CoLoc3D v.1

User manual

Jaroslaw Jacak

Medical Engineering Dept.
Upper Austria University of Applied Sciences
Linz, Austria
jaroslav.jacak@fh-linz.at

2022

CoLoc3D are tools for calibration, registration and co-localization of 3D dual-color single molecule localization microscopy images.

CoLoc3D contains published Matlab functions (see comments in the functions) as well as several own functions and scripts for handling individual menu items. CoLoc3D allows channel calibration, registration and co-localization of 3D dual-color single molecule localization microscopy images. In our system, the input data structure is limited to a specified Matlab format. Here, the dataset is a structure array named "par". The structure contains the following fields: par.pkmatrix – array of numerical data with row (measurements) and columns (data coordinates). par.pkdesc - contains two cells par.pkdesc.desc, and par.pkdesc.units: par.pkdesc.desc gives a description of the pkmatrix-columns line with at least the following string variables: "frame"; "x"; "y"; "z"; "pa"; "paz"; "intensity" where "pa" is the position accuracy of (x,y) localization and "paz" is position accuracy of (z) localization. "frame" is the frame number. par.pkdesc.units contains a description of the units for each column in string format.
To enable analysis of data in another format, the system has an interface to Excel that imports and transfers Excel data to the Matlab format described above. Excel data spreadsheet must be column-oriented and must have at least 3 columns (3D point localizations) with column headers "x", "y", "z".
If possible, the following additional columns contain: position accuracy (in nm) in (x,y) with header "pa", position accuracy (in nm) in (z) with header "paz", intensity of the point with the header "intensity", and the frame number with the "frame" header. In the absence of these columns, its values will be automatically generated with default numbers typical for 3D dSTORM nanoscopic samples.

Installation

Download COLOC3D_V1.zip. Unpack the complete directory COLOC3D_V1. From the *CoLoc3d_code* subdirectory, run the Matlab script **COLOC**. The main menu will open. The subdirectory *CoLoc3d_Example* contains sample results of the analysis.

Main menu

The system starts in Matlab (up to version R2018) with the **COLOC** command from subdirectory *CoLoc3d_code*. Then the main menu opens with buttons for possible processing and analysis options for two 3D single molecule localization samples (point-clouds).

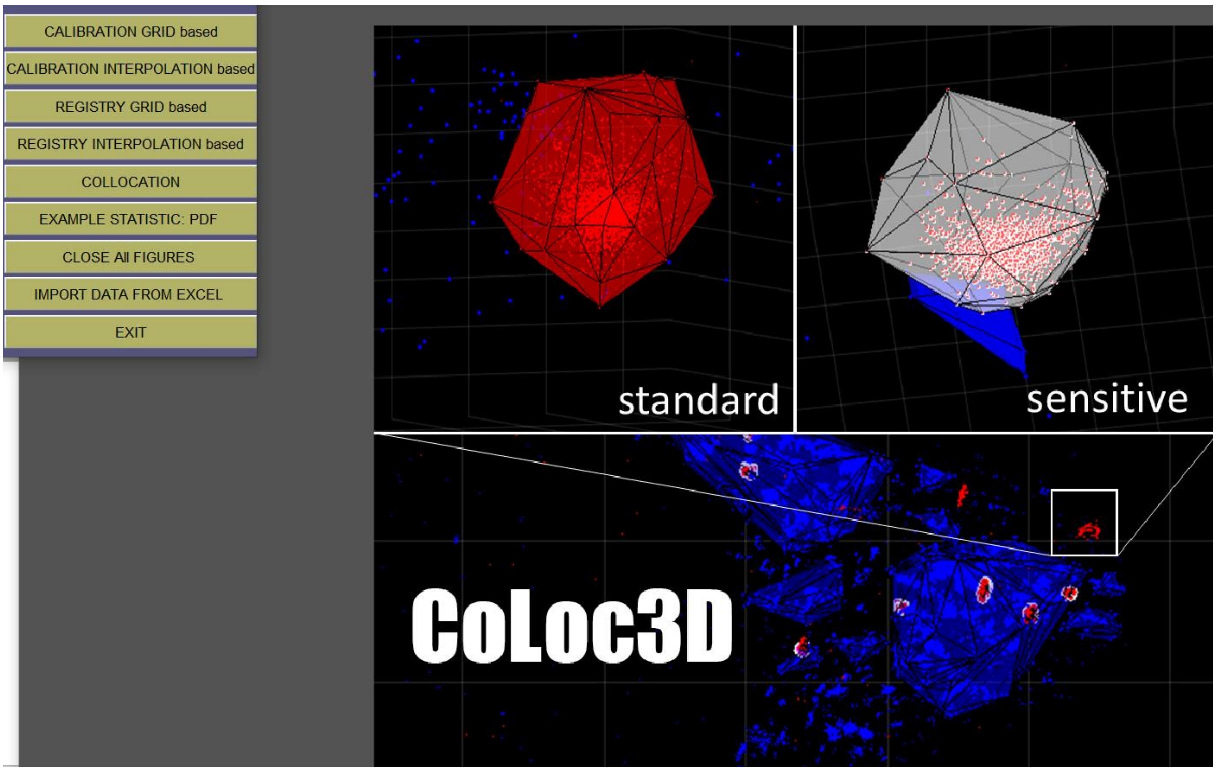


Fig 1. Start page and the main menu of the COLOC -system.

The navigation in the system is very intuitive. Each analysis, triggered by the corresponding button of main menu, is assisted by many additional messages and questions (displayed on the screen) which guide the user during the data processing. After completing the execution of options, the system automatically returns to the main menu.

Button **IMPORT DATA from Excel**

In order to enable analysis of data in another format, the system has an interface to Excel that imports and transfers Excel data to the Matlab format.

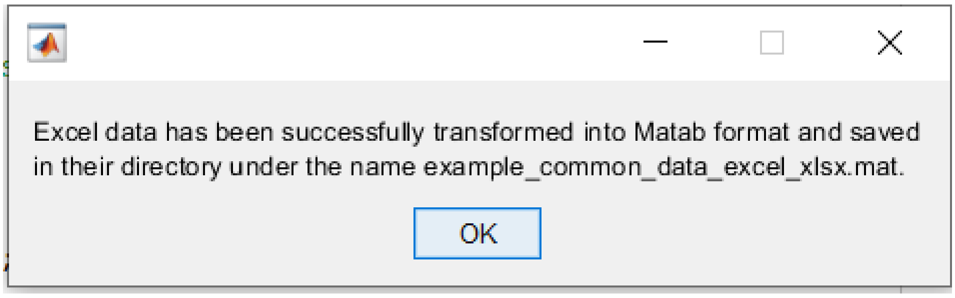


Fig 2. Data transformation from Excel to Matlab is accompanied by a message with the name and prefix “ex” the data is stored.

Excel data spreadsheet must be column-oriented and must have at least 3 columns (3D point localizations) with column headers "x", "y", "z" (for calibration data all "z" can be zeros).

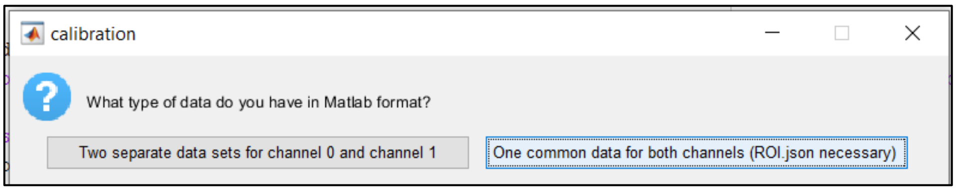
If possible, the following additional columns should contain: position accuracy (in nm) in (x,y) with header "pa", position accuracy (in nm) in (z) with header "paz", intensity of the point with the header "intensity", and the frame number with the "frame" header. The system accepts Excel data without these columns. They will be replaced with default values as necessary.

Button **CAIBRATION GRID based**

Calibration:

The calibration is based on the geometric transformation of the image data of channel 1 (color2) to the data of channel 0 (color1). Images of both channels show fiducial markers, which are necessary to calibrate the system for further experiments. Herein, we use calibration data as described in [DOI: 10.1364/BOE.424016], determined by 3D imaging of multispectral beads.

Typically, the fiducial marker location data of both channels are contained in a one common set of points in Matlab *par* format. To separate the data into two separate channels, it is necessary to define their ROI parameters such as offset, height and length. (Option*: One common data for both channels)*



If the fiducial marker positions in both channels are contained in separate data sets, they can also be loaded with the option: *Two separate data sets for channel 0 and channel 1*

Grid based calibration can be performed using two methods that differ only in the form of the input data:

Fiducial marker points:

Input data: fiducial marker points filtered from images of two channels: channel 0 (color1) called ***fix point-cloud*** and channel 1 (color2) ***mobile point-cloud***.

Since the extracted fiducial marker points of both channels are typically in one set, it is necessary to cut points for each channel separately. This is done with the help of ROI (Region of interest, *roi.json*) parameters.

The points from the ROI of channel 1 (mobile points) will be moved to the points within the ROI of channel 0 (fix points). Brightness threshold and DBscan clustering filter both point clouds. Points that do not belong to any cluster are removed (see Fig.3).

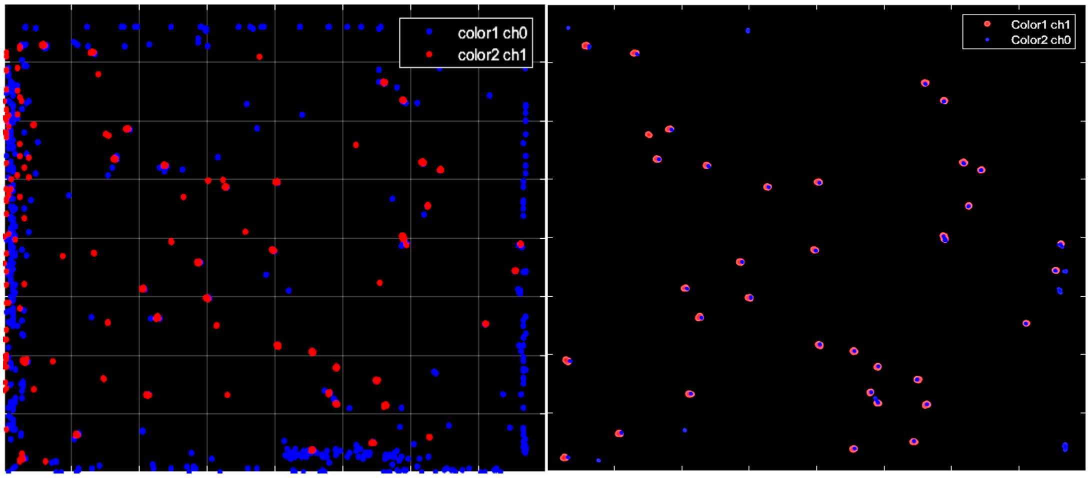


Fig 3. Point Clouds (blue-channel 0; red-channel 1 before (left) and after filtration (right).

Then find the optimal rigid transformation (translation and rotation around the z-axis) of the channel 1 (mobile points) to channel 0 (fix points).

Since the optical system that captures fiducials in both channels is non-linear, the transformation computed for the full images averages the point shift, which results in inaccurate calibration at the periphery of the images.

For this reason, a method has been proposed for dividing both sets of points (corresponding to images from both channels) into sectors (segments) by overlaying a regular n×n grid. This method is a generalization of the field displacement method used in image registration.

For each grid sector, an optimal rigid transformation (translation and rotation about the z-axis) of the sector's mobile points is calculated individually. The optimal transformation for each sector results from **minimizing the sum of minimum distances** of mobile points to fix points.

The results in an n×n displacement vector field representing the transformation for each grid sector (segment) separately. Typically, fiducials are not densely scattered in the images, so increasing the size of the grid can result in a blank sectors without points in one of the clouds. In this case, the transformation for these blank sectors is calculated by averaging the transformations from neighbouring sectors.

The displacement vectors grid is stored and can always be used for registration and calibration of point cloud samples obtained in further experiments. The result will be stored with the date and name of the calibrated sample as *'calibration_day_month_name'*

Figure 4 shows displacement vectors grids of various dimensions and their impact on the calibration accuracy measured as total MSE (mean squared error) after registration of mobile points. Grids of large size show the nonlinearity of image capture by the optical system.

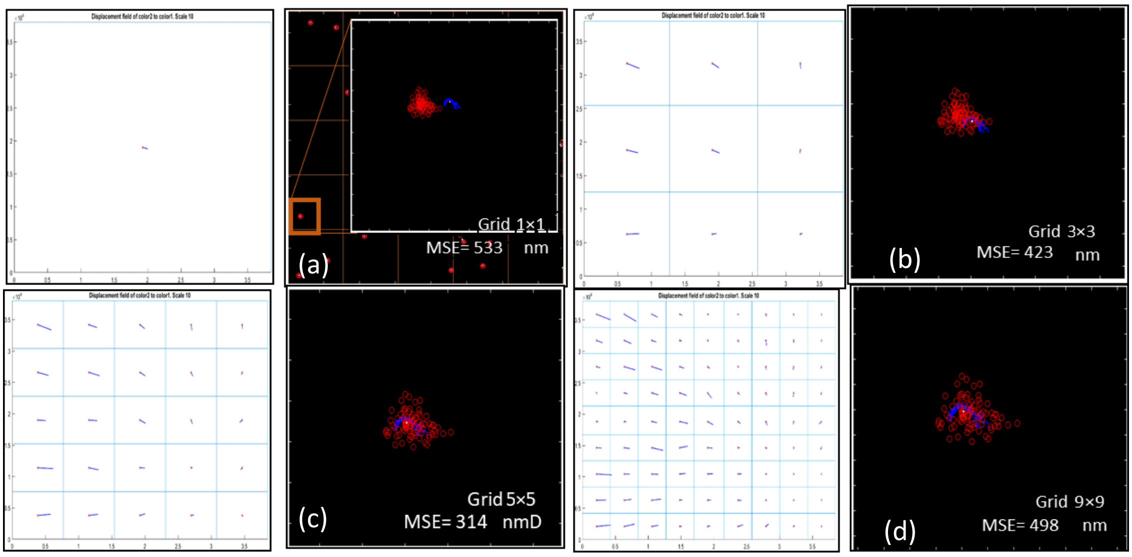


Fig 4: Displacement vectors for different grid-sizes in scale 10 and selected peripheral fiducial marker (blue - color 1 (fix), red - color 2 (mobile) after calibration). (a) Displacement vector for grid 1x1, total Mean Squared Error (MSE) of calibration about 530 nm. (b) Displacement vector field for grid 3x3. MSE of calibration about 423 nm. (c) Displacement vector field for grid 5x5. Mean squared error after calibration about 314 nm. (d) Displacement vector field for grid 9x9. Mean squared error after calibration about 500 nm.

By increasing the grid size, the number of empty sectors increases. This causes the global MSE to increase. The best results are obtained for grid sizes 3 to 7.

Button **CALIBRATION INTERPOLATION based**

The second calibration method takes advantage of the fact that individual image frames can be uniformly assigned values on the z-axis.

After filtering as in the previous method, one can search for each pair of (the same) frames from channel 0 and channel 1 for the optimal 2D transformation of the frame points originating from channel 1 to the channel 0.

The found optimal translations X and Y can be assigned via the frame number to the value Z. In this way, the empirical values of the shift function along the axis X and Y depending on the value of Z are obtained.

The data is interpolated by the smoothing spline method and the resulting functions shift_X(z) and shit_Y(z) (see Fig. 5 and 6) are stored (as *'interpolated_calibration_day_month_name')* for further calibration and registration.

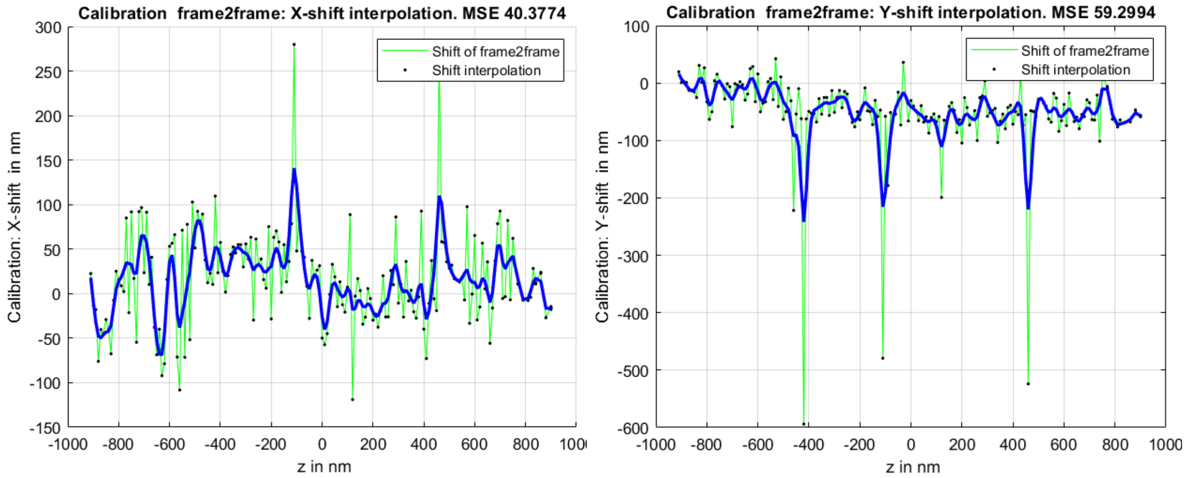


Fig 5. Interpolated functions shift_X(z) and shift_Y(z)

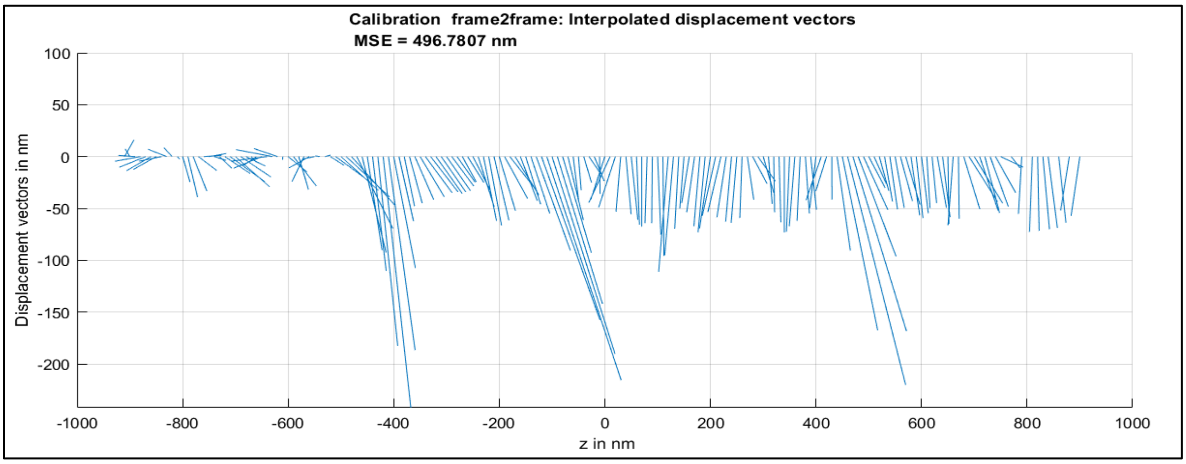


Fig. 6. 2D displacement vector as a function of z-value.

Comparing the total MSE generated by both methods, it is obvious that the interpolation-method overlaps with the grid-method only for a full grid consisting of one segment. For larger grid sizes, the grid method gives smaller MSE (see Fig 7).

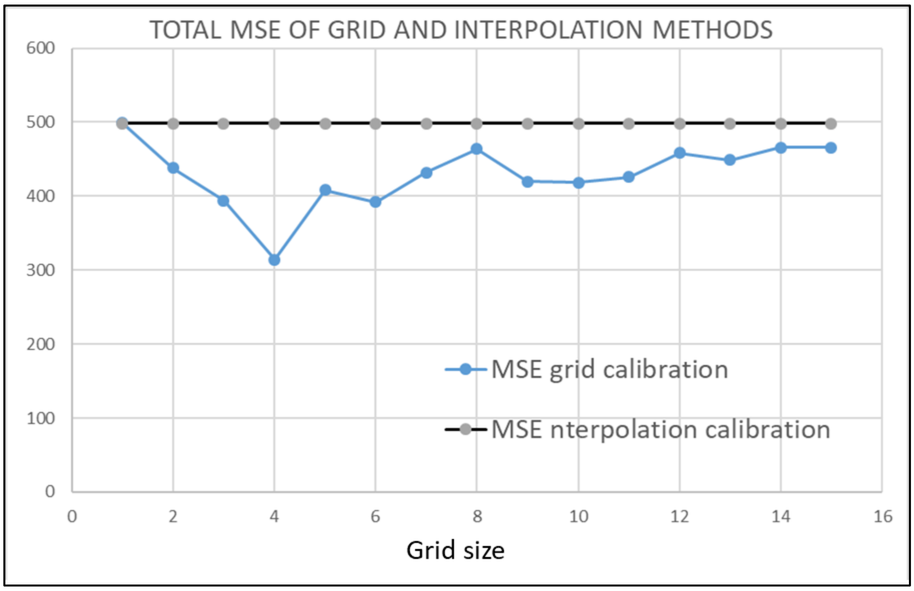


Fig. 7. Comparison of MSE for different grid sizes and interpolations

Button **REGISTRY GRID based**

Registration: Applying stored transformations in the form of a displacement vectors grid to other samples with point clouds collected from channel 0 and channel 1 executes the **REGISTRY GRID based** option.

The program interactively loads the selected displacement vectors grid (calibration file) and requests loading of a fix point cloud and a mobile point cloud to be registered.

Mobile points are subsequently divided into sectors, which have a selected displacement grid. Points from each sector are transformed locally according to the 3D vectors from displacement grid. The transformation progress is illustrated in Fig. 8. The calculated mobility point cloud is stored in its original directory with the prefix *'registred_name of sample'*.

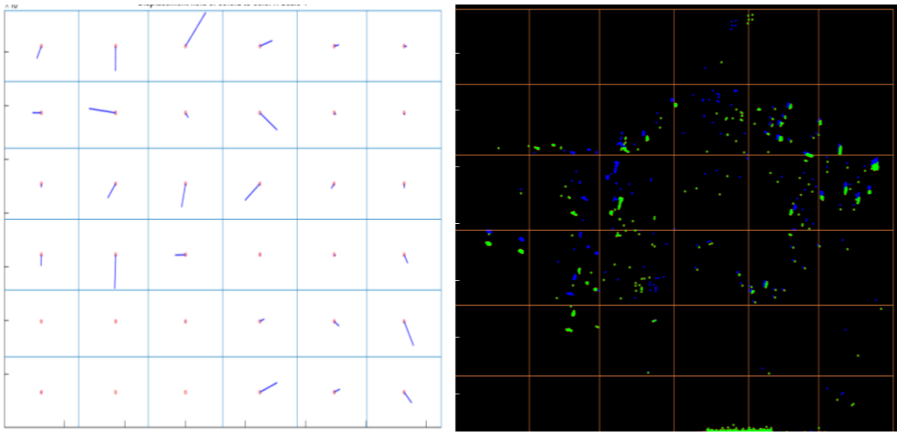


Fig. 8. Registration of sample (right) with displacement grid (left).
Blue points- before registration and yellow points- after registration

Button **REGISTRY INTERPOLATION based**

Applying the stored transformations in the form of X and Y shift functions of Z to other samples with point clouds collected from channel 0 and channel 1 causes the **REGISTRY INTERPOLATION based** option to be executed.

The program interactively loads the selected X and Y shift functions (interpolated_calibration file) and asks to load the fix point cloud and mobile point cloud to be registered. Then the y values x and y are corrected for each z value according to the shift_X/Y function. The calculated mobility point cloud is stored in its original directory with the prefix *'registred_interpolated_name of sample'*.

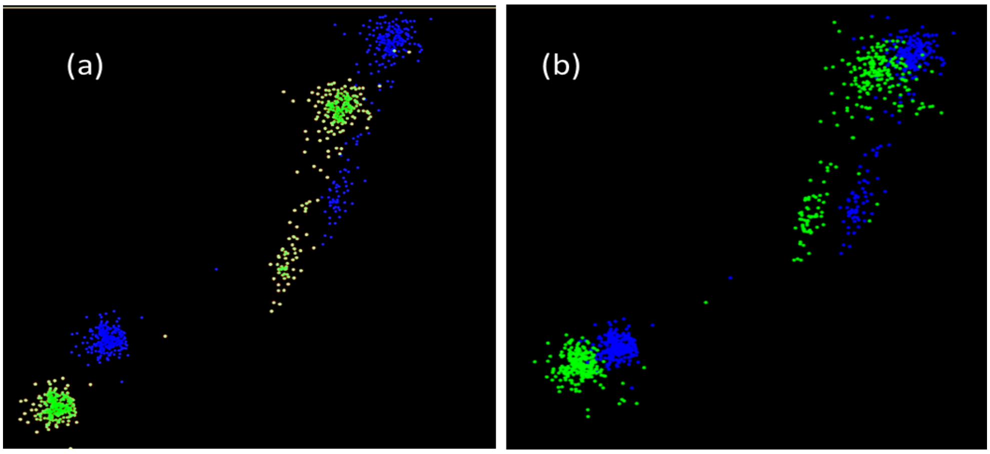


Fig. 9. Registration of sample with displacement grid (a) and interpolation based method (b).
Blue points before registration and yellow points-after registration

Comparison of registration of mobile points with grid and interpolation methods. Correction of point positions in both methods occurs in the same direction, but the transformation based on the displacement grid is larger.

Button **COLLOCATION**

Collocation : If two samples contain point clouds in the same space area (region), then the collocation problem can be formulated as the problem of finding places in this region where points from both samples lie close to each other. Due to the high accuracy of the locations of the points, it is very probable that the classic intersection of the two clouds (point sets) is an empty set. Therefore, the collocation problem is to find groups of points in both samples that overlap each other. Collocation computing started with command **COLLOCATION.** The task of collocation can be solved by two methods: method 1 (called standard method) looks for collocation points based on individual cloud cluster analysis, method 2 (called sensitive method) looks for collocation points based on cluster analysis in the union of both clouds.

Method 1: Standard search

The first step is to load both samples, the first as the passive (fixed) sample and the second as the active (mobile) sample. It is recommended to select as the active sample the sample with a smaller cloud size to speed up the analysis (less number of clusters). Next, the two samples are clustered separately with the DBscan method. Each cluster of the active sample is tested for intersection (overlap) with the clusters of the passive sample. The intersection test is performed by computing the convex hull intersection of the tested active cluster (cluster from active sample) with K convex hulls constructed for its closest neighbours among the clusters of the passive sample (default K=9). The closest neighbours of the tested cluster are determined based on the distance between its points and the points of the clusters from the passive sample.

If for any pair of clusters, the intersection of their convex hulls is not empty then the volume of the intersection set is calculated in relation to the volume of both convex hulls. This Intersection-over-Union (IoU) coefficient is stored for the cluster pair under consideration. In case the intersection with all K convex hulls is empty then the smallest distance between the convex hull of the tested cluster and the hulls of its K-neighbours from the passive sample will be stored.

The precision and quality of colocation depends on the selection of DBscan parameters, i.e. radius R as threshold for a neighbourhood search and a minimum number of neighbours *minpts. minpts* must be at least 4 to be able to construct 3D convex hulls. The radius R is determined for each sample separately as a multiple of the average minimum distance between the cloud points in sample, i.e. R *= alpha * average_min_distance*. For a small radius of clusters, it is unlikely that an intersection will occur (Fig. 10) while for large radii there is intersection at the edges or inside the clusters, where there may not be any point groups from both samples (Fig. 10). Thus, the selection of the radius R via the *alpha* factor is critical for the quality of the collocation detection.

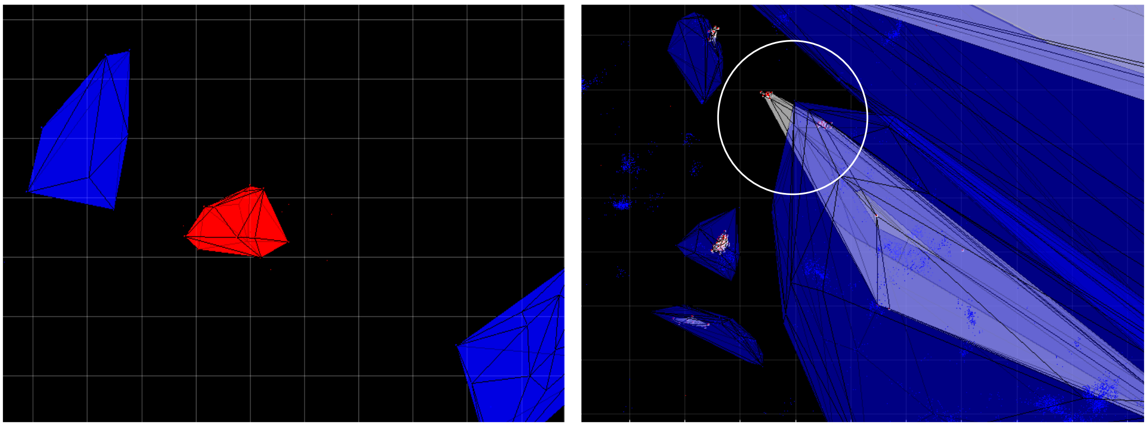


Fig 10. Active cluster (red) and passive clusters (blue) for alpha =1 (left) and alpha = 10 (right)

Thus, the quality of the collocation calculation depends on the *alpha* coefficient determining the size (radius) of the clusters.

High quality collocation would be achieved with a large number of cluster intersections while having a small cluster size. Both criteria are opposed to each other. Quality as the number of intersections is directly proportional to alpha and as a cluster dimension is indirectly proportional to alpha. The empirically determined saddle point of both criteria lies between *alpha* = 1.5 and *alpha* = 2.5. *Alpha* = 2 was assumed as default. Examples of cluster intersection for *alpha* = 2 are shown in Fig. 11.

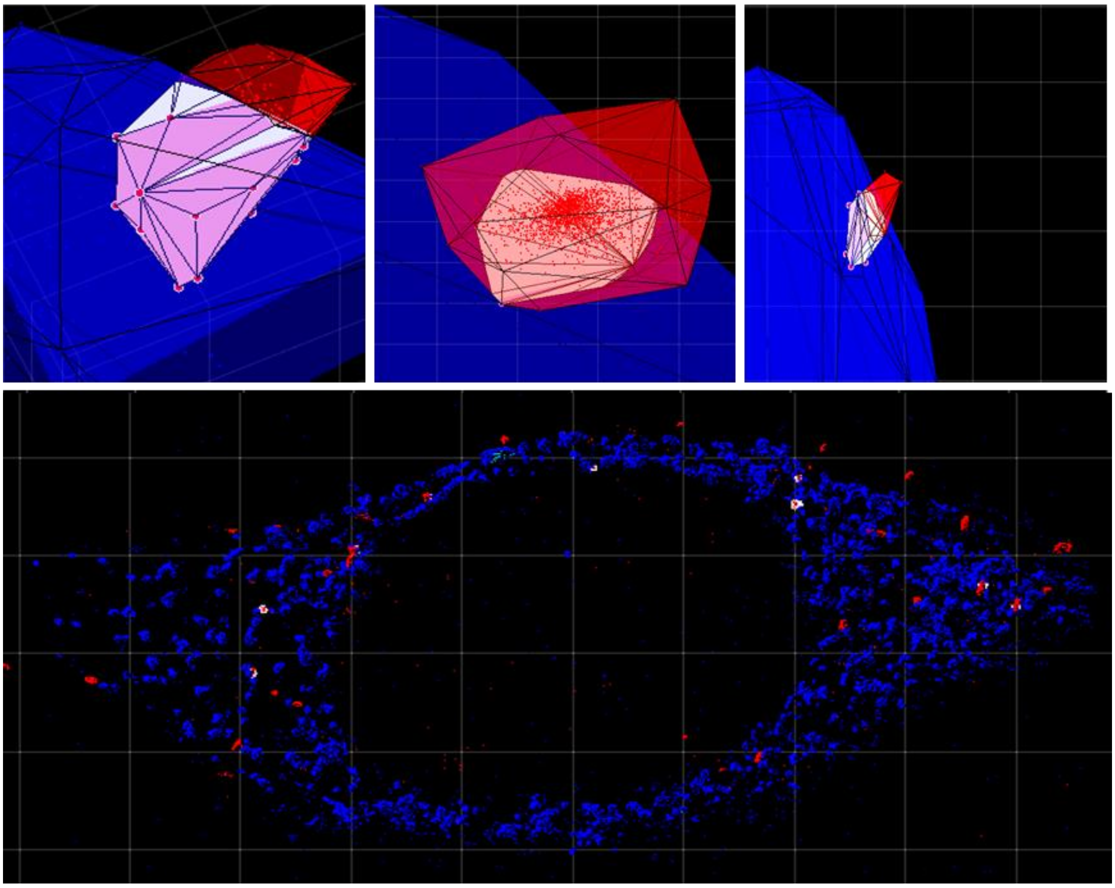


Fig. 11. Standard method: Examples of cluster intersection for alpha = 2
(passive cluster in blue, active cluster in red).

Method 2: Sensitive search

First, a union of both point clouds from the passive and active sample is performed. This new large set of points is clustered using the DBscan algorithm with the same *alpha* coefficient as described above. For large point clouds (more than 100 000 points), the process of clustering union cloud can take several minutes.

Then a search is made for clusters containing points from both clouds. These clusters are treated as collocation clusters. The convex hulls for points from the passive and active cloud and their volumes are calculated separately (Fig. 12). Each part of the cluster (passive and active) must contain at least 3 points. The IoU coefficient is calculated as the volume of the smaller of convex hulls in relation to the volume of both hulls and is stored for the analysed cluster.

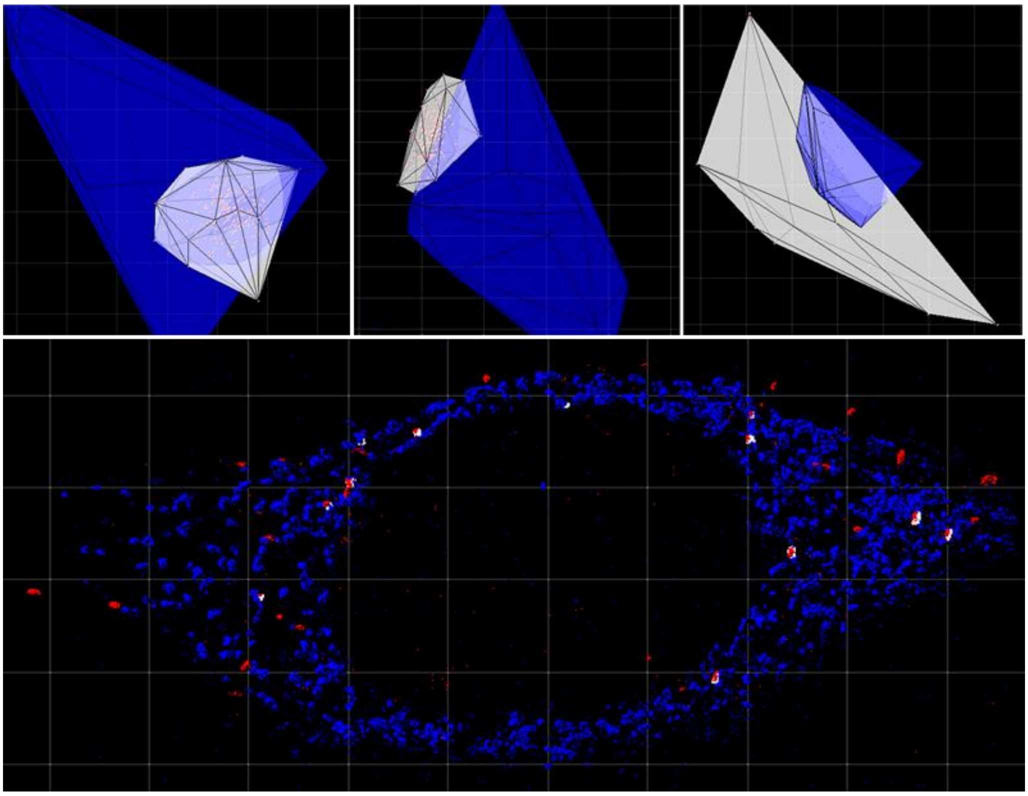


Fig. 12.Sensitive method: Examples of cluster intersection for alpha = 2
(passive cluster in blue, active cluster in red).

Statistics

After the collocation process is performed, the four main parameters of the collocation are computed:

- Volumetric average collocation for standard method (IoU) as the quotient of the sum of the volumes of all intersections by the sum of the volumes of their cluster pairs.
- Volumetric average collocation for sensitive method (IoU) as the quotient of the sum of the volumes of all smaller parts of the clusters by the sum of the volumes of the whole clusters of the passive and active part.
- Quantitative average collocation for standard method as the quotient of the number of points in the intersections to the number of all points in the active sample.
- Quantitative average collocation for sensitive method as the quotient of the number of active points lying in common clusters to the number of all points of the active sample.

This is illustrated in Fig. 13, which compares the collocations of points of the active sample obtained by method 1 (standard) with the collocations of points of the same sample obtained by method 2 (sensitive).

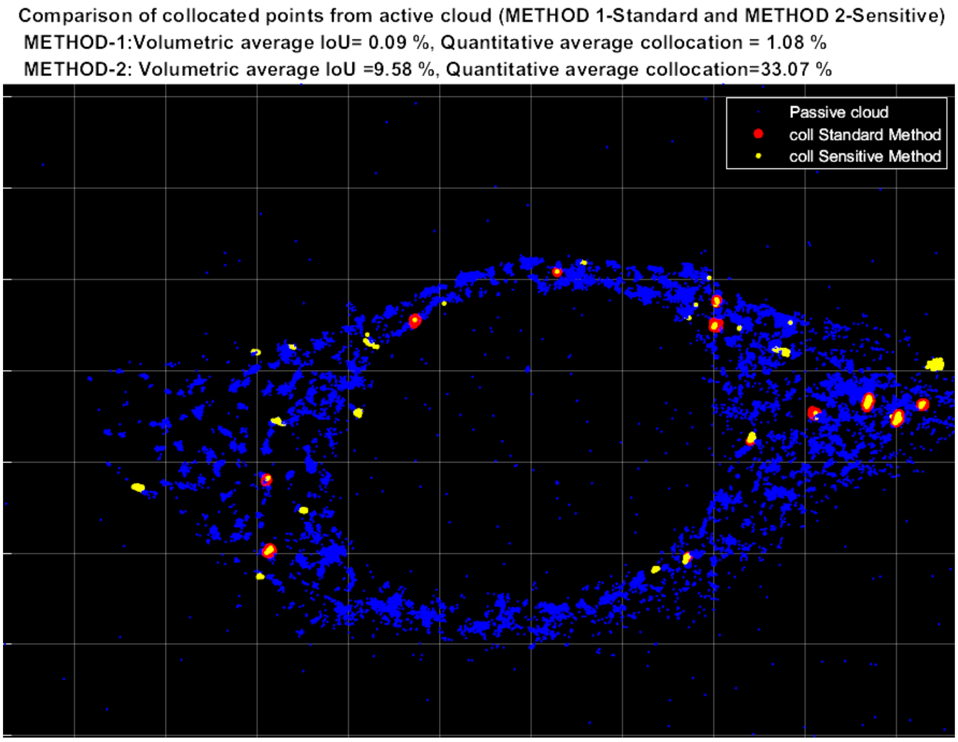


Fig 13 Comparison of collocated points detected by both methods:
standard method in red points, sensitive method in yellow points, passive cloud in blue points.

All collocations of points detected by standard method (red points) are also detected by sensitive method (yellow points). The sensitive method detects additional collocation places. This is illustrated in Fig. 14. The standard method does not create clusters in this area of points (blue) from the passive sample (too large point distances). There is only a cluster of points from the active sample (red). The method based on the union of point clouds constructs a common cluster from the passive and active points resulting in a new collocation place.

The next statistical characteristics are:

- The distribution of the minimum distance between the convex hulls of the clusters of the active sample with their closest neighbours of the convex hulls of the passive sample
- Histogram of the distribution of these distances.

Negative values of the x-axis denote intersections of convex hulls, i.e. collocations that occur (Fig. 15).

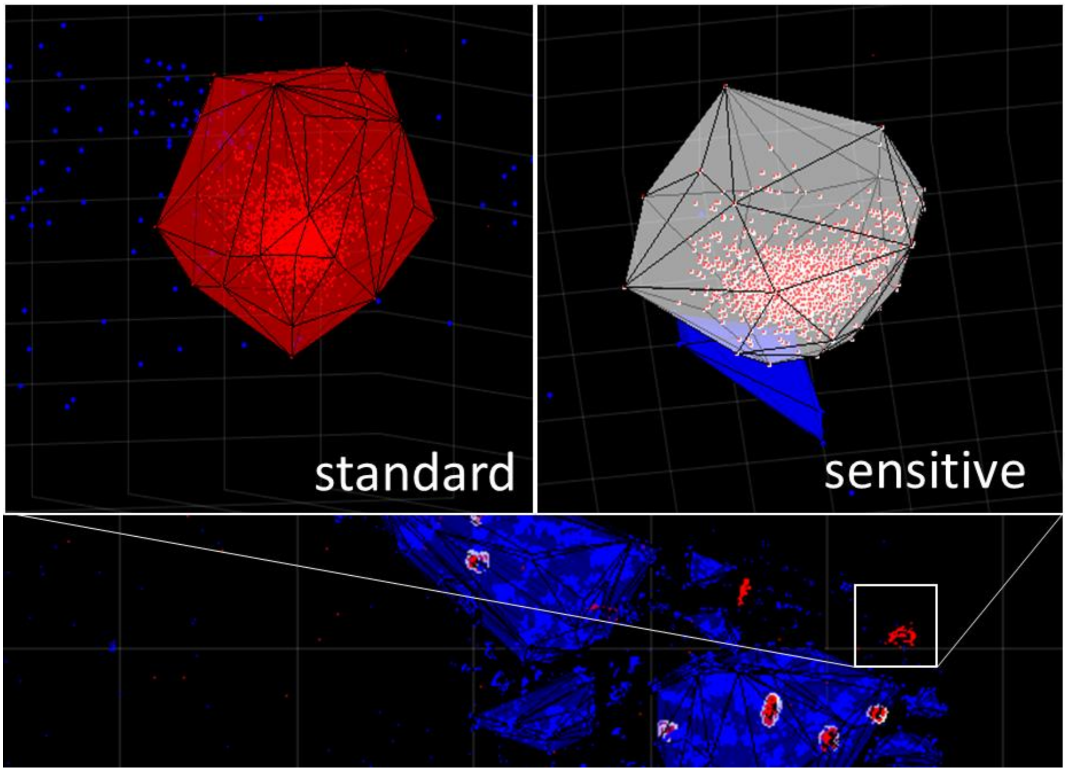


Fig 14. The standard method does not detect colocations because it does not construct a cluster from passive points in this region (left). The sensitive method constructs a common cluster and detects collocations (right).

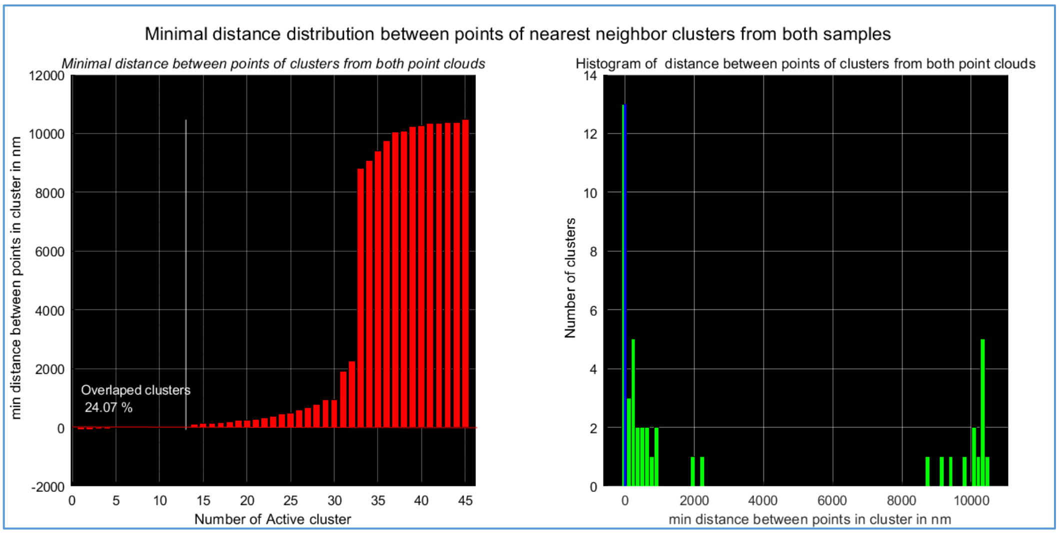


Fig. 15. Distribution of the minimum distance between convex hulls from clusters of the active sample and their closest neighbours of convex hulls from the passive sample and their histogram

Statistical data are automatically exported in excel format and are stored in its original directory with the prefix *'COLOC_typ_stat_name of sample*.

Button **EXAMPLE STATISTIC: PDF**

By performing multiple experiments on the same biological material, it is possible to cumulate the results of collocations and calculate an empirical probability distribution function PDF for the distance and size of cluster collocations. Examples of such PDF of distances between clusters of passive and active data are realized by this option (Fig. 16).

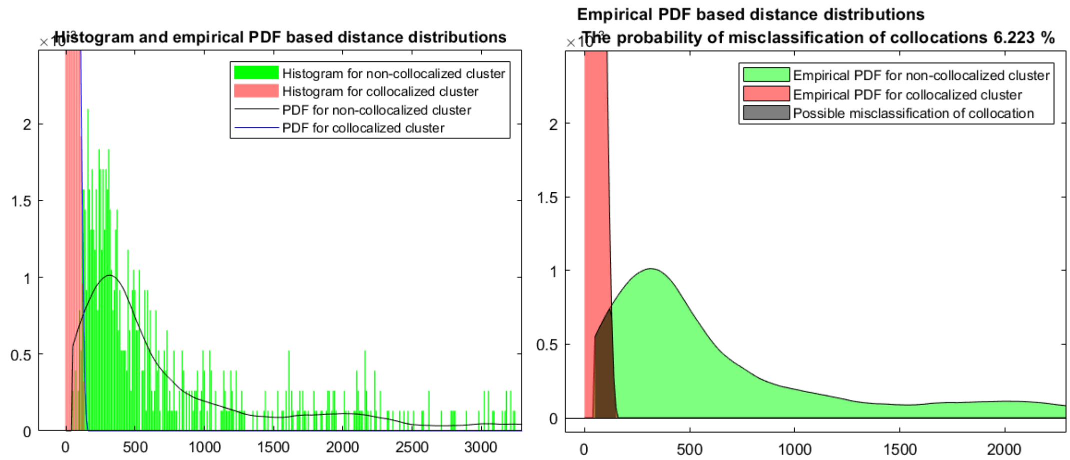


Fig. 16. Empirical PDF Distribution of the distance between clusters of the active sample and their closest neighbours of cluster from the passive sample and their histogram

Button **CLOSE ALL FIGURES**

Closes all open figures.

Button **EXIT**

Causes the system termination.