Bioinformatics, 2019, 1–8
doi: 10.1093/bioinformatics/btz423
Advance Access Publication Date: 22 May 2019
Original Paper

Bioimage informatics

NRStitcher: non-rigid stitching of terapixel-scale volumetric images

Arttu Miettinen[1,2,*,†], **Ioannis Vogiatzis Oikonomidis**[2,3], **Anne Bonnin**[2] **and Marco Stampanoni**[2,4]

[1]Centre d'imagerie biomédicale, École polytechnique fédérale de Lausanne, 1015 Lausanne, Switzerland, [2]Swiss Light Source, Paul Scherrer Institute, 5234 Villigen, Switzerland, [3]Institute of Anatomy, University of Bern, 3012 Bern, Switzerland and [4]Institute for Biomedical Engineering, University and ETH Zürich, 8092 Zürich, Switzerland

*To whom correspondence should be addressed.
†Present address: Institute for Biomedical Engineering, University and ETH Zürich, 8092 Zürich, Switzerland

Associate Editor: Robert Murphy
Received on December 4, 2018; revised on April 9, 2019; editorial decision on May 13, 2019; accepted on May 17, 2019

Abstract

Summary: In modern microscopy, the field of view is often increased by obtaining an image mosaic, where multiple sub-images are taken side-by-side and combined post-acquisition. Mosaic imaging often leads to long imaging times that can increase the probability of sample deformation during the acquisition due to, e.g. changes in the environment, damage caused by the radiation used to probe the sample or biologically induced deterioration. Here we propose a technique, based on local phase correlation, to detect the deformations and construct an artifact-free image mosaic from deformed sub-images. The implementation of the method supports distributed computing and can be used to generate teravoxel-size mosaics. We demonstrate its capabilities by assembling a 5.6 teravoxel tomographic image mosaic of microvasculature in whole mouse brain. The method is compared to existing rigid stitching implementations designed for very large datasets, and observed to create artifact-free image mosaics in comparable runtime with the same hardware resources.

Availability and implementation: The stitching software and C++/Python source code are available at GitHub (https://github.com/arttumiettinen/pi2) along with an example dataset and user instructions.

Contact: arttu.miettinen@psi.ch

1 Introduction

In most commonly used forms of microscopy the routinely achievable resolution has been steadily increasing during the last years. Usually, the field of view of the microscope decreases with increasing resolution, often severely limiting the region of the sample that can be seen in one high-resolution image. In order to increase the size of the imaged region, automated acquisition of multiple images side-by-side, with small overlap, is often done. This procedure leads to the problem of stitching or mosaicing, i.e. connecting the sub-images taken side-by-side into one large image. Typically, the stitching algorithm finds optimal location for each sub-image based on image content in the overlapping regions. This procedure ensures that the final mosaic is free of geometrical artifacts (such as blurred or repeated details) in the overlapping regions, despite inaccuracies in the initial position estimates of the sub-images.

A particular type of microscopy where image mosaicing is often applied is X-ray microtomography (CT). It is a type of microscopy that results in a 3D image of the sample mathematically reconstructed from 2D X-ray projection images (Kak and Slaney, 1988). Recent developments in CT have enabled fast acquisition of high-resolution images, making creation of large 3D image mosaics possible in reasonable imaging time. For example, acquisition of a 10 × 10 × 10 mosaic using equipment described in Mokso *et al.* (2017) would be possible in <1 day with image resolution in the single-digit micrometer range. The size of a single sub-image would be near

(2000 voxels)^3, and the total data size would be ~15 TB, assuming 16-bit voxels. It is obvious that such amount of data cannot be easily stitched without an efficient workflow that is optimized for datasets that do not fit into the random access memory (RAM) of a typical workstation.

Many biologically interesting materials tend to show small changes and deformations in their local structure as a function of time, caused by processes such as heating or cooling, drying, chemical reactions and radiation damage. In principle these deformations could be avoided by keeping the sample in a stable environment, but that is often not possible, in particular in the case of CT where the sample is probed with X-radiation. Generally the magnitude and probability of the deformations increases with time, and therefore the total time used to image a sample must be limited such that the reconstructed image shows no artifacts due to the deformations. When acquiring a large image mosaic the total acquisition time of the whole mosaic should be set similarly. This severely limits the time available for acquiring large image mosaics, and in practice the total acquisition time is often set such that each individual sub-image is free of artifacts. The sub-images that are to be placed next to each other might be taken some time between them (e.g. the first image in the first row and the first image in the second row). Consequently, the sample may have accumulated deformations such that the overlapping regions of the two sub-images are not geometrically congruent, i.e. the image details do not overlap correctly when the images are overlaid. If not accounted for, stitching of such sub-images leads to stitching artifacts like discontinuous structures and blurred details at the boundaries of the sub-images.

In this work a method to stitch images with arbitrary small local deformations between them is presented. In the following, the term ‘non-rigid’ stitching is used to refer to such methods, and the term ‘rigid’ to denote methods or transformations that do not incorporate arbitrary local deformations. In contrast to many previous non-rigid stitching algorithms (Castanheira de Souza et al., 2012; Yan *et al.*, 2017; Yeung *et al.*, 2008), the proposed method is readily suitable for stitching both 2D and 3D image mosaics. It is based on computing the local deformation field between the overlapping regions of the sub-images non-iteratively using (local) phase correlation (Kuglin and Hines, 1975; Tajima *et al.*, 2012). The local deformations are post-processed in order to filter out cases where the initial computation failed. The subsequent results are then used to position and deform the sub-images optimally. This process is formulated as a simple non-iterative algorithm that is easy to parallelize and optimize for very large images (compare e.g. to Gao *et al.* 2019; Wachinger *et al.*, 2008; Yigitsoy *et al.*, 2015).

As a showcase, we assembled a tomographic image mosaic of the entire micro-vascular network in a whole mouse brain sample with 0.65 µm voxel size. The stitched mosaic, consisting of 5.6 teravoxels, is, to the best of our knowledge, one of the largest synchrotron-based tomographic microscopy datasets to date (compare to Vescovi *et al.*, 2018). Such images offer an unprecedented possibility to analyze biological systems in high detail. Previously, analysis methods for images of comparable size have been demonstrated, e.g. in materials science (Mattila *et al.*, 2016), albeit only for artificially generated data. Additionally, we show the efficiency of the stitching method on smaller datasets showing microstructure of mouse lung.

Finally, the stitching results and computation time are compared to those of TeraStitcher (Bria and Iannello, 2012) and BigStitcher (Hörl *et al.*, 2018). The proposed non-rigid method effectively eliminates geometrical stitching artifacts that the rigid stitching methods are unable to remove. A freely available implementation of the method supporting distributed processing in a computer cluster environment is given.

2 Algorithm

The proposed non-rigid stitching algorithm consists of three main parts (see Fig. 1):

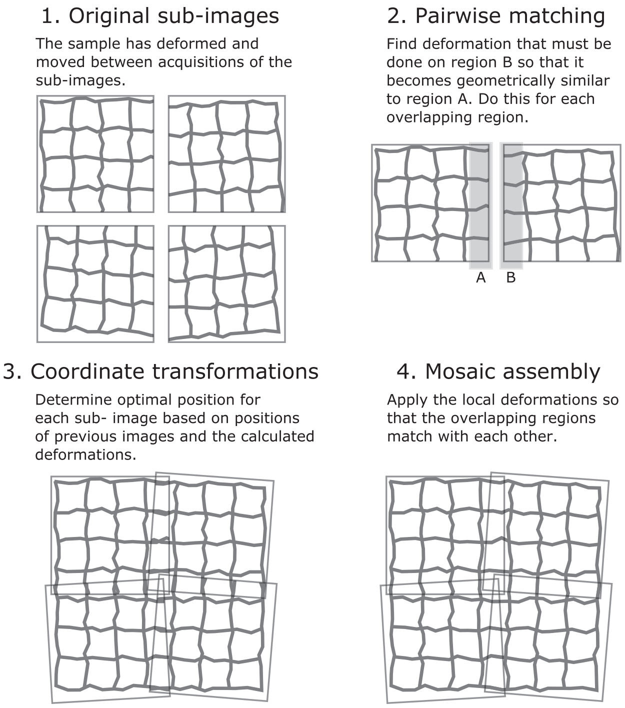
1. Pairwise matching, where corresponding points in each overlapping sub-image pair are determined.
2. Determination of coordinate transformation between stitched mosaic and each sub-image such that all the overlapping regions of the sub-images become geometrically congruent.
3. Assembly of the stitched mosaic.

In the following, each of the three parts is discussed separately.

2.1 Pairwise matching

The first step in the stitching process is pairwise matching. The goal of this step is to find corresponding locations in two partially overlapping sub-images. The correspondences are later used to deform the sub-images such that the overlapping region is geometrically congruent in both of them. In this step each pair of overlapping sub-images in each coordinate direction is considered separately. It is assumed that initial estimates of the relative positions (and overlaps) of the images are available from the stage position data read from the microscope.

Consider two overlapping images I_1 and I_2. The overlapping region between them is assumed to be rectangular in the coordinates of I_1. A rectangular grid of points with user-specified spacing in each coordinate direction is placed to the overlapping region in I_1 and denoted by $\vec{x}_i^1$. Initially, estimates of the corresponding locations in I_2 are determined from the positions of the two images and denoted by $\vec{x}_i^2$. Spacing between the points is chosen such that sample deformations can be approximated to be linear in between the points (see also Sections 3.2 and 4 for further discussion).


1. Original sub-images

The sample has deformed and moved between acquisitions of the sub-images.

2. Pairwise matching

Find deformation that must be done on region B so that it becomes geometrically similar to region A. Do this for each overlapping region.

3. Coordinate transformations

Determine optimal position for each sub- image based on positions of previous images and the calculated deformations.

4. Mosaic assembly

Apply the local deformations so that the overlapping regions match with each other.

Fig. 1. Schematic representation of the stitching process. The example is given in 2D for simplicity and clarity of presentation. The same principles apply for the 3D case. Notice that in Step 3 the overlapping regions do not fully match, and therefore assembling a mosaic without accounting for local deformations would introduce artifacts

The location of each $\vec{x}_i^2$ is refined by a multiresolution phase correlation process (Kuglin and Hines, 1975; Tajima *et al.*, 2012) on rectangular blocks of I_1 and I_2 around points $\vec{x}_i^1$ and the last estimate of $\vec{x}_i^2$, respectively. The blocks are downscaled by a factor of n, where typically $n = N$ in the first resolution step and $n = 1$ in the last resolution step, with N being the number of steps. The estimate of $\vec{x}_i^2$ at step n is given by

$$\vec{x}_{i,n}^2 = \vec{x}_{i,n-1}^2 + n \arg\max_{\vec{x}}(r_n(\vec{x})), \tag{1}$$

where the cross-correlation r_n is defined by

$$r_n = \mathcal{F}^{-1}\left\{\frac{G_1 \circ G_2{}^*}{|G_1 \circ G_2{}^*|}\right\}. \tag{2}$$

Here, $G_j = \mathcal{F}(S(g_j, n))$ is the Fourier transform of a downscaled block of image I_j. The operator $S(g_j, n)$ scales g_j by a factor of $1/n$, and g_j denotes a block of I_j around $\vec{x}_i^j$ or its previous estimate. The operator $\circ$ denotes elementwise product (Hadamard product), and $*$ denotes complex conjugation.

The size of the blocks I_1 and I_2 determines the maximal local shift that can be recognized. Consequently, the multiresolution process enables computationally efficient use of spatially large (but downscaled) blocks to find large shifts. The accuracy of the estimate is then improved by using smaller blocks taken around the last estimate of the shift.

Depending on the ratio between resolution and pixel size of the sub-images, it may be possible to stop refining $\vec{x}_i^2$ before reaching $n = 1$, or it may be necessary to continue the refining process to sub-pixel accuracy by using real values in range $0 < n < 1$. In all the cases sub-pixel accurate estimation of $\vec{x}_{i,n}^2$ may be used to improve accuracy (see e.g. Foroosh *et al.*, 2002; Feng *et al.*, 2012). Such algorithms are particularly favorable over direct upscaling, and enable use of smaller number of resolution steps in Equation (1) or skip the highest-resolution steps altogether. Selection of proper number of resolution steps depends on the content of the images and magnitudes of the shifts, but in many practical cases it seems to be enough to use one low-resolution step that captures large shifts followed by one high-resolution step for increased accuracy.

Near the edges of a sub-image some regions of the block g_j might fall outside the sub-image. In such situations the missing pixel values are replaced by the nearest valid value at the edge of the sub-image. This choice of boundary condition suppresses spurious maxima in the cross-correlation r_n caused by the edges of the blocks.

Before further processing, the set of correspondences consisting of point pairs $(\vec{x}_i^1, \vec{x}_i^2)$ is filtered to remove unrealistic values of $\vec{x}_i^2$. Such values can be encountered when the phase correlation process fails to find the correct position. This usually happens in regions where there are only few corresponding details visible in the sub-images. To that end, first define shift $\vec{s}_i$ as

$$\vec{s}_i = \vec{x}_i^2 - \vec{x}_i^1 - \langle \vec{x}_j^2 - \vec{x}_j^1 \rangle, \tag{3}$$

where the average is taken over all point pairs. All s_i whose some component s_{ik} satisfies

$$|s_{ik} - M(s_{ik}, r)| > T \tag{4}$$

are removed. The function $M(s_{ik}, r)$ is the median of all s_k in a sphere of radius r around s_{ik}, and T is a user-specified threshold value. The value of the parameter T can be chosen, e.g. based on the measure of the repeatability of the microscope stage as shown in Chalfoun et al. (2017). The removed $\vec{s}_i$ are replaced by values interpolated from the surrounding non-removed points using method introduced in Garcia (2010) and Wang *et al.* (2012). Finally, filtered values of $\vec{x}_i^2$ are calculated by inverting Equation (3).

2.2 Determination of stitched-to-local transformations

In this second step, the correspondences between overlapping sub-images determined with the phase correlation process (see Section 2.1) are used to find the coordinate transformation between the final stitched mosaic and each sub-image. In the following, we will call such coordinate transformation stitched-to-local transformation (StL-T).

In order to determine StL-Ts for all sub-images, the optimal rigid transformation is first found for each of them using the algorithm discussed in Hörl *et al.* (2018). To this end, let us denote a rigid transformation from the coordinates of the stitched mosaic to the coordinates of sub-image n by

$$\vec{x}^n = T_n(\vec{x}) = R_n \vec{x} + \vec{\Delta}_n, \tag{5}$$

where R_n is a rotation matrix and $\vec{\Delta}_n$ is a translation vector. The optimal T_n are determined by minimizing the value of the weight function

$$\sum_{(n,m)} \sum_{(\vec{x}^i, \vec{x}^j) \in C_{nm}} ||T_n{}^{-1}(\vec{x}^i) - T_m{}^{-1}(\vec{x}^j)||^2 \tag{6}$$

over T_n (practically, over rotation matrices R_n and translations Δ_n), keeping T_1 constant. The first sum is taken over all overlapping sub-image pairs (n, m), and the second sum is taken over C_{nm}, the set of all filtered corresponding points between images n and m as determined using the phase correlation process.

After the optimal T_n has been found for each sub-image, the StL-T for sub-image I_n is determined from the StL-Ts of those overlapping sub-images I_m whose StL-Ts have been determined so far. The processing order is discussed later. The transformation is defined differently in separate regions of the image I_n (see Fig. 2).

In the region where no I_m overlaps with I_n (region A in Fig. 2), the StL-T equals to the T_n.

In the region where one or more sub-images I_m overlap with I_n (regions B and C in Fig. 2), the StL-T is calculated using the filtered point correspondences. In this region

$$\vec{x}^n = \langle f(\vec{x}^m(\vec{x}), \{(\vec{x}_i^m, \vec{x}_i^n)\}) \rangle, \tag{7}$$

where the function f performs cubic interpolation of data points $\{(\vec{x}_i^m, \vec{x}_i^n)\}$ at $\vec{x}^m(\vec{x})$. The point $\vec{x}^m(\vec{x})$ is position $\vec{x}$ in the coordinates of the stitched mosaic converted to coordinates of sub-image I_m. The average is taken over all m, i.e. over all sub-images that overlap with sub-image n. In the example shown in Figure 2, this process results the StL-T being determined from point correspondences between I_n and one overlapping sub-image in regions B. In region C the StL-T will be average of the three transformations calculated from the point correspondences between I_n and the three overlapping sub-images.

The transformation defined above by Equations (5) and (7) might not be smooth at the boundary between the two regions (dashed region in Fig. 2). Such abrupt changes in the transformation may exhibit themselves as artificially stretched strips in the stitched mosaic. In order to mitigate such artifacts, the boundaries between the regions are smoothed by replacing a strip of values near the boundary (in region A) by values interpolated from the surroundings using Garcia, (2010) and Wang *et al.* (2012).

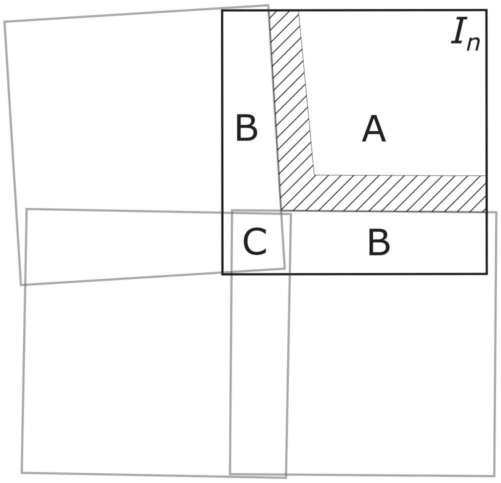


Fig. 2. Various regions encountered while determining the StL-T. The image I_n is depicted by a rectangle with black border, and images that overlap with it and whose StL-Ts have been determined are depicted by rectangles with gray border. In region A the StL-T of I_n equals its globally optimized rigid transformation. In regions B and C the transformation is calculated from the point correspondences determined in the pairwise matching phase. In the dashed region the two transformations are interpolated in order to ensure a smooth change between the regions

Determination of the StL-Ts requires that the sub-images are processed in a certain order. A consequence of the ordering is that the local deformations applied to a sub-image make it match to all the overlapping sub-images processed before it. The selection of the processing order can be made arbitrarily, and it does not affect the general shape of the stitched mosaic as that is controlled by the transformation optimization step described above (Equation 6). As the amount of local deformations may increase with increasing acquisition time (see Section 1), it seems natural to determine the StL-Ts of the sub-images in chronological order.

2.3 Mosaic assembly

The final stitched mosaic is assembled by transforming each sub-image using its StL-T, and adding the transformed sub-image to the stitched mosaic. In the regions where multiple sub-images overlap, the sub-images are blended linearly, i.e. the pixel values of a sub-image are weighted by the distance to the nearest edge of the sub-image. The total weight accumulated per pixel of the stitched mosaic is stored in a separate image, having the same size as the stitched mosaic. After all the sub-images have been processed, each pixel of the stitched mosaic is divided by the total weight accumulated to that pixel.

3 Implementation

The proposed method is particularly suitable for distributed processing on a computer cluster. All the pairwise matching tasks are independent and can be done in parallel. The determination of the StL-T for a sub-image is independent of other non-overlapping sub-images. The mosaic assembly can be done for a selected region of the final stitched mosaic per process as the sub-images and the corresponding StL-Ts are completely independent. Additionally, the tasks themselves contain parts that use the same input data but can be processed in parallel, e.g. the pairwise matching process for each grid point, and the determination of the StL-T of each pixel of the final mosaic.

Given the two levels of parallelism it seems to be natural to divide the independent tasks to multiple computing nodes with additional thread parallelism in each node. If multiple computing nodes are not available, the independent tasks can be executed sequentially on the local workstation. Notice that there is no need for communication between the parallel tasks.

The computationally intensive parts of the algorithm, i.e. phase correlation, determination of coordinate transformations and mosaic assembly, are implemented in C++ using OpenMP (Dagum and Menon, 1998) and FFTW libraries (Frigo and Johnson, 2005). A high-level Python driver routine calls the C++ submodules. The Python routine sets the initial positions of the sub-images based on the sample position data recorded during imaging, and separates individual jobs to run independently. The jobs are monitored and failed jobs are re-run (e.g. in the case of hardware problem on single compute node). The system reads the sub-images from 3D TIFF files, TIFF or PNG sequences, or flat binary files (RAW format, various pixel data type possibilities) and outputs a single file containing the stitched mosaic in RAW format. Various input arguments are stored in a separate human-readable text file. The implementation is kept as simple and lightweight as possible in order to enable extensions and improvements by third parties.

3.1 Test cases

Validation of the proposed method by comparing stitched image to ground truth is challenging as the method does not correct for deformations in the sub-images, but merely deforms the sub-images such that they correspond to each other. Therefore, even though the stitched image is sharp and details visible in the sub-images overlap correctly, it may not be similar to the ground truth. Instead of a direct comparison between images, a comparison of measurements made from ground truth and stitched images are made. To this end, images of size 1500×1500 pixels were generated by plotting random 2 pixels thick lines to an empty image until the fraction of the area covered by lines was ~0.2. This image was taken to be the ground truth. It was cut into tiles of size 400×400 pixels with 190 pixels overlap, and each of the tiles were individually deformed using random displacement field generated from OpenSimplex noise (see https://github.com/lmas/opensimplex). Average and maximal displacements were ~3.0 pixels and 6.7 pixels, respectively. Noise with SD of 8% of the full intensity range was added to all the images. The generated sub-images were stitched using rigid (translations and rotations only) stitching and the proposed non-rigid stitching methods. The lines were segmented from the ground truth and the stitched images by simple thresholding, and lengths of branches between intersecting lines were determined using skeleton analysis. Branches <4 pixels in length were pruned as those did not seem to correspond to the plotted lines. The average branch length distributions over 100 test images is shown in Figure 3, where it can be seen that non-rigid stitching resolves the original branch length distribution more accurately than rigid stitching. Kolmogorov–Smirnov test statistic between the true and the measured branch length probability distributions is 0.048 and 0.0067 for rigid and non-rigid stitching, respectively.

The practical performance of the proposed method is demonstrated by applying it to two CT volume image mosaics acquired at the TOMCAT beamline of the Swiss Light Source at Paul Scherrer Institute. The ‘brain’ dataset shows whole mouse brain and it consists of $9 \times 9 \times 15$ sub-images of 2048^3 voxels each. Individual

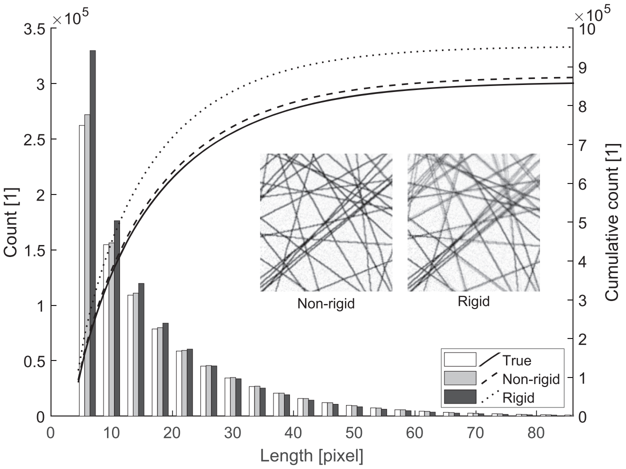


Fig. 3. Branch length distributions (bar plots, left axis) and the corresponding cumulative distributions (line plots, right axis) in artificially generated test images containing a network of lines. The distributions are normalized to the total number of detected branches in all the 100 test images of 1500^2 pixels each. The inset shows small parts of images stitched with non-rigid and rigid (translations + rotations) methods

sub-images overlap 30% of their diameter in directions perpendicular to the rotation axis, and ~10% in the direction parallel to the rotation axis. Each of the sub-images has been reconstructed from 1001 X-ray projection images with the GridRec algorithm (Marone and Stampanoni, 2012), applying Paganin phase retrieval method (Paganin *et al.*, 2002) before reconstruction. The projection images have been acquired with 20 keV monochromatic X-ray beam, 0.65 μm voxel size and 50 ms exposure time. The total acquisition time was ~57 h. The 'lung' dataset shows mouse lungs in fresh post-mortem state, and it consists of 2 × 2 sub-images, each having 2016^3 voxels reconstructed from 1500 projection images with similar procedure as the 'brain' dataset. Here, X-ray beam energy was 21 keV, voxel size was 1.1 μm, exposure time was 5 ms and Moosmann filtering was used instead of Paganin. The total image acquisition time was ~1 min.

Figure 4 shows small parts of the 'brain' and the 'lung' datasets, and a comparison with rigid (translations only) stitching calculated using TeraStitcher. In the 'brain' dataset, the rigidly stitched mosaics show the same details multiple times side-by-side in the regions where multiple sub-images overlap. The same artifact is visible also in the 'lung' dataset as blurred parenchyma (bright walls between airways). The artifacts have been completely eliminated in the results obtained with our proposed non-rigid stitching method. Notice that despite the artifacts, some parts of the overlapping regions in the rigidly stitched images are sharp and well-stitched. This indicates that the sub-images have been locally deformed between acquisitions and are thus impossible to stitch with strictly rigid deformations without producing artifacts. There is thus a definite need to use a non-rigid stitching method with these datasets. Finally, Figure 5 shows visualizations of the full 'brain' dataset highlighting the multiple length scales that can be probed simultaneously using mosaic imaging.

The magnitude of the local deformations can be analyzed in various ways using the StL-Ts. For example, the StL-Ts can be used to calculate the Biot finite strain tensor field, whose eigenanalysis gives the maximal local absolute normal strain $|\epsilon_{\max}|$ and maximal local shear strain $\gamma_{\max}$ (Lubliner, 2008). The distributions of $|\epsilon_{\max}|$ and $\gamma_{\max}$ are shown in Figure 6. Despite filtering of deformation fields as

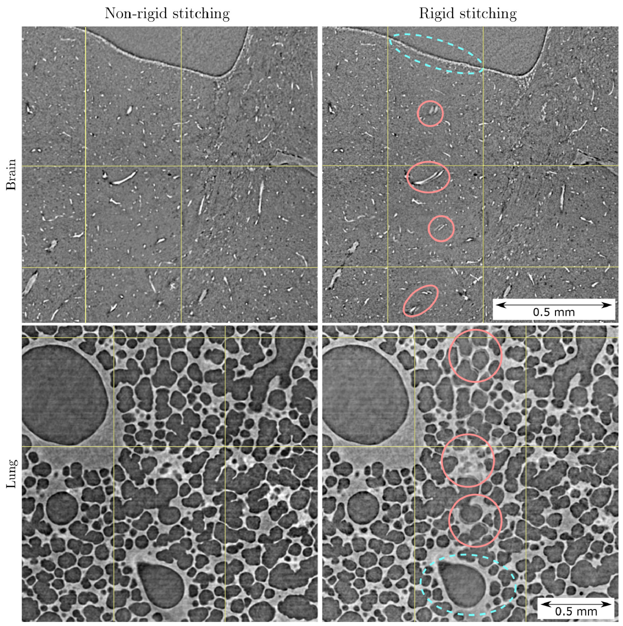


Fig. 4. Comparison of the proposed non-rigid stitching method to rigid (translation-only) stitching (Bria and Iannello, 2012). The top and bottom rows show small parts of slices through 3D CT image of mouse brain and mouse lung, respectively. The sub-images overlap between the yellow lines. The left and right columns show the results of the proposed non-rigid stitching method and rigid stitching made with TeraStitcher, respectively. Stitching artifacts are visible as doubled details and blur in the rigid stitching results in the regions where the sub-images overlap (right, between yellow lines), but not in the non-rigid stitching results (left). Some artifacts have been marked with red ellipses. Regions marked with blue dashed ellipses are well-stitched, indicating that no rigid transformation will result in good stitching everywhere in the slice. Contrast of the images has been scaled to show the details well (Color version of this figure is available at *Bioinformatics* online.)

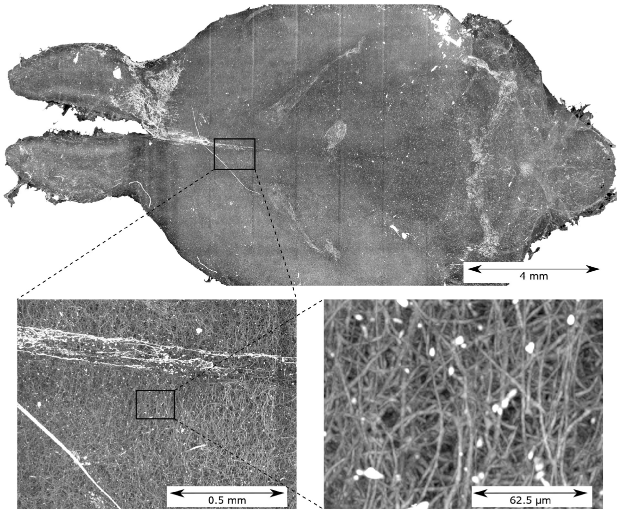


Fig. 5. Maximum intensity projection of a stitched 9 × 9 × 15 CT mosaic showing blood vessels in a whole mouse brain (top), and details of a selected location at two different scales (bottom)

described in Section 2.1, the maximal strains seem to often occur in regions outside of the sample or on the edge of it. The maximal strains inside the sample are better depicted by the 99th percentiles of the strain distributions that are 0.16 and 0.25 for $|\epsilon_{\max}|$ and $\gamma_{\max}$, respectively. These values indicate that any short linear structure in the stitched image is at most 16% longer or shorter than in any individual sub-image, and that any small cube in a sub-image is sheared at most 25% of its edge length in the stitched image. The average values of $|\epsilon_{\max}|$ and $\gamma_{\max}$ are 0.018 and 0.041, respectively, indicating that the local deformations are generally relatively small. Furthermore, Figure 6 shows that almost one half of the total volume of the sub-images has approximately zero strain, indicating the volume that is not locally deformed or is not overlapping with any other sub-image.

3.2 Runtime and memory requirements

The computational complexity of the pairwise matching process is $O(nM\log(M))$, where n is the number of grid points and M is the count of pixels in the blocks used in the phase correlation algorithm. The complexity of the StL-T calculation is $O(N\log(N))$, where N is the number of pixels in a sub-image, due to Discrete Cosine Transforms taken in the interpolation process. The mosaic assembly is an $O(N)$ process.

Practically, the processing speed is limited also by the read and write speeds of the hard disk system. Therefore, the total amount of disk operations is minimized by reading the sub-images only two times and by avoiding random access to the image files. The pairwise matching process requires reading of the overlapping regions of all the sub-images from the disk, and the mosaic assembly process requires reading of all the sub-images from the disk and writing the whole stitched mosaic to the disk. In particular, each pairwise matching job computes point correspondences for single overlapping image pair and thus reads image data from two images, from the overlapping region only. Each of the mosaic assembly jobs corresponds to a block of the final stitched mosaic. These jobs read the images that contribute to the output block one-by-one, and write the block of the output file they have constructed. The determination of the StL-Ts does not include reading of the input images.

The most RAM-intensive part of the stitching process is the final assembly of the stitched image, where one must have at least a single sub-image, a part of the stitched mosaic, and the corresponding part of the weight image in RAM at the same time. In the current implementation, the memory requirement can be tuned by setting the size of the part of the stitched mosaic that is generated by a single process. Minimal practical RAM requirement is thus a few times the size of one sub-image.

The runtime of the proposed method was compared to TeraStitcher and BigStitcher on a computer with two Intel Xeon E5-2690v3 processors (total of 24 physical cores) and 180 GiB of RAM available to the programs. The proposed method was also tested on a cluster of 10 computing nodes with the same specifications. Runtimes of each of the three programs were measured for a 9 × 9 × 1 block of the ‘brain’ dataset, scaled to four different sizes by a factor $s = 1$, $3/4$, $1/2$ or $1/4$, leading to sub-image sizes of 2048^3, 1536^3, 1024^3 and 512^3, respectively. The stitching settings were set to realistic values that a normal user would have chosen to stitch the images. In particular, for the proposed method, the grid spacing was set to $s \times 60$ pixels, and block size to $s \times 120$ pixels $+ 1$ pixels. Single resolution step was sufficient in the phase correlation, with downscaling factor equal to $4s$. Filtering parameter T was set to $s \times 12$ pixels. TeraStitcher was run with the default settings, outputting 2D TIFF slices. For BigStitcher, the sub-image data were first converted to HDF5 format including a pre-computed downsampled version with the same scaling factor that was used in the proposed method, except for image with $s = 3/4$ where downsampling of 4 was used instead of 3. The stitching was run using the default settings, without the Iterative Closest Point (ICP) option, in ‘virtual’ mode. The stitched image was saved as uncompressed TIFF slices. Other settings were left to the default values.

The measured runtimes shown in Figure 7 indicate that the runtimes of all the methods scale essentially linearly with the number of pixels. Notice also that in this benchmark the number of overlapping pixels increases linearly with the total number of pixels. Additionally it should be noted that the experimental runtimes and scaling behavior may vary considerably with factors such as stitching settings, disk system speed or the data format used to store the sub-images. In particular, BigStitcher runtimes would be shorter if the sub-images were saved to suitable HDF5 file(s) so that no conversion would be required, and the runtimes of the proposed method would be significantly longer if larger block size or smaller grid step was used.

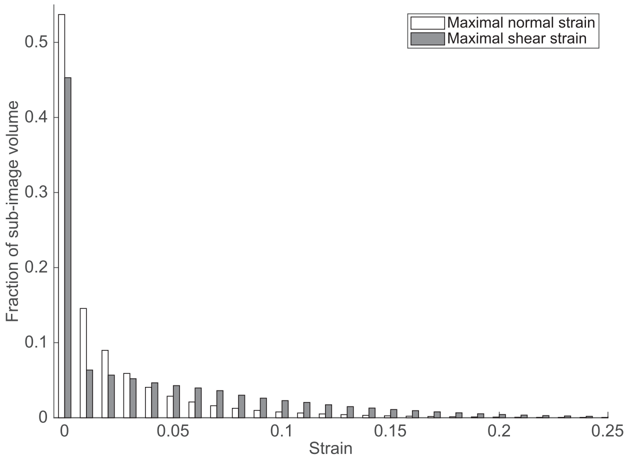


Fig. 6. Distribution of the local maximal absolute normal strain $|\epsilon_{\max}|$ and the local maximal shear strain $\gamma_{\max}$ between sub-images in the ‘brain’ dataset. Zero strain corresponds to regions with rigid deformations. Average values of $|\epsilon_{\max}|$ and $\gamma_{\max}$ are 0.018 and 0.041, respectively. The 99th percentiles are 0.16 and 0.25, giving estimates of the maximal normal and shear strains in the sample

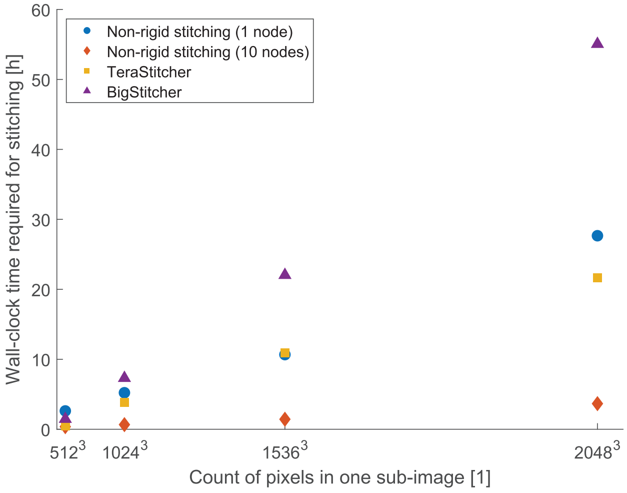


Fig. 7. Comparison of the runtime of the proposed method to those of TeraStitcher and BigStitcher for images of different size. The stitched mosaic consists of 9 × 9 × 1 sub-images. Results for TeraStitcher and BigStitcher are shown only for a single compute node as they do not support distributed processing

Figure 7 also shows that by using a 10-node cluster for the computations, the total computing time required by the proposed method can be reduced by a factor of 7–8. However, the maximum count of compute nodes that the proposed method can take advantage of is limited in the present implementation. In the pairwise matching phase one job is generated for each overlapping image pair, and thus the number of overlaps limits the count of jobs. When determining the StL-Ts, one job is generated for each sub-image. In the mosaic assembly phase the count of jobs is determined by the amount of RAM available in the compute nodes. Practically, in the case of the full 9 × 9 × 15 sub-image 'brain' mosaic there are several thousands of pairwise matching jobs, 1215 StL-T generation jobs and ∼500 mosaic assembly jobs. The stitching takes ∼51 h with 10 compute nodes. Most of the jobs are independent so the total runtime should decrease considerably if more than 10 compute nodes are available.

4 Discussion

A parallel non-rigid stitching method for volume images was proposed. The method can account for translations, small rotations and small local deformations between the sub-images. It is based on estimation of the deformations between the sub-images using local phase correlation. The deformations are used to calculate position and orientation of each sub-image in the stitched mosaic, and to deform the sub-images such that they are geometrically congruent in the regions where they overlap. The proposed method was tested on CT images and it improved the stitching results over traditional rigid stitching methods. The runtime of the method is not excessively high compared to existing rigid stitching methods run on datasets of the same size, while it provides better image quality. The RAM resources required by the method seem to be reasonable when compared to existing rigid stitching software.

The proposed method does not generally correct deformations of the sample but only deforms sub-images more so that the resulting stitched mosaic is geometrically consistent, i.e. structures are continuous over the boundaries of the sub-images. There is no guarantee that the mosaic would be geometrically correct, e.g. parallel lines in the sub-images might not be parallel everywhere in the stitched mosaic. These distortions originate from the local deformations applied to the sub-images and their magnitude is bounded by the magnitude of deformation of the sample occurred during the imaging process. Furthermore, the magnitude can be quantified from the StL-Ts. In practice the geometrical distortions seem to be small and the added geometrical consistency in the overlapping regions outweighs the distortions in most cases. For example, analysis of the vascular network in the 'brain' test image would be impossible if the thin blood vessels were discontinuous on the boundaries of the sub-images as is the case for a mosaic stitched with rigid stitching methods. A side effect of the possible geometrical distortions caused by the proposed method is that a quantitative measurement of the accuracy of the stitching is hard, as an artifact-free mosaic might be geometrically different from a ground-truth image.

A limitation of the proposed method and its current implementation is that it requires larger overlap between the sub-images than rigid stitching. If the overlap is too small, there may not be enough corresponding image details for accurate phase correlation. Practical observations of the performance of the method indicate that

- the block size in the pairwise matching step (see Section 2.1) must be set such that most of the blocks contain multiple details of interest,
- the spacing between the calculation grid points should be determined so that deformations can be approximated to be linear in between the grid points and that there are many grid points in all non-background regions and
- overlap between images must correspond to at least 3 calculation grid points.

It may not be possible to fulfill all the three requirements if some of the sub-images contain only small parts of the sample or if the deformation gradient is very large in some regions. In these cases the stitching may fail and lead to local artifacts. The 30% overlap between sub-images used in the examples in Figure 4 corresponds to ∼10 calculation grid points and seems to be sufficient for bulk materials containing large number of features.

This work was focused on correcting the geometrical mismatch between deformed sub-images and thus radiometric alignment (i.e. contrast matching) has been left as a topic for future work. Instead of the simple linear blending used in this work, methods such as graph cuts based 'puzzling' proposed in Oikonomidis *et al.* (2016) could potentially improve the quality of the mosaic even further.

The programs and source code used to generate the results are freely available in GitHub and are licensed under the GNU General Public License. The source code has been compiled and tested in Windows 7, Windows 10 and CentOS Linux 7 operating systems using MSVC 19.12, MSVC 19.16 and GCC 7.3.0 compilers, respectively. Python version 3.6 is required. User instructions and a small test dataset are also provided in the GitHub repository.

Acknowledgements

The authors would like to acknowledge Alessandra Patera for designing a sample preparation and imaging protocol used in acquisition of the 'brain' dataset, and members of the TOMCAT research group and Jussi Virkajärvi for fruitful discussions during the development of the stitching method.

Funding

This work was supported by Swiss National Science Foundation grants [CR23I2-135550, 310030-153468].

Conflict of Interest: none declared.

References

Bria,A. and Iannello,G. (2012) TeraStitcher—a tool for fast automatic 3D-stitching of teravoxel-sized microscopy images. *BMC Bioinformatics*, **13**, 316.

Castanheira de Souza,R.H. *et al.* (2012) Real-time image mosaicing using non-rigid registration. In: Yo-Sung,H. (ed.), *Advances in Image and Video Technology, Proceedings of the Pacific-Rim Symposium on Image and Video Technology. Lecture Notes in Computer Science*. Vol. 7087. Springer-Verlag, Berlin, Heidelberg, Germany.

Chalfoun,J. *et al.* (2017) MIST: accurate and scalable microscopy image stitching tool with stage modeling and error minimization. *Sci. Rep.*, **7**, 4988.

Dagum,L. and Menon,R. (1998) OpenMP: an industry standard API for shared-memory programming. *Comput. Sci. Eng., IEEE*, **5**, 46–55.

Feng,S. *et al.* (2012) A subpixel registration algorithm for low PSNR images. In: *IEEE Fifth International Conference on Advanced Computational Intelligence (ICACI) Nanjing, Jiangsu, China.*

Foroosh,H. *et al.* (2002) Extension of phase correlation to subpixel registration. *IEEE Trans. Image Process.*, **11**, 188–200.

Frigo,M. and Johnson,S.G. (2005) The design and implementation of FFTW3. In: *Proc. IEEE. Special Issue on "Program Generation, Optimization, and Platform Adaptation"*, 93, pp. 216–231.

Gao,R. *et al.* (2019) Cortical column and whole-brain imaging with molecular contrast and nanoscale resolution. *Science*, **363**, eaau8302.

Garcia,D. (2010) Robust smoothing of gridded data in one and higher dimensions with missing values. *Comput. Stat. Data Anal.*, **54**, 1167–1178.

Hörl,D. *et al.* (2018) BigStitcher: reconstructing high-resolution image datasets of cleared and expanded samples. https://www.biorxiv.org/content/10.1101/343954v1.

Kak,A.C. and Slaney,M. (1988) *Principles of Computerized Tomographic Imaging*. IEEE Press, New York, USA.

Kuglin,C.D. and Hines,D.C. (1975) The phase correlation image alignment method. In: *Proceeding of IEEE International Conference on Cybernetics and Society, New York, USA*. IEEE.

Lubliner,J (2008). *Plasticity Theory*. Dover Publications, Mineola, New York, USA.

Marone,F. and Stampanoni,M. (2012) Regridding reconstruction algorithm for real-time tomographic imaging. *J. Synchrotron Radiat.*, **19**, 1029–1037.

Mattila,K. *et al.* (2016) A prospect for computing in porous materials research: very large fluid flow simulations. *J. Comput. Sci.*, **12**, 62–76.

Mokso,R. *et al.* (2017) GigaFRoST: the gigabit fast readout system for tomography. *J. Synchrotron Radiat.*, **24**, 1250–1259.

Oikonomidis,I.V. *et al.* (2016) Imaging samples larger than the field of view: the SLS experience. In: *Proceedings of X-ray microscopy conference 2016, Oxford, UK Institute of Physics*.

Paganin,D. *et al.* (2002) Simultaneous phase and amplitude extraction from a single defocused image of a homogeneous object. *J. Microsc.*, **206**, 33–40.

Tajima,Y. *et al.* (2012) A non-rigid registration method for medical volume data using 3D phase-only correlation. In: *Proceedings of the 21st International Conference on Pattern Recognition (ICPR 2012) Tsukuba, Japan*. International Association for Pattern Recognition.

Vescovi,R. *et al.* (2018) Tomosaic: efficient acquisition and reconstruction of teravoxel tomography data using limited-size synchrotron X-ray beams. *J. Synchrotron Radiat.*, **25**, 1478–1489.

Wachinger,C. *et al.* (2008) Deformable mosaicing for whole-body MRI. In: Metaxas,D. *et al.*, (eds.), *Proceedings of Medical Image Computing and Computer-Assisted Intervention. Lecture Notes in Computer Science*. Vol. 5242. Springer, Berlin, Heidelberg.

Wang,G. *et al.* (2012) A three-dimensional gap filling method for large geophysical datasets: application to global satellite soil moisture observations. *Environ. Model. Softw.*, **30**, 139–142.

Yan,W. *et al.* (2017) Robust multi-homography method for image stitching under large viewpoint changes. *Int. J. Hybrid Inf. Technol.*, **10**, 1.

Yeung,S.-K. *et al.* (2008) Enforcing stochastic inverse consistency in non-rigid image registration and matching. In: *Proceedings of IEEE Conference on Computer Vision and Pattern RecognitionAnchorage, Alaska, USA*. IEEE.

Yigitsoy,M. *et al.* (2015) Structure propagation for deformable image stitching. In: *Proceedings of IEEE 12th International Symposium on Biomedical Imaging, New York, USA*. IEEE.