Efficient Bayesian-based Multi-View Deconvolution

Stephan Preibisch, Fernando Amat, Evangelia Stamataki,
Mihail Sarov, Robert H. Singer, Gene Myers and Pavel Tomancak

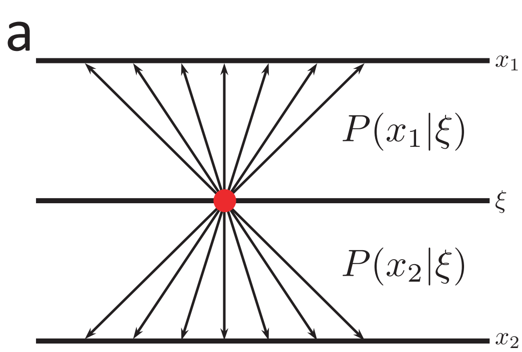
Nature Methods: doi:10.1038/nmeth.2929

Note: Supplementary Videos 1–11 are available for download on the Nature Methods homepage.

Nature Methods: doi:10.1038/nmeth.2929

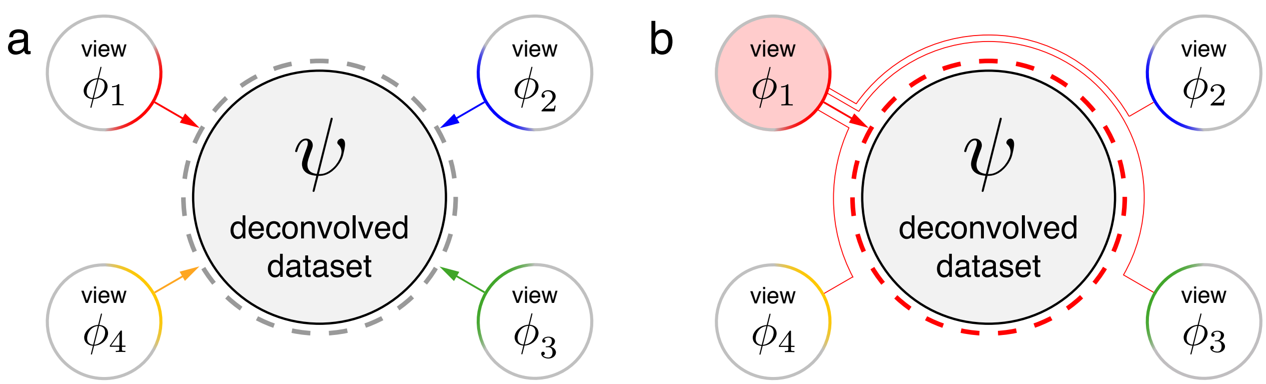
SUPPLEMENTARY FIGURES

SUPPLEMENTARY FIGURE 1: Illustration of conditional probabilities describing the dependencies of two views



Supplementary Figure 1: *Illustration of conditional probabilities describing the dependencies of two views.* (**a**) illustrates the conditional independence of two observed distributions $\phi_1(x_1)$ and $\phi_2(x_2)$ if it is known that the event $\xi = \xi'$ on the underlying distribution $\psi(\xi)$ occured. Given $\xi = \xi'$, both distributions are conditionally independent, the probability where to expect an observation only depends on $\xi = \xi'$ and the respective individual point spread function $P(x_1|\xi)$ and $P(x_2|\xi)$, i.e. $P(x_1|\xi, x_2) = P(x_1|\xi)$ and $P(x_2|\xi, x_1) = P(x_2|\xi)$. (**b**) illustrates the relationship between an observed distribution $\phi_2(x_2)$ and $\phi_1(x_1)$ if the event $x_1 = x_1'$ occured. Solely the 'inverse' point spread function $Q(\xi|x_1)$ defines the probability for any event $\xi = \xi'$ to have caused the observation $x_1 = x_1'$. The point spread function $P(x_2|\xi)$ consecutively defines the probability where to expect a corresponding observation $x_2 = x_2'$ given the probability distribution $\psi(\xi)$.

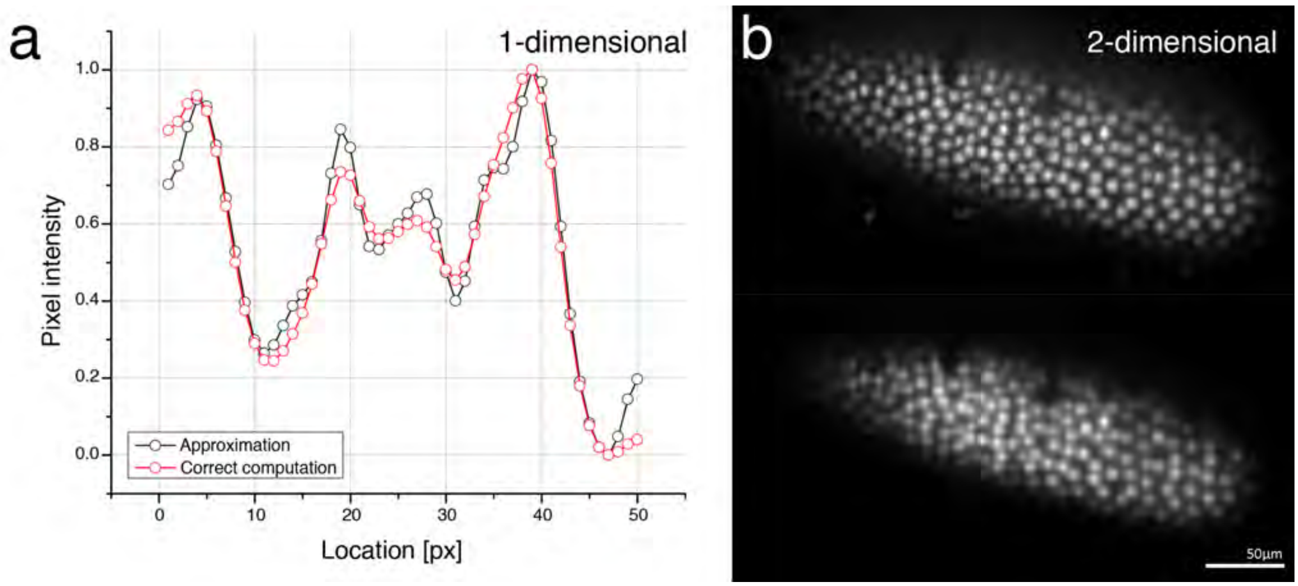
SUPPLEMENTARY FIGURE 2: The principle of 'virtual' views and sequential updating



Supplementary Figure 2: *The principle of 'virtual' views and sequential updating.* (**a**) The classical multi-view deconvolution[1–4] where an update step is computed individually for each view and subsequently combined into one update of the deconvolved image. (**b**) Our new derivation considering conditional probabilities between views. Each individual update step takes into account all other views using virtual views and additionally updates the deconvolved image individually, i.e. updates are performed sequentially[5] and not combined.

Nature Methods: doi:10.1038/nmeth.2929

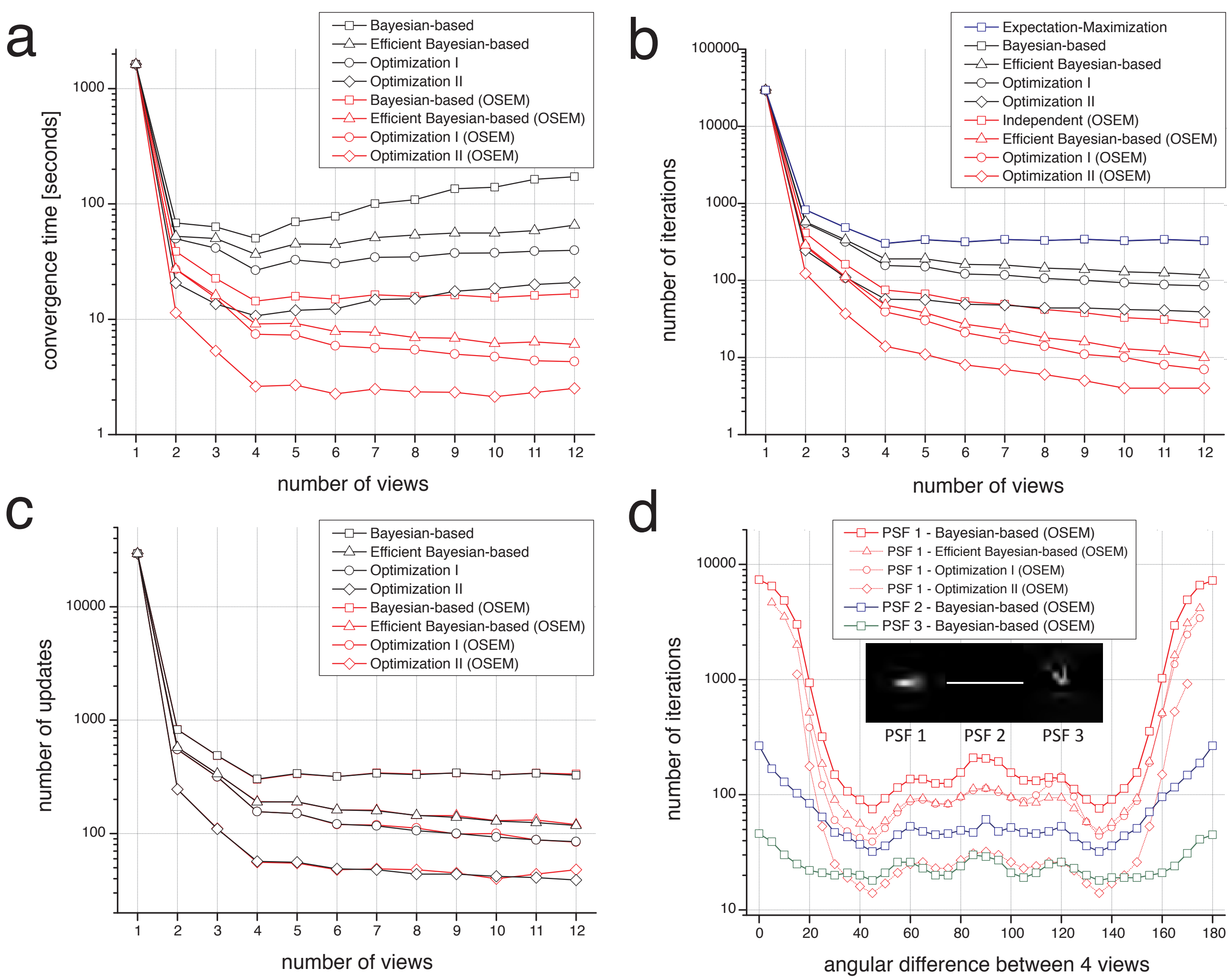
SUPPLEMENTARY FIGURE 3: Illustration of assumption required for incorporating 'virtual' views without additional computational effort



Supplementary Figure 3: *Illustration of assumption in equation 91.* (**a**) shows the difference in the result when computing $(f * g) \cdot (f * h)$ in red and the approximation $f * (g \cdot h)$ in black for a random one-dimensional input sequence (f) and two kernels with σ=3 (g) and σ=2 (h) after normalization. (**b**) shows the difference when using the two-dimensional image from supplementary figure 5a as input (f) and the first two point spread functions from supplementary figure 5e as kernels (g, h). The upper panel pictures the approximation, the lower panel the correct computation. Note that for (**a,b**) the approximation is slightly less blurred. Note that the beads are also visible in the lower panel when adjusting the brightness/contrast.

Nature Methods: doi:10.1038/nmeth.2929

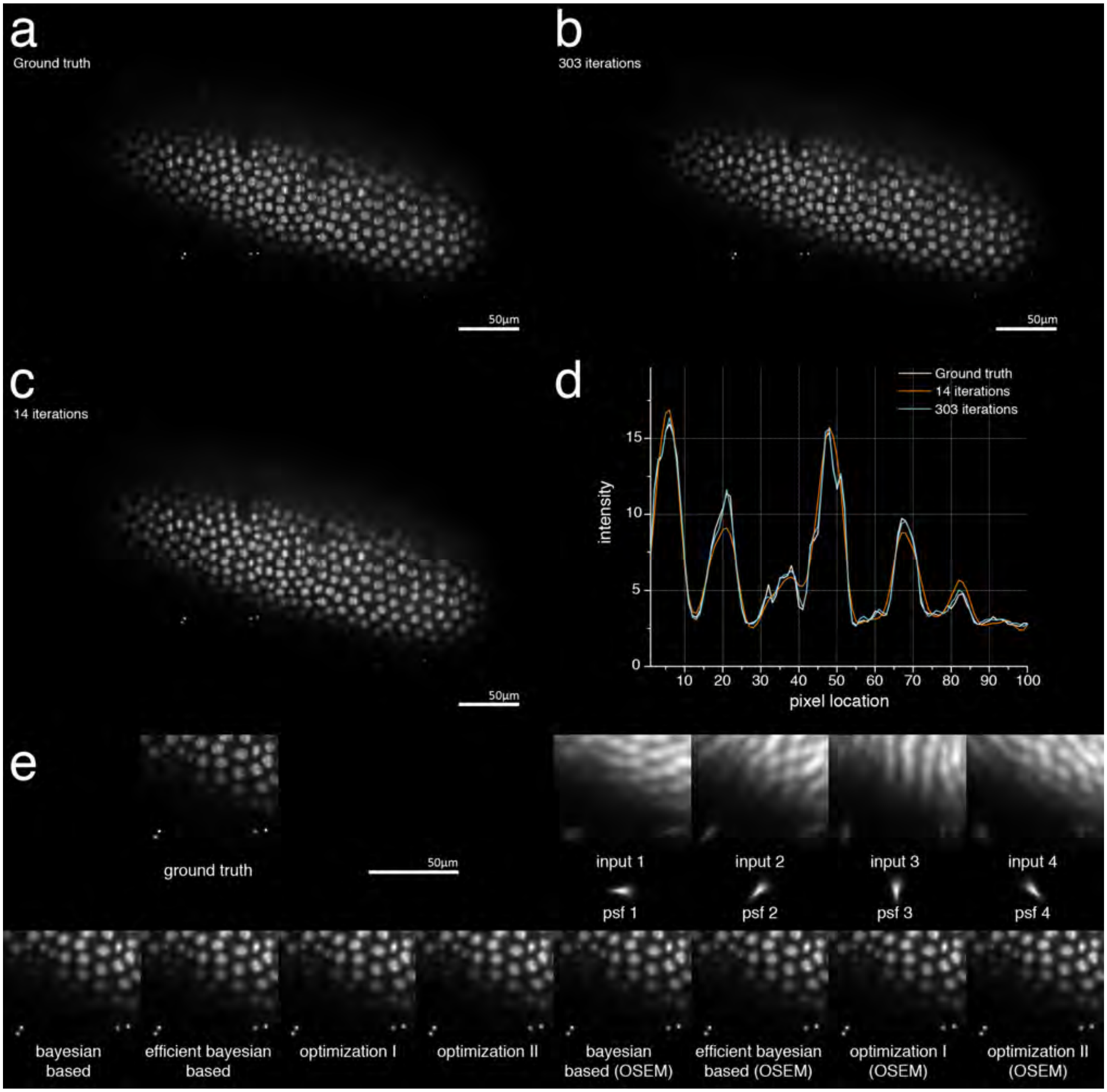
SUPPLEMENTARY FIGURE 4: Performance comparison of the multi-view deconvolution methods and dependence on the PSF



Supplementary Figure 4: *Performance comparison and dependence on the PSF* (**a**) The convergence time of the different algorithms until they reach the same average difference to the ground truth image shown in supplementary figure 5e. (**b**) The number of iterations required until all algorithms reach the same average difference to the ground truth image. One 'iteration' comprises all computional steps until each view contributed once to update the underlying distribution. Note that our Bayesian-based derivation and the Maximization-Likelihood Expectation-Maximization[1] method perform almost identical (**c**) The total number of updates of the underlying distribution until the same average difference is reached. (**d**) The number of iterations required until the same difference to the ground truth is achieved using 4 views. The number of iterations is plotted relative to the angular difference between the input PSFs. An angular difference of 0 degrees refers to 4 identical PSFs and therefore 4 identical input images, an example of an angular difference of 45 degrees is shown in supplementary figure 5e. Plots are shown for different types of PSFs. (**a-d**) y-axis has logarithmic scale, all computations were performed on a dual-core Intel Core i7 with 2.7Ghz.

Nature Methods: doi:10.1038/nmeth.2929

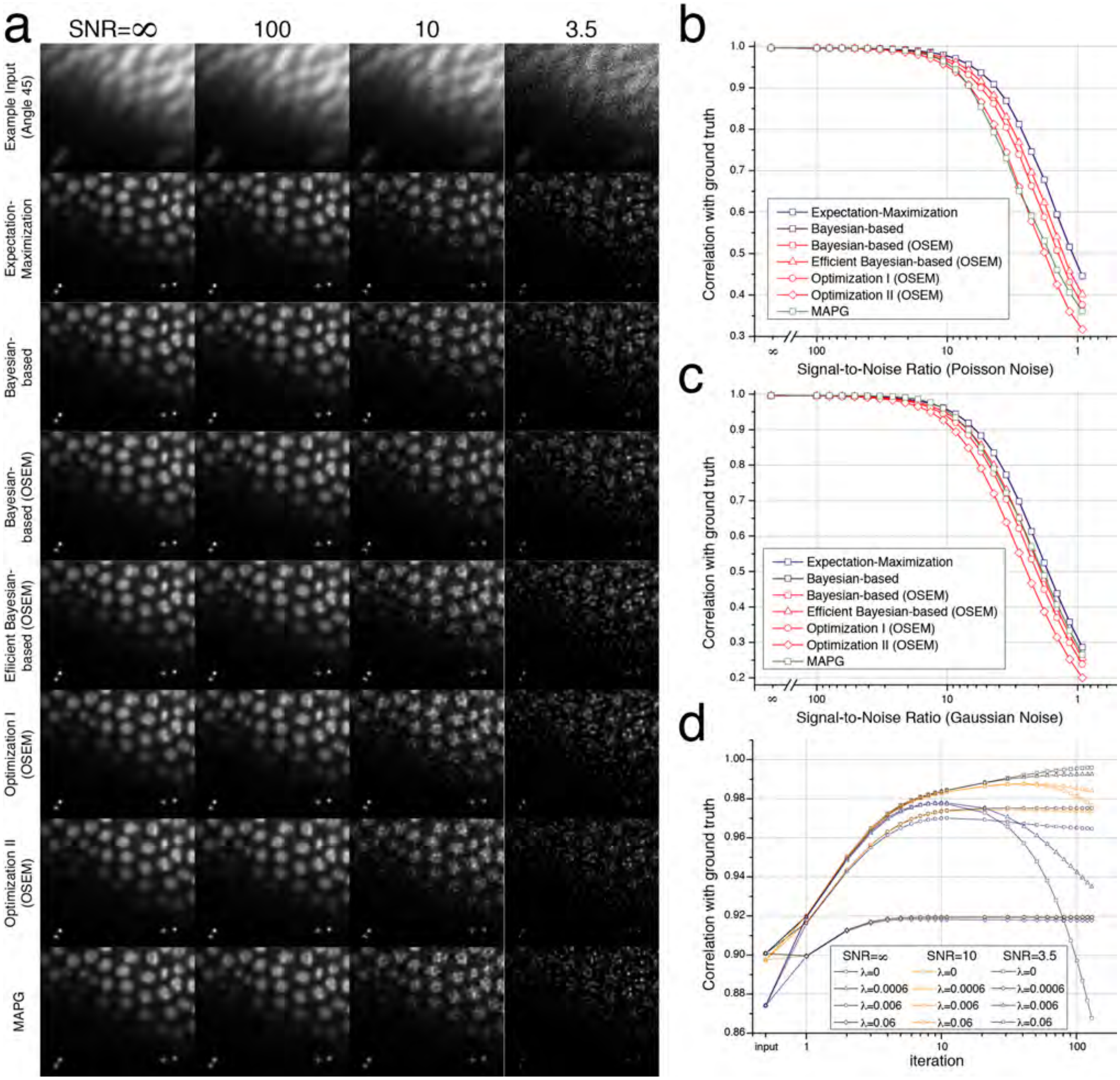
SUPPLEMENTARY FIGURE 5: Images used for analysis and visual performance



Supplementary Figure 5: *Images used for analysis and visual performance.* (**a**) The entire ground truth image used for all analyses shown in the supplement. (**b**) Reconstruction quality after 301 iterations using optimization II and sequential updates on 4 input views and PSF's as shown in (e). (**c**) Reconstruction quality after 14 iterations for the same input as (b). (**d**) Line-plot through the image highlighting the deconvolution quality after 301 (b) and 14 (c) iterations compared to the ground truth (a). (**e**) Magnificantion of a small region of the ground truth image (a), the 4 input PSF's and 4 input datasets as well as the results for all algorithms as used in supplementary figure 4a-c for performance measurements.

Nature Methods: doi:10.1038/nmeth.2929

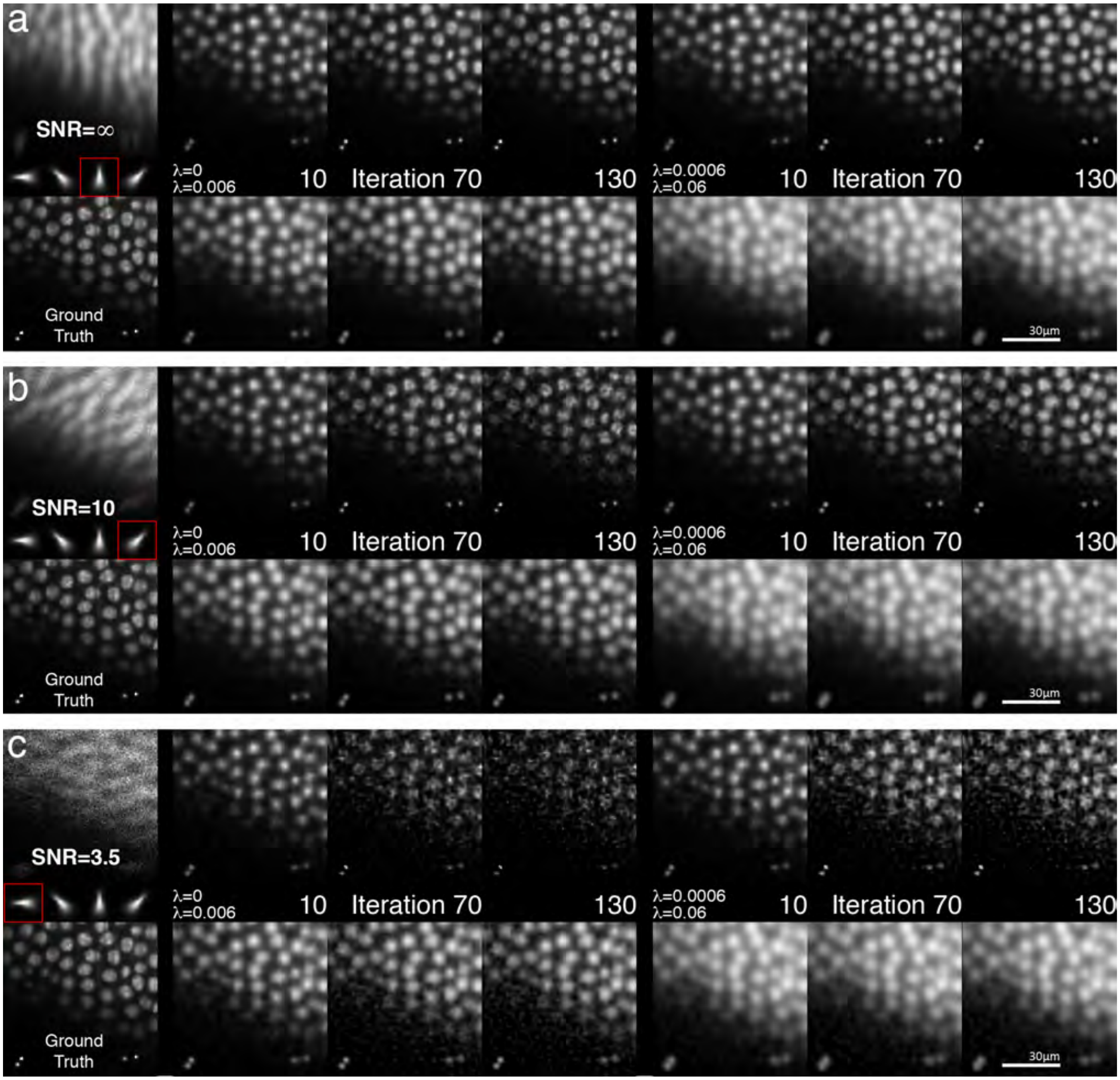
SUPPLEMENTARY FIGURE 6: Effect of noise on the deconvolution results



Supplementary Figure 6: *Effect of noise on the deconvolution results.* (**a**) Deconvolved images corresponding to the points in graph (b) to illustrate the resulting image quality corresponding to a certain correlation coefficient. (**b,c**) The resulting cross-correlation between the ground truth image and the deconvolved image depending on the signal-to-noise ratio in the input images. (**b**) Poisson noise, (**c**) Gaussian noise. (**d**) The cross correlation between the ground truth image and the deconvolved image at certain iteration steps during the deconvolution shown for different signal-to-noise ratios (SNR=∞ [no noise], SNR=10, SNR=3.5) and varying parameters of the Tikhonov regularization (λ=0 [no regularization], λ=0.0006, λ=0.006, λ=0.06). Supplementary figure 7 shows the corresponding images for all data points in this plot. This graph is based on the Bayesian-based derivation using sequential updates in order to be able to illustrate the behaviour in early stages of the devonvolution.

Nature Methods: doi:10.1038/nmeth.2929

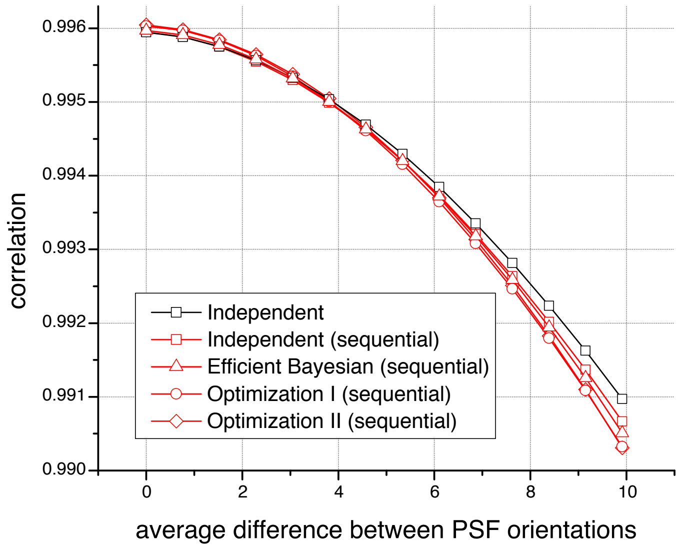
SUPPLEMENTARY FIGURE 7: Intermediate stages of deconvolution results for varying SNR's and regularization



Supplementary Figure 7: *Intermediate stages of deconvolution results for varying SNR's and regularization.* (**a-c**) 1st row shows input data for the PSF in the red box, PSF's and ground truth, the other rows show the images at iteration 10, 70 and 130 for varying parameters of the Tikhonov regularization (λ=0 [no regularization], λ=0.0006, λ=0.006, λ=0.06). (**a**) Results and input for SNR=∞ (no noise). Here, λ=0 shows best results. (**b**) Results and input for SNR=10 (Poisson noise). Small structures like the fluorescent beads close to each other remain separable. (**c**) Results and input for SNR=3.5 (Poisson noise). Note that although the beads cannot be resolved anymore in the input data, the deconvolution produces a reasonable result, visually best for a λ between 0.0006 and 0.006. This graph as well as based on the Bayesian-based derivation using sequential updates (see supplementary figure 6d).

Nature Methods: doi:10.1038/nmeth.2929

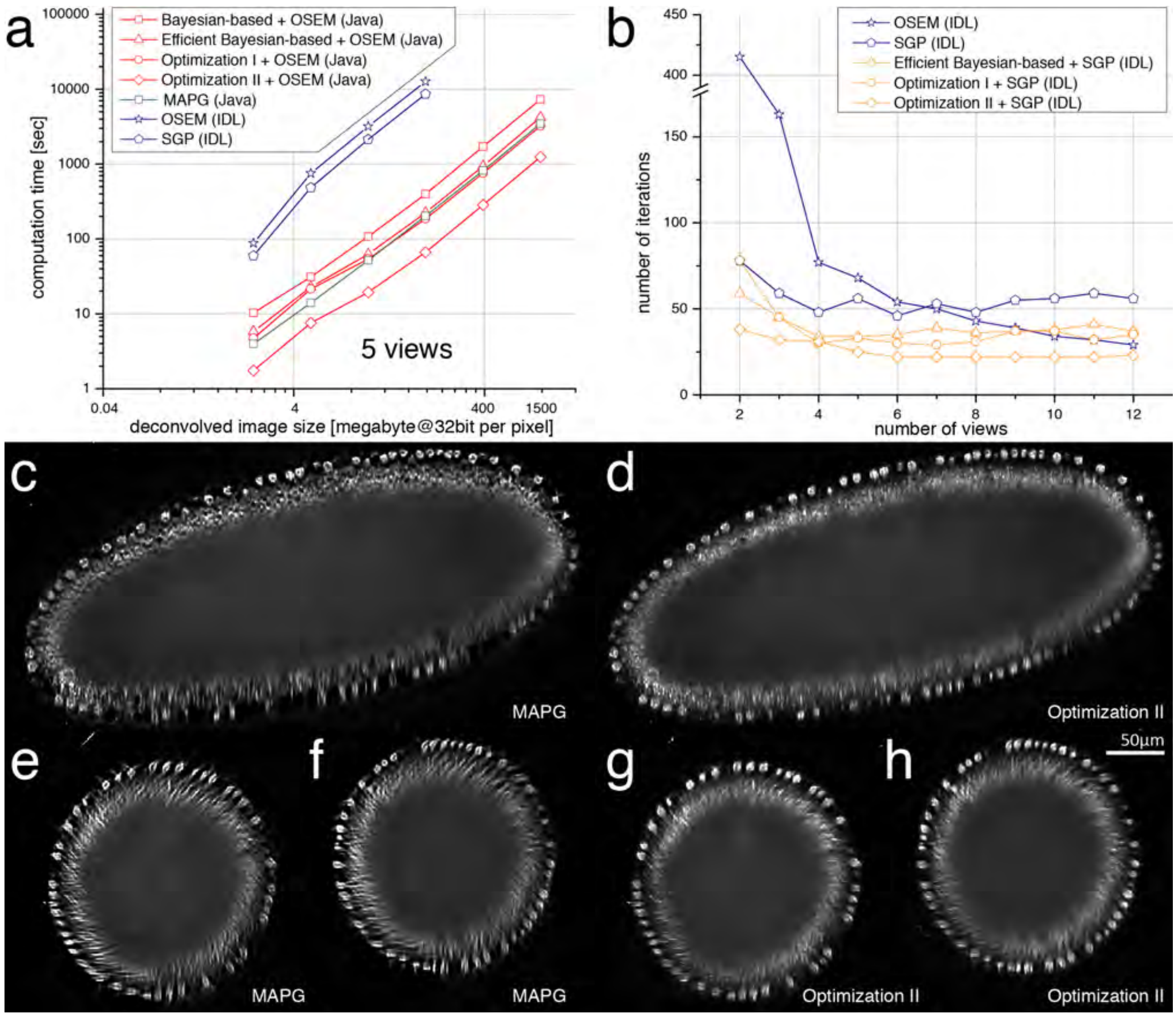
SUPPLEMENTARY FIGURE 8: Quality of deconvolution for imprecise estimation of the PSF



Supplementary Figure 8: *Quality of deconvolution for imprecise estimation of the PSF*. The cross-correlation between the deconvolved image and the ground truth images when the PSF's used for deconvolution were rotated by random angles relative to the PSF's used to create the input images.

Nature Methods: doi:10.1038/nmeth.2929

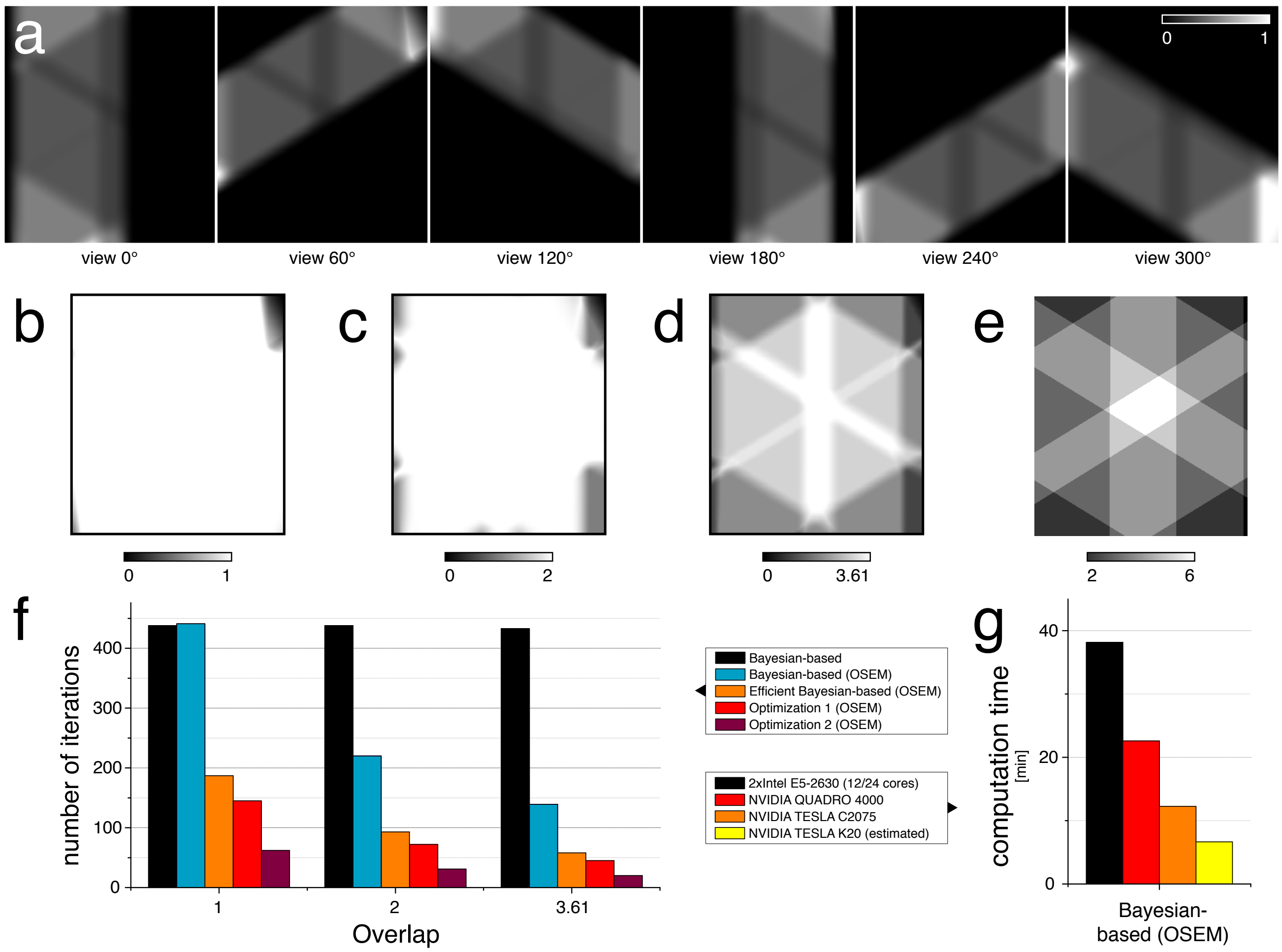
SUPPLEMENTARY FIGURE 9: Comparison to other optimized multi-view deconvolutions



Supplementary Figure 9: *Comparison to other optimized multi-view deconvolution schemes.* (**a,b**) Compares optimized versions of multi-view deconvolution, including the IDL implementations of Scaled Gradient Projection (SGP),[4] Ordered Subset Expectation Maximization (OSEM),[5] Maximum a posteriori with Gaussian Noise (MAPG),[6] and our derivations combined with OSEM (see also main text figure 1e,f). All computations were performed on a machine with 128 GB of RAM and two 2.7 GHz Intel E5-2680 processors. (**a**) Correlates computation time and image size until the deconvolved image reached the same difference to the known ground truth image. All algorithms perform relatively proportional, however the IDL implementations run out of memory. (**b**) illustrates that our optimizations can also be combined with SGP in order to achieve a faster convergence. (**c-h**) compare the reconstruction quality of MAPG and Optimization II using the 7-view acquisition of the *Drosophila* embryo expressing His-YFP (main text figure 3c,d,e). Without ground truth we chose a stage of similar sharpness (26 iterations of MAPG and 9 iterations of Optimization II, approximately in correspondence with main figure 1f) not using any regularization. Optimization II achieves a visually higher image quality, while MAPG shows some artifacts and enhances the stripe pattern arising from partially overlapping input images. (**c,d**) show a slice in lateral orientation of one of the input views, (**e-h**) show slices perpendicular to the rotation axis.

Nature Methods: doi:10.1038/nmeth.2929

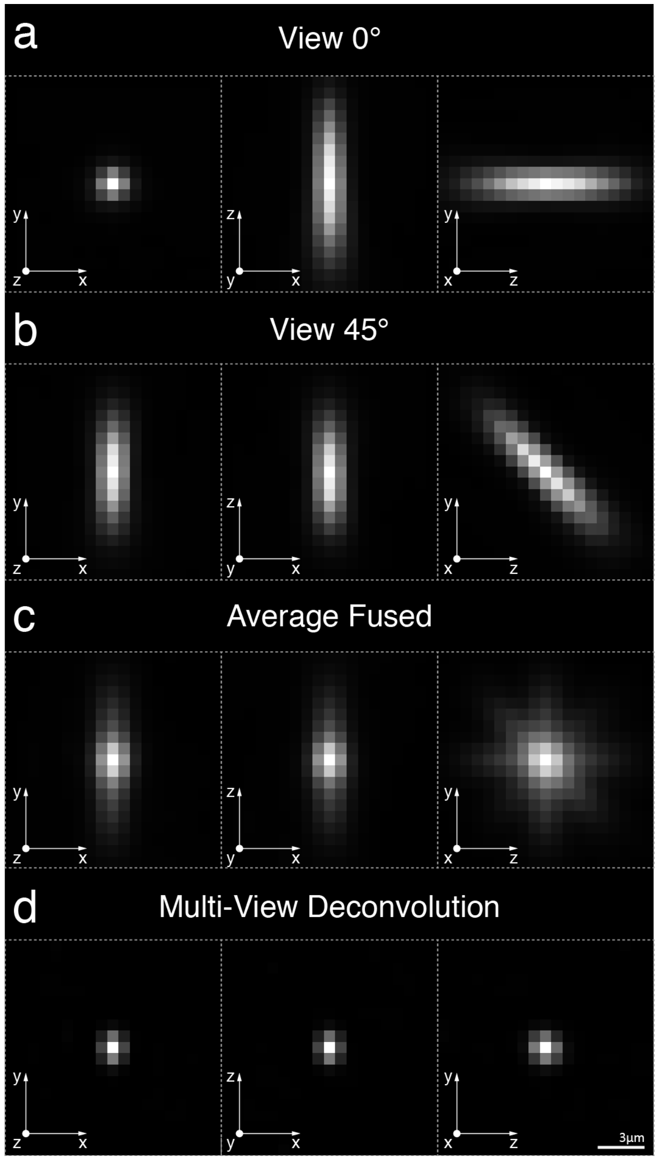
SUPPLEMENTARY FIGURE 10: Effects of partial overlap and CUDA performance



Supplementary Figure 10: *Effects of partial overlap and CUDA performance.* (**a**) shows for one slice perpendicular to the rotation axis the weights for each pixel of each view of a six-view SPIM acquistion (see supplementary figure 14e-j for image data). Every view only covers part of the entire volume. Close to the boundaries of each view we limit its contribution using a cosine blending function preventing artifacts due to sharp edges.[7] For each individual pixel the sum of weights over all views is normalized to be ≤ 1. Black corresponds to a weight of 0 (this view is not contributing), white to a weight of 1 (only this view is contributing). (**b-d**) illustrate how much each pixel is deconvolved in every iteration when using different amounts of OSEM speedup (i.e. assuming a certain amount of overlapping views). Note that individual weights must not be >1. (**b**) normalizing the sum of all weights to ≤ 1 results in a uniformly deconvolved image except the corners where the underlying data is missing, however no speedup is achieved by OSEM (**f** left). Note that summing up all 6 images from (**a**) results in this image. (**c**) two views is the minimal number of overlapping views at every pixel (see **e**), so normalization to ≤ 2 still provides a pretty uniform deconvolution and a 2-fold speed up (**f** center). (**d**) normalizing to ≤ 3.61 (average number of overlapping views) results in more deconvolution of center parts, which is not desireable. Many parts of the image are not covered by enough views to achieve a sum of weights of 3.61. (**f**) performance improvement of partially overlapping datasets using the weights pictured above and a cropped version of the ground truth image (supplementary figure 5). The effect is identical to perfectly overlapping views, but the effective number of overlapping views is reduced. Our new optimizations improve performance in any case. (**g**) the relative speed-up of deconvolution performance that can be achieved using our CUDA implementation.

Nature Methods: doi:10.1038/nmeth.2929

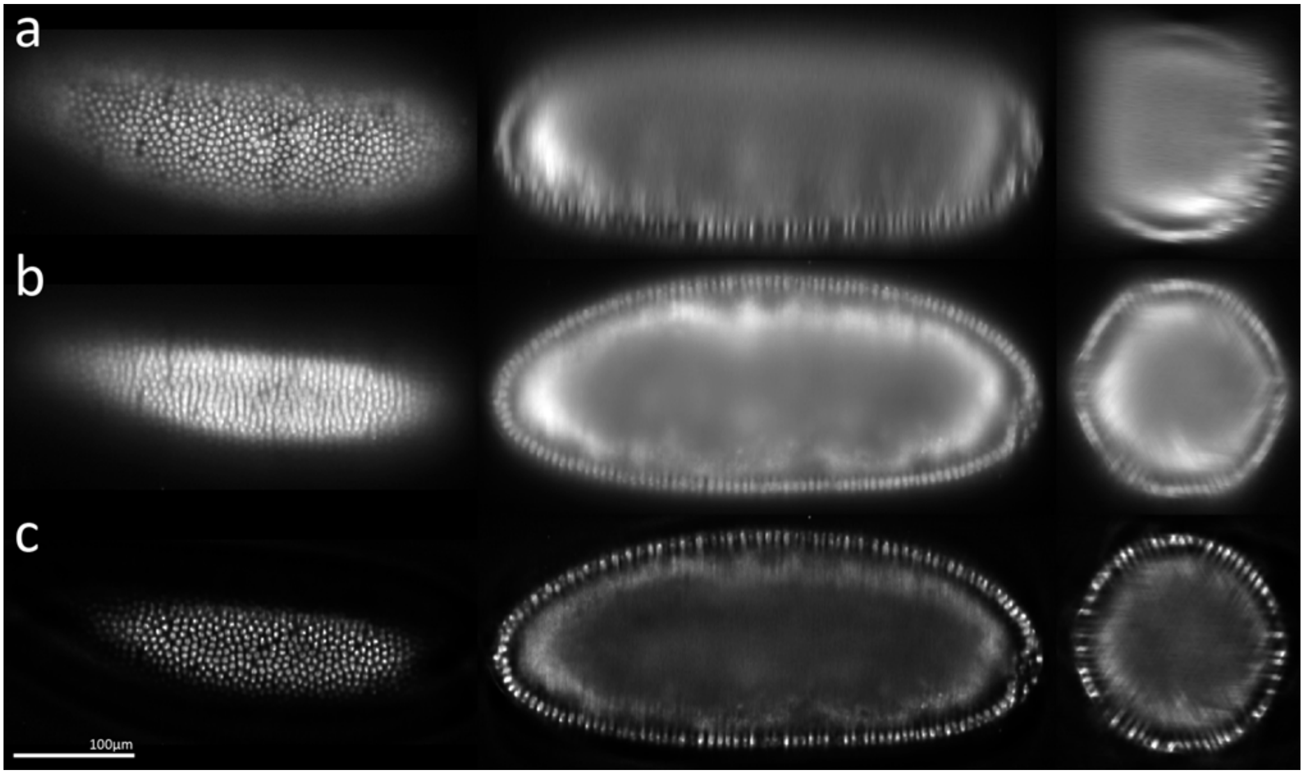
SUPPLEMENTARY FIGURE 11: Quantification of resolution enhancement by Multi-View Deconvolution



Supplementary Figure 11: *Quantification of resolution enhancement by Multi-View Deconvolution.* (**a-d**) compare the average of all fluorescent beads matched by the bead-based registration[7] for two input views (**a,b**), after multi-view fusion (**c**), and after multi-view deconvolution (**d**). The resolution enhancement is apparent, especially along the rotation axis (third column, yz) between (**c**) and (**d**). The dataset used for this analysis is the 7-view acquisition of a developing *Drosophila* embryo (see main text figure 3c-e), deconvolved for 15 iterations with λ=0.0006 using Optimization I.

Nature Methods: doi:10.1038/nmeth.2929

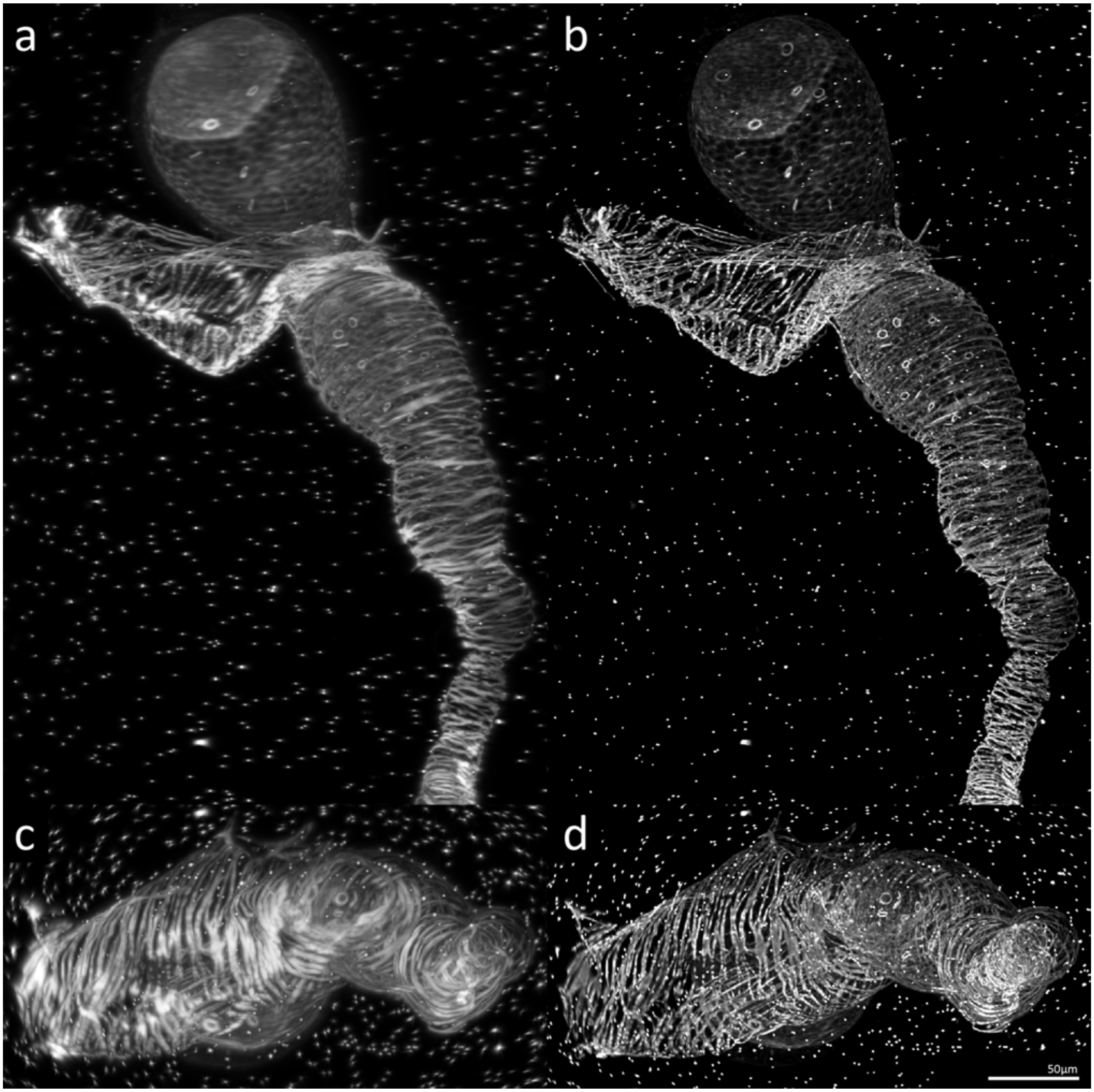
SUPPLEMENTARY FIGURE 12: Reconstruction quality of an OpenSPIM acquistion



Supplementary Figure 12: *Comparison of reconstruction quality on the OpenSPIM.* (**a**) Quality of one of the input views as acquired by the OpenSPIM microscope. (**b**) Quality of the content-based fusion of the registered dataset. (**c**) Quality of the deconvolution of the registered dataset. (**a-c**) The first column shows a slice in the lateral orientation of the input dataset, the second column shows an orthogonal slice, the third column shows a slice perpendicular to the rotation axis. All slices are in the exactly same position and show the identical portion of each volume and are directly comparable. The light sheet thickness of the OpenSPIM is larger than of Zeiss prototype, therefore more out-of-focus light is visible and (a,b) are more blurred. Therefore the effect of deconvolution is especially visible, most dominantly in the third column showing the slice perpendicular to the rotation axis. The dataset has a size of $793 \times 384 \times 370$ px, acquired with in 6 views totalling around 680 million pixels and 2.6 gigabytes of data. Computation time for 12 iterations was 12 minutes on two Nvidia Quadro 4000 GPU's using optimization I.

Nature Methods: doi:10.1038/nmeth.2929

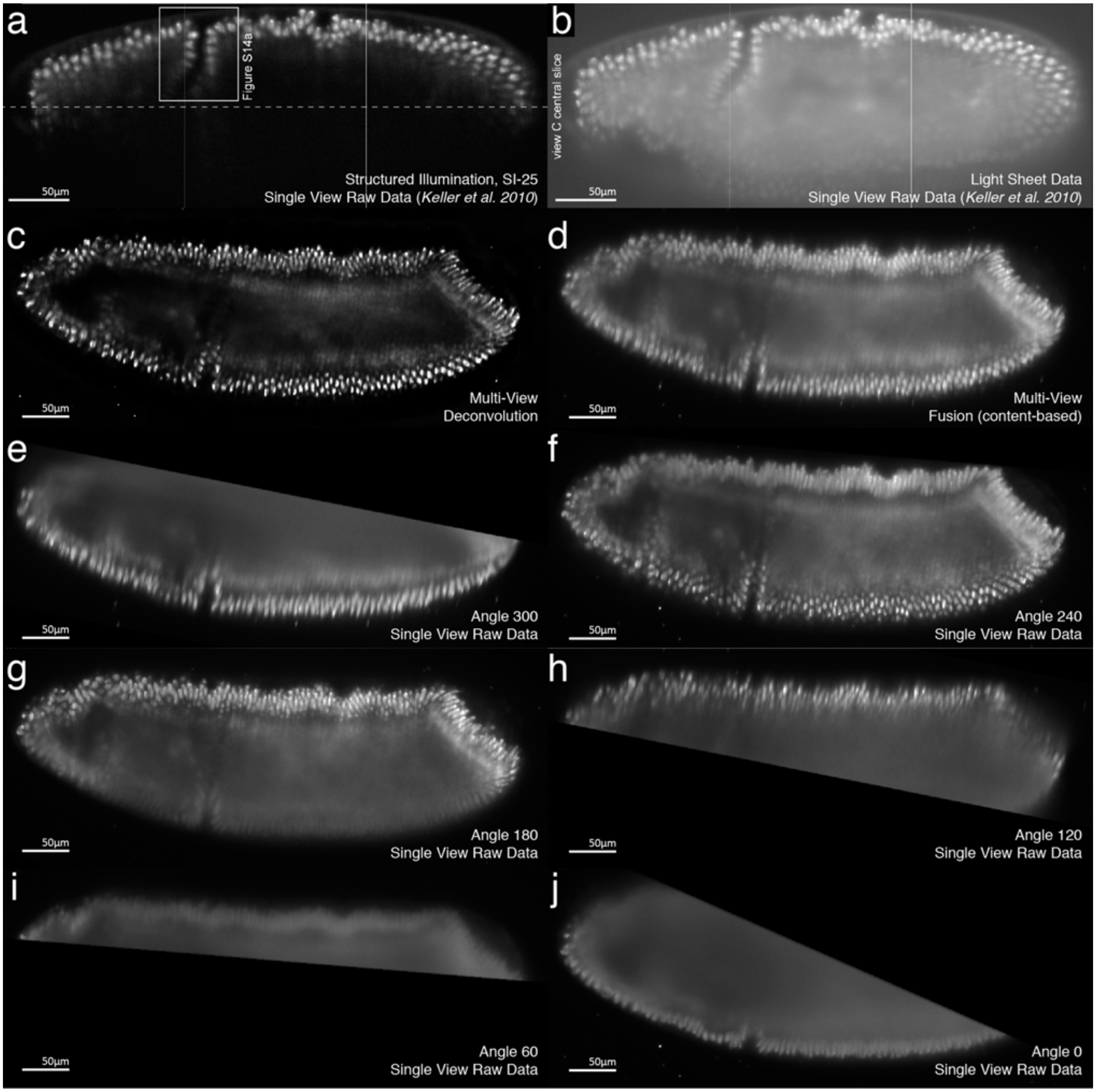
SUPPLEMENTARY FIGURE 13: Quality of reconstruction of Drosophila ovaries



Supplementary Figure 13: *Comparison of reconstruction quality of* Drosophila *ovaries acquired on the Zeiss SPIM prototype using maximum intensity projections.* (**a**) shows the content-based fusion along the orientation of one of the acquired views. (**b**) shows the same image deconvolved. (**c**) shows the projection along the rotation axis of the content-based fusion, (**d**) of the deconvolved dataset. The final dataset has a size of $822 \times 1211 \times 430$ px, acquired in 12 views totalling an input size of around 5 billion pixels and 19 gigabytes of data (32 bit floating point data required for deconvolution). Computation time for 12 iterations was 36 minutes on two Nvidia Quadro 4000 GPU's using optimization I.

Nature Methods: doi:10.1038/nmeth.2929

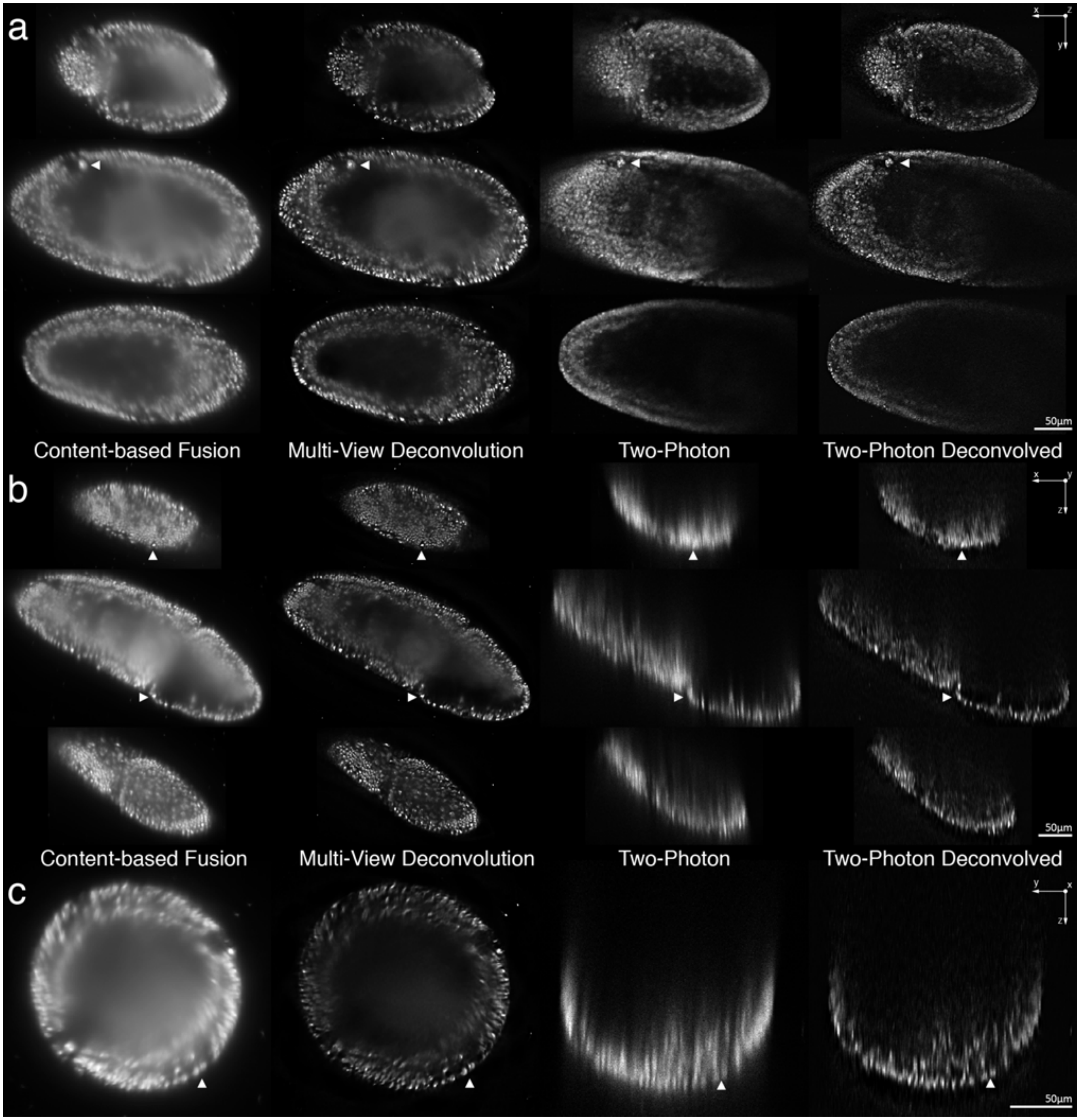
SUPPLEMENTARY FIGURE 14: Comparison of Multi-View Deconvolution to Structured Illumination Light Sheet Data



Supplementary Figure 14: *Comparison of Multi-View Deconvolution to Structured Illumination Light Sheet Data.* (**a**) Slice through a *Drosophila* embryo expressing a nuclear marker acquired with DSLM and structured illumination (SI). (**b**) Corresponding slice acquired with standard light sheet microscopy. (**a**) and (**b**) taken from Keller et al.[8] (**c-j**) Slice through a *Drosophila* embryo in a similar stage of embryonic development expressing His-YFP. (**c**) shows the result of the multi-view deconvolution, (**d**) the result of the content-based fusion and (**e-j**) shows a slice through the aligned[7] raw data as acquired by the Zeiss demonstrator B.

Nature Methods: doi:10.1038/nmeth.2929

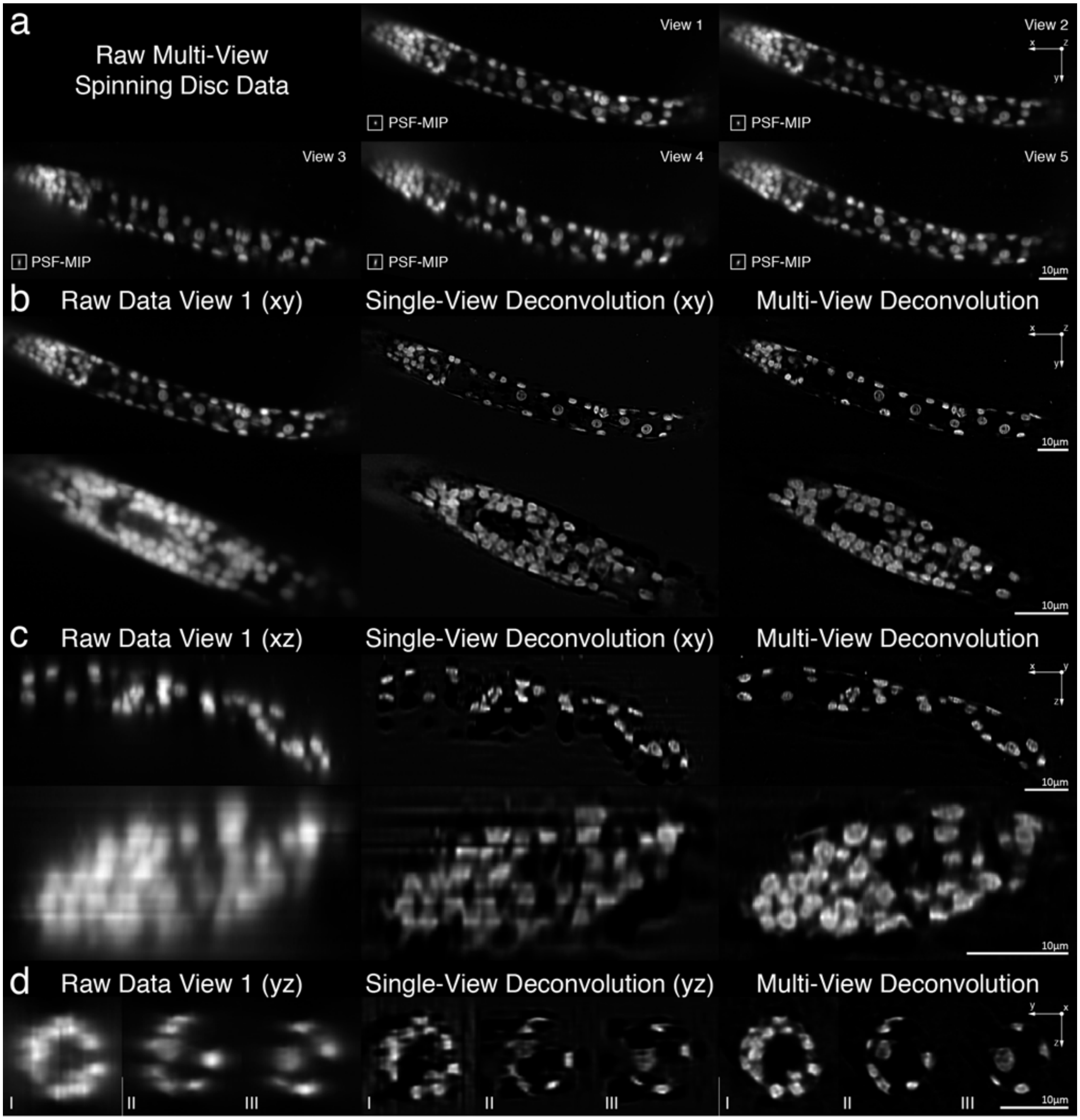
SUPPLEMENTARY FIGURE 15: Comparison of Multi-View Deconvolution to 2p Microscopy



Supplementary Figure 15: *Comparing multi-view deconvolution to two-photon (2p) microscopy.* (**a-c**) slices through a fixed *Drosophila* embryo stained with Sytox green labeling nuclei. Same specimen was acquired with the Zeiss SPIM prototype (20x/0.5NA water dipping obj.) and directly afterwards with a 2p microscope (20x/0.8NA air obj.). We compare the quality of content-based fusion, multi-view deconvolution, raw 2p stack and single view deconvolution of the 2p acquisition. (**a**) lateral (xy), (**b**) axial (xz), (**c**) axial (yz) orientation of the 2p stack, SPIM data is aligned relative to it using the beads in the agarose. Arrows mark corresponding nuclei.

Nature Methods: doi:10.1038/nmeth.2929

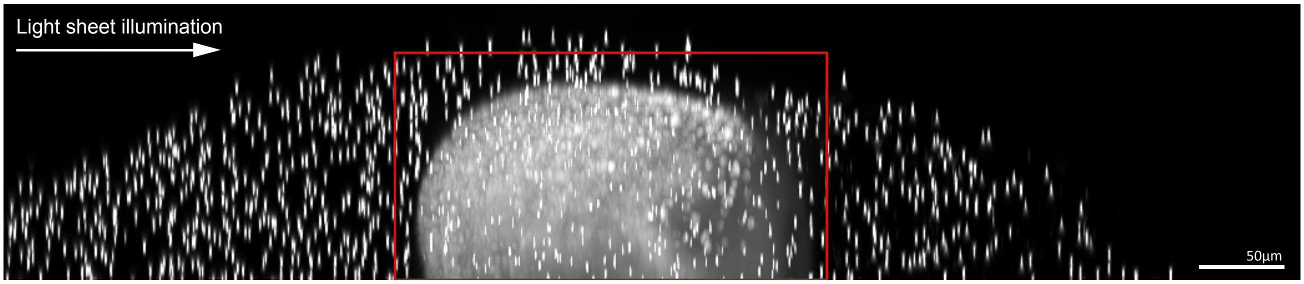
SUPPLEMENTARY FIGURE 16: Multi-View Deconvolution of Spinning-Disc Confocal Data



Supplementary Figure 16: *Multi-View Deconvolution of a Spinning-Disc Confocal Dataset.* (**a-d**) show slices through a fixed C. elegans in L1 stage stained with Sytox green labeling nuclei. The specimen was acquired on a spinning disc confocal microscope (20x/0.5NA water dipping objective). The sample was embedded in agarose and rotated using a self-build device.[7] (**a**) Slice through the aligned input views; insets show averaged MIP of the PSF. (**b-d**) slices with different orientations through the larva comparing the quality of the first view of the input data, the single-view deconvolution of view 1 and the multi-view deconvolution of the entire dataset.

Nature Methods: doi:10.1038/nmeth.2929

SUPPLEMENTARY FIGURE 17: Variation of PSF across the light sheet in SPIM acquistions



Supplementary Figure 17: *Variation of the PSF across the light sheet in SPIM acquistions*. The maximum intensity projection perpendicular to the light sheet of a *Drosophila* embryo expressing His-YFP in all nuclei. The fluorescent beads have a diameter of 500nm. The arrow shows the illumination direction of the light sheet. The fluorescent beads should reflect the concave shape of a light sheet. The red box illustrates the area that is approximately used for deconvolution.

Nature Methods: doi:10.1038/nmeth.2929

SUPPLEMENTARY TABLES

SUPPLEMENTARY TABLE 1: Summary of datasets used in this publication

Dataset	Size, Lightsheet Thickness, SNR*	Computation Time, Iterations, Method	Machine
Drosophila embryo expressing His-YFP in all cells acquired with Zeiss SPIM prototype using a 20x/0.5 detection objective (**Fig. 2c-e**, **Supp. Fig. 11**)†	720×380×350 px, 7 views, LS∼5μm, SNR∼30	7 minutes, 12 iterations, optimization I, λ = 0.006	2× Nvidia Quadro 4000‡, 64 GB RAM
Drosophila embryo expressing His-YFP in all cells acquired with the Open-SPIM using a 20x/0.5 detection objective (**Supp. Fig. 12**)	793×384×370 px, 6 views, LS∼10μm, SNR∼15	12 minutes§, 12 iterations, optimization I, λ = 0.006	2× Nvidia Quadro 4000‡, 64 GB RAM
Drosophila ovaries acquired on the Zeiss SPIM prototype using a 20x/0.5 detection objective (**Supp. Fig. 13**)	1211×822×430 px, 12 views, LS∼5μm, SNR∼19	36 minutes, 12 iterations, optimization I, λ = 0.006	2× Nvidia Quadro 4000‡, 64 GB RAM
Drosophila embryo expressing His-YFP in all cells acquired with Zeiss SPIM prototype using a 20x/0.5 detection objective (**Supp. Video 8-10**,**Supp. Fig. 14**)	792×320×310 px, 6 views, 236 timepoints, LS∼5μm, SNR∼26	24.3 hours, 12 iterations, optimization I, λ = 0.006	2× Nvidia Quadro 4000‡, 64 GB RAM
Drosophila embryo expressing Histone-H2Av-mRFPruby fusion in all cells imaged on Zeiss Lightsheet Z1 with a 20x/1.0 detection objective and dual-sided illumination	928×390×390 px, 6 views, 715 timepoints, LS∼5μm, SNR∼21	35 hours, 10 iterations, optimization I, λ = 0.0006	4× Nvidia TESLA¶, 64 GB RAM
C. elegans embryo in 4-cell stage expressing PH-domain-GFP fusion acquired with Zeiss SPIM prototype using a 40x/0.8 detection objective (**Fig. 2a,b**)†	180×135×180 px, 6 views, LS∼3.5μm, SNR∼40	1 minute, 20 iterations, optimization I, λ = 0.006	2× Intel Xeon E5-2630, 64 GB RAM
Fixed *C. elegans* larvae in L1 stage expressing LMN-1::GFP and stained with Hoechst imaged on Zeiss Lightsheet Z1 with a 20x/1.0 detection objective (**Fig. 2f,g** and **Supp. Video 4-7**)	1640×1070×345 px, 4 views, 2 channels, LS∼2μm, SNR∼62 (Hoechst), SNR∼24 (GFP)	2×160 minutes, 100 iterations, optimization II, λ = 0	2× Intel Xeon E5-2690, 128 GB RAM

*The SNR is estimated by M.S., E.M., and P.T. were additionally supported by the Bundesministerium fr Bildung und Forschung grant 031A099.computing the average intensity of the signal, divided by the standard deviation of the signal in areas with homogenous sample intensity

†This SPIM acquisition was already used in Preibisch (2010)[7] to illustrate the results of the bead-based registration and multi-view fusion; we use the underlying dataset again to illustrate the improved results of the multi-view deconvolution.

‡Two graphics cards in one PC, which can process two 512×512×512 blocks in parallel

§Note that the increased computation time is due to larger anisotropy of the acquired stacks leading to larger effective

Nature Methods: doi:10.1038/nmeth.2929

SUPPLEMENTARY TABLE 1 (CONTINUED): Summary of datasets used in this publication

Dataset	Size, Lightsheet Thickness, SNR	Computation Time, Iterations, Method	Machine
*F*ixed C. elegans in L1 stage stained with Sytox green acquired with Spinning Disc Confocal using a 20x/0.5 detection objective (**Supp. Fig. 16**)[‖]	1135×400×430 px, 5 views, LS N/A, SNR∼28	36 minutes, 50 iterations, optimization II, λ = 0.0006	2× Intel Xeon E5-2680, 128 GB RAM
*F*ixed C. elegans in L1 stage stained with Sytox green acquired with Spinning Disc Confocal using a 20x/0.5 detection objective (**Supp. Fig. 16**)[**]	1151×426×190 px, 1 view, LS N/A, SNR∼28	202 minutes, 900 iterations, Lucy-Richardson, λ = 0.0006	2× Intel Xeon E5620, 64 GB RAM
*F*ixed Drosophila embryo stained with Sytox green acquired on the Zeiss SPIM prototype using a 20x/0.5 detection objective (**Supp. Fig. 15**)	642×316×391 px, 9 views, LS∼5μm, SNR∼20	15 minutes, 15 iterations, optimization I, λ = 0.006	2× Intel Xeon E5-2680, 128 GB RAM
*F*ixed Drosophila embryo stained with Sytox green acquired on a Two-Photon Microscope using a 20x/0.8 detection objective (**Supp. Fig. 15**)	856×418×561 px, 1 view, LS N/A, SNR∼7	160 minutes, 300 iterations, Lucy-Richardson, λ = 0.006	2× Intel Xeon E5620, 64 GB RAM

Supplementary Table 1: *Summary of all datasets used in this publication*. Note that the multi-view deconvolution of the *C. elegans* larvae in L1 stage (SPIM & Spinning Disc Confocal) required an additional registration step, which is explained in the **Online Methods.**

PSF sizes, which increases computational effort. The image could therefore not be split up into two 512×512×512 blocks.

[¶]Run on a cluster with 4 nodes that are equipped with one Nvidia TESLA and 64 GB of system memory

[‖]This multi-view spinning disc acquisition was already used in Preibisch (2010)[7] to illustrate the applicability of the bead-based registration and multi-view fusion to other technologies than SPIM; we use the underlying dataset again to illustrate the improved results and applicability of the multi-view deconvolution.

[**]This is the same dataset as in the row above, but showing the time it took to compute the single-view deconvolution.

Nature Methods: doi:10.1038/nmeth.2929

SUPPLEMENTARY NOTE 1: BAYESIAN-BASED SINGLE-VIEW AND MULTI-VIEW DECONVOLUTION

REMARKS

This document closely follows the notation introduced in the paper of L. B. Lucy[9] whenever possible. Note that for simplicity all derivations in this document only cover the one-dimensional case. Nevertheless, all equations are valid for any *n*-dimensional case.

SUPPLEMENTARY NOTE 1.1: Derivation of the Bayesian-based single-view deconvolution

This section re-derives the classical Bayesian-based Richardson[10]-Lucy[9] deconvolution for single images, other derivations presented in this document build up on it. The goal is to estimate the frequency distribution of an underlying signal $\psi(\xi)$ from a finite number of measurements $x^{1'}, x^{2'}, ..., x^{N'}$. The resulting observed distribution $\phi(x)$ is defined as

$$\phi(x) = \int_{\xi} \psi(\xi) P(x|\xi) d\xi \tag{1}$$

where $P(x|\xi)$ is the probability of a measurement occuring at $x = x'$ when it is known that the event $\xi = \xi'$ occured. In more practical image analysis terms equation 1 describes the one-dimensional convolution operation where $\phi(x)$ is the blurred image, $P(x|\xi)$ is the kernel and $\psi(\xi)$ is the undegraded (or deconvolved) image. All distributions are treated as probability distributions and fulfill the following constraints:

$$\int_{\xi} \psi(\xi) d\xi = \int_{x} \phi(x) dx = \int_{x} P(x|\xi) dx = 1 \quad and \quad \psi(\xi) > 0,\ \phi(x) \geq 0,\ P(x|\xi) \geq 0 \tag{2}$$

The basis for the derivation of the Bayesian-based deconvolution is the tautology

$$P(\xi = \xi' \wedge x = x') = P(x = x' \wedge \xi = \xi') \tag{3}$$

It states that it is equally probable that the event ξ' results in a measurement at x' and that the measurement at x' was caused by the event ξ'. Integrating equation 3 over the measured distribution yields the joint probability distribution

$$\int_{x} P(\xi \wedge x) dx = \int_{x} P(x \wedge \xi) dx \tag{4}$$

which can be expressed using conditional probabilities

$$\int_{x} P(\xi) P(x|\xi) dx = \int_{x} P(x) P(\xi|x) dx \tag{5}$$

and in correspondence to Lucy's notation looks like (equation 1)

$$\int_{x} \psi(\xi) P(x|\xi) dx = \int_{x} \phi(x) Q(\xi|x) dx \tag{6}$$

where $P(\xi) \equiv \psi(\xi), P(x) \equiv \phi(x), P(\xi|x) \equiv Q(\xi|x)$. $Q(\xi|x)$ denotes what Lucy calls the 'inverse' conditional probability to $P(x|\xi)$. It defines the probability that an event at ξ' occured, given a specific measurement at x'. As $\psi(\xi)$ does not depend on x, equation 6 can be rewritten as

$$\psi(\xi) \overbrace{\int_{x} P(x|\xi) dx}^{=1} = \int_{x} \phi(x) Q(\xi|x) dx \tag{7}$$

hence (due to equation 2)

$$\psi(\xi) = \int_{x} \phi(x) Q(\xi|x) dx \tag{8}$$

Nature Methods: doi:10.1038/nmeth.2929

which corresponds to the inverse of the convolution in equation 1. Although $Q(\xi|x)$ cannot be used to directly compute $\psi(\xi)$, *Bayes' Theorem* and subsequently equation 1 can be used to reformulate it as

$$Q(\xi|x) = \frac{\psi(\xi)P(x|\xi)}{\phi(x)} = \frac{\psi(\xi)P(x|\xi)}{\int_\xi \psi(\xi)P(x|\xi)d\xi} \tag{9}$$

Replacing $Q(\xi|x)$ in equation 8 yields

$$\psi(\xi) = \int_x \phi(x)\frac{\psi(\xi)P(x|\xi)}{\int_\xi \psi(\xi)P(x|\xi)d\xi}dx = \psi(\xi)\int_x \frac{\phi(x)}{\int_\xi \psi(\xi)P(x|\xi)d\xi}P(x|\xi)dx \tag{10}$$

which exactly re-states the deconvolution scheme introduced by Lucy and Richardson. The fact that both sides of the equation contain the desired underlying (deconvolved) distribution $\psi(\xi)$ suggests an iterative scheme to converge towards the correct solution

$$\psi^{r+1}(\xi) = \psi^r(\xi)\int_x \frac{\phi(x)}{\int_\xi \psi^r(\xi)P(x|\xi)d\xi}P(x|\xi)dx \tag{11}$$

where $\psi^0(\xi)$ is simply a constant distribution with each value being the average intensity of the measured distribution $\phi(x)$.

Equation 11 turns out to be a maximum-likelihood (ML) expection-maximization (EM) formulation,[11] which works as follows. First, it computes for every pixel the convolution of the current guess of the deconvolved image $\psi^r(\xi)$ with the kernel (PSF) $P(x|\xi)$, i.e. $\phi^r(x) = \int_\xi \psi^r(\xi)P(x|\xi)d\xi$. In EM-terms $\phi^r(x)$ describes the *expected value*. The quotient between the input image $\phi(x)$ and the *expected value* $\phi^r(x)$ yields the disparity for every pixel. These values are initially large but will become very small upon convergence. In an ideal scenario all values of $\phi^r(x)$ and $\phi(x)$ will be identical once the algorithm converged. This ratio is subsequently convolved with the point spread function $P(x|\xi)$ reflecting which pixels influence each other. In EM-terms this is called the *maximization step*. This also preserves smoothness. These resulting values are then pixel-wise multiplied with the current guess of the deconvolved image $\psi^r(\xi)$, which we call an RL-update (Richardson-Lucy). It results in a new guess for the deconvolved image.

Starting from an initial guess of an image with constant values, this scheme will converge towards the correct solution if the guess of the point spread function is correct and if the observed distribution is not degraded by noise, transformations, etc.

SUPPLEMENTARY NOTE 1.1.1: Integrating ξ and x

Note that convolution of $\psi^r(\xi)$ with $P(x|\xi)$ requires integration over ξ, while the convolution of the quotient image with $P(x|\xi)$ integrates over x. Integration over x can be formulated as convolution if $P(x|\xi)$ is constant by using inverted coordinates $P(-x|\xi)$. Note that it can be ignored if the kernel is symmetric $P(x|\xi) = P(-x|\xi)$. For single-view datasets this is often the case, whereas multi-view datasets typically have non-symmetric kernels due to their transformations resulting from image alignment.

Nature Methods: doi:10.1038/nmeth.2929

SUPPLEMENTARY NOTE 1.2: Derivation of the Bayesian-based multi-view deconvolution

This section shows for the first time the entire derivation of Bayesian-based multi-view deconvolution using probabilty theory. Compared to the single-view case we have a set of views $V = \{v_1...v_N : N = |V|\}$ comprising N observed distributions $\phi_v(x_v)$ (input views acquired from different angles), N point spread functions $P_v(x_v|\xi)$ corresponding to each view, and one underlying signal distribution $\psi(\xi)$ (deconvolved image). The observed distributions $\phi_v(x_v)$ are accordingly defined as

$$\phi_1(x_1) = \int_\xi \psi(\xi) P(x_1|\xi) d\xi \tag{12}$$

$$\phi_2(x_2) = \int_\xi \psi(\xi) P(x_2|\xi) d\xi \tag{13}$$

$$... \tag{14}$$

$$\phi_N(x_N) = \int_\xi \psi(\xi) P(x_N|\xi) d\xi \tag{15}$$

The basis for the derivation of the Bayesian-based multi-view deconvolution is again a tautology based on the individual observations

$$P(\xi = \xi' \wedge x_1 = x_1' \wedge ... \wedge x_N = x_N') = P(x_1 = x_1' \wedge ... \wedge x_N = x_N' \wedge \xi = \xi') \tag{16}$$

Integrating equation 16 over the measured distributions yields the joint probability distribution

$$\int_{x_1} ... \int_{x_N} P(\xi \wedge x_1 \wedge ... \wedge x_N) dx_1 ... dx_n = \int_{x_1} ... \int_{x_N} P(x_1 \wedge ... \wedge x_N \wedge \xi) dx_1 ... dx_N \tag{17}$$

shortly written as

$$\int_{\overline{x}} P(\xi, x_1, ..., x_N) d\overline{x} = \int_{\overline{x}} P(x_1, ..., x_N, \xi) d\overline{x} \tag{18}$$

By expressing the term using conditional probabilities one obtains

$$\int_{\overline{x}} P(\xi) P(x_1|\xi) P(x_2|\xi, x_1) \ldots P(x_N|\xi, x_1, ..., x_{N-1}) d\overline{x} = \int_{\overline{x}} P(x_1) P(x_2|x_1) P(x_3|x_1, x_2) \ldots P(\xi|x_1, ..., x_N) d\overline{x} \tag{19}$$

On the left side of the equation all terms are conditionally independent of any x_v given ξ. This results from the fact that if an event $\xi = \xi'$ occured, each individual measurement x_v depends only on ξ' and the respective point spread function $P(x_v|\xi)$ (supplementary figure 1a for illustration). Equation 19 therefore reduces to

$$P(\xi) \overbrace{\int_{\overline{x}} P(x_1|\xi) P(x_2|\xi) \ldots P(x_N|\xi) d\overline{x}}^{=1} = \int_{\overline{x}} P(x_1) P(x_2|x_1) P(x_3|x_1, x_2) \ldots P(x_N|x_1, ..., x_{N-1}) P(\xi|x_1, ..., x_N) d\overline{x} \tag{20}$$

Assuming independence of the observed distributions $P(x_v)$ equation 20 further simplifies to

$$P(\xi) = \int_{\overline{x}} P(x_1) P(x_2) P(x_3) \ldots P(x_N) P(\xi|x_1, ..., x_N) d\overline{x} \tag{21}$$

Although independence between the views is assumed,[2,3,12] the underlying distribution $P(\xi)$ still depends on the observed distributions $P(x_v)$ through $P(\xi|x_1, ..., x_N)$.

Note: In section 3 we will show that that the derivation of Bayesian-based multi-view deconvolution can be be achieved without assuming independence of the observed distributions $P(x_v)$. Based on that derivation we argue that there is a relationship between the $P(x_v)$'s (supplementary figure 1b for illustration) and that it can be incorporated into the derivation to achieve faster convergence as shown in sections 4 and 6.

Nature Methods: doi:10.1038/nmeth.2929

We cannot approximate $P(\xi|x_1, ..., x_N)$ directly and therefore need to reformulate it in order to express it using individual $P(\xi|x_v)$, which can subsequently be used to formulate the deconvolution task as shown in section 1.1. Note that according to Lucy's notation $P(\xi|x_1, ..., x_N) \equiv Q(\xi|x_1, ..., x_N)$ and $P(\xi|x_v) \equiv Q(\xi|x_v)$.

$$P(\xi|x_1, ..., x_N) = \frac{P(\xi, x_1, ..., x_N)}{P(x_1, ..., x_N)} \tag{22}$$

$$P(\xi|x_1, ..., x_N) = \frac{P(\xi)P(x_1|\xi)P(x_2|\xi, x_1)...P(x_N|\xi, x_1, ..., x_{N-1})}{P(x_1)P(x_2|x_1)...P(x_N|x_1, ..., x_{N-1})} \tag{23}$$

Due to the conditional independence of the $P(x_v)$ given ξ (equation 19 $\rightarrow$ 20 and supplementary figure 1a) and the assumption of independence between the $P(x_v)$ (equation 20 $\rightarrow$ 21) equation 23 simplifies to

$$P(\xi|x_1, ..., x_N) = \frac{P(\xi)P(x_1|\xi)...P(x_N|\xi)}{P(x_1)...P(x_N)} \tag{24}$$

Using *Bayes' Theorem* to replace all

$$P(x_v|\xi) = \frac{P(x_v)P(\xi|x_v)}{P(\xi)} \tag{25}$$

yields

$$P(\xi|x_1, ..., x_N) = \frac{P(\xi)\ P(\xi|x_1)P(x_1)...P(\xi|x_N)P(x_N)}{P(x_1)...P(x_N)\ P(\xi)^N} \tag{26}$$

$$P(\xi|x_1, ..., x_N) = \frac{P(\xi)\ P(\xi|x_1)...P(\xi|x_N)}{P(\xi)^N} \tag{27}$$

$$P(\xi|x_1, ..., x_N) = \frac{P(\xi|x_1)...P(\xi|x_N)}{P(\xi)^{N-1}} \tag{28}$$

Substituting equation 28 in equation 21 yields

$$P(\xi) = \frac{\int_{\overline{x}} P(x_1) ... P(x_N)P(\xi|x_1)...P(\xi|x_N)d\overline{x}}{P(\xi)^{N-1}} \tag{29}$$

and rewritten in Lucy's notation

$$\psi(\xi) = \frac{\int_{\overline{x}} \phi_1(x_1) ... \phi_N(x_N)Q(\xi|x_1)...Q(\xi|x_N)d\overline{x}}{\psi(\xi)^{N-1}} \tag{30}$$

$$\psi(\xi) = \frac{\int_{x_1} \phi_1(x_1)Q(\xi|x_1)dx_1 ... \int_{x_N} \phi_N(x_N)Q(\xi|x_N)dx_N}{\psi(\xi)^{N-1}} \tag{31}$$

$$\psi(\xi) = \frac{\prod_{v \in V} \int_{x_v} \phi_v(x_v)Q(\xi|x_v)dx_v}{\psi(\xi)^{N-1}} \tag{32}$$

Nature Methods: doi:10.1038/nmeth.2929

As in the single view case we replace $Q(\xi|x_v)$ with equation 9

$$\psi(\xi) = \frac{\prod_{v \in V} \int_{x_v} \phi_v(x_v) \frac{\psi(\xi) P(x_v|\xi)}{\int_\xi \psi(\xi) P(x_v|\xi) d\xi} dx_v}{\psi(\xi)^{N-1}} \tag{33}$$

$$\psi(\xi) = \frac{\psi(\xi)^{\cancel{N}} \prod_{v \in V} \int_{x_v} \phi_v(x_v) \frac{P(x_v|\xi)}{\int_\xi \psi(\xi) P(x_v|\xi) d\xi} dx_v}{\cancel{\psi(\xi)}^{N-1}} \tag{34}$$

$$\psi(\xi) = \psi(\xi) \prod_{v \in V} \int_{x_v} \phi_v(x_v) \frac{P(x_v|\xi)}{\int_\xi \psi(\xi) P(x_v|\xi) d\xi} dx_v \tag{35}$$

As in the single view case, both sides of the equation contain the desired deconvolved distribution $\psi(\xi)$. This again suggests the final iterative scheme

$$\psi^{r+1}(\xi) = \psi^r(\xi) \prod_{v \in V} \int_{x_v} \frac{\phi_v(x_v)}{\int_\xi \psi^r(\xi) P(x_v|\xi) d\xi} P(x_v|\xi) dx_v \tag{36}$$

where $\psi^0(\xi)$ is considered a distribution with a constant value. Note that the final derived equation 36 ends up being the per pixel multiplication of the single view RL-updates from equation 11.

It is important to note that the maximum-likelihood expectation-maximization based derivation[1] yields an additive combination of the individual RL-updates, while our derivation based probability theory and *Bayes' Theorem* ends up being a multiplicative combination. However, our derivation enables us to prove (section 3) that this formulation can be achieved without assuming independence of the observed distributions (input views), which allows us to introduce optimizations to the derviation (sections 4 and 6). We additionally proof of in section 2 the convergence of our multiplicative derivation to the maximum-likelihood solution.

SUPPLEMENTARY NOTE 1.3: Expression in convolution algebra

In order to be able to efficiently compute equation 36 the integrals need to be expressed as convolutions, which can be computed in *Fourier Space* using the *Convolution Theorem*. Expressing equation 36 in convolution algebra (see also equation 1) requires two assumptions. Firstly, we assume the point spread functions $P(x_v|\xi)$ to be constant for every location in space. Secondly, we assume that the different coordinate systems ξ and $x_1 \ldots x_N$ are identical, i.e. they are related by an identity transformation. We can assume that, as prior to the deconvolution the datasets have been aligned using the bead-based registration algorithm.[7] The reformulation yields

$$\psi^{r+1} = \psi^r \prod_{v \in V} \frac{\phi_v}{\psi^r * P_v} * P_v^* \tag{37}$$

where $*$ refers to the convolution operator, $\cdot$ and $\prod$ to scalar multiplication, $-$ to scalar division and

$$P_v \equiv P(x_v|\xi) \tag{38}$$

$$\phi_v \equiv \phi_v(x_v) \tag{39}$$

$$\psi^r \equiv \psi^r(\xi) \tag{40}$$

Note that P_v^* refers to the mirrored version of kernel P_v (see section 1.1.1 for the explanation).

Nature Methods: doi:10.1038/nmeth.2929

SUPPLEMENTARY NOTE 2: PROOF OF CONVERGENCE FOR BAYESIAN-BASED MULTI-VIEW DECONVOLUTION

This section proofs that our Bayesian-based derivation of multi-view deconvolution (equation 36) converges to the maximum-likelihood (ML) solution using noise-free data. We chose to adapt the proof developed for *Ordered Subset Expectation Maximization* (OS-EM)[5] due to its similarity to our derivation (see section 5).

SUPPLEMENTARY NOTE 2.1: PROOF FOR NOISE-FREE DATA

Assuming the existence of a feasible solution ψ^* it has been shown[5] that the likelihood $L^r := L(\psi^r; \psi^*)$ of the solution ψ at iteration r can be computed as

$$L(\psi^r; \psi^*) = -\int_{\xi} \psi^*(\xi) \log \frac{\psi^*(\xi)}{\psi^r(\xi)} d\xi \tag{41}$$

Following the argumentations of Shepp and Vardi,[1] Kaufmann[13] and Hudson and Larkin[5] convergence of the algorithm is proven if the likelihood of the solution ψ increases with every iteration since L is bounded by 0. In other words

$$\begin{aligned} \Delta L &= L^{r+1} - L^r && (42)\\ &\geq 0 && (43) \end{aligned}$$

We will now prove that ΔL is indeed always greater or equal to zero. Replacing equation 41 in equation 42 yields

$$\begin{aligned} \Delta L &= \int_{\xi} \psi^*(\xi) \log \frac{\psi^*(\xi)}{\psi^r(\xi)} - \psi^*(\xi) \log \frac{\psi^*(\xi)}{\psi^{r+1}(\xi)} d\xi && (44)\\ &= \int_{\xi} \psi^*(\xi) \log \frac{\psi^{r+1}(\xi)}{\psi^r(\xi)} d\xi && (45) \end{aligned}$$

Next, we substitute $\psi^{r+1}(\xi)$ with our derivation of Bayesian-based multi-view deconvolution (equation 36). Note that for simplicity we replace $\int_{\xi} \psi^r(\xi) P(x_v|\xi) d\xi$ with $\phi_v^r(x_v)$, which refers to the current estimate of the observed distribution given the current guess of the underlying distribution ψ^r (or in EM terms the *expected value*).

$$\begin{aligned} \Delta L &= \int_{\xi} \psi^*(\xi) \log \frac{\psi^r(\xi) \left(\prod_{v \in V} \int_{x_v} \frac{\phi_v(x_v)}{\phi_v^r(x_v)} P(x_v|\xi) dx_v \right)^{\frac{1}{|V|}}}{\psi^r(\xi)} d\xi && (46)\\ &= \int_{\xi} \psi^*(\xi) \log \left(\prod_{v \in V} \int_{x_v} \frac{\phi_v(x_v)}{\phi_v^r(x_v)} P(x_v|\xi) dx_v \right)^{\frac{1}{|V|}} d\xi && (47)\\ &= \frac{1}{|V|} \sum_{v \in V} \int_{\xi} \psi^*(\xi) \log \int_{x_v} \frac{\phi_v(x_v)}{\phi_v^r(x_v)} P(x_v|\xi) dx_v d\xi && (48) \end{aligned}$$

As equation 36 expresses a proportion, it is necessary to normalize for the number of observed distributions $|V|$ and apply the $|V|$'th root, i.e. compute the geometric mean. Note that this normalization is the equivalent to the division by $|V|$ as applied in the ML-EM derivations[1,3] that use the arithmetic mean of the individual RL-updates in order to update underlying distribution.

Using Jensen's inequality equation 48 can be reformulated (equation 49) and further simplified

$$\begin{aligned} \Delta L &\geq \frac{1}{|V|} \sum_{v \in V} \int_{\xi} \psi^*(\xi) \int_{x_v} \log \left(\frac{\phi_v(x_v)}{\phi_v^r(x_v)} \right) P(x_v|\xi) dx_v d\xi && (49)\\ &= \frac{1}{|V|} \sum_{v \in V} \int_{x_v} \log \left(\frac{\phi_v(x_v)}{\phi_v^r(x_v)} \right) \int_{\xi} \psi^*(\xi) P(x_v|\xi) d\xi dx_v && (50) \end{aligned}$$

Nature Methods: doi:10.1038/nmeth.2929

Substituting equation 1 in equation 50 yields

$$\Delta L \geq \frac{1}{|V|} \sum_{v \in V} \int_{x_v} \log \left(\frac{\phi_v(x_v)}{\phi_v^r(x_v)} \right) \phi_v(x_v) dx_v \tag{51}$$

It follows directly that in order to prove that $\Delta L \geq 0$, it is sufficient to prove that

$$\int_{x_v} \log \left(\frac{\phi_v(x_v)}{\phi_v^r(x_v)} \right) \phi_v(x_v) dx_v \geq 0 \tag{52}$$

Based on the inequality $\log x \geq 1 - x^{-1}$ for $x > 0$ proven by *Adolf Hurwitz*, we need to show that

$$\begin{aligned}
\int_{x_v} \log \left(\frac{\phi_v(x_v)}{\phi_v^r(x_v)} \right) \phi_v(x_v) dx_v &\geq \int_{x_v} \left(1 - \frac{\phi_v^r(x_v)}{\phi_v(x_v)} \right) \phi_v(x_v) dx_v \geq 0 && (53) \\
&= \int_{x_v} \phi_v(x_v) dx_v - \int_{x_v} \phi_v^r(x_v) dx_v && (54) \\
&= \int_{x_v} \phi_v(x_v) dx_v - \int_{x_v} \int_\xi \psi^r(\xi) P(x_v|\xi) d\xi dx_v && (55) \\
&= \int_{x_v} \phi_v(x_v) dx_v - \int_\xi \psi^r(\xi) \overbrace{\int_{x_v} P(x_v|\xi) dx_v}^{=1} d\xi && (56) \\
&= \int_{x_v} \phi_v(x_v) dx_v - \int_\xi \psi^r(\xi) d\xi && (57) \\
&\geq 0 \iff \int_{x_v} \phi_v(x_v) dx_v \geq \int_\xi \psi^r(\xi) d\xi && (58)
\end{aligned}$$

In other words, convergence is proven if we show that the energy of the underlying distribution $\psi(\xi)$ (deconvolved image) is never greater than energy of each observed distribution $\phi_v(x_v)$ (input views). Replacing $\psi^r(\xi)$ with our Bayesian-based derivation (equation 36) shows that proving the condition in equation 58 is equivalent to proving

$$\int_{x_v} \phi_v(x_v) dx_v \geq \int_\xi \psi^r(\xi) \left(\prod_{v \in V} \int_{x_v} \frac{\phi_v(x_v)}{\phi_v^r(x_v)} P(x_v|\xi) dx_v \right)^{\frac{1}{|V|}} d\xi \tag{59}$$

As this inequality has to hold for any iteration r, we omit writing $r - 1$ for simplicity. As the arithmetic average is always greater or equal than the geometric average[14] it follows that proving equation 59 is equivalent to

$$\int_{x_v} \phi_v(x_v) dx_v \geq \int_\xi \psi^r(\xi) \frac{1}{|V|} \sum_{v \in V} \int_{x_v} \frac{\phi_v(x_v)}{\phi_v^r(x_v)} P(x_v|\xi) dx_v d\xi \geq \int_\xi \psi^r(\xi) \left(\prod_{v \in V} \int_{x_v} \frac{\phi_v(x_v)}{\phi_v^r(x_v)} P(x_v|\xi) dx_v \right)^{\frac{1}{|V|}} d\xi \tag{60}$$

and we therefore need to prove the inequality only for the arithmetic average. We simplify equation 60 as follows

$$\begin{aligned}
\int_{x_v} \phi_v(x_v) dx_v &\geq \frac{1}{|V|} \sum_{v \in V} \int_\xi \psi^r(\xi) \int_{x_v} \frac{\phi_v(x_v)}{\phi_v^r(x_v)} P(x_v|\xi) dx_v d\xi && (61) \\
&= \frac{1}{|V|} \sum_{v \in V} \int_{x_v} \frac{\phi_v(x_v)}{\phi_v^r(x_v)} \int_\xi \psi^r(\xi) P(x_v|\xi) d\xi dx_v && (62) \\
&= \frac{1}{|V|} \sum_{v \in V} \int_{x_v} \frac{\phi_v(x_v)}{\phi_v^r(x_v)} \phi_v^r(x_v) dx_v && (63) \\
&= \frac{1}{|V|} \sum_{v \in V} \int_{x_v} \phi_v(x_v) dx_v && (64)
\end{aligned}$$

As the integral of all input views is identical (equation 2), equation 64 is always true, and we proved that our Bayesian-based derivation of multi-view deconvolution always converges to the Maximum Likelihood.

Nature Methods: doi:10.1038/nmeth.2929

SUPPLEMENTARY NOTE 3: DERIVATION OF BAYESIAN-BASED MULTI-VIEW DECONVOLUTION WITHOUT ASSUMING VIEW INDEPENDENCE

Previous derivations of the Richardson-Lucy multi-view deconvolution[1–3,12] assumed independence of the individual views in order to derive variants of equation 36. The following derivation shows that it is actually not necessary to assume independence of the observed distributions $\phi_v(x_v)$ (equation 20 $\rightarrow$ 21 and equation 23 $\rightarrow$ 24) in order to derive the formulation for Bayesian-based multi-view deconvolution shown in equation 36.

We therefore rewrite equation 21 without assuming independence (which is then identical to equation 20) and obtain

$$P(\xi) = \int_{\overline{x}} P(x_1)P(x_2|x_1)\dots P(x_N|x_1,...,x_{N-1})P(\xi|x_1,...,x_N)d\overline{x} \tag{65}$$

We consequently also do not assume independence in equation 24, which intends to replace $P(\xi|x_1,...,x_N)$, and obtain

$$P(\xi|x_1,...,x_N) = \frac{P(\xi)P(x_1|\xi)\dots P(x_N|\xi)}{P(x_1)P(x_2|x_1)\dots P(x_N|x_1,...,x_{N-1})} \tag{66}$$

Replacing equation 66 in 65 yields

$$P(\xi) = \int_{\overline{x}} P(x_1)P(x_2|x_1)\dots P(x_N|x_1,...,x_{N-1})\frac{P(\xi)P(x_1|\xi)\dots P(x_N|\xi)}{P(x_1)P(x_2|x_1)\dots P(x_N|x_1,...,x_{N-1})}d\overline{x} \tag{67}$$

Cancelling out all terms below the fraction bar (from equation 66) with the terms in front of the fraction bar (from equation 65) results in

$$P(\xi) = \int_{\overline{x}} P(\xi)P(x_1|\xi)\dots P(x_N|\xi)d\overline{x} \tag{68}$$

Using again *Bayes' Theorem* to replace all

$$P(x_v|\xi) = \frac{P(x_v)P(\xi|x_v)}{P(\xi)} \tag{69}$$

yields

$$P(\xi) = \frac{\int_{\overline{x}} P(\xi)P(x_1)P(\xi|x_1)\dots P(x_N)P(\xi|x_N)d\overline{x}}{P(\xi)^N} \tag{70}$$

$$P(\xi) = \frac{\int_{\overline{x}} P(x_1)\dots P(x_N)P(\xi|x_1)\dots P(\xi|x_N)d\overline{x}}{P(\xi)^{N-1}} \tag{71}$$

which is identical to equation 29. This proofs that we can derive the final equation 36 without assuming independence of the observed distributions.

Nature Methods: doi:10.1038/nmeth.2929

SUPPLEMENTARY NOTE 4: EFFICIENT BAYESIAN-BASED MULTI-VIEW DECONVOLUTION

Section 3 shows that the derivation of Bayesian-based multi-view deconvolution does not require the assumption that the observed distributions (views) are independent. We want to take advantage of that and incorporate the relationship between them into the deconvolution process to reduce convergence time. In order to express these dependencies we need to understand and model the conditional probabilities $P(x_w|x_v)$ describing how one view $\phi_w(x_w)$ depends on another view $\phi_v(x_v)$.

SUPPLEMENTARY NOTE 4.1: MODELING CONDITIONAL PROBABILITIES

Let us assume that we made an observation $x_v = x'_v$ (see also supplementary figure 1b). The 'inverse' point spread function $Q(\xi|x_v)$ defines a probability for each location of the underlying distribution that it caused the event $\xi = \xi'$ that lead to this observation. Based on this probability distribution, the point spread function of any other observation $P(x_w|\xi)$ can be used to consecutively assign a probability to every of its locations defining how probable it is to expect an observation $x_w = x'_w$ corresponding to $x_v = x'_v$. Assuming the point spread function $P(x_w|\xi)$ is known, this illustrates that we are able to estimate the conditional probability $P(x_w|x_v = x'_v)$ for every location $x_w = x'_w$ as well as we can estimate the 'inverse' point spread function $Q(\xi|x_v)$.

However, we want to be able to compute an entire 'virtual' distribution, which is based on not only one singluar event $x_v = x'_v$, but an entire observed distribution $\phi_v(x_v)$. Such a 'virtual' distribution is solely based on the conditional probabilities $P(x_w|x_v)$ and summarizes our knowledge about a distribution $\phi_w(x_w)$ by just observing $\phi_v(x_v)$ and knowing the (inverse) point spread functions $Q(\xi|x_v)$ and $P(x_w|\xi)$. We denote a 'virtual' distribution $\phi_v^{V_w}(x_w)$; the subscript v denotes the observed distribution it is based on, w defines the distribution that is estimated and V labels it as 'virtual' distribution.

The derviation of the formulation for a 'virtual' distribution is based on equations 1 and 8. The 'inverse' point spread function $Q(\xi|x_w)$ relates $\phi_v(x_v)$ to the underlying signal distribution $\psi(\xi)$, and the point spread function $P(x_w|\xi)$ consecutively relates it to the conditionally dependent signal distribution $\phi_w(x_w)$ (see also supplementary figure 1b)

$$\psi(\xi) = \int_{x_v} \phi_v(x_v) Q(\xi|x_v) dx_v \quad (72)$$

$$\phi_w(x_w) = \int_{\xi} \psi(\xi) P(x_w|\xi) d\xi \quad (73)$$

Substituting equation 72 in equation 73 yields

$$\phi_w(x_w) = \int_{\xi} \int_{x_v} \phi_v(x_v) Q(\xi|x_v) dx_v P(x_w|\xi) d\xi \quad (74)$$

As discussed in sections 1.1 and 1.2, we cannot use $Q(\xi|x_v)$ directly to compute $\psi(\xi)$. Using *Bayes' theorem* it can be rewritten as

$$Q(\xi|x_v) = \frac{\psi(\xi) P(x_v|\xi)}{\phi_v(x_v)} \quad (75)$$

Assuming $\phi_v(x_v)$ and $\psi(\xi)$ constant (or rather identical) simplifies equation 75 to

$$Q(\xi|x_v) = P(x_v|\xi) \quad (76)$$

This assumption reflects that initially we do not have any prior knowledge of $Q(\xi|x_v)$ and therefore need to set it equal to the PSF $P(x_v|\xi)$, which states the worst-case scenario. In other words, the PSF constitutes an *upper bound* for all possible locations of the underlying distribution $\psi(\xi)$ that could contribute to the observed distribution given an observation at a specific location $x_v = x'_v$. Thus, this assumption renders the estimate $\phi_v^{V_w}(x_w)$ less precise (equation 77 and main text figure 1c), while not omitting any of the possible solutions. Note that it would be possible to improve the guess of $Q(\xi|x_v)$ after every iteration. However, it would require a

Nature Methods: doi:10.1038/nmeth.2929

convolution with a different PSF at every location, which is currently computationally not feasible and is therefore omitted. Replacing equation 76 in equation 74 yields

$$\phi_w(x_w) \approx \phi_v^{V_w}(x_w) = \int_\xi \int_{x_v} \phi_v(x_v) P(x_v|\xi) dx_v P(x_w|\xi) d\xi \tag{77}$$

Equation 77 enables the estimation of entire 'virtual' distributions $\phi_v^{V_w}(x_w)$, see and main text figure 1c for a visualization. These 'virtual' distributions constitute an *upper boundary* describing how a distribution $\phi_w(x_w) \approx \phi_v^{V_w}(x_w)$ could look like while only knowing $\phi_v(x_v)$ and the two PSF's $P(x_v|\xi)$ and $P(x_w|\xi)$. We denote it *upper boundary* as it describes the combination of all possiblities of how a observed distribution $\phi_w(x_w)$ can look like.

SUPPLEMENTARY NOTE 4.2: INCORPORATING VIRTUAL VIEWS INTO THE DECONVOLUTION SCHEME

In order to incorporate this knowledge into the deconvolution process (equation 36), we perform updates not only based on the observed distributions $\phi_v(x_v)$ but also all possible virtual distributions $\phi_v^{V_w}(x_w)$ as modelled by equation 77 and shown in and main text figure 1c. Based on all observed distributions

$$V = \{\phi_1(x_1), \phi_2(x_2), ...\phi_N(x_N)\} \tag{78}$$

we can estimate the following 'virtual' distributions

$$W = \{\phi_1^{V_2}(x_2), \phi_1^{V_3}(x_3), ..., \phi_1^{V_N}(x_N), \phi_2^{V_1}(x_1), \phi_2^{V_3}(x_3), ..., \phi_2^{V_N}(x_N), ..., \phi_N^{V_{N-1}}(x_{N-1})\} \tag{79}$$

where

$$|W| = (N-1)^N \;:\; N = |V| \tag{80}$$

Note that if only one input view exists, $W = \emptyset$. We define subsets $W_v \subseteq W$, which depend on specific observed distributions $\phi_v(x_v)$ as follows

$$W_1 = \{\phi_1^{V_2}(x_2), \phi_1^{V_3}(x_3), ..., \phi_1^{V_N}(x_N)\} \tag{81}$$

$$W_2 = \{\phi_2^{V_1}(x_1), \phi_2^{V_3}(x_3), ..., \phi_2^{V_N}(x_N)\} \tag{82}$$

$$... \tag{83}$$

$$W_N = \{\phi_N^{V_1}(x_1), \phi_N^{V_2}(x_2), ..., \phi_N^{V_{N-1}}(x_{N-1})\} \tag{84}$$

where

$$W = \bigcup_{v \in V} W_v \tag{85}$$

Incorporating the virtual distributions into the multi-view deconvolution (equation 36) yields

$$\psi^{r+1}(\xi) = \psi^r(\xi) \prod_{v \in V} \int_{x_v} \frac{\phi_v(x_v)}{\int_\xi \psi^r(\xi) P(x_v|\xi) d\xi} P(x_v|\xi) dx_v \prod_{w \in W_v} \int_{x_w} \frac{\phi_v^{V_w}(x_w)}{\int_\xi \psi^r(\xi) P(x_w|\xi) d\xi} P(x_w|\xi) dx_w \tag{86}$$

This formulation is simply a combination of observed and 'virtual' distributions, which does not yield any advantages in terms of computational complexity yet. During the following steps we will show that using a single assumption we are able to combine the update steps of the observed and 'virtual' distributions into one single update step for each observed distribution.

For simplicity we focus on one oberved distribution $\phi_v(x_v)$ and its corresponding subset of 'virtual' distributions W_v. Note that the following assumptions and simplifications apply to all subsets individually.

$$\int_{x_v} \frac{\phi_v(x_v)}{\int_\xi \psi^r(\xi) P(x_v|\xi) d\xi} P(x_v|\xi) dx_v \prod_{w \in W_v} \int_{x_w} \frac{\phi_v^{V_w}(x_w)}{\int_\xi \psi^r(\xi) P(x_w|\xi) d\xi} P(x_w|\xi) dx_w \tag{87}$$

Nature Methods: doi:10.1038/nmeth.2929

First, the 'virtual' distributions $\phi_v^{V_w}(x_w)$ are replaced with equation 77 which yields

$$\int_{x_v} \frac{\phi_v(x_v)}{\int_\xi \psi^r(\xi)P(x_v|\xi)d\xi} P(x_v|\xi)dx_v \prod_{w\in W_v} \int_{x_w} \frac{\int_\xi \int_{x_v} \phi_v(x_v)P(x_v|\xi)dx_v P(x_w|\xi)d\xi}{\int_\xi \psi^r(\xi)P(x_w|\xi)d\xi} P(x_w|\xi)dx_w \tag{88}$$

Note that $\int_\xi \psi^r(\xi)P(x_w|\xi)d\xi$ corresponds to our current guess of the observed distribution $\phi_w(x_w)$, which is based on the current guess of the underlying distribution $\psi^r(\xi)$ and the point spread function $P(x_w|\xi)$ (equation 1). In order to transform it into a 'virtually' observed distribution compatible with $\phi_v^{V_w}(x_w)$, we also apply equation 77, i.e. we compute it from the current guess of the observed distribution $\phi_v(x_v)$ yielding

$$\int_{x_v} \frac{\phi_v(x_v)}{\int_\xi \psi^r(\xi)P(x_v|\xi)d\xi} P(x_v|\xi)dx_v \prod_{w\in W_v} \int_{x_w} \frac{\int_\xi \int_{x_v} \phi_v(x_v)P(x_v|\xi)dx_v P(x_w|\xi)d\xi}{\int_\xi \int_{x_v} \int_\xi \psi^r(\xi)P(x_v|\xi)d\xi P(x_v|\xi)dx_v P(x_w|\xi)d\xi} P(x_w|\xi)dx_w \tag{89}$$

To better illustrate the final simplifications we transform equation 89 into convolution algebra (section 1.3). The reformulation yields

$$\frac{\phi_v}{\psi^r * P_v} * P_v^* \prod_{w\in W_v} \frac{\phi_v * \overbrace{P_v^* * P_w}}{\psi^r * P_v * \underbrace{P_v^* * P_w}} * P_w^* \tag{90}$$

Additional simplification of equation 90 requires an assumption in convolution algebra that we incorporate twice. Given three functions f, g and h we assume

$$(f * g) \cdot (f * h) \approx f * (g \cdot h) \tag{91}$$

We illustrate in supplementary figure 3 on a one-dimensional and two-dimensional example that for *Gaussian*-like distributions this assumption may hold true after normalization of both sides of the equation. Note that the measured PSF's usually resemble a distribution similiar to a gaussian (supplementary figure 5 and 17).

The numerator and the denominator of the ratio of the 'virtual' distribution in equation 90 both contain two consecutive convolutions with P_v^* and P_w as indicated by brackets. Based on equation 91 we assume

$$\frac{(g * f)}{(h * f)} \approx \left(\frac{g}{h}\right) * f \tag{92}$$

where

$$f \equiv P_v^* * P_w \tag{93}$$
$$g \equiv \phi_v \tag{94}$$
$$h \equiv \psi^r * P_v \tag{95}$$

and

$$\frac{(g * f)}{(h * f)} = (g * f) \cdot \frac{1}{(h * f)} = (g * f) \cdot \left(\frac{1}{h} * f\right) = \overbrace{(f * g) \cdot \left(f * \frac{1}{h}\right) \approx f * \left(g \cdot \frac{1}{h}\right)}^{equation\ 91} = f * \left(\frac{g}{h}\right) = \left(\frac{g}{h}\right) * f \tag{96}$$

Based on this assumption we can rewrite equation 90 as

$$\overbrace{\frac{\phi_v}{\psi^r * P_v}} * P_v^* \prod_{w\in W_v} \overbrace{\frac{\phi_v}{\psi^r * P_v}} * P_v^* * P_w * P_w^* \tag{97}$$

Note that this reformulation yields two identical terms as outlined by brackets describing the ratio between the observed distribution ϕ_v and its guess based on the current iteration of the deconvolved distribution $\psi^r * P_v$. To

Nature Methods: doi:10.1038/nmeth.2929

further simplify equation 97 we apply the assumption (equation 91) again where

$$f \equiv \frac{\phi_v}{\psi^r * P_v} \tag{98}$$

$$g \equiv P_v^* \tag{99}$$

$$h \equiv P_v^* * P_w * P_w^* \tag{100}$$

which yields

$$\frac{\phi_v}{\psi^r * P_v} * \left(P_v^* \prod_{w \in W_v} P_v^* * P_w * P_w^* \right) \tag{101}$$

In the context of all observed distributions, the final formula for efficient Bayesian-based multi-view deconvolution reads

$$\psi^{r+1} = \psi^r \prod_{v \in V} \frac{\phi_v}{\psi^r * P_v} * \overbrace{\left(P_v^* \prod_{w \in W_v} P_v^* * P_w * P_w^* \right)}^{P_v^{compound}} \tag{102}$$

Equation 102 incorporates all observerd and 'virtual' distributions that speed up convergence, but requires the exactly same number of computations as the normal multi-view deconvolution (equation 36) derived in section 1.2. The only additional computational overhead is the initial computation of the *compound* kernels for each observed distribution

$$P_v^{compound} = P_v^* \prod_{w \in W_v} \overbrace{P_v^* * P_w * P_w^*}^{P_{w_v}^{compound}} \tag{103}$$

The compound kernel for a specific observed distribution ϕ_v is computed by scalar multiplication of its mirrored point spread function P_v^* with all 'virtual' compound kernels $P_{w_v}^{compound}$ based on the corresponding 'virtual' distributions $\phi_v^{V_w}(x_w) \in W_v$. All individual 'virtual' compound kernels are computed by convolving P_v^* with P_w and sequentially with P_w^*. For most multi-view deconvolution scenarios the computational effort for the pre-computation of the compound kernels can be neglected as the PSF's a very small compared to the images and they need to be computed only once.

Nature Methods: doi:10.1038/nmeth.2929

SUPPLEMENTARY NOTE 5: ALTERNATIVE ITERATION FOR FASTER CONVERGENCE

To further optimize convergence time we investigated the equations 36 and 102 in detail. Both multi-view deconvolution formulas evaluate all views in order to compute one single update step of $\psi(\xi)$. It was already noted that in both cases each update step is simply the ***multiplication*** of all contributions from each observed distribution. This directly suggests an alternative update scheme where the individual contributions from each observed distribution are directly multiplied to update $\psi(\xi)$ in order to save computation time. In this iteration scheme, equation 102 reads as follows

$$\psi^{r+1} = \psi^r \frac{\phi_1}{\psi^r * P_1} * \left(P_1^* \prod_{w \in W_1} P_1^* * P_w * P_w^* \right) \tag{104}$$

$$\psi^{r+2} = \psi^{r+1} \frac{\phi_2}{\psi^{r+1} * P_2} * \left(P_2^* \prod_{w \in W_2} P_2^* * P_w * P_w^* \right) \tag{105}$$

$$... \tag{106}$$

$$\psi^{r+N} = \psi^{r+N-1} \frac{\phi_N}{\psi^{r+N-1} * P_N} * \left(P_N^* \prod_{w \in W_N} P_N^* * P_w * P_w^* \right) \tag{107}$$

$$\tag{108}$$

Note that for equation 36 (equation 37) the iterative scheme looks identical when the multiplicative part of the compound kernel is left out; it actually corresponds to the sequential application of the standard Richardson-Lucy (RL) updates (equation 11), which corresponds to the principle of *Ordered Subset Expectation Maximization*[5] (OS-EM, see section 5.1).

SUPPLEMENTARY NOTE 5.1: RELATIONSHIP TO ORDERED SUBSET EXPECTION MAXIMIZATION (OS-EM)

The principle of OS-EM is the sequential application of *subsets* of the observed data to the underlying distribution $\psi(\xi)$ using standard Richardson-Lucy (RL) updates (equation 11). Note that in a multi-view deconvolution scenario, each observed distribution $\phi_v(x_v)$ is equivalent to the OS-EM definition of a *balanced subset* as all elements of the underlying distribution are updated for each $\phi_v(x_v)$.

$$\psi^{r+1}(\xi) = \psi^r(\xi) \int_{x_1} \frac{\phi_1(x_1)}{\int_\xi \psi^r(\xi) P(x_1|\xi) d\xi} P(x_1|\xi) dx_1 \tag{109}$$

$$\psi^{r+2}(\xi) = \psi^{r+1}(\xi) \int_{x_2} \frac{\phi_2(x_2)}{\int_\xi \psi^{r+1}(\xi) P(x_2|\xi) d\xi} P(x_2|\xi) dx_2 \tag{110}$$

$$... \tag{111}$$

$$\psi^{r+N}(\xi) = \psi^{r+N-1}(\xi) \int_{x_N} \frac{\phi_N(x_N)}{\int_\xi \psi^{r+N-1}(\xi) P(x_N|\xi) d\xi} P(x_N|\xi) dx_N \tag{112}$$

As pointed out in the main text, the obvious relationship to OS-EM is that the sequential application of RL-updates is directly suggested by our multiplicative derivation (equation 36), compared to the additive EM derivation.[1]

Nature Methods: doi:10.1038/nmeth.2929

SUPPLEMENTARY NOTE 6: AD-HOC OPTIMIZATIONS OF THE EFFICIENT BAYESIAN-BASED MULTI-VIEW DECONVOLUTION

The efficient Bayesian-based multi-view deconvolution derived in section 4 offers possibilites for optimizations as the assumption underlying the estimation of the conditional probabilities (section 4.1) results in smoothed guess of the 'virtual' distributions (and main text figure 1c). Therefore, the core idea underlying all subsequently presented alterations of equation 102 is to change how the 'virtual' distributions are computed. Due to the optimizations introduced in the last section, this translates to modification of the 'virtual' compound kernels

$$P_{w_v}^{compound} = P_v^* * P_w * P_w^* \tag{113}$$

The goal is to decrease convergence time while preserving reasonable deconvolution results. This can be achieved by sharpening the 'virtual' distribution (and main text figure 1c) without omitting possible solutions or rendering them too unlikely.

SUPPLEMENTARY NOTE 6.1: OPTIMIZATION I - REDUCED DEPENDENCE ON VIRTUALIZED VIEW

The computation of the 'virtual' compound kernels contains two convolutions with the point spread function of the 'virtualized' observation, one with P_w and one with P_w^*. We found that skipping the convolution with P_w^* significantly reduces convergence time while producing almost identical results even in the presence of noise (section 7 and 7.5).

$$\psi^{r+1} = \psi^r \prod_{v \in V} \frac{\phi_v}{\psi^r * P_v} * \left(P_v^* \prod_{w \in W_v} P_v^* * P_w \right) \tag{114}$$

SUPPLEMENTARY NOTE 6.2: OPTIMIZATION II - NO DEPENDENCE ON VIRTUALIZED VIEW

We determined empirically that further assuming P_w to be constant still produces reasonable results while further reducing convergence time. We are aware that this is quite an ad-hoc assumption, but in the presence of low noise levels still yields adequate results (section 7 and 7.5).

$$\psi^{r+1} = \psi^r \prod_{v \in V} \frac{\phi_v}{\psi^r * P_v} * \prod_{v, w \in W_v} P_v^* \tag{115}$$

Interestingly, this formulation shows some similarity to an optimization of the classic single-view Richardson-Lucy deconvolution, which incorporates an exponent into the entire 'correction factor',[15] not only the PSF for the second convolution operation. Our derivation of the efficient Bayesian-based multi-view scenario intrinsically provides the exponent for the second convolution that can be used to speed up computation and achieve reasonable results.

SUPPLEMENTARY NOTE 6.3: NO DEPENDENCE ON OBSERVED VIEW

Only keeping the convolutions of the 'virtualized' observation, i.e. P_w and P_w^* yields a non-functional formulation. This is in agreement with the estimation of the conditional probabilities (section 4.1).

Nature Methods: doi:10.1038/nmeth.2929

SUPPLEMENTARY NOTE 7: BENCHMARKS & ANALYSES

We compare the performance of our new derivations against classical multi-view deconvolution (section 7.1) and against other optimized multi-view deconvolution schemes (section 7.2) using ground truth images. Subsequently, we investigate the general image quality (section 7.3), the dependence on the PSF's (section 7.4), analyze the effect of noise and regularization (section 7.5) and show the result of imperfect point spread functions (section 7.6).

The iteration behaviour of the deconvolution depends on the image content and the shape of the PSF (supplementary figure 4d). In order to make the simulations relatively realistic for microscopic multi-view acquisitions, we chose as ground truth image one plane of a SPIM acquistion of a *Drosophila* embryo expressing His-YFP in all cells (supplementary figure 5a,e) that we blur with a maximum intensity projection of a PSF in axial direction (xz), extracted from an actual SPIM acquistion (supplementary figure 5e).

SUPPLEMENTARY NOTE 7.1: CONVERGENCE TIME, NUMBER OF ITERATIONS & UPDATES COMPARED TO CLASSICAL MULTI-VIEW DECONVOLUTION

First, we compare the performance of the efficient Bayesian-based multi-view deconvolution (section 4, equation 102) and its optimizations I & II (sections 6.1 and 6.2, equations 114 and 115) against the Bayesian-based derivation (sections 1.2 and 3, equations 36 and 37) and the original Maximum-Likelihood Expectation-Maximization derviation[1] for combined (sections 1.2 – 6) and sequential (section 5, OSEM[5]) updates of the underlying distribution.

Supplementary figure 4a-c illustrate computation time, number of iterations and number of updates of the underlying distribution that are required by the different derivations to converge to a point, where they achieve exactly the same average difference between the deconvolved image and the ground truth. Detailed parts of the ground truth, PSFs, input images and results used for supplementary figure 4 are exemplarily pictured for 4 views in supplementary figure 5e, illustrating that all algorithms actually converge to the same result. The entire ground truth picture is shown in supplementary figure 5a, the deconvolution result as achieved in the benchmarks is shown in supplementary figure 5b.

Supplementary figure 4a shows that our efficient Bayesian-based deconvolution (equation 102) outperforms the Bayesian-based deconvolution (equation 36) by a factor of around 1.5–2.5, depending on the number of views involved. Optimization I is faster by a factor of 2–4, optimization II by a factor of 3–8. Sequential updates (OSEM) pictured in red additionally speed up the computation by a factor of approximately n, where n describes the number of views involved. This additional multiplicative speed-up is independent of the derivation used. Supplementary figure 4b illustrates that our Bayesian-based deconvolution behaves very similar to the Maximum-Likelihood Expectation-Maximization method.[1] Note the logarithmic scale of all y-axes in supplementary figure 4.

It is striking that for combined updates (black) the computation time first decreases, but quickly starts to increase with the number of views involved. In contrast, for sequential updates (OSEM) the computation time descreases and then plateaus. The increase in computation time becomes clear when investigating the required number of iterations[††] (supplementary figure 4b). The number of iteration for combined updates (black) almost plateaus at a certain level, however, with increasing number of views, the computational effort to compute one update increases linearly. This leads to an almost linear increase in convergence time with an increasing number of views when using combined updates. When using sequential updates (red), the underlying distribution is updated for each view individually, hence the number of required iterations continuously decreases and only the convergence time plateaus with an increasing number of views. Supplementary figure 4c supports this interpretation by illustrating that for each derivation the number of updates of the underlying distribution defines when the same quality of deconvolution is achieved.

[††]Note that we consider one iteration completed when all views contributed to update the underlying distribution once. In the case of combined updates this refers to one update of the underlying distribution, in case of sequential updates this refers to n updates.

Nature Methods: doi:10.1038/nmeth.2929

In any case, having more than one view available for the deconvolution process decreases computation time and number of required updates significantly. This effect is especially prominent at a low number of views. For example adding a second view decreases the computation time in average 45–fold, a third view still on average another 1.5–fold.

One can argue that using combined update steps allows better parallelization of the code as all view contributions can be computed at the same time, whereas sequential updating requires to compute one view after the other. In practice, computing the update step for an individual view is already almost perfectly multi-threadable. It requires two convolutions computed in Fourier space and several per-pixel operations. Even when several GPU's are available it can be parallelized as it can be split into blocks. Using sequential updates additionally offers the advantage that the memory required for the computation is significantly reduced.

SUPPLEMENTARY NOTE 7.2: CONVERGENCE TIME, NUMBER OF ITERATIONS & UPDATES COMPARED TO OPTIMIZED MULTI-VIEW DECONVOLUTION

Previously, other efficient methods for optimized multi-view deconvolution have been proposed. We compare our methods against Scaled-Gradient-Projection (SGP),[4] Ordered Subset Expectation Maximization (OSEM)[5] and Maximum a posteriori with Gaussian Noise (MAPG).[6] Again, we let the algorithms converge until they achieve the same average difference of 0.07 to the known ground truth image.

In order to compare the different algorithms we re-implemented MAPG in Java, based on the Python source code kindly provided by Dr. Peter Verveer. In order to compare to SGP, we downloaded the IDL source code. In order to allow a reasonable convergence, it was necessary to add a constant background to the images before processing them with SGP. It was also necessary to change the size of the input data to a square, even dimension (614$\times$614) to not introduce a shift of one pixel in the deconvolved data by the IDL code. OSEM is identical to our sequential updates and therefore requires no additional implementation. Main text figure 1f illustrates that our Java implementation (Bayesian-based + OSEM) and the IDL OSEM implementation require almost the identical amount of iterations.

Regarding the number of iterations (main text figure 1f), the efficient Bayesian-based deconvolution and Optimization I perform better compared to all other efficient methods for more than 4 overlapping views, Optimization II already for more than 2 views. For example at 7 views, where OSEM (50), SGP (53) and MAPG (44) need around 50 iterations, the efficient Bayesian-based deconvolution requires 23 iteration, Optimization I 17 iterations, and Optimization II 7 iteration in order to converge to the same result.

Concerning computation time (main text figure 1e), any algorithm we implemented in Java completely outperforms any IDL implementation. For the almost identical implementation of OSEM Java is in average 8$\times$ faster than IDL on the same machine (2$\times$ Intel Xeon E5-2680, 128 GB RAM), which slightly increases with the number of views (7.89$\times$ for 2 views, 8.7$\times$ for 11 views). Practically, Optimization II outperforms all methods, except MAPG at 2 views. At 7 views where SGP (IDL) and OSEM (IDL) require around 80 seconds to converge, MAPG converges in 4 seconds, the efficient Bayesian-based deconvolution in 5 seconds, Optimization I in 3.7 seconds and Optimization II in 1.6 seconds.

Note that MAPG is conceptually different to all other deconvolution methods compared here. It assumes Gaussian noise and performs the deconvolution on a fused dataset, which results in a reduced reconstruction quality on real datasets (see also section 7.3.1). It also means that its computation time is theoretically independent on the number of views, a property that is shared with the classical OSEM (Supplementary figure 4c). However, it is obvious that up to 4 views, the deconvolution performance significantly increases with an increasing number of views. We speculate that the reason for this is the coverage of frequencies in the Fourier spectrum (thanks for a great discussion with H. Shroff, NIH). Each PSF view blurs the image in a certain direction, which means that certain frequencies are more preserved than others. For more than 4 views, it seems that most high frequencies are contributed by at least one of the views and therefore the performance does not increase any more for algorithms that do not take into account the relationships between the individual views. Note, that our optimized multi-view deconvolution methods still signifcantly increase their performance if more than 4 views contribute (main text figure 1d,e and supplementary figure 4a,b,c).

Nature Methods: doi:10.1038/nmeth.2929

Supplementary figure 9a plots the computation time versus the image size of the deconvolved image for a dataset consisting of 5 views. All methods behave more or less proportional, however, the IDL code is only able to process relatively small images.

Supplementary figure 9b illustrates that our optimizations can theoretically also be combined with SGP, not only OSEM. The number of iterations is in average reduced 1.4-fold for the efficient Bayesian-based deconvolution, 2.5-fold for Optimization I, and 2.5-fold for Optimization II.

SUPPLEMENTARY NOTE 7.3: VISUAL IMAGE QUALITY

Supplementary figure 5c shows the result using optimization II and sequential updates (OSEM) after 14 iterations, the same quality as achieved by all algorithms as shown in supplementary figure 5e and used for the benchmarks in supplementary figure 4. In this case the quality of the deconvolved image is sufficient to separate small details like the fluorescent beads, which is not possible in the input images (supplementary figure 5e, right top). 301 iterations almost perfectly restore the image (supplementary figure 5b,d). In comparison, the Bayesian-based derivation (equation 36) needs 301 iteration to simply arrive at the quality pictured in supplementary figure 5c,e.

SUPPLEMENTARY NOTE 7.3.1: COMPARISON TO MAPG

Supplementary figure 9c-h compares our fastest Optimization II to MAPG using the same 7-view acquisition of a *Drosophila* embryo expressing His-YFP as in main text figure 3c-e. It shows that, despite being signifcantly faster then MAPG (main text figure 1e,f), Optimization II clearly outperforms MAPG in terms of overall image quality (supplementary figure 9c-h). Using the same blending scheme (supplementary figure 10) MAPG produces artifacts close to some of the nuclei (e.g. top left of supplementary figure 9e) and enhances stripes inside the sample that arise from the beginning/ending of partially overlapping input views. Especially the lower part of supplementary figure 9c shows reduced quality, which most likely arises from the fact that in that area one input views less contributes to the final deconvolved image (note that the 7 views are equally spaced in 45 degree steps from 0–270 degrees, however every pixel is covered by at least 2 views). Note that also Optimization II shows slightly reduced image quality in this area, but is able to compensate the reduced information content significantly better.

SUPPLEMENTARY NOTE 7.4: GENERAL DEPENDENCE ON THE PSF's

For supplementary figure 4a-c the PSF's are arranged in a way so that the angular difference between them is maximal in the range from 0–180 degrees (supplementary figure 5e). Supplementary figure 4d visualizes for 4 views that the angular difference between the views significantly influences the convergence behaviour. Looking at two extreme cases explains this behaviour. In this synthetic environment a difference of 0 degrees between PSF's corresponds to 4 identical PSF's and therefore 4 identical input images. This constellation is identical to having just one view, which results in a very long convergence time (supplementary figure 4a). The same almost applies for 180 degrees as the PSF that was used is quite symmetrical. In those extreme cases our argument that we can learn something about a second view by looking at the first view (section 4.1) does not hold. Therefore our efficient Bayesian-based deconvolution as well as the optimizations do not converge to the identical result and few datapoints close and equal to 0 and 180 degrees are omitted. Note that they still achieve a reasonable result, but simply cannot be plotted as this quality of reconstruction is not achieved.

In general, convergence time decreases as the level of overlap between the PSFs decreases. In case of non-isotropic, gaussian-like PSFs rotated around the center (as in multi-view microscopy), this translates to a decrease in convergence time with an increase in angular difference. From this we can derive that for overlapping multi-view acquisitions it should be advantageous to prefer an odd over an even number of equally spaced views.

Supplementary figure 4d also illustrates that convergence time significantly depends on the shape and size of the PSF. Different PSF's require different amount of iterations until they reach the same quality. Intuitively this has to be true, as for example the most simple PSF consisting only of its central pixel does not require any

Nature Methods: doi:10.1038/nmeth.2929

deconvolution at all. Conversely, this also holds true for the images themselves; the iteration time required to reach a certain quality depends on the content. For example, the synthetic image used in Supplementary Movie 1 takes orders of magnitude longer to converge to same cross correlation of 0.99 to ground truth, compared to the image in supplementary figure 5a using the same PSF's, algorithm and iteration scheme.

SUPPLEMENTARY NOTE 7.5: NOISE AND REGULARIZATION

Although the signal-to-noise ratio is typically very high in light-sheet microscopy (see Supplementary Table 1), it is a common problem and we therefore investigated the effect of noise on the performance of the different algorithms. As Poisson noise is the dominant source of noise in light microscopy we created our simulated input views using a Poisson process with variable SNR:

$$SNR = \sqrt{N} \tag{116}$$

where N is the number of photons collected. These images were then used to run the deconvolution process. The first row in supplementary figure 6a and first column in supplementary figure 7 show the resulting input data for varying noise levels. For supplementary figure 7c we added Gaussian noise with an increasing mean to simulate the effects of Gaussian noise.

A comparison as in the previous section is unfortunately not possible as in the presence of noise none of the algorithms converges exactly towards the ground truth. Note that still very reasonable results are achieved as shown in supplementary figure 6a and 7. Therefore, we devised a different scenario to test the robustness to noise. For the case of no noise (SNR $= \infty$) we first identified the number of iterations required for each algorithm to reach the same quality (supplementary figure 6c, 1st column). With increasing noise level we iterate the exact same number of iterations for each algorithm and analyze the output.

Supplementary figure 6a,b,c show that for the typical regime of SNR's in light sheet microscopy (see Supplementary Table 1, estimation range from 15 to 63) all methods converge to visually identical results.

For low SNR's (independent of Poisson or Gaussian noise) the Bayesian-based deconvolution (equation 36), the Maximum-Likelihood Expectation-Maximization (ML-EM) and the sequential updates (OSEM) score best with almost identical results. For Poisson noise, MAPG and Optimization II show comparable results with lower quality, Optimization I and the efficient Bayesian-based derivation lie in between. For Gaussian noise, MAPG, the Bayesian-based derivation and Optimization I produce very similar results while Optimization II shows lower quality.

To compensate for noise in the deconvolution we added the option of Tikhonov-regularization. Supplementary figure 6c illustrates the influence of the λ parameter on the deconvolution results. Supplementary figure 7 shows corresponding images for all data points. We think that although the Tikhonov regularization slows down convergence (supplementary figure 6c), a low λ might be a good choice even in environments of a high SNR (supplementary figure 7).

SUPPLEMENTARY NOTE 7.6: PSF-ESTIMATION

Another common source of errors is an imprecise estimation of the PSF's. In the previous sections we always assumed to know the PSF exactly. In real life PSF's are either measured or theoretically computed and might therefore not precisely resemble the correct system PSF of the microscope due to misalignement, refractions, etc.

In order to be able to estimate the effect of using imprecise PSF's for the deconvolution we randomly rotated the PSF's we used to create the input images before applying them to the deconvolution (supplementary figure 8). We used the same scheme to analyze the results as discussed in section 7.5. Surprisingly, the effect on the deconvolution result is hardly noticable for all algorithms, even at an average rotation angle of 10 degrees. The deconvolved images are practically identical (therefore not shown), the maximal difference in the correlation coefficient is r=0.017. We suspect that this is a result of the almost Gaussian shape of the PSF's. Although the correct solution becomes less probable, it is still well within range.

Nature Methods: doi:10.1038/nmeth.2929

We investigated the change of the PSF of a SPIM system that should occur due to concavity of the light sheet across the field of view. Typical light-sheet microscopic acquisitions as shown in supplementary figure 17 and 13 show no visible sign of change, even across the entire field of view. Given the tolerance of the deconvolution regarding the shape of the PSF we concluded that it is not necessary to extract different PSFs at different physical locations. Note that the option to perform the deconvolution in blocks (**Online Methods**) would easily allow such an extension. We think that another real improvement in deconvolution quality could be achieved by being able to measure the correct PSF inside the sample, which could be combined with the work from Blume et al.[16] Additionally to the experimental burden, it is unfortunately far from being computationally tractable.

SUPPLEMENTARY NOTE 8: LINKS TO THE CURRENT SOURCE CODES

The source code for the efficient Bayesian-based multi-view deconvolution is implemented in ImgLib2[17] and available on Github: `https://github.com/fiji/spimreconstruction`. The source code most relevant to the deconvolution can be found in the package `src.main.java.mpicbg.spim.postprocessing.deconvolution2`. The code for the CUDA implementation is available online (`http://fly.mpi-cbg.de/preibisch/nm/CUDA_code_conv3d.zip`). Newer versions will be hosted using github, announcements will be done on the github page (`https://github.com/fiji/spimreconstruction`) and on the Fiji wiki (`http://fiji.sc/Multi-View_Deconvolution`).

The simulation of multi-view data (**Online Methods**) and the 3d-rendering as shown in main text figure 2a are implemented in ImgLib2.[17] The source code for the simulation is available on Github: `https://github.com/StephanPreibisch/multiview-simulation`. The source code for the 3d volume rendering can be found on Github as well `https://github.com/StephanPreibisch/volume-renderer`. Please note that the majority of the 3d-rendering code is a fork of the volume renderer written by Stephan Saalfeld `https://github.com/axtimwalde/volume-renderer`, the relevant class for rendering the sphere is `net.imglib2.render.volume.RenderNatureMethodsPaper.java`

REFERENCES

[1] Shepp, L. and Vardi, Y., "Maximum likelihood reconstruction for emission tomography," *Medical Imaging, IEEE Transactions on* **1**(2), 113–122 (1982).

[2] Krzic, U., *Multiple-view microscopy with light-sheet based fluorescence microscope*, phd thesis, University of Heidelberg (2009).

[3] Temerinac-Ott, M., Ronneberger, O., Ochs, P., Driever, W., Brox, T., and Burkhardt, H., "Multiview deblurring for 3-d images from light-sheet-based fluorescence microscopy," *Image Processing, IEEE Transactions on* **21**, 1863 –1873 (april 2012).

[4] Bonettini, S., Zanella, R., and Zanni, L., "A scaled gradient projection method for constrained image deblurring," *Inverse Problems* **25**(1), 015002 (2009).

[5] Hudson, H. and Larkin, R., "Accelerated image reconstruction using ordered subsets of projection data," *Medical Imaging, IEEE Transactions on* **13**(4), 601–609 (1994).

[6] Verveer, P., Swoger, J., Pampaloni, F., Greger, K., Marcello, M., and Stelzer, E. H. K., "High-resolution three-dimensional imaging of large specimens with light sheet-based microscopy," *Nature Methods* **4**(4), 311–313 (2007).

[7] Preibisch, S., Saalfeld, S., Schindelin, J., and Tomancak, P., "Software for bead-based registration of selective plane illumination microscopy data," *Nature Methods* **7**, 418–419 (Jun 2010).

[8] Keller, P. J., Schmidt, A. D., Santella, A., Khairy, K., Bao, Z., Wittbrodt, J., and Stelzer, E. H., "Fast, high-contrast imaging of animal development with scanned light sheet-based structured-illumination microscopy," *Nat. Methods* **7**, 637–642 (Aug 2010).

[9] Lucy, L. B., "An iterative technique for the rectification of observed distributions," *Astronomical Journal* **79**(6), 745–754 (1974).

Nature Methods: doi:10.1038/nmeth.2929

[10] Richardson, W. H., "Bayesian-based iterative method of image restoration," *Journal of the Optical Society of America* **62**, 55–59 (1972).

[11] Dempster, A. P., Laird, N. M., and Rubin, D. B., "Maximum likelihood from incomplete data via the em algorithm," *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B* **39**(1), 1–38 (1977).

[12] Preibisch, S., *Reconstruction of Multi-Tile Microscopic Acquisitions*, phd thesis, Technical University Dresden (2011).

[13] Kaufman, L., "Implementing and accelerating the em algorithm for positron emission tomography," *Medical Imaging, IEEE Transactions on* **6**(1), 37–51 (1987).

[14] Merziger, G., Wirth, T., Wille, D., and Mühlbach, G., [*Formeln + Hilfen zur höheren Mathematik*], Binomi Verlag, 4th ed. (2004).

[15] Singh, M. K., Tiwary, U. S., and Kim, Y.-H., "An adaptively accelerated lucy-richardson method for image deblurring," *EURASIP J. Adv. Signal Process* **2008** (2008).

[16] Blume, M., Zikic, D., Wein, W., and Navab, N., "A new and general method for blind shift-variant deconvolution of biomedical images," in [*MICCAI '07*], 743–750, Springer (2007).

[17] Pietzsch, T., Preibisch, S., Tomančák, P., and Saalfeld, S., "ImgLib2 – generic image processing in Java," *Bioinformatics* **28**, 3009–3011 (Nov 2012).

Nature Methods: doi:10.1038/nmeth.2929