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

Contents

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 algorithm implementation details of Markerauto2 (see section S4), the instructions for installation (see section S2) and 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**

```
sudo apt install gcc
```

- **CMake**

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

- **Opencv4.5 (C++ version)**

```
sudo apt-get install build-essential cmake git libgtk2.0-dev pkg-config libavcodec-dev libavformat-dev libswscale-dev
Download the Opencv4.5.0 installation package (https://opencv.org/releases/).
unzip opencv-4.5.0.zip
cd opencv-4.5.0
mkdir build
cd build
cmake -DCMAKE_BUILD_TYPE=Release -DOPENCV_GENERATE_PKGCONFIG=ON -DCMAKE_INSTALL_PREFIX=/usr/local ..
make -j2
sudo make install
```

- **Ceres Solver2.1 (C++ version)**

```
sudo apt-get install libeigen3-dev libatlas-base-dev libsuitesparse-dev
git clone https://ceres-solver.googlesource.com/ceres-solver
tar zxf ceres-solver-2.1.0.tar.gz
mkdir ceres-bin
cd ceres-bin
cmake ../ceres-solver-2.1.0
make -j3
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**

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

- **Compilation**

```
cd ./Markerauto2/
mkdir build
cd ./build
cmake ..
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 inintangle.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 w`-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 downloaded 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 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 serveral 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:**

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

- **For IMOD:**

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

S4 Detailed implementation of Markerauto2

S4.1 Fiducial Marker Detection

The detection algorithm based on wavelet transform has three main steps(Hou *et al.*, 2023): (i) Wavelet-based template generation, (ii) Candidate fiducial marker generation, (iii) Markers determination using statistic-based filter and localization scheme. The pseudo-code of this algorithm is shown in Algorithm 1.

Wavelet-based template generation. The generation of fiducial marker template is rather important since it directly determines the success of template matching. However, the high-level noise in cryo-ET tilt series and the shape distortion caused by imaging bring great challenges to perfect template generation. To solve this problem, a translation-invariant wavelet algorithm(Izeddin *et al.*, 2012; Olivo-Marin, 2002; Shensa *et al.*, 1992) is utilized to capture the high-frequency shape characteristics of fiducial marker based on contrast information.

Firstly, the input micrograph is convoluted with a specific wavelet kernel to obtain the approximation coefficients. Then, the detailed coefficients of $j-th$ scale, W_j, which contain the features of fiducial markers, are extracted from the difference of adjacent scale approximation coefficients. The detailed calculations are illustrated as follows:

$$A_j\left(x,y\right)=\sum_n k_{j-1}\left(n\right)\sum_m k_{j-1}\left(m\right)A_{j-1}\left(x-n-\frac{L}{2},y-m-\frac{L}{2}\right),j\geq 1 \tag{1}$$

$$W_j\left(x,y\right)=A_{j-1}\left(x,y\right)-A_j\left(x,y\right),j\geq 1 \tag{2}$$

where A_j indicates the j^{th} scale approximation coefficients, $A_0\left(x,y\right)$ is the input micrograph, W_j indicates the j^{th} scale detailed coefficients, k_0 is set as the $k_0=\left[\frac{1}{16},\frac{1}{4},\frac{3}{8},\frac{1}{4},\frac{1}{16}\right]$, k_j is calculated by adding zeros between each element of k_{j-1}, L is the length of k_0. Analogically, this process is repeated serveral times. After iterative optimization, the noise in the detailed coefficients will be eliminated to a certain extent, and the shape characteristic of fiducial marker will become more and more obvious.

Next, the detailed coefficient map will be post-processed to ease the location of markers. Empirically, the detailed coefficients of the second or third scale are chose for extracting the features. The post-process contains three steps: (i) a threshold criterion (as shown in Equation 3) is proposed to alleviate the influence of remaining high-frequency signals in the detailed coefficients. (ii) the detailed coefficients is binarized to remove the outliers. (iii) the foreground in the detailed coefficients will be filtered by the shape-based criterion (as shown in Equation 4).

$$\tau\left(W_i, t_i\right) = \begin{cases} W_i & W_i \geqslant t_i \\ 0 & W_i < t_i \end{cases} \tag{3}$$

where $\tau_i = 3\sigma_i$, σ_i is the standard deviation of the detailed coefficients at scale i. We use a robust estimation of i that $\sigma_i = \overline{\sigma}/0.67$ (Starck *et al.*, 1995), which is obtained from the MAD estimate(Sadler and Swami, 1999).

$$C = \frac{4\pi S}{L^2} \tag{4}$$

where S represents the area of the region, while L represents the perimeter of the region.

Candidate fiducial marker generation. The second stage is responsible for finding candidates using the improved template matching algorithm. Given the generated template, the normalized cross-correlation (NCC) is calculated between the micrograph and template to reflects the similarity. The candicates can be located based on NCC value and pixel intensity by grid-based maximum search. To accelerate the algorithm, the NCC and pixels are weak filtered with hyperparameters τ_r and τ_p, which are list as follows:

$$\begin{aligned} \tau_r &= \mu_r + 2\sigma_r \\ \tau_p &= \mu_p \end{aligned} \tag{5}$$

where μ_r, σ_r is the mean and variance of NCC and μ_p is the mean of pixel value among the whole dataset. The candidates are defined as pixels with $NCC > \tau_r$ and $pixelValue > \tau_p$.

Markers determination using statistic-based filter and localization scheme. After obtaining the candidates, the further filtering is required to remove false-positive markers. The candidates can be categorized into two sets based on whether they contain detailed coefficient information.

The first statistical filter is crafted for candidates lacking detailed coefficient information, typically associated with very few fiducial markers. Gaussian distribution properties are employed to filter candidates that exhibit fiducial marker characteristics. Candidates with a value of the attribute $NCC \times$ pixel greater than a threshold τ_{np1} are retained. The formulation for τ_{np1} is as follows:

$$\tau_{np1} = \mu_{np1} + 3 \times \sigma_{np1} \tag{6}$$

, where μ_{np1} and σ_{np1} respectively represent the mean and variance of the attribute $NCC \times pixel$ for the candidates without detailed coefficients information.

After the first filter, the remaining candidates without detailed coefficient information are combined with candidates containing detailed coefficient information. The second statistical filter is devised for the merged set of candidates, utilizing the same method as the first filter, employing a threshold τ_{np2} for the value of $NCC \times$ pixel. Candidates with a value surpassing τ_{np2} are identified as fiducial markers. τ_{np2} is formulated as

$$\tau_{np2} = \mu_{np2} - 0.5 \times \sigma_{np2} \tag{7}$$

, where μ_{np2} and σ_{np2} respectively represent the mean and variance of $NCC \times pixel$ for the merged set of candidates.

The final step is to further refine the location of the detected fiducial makers. Considering that the characteristics of fiducial markers follow the Gaussian distribution, the Gaussian fitting is chose to do the refinement.

Algorithm 1 Wavelet Transform for Fiducial Marker Detection

procedure DETECTING(img)
    Using wavelet transform to **img** to obtain detail coefficients **W**: **W** $\leftarrow$ WaveletTransform(**img**);
    Threshold **W** and remove foregrounds **f** with small areas;
    Define the patch list **L** $\leftarrow \Phi$;
    Filter **f** and store the results of the filter in **L**;
    Calculate **diameter** in **L**, then use diameter to calculate the template average **t** of the element in **L**;
    Calculate the image of NCC by template matching: $I_{NCC} \leftarrow$ TemplateMatching (**img, t**);
    Calculate the mean μ_r, variance σ_r of the I_{NCC}, and the mean μ_p of pixel value.
    Define the patch list **c** $\leftarrow \Phi$;
    for i=0 to $I_{NCC}.width$ in step length of t.width **do**
        for j=0 to $I_{NCC}.height$ in step length of t.height **do**
            Define the grid $g_{i,j} = I_{NCC}.height[i + t.width, j + t.height]$;
            Find the point $p_{i,j}$ with the max NCC in grid $g_{i,j}$;
            if $p_{i,j}.ncc \geq \mu_r + 2\sigma_r$ and $p_{i,j}.pixel \geq \mu_p$ **then**
                $\mathbf{c} \leftarrow \mathbf{c} \cup p_{i,j}$
            end if
        end for
    end for
    Define the candidates without detailed coefficients information $c_1 \leftarrow \Phi$;
    Define the candidates with detailed coefficients information $c_2 \leftarrow \Phi$;
    for $p_{i,j}$ in c **do**
        if $p_{i,j}.information = 255$ **then**
            $c_2 = c_2 \cup p_{i,j}$;
        else
            $c_1 = c_1 \cup p_{i,j}$;
        end if
    end for
    Calculate the mean μ_{np1}, variance σ_{np1} of the $NCC \times pixel$ for the set c_1;
    for $p_{i,j}$ in c_1 **do**
        if $p_{i,j}.ncc * p_{i,j}.pixel \leq \mu_{np1} + 3\sigma_{np1}$ **then**
            $c_1 = c_1 - p_{i,j}$;
        end if
    end for
    Define $c_3 = c_1 \cup c_2$;
    Calculate the mean μ_{np2}, variance σ_{np2} of the $NCC \times pixel$ for the set c_3;
    for $p_{i,j}$ in c_1 **do**
        if $p_{i,j}.ncc * p_{i,j}.pixel \leq \mu_{np2} - 0.5\sigma_{np2}$ **then**
            $c_3 = c_3 - p_{i,j}$;
        end if
    end for
    Refine the location of the detected fiducial makers in c_3;
end procedure

S4.2 Fiducial marker correspondence establishment

Let 'point set' represent the positions of fiducial markers extracted from a projection. Given two point sets ($\mathcal{M}$ and $\mathcal{S}$) from different views, an affine transformation $\mathcal{T}(\cdot)$ is applied to the moving 'model' point set $\mathcal{M}$. The goal is to find an affine transformation $\mathcal{T}(\cdot)$ such that there exists a subset of $\mathcal{T}(\mathcal{M})$ with the maximum cardinality, and the points in this subset correspond to the points from a subset of the static 'scene' set $\mathcal{S}$ under a selected measure of distance.

Markerauto2 incorporates a robust and ultrafast fiducial marker matching algorithm (Han *et al.*, 2021). The determination of fiducial marker correspondence is divided into two stages: (i) initial transformation determination, and (ii) local correspondence refinement.

S4.2.1 Initial transformation determination

In this stage, the initial transformation estimation is conceptualized as a correspondence pair inquiry and verification problem. Local 4-point affine invariant features are then utilized for a fast comparison of geometric similarity among different fiducial marker positions on various micrographs (Aiger *et al.*, 2008; Koenderink and Van Doorn, 1991) . In each similarity inquiry, the extraction range of local features is confined, and a multiple-check technique of the extracted features is introduced for rapid consistency verification.

4-point affine invariant feature: For a point set $\mathcal{P} = \{p_a, p_b, p_c, p_d\}$ where line $l_{ab} = p_a + \alpha(p_b - p_a)$ intersects with line $l_{cd} = p_c + \beta(p_d - p_c)$ at the point p_e, the ratios $r_1 = \frac{||p_a - p_e||}{||p_a - p_b||}$ and $r_2 = \frac{||p_c - p_e||}{||p_c - p_d||}$ are preserved under any affine transformation. The 4-point set $\mathcal{P}$ along with the ratios r_1 and r_2 compose the 4-point invariant feature.

Local geometric constraint and numerical stability: The lines l_{ab} and l_{cd} are defined as the diagonal of an invariant feature $\mathcal{P}$, and the 4-point invariant feature is extracted within the length constraints of the diagonals. Specifically, first, build a nearest neighbor search tree (Bentley, 1975) to get the set of the distance $\{l_k\}$ between each point in a point set $\mathcal{S}$ with its nearest neighbor. Second, get the average l_{avg} and standard deviation l_{sdv} of $\{l_k\}$. Next, define a diagonal length for a 4-point invariant feature to be no more than $l_{max} = 3\sqrt{2}l_{avg} + l_{sdv}$. Finally, get all the possible point pairs within the point set $\mathcal{S}$, and only select the point pairs with length no more than l_{max} to produce the 4-point invariant feature.

Fast inquiry and verification of invariant feature: For a 4-point invariant feature $\mathcal{P} \in \mathcal{M}$ with invariant ratios r_1 and r_2, we try to check its correspondence in $\mathcal{S}$ based on the supposed position of its intersection e. If a point pair l_{ij} corresponds to one of $\mathcal{P}$'s diagonals, it has:

$$\begin{cases} p_{e_{r_1(ij)}} = p_i + r_1(p_j - p_i) \quad \text{or} \quad p_{e_{r_1(ji)}} = p_j + r_1(p_i - p_j), \\ p_{e_{r_2(ij)}} = p_i + r_2(p_j - p_i) \quad \text{or} \quad p_{e_{r_2(ji)}} = p_j + r_2(p_i - p_j), \end{cases} \tag{8}$$

where $e_{r1(ij)}, e_{r1(ji)}$ represent the two types of intersection e_{r1} defined by r_1, and $e_{r2(ij)}, e_{r2(ji)}$ represent the two types of intersection e_{r2} defined by r_2. Consequently, to find $\mathcal{P}$'s corresponding 4-point feature in $\mathcal{S}$, a nearest search tree of $\{e_{r2}\}$ is built and search e_{r1} is conducted based on the nearest search tree. To constrain the calculation in a local region, a *min&max* rule for the 4-point invariant feature is devised, where the the point pairs with $\|l_{p_ip_j}\| > l_{max}$ or $\|l_{p_ip_j}\| < l_{min}$ are discarded. By default, $l_{max} = 3\sqrt{2}l_{avg} + l_{sdv}$, and l_{min} depends on the nanoscale of fiducial markers and the pixel size of the micrograph.

Fast consistency verification: Given a micrograph with tilt angle β_1 and another one with tilt angle β_2 , the area of the corresponding plane shapes on these two micrographs have an approximate ratio of $cos\beta_1/cos\beta_2$. Therefore, given a candidate correspondence $\mathcal{P}'$ of the 4-point invariant subset $\mathcal{P}$, $\mathcal{P}'$ is accepted if and only if $(1-\xi)\frac{cos\beta_1}{cos\beta_2} \leq \frac{S_{\mathcal{P}}}{S_{\mathcal{P}'}} \leq (1+\xi)\frac{cos\beta_1}{cos\beta_2}$.

Robust estimation of the affine transformation: The random sample consensus (RANSAC) is utilized to perform robust estimation of affine transformations between two point sets. Multiple trials of congruent subset inquiry is carried out as follows:

1. Compose a random 4-point set $\mathcal{P}$ using point pairs within $\mathcal{M}$, and get its candidate congruent subsets Θ in $\mathcal{S}$ through the fast inquiry algorithm.

2. For each congruent subset $\mathcal{P}_S$ within Θ, compute the potential transform $\mathcal{T}(\cdot)$ using the method of least squares estimation. Apply the transform to $\mathcal{M}$ and determine the count of points in $T(M)$ that are close enough (congruent) to the points in $\mathcal{S}$.

3. If the number of congruent points between $\mathcal{T}(\mathcal{M})$ and $\mathcal{S}$ exceeds the current maximum record c_{max}, save the current value as c_{max} and update the termination condition I.

4. Repeat Steps $1 \sim 4$ until the termination condition is met.

S4.2.2 Local correspondence refinement

Although the initial transformation can be estimated with local geometric constraints, the distortion caused by sample deformation or beam-induced motion is non-negligible(Fernandez *et al.*, 2019; Lawrence *et al.*, 2006; Zheng *et al.*, 2017), therefore, requiring further local refinement.

First, large deviations and outliers in $\mathcal{M}$ are corrected by the calculated affine transformation $\mathcal{T}(\cdot)$. Then, the data interpolation and transformation refinement are carried out by the non-rigid coherent point drift, based on the Gaussian mixture model (GMM) (Myronenko and Song, 2010; Jian and Vemuri, 2010). We describe the probability that a point $s \in \mathcal{S}$ is corresponding to a point $m \in \mathcal{M}$ as an isotropic Gaussian function:

$$p\left(s|m\right) = \frac{1}{2\pi\sigma^2} exp\left(-\frac{\|s-m\|^2}{2\sigma^2}\right) \tag{9}$$

where σ is a parameter to describe the instability of the system. The probability that the point s belongs to the point set $\mathcal{M}$ can be defined as $p\left(s|\mathcal{M}\right) = \sum_{i=1}^{n} P(i)p(s, m_i)$, where $P(i)$ is the prior of s under the condition of the ith point m_i. Considering a drift transform $\mathcal{T}\left(\cdot; v\right)$ applied to the model for distortion correction, and assuming the point s either belongs to outliers with w probability or sampled from the point set $\mathcal{Y}$ with a uniform distribution, the GMM probability density function can be defined as:

$$\begin{aligned} p\left(s|v_i\right) &= w\frac{1}{K} + (1-w)\sum_{i=1}^{N}\frac{1}{N}p\left(s|m_i\right) \\ &= w\frac{1}{K} + (1-w)\sum_{i=1}^{N}\frac{1}{N}\frac{1}{2\pi\sigma^2}e^{-\frac{\left\|x-\mathcal{T}(m_i;v_i)^2\right\|}{2\sigma^2}} \end{aligned} \tag{10}$$

where v_i is the drift corresponding to the ith point m_i. Our aim is to find such a transformation $\mathcal{T}\left(\cdot; v\right)$ and parameter σ applied to all the points $m \in \mathcal{M}$ so that the negative log-likelihood from $\mathcal{M}$ to $\mathcal{S}$ is minimized:

$$E\left(\mathcal{T}\left(\cdot\right), \sigma^2\right) = -\sum_{j=1}^{K} log \sum_{i=1}^{N+1} P\left(i\right) p\left(s|m_i\right) \tag{11}$$

where $P(i)$ is the reweighted i.d.d prior and $p\left(s|m^{N+1}\right) = \frac{w}{K}$ presents the probability of outliers. A negative log-likelihood objective function could be effectively solved by the expectation-maximization (E-M) optimization, where the algorithm iterates between the E-step and the M-step until convergence. With Jensen's inequality (Jensen, 1906; Redner and Walker, 1984), the negative log-likelihood defined in Equation 11 is upper bounded by the following function in each iteration:

$$Q = -\sum_{j=1}^{K} \sum_{i=1}^{N+1} p\left(i|s_j\right) log\left(P^{new}\left(i\right) p^{new}\left(s_j|m_i\right)\right) \tag{12}$$

where $p(i|s_j) = P(i)p(s_j|m_i)/p(s_j)$ is the probability that m_i corresponds to s_j, the 'old' superscript indicates that a parameter is guessed in the E-step and the 'new' superscript indicates that a parameter is optimized from the negative log-likelihood function in the M-step. With the initial transformation inherited from the first stage, this iterative optimization is able to quickly converge.

S4.3 Calibration of projection parameters

The projection parameter model (Han *et al.*, 2019) in ET can be describle as follows:

$$\begin{pmatrix} u \\ v \end{pmatrix} = sR_\gamma P R_\beta R_\alpha \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + t \tag{13}$$

where $(X, Y, Z)^T$ is a coordinate representing a spatial point located in the ultrastructure; α represents the pitch angle along the tilt axis; β represents the tilt angle; γ represents the in-plane rotation within the projection plane; s represents the scale change of the view; $t = (t_0, t_1)^T$ represents the translation of the view; $(u, v)^T$ is the measured projection point and P denotes the orthogonal projection matrix. The detailed R_α, R_β, P and R_γ are defined as follows:

$$\begin{gathered} R_\alpha = \begin{pmatrix} 1 & 0 & 0 \\ 0 & cos\alpha & sin\alpha \\ 0 & -sin\alpha & cos\alpha \end{pmatrix}, R_\beta = \begin{pmatrix} cos\beta & 0 & -sin\beta \\ 0 & 1 & 0 \\ sin\beta & 0 & cos\beta \end{pmatrix}, \\ R_\gamma = \begin{pmatrix} cos\gamma & sin\gamma \\ -sin\gamma & cos\gamma \end{pmatrix}, P = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \end{gathered} \tag{14}$$

Our aim is to find the most appropriate s, α, β, γ and t. It is worth noting that since the tilt angle of the projection is already known as well, β is equal to the assumed tilt value. The tilt angle of the rotary axis of the projection during the imaging process is small, α can be initialized to 0 directly. The scale change of the view s is set to 1. So our initial estimation focuses on γ and t.

Initialization of projection parameters for high angle images can be achieved by means of affine transformation matrices $T(\cdot)$ for different images:

$$\begin{pmatrix} u_1 \\ v_1 \end{pmatrix} = T \begin{pmatrix} u_2 \\ v_2 \end{pmatrix} \tag{15}$$

We transform the projection model (13) by writing the rotation parameters α, β and the translation variable t into the same matrix. Bringing the changed projection model into Equation 15 gives the following expression:

$$\begin{aligned} & s_1 \begin{pmatrix} cos\gamma_1 & sin\gamma_1 & 0 \\ -sin\gamma_1 & cos\gamma_1 & 0 \end{pmatrix} \begin{pmatrix} cos\beta_1 & sin\alpha_1 sin\beta_1 & -cos\alpha_1 sin\beta_1 & t_0 \\ 0 & cos\alpha_1 & sin\alpha_1 & t_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ = T \cdot & s_2 \begin{pmatrix} cos\gamma_2 & sin\gamma_2 & 0 \\ -sin\gamma_2 & cos\gamma_2 & 0 \end{pmatrix} \begin{pmatrix} cos\beta_2 & sin\alpha_2 sin\beta_2 & -cos\alpha_2 sin\beta_2 & t_0^{'} \\ 0 & cos\alpha_2 & sin\alpha_2 & t_1^{'} \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{aligned} \tag{16}$$

Since the transformation approximates a rigid transformation, the scale changes of the view, s_1 and s_2, are set to 1. Therefore, Equation 16 can be expressed in the following form:

$$
\begin{pmatrix} cos\gamma_1 cos\beta_1 & sin\alpha_1 sin\beta_1 cos\gamma_1 + sin\gamma_1 cos\alpha_1 & -cos\alpha_1 sin\beta_1 cos\gamma_1 + sin\alpha_1 sin\gamma_1 & cos\gamma_1 t_0 + sin\gamma_1 t_1 \\ -sin\gamma_1 cos\beta_1 & -sin\alpha_1 sin\beta_1 sin\gamma_1 + cos\alpha_1 cos\gamma_1 & cos\alpha_1 sin\beta_1 sin\gamma_1 + sin\alpha_1 cos\gamma_1 & -sin\gamma_1 t_0 + cos\gamma_1 t_1 \end{pmatrix}
$$

$$
= T \cdot \begin{pmatrix} cos\gamma_2 cos\beta_2 & sin\alpha_2 sin\beta_2 cos\gamma_2 + sin\gamma_2 cos\alpha_2 & -cos\alpha_2 sin\beta_2 cos\gamma_2 + sin\alpha_2 sin\gamma_2 & cos\gamma_2 t_0^{'} + sin\gamma_2 t_1^{'} \\ -sin\gamma_2 cos\beta_2 & -sin\alpha_2 sin\beta_2 sin\gamma_2 + cos\alpha_2 cos\gamma_2 & cos\alpha_2 sin\beta_2 sin\gamma_2 + sin\alpha_2 cos\gamma_2 & -sin\gamma_2 t_0^{'} + cos\gamma_2 t_1^{'} \end{pmatrix} \tag{17}
$$

For the sake of clarity, we express the right-hand side of Equation 17 as follows:

$$
T \begin{pmatrix} u_2 \\ v_2 \end{pmatrix} = \begin{pmatrix} A & B & C & D \\ E & F & G & H \end{pmatrix} \tag{18}
$$

We can obtain the relationship as in Equation 19 and subsequently solve for γ_1.

$$
\begin{gathered} -\frac{sin\gamma_1 cos\beta_1}{cos\gamma_1 cos\beta_1} = \frac{E}{A} \\ tan\gamma_1 = -\frac{E}{A}, \gamma_1 = arctan(-\frac{E}{A}) \end{gathered} \tag{19}
$$

Once γ_1 is determined, the translation parameters t_1 and t_2 can be obtained by solving Equation 20.

$$
\begin{cases} \cos\gamma_1 t_0 + \sin\gamma_1 t_1 = D \\ -\sin\gamma_1 t_0 + \cos\gamma_1 t_1 = H \end{cases} \tag{20}
$$

With the aforementioned calculations, we can finalize the initialization of the projection parameters.

References

Aiger, D., Mitra, N. J., and Cohen-Or, D. (2008). 4-points congruent sets for robust pairwise surface registration. In *ACM SIGGRAPH 2008 papers*, pages 1–10.

Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. *Communications of the ACM*, **18**(9), 509–517.

Fernandez, J.-J., Li, S., and Agard, D. A. (2019). Consideration of sample motion in cryo-tomography based on alignment residual interpolation. *Journal of structural biology*, **205**(3), 1–6.

Han, R., Bao, Z., Zeng, X., Niu, T., Zhang, F., Xu, M., and Gao, X. (2019). A joint method for marker-free alignment of tilt series in electron tomography. *Bioinformatics*, **35**(14), i249–i259.

Han, R., Li, G., and Gao, X. (2021). Robust and ultrafast fiducial marker correspondence in electron tomography by a two-stage algorithm considering local constraints. *Bioinformatics*, **37**(1), 107–117.

Hou, G., Yang, Z., Zang, D., Fernández, J.-J., Zhang, F., and Han, R. (2023). Markerdetector: A method for robust fiducial marker detection in electron micrographs using wavelet-based template. *Journal of Structural Biology*, page 108044.

Izeddin, I., Boulanger, J., Racine, V., Specht, C., Kechkar, A., Nair, D., Triller, A., Choquet, D., Dahan, M., and Sibarita, J. (2012). Wavelet analysis for single molecule localization microscopy. *Optics express*, **20**(3), 2081–2095.

Jensen, J. L. W. V. (1906). Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta mathematica*, **30**(1), 175–193.

Jian, B. and Vemuri, B. C. (2010). Robust point set registration using gaussian mixture models. *IEEE transactions on pattern analysis and machine intelligence*, **33**(8), 1633–1645.

Koenderink, J. J. and Van Doorn, A. J. (1991). Affine structure from motion. *JOSA A*, **8**(2), 377–385.

Lawrence, A., Bouwer, J. C., Perkins, G., and Ellisman, M. H. (2006). Transform-based backprojection for volume reconstruction of large format electron microscope tilt series. *Journal of structural biology*, **154**(2), 144–167.

Myronenko, A. and Song, X. (2010). Point set registration: Coherent point drift. *IEEE transactions on pattern analysis and machine intelligence*, **32**(12), 2262–2275.

Olivo-Marin, J.-C. (2002). Extraction of spots in biological images using multiscale products. *Pattern recognition*, **35**(9), 1989–1996.

Redner, R. A. and Walker, H. F. (1984). Mixture densities, maximum likelihood and the em algorithm. *SIAM review*, **26**(2), 195–239.

Sadler, B. M. and Swami, A. (1999). Analysis of multiscale products for step detection and estimation. *IEEE Transactions on Information Theory*, **45**(3), 1043–1051.

Shensa, M. J. *et al.* (1992). The discrete wavelet transform: wedding the a trous and mallat algorithms. *IEEE Transactions on signal processing*, **40**(10), 2464–2482.

Starck, J.-L., Murtagii, F., and Bijaoui, A. (1995). Multiresolution support applied to image filtering and restoration. *Graphical models and image processing*, **57**(5), 420–431.

Zheng, S. Q., Palovcak, E., Armache, J.-P., Verba, K. A., Cheng, Y., and Agard, D. A. (2017). Motioncor2: anisotropic correction of beam-induced motion for improved cryo-electron microscopy. *Nature methods*, **14**(4), 331–332.