Enabling Manual Intervention for Otherwise Automated Registration of Large Image Series

Roman Grothausmann[1,2](✉), Dženan Zukić[3], Matt McCormick[3], Christian Mühlfeld[1,2], and Lars Knudsen[1,2]

[1] Institute of Functional and Applied Anatomy,
Hannover Medical School, Hannover, Germany
grothausmann.roman@mh-hannover.de
[2] Biomedical Research in Endstage and Obstructive Lung Disease Hannover
(BREATH), Hannover, Germany
[3] Kitware, Inc., Carrboro, NC, USA

Abstract. Aligning thousands of images from serial imaging techniques can be a cumbersome task. Methods ([2,11,21]) and programs for automation exist (e.g. [1,4,10]) but often need case-specific tuning of many meta-parameters (e.g. mask, pyramid-scales, denoise, transform-type, method/metric, optimizer and its parameters). Other programs, that apparently only depend on a few parameter often just hide many of the remaining ones (initialized with default values), often cannot handle challenging cases satisfactorily.

Instead of spending much time on the search for suitable meta-parameters that yield a usable result for the complete image series, the described approach allows to intervene by manually aligning problematic image pairs. The manually found transform is then used by the automatic alignment as an initial transformation that is then optimized as in the pure automatic case. Therefore the manual alignment does not have to be very precise. This way the worst case time consumption is limited and can be estimated (manual alignment of the whole series) in contrast to tuning of meta-parameters of pure auto-alignment of complete series which can hardly be guessed.

1 Introduction

The general approach to reconstruct 3D by 2D serial sections (also termed array tomography) is long known and can be applied with various imaging techniques [2,7,22,30]. This method has the common drawback that the images of the serial sections need to be aligned to the image of the adjacent slice. While this can be done manually with various programs (e.g. midas of IMOD, Fiji/ImageJ, VV, Gimp, PhotoShop), this can be very tedious labour. Although visual inspection seems easy, it often is hard to decide which transform is the “best”, one reason being the fact that adjacent images in general contain similar but not equal

Ž. Špiclin et al. (Eds.): WBIR 2020, LNCS 12120, pp. 23–33, 2020.
https://doi.org/10.1007/978-3-030-50120-4_3

content due to the structure change in the 3rd dimension. This becomes of particular importance when employing registration allowing local deformations, because the natural 3D structure change is not meant to be corrected by local deformation.

Therefore, various procedures for digital automatic alignment have been investigated, which in general are based on finding a transformation that optimizes a metric (a well defined quantification in contrast to visually "best"). Many types of transformations, metrics and optimizers have been developed of which specific ones need to be chosen depending on the given data and desired results. Apart from the parameters of the transformation that get optimized during the processing, parameters of the chosen optimizer, metric and general ones such as denoising, size/shape of a mask and pyramid resolutions need to be set before the processing can start [4,10,32]. These parameters are referred to as meta-parameters and need to be tuned with expert knowledge in order to get an acceptable results for as many consecutive images as possible. The more serial sections the image series contains, the more difficult and time consuming this task can become. Experience shows, that for a series of a few hundred up to a few thousand realistic (i.e. non-ideal) images, the finding of suitable meta-parameters can take a few weeks, without a guarantee to succeed at all.

In order to have a better guarantee to succeed in practice, the procedure described in this paper limits the time consumption to that needed for a pure manual alignment of the whole series, while trying to use automation as much as possible.

2 Method

First Elastix [10], later SimpleElastix [16] was chosen as the framework that provides the means for automated registration. In general other implementation could be chosen, however Elastix (based on ITK [8]) already accepts initial transformations. Even if an initial transformation is provided by manual alignment, it can still enter the automatic optimization and therefore get improved quantitatively as in the default case of pure automated optimization. In other words, if the automated optimization gets trapped in a local optimum i.e. fails to find the global optimum, a manual initial transform provides a different start point for the optimization such that the global optimum is reached.

The work presented here is based on three distinct pieces of software:

1. The Python-implemented registration program `recRegStack.py` which employs SimpleElastix.[1]
2. Extra programs and commands needed to convert gigapixel slice scans from a Carl Zeiss slide scanner (in CZI fromat) to an image series usable by `recRegStack.py`.

[1] http://github.com/romangrothausmann/elastix_scripts/.

3. A build and invocation system to apply these to a full-size image series, with adjustments needed for the specific data at hand, using gnuplot [31] and GNU Parallel [29].[2]

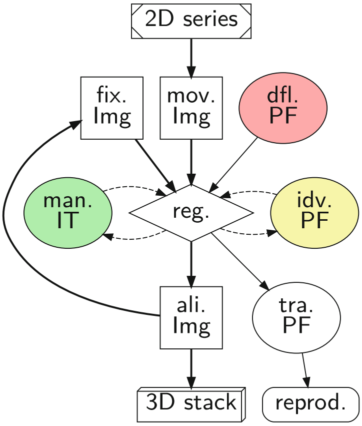

Fig. 1. Processing dependencies
Schematic flow graph to visualize the dependencies involved in the iterative process. Images are represented by squares, text-files by ovals. The parameters used by default (dfl. PF, dPF) during the registration (reg.) need to be tuned for the integral image series (2D series) in order to reduce the need for manual interactions. The first image at the start is copied unchanged. The last aligned image (ali. Img) is used as fixed image (fix. Img, fI) for registering the next image from the series (mov. Img, mI). In the case that the default parameters do not yield an acceptable result for an individual image pair, it is possible to supply a manual initial transform (man. iT, mIT) and/or provide individual registration parameters (idv. PF, iPF). Tuning the default parameters (dPF) is the most difficult (time consuming, red) task, adjusting some individual parameters less problematic (iPF, yellow) while creating a manual initial transform (mIT, green) with e.g. `midas` is easiest. In case some images need to be re-scanned (due to scan-artefacts, defocus, etc), the transform parameter files (tra. PF, tPF) can be used to register the new image exactly the same way (reprod.) or the registration process can be re-initiated to make use of the improved image quality.

`recRegStack.py` takes an Elastix/ITK parameter file (containing the definition of various meta-parameters) into account, which allows changing the default values used by SimpleElastix. The last transformed image is used to register the following one, see Fig. 1.

For the proof of principle, `midas` of the IMOD package [19] was chosen for manual alignment due to its superior precision and interaction possibilities. Manually created initial transform files (mITs, Fig. 1) will then be taken into account

[2] http://github.com/romangrothausmann/CZIto3D.

by `recRegStack.py` when continuing the automated alignment. In addition, it is possible to adjust the meta-parameters for individual image pairs (iPF, Fig. 1) in case the mIT together with the global defaults (dPF, Fig. 1) do not lead to a satisfactory result.[3] This can happen for example if the fixed image (fI) and the moving image (mI) come from different section bands, possibly differing significantly in focus quality.

3 Application and Results

The described approach was applied to an image series of about 2600 histological serial section of lung tissue (rat, details can be found in [27]), referred to as K2-dataset. An EM UC7 microtome (Leica, Germany) was used to cut semi-thin sections with a thickness of 1 µm connected to bands of about 1 to 20 sections. These bands were placed on 177 glass slides (where possible as a single row)

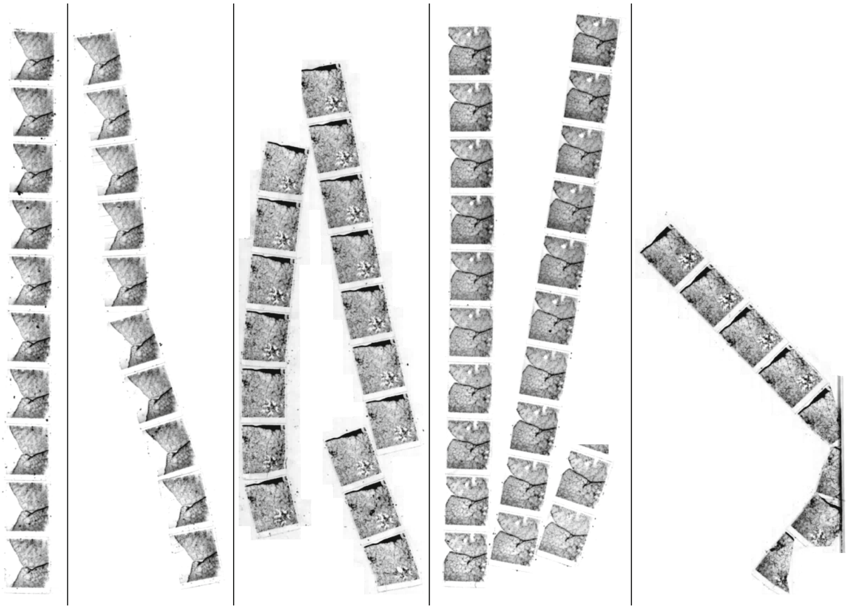

Fig. 2. Bands of serial sections of stained lung tissue on glass slides
Some exemplary thumbnails of slide scans with bands of serial sections of lung tissue stained on glass slides. Ranging from good (left, one band well aligned and no significant staining variations) to bad (right, broken bands with unobvious order, staining variations and slice loss due to folds and extending slide border). Lines indicate different glass slide.

[3] The iPF also allows to suppress further auto adjustments of an mIT in case the global optimum does not represent the correct transformation, which can happen for very destorted slices with repetitive, similar structures.

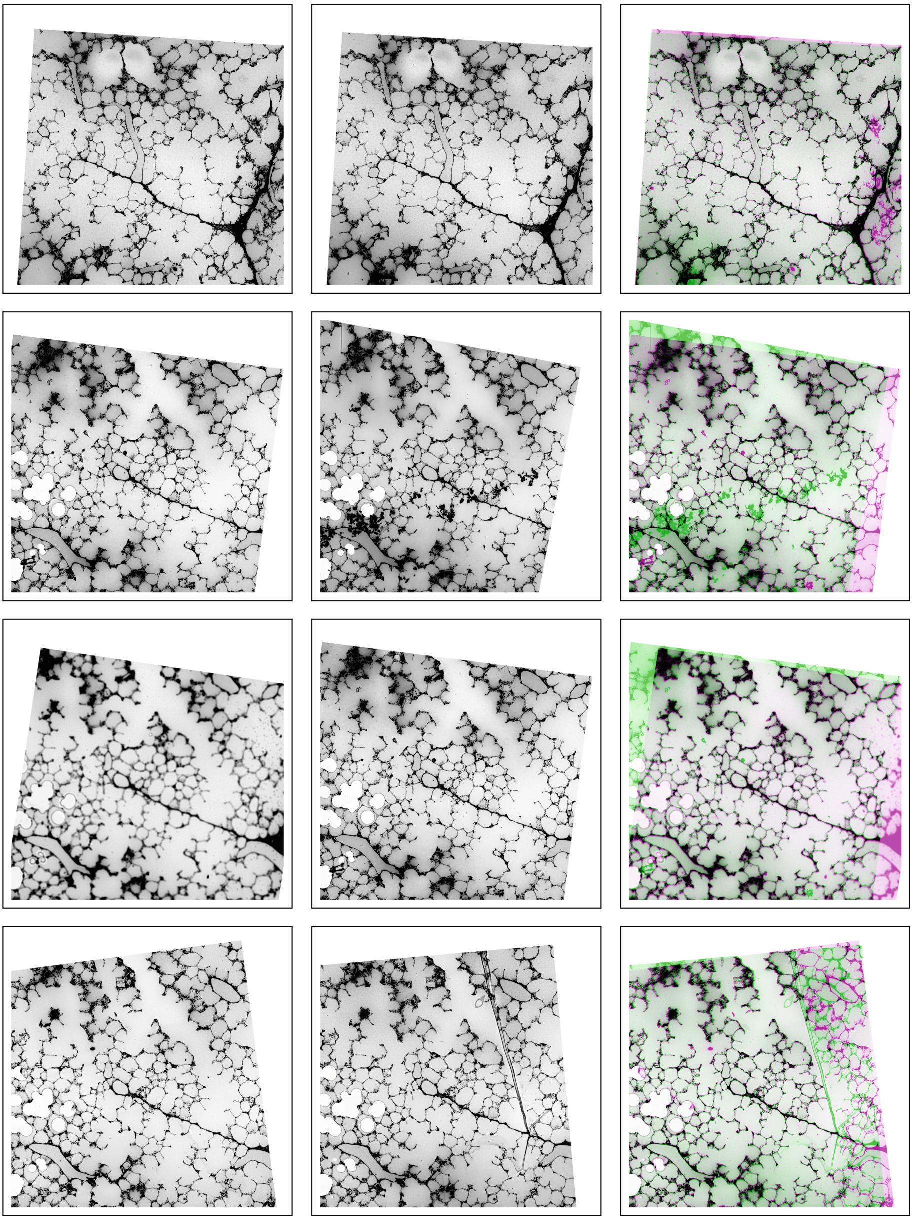

Fig. 3. Exemplary image pairs
Left column: fixed image, middle column: moving image, right column: Magenta-Green overlay of the image pair (similar to midas). Image pair rows:
1. ideal (no mIT or iPF needed).
2. dirt (mIT but no iPF needed).
3. defocus (no mIT or iPF needed).
4. folds (no mIT or iPF needed).
Metric for 3 without mIT is 34974 and with mIT 20214. (Color figure online)

see Fig. 2. After staining with toluidine blue, the slides were digitalized at a magnification of 10X by an AxioScan Z1 (Zeiss, Germany) using a single-channel fluorescence camera with a very low transmission light in order to get greyscale images (in CZI format) with a dynamic range above 8-bits.

The build and invocation system for this image series can be found in http://gitlab.com/romangrothausmann/K2_fibrosis/. This git repository holds references to the raw-data (CZIs) in an annex (https://git-annex.branchable.com/), imports http://github.com/romangrothausmann/CZIto3D as a subtree for local adjustments as needed for the specific data and serves as processing protocol. `recRegStack.py` from http://github.com/romangrothausmann/elastix_scripts/ is invoked via a docker-image (http://www.docker.com/) containing all the needed libraries to reproducibly register the images. There is a short (downscaled) image series for testing in tests/recRegStack/.

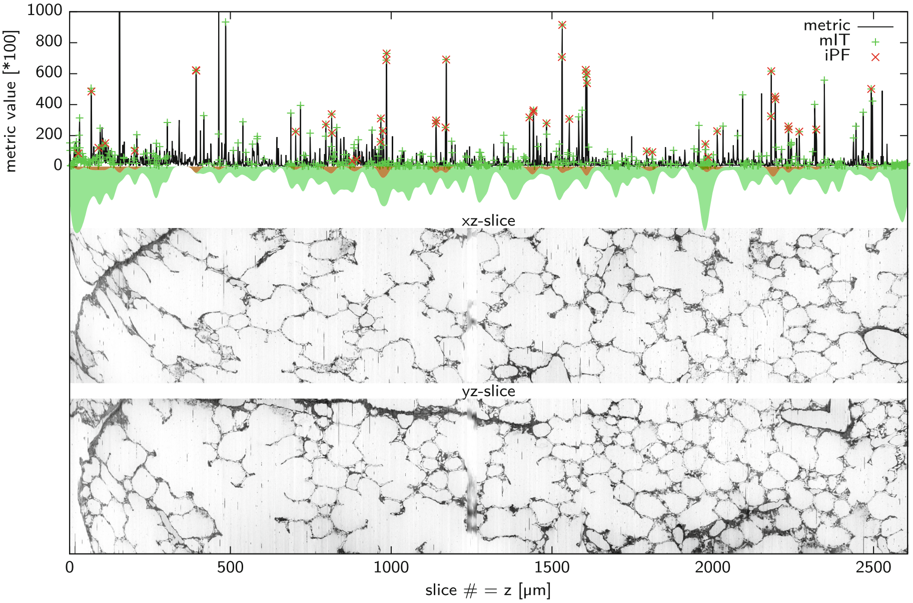

Fig. 4. Plot of metric values with markers for mIT and iPF
The point densities of mITs and iPFs are visualized with kernel density plots on the negative y-axis (unrelated to metric value, $\sigma = 10$). Mean of metric: ≈ 4100, Std. Dev.: ≈ 9300, some values are outside of the plot range, iPFs (52) are needed less often then mITs (621), mostly in cases of high metric values. The largest interval without any manual intervention (no mITs) is from 305 to 393 even though the default parameter file (dPF) was tuned at different locations (e.g. slice 929, 1266 and 1379). The centre xz- and yz-slice of the result stack (as shown in Fig. 5) are plotted for comparison. Distortions due to alignment drift can be seen, especially in the xz-slice up to slice 500.

Fiji [6] was used to roughly mark the centre of each section in thumbnail images of the gigapixel scans (czi2stack/Makefile). These centres were then used to automatically extract the region of each slice as its own image (3000 x 3000 pixel, PNG) with bfconvert [23] (czi2stack/Makefile). In case of broken bands, the ordering implied by the centre marks had to be adjusted to match the physical order (czi2stack/slides/slideOrder.lst, discrepancies often only visible after a registration). This order was then used to register the consecutive slices (czi2stack/Makefile). The mask for registration and the default parameter file (dPF) were adjusted to fit the K2-dataset, applying rigid registration using a "MultiResolutionRegistration" with "AdvancedMeanSquares" metric and "AdaptiveStochasticGradientDescent" as optimizer, see czi2stack/parameterFile.txt for details. Still, 621 mITs (in average every 4th image) and 52 iPFs (in average every 50th image) were needed to align all 2607 images, see Fig. 4. Some exemplary image pairs (good, dirt, defocus, folds) are shown in Fig. 3. Since the registration reconstructs the spatial correspondence in the 3rd dimension, the resulting image stack can then be regarded as a 3D dataset of 3000 x 3000 x 2607 voxel (about 44 GB @ 16-bit), see Fig. 5.

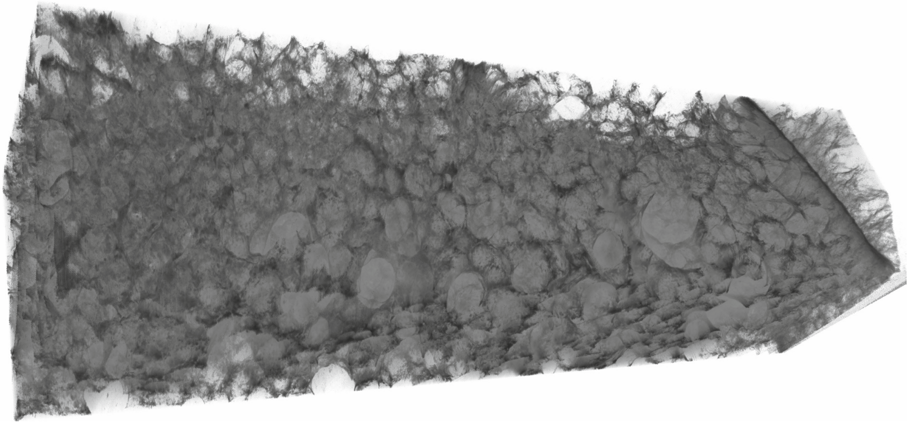

Fig. 5. Volume rendering of the 3D stack
Volume rendered visualization of the reconstructed 3rd dimension of the lung tissue block (sub-extent of 1000 × 1000 × 2607 voxel). Tissue dark, resin semi-transparent grey (airspaces and blood vessels).

The volume in lung samples occupied by tissue is only about 10%–20% [3,7], so there is about 90%–80% mostly non-correlating texture in image pairs which disturbs the registration process. This is one reason why the registration of serial sections of lung tissue is challenging. A possible countermeasure is to "fill" the non-tissue space with (roughly) correlating data. This can be achieved by first auto-thresholding the image to roughly binarize tissue and non-tissue and then generating a distance map, which is implemented in the branch ot+dm and leads to image pairs as in Fig. 6. However, applying this version of `recRegStack.py` to the K2-dataset (K2_fibrosis@a5617581) showed, that more mITs are needed.[4] A reason for this might be that this approach is more sensitive to dirt, which ends up in the tissue segment and therefore causes significant disturbance in the distance transform, see Fig. 6. Another promising approach could be registration based on landmarks generated by SIFT [14,15], similar to Fiji's "Register Virtual Stack Slices" [4] but using Elastix/ITK in order to keep the features of manual intervention (mIT and iPF)[5].

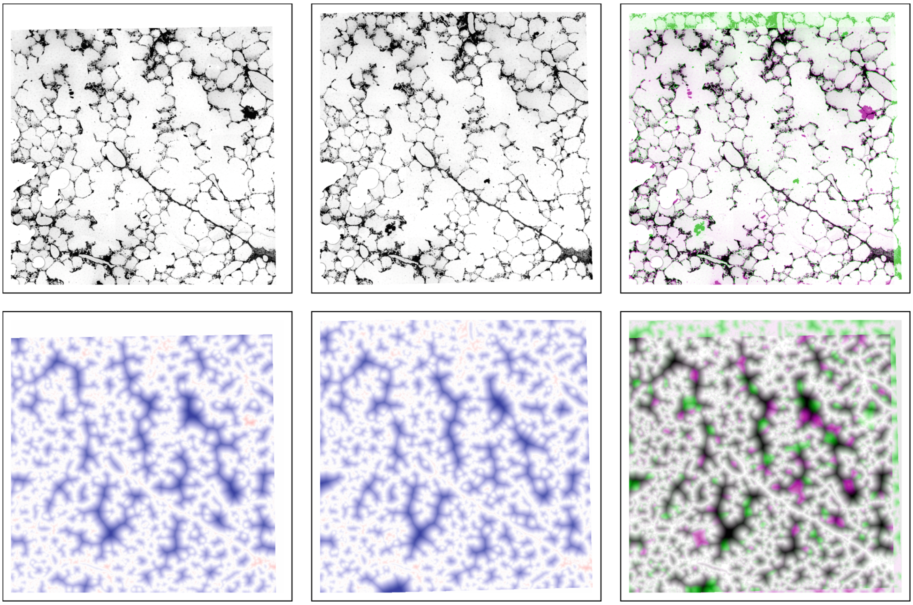

Fig. 6. Exemplary image pair for ot+dm variant
Top row: image pair as is (no mIT or iPF needed), fixed image, moving image, Magenta-Green overlay. Bottom row: ot+dm image pair for images on the left (blue-white-red LUT from -50 to 50, blue: air, blood space; red: tissue, dirt). (Color figure online)

[4] A full alignment with ot+dm was not pursued further because currently SimpleITK of SimpleElastix does not allow to set a mask for the otsu-threshold calculation (ot-mask). Therefore, the continuation feature of `recRegStack.py` cannot be used so that after each mIT creation the registration has to start from the beginning and therefore the whole procedure needs much more time. An alternative would be to port `recRegStack.py` to ITKElastix.

[5] The lack of the possibility for manual intervention and a trial of more than two weeks to find meta-parameters to register the K2-dataset with Fiji's "Register Virtual Stack Slices" was actually the motivation for implementing `recRegStack.py`.

4 Discussion

The described proof of principle combines automated alignment with manual intervention such that ideally the automation does the whole work but also ensures that “in the worst case” at least the result of a pure manual alignment will be achieved. This comes with a need to balance the two time consuming tasks: Either tuning the meta-parameters for the automated alignment of as many images in the series as possible (total time consumption can be unlimited) or helping the automation with initial manual alignments (total time is limited).

While the current implementation with Makefiles serves as a proof of principle, it can be further improved to a more intuitive and user-friendly program. For example, by incorporating the processing done by various commands into the Python code, as well as avoiding `midas` and other IMOD-tools or aborting auto alignment in order to provide a manual alignment to continue with. A graphical user interface (GUI) could provide visual feedback (similar to the image viewer `geeqie`), incorporate the needed `midas` functionality and offer a “manual intervention” button. In principle, a threshold on the final metric value could be used to automatically trigger the suggestion of a manual alignment.

Another variant of `recRegStack.py` (branch combT_01) accumulates all former transforms and adds the newly found transform of the processed image pair as in [21]. While the two approaches should yield similar results, accumulation of a few hundred transforms can become problematic but in return can avoid larger differences in case of already similarly recorded image pairs. A third variant (branch reg2tra+prevTra) uses both approaches and chooses the result with the better metric value finally achieved.

Ideally, a (non-destructive) tomogram or some form of markers should be used for guiding 3D reconstruction of serial sections (e.g. constrain local deformation corrections and avoid continuous drift), such functionality is offered by e.g. HistoloZee [1].

`recRegStack.py` provides the option to ignore some images of the series in case some slices are too distorted for registration or lost during the preparation (czi2stack/lostSlieds.lst). After aligning the next usable image of the series, a reconstruction of the lost slice can be created from the adjacent slices [14].

Acknowledgement. Special thanks go to Susanne Faßbender (for excellent technical assistance), Lena Ziemann (for slide scanning and creation of mITs and iPFs), Kasper Marstal, Fabien Pertuy, Stefan Klein and Marius Staring (for feedback on use of Elastix, SimpleElastix and debugging) and to David Mastronarde, Daniel Adler (for feedback on IMOD and HistoloZee details) and Oleg Lobachev (for feedback on the general approach).

References

1. Adler, D., Yushkevich, P.: HistoloZee, Penn Image Computing and Science Laboratory (PICSL), University of Pennsylvania. http://picsl.upenn.edu/software/histolozee/

2. Arganda-Carreras, I., Fernàndez-Gonzàlez, R., Muñoz-Barrutia, A., Ortiz-De-Solorzano, C.: 3D reconstruction of histological sections: application to mammary gland tissue. Microsc. Res. Tech. **73**(11), 1019–1029 (2010). https://doi.org/10.1002/jemt.20829
3. Buchacker, T., et al.: Assessment of the alveolar capillary network in the postnatal mouse lung in 3D using serial block-face scanning electron microscopy. Front. Physiol. **10**, (2019). https://doi.org/10.3389/fphys.2019.01357
4. Cardona, A., Arganda-Carreras, I., Saalfeld, S.: Register Virtual Stack Slices (Fiji). http://imagej.net/Register_Virtual_Stack_Slices
5. Cheung, W., Hamarneh, G.: n-SIFT: n-dimensional scale invariant feature transform. IEEE Trans. Image Process. **18**(9), 2012–2021 (2009). https://doi.org/10.1109/TIP.2009.2024578
6. Fiji/ImageJ development team: Fiji (ImageJ), see [25, 26]. http://fiji.sc
7. Grothausmann, R., Knudsen, L., Ochs, M., Mühlfeld, C.: Digital 3D reconstructions using histological serial sections of lung tissue including the alveolar capillary network. Am. J. Physiol.- Lung Cell. Mol. Physiol. **312**(2), L243–L257 (2017). https://doi.org/10.1152/ajplung.00326.2016
8. Ibanez, L., et al.: ITK, see [10, 33]. Kitware Inc. http://www.itk.org
9. Johnson, H.J., McCormick, M.M., Ibàñez, L.: The Insight Software Consortium: The ITK Software Guide. Kitware Inc., Clifton Park (2018)
10. Klein, S., Staring, M.: elastix, see [12, 27]. Image Sciences Institute, University Medical Center Utrecht. http://elastix.isi.uu.nl/
11. Klein, S., Staring, M., Murphy, K., Viergever, M.A., Pluim, J.P.: elastix: a toolbox for intensity-based medical image registration. IEEE Trans. Med. Imaging **29**(1), 196–205 (2010). https://doi.org/10.1109/TMI.2009.2035616
12. Kremer, J.R., Mastronarde, D.N., McIntosh, J.R.: Computer visualization of three-dimensional image data using IMOD. J. Struct. Biol. **116**(1), 71–76 (1996). https://doi.org/10.1006/jsbi.1996.0013
13. Linkert, M., et al.: Metadata matters: access to image data in the real world. J. Cell Biol. **189**(5), 777–782 (2010). https://doi.org/10.1083/jcb.201004104
14. Lobachev, O.: The tempest in a cubic millimeter: image-based refinements necessitate the reconstruction of 3D microvasculature from a large series of damaged alternately-stained histological sections. IEEE Access (2020). https://doi.org/10.1109/access.2020.2965885
15. Lowe, D.G.: Object recognition from local scale-invariant features. **2**, 1150–1157 (1999). https://doi.org/10.1109/ICCV.1999.790410
16. Marstal, K.: SimpleElastix, see [18]. Image Sciences Institute, University Medical Center Utrecht. http://simpleelastix.github.io/
17. Marstal, K., Berendsen, F., Staring, M., Klein, S.: SimpleElastix: a userfriendly, multi-lingual library for medical image registration. In: Schnabel, J., Mori, K. (eds.) International Workshop on Biomedical Image Registration (WBIR). IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 574–582, Las Vegas, Nevada, USA (2016)
18. Mastronarde, D.N.: Dual-axis tomography: an approach with alignment methods that preserve resolution. J. Struct. Biol. **120**(3), 343–352 (1997). https://doi.org/10.1006/jsbi.1997.3919
19. Mastronarde, D.N.: IMOD, see [21, 19, 13]. http://bio3d.colorado.edu/imod/
20. Mastronarde, D.N., Held, S.R.: Automated tilt series alignment and tomographic reconstruction in IMOD. J. Struct. Biol. **197**(2), 102–113 (2017). https://doi.org/10.1016/j.jsb.2016.07.011. SI: Electron Tomography

21. Mueller, D., Vossen, D., Hulsken, B.: Real-time deformable registration of multi-modal whole slides for digital pathology. Comput. Med. Imaging Graph. **35**(7–8), 542–556 (2011). https://doi.org/10.1016/j.compmedimag.2011.06.006
22. Ochs, M., Knudsen, L., Hegermann, J., Wrede, C., Grothausmann, R., Mühlfeld, C.: Using electron microscopes to look into the lung. Histochem. Cell Biol. **146**(6), 695–707 (2016). https://doi.org/10.1007/s00418-016-1502-z
23. Open Microscopy Environment: bftools (scripts using Bio-Formats), see [14]. https://docs.openmicroscopy.org/bio-formats/5.7.2/users/comlinetools/index.html
24. Schindelin, J., et al.: Fiji: an open-source platform for biological-image analysis. Nat. Methods **9**(7), 676–682 (2012). https://doi.org/10.1038/nmeth.2019
25. Schneider, C.A., Rasband, W.S., Eliceiri, K.W.: NIH Image to ImageJ: 25 years of image analysis. Nat. Method **9**(7), 671–675 (2012). https://doi.org/10.1038/nmeth.2089
26. Shamonin, D., Bron, E., Lelieveldt, B., Smits, M., Klein, S., Staring, M.: Fast parallel image registration on CPU and GPU for diagnostic classification of Alzheimer's disease. Front. Neuroinform. **7**, 50 (2014). https://doi.org/10.3389/fninf.2013.00050
27. Steffen, L., et al.: Surfactant replacement therapy reduces acute lung injury and collapse induration-related lung remodeling in the bleomycin model. Am. J. Physiol.-Lung Cell. Mol. Physiol. **313**(2), L313–L327 (2017). https://doi.org/10.1152/ajplung.00033.2017
28. Tange, O.: GNU parallel - the command-line power tool. Login: The USENIX Mag. **36**(1), 42–47 (2011)
29. Tange, O.: GNU parallel - the command-line power tool, see [29]. http://www.gnu.org/s/parallel/
30. Wang, C.-W., Gosno, E.B., Li, Y.-S.: Fully automatic and robust 3D registration of serial-section microscopic images. Sci. Rep. **5**, 15051 (2015). https://doi.org/10.1038/srep15051
31. Williams, T., Kelley, C.: gnuplot. http://gnuplot.sourceforge.net/docs_4.4/gnuplot.pdf
32. Yoo, T., et al.: Insight into Images: Principles and Practice for Segmentation, Registration, and Image Analysis. A K Peters Ltd., Natick (2004)