CHAPTER

Open-source solutions for SPIMage processing 27

Christopher Schmied, Evangelia Stamataki, Pavel Tomancak
Max Planck Institute of Molecular Cell Biology and Genetics, Dresden, Germany

CHAPTER OUTLINE

Abstract

Light sheet microscopy is an emerging technique allowing comprehensive visualization of dynamic biological processes, at high spatial and temporal resolution without significant damage to the sample by the imaging process itself. It thus lends itself to time-lapse observation of

Methods in Cell Biology, Volume 123, ISSN 0091-679X, http://dx.doi.org/10.1016/B978-0-12-420138-5.00027-6

fluorescently labeled molecular markers over long periods of time in a living specimen. In combination with sample rotation light sheet microscopy and in particular its selective plane illumination microscopy (SPIM) flavor, enables imaging of relatively large specimens, such as embryos of animal model organisms, in their entirety. The benefits of SPIM multiview imaging come to the cost of image data postprocessing necessary to deliver the final output that can be analyzed. Here, we provide a set of practical recipes that walk biologists through the complex processes of SPIM data registration, fusion, deconvolution, and time-lapse registration using publicly available open-source tools. We explain, in plain language, the basic principles behind SPIM image-processing algorithms that should enable users to make informed decisions during parameter tuning of the various processing steps applied to their own datasets. Importantly, the protocols presented here are applicable equally to processing of multiview SPIM data from the commercial Zeiss Lightsheet Z.1 microscope and from the open-access SPIM platforms such as OpenSPIM.

INTRODUCTION

BRIEF OVERVIEW OF LIGHT SHEET MICROSCOPY FLAVORS

The basic principle of light sheet microscopy is that optical sectioning is achieved by illuminating the sample from the side with a thin sheet of laser light that will excite the fluorophores in the sample only in a relatively narrow plane. Photons emitted by these fluorophores are captured by a detection objective oriented perpendicularly to the light sheet and imaged onto a detection device such as a CCD camera (Fig. 27.1A). Thus, light sheet microscopy allows very fast acquisition, offering high temporal resolution. Since only the observed plane is illuminated, bleaching and phototoxicity are reduced to a minimum (Huisken, Swoger, Del Bene, Wittbrodt, & Stelzer, 2004).

The realization of this theta microscopy principle (Stelzer & Lindek, 1994) varies among the various types of light sheet microscopes. The illumination and detection axes may be fixed, with the detection objective permanently focused to the center of the light sheet, and the sample is moved through the focal volume to achieve 3D acquisition (Huisken et al., 2004). Alternatively, the light sheet is scanned axially through the sample, and the focus of the detection objective is continuously adjusted with a fine motor (Krzic, Gunther, Saunders, Streichan, & Hufnagel, 2012; Tomer, Khairy, Amat, & Keller, 2012) or using an electrically tunable lens (Fahrbach, Voigt, Schmid, Helmchen, & Huisken, 2013).

The light sheet itself can be static, formed by a cylindrical lens, the so-called selective plane illumination microscopy (SPIM; Huisken et al., 2004), or it can be generated dynamically by scanning a point source of laser light at high frequency across the sample, the so-called digital scanned laser light sheet fluorescence microscopy (DSLM; Keller & Stelzer, 2008). The static light sheet may be pivoted at high frequency parallel to the imaging plane, which leads to the reduction of the notorious stripe artifacts caused by absorption and scattering of the illumination light sheet across the field of view (Huisken & Stainier, 2007). The dynamic light sheet can

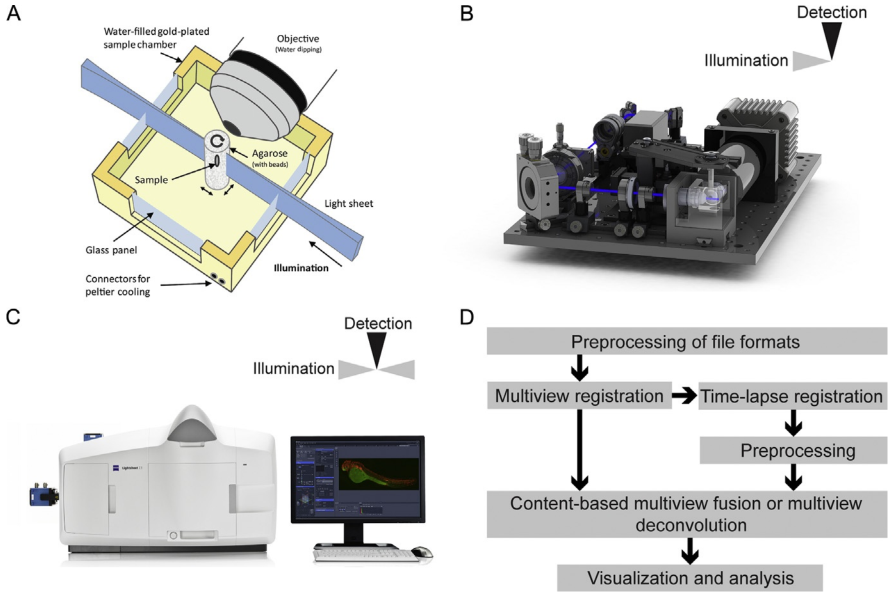

A
Water-filled gold-plated sample chamber
Objective (Water dipping)
Agarose (with beads)
Sample
Light sheet
Glass panel
Illumination
Connectors for peltier cooling
B
Detection
Illumination
C
Detection
Illumination
D
Preprocessing of file formats
Multiview registration
Time-lapse registration
Preprocessing
Content-based multiview fusion or multiview deconvolution
Visualization and analysis

FIGURE 27.1

Illustration of the SPIM principle. (A) The sample is suspended in an agarose column and immersed in a buffer. A thin laser light sheet with a concave geometry is illuminating the sample. A water-immersion lens is positioned perpendicular to the light sheet, and the emitted light is captured on a CCD or sCMOS camera. By moving the sample through the light sheet, stacks are acquired. Rotation allows the acquisition of multiple views. (B) OpenSPIM—an open-access platform for light sheet microscopy. (C) Zeiss Lightsheet Z.1—a commercial realization of light sheet microscope from Carl Zeiss Microscopy. (D) Workflow of the Fiji SPIM image-processing pipeline.

Figure (A) from Preibisch, Saalfeld, Rohlfing, and Tomancak (2009). Figure (B) adapted from Pitrone et al. (2013). Figure (C) © Copyright of Carl Zeiss Microscopy GmbH

be modulated by structured illumination, which leads to the reduction of the background and increase in signal-to-noise ratio (Keller et al., 2010). An axicon element can be included in the illumination beam path leading to a formation of a self-reconstructing Bessel beam that provides a more uniform illumination across the field of view and reduced light sheet thickness resulting in higher z resolution of the acquisition (Planchon et al., 2011).

In the DSLM mode, a two-photon light sheet can be formed, which provides better penetration into thick biological specimen (Truong, Supatto, Koos, Choi, & Fraser, 2011). Uniformity of illumination across the field of view can be increased by bringing the light sheet into the sample from two sides sequentially or even simultaneously, the so-called dual-sided illumination principle (Huisken & Stainier, 2007). Finally, sample rotation provides the ability to image the same specimen from multiple angles and to cover the "dark side" of the sample that happens to be away from the detection lens. The need for rotation can be partially ameliorated by building a second detection axis that positions the detection lens physically on the dark side of the specimen, the so-called dual-sided detection principle (Huisken et al., 2004; Krzic et al., 2012; Tomer et al., 2012).

The detection arm of a light sheet microscope can be equipped with a regular CCD camera, EM-CCDs providing better sensitivity, or more recently sCMOS cameras with large chips and fast readout times. The magnification of the detection lens has to match the numerical aperture (NA) of the illumination optics.

With higher NA of the illumination objective, the light sheet is thinner in the center of the field of view but thicker at its borders, thus reducing the usable field of view. A thicker light sheet is more uniform across a larger area and increases the field of view.

Light sheet microscopy remains an active field in optical technology development. Detailed discussion of all light sheet microscopy variations developed so far is beyond the scope of this chapter, and we recommend the many excellent reviews written by the movers of the light sheet field (Huisken & Stainier, 2009; Keller, 2013; Reynaud, Krzic, Greger, & Stelzer, 2008) and Chapter 11 of this book.

APPLICATIONS OF LIGHT SHEET MICROSCOPY

The broad range of light sheet arrangements results in a diverse application of the technology. Light sheets are instrumental in monitoring fast biological processes such as beating hearts (Arrenberg, Stainier, Baier, & Huisken, 2010), neuronal activity in intact fish brain (Ahrens, Orger, Robson, Li, & Keller, 2013), and movements of RNA molecules in embryos (Siebrasse, Kaminski, & Kubitscheck, 2012). The most advanced structured illumination Bessel beam microscopes developed by Erik Betzig have impressively demonstrated that the light sheet paradigm will be instrumental in systems level monitoring of cell biological processes in living cells including cortical dynamics, endocytosis, or cell division (Gao et al., 2012). But these cutting-edge microscopes are not yet broadly available, and the details of data processing required remain rather opaque.

From the beginning, light sheet microscopy has been applied in a spectacular manner to monitoring of developmental processes in intact embryos (Keller, Schmidt, Wittbrodt, & Stelzer, 2008), and this particular application is the focus of this chapter. Development forms a complex multicellular organism from a single cell through cell proliferation, cell differentiation, and morphogenesis. In order to study these relatively fast and transient processes and the underlying genetic programs that control them, we require the ability to image living embryos with high spatiotemporal resolution throughout embryogenesis (Megason & Fraser, 2007). Specifically, SPIM is ideal for this particular application. SPIM is capable of imaging large specimens with single-cell resolution. It provides fast acquisition with a minimum of phototoxicity and bleaching of the labeled molecular markers. Moreover, by combining stacks acquired from several angles, SPIM can achieve isotropic resolution and allows capturing a single living embryo in its entirety.

The SPIM technology is mature, and it is available both as a commercial product and as prototypes developed by individual labs or public open-access communities (Gualda et al., 2013; Pitrone et al., 2013). Thus, many researchers are or, in the near future, will be acquiring SPIM data and need to be aware of the image analysis challenges awaiting them and become equipped to solve them, which is the purpose of this chapter.

In SPIM, the suspended sample, typically mounted in agarose inside a glass capillary (for review on sample mounting, see Reynaud et al. (2008)), can be moved and rotated freely through the light sheet, which allows acquisition of 3D stacks of 2D planes from multiple angles (in SPIM jargon referred to as views). By imaging the sample from different angles, limitations in penetration depth and degradation of the signal can be at least partially circumvented. Registration of the views and their combination into a single output image (so-called fusion) enables the reconstruction of the sample with nearly isotropic resolution (Preibisch, Saalfeld, Schindelin, & Tomancak, 2010; Swoger, Verveer, Greger, Huisken, & Stelzer, 2007).

In this guide, we will discuss how to process raw time-lapse, multiview SPIM data in order to visualize and analyze them. We are going to use the output of two distinct systems as examples: One is an open-source platform for light sheet microscopy OpenSPIM (Pitrone et al., 2013; Fig. 27.1B) and the other one is a commercially available system from Carl Zeiss Microscopy marketed under the name Lightsheet Z.1 (Fig. 27.1C). OpenSPIM implements single-sided illumination and single-sided detection and is limited to single-channel imaging, while Lightsheet Z.1 offers dual-sided illumination, multichannel acquisition, and much more. From the point of view of image processing, it is important to note that both systems use sample rotation to achieve multiview imaging.

Our pipeline, presented in the succeeding text, is not limited by the use of a specific system and can easily be applied to other SPIM applications. It has been released in the open-source platform Fiji Is Just ImageJ (Fiji; Schindelin et al., 2012), a freely available distribution of ImageJ, which greatly facilitates its adaptation to various multiview imaging scenarios. Importantly, the open-source nature of all the software discussed here makes it possible to deploy the individual steps of the processing pipeline on a cluster computer. The sheer amount of image data produced by light sheet microscopes is tremendous, it rivals the data volumes produced in particle physics, and biologists are not ready to deal with this data deluge. In "Cluster Processing," we discuss parallelization of SPIM image processing on high-performance computing hardware.

27.1 PREREQUISITES

27.1.1 PARAMETERS OF EXAMPLE DATASET

We will discuss all processing steps using an SPIM recording of a *Drosophila melanogaster* embryo acquired with the Zeiss Lightsheet Z.1. To visualize cell behavior during embryogenesis, we have used a transgenic fly line expressing histone H2Av labeled with mRFPruby (Fischer, Haase, Wiesner, & Müller-Taubenberger, 2006).

The same procedure applies for the data from OpenSPIM with the exception of handling transformation of the file formats described in "Section 27.1.5."

The embryo was imaged with a Zeiss 20×/1.0 water-immersion Plan Apochromat objective lens with 0.8 zoom at 25 °C. 1.5 mW of the 561 nm laser was used for the first 67 time points; the power was later increased to 2.5 mW. Each view contains 130 slices with a step size of 1 μm. 6 views at 60° angles were recorded per time point. The embryo was imaged continuously for over 18 h, with each time point taking 90 s to acquire. 715 time points were recorded. Exposure time was 20 ms per slice. Each slice consists of 1920 × 1060 pixels with a pixel size of 0.2859 μm and a bit depth of 16 bits. We used dual-sided illumination with Zeiss 10×/0.2 illumination lenses. The light sheet thickness was 4.03 μm at the center of the field of view and 13.3 μm at the borders. Both sides were fused as a preprocessing step directly after acquisition using a mean fusion in the ZEN software. This reduces the size of the acquired recording by half to 5.63 TB.

27.1.2 FLUORESCENT BEADS AS FIDUCIAL MARKERS FOR REGISTRATION

For purposes of multiview registration, we included a 1:10,000 dilution of fluorescent microspheres (FY050 Estapor microspheres from Merck Millipore) with a diameter of 500 nm in the agarose mounting medium. These beads were excited with the 561 nm laser and have their predominant peak of emission around 680 nm. The beads should be selected such that they can be excited and imaged in the chosen channel but without using their dominant excitation and emission wavelengths. Thus, they are still visible and can be segmented but do not dominate the image.

We recommend using the FY050 beads for imaging GFP or YFP alone since their fluorescence is weaker than the signal when excited with the 488 nm lasers. These beads have a strong fluorescence when excited with the 561 nm laser used for imaging mRFPruby. Thus, we recommend the FZ050 beads when imaging mRFPruby alone or GFP and mRFPruby in two channels.

27.1.3 INSTALLATION AND CONFIGURATION OF FIJI

We will be using Fiji for all SPIM image data processing steps (SPIMage processing). Fiji is a Java open-source biological image analysis platform (Schindelin et al., 2012) that is freely available to download and install on any computer platform including PC, Mac OS X, and Linux (http://fiji.sc). Fiji is a modular software that uses the system of the so-called plug-ins, independent software modules that are linked to the main program and extend it in some useful way. It is important to note that Fiji uses an update mechanism that allows continuous renewal of the program on the user's computer with the latest versions of the plug-ins and the addition of new functionality. These plug-ins, collectively, together with the Fiji's core (which is nothing else but ImageJ; Schneider, Rasband, & Eliceiri, 2012), represent the Fiji distribution. The plug-ins required for SPIMage processing are part of the Fiji distribution and can be found upon download in the main Fiji *Plugins menu* in the subfolder *SPIM Registration*. We recommend updating Fiji before attempting any reconstruction.

27.1.4 HARDWARE REQUIREMENTS

Multiview SPIM image processing is computationally expensive due to the size of the image dataset. Typical acquisition will consist of multiple views of 100–400 MB each. Those large images have to be at least temporarily held in memory, usually in multiple copies, and rotated, which increases the effective size dramatically. Therefore, we recommend investing in *RAM*. Most of the Fiji code involved in SPIMage processing is in Java and is multithreaded (i.e., will make use of multiple cores when available); thus, investing in *multicore processors* is also advisable. The output of the SPIMage processing may increase the already substantial total raw data volume, and so, it is prudent to plan for ample *hard drive space*. Multiview deconvolution, 3D rendering, and in general visualization of SPIM data are placing high demands on graphics hardware (GPU). We recommend investing in *CUDA capable graphics cards* such as NVIDIA Tesla or Quadro or GeForce. Moving large amounts of data characteristic to SPIM data to and from the computer takes time, and every network bottleneck between the end points should be removed starting with a 10 gigabit network interface. The processing described in this tutorial is performed on the following hardware configuration:

Processor: *Two Intel Xeon Processor E5-2630* (*six-core*, *2.30 GHz Turbo*, 15 MB, 7.2 GT/s)
Memory: *128 GB* (16 × 8 GB) 1600 MHz DDR3 ECC RDIMM
Hard drive: *4 × 2TB* 3.5 inch Serial ATA (7.200 Rpm) hard drive
HDD controller: PERC H310 SATA/SAS controller for Dell Precision
HDD configuration: C1 SATA 3.5 inch, 1–4 hard drives
Graphics: *Dual 2 GB NVIDIA Quadro 4000* (2 cards w/2DP and 1DVI-I each; 2DP-DVI and 2DVI-VGA adapter; MRGA17H)
Network: Intel X520-T2 Dual Port *10GbE Network Interface Card*

27.1.5 FILE FORMATS PREPROCESSING AND NAMING CONVENTIONS

The first step and a prerequisite for the SPIMage processing pipeline in Fiji is to be able to open the image files. This is not a trivial task, since the paradigm of SPIM multiview imaging is relatively new and both commercial and open-source platforms did not yet incorporate the extra dimension into their data models. Moreover, SPIM acquisitions tend to be long-term, time-lapse, multiview, and multichannel acquisitions, which may be stored on the hard drive in countless permutations (for instance, one file for every dimension, one file per time point, or one file per view). Fiji relies on the Bio-Formats library (Goldberg et al., 2005) to import image data. SPIM registration Fiji plug-ins are currently working only with *.tif* or *.lsm* file types. In order to work with files from Lightsheet Z.1, it is necessary to open the files in Fiji and resave them as regular *.tif* files. This is easy to do manually for a single multiview acquisition but becomes impractical for any larger time-lapse recordings. Fortunately, Fiji possesses a powerful macro language that makes it relatively straightforward to automate this task. The exact description of how to do that is beyond the scope of this manuscript; detailed instructions on how to open *.czi* files and resave them as *.tif files* can be found here (http://openspim.org/Pre-processing). Note that the same resaving procedure applies for OpenSPIM and Lightsheet Z.1 data.

The issue of file formats is closely connected to the issue of naming convention for the files. The SPIM plug-ins currently rely on encoding the information about the basic metadata about the experiment (i.e., which time point, view, or channel the file contains) in the file name. Although other possibilities are feasible, we recommend using the following naming convention:

spim_TL<*number*>**_ch**<*number*>**_Angle**<*number*>.tif

The numbers are usually zero-padded (i.e., 001, 002, and 003). Note that channels and angles can be discontinuous, while the time points usually represent a series of integers. The renaming into this naming convention is encoded in the macro script used for resaving from *.czi* to *.tif* (http://openspim.org/Pre-processing).

27.2 OVERVIEW OF THE SPIM IMAGE-PROCESSING PIPELINE

In order to reconstruct the sample, we need to register and fuse the acquired multiple views into a single output volume. We use subresolution fluorescent beads embedded in the rigid agarose medium as fiduciary markers to achieve this registration (Preibisch et al., 2009, 2010). Additionally, we exploit the beads to measure the point spread function (PSF) of the system to constrain the multiview deconvolution algorithm (Preibisch, Amat, Stamataki, Sarov, Singer, Myers & Tomancak, 2014).

Practically, the processing pipeline consists of the following steps (Fig. 27.1D):

1. *Multiview registration* (bead-based registration): During the first steps, the beads in each view are segmented. Then, the segmented beads are matched between the different views, and the views are registered resulting in a transformation model for each view that is able to arrange the views optimally in 3D space with respect to each other.
2. *Time-lapse registration* (bead-based registration): When dealing with time-lapse data, each time point of the recording is registered onto a reference time point. This allows compensating for slight differences in the positioning of each time point due to agarose deformation or sliding out of the capillary. This step is necessary for smooth visualization of the recording and tracking of labeled components across time.
3. *Preprocessing before* content-based multiview fusion/ multiview deconvolution (multiview fusion): Since the content-based multiview fusion/ multiview deconvolution processes are very memory-intensive, it is beneficial to first define a minimal area of interest using downsampled data. The region beyond this minimal area will not contribute to the fusion, and thus, it reduces the memory footprint and the required processing time.

4. Content-based multiview fusion/ multiview deconvolution (multiview fusion): The 3D images of the individual views are then fused together into a single output image by either content-based multiview fusion or multiview deconvolution using the transformation parameters from registration (step 1) and using the crop area defined in step 3.

It should be noted that while multiview registration must always precede fusion or deconvolution, the time-lapse registration could be performed out of order, immediately after multiview registration or after the fusion. The pipeline as presented in Fig. 27.1D is the most efficient order when dealing with massive time-lapse data. Some steps of the pipeline in Fig. 27.1D can be parallelized (steps 1 and 4), that is, each time point can be processed by a separate computer, and we describe how to leverage such distributed hardware in "cluster processing."

27.3 BEAD-BASED REGISTRATION

The first step in SPIMage processing is to register the 3D stacks of the same specimen acquired from different angles. This is achieved by fixing one of the views in 3D space and registering all other views to it. The axial resolution in SPIM is limited by the width of the light sheet and is significantly lower compared to the lateral resolution, which is determined by the NA of the objective lens. Thus, the stacks of individual views are highly anisotropic. The signal degrades along the illumination axis, because the light sheet gets absorbed and scattered by the specimen (Fig. 27.2A). To partially compensate for signal degradation along the illumination axis, it is possible to illuminate the sample from both sides sequentially; however, it does not completely remove the degradation depending on optical properties of the sample. Moreover, in SPIM, the signal degrades along the detection axis, similarly to a confocal, as the emitted light has to pass through the sample to reach the objective lens and is thus scattered and absorbed. Parts of the sample, which are further away from the objective lens, will be increasingly blurry (Fig. 27.2A). Finally, the specimen is typically alive and labeled components change position. In combination with anisotropy and signal degradation, this makes registration using sample intensities problematic (Preibisch, Ejsmont, Rohlfing, & Tomancak, 2008; Preibisch, Rohlfing, Hasak, & Tomancak, 2008; Swoger et al., 2007).

The bead-based registration plug-in solves the problem by uncoupling the registration from sample intensities by using subresolution fluorescent beads included in the rigid agarose medium as fiduciary markers (Preibisch et al., 2009, 2010). It consists of three principal steps: finding the beads, establishing which beads are the same between the different pairs of views, and minimizing the displacement of the corresponding beads.

For bead segmentation, we use a standard image-processing algorithm, *difference of Gaussian*, to detect local maxima in the image. For matching beads across views, we construct *local geometric descriptors*, which are essentially constellations of four beads expressed in relative, local coordinate systems that allow finding similar constellations in other views independent of rotation and translation (Fig. 27.2B).

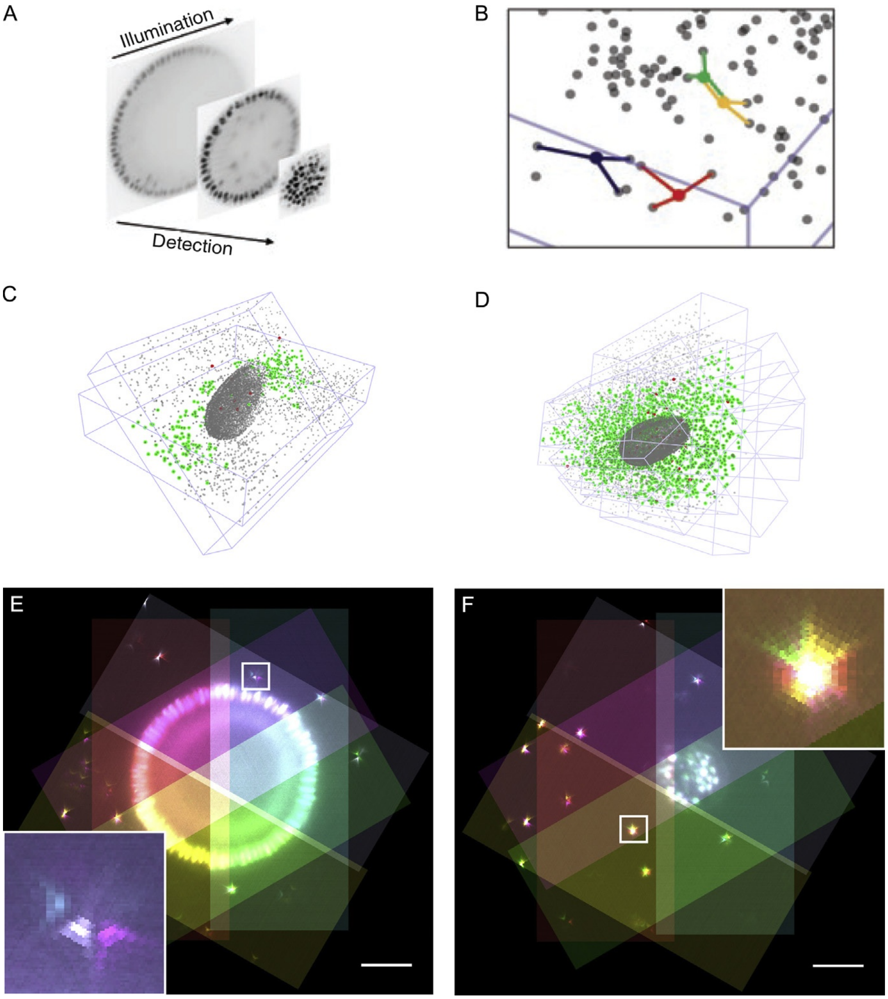


FIGURE 27.2

Multiview registration. (A) Degradation of the signal along the illumination and the detection axes. (B) Segmented beads in a 3D volume. Each bead (gray) is described by a local descriptor using the location of the three nearest neighbors in respect to the described bead. Four beads with their 3 nearest neighbors are highlighted by connecting lines. (C) Schematic of two views that are registered onto each other. The boundary box of each view and the sample are depicted in light gray. Bead correspondences are shown as dark grey dots. (D) An affine transformation model is used to register all views onto a fixed reference view. (E and F) Overlap of the different views in the reconstruction. The insets in each panel show the overlapping PSFs of the beads. In certain parts of the reconstruction, only a subset of the views might contribute to the reconstruction (E), whereas for the tips of the embryos, nearly all six views contributed (F). Scale bars represent 50 μm.

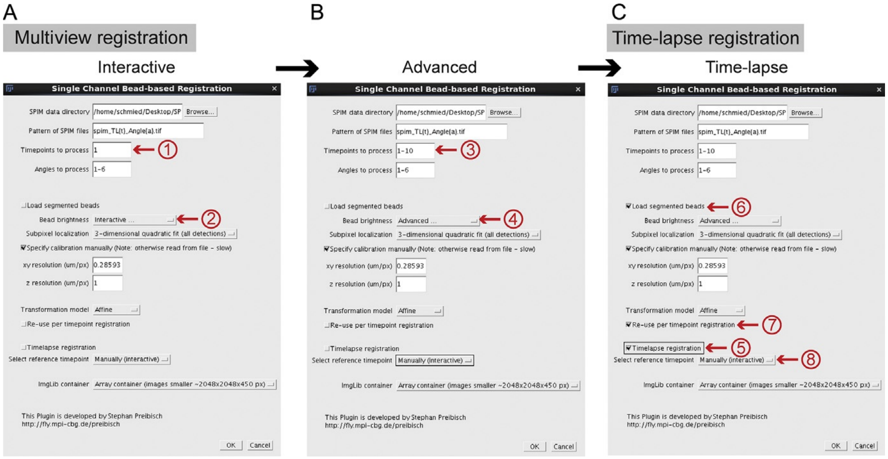
Figures (A and B) from Preibisch et al. (2010) and (C and D) from Preibisch et al. (2009).

Since this process is likely to produce spurious matches (bead constellations that are similar by chance), we use the *random sample consensus* (*RANSAC*; Fischler & Bolles, 1981) algorithm to exclude all false correspondences. The basic assumption of RANSAC is that all true correspondences are agreeing on one transformation model (Fig. 27.2C green dots), whereas each false correspondence will point to a different transformation model (Fig. 27.2C red dots). Finally, the true corresponding bead descriptors between the different views are used to identify the affine transformation of each view that minimizes their displacement. A *global optimization* using an iterative optimization algorithm is used to determine the optimal transformation model to match all views onto a fixed view as a common reference (Fig. 27.2D).

27.3.1 WORKFLOW

In practice, the registration process is performed using the *bead-based registration* plug-in. For detailed cookbook-style tutorial, see http://openspim.org/Registration and http://openspim.org/Timelapse_Registration.

1. We first select the segmentation method. *Difference of mean* is in most cases sufficient and has the benefit of running faster compared with *difference of Gaussian*. It is definitely the method of choice when attempting to define the segmentation parameters interactively.
2. Next, we load the raw data, named according to the naming convention described in Section 27.1.5 (Fig. 27.3A–C top part of each screenshot). Initially, we will process a single time point. For a long time series, it is reasonable to select a time point in the middle of the time series.
3. During the first run through the registration pipeline, we will determine the best bead segmentation parameters. We are doing this interactively on one representative angle of one time point of our dataset (Fig. 27.3A). The plug-in will load the selected stack in a window. The effect of the selected segmentation settings can be observed in the stack window with the segmented beads circled in green. For typical subresolution fluorescent beads, the default radius settings ($r_1 = 2$ and $r_2 = 3$) are sufficient, and only the threshold needs to be adjusted. The lower the threshold, the more beads will be detected. The aim is to segment as many beads as possible, without getting too much false detection in or around the sample and multiple detections on one bead. Note that detections inside the sample are unavoidable and will not compromise the registration.
4. After selecting the parameters, we execute the registration on the selected time point. The plug-in will first open each view and run the segmentation algorithm selected in step 1. This should result in several thousand detections for each view. Since the overlap of the views is typically limited and many detections represent densely packed structures inside the specimen, only a fraction of these detections will form descriptors that can be matched between views. Some of these will be similar by chance and will be excluded by RANSAC as false correspondences. The *ratio of true versus false correspondences* is reported by the algorithm and should be ideally over 90%, and the total *number of true correspondences* should be as high as possible.



A — Single Channel Bead-based Registration

SPIM data directory: /home/schmied/Desktop/SP
Pattern of SPIM files: spim_TL{t}_Angle{a}.tif
Timepoints to process: 1
Angles to process: 1-6
□ Load segmented beads
Bead brightness: Interactive ...
Subpixel localization: 3-dimensional quadratic fit (all detections)
☑ Specify calibration manually (Note: otherwise read from file - slow)
xy resolution (um/px): 0.28593
z resolution (um/px): 1
Transformation model: Affine
□ Re-use per timepoint registration
□ Timelapse registration
Select reference timepoint: Manually (interactive)
ImgLib container: Array container (images smaller ~2048x2048x450 px)
This Plugin is developed by Stephan Preibisch
http://fly.mpi-cbg.de/preibisch

B — Single Channel Bead-based Registration

SPIM data directory: /home/schmied/Desktop/SP
Pattern of SPIM files: spim_TL{t}_Angle{a}.tif
Timepoints to process: 1-10
Angles to process: 1-6
□ Load segmented beads
Bead brightness: Advanced ...
Subpixel localization: 3-dimensional quadratic fit (all detections)
☑ Specify calibration manually (Note: otherwise read from file - slow)
xy resolution (um/px): 0.28593
z resolution (um/px): 1
Transformation model: Affine
□ Re-use per timepoint registration
□ Timelapse registration
Select reference timepoint: Manually (interactive)
ImgLib container: Array container (images smaller ~2048x2048x450 px)
This Plugin is developed by Stephan Preibisch
http://fly.mpi-cbg.de/preibisch

C — Single Channel Bead-based Registration

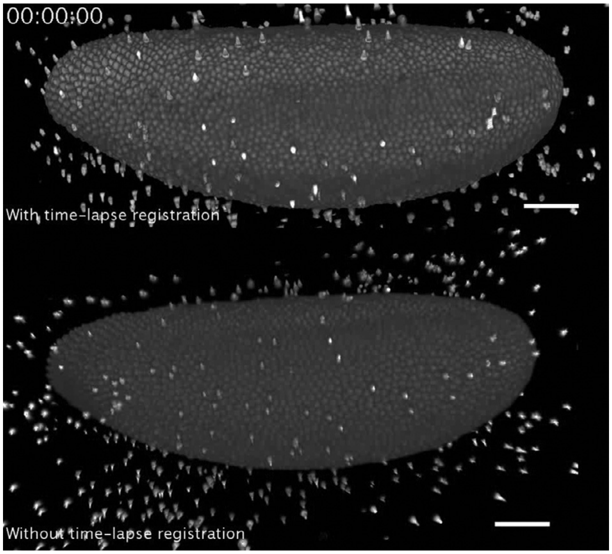
SPIM data directory: /home/schmied/Desktop/SP
Pattern of SPIM files: spim_TL{t}_Angle{a}.tif
Timepoints to process: 1-10
Angles to process: 1-6
☑ Load segmented beads
Bead brightness: Advanced ...
Subpixel localization: 3-dimensional quadratic fit (all detections)
☑ Specify calibration manually (Note: otherwise read from file - slow)
xy resolution (um/px): 0.28593
z resolution (um/px): 1
Transformation model: Affine
☑ Re-use per timepoint registration
☑ Timelapse registration
Select reference timepoint: Manually (interactive)
ImgLib container: Array container (images smaller ~2048x2048x450 px)
This Plugin is developed by Stephan Preibisch
http://fly.mpi-cbg.de/preibisch

FIGURE 27.3

Overview of the registration process. Shown are screenshots of the bead-based registration plug-in window in Fiji. (A) During the initial run, we select a single time point (*1*) and launch the interactive bead segmentation window (*2*). (B) In the second run, we apply the segmentation parameters to all time points (*3*) in the time series by selecting the "advanced" segmentation option (*4*). (C) In the final run, we perform time-lapse registration (*5*) while reusing the bead segmentations (*6*) and registration (*7*) from the previous run. We are choosing the reference time point interactively (*8*).

In the final phase of the registration run, the global optimization will find the best affine transformation model that minimizes the displacement of true correspondences. The result of the global optimization is reported as *average displacement* of the corresponding bead descriptors, and it should be as low as possible, typically less than one pixel. If the number of true correspondences is low or average displacement high, one should try a different segmentation threshold. If the number of true correspondences cannot be increased over the built-in threshold of 12 per pair of views, it may be necessary to repeat the imaging experiment with more beads. It is important to note that it is not necessary that all views are connected to all other views; it is in principle sufficient that the views are connected pair-wise forming a closed chain.

5. Once a good segmentation threshold is identified, we can apply the selected parameter to the whole recording and perform the multiview registration on each time point using the advanced bead segmentation option (Fig. 27.3B). The actual output of the registration process will be coordinates of the geometric descriptors and *affine transformation matrices*. The files containing these numbers will be stored in a new directory called */registration*, which will be created in the directory that contains the raw data.
6. The exact relative positioning of each time point may vary over time due to imprecise motor movements or slight drift of the agarose column inside the capillary. These differences in positioning would appear as drift if played as a time-lapse



MOVIE 1

Time-lapse registration. Renderings of the Lightsheet Z.1 recording after content-based multiview fusion with time-lapse registration (upper panel) and without time-lapse registration (lower panel). Registration onto the reference time point stabilizes the recording and removes the drift introduced by imprecise motor movements and movement of the agarose.

movie (Movie 1 on http://dx.doi.org/10.1016/B978-0-12-420138-5.00027-6). To counteract this, we perform a *time-lapse registration*. In this step, we register all time points onto a reference time point. The time-lapse registration uses the same *bead-based registration* plug-in and works analogous to the multiview registration exploiting the beads segmented in the previous run (Fig. 27.3C). The reference time point can be selected manually, and it is advisable to select the same time point on which we optimized the segmentation parameters in the middle of the time series. At the end of the time-lapse registration, the plug-in will display a graph of the average, minimum, and maximum correspondence displacements. At this point, we can optionally select the time point with the lowest average displacement and rerun the registration using that time point as a reference. The output is, similarly to the registration for each time point, an affine transformation matrix stored in the */registration* subdirectory. The new registration files have a suffix (e.g., *.to_<reference_timepoint>*) to indicate that they contain the time-lapse registration data.

27.3.2 RESULTS

The results of bead-based registration are affine transformation matrices that minimize the displacement of potentially thousands of corresponding bead descriptors surrounding the imaged specimen. Figure 27.2E and F shows sections through the data orthogonal to the rotation axis after the transformation matrices were applied to the raw 3D image stacks. The data are not combined but simply displayed on top of each other with each view having assigned a different color. The insets show that the color-coded axially elongated PSFs of the beads overlap, which is the principle of the bead-based registration method. Note that the inset in Fig. 27.2E is composed of PSFs from three views, while the inset in Fig. 27.2F is made of four overlapping PSFs. Similarly, the boxes that represent the data indicate that there are areas of the specimen covered by two, three, four, and very rarely five views. Only at the tip of the specimen do the three- and four-way overlaps contribute to data (Fig. 27.2F); most of the specimen in the center slice is covered by two or maximally three views (Fig. 27.2E). Such limited view overlap is typical for multiview SPIM recording since imaging far beyond the equator of the sample does not usually contribute useful information. Therefore, we typically stop the imaging approximately halfway through the sample, which also decreases acquisition time. Nevertheless, as long as there are beads in the overlapping areas of the views, the registration will succeed.

27.4 MULTIVIEW FUSION

After successful registration, it is necessary to combine the data into a single three-dimensional output volume. Several strategies to achieve that task have been published (Rubio-Guivernau et al., 2012; Swoger et al., 2007; Temerinac-Ott, Ronneberger, Nitschke, Driever, & Burkhardt, 2011; Temerinac-Ott et al., 2012), ranging from simple averaging of the values to sophisticated multiview deconvolution strategies. We focus here on the two fusion methods implemented in Fiji, namely, *content-based multiview fusion* (Preibisch, Rohlfing, et al., 2008) and *efficient Bayesian-based multiview deconvolution* (Preibisch et al., 2014).

27.4.1 CONTENT-BASED MULTIVIEW FUSION

The transformation parameters determined in the multiview registration are the prerequisite to fuse the views into a single isotropic image (Fig. 27.4A). The views in multiview acquisition are typically overlapping only partially (Fig. 27.4B), and the deeper we image inside the specimen, the more blurry the images become due to light scattering along the illumination and detection axes (Fig. 27.3A). Therefore, simple addition or averaging of the values from the different views in the registered volume would lead to degradation of the sharp data from one view with the blurred data from the other view. The content-based multiview fusion attempts to prevent this degradation by evaluating local information content or entropy in the overlapping parts of the multiview acquisition and suppressing the blurry data with low entropy or in other words by maximizing the contribution of the sharp parts to the output image (Fig. 27.4C–E).

27.4.1.1 Workflow

Practically, the *content-based multiview fusion* is implemented by a dedicated plugin accessible from the SPIM Registration Method submenu in Fiji. For detailed cookbook-style tutorial, see http://openspim.org/Fusion.

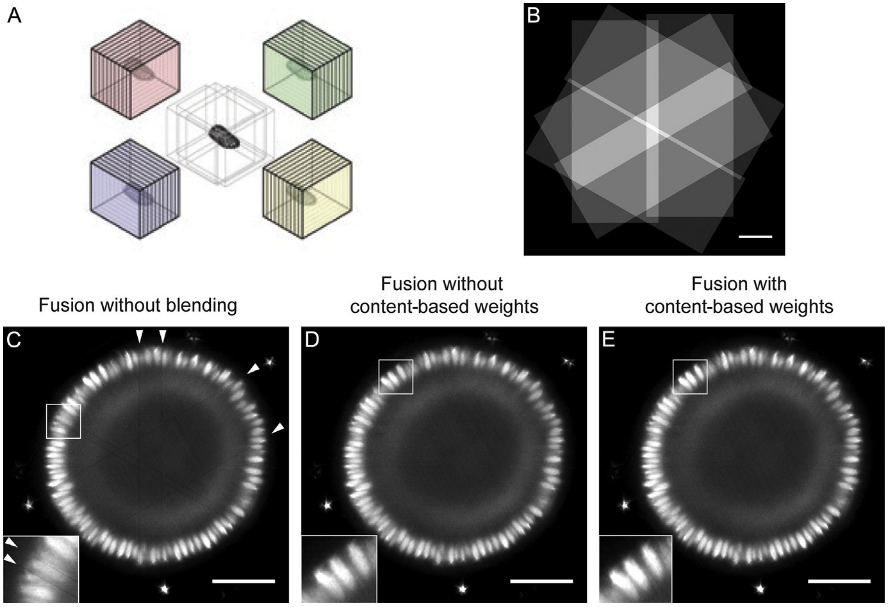
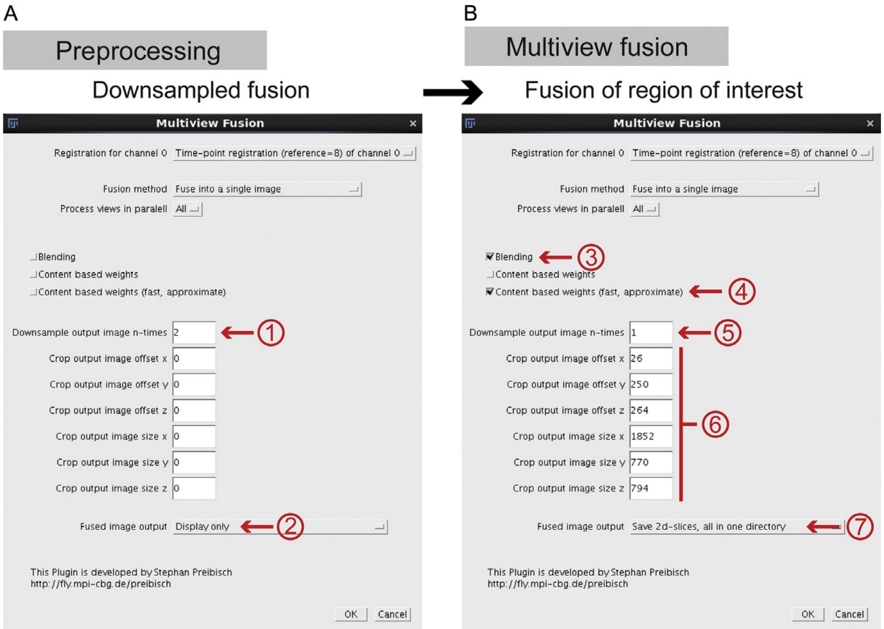

A
B
Fusion without blending
Fusion without content-based weights
Fusion with content-based weights
C
D
E

FIGURE 27.4

Content-based multiview fusion. (A) 3D stacks acquired from different angles are fused into a single 3D volume. (B) Schematic representation of the overlap between the six different views in our example dataset. (C) y–z section of Lightsheet Z.1 data after content-based multiview fusion without blending. The line artifacts are visible (*white arrowheads*, *inset*). (D) y–z section after content-based multiview fusion with blending but without content-based weights (*insets*). (E) y–z section after content-based multiview fusion using the content-based weights (*inset*). Scale bars represent 50 μm.

Figure (A) from Preibisch et al. (2010)

1. Similarly to multiview registration, the content-based multiview fusion is realized by two consecutive runs of the content-based multiview fusion plug-in. The algorithm is very memory-intensive because the input 3D stacks are rotated using the transformation matrices from registration and thus the size of the bounding box of the output volume increases dramatically. The data outside the specimen are important for the registration since they contain the beads but should be excluded from the fusion to reduce processing time. Therefore, in the first run of the plug-in, the processing is done on a single time point *downsampled 2–4 times*, and the fusion is applied without any computationally intensive steps such as blending (see in the succeeding text) and content-based weights (Fig. 27.5A). Running the plug-in using these parameters is equivalent to performing fusion by averaging the values from each view, and due to downsampling, it runs interactively even for large input data.



A

Preprocessing

Downsampled fusion

Registration for channel 0: Time-point registration (reference=8) of channel 0

Fusion method: Fuse into a single image

Process views in paralell: All

□ Blending

□ Content based weights

□ Content based weights (fast, approximate)

Downsample output image n-times: 2 ← (1)

Crop output image offset x: 0

Crop output image offset y: 0

Crop output image offset z: 0

Crop output image size x: 0

Crop output image size y: 0

Crop output image size z: 0

Fused image output: Display only ← (2)

This Plugin is developed by Stephan Preibisch
http://fly.mpi-cbg.de/preibisch

B

Multiview fusion

Fusion of region of interest

Registration for channel 0: Time-point registration (reference=8) of channel 0

Fusion method: Fuse into a single image

Process views in paralell: All

☑ Blending ← (3)

□ Content based weights

☑ Content based weights (fast, approximate) ← (4)

Downsample output image n-times: 1 ← (5)

Crop output image offset x: 26

Crop output image offset y: 250

Crop output image offset z: 264

Crop output image size x: 1852

Crop output image size y: 770

Crop output image size z: 794

(6)

Fused image output: Save 2d-slices, all in one directory ← (7)

This Plugin is developed by Stephan Preibisch
http://fly.mpi-cbg.de/preibisch

FIGURE 27.5

Overview of the content-based multiview fusion. (A) Screenshot of the first run of the fusion plug-in on downsampled data (*1*). This run is meant to define the minimal crop area containing the sample. The output will not be saved (*2*). (B) The second run through the same plug-in with blending (*3*) and content-based multiview fusion (*4*) turned on, downsampling turned off (*5*), and crop parameters entered (*6*). The results will be saved into an output directory (*7*).

2. From the prefused data, we determine the coordinates of the bounding box, which only includes the sample. It is very important to use the reference time point of the time-lapse registration to determine the correct cropping parameters for a time-lapse recording. The bounding box would be different for the registration derived from processing of a single time point and from time-lapse registration. We use the core Fiji tools to collect the coordinates of the bounding cube and either write them down or record them using the macro recorder of ImageJ.
3. In the second run, we enter the recorded cropping parameters and perform the fusion without downsampling only on the part of the image defined by the crop area (Fig. 27.5B). It is important to multiply the coordinates of the crop area by the factor used in downsampling (i.e., if we downsampled 4 times, we multiply each coordinate by a factor of four). During this run, we apply *nonlinear blending* and *content-based weights*. The fusion can be applied sequentially to

multiple time points using the time-lapse registration files in the */registration* directory. The resulting fused output volume is saved in a new directory/ *output*. Each time point can be saved as a series of 2D image planes in its own directory that is named after the time point index (Fig. 27.5B). The result can be viewed either by dragging and dropping each directory into the main window of Fiji, or alternatively, it can be opened as image sequence (File → Import → Image Sequence).

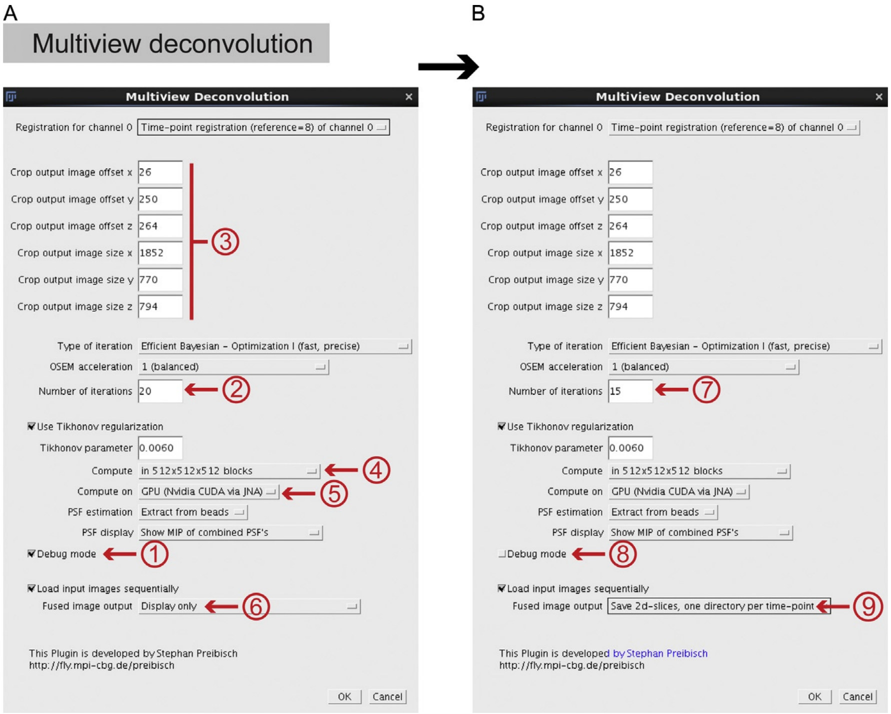
27.4.1.2 Results

As discussed in "Section 27.3," in SPIM, we do not typically image the entire sample in each view. Thus, the fusion algorithm has to deal with a situation when one view ends abruptly or the other begins. Without compensation, the abrupt borders of the views result in clearly visible line artifacts in the fused image (Fig. 27.4C white arrow heads). These line artifacts are eliminated by the use of nonlinear blending (Fig. 27.4C compared with Fig. 27.4D) that smooths the transition between the views. It should be, however, noted that the blending is altering the raw image data and should not be used when quantitative information need to be extracted from the images.

Content-based multiview fusion ensures that for each view, mainly the sharpest parts are contributing to the final fused image using location-dependent weighting factors for each pixel in each view (Fig. 27.4D compared with Fig. 27.4E insets). Since the PSF of the microscope is anisotropic and axially elongated, this simple fusion approach based on the information theory necessarily degrades the quality of the output image compared to the raw single-view data viewed laterally. Multiview deconvolution addresses this shortcoming of the fusion process.

27.4.2 MULTIVIEW DECONVOLUTION

Multiview deconvolution is another fusion strategy that takes the PSFs of the SPIM system into account to combine the images of different views into a single output image. In general, deconvolution attempts to reconstruct the underlying true image that gave rise to the observed microscope image given the PSF of the imaging system. This is obviously a very demanding task; however, the multiview scenario is particularly well suited for deconvolution since the same location in the same specimen is observed multiple times from different angles, which makes the ill-posed problem of inverting the convolution process more tractable. Several approaches to multiview deconvolution have been developed; however, none are capable of processing on large input datasets (Temerinac-Ott et al., 2011, 2012). We focus therefore on the *Bayesian-based multiview deconvolution plug-in* implemented in Fiji (Fig. 27.6; Preibisch et al., 2014). This plug-in implements an optimized multiview deconvolution approach that exploits the conditional probabilities between views (i.e., by observing one view, we learn something about the other view of the same specimen), which leads to dramatic decrease in number of iterations required to reach a certain deconvolution quality compared to competing approaches. Thanks



A — Multiview Deconvolution

Registration for channel 0: Time-point registration (reference=8) of channel 0

Crop output image offset x: 26
Crop output image offset y: 250
Crop output image offset z: 264
Crop output image size x: 1852
Crop output image size y: 770
Crop output image size z: 794

Type of iteration: Efficient Bayesian - Optimization I (fast, precise)
OSEM acceleration: 1 (balanced)
Number of iterations: 20

☑ Use Tikhonov regularization
Tikhonov parameter: 0.0060
Compute: in 512x512x512 blocks
Compute on: GPU (Nvidia CUDA via JNA)
PSF estimation: Extract from beads
PSF display: Show MIP of combined PSF's
☑ Debug mode

☑ Load input images sequentially
Fused image output: Display only

This Plugin is developed by Stephan Preibisch
http://fly.mpi-cbg.de/preibisch

B — Multiview Deconvolution

Registration for channel 0: Time-point registration (reference=8) of channel 0

Crop output image offset x: 26
Crop output image offset y: 250
Crop output image offset z: 264
Crop output image size x: 1852
Crop output image size y: 770
Crop output image size z: 794

Type of iteration: Efficient Bayesian - Optimization I (fast, precise)
OSEM acceleration: 1 (balanced)
Number of iterations: 15

☑ Use Tikhonov regularization
Tikhonov parameter: 0.0060
Compute: in 512x512x512 blocks
Compute on: GPU (Nvidia CUDA via JNA)
PSF estimation: Extract from beads
PSF display: Show MIP of combined PSF's
□ Debug mode

☑ Load input images sequentially
Fused image output: Save 2d-slices, one directory per time-point

This Plugin is developed by Stephan Preibisch
http://fly.mpi-cbg.de/preibisch

FIGURE 27.6

Overview of the multiview deconvolution processing. (A) The initial run is performed in the debug mode (*1*), in order to determine how many iterations are necessary (*2*). The deconvolution is performed on the cropped dataset (*3*) in blocks (*4*) using the GPU (*5*). For the initial run, we will only display the resulting stacks (*6*). (B) The final run is performed on the whole time-lapse using the determined number of iterations (*7*). The debug mode is turned off (*8*) and the resulting stacks are saved (*9*).

to advanced software engineering, the multiview deconvolution plug-in in Fiji is also able to execute each iteration faster, which together with less iterations leads to dramatic improvements in computation time. It is also the only approach able to deal with partially overlapping data and very large datasets. A final speed gain can be achieved by exploiting the parallel computation on graphics devices capable of CUDA (GPU computing), making this plug-in currently the only approach able to deconvolve long-term, time-lapse, multiview SPIM data in reasonable time.

27.4.2.1 Workflow

The *Bayesian-based multiview deconvolution* is implemented by a dedicated plug-in accessible from the SPIM Registration Method submenu in Fiji. For detailed information about advanced parameters of the plug-in, see http://fiji.sc/Multi-View_Deconvolution.

1. The prerequisite for multiview deconvolution is *precise registration* of the views in multiview acquisition as described in "Section 27.3." We also need to know the *PSF of the imaging system*. Since for the purposes of registration we included subresolution fluorescent beads that effectively measure the PSF, we use the true correspondences from the registration step to construct an average PSF for each view. Alternatively, a computed PSF can be provided as an image. Finally, even with all the speedups described in the preceding text, multiview deconvolution is computationally very demanding, and thus, we need to minimize the amount of input data. We use the *crop area* defined during the fusion step described in "Section 27.4" to limit the size of the deconvolved data. This means that in a typical workflow, we first perform a fusion to extract the crop parameters before proceeding with the multiview deconvolution. It is imperative to use the same registration files for both fusion steps.
2. For some large data, it may be beneficial to perform the *deconvolution on a downsampled dataset* to further speed up the process. Fiji offers the possibility to prescale the data before launching the deconvolution by running the *Apply External Transformation* plug-in. This plug-in will prepend a transformation model to the data (e.g., a scale factor of 0.5) by directly modifying the registration output files in /*registration* directory. Note that this process is irreversible, and so, it is recommended to backup the registration files before downscaling. Importantly, the downscaling is applied to the data after cropping, and so, the crop area must be expressed in the coordinates of the full-scale image.
3. As before, the deconvolution plug-in is run in two rounds. During the first round, we activate the *debug mode* that will give us an opportunity to examine the results of the deconvolution after each iteration and thus help us decide how far to run the iterative deconvolution process. The only other free parameter of the method is the number of iterations, where 10–20 is a good starting point. All other parameters of the deconvolution plug-in are launched with reasonable default values and should be changed only after thoroughly understanding the principles of the method. The implementation can make use of GPUs, and it is usually necessary to perform the operation in blocks because the memory of GPUs is small compared to the size of the SPIM input data.
4. After the initial debug run, which helps us to determine the number of iterations by examining the output, we can apply the deconvolution process sequentially to an entire time series. Note that for typical SPIM datasets, even when using the power of GPUs, this may take quite some time, and therefore, we recommend to explore the possibilities of parallelizing the process on a cluster as described in the preceding text.

27.4.2.2 Results

Multiview deconvolution dramatically improves the resolution and contrast of the fused image compared to content-based multiview fusion, and the quality of the reconstructed data is superior even to single views in the axial direction (Fig. 27.7A–F). The success of the deconvolution can be monitored by the effect

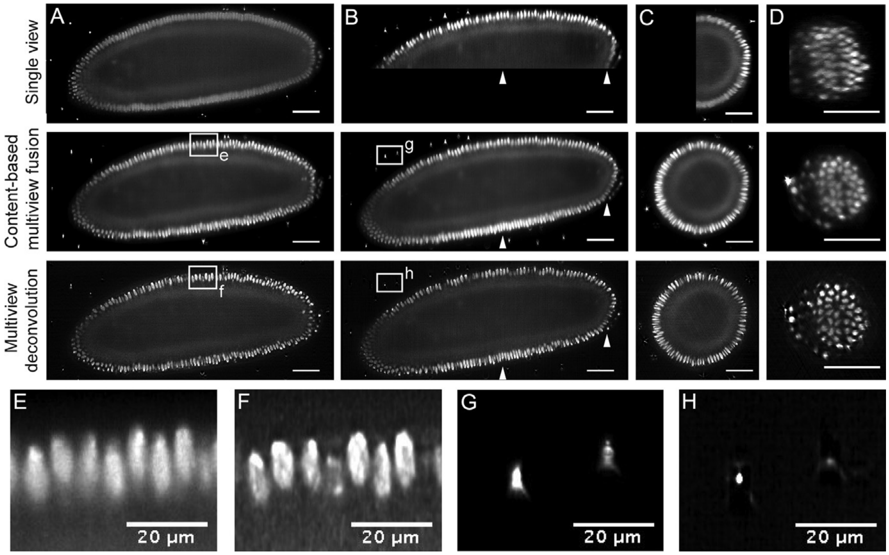


FIGURE 27.7

Multiview *deconvolution*. (A) Lateral, (B) axial, and (C and D) y–z sections through raw, fused, and deconvolved Lightsheet Z.1 data. The positions of the y–z sections are indicated with white arrowheads (B). Lightsheet Z.1 offers dual-sided illumination, and therefore, both sides of the sample are evenly illuminated (A, *single view*). The degradation of the signal in the detection axis is visible in the axial section (B, *single view*) and the y–z section of the single view (C and D, *single view*). The boxes mark areas enlarged in (E–H). (E) Blowup of nuclei after content-based multiview fusion. (F) Blowup of the same nuclei after multiview deconvolution. The resolution and contrast are greatly enhanced by the deconvolution. (G) Blowup of the PSFs of two beads after content-based multiview fusion. (H) Blowup of the same beads after multiview deconvolution. The PSFs of the beads collapse into a single point. Scale bars represent 50 μm except otherwise indicated.

it has on the fluorescent beads that are ideally collapsed to single intense points (Fig. 27.7G and H). The deconvolution is a nonlinear process that dramatically affects the signal intensity distribution, and so, it should be used with caution when applied to quantitative data analysis.

27.5 PROCESSING ON A HIGH-PERFORMANCE CLUSTER

The registration, fusion, and deconvolution pipelines are typically applied to multiview time-lapse SPIM datasets consisting of potentially hundreds of time points. While the individual steps last from minutes (registration) to hours (deconvolution on CPU) on one time point, when applied to large-scale time-lapse data, the minutes become hours and hours become days. It is therefore quite beneficial to parallelize the processing by employing many computers at the same time. Since, with the exception of time-lapse registration, all steps of the SPIMage processing pipeline are independent from time point to time point, they can be relatively straightforwardly distributed to a cluster computer. Although Fiji is relying heavily on a graphical user interface, we exploit the trick that the screen can be simulated in memory, which allows processing on a cluster node that lacks a monitor. We have developed a set of Bash and BeanShell scripts that deploy the SPIM plug-ins to a cluster computer. The source code details are beyond the scope of this chapter and are available at http://fiji.sc/SPIM_Registration_on_cluster. The basic premise is to first perform the registration locally on a graphics workstation and collect the key parameters such as z-scaling of the data, segmentation threshold, crop area, or number of iterations for multiview deconvolution. These parameters are inserted in appropriate places in the scripts and are eventually passed to a Fiji instance executed on a cluster node using raw multiview data for a single specific time point. The scripts can be easily adapted to a different cluster hardware and software. As is the case for processing single time points, the cluster needs to have fairly state-of-the-art hardware parameters, especially when it comes to read and write operations and the amount of memory available for each node. With the appropriate hardware, it becomes possible to register, fuse, and deconvolve an entire time-lapse multiview SPIM recording as fast as the data are acquired (Fig. 27.8 and Movie 2 on http://dx.doi.org/10.1016/B978-0-12-420138-5.00027-6).

27.6 FUTURE APPLICATIONS

The SPIMage processing pipeline is a product of research at the interface between biology and computer science. Although it is already quite useful for processing multiview, time-lapse SPIM data, it is far from finished. It will be important in the future to make the input to the plug-ins more flexible, utilizing the metadata from the microscopes to automatically understand the structure of the data and to process on them directly, without format conversions. The plug-ins themselves will become more integrated so that the tedious steps of defining the crop area and moving parameters manually between the different runs are more convenient. We are working on a Big Data Viewer plug-in that uses HDF5 database to enable interactive navigation of terabyte scale, multiangle data in Fiji. Eventually, all SPIMage processing plug-ins will write the output data into the HDF5 container, and that will enable to examine various steps of the processing pipelines immediately after they are finished. This will finally realize the vision of acquiring multiview SPIM data on a microscope and immediately being able to examine the reconstruction in three dimensions. Finally, processing of the SPIM data is only the beginning. The Fiji platform is running several open-source projects to analyze large-scale microscopy datasets in developmental and cell biological contexts that involve segmentation and tracking of labeled components through space and time. These tools, manual and automated, need to be integrated on one hand with the outputs of the SPIMage processing pipeline and on the other hand with visualization and data analysis tools in Fiji and beyond. Most of the SPIMage processing is using the powerful Java image-processing library ImgLib2 (Pietzsch, Preibisch, Tomancak, & Saalfeld, 2012) that

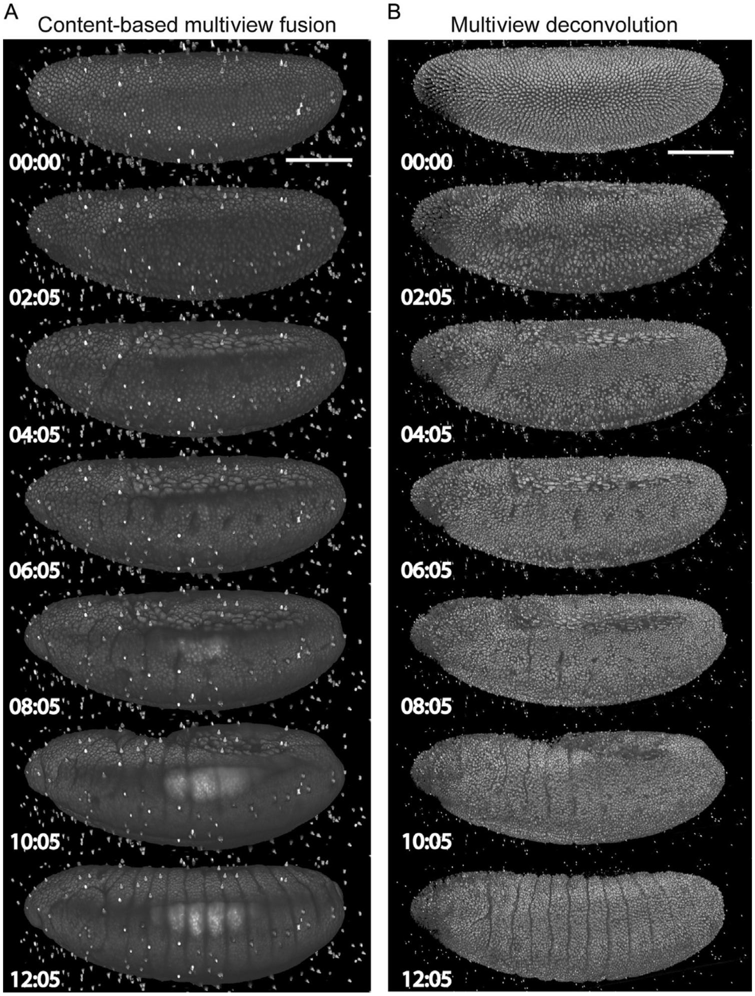
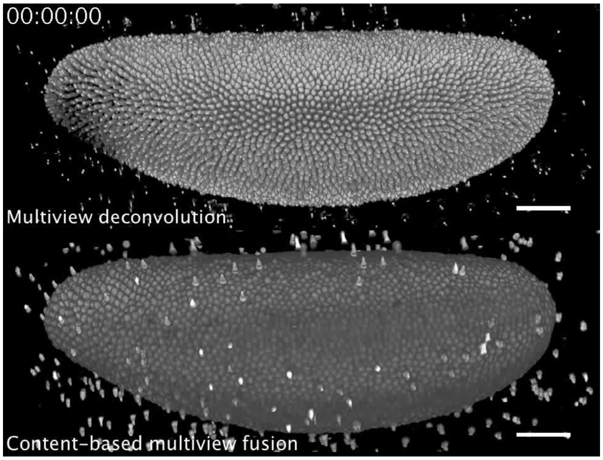


FIGURE 27.8

Processed time-lapse data. (A) Rendered time lapse of the Lightsheet Z.1 recording after content-based multiview fusion. (B) Rendered time lapse of the same recording after multiview deconvolution. The scale bars represent 100 μm.

is capable of building bridges to other open-source platforms. This enables building complex, multiplatform solutions for SPIMage processing and analysis utilizing the strengths of diverse image analysis software (Eliceiri et al., 2012). The open-source model is instrumental in this effort (Cardona & Tomancak, 2012).

We demonstrated the performance of the SPIMage processing platform on data acquired from the commercial Lightsheet Z.1 instrument. It is equally applicable to processing data from homemade SPIM systems, including the open-access hardware



MOVIE 2

Multiview Deconvolution. The rendered Lightsheet Z.1 data after multiview devconvolution and content-based multiview fusion. The contrast and the resolution is significantly improved using multiview deconvolution.

OpenSPIM concept. The synergy of open-source software and open-access hardware ensures that the interdisciplinary research in processing of multiview microscopy data will continue bringing new ideas from computer science to the hands of biologists interested in applying the light sheet technology.

REFERENCES

Ahrens, M. B., Orger, M. B., Robson, D. N., Li, J. M., & Keller, P. J. (2013). Whole-brain functional imaging at cellular resolution using light-sheet microscopy. *Nature Methods*, *10*(5), 413–420.

Arrenberg, A. B., Stainier, D. Y., Baier, H., & Huisken, J. (2010). Optogenetic control of cardiac function. *Science*, *330*(6006), 971–974. http://dx.doi.org/10.1126/science.1195211.

Cardona, A., & Tomancak, P. (2012). Current challenges in open-source bioimage informatics. *Nature Methods*, *9*(7), 661–665. http://dx.doi.org/10.1038/nmeth.2082.

Eliceiri, K. W., Berthold, M. R., Goldberg, I. G., Ibáñez, L., Manjunath, B. S., Martone, M. E., et al. (2012). Biological imaging software tools. *Nature Methods*, *9*(7), 697–710. http://dx.doi.org/10.1038/nmeth.2084.

Fahrbach, F. O., Voigt, F. F., Schmid, B., Helmchen, F., & Huisken, J. (2013). Rapid 3D light-sheet microscopy with a tunable lens. *Optics Express*, *21*(18), 21010. http://dx.doi.org/10.1364/OE.21.021010.

Fischer, M., Haase, I., Wiesner, S., & Müller-Taubenberger, A. (2006). Visualizing cytoskeleton dynamics in mammalian cells using a humanized variant of monomeric red fluorescent protein. *FEBS Letters*, *580*(10), 2495–2502. http://dx.doi.org/10.1016/j.febslet.2006.03.082.

Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, *24*(6), 381–395. http://dx.doi.org/10.1145/358669.358692.

Gao, L., Shao, L., Higgins, C. D., Poulton, J. S., Peifer, M., Davidson, M. W., et al. (2012). Noninvasive imaging beyond the diffraction limit of 3D dynamics in thickly fluorescent specimens. *Cell*, *151*(6), 1370–1385. http://dx.doi.org/10.1016/j.cell.2012.10.008.

Goldberg, I. G., Allan, C., Burel, J.-M., Creager, D., Falconi, A., Hochheiser, H., et al. (2005). The open microscopy environment (OME) data model and XML file: Open tools for informatics and quantitative analysis in biological imaging. *Genome Biology*, *6*(5), R47. http://dx.doi.org/10.1186/gb-2005-6-5-r47.

Gualda, E. J., Vale, T., Almada, P., Feijó, J. A., Martins, G. G., & Moreno, N. (2013). OpenSpinMicroscopy: An open-source integrated microscopy platform. *Nature Methods*, *10*(7), 599–600. http://dx.doi.org/10.1038/nmeth.2508.

Huisken, J., & Stainier, D. Y. R. (2007). Even fluorescence excitation by multidirectional selective plane illumination microscopy (mSPIM). *Optics Letters*, *32*(17), 2608–2610.

Huisken, J., & Stainier, D. Y. R. (2009). Selective plane illumination microscopy techniques in developmental biology. *Development*, *136*(12), 1963–1975. http://dx.doi.org/10.1242/dev.022426.

Huisken, J., Swoger, J., Del Bene, F., Wittbrodt, J., & Stelzer, E. H. (2004). Optical sectioning deep inside live embryos by selective plane illumination microscopy. *Science*, *305*(5686), 1007–1009.

Keller, P. J. (2013). Imaging morphogenesis: Technological advances and biological insights. *Science*, *340*(6137), 1234168. http://dx.doi.org/10.1126/science.1234168 [Review].

Keller, P. J., Schmidt, A. D., Santella, A., Khairy, K., Bao, Z., Wittbrodt, J., et al. (2010). Fast, high-contrast imaging of animal development with scanned light sheet-based structured-illumination microscopy. *Nature Methods*, *7*(8), 637–642. http://dx.doi.org/10.1038/nmeth.1476.

Keller, P. J., Schmidt, A. D., Wittbrodt, J., & Stelzer, E. H. K. (2008). Reconstruction of zebrafish early embryonic development by scanned light sheet microscopy. *Science*, *322*(5904), 1065–1069. http://dx.doi.org/10.1126/science.1162493.

Keller, P. J., & Stelzer, E. H. (2008). Quantitative in vivo imaging of entire embryos with digital scanned laser light sheet fluorescence microscopy. *Current Opinion in Neurobiology*, *18*(6), 624–632. http://dx.doi.org/10.1016/j.conb.2009.03.008.

Krzic, U., Gunther, S., Saunders, T. E., Streichan, S. J., & Hufnagel, L. (2012). Multiview light-sheet microscope for rapid in toto imaging. *Nature Methods*, *9*(7), 730–733. http://dx.doi.org/10.1038/nmeth.2064.

Megason, S. G., & Fraser, S. E. (2007). Imaging in systems biology. *Cell*, *130*(5), 784–795. http://dx.doi.org/10.1016/j.cell.2007.08.031.

Pietzsch, T., Preibisch, S., Tomancak, P., & Saalfeld, S. (2012). ImgLib2—Generic image processing in Java. *Bioinformatics*, *28*(22), 3009–3011. http://dx.doi.org/10.1093/bioinformatics/bts543.

Pitrone, P. G., Schindelin, J., Stuyvenberg, L., Preibisch, S., Weber, M., Eliceiri, K. W., et al. (2013). OpenSPIM: An open-access light-sheet microscopy platform. *Nature Methods*, *10*(7), 598–599. http://dx.doi.org/10.1038/nmeth.2507.

Planchon, T. A., Gao, L., Milkie, D. E., Davidson, M. W., Galbraith, J. A., Galbraith, C. G., et al. (2011). Rapid three-dimensional isotropic imaging of living cells using Bessel beam plane illumination. *Nature Methods*, *8*(5), 417–423. http://dx.doi.org/10.1038/nmeth.1586.

Preibisch, S., Amat, F., Stamataki, E., Sarov, M., Singer, R. H., Meyers, E., et al. (2014). Efficient Bayesian-based multiview deconvolution. *Nature Methods*, In Print. http://dx.doi.org/10.1038/nmeth.2929.

Preibisch, S., Ejsmont, R., Rohlfing, T., & Tomancak, R. (2008). *Towards digital representation of Drosophila embryogenesis. IEEE.* (324–327).

Preibisch, S., Rohlfing, T., Hasak, M. P., & Tomancak, P. (2008). Mosaicing of single plane illumination microscopy images using groupwise registration and fast content-based image fusion. In *Proceedings of SPIE: Vol. 6914. Medical imaging 2008: Image processing* (p. 69140E) http://dx.doi.org/10.1117/12.770893(1).

Preibisch, S., Saalfeld, S., Rohlfing, T., & Tomancak, P. (2009). Bead-based mosaicing of single plane illumination microscopy images using geometric local descriptor matching. *SPIE Medical Imaging*, *7259*, 72592S–725102S.

Preibisch, S., Saalfeld, S., Schindelin, J., & Tomancak, P. (2010). Software for bead-based registration of selective plane illumination microscopy data. *Nature Publishing Group*, *7*(6), 418–419.

Reynaud, E. G., Krzic, U., Greger, K., & Stelzer, E. H. K. (2008). Light sheet-based fluorescence microscopy: More dimensions, more photons, and less photodamage. *HFSP Journal*, *2*(5), 266–275. http://dx.doi.org/10.2976/1.2974980.

Rubio-Guivernau, J. L., Gurchenkov, V., Luengo-Oroz, M. A., Duloquin, L., Bourgine, P., Santos, A., et al. (2012). Wavelet-based image fusion in multi-view three-dimensional microscopy. *Bioinformatics*, *28*(2), 238–245. http://dx.doi.org/10.1093/bioinformatics/btr609.

Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., et al. (2012). Fiji: An open-source platform for biological-image analysis. *Nature Methods*, *9*(7), 676–682. http://dx.doi.org/10.1038/nmeth.2019.

Schneider, C. A., Rasband, W. S., & Eliceiri, K. W. (2012). NIH Image to ImageJ: 25 years of image analysis. *Nature Methods*, *9*(7), 671–675. http://dx.doi.org/10.1038/nmeth.2089.

Siebrasse, J. P., Kaminski, T., & Kubitscheck, U. (2012). Nuclear export of single native mRNA molecules observed by light sheet fluorescence microscopy. *Proceedings of the National Academy of Sciences*, *109*(24), 9426–9431. http://dx.doi.org/10.1073/pnas.1201781109/-/DCSupplemental/Appendix.pdf.

Stelzer, E. H., & Lindek, S. (1994). Fundamental reduction of the observation volume in far-field light microscopy by detection orthogonal to the illumination axis: Confocal theta microscopy. *Optics Communications*, *111*(5), 536–547.

Swoger, J., Verveer, P., Greger, K., Huisken, J., & Stelzer, E. H. K. (2007). Multi-view image fusion improves resolution in three-dimensional microscopy. *Optics Express*, *15*(13), 8029–8042.

Temerinac-Ott, M., Ronneberger, O., Nitschke, R., Driever, W., & Burkhardt, H. (2011). *Spatially-variant Lucy–Richardson deconvolution for multiview fusion of microscopical 3D images. IEEE.* (pp. 899–904).

Temerinac-Ott, M., Ronneberger, O., Ochs, P., Driever, W., Brox, T., & Burkhardt, H. (2012). Multiview deblurring for 3-D images from light-sheet-based fluorescence microscopy. *IEEE Transactions on Image Processing*, *21*(4), 1863–1873. http://dx.doi.org/10.1109/TIP.2011.2181528, A Publication of the IEEE Signal Processing Society.

Tomer, R., Khairy, K., Amat, F., & Keller, P. J. (2012). Quantitative high-speed imaging of entire developing embryos with simultaneous multiview light-sheet microscopy. *Nature Methods*, *9*(7), 1–14. http://dx.doi.org/10.1038/nmeth.2062.

Truong, T. V., Supatto, W., Koos, D. S., Choi, J. M., & Fraser, S. E. (2011). Deep and fast live imaging with two-photon scanned light-sheet microscopy. *Nature Methods*, *8*(9), 757–760. http://dx.doi.org/10.1038/nmeth.1652.