Journal of Microscopy, Vol. 233, Pt 1 2009, pp. 42–60
Received 11 February 2008; accepted 19 June 2008

XuvTools: free, fast and reliable stitching of large 3D datasets

M. EMMENLAUER[*,‖], O. RONNEBERGER[*,‖], A. PONTI[‡], P. SCHWARB[‡], A. GRIFFA[†], A. FILIPPI[§,‖], R. NITSCHKE[¶,‖], W. DRIEVER[§,‖] & H. BURKHARDT[*,‖]
[*]*Computer Science, Albert-Ludwigs-University Freiburg, Germany*
[†]*BIOp, Ecole Polytechnique Fédérale de Lausanne, Switzerland*
[‡]*FMI (Part of Novartis Research Foundation), Basel, Switzerland*
[¶]*ZBSA, Live Imaging Center, Albert-Ludwigs-University Freiburg, Germany*
[§]*Faculty of Biology, Albert-Ludwigs-University Freiburg, Germany*
[‖]*Centre for Biological Signaling Studies (bioss), Albert-Ludwigs-University Freiburg, Germany*

Key words. Bleaching, correlation, large scale microscopy, mosaicing, stitching, virtual microscopy, 3D.

Summary

Current biomedical research increasingly requires imaging large and thick 3D structures at high resolution. Prominent examples are the tracking of fine filaments over long distances in brain slices, or the localization of gene expression or cell migration in whole animals like *Caenorhabditis elegans* or zebrafish. To obtain both high resolution and a large field of view (FOV), a combination of multiple recordings ('tiles') is one of the options. Although hardware solutions exist for fast and reproducible acquisition of multiple 3D tiles, generic software solutions are missing to assemble ('stitch') these tiles quickly and accurately.

In this paper, we present a framework that achieves fully automated recombination of tiles recorded at arbitrary positions in 3D space, as long as some small overlap between tiles is provided. A fully automated 3D correlation between all tiles is achieved such that no manual interaction or prior knowledge about tile positions is needed. We use (1) phase-only correlation in a multi-scale approach to estimate the coarse positions, (2) normalized cross-correlation of small patches extracted at salient points to obtain the precise matches, (3) find the globally optimal placement for all tiles by a singular value decomposition and (4) accomplish a nearly seamless stitching by a bleaching correction at the tile borders. If the dataset contains multiple channels, all channels are used to obtain the best matches between tiles. For speedup we employ a heuristic method to prune unneeded correlations, and compute all correlations via the fast Fourier transform (FFT), thereby achieving very good runtime performance.

We demonstrate the successful application of the proposed framework to a wide range of different datasets from whole zebrafish embryos and *C. elegans*, mouse and rat brain slices and fine plant hairs (trichome). Further, we compare our stitching results to those of other commercially and freely available software solutions.

The algorithms presented are being made available freely as an open source toolset 'XuvTools' at the corresponding author's website (http://lmb.informatik.uni-freiburg.de/people/ronneber), licensed under the GNU General Public License (GPL) v2. Binaries are provided for Linux and Microsoft Windows. The toolset is written in templated C++, such that it can operate on datasets with any bit-depth. Due to the consequent use of 64bit addressing, stacks of arbitrary size (i.e. larger than 4 GB) can be stitched. The runtime on a standard desktop computer is in the range of a few minutes. A user friendly interface for advanced manual interaction and visualization is also available.

Correspondence to: O. Ronneberger. Tel: +49-(0)761-203-8285; Fax: +49-(0)761-203-8262; e-mail: ronneber@informatik.uni-freiburg.de

Introduction

Recent applications in systems biology require optical analyses of very large samples at sub-micrometer resolution. Optical resolution in the sub-micron range can only be achieved with lenses of high numerical aperture (NA), which are characterized by relatively short working distances. In addition, high sampling rates and magnification are often required. As a consequence of the high magnification, the field of view (FOV) captured by a camera chip of finite size is rather small, and it will need a large number of 3D image stacks ('tiles') to capture a structure with diameters in the millimeter range, which is a typical size for a small animal organ, an embryo, or other biologically relevant specimen.

Thanks to the steady evolution of 3D microscopy techniques over the past decades, the acquisition of large datasets is no longer the limiting factor in the analysis of large specimen. Recent techniques like Two-Photon microscopy have extended the capability of the confocal microscope to image through thicker tissues, whereas very fast acquisition is made possible by spinning-disc or slit-scan confocal microscopy. Moreover, recent objectives provide a nearly distortion-free image over the whole extent of the FOV.

Tiles are acquired by sequentially moving the stage of the microscope to a new position (that is partially overlapping with the previous) and locally imaging the specimen in 3D at high resolution. Different modalities exist: between consecutive acquisitions, the user can manually move the stage by using a joystick, or the stage can be programmed to move on a regular grid or other specified pattern. Depending on the equipment, the stage positions might be recorded along with the data. It is important to notice, however, that these positions usually have a significant error and a stitching algorithm that only relies on them will generate inaccurate mosaics.

Once the series of tiles has been acquired, the challenge lies in the accurate reconstruction of the sample from the number of independently imaged 3D sub-volumes. While 2-dimensional stitching of single-plane (2D) images (especially in photography) is now well established and several software solutions exist, 3D stitching is still in its infancy. Simply extending these tools to the 3D case is in most cases not feasible, since they are optimized for the largely different transformation and distortion models that are needed in photography.

When refractive index mismatches are avoided as far as possible, highly corrected objectives like Apochromats, and no negative zoom optic are used, one can safely assume that complex distortions and rotations in the signal between successively acquired tiles are negligible and that translation is the only transformation that has to be taken into account when performing the reconstruction. Neglecting complex distortions and rotations drastically reduces the complexity of the problem.

In this article, we describe an approach for fully automated, fast and precise stitching of 3D datasets. The consequent limitation to the compensation of 3D translations only (all the other distortions are better compensated on the imaging side by use of appropriate optics) allows the usage of very robust techniques that are in most cases equivalent to an exhaustive search over all possible placements of the tiles. At the same time considerable effort was spent to reduce the run times of the algorithms.

The article is structured as follows. In the section 'State of the art', we give an overview of the state of the art in stitching algorithms and available software. 'Materials and methods' contains the detailed description of the proposed stitching approach with the sub-sections 'finding pairwise displacements', 'determination of the absolute tile positions', and 'compensation of bleaching artifacts'. 'The XuvTools software' section highlights implementation details and workflow of our XuvTools toolset. Finally, the results of the proposed stitcher on a wide range of datasets are presented in the 'Results' section and compared to the results of commercially and freely available stitching programs.

State of the art

The number of publications concerning the stitching (or mosaicing) of 2D photographs is enormous (e.g. at the time of writing this article, the Annotated Computer Vision Bibliography lists 362 papers in the chapter 'Mosaic Generation' (Price, 2008)). The best solutions in this field are not necessarily the ones best suited for 3D microscopic applications. This is mainly due to the limitation of these approaches to 2D images, and due to the significant perspective and lens distortions found in photographs that need high-order transformation models. An exhaustive search in such a high-dimensional parameter space is unfeasible, such that in this field the use of landmarks (found manually or automatically) is the most appropriate approach.

In contrast to this, the stitching of image tiles recorded by a microscope equipped with a scanning stage only needs to take a comparably low-dimensional parameter space into account. Due to the fixed optics no perspective distortions or scale changes are present. Microscope objectives are well corrected such that lens distortions are very low, and, due to the scanning stage, rotations of the tiles have little impact. In the remaining low-dimensional parameter space not only landmark based techniques (e.g. Becker *et al.*, 1996; Can *et al.*, 2002; Al-Kofahi *et al.*, 2003; Bajcsy *et al.*, 2006) but also pixel-intensity-based techniques like cross-correlation or mutual information are applied (e.g. Capek & Krekule 1999; Karen *et al.*, 2003; Slamani *et al.*, 2006; Thévenaz & Unser, 2007). Generally, landmark-based techniques are quite fast, but need to make certain assumptions about the dataset, e.g. the existence of blob-like objects (Becker *et al.*, 1996), fine filaments (e.g., Can *et al.*, 2002; Al-Kofahi *et al.*, 2003) or closed contours (Bajcsy *et al.*, 2006). Pixel-intensity-based techniques on the other hand are usually slower, but applicable to every type of dataset. A possibility to combine the advantages of both techniques is to search for salient points (like edges or corners, which exist in every type of dataset) and limit the pixel based techniques to small windows around these salient points (e.g., Chow *et al.*, 2006; Sun *et al.*, 2006). Correlation on such small windows is quite fast and insensitive to global intensity gradients.

While this search for best correspondences between tiles is the most crucial part in a stitching approach, there are some pre- and post-processing steps needed for a fully functional framework. Before the search for the correspondences takes place, nearly all algorithms need approximative positions of the tiles. All pixel-based approaches mentioned above rely on positions of the scanning stage or require manual pre-alignment. Only some of the landmark-based techniques can infer this prior information from the dataset itself.

After a successful correspondence search, one usually is confronted with the problem that every tile overlaps with multiple other tiles, such that a global solution must be found, which best satisfies all relative correspondences. Typical approaches for this are the search for an optimum that satisfies the relative displacements by the method of least squares (e.g. Sun *et al.*, 2006), fitting techniques robust to outliers like Ransac (Fischler & Bolles, 1981), graph-based methods (e.g. Chow *et al.*, 2006; Bujnak *et al.*, 2007), or dynamic programming techniques (e.g. Appleton *et al.*, 2005).

Finally, for the combination of all tiles, certain intensity corrections and blending techniques are needed, to obtain a seamlessly stitched mosaic image. This correction must be done according to the type of dataset. For example, histology images, recorded with transmitted light techniques, have as the main source for intensity variations at the border the shading effects of the microscope. (Sun *et al.*, 2006) correct for them by estimating these shading effects using a second order polynomial function. For mosaics with a very high number of tiles, and homogeneous intensities over the whole sample, such a shading can be estimated by min- and max-images over all tiles (Chow *et al.*, 2006). Other approaches suggest a combination of the images in the gradient domain and find the resulting image as a solution of the Poisson equation (e.g. Zomet *et al.*, 2006).

Available software solutions

Still many biologists apply manual stitching to their datasets, for a lack of helpful automated software. This can be a tedious and difficult task. To the aid comes automatic stitching software (also denoted as mosaicing software) that tries to recombine the tiles either based on an initial pre-alignment or fully automatically. The pre-alignment can be gained from the scanning stage positions for the individual tiles (if available), or can be provided by the user via interactive positioning of the tiles on the screen.

Software that uses pre-alignment is usually fast and quite successful in adjusting small positional errors, but can seldom cope with coarse initial misalignment. This can be a major drawback because the outcome is very much user-dependent. Furthermore, even if the scanning stage positions of the individual tiles are stored, it is usually hard to import them into the stitching software. So an important feature for a stitching software is the ability to find the initial positions without any additional information.

With the objective of providing a brief survey of the common stitching software available at the present, we consider the applications reported in Table 1.

Adobe Photoshop is an example among others of a tool that can perform completely automatic 2D stitching without any *a priori* knowledge on the stage positions. However, we experienced mosaic reconstruction failures when the content of information in an image is low, as can be the case with 2D neuron tracing.

The other tools we consider have been specifically developed for microscopy. They perform semi-automatic 2D stitching, meaning that they need some form of a priori knowledge about the tile positions. *ImageAccess* employs pairwise tile stitching by asking the user to define a couple of correspondence points in the two images. Then a rigid transformation of one tile with respect to the other is performed. It is evident that this approach is very much user-dependent and requires some expertise. Results are typically acceptable for small mosaics, but the approach is not applicable in the case of huge datasets, also because in practice the stitching operation is particularly slow.

The stitch functionality in *MediaCybernetics Image-Pro* is targeted towards stitching of overlapping grid-mode acquisitions: a general scheme of acquisition that works as an initial definition of tile positions has to be defined. The possibility of inserting blank images in the list of tiles to be stitched and the fact that consistent movements of the tiles with respect to the initial grid positions are allowed permits the stitching of free-mode acquisitions too. However, the setup is time consuming and could require more than one attempt. Moreover, the more the acquisition scheme strays from a grid-like regularity, the more the stitching operation is subject to failure.

Table 1. Commercially and freely available stitching software we have tested.

Company/Institute	Name	Data type	Automation	*A priori* info	License
Adobe	Photoshop	2D	full	None	Commercial
Imagic AG	ImageAccess	2D	semi	Reference points	Commercial
MediaCybernetics	Image-Pro	2D	semi	Defined grid	Commercial
Biomedical Imaging, EPFL	MosaicJ for ImageJ	2D	semi	Manual pre-positioning	Free
CarlZeis MicroImaging	AxioVision (> V. 4.5)	2D, 3D	semi	Scanning stage positions	Commercial
Molecular Devices	Metamorph	2D, 3D	semi	Scanning stage positions	Commercial
Biomathematics, ASCR	GlueMRC / LinkMRC	2D, 3D	semi	Reference points	Free

MosaicJ is a 2D stitching plugin for the public-domain image analysis package ImageJ, developed by the biomedical imaging group of the École Polytechnique Fédérale de Lausanne. Starting from an initial manual positioning of the tiles, it performs image registration with sub-pixel precision (Thévenaz & Unser 2007). Because of the manual initialization, the stitching of large numbers of tiles usually takes quite long and depends on the user's capability of reconstructing the initial placement.

Zeiss AxioVision and *Molecular Devices Metamorph* are tools that offer the possibility to stitch 3D stacks as well as 2D ones. (To be precise, Metamorph only performs 2D stitching out of the box. The pseudo-3D stitching is provided by Visitron Systems GmbH, an European Molecular Devices distributor, in form of a Metamorph journal.) Both applications can process only specific file formats (zvi for AxioVision, stk for Metamorph) and need to find the information about scanning stage positions in the image metadata. Thanks to the fact that they use the stage positions as initial coordinates, the registration results on the plane are generally good. Concerning the 3D stitching, a real volumetric approach is missing. Stacks are stitched plane-by-plane, and no correction is performed in the third dimension. This can cause the appearance of artifacts in the stitched volume not only between different tiles but also between the planes of a single tile. Moreover, stacks with different dimensions in z-direction cannot be processed. Both software tools had problems if the size of the mosaic image exceeded 1 Gigabyte. Several of our test-datasets exceed this limit by far.

The Department of Biomathematics of the Academy of Sciences of the Czech Republic distributes its software tools *GlueMRC* and *LinkMRC* for free (Karen *et al.*, 2003). They represent the only software we could obtain that is able to perform real 3D stitching, in contrast to the pseudo-3D approach used by Metamorph and AxioVision that stitch plane by plane independently or use a single reference plane for the whole stack. GlueMRC is a Windows software with a straightforward user interface. Stitching is performed by adding one tile at a time to the current mosaic image. For each added tile, the user is asked to select a feature in the tile and a corresponding feature in the mosaic image. From these, the 3D fine placement is obtained automatically by the software, typically with very good results (compare Fig. 21). The main drawback is the need to specify corresponding features by hand, which requires the user not only to know the rough recording positions, but also to have enough expertise and patience to find matching structures. Especially if heavy bleaching is evident in the sample, this can become difficult. Another limitation is that GlueMRC cannot guarantee that the stitched dataset will be the one with best possible placement of the tiles that compose it. Since at iteration j the user can only specify one feature point in tile f_j and one in the current mosaic, the correct positioning of tile f_j is calculated only against the one tile in the mosaic that contains the user-selected feature. Information from overlaps with other areas of the mosaic are not considered for refinement. This can lead to increasingly large errors the more stacks are added to the mosaic.

Main contributions

As mentioned above, the main goal of our proposed 3D stitching algorithms is robustness, speed and accuracy. The software should be well suited for daily routine work and be able to cope with any type of structures that appear in biological research.

- Omitting rotation and subpixel-displacements allows to place the raw data without modification (i.e. no interpolation) into a final mosaic.
- True 3D approach. Tiles may be imaged anywhere in 3D space (as long as sufficient overlap is given). No reduction to 2D as in other approaches.
- No need for any meta-data or assumptions about the type of objects (like in landmark-based approaches). Just give the tiles to the program and get the final mosaic.
- Full multi-channel support. Finds the solution that best satisfies the correspondences in all channels (no reduction to single channel as in other approaches). The final mosaic image has all channels precisely aligned with each other.
- Very fast, but still a good approximation of an exhaustive search over all possible placements of all tiles.
- No limitations on data type (8bit gray, 16bit, 32bit float), no limitations on size (especially no 4 GB size limit).
- Quality control: if no perfect solution can be found for the whole set of tiles, the user is informed about the largest possible subset(s) that can be stitched. No suboptimal solution is returned.
- Maximization of the space occupied by unbleached tiles rather than blending between bleached and unbleached ones, which could result in smearing and lower signal to noise ratio. Tries to adjust bleached stacks to eliminate discontinuities.

Materials and methods

The recombination of multiple datasets is logically split into three distinct parts (see Fig. 1). First, the *pairwise displacements* between tiles must be estimated. This is done by analyzing the tiles for similarities. Second, the *absolute positions* of the tiles have to be determined in a way so that the positions optimally satisfy all pairwise displacements. Third, all tiles are combined into a *single large image* (in the following denoted 'mosaic image'), where care has to be taken at the overlapping tiles borders that discontinuities are minimized and useful information maximized.

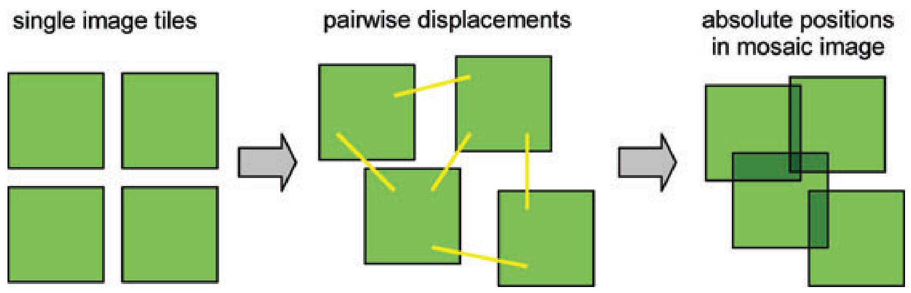


Fig. 1. From single 3D image tiles pairwise displacements are determined. These displacements in turn allow the computation of the absolute tile positions in the final 3D mosaic image.

Fast pairwise displacement estimation

If no information about the positioning of tiles is available, neither from the scanning stage positions nor from a manual pre-alignment, this information needs to be obtained by estimating the pairwise displacements (or translations) between tiles from similarities in the signal. Comparing all tiles in a brute force approach can be computationally very expensive, since there are $N(N-1)/2$ stack tuples to test against each other. This requires a very fast and optimized algorithm for the comparison. In the sub-section, 'Heuristic for finding complete graphs', we introduce a heuristic method that can often prune displacement tests; however, the worst case scenario of comparing all possible tuples remains valid.

Our approach for the estimation of displacements utilizes phase-only correlation (POC) based on the normalized Fourier spectrum. This method allows to efficiently determine pairwise best displacements while being invariant to linear gray value changes. In the following we derive the expression for the POC (Eq. 5).

Given two image stacks $f(x, y, z)$ and $g(x, y, z)$, the 3D correlation c_{fg} under translation (t_x, t_y, t_z) is defined as

$$c_{fg}(t_x, t_y, t_z) := (f \circ g)(t_x, t_y, t_z)$$
$$= \sum_{x=0}^{N_x-1} \sum_{y=0}^{N_y-1} \sum_{z=0}^{N_z-1} f(x, y, z) \cdot g(x - t_x, y - t_y, z - t_z). \quad (1)$$

The Fourier isomorphism states that Eq. (1) can be computed in frequency space as follows:

$$c_{fg} = f \circ g = \mathrm{DFT}^{-1} \left(\mathrm{DFT}(f) \cdot \mathrm{DFT}(g)^*\right), \quad (2)$$

where * denotes the complex conjugate and DFT denotes the discrete Fourier transform. Since the DFT assumes periodic signals, the signals f and g must be zero-padded to double size before the DFT is applied. The discrete Fourier spectra $F(u, v, w)$ and $G(u, v, w)$ are given by the DFT:

$$F(u, v, w) = \frac{1}{N_x N_y N_z} \cdot \sum_{x=0}^{N_x-1} \sum_{y=0}^{N_y-1} \sum_{z=0}^{N_z-1} f(x, y, z)\, e^{-j2\pi\left(\frac{xu}{N_x} + \frac{yv}{N_y} + \frac{zw}{N_w}\right)} \quad (3)$$

$$G(u, v, w) = \frac{1}{N_x N_y N_z} \cdot \sum_{x=0}^{N_x-1} \sum_{y=0}^{N_y-1} \sum_{z=0}^{N_z-1} g(x, y, z)\, e^{-j2\pi\left(\frac{xu}{N_x} + \frac{yv}{N_y} + \frac{zw}{N_w}\right)}. \quad (4)$$

The DFT is efficiently implemented by the FFT, which has a total run time complexity of $O(N^3 \log N)$ for 3D signals of size N^3.

The POC c_{pho} is obtained from the correlation by removal of the amplitude information from the Fourier spectrum, leaving the more information-rich phase intact:

$$c_{\mathrm{pho},fg} = \mathrm{DFT}^{-1} \left(\frac{F \cdot G^*}{|F| \cdot |G|} \right) \quad (5)$$

It was shown by Ito *et al.* (2004) that the POC has a precise peak for shifted images. The position of the peak is used as an estimation of the pairwise displacement, and is invariant to linear gray value changes and very robust against noise. Moreover, correlation, and i.e. POC, performs well on downscaled signals. Even with low quality downscaling like nearest neighbor, the relative height of the peak stays almost the same. At scalings down to 10% of the original image size, almost no notable neighbouring maxima occurred in our experiments (see Fig. 2), while downscaling significantly reduces runtime complexity of the FFT. In the implementation, tiles are scaled to a size of a power of two, since the FFT can work more efficiently for these.

Precise pairwise displacement estimation

The POC is applied to downscaled image tiles, so the precision of the location of the peak is limited by the scaling factor. The precise pairwise displacement is determined on the unscaled tiles using normalized cross-correlation (NCC). The correlation coefficient c_{norm} of the NCC also delivers a quantitative measure of similarity that could not have been gained from the POC. The NCC c_{norm} of signals f, g under displacement

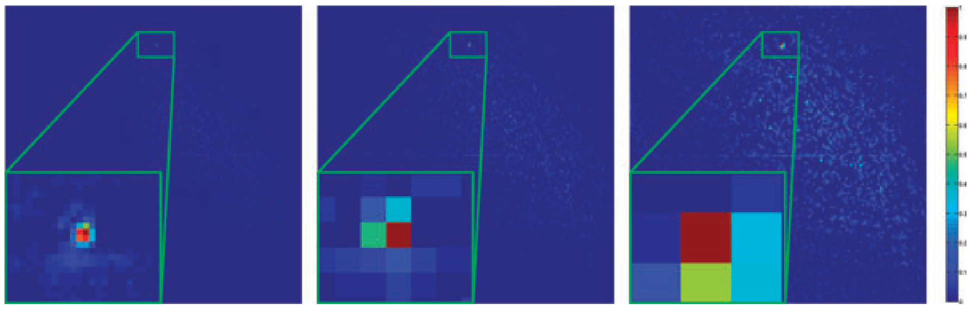


Fig. 2. POC of two tiles, scaled to 100%, 25% and 12.5% of the original tile size. While the size of the peak grows with the pixel sizes, the relative height of the peak compared to the other correlation coefficients stays almost the same.

(t_x, t_y, t_z) is given in Eq. (6):

$$c_{\mathrm{norm}}(t_x, t_y, t_z) = \frac{\sum_{x=0}^{N_x-1}\sum_{y=0}^{N_y-1}\sum_{y=0}^{N_z-1}\left(f(x,y,z)-\overline{f}\right)\cdot\left(g_{t_x,t_y,t_z}(x,y,z)-\overline{g_{t_x,t_y,t_z}}\right)}{\sigma(f)\cdot\sigma(g_{t_x,t_y,t_z})}, \quad (6)$$

where $\sigma(f)$ denotes the standard deviation of signal f, and g_{t_x,t_y,t_z} denotes the shifted version of the second signal g. Normalizing a signal f by subtracting the mean $\overline{f}$ and dividing through the standard deviation $\sigma(f)$ maps all linear transformations $f' = af + b, a, b \in \mathbb{R}$ to the normalized signal f_n, making it invariant to linear gray value changes.

In our application, the shifted signal g_{t_x,t_y,t_z} is a sub-signal that is extracted at the relative position $(-t_x, -t_y, -t_z)$. Note that both the mean value as well as the standard deviation of this signal are different for each (t_x, t_y, t_z). This dependency is often neglected, especially if the cross-correlation is computed by means of the FFT. In images with very heterogeneous structures (and consequently with large variations of the mean and the standard deviation within small areas), this approximation can lead to significantly biased results. Therefore, to compute the unbiased normalized cross-correlation by means of the DFT, some additional efforts are necessary (Ronneberger, 1998). First we use the equivalence

$$\sum_{\mathbf{x}}\left(f(\mathbf{x})-\overline{f}\right)\cdot\left(g_{\mathbf{t}}(\mathbf{x})-\overline{g_{\mathbf{t}}}\right) = \sum_{\mathbf{x}} f(\mathbf{x})g_{\mathbf{t}}(\mathbf{x}) - \frac{1}{N}\sum_{\mathbf{x}} f(\mathbf{x})\sum_{\mathbf{x}} g_{\mathbf{t}}(\mathbf{x}) \quad (7)$$

to move the mean values $\overline{f}$ and $\overline{g_{\mathbf{t}}}$ to a second independent term. For a more compact notation we use a vectorial notation here with $\mathbf{x} = (x, y, z)$, $\mathbf{t} = (t_x, t_y, t_z)$ and $N = N_x \cdot N_y \cdot N_z$. To replace the direct correlation in the first term with the cyclic correlation that is provided by the DFT-approach, both signals are enlarged to $(N_x + T_x) \times (N_y + T_y) \times (N_z + T_z)$, where T_x, T_y, T_z are the number of translations in x, y, z-direction. Signal f is enlarged by padding with zeros, and signal g by simply cropping a larger region. The enlarged signals are denoted as f_E and g_E. The normalized cross-correlation (6) can thus be rewritten using the DFT as

$$c_{\mathrm{norm}}(\mathbf{t}) = \frac{\mathrm{DFT}^{-1}\left(\mathrm{DFT}(f_E)\cdot\mathrm{DFT}(g_E)^*\right) - \frac{1}{N}\sum_{\mathbf{x}} f(\mathbf{x})\sum_{\mathbf{x}} g_{\mathbf{t}}(\mathbf{x})}{\sigma(f)\cdot\sigma(g_{\mathbf{t}})}. \quad (8)$$

The fast computation of the sums $\sum_{\mathbf{x}} g_{\mathbf{t}}(\mathbf{x})$ and standard deviations $\sigma(g_{\mathbf{t}})$ is done by an iterative scheme with linear complexity. For the computation of $\sigma(g_{\mathbf{t}})$ again the equivalence relation

$$\sum_{\mathbf{x}}\left(g_{\mathbf{t}}(\mathbf{x})-\overline{g_{\mathbf{t}}}\right)^2 = \sum_{\mathbf{x}} g_{\mathbf{t}}^2(\mathbf{x}) - \frac{1}{N}\left(\sum_{\mathbf{x}} g_{\mathbf{t}}(\mathbf{x})\right)^2 \quad (9)$$

is used, such that the same fast iterative scheme for its computation can be used.

Placement of correlation windows at interest points

The overlapping region of tiles can be used for correlation. For this, we divide the overlapping region into smaller windows, which has several advantages compared to blindly correlating the whole overlap. First, using multiple windows (and normalizing each of them independently to zero mean and unit variance as described for the NCC) minimizes the influence of large-scale intensity variations inside the overlapping region (which can arise i.e. from bleaching). Second, correlation is faster as long as the combined volume of the windows is smaller than that of the whole overlapping region. A further advantage is given by targeting the windows to regions with high image contrast. These regions are likely candidates for high information density and should give more informative correlation result. As a consequence the signal-to-noise ratio of the obtained correlation is better than that of a correlation of the full overlapping region. This selection is depicted in Fig. 3, where fine neuronal axons are detected (blue windows) in the overlapping part (yellow dashed region) of two neighbouring tiles. Only at these positions the normalized cross-correlation will be calculated.

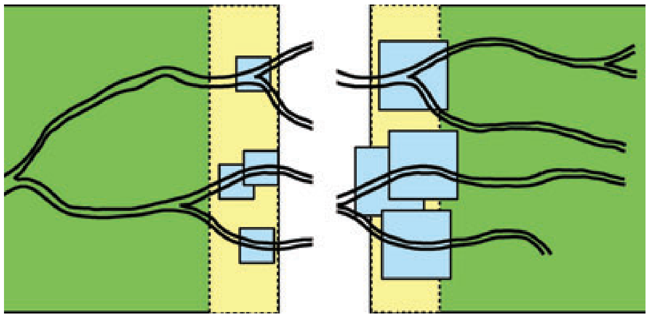


Fig. 3. Schematic of neuronal processes in two neighbouring tiles. The blue windows in the left tile are the interest points P. The larger blue windows in the right tile mark the region against which the interest point will be correlated using NCC.

As a measure of image contrast we use the gradient magnitude $||\nabla f||$ of signal f, and integrate over small regions (about 2/3 of the correlation window size). The n regions with the highest summed contrast are selected, and their centers $\mathbf{p}_i$ are stored as a set of 'interest points' $P = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n\}$. It is sufficient to extract interest points from the overlapping region of one of the tiles, since the overlaps are assumed to be identical. For this, we have arbitrarily chosen the overlapping region with the higher mean gray value, as the darker one usually has a worse signal to noise ratio (again as a result of bleaching). If more than one channel is present, the interest point detection is performed on all available channels individually, since distinctive structures can vary between them, resulting in an individual set of interest points $P_1, \ldots, P_k$ for each of the k channels. Once interest points $P_1, \ldots, P_k$ have been extracted for all channels, the per-window NCC can be calculated. The iteration over all channels and interest points is shown in pseudo-code in Algorithm 1. Note that the coefficients computed by the NCC can be summarized over the channels, leading to an additive information gain. For every pair of tiles it is sufficient if one of the channels provides enough information for a successful stitching, no matter which channel. See Fig. 14 for an example where in some regions only channel one, in others only channel two contained useful structure. Stitching with only one of the channels would not have been possible.

Algorithm 1 Interest-point-based, normalized cross-correlation for two tiles f and g

```
Input: For the k channels: the image tiles $f_1, \ldots, f_k$ and $g_1, \ldots, g_k$,
  sets of interest points $P_1, \ldots, P_k$, displacement estimate $\mathbf{d}_{\mathrm{estimate}fg}$
  between tiles $f$ and $g$.
Output: $\mathbf{d}_{fg}$ as the pixel-precise displacement between tiles $f$ and $g$.
// Initialize correlation coefficients to zero
1:  $c_{\mathrm{norm}} := 0$
2:  for all channels do
3:    for all interest points $\mathbf{p}_i \in P_{channel}$ do
4:      $w_f$ := Window taken from tile $f$ at position $\mathbf{p}_i$, zero-padded to
        the size of the fixed window
5:      $w_g$ := Window taken from tile $g$ at position $\mathbf{p}_i + \mathbf{d}_{\mathrm{estimate}fg}$,
        zero-padded where overlapping the tile border
//      Summarize correlation coefficients over all channels and positions
6:      $c_{\mathrm{norm}}$ += normalized cross-correlation of $w_f$ and $w_g$
7:    end for
8:  end for
9:  Normalize $c_{\mathrm{norm}}$ with number of loop iterations:
    $c_{\mathrm{norm}} /= |\mathrm{channels}| \cdot |\mathrm{interestpoints}|$
10: $\mathbf{d}_{fg} := d_{\mathrm{estimate}fg}$ + Position of the maximum of $c_{\mathrm{norm}}$
```

Finding absolute tile positions from pairwise displacements

Once the precise pairwise displacements have been determined, the absolute tile positions can be computed from them. The normalized correlation coefficient c_{norm} from the NCC is used as a measure of quality of the corresponding correlation. Using a threshold δ, the displacements with low correlation coefficient $c_{\mathrm{norm}} < \delta$ can be removed from the set.

From the remaining displacements with high coefficient, we estimate the absolute stack positions. Two problems arise: first, there can be more displacements than tiles (as each tile overlaps with several others) so the placing is not straightforward. Second, if the displacements form loops (compare Fig. 4) they can contradict each other. This second problem is rare, but currently being worked on using an improved placement algorithm. To find a globally optimal placement from more displacements than tiles, we use a solution that weights all displacements equally and solves them by minimizing the squared error. For that, we set up the system of linear equations that relates the required absolute positions p_f, p_g to the measured displacement d_{fg}:

$$d_{fg} = p_f - p_g. \tag{10}$$

For N tiles there can be up to $N(N-1)/2$ linear equations, forming a probably over-determined equation system as

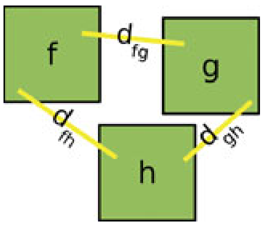


Fig. 4. Three tiles f, g and h with pairwise displacements between them. If all three displacements have a sufficiently high correlation coefficient they will be used for computing absolute positions. Since they form a loop they can be contradictory (e.g. $d_{fh} + d_{gh} < d_{fg}$).

shown exemplary for three tiles:

$$\begin{pmatrix} d_{fg} \\ d_{gh} \\ d_{fh} \end{pmatrix} = \begin{pmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} p_f \\ p_g \\ p_h \end{pmatrix} \qquad (11)$$

The position of one of the tiles must be fixed in advance, the other positions will then be computed relative to this arbitrary position. We chose the position of the first tile to be zero $p_f = (0, 0, 0)$ and can then drop the corresponding (i.e. the first) column in the matrix:

$$\begin{pmatrix} d_{fg} \\ d_{gh} \\ d_{fh} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ -1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} p_g \\ p_h \end{pmatrix}. \qquad (12)$$

Solving the equation system is done using the singular value decomposition (SVD) for all dimensions x, y, z independently. The SVD leads to a solution that is optimal in the sense of smallest squared error.

In some cases it can happen that not all tiles are connected to the mosaic with a sufficiently high correlation coefficient. This is especially the case if a tile (or its borders) consist only of background, or if the overlap between tiles is not wide enough for correlation. In the case that not all tiles are connected to each other via displacements after applying the threshold, the user is informed about the problem, is presented with the list of connected components and can stitch these independently of each other. Examples can be seen in Fig. 15 where two large components are stitched independently due to a too small overlap between them, and in Fig. 13 where the tiles containing only background noise could not be stitched and were left out.

Heuristic for finding complete graphs

Our software does not make assumptions by default about initial tile positions but tries to find them automatically, yielding $N(N-1)/2$ pairwise tests in the worst case. If the recording positions are available from the microscope, they are used to estimate which tiles have been recorded close to each other, and only these are used for the correlation. In this case, we test a tile against its neighbours in a relatively large region around it (i.e. a radius of half the tiles size), which is motivated by the fact that table positions are typically biased with errors.

If no recording positions are available, it is unknown which tiles have to be correlated against which others. We have arbitrarily selected to start with tiles from close positions in their storage order, i.e. first test each tile with the immediately following one in the file, then continue with increasing distance. This is motivated by the fact that in a typical microscopic imaging scenario, tiles are recorded next to each other to avoid gaps in the mosaic image, yielding close positions in the file storage. Note that this heuristic method does not reduce the total number of tested pairs of tiles.

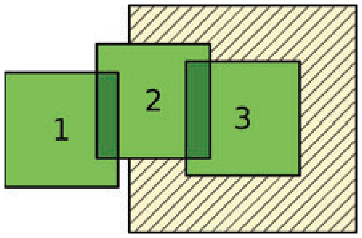


Fig. 5. Tiles that have been correlated form connected components, of which the relative positions can already be computed. These allow implications about the remaining correlation candidates, i.e. the pair (1, 3) does not need to be tested anymore since tile 1 is not in the search radius around tile 3.

However, in our tests it has shown to find the good correlations earlier on.

Once several good correlations have been found, the pairs of tiles that match with high certainty (i.e. with sufficiently high correlation coefficient) form a connected component. The relative positions of tiles in a connected component can be computed by means of the SVD as described above. These positions allow implications about other test candidates from the same component. Figure 5 shows an example of three tiles, where the relative positions of (1, 2) and (2, 3) indicate that (1, 3) do not overlap, since tile 1 is not in the search radius (depicted as yellow hatched region) of tile 3. All pairs (f, g) from the same connected component where f is not inside the search radius of g can be pruned from the test set. It is evident that the sooner large connected components emerge, the more implications about the remaining test candidates can be drawn (resulting in a large speedup). From our tests it has emerged that this heuristic is useful for arbitrary recording positions, and results in an especially large speedup for many common recording patterns (see Table 2).

Bleaching correction

When recombining image tiles to one mosaic image, borders with a strong drop in intensity can emerge (see Fig. 6a). These borders are not artifacts introduced by the positioning, but result from bleaching, the destruction of fluorophores that are hit by the excitation light during acquisition of the prior recorded tiles. This bleaching of neighboring tiles occurs even if the laser (in the case of confocal microscopy) is switched off exactly at the border of the current tile. See Fig. 7 for an illustration. In the previous processing step 'Precise pairwise displacement estimation', the pixel-precise match for all overlapping tiles was obtained. Making use of the fact that two overlapping regions should have identical content, the pixel-by-pixel gray value mapping in the overlapping region can be computed. Let f be the unbleached tile and g the bleached tile, $\mathbf{p}_i$ denotes a position in f and $\mathbf{p}_i' = \mathbf{p}_i + \mathbf{d}_{fg}$ the corresponding position in g. By mapping the gray values from the darker overlapping region to the brighter one, we

Table 2. Amount of correlations performed with our proposed heuristic, compared to brute force and known recording positions.

Dataset	3D tiles	Brute force	With heuristic	With stage positions
Cortex of rat (Fig. 16)	$4 \times 5 = 20$	190	73	55

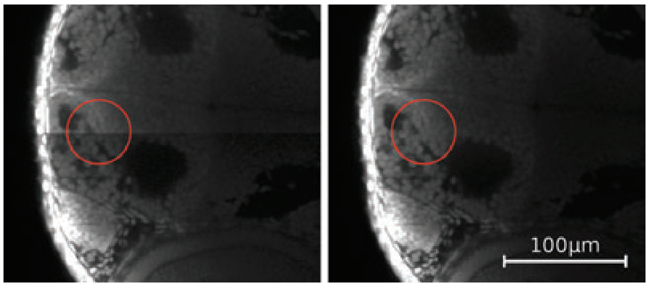


Fig. 6. Zebrafish 72hpf embryo stained with TOTO3 to visualize all cell nuclei, two neighbouring tiles of one recording. (a) The top tile has been imaged first, the bottom tile third in the series. Bleaching in the bottom tile is clearly visible. (b) Our proposed bleaching correction applied to the bottom tile before recombination. No visible discontinuities remain.

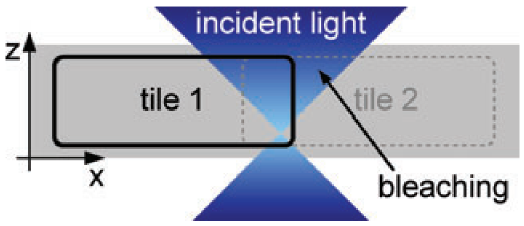


Fig. 7. During recording of the first tile not only the overlapping regions, but also certain areas beside the tile are bleached. This is a consequence of the recording in different layers and the conical shape of the excitation light beam. If bleaching is not corrected, significant discontinuities can appear in the stitched image.

compute the gray value co-occurrence matrix. We have found bleaching to be modelled by a linear process $f(\mathbf{p}_i) = m \cdot g(\mathbf{p}_i^{'}) + b$ that maps gray values from the bleached overlapping region to those of the unbleached one. From the co-occurrence matrix, the linear parameters m, b of the mapping can be deduced by means of the least squares method. We also infer the mapping variance as a measure of quality of the linear parameter estimation. An example is shown in Fig. 8 (right).

In most experiments, a small step in the bleaching effects at the border of the overlapping region was observed. This is modelled by the correction of the factor m with a certain constant suppression factor $w < 1$ outside the overlapping region.

The original (i.e. unbleached) gray values $g^{'}$ of the bleached tile g are reconstructed only outside of the overlapping region, because inside this region the unbleached tile f provides the identical content with better signal to noise ratio. For the reconstruction, the linearly corrected intensities are cross-faded to the original intensities with increasing distance d to the tile border. The decreasing contribution of the corrected signal is modeled with a Gaussian-shaped distribution

$$N_\sigma(d) = e^{-d^2/\sigma^2}, \tag{13}$$

where $N_\sigma(d) \in [0, 1]$. The reconstructed gray values in tile g are

$$g'(\mathbf{p}_i') = N_\sigma(d) \cdot (m \cdot g(\mathbf{p}_i') + b) + (1 - N_\sigma(d)) \cdot g(\mathbf{p}_i'). \tag{14}$$

The standard deviation σ can be seen as the bleaching width in µm, and has to be estimated by the user through visual inspection, since no prior knowledge about the unbleached gray values outside the overlapping region is available. However, σ stays typically the same for all samples with the same staining and recording settings, and can therefore be re-used. Fig. 16 shows an example of bleaching, and our proposed corrected mosaic image.

The XuvTools software

Implementation details

The software we have described in this paper is freely available as an open source toolbox. We currently provide ready-to-use binaries for Linux and Microsoft Windows, and a complete build environment for both platforms. The software is entirely written in C++ and should be easily portable to other platforms as well. We make use of several open source libraries, most notably the fastest Fourier Transform in the West (FFTW) (http://www.fftw.org/) and a highly optimized C++ class library for scientific computing 'blitz++' (http://www.oonumerics.org/blitz/). For file input/output, at present NetCDF (http://www.unidata.ucar.edu/software/netcdf/), HDF5 (http://www.hdfgroup.org/products/hdf5/index.html) and the current Bitplane Imaris 5.5 (ims) are supported. More major microscopy formats are being worked on, the most important ones being tiffs, tiff stacks, Zeiss lsm and Metamorph stk. Users interested in working with NetCDF or HDF5 might be interested in our ImageJ reader/writer plugin (freely available at http://lmb.informatik.uni-freiburg.de/lmbsoft/imagej_plugins/).

Our software is split into core library, command line tools for batch processing and a graphical user interface (GUI, in the XuvTools-GUI module). Certain features like interactive

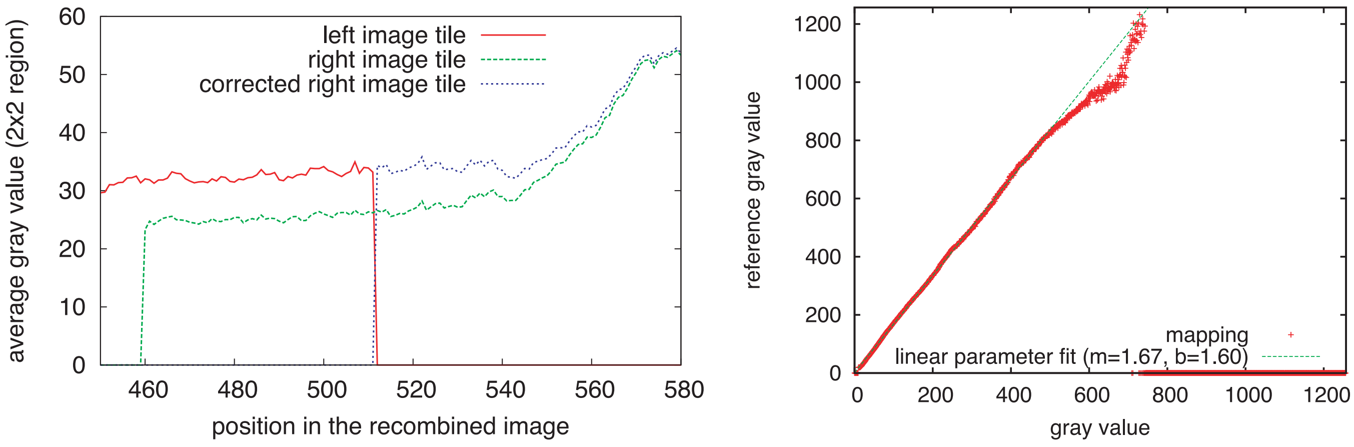


Fig. 8. Left figure: border region between two overlapping image tiles. The average gray value of the left tile is shown in red, the average gray value of the right tile in green. The blue curve denotes the gray value after correction, with an average factor of $m = 1, 29$ in the overlapping part, and a smooth decrease of this factor to 1 beside the stitching border. Right figure: mapping of gray values inside the overlapping region of two image tiles.

refinement or manual correction of the stitching are best used through the GUI. Most notable implementation details are:

- Independence of data type through the use of C++ templates.
- All internal functions and libraries have been tested to support arbitrarily large datasets, i.e. can read and write files of more than 4 GB in size and utilize more than 4 GB of RAM on 64bit platforms.
- We make use of the highly optimized FFTW for the Fourier transform.
- For fastest operation, all datasets are kept in RAM. Typical memory usage is twice the size of the dataset. Optimizations that allow operation on low-memory machines are being worked on.
- The command line tools are split into three parts ('tile alignment', 'bleaching correction', 'stitching with blending'). The parts communicate via status-files, therefore it is easy to replace one part with a different program (e.g. leave out bleaching correction for samples without bleaching artifacts).
- Support for multi-threading via OpenMP is already implemented (currently in alpha status).

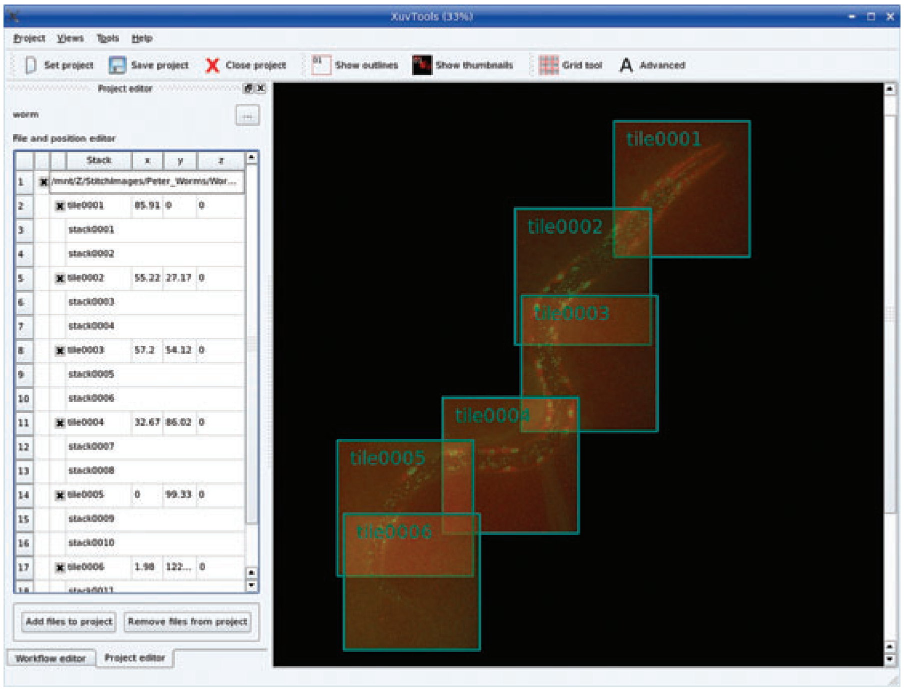
The goal of the XuvTools-GUI module is to offer a user interface to the command line XuvTools (see Fig. 9) that is on one side appealing to users that are less comfortable with the command line, and on the other hand greatly enhances the interactivity with the software. In more detail, the GUI allows the iterative refinement of the stitching by allowing user intervention at all steps. Thus, for instance, the fast pairwise displacement estimation can be reviewed graphically and corrected by dragging misplaced tiles in the approximately correct position. Parameters for the creation of the final overlay like the blending between tiles or the strength and width of the bleaching correction can be adapted interactively. Moreover, the graphical user interface offers advanced handling of the initial tile positioning by allowing their initialization on regular shapes and grids. The XuvTools-GUI module is written in C++ using Trolltech's cross-platform framework Qt (http://trolltech.com/products/qt), which allows to create Linux and Microsoft Windows programs from the identical source code.

A sample workflow

The actual use of the XuvTools software is structured in a clear step-wise approach. Both the command line and the GUI share the same workflow (as depicted in Fig. 10), and cover the sequence of algorithms presented in 'Materials and methods'. In the following, a typical workflow is presented.

First, input files are added to the project. The files are scanned, and the information relevant to the stitching is extracted. This information can be reviewed and adjusted in the GUI with the Project Editor, that also allows to disable incidental tiles that should be omitted from the stitching. The command line tools export the result of the file inspection as a comma-separated value (CSV) table that can be read and written by many common office packages like OpenOffice Calc or Microsoft Excel. This allows for batch processing of datasets that were taken with the same microscope settings. For the file formats NetCDF and HDF5 that are not bound to a specific internal organization, the user manually assigns the internal paths to the datasets in the GUI or CSV-file.

If stage positions could be gathered from the input file(s), they are used for the display of the tiles in the GUI, otherwise the user can use the Grid Tool to arrange the tiles in one from a series of common grid-layouts. Also, tiles can be (re)arranged by hand. The positions can optionally be used as initial values for the stitching, since positioning can improve speed and decrease the chance of failure when stitching more difficult datasets. However, this is not essential to a successful outcome



	Stack	x	y	z
1	/mnt/Z/StitchImages/Peter_Worms/Wor...			
2	tile0001	85.91	0	0
3	stack0001			
4	stack0002			
5	tile0002	55.22	27.17	0
6	stack0003			
7	stack0004			
8	tile0003	57.2	54.12	0
9	stack0005			
10	stack0006			
11	tile0004	32.67	86.02	0
12	stack0007			
13	stack0008			
14	tile0005	0	99.33	0
15	stack0009			
16	stack0010			
17	tile0006	1.98	122...	0
18	stack0011			

Fig. 9. The XuvTools–GUI user interface. The mosaic image shows a tiling of a worm (*Caenorhabditis elegans*) expressing nuclear periphery marker (GFP lmn-1).

and was not used for the results we present in the next chapter.

The GUI can display the progress of stitching graphically by combining the stitching algorithms (first three sub-sections of ‘Materials and Methods’) with the algorithm for finding the absolute tile positions (see ‘Materials and methods’), that will reposition the tiles to the currently estimated position at every iteration. In the ideal case, at the end of stitching all tiles will already belong to the same connected component (i.e. the complete mosaic). If this is not the case, the user can change the threshold parameter for finding the absolute tile positions, and apply it again. This is very fast, since the displacement estimation does not need to be repeated.

In the GUI, this is done by changing the slider ‘correlation threshold’. On the command line, the tool find_absolute_positions can be run, passing the new value as a parameter. The threshold needs to be decreased if more than one connected component was found. The threshold should be increased if tiles tend to merge to a large blob. A good strategy is to start with a high threshold, and to decrease it stepwise until a complete mosaic image is found.

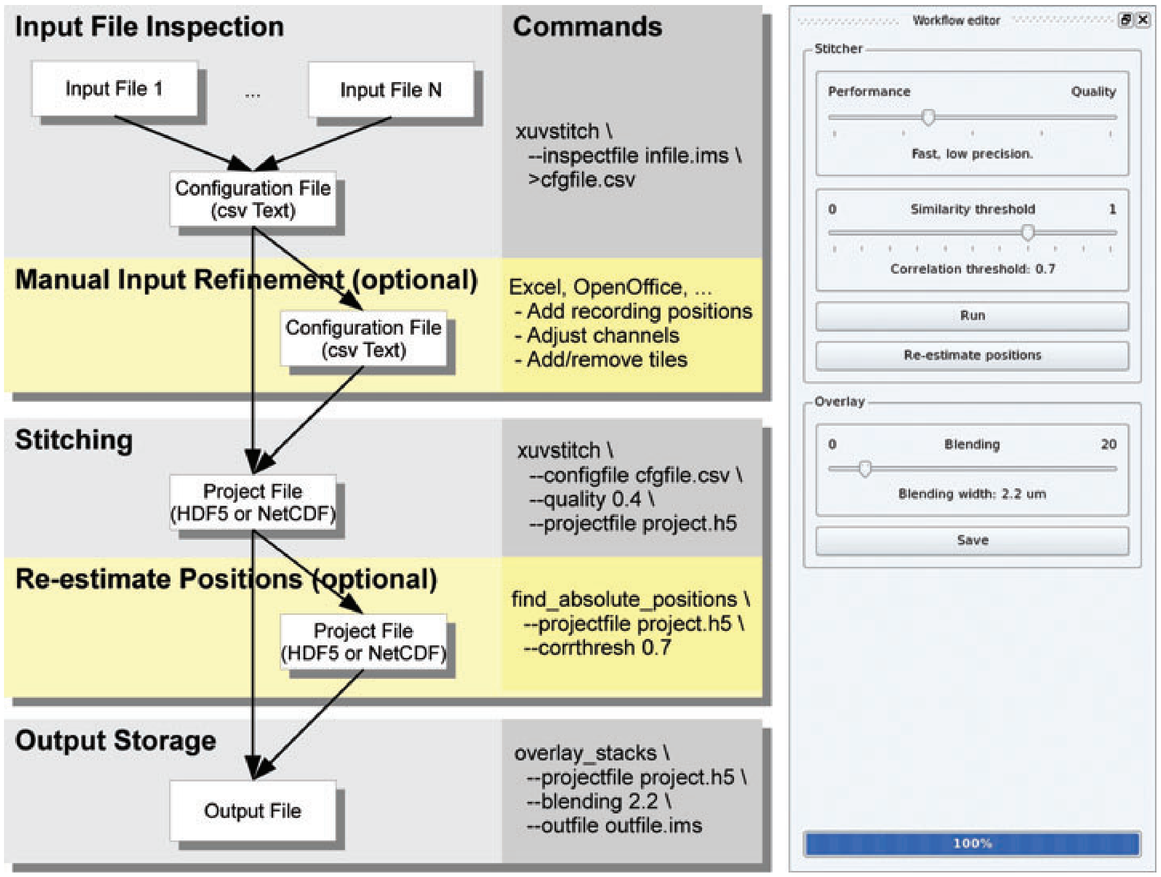
At any time the current layout of the dataset can be saved to an output file, which is typically done as the last step of the workflow. On the command line, overlay_stacks is run on the project file to export the current mosaic to a file. In the GUI the save-button can be used to select an output file.

Additional parameters (e.g. the size and search radius of correlation windows, etc.) are available for advanced users, however there was no need to change the default values for all test datasets described in this paper. A detailed description can be found in the supplied manual file. It is worthwhile to note that our implementation was fast enough for a fluent work flow even on the largest datasets we have encountered so far. A state of the art single core computer is sufficient, as long as it provides RAM of roughly twice the size of the dataset.

Results

We have applied our software toolset to various biomedical datasets in which cellular structures need to be imaged at high resolution in the context of whole organs or organisms. These include: whole zebrafish brain and head at single cell resolution, neurons and axons in mouse and rat brain slices, whole *C. elegans* animals, and trichome hairs of the *Arabidopsis* plant. With the pruning method (‘Heuristic for finding complete graphs’), very large datasets of up to 68 tiles of size 512 × 512 × 73 voxels (Fig. 15) recorded at arbitrary positions could be stitched in less than 6 min on a state of the art single core computer, given a sufficient amount of RAM. For smaller datasets, stitching takes on average less time than needed to load and store the data, typically below 3 min. See ‘Results’ for more detailed benchmark results.

To measure the quality of the final mosaic image, visual inspection is most often sufficient, discontinuities and artifacts are obvious to laymen. However, if the final mosaic image should be used for automatic processing, like in filament tracing, more stringent constraints about the introduction of discontinuities in the image structure need to be enforced.



```
xuvstitch \
  --inspectfile infile.ims \
  >cfgfile.csv
```

Excel, OpenOffice, ...
- Add recording positions
- Adjust channels
- Add/remove tiles

```
xuvstitch \
  --configfile cfgfile.csv \
  --quality 0.4 \
  --projectfile project.h5
```

```
find_absolute_positions \
  --projectfile project.h5 \
  --corrthresh 0.7
```

```
overlay_stacks \
  --projectfile project.h5 \
  --blending 2.2 \
  --outfile outfile.ims
```

Correlation threshold: 0.7

Blending width: 2.2 um

Fig. 10. Sample workflow of the XuvStitch command line tools (left), and a screen shot of the workflow pane of the graphical user interface. The optional bleaching correction that can be applied after running stitching is not shown here. In the command line example, 'infile.ims' is an Imaris file that contains the recorded tiles (stored as time series), 'outfile.ims' is an Imaris file that contains the stitched Volume.

Discontinuities can be both in space (from bad positioning of tiles) as well as intensity (most often as a result of bleaching). We use color overlay for visual inspection, stitching artifacts become visible by imbalanced color mixing of colorized overlapping tiles. Figure 18 is an example of errors that can very well be detected in the color overlay. We have also applied a filament tracing software ('NeuronTracer' from Bitplane AG) to stitched neuron images to compare correctly and incorrectly traced filaments (Fig. 20).

Datasets that benefit from small correlation windows

The zebrafish dataset shown in Fig. 11 has a large extent in z-direction of about 500 planes (respectively 500 μm). Inside the overlapping regions, structures vary between homogeneous gray (i.e. in the eye) to fine neuronal axons in the brain. This along with the strong nonlinear gradient introduced by bleaching, makes the use of small correlation windows preferable. A smaller window is subject to only a portion of the gradient of bleaching, so the normalized cross-correlation is less affected by it, leading to good correlation results on this dataset. The interest point detection was also rendered more robust by avoiding low-contrast areas, i.e. where no staining took place like the mouth/nasal area, eye pupil, etc.

Another example for the effectiveness of our approach is shown in Fig. 12, where axonal trajectories, and fine filament structures generate signal intensities only little over the background noise. NCC of the whole overlapping region would yield a low correlation coefficient due to the low overall contrast. This stitching also shows that the proposed computation of absolute tile positions is robust. Even though not all displacement estimations where correctly determined by the POC, a subset of the displacements was sufficient for placing the tiles.

A more complicated example is introduced by the neuronal cell in Fig. 13, where the cell and its fibers are highly overexposed, whereas the background consists mainly of noise. The image was done as a recording example for structured light imaging (Apotome, Carl Zeiss MicroImaging). The overexposed cell has a constant value of 255 and does not contain structure, whereas the background is superimposed by noise and not useful for correlation. Here, our proposed

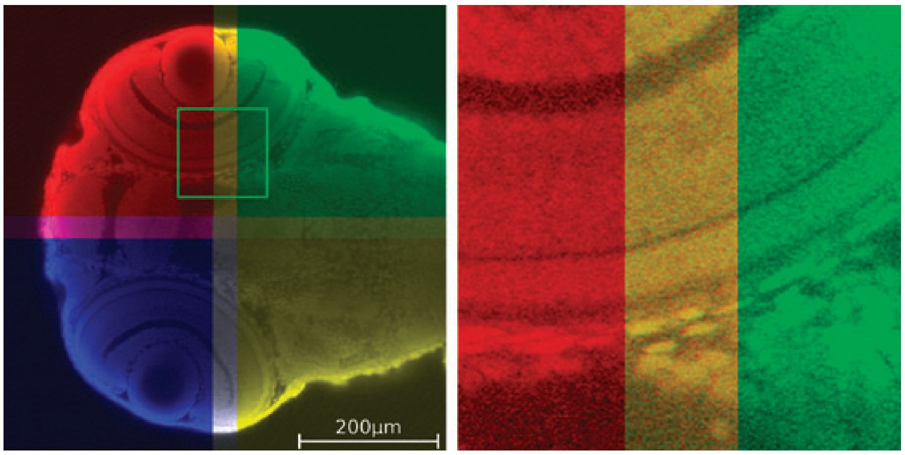


Fig. 11. Zebrafish embryo stained with TOTO3 to visualize all cell nuclei. Middle section of a confocal stack, gamma corrected ($\gamma = 0.3$). For quality control each tile is colored differently. The homogeneous mix color in the overlapping regions indicates a good placement of the tiles.

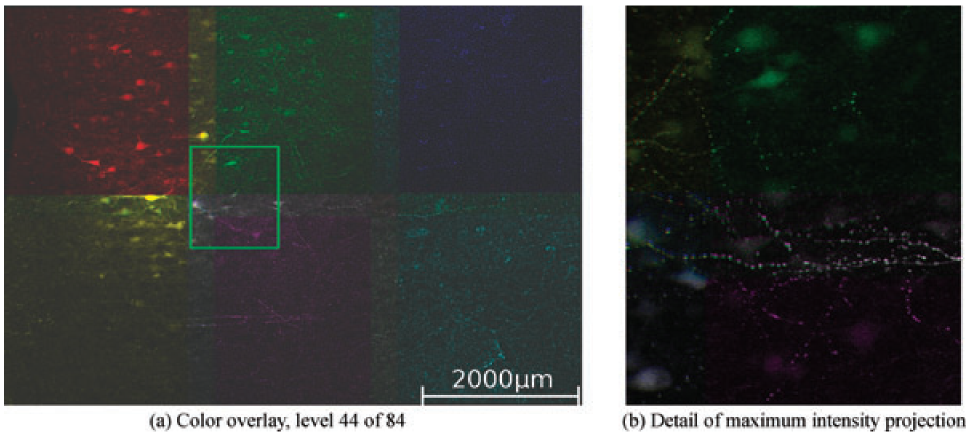

2000μm
(a) Color overlay, level 44 of 84
(b) Detail of maximum intensity projection

Fig. 12. Biocytin filled bitufted interneuron in the primary somatosensory cortex of the rat, visualized by Alexa Fluor-488. (a) Color overlay indicates a good matching of the long, fine axon in the center that is hard to see even for humans. (b) The fine axonal branches can be seen better in the maximum intensity projection.

interest point detection based on the gradient magnitude reaches maximizing positions at the exact border of the cell where the value drops from overexposed to almost zero. This is indeed the only area with sufficient structure to be correlated, which was successfully done. Note that some tiles could not be stitched because they contained only background. These can be detected from their low correlation coefficient. They were rejected by the software rather than placed in a suboptimal way.

Results that make special use of the automatic pre-alignment

For almost none of our tested datasets, microscope stage positions were available. After stitching, however, often a regular grid structure emerged that showed proof that the computed stitching positions were indeed correct. The *Arabidopsis* trichome dataset from Fig. 14, however, was recorded at arbitrary positions in 3D space with no regular pattern, as can be seen from the estimated tile positions in Fig. 14(a). Since most tiles overlap mainly in z-direction, it is very difficult for the scientist to place the tiles manually. Also, from the maximum intensity projection in Fig. 14(c) it can be seen that in z-direction the top region (trichomes) is only imaged in the green channel whereas the bottom region (*Arabidopsis* leaf) is only imaged in the red channel. The phase-only correlation was able to find the displacements between tiles automatically, and by combining the information from both channels it was successful on the whole leaf.

The very large dataset in Fig. 15, consisting of 68 tiles, was also recorded at arbitrary positions with no stage positions available. The tiles were recorded with 16 bit precision to maintain the details in spite of the large intensity difference

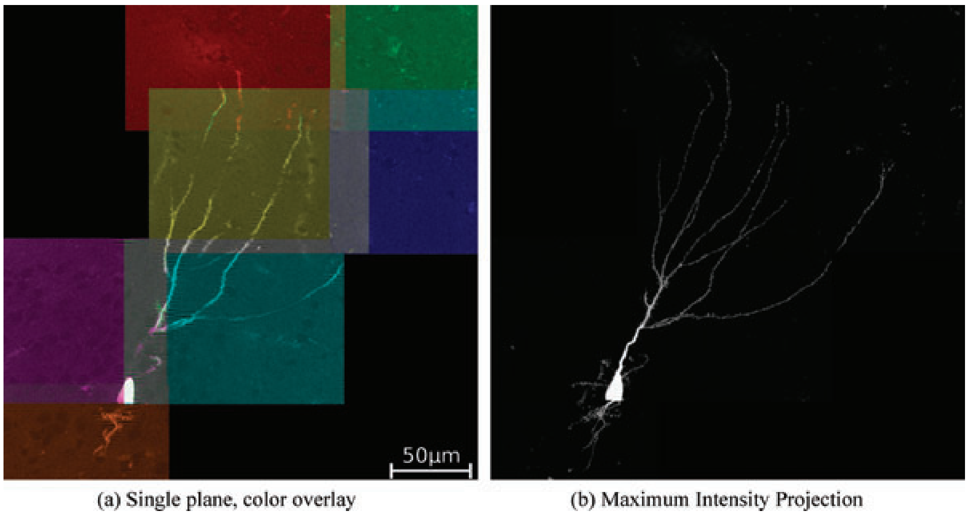

50μm
(a) Single plane, color overlay
(b) Maximum Intensity Projection

Fig. 13. Mouse hippocampal granule cell selected as an example for very difficult stitching conditions. Recorded with structured light technique (Zeiss Apotome) and too high exposure times (necessary to visualize the fine intensity structures in the sample), which induced significant stripe artifacts to the stacks. Nonetheless, the proposed algorithms were able to find a reasonable solution for the seven tiles (out of 12) that contain filaments.

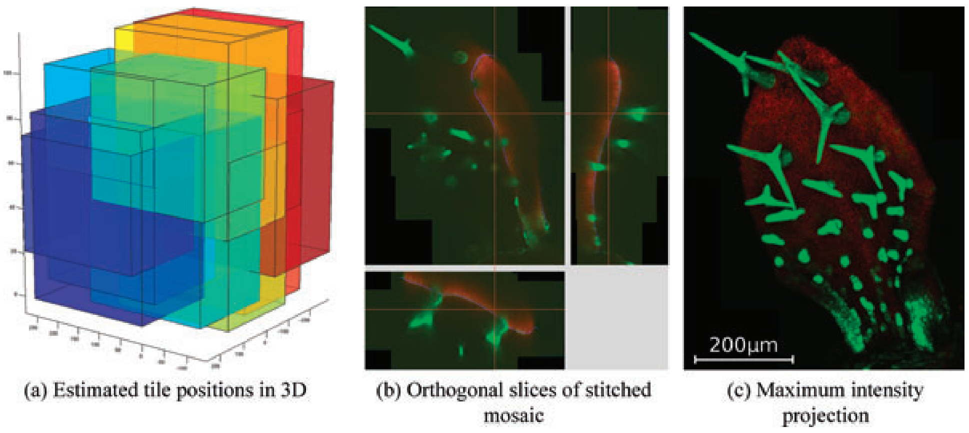

200μm
(a) Estimated tile positions in 3D
(b) Orthogonal slices of stitched mosaic
(c) Maximum intensity projection

Fig. 14. *Arabidopsis* trichome dataset. Image consists of 12 stacks with two channels, recorded at arbitrary 3D positions (a). This image could not be stitched by using only one of the two channels, because some overlapping regions contain valuable information only in the red channel, others only in the green channel (c).

between the upper region and the lower region. Therefore, for the human reference stitching we needed to apply gamma correction to achieve a more homogeneous brightness distribution. This on the other hand is a disadvantage for the visibility of the fine fibers in the lower region, so stitching remained difficult for humans. The POC, however, benefits from the additional information delivered by 16 bit data, and provides very precise and distinct results on the dataset.

Bleaching correction and recombination of the mosaic image

The large dataset in Fig. 16 consists of two channels, of which only the first exhibited strong bleaching. By visual inspection, we estimated a bleaching width $\sigma = 45$ μm and a suppression factor $w = 0.8$ for the intensity drop. Both parameters need to be estimated manually, but can often be re-used from sample to sample as long as the same staining and imaging parameters are used. The mosaic image built from the bleaching-corrected tiles shows significantly less discontinuities at the tile borders (Fig. 16b).

If visible borders emerge in the recombined mosaic image that are not caused by bleaching, blending of tiles is a good way to improve the continuity. We implement a sigmoid blending with variable width. All previously shown mosaic images did not have these artifacts and have therefore been blended with only a very narrow sigmoid (width approx. 4 pixels). The *C. elegans* dataset in Fig. 17, however, had visible artifacts at the tile borders from stray light in the microscope. Blending with a

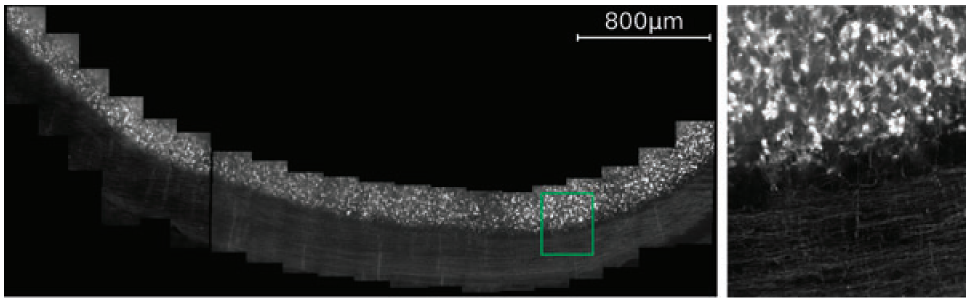


Fig. 15. *Ex vivo* image of an horizontal section of the cerebellar lobule V of a transgenic mouse expressing membrane-bound green fluorescent protein (Thy-1 mGFP) in a subset of neuronal cells. Stitching sectioned in 20 + 43 stacks. Due to too small overlap of less than 16 pixels between neighboring tiles in left third, it was not possible to stitch it to one complete mosaic image.

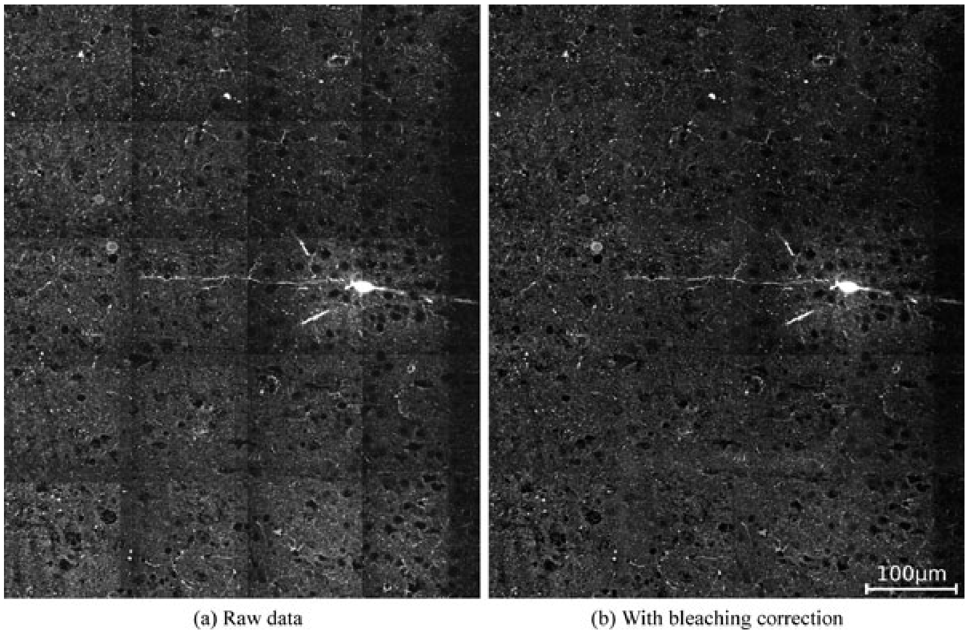


Fig. 16. Biocytin filled pyramidal neuron and parvalbumin specific antibody labeled interneurons in the primary somatosensory cortex of the rat. (a) The stitched image using the raw data of the tiles shows significant dark stripes at the tile borders. (b) The proposed bleaching correction is able to significantly reduce these discontinuities. A full correction can not be achieved by this technique, though.

sigmoid of 30 pixels width mostly eliminates the artifacts, and thereby improves the overall continuity in the signal.

Lens distortions

As noted earlier, our software does not correct for lens distortions as that would significantly increase the number of degrees of freedom of the transformation. These distortions should better be corrected by improving image acquisition. An example for barrel distortions is shown in Fig. 18 where the correction ring of a multi-immersion objective was not set to the correct refractive index of the used immersion medium. This lead to aberrations in the periphery of each image and thus at the tile border.

Comparison with other stitching software

Among the stitching software solutions we have considered in our comparison ('State of the art'), only Metamorph from Molecular Devices, AxioVision from Carl Zeiss MicroImaging, and GlueMRC from the Academy of Sciences of the Czech Republic are able to perform a 3D stitching.

We have been provided with many test datasets from different imaging facilities, recorded in many different file formats. AxioVision and Metamorph can read scanning stage positions only from their own file formats (zvi and stk, respectively), but not from any of the others. Since both software toolsets essentially require scanning stage positions for stitching, we could only test them on the datasets that were recorded with the according software. This is an implicit

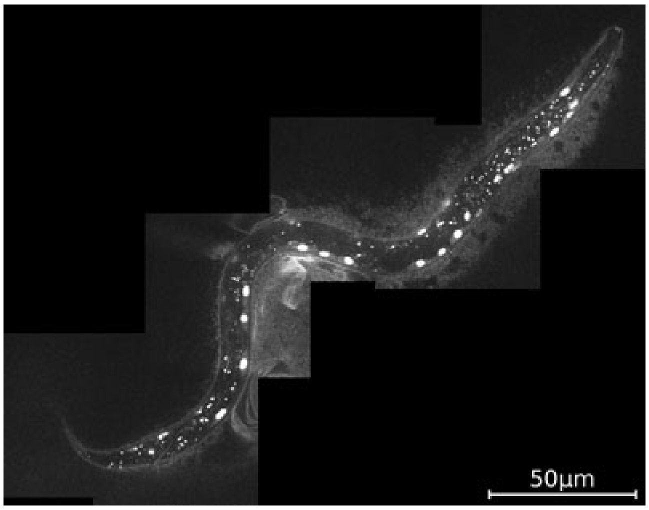


Fig. 17. Tiling of a worm (*Caenorhabditis elegans*, first larval stage) expressing a muscle-specific nuclear GFP at low level. Diameter of the worm is about 15 µm. The stitching consists of six tiles, which have rather low signal to noise ratio.

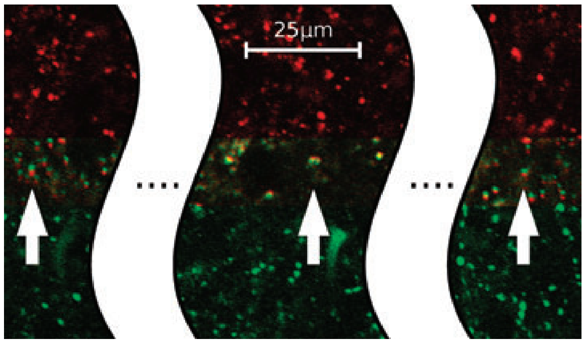


Fig. 18. Example for very strong barrel distortions. At the left and right sides the red tile should have been placed higher (visible red edge at bottom of particle). At the center the red tile should have been placed lower (visible red edge at top of particle). The proposed algorithm does not correct for such distortions, because this would significantly reduce speed and robustness. Recent high-end objectives are so well corrected that such problems can be solved on the imaging side (in this case the correction ring for the refractive index was misadjusted). The total tile width is 1366 pixels, or 190 µm.

limitation of the software and can not be easily circumvented on the users side.

In AxioVision's zvi file format, we had only one rather 'easy' dataset available. This dataset contains dense structures and does not need corrections in z-direction, so the pseudo 3D approach of AxioVision performed well.

In Metamorph's stk format, several datasets with different degrees of difficulty were available. While Metamorph's own pseudo 3D approach performed well on the datasets with dense structures and no z-offset, it had significant problems with the more difficult datasets that contain only very sparse structures and had axial mismatches. Since Metamorph performs a plane-by-plane stitching of 3D stacks, good results in one plane do not necessarily imply good results in other planes. We therefore validate the results on z-direction projections and volumetric rendering.

The maximum intensity projection (MIP) shown in Fig. 19 is a detailed view of a stitching consisting of 22 stacks. While in several of the single planes the Metamorph stitching appears seamless, in some planes discontinuities can be found. As seen in the projection, artifacts are also introduced by the dislocation of planes against each other. These artifacts are most prominent not at the tiles borders but instead in the center, leading to severe misjudgement of the tile content. The highlighted region of Fig. 19 shows two examples of structures that appear multiple times in the projection while being there only once in the original tiles. The manifold structures can be seen in the volume rendering as well (Fig. 20). When applying filament tracing (NeuronTracer from Bitplane AG) the additional fibers are detected as neuron fibers or, with worse results, introduce discontinuities. This can make a successful tracing more difficult.

The stitching with GlueMRC can generally be considered as good. From the manually selected pair of features between the currently added tile and the mosaic, it is able to fine-place the added tile automatically and with high precision, as shown in Fig. 21(a). Due to the high amount of time it takes for the user to manually select corresponding features, we have stitched only four of the originally 40 stacks with GlueMRC. Our XuvTools stitching of the same four stacks provides a result of quality similar to the stitching of GlueMRC, see Fig. 21(b). It was achieved fully automatically and in a fraction of the time.

Benchmark

All results shown in this paper have been computed on a Linux server in our lab. For our tests a single core of an 8-core Xeon server running at 3 Ghz equipped with 32 GB of RAM was used. The results should be equivalent to those on a state of the art Core 2 Duo desktop computer with 16 GB of RAM. Note

Table 3. Benchmark for Stitching on 2.6 GHz Xeon (single core) with 32 GB of RAM.

Dataset	Figure	Final Size [voxel]	Tiles × Channels	Time
Cortex of the rat	16	5000 × 6300 × 128	20×2	2 min 33 s
Mouse, cereb. sec.	15	9300 × 4200 × 77	68	6 min 10 s
Single neuron cell	12	2800 × 2000 × 80	6	13 s
Zebrafish	11	970 × 970 × 480	4	30 s

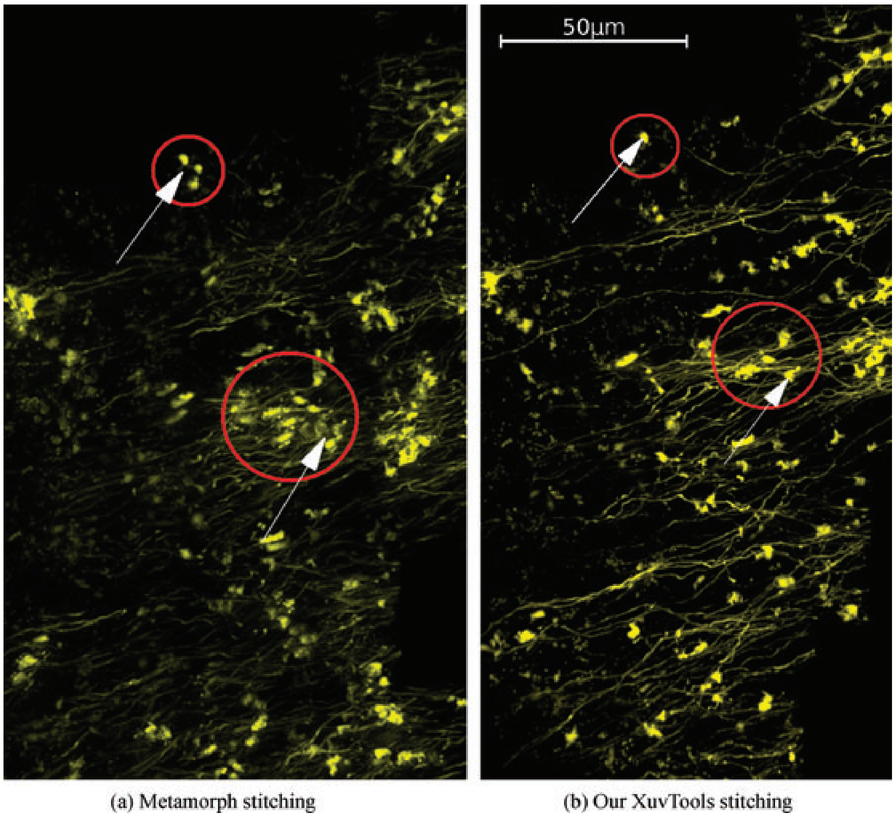


Fig. 19. *Ex vivo* image of the mossy fiber projection in the hippocampal CA3 region of a 3 month-old Thy-1 mGFP transgenic mouse. The maximum intensity projection shows ambiguities in the Metamorph stitching that introduce the impression of more presynaptic terminals than available in the original recording.

that while improvements are being implemented to be able to stitch large datasets on low-memory machines, the algorithms benefit more from a sufficient amount of RAM than from a fast CPU.

Conclusions and outlook

The stitching quality and speed that was reached with XuvTools is significantly higher than those of other available tools. From our tests we conclude that a distinguishing feature of our software is that it works fully automatically, and does not require *a priori* information on the tile (stage) positions.

The POC we use for the automatic pre-alignment works robustly and (due to scaling) also very fast. With 16bit data as seen in Fig. 15, it benefits from the additional information that in contrast severely hinders humans when manually pre-aligning tiles. Our proposed interest-point detection based on the gradient magnitude successfully detects structures that can be correlated even under difficult circumstances. This is well demonstrated by the highly overexposed mouse hippocampal granule cell dataset, yielding a sufficiently good result (Fig. 13). The normalized cross-correlation (NCC) on the proposed small correlation windows gives strong peak results even with only fine fibers, and in the presence of bleaching.

Concerning the high difficulty when stitching noisy and very sparse neuronal datasets, we assume that every additional degree of freedom in the transformation parameters may cause a complete failure of the transformation parameter estimation. Due to the intentional limitation of our approach to the minimal set of necessary transformations, the demands to the imaging hardware are a little bit higher than those of other stitching approaches. However, for most biological applications the important feature is robustness and precision of the overall work flow. An investment into high-quality objectives therefore appears highly recommended compared to spending hours of man-power on manual stitching.

At the current state XuvTools work very smoothly and enable a fast work flow on a state of the art computer (see ‘Benchmark’ section, in ‘Results’).

Still, there are some possibilities to further optimize the XuvTools for productive use in the biomedical field. From the algorithmic side, the estimation of the tile positions should become less sensitive to spurious but high correlation peaks, that appear in rare cases, when no valuable information is available in the overlapping regions. In these cases, the estimation of absolute tile positions using singular value

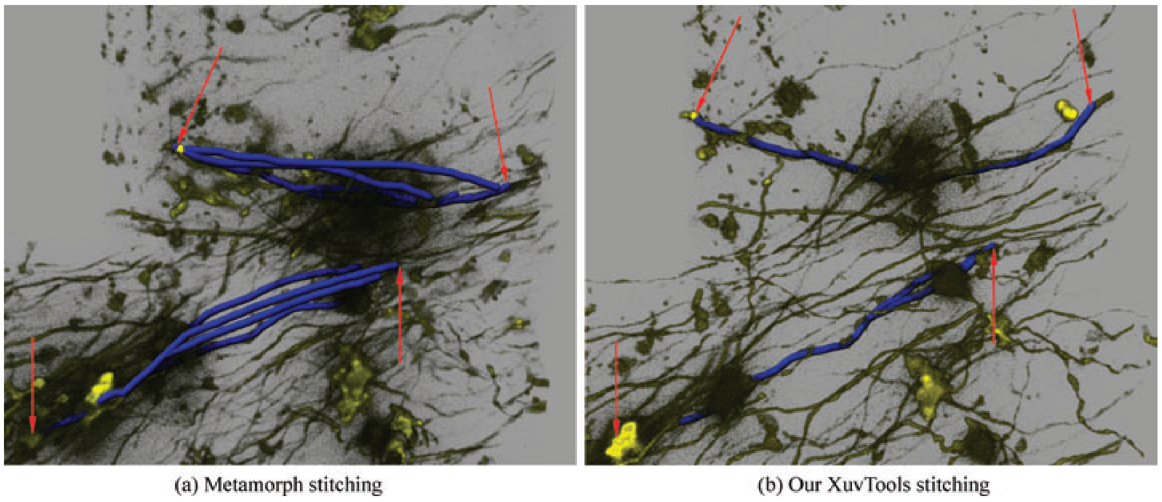


Fig. 20. Section of a 3D rendering of Fig. 19, with neuron tracing applied. (a) Erroneous duplicate tracings in the Metamorph stitching due to introduced ambiguities are clearly visible. (b) Tracing in the XuvTools stitching.

decomposition (SVD) is inaccurate for all tiles, and not just for the tile with the faulty correlation. An improved estimation is currently being worked on that optimizes all placements individually, thereby achieving a good overall positioning even in case of errors with the NCC.

For a better user experience with the XuvTools, in addition, we are currently working on the support of more file formats, and the addition of an alternative memory management that is optimized for lower memory usage at the cost of a slight runtime penalty. Both these features are currently being worked on.

Acknowledgement

We wish to thank the following people for their cooperation and for providing us with biomedical datasets: Csaba Dávid and Jochen Staiger, Institute of Anatomy and Cell Biology, Albert-Ludwigs-University Freiburg, Germany; Katja Wester

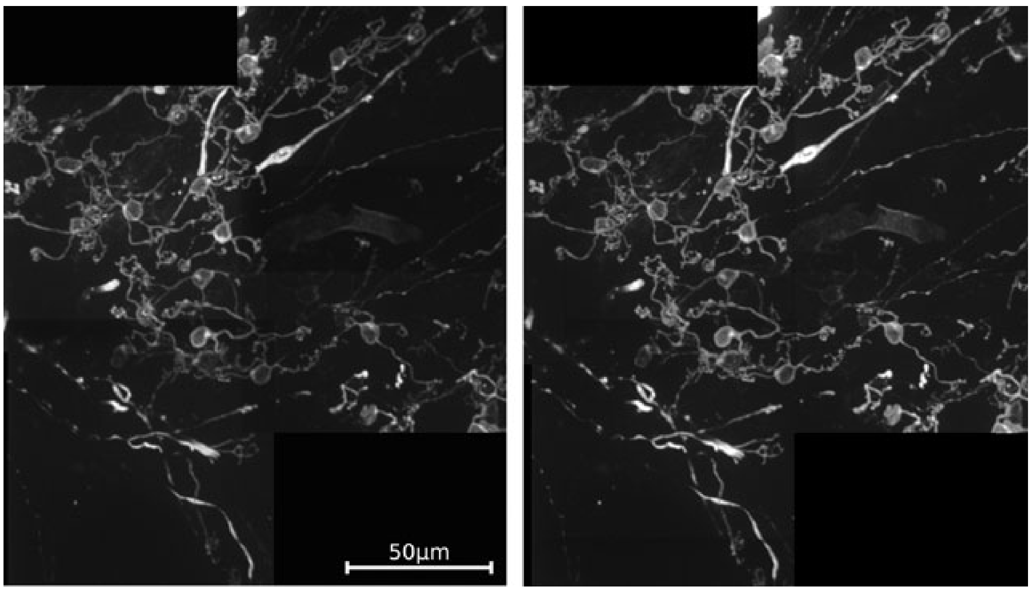


Fig. 21. *Ex vivo* image of the lobule III of the cerebellar cortex of a three month-old transgenic mouse expressing Thy-1 mGFP in a subset of neuronal cells. (a) The GlueMRC stitching is precise and correct in z-direction. Stitching of large datasets with GlueMRC requires patience and expertise from the user, as he has to label corresponding structures in tiles to initialize the automatic fine placement. (b) Our stitching provides a result very similar to GlueMRC, but was achieved without manual interaction, and thus in a fraction of the time (about 15 sec). Note: The brightness difference is due to gamma correction of the originally 16bit image mosaic, and was performed manually for better visibility in this presentation.

and Martin Hülskamp, Botanical Institute III, University of Cologne, Germany (their work was funded by the DFG Sonderforschungsbereich 572); Christina Eberle and Jakob Wolfart, Department of Neurosurgery, University Hospital Freiburg, Germany; Claudia Vittori and Nadine Gogolla from Group Pico Caroni, and Peter Meister from Group Susan Gasser, Friedrich Miescher Institute for Biomedical Research (Part of Novartis Research Foundation), Basel, Switzerland.

This work was supported by the DFG Sonderforschungsbereich 592 projects A3 (WD) and Z2 (RN, WD), and a Landesforschungsschwerpunktprogramm des Landes Baden-Württemberg (HB, WD, RN). The study was also supported by the Excellence Initiative of the German Federal and State Governments (EXC 294).

References

Al-Kofahi, O., Can, A., Lasek, S., Szarowski, D.H., Turner, J.N. & Roysam, B. (2003) Algorithms for accurate 3D registration of neuronal images acquired by confocal scanning laser microscopy. *J. Microsc.* **211**, 8–18.

Appleton, B., Bradley, A. & Wildermoth, M. (2005) Towards optimal image stitching for virtual microscopy. *Digital Image Computing: Techniques and Applications*, pp. 299–306. IEEE, Cairns.

Bajcsy, P., Lee, S.-C., Lin, A. & Folberg, R. (2006) Three-dimensional volume reconstruction of extracellular matrix proteins in uveal melanoma from fluorescent confocal laser scanning microscope images. *J. Microsc.* **221**, 30–45.

Becker, D.E., Ancin, H., Szarowski, D.H., Turner, J.N. & Roysam, B. (1996) Automated 3-D montage synthesis from laser-scanning confocal images: application to quantitative tissue-level cytological analysis. *Cytometry* **25**, 235–245.

Bujnak, M., Bujnak, M. & Sara, R. (2007) A robust graph-based method for the general correspondence problem demonstrated on image stitching. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, edited by R. Sara, 1–8.

Can, A., Can, A., Stewart, C., Roysam, B. & Tanenbaum, H. (2002) A feature-based, robust, hierarchical algorithm for registering pairs of images of the curved human retina. *IEEE Trans. Pattern Anal. Mach. Intell.* **24**, 347–364.

Capek, M. & Krekule, I. (1999) Alignment of adjacent picture frames captured by a clsm. *IEEE Trans. Inf. Technol. Biomed.* **3**, 119–124.

Chow, S. K., Hakozaki, H., Price, D.L., Maclean, N.A.B., Deerinck, T.J., Bouwer, J.C., Martone, M.E., Peltier, S.T. & Ellisman, M.H. (2006) Automated microscopy system for mosaic acquisition and processing. *J. Microsc.* **222**, 76–84.

Fischler, M.A. & Bolles, R.C. (1981) Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm ACM* **26**, 381–395.

Ito, K., Nakajima, H., Kobayashi, K., Aoki, T. & Higuchi, T. (2004) A fingerprint matching algorithm using phase-only correlation. *IEICE Trans. Fundam.* **E87-A**, 682–691.

Karen, P., Jirkovská, M., Tomori, Z., Demjénová, E., Janáček, J. & Kubínová, L. (2003) Three-dimensional computer reconstruction of large tissue volumes based on composing series of high-resolution confocal images by gluemrc and linkmrc software. *Microsc. Res. Tech.* **62**, 415–422.

Price, K. (2008) *Annotated Computer Vision Bibliography*, chap. 18.4.1 Mosaic Generation. URL: http://www.visionbib.com/bibliography/contents.html.

Ronneberger, O. (1998) *Messung aller drei Geschwindigkeitskomponenten mit Hilfe der 'Particle Image Velocimetry' mittels einer Kamera und zweier paralleler Lichtschnitte*. DLR-FB 98-40, ISSN 1434-8454. Deutsches Zentrum für Luft- und Raumfahrt.

Slamani, M.-A., Krol, A., Beaumont, J., Price, R.L., Coman, I.L. & Lipson, E.D. (2006) Application of phase correlation to the montage synthesis and three-dimensional reconstruction of large tissue volumes from confocal laser scanning microscopy. *Microsc. Microanal.* **02**, 106–112.

Sun, C., Beare, R., Hilsenstein, V. & Jackway, P. (2006) Mosaicing of microscope images with global geometric and radiometric corrections. *J. Microsc.* **224**, 158–165.

Thévenaz, P. & Unser, M. (2007) User-friendly semiautomated assembly of accurate image mosaics in microscopy. *Microsc. Res. Tech.* **70**, 135–146.

Zomet, A., Levin, A., Peleg, S. & Weiss, Y. (2006) Seamless image stitching by minimizing false edges. *IEEE Trans. Image Process.* **15**, 969–977.