

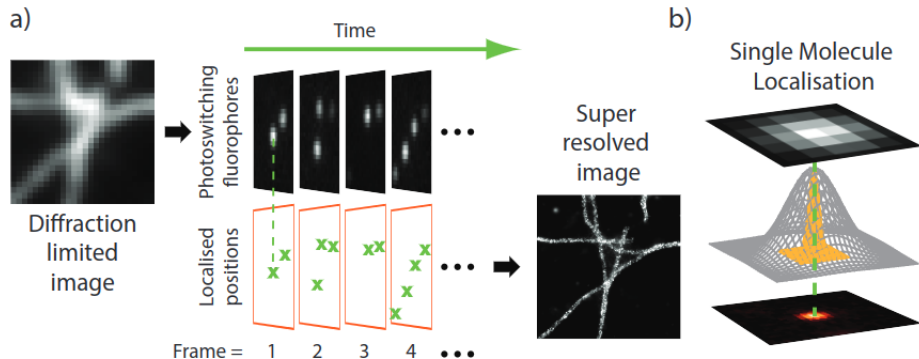
Single molecule localization microscopy

Clayton W. Seitz

April 17, 2023

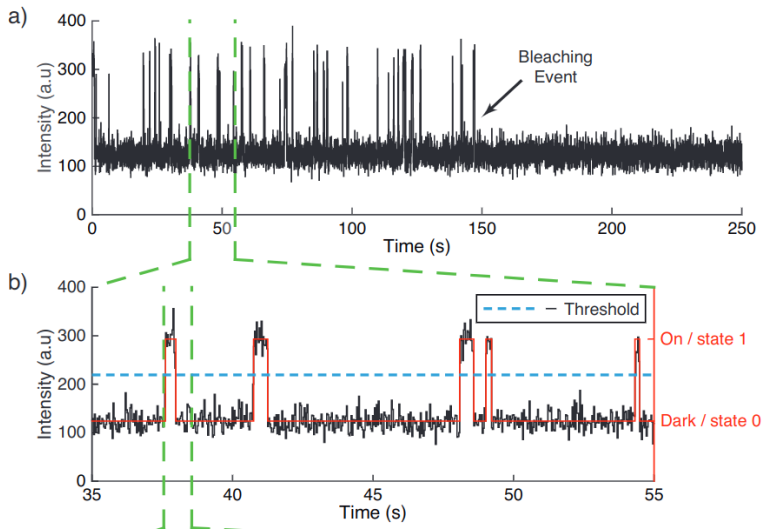
A (Very) Brief Summary of SMLM

General principle of single molecule localization microscopy (SMLM)

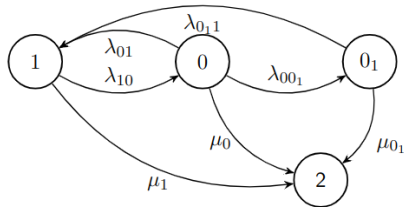
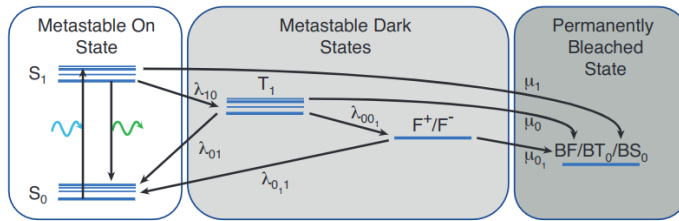


Patel et al. A hidden Markov model approach to characterizing the photo-switching behavior of fluorophores. Annals of Applied Statistics 2019

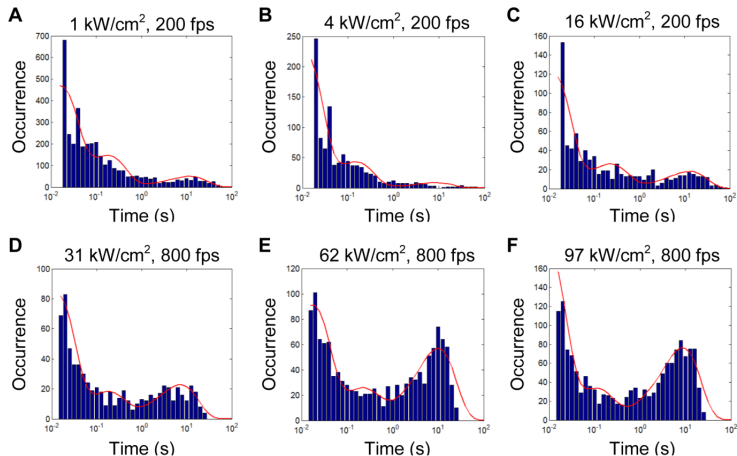
SMLM in the time domain



Photoswitching is a continuous time Markov process (CTMC)



The OFF state lifetime depends on laser power (Alexa-647, WFM)



Quantifying and Optimizing Single-Molecule Switching Nanoscopy at High Speeds

Resolution is dependent on photoswitching kinetics

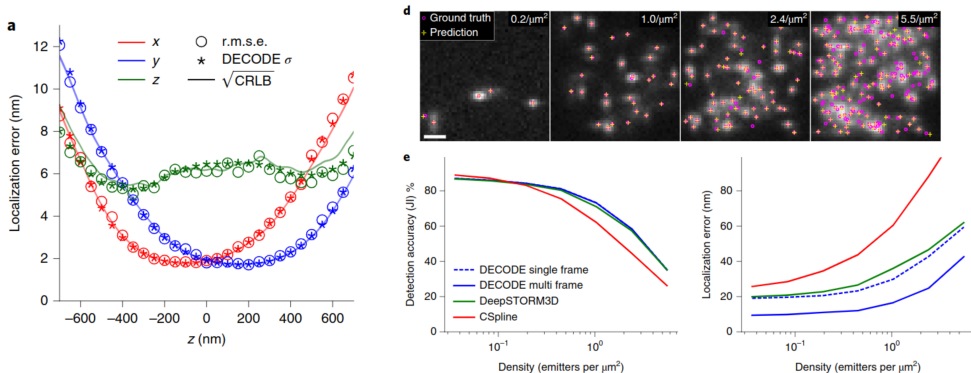
- ▶ Define ratio of OFF and ON state lifetimes as $r = \tau_{OFF}/\tau_{ON}$
- ▶ High values of r assure a low number of fluorophores in the ON state
- ▶ Low r values ensure precise localization, but acquisitions are slow
- ▶ Long acquisitions are undesirable due to sample drift

Params for a single molecule are $\theta = (x_0, y_0)$ What we call lateral “resolution” in

SMLM is the lowest possible uncertainty in θ , given a perfect estimator (The Cramer-Rao lower bound)

Current state of the art for dense localization in 3D (astigmatism)

DECODE uses a deep generative model to localize dense fluorophores



Deep learning enables fast and dense single-molecule localization with high accuracy

Mathematical Details

sCMOS noise model

The Poisson rate parameter for a single pixel is

$$\mu_k = \eta \Delta t (N_0 \lambda_k + B_0)$$

where Δt is the camera exposure time and N_0 and B_0 are the fluorophore and background emission rates respectively.

$$\lambda_k = \int_{\text{pixel}} G(x, y) dx dy$$

where the 2D function $G(x, y)$ is a normalized Gaussian density over the pixel array

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x-x_0)^2 + (y-y_0)^2}{2\sigma^2}}$$

How to compute λ_k at each pixel

We can replace this integral with error functions:

$$\lambda_x(x) = \frac{1}{2} \left(\operatorname{erf} \left(\frac{x + a/2 - x_0}{\sqrt{2}\sigma} \right) - \operatorname{erf} \left(\frac{x - a/2 - x_0}{\sqrt{2}\sigma} \right) \right)$$
$$\lambda_y(y) = \frac{1}{2} \left(\operatorname{erf} \left(\frac{y + a/2 - y_0}{\sqrt{2}\sigma} \right) - \operatorname{erf} \left(\frac{y - y/2 - y_0}{\sqrt{2}\sigma} \right) \right)$$

For multiple emitters

$$\lambda(x, y) = \sum_n \lambda_{n,x}(x) \lambda_{n,y}(y)$$

The *true signal* is then

$$\vec{S} = [\text{Poisson}(\lambda_1), \text{Poisson}(\lambda_2), \dots, \text{Poisson}(\lambda_N)]$$

Poisson approximation of pixel values

However, due to readout noise, we measure

$$\vec{H} = \vec{S} + \vec{\xi}$$

The distribution of H_k is the convolution:

$$\begin{aligned} P(H_k|\theta) &= P(S_k) \circledast P(\xi_k) \\ &= A \sum_{q=0}^{\infty} \frac{1}{q!} e^{-\mu_k} \mu_k^q \frac{1}{\sqrt{2\pi}\sigma_k} e^{-\frac{(H_k - g_k q - o_k)^2}{2\sigma_k^2}} \end{aligned}$$

where $P(\xi_k) = \mathcal{N}(o_k, \sigma_k^2)$ and $P(S_k) = \text{Poisson}(g_k \mu_k)$. In practice, this expression is difficult to work with, so we look for an approximation. Notice that

$$\xi_k - o_k + \sigma_k^2 \sim \mathcal{N}(\sigma_k^2, \sigma_k^2) \approx \text{Poisson}(\sigma_k^2)$$

The model log likelihood and Hessian matrix

Since $H_k = S_k + \xi_k$, we transform $H'_k = H_k - o_k + \sigma_k^2$, which is distributed according to

$$H'_k \sim \text{Poisson}(\mu'_k) \quad \mu'_k = g_k \mu_k + \sigma_k^2$$

Since each Poisson r.v. is independent, the negative log likelihood reads

$$\begin{aligned} \ell(\vec{H}) &= -\log \prod_k \frac{e^{-(\mu'_k)} (\mu'_k)^{n_k}}{n_k!} \\ &= \sum_k \log n_k! + \mu'_k - n_k \log (\mu'_k) \end{aligned}$$

The model log likelihood and Hessian matrix

Fortunately, we can compute the Hessian of the negative log likelihood using the chain-rule for Hessian matrices.

$$\hat{H}_{(\ell,\theta)} = \hat{J}_{(\lambda,\theta)}^T \hat{H}_{(\ell,\lambda)} \hat{J}_{(\lambda,\theta)} + (J_{(\ell,\lambda)} \otimes I_n) \hat{H}_{(\lambda,\theta)}$$

By calculating $\hat{H}_{(\ell,\theta)}$ at θ_{MLE} , we can get an estimate of the observed information matrix. To do that $J_{(\lambda,\theta)}$ and $H_{(\lambda,\theta)}$ are best left to symbolic calculators.