# PyHoLo Manual (v. 4.0)

## A) Tools required for running PyHoLo software

In order to run PyHoLo you will require:

- Windows Vista, 7, 8, 10 (≥ Vista),
- Python language interpreter version 3.6 or later (e.g. Anaconda3),
- external Python libraries, such as PyQt5, scikit-image.

1. Download the Anaconda environment from https://www.anaconda.com/products/individual. Choose a version compatible with Python 3.6 or later.

2. Install the Anaconda environment by running the installation file and following the instructions on the screen. We recommend you to install Anaconda in the default location suggested by the instalator. In Windows it will be 'C:\Users\user_name\Anaconda3'.

3. After installation open Anaconda Prompt. In Windows 7 it can be found under Start menu, in All Programs → Anaconda3 → Anaconda Prompt.

4. Newer versions of Anaconda (i.e. versions compatible with Python 3.7 or later) come with all libraries required for PyHoLo to work. However, older Anaconda versions may be missing newer distributions of those libraries, e.g. PyQt5, scikit-image. To install those packages manually (i.e. if you stumble upon some error during running PyHoLo), type the following command in the Anaconda Prompt:

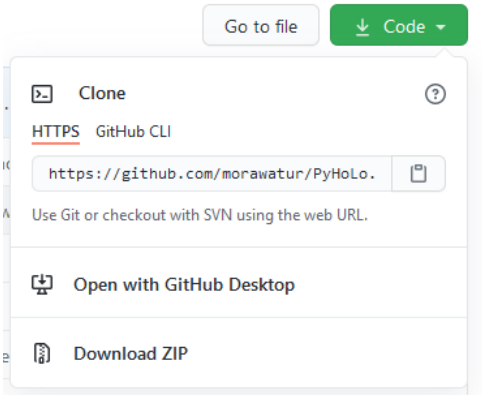
```
conda install <package_name>
```

(e.g. `conda install pyqt`)

After each command you will be asked for confirmation of the library name which you want to install. You can confirm it by typing 'y' into the Anaconda Prompt and pressing Enter. If the installation of all libraries is successful, you can close the Anaconda Prompt.

## B) Running and using PyHoLo software

1. Download PyHoLo software from https://github.com/morawatur/PyHoLo by clicking on the green 'Code' button in the upper right part of the file list and then selecting 'Download ZIP' option.




2. Extract files from zip archive to a directory on local disk (e.g. C:\python_programs\PyHoLo).

3. Open the Anaconda Prompt. In Windows 7 it can be found under Start menu, in All Programs → Anaconda3 → Anaconda Prompt.

4. From the Anaconda Prompt level go to the directory with extracted files. You can do it by entering the following command:

```
cd path_to_directory
```

(e.g. `cd C:\python_programs\PyHoLo`)

5. Run the PyHoLo.py script by entering the following command:

```
python PyHoLo.py
```

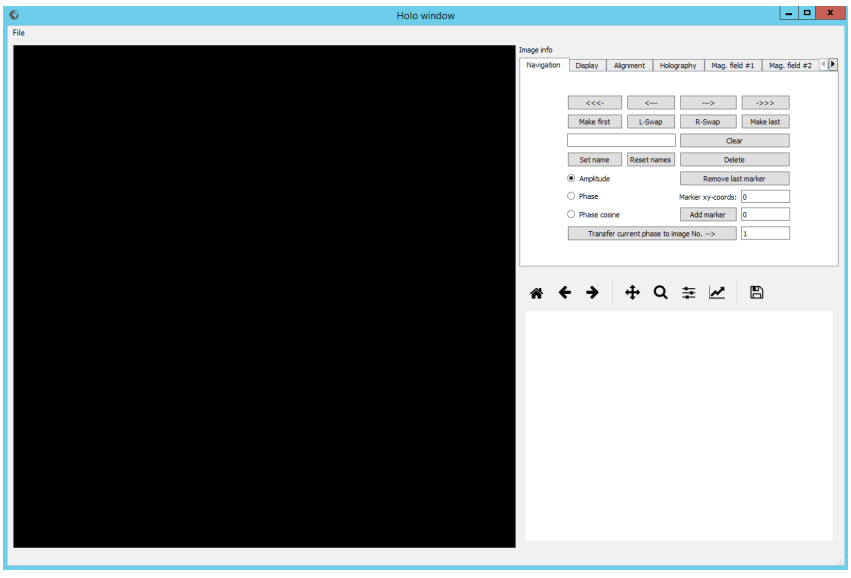



Fig. 1. PyHoLo window after starting the program.

PyHoLo window should appear on the screen, with blank display on the left side, control panel in the upper right and plotting area in the lower right part of the window, see Fig. 1. At this point program doesn't offer much functionality. In order to start working with PyHoLo you need to load images (holograms) from the disk. This can be done by expanding the File menu in the menu bar and selecting one of the two available options (Fig. 2). Let's start with the second option - 'Open dm3 series' - which was the only way of reading images in the older versions of PyHoLo. After clicking on that option a dialog window should appear, asking you to point to a directory with series of images.

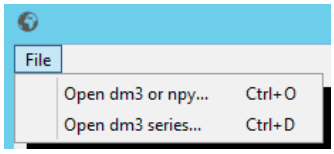



Fig. 2. Use File menu to load images into the PyHoLo.

Note 1: In this case there is a strict requirement for naming files which belong to one series. Images (files) in a series must have a common set of characters denoting name of a series and they must be numbered starting from '1' or '01'. For example you could name your files as follows:

holo_1.dm3, holo_2.dm3, holo_3.dm3, …

img01.dm3, img02.dm3, img03.dm3, ...

(notice that all characters apart from the image number are the same in all filenames).

Note 2: Using 'Open dm3 series' (as name suggests) requires the images to be stored in dm3 file format. It is the internal file format of Gatan Microscopy Suite (formerly: DigitalMicrograph), software which is commonly used in TEM image analysis.

To start reading images, use the dialog to go to the directory with series of dm3 files, then select the first file in the series and press Open. Information about the progress of reading images is displayed in the interpreter window. When all images have been loaded into the PyHoLo, first image in series is shown in the display window.

By default the loaded images are treated as amplitude images. PyHoLo distinguishes two types of images: amplitude and phase. To change the default type of loaded image you can place the optional 'info.txt' file in the directory with the image files. Info file should consist of three tab-separated columns with the number of rows corresponding to the number of images. In Fig. 3 you can see an example showing the structure of info file content. It tells us that there are four images in the series. In the 1st column of info file there are the filenames (without the 'dm3' extension) of all image files. In the 2nd column there are image names which should briefly describe the content/character of images. In this example we have four holograms, two of which were acquired for the upside orientation of the sample and another two for its downside orientation, hence the 'up' and 'down' words. Moreover, each pair of holograms consists of a reference hologram and an object hologram, hence 'ref' and 'obj' words. Image name from the

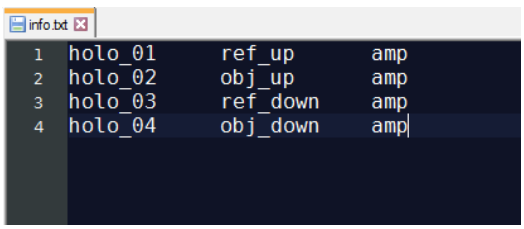



Fig. 3. Example of 'info.txt' file content.

second column will be displayed in the PyHoLo window, next to the corresponding image. Its main purpose is to help identify the image in the queue.

Finally, the 3rd column of info file lists types of images. There are two possible options: 'amp' for amplitude and 'phs' for phase. If any other string of characters is given (or no string at all) image will be identified as amplitude image (default type). In most cases, source images acquired on TEM are amplitude images, but 'phs' option can be handy if you would like to load the phase images resulting from hologram reconstruction back to the PyHoLo and analyze them further. This however would require you to convert the raw binary data exported from PyHoLo to dm3 image with the use of GMS software (more on this topic will be discussed later, in the section dedicated to exporting images).

Now let's go back to the first option in the File menu: 'Open dm3 or npy file…'. This function allows you to open any number of images stored in dm3 format or npy format (see Note 4) and there are no special requirements for naming image files. User decides which files are to be loaded by selecting one or more files in the dialog and clicking Open. This option doesn't make use of an info file, which means that file names will be treated as image names after they are opened in PyHoLo. You can indicate the type of images which are to be loaded (before selecting 'Open dm3 or npy file…'), by clicking on Amplitude or Phase radiobutton in Navigation tab (see black arrow in Fig. 5). Choosing third radiobutton - Phase cosine – will also result in phase image type.

The upside of using 'Open dm3 or npy file…' is that you don't have to follow any file naming requirements. On the other hand, in one batch you can load only one type of images (amplitude or phase). Moreover, you might need to assign more descriptive names to images (as in 2nd column of info file) one-by-one after they are opened in PyHoLo.

Note 3: Another upside of 'Open dm3 or npy file…' is that it allows to open numpy array files (files with npy extension). PyHoLo uses Pyhon library *numpy* for a large part of calculations performed on 2-dimensional arrays of data (i.e. images) and can easily export such data to separate npy files. Those npy files can then be loaded back to PyHoLo later, without any conversions. Unfortunately this isn't the case for dm3 files - PyHoLo can read dm3 files, but it cannot export data to a dm3 file. It is however possible for PyHoLo to export data to raw binary file. Then such file can be imported to GMS software and saved as dm3 file.

Note 4: Sample series of four electron holograms is located in the 'input' folder along with the rest of PyHoLo files.

After selecting files and clicking 'Open' you will be able to see the information about next images being opened in the interpreter window. At the same time those images will be displayed one-by-one in the PyHoLo window.

## C) General description of PyHoLo GUI

PyHoLo window (Fig. 4) consists of three main parts: [1] image display, [2] control panel divided into 7 different tabs and [3] a plotting area. Below you will find the description of all program functions.

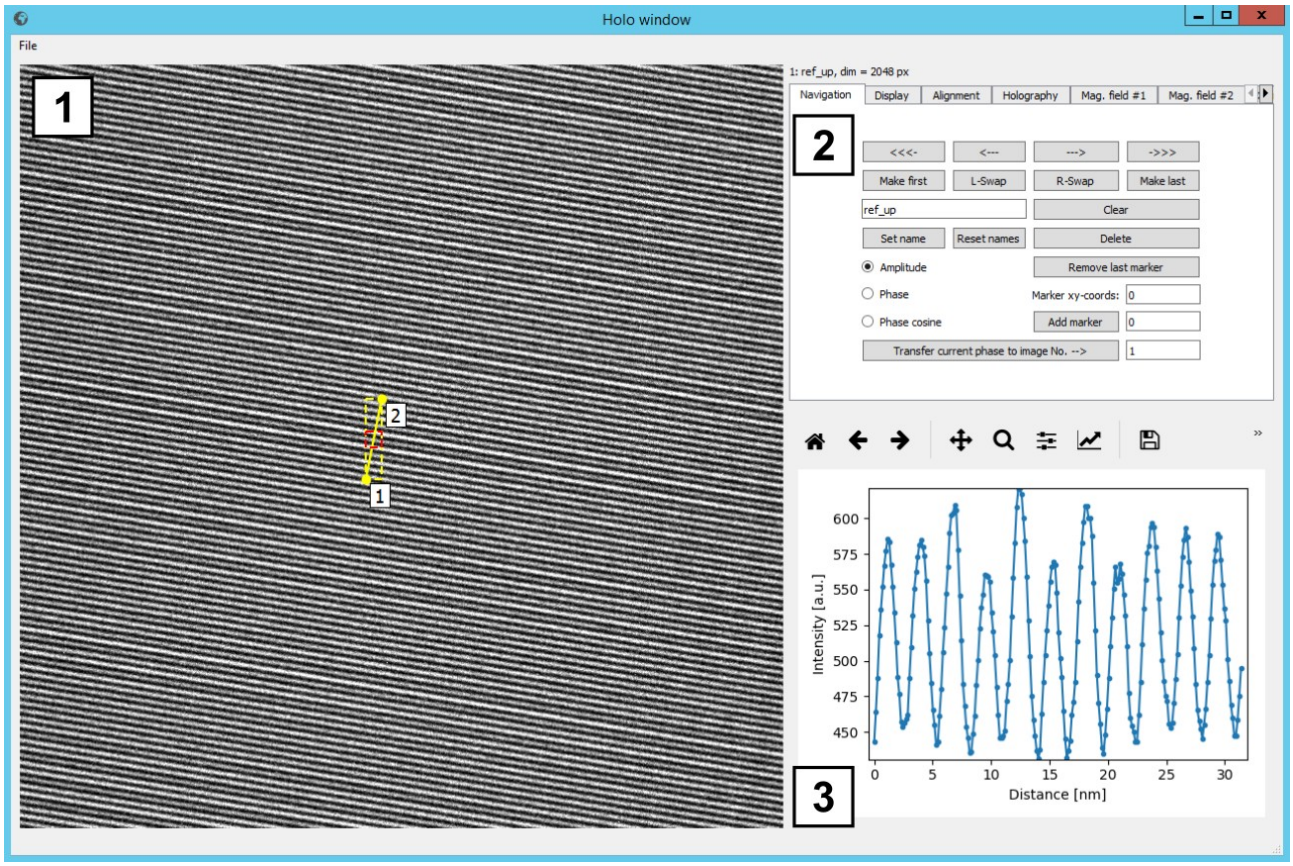



Fig. 4. Graphical user interface of PyHoLo program.

### 1. Display for images

This window displays images from the current image queue, including holograms, amplitude and phase images, phase contours and the results of other operations, such as FFT, sum, difference and gradient calculation. If you click on the display window with your mouse you can mark a point. Marking points is necessary for some of the procedures to take action, e.g. image alignment. Each time you mark a point on the displayed image the information about amplitude and phase values of the selected pixel will be displayed in the terminal window.

### 2. Control panel

Control panel is divided into 7 tabs: Navigation, Display, Alignment, Holography, Mag. field #1, Mag. field #2 and Corrections.

The list of features (and their functions) included in individual tabs are listed below.

## 2.1. Navigation tab

**<<<- / - >>>** Go to the first/last image in series (queue).
**<--- / --->** Go to the previous/next image in series (queue).

**Make first**, **Make last** - move current image to the beginning/the end of queue.

**L-Swap**, **R-Swap** - swap currently displayed image with the prevoius/the next one (but keep the same image displayed).

**Set name** - assign name to the current image in order to discern it from other images in the queue. Enter image name in the textbox above the button. The name will be saved after clicking the **Set name** button. If user decides to export the image to a file then this name will also be used by default as the file name.

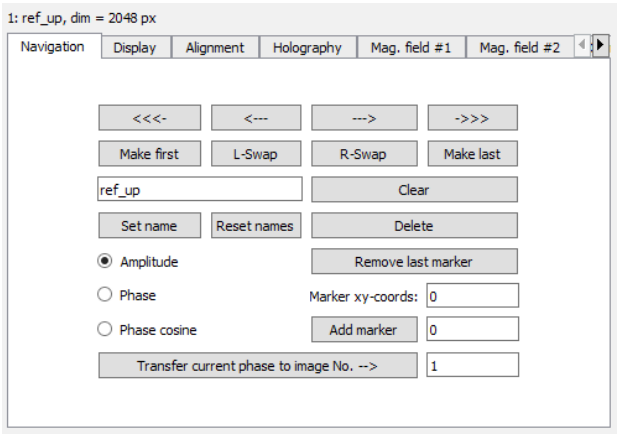



Fig. 5. Navigation tab

**Reset names** - reset names of all images in the queue to 'img_01', 'img_02', 'img_03', etc.

**Amplitude/Phase/Phase cosine** - switch between amplitude, phase and phase cosine of the current image.

**Clear** - remove all points marked on the current image.

**Delete** - remove the current image from the image queue (this operation cannot be undone).

**Remove last marker** - remove last point from the set of points marked on the current image.

**Add marker** - draw marker on the current image at the selected x-y coordinates. In general, user can add marker to the image at any time by clicking on the selected place in the image with the mouse cursor. Alternatively she can type in the (x, y) coordinates of a marker to be drawn and click **Add marker** button (x - upper input box, next to the 'Marker x-y coords:' label; y – lower input box). Values of x and y correspond to the numbers of pixels starting from the left and the top edges of the image display respectively. Given coordinates are limited by the display size which is 768x768 pixels. It means that the top-left corner of the display corresponds to (x, y) = (0, 0), and its bottom-right corner corresponds to (x, y) = (767, 767). Whenever you add a marker to the image the information about its position and the corresponding pixel value is printed to the interpreter console. This information consists of: display coordinates (limited by image display size), actual coordinates (limited by actual image size), as well as values of amplitude and phase of the pixel in that position.

**Transfer current phase to image No. -->** button is pretty much self-explanatory, i.e. assign phase of the currently displayed image to another image, which is identified by the number given in the textbox next to the button.

Note 5: Alternatively you can navigate through the image series with A (previous image) and D (next image) keys on your keyboard. You can also switch between Amplitude, Phase and Phase cosine images with W, S and X keys, respectively.

## 2.2. Display tab

### 2.2.a) Display panel

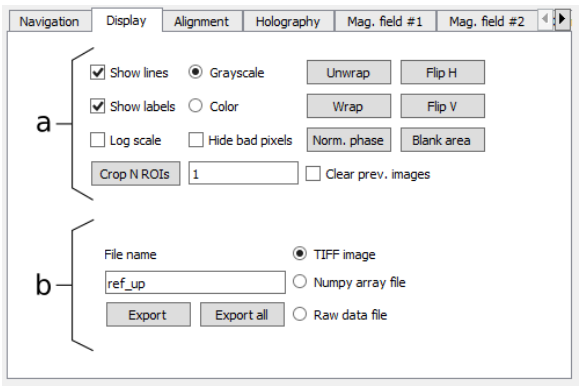



Fig. 6. Display tab

**Show lines** - if this box is checked then lines will be drawn between points marked on the image.

**Show labels** - if this box is checked then numbering labels will be displayed next to the markers.

**Log scale** - if this box is checked then the image will be displayed in logarithmic scale. Unchecking this box will cause the image to be displayed in linear scale.

**Grayscale/Color** - switch between grayscale and RGB color maps.

**Hide bad pixels** - hide outlier pixels (i.e. pixels with values significantly greater or lower than the mean value for that image) to obtain better contrast of the image. This option is used only for display reasons and doesn't affect the pixel values during any of the calculation procedures.

**Unwrap** - unwrap the image phase so that the phase is not limited to the $(-\pi, +\pi)$ range, i.e. discrete changes (jumps and drops) in phase will become continuous.

**Wrap** - wrap the image phase, i.e. performs modulo operation which transforms $+\pi+\Delta\pi$ values (where $0 < \Delta\pi < 2\pi$ ) into $-\pi+\Delta\pi$ values.

**Norm. phase** - normalize phases of images in the queue, so that the phase value (or average phase value) of selected pixel (or area) is the same for all images (i.e. 0). Before using this function user must select one or two points in the current image to indicate specific pixel or area of pixels, respectively. Then the phase value of selected pixel (or average phase value in selected area) will be subtracted from all pixels in every image in the queue.

**Flip H** - replace current image (its amplitude as well as its phase) with its horizontal mirror image.

**Flip V** - replace current image (its amplitude as well as its phase) with its vertical mirror image.

**Blank area** - blank pixels (i.e. assign zero amplitude and zero phase) in the region defined by user. In PyHoLo region of interest can be defined by adding two markers, top-left and bottom-right, to the image (see Fig. 7).

**Crop N ROIs** - crop square fragment from the current image and the next N-1 images. User must define the cropping area and specify number of images which are to be cropped (inluding the current image), see Fig. 7). Defined rectangle area will be automatically transformed to a square with side length equal to the shorter side of a rectangle. Cropped fragments will be placed after the original N images. If the **Remove original images** checkbox is checked then the source (original) images will be deleted from the queue after the cropping is finished.

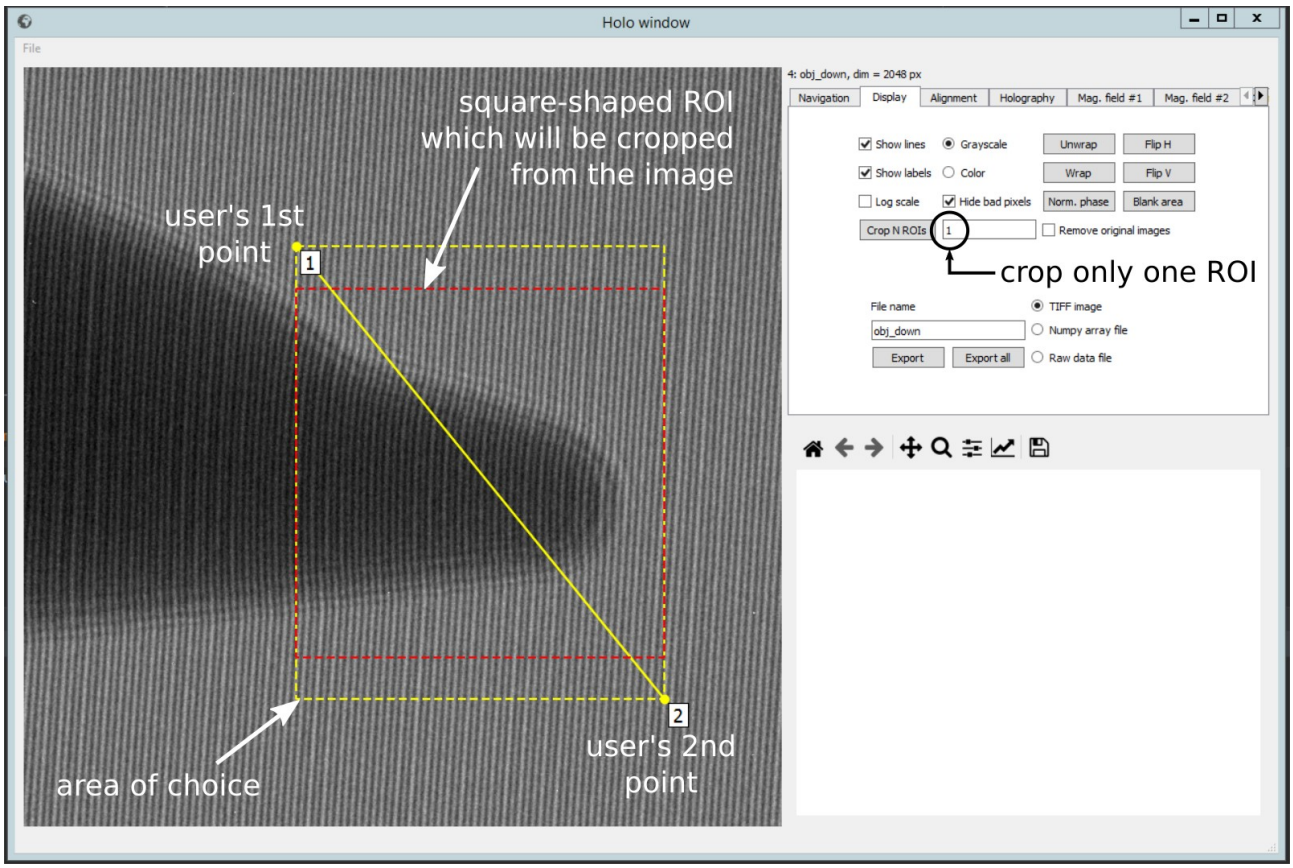



Fig. 7. The way of selecting an area for cropping. Yellow dashed line shows the rectangle created by selection of two points (1 - top-left, 2 - bottom-right). Red dashed line on the other hand shows the actual fragment which will be cropped from N images. In the above example only one fragment will be cropped, because the N value in the textbox indicated by black circle is 1.

### 2.2.b) Export panel

**File name** - in this textbox you can enter path and name of the exported file. You don't have to enter file extension. You can specify the file format of exported image by clicking on one of the corresponding radiobuttons (**TIFF image/Numpy array file/Raw data file**). In the case of images saved to TIFF format or Numpy array format the extension (png or npy) will be added automatically to the file name. Raw binary files don't need an extension. The default path for exporting data is the 'output' catalogue in the directory which contains source files of PyHoLo. For example, typing 'img' in the **File name** textbox will cause the image to be saved in the '<<PyHoLo dir>>/output' directory, under the name 'img.png', 'img.npy' or 'img' depending on the selected file format.
**Export** - save current image to the given file. Note that after clicking **Export** button only one of the image representations is saved, i.e. amplitude, phase or phase cosine, depending on the image type currently displayed.
**Export all** - save all images to the default directory and use image names as file names. All images will be saved according to the same image representation, i.e. amplitude, phase or phase cosine, depending on the currently selected image type. The same applies to the selection of **Grayscale/Color** and **TIFF image/Numpy array file/Raw data file** options.

## 2.3. Alignment tab

### 2.3.a) Manual alignment panel

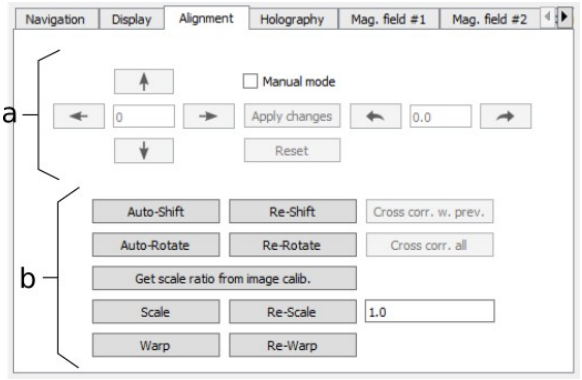



Fig. 8. Alignment tab

After checking the **Manual mode** checkbox the buttons for manual alignment of images (both amplitude and phase) become active, including buttons for shifting image horizontally (← →), vertically (↑ ↓) and rotating image clockwise and counter-clockwise (↷ ↶). In the textbox between shifting buttons you can enter the number of pixels by which the image is to be shifted in the chosen direction. Similarly you can define the angle of rotation (in degrees) in the textbox between rotation buttons.

**Apply changes** - save changes introduced to the image by manual alignment.

**Reset** - revoke introduced changes. Changes will also be revoked if you switch to another image or uncheck the Manual mode checkbox before applying changes done to the image.

### 2.3.b) Automatic alignment panel

This panel allows to align the current image with the preceding image semi-automatically. Generally, to align images, you must mark two sets of points indicating the same characteristic features on both images (the current and preceding images).

**Auto-Shift** - shift the current image to align it with the preceding image, based on two sets of marked points. Average shift between images is calculated based on the shifts between individual pairs of points. New, shifted image is added to the queue.

Note 6: Both sets of points must contain the same number of points, and the points must be marked in the same order on both images. In the case of automatic shift at least one point has to be marked on each image.

**Auto-Rotate** - rotate the current image to align it with the preceding image (shift operation may also be required after rotation). Average rotation angle is calculated based on the rotation angles determined from respective pairs of marked sections. New, rotated image is added to the queue.

Note 7: In the case of automatic rotation at least two points have to be marked on each image (two points define a section; two sections can be used to determine rotation angle between images).

Shifting as well as rotating an image will cause a loss of data on the sides of that image. At the same time, areas near opposite sides will have to be filled with zero values, i.e. shifting an image by [+10, +10] pixels will produce black stripes on upper and left sides of the image. In such case the

**Crop N**(=2) **ROIs** button may become handy to cut off those black areas.

**Re-Shift** - after each shift-operation last shift value is saved in a program's memory. When Re-shift button is clicked the current image gets shifted in the same way as the last shifted image.
**Re-Rotate** - similarly, after each rotate-operation last value of rotation angle is saved. When Re-rotate button is clicked the current image gets rotated by the same angle as the last rotated image.

**Scale** - use sets of points marked on two images (current and preceding images) to determine difference in scale between them. Then rescale the current image to make both images scaled equally. Rescaled copy of the second image is added to the queue. If there are no points marked on images or if the number of points doesn't match, the value from adjacent input textbox is used to scale the current image automatically.
**Re-Scale** - after each rescale operation the scaling factor is saved so it can be used to rescale any other image in the same way.
**Get scale ratio from image calib.** - source dm3 images usually contain information about image calibration (pixel size). Clicking Get scale ratio… button determines scaling factor from the relation between image calibrations of two images. This value is displayed in the input textbox, adjacent to the Re-Scale button, and can be used in the manual scaling operations.

**Warp** - transform one of the images geometrically to make one set of points (on the first image) correspond exactly to the other set of points (on the second image), i.e. after warping transformation points with the same numbers on both images will overlap with each other. Clicking Warp button will add a warped copy of the second image to the image queue.
**Re-Warp** - repeat last warp transformation on the current image.

Note 8: In all cases which involve selecting two sets of points on two images the function button (Auto-Shift, Auto-Rotate, Scale, Warp) must be clicked when the second image is selected (displayed). Then the software will know to use the previous image as the reference image and to modify (align, rescale, warp) the current image.

Note 9: For the warping effect to be satisfactory, you should mark a significantly large number (i.e. above 4) of points on both image.

## 2.4. Holography tab

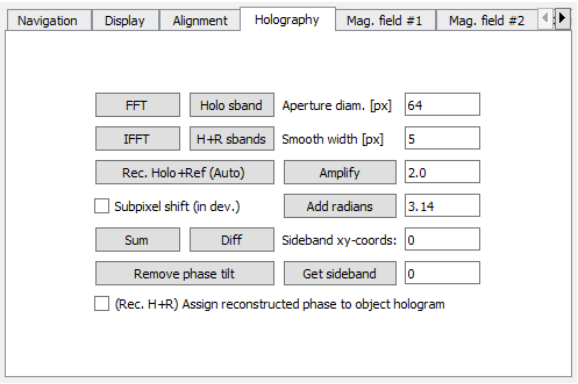



Fig. 9. Holography tab

**Aperture diam.** - diameter (in pixels) of numeric aperture which is used to mask one of the sidebands on the Fourier transform of the hologram (i.e. filter out everything except that sideband). Size of the aperture determines the range of spatial frequencies which will be considered in the phase reconstructed from hologram – smaller aperture means that only lower frequencies will be taken into account causing the phase image to be less noisy. On the other hand significantly large aperture is needed in order to observe faster changes of phase, i.e. better resolution of phase image. The default diameter of numeric aperture is set to 64 pixels.
**Smooth width** – smoothing width of the aperture edge (in pixels). Tukey window is imposed on the image of the (cropped and centered) sideband in order to smooth out the sudden drops of intensity on the edges of aperture. The default smoothing width is set to 5 pixels. The effective size of the aperture is: aperture diam. + 2*smooth width.
**FFT** - calculate and display Fourier transform (FFT) of the current image.
**IFFT** - calculate and display inverse Fourier transform of the current FFT image.

Note 10: After clicking FFT button the Log scale display mode is automatically switched on in order to see all features on the FFT image. Similarly, after clicking IFFT button the Log scale is automatically switched off.

**Holo sband** - filter the sideband selected for hologram reconstruction. Before using this function you must calculate FFT of the hologram (FFT button) and mark the area around the sideband chosen for hologram reconstruction by selecting two points: top-left and bottom-right (see Fig. 10). When you click the Holo sband button the program will find center of the sideband (i.e. position of maximum intensity) inside of the marked area. Then it will crop the sideband with the circular aperture of a given diameter (i.e. rest of the FFT will be masked) and shift the sideband to the center of FFT image. Moreover a Tukey window will be applied to this modified FFT image in order to smooth out edges of the aperture. This sideband image will be added to the queue. In order to see the result of hologram reconstruction you need to calculate the IFFT of the sideband image (IFFT button).
**H+R sbands** - this function works in a similar way as Holo sband, but it uses also the reference hologram (i.e. equivalent sidebands are filtered from object hologram and from reference hologram). In this case you must calculate the FFT of reference hologram first, before clicking H+R sbands button. Also, both holograms must be ordered in the appropriate way – first the reference hologram and then the object hologram. After FFT calculation you have to mark one of the sidebands and click H+R sbands button. This will cause the program to locate center of the sideband (on the reference FFT image) and to crop it with the aperture from the reference FFT image as well as from the object FFT image.

Note 11: You don't have to calculate the FFT of the object hologram yourself. The program will do it for you after you click H+R sbands button.

Two images with centered sidebands will be added to the queue. Then you can calculate IFFT from both of them in order to observe the reference phase shift as well as the total phase shift introduced to wave function by internal fields of a sample. You may need to use the Unwrap button (Display tab) to unwrap phases before applying next steps. Reference holograms are used to remove the artifacts related to the specific characteristic (information transfer) of CCD camera used for hologram registration. You can remove those artifacts by calculating the difference between total object phase and reference phase. You can accomplish this by using Diff function which will subtract phase of the previous image (reference phase) from phase of the currently displayed image (object phase).

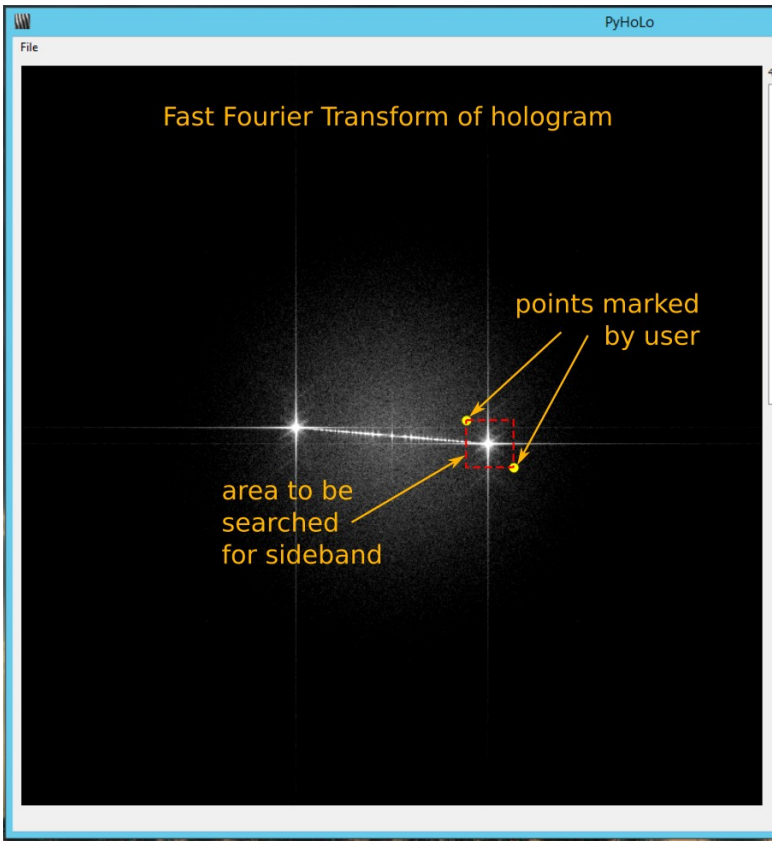



Fig. 10. The way of locating sideband on Fourier transform of typical hologram. Program searches for center of the sideband (i.e. pixel of maximum intensity) inside of the square area marked by user.

**Rec. Holo+Ref (Auto)** - automatically performs full reconstruction of the object hologram with the assist of the reference holograms. All you need to do is to calculate FFT of the reference image, mark the sideband and click the Rec. Holo+Ref (Auto). After a short while a total phase (after substracting the reference phase) should be displayed.

**(Rec. H+R) Assign reconstructed phase to object hologram** - check this option to automatically assign the phase resulting from Rec. Holo+Ref (Auto) operation as the phase of the object hologram, which was used for reconstruction.

**Sum** - calculate the sum of two phases: 1 - phase of currently displayed image and 2 - phase of the previous image. Resulting image will be added to the queue after both images.
**Diff** - calculate the difference between two phases: 1 - phase of currently displayed image and 2 - phase of the previous image. Resulting image will be added to the queue after both images.

Note 12: When you apply Sum or Diff function the resulting image will have its amplitude equal to the product or quotient of amplitudes of two previous images, respectively.

**Amplify** - amplify the phase image $\phi$ according to the following formula $\phi' = \cos(\alpha \cdot \phi)$, where amplification factor $\alpha$ can be specified in the adjacent textbox. This operation adds a new, amplified image to the queue.
**Add radians** - add a constant number of radians, specified in the adjacent textbox, to all pixels of the current phase image. This operation adds a new, modified image to the queue.

**Remove phase tilt** - remove constant phase gradient from the phase image. Four points in the image are taken, two for each axis, to determine the phase gradient. By default (if no points are marked by user) these 4 points are: [x, y] = [0, h/2], [w, h/2], [w/2, 0], [w/2, h] (see red points in Fig. 11). User can select different 4 points if he wants them to be located further from image borders. For example, let's say that the image has the size of 1024x1024 pixels. If no points are marked on the image then the 4 points used for tilt calculation will be: [0, 512], [1024, 512], [512, 0], [512, 1024]. On the other hand, if the user marks the following points [34, 509], [999, 498], [522, 24], [510, 1001], then the actual points used for tilt calculation will be: [34, 512], [999, 512], [512, 24], [512, 1001] (see orange points in Fig. 11). In other words, those 4 points will always lay somewhere on the x, y axes, which intersect in the middle of the image. Alternatively user can mark only two point (top-left and bottom-right) to specify a subregion for calculation and removal of phase tilt. In such case the 4 points which are used for calculations will lay in the middle of each side of that area (see the yellow rectangle marked in Fig. 11).

**Get sideband** - user can manually specify the center of the sideband, which is to be filtered, by entering its x, y coordinates in the two textboxes adjacent to the button. x, y values should correspond to the coordinates on the real image, not on the displayed image which is always 768x768. In other words, if the image is 1024x1024 pixels, then x, y values should be from (0, 1023) range. As a result the circular area around the specified location will be cropped and centered, and the aperture edges will be smoothed out in the similar manner as in the case of Holo sband function. Image with the cropped sideband is then added to the queue.

**Subpixel shift (in dev.)** - (not recommended) this option is experimental and allows for finding center of sideband with the subpixel accuracy (using center of mass approach). Unfortunately, it produces strange image-tearing artifacts when the fractional part of one of the sideband

coordinates is close to 0.5.

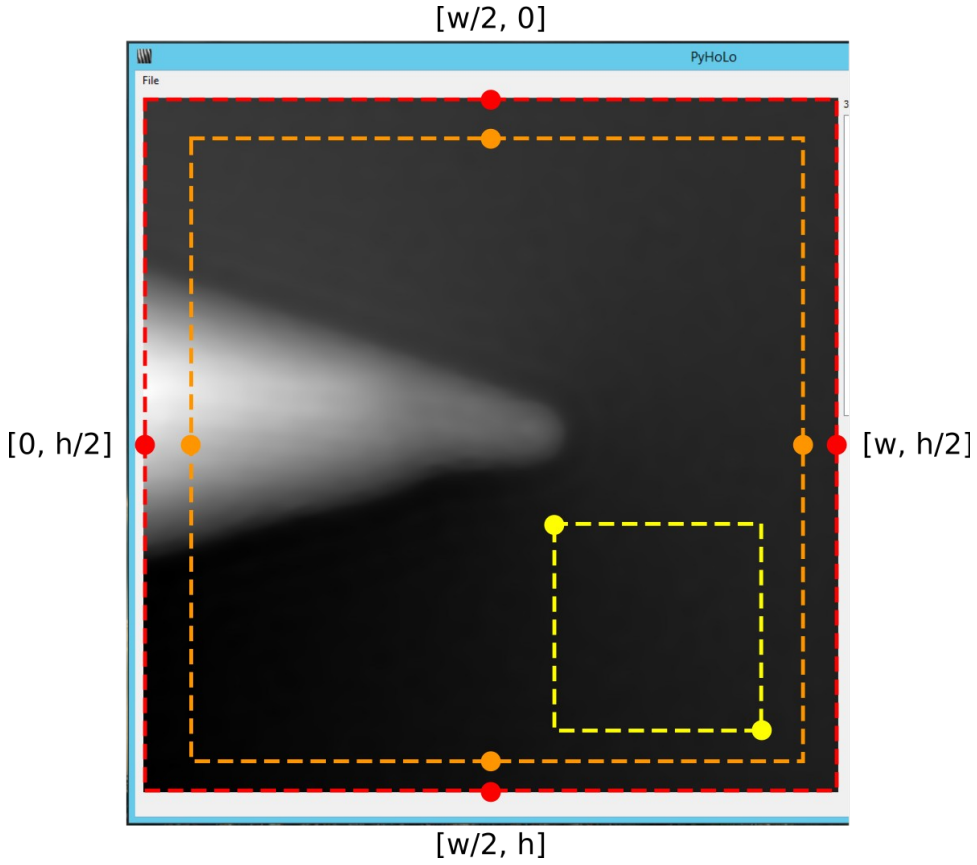



Fig. 11. Three possibilities of removing phase tilt from phase image. Red points in the image are default pixels taken for phase tilt removal, i.e. if no points are marked by user. User can mark four arbitrary points instead (orange points in the image) or two points (yellow points forming a rectangle in the image) if he wants to take into account some specific area.

## 2.5. Magnetic field tab #1

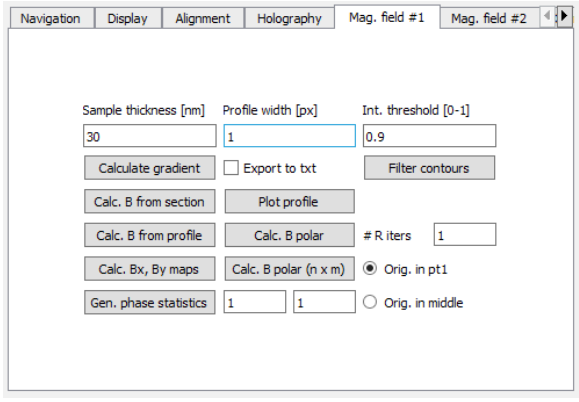



Fig. 12. Magnetic field tab #1

This tab allows for plotting intensity/phase profiles, calculating magnetic field B values based on magnetic phase images and generating polar plots of magnetic field B.

**Plot profile** - display the profile of intensity corresponding to the line marked on the image by user. Profile intensity is displayed in the Plotting area of PyHoLo window (see section 3). If **Export to txt** option is checked then, at the same time, the displayed profile will be saved to the two-column text file (distance [nm] vs. intensity [a.u.] or phase [rad]), in the default output directory.

**Profile width** - integration width of the intensity profile given in pixels (see Fig. 13). If its value is set to be > 1 then each point of the intensity profile will be calculated as a sum of pixels placed on

a line perpendicular to the marked profile at a distance from the range (-d/2, +d/2) from that profile (d is a given profile width).

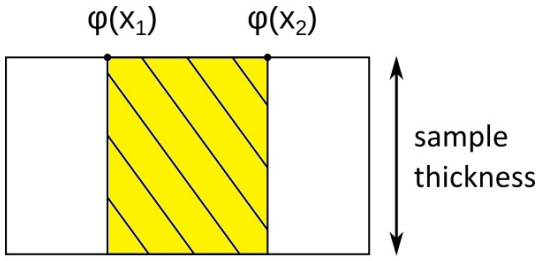
**Calculate gradient** - calculate directional x,y-gradients and total gradient of the current phase image and add the three resulting images to the queue.

**Sample thickness** - input for sample thickness (in nm) in the area of interest. Its value is necessary to calculate magnetic field B in this area.

**Calc. B from section** - calculate magnetic field B based on the section (two points) marked on the image. The calculation of B follows the formula:

(Eq. 1)
$$\left|\vec{B}_{\perp}\right| = \frac{\hbar}{et}\frac{\partial\varphi_m}{\partial x} \approx \frac{\hbar}{e}\frac{\Delta\varphi_m}{t\Delta x} = \frac{\hbar}{e}\frac{(\varphi_2 - \varphi_1)}{t\Delta x}$$

where t is a sample thickness and $\frac{\partial \varphi_m}{\partial x}$ is a gradient of magnetic phase across the selected section. The calculated B value corresponds to the magnetic flux passing through the rectangular surface, which is defined by two points marked on the phase image and by the sample thickness (see the image below). The result (in mT units) is displayed in the interpreter window.




$\Delta\varphi_m = \varphi(x_2) - \varphi(x_1)$

**Calc. B from profile** - calculate magnetic field B based on the phase profile displayed in the plotting area and the information about sample thickness in the area of interest. To use this function you will need to mark a straight line on the magnetic phase image (the procedure of getting such image will be described later in this manual) and plot the phase shift profile across this line. Next, you will have to select two points on the phase profile by clicking on a graph with your mouse (after each click a red circle should appear on the plot). Those two points will determine a section of phase profile on which the phase gradient is to be calculated. Thus you should select a section where phase change is roughly constant (i.e. the phase gradient is constant). Only x-coordinates of selected points are used by program; y-values (phase shifts) are derived from the phase profile automatically) so the points don't necessarily have to coincide with the plot. After you do all of the above steps and click the Calc. B from profile button the magnetic field value (in mT units) will be displayed in the interpreter window.

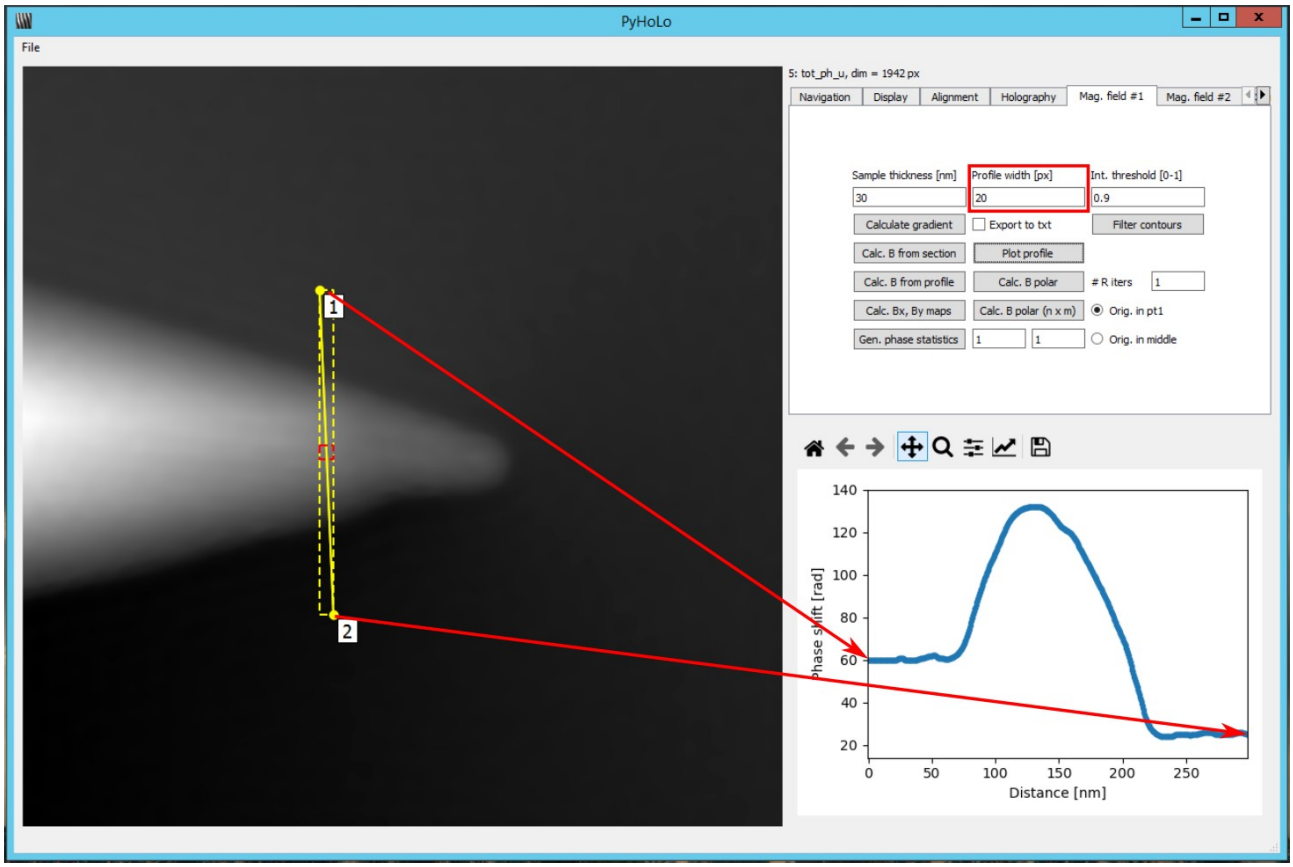



Fig. 13. Plotting the 20-pixel-wide profile of phase shift across the yellow line marked on image.

**Calc. Bx, By maps** - calculate maps of magnetic field B in two directions, x and y, based on currently displayed image of magnetic phase shift. Each pixel value is calculated based on two adjacent pixels (distance between pixels is constant and equal to the pixel size) and their phase values. Sample thickness is taken from the appropriate input textbox. In other words Bx, By maps are just gradient maps multiplied by specific coefficient.

**Gen. phase statistics** - generate phase statistics (min/max values, average value, standard deviation) for the currently displayed phase image. All values are displayed in the terminal window.

**Filter contours** - filter image features with intensity greater than the given fraction (Int. threshold) of maximum intensity.
**Int. threshold** - value from the (0, 1) range (where maximum intensity in the image corresponds to 1.0), below which all pixels will be blackened when you use Filter contours function.

**Calc. B polar** - generate polar plot of B values in different radial directions. First you have to mark two points (a section) on the magnetic phase image. These points will correspond to the $\varphi_1$ and $\varphi_2$ values. Additionally you have to specify if the rotation center of this section will be in the $\varphi_1$-point (i.e. first selected point) or in the middle of the section. For this you have to select one of the **Orig. in pt1/Orig. in middle** radiobuttons. Moreover you can define the number of iterations for increasing the radius, i.e. the distance between rotation center and the $\varphi_2$-point (**#R input textbox**). For example if the distance between two selected points is 24 pixels, and you specify 3

#R iterations, then you will be able to generate single chart with three polar plots corresponding to 24, 48(=2*24) and 72(=3*24) pixels between rotation center and the $\varphi_2$-point.

## 2.6. Magnetic field tab #2

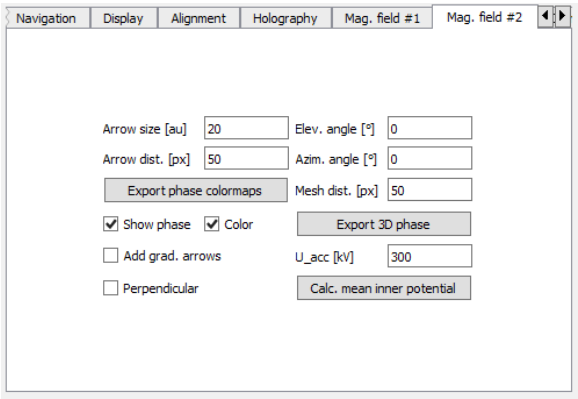



Fig. 14. Magnetic field tab #2

This tab allows for generating series of phase colormaps with superimposed gradient arrows (or arrows perpendicular to the gradient direction), as well as for generating 3d phase maps.

**Export phase colormaps** - export all phase images in the image queue in colorscale or grayscale (check or uncheck **Color** checkbox) with superimposed gradient arrows (check **Add grad. arrows** checkbox to draw arrows parallel to the direction of highest gradient; additionally you can check **Perpendicular** checkbox if you want the arrows to be perpendicular to the highest gradient direction, i.e. if you want the arrows to be parallel to the direction of magnetic force lines). You can modify the size of arrows and distances between them by changing the values given in **Arrow size** and **Arrow dist** input textboxes, respectively. If you uncheck the **Show phase** checkbox, blank images with superimposed arrows will be exported. All phase colormaps will be exported to the 'output' directory as png files. They won't be displayed in the PyHoLo display window. Last phase colormap in the queue will have the (global) colorbar scale attached, corresponding to all images.

**Export 3D phase** - generate and export 3D plot of the currently displayed phase image in the 3D projection given by elevation angle (**Elev. angle**) and azimuthal angle (**Azim. angle**). Specify smaller value of **Mesh dist.** to generate more detailed 3D plot (with smaller step). The 3D phase image will be exported as png file directly to the 'output' directory, so it won't be displayed in the PyHoLo display window.

**Calc. mean inner potential** – use the currently displayed electric phase map to generate and display map of mean inner potential for the given accelerating voltage of TEM in kV (**U_acc**). Pixel values of the resulting map will be given in Volt units.

## 2.7. Corrections tab

Here you can change brightness, contrast and gamma of the currently displayed image by adjusting the corresponding sliders or entering values into the corresponding textboxes.

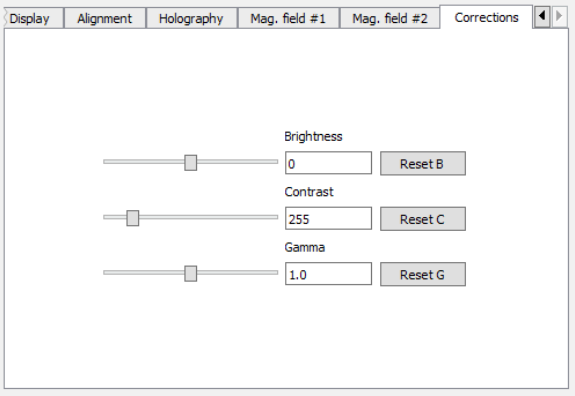



Fig. 15. Corrections tab

## 3. Plotting area

For more information about plotting profiles see the descriptions of **Plot profile** button and **Profile width** input in the section regarding Magnetic field tab #1.

The most useful functions for operating on plots are listed below:

– reset the plot to the original state.
– move the plot across the graph area with your mouse.
– zoom a section of the plot.
– modify axis parameters and labels, choose color and style of the plot.
– save the plot on a disk.