Fast Point Cloud Fusion For Incomplete SMLM Particles

Sabri Bolkar*

Abstract

Single molecule localization microscopy (SMLM) allows fluorescent imaging of nano structures well below the diffraction limit, but it suffers from low spatial resolution. Registration of identical particles offers a straightforward solution to increase the apparent resolution and SNR of localization microscopy sets. In this technical report, we propose a fast and global joint registration framework for SMLM particle averaging. We compare our results with the all-to-all registration framework proposed in [1].

The main motivation of this work is to develop a fast algorithm that will allow global registration of under-labeled 3d SMLM particles. Current state-of-the-art [1] is based on all-successive pairwise registration of particles that has quadratic computational complexity, hence it is not feasible to densely sample azimuth and elevation initialization angles in order to escape local optima points. The proposed joint registration method is inspired by [2], but we put additional constraints on transformation parameters by prohibiting reflection during registration. We further enhance the idea by enabling multiple initializations that allow us to reach the global maximizer. It is a work-in-progress with ongoing improvements that are summarized at the end of the report.

In SMLM particle averaging, each particle can be treated as noisy and incomplete realizations of the same underlying structure (i.e., binding sites). Furthermore, emission from a binding site generates multiple localizations captured during image acquisition. It is possible to use the exactly same scenario when modeling the registration algorithm. In the end, this will provide us a physically plausible particle averaging framework with inherent localization clustering abilities.

Let assume a central Gaussian mixture model with K components $G(x, \mu_k, \Sigma_k),\ k \in \{1, K\}$ having the means μ_k and the covariance matrices Σ_k actually generated the observed set of M particles $V : \{v_j\}_{j=1}^{M}$. But these particles are incomplete realizations of the central GMM by rigid transformations (except reflection) $T : \{R_j, t_j\}_{j=1}^{M}$. Now we can model probability distributions of each SMLM particle v_j with total N_j localizations using the central GMM as following:

$$P(v_j) = \sum_{i=1}^{N_j} \sum_{k=1}^{K} p_k G((v_{ji} - t_j)R_j^{-1}; \mu_k, \Sigma_k). \tag{1}$$

where p_k is the weight of each GMM component. This formulation exposes the unknown parameters denoted by θ that need to be estimated by our registration algorithm: $\theta = \{\{R_j, t_j\}_{j=1}^{M}, \{\mu_k, \Sigma_k, p_k\}_{k=1}^{K}\}$.

This registration framework can be cast as a maximum likelihood problem where log likehood $l(\theta) = \sum_j ln(P(v_j;\theta))$ is maximized. But under-labeled characteristics of particles prevent direct estimation of parameters. In this work, expectation conditional maximization (ECM) [3] is used to iteratively estimate each parameter by holding the others constant. This algorithm is a variant of famous expectation maximization (EM) [4] that is known to handle missing data well. But, this would still not be sufficient to obtain high quality particle averaging results. Due to lack of control at nano-level, SMLM particles have rather arbitrary orientation and directly performing registration will be plagued by local optima. To reach global maximizer, we also initialize each particle by rotating them with predefined angles uniformly sampled from $[0, 360]$, then choosing the parameters giving the best fit to the central GMM. In the below sections, we first describe our algorithm then present the details about the implementation.

Theoretical Analysis

Let us explicitly write the log likelihood function $l(\theta)$ previously stated in the above paragraph:

$$l(\theta) = \sum_{j=1}^{M} \sum_{i=1}^{N_j} ln\Big(P(v_{ji};\theta)\Big) = \sum_{j=1}^{M} \sum_{i=1}^{N_j} ln\Big(\sum_{k=1}^{K} p_k G_k(v_{ji};\theta)\Big). \tag{2}$$

*This work is carried out at TU Delft Quantitative Imaging Group

The main problem in (2) is that summation of Gaussian terms appears inside the logarithmic function which in turn complicates analysis and prevents obtaining closed-form expressions for the parameters θ [5]. The trick applied in EM algorithm is to assume the existence of a hidden random variable $Z : \{z_k\}_{k=1}^{K}$, which is included in our formulation by defining $P(V)$ as the marginal distribution of the complete-data distribution $P(V, Z)$ (double summation is avoided for brevity):

$$l(\theta) = \sum_{j,i} ln\bigg(\sum_{h=1}^{K} P(v_{ji}, z_h; \theta)\bigg). \tag{3}$$

Here, Z is an assignment variable representing which Gaussian component is responsible for which localizations. It takes values $\{0, 1\}$ such that $P(v_{ji}|z = k) = G_k(v_{ji}; \theta)$ and $P(z = k) = p_k$ (i.e., also denoted by $P(z_k)$ for simplification) for corresponding kth assignment. Therefore, being aware of the joint distribution $P(V, Z)$ simplifies the formulation by removing the summation appearing inside logarithmic function in (2). However, Z is actually not known to us a priori and we need additional considerations to solve the problem.

At this point, we will exploit the Jensen's inequality, $f(E[x]) \leq E[f(x)]$, which is defined for convex functions $f(x) : x \in R$. Let us introduce a distribution, denoted by $Q(Z)$, into our formulation in (3):

$$l(\theta) = \sum_{j,i} ln\bigg(\sum_{h=1}^{K} Q(z_h) \frac{P(v_{ji}, z_h; \theta)}{Q(z_h)}\bigg), \tag{4}$$

the expression in (4) can be expressed as expectation:

$$l(\theta) = \sum_{j,i} ln\bigg(E(\frac{P(v_{ji}, z_h; \theta)}{Q(z_h)})\bigg), \tag{5}$$

now we can apply Jensen's inequality to (5) and find a lower bound that can be used during maximization:

$$\sum_{j,i} ln\bigg(E(\frac{P(v_{ji}, z_h; \theta)}{Q(z_h)})\bigg) \leq \sum_{j,i} E(ln\bigg(\frac{P(v_{ji}, z_h; \theta)}{Q(z_h)}\bigg)). \tag{6}$$

As can be seen in (6), now logarithmic function operates inside the summation. In a way, Jensen's inequality allow us to carry out the optimization in a more straightforward manner. Further tightening the inequality to an equality condition will result in quite convenient expression that we will use during EM optimization. Equality holds if the expectation is performed over a constant valued function, thus we can choose $Q(Z) = cP(V, Z; \theta)$ where c is a constant real value. As both $Q(z)$ and $P(V, Z; \theta)$ are distributions, to satisfy $\sum_h Q(z_h) = 1$, $Q(Z)$ can be chosen to be the normalized expression given by [6]:

$$Q(z) = \frac{P(v, z; \theta)}{\sum_z P(v, z; \theta)} = P(z|v; \theta). \tag{7}$$

This expression gives the E-step of the EM algorithm where posterior distributions of Z given V are computed. Then optimization is performed to find maximizers for each parameter in θ (M-step). Considering Jensen's inequality, here we actually optimize the **lower bound** instead of the original formulation. It can be shown that both converges to the same local optima [5].

Now let us define the EM optimization as following:

$$\theta = argmax_\theta\Bigg(\sum_{i,j,h} P(z_h|v_{ji}; \theta) ln\bigg(\frac{P(v_{ji}, z_h; \theta)}{P(z_h|v_{ji}; \theta)}\bigg)\Bigg). \tag{8}$$

Due to the extended set of parameters, derivation of closed form expressions for all parameters $\theta = \{\{R_j, t_j\}_{j=1}^{M}, \{\mu_k, \Sigma_k, p_k\}_{k=1}^{K}\}$ is rather difficult using EM directly. Hence, as done in [2], we apply Expectation Conditional Maximization algorithm where maximization steps of EM (i.e., M-steps) are divided into smaller steps where closed form update expressions for each of the parameters obtained by taking the rest of the parameters as they are. In the below section, we present details for each step in our algorithmic implementation.

Algorithmic Analysis and Implementation

Theoretical analysis explained in the above paragraphs gives little intuition about how this framework can actually be implemented. In this section, specific details related to the algorithm is provided. First, a pseudo-code of our iterative algorithm is shown below:

```
Initialize $\theta = \{\{R_j, t_j\}_{j=1}^{M}, \{\mu_k, \Sigma_k, p_k\}_{k=1}^{K}\}$;
while $iter<$ MaxIter (e.g., 100) do
    if iter<FineIter (e.g., 20) then
        1) Coarse initialization: Rotate particles $\{R_j^a\}_{j=1}^{M}$ using coarse sampled angles, $a \in \{1, 4\}$;
        2) E step: Update $P^a(Z|V;\theta), \forall a$;
        3) M-T step: Update $\{R_j^a, t_j^a\}, \forall a$;
        4) Clamp step: When detected, clamp reflection in $R_j^a$ by $R_j^a \leftarrow I$;
        5) Transform particles using updated $\{R_j^a, t_j^a\}_{j=1}^{M}, \forall a$;
        6) Max step: Choose the best $R_j, t_j$ and $P(Z|V)$ from $\{R_j^a, t_j^a, P^a(Z|V)\}$ giving
         $max\big(similarity(V, GMM)\big)$;
        7) M-GMM step: Update $\{\mu_k, \Sigma_k\}$;
        8) Continue with the particles transformed using the best parameters
    else
        1)Fine initialization: Rotate particles $\{R_j^a\}_{j=1}^{M}$ using fine sampled angles, $a \in \{1, 64\}$;
        ... (Repeat steps 2 to 8) ...;
    end
end
```

Algorithm 1: Global joint SMLM particle averaging algorithm

The algorithm consist of alternating E and M steps of ECM algorithm until a predefined iteration is reached (i.e., 100). At the outset, we initialize all parameters. Rotation matrices are initialized by using uniformly sampled angles from 0 to 360 degrees and R_j^a of jth particle is estimated at each initialization (a). Then, the one initialization with the highest Battacharyya similarity value with respect to the central GMM is kept to be used for the next steps. Battacharyya similarity is defined exactly as in [1]. Here, the term coarse to fine used to indicate that we actually increase number of initializations from 4 to 64 after a predefined iteration passes (e.g., 20th iteration) to exploit roughly aligned particles for subsequent fine registration. Central Gaussian components are initialized from one of the particles. The chosen particle is down-sampled to ensure that the number of GMM components are lower than the number of localizations (i..e, 50-100 GMM means preferred). This scheme encourages clustering of localizations around central GMM components.

E Step

In the E step, posterior distributions of Z given particles V and GMM centers, $P(Z|V;\theta)$ are determined. Briefly, it is a soft assignment step where kth GMM component's probability of being assigned to the localization v_{ji} is estimated:

$$a_{jik} = P(z_{ji} = k | v_{ji}; \theta) = \frac{G_{jik}}{\sum_k G_{jik}}, \tag{9}$$

where isotropic Guassian term G_{jik} with $\Sigma_k = \sigma_k I$ is defined as:

$$G_{jik} = \frac{p_k}{\sigma_k} exp\bigg(\frac{-||R_j v_{ji} + t_j - \mu_k||^2}{2\sigma_k^2}\bigg). \tag{10}$$

Note that we assumed spherical (i.e. isotropic) Gaussian mixture components in the implementation. Previous works indicate no sign of significant improvement when using non-isotropic GMM mixtures [2].

M-T Step

In this step, transformation parameters for each particle v_{ji} are estimated fixing current a_{jik} values and

GMM components. This section simply formulates the problem with least-squares minimization and updates parameters using the work of Umeyana [7]. The $\{R_j, t_j\}_{j=1}^{M}$ updates are computed given the soft assignments of each localization to the Gaussian components (a_{jik}). The minimization can be stated as following:

$$\begin{cases} min_{R_j,t_j}\big(||(W_j + t_j e^T - M)\Lambda||_F^2\big) \\ R^T R = I,\ det(R_j) = 1 \end{cases}$$

where $\Lambda_j \in \mathrm{I\!R}^{KxK}$ is a diagonal matrix with entries as $\lambda_{jkk} = (\sum_i a_{jik}/\sigma_k)^{\frac{1}{2}}$, M indicates vectorized GMM centers $M = [\mu_1, .., \mu_K] \in \mathrm{I\!R}^{dxK}$ (d: dimension) and $W_j = [w_{j1}, .., w_{jK}] \in \mathrm{I\!R}^{dxK}$ such that each w_{jk} defined as:

$$w_{jk} = \frac{\sum_i a_{jik} v_{ji}}{\sum_i a_{jik}}. \tag{11}$$

This minimization problem has closed form solutions for both rotation and translation using singular value decomposition (see [7] for details).

Clamp Step

It should be noted that M-T step can be improved by simply using a more modern technique for estimation of point-to-point transformation parameters. The main reason of this claim is that, for under-labeled particles, Umeyana's method fails to correct for reflection. Therefore, we include a reflection clamping step when estimating rotation parameters. In this step, reflection is checked and equalized to identity matrix when detected.

Max Step

In this step, the parameters belonging to the highest Bhattacharyya similarity computed between the transformed particle and the central GMM are preserved for the next iteration. This operation is performed for all particles in the dataset.

M-GMM Step

As the last step in the ECM algorithm, GMM parameters $\{\mu_k, \sigma_k\}_{k=1}^{K}$ are estimated by from the best transformation parameters and posterior distributions. The expression for the local optima points are simply obtained by equalizing the respective derivatives to zero. It can be shown that actually the GMM centers maximizing log likelihood are actually weighted average of the transformed particles:

$$\mu_k = \frac{\sum_{ji} a_{jik}(R_j v_{ji} + t_j)}{\sum_{ji} a_{jik}} \tag{12}$$

$$\sigma_k = \frac{\sum_{ji} a_{jik}||R_j v_{ji} + t_j - \mu_k||_2^2}{\sum_{ji} a_{jik}} \tag{13}$$

In the implementation, we ignored updates for weights (a.k.a priors) p_k and fixed them to the uniform distribution $p_k = 1/k$. Previous works show no significant improvement for updates applied to the GMM weights.

Computational Complexity

Our joint registration framework has computational complexity of $O(MNK)$ where N is the average number of localizations per particle, M is the number of particles, and K is the number of GMM components. On the other hand, all-to-all framework has complexity of $O((MN)^2)$, which is order of magnitude slower comparing to the proposed algorithm.

Results

Particle averaging results for simulation and experimental datasets are shown in Figure 2 and Figure 3 respectively. We compare our reconstruction outcomes against the all-to-all framework [1].

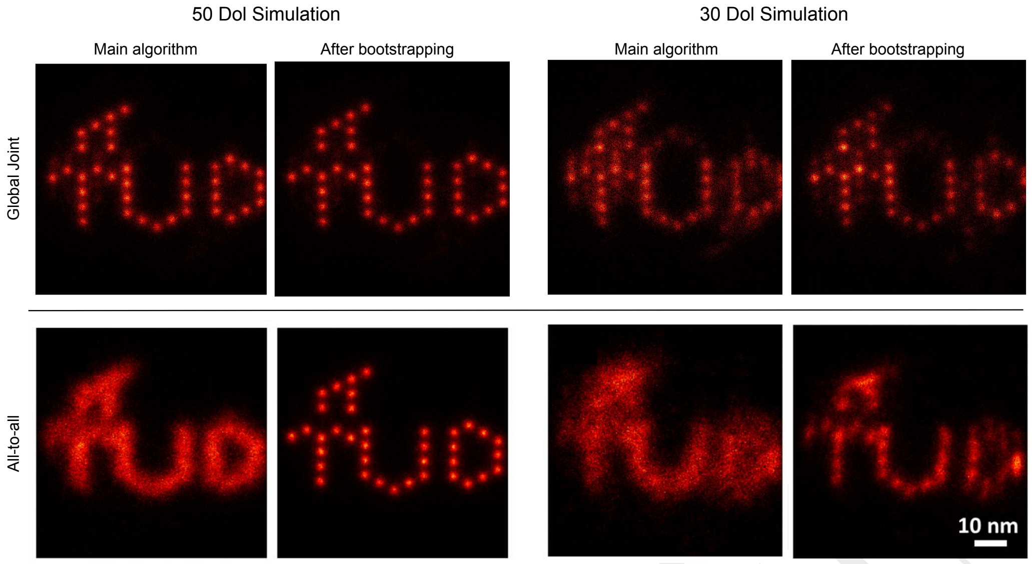

Figure 1: Qualitative comparison of Global Joint and All-to-all registration for simulation set.

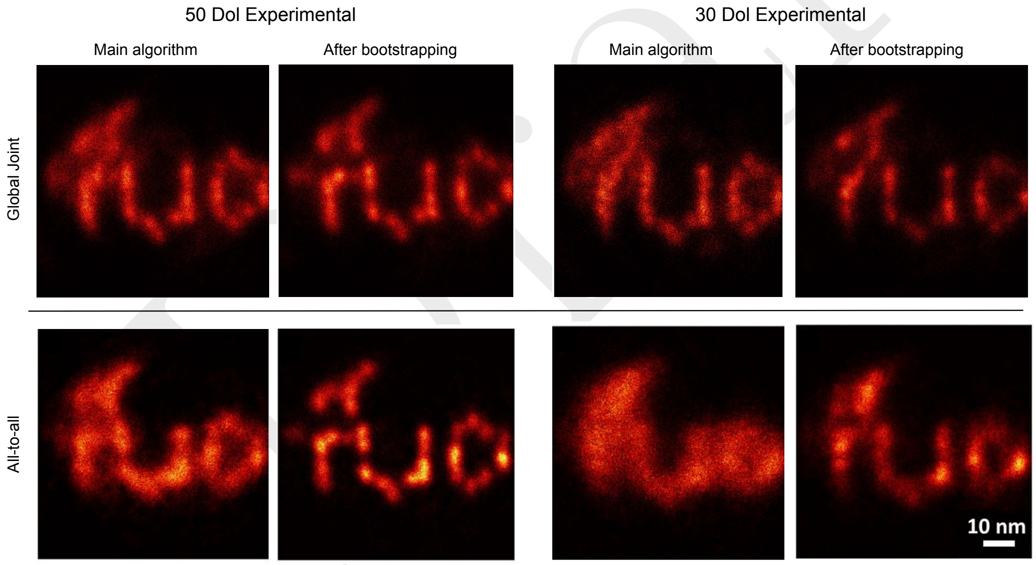

Figure 2: Qualitative comparison of Global Joint and All-to-all registration for experimental set.

Future Work

Global joint SMLM particle averaging is a work-in-progress with many aspects to include into the framework:

- Include individual localization uncertainties. At the moment, algorithm assumes fixed uncertainties for all localizations.
- Implement a better formulation for the M-T step instead of Umeya's solution developed in 1991.
- Improve global optimization framework by using Bayesian optimization rather then uniform sampling of rotation space.

- Implement a consistency check block such that transformed particles with less similar to the central GMM are re-processed at later iterations.
- Enable paralleled GPU computation when estimating transformation parameters as the algorithm steps are independent for each particle (except M-GMM step).

References

[1] Hamidreza Heydarian, Florian Schueder, Maximilian T Strauss, Ben Van Werkhoven, Mohamadreza Fazel, Keith A Lidke, Ralf Jungmann, Sjoerd Stallinga, and Bernd Rieger, "Template-free 2d particle fusion in localization microscopy," *Nature methods*, vol. 15, no. 10, pp. 781, 2018.

[2] Georgios Dimitrios Evangelidis and Radu Horaud, "Joint alignment of multiple point sets with batch and incremental expectation-maximization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 6, pp. 1397–1410, 2018.

[3] Xiao-Li Meng and Donald B Rubin, "Maximum likelihood estimation via the ecm algorithm: A general framework," *Biometrika*, vol. 80, no. 2, pp. 267–278, 1993.

[4] Arthur P Dempster, Nan M Laird, and Donald B Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, no. 1, pp. 1–22, 1977.

[5] Christopher M Bishop, *Pattern recognition and machine learning*, springer, 2006.

[6] Andrew Ng, "Cs229 lecture notes," *CS229 Lecture notes*, vol. 1, no. 1, pp. 1–3, 2000.

[7] Shinji Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, , no. 4, pp. 376–380, 1991.