Protomo Tilt Series Tutorial

version 3.1

Hanspeter Winkler

Contents

Revision 0 – 2022-10-31

1 Getting started

1.1 Sample data set

This tutorial uses a small sample data set of insect flight muscle (IFM). The specimen is a section through a sarcomere that was stained and plastic embedded. The section shows a myac layer with alternating thick filaments (myosin) and thin filaments (actin) that run vertical in the images. The dual-axis tilt series was acquired on a CM300 electron microscope at a magnification of 19500, with the Saxton scheme. The initial angular increment at the untilted state is 4° and the maximal tilt angles are ±61°. There are 39 images in this tilt series. After collecting the first tilt series, the specimen grid was rotated by roughly 90° and a second tilt series with an additional 39 images was recorded. To demonstrate the single-axis tilt series alignment, we use the first tilt series. The procedure for dual-axis alignment is essentially the same as for single-axis alignment. The additional steps needed for dual-axis alignment are described in the second part of this tutorial.

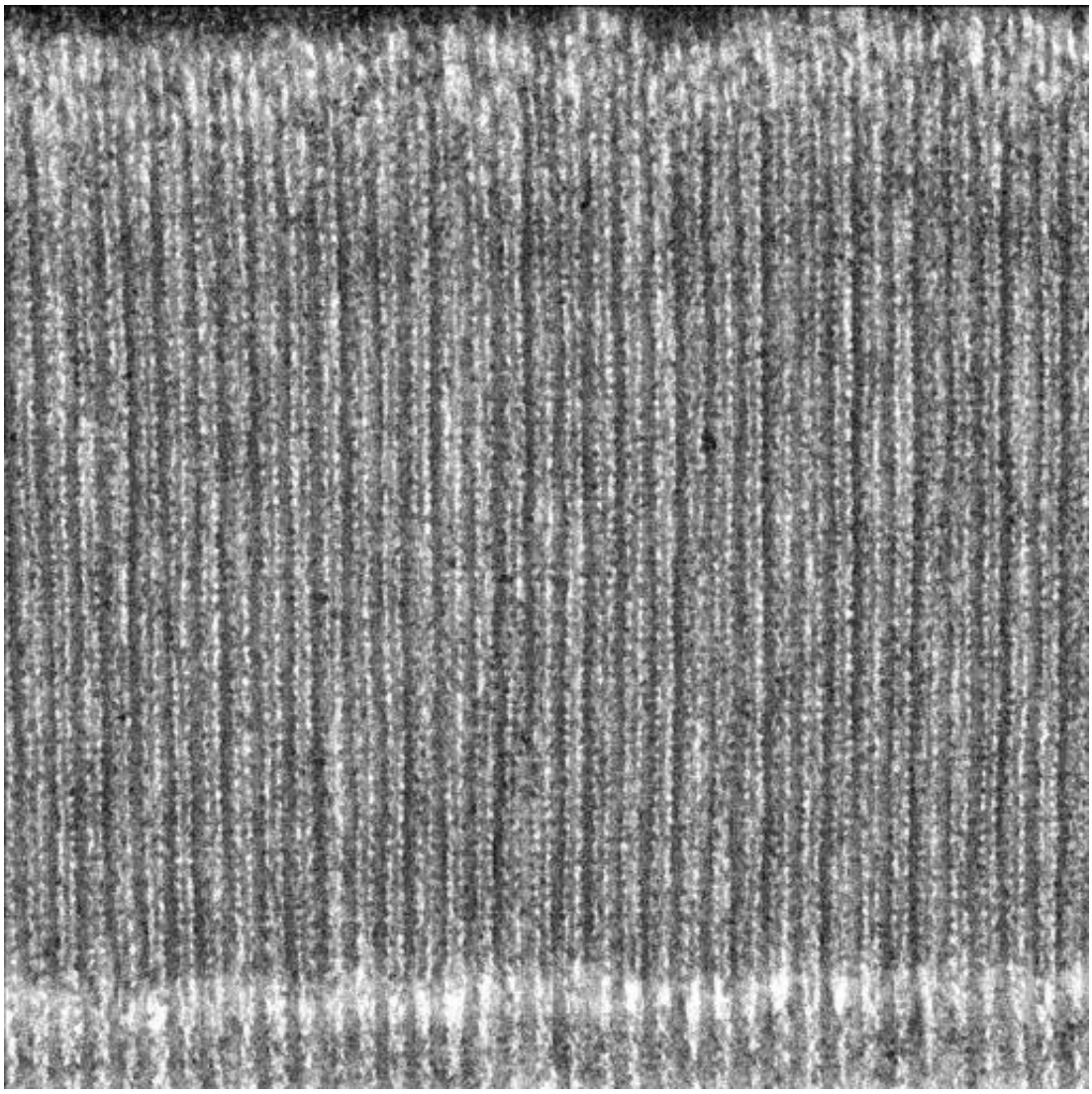

Figure 1: *Untilted view of the myac layer*

1.2 Initial setup

Make sure that the latest version of the *protomo* software is installed. Unpack the tutorial data by typing the following command at the shell prompt:

```
tar -xjf protomo-tutorial-3.x.tar.bz2
```

where `3.x` is the current release number. This command will create a new subdirectory in the current working directory, and in that subdirectory you will find all the required files and data. The raw input data is stored in the subdirectory "`raw-x`" for the first tilt series, and in "`raw-y`" for the second one of the dual-axis tilt series. The single-axis alignment will only use the files in "`raw-x`". Some of the results produced by this tutorial can be found in the "`results`" subdirectory. First we set the directory of the single-axis tilt series as our working directory, and we create two subdirectories in it:

```
cd protomo-tutorial-3.x/singleaxis
mkdir cache out
```

The "`cache`" directory is used by the software to store intermediate files and should preferably reside on a local filesystem, and not on a networked filesystem, for efficient disk input/output. Files with the suffix "`.i3c`" are cache files and can be deleted anytime in the course of alignment. They will be automatically regenerated if they are needed again. The "`out`" directory receives output files for diagnostic purposes. These files are not needed in subsequent alignment steps and can also be deleted anytime.

2 Parameter files

2.1 Tilt geometry

The tilt geometry is specified in a formatted text file. The file name of the text file usually has the suffix "`.tlt`". The file could be created with a text editor, but since it requires a specific formatting, it is easier to use a shell script that takes a simple list as input. The list is stored in the file "`max.dat`" and contains one line of data for each tilt, i. e. the names of the raw image files, the tilt angles, the common origin coordinates, and an in-plane rotation. A complete description of the format and contents of the list file and the geometry file can be found in the *protomo user's guide*.

The tilt geometry file is generated with the list file as input with the command:

```
tomomaketlt.sh max.dat max -80.953 0 101 >max.tlt
```

where the first parameter is the file name of the list, the second one an identification string (a name for the tilt series), and the following ones the tilt axis azimuth (-80.935), the specimen orientation angle (0), and the start number (101) for numbering the images in the tilt series. We use 101 as the start number here, and later, for the dual-axis tilt series, we choose 101 for the first, and 201 for the second series. Finally, the output of this script is redirected to a file named "`max.tlt`".

2.2 Processing parameters

The files that specify all the necessary parameters for processing a data set are also plain text files. In such a parameter file, the relevant parameters for tilt series alignment are contained in a section called "`tiltseries`" (see for instance the file "`max.param`"). The format of this file and the meaning of the parameters are described in detail in the *protomo user's guide*. In our tutorial, all parameters were assigned reasonable values, so that the tutorial will work "out of the box". The file name prefix, "`max`" in our case, identifies a particular tilt series and must not be changed during processing. The additional parameter files in this tutorial with cycle numbers appended contain changes needed for the later alignment cycles and will be copied to the file "`max.param`" and not used directly.

3 Alignment

3.1 General considerations

Preprocessing the raw image data and applying filters are important aspects of tilt series alignment. Preprocessing removes linear density gradients and density outliers ("hot pixels"), that may adversely affect the cross-correlation alignment. Typically, band pass filters are applied to the cross-correlation function, and the high pass and low pass filter limits should be set sensibly. The effect that the filters have on the raw images can be examined visually with the commands described in the following sections.

Filter limits are specified relative to the size of the resampled area. The units are reciprocal real space pixels of the resampled image, so a filter diameter of 1.0 for a low-pass filter would filter out the signal at and beyond the Nyquist frequency. Note, that the filter parameters are specified as the dimensions of a mask (diameter) that is applied to the Fourier transform, and not as the spatial frequency cutoff. The Nyquist frequency cutoff has always a value of 0.5 in this scheme (mask diameter 1.0) regardless of image size. Filter limits are thus specified as a multiplier or fraction of the Nyquist frequency, and not in terms of Fourier space pixels. The reason for this is that the spatial frequency corresponding to a Fourier space pixel depends on the real space image size in the discrete Fourier transform (DFT).

The values of the filter limits should still be set with the real space sampling in mind, though. For instance, if we want to filter at a particular spatial frequency relative to the sampling of the original data, and we change the sampling or binning factor, we also need to change the filter limits accordingly. One way to do this is to multiply the filter limits by the sampling factor. If the product exceeds the value of 1.0, no filter is being applied, and it would mean that we attempted to filter beyond the theoretical resolution limit.

A low-pass filter eliminates the higher spatial frequencies which tend to be dominated by noise contributions. The filter limit should be chosen with the expected resolution of specimen features in mind. The choice of an appropriate high pass filter is more critical. Generally, the high pass filter should be set in such a way as to enhance specimen features of interest by suppressing low spatial frequencies that contribute little to the visibility of such features, and to eliminate nonlinear background density variations that cannot be corrected by the preprocessing procedure.

3.2 Coarse alignment

The images in the tilt series can be aligned manually if needed, for example if the image acquisition software does not keep the region of interest centered in the field of view. The resulting large shifts may then cause the automatic coarse alignment procedure to fail. With the interactive graphical user interface we can correct the alignment of the affected images easily. The command to start the interface is listed below:

```
tomoalign-gui -log -tlt max.tlt max.param
```

Since this is the very first time we read the raw images, and we have selected preprocessing options in the parameter file, a cache file with preprocessed images is first created ("`cache/max_pre.i3c`") which may take some time to execute. Since we also have selected binning with a sampling factor of 2, a second cache file with the binned images is created ("`cache/max_pre_smp2.i3c`") using the preprocessed data. In all subsequent operations, data is directly read from the cache files, unless they have been deleted, in which case they are regenerated automatically.

The alignment parameters (transformation matrices and origins) are stored in a separate binary file ("`max.i3t`") which is also created automatically if it does not exist. If `tomoalign-gui` is run a second time, this file is read instead of "`max.tlt`", and the option "`-tlt`" is not needed anymore.

If we do not want to do an alignment initially and use the default origins in the geometry file, we can also initialize the tilt series directly with the command line program "`tomoinit`", which takes the same arguments as "`tomoalign-gui`":

```
tomoinit -tlt max.tlt max.param
```

In the graphical interface we first increase the area to be aligned by zooming out. Select "`View` → `zoom out`" on the top menu bar. For manual alignment, left-click on the image and drag it while keeping the mouse button down. The reference image is displayed in red color, and the image that is aligned in green color. If aligned, the addition of the two colors produces a grey-level image. A single image can also be aligned automatically by pressing the "`align`" button. The "`reset`" button returns to the initial state for the displayed image.

In our case, the tilt series is relatively well aligned, so we can use the automated function for translational alignment of the whole tilt series. We start the alignment process by choosing "`Actions` → `align all`". When the alignment has finished (the "stop" button, to the right of the "reset" button, changes to "align" again), we notice that the peripheral parts of the images do not match the reference as well as the central parts, i. e. the central part appears grey, but the periphery still shows red or green colored specimen features. This is because overall alignment cannot be achieved by a simple translation and we need the area matching technique described in the following section.

We can check the alignment quality with an animated display. First, switch to single image display with the menu option "`View` → image", then choose "`Actions` → show movie". Smooth transitions between images indicate good alignment, skips indicate misalignment. We now close the graphical interface and save the result. It is advised to make a backup copy of the alignments now, if you want to start again from this point without repeating the previous steps:

```
cp -p max.i3t maxsaved.i3t
```

3.3 Alignment by area matching

3.3.1 Setting up the alignment

The following tasks are carried out with the program "`protomo`" which is started at the shell prompt. When the protomo command prompt "`tomo>`" appears, we first set the "`log`" option, then read the parameter file and open the tilt series which makes the geometry information in the previously created binary file "`max.i3t`" available for processing. A file name need not be specified for the binary file, since the name of the file will be derived from the prefix of the parameter file name ("`max`"). This prefix will also be used subsequently as default when file names are generated for output files.

```
protomo
tomo> set log
tomo> paramfile max.param
tomo> open
```

The "`log`" option enables printing of diagnostic messages for subsequent commands. The "`paramfile`" command reads the parameter file, while the "`open`" command retrieves the geometric parameters from the file "`max.i3t`", which keeps track of the alignment parameters. If we quit a session and leave this file unchanged, we can resume the alignment in a new session at the point where we left off.

Before we start the alignment we want to check whether the mask and filter parameters have reasonable values. For this purpose we extract a window from the untilted image (number 120) with the parameters loaded previously and display it. The "`image`" command writes the window to an image file first that is displayed with the "`show`" command:

```
tomo> image 120
tomo> show
```

The transform of this image is generated and displayed as follows:

```
tomo> transform 120
tomo> show
```

We need to adjust the contrast so that we can see the applied band-pass filter by clicking on the menu items "Image → Histogram". In the histogram window, we decrease the upper limit until it nearly reaches the lower limit. In a similar way we can inspect the filtered image:

```
tomo> filter 120
tomo> show
```

3.3.2 The first alignment cycle

We are now ready to start the alignment:

```
tomo> align
```

The "`align`" command calculates and stores a correction to the initially supplied geometry for each image by matching equivalent image areas in the tilt series. Since we have selected the "`log`"

option, a line with translational shifts, correction values and the cross-correlation coefficient is written to the terminal for each image.

The alignment terminates prematurely at image 109 and 131, because the specified maximal correction of 4% in the parameter file (`maxcorrection: 0.04`) is exceeded for these images. In an interactive session we can display a plot of the correction factors for the aligned images as follows:

```
tomo> plot
```

Another option is to write the factors to a file for subsequent plotting with the command "`corr`". If no file name is specified, the name is generated from the output directory name specified in the parameters and the tilt series prefix by appending the cycle number, which is "00" for the first cycle. The following two commands are thus equivalent:

```
tomo> corr
tomo> corr to out/max_00_cor.dat
```

The output is a simple text file containing numbers that can be plotted with the program of your choice. In a separate shell you could produce the plots in postscript format with the shell script `tomoplot.sh` if your working directory is `protomo-tutorial-3.x`.y:

```
tomoplot.sh out/max_00_cor.dat
```

The postscript file with the correction factors generated by the script is called "`max_00_cor_cof.ps`" and is located in the subdirectory "`out`". Two additional files, one with a plot of the direction of stretching/compression indicated by the correction factors (`max_00_cor_coa.ps`), and another one with the in-plane rotations (`max_00_cor_rot.ps`) are also produced. The angles are measured anti-clockwise from the x-axis, in degrees.

We also requested the output of correlation peak images of size 128×128 pixels in the parameter file. These were also written into the "`out`" subdirectory as a 3D stack of images (file "`max00_cor.img`"). The z-coordinate of each section corresponds to the sequence number of the image. The sequence number was the number enclosed in square brackets that appeared in the terminal log output during alignment. It always starts at 0 and is not necessarily the same as the image number. Note, that there is no correlation peak for image 120, sequence number [19], because this is the reference image which does not need to be aligned. Also, there are no peaks in the sections that correspond to images that have not been aligned yet due to the premature termination. We can display the correlation peaks after reading the file (scroll in the z-direction with the up- and down-arrow keys when the display window pops up):

```
tomo> show out/max_00_cor.img
```

To prepare for the next cycle, we re-evaluate the tilt geometry using the corrections obtained by area matching. The align command computed a shift vector and four correction parameters (a 2×2 matrix) for each image which were recorded internally in the file "`max.i3t`". From these data, new geometric parameters are calculated by a least squares fit:

```
tomo> fit
```

The fitting could be executed multiple times with different parameters, if so desired. The results are stored in a temporary location and are overwritten each time the "`fit`" command is executed. The command prints a list of the geometric parameters to the terminal if logging is enabled. For

each parameter, the number in the left column is the new value, the number in the right column the change from the previous value. The left and right columns have the same values this time, except for the tilt azimuth, because we initialized the parameters with zero. The tilt azimuth changed by more than 1°. Generally the initial estimate of the azimuth need not be very accurate; in many cases the first alignment can tolerate deviations as large as 10° for the procedure to still find the correct value.

The results of the most recent fit will only be stored permanently if updated, otherwise they are lost. The following command replaces the current geometry the with the new, fitted results and saves them in the binary file:

```
tomo> update
```

3.3.3 More alignment cycles

We use the same parameters for the next few cycles, write the correction data to a file for later examination, and display a plot thereof:

```
tomo> align
tomo> corr
tomo> plot
```

The second cycle completes successfully for all images and the deviations of the correction factors are an order of magnitude better, when plotted. Ideally, these factors should all be 1. We recompute the geometry again:

```
tomo> fit
```

The tilt azimuth changed by a fraction of a degree only in this cycle. The change of the in-plane rotations is insignificant for the images that were aligned before the cycle terminated in the previous run. We make these changes permanent with the following command:

```
tomo> update
```

and run another cycle:

```
tomo> align
tomo> corr
tomo> fit
tomo> update
```

There was no substantial improvement of the corrections, so we want to compute a preliminary back-projection map to check the thickness of the specimen:

```
tomo> map
tomo> show
```

When displaying the map, we find that sections in the range $-14 \leq z \leq 19$ show the most contrast. The estimated thickness would then be 34 pixels at a sampling factor of 2 and we use this number for the next cycle (see parameter file `max03.param`).

3.3.4 Maximal usable area

We are now trying to determine the maximal usable area in the tilt series. The "`area`" command generates a superposition of binary images that indicates the coverage of the aligned specimen area by the raw data. When a raw image is re-interpolated according to the stored geometric parameters and the interpolated pixel lies within the raw image, it is assigned a value of 1, otherwise if the source location falls outside the raw image, it is assigned a value of 0 in the output image. Since corresponding pixels in the interpolated images represent the same specimen points in the projections, the pixel value in the superposition is a count of images that contributed to the particular pixel at a specific specimen point. When the output image is displayed and zoomed out, the white area, or more precisely, the area with the highest density value represents the equivalent projected specimen areas that are common to all members of the tilt series.

```
tomo> area to maxarea.img
tomo> show
```

We can see in the resulting image that the individual projected regions overlap better vertically than horizontally which suggests a horizontal drift during data collection. The displayed image should be zoomed out and thresholded just below the density 39, which is the number of images in the tilt series (by default, the zoom factor is 1 and only the central white area will be displayed). The terminal output indicates that the maximal area has a size of 942×943 pixels, which is the rectangle that can be fit in the common area. It is slightly off-center by 46 pixels on the x-axis and -44 pixels on the y-axis.[1] Note, that the rectangle size is scaled by the sampling factor, whereas the origin is reported with respect to the unsampled, original images.

3.3.5 Alignment cycle with adjusted area and thickness

We now choose a slightly smaller area than reported which has the size of 936×936 pixels, and we do not correct for the relatively small shift of 23 or -22 pixels (at a sampling factor 2) in the x- or y-direction. The number 936 was chosen, because it is a product of small prime numbers for which the Fourier transform computation is more efficient. The change to the window size and the adjustment of the thickness that we determined earlier by visual inspection of the map have been recorded in the file "`max03.param`".

The new alignment cycle is now carried out with the increased area and the adjusted thickness. We first read the new parameter set from the text file "`max03.param`":

```
tomo> paramfile max03.param
```

The old parameters will be replaced with the new ones by the "`paramfile`" command. The following commands are the same as in the earlier cycles:

```
tomo> align
tomo> corr
tomo> fit
tomo> update
```

[1] The reported numbers may vary slightly depending on the versions of third-party libraries used.

3.3.6 Final cycle

We set the sampling factor to 1, increase the window size to 1800×1800, and double the thickness parameter for the final alignment at full resolution. Again, the new parameters are read from a text file first:

```
tomo> paramfile max04.param
tomo> align
tomo> corr
tomo> fit
tomo> update
```

The final alignment parameters can be exported to a text file as follows:

```
tomo> geometry
```

3.4 Back-projection map

The x-y size of the final map could now be set to the same values of the window size used in the final alignment cycle (1800×1800). We reduce it for this tutorial to a smaller size to speed up the map computation:

```
tomo> param map.size { 512 256 96 }
tomo> map
```

Since no file name was provided in the "`map`" command, the map is written to the default location "`out/max05_bck.img`".

3.5 Batch processing

The batch processing scripts are designed to run alignments of multiple tilt series from the same specimen. The scripts assume a particular directory structure. All input and output files are referenced with respect to what we call the top directory. In our case it has the name "`batch`":

```
cd protomo-tutorial-3.x/batch
```

Each tilt series is given an identifier, and subdirectories will be created in the top directory with that identifier as their name. All results and intermediate files are stored separately for each tilt series in the subdirectories. The scripts for batch processing must be run with the top directory set as working directory. Generally, no terminal output is generated by the scripts. Diagnostic output and error messages are written to log files. If the log files remain in the top directory, it means that the script did not complete successfully.

Assuming that we already created the initial geometry file earlier in this tutorial, we copy it and give it a different name (`single_initial.tlt`). "`single`" is our tilt series identifier, and the suffix "`_initial.tlt`" is required by the scripts.

```
cp ../singleaxis/max.tlt ./single_initial.tlt
```

The parameter file "`initial.param`" is a modified copy of the file "`max.param`" in which the parameter "`pathlist`" has been changed to reflect the different directory structure. Since the processing scripts are executed in the subdirectory "`batch/single`", the relative path specification has to be adapted to point to the correct directory with the raw image files. With these changes made, we initialize the alignment for our tilt series:

```
tomoinitialize.sh single initial.param .
```

Note the dot as the last argument of the script. This is the directory where the geometry files are located, in our case it is the current working directory which is designated with the dot. Results are stored in the newly created directory "`single`". While the script is running, diagnostic messages and error messages are directed to a log file in the top directory. If the script terminates abnormally, the log file is left where it was created in the top directory, otherwise it is moved to the directory "`single/history`". No terminal output is generated during execution.

After initialization, we run a coarse alignment with six iterations of a simple translational alignment followed by four iterations of area matching:

```
tomoaligninitial.sh single 6 4
```

A parameter file is not needed for this step. The parameters used are those specified during initialization. Among the usual output, the script produces a preliminary map at the end of the procedure which can be examined to determine the actual thickness. As before, we find the same range of $-14 \leq z \leq 19$ with substantial contrast, and we set the thickness accordingly in the new parameter file "`cycle-02.param`". We run two iterations of area matching and also update the window size which has been recalculated automatically in the previously completed cycle to use the maximal available area:

```
tomoalign.sh single cycle-02.param 2 update
```

This script requires a parameter file name (`cycle-02.param`) in addition to the tilt series identifier (`single`), and the number of iterations (`2`) to execute. The keyword "`update`" indicates that the automatically estimated values for the maximal usable area are to be used for the window size instead of the values from the parameter file.

A map is computed with the following command:

```
tomomap.sh single cycle-02.param size 512 256 96
```

The size option overrides the map size specified in the parameter file. The map can be found in the subdirectory "out" and can be displayed with the command:

```
i3display single/out/single_03_smp2.img
```

We run an additional cycle with two iterations of area matching after changing the sampling factor to 1 and the window size to 1800×1800 in the parameter file "`cycle-03.param`":

```
tomoalign.sh single cycle-03.param 2
```

The final map is computed with the same command as above but with a different parameter file:

```
tomomap.sh single cycle-03.param size 512 256 96
```

4 Dual-axis alignment

We first set the working directory for dual-axis alignment and create the required subdirectories:

```
cd protomo-tutorial-3.x/dualaxis
mkdir cache out
```

The tilt geometry files are generated from two input data files (`max.dat` and `may.dat`) with the commands

```
tomomaketlt.sh max.dat max -80.953 0 101 >max.tlt
tomomaketlt.sh may.dat may -80.953 -90 201 >may.tlt
```

where the command line arguments are the file name prefixes and the starting numbers for the images. Images of the first series are numbered from 101 to 139, and for the second series from 201 to 239 with these settings. The scripts create two output files (`max.tlt` and `may`.tlt), which we combine later into a single file.

The next step is to determine the overall orientation of the second tilt series relative to the first one. We need to define all geometric parameters relative to the same coordinate system that is fixed with respect to the specimen structure. Since the specimen was rotated by about 90° before the second tilt series was recorded, we expect the two images at 0° tilt to be rotated by the same amount. This is reflected in the geometry file by a difference in the two orientation angles PHI of roughly 90° (the grid or specimen rotation). Note, that the tilt azimuths of the two tilt series are the same, since they are defined in a coordinate system fixed with respect to the microscope.

While we could create a combined geometry file simply with a text editor if the rotation angle was known, we run a script here to measure the relative orientation, and to modify and combine the geometry files created by `tomomaketlt.sh`. The script calls `tomoalign-gui` to manually align the two zero degree tilts and takes the file name prefixes of input and output files as arguments:

```
tomodualorient.sh max may maxy
```

It requires a new parameter file named `maxy_zero.param` to run. We adapt the single-axis parameter file by adding the path of the raw images of the second tilt series, and inserting an additional parameter "`startangle`". In the graphical interface it is best to zoom out the image, so that we get a better overview, then manually rotate the green overlay image until the filaments are vertical, then switch to "translate". Clicking the "align" button should then shift the overlay down and to the right for a considerable amount. From here we can fine-tune the alignment by switching between translational and rotational alignment until the superposition of the images assumes a grey tone over the entire overlapping region. After saving and quitting the interface a new geometry and parameter file is created. In the geometry file `maxy.tlt`, the orientation angle and the origins have been adjusted. In the parameter file `maxy.param`, which is a copy of `maxy_zero.param`, the 0° tilt of the second series has been excluded in the combined tilt series.

Coarse alignment of the dual-axis tilt series must be carried out separately for the time being, because the program `tomoalign-gui` cannot handle the combined tilt series yet. After initializing, we choose "`Actions` → `align all`" in the graphical interface, and repeat the alignment once when the first pass has finished, for both parts of the tilt series separately:

```
tomoinit -tlt maxy.tlt maxy.param
tomoalign-gui -log -a 0 maxy.param
```

```
tomoalign-gui -log -a 1 maxy.param
cp -p maxy.i3t maxysaved.i3t
```

If all parameters were specified correctly, the actin and myosin filaments in the alignment display are oriented vertically in both `tomoalign-gui` invocations. The first part shows shifts of about 20 pixels in the first alignment pass. The second part is not as well in register as the first part, probably due to less accurate tracking in the microscope, and the maximal shifts at the higher tilt angles are more than 100 pixels.

From here on the alignment process is essentially the same as for single-axis alignment. The only change that needs to be made in the parameter file is the addition of the "`startangle`" option, which selects an alignment mode that alternates between the two parts of the tilt series and aligns the images strictly in the order of increasing magnitude of the tilt angle.

The following commands have been used to produce the results of the first two cycles of dual-axis alignment in this tutorial:

```
protomo
tomo> set log
tomo> paramfile maxy.param
tomo> open

tomo> align
tomo> corr
tomo> fit
tomo> update

tomo> align
tomo> corr
tomo> fit
tomo> update
```

From here on we also refine a scale factor between the two parts of the tilt series to take into account slight magnification differences. The following parameter takes effect when we compute the new geometry:

```
tomo> param fit.scale true
```

We run two more cycles and compute a preliminary map to determine the specimen thickness as described for single-axis alignment:

```
tomo> align
tomo> corr
tomo> fit
tomo> update

tomo> align
tomo> corr
tomo> fit
tomo> update

tomo> map
tomo> show
```

Now we read an updated parameter file in which the specimen thickness has been changed. After one more alignment cycle we determine the maximal usable area as before in the single-axis alignment:

```
tomo> paramfile maxy04.param
tomo> align
tomo> corr
tomo> fit
tomo> update

tomo> area
```

The terminal output indicates that the maximal area is 794×772 pixels, and the center of that area lies at coordinates 230, -176. Since this is a substantial displacement, we shift our region of interest and then increase the window size:

```
tomo> origin 230 -176
tomo> param window.size { 768 768 }
```

Note, that the size is scaled with the sampling factor, so the actual window relative to the raw images has twice the linear dimensions. The origin coordinates are also measured relative to the raw images. To check that the chosen size is small enough to contain all specimen features, we can extract a stack of windows. The windows are stretched to compensate the foreshortening. When scrolling through the stack with the up- and down-arrow keys, the displayed windows should not show any blank regions:

```
tomo> windows
tomo> show
```

We run another alignment cycle with the new size and origin:

```
tomo> align
tomo> corr
tomo> fit
tomo> update
```

For the final cycle we set the sampling factor to 1 and double the window size and start the alignment after reading the new parameters from a file. Then we write the final geometry and compute the final map:

```
tomo> paramfile maxy06.param
tomo> align
tomo> corr
tomo> fit
tomo> update

tomo> geometry
tomo> param map.size { 512 256 96 }
tomo> map
```