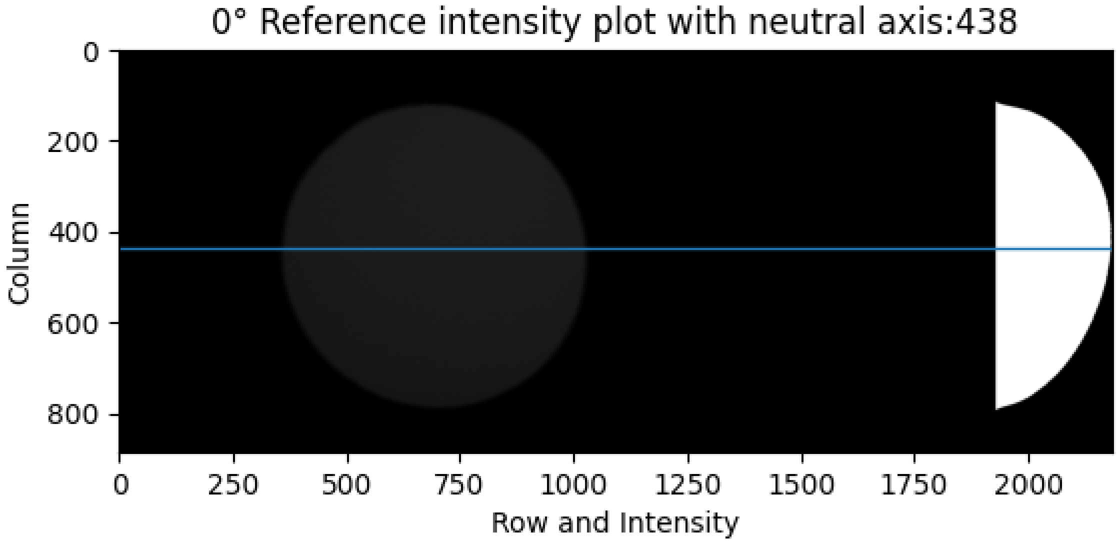




Figure

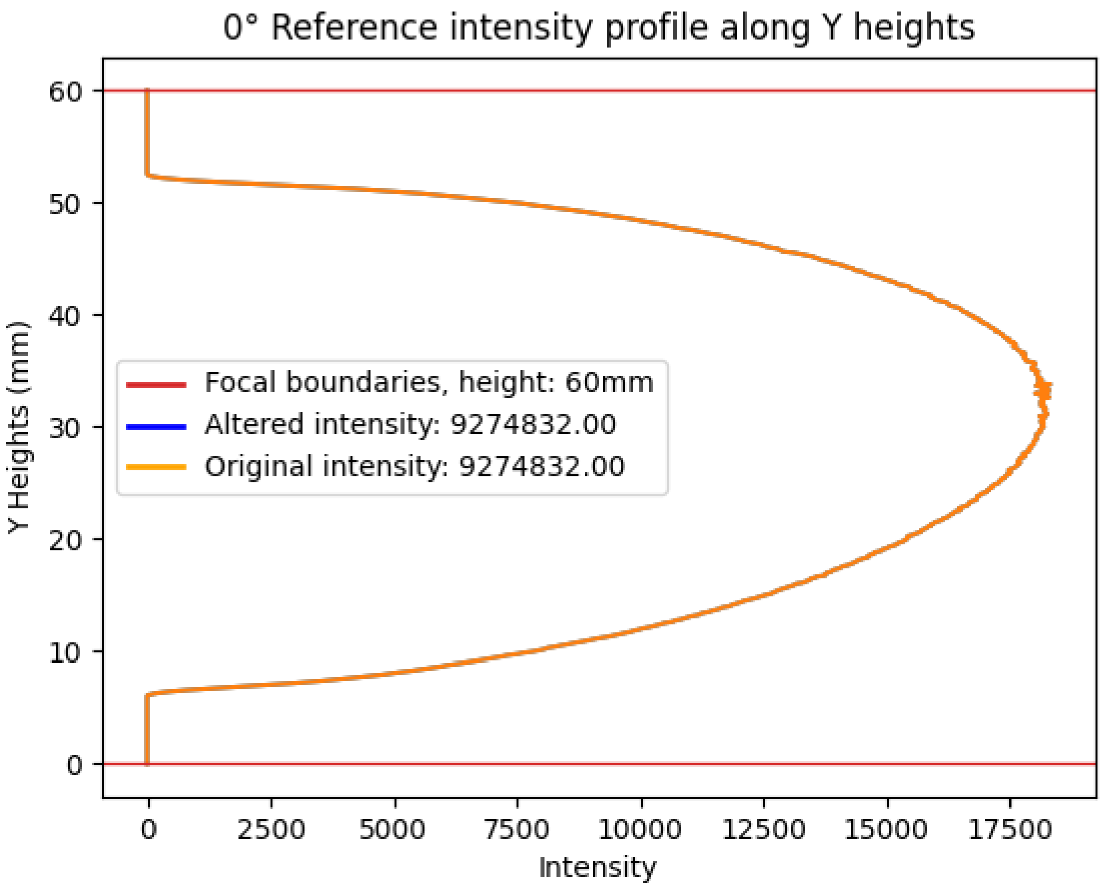

0° Etendue intensity plot with neutral axis:410
Column
0
200
400
600
800
0
250
500
750
1000
1250
1500
1750
2000
Row and Intensity

Figure



```
An error occurred trying to create min_hline, in 'show focal energy'-function: local variable 'index_to_keep_min' referenced before assignm
ent
 the hline was set to the first index of the list
An error occurred trying to create max_hline, in 'show focal energy'-function: local variable 'index_to_keep_max' referenced before assignm
ent
 the hline was set to the last index of the list
```

Figure



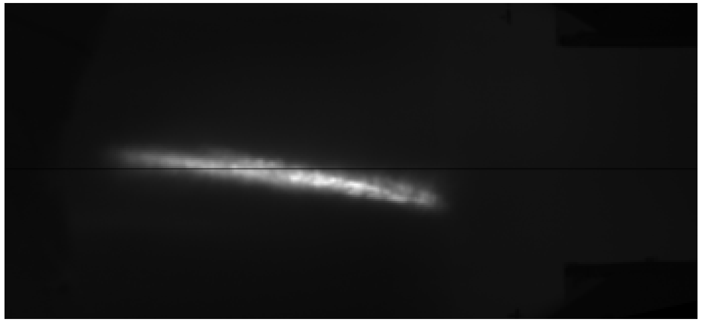
```
referece flux = 9274832
 etendue flux = 7687888
  transmission = 82%
```

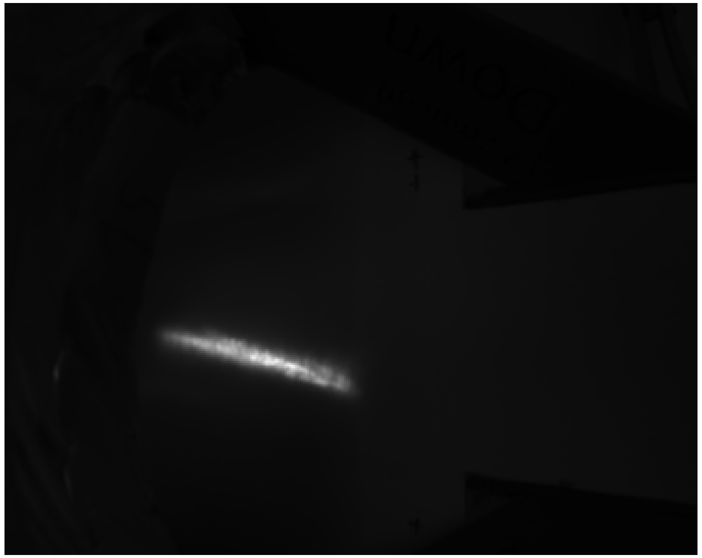
In [317…

```
clicked_points = [(209, 406), (772, 301), (768, 978), (193, 837)]
output_comparison("ImageAnalysis/15i/11.75.tif","ImageAnalysis/15i/11.75ref.tif","ImageAnalysis/15i/plots/11.75deg",17,18)
```

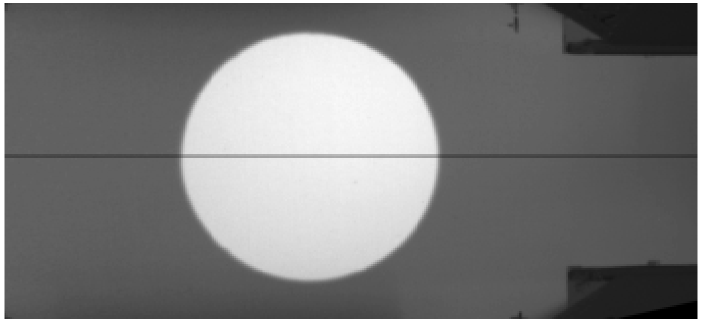
```
Clicked coordinates: [(209, 406), (772, 301), (768, 978), (193, 837)], Testing: False
Clicked coordinates: [(209, 406), (772, 301), (768, 978), (193, 837)], Testing: False
```

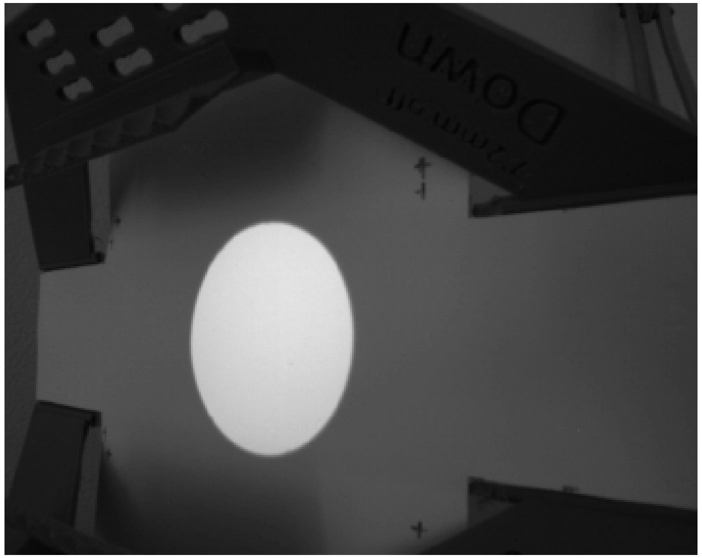
Figure



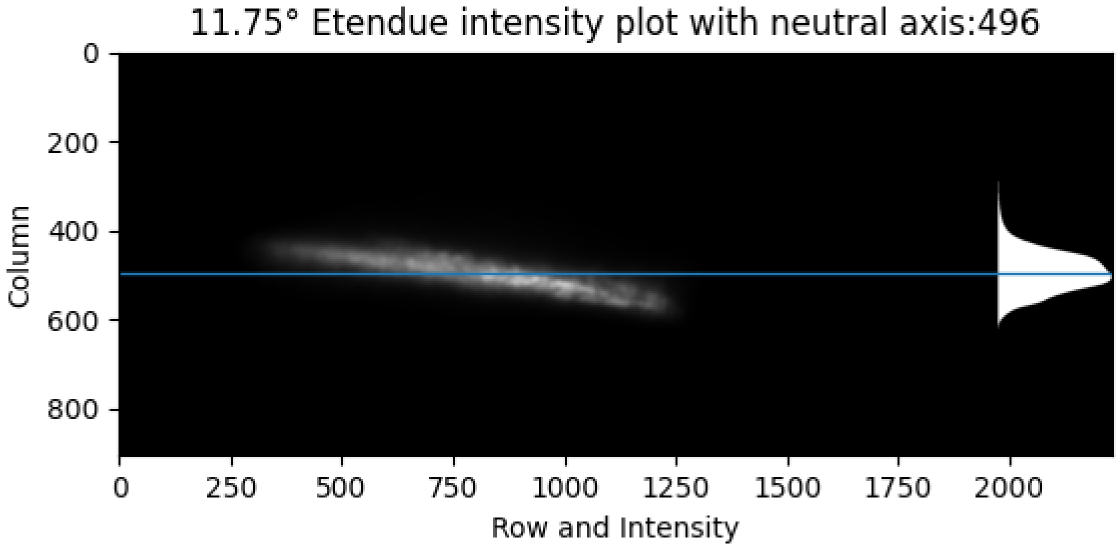


Figure

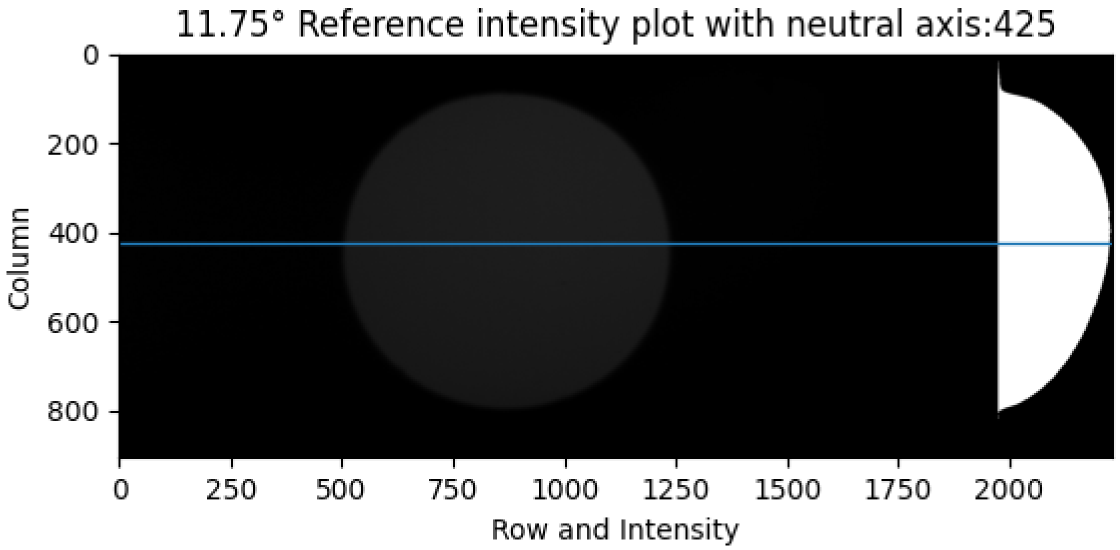




Figure

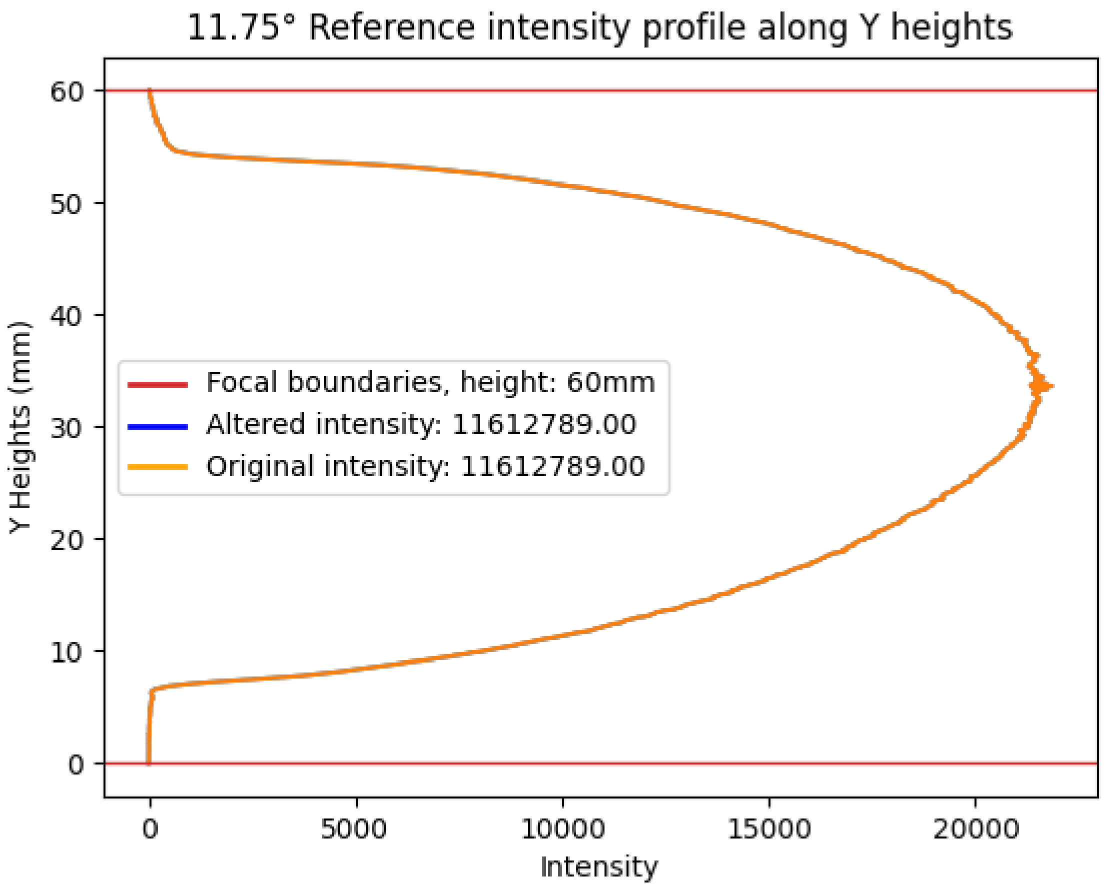

11.75° Etendue intensity plot with neutral axis:496
Column
0
200
400
600
800
0
250
500
750
1000
1250
1500
1750
2000
Row and Intensity

Figure



```
An error occurred trying to create min_hline, in 'show focal energy'-function: local variable 'index_to_keep_min' referenced before assignm
ent
 the hline was set to the first index of the list
An error occurred trying to create max_hline, in 'show focal energy'-function: local variable 'index_to_keep_max' referenced before assignm
ent
 the hline was set to the last index of the list
```

Figure



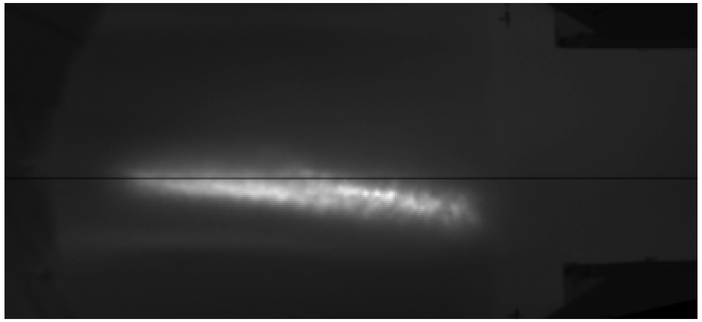
```
referece flux = 11612789
 etendue flux = 6429656
  transmission = 55%
```

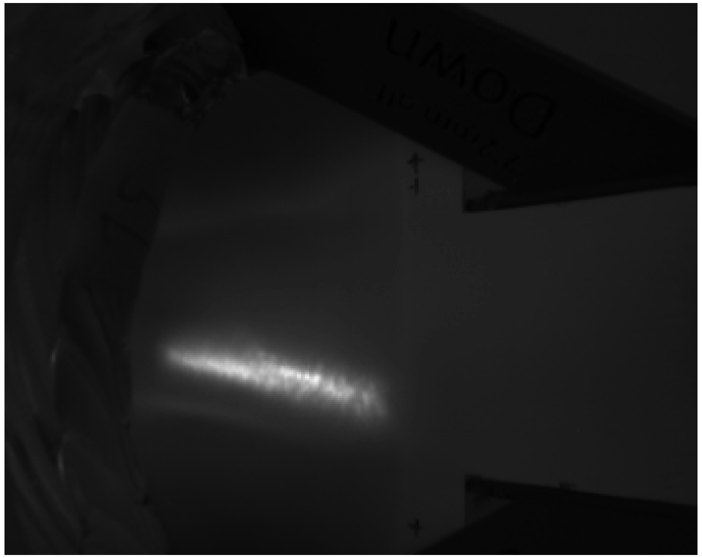
In [318…

```
clicked_points = [(211, 407), (774, 303), (769, 982), (194, 839)]
output_comparison("ImageAnalysis/15i/23.5.tif","ImageAnalysis/15i/23.5ref.tif","ImageAnalysis/15i/plots/23.5deg",16,16)
```

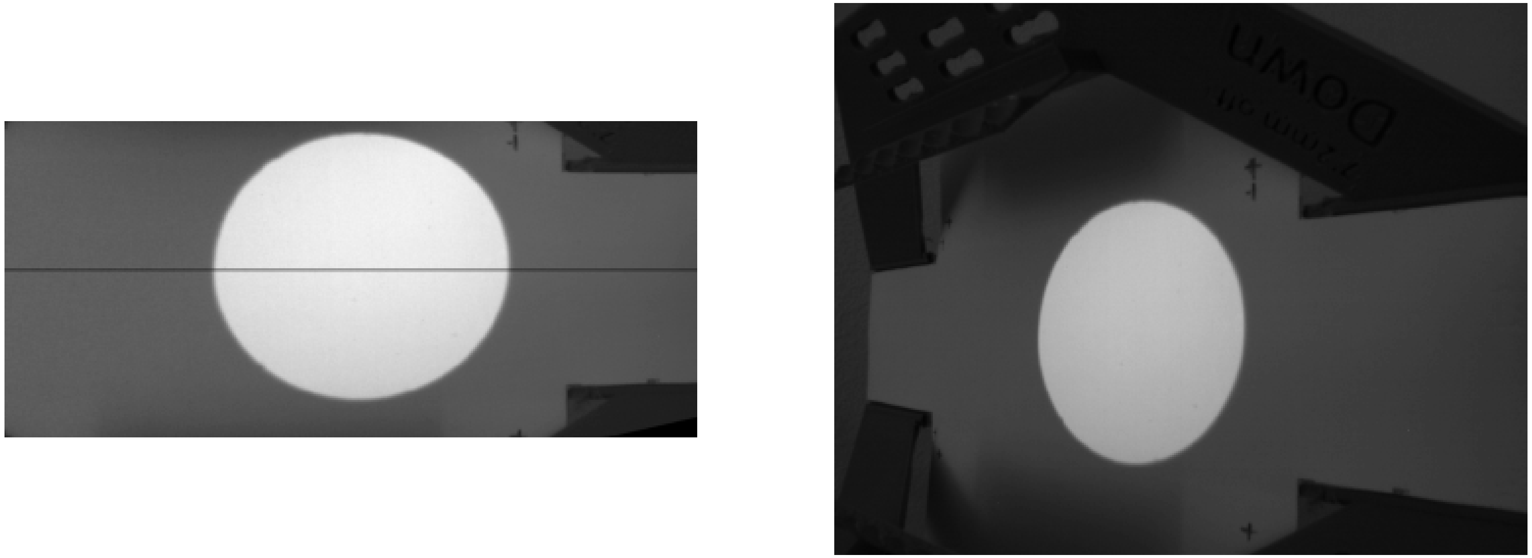
```
Clicked coordinates: [(211, 407), (774, 303), (769, 982), (194, 839)], Testing: False
Clicked coordinates: [(211, 407), (774, 303), (769, 982), (194, 839)], Testing: False
```

Figure

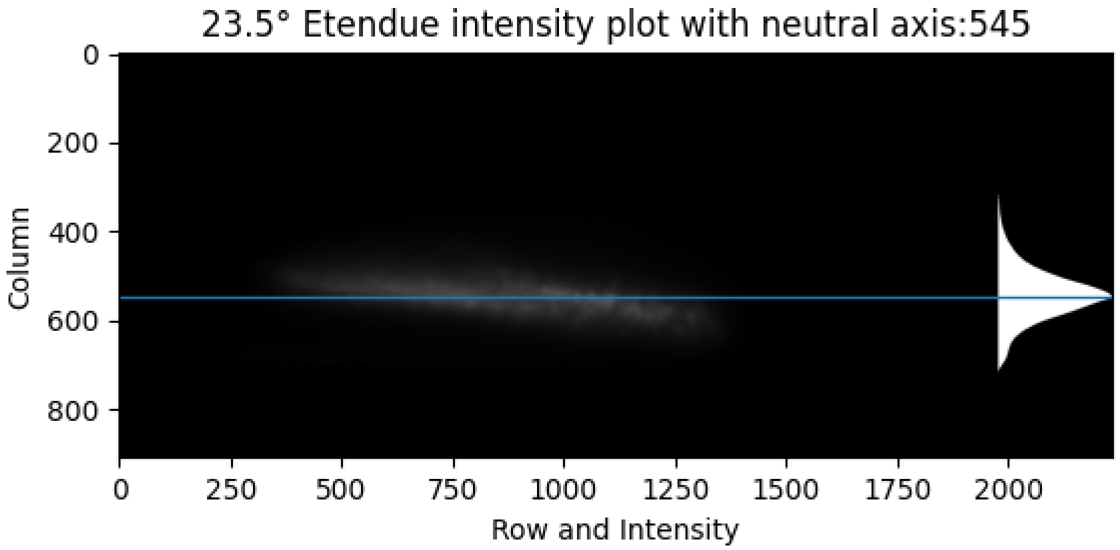




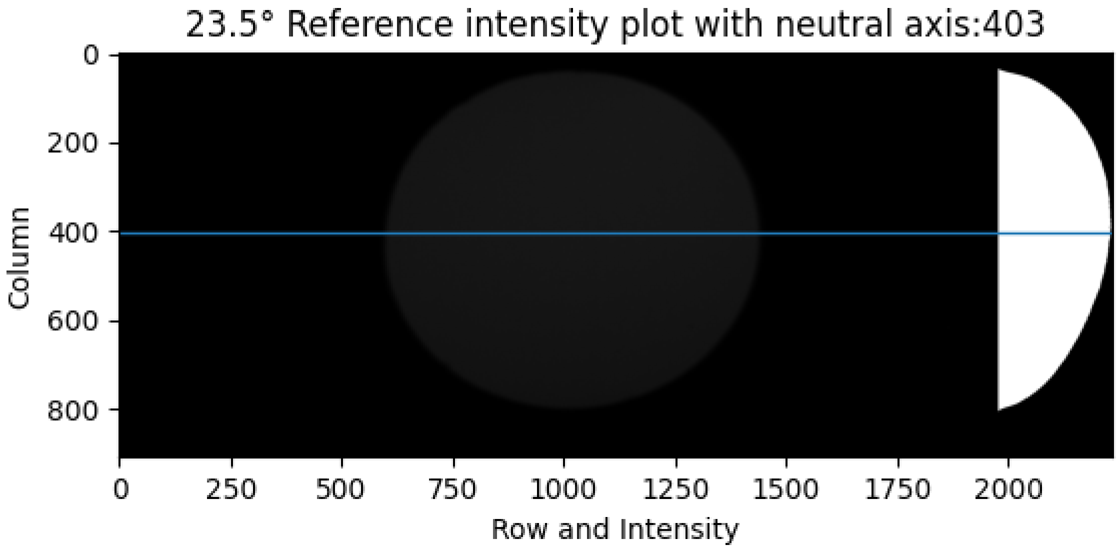
Figure



Figure

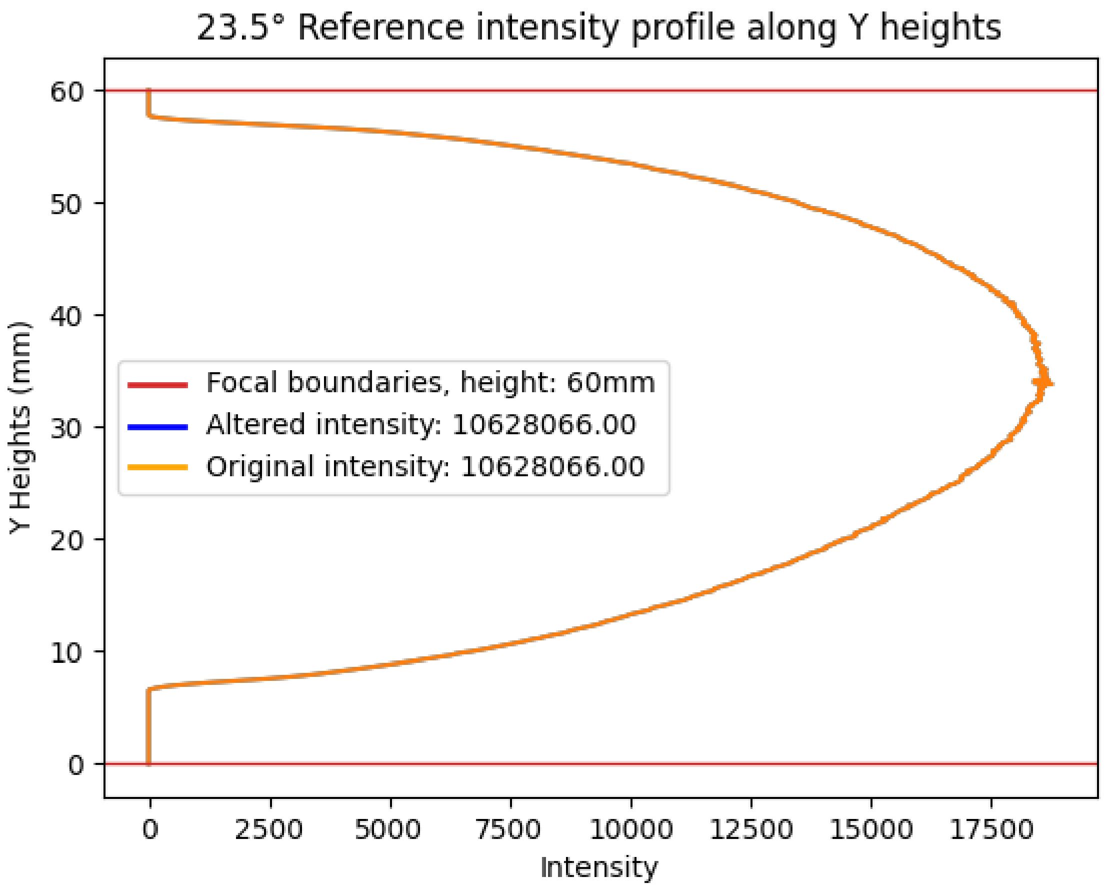

23.5° Etendue intensity plot with neutral axis:545
Column
0
200
400
600
800
Row and Intensity
0
250
500
750
1000
1250
1500
1750
2000

Figure



```
An error occurred trying to create min_hline, in 'show focal energy'-function: local variable 'index_to_keep_min' referenced before assignm
ent
 the hline was set to the first index of the list
An error occurred trying to create max_hline, in 'show focal energy'-function: local variable 'index_to_keep_max' referenced before assignm
ent
 the hline was set to the last index of the list
```

Figure



```
referece flux = 10628066
 etendue flux = 4764645
  transmission = 44%
```

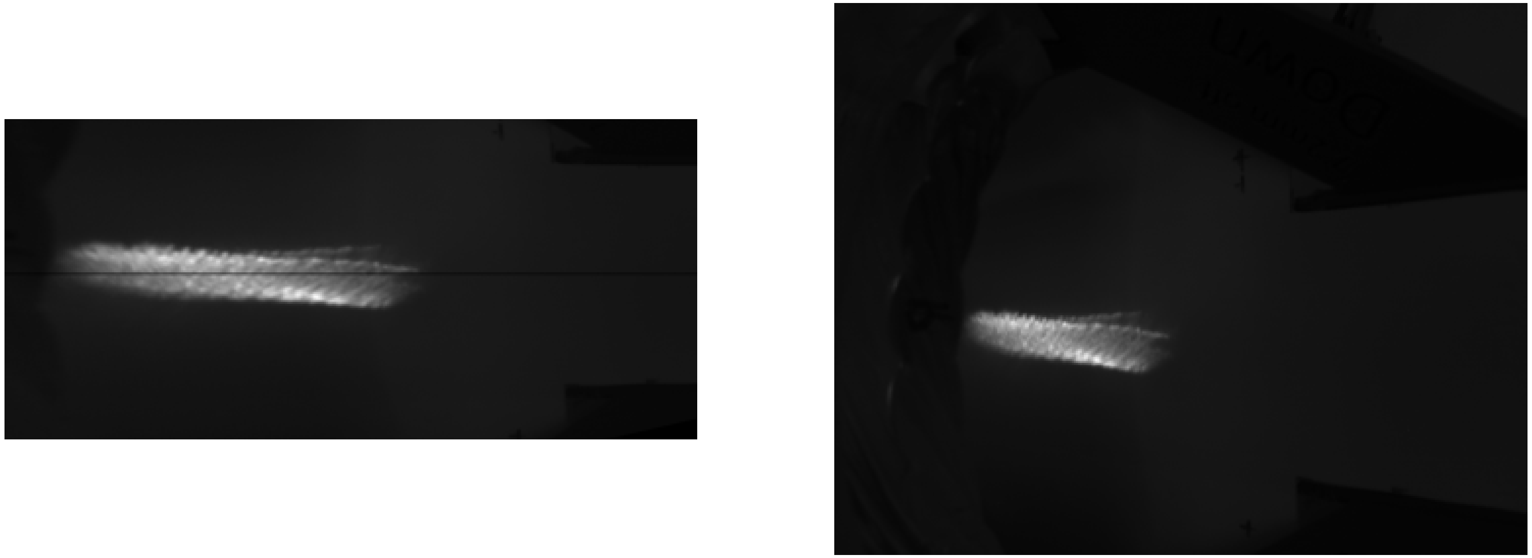
9i

In [320…

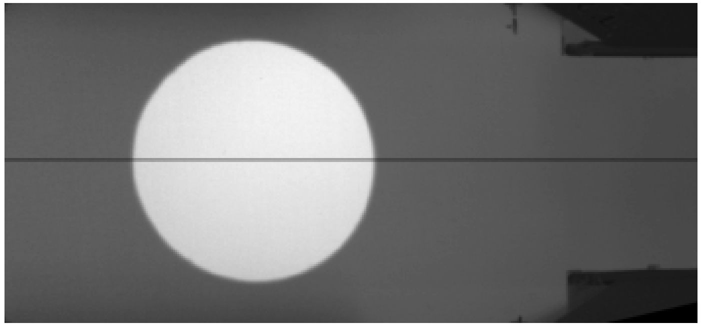
```
clicked_points = [(209, 408), (779, 308), (768, 982), (191, 838)]
output_comparison("ImageAnalysis/9i/0deg.tif","ImageAnalysis/9i/0degref.tif","ImageAnalysis/9i/plots/0deg",0,0)
```

```
Clicked coordinates: [(209, 408), (779, 308), (768, 982), (191, 838)], Testing: False
Clicked coordinates: [(209, 408), (779, 308), (768, 982), (191, 838)], Testing: False
```

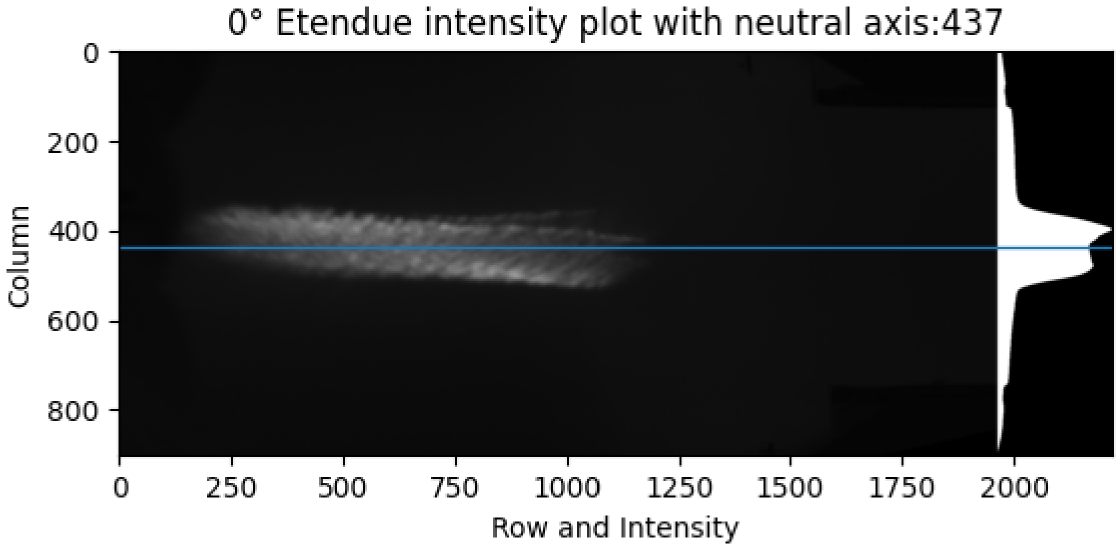
Figure



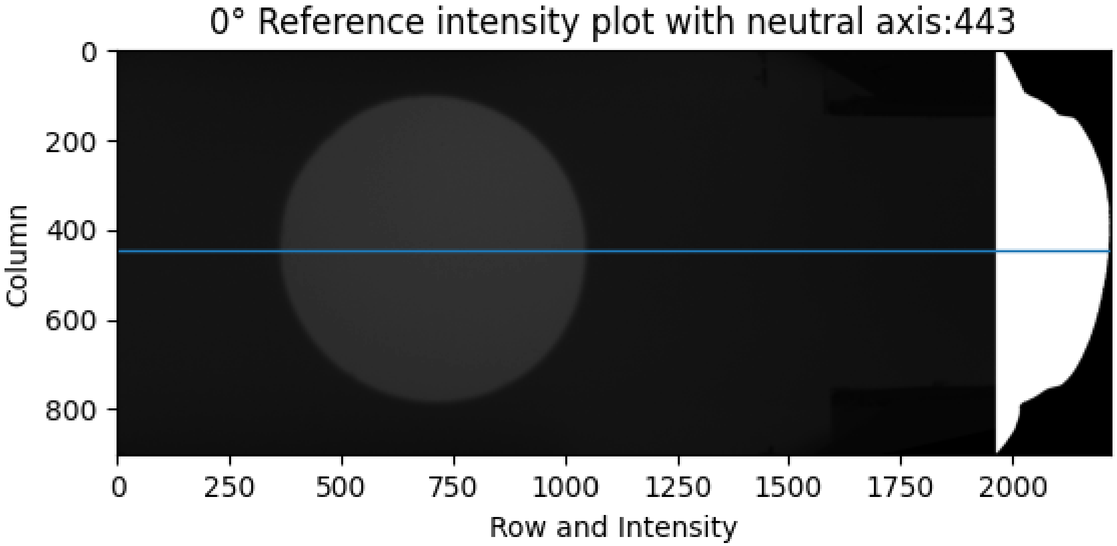
Figure



Figure

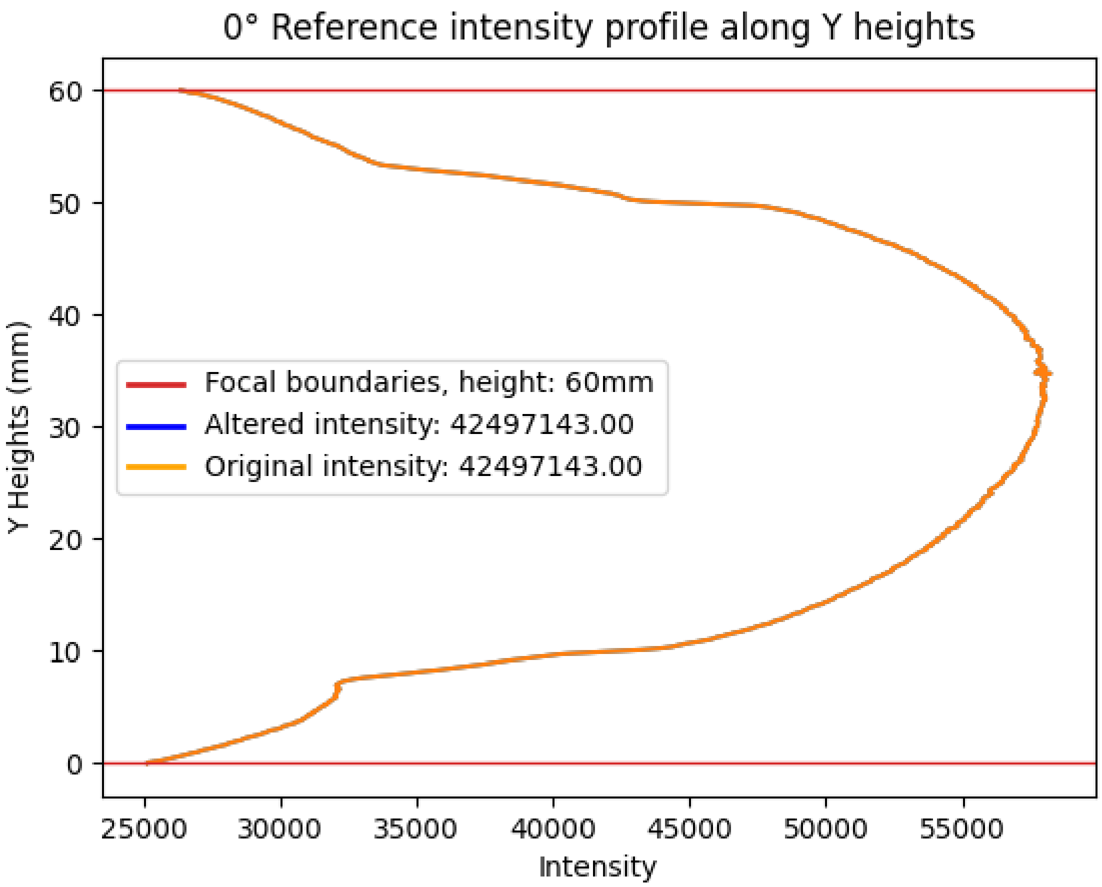

0° Etendue intensity plot with neutral axis:437
Column
0
200
400
600
800
0
250
500
750
1000
1250
1500
1750
2000
Row and Intensity

Figure



```
An error occurred trying to create min_hline, in 'show focal energy'-function: local variable 'index_to_keep_min' referenced before assignm
ent
 the hline was set to the first index of the list
An error occurred trying to create max_hline, in 'show focal energy'-function: local variable 'index_to_keep_max' referenced before assignm
ent
 the hline was set to the last index of the list
```

Figure



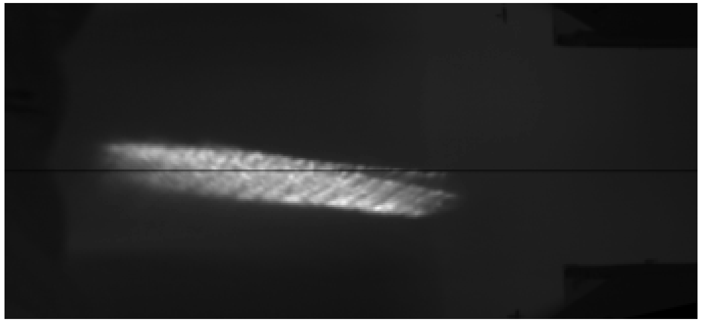
```
referece flux = 42497143
 etendue flux = 12980923
  transmission = 30%
```

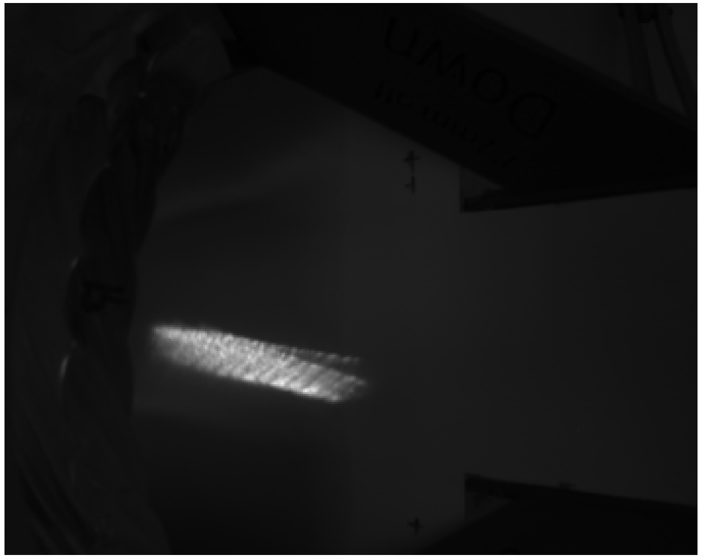
In [321…

```
clicked_points = [(203, 409), (771, 307), (767, 976), (186, 838)]
output_comparison("ImageAnalysis/9i/11.75.tif","ImageAnalysis/9i/11.75ref.tif","ImageAnalysis/9i/plots/11.75deg",0,0)
```

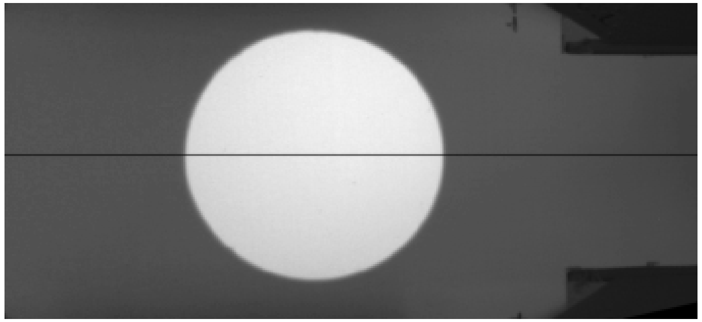
```
Clicked coordinates: [(203, 409), (771, 307), (767, 976), (186, 838)], Testing: False
Clicked coordinates: [(203, 409), (771, 307), (767, 976), (186, 838)], Testing: False
```

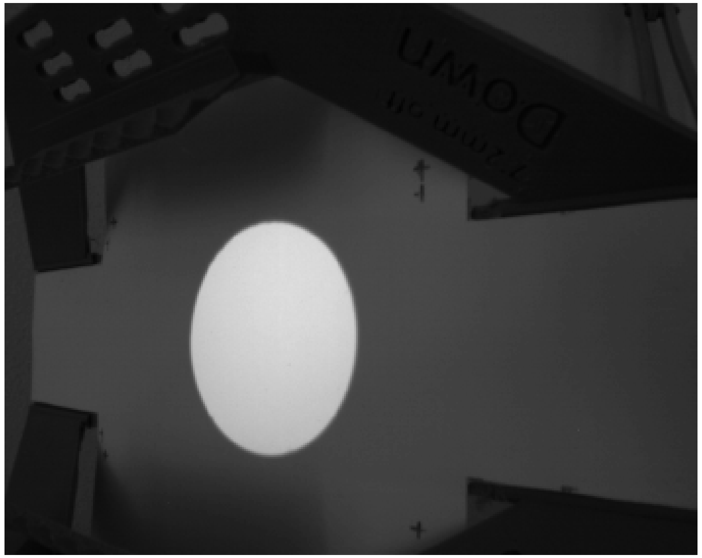
Figure



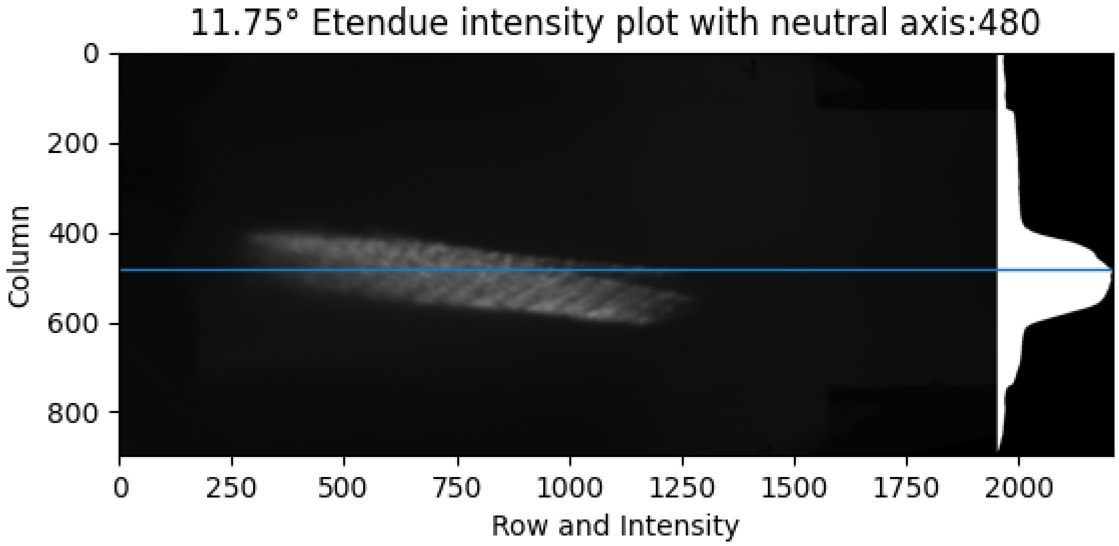


Figure

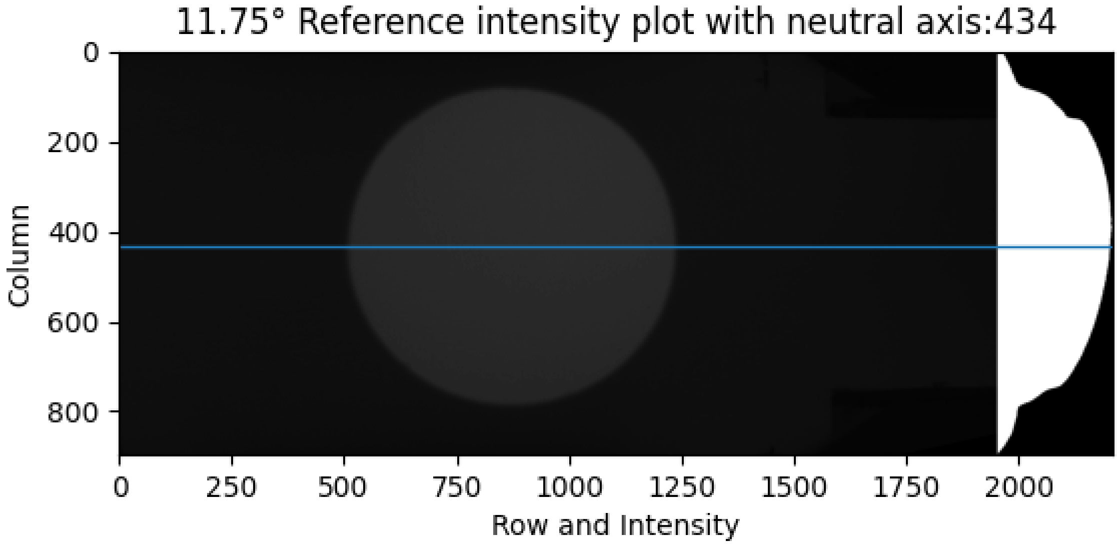




Figure

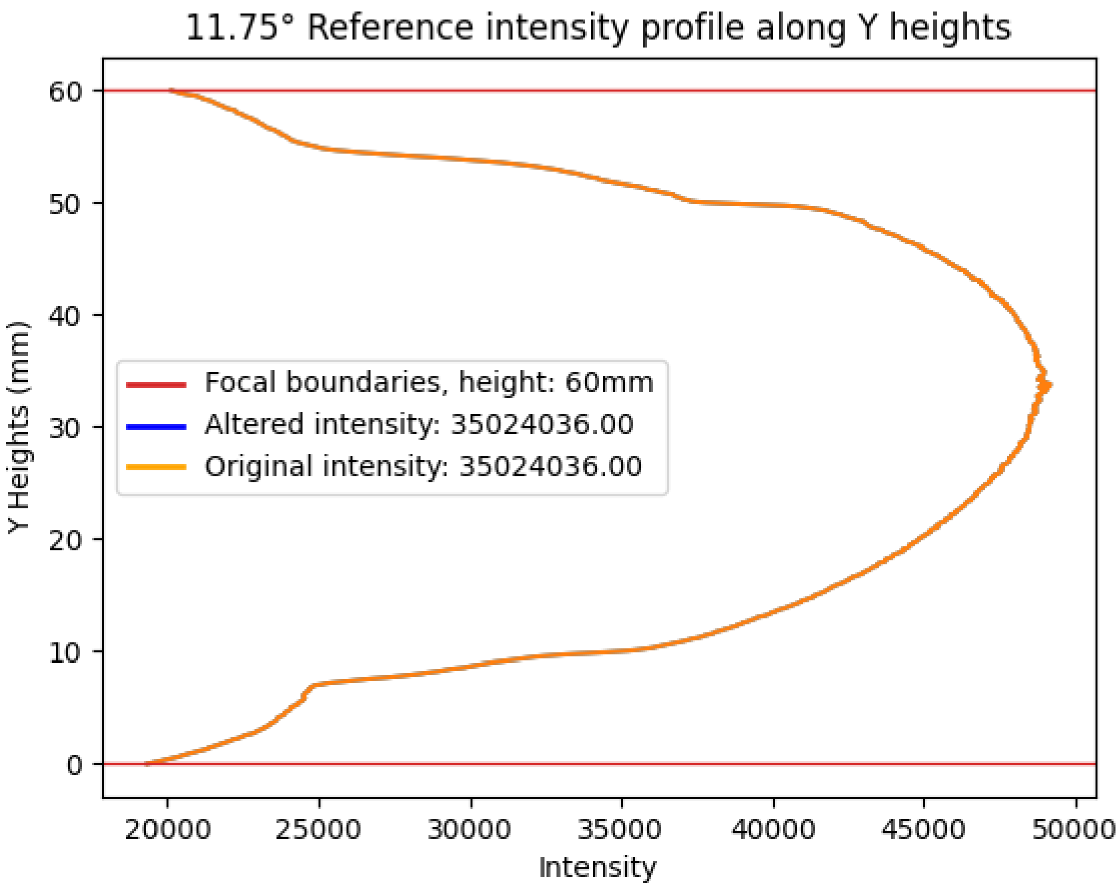

11.75° Etendue intensity plot with neutral axis:480
Column
0
200
400
600
800
0
250
500
750
1000
1250
1500
1750
2000
Row and Intensity

Figure



```
An error occurred trying to create min_hline, in 'show focal energy'-function: local variable 'index_to_keep_min' referenced before assignm
ent
 the hline was set to the first index of the list
An error occurred trying to create max_hline, in 'show focal energy'-function: local variable 'index_to_keep_max' referenced before assignm
ent
 the hline was set to the last index of the list
```

Figure



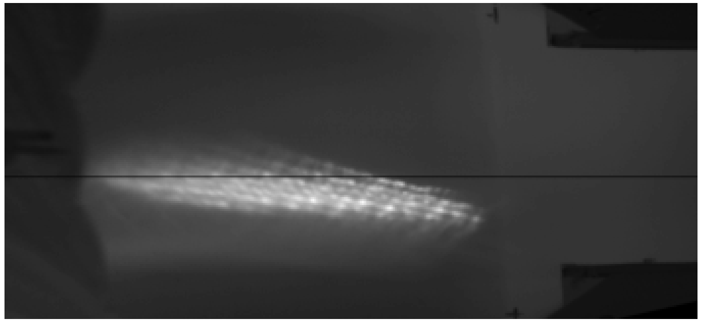
```
referece flux = 35024036
 etendue flux = 11626317
  transmission = 33%
```

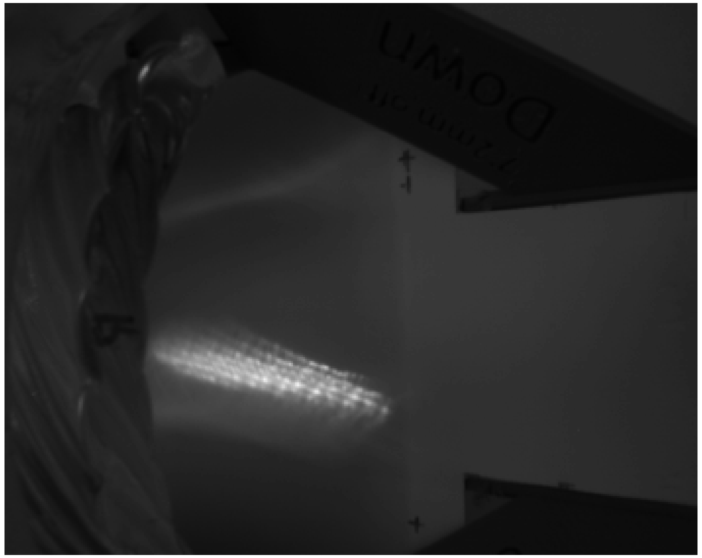
In [323…

```
clicked_points = [(211, 407), (776, 305), (768, 978), (192, 839)]
output_comparison("ImageAnalysis/9i/23.5.tif","ImageAnalysis/9i/23.5ref.tif","ImageAnalysis/9i/plots/23.5deg",0,0)
```

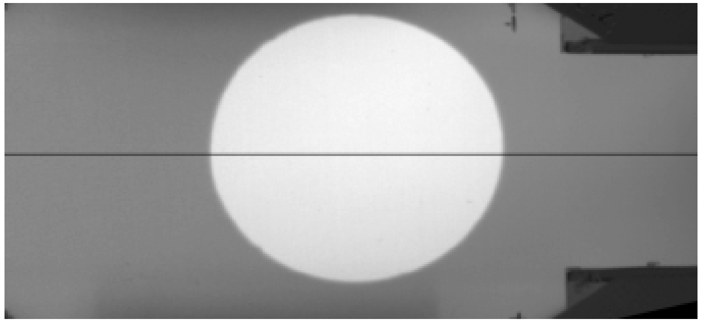
```
Clicked coordinates: [(211, 407), (776, 305), (768, 978), (192, 839)], Testing: False
Clicked coordinates: [(211, 407), (776, 305), (768, 978), (192, 839)], Testing: False
```

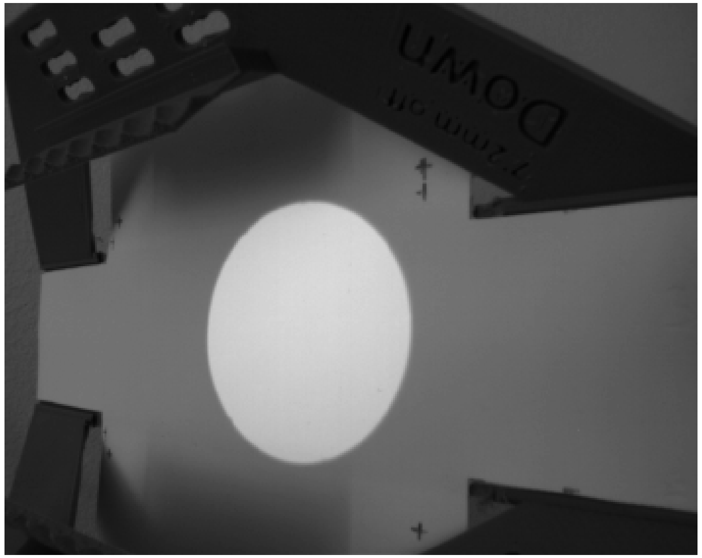
Figure



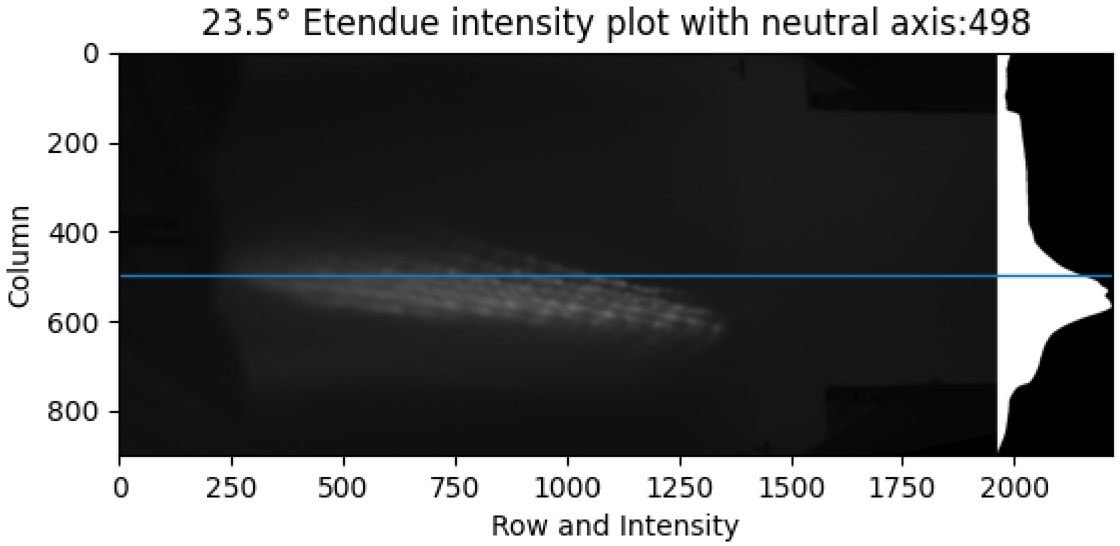


Figure

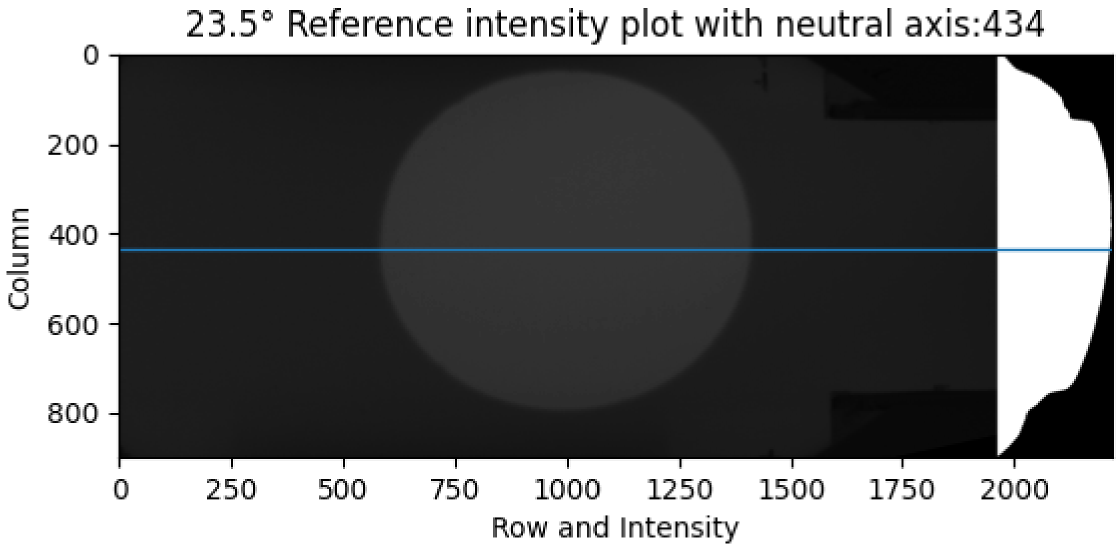




Figure

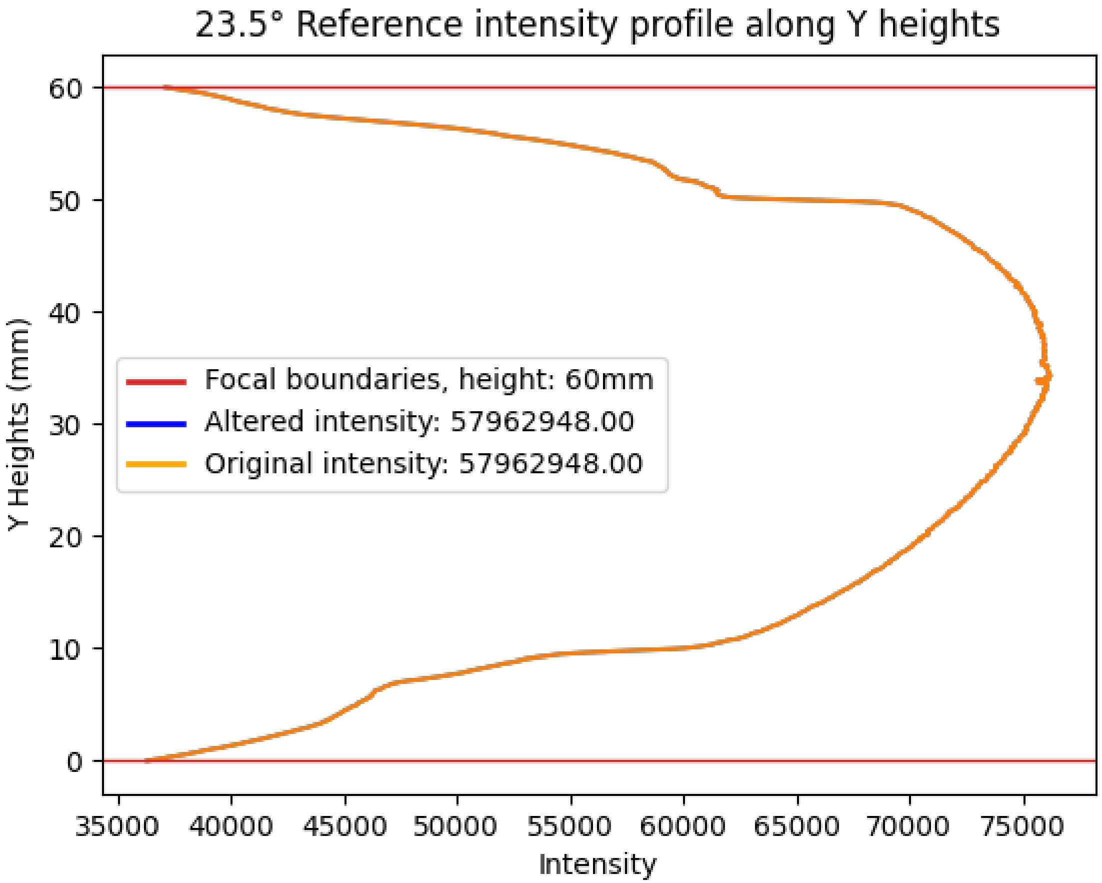

23.5° Etendue intensity plot with neutral axis:498
Column
0
200
400
600
800
0
250
500
750
1000
1250
1500
1750
2000
Row and Intensity

Figure



```
An error occurred trying to create min_hline, in 'show focal energy'-function: local variable 'index_to_keep_min' referenced before assignm
ent
 the hline was set to the first index of the list
An error occurred trying to create max_hline, in 'show focal energy'-function: local variable 'index_to_keep_max' referenced before assignm
ent
 the hline was set to the last index of the list
```

Figure



```
referece flux = 57962948
 etendue flux = 12339717
  transmission = 21%
```

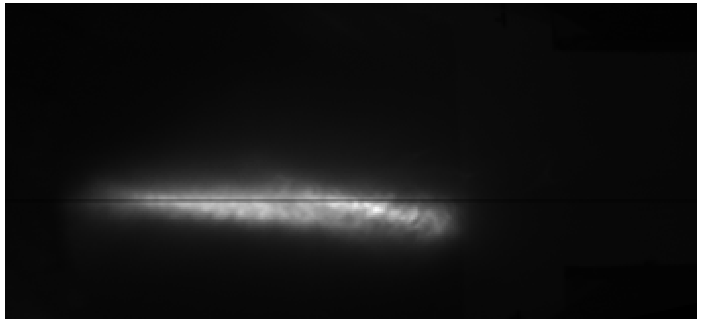
15i_2

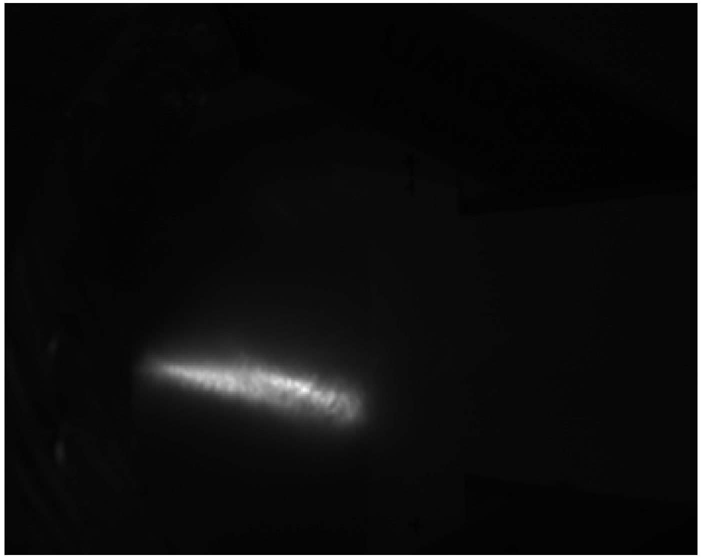
In [324…

```
clicked_points = [(203, 403), (770, 302), (765, 973), (184, 834)]
output_comparison("ImageAnalysis/15i/15i_2/23.5_2.tif","ImageAnalysis/15i/15i_2/23.5_2ref.tif","ImageAnalysis/15i/15i_2/plots/23.5deg",6,6,
```

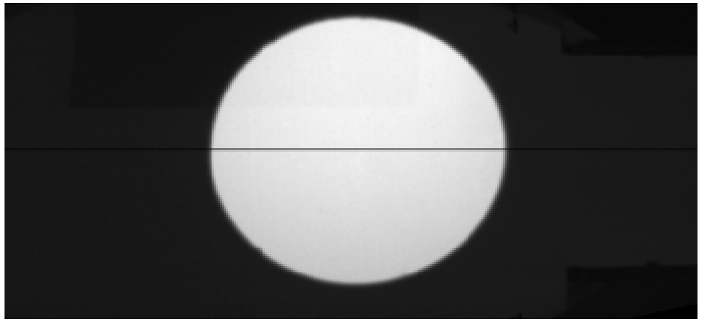
```
Clicked coordinates: [(203, 403), (770, 302), (765, 973), (184, 834)], Testing: False
Clicked coordinates: [(203, 403), (770, 302), (765, 973), (184, 834)], Testing: False
```

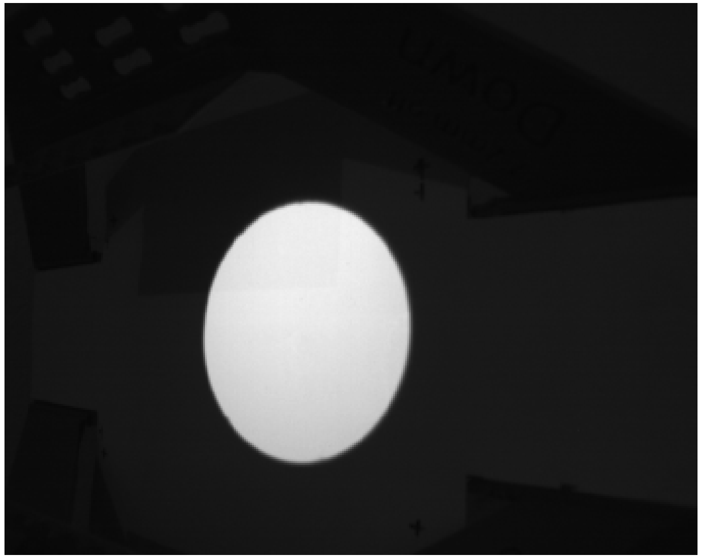
Figure



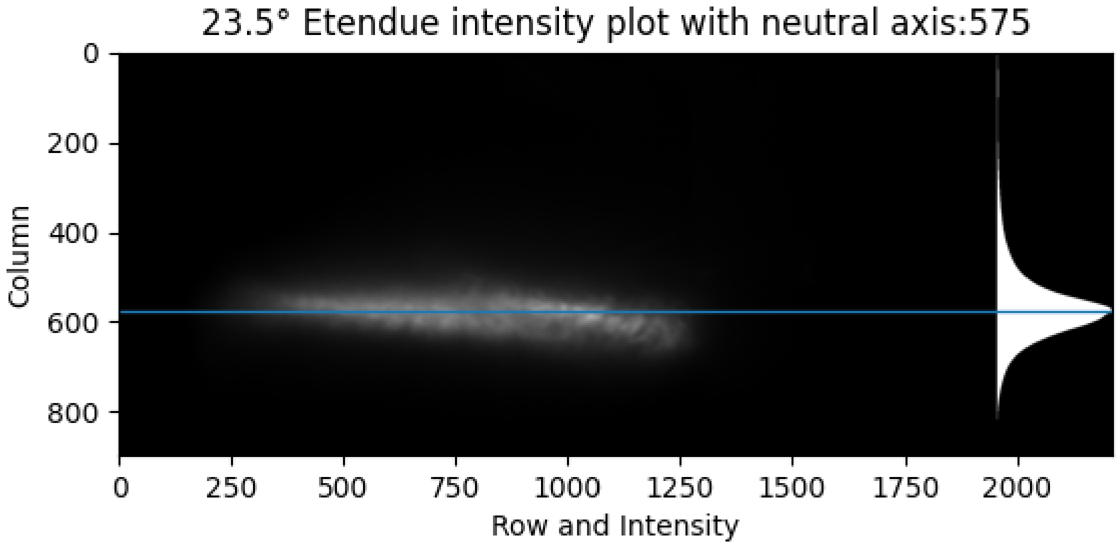


Figure

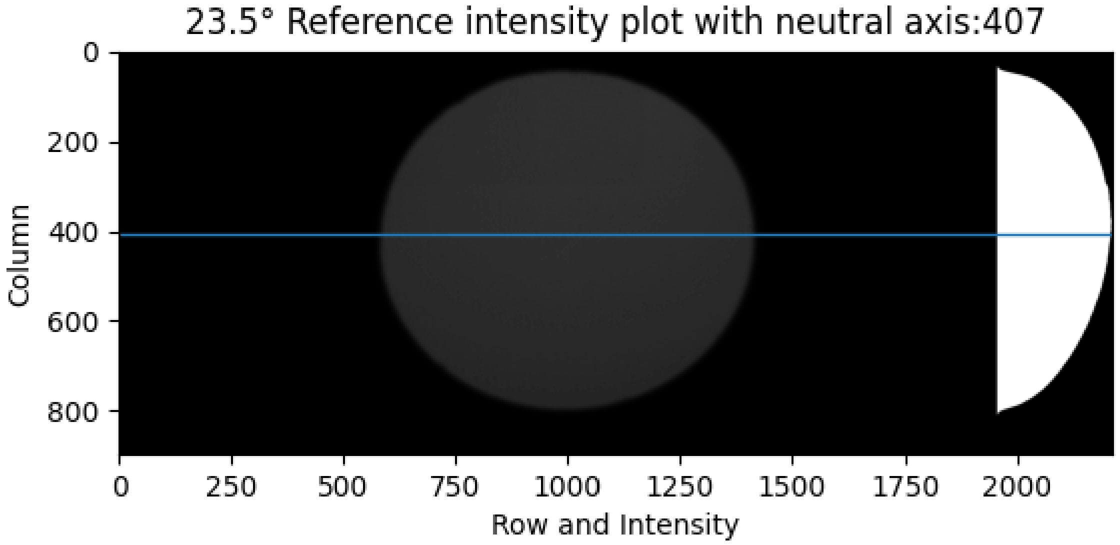




Figure

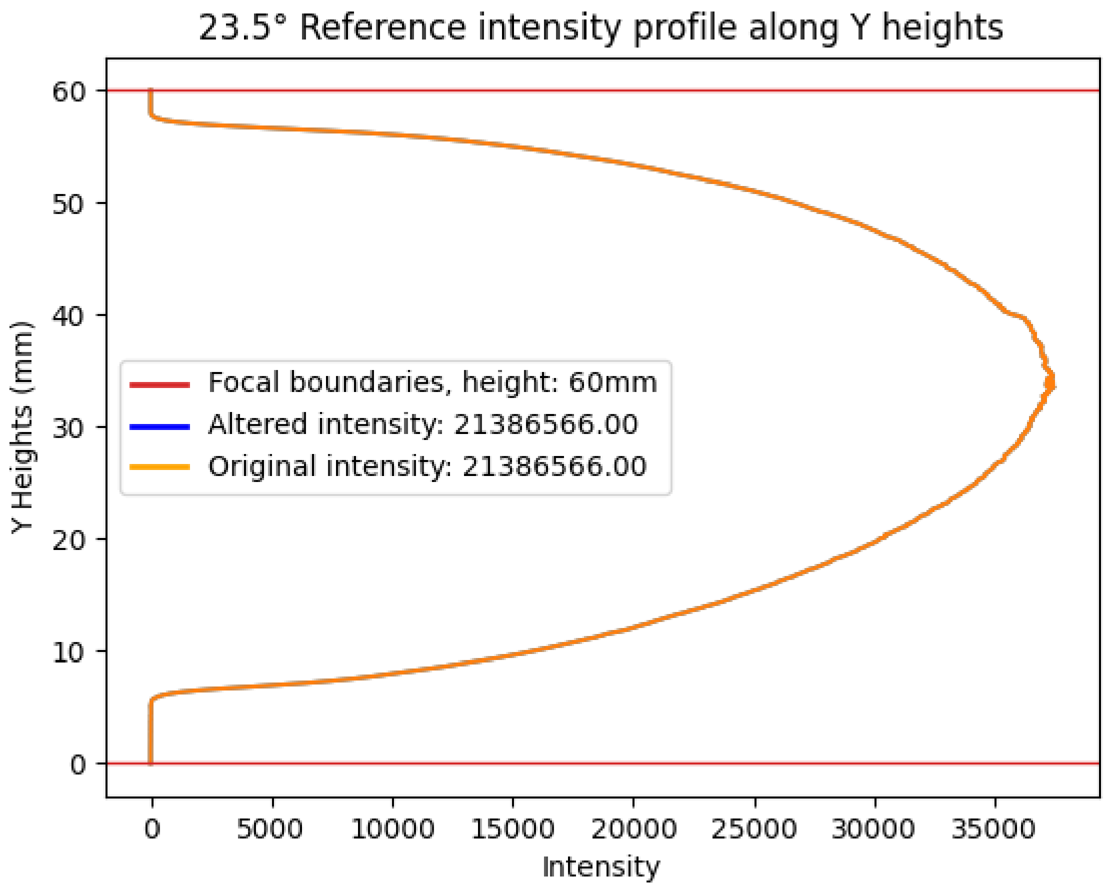

23.5° Etendue intensity plot with neutral axis:575
Column
0
200
400
600
800
0
250
500
750
1000
1250
1500
1750
2000
Row and Intensity

Figure



```
An error occurred trying to create min_hline, in 'show focal energy'-function: local variable 'index_to_keep_min' referenced before assignm
ent
 the hline was set to the first index of the list
An error occurred trying to create max_hline, in 'show focal energy'-function: local variable 'index_to_keep_max' referenced before assignm
ent
 the hline was set to the last index of the list
```

Figure



```
referece flux = 21386566
 etendue flux = 10589556
  transmission = 49%
```

In [137…

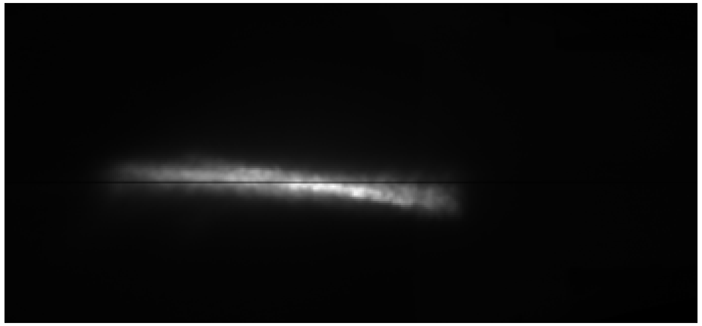
```
plt.close("all")
```

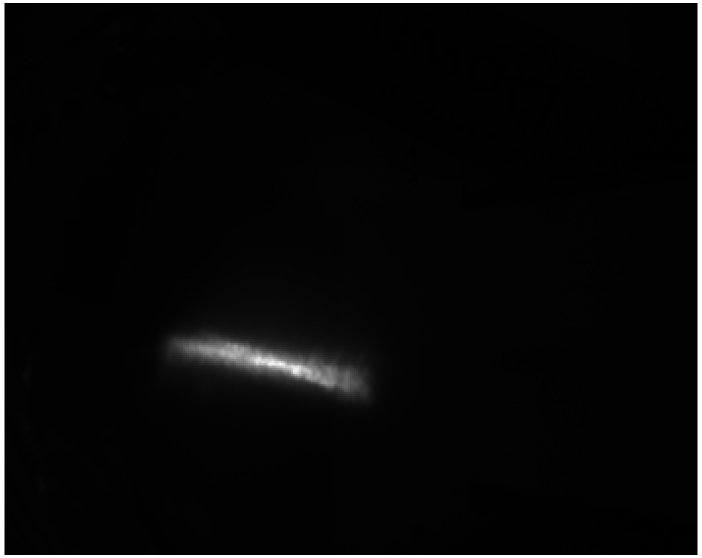
In [325…

```
clicked_points = [(228, 401), (777, 292), (771, 989), (215, 834)]
output_comparison("ImageAnalysis/15i/15i_2/11.75_2.tif","ImageAnalysis/15i/15i_2/11.75_2ref.tif","ImageAnalysis/15i/15i_2/plots/11.75deg",4
```

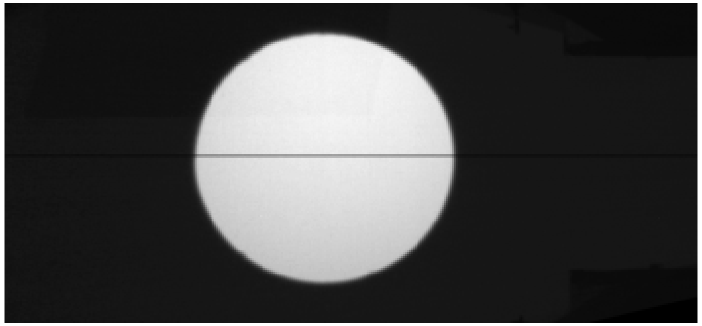
```
Clicked coordinates: [(228, 401), (777, 292), (771, 989), (215, 834)], Testing: False
Clicked coordinates: [(228, 401), (777, 292), (771, 989), (215, 834)], Testing: False
```

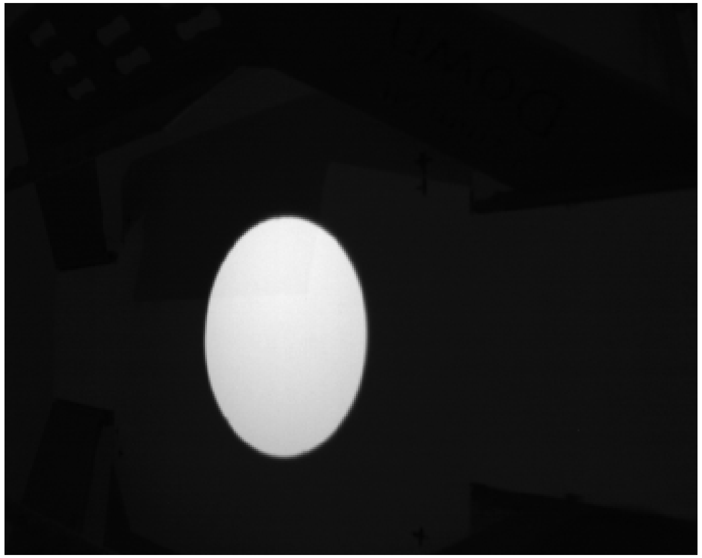
Figure



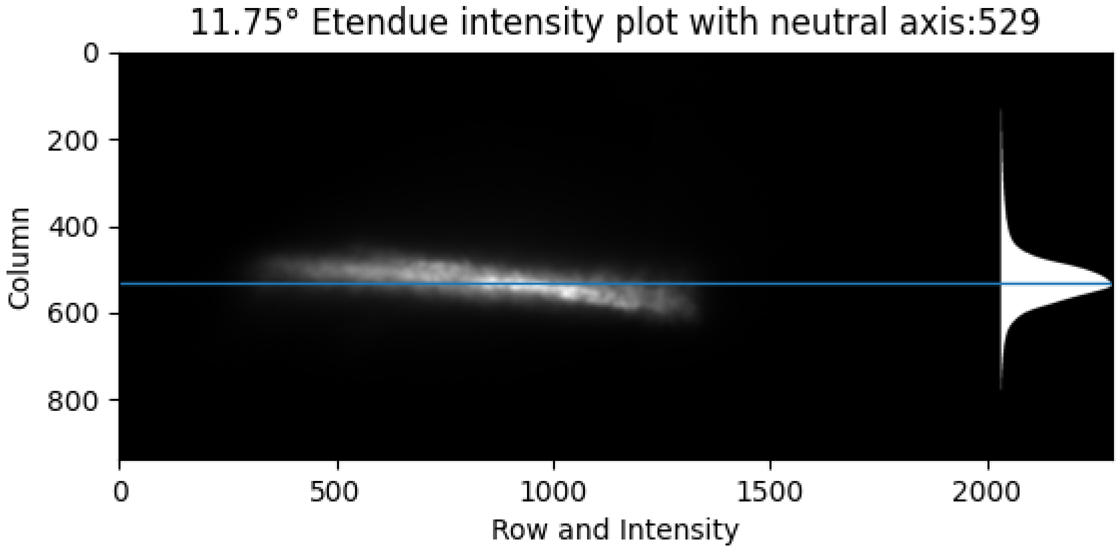


Figure

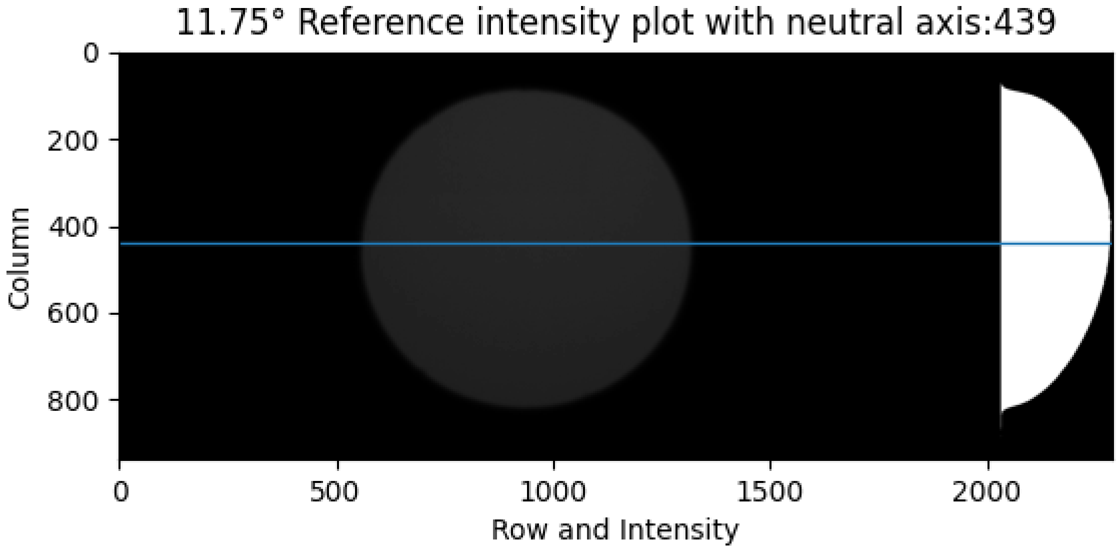




Figure

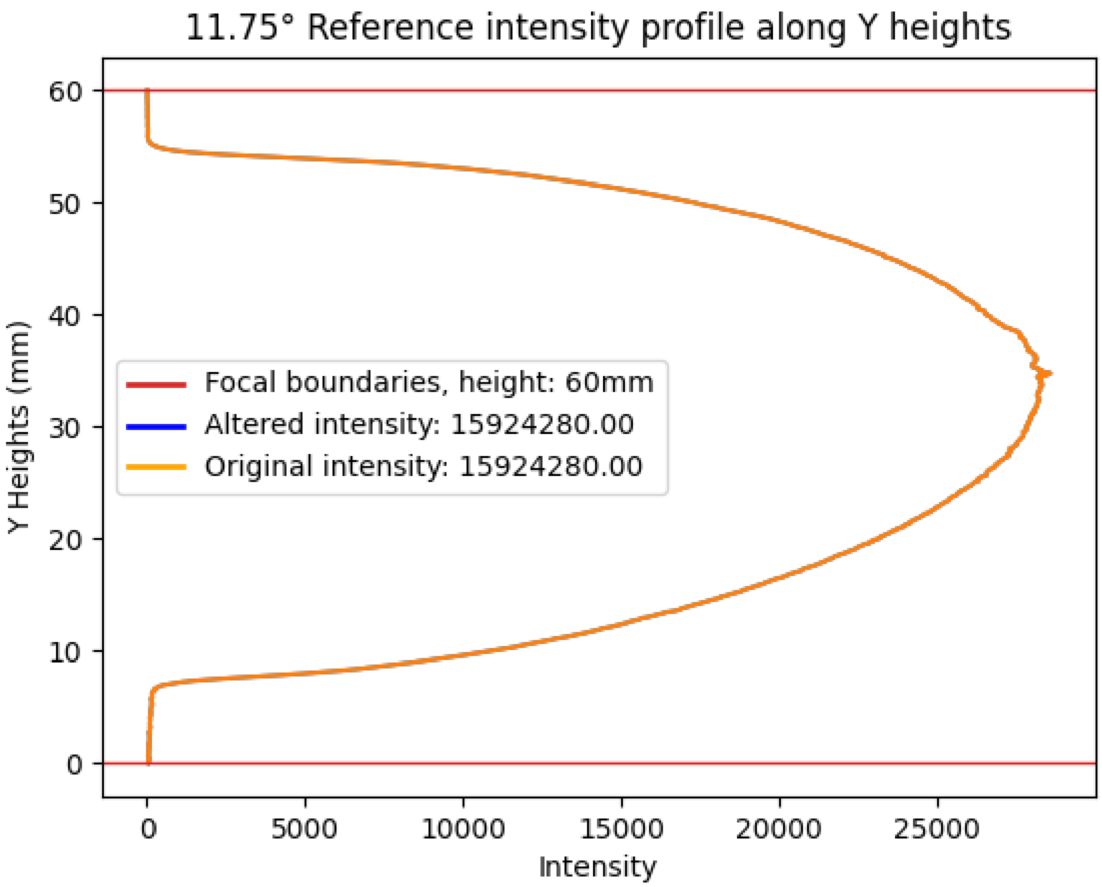

11.75° Etendue intensity plot with neutral axis:529
Column
0
200
400
600
800
0
500
1000
1500
2000
Row and Intensity

Figure



```
An error occurred trying to create min_hline, in 'show focal energy'-function: local variable 'index_to_keep_min' referenced before assignm
ent
 the hline was set to the first index of the list
An error occurred trying to create max_hline, in 'show focal energy'-function: local variable 'index_to_keep_max' referenced before assignm
ent
 the hline was set to the last index of the list
```

Figure



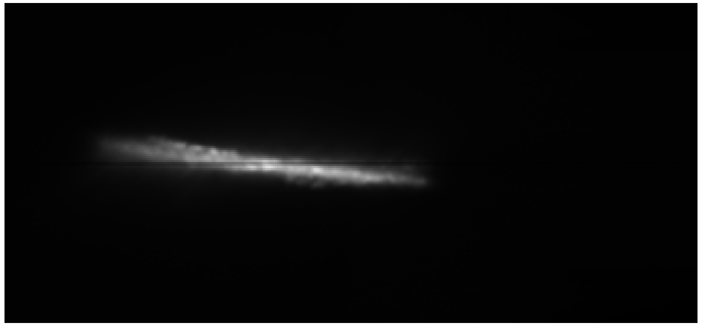
```
referece flux = 15924280
 etendue flux = 10679580
  transmission = 67%
```

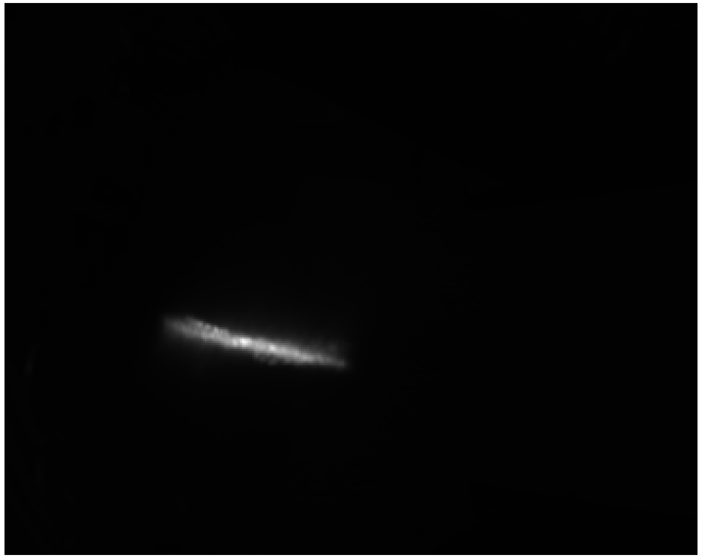
In [326…

```
clicked_points = [(234, 405), (776, 297), (769, 989), (218, 835)]
output_comparison("ImageAnalysis/15i/15i_2/0_2.tif","ImageAnalysis/15i/15i_2/0_2ref.tif","ImageAnalysis/15i/15i_2/plots/0deg",4,4,"15i 0\u0
```

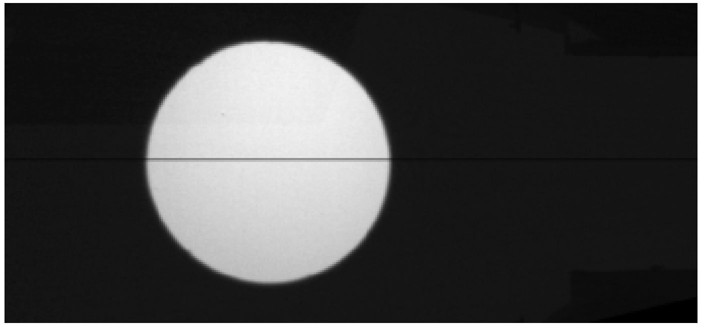
```
Clicked coordinates: [(234, 405), (776, 297), (769, 989), (218, 835)], Testing: False
Clicked coordinates: [(234, 405), (776, 297), (769, 989), (218, 835)], Testing: False
```

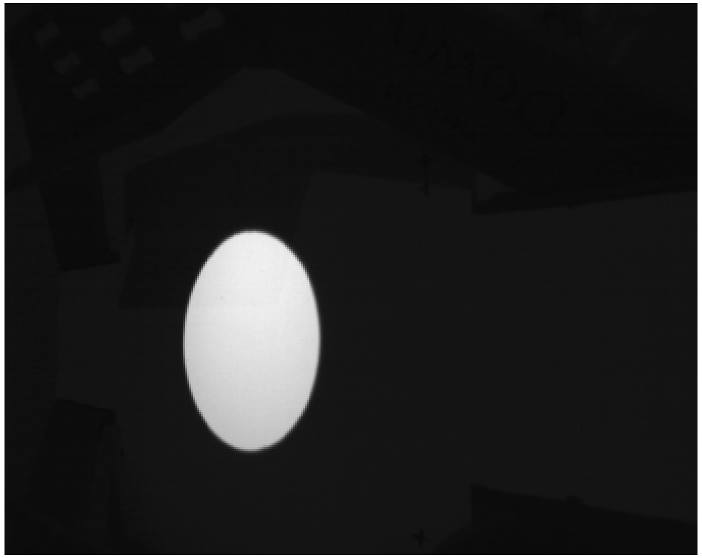
Figure



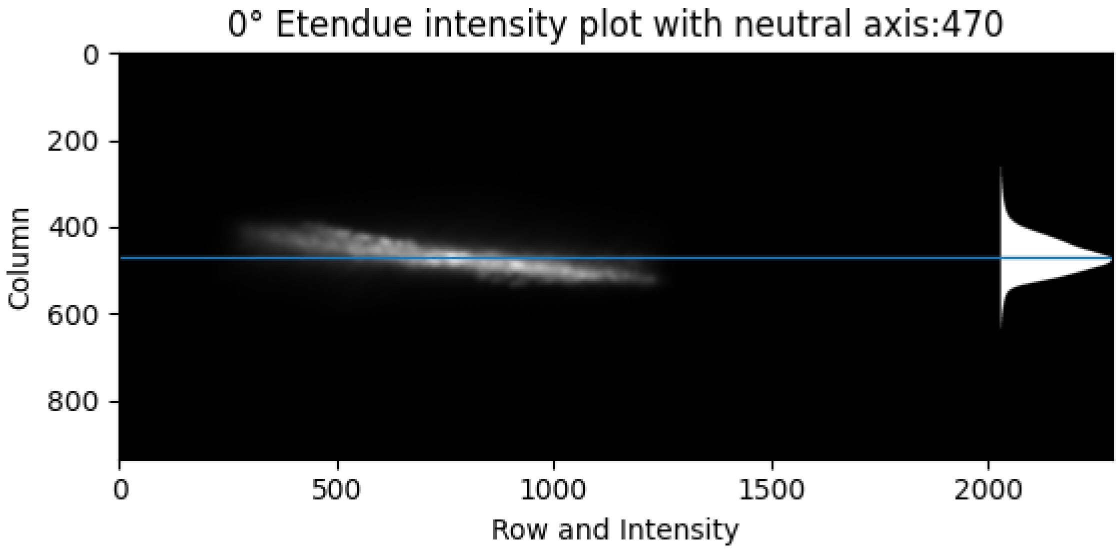


Figure

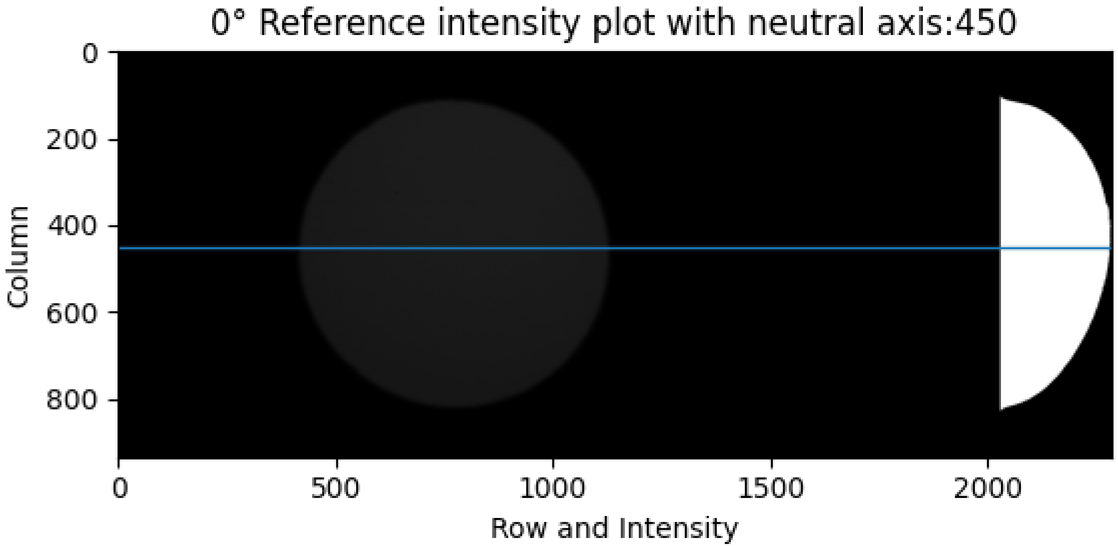




Figure

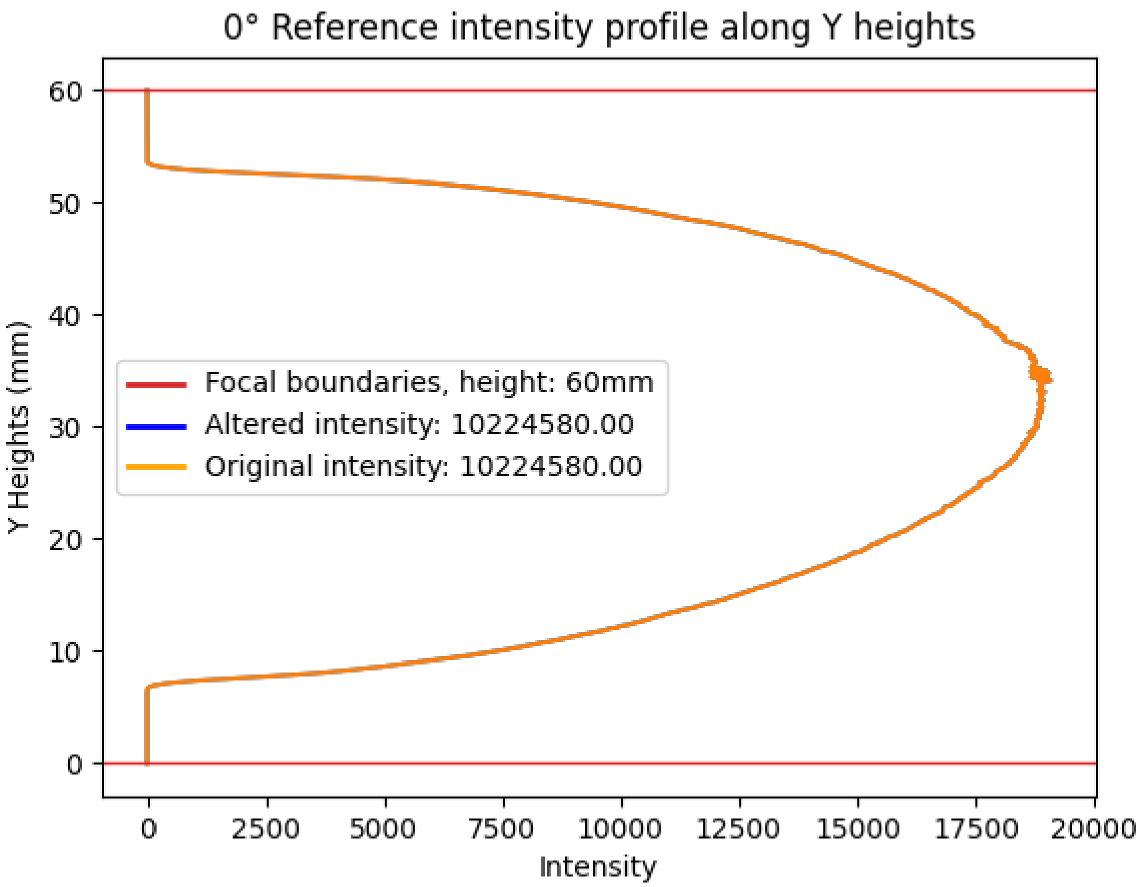

0° Etendue intensity plot with neutral axis:470
Column
0
200
400
600
800
0
500
1000
1500
2000
Row and Intensity

Figure



```
An error occurred trying to create min_hline, in 'show focal energy'-function: local variable 'index_to_keep_min' referenced before assignm
ent
 the hline was set to the first index of the list
An error occurred trying to create max_hline, in 'show focal energy'-function: local variable 'index_to_keep_max' referenced before assignm
ent
 the hline was set to the last index of the list
```

Figure



```
referece flux = 10224580
 etendue flux = 7538652
  transmission = 73%
```

This notebook contains code and analysis assisted by OpenAI's GPT-4 model.

```
In [ ]:
```