**Head\_orientation calibration app manual**

08/02/2021

Akihiro Itahara

# Table of Contents

1. Equipment and Software.
2. Prepare images for camera calibration.
3. Prepare images of the target object.
4. Use MATLAB app ‘cameraCalibrator’ to calibrate the camera.
5. Use the custom-made ‘Head\_orientation\_calibration\_app’ to reconstruct the target object.

# Equipment and Software

## Equipment

* Camera or Video camera (with a manual focus mode)
* A ‘scale’ object (An equilateral triangle with known lengths)
* PC (specifications: <https://jp.mathworks.com/support/requirements/matlab-system-requirements.html>)
* Checkerboard patterns

## Software

* MATLAB
* MATLAB Image Processing Toolbox
* MATLAB Computer Vision Toolbox

# Prepare for camera calibration.

To accurately reconstruct the target object from 2D images, calibrate the camera. To do this, use the ‘cameraCalibrator’ app in MATLAB to estimate the camera intrinsic, extrinsic, and lens distortion parameters. If you do this in an outdoor condition, it is recommended to do this procedure every time before the next step as lighting conditions tend to differ between time/days.

**Workflow**

1. Prepare a camera or a video camera.
2. Prepare a checkerboard pattern.
3. Take photos or extract still frames from the movies of the checkerboard pattern.
4. Save images in the folder.
5. Prepare a camera or a video camera.

Set the camera or video camera to manual focus mode.

1. Prepare a checkerboard pattern.

Check this web site (about the manual of camera Calibrator app) (<https://jp.mathworks.com/help/vision/ug/single-camera-calibrator-app.html?lang=en>).

A checkerboard is available in MATLAB. Run the command in the MATLAB command window, and you can get a checkerboard.

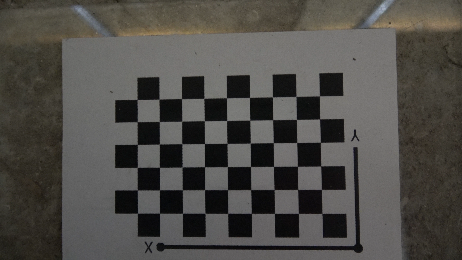


fig . The checkerboard image. The printed checkerboard needs to be attached to a flat surface.

*‘open checkerboardPattern.pdf’*

Print the checkerboard that roughly matches the size of the target object.

Attach the printed checkerboard to a flat surface (fig1).

Measure the length of checkerboard’s square in mm.

1. Take photos or extract still frames from the movies of the checkerboard pattern.

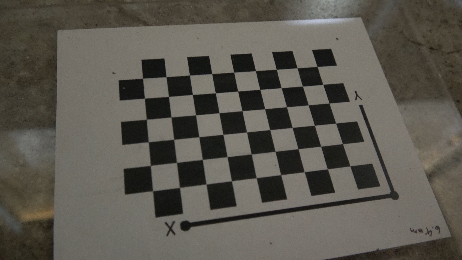


fig . Take images of checkerboard from various angles.

Put the attached checkerboard on the place where you will take photos or still frames from the movies of the target object.

Focus the camera on the checkerboard using manual focus. Do not use auto focus (as this will make different intrinsic parameters across photos of different angles). Make sure to use the same camera with the same settings when you take several photos of the target object.

Take photos of checkerboard from various angles (fig2). At least three images are needed, but between 10 and 20 images are recommended.

If you are using a video camera, take movies of the checkerboard from various angles and cut the images with editing software, such as Adobe Premiere Pro (it is often convenient to make a click sound to indicate the best frames while taking a movie).

1. Save images in the folder.

Save images in [Head\_orientation\_calibration\_app\_set/calibration/date/subjectname/\*.png] file. ‘date’ can contain only numbers (e.g., 210119). File names must be serial number (e.g., 1.png, 2.png, 3.png).

# Prepare images of the target object.

**Workflow**

1. Prepare a camera or a video camera.
2. Prepare the target object.
3. Take photos or extract still frames from the movies of the checkerboard pattern.
4. Save images in the folder.
5. Prepare camera or video camera.

Set the camera or video camera to the same focus (in manual focus mode) as when you took images of the checkerboard.

1. Prepare the target object.

The scale object (e.g., triangle with 30 mm in each side) needs to be attached to the target object. It must have more than 2 points to estimate the reconstruction accuracy later. I attached a 30mm equilateral triangle to crow beak using double sided tape.

1. Take photos or extract still frames from the movies of the checkerboard pattern.

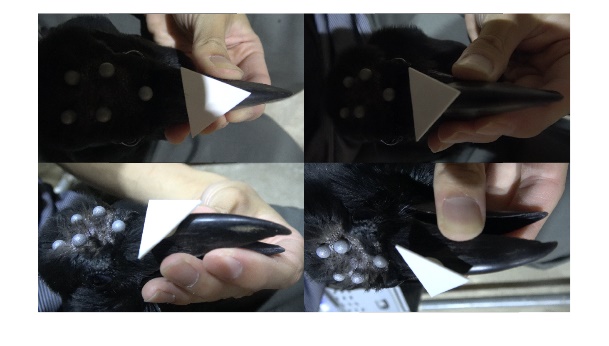


fig . Object images from 4 angles.

Take images of the target object from at least 4 angles (fig3). You must include the same reference points (e.g., markers and scale edges), at least 5 points, in all frames to estimate the position and orientation of cameras in the global coordinate (called ‘sfm-calibration’). The points you do not use for ‘sfm-calibration’ (e.g., eyes, beak) do not need to be included in all frames but must be included in at least 2 frames. Film the target object as large as possible in each image. When I took movies of crows, I held the crow body with my thighs, the upper beak with my left hand, and the video camera with my right hand. To take images from 4 angles, it needed around 10 seconds.

1. Save images in the folder.

Save images in [Head\_orientation\_calibration\_app\_set/photos/date/subjectname/\*.png] file. Date can contain only numbers (e.g., 210119). File names must be serial number (e.g., 1.png, 2.png, 3.png).

# Use MATLAB app ‘cameraCalibrator’ to calibrate the camera.

Use the ‘cameraCalibrator’ app in MATLAB, and estimate camera intrinsic, extrinsic, and lens distortion parameters.

**Workflow**

|  |
| --- |
| 1. Open MATLAB. 2. Run the command in the MATLAB command window; ‘*cameraCalibrator*’. Or select ‘cameraCalibrator’ app from MATLAB Toolstrip: On the Apps tabs in ‘Image Processing and Computer Vision’ section. 3. Add images to ‘cameraCalibrator’ and get camera parameters. 4. Select calibration options. 5. Evaluate calibration results. 6. Save all data in the folder. |

1. Open MATLAB
2. Run the command in the MATLAB command window.

‘*cameraCalibrator*’

Or select ‘cameraCalibrator’ app from MATLAB Toolstrip: On the Apps tabs in ‘Image Processing and Computer Vision’ section (fig4)

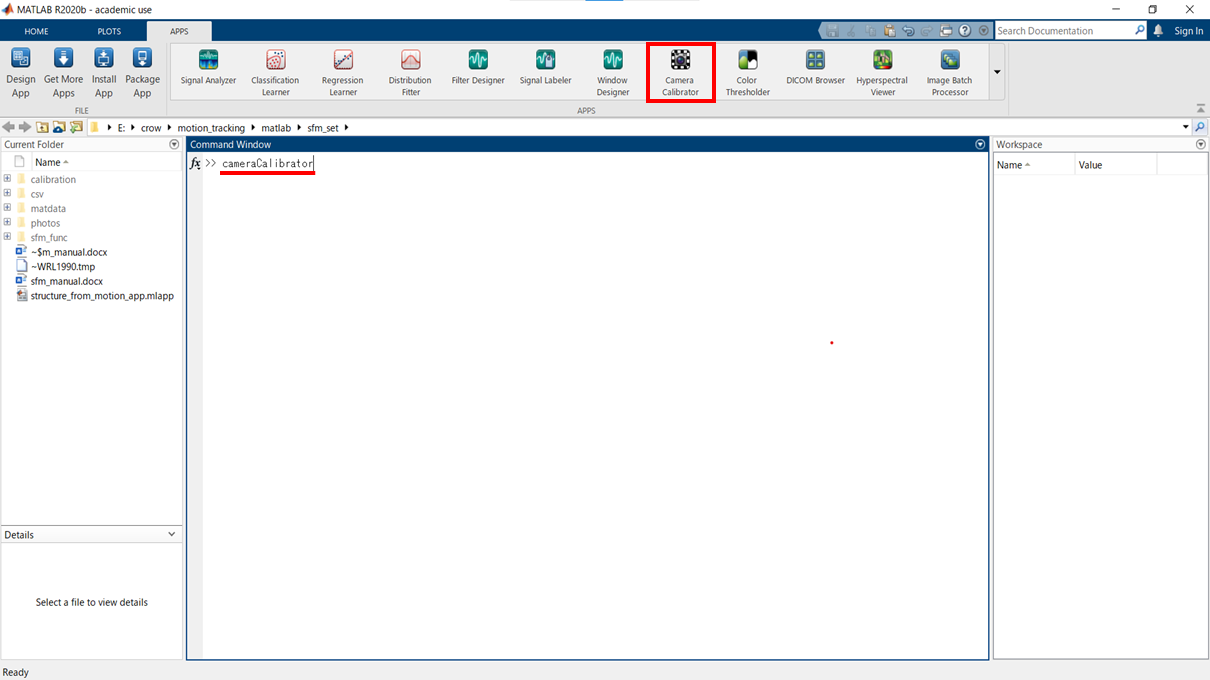


fig . Click Camera Calibrator app or run the command ‘cameraCalibrator’

1. Add images to ‘cameraCalibrator’ and get camera parameters.

Click ‘Add Images’ and select checkerboard images (10-20 images are recommended) (fig5).

Enter the length of one side of square of the checkerboard in the ‘Checkerboard Square Size’ dialog box (fig6).

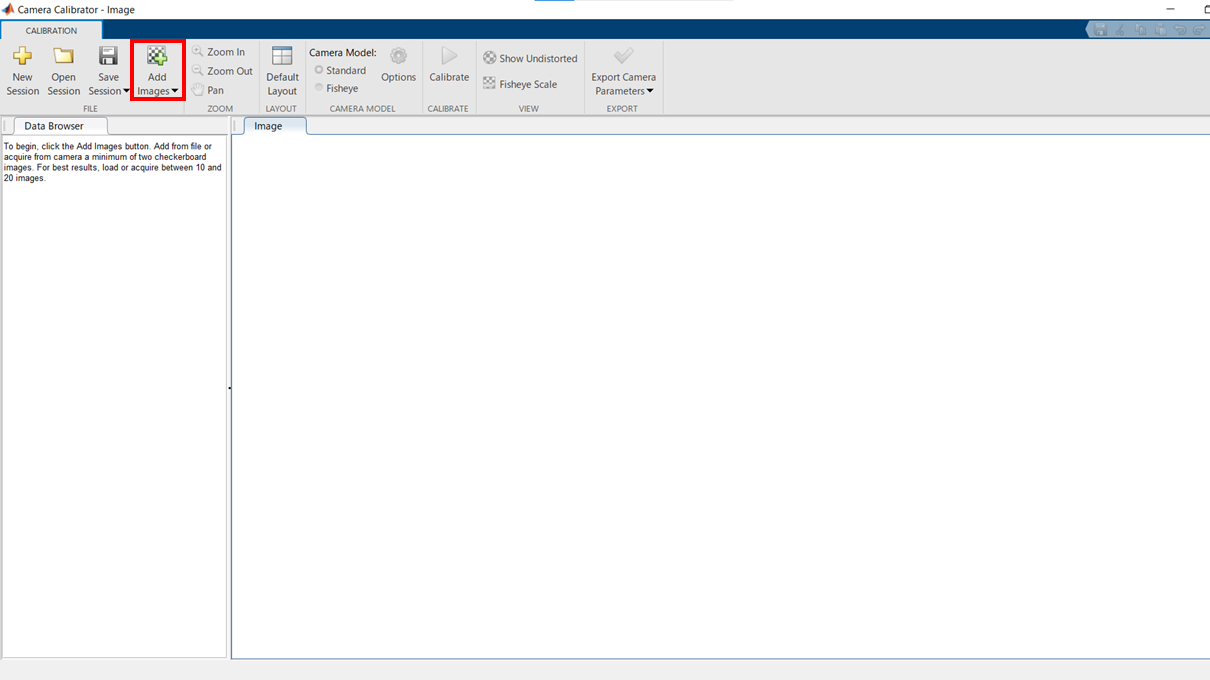


fig . Click ‘Add Images’

If any images are rejected because the focus is blurry or they are too bright or too dark, ‘Detection Results’ dialog box appears. It shows ‘Total images processed’, ‘Added images’ and ‘Rejected images’,

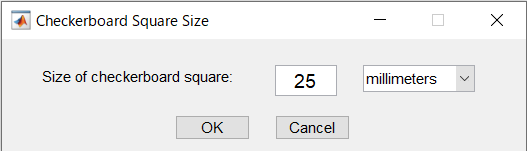


fig 6. Fill the size of checkerboard square(mm)

and then you can see added images with ‘Checkerboard origin’ (yellow square), ‘Detected points’ (green circles), and X-Y axes which are automatically detected based on the checkerboard pattern (fig7).

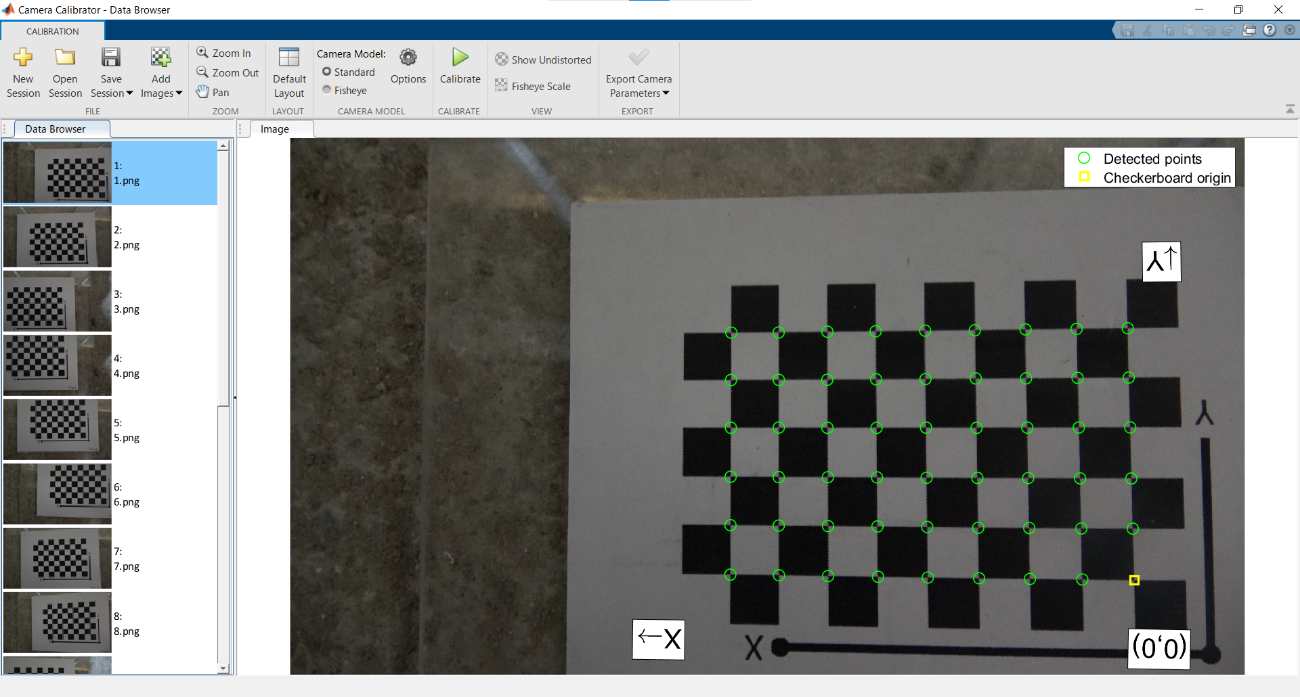


fig 7. Check detected points and checkerboard origin.

1. Select calibration options.

* Select the camera lens option (standard or fisheye)
* Click Options
  + Standard camera model

Radial Distortion (2 Coefficients/ 3 Coefficients)

Default: 2 Coefficients

When severe distortion is found, use 3 Coefficients.

Compute Skew

Default: Off

When your camera sensors contain imperfections that cause the x- and y-axes of the image to not be perpendicular, ‘Compute Skew’ should be turned on.

Compute Tangential Distortion

Default: Off

When the lens and the image plane are not parallel, ‘Compute Tangential Distortion’ should be turned on.

* + Fisheye model

Estimate Alignment

When the optical axis of the fisheye lens is not perpendicular to the image plane, ‘Estimate Alignment’ should be turned on.

* Click ‘Calibrate’

1. Evaluate calibration results.

* Check ‘Reprojection Errors Graph’ to find which images substantially lower the calibration quality (fig8). When I calibrate the camera (3840\*2160 pixel), reprojection errors were around 1.5 pixels.
* Check whether ‘Detected points’ and ‘Reprojected points’ fit well or not in each image.
* Check both ‘Camera-centric’ view of the patterns and ‘Pattern-centric’ view of the camera.
* Select images that substantially lower the calibration quality, do right click on it, and remove the images (fig9). Calibration will be automatically conducted again. Then evaluate calibration results again.

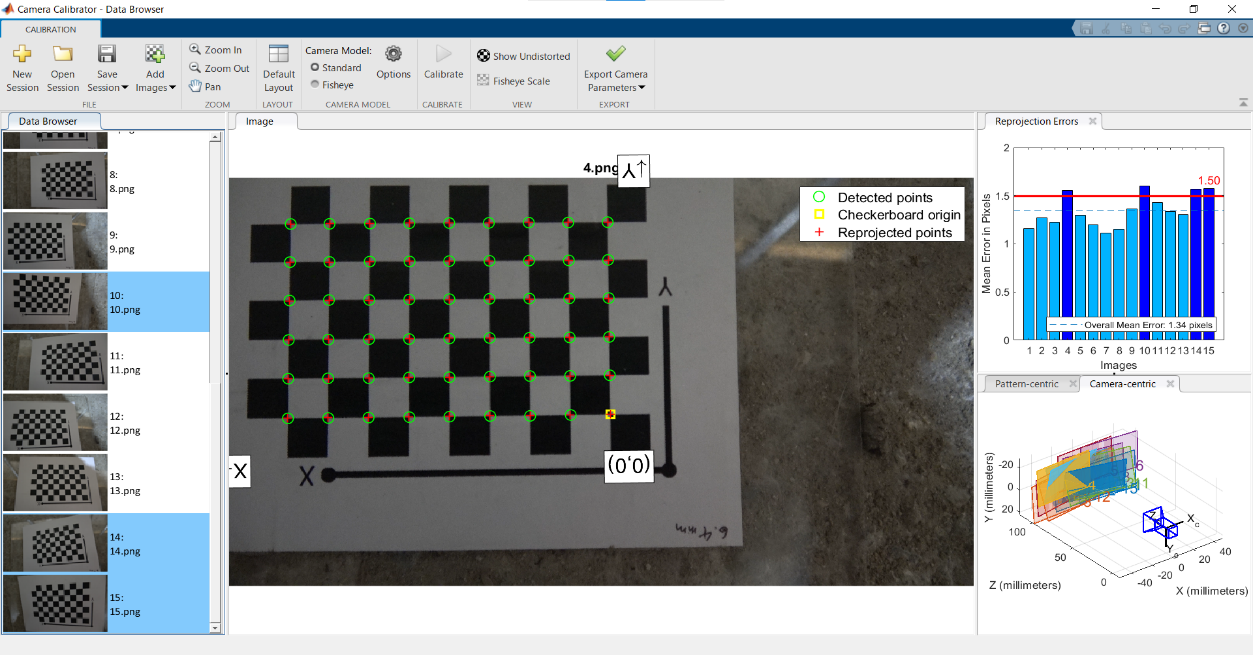


fig 9. Select the images that worsen calibration result and delate them.

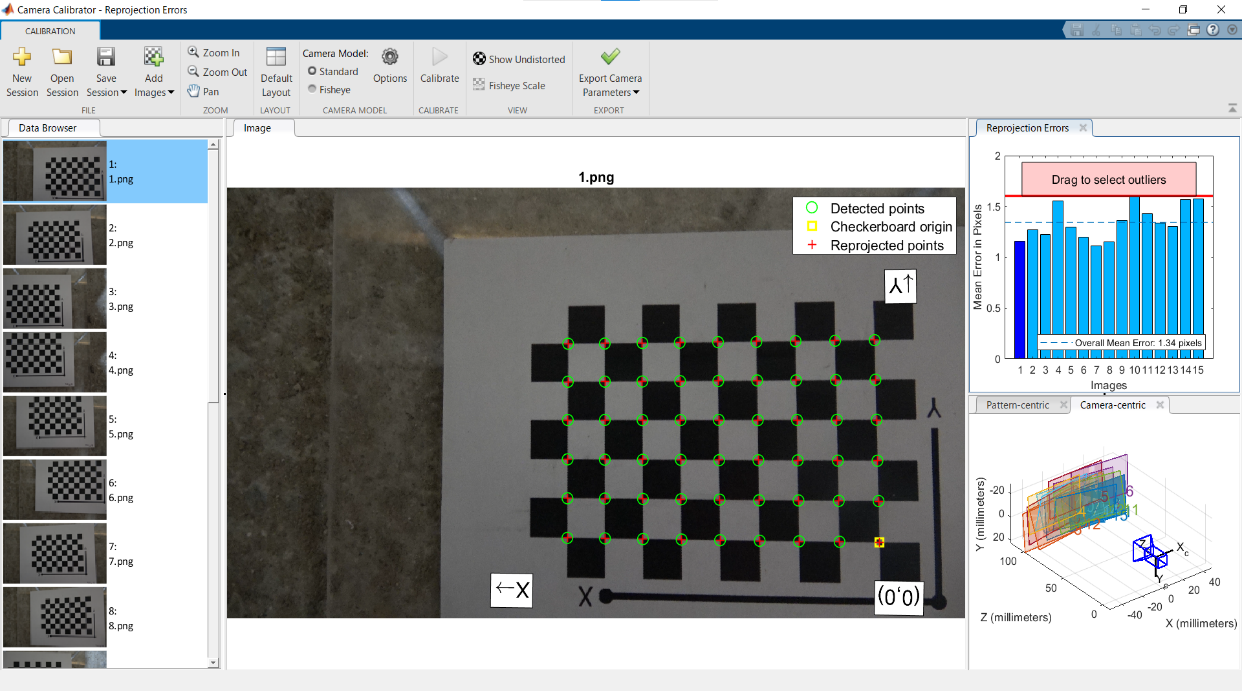


fig 8. Check reprojection errors, reprojected points, camera-centric and pattern-centric views, to find which images worsen the calibration result.

1. Save all data in the folder.

Click ‘Save Session’. Save the camera calibration data in [Head\_orientation\_calibration\_app\_set/calibration/ date/subjectname/calibrationSession.mat] file. Date can contain only numbers (e.g., 210119).

# Use ‘Head\_orientation\_calibration\_app’ to reconstruct the target object.

Reconstruct 3D object from 2D images using the custom-made ‘Head\_orientation\_calibration\_app.mlapp’. It has two processes. First, it estimates the position and orientation of the cameras in the global coordinates (called ‘sfm-calibration’) from calibration points (captured in all frames). Second, it reconstructs other points (does not need to be captured in all frames).

**Workflow**

1. Open ‘Head\_orientation\_calibration\_app’.
2. Fill ‘plots\_name’, ‘scale\_length(mm)’, and ‘reprojection\_error threshold(pixel)’.
3. Make sure to set ‘skip\_get\_plots’ to ‘No’.
4. Check the parameters in ‘Head\_orientation\_calibration\_app’.mlapp.
5. Click ‘start’.
6. Get points.
7. Check reconstruction results.
8. Check reprojection errors.
9. Check the reconstructed the lengths of the scale object.
10. Data are stored in the folders.
11. Open ‘Head\_orientation\_calibration\_app’

‘Head\_orientation\_calibration\_app.mlapp’ is in [Head\_orientation\_calibration\_app\_set] folder. To start the app, MATLAB and MATLAB Image Processing Toolbox are required.

1. Fill ‘plots\_name’, ‘scale\_length(mm)’, and ‘reprojection\_error threshold(pixel)’.

* Plots\_name

Separate point names with commas (fig10).

The names of scale object’s points start with ‘Scale’.

The names of the points which are not used to calibrate the camera location and angle start with ’nc‘.

* Scale\_length(mm)

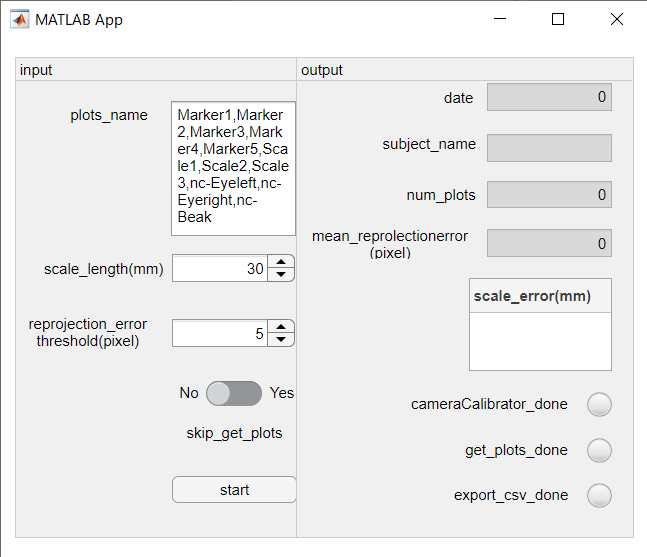
Input the scale length of one side which you attached to the target object.

* Reprojection\_error threshold(pixel)

Input the threshold of reprojection error. Default is 5 (3840 \* 2160 pixel camera) but note that this value could be changed (so far, up to 6.35 seems to be tolerable in my experiences – for the calibrations with crows).

fig 10. ‘Head\_orientation\_calibration\_app’ UI. Fill ‘plots\_name’, ‘scale\_length(mm)’ and ‘reprojection\_error threshold(pixel)’, and

‘set skip\_get\_plots’ to ‘No’.



1. Make sure to set ‘skip\_get\_plots’ to ‘No’.

If you set ‘skip\_get\_plots’ to ‘Yes’, the app skips getting 2D plots from 2D images. When you try to reconstruct 3D images from the data you have already acquired, set ‘skip\_get\_plots’ to ‘Yes’.

1. Check the parameters of the function ‘cl\_reconstruction’.

Check the parameters from 56 to 64 row in ‘Head\_orientation\_calibration\_app.mlapp’.

The value of ‘data.cl\_reconstruction.numbundle’ is set to 1000 (so far, this is enough for successful reconstruction), but this value could be changed.

Other values are set to default.

1. Click ‘start’

Start this app. Choose the folder in which 2D images of the target object are located. When you set ‘skip\_get\_plots’ to ‘Yes’, the message ‘Loading200101-BY point coordinates. Processing the data now.’ is displayed in MATLAB Command Window (date and subject name change depending on the file you choose).

1. Get points.

Plot the points (fig11). When points cannot be seen from the images, you can skip plotting by pressing ‘enter’ key (in this case, eyes, and beak; these are the points which are not used to sfm-calibration). If you make a mistake in the plotting position, restart getting plots from the beginning. After plotting, the message ‘200101-By point coordinates are saved. Processing the data now.’ is displayed in MATLAB Command Window (date and subject name change depending on the file you choose).

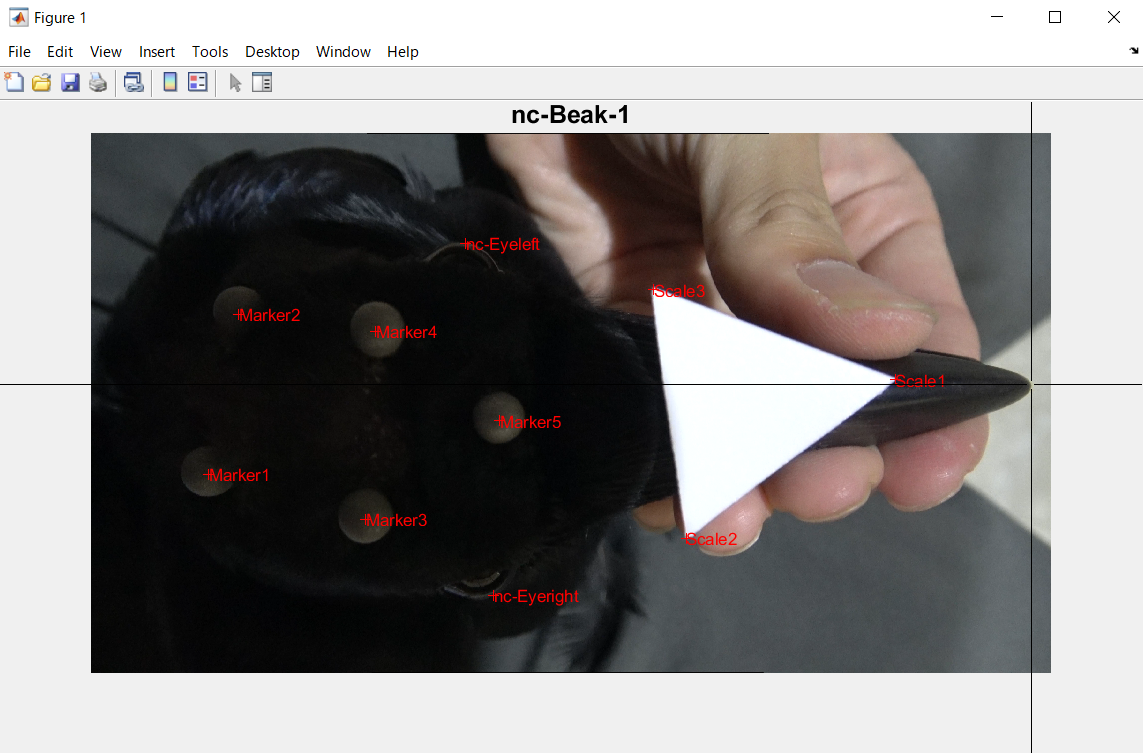


fig 11. Plot the points of the objects. The points you have plotted are displayed in ‘+’.

1. Check reconstruction results.

If the reconstruction is done successfully, three figures are displayed.

Fig1: Mean reprojection error transition plots (fig 12).

Fig2: The result of sfm-calibration (fig 13).

Fig3: the result of 3D reconstruction (fig 14).

Check the reconstructed 3D image.

1. Check reprojection errors.

Check the mean reprojection error transition plots. Its horizontal axis is num of bundle adjustment, vertical axis is mean reprojection error. So far, the result is acceptable if the value of mean reprojection error is 5 or less(pixel), but this threshold could be changed.

1. Check the lengths of the scale object.

Check ‘scale\_error (mm)’ in the user interface (fig15). In my experience, the error value is around 1% of the length of the scale object (e.g., error value is around 0.3mm in 30mm triangle)

1. Data are stored in the following folders.

‘.mat’ data is saved in ‘matdata’ folder.

‘.csv’ data is saved in ‘csv’ folder.

fig 12. Check reprojection error. Vertical axis is reprojection error value, horizontal axis is num of bundle adjustment.

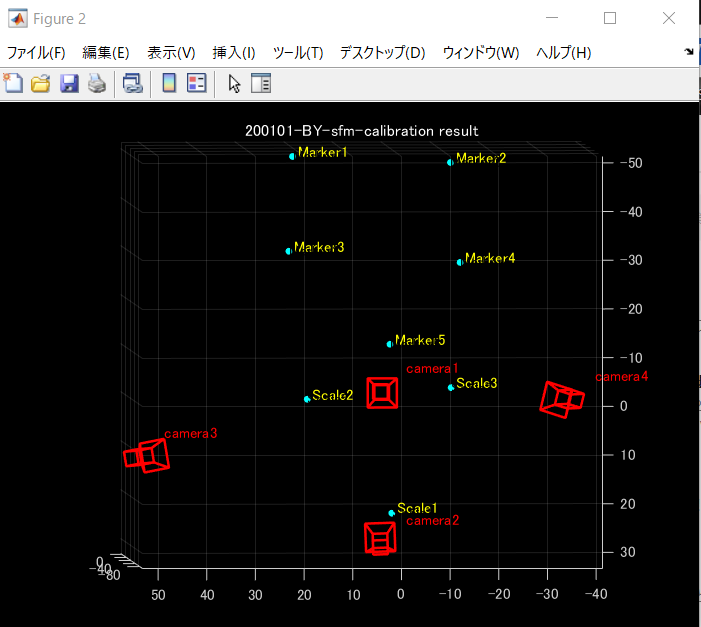
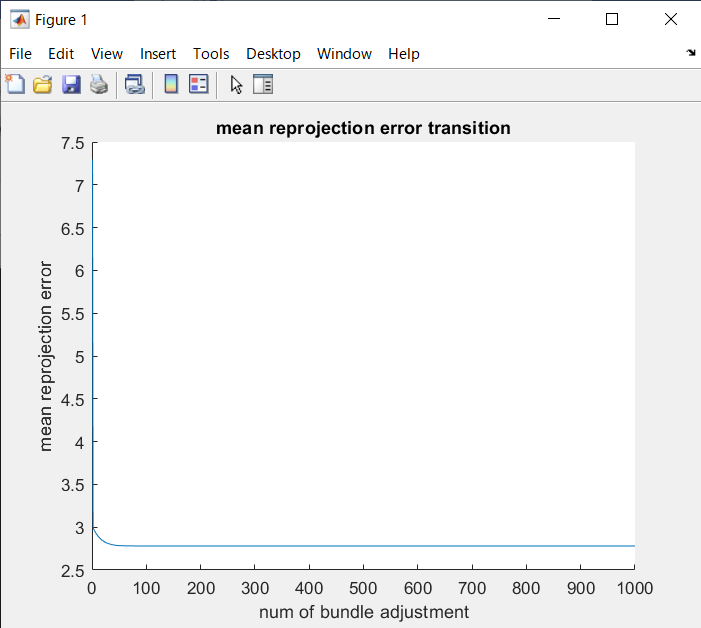


Fig 13. Check the marker and the camera locations by rotating the figure.

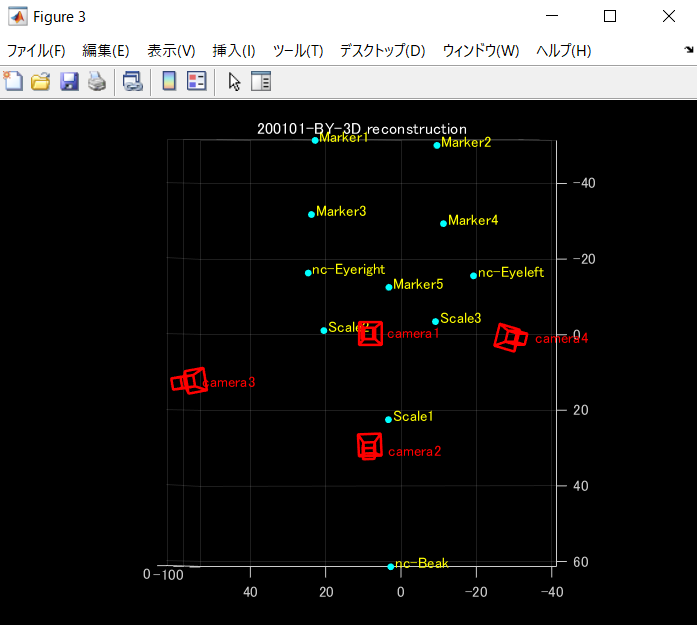


fig 14. Check the 3D reconstruction result.

fig 15. Check mean reprojection error(pixel) and scale error(mm). If these values are small enough, the object is reconstructed accurately.

