

# Markerauto2: A fast, robust and fully automatic fiducial marker-based tilt series alignment software for electron tomography

## Contents

<b>S1 Overview</b>	<b>2</b>
<b>S2 Installation</b>	<b>2</b>
S2.1 Prerequisites . . . . .	2
S2.2 Installtion of Markerauto2 . . . . .	4
<b>S3 Explanation of parameters and useful examples</b>	<b>4</b>
S3.1 Parameter explanation of Markerauto2 . . . . .	4
S3.2 Explanation of the content in the output folder . . . . .	5
S3.3 Examples . . . . .	7
S3.4 Generate final stack . . . . .	7

## S1 Overview

Here we present a fast, robust and fully automatic software, Markerauto2, for efficient and effective cryo-electron tomography (cryo-ET) tilt series alignment. Firstly, Markerauto2 implement the fiducial marker detection based on an fast accelerated wavelet template-based detection algorithm. After detecting fiducial markers, Markerauto2 utilizes a novel two-stage algorithm to establish correspondences in electron tomography, enhancing both speed and accuracy in this crucial step. Then, Markerauto2 employs a high-angle fiducial marker supplementation strategy designed to identify undetected markers in high-angle projections and generate more complete tracks. Finally, Markerauto2 utilizes a group-weighted parameter optimization algorithm for precise calibration of projection parameters. In comparison to other alignment software and method, Markerauto2 stands out through faster, more robust alignment while it is also fully automated.

This manual introduces the instructions for installation (see section S2), the detailed explanation of software parameters and several useful examples(see section S3).

## S2 Installation

The sections below explain how to download and install Markerauto2 on your computer.

### S2.1 Prerequisites

Note that Markerauto2 depends on and uses several external programs and libraries.

- **Linux or Unix-like operating systems**
- **GCC**

```
1 sudo apt install gcc
```

- **CMake**

```
1 sudo apt-get install cmake
```

- **Opencv4.5.5 (C++ version)**

```
1 sudo apt-get install build-essential cmake git libgtk2.0-dev pkg-config
   libavcodec-dev libavformat-dev libswscale-dev
2 Download the Opencv4.5.5 installation package (https://opencv.org/releases/).
3 unzip opencv-4.5.5.zip
4 cd opencv-4.5.5
5 mkdir build
6 cd build
7 cmake -DCMAKE_BUILD_TYPE=Release -DOPENCV_GENERATE_PKGCONFIG=ON -
   DCMAKE_INSTALL_PREFIX=/usr/local ..
8 make -j2
9 sudo make install
```

- **Ceres2.1 (C++ version)**

```
1 sudo apt-get install libeigen3-dev libatlas-base-dev libsuitesparse-dev
2 git clone https://ceres-solver.googleusercontent.com/ceres-solver
3 tar xzf ceres-solver-2.1.0.tar.gz
4 mkdir ceres-bin
5 cd ceres-bin
6 cmake ../ceres-solver-2.1.0
7 make -j3
8 sudo make install
```

## S2.2 Installtion of Markerauto2

We store the public release versions of Markerauto2 on GitHub, a site that provides code-development with version control and issue tracking through the use of git. We will not describe the use of git in general, as you will not need more than very basic features. Below we outline the few commands needed on a UNIX-system, please refer to general git descriptions and tutorials to suit your system. To get the code, you clone or download the repository. We recommend cloning, because it allows you very easily update the code when new versions are released. To do so, use the shell command-line:

- **Download**

```
1 git clone https://github.com/icthrm/Markerauto2.0.git
```

- **Compilation**

```
1 cd ./Markerauto2/
2 mkdir build
3 cd ./build
4 cmake ..
5 make -j2
```

## S3 Explanation of parameters and useful examples

### S3.1 Parameter explanation of Markerauto2

Markerauto2: command line arguments

==== Required options ====

- `--input (-i)`: specify the inputfile (The inputfile extension is `.mrc` or `.st`).  
`--input inputfile.st` or `-i inputfile.st`
- `--output (-o)`: specify the outputfile.  
`--output outputfile.st` or `-o outputfile.st`

- `--initangle (-a)`: specify the init angle file.  
`--initangle initangle.rawtlt` or `-a initangle.rawtlt`
- `--newangle (-n)`: specify the new angle filename.  
`--newangle newanglefile.tlt` or `-n newanglefile.tlt`
- `--diameter (-d)`: specify the marker diameter (pixel).  
`--diameter n` or `-d n` ( $n$  is the marker diameter. If `-d -1`, software will automatically initialize the diameter)

==== Optional options ====

- `--help (-h)`: print help content, no extra argument needed.
- `--rotationangle (-r)`: image rotation.  
`--rotationangle 0` or `-r 0`
- `--verbose (-v)`: display verbose information. Number greater than 1 enables this mode.  
If you do not provide this argument, it will run as `'-v 0'`.  
`--verbose 0` or `-v 1`
- `-t`: enable fast mode. In the fast mode, two-stage algorithm for the fiducial marker correspondence establishment is used. Otherwise, software use the global algorithm to match the fiducial markers.

We have supplied examples of the input file format on Github (<https://github.com/icthrm/Markerauto2.0.git>), which users can download and viewed.

### S3.2 Explanation of the content in the output folder

- `'xtiltangle.txt'`
  - Record the rotation angle of the each micrograph  $\alpha/\pi * 180$ .
- `'points.mot'`
  - The final coordinates of 3D point in space.

- 'cameras.params'
  - Save camera parameters for different images:  $s, \alpha, \beta, \gamma, t_0, t_1$ .
- 'BBa\_new.tlt'
  - Optimized new angle file.
- 'BBa\_fin.xf'
  - The file to be applied to the linear transformation of the image, calculated from the optimized camera parameters. The file is processed with Autom or IMOD to perform the image transformation to obtain the aligned image sequence (see subsection S3.4).
- 'fids.txt'
  - Output the width, height, number of photos, and ratio of the micrograph.
  - Output the  $i$ -th micrograph (frame).
  - Output the number of fiducial markers in the  $i$ -th micrograph.
  - The number of fiducial markers in each micrograph and output its coordinates  $(x, y)$  in the micrograph.
- 'proadd.txt'
  - Output high-angle fiducial markers coordinates  $(x, y)$  supplemented by reprojection.
- 'transmx folder'
  - Stores affine matrices between pairs of micrographs. The number of affine matrices may not be unique.
- 'matches folder'
  - Store the coordinates of the corresponding fiducial markers in the two matched micrographs.

### S3.3 Examples

When using Markerauto2 for alignment, the necessary parameters provided by the user include the initial micrograph, the tilt angle of the micrograph, the linear transformation \*.xf file storage path to align the images, and the optimized tilt angle storage path. Other parameters are optional and users can choose according to their needs (see subsection S3.1). The following are several useful examples of input from the command line:

**The software will automatically initialize the diameter, no fast mode, no log output:**

```
./markerauto2 -i BBa.st -a BBa.rawtlt -n BBa_new.tlt -o BBa_fin.xf -d -1
```

**User gives the initial diameter value, no fast mode, no log output:**

```
./markerauto2 -i BBa.st -a BBa.rawtlt -n BBa_new.tlt -o BBa_fin.xf -d 8
```

**The software will automatically initialize the diameter, with log output:**

```
./markerauto2 -i BBa.st -a BBa.rawtlt -n BBa_new.tlt -o BBa_fin.xf -d -1 -v
```

**The software will automatic initialize the diameter, with fast mode:**

```
./markerauto2 -i BBa.st -a BBa.rawtlt -n BBa_new.tlt -o BBa_fin.xf -d -1 -t
```

### S3.4 Generate final stack

After users use Markerauto2 to obtain optimized projection parameters, they can linearly transform the image sequence through the projection parameters to achieve image alignment. Files that perform linear transformations on images are stored in \*.xf files. Users can use Autom or IMOD to process the \*.xf file to realize the linear transformation with the following commands:

- **For Autom:**

```
1 mrcstack -i BBa.st -o BBa_fin.mrc -x BBa_fin.xf
```

- **For IMOD:**

```
1 newstack -input BBa.st -output BBa_fin.mrc -xf BBa_fin.xf
```