Biomedical Optics EXPRESS

MusiJ: an ImageJ plugin for video nanoscopy

SEBASTIAN ACUÑA,[1,3] FLORIAN STRÖHL,[1,3] IDA S. OPSTAD,[1] BALPREET S. AHLUWALIA,[1,2] AND KRISHNA AGARWAL[1,*]

[1] *Department of Physics and Technology, UiT The Arctic University of Norway, NO-9037 Tromsø, Norway*
[2] *Department of Clinical Science, Intervention and Technology, Karolinska Insitute, 17177 Stockholm, Sweden*
[3] *These authors contributed equally*
[*] *krishna.agarwal@uit.no*

Abstract: We present an open-source implementation of the fluctuation-based nanoscopy method MUSICAL for ImageJ. This implementation improves the algorithm's computational efficiency and takes advantage of multi-threading to provide orders of magnitude faster reconstructions than the original MATLAB implementation. In addition, the plugin is capable of generating super-resolution videos from large stacks of time-lapse images via an interleaved reconstruction, thus enabling easy-to-use multi-color super-resolution imaging of dynamic systems.

1. Introduction

The past two decades have witnessed a huge development in nanoscopy techniques that allow to surpass the resolution limit of optical microscopy and thus provide *super-resolution*[1]. One way of classifying the broad range of those techniques is to distinguish all-optical, hybrid, and purely computational approaches. All-optical nanoscopy techniques manage to shrink the effective point spread function before detection and include stimulated emission depletion (STED) microscopy [2] or instant structured illumination microscopy (iSIM) [3]. Hybrid and purely computational techniques, in contrast, make use of temporal changes in the sample's fluorescent emission profile and extract additional information from time-series of raw frames of the same underlying sample structure. Such changes can be induced extrinsically via spatially varying illumination patterns [4,5] or intrinsically by exploiting fluorophore photokinetics that result in fluctuations in the measured fluorescence intensity [6,7].

Despite the often simplified optical setup of computational and hybrid nanoscopy techniques in comparison to all-optical ones, a lack of user-friendly and open-source implementations has often hindered fast integration of nanoscopy into biological research routines. A prime example for such a delay is the case of structured illumination microscopy (SIM). Its complex reconstruction algorithm was published in 2000, just to be implemented anew countless times in microscopy laboratories all over the world until the release of the easy-to-use FairSIM plugin [8], more than 15 years after the original publication. Further, as all nanoscopy techniques vary in their strengths and weaknesses, it is desirable to make as many different techniques available as possible, ideally in a single standard analysis environment. For microscopy, this environment is the image processing toolbox ImageJ [9] or its advanced version Fiji (*Fiji is just ImageJ*) [10].

Akin to the SIM reconstruction plugin FairSIM, single molecule localisation microscopy (SMLM) software has been made freely available by a vibrant community (for a comprehensive list see [11–13]). Similar to SIM, SMLM techniques can be regarded as hybrid nanoscopy methods due to the requirements of multiple high-power lasers and additional optical elements for field flattening [14] or when 3D information is desired [15]. A notable exception are 3D SMLM techniques with purely computational 3D information extraction based on aberrations in the microscope's point spread function (PSF) - these are also available in Fiji [16]. Despite their impressive resolution, SMLM reconstruction algorithms require data sets comprising thousands

#382735 https://doi.org/10.1364/BOE.382735
Journal © 2020 Received 26 Nov 2019; revised 20 Mar 2020; accepted 20 Mar 2020; published 14 Apr 2020

of raw frames with sparse single molecule blinking events, which renders live-cell, let alone time-lapse imaging challenging.

Approaches to extract sub-diffraction features from data sets with densely packed emitters and with well below thousand raw frames taken on conventional, non-specialized microscopes, can be grouped as fluctuation-based nanoscopy techniques. Fluctuation-based algorithms exploit small intensity variations in time-series via statistical analyses to generate better resolved images. The actual algorithm depends on the statistical approach, thus giving rise to various flavors of this idea. Examples include SOFI (super-resolution optical fluctuation imaging) [17], 3B (Bayesian analysis of blinking and bleaching) [18], ESI (entropy-based super-resolution imaging) [19], and SRRF (super-resolution radial fluctuations) [20]. All of the above mentioned algorithms (except SOFI) have been translated to ImageJ. The recently developed fluctuation-based method MUSICAL (multiple signal classification algorithm) [21], however, has not been translated to ImageJ until now.

MUSICAL's origin can be traced to MUSIC (multiple signal classification) first developed for direction of arrival measurements [22]. In MUSIC, the number of independent sources, often equal to the number of targets (aeroplanes or ships for example), is determined by the number of non-zero eigenvalues if the number of sources is less than the number of independent measurements taken at multiple time instances over an array of radar/sonar sensors. The multiple time instances provide multiple independent measurements from the sources and the problem of determining the direction of arrival is an inverse source problem. An indicator function is computed for each candidate point (also called test point) r_{test} in the target region by taking the reciprocal of a distance function d_n.

$$f(r_{\text{test}}) = \frac{1}{d_n(r_{\text{test}})} \tag{1}$$

The distance function d_n is the projection of the expected output vector for a test point onto the eigenvectors with zero eigenvalues. The presence and direction of arrival of the target is then indicated if the indicator function has a large value at a candidate target point. MUSIC has survived the test of times and is constantly being reinvented for modern applications such as cognitive radars [23] and inverse imaging in the microwave domain [24,25]. In the case of inverse imaging in the microwave domain, the full electromagnetic wave model of scattering applies and a multiple input multiple output system is used for taking measurements. Compared to the case of radars, the problem here is an inverse scattering problem and not an inverse source problem. This means the number of non-zero eigenvalues indicates the number of independent dipoles induced on the scatterers, as long as the number of transmitters and receivers is more than the number of induced dipoles. This number is equal to or more than the number of scatterers, depending upon the degrees of freedom for the induced dipole, which in turn depends upon isotropicity of the scatterers as well as the polarization constraints on the incident waves [24].

In the case of optical microscopy or nanoscopy, the measurements are comprised of fluorescent intensity detections on an optical detector array (i.e. the microscope's camera) at multiple time instances. At each time instance, the number of photons emitted by any fluorescent molecule is independent of the other molecules and follows a statistical distribution [26]. Therefore, the underlying problem is an inverse source problem like as in MUSIC. Nonetheless, adapting MUSIC to MUSICAL for optical nanoscopy is a non-trivial task and only a short intuitive understanding shall be given here without resorting to the detailed mathematical formulation.

Consider a small window around a given pixel of the size of the point spread function (PSF) of the microscope as the region of influence for the fluorescent emitters within the region of that pixel. Despite the main region of interest being within the pixel, the structure on which fluorophores are attached may extend beyond it. Therefore, it is important that the MUSIC indicator function is computed over the entire window. This is applicable to a first approximation. Nevertheless, as PSFs has not hard boundary, the PSF of fluorophores never lie completely within the window but will stretch outside. Similarly, the window may contain data from the trailing part of the PSFs of fluorophores completely outside the window. This non-reliability is suppressed in MUSICAL by using a soft window function on the measurement in the window as well as the PSF of candidate locations of emitters (i.e. the test points). This is equivalent to weighing the indicator function at a test point on the basis of distance from the center pixel of the window. Moreover, for stitching the reconstructions of all the windows, instead of using conventional image-processing techniques, MUSICAL takes a physics-based route in which an additional distance metric d_s in the numerator is the indicator function.

$$f(r_{\text{test}}) = \frac{d_s(r_{\text{test}})}{d_n(r_{\text{test}})} \tag{2}$$

The distance function d_s is the projection of the expected output vector for a candidate target point on the eigenvectors with non-zero eigenvalues. Introducing this distance is equivalent to stitching the reconstructed images based on the energy contributed by the test point in the numerical space of measurements.

In essence, MUSICAL identifies spatio-temporal patterns present in the image sequence through patch-wise singular value decomposition. A manually selected threshold then partitions the spatio-temporal patterns (i.e. eigenvectors) into two sets, 'signal' set Q_s that contains those eigenvectors whose corresponding singular value is larger than the threshold and 'noise' set Q_n containing the remaining eigenvectors. This is illustrated in Fig. 1). The final MUSICAL values are then computed as the ratio between sub-sampled image patches projected onto signal and noise vectors. A detailed derivation can be found in Appendix A and in the orginial MUSICAL publication [21].

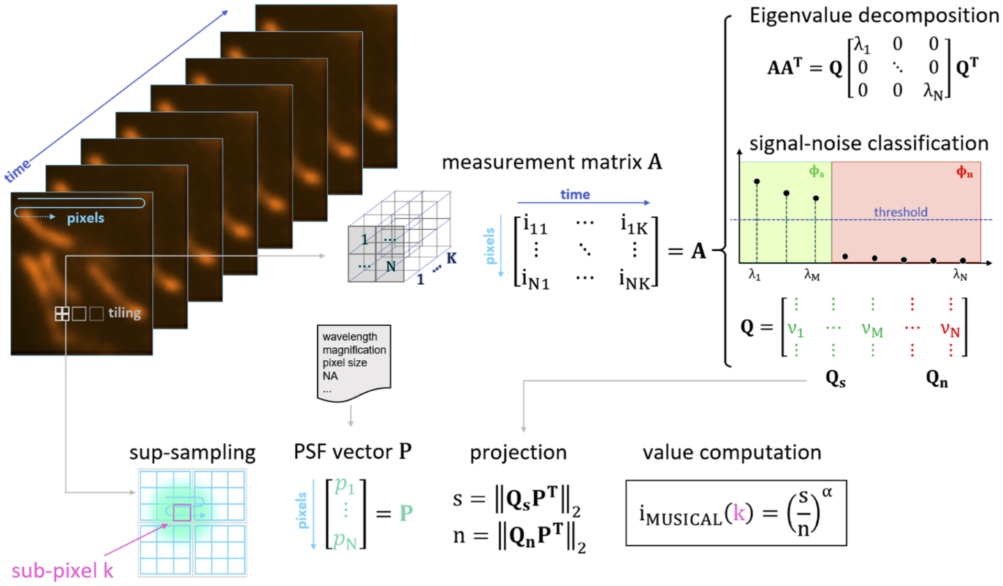

time
pixels
tiling
1
N
K
1
measurement matrix A
time
pixels
$\begin{bmatrix} i_{11} & \cdots & i_{1K} \\ \vdots & \ddots & \vdots \\ i_{N1} & \cdots & i_{NK} \end{bmatrix} = \mathbf{A}$
Eigenvalue decomposition
$\mathbf{AA^T} = \mathbf{Q}\begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_N \end{bmatrix}\mathbf{Q^T}$
signal-noise classification
ϕ_s
ϕ_n
threshold
λ_1
λ_M
λ_N
$\mathbf{Q} = [v_1 \cdots v_M \cdots v_N]$
$\mathbf{Q_s}$
$\mathbf{Q_n}$
wavelength
magnification
pixel size
NA
...
sup-sampling
PSF vector P
projection
value computation
pixels
$\begin{bmatrix} p_1 \\ \vdots \\ p_N \end{bmatrix} = \mathbf{P}$
$s = \|\mathbf{Q_s P^T}\|_2$
$n = \|\mathbf{Q_n P^T}\|_2$
$i_{\text{MUSICAL}}(k) = \left(\frac{s}{n}\right)^{\alpha}$
sub-pixel k

Fig. 1. Summary of the multiple signal classification algorithm (MUSICAL) [21].

MUSICAL was initially implemented in MATLAB with a focus on code-readability with respect to the mathematical background of the technique, rather than computational efficiency. Also, the MATLAB version provided only a rudimentary user interface with no extended capabilities for video generation or multi-color imaging. Hence, for a successful translation into a handy tool, we have developed MusiJ, a plugin for Fiji that improves both on the front end and back end of the original MATLAB implementation in several ways.

2. MusiJ: MUSICAL for ImageJ

2.1. Back end

MusiJ has three main differences compared to the previously released MATLAB version in terms of back end implementation of the algorithm. The most basic change is the data type. MATLAB uses by default double-precision floating-point format, defined by the IEEE Standard 754 [27] (named as *binary64* from 2008) which means that every value requires 64 bits of memory. In contrast, our implementation works with the single-precision floating-point or *binary32* format which halves the memory usage and speeds up individual computation steps. Although, in principle, this comes at the cost of numerical precision, we found no noticeable difference in image quality between the outputs generated by the two data types in practice. The second change is in the computation of MUSICAL image values i_{MUSICAL} (also called the indicator function) during image synthesis (see Fig. 1). In order to computer the values **s** and **n** for the indicator function i_{MUSICAL}, we perform eigenvalue decomposition to obtain **Q**. The columns of **Q** correspond to a basis with orthonormal columns. Thanks to the Pythagorean theorem, it is therefore sufficient to compute only one of them since the vector **P** is projected into the subspace spanned by $\mathbf{Q}_s$ and its orthogonal complement $\mathbf{Q}_n$. Hence, the following holds:

$$||\mathbf{P}||^2 = ||\mathbf{Q}_s\mathbf{P}^T||^2 + ||\mathbf{Q}_n\mathbf{P}^T||^2. \tag{3}$$

In practice, the cardinality of $\mathbf{Q}_s$ is significantly smaller than that of $\mathbf{Q}_n$. Moreover, the PSF vector **P** is purely defined by optical system parameters. Therefore, we redefine the indicator function in its equivalent form given in Eq. (4). This permits a reduction in the number of operations by computing the norm of **P** in advance.

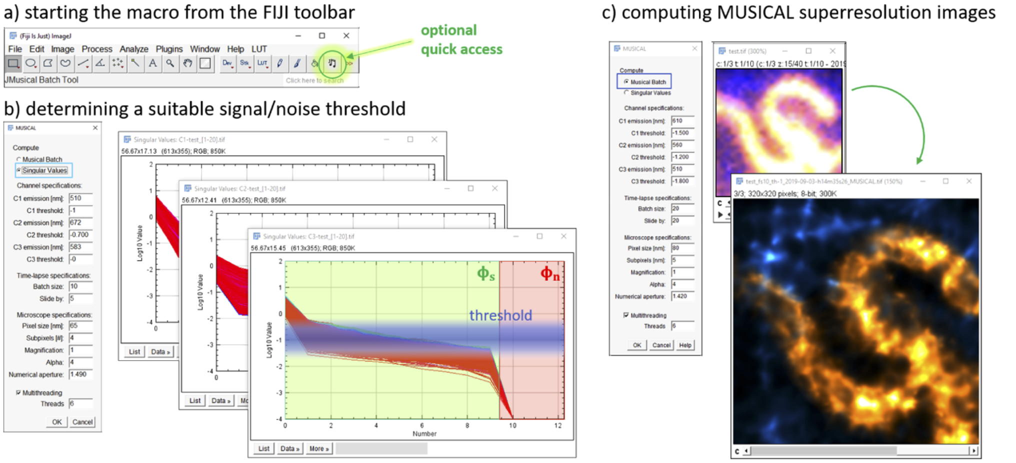
$$i_{\mathrm{MUSICAL}} = \left(\frac{\mathbf{s}^2}{||\mathbf{P}||^2 - \mathbf{s}^2}\right)^{\frac{\alpha}{2}}. \tag{4}$$

The final improvement is via multi-threading. As the image contains many non-overlapping regions, it is possible to process them in different threads of execution simultaneously, before merging the results into a single final image. This improvement is available as an option to the user, and the user may specify the number of used threads based on their system configuration and the load that the system may be experiencing due to other applications executing concurrently.

2.2. Front end

Along with changes in the computational efficiency in MusiJ, the developed plugin offers a range of features to simplify the usage and adds to MUSICAL's capabilities. The most prominent feature is the graphical user interface, GUI, which is shown in Fig. 2.

It provides easy access to the plugin's two main functions: (1) singular value computation and (2) MUSICAL image computation. Note that eigenvalues are the squares of singular values and thus equivalent for the purpose of signal/noise thresholding. In accordance with the original MUSICAL publication the plugin hence displays singular values. Additionally, a quick-access button can be added to the Fiji toolbar, which is especially convenient for heavy use. In the main GUI window, all necessary parameters can be filled in for MUSICAL image computation. It is possible and recommended to change the values stored as default in the accompanying MusiJ macro when the same parameters are in regular use, for example for repeat-experiments on the same microscope. This is to save time and to avoid typographical errors. The required thresholds to separate 'signal' from 'noise' (one per color channel) is estimated through visual inspection of the singular value plots and normally computed before the MUSICAL image generation. At the top of the GUI, the user may select among two options ('Singular Values' or 'Musical Batch'). The first option allows visual inspection and selection of the threshold value to separate signal


a) starting the macro from the FIJI toolbar
(Fiji Is Just) ImageJ
File Edit Image Process Analyze Plugins Window Help LUT
JMusical Batch Tool
optional quick access
b) determining a suitable signal/noise threshold
Singular Values: C1-test_[1-20].tif
Singular Values: C2-test_[1-20].tif
Singular Values: C3-test_[1-20].tif
Log10 Value
Number
Φ_s
Φ_n
threshold
List Data » More »
c) computing MUSICAL superresolution images
c: 1/3 t: 1/10
3/3; 320x320 pixels; 8-bit; 300K

MUSICAL

Compute
○ Musical Batch
● Singular Values
Channel specifications:
C1 emission [nm]: 510
C1 threshold: -1
C2 emission [nm]: 672
C2 threshold: -0.700
C3 emission [nm]: 583
C3 threshold: -0
Time-lapse specifications:
Batch size: 10
Slide by: 5
Microscope specifications:
Pixel size [nm]: 65
Subpixels [#]: 4
Magnification: 1
Alpha: 4
Numerical aperture: 1.490
☑ Multithreading
Threads 6
OK Cancel

MUSICAL

Compute
● Musical Batch
○ Singular Values
Channel specifications:
C1 emission [nm]: 610
C1 threshold: -1.500
C2 emission [nm]: 560
C2 threshold: -1.200
C3 emission [nm]: 510
C3 threshold: -1.800
Time-lapse specifications:
Batch size: 20
Slide by: 20
Microscope specifications:
Pixel size [nm]: 80
Subpixels [nm]: 5
Magnification: 1
Alpha: 4
Numerical aperture: 1.420
☑ Multithreading
Threads 6
OK Cancel Help

Fig. 2. The graphical user interface of MusiJ. (a) The plugin can be found in the regular 'Plugins' tab or directly launched via the optional quick-access button. (b) Using the raw data and microscope parameters as input, singular values can be computed that allow the user to find suitable thresholds for signal/noise classification. The blue region in the image above shows a region in which a suitable threshold is likely to lie. Note that each color channel has its own threshold. (c) With the determined thresholds, a nanoscopy image or a time-series can be computed from the raw data temporal stack. Here, mitochondria (orange) and microtubules (blue) are shown.

and noise. The other option allows batch processing of the entire data set for the pre-selected and specified threshold value. Under 'Time-lapse specifications', parameters for video generation can be set. For instance, if only a single image is to be reconstructed from the first 100 images of a much longer time sequence (e.g to optimize thresholds quickly), use 'Batch size' 100, and 'Slide by' a number larger than the remaining number of frames in the image stack subjected to analysis. To reconstruct super-resolved details and visualize the changes over time, use 'Slide by' equal to or smaller than the batch size. We call this feature *interleaved reconstruction*, and it allows for a time-resolution smaller than dictated by the total acquisition time of all frames used for image reconstruction. The maximum interleave of an image stack (at a significant increase in computation time) is achieved by using 'Slide by' 1. We do not recommend this as a starting point for MUSICAL video analysis. The 'Multithreading' option allows to choose how much of the computers resources to be made accessible for MUSICAL image computation. For fastest multi-thread reconstructions use 'Threads' equal to the number of CPU cores. In practice, if running MusiJ on an office computer, we recommend to not use all cores but spare some processing power for other applications to continue executing in the background. When all parameters are set, clicking on the 'OK' button generates a super-resolved MUSICAL image or time-lapse batch and saves it along with a log file of all parameters.

3. Results and discussion

A summary of the improvements upon the MATLAB version and new capabilities only available in MusiJ is provided in Table 1.

We tested both MATLAB and ImageJ implementations on a desktop computer running Windows 10, with an Intel Xeon Gold 5118 processor (12 physical cores) and 128 GB DDR4 RAM. The MATLAB version was obtained from the official MUSICAL website (https://sites.google.com/site/uthkrishth/musical) and executed using MATLAB version R2018b. MusiJ was tested using FIJI 2.0.0-rc-69 with ImageJ 1.52b. For algebraic operations, MusiJ relies

Table 1. Comparison of MUSICAL implementations for MATLAB and MusiJ.

	MATLAB	MusiJ
processing time	417.25 s	14.75 s
graphical user interface	(✓)	✓
multi-color capability	✗	✓
video capability	✗	✓
interleaved reconstruction	✗	✓

on Nd4j version 1.0.0-beta2 using CPU as back end, with MKL 2019.1 installed. In addition, we set the number of threads used by this library to 1 by setting the *Environment Variable OMP_NUM_THREADS* in our system. To test the speed of both implementations, we used a 256×256 pixel image stack with 50 frames and set the subpixel parameter to 10. For MusiJ we used an increasing number of threads from 1 to 8. The results are shown in the plot of Fig. 3 and a visual comparison of the generated MusiJ reconstruction to the MATLAB reconstruction is provided in Supplementary Figure S3.

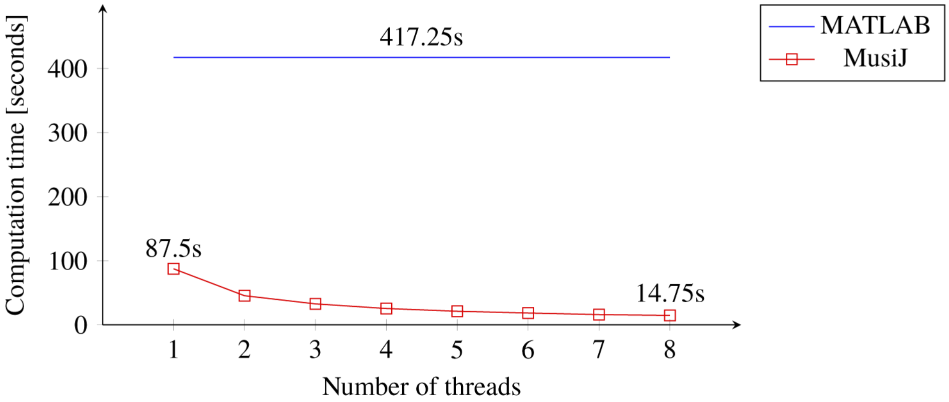


Fig. 3. Time used for a single reconstruction of an image of size 256×256 pixels, 50 frames, and subpixelation of 10. The results for MATLAB were replicated to make this comparison.

Due to the increased speed of the reconstruction process, a multitude of MUSICAL frames can be computed from a long time-series with overlapping raw-frames, termed as interleaved reconstruction. This is beneficial to enhance time-resolution when the imaged objects are changing their morphology or moving fast compared to the capture time of the entire raw frame series to determine the onset of events. The exact number of raw frames used for each MUSICAL time-point has to be adapted individually to the system dynamics. As can be seen in Fig. 4, interleaved reconstruction presents a trade-off between time and spatial resolution. Figure 4 illustrates the principle of interleaved reconstruction on exemplary time-lapse image data of mitochondria.

Many sources of signal fluctuations arise in living cells in addition to the intrinsic photokinetic fluctuations of fluorescent molecules that MUSICAL relies on. This is a challenge for threshold selection and when interpreting the results. Objects moving in and out of the imaged focal plane, or any other motion of the fluorescent emitters at nanometer scales, create signal fluctuations that are picked up by the algorithm. These different sources of signal fluctuations can be a potential source of misinterpretation. Trying different thresholds and cross-checking the reconstruction results with the system dynamics visible in the raw data is thus helpful and necessary to reach interpretations consistent with both raw data and MUSICAL reconstruction. On our data, a

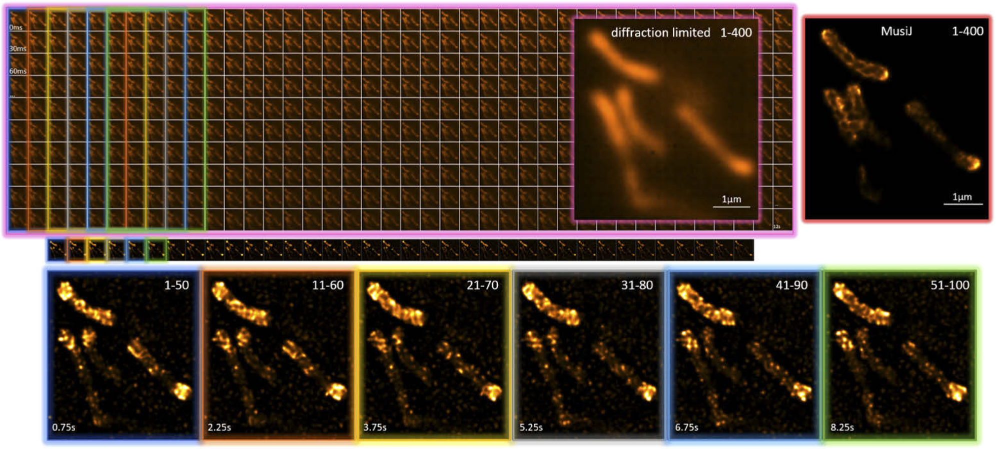


Fig. 4. Using MusiJ's interleaved reconstruction feature, dynamics of sub-cellular organelles like mitochondria can be visualised with super-resolution. In this example, a total of 400 time frames (30ms raw frame exposure time) were reconstructed into the MUSICAL image in the rightmost panel, while batches of 50 images with a 10 frame 'Slide-by' were converted into the time series displayed underneath. In the bottom six panels, zooms of the first few frames are shown. Labelled is the inner membrane of mitochondria in a cardiomyoblast cell line using dsRed. The imaging set-up was a commercial OMX microscope in widefield mode with LED illumination. A 60× 1.42NA oil immersion objective was equipped.

suitable threshold for most samples was found to be in the mid-range of the 2nd singular values (around the first *elbow* visible in the singular value plot of Fig. 2). For data with strong signal and low background fluctuations, the threshold can be set even lower to include more information in the computation for enhanced resolution. Figures S1 and S2 contain a visual comparison of images generated with different thresholds.

4. Conclusion

We have presented a user-friendly implementation of the fluctuation-based super-resolution algorithm MUSICAL for ImageJ/Fiji with a significant speed-up by a factor of almost 30 compared to the previous MATLAB version. The plugin can be kept up-to-date automatically via Fiji's update site. A step-by-step tutorial for installation and usage can be found at github.com/sebsacuna/MusiJ. Fluctuation-based video nanoscopy is an advancing field, but requires further experimentation and computational speed-up for increased understanding and usability of these techniques. Hence, this plugin was created with the objective of advancing the availability and usability of computational live-cell friendly super-resolution methods.

Appendix A: Mathematical background of MUSICAL

MUSICAL is an algorithm that allows to obtain super-resolution from a short (<100) sequence of frames. Here, a brief mathematical background is presented.

For a sensor with M pixels and a sample composed of N emitters, and under the assumption that emitters' locations do not change with time, the imaging model can be approximated as the matrix-vector multiplication shown in Eq. (5). This model is generalizable to moving emitters by making a hypothetical list of emitters, which take unique positions along the motion trajectory of the emitter. One hypothetical emitter is then modeled as having zero emissions at all other times except at the time when the real emitter is at the location of the hypothetical emitter.

$$\bar{I}(t) = \begin{bmatrix} G(\vec{r}^{(1)}_{em},\vec{r}^{(1)}_{im}) & \dots & G(\vec{r}^{(N)}_{em},\vec{r}^{(1)}_{im}) \\ \vdots & \ddots & \vdots \\ G(\vec{r}^{(1)}_{em},\vec{r}^{(M)}_{im}) & \dots & G(\vec{r}^{(N)}_{em},\vec{r}^{(M)}_{im}) \end{bmatrix} \begin{bmatrix} e_1(t) \\ \vdots \\ e_N(t) \end{bmatrix} \tag{5}$$

This model defines the acquired image in time $\bar{I}(t)$ as a column vector where each element correspond to the intensity value for every pixel. The matrix that contains the values obtained from the mapping function $G(\vec{r}_{em}, \vec{r}_{im})$ will be referred to as (G). The function $G(\vec{r}_{em}, \vec{r}_{im})$ maps the intensity produced by an emitter located at $\vec{r}_{em}$ to the pixel located in $\vec{r}_{im}$ using the known point spread function (PSF) of the system. Finally, $e_i(t)$ corresponds to the brightness of emitter i during time t. Note that each image is obtained then as a linear combination of the columns of $\mathbf{G}$ which is not time dependent.

Let's consider now a sequence of K image vectors to form the matrix $\mathbf{I}$ and corresponding Singular Value Decomposition (SVD) shown in Eq. (6). This allow us to generate an orthonormal basis for $M-$ dimensional space of real numbers $\Re^M$ given by the columns of $\mathbf{U}$.

$$\mathbf{I} = \mathbf{USV}^T \tag{6}$$

Equation (5) and Eq (6) are two fundamental relations used by MUSICAL. The simplest case is when the number of emitters is less than the number of pixels ($N<M$), and assuming $M<K$. In this case, $\mathbf{G}$ has N columns, meaning that its rank can be at most N. These columns span a subspace of $\Re^M$, and this is what we will call the signal space, corresponding to all the images that a set of N emitters can produce. Another implication is that the rank of $\mathbf{I}$ is equal to the rank of $\mathbf{G}$, which means that there must be N non-zero singular values. The vectors associated to these singular values then, must span the same subspace as $\mathbf{G}$. Alternatively, the subspace associated to the vectors with singular value zero, referred to as the null space, is orthogonal to the signal space. In this scenario, we can test if a point $\bar{r}_s$ belongs to the set of emitters by evaluating the expression shown in Eq. (7).

$$\bar{G}(\vec{r}_s) \cdot \bar{u}_{\sigma=0} = \begin{cases} 0 & \text{if an emitter is present at} \bar{r}_s \\ \text{non-zero} & \text{if no emitter is present at} \bar{r}_s \end{cases} \tag{7}$$

In reality, noise coming from undesired emission in the sample, shot-noise, and electronics is present in the images. Due to these factors, the singular values are unlikely to be zero. In order to split the space into signal and null space, a threshold σ_0 is given by the user. The final function used by MUSICAL is given by Eq. (8).

$$f(\bar{r}_{test}) = \left(\frac{\sqrt{\sum_{\sigma<\sigma_0} ||G(\bar{r}_{test}) \cdot \bar{u}_i||^2}}{\sqrt{\sum_{\sigma\geq\sigma_0} ||G(\bar{r}_{test}) \cdot \bar{u}_i||^2}} \right)^{\alpha} \tag{8}$$

Appendix B: Effect of hyper-parameters

MusiJ works over a stack of images, following a workflow similar to SRRF. Figure 5 presents a series of reconstructed images from the same source file comparing MUSICAL and SRRF. The sample used as example was cardiomyoblast cells with labelled mitochondria and is available in [28]. Since the sample presented significant motion only 50 frames were used for reconstructions. The parameters relevant for a MUSICAL reconstruction were as follows:

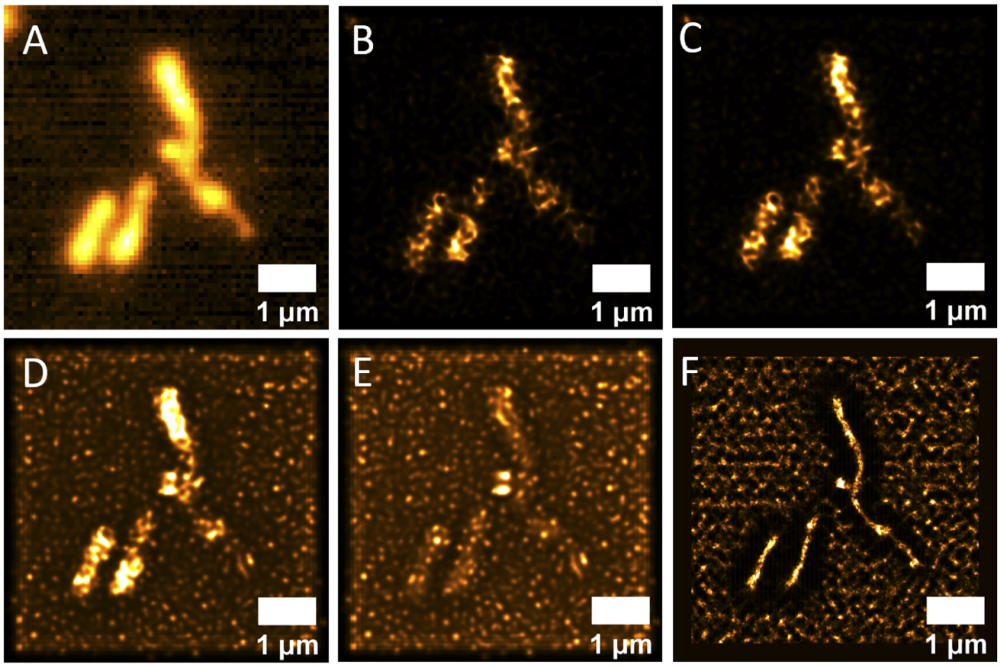


Fig. 5. A comparison between fluctuation images of the same raw data generated by MusiJ and SRRF. (A) average of 50 frames. (B-E) are MUSICAL reconstructions using MusiJ with thresholds -1.0, -0.75, -0.5 and -0.25. (F) correspond to a reconstruction made with SRRF.

- Emission wavelength of the fluorophore: 525 nm
- Numerical aperture: 1.42
- Magnification: 1× (as pixel size was scaled directly)
- Pixel size: 80 nm
- Exposure time per frame: 30 ms
- Number of raw image frames: 50

The parameter for alpha was set to 4, and the subpixelation to 10. In the case of SRRF the method picked was TRAC with all parameters set to default. Figure 5 address the importance of the threshold in the quality of the reconstruction. Note that as the threshold increases the relation between background and foreground get diffused. The corresponding SRRF image shows as characteristic property that all features are presented as uniform lines. This discrepancy between fluctuation techniques is known and a current topic of research. Figure 6 shows the parameters used for MUSICAL reconstructions in Fig. 5. This figure uses the plotting tool included in MusiJ.

Fig. 7 presents a comparison between the previous MATLAB implementation and MusiJ. Note that, by default, MATLAB uses 64 bits as data type, but the final printed result is presented as a PNG image of 8bits. It is hence less than the resolution used internally. Another minor difference in the implementation is the size of the sliding window. In MATLAB, the window size (in pixels) is always matched to exactly an airy disk, while in MusiJ the minimum size is 7 pixels. Hence, whenever the computed size is less than 7, the sliding window adds additional content. Nevertheless, no significant difference can be seen between the implementations.

Biomedical Optics EXPRESS

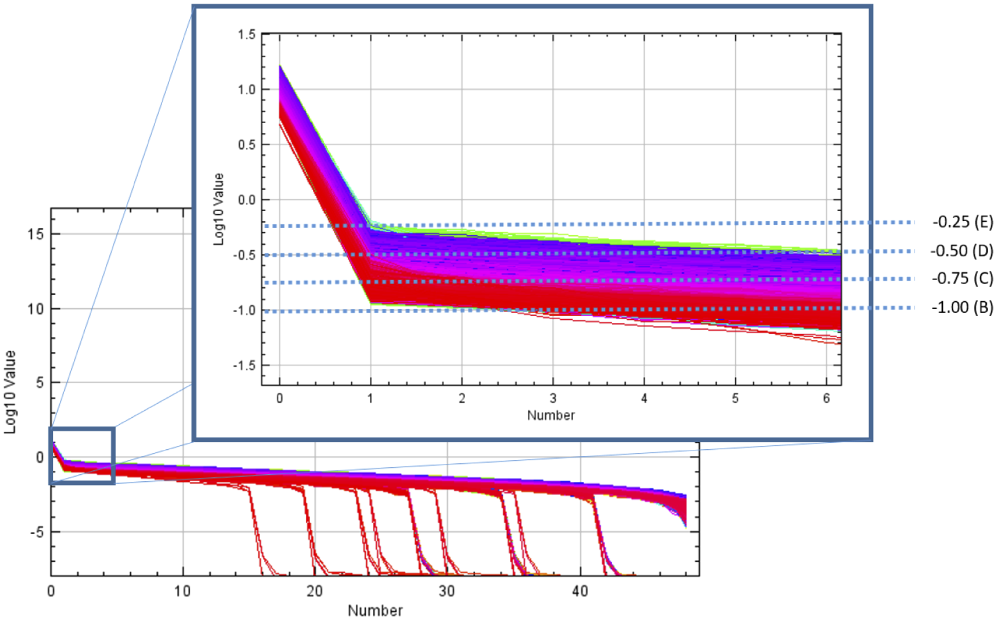


Fig. 6. The singular values for each sliding window in the cardiomyoblast sample shown in Fig. 5. This was obtained using the tool included in MusiJ. The horizontal axis corresponds to the index of the singular value. The dashed lines correspond to the threshold used for every reconstruction.

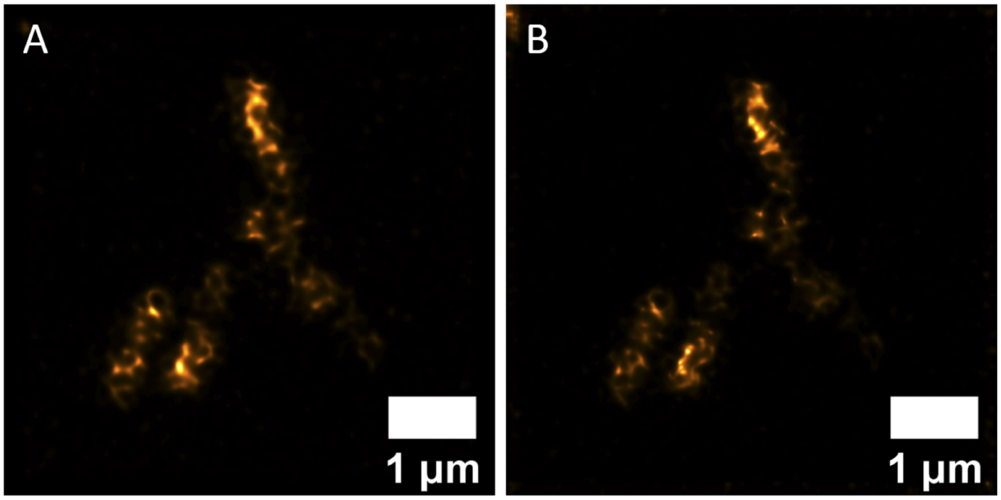


Fig. 7. A reconstructed nanoscopy image as produced by the previous (A) Matlab implementation and (B) MusiJ. A threshold of -1.0 was used in both versions of MUSICAL. The other parameters were the same as used in Fig. 5.

Funding

H2020 Marie Skłodowska-Curie Actions (836355, SEP-210382872); Norges Forskningsråd (285571); H2020 European Research Council (336716, 804233).

Acknowledgements

FS acknowledges funding from a Horizon 2020 Marie Skłodowska-Curie Action (836355). KA acknowledges funding from a Horizon 2020 Marie Skłodowska-Curie Action (SEP-210382872) and a Horizon 2020 ERC Starting Grant (804233). SA also acknowledges funding from the Horizon 2020 Marie Skłodowska-Curie Action (SEP-210382872). BSA acknowledges funding from a FP7 ERC Starting Grant (336716) and RCN BioTek2021 (285571). The authors thank Åsa Birgisdottir for kind provision of cultured cells.

KA conceived the project. SA developed the eigenvalue-based version of MUSCIAL images and implemented the back end of the ImageJ plugin. ISO prepared samples for microscopy, acquired data for MUSICAL and tested the plugin on a broad range of multi-color live-cell data for different reconstruction parameters. ISO provided feedback on necessary functionalities of the plugin and ease of use. FS and ISO designed the front end of the plugin. FS wrote the front end wrapper macro. FS, SA, ISO, and KA wrote the manuscript. BSA provided guidance and research tools. All authors commented on the manuscript.

Disclosures

The computational enhancements in the back end are a part of the invention disclosure 2019/3595 claimed by UiT The Arctic University of Norway. This part of the invention disclosure is not intended for commercialization and does not conflict with the commercialization of the remaining part. The authors declare that there are no conflicts of interest related to this article.

References

1. L. Schermelleh, A. Ferrand, T. Huser, C. Eggeling, M. Sauer, O. Biehlmaier, and G. P. Drummen, “Super-resolution microscopy demystified,” Nat. Cell Biol. **21**(1), 72–84 (2019).
2. S. W. Hell and J. Wichmann, “Breaking the diffraction resolution limit by stimulated emission: stimulated-emission-depletion fluorescence microscopy,” Opt. Lett. **19**(11), 780–782 (1994).
3. A. G. York, P. Chandris, D. Dalle Nogare, J. Head, P. Wawrzusin, R. S. Fischer, A. Chitnis, and H. Shroff, “Instant super-resolution imaging in live cells and embryos via analog image processing,” Nat. Methods **10**(11), 1122–1126 (2013).
4. M. G. Gustafsson, “Surpassing the lateral resolution limit by a factor of two using structured illumination microscopy,” J. Microsc. **198**(2), 82–87 (2000).
5. E. Mudry, K. Belkebir, J. Girard, J. Savatier, E. Le Moal, C. Nicoletti, M. Allain, and A. Sentenac, “Structured illumination microscopy using unknown speckle patterns,” Nat. Photonics **6**(5), 312–315 (2012).
6. M. J. Rust, M. Bates, and X. Zhuang, “Sub-diffraction-limit imaging by stochastic optical reconstruction microscopy (STORM),” Nat. Methods **3**(10), 793–796 (2006).
7. T. Dertinger, R. Colyer, G. Iyer, S. Weiss, and J. Enderlein, “Fast, background-free, 3D super-resolution optical fluctuation imaging (SOFI),” Proc. Natl. Acad. Sci. **106**(52), 22287–22292 (2009).
8. M. Müller, V. Mönkemöller, S. Hennig, W. Hübner, and T. Huser, “Open-source image reconstruction of super-resolution structured illumination microscopy data in ImageJ,” Nat. Commun. **7**(1), 10980 (2016).
9. C. A. Schneider, W. S. Rasband, and K. W. Eliceiri, “NIH image to ImageJ: 25 years of image analysis,” Nat. Methods **9**(7), 671–675 (2012).
10. J. Schindelin, I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, S. Preibisch, C. Rueden, S. Saalfeld, B. Schmid, J.-Y. Tinevez, D. J. White, V. Hartenstein, K. Eliceiri, P. Tomancak, and A. Cardona, “Fiji: an open-source platform for biological-image analysis,” Nat. Methods **9**(7), 676–682 (2012).
11. D. Sage, H. Kirshner, T. Pengo, N. Stuurman, J. Min, S. Manley, and M. Unser, “Quantitative evaluation of software packages for single-molecule localization microscopy,” Nat. Methods **12**(8), 717–724 (2015).
12. D. Sage, T. A. Pham, H. Babcock, T. Lukes, T. Pengo, J. Chao, R. Velmurugan, A. Herbert, A. Agrawal, S. Colabrese, A. Wheeler, A. Archetti, B. Rieger, R. Ober, G. M. Hagen, J. B. Sibarita, J. Ries, R. Henriques, M. Unser, and S. Holden, “Super-resolution fight club: assessment of 2D and 3D single-molecule localization microscopy software,” Nat. Methods **16**(5), 387–395 (2019).
13. S. van de Linde, “Single-molecule localization microscopy analysis with ImageJ,” J. Phys. D: Appl. Phys. **52**(20), 203002 (2019).
14. C. J. Rowlands, F. Ströhl, P. P. V. Ramirez, K. M. Scherer, and C. F. Kaminski, “Flat-field super-resolution localization microscopy with a low-cost refractive beam-shaping element,” Sci. Rep. **8**(1), 5630 (2018).
15. H. Deschout, F. C. Zanacchi, M. Mlodzianoski, A. Diaspro, J. Bewersdorf, S. T. Hess, and K. Braeckmans, “Precisely and accurately localizing single emitters in fluorescence microscopy,” Nat. Methods **11**(3), 253–266 (2014).

16. C. Franke, M. Sauer, and S. van de Linde, “Photometry unlocks 3D information from 2D localization microscopy data,” Nat. Methods **14**(1), 41–44 (2017).
17. T. Dertinger, R. Colyer, R. Vogel, M. Heilemann, M. Sauer, J. Enderlein, and S. Weiss, “Superresolution optical fluctuation imaging (SOFI),” in *Nano-Biotechnology for Biomedical and Diagnostic Research*, (Springer, 2012), pp. 17–21.
18. S. Cox, E. Rosten, J. Monypenny, T. Jovanovic-Talisman, D. T. Burnette, J. Lippincott-Schwartz, G. E. Jones, and R. Heintzmann, “Bayesian localization microscopy reveals nanoscale podosome dynamics,” Nat. Methods **9**(2), 195–200 (2012).
19. I. Yahiatene, S. Hennig, M. Müller, and T. Huser, “Entropy-based super-resolution imaging (ESI): From disorder to fine detail,” ACS Photonics **2**(8), 1049–1056 (2015).
20. N. Gustafsson, S. Culley, G. Ashdown, D. M. Owen, P. M. Pereira, and R. Henriques, “Fast live-cell conventional fluorophore nanoscopy with ImageJ through super-resolution radial fluctuations,” Nat. Commun. **7**(1), 12471 (2016).
21. K. Agarwal and R. Macháň, “Multiple signal classification algorithm for super-resolution fluorescence microscopy,” Nat. Commun. **7**(1), 13752 (2016).
22. R. Schmidt, “Multiple emitter location and signal parameter estimation,” IEEE Trans. Antennas Propag. **34**(3), 276–280 (1986).
23. A. M. Elbir, “Deepmusic: Multiple signal classification via deep learning,” arXiv pp. arXiv–1912 (2019).
24. K. Agarwal and X. Chen, “Applicability of music-type imaging in two-dimensional electromagnetic inverse problems,” IEEE Trans. Antennas Propag. **56**(10), 3217–3223 (2008).
25. K. Agarwal, L. Pan, Y. K. Leong, M. Han, O. Y. Chan, X. Chen, and S. P. Yeo, “Practical applications of multiple signal classification,” Int. J. RF Microw. Comput. Eng. **22**(3), 359–369 (2012).
26. G. T. Dempsey, J. C. Vaughan, K. H. Chen, M. Bates, and X. Zhuang, “Evaluation of fluorophores for optimal performance in localization-based super-resolution imaging,” Nat. Methods **8**(12), 1027–1036 (2011).
27. D. Zuras, M. Cowlishaw, A. Aiken, M. Applegate, D. Bailey, S. Bass, D. Bhandarkar, M. Bhat, D. Bindel, and S. Boldo, “IEEE standard for floating-point arithmetic,” IEEE Std **754**, 1–70 (2008).
28. S. Acuña and I. Opstad, “MusiJ: example of results,” https://www.doi.org/10.6084/m9.figshare.12006615.v3 (2020). Accessed: 2020-03-20.