archived swh:1:dir:fc8dc8178cb109c4990a92415a6c120fe00a55fb
platform macOS | Windows | Linux

license BSD

`Version 2.9.0` (December 31, 2025)

PyPOLAR documentation

PyPOLAR is a Python-based application designed to analyze polarization-resolved microscopy data, enabling the measurement of molecular orientation and order in biological samples.

🚀 Getting Started

Desktop Installers: Check out the Latest Release to download the standalone version for your OS:

- **Windows:** `PyPOLAR_installer.exe`
- **macOS:** `PyPOLAR.dmg`

Via PyPI: PyPOLAR can be installed on any OS (Windows, macOS, Linux) via pip. See the PyPI Installation Guide for details.

	Website: fresnel.fr/polarimetry
	Source Code: github.com/cchandre/Polarimetry

PyPOLAR is based on a code originally developed by **Sophie Brasselet** (Institut Fresnel, CNRS).

📫 Bug Reports & Contact

If you encounter any issues, please reach out to the development team:

- **Manos Mavrakis** (Institut Fresnel, CNRS): manos.mavrakis@cnrs.fr
- **Cristel Chandre** (Institut de Mathématiques de Marseille, CNRS): cristel.chandre@cnrs.fr
- **Sophie Brasselet** (Institut Fresnel, CNRS): sophie.brasselet@fresnel.fr

🛠️ Built With

PyPOLAR is developed in **Python 3.8.16** utilizing the following core libraries:

- **GUI:** CustomTkinter and tksheet.
- **Scientific:** NumPy, SciPy, OpenCV, and scikit-image.
- **Data & Icons:** Matplotlib, colorcet, and Material Design icons.

For a full list of dependencies, see `requirements.txt`.

📜 License

BSD 2-Clause License

📦 Distribution

Compilation: PyInstaller 5.12.0
macOS DMG: DMG Canvas
Windows EXE: InstallForge

Table of Contents

Left Panel

The basic elements to perform an analysis are listed on the left panel: select a polarimetry method, a stack to be analyzed and a method of analysis; then click the button `Analysis`

select a polarimetry method

The list of methods is available here

Note that when `4POLAR2D` or `4POLAR3D` is selected, an image registration is required (see here for more details)

download data to be analyzed

`Open file` : select a `.tiff` or `.tif` stack file

`Open folder` : select a folder containing `.tiff` or `.tif` stack files

`Open figure` : select a `.pyfig` pickled file. Info here

select the method of analysis

Version: `2.9.0` or later

- `Thresholding` : Manually define threshold levels and select regions of interest (ROI) within the interface.
- `Mask` : Apply external binary segmentation masks to your analysis.
    - **Action**: Browse and select the folder containing your mask files.
    - **Naming Convention**: The mask filename must be **identical** to the image stack filename.
    - **File Format**: The mask must be a `.png` file.

- `ROI` : Apply external ROIs to your analysis.
    - **Action**: Browse and select the folder containing your ROI files.
    - **Naming Convention**: The ROI filename must be **identical** to the image stack filename.
    - **File Format**: The ROI must be a `.pyroi` (PyPOLAR ROIs) file, a `.roi` (ImageJ) file, or a `.zip` (ImageJ) file.

Version: prior to `2.9.0`

- `Thresholding (manual)` : thresholding and regions of interest are manually selected for each stack
- `Thresholding (auto)` : thresholding and regions of interest are selected for the first stack and applied to each stack to be analyzed in batch mode
- `Mask (manual)` : a binary segmentation mask is applied to the analysis; the name of the segmentation mask has to be identical to the one of the stack with a `.png` extension. The mask file should be in the same folder as the stack.
- `Mask (auto)` : batch analysis, similar to `Thresholding (auto)` but with a segmentation mask applied to each stack. The mask files should be in the same folder as the stacks.

For `Thresholding (auto)` and `Mask (auto)` , the menu icon is changed to 🔧

add one or multiple region(s) of interest (ROI)

Click `Add ROI` . This selection is made on the Thresholding/Mask tab; for more details on ROIs

- For polygon-shaped ROIs, click successively the left mouse button.
- For freeform ROIs, click continuously the right mouse button.
- Double click to close the ROI.

click on `Analysis`

Once the methods and data are selected, click on `Analysis` to perform the analysis; the icon becomes red ⏸ during the analysis

The analysis is performed with 8 or 16-bit images. In case of 32-bit images, they are converted to 16-bit images before analysis.

✕ close figures

Click on close figures to close all windows containing output figures

Back to Table of Contents ↑

Intensity Tab

The intensity tab displays the total intensity (total = sum over the polarization angles) image of the stack to be analyzed if the `Stack` slider in the lower right part of the tab is set to 'T', or the image *n* of the stack if the value of the `Stack` slider is set to *n*. The selected dark value has been removed from the total intensity (see Dark value for more information on the computation of the dark value).

Information on the Navigation toolbar can be found here.

- ◐ The contrast can be adjusted with the contrast slider on the right hand side of the tab. This value of the contrast will be used in the intensity images in the figures, but does not affect the analysis.
- Click the 📐 button to display the angle (in deg) and the length (in number of pixels) of the segment selected on the intensity image. In order to display the distance in μm, enter the length of a pixel in nm in the entry box right below. The angle is defined counter-clockwise in the field of view (see Angle orientation). Click the left mouse button to select the start and end of this segment. Click the square button at any time to exit the line segment drawing mode.

The lower left part of the tab indicates the name of the stack to be analyzed.

Back to Table of Contents ↑

Thresholding/Mask Tab

The **Thresholding/Mask** tab displays the total intensity image (the sum over all angles) of the stack to be analyzed, based on the threshold specified in `ILow`. Note that the Dark value is subtracted prior to displaying this intensity.

For general interface navigation, refer to the Navigation toolbar documentation.

Intensity Control (`ILow`)

The `ILow` value determines the threshold for analysis. You can adjust it using the slider or by entering a value manually in the text box. The `ILow` value used during processing is automatically saved in the exported MS Excel file (see the Options tab).

Visualization & Tools

- **Contrast & Transparency:** Adjust visualization contrast using the right-hand slider. Use the `Transparency` slider to view pixels below the threshold.
  Note: These settings assist with visualization only and do not affect the final analysis.
- **Colormap:** Click to toggle the thresholded image colormap between `hot` and `gray`.
- **Background Mode:** Switch between dark mode or light mode for the background.
- **Export Mask:** Save the binary mask as a `.png`. You can generate masks from ROIs, the intensity threshold, or a combination of both.

  > **Note:** Files are exported with the suffix `_roi` if `ROI` is selected. If `per ROI` is selected in the `Options` Tab, the masks are exported individually for each ROI with the suffix `_roi#`. If combined with `Intensity` an additional suffix `_intensity` is added. To use this mask for future PyPOLAR sessions, remove the suffix so the filename matches the name of the stack exactly.

- **ROI Manager:** Opens the ROI Manager for detailed region selection.
- **Edge Detection:** Opens the Edge Detection window. Use `Compute` to detect edges from current threshold settings or `Download` to import an external mask for contour determination.

Back to Table of Contents ↑

Options Tab

Figures

The `Show/Save` table lists all display outputs (`Composite` , `Sticks` , `Histogram` , `Intensity`):

- `Composite` image of the variable *C* displays the values of *C* as color-coded pixels on top of the intensity image using the colors in the colormap. For a list of variables for each polarimetry method, click here
- `Sticks` image of the variable *C* displays the values of *C* as color-coded sticks (centered around pixels) on top of the intensity image with a color given by the value of *C* and an orientation given by the value of *ρ*
- `Histogram` displays the histograms of the selected variables *C*. NB: the histograms for the variables *ρ* (for all methods) and *η* (for `4POLAR 3D`) are displayed as polar histograms
- `Intensity` image displays the intensity image of the Intensity tab with the applied contrast and the selected numbered ROIs

Check the boxes in the `Show` column for the figure types to be displayed, and in the `Save` column for the figures to be saved. The format of the figures to be saved is selected using the option menu in the frame `Save output` .

Variables

The `Variables` table lists all the possible variables *C*. Check the boxes for the variables *C* to be displayed and/or saved in the analysis. See the list of variables for each polarimetry method here. The second and third columns display the minimum and maximum values of the variables used for the colorbars of histograms and composite and stick maps. These elements are editable (except for ρ) if the toggle is activated.

The colormaps for each variable are detailed here.

Preferences

The features below act interactively on displayed histograms and figures.

- Checkbox **Axes on figures**: if ticked, the pixel numbers on the axes of each figure are displayed on the showed or saved figure images
- Checkbox **Colorbar on figures**: if ticked, the colorbar of each composite or stick figure is displayed on the showed or saved figure images
- Checkbox **Colorblind-friendly**: if ticked, uses colorblind-friendly colormaps on the showed or saved figure images
- Spinners for the number of pixels separating sticks on stick maps: **vertical** = number of pixels separating sticks vertically, **horizontal** = number of pixels separating sticks horizontally (i.e., 1 means every pixel, 2 means every other pixel, etc...); the spinners apply directly to open stick figures
- Spinner for the number of bins used in histograms (default=60, increment=5)
- Button : click to open the Crop Manager

Miscellaneous tools

- Button : Click this button to visualize the accuracy of the fitting per pixel. The selection of the pixel is done on the composite figure of ρ.
- Button : Click this button to merge the histograms of the variables selected in the Variables table. Select the folder containing the `.mat` files to be concatenated.
- Checkbox `per ROI` : if ticked, the results in histograms and in the excel files are displayed and saved separately for each ROI; otherwise, the results are displayed and saved by grouping all ROIs. It acts interactively on displayed histograms of grouped ROIs to convert them to histograms per ROI and vice versa.

Save output

The `Save output` table lists the saving options: `Data (.mat)` for saving the values of the variables for each pixel used in the analysis as a MATLAB `.mat` file, `Mean values (.xlsx)` for saving the mean values of the variables in a MS Excel file, and `Movie (.gif)` for an animated gif file of the

acquired stack or a region of the stack. Use the Crop Manager to define this region.
Note that the stem (i.e., filename without the extension) of the saved files is the stem of the analyzed polar stack. If the same polar stack is used for multiple analyses, the data are appended in the same MS Excel file. For a single ROI, its labels are displayed in the MS Excel file only if `perROI` is clicked.
Using the option menu `Figures`, select the format of the figures to be saved (as selected in the Figures table for the variables selected in the Variables table). Possible formats are `.pdf` (default), `.png`, `.jpeg`, `.tif`, `.tif(ImageJ)` (for later use in ImageJ/Fiji) and `.pyfig` (for later use in `Open figure` in PyPOLAR). The `.tif(ImageJ)` format saves a composite image of the intensity image and the image of the variable values per analyzed pixel. To save figures in the `.tif(ImageJ)` format, `Save` for the `Composite` image must be selected.

`.pyfig` files are PyPOLAR files storing (as a pickled file) all information on the produced figures during an analysis. This is used for minor changes in the layout of the produced figures (such as zooming, cropping, removing axes, changing colormaps, removing colorbars...) without redoing the analysis from scratch. A `.pyfig` file of the composite image displays the value of the selected variable (e.g., rho, psi...) per pixel in the image.

🎨 click this button to open a Jupyter notebook to visualize, customize and save the colorbars used in PyPOLAR. This Jupyter notebook can be found here. For more details on this Jupyter notebook and the options to generate the colorbars, click here

🗑 reinitializes the Show/Save and Variables tables.

Back to Table of Contents ↑

Advanced Tab

Dark

The dark signal can be calculated automatically or experimentally. The default is the automatic calculation which is displayed as the `Calculated dark value`. The experimentally calculated dark value can be entered in the `Used dark value` entry box by activating the Dark toggle. For `1PF`, the minimum possible value is 480.

For more details on the dark value calculation and recommendations, click here

Polarization

- The value of the offset angle used in the analysis is indicated. This angle (in degrees) is measured according to the convention mentioned here. In order to manually change this value, activate the toggle.
- ↻ or ↺ : click to select the polarization direction as clockwise or counter-clockwise. The direction shown on the button is the direction chosen in the analysis.
  The offset angle and the polarization direction are saved in the MS Excel file (see Save output).

Disk cone / Calibration data (for 1PF and 4POLAR)

For 1PF: The drop down menu lists all the disk cones included in the app. If the disk cone to be used is not in the list, select `other` and download the appropriate disk cone. Click the button 🖼 to visualize the disk cone used in the `1PF` analysis. The name of the disk cone used in the analysis is displayed below the dropdown menu. The name of the disk cone is saved in the MS Excel file (see Save output).

Click the button 🛠 for the calibration of disk cones for 1PF (for expert users); see Disk cone calibration for more details.

For 4POLAR: The upper drop down menu lists all the calibration data included in the app. If the calibration data to be used is not in the list, select `other` and download the appropriate calibration data. The calibration data is a `.txt` file containing a 4x3 matrix for `4POLAR 2D` (see template) or a 4x4 matrix for `4POLAR 3D` (see template). The name of the calibration data used in the analysis is displayed below the dropdown menu.
Select the distribution of the polarization angles (0°, 45°, 90°, 135°) using the lower drop down menu: UL (upper left), UR (upper right), LR (lower right) and LL (lower left). The choice of distribution is saved in the MS Excel file (see Save output).

Binning

This option is used to improve the quality of the stack if the signal is too weak. It performs a convolution of the stack with a kernel of 1's of size `Bin width x Bin height`. A side effect is a blurring of the stack as displayed in the intensity images. The binning size is saved in the MS Excel file (see Save output).

For more information on binning, click here

Rotation

- `Stick (deg)`: (editable) value of the angle (in degrees) to arbitrarily rotate the sticks (following the convention: positive = counter-clockwise, negative = clockwise)
- `Figure (deg)`: (editable) value of the angle (in degrees) to arbitrarily rotate the entire figure (following the convention: positive = counter-clockwise, negative = clockwise)
- `Reference (deg)`: (editable) value of the angle (in degrees) with respect to which ρ is normalized; if the change of reference angle is combined with a figure rotation, the reference angle has to be determined in the rotated figure.

Intensity removal

This option is used to remove some intensity from the stack. First, define the size in pixels (`Bin width` and `Bin height`) of the bin in the part of the intensity image. Second, select a point (center of the bin of size `Bin width x Bin height`) in the intensity image by clicking the button ⊡. Third, choose the value `Factor` (between 0 and 1) for the fraction of the mean intensity of the bin to be removed from the stack. The mean value over the selected bin weighted by the chosen factor is removed from the entire stack. The value `Removed intensity value` which is subtracted from the stack is indicated in the lower part of the panel.

Back to Table of Contents ↑

Navigation Toolbar

The Intensity and Thresholding/Mask tabs, as well as all generated figures, contain the following set of buttons that help navigate the displayed images or figures.

- ⌂ ← → : Home, Forward and Back buttons
  Forward and Back are used to navigate back and forth between previously defined views in Intensity and Thresholding/Mask tabs. They have no meaning unless you have already navigated somewhere else using the pan and zoom buttons. Home takes you to the first, default view of your data.

- ✥ Pan/Zoom button
  This button has two modes: pan and zoom. Click the Pan/Zoom button to activate panning and zooming, then put your mouse somewhere over an axes. Press the left mouse button and hold it to pan the figure, dragging it to a new position. When you release it, the data under the point where you pressed will be moved to the point where you released. If you press 'x' or 'y' while panning the motion will be constrained to the x or y axis, respectively. Press the right mouse button to zoom, dragging it to a new position. The x axis will be zoomed in proportionately to the rightward movement and zoomed out proportionately to the leftward movement. The same is true for the y axis and up/down motions (up zooms in, down zooms out). The point under your mouse when you begin the zoom remains stationary, allowing you to zoom in or out around that point as much as you wish. You can use the modifier keys 'x', 'y' or 'CONTROL' to constrain the zoom to the x axis, the y axis, or aspect ratio preserve, respectively.

- 🔍 Zoom-to-Rectangle button
  Put your mouse somewhere over an axes and press a mouse button. Define a rectangular region by dragging the mouse while holding the button to a new location. When using the left mouse button, the axes view limits will be zoomed to the defined region.

- 💾 Save button
  Click this button to launch a file save dialog. You can save files with the following extensions: `.png` , `.eps` , `.jpg` , `.pdf` , `.pgf` , `.ps` , `.raw` , `.svg` , `.tif` , `.webp` .

In `Composite` figures, one can display the coordinates, the intensity and the calculated parameter values by hovering the mouse pointer over the pixels of interest.

In `Histograms` , one can display the calculated parameter values and respective counts by hovering the mouse pointer over the region of interest in the histogram.

Back to Table of Contents ↑

Region of Interest

A Region of Interest (ROI) is a closed area within an image used to isolate specific data for analysis. You can define multiple polygon or freeform ROIs before starting the polarimetry analysis.

> **Important:** To ensure accurate data, ROIs should not overlap with one another.

To manage your selections, open the ROI Manager by clicking the icon.

How to add an ROI

1. **Threshold First:** Adjust the thresholding intensity on the Thresholding/Mask tab **before** drawing. This associates the current `ILow` value with that specific ROI.
   - *Note: You can later modify this value in the `ILow` column of the ROI Manager.*
2. **Activate Tool:** Click the `Add ROI` button .
3. **Draw on Image:**
   - **Polygon Shape:** Click the **left mouse button** successively to place vertices.
   - **Freeform Shape:** Click and hold the **right mouse button** to draw manually.
   - **Close ROI: Double-click** to complete the shape.

Once confirmed, each ROI is numbered and visible on the Intensity tab, the Thresholding/Mask tab, and within the ROI Manager list.

> **Note:** Click the `Add ROI` button at any time to exit the drawing mode. When `Add ROI` is active, it is not possible to use the navigation toolbar (for, e.g., pan, zoom). First close the ROI, or click again the button `Add ROI` , before using the navigation toolbar again.

Back to Table of Contents ↑

ROI Manager

The `ROI Manager` provides a centralized interface to edit, organize, tag, and export your Regions of Interest (ROI). Using the manager ensures that specific regions can be tracked accurately across different analysis sessions.

Open the manager by clicking the ☰ icon.

ROI Table Columns

- `ROI` : The index number of the ROI. This matches the labels shown on the Intensity and Thresholding images and is used in the Excel export. You can manually renumber these if necessary.
- `ILow` : The intensity threshold specific to that ROI. You can modify this value directly in the table.
  - *Note: When importing from ImageJ/Fiji, this defaults to 0 and should be updated for proper analysis.*
- `Label 1, 2, 3` : Custom metadata tags (e.g., cell type, treatment, or condition). These labels are included in the final Excel output.
- `Select` : Use the checkboxes to include or exclude specific ROIs from the current analysis. Toggle the column header to select/deselect all.
- `Delete` : Mark specific ROIs for removal. Toggle the column header to select/deselect all.

Action Buttons

Button	Description
Save	Exports all current ROIs and their associated `ILow` thresholds into a PyPOLAR `.pyroi` file.
Load	Imports ROIs from a `.pyroi` file or an ImageJ/Fiji `.zip` / `.roi` file. For ImageJ imports, `Label 1` is automatically populated with the ImageJ ROI name.
Delete	Permanently removes all ROIs checked in the `delete` column. The remaining ROIs are automatically renumbered in the manager and on the active images.

Back to Table of Contents ↑

Crop Manager

The `Crop Manager` provides a centralized interface to explore and edit the figures. Using the manager ensures that the exact same regions can be tracked accurately across different figures.

Open the manager by clicking the icon.

- enter the x-range (`xlim`) and y-range (`ylim`) of interest in the entry boxes by clicking enter after entering each value. The new xlim and ylim are automatically applied to all displayed figures
- click `Get` to get the x-range and y-range of the active figure
- click `Crop` for cropping figures to the same xlim and ylim as the active figure
- click `Resize` to resize all the figure windows to the same size as the active figure
- click on `Create ROI` to create a rectangular ROI with the set limit on the active figure; the values are also the ones used in the saved animated gif (see Save output)
- click on `Reset` to go back to the original x-range and y-range

Back to Table of Contents ↑

Polarimetry Methods

`1PF` : one-photon fluorescence

- variables: (ρ, ψ).
- defined as: given $I(\mathbf{x}, \alpha) = a_0(\mathbf{x}) + a_2(\mathbf{x}) \cos 2\alpha + b_2(\mathbf{x}) \sin 2\alpha$ where $\mathbf{x}$ is the position of the pixel and α the polarization angle, the values of ρ and ψ are obtained using a linear interpolation on a pre-loaded disk cone. This disk cone can be changed in the Advanced option tab
- $\rho \in [0, 180^\circ[$ (orientation) and $\psi \in [0^\circ, 180^\circ]$
- reference: A. Kress, X. Wang, H. Ranchon, J. Savatier, H. Rigneault, P. Ferrand, S. Brasselet, *Mapping the local organization of cell membranes using excitation-polarization-resolved confocal*

fluorescence microscopy, Biophys. J. 105, 127-136 (2013)

`CARS` : coherent anti-Stokes Raman scattering

- variables: (ρ, S_2, S_4)
- defined as: given $\sqrt{I(\mathbf{x},\alpha)} = a_0(\mathbf{x}) + a_2(\mathbf{x})\cos 2\alpha + b_2(\mathbf{x})\sin 2\alpha + a_4(\mathbf{x})\cos 4\alpha + b_4(\mathbf{x})\sin 4\alpha$ where $\mathbf{x}$ is the position of the pixel and α the polarization angle,
    - $S_2(\mathbf{x}) = 3\frac{\sqrt{a_2^2+b_2^2}}{2a_0}$
    - $S_4(\mathbf{x}) = 6\frac{\sqrt{a_4^2+b_4^2}}{a_0}\cos 4(\phi-\rho)$ where $\rho(\mathbf{x}) = \mathrm{atan2}\frac{b_2}{a_2}$ and $\phi(\mathbf{x}) = \mathrm{atan2}\frac{b_4}{a_4}$
- $\rho \in [0, 180^\circ[$ (orientation), $S_2 \in [0,1]$ and $S_4 \in [-1,1]$
- reference: J. Duboisset, P. Berto, P. Gasecka, F.Z. Bioud, P. Ferrand, H. Rigneault, S. Brasselet, *Molecular orientational order probed by coherent anti-Stokes Raman scattering (CARS) and stimulated Raman Scattering (SRS) microscopy: a spectral comparative study*, J. Phys. Chem. B 119, 3242–3249 (2015)

`SRS` : stimulated Raman scattering

- variables: (ρ, S_2, S_4)
- defined as: given $I(\mathbf{x},\alpha) = a_0(\mathbf{x}) + a_2(\mathbf{x})\cos 2\alpha + b_2(\mathbf{x})\sin 2\alpha + a_4(\mathbf{x})\cos 4\alpha + b_4(\mathbf{x})\sin 4\alpha$ where $\mathbf{x}$ is the position of the pixel and α the polarization angle,
    - $S_2(\mathbf{x}) = 3\frac{\sqrt{a_2^2+b_2^2}}{2a_0}$
    - $S_4(\mathbf{x}) = 6\frac{\sqrt{a_4^2+b_4^2}}{a_0}\cos 4(\phi-\rho)$ where $\rho(\mathbf{x}) = \mathrm{atan2}\frac{b_2}{a_2}$ and $\phi(\mathbf{x}) = \mathrm{atan2}\frac{b_4}{a_4}$
- $\rho \in [0, 180^\circ[$ (orientation), $S_2 \in [0,1]$ and $S_4 \in [-1,1]$
- reference: J. Duboisset, P. Berto, P. Gasecka, F.Z. Bioud, P. Ferrand, H. Rigneault, S. Brasselet, *Molecular orientational order probed by coherent anti-Stokes Raman scattering (CARS) and stimulated Raman Scattering (SRS) microscopy: a spectral comparative study*, J. Phys. Chem. B 119, 3242–3249 (2015)

`SHG` : second-harmonic generation

- variables: (ρ, S_{SHG}, S_2, S_4)
- defined as: given $I(\mathbf{x},\alpha)=a_0(\mathbf{x})+a_2(\mathbf{x})\cos 2\alpha+b_2(\mathbf{x})\sin 2\alpha+a_4(\mathbf{x})\cos 4\alpha+b_4(\mathbf{x})\sin 4\alpha$ where $\mathbf{x}$ is the position of the pixel and α the polarization angle,
  - $S_2(\mathbf{x})=3\frac{\sqrt{a_2^2+b_2^2}}{2a_0}$
  - $S_4(\mathbf{x})=6\frac{\sqrt{a_4^2+b_4^2}}{a_0}\cos 4(\phi-\rho)$ where $\rho(\mathbf{x})=\mathrm{atan2}\frac{b_2}{a_2}$ and $\phi(\mathbf{x})=\mathrm{atan2}\frac{b_4}{a_4}$
  - $S_{\mathrm{SHG}}(\mathbf{x})=-\frac{\sqrt{a_4^2+b_4^2}-\sqrt{a_2^2+b_2^2}}{2\left(\sqrt{a_4^2+b_4^2}+\sqrt{a_2^2+b_2^2}\right)}-0.65$
- $\rho\in[0,180^\circ[$ (orientation), $S_2\in[0,1]$, $S_4\in[-1,1]$ and $S_{\mathrm{SHG}}\in[-1,1]$

`2PF` : two-photon fluorescence

- variables: (ρ, S_2, S_4)
- defined as in `SRS`
- reference: P. Ferrand, P. Gasecka, A. Kress, X. Wang, F.Z. Bioud, J. Duboisset, S. Brasselet, *Ultimate use of two-photon fluorescence microscopy to map orientational behavior of fluorophores*. Biophys. J. 106 2330–2339 (2014)

`4POLAR 2D` : 2D 4POLAR fluorescence

- variables: (ρ, ψ)
- $\rho\in[0,180^\circ[$ (orientation) and $\psi\in[0^\circ,180^\circ]$

`4POLAR 3D` : 3D 4POLAR fluorescence (beta)

- variables: (ρ, ψ, η)
- $\rho\in[0,180^\circ[$ (orientation), $\psi\in[0^\circ,180^\circ]$ and $\eta\in[0^\circ,90^\circ]$

The colormap for ρ is `hsv` (and `colorwheel` from Colorcet for colorblind-friendly visualization).

The colormap for ψ, S_2, S_4, S_{SHG} is `jet` (and `viridis` for colorblind-friendly visualization).

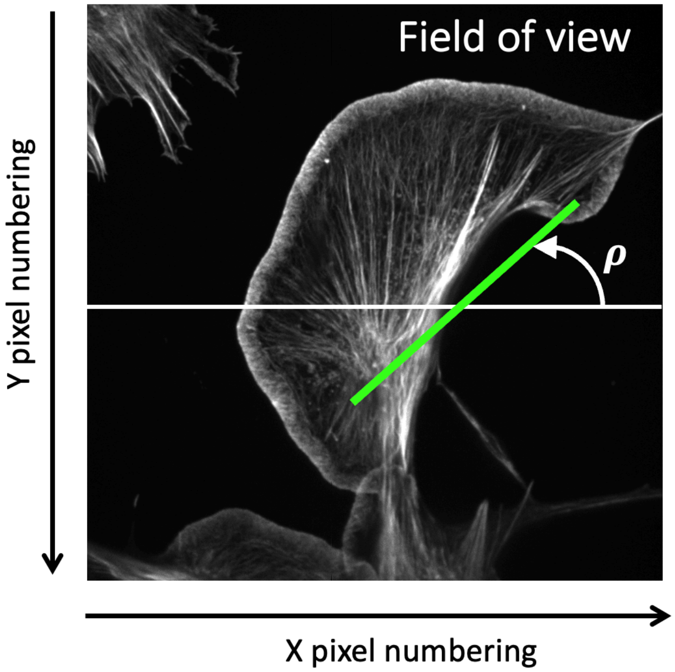
The colormap for η is `plasma`.

Back to Table of Contents ↑

Convention for Angle Origins

In-Plane Orientation (ρ)

The orientation angle ρ (used for sticks and histograms) is calculated **counter-clockwise** within the field of view. The following image defines the origin and the direction of increase. Note the specific pixel numbering convention used for the horizontal (x) and vertical (y) axes.



Out-of-Plane Orientation (η)

For **4POLAR** methods, the orientation angle η is defined relative to the imaging depth:

- 0°: The transition dipole is **perpendicular** to the image plane.
- 90°: The transition dipole is **parallel** to the image plane.

Back to Table of Contents ↑

Image Registration

For methods such as `4POLAR 2D` or `4POLAR 3D`, **image registration** is a mandatory step. This process aligns the four quadrants of your sensor—each capturing a different polarization angle (0°, $45^\circ, 90^\circ, 135^\circ$)—so they perfectly overlay for analysis. PyPOLAR utilizes the Scale-Invariant Feature Transform (SIFT) algorithm to achieve sub-pixel alignment accuracy.

Registration Workflow

When the software detects a 4POLAR method, choose one of the following:

- **Load Existing**: If you have already registered this specific optical setup, click **'Load'** to import a previously created registration file (`*.pyreg`).
- **Perform New**: If no registration exists, ensure you have a **Beads image** (`*.tif`) and a **Whitelight image** (must be named `Whitelight.tif`) in the same folder.
  - Click **'Perform'** to run the registration.
  - Review the alignment figure. If accurate, click **'Validate'**.
  - To reuse this alignment later, click **'Save'** to generate a `.pyreg` file in the current folder.
  - If the result is unsatisfactory, click **'Change'** to adjust the SIFT parameters.

Method Details

The registration pipeline follows three main stages:

1. **FOV Segmentation**: PyPOLAR analyzes `Whitelight.tif` using Otsu's thresholding and contour detection (`cv2.findContours`) to identify the boundaries of the four polarization fields.
2. **Bead Extraction**: The beads image is split into four separate images based on the Whitelight contours. Intensities are normalized and thresholded to isolate beads as distinct features.
3. **Feature Matching**:
   - The **Top-Left** quadrant is the "Fixed" (reference) image.
   - The other three quadrants are "Moving" images.
   - SIFT detects local features, and the software matches bead locations by solving a linear sum assignment problem (`scipy.optimize.linear_sum_assignment`).

- A **Homography Matrix** (`cv2.findHomography`) is calculated to map the moving images onto the fixed image.

Parameters

Adjust these parameters if the software fails to find enough matching points or detects too much noise:

- **`contrastThreshold`** : Used to filter out weak features. **Increase** this value to ignore low-contrast noise; **decrease** it if valid beads are being ignored.
- **`sigma`** : The Gaussian blur applied at the initial stage. If your images are "soft" or captured with a low-resolution camera, reducing this number may improve feature detection.

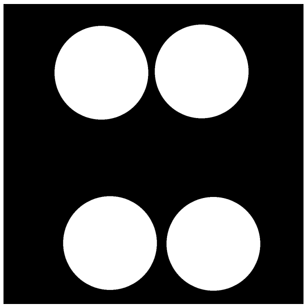
See the OpenCV SIFT Reference for more technical details.

Troubleshooting: Creating a Whitelight File

If `Whitelight.tif` is missing, you can create one manually in Fiji/ImageJ to define the quadrant locations:

1. Open your **Beads** image in Fiji.
2. Adjust **Brightness/Contrast** (`Image > Adjust > Brightness/Contrast`) until the four quadrants are visible.
3. Use the **Oval** selection tool (hold `Shift` for a perfect circle) to draw a circle over the upper-left quadrant.
4. Right-click the selection and choose `Create Mask` .
5. Move that selection to the next quadrant on the original image and repeat "Create Mask" until all four are done.
6. **Invert** the resulting mask image (`Image > Lookup Tables > Invert LUT`) so the circles are white on a black background.
7. Save the file as `Whitelight.tif` .

Example of a Fiji-built Whitelight mask:



Back to Table of Contents ↑

Edge Detection

Edge detection identifies cell boundaries and contours where image brightness changes sharply. This allows for the analysis of molecular orientation relative to structural features.

Open the module by clicking the icon. An `Edge Detection` tab appears on the right of the `Advanced` tab. The button remains blue while the module is active.

Computational Workflow

1. **Detection**: Uses a Gaussian blur followed by the Canny Edge Detector.
2. **Filtering**: Contours are extracted and filtered by a minimum **Length** parameter.
3. **Smoothing**: A Savitzky-Golay filter (`scipy.savgol_filter`) is applied to create smooth, differentiable curves.
4. **Relative Analysis**: PyPOLAR calculates the local normal to the contour and measures orientation angles ρ relative to these boundaries within a defined **Layer width**.

Parameters

The following parameters can be adjusted on the `Edge Detection` tab:

Parameter	Type	Description
Low threshold	Detection	Hysteresis lower bound (0-255).
High threshold	Detection	Hysteresis upper bound (0-255).

Parameter	Type	Description
Length	Filter	Minimum pixel number for valid contours.
Smoothing window	Smoothing	Window size for the Savitzky-Golay filter.
Distance from contour	Layer	Offset from the detected boundary (in pixels).
Layer width	Layer	Thickness of the analysis region (in pixels).

> **Tip:** If your edges look "jagged," increase the `Smoothing window`. If you are picking up too much background noise, increase the `Low threshold`.

Dark Signal Subtraction

Image processing in PyPOLAR includes the subtraction of the **dark signal**—the electronic noise produced by the camera sensor even when no photons are present. This value depends on your camera temperature, gain, and exposure time.

Automatic Calculation

By default, PyPOLAR estimates the dark signal using the following method:

1. The first image in the stack is tiled into **20x20 pixel cells**.
2. The software identifies the cell with the **lowest mean intensity**.
3. The average of this cell across all polarization angles is defined as the `Calculated dark value`.

Manual Override

If your sample fills the entire Field of View (FOV), the automatic method will mistakenly treat the darkest part of your sample as "dark noise," leading to over-subtraction.

To set a manual value:

1. Go to the Advanced Options tab.
2. Activate the **"Dark"** toggle.
3. Enter your measured value (e.g., **480** for the Institut Fresnel spinning disk setup).

> **Caution:** If your FOV is entirely filled with sample (e.g., tissue slices or dense monolayers), **do not use the automatic calculation**. You must provide a manual dark value to avoid losing real

> data signal.

Back to Table of Contents ↑

Disk Cone Calibration

Disk Cone Calibration compensates for optical distortions by identifying the most accurate disk cone for your dataset post-acquisition. The process analyzes image stacks containing a broad range of ρ values but a narrow range of ψ values. By calculating results for thousands of potential disk cones, the system identifies those with the lowest dispersion (standard deviation) in ψ.

Step-by-Step Analysis

1. **Open Calibration:** Click the 🛠 icon to open the Calibration window.
2. **Select Folders:**
   - **Disk Folder:** Select the directory containing the library of disk cones.
   - **Stack Folder:** Select the directory containing your image stacks and their corresponding mask or ROI files.
3. **Run:** Click `Start`. The interface will track the number of disks analyzed in real time.
4. **Save:** Once finished, an MS Excel file containing the full results is automatically saved in the **Stack Folder**.

> **Note: Naming Convention**: The ROI or mask filename must be identical to the image stack filename. **File Format**: The mask must be a `.png` file. The ROI must be a `.pyroi` (PyPOLAR ROIs) file, a `.roi` (ImageJ) file, or a `.zip` (ImageJ) file.

Visualization and Results

After the analysis, use the following buttons to evaluate performance:

- **Plot All:** Generates a plot showing the standard deviation of ψ for every analyzed disk.
- **Plot:** Focuses on the **best-performing disks** (lowest standard deviation). This displays:

- Standard deviation of ψ per disk.
- ρ and ψ values per disk.
- ψ values plotted as a function of ρ.

> **Tip:** Filtering Results: By default, the top 20 disk cones are displayed with their ID, full name, and standard deviation. Use the spinner to increase or decrease the number of top performers shown. Note that generated plots must be saved manually.

Back to Table of Contents ↑

Colorbars

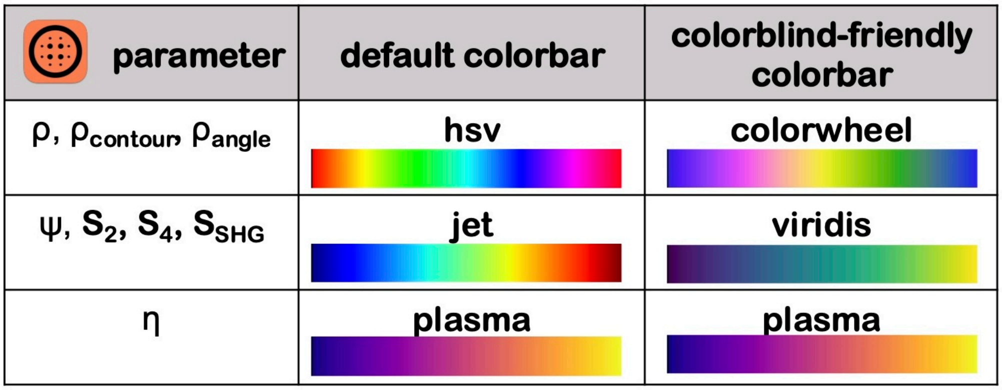
🎨 Colorbar Customization

You can visualize and export customized colorbars using our hosted Jupyter notebook:

Open Colorbar Generator (Google Colab)

Colormap Assignments

- **Orientation (ρ):** Uses `hsv` or `colorwheel` (colorblind-friendly).
- **Order (ψ, S_2, S_4):** Uses `jet` or `viridis` (colorblind-friendly).
- **Out-of-plane (η):** Uses `plasma` .



parameter	default colorbar	colorblind-friendly colorbar
ρ, $\rho_{contour}$, ρ_{angle}	hsv	colorwheel
ψ, S_2, S_4, S_{SHG}	jet	viridis
η	plasma	plasma

🔬 Look-Up Tables (LUT) for ImageJ/Fiji

To use PyPOLAR colormaps in Fiji:

1. Download the LUT files here.
2. Place them in your `Fiji.app/luts/` directory.
3. Apply them via `Image > Lookup Tables` after opening your exported `*_ImageJ.tif` file.

Back to Table of Contents ↑

Install and Run PyPOLAR with PyPI

Requirements for installing packages

- Ensure you can run Python from the command line

Open a Terminal window (or a PowerShell on Windows) and run the following command:

```
python3 --version
```

If Python is not installed on your computer or if the version is lower than 3.8, please install or upgrade to a new version of Python. For a complete guide on how to do it, click here

- Ensure you can run pip from the command line

```
python3 -m pip --version
```

For more information about these requirements, click here

Create a Python virtual environment

Run the following command once and for all on the computer where you want you to install PyPOLAR:

```
python3 -m venv pypolar_env
```

Once `pypolar_env` created, it serves as the virtual environment where PyPOLAR will be able to run. If you want to know more about Python virtual environments, click here. For a summary with the main steps, click here.

Installing PyPOLAR from PyPI

Run the following command for each installation and upgrade of PyPOLAR:

```
source pypolar_env/bin/activate
python3 -m pip install polarimetry
```

For installing a specific version:

```
source pypolar_env/bin/activate
python3 -m pip install polarimetry==XXX
```

where XXX is the version number, e.g., 2.6.4

Run PyPOLAR

Run the following command each time you want to use PyPOLAR

```
source pypolar_env/bin/activate
pypolar
```

Back to Table of Contents ↑

Python Virtual Environment Primer (Windows, macOS, Linux)

This guide helps you create and activate an isolated Python environment using `venv` on **Windows**, **macOS**, and **Linux**.

Prerequisites

- **Python 3.x** installed (preferably 3.6+)
- `venv` module is included by default with Python 3.3+
- Verify your installation:

  ```
  python --version     # or python3 --version
  ```

1. Create a Virtual Environment

To create a virtual environment (named `pypolar_env`, though you can choose any name), open a terminal or command prompt in your project directory and run the following command:

💻 Windows

```
python -m venv pypolar_env
```

🍎 macOS / 🐧 Linux

```
python3 -m venv pypolar_env
```

This creates a subdirectory called `pypolar_env/` that contains the isolated Python interpreter and dependencies.

2. Activate the Virtual Environment

💻 Windows (Command Prompt)

```
pypolar_env\Scripts\activate
```

💻 Windows (PowerShell)

```
pypolar_env\Scripts\Activate.ps1
```

> **Note:** If you get an execution policy error in PowerShell, run:
>
> ```
> Set-ExecutionPolicy -Scope Process -ExecutionPolicy Bypass
> ```

🍎 macOS / 🐧 Linux

```
source pypolar_env/bin/activate
```

When activated, your shell prompt will prefix with `(pypolar_env)`.

3. Use the Environment

Once the environment is active:

- Install packages with `pip`, e.g., `Polarimetry`:

  ```
  pip install polarimetry
  ```

- These packages will be isolated to the virtual environment.

4. Deactivate the Environment

When you're done working in the virtual environment, deactivate it:

```
deactivate
```

5. Optional Tips

Check installed packages:

```
pip list
```

Save your environment:

```
pip freeze > requirements.txt
```

Recreate environment from file:

```
pip install -r requirements.txt
```

Additional Notes

- You can safely delete the `pypolar_env/` folder to remove the environment.
- IDEs like **VS Code** automatically detect and use the `pypolar_env` if placed in your project folder.
- For project-based work, it is good practice to include the `pypolar_env/` folder in your `.gitignore`.

Back to Table of Contents ↑