# Spatially Variable Deconvolution for Lightsheet Microscopy

**Bogdan Toader**[1], **Stéphane Chrétien**[2], **Andrew Thompson**[3]

[1]University of Cambridge, [2]University of Lyon 2, [3]National Physical Laboratory

UNIVERSITY OF CAMBRIDGE

## Lightsheet microscopy

TODO: a few words about why lightsheet is good
Light sheet microscopy is a type of fluorescence microscopy used in cell biology due to its fast acquisition times and low photo-damage to the sample. This is achieved by selectively illuminating a slice of the sample using a sheet of light and detecting the emitted fluorescence signal using a dedicated objective orthogonal to the plane of the sheet.

- Project aim: computationally improve the quality of light sheet microscopy images.

### Light-sheet microscope

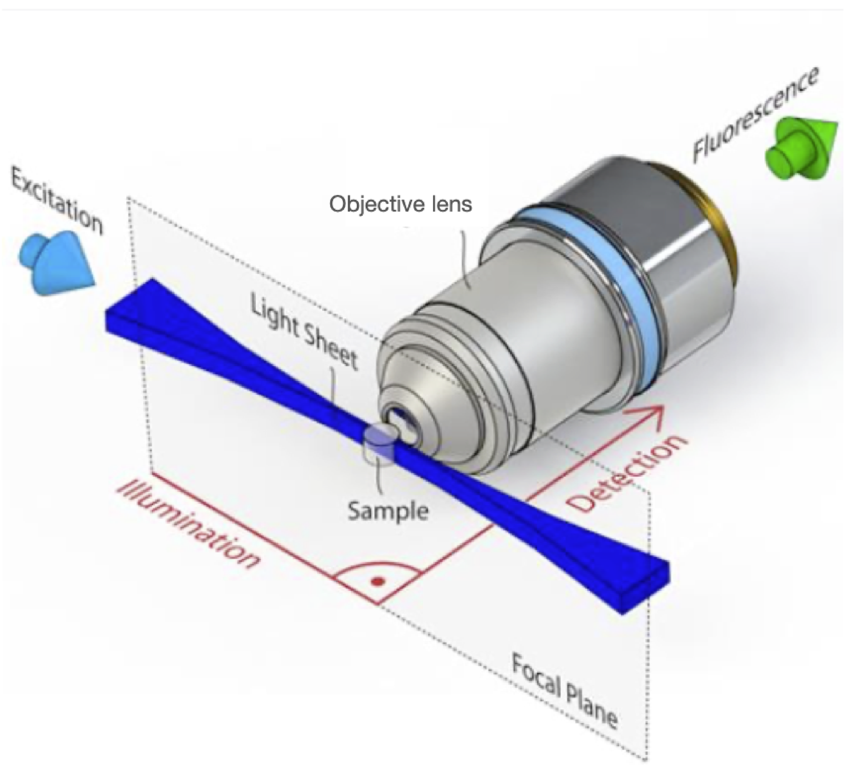
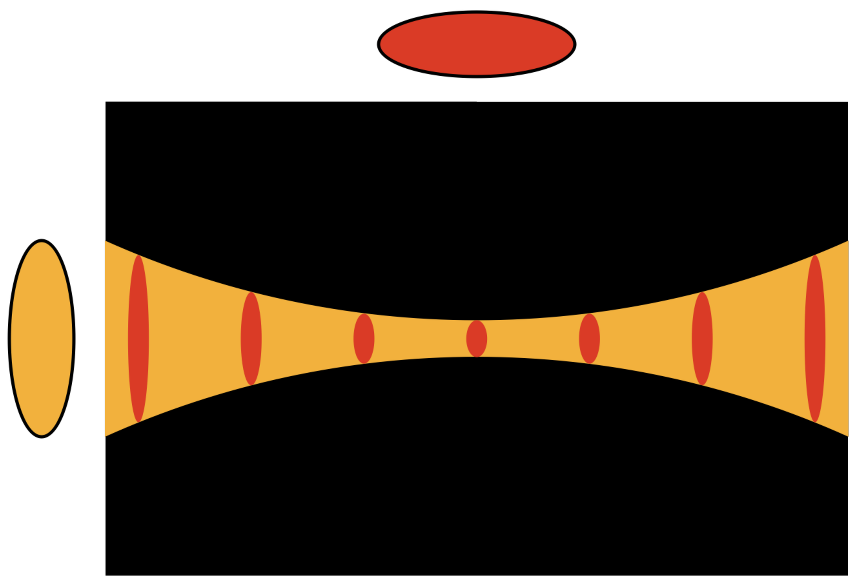



*Figure credit: Jörg Ritter, PhD thesis (2011)*

A light-sheet microscope consists of two objectives: TODO etc

### Spatially varying PSF



### Problem

Due to the combination of light-sheet and the objective PSF, the effective PSF of the system is spatially varying, and therefore standard deconvolution approaches are not applicable.

## Model

The measure $x$ can be recovered exactly by solving the TV norm minimisation problem over non-negative measures $z$ on $[0,1]$:

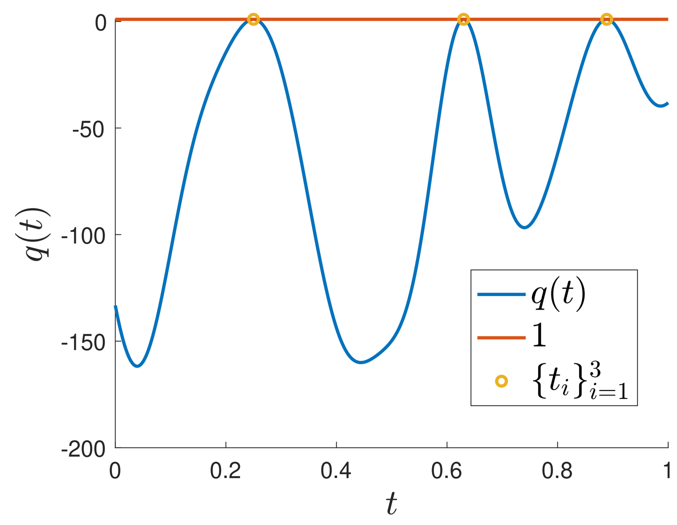
$$\min_{z\geq 0} \|z\|_{TV} \quad \text{subject to} \quad y_j = \int_{[0,1]} \phi(t-s_j)x(\mathrm{d}t), \quad \forall j = 1,\dots,m, \tag{1}$$

or its dual:

$$\max_{\lambda\in\mathbb{R}^m} y^T\lambda \quad \text{subject to} \quad \sum_{j=1}^{m} \lambda_j \phi(t-s_j) \leq 1, \quad \forall t\in[0,1]. \tag{2}$$

Having the solution $\lambda$ to the dual problem (2), the locations $\{t_i\}_{i=1}^k$ in the input signal are given by the global maximisers of the dual certificate:

$$q(t) = \sum_{j=1}^{m} \lambda_j \phi(t-s_j).$$




## Results – simulated data

## Fitting the PSF

### Algorithm

We solve $\min_{\lambda\in\mathbb{R}^m} f(\lambda)$ for the exact penalty $f$:

$$f(\lambda) = -y^T\lambda + \Pi \cdot \max\{\sup_s \sum_{j=1}^{m} \lambda_j\phi(s-s_j) - 1, 0\}$$

in the set $Q = \{\lambda \in \mathbb{R}^m \mid \|\lambda\|_\infty \leq B\}$. As described in [1], at every iteration $p$:

1. Compute the piece-wise linear model:

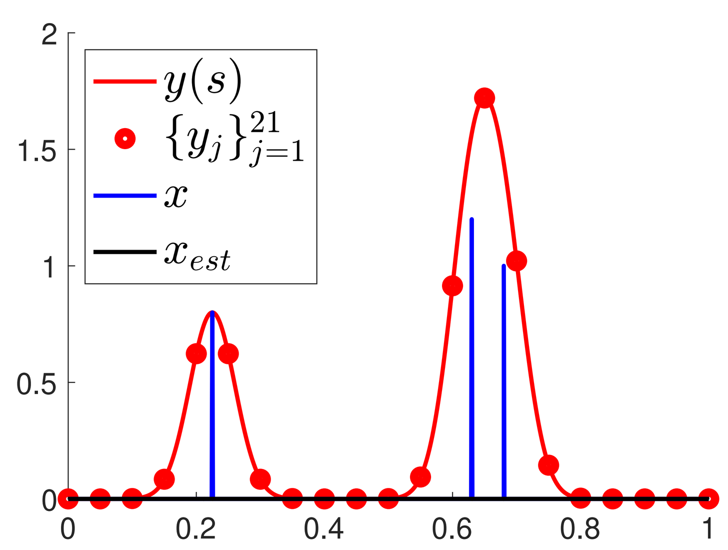
   $$\hat{f}_p(\lambda) = \max_{0\leq l\leq p} \left[ f(\lambda_l) + \langle g(\lambda_l), \lambda - \lambda_l\rangle \right],$$

   where $g(\lambda_l)$ are some subgradients of $f$ at $\lambda_l$.
2. Find the optimal value $\hat{f}_p^*$ of the current model $\hat{f}_p$ and the smallest value $f_p^*$ of the objective $f$ so far.
3. Calculate the next iterate $\lambda_{p+1}$ as the projection of $\lambda_p$ on the level set $\mathcal{L}_p(\alpha) = \{\lambda \in Q \mid \hat{f}_p(\lambda) \leq \ell_p(\alpha)\}$ where $\ell_p(\alpha) = (1-\alpha)\hat{f}_p^* + \alpha f_p^*$ for some $\alpha \in (0,1)$.

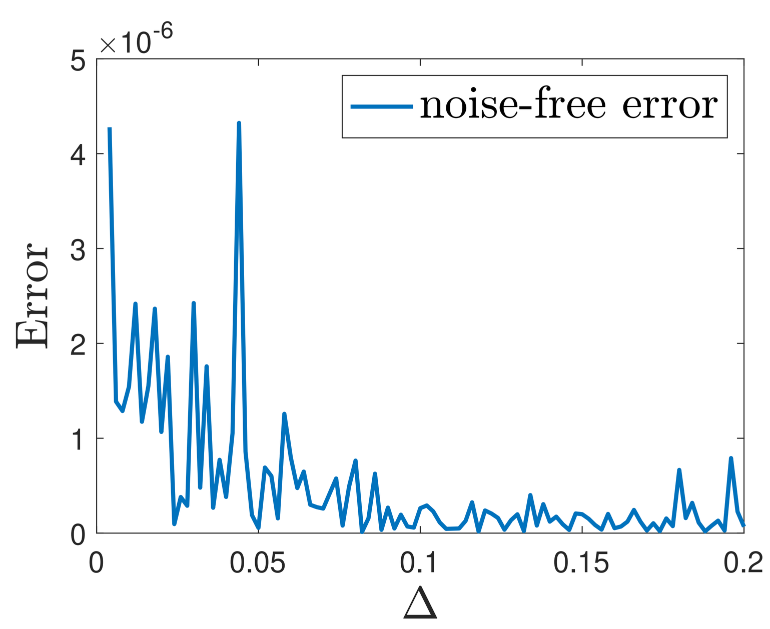
## Results – real data

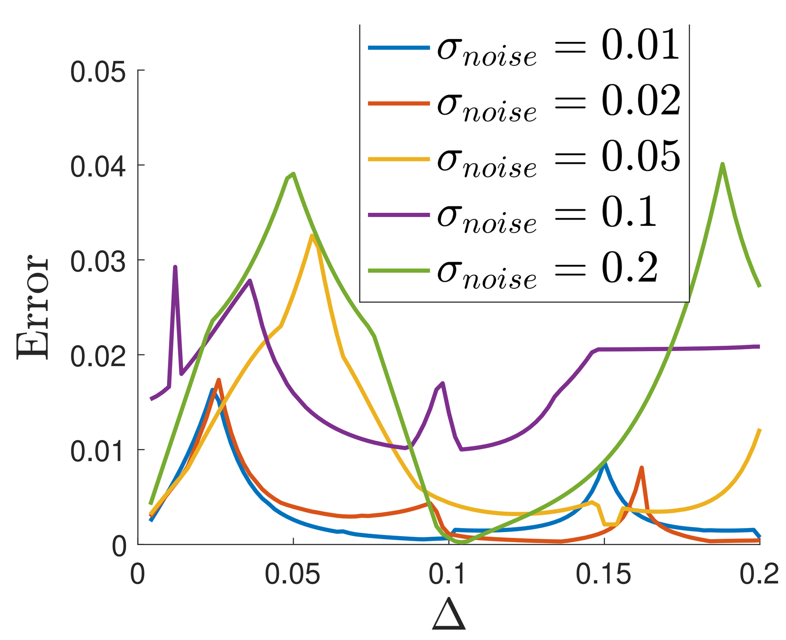
### Beads

- **Exact measurements:** the level method finds the correct locations of the point sources with high accuracy – see the localisation error for two sources as the distance between them goes to zero in the bottom left figure.
- **Noisy measurements:** As the distance between two sources decreases, they are replaced with one source with higher intensity in the reconstructed signal.




*Example of setup with noise-free measurements, solved to accuracy $10^{-6}$.*

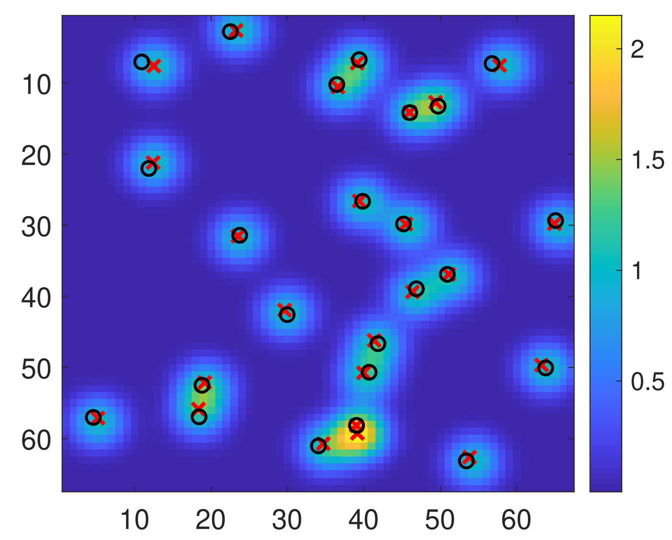


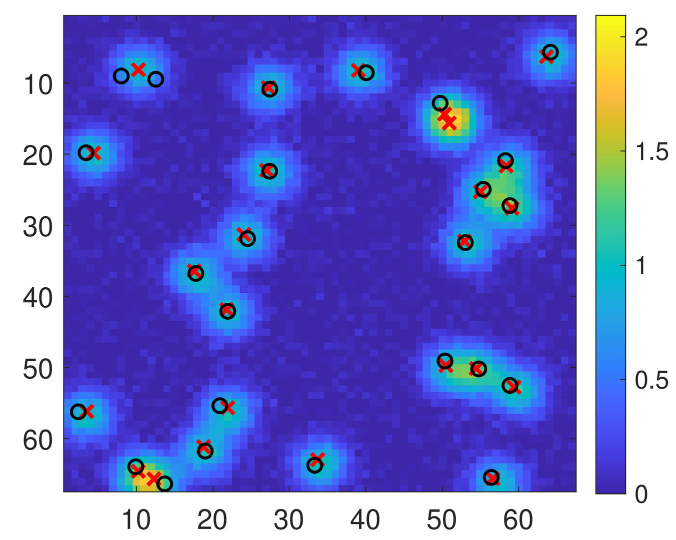

*Localisation error for two point sources as the distance $\Delta$ between them increases from 0.004 to 0.2 in the noise-free case (left) and noisy case for Gaussian noise with standard deviation 0.01, 0.02, 0.05, 0.1, 0.2 (right). The convolution kernel is Gaussian with $\sigma = 0.05$, intensities are 1 and $m = 21$ equispaced samples.*

### Marchantia

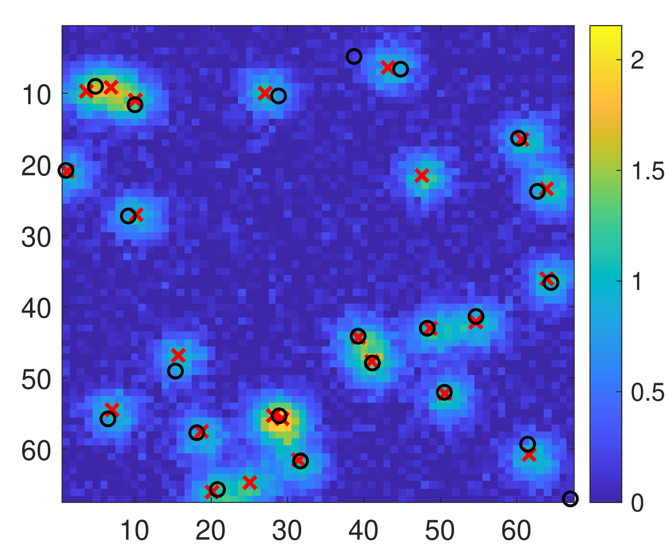
Examples of $k = 25$ sources in an image with $67 \times 67$ pixels (giving $m = 4489$ measurements), Gaussian kernel with $\sigma = 0.05$ and level method parameters $\Pi = 100$ and $B = 100$. Red crosses × are true sources and black circles ○ are estimated sources.



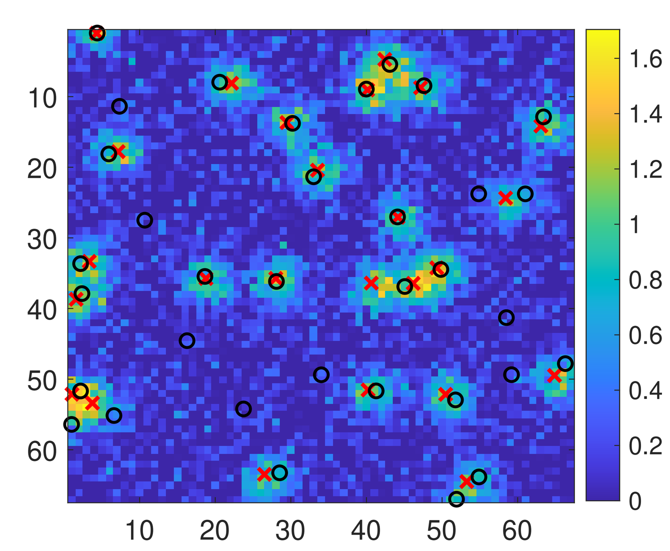
*Noise-free measurements.*



*Gaussian noise with standard deviation 0.05.*



*Gaussian noise with standard deviation 0.1.*



*Gaussian noise with standard deviation 0.2.*