AUTOMATED HIGH SPEED STITCHING OF LARGE 3D MICROSCOPIC IMAGES

Yang Yu[+], *Hanchuan Peng*[*+]

Janelia Farm Research Campus, Howard Hughes Medical Institute, Ashburn, VA 20147

ABSTRACT

High-resolution microscopic imaging of biological samples often produces multiple 3D image tiles to cover a large field of view of specimen. Usually each tile has a large size, in the range of hundreds of megabytes to several gigabytes. For many of our image data sets, existing software tools are often unable to stitch those 3D tiles into a panoramic view, thus impede further data analysis. We propose a simple, but accurate, robust, and automatic method to stitch a group of image tiles without *a priori* adjacency information of them. We first use a multiscale strategy to register a pair of 3D image tiles rapidly, achieving about 8~10 times faster speed and 10 times less memory requirement compared to previous methods. Then we design a minimum-spanning-tree based method to determine the optimal adjacency of tiles. We have successfully stitched large image stacks of model animals including C. *elegans*, fruit fly, dragonfly, and mouse, which could not be stitched by several existing methods.

Index Terms— Image Stitching, Phase Correlation, Correlation Coefficient, Multiscale Analysis, FFT

1. INTRODUCTION

With the development of modern microscopy, there is a growing demand of imaging large and thick 3D biological specimens at high resolution. In many cases, the prepared samples are too large to fit into the field of view of a microscope. Motorized stage has become important to image large areas in separate tiles, each containing the picture of a piece of the specimen. Although physical coordinates from the stage may be recorded and used in stitching of all tiles into a panoramic view, there are many situations that these coordinates may not be precise enough. There are also cases that such physical displacement information is not available. Hence, automatic 3D stitching of multiple un-organized image tiles is needed.

Image stitching methods often find a geometric transformation to maximize a similarity cost function between adjacent images. Many cost functions have been proposed for image stitching problems including feature-based, intensity-based and hybrid approaches [1-3]. Feature-based methods search for corresponding salient points, which are often edge points or corners, in the overlap regions of a pair of images. These methods may be sensitive to the detection of the salient points and the determination of the correspondence among these points. Intensity-based methods are to perform a brute-force search of the transformation (mapping from one vector space to another) space, which can be optimized incorporating multi-resolution strategies. Hybrid methods combine the advantages of both techniques, which make intensity-based alignment metrics computation restricted in local regions around the detected salient points.

MosaicJ [1] is a 2D semi-automatic stitching software solution in the form of the ImageJ plugin. For stitching a pair of images, it can be fully automatic. However, it will be time-consuming for aligning a pair of 3D images directly extended from its rigid transformation model with a rotation. For group-wise image stitching, the stitching of a large number of tiles may be prohibitively expensive due to its requirement of manual positioning of the tiles. Another ImageJ stitching plugin, which distributed as a part of the Fiji project [2], is intensity-based automated stitching. It is multi-threaded and thus takes advantage of multi-core CPUs. However, for large-scale 3D microscopic images of our datasets, we find it easily run out of physical memory. XuvTools [3] is another automated 3D stitching software using a hybrid registration method. It uses multiscale strategies for stitching. At a coarse scale, it uses phase-only correlation to obtain rough estimation. In the fine scale, it detects salient points that would appear in overlap regions, followed by aligning the data based on maximizing correlation coefficient in a series of small windows around the salient points. Since both phase correlation and salient points detection may be sensitive to noise, the stitching results may not be consistent when choosing different scaling factors. Zhao et al also developed a 3D image stitching approach, which has been used in applications such as tiled fruit fly confocal images (unpublished data). Its computational performance is similar to that of the method presented in Fiji. For many cases in our testing, it has not yet produced ideal results as one would like to have.

Despite above existing methods, automatic and robust stitching of large-scale 3D microscopic images remains challenging. We note that for microscopic images, the

[*] Corresponding author: Hanchuan Peng (pengh@janelia.hhmi.org)
[+] Equal Contribution.

magnification is usually a constant for all tiles. In addition, in most cases, it is not essentially necessary to correct the potential lens distortion in imaging. The potential variation of orientation of successive tiles is often negligible. Thus, the main degree of freedom would be the translation among different tiles, as well as potential intensity variation of different tiles. Moreover, the order, or configuration, of all tiles may be unknown. In this paper, for the above most common situation(s) one may encounter in a series of 3D image datasets of different model animals, we propose an efficient and robust multiscale stitching approach.

2. METHOD

To make this section self-contained, let's re-state the goal of this stitching problem. Given an un-organized set of image tiles $S = \{T_1, T_2, \ldots T_N\}$, where each T_i, (i=1, 2, …N) is a 3D image stack and N is the number of tiles, the goal is to automatically determine their order (i.e. the configuration of their spatial adjacency) as well the relative displacement of adjacent tiles in 3D, with the assumption that there could be intensity fluctuation between tiles but other geometrical variations are negligible.

To solve this problem, we consider a two-stage stitching approach. The first stage, called pair-wise registration, is to best align every possible pair of tiles and thus obtain the respective distance score between this pair of images. In this step, all possible translations will also be estimated. The second stage is called group registration, during which we formulate a graph, where each tile is treated as a graph node and the edge weight between two tiles "nodes" is chosen as the distance of the two tiles. Then from the initial graph we compute a minimum spanning tree, which indicates the order/configuration of tiles.

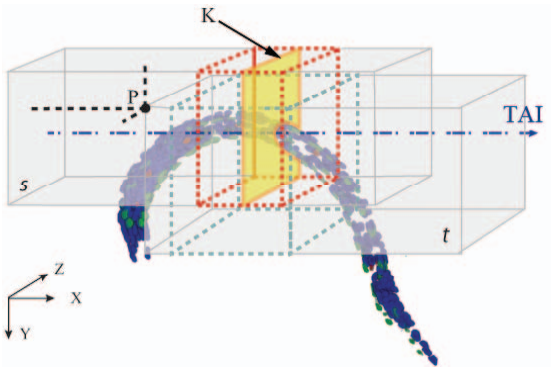

Fig. 1. Illustration of our stitching method at a fine scale. See Box 1 for the notations. Image tiles of *C. elegans* are shown in this example. Red, green and blue in the image: mCherry, GFP, and DAPI staining of cells.

2.1. Pair-wise Registration

There are many ways to best align a pair of image tiles and determine their optimal translation. Here we consider the image intensity based approach. In this category, a brute-force approach in the spatial domain would require trying all possible displacement of the two images along X, Y, Z three dimensions. The best displacement can be defined as the location where two images have the smallest overall intensity difference, or almost equivalently, the largest correlation. In many cases, correlation of two images [4] is a good criterion when the baseline intensities of the two images are different. Such difference will be naturally removed when correlation is used. Thus we seek to find the translation that maximizes the correlation of two images.

Previous studies suggested that both phase correlation (PC) [3] and normalized cross-correlation (NCC) (Zhao, unpublished data), or their combination [2], can be used in stitching 3D images. Suppose we have a generic pair of images, t and s. We call t the *target* and keep it still during stitching. We call s the *subject* and translate it to best match the target. PC [5] can be computed efficiently used the fast Fourier transform, as shown below

$$PC(s,t) = F^{-1}\left(\frac{F(s)F^*(t)}{\left|F(s)F^*(t)\right|}\right),$$

where F denotes the discrete Fourier transform, -1 and * denote inverse and conjugate transforms respectively. On the other hand, NCC is defined in the following way,

$$NCC(s,t) = \frac{\sum(s-\bar{s})(t-\bar{t})}{\sqrt{\sum(s-\bar{s})^2\sum(t-\bar{t})^2}}.$$

NCC can also be expressed in term of the Fourier transform,

$$NCC(s,t) = \frac{F^{-1}\left(F(s)F^*(t)\right) - \frac{1}{N}\sum s\sum t}{\sqrt{\left(\sum s^2 - \frac{1}{N}\left(\sum s\right)^2\right)\left(\sum t^2 - \frac{1}{N}\left(\sum t\right)^2\right)}}.$$

In our method, we combine both PC and NCC. Previous studies [1, 2, 3] show that NCC is typically more faithful due to normalization of the overlapping region, but in the extreme cases (e.g. the full image or a single voxel) it won't extrapolate as well as PC. Hence we use PC to find the top M candidates (typically M = 8), defined as the best M local maxima in the PC map. Then within a small surrounding window (typically 3×3×3) of each local maximum, we select the locations that yield the largest NCC value. Finally the best translation location is chosen as the one that has the largest NCC value among the M candidates. We call this basic algorithm PCNCC.

Obviously, to implement of this PCNCC method, we need to do only one pass of the forward Fourier transform and two passes of the inverse transforms, one for PC and one for NCC. We also use pre-computed running sum tables of the image and squared image [4] to accelerate the computation. Generally speaking, this is efficient. However, we note that for large 3D microscopic images each of which has hundreds mega-voxels to several giga-voxels, this direct implementation may still be slow.

To further reduce the computational complexity, we design a simple but highly effective two-scale hierarchical

method to use PCNCC. While these two scales share the same PCNCC algorithm, they use different portions of the images as their input.

At a coarse scale, we downsample both *t* and *s* 5 times along each dimension. As a result, aligning the smaller images yield a much faster, but coarse, alignment. This may also make the PC peaks spread to more distal locations. In this way, we first produce a rough estimation of the translation offset-triple (along three axes) of *t* and *s*. This rough offset-triple is denoted as {*dcx*, *dcy*, *dcz*}.

Then at a fine scale, we search the optimal offsets, but only within a local area that is close to the best location estimated in the coarse scale. This method is shown in Fig. 1. We first find the axis (among X, Y, and Z) for which *t* and *s* have the smallest overlap, defined by the translation offsets *dcx*, *dcy*, *dcz* normalized by the dimensions of image tiles. This axis is called the translation axis of interest (TAI). Then, within the rough overlapping region we compute a key plane, which is orthogonal to the TAI and at the same time has the largest image contrast (and thus presumably most informative). Next, we extract a local volume of interest (VOI) close to the key plane for both *s* and *t*. Typically such a VOI consists of only 15% of the *overlapping* image planes along TAI, and thus has a very small size. We run the same PCNCC algorithm on the VOIs extracted from *t* and *s*, and obtain the final translation offset. Overall, our pair-wise registration algorithm can be summarized as Box 1. The novel idea is to use a two-scale algorithm to best optimize both PC and NCC values for small volumes of the original image tiles. This algorithm is about 10 times faster than existing approaches.

Box. 1. Pair-wise image stitching algorithm.

Input: two tiles *s* and *t*
Output: the displacement between tiles *s* and *t*
// coarse scale
1: compute phase correlation (PC) using FFT to obtain *M* peaks as translation offset candidates
2: compute normalization cross correlation (NCC) in small windows around *M* candidates to obtain translation offset estimation (*P* in Fig.1)
// fine scale
3: extract a plane with most foreground information from subject image s as the key plane (*K* in Fig. 1), which is orthogonal to the translation axis of interest (TAI)
4: extract volumes of interests (VOIs, shown as red and green dashed boxes in Fig. 1) from *s* and *t* corresponding to *K*
5: compute PCNCC of VOIs to obtain the final displacement estimation

2.2. Group Registration

Our global approach determines the translation set for all tiled images that are connected to the graph where the tiles are the nodes and the edges are correlations between adjacent image tiles. In the case of without prior knowledge of adjacency relationship, this computation cost could be very high on the construction of a mosaic by blindly attempting to align each tile to the remaining ones. In our approach, the geometry topology is established by finding single distinctive maximum spanning tree (MST) at a coarse scale. This rough alignment from the spanning tree will be refined by means of our multiscale pairwise stitching.

We obtain the optimal configuration (adjacency) via solving a maximum spanning tree problem to produce a globally optimal alignment of all tiles. The solution gives the set of translational offsets with the maximum correlation scores between all image pairs. We use Prim's algorithm [6] to find the maximum spanning tree.

Computation at this stage can be reduced. Indeed, we use the NCC values computed at the coarse level of Box 1 algorithm, instead of the optimal NCC values at the fine level, to set the edge weight of the graph. Then after we have figured out which pair of tiles would be adjacent, we optimize the respective translation offset.

2.3. Other Implementation Issues

We compute the discrete Fourier transform using the well-known FFTW library [7], which is able to handle a wide range of size, dimension and stride of the data vector, and also provide in-place transform.

Microscopic images often come with multiple color channels. Typically we choose the color channel with the best signal-to-noise ratio by eyes as the reference for stitching.

To fuse the overlapping regions in adjacent tiles, we use the average intensity of these regions. This simple method works in most cases in our testing of a variety of data. This can certainly be extended as linear blending, as in [2].

3. RESULTS

We applied our approach to various microscopic datasets where the specimens were imaged at high resolution. We generated and proofread 3D montages of large datasets from multiple laboratories containing more than 200 microscopic images from mouse brain, *Drosophila* (Fig. 2), and *C. elegans*. In a comprehensive testing using all these images, methods in XuvTools [3] and others often failed or did not produce consistent results when choosing different scaling factor. Therefore, in the following, we focus on comparing our method to Preibisch's method [2] because it could stitch most our 3D microscope images.

For pairwise image stitching (Fig. 3), it is clearly that for our test datasets, our method used at most about 2.5G memories for the biggest dataset, whereas method in [2] used 16G memory in such a case. The speed of [2] was also slow. For instance in the mouse brain test case, that method used almost 200 seconds whereas our method used only 20 seconds. Thus the overall computational complexity of our method was about 80 ~ 100 times less than that of [2].

For group registration (Table 1), our method was able to stitch 9 tiles in about 10 minutes, while method in [2] failed.

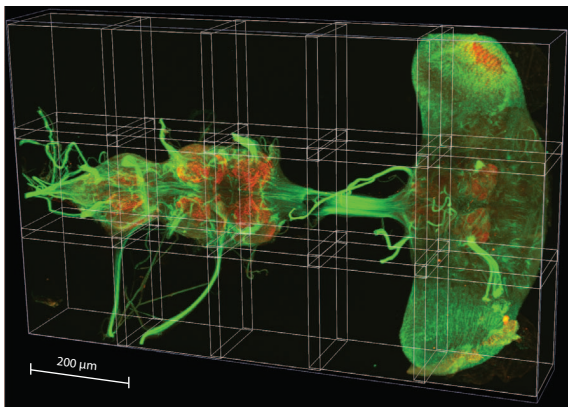

Fig. 2. 3D stitching of a complete *Drosophila* nervous system, including both brain (right) and ventral nerve cord (left), from 15 3D tiles. Our stitcher used about 6 minutes to produce this result. The Fiji tool failed in this case probably due to a few tiles have very week signal. Green, anti-Neuroglian (BP104); red, anti-DN-cadherin (DN-EX #8).

For the choice of parameters, we only need to readjust sampling factors for 1 out of 30 dataset of *C. elegans* and 1 out of 20 dataset of fruit fly brain.

Tiles	Stitched Image Size	Timing
3	1.9GB	1m0s ours 9m20s Preibisch's
6	3.8GB	4m38s ours 37m43s Preibisch's
9	6.0GB	10m13s ours (Out of memory) Preibisch's

Table 1. Illustration of stitching performance on tiled microscopic images computed on a Linux machine with Intel® 8-Core CPU (2.66 GHz) and 35 GB of RAM. Single tile dimension is 1476×1476×160.

4. CONCLUSIONS AND AVAILABILITY

We present a fully automated high-speed multiscale image stitching method for large-scale 3D microscopic images. The specific stitching strategy yields successful results on microscope images. Furthermore, the computational complexity of our method, in terms of computing time and memory requirement, is much lower than existing methods. This may provide a very useful tool for a wide range of applications. Our program supports different data types, such as 8-bit, 16-bit and 32-bit images.

Our stitching method is implemented in C/C++, and is provided as plugin of V3D [8] (see link below), which is an open source and freely downloadable platform for high-performance 3D+ image analysis and visualization. The web site is http://penglab.janelia.org/proj/v3d. Our stitching plugin and tutorial can be found at following the website http://penglab.janelia.org/proj/stitching/.

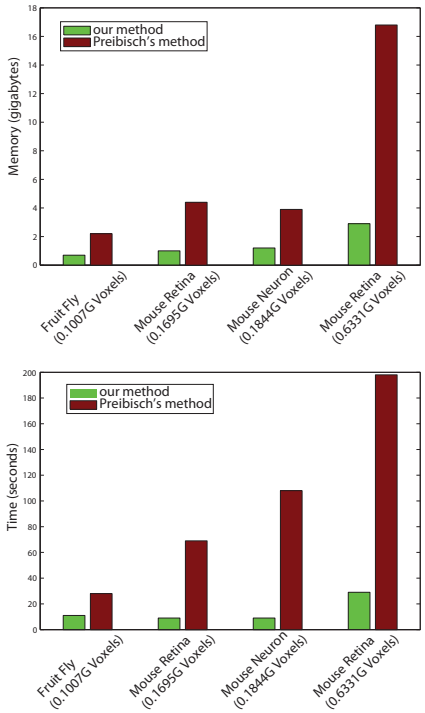

Fig. 3. Comparison of single-pair image-stitching with Preibisch's method in memory usage and time consumption with different image data set.

5. ACKNOWLEDGEMENTS

We thank Xiao Liu, Ananya Mitra, Richard Tsien, Bart Borghuis, Samuel Gale, Jinhyun Kim, and David Mellert for providing microscopic images. We also thank Ting Zhao and Steven G. Johnson for their helpful suggestions on how to efficient implementation of DFT using FFTW library. We also thank people in Gene Myers lab and Hanchuan Peng lab for discussion.

6. REFERENCES

[1] P. Thevenaz, and M. Unser, "User-friendly semiautomated assembly of accurate image mosaics in microscopy," Microscopy Research and Technique, 70, pp. 135-146, 2007.
[2] S. Preibisch, et al. "Globally optimal stitching of tiled 3D microscopic image acquisitions," Bioinformatics, 25, no. 11, pp. 1463-1465, 2009.
[3] M. Emmenlauer, et al., "XuvTools: free, fast and reliable stitching of large 3D datasets," Journal of Microscopy, 233(1), pp. 42–60, 2009.
[4] J.P. Lewis, "Fast template matching," Vision Interface, pp. 120-123, 1995.
[5] C. Kuglin, and D. Hines, "The Phase Correlation Image Alignment Method," IEEE International Conference on Systems, Man and Cybernetics, pp. 163-165, 1975.
[6] R.C. Prim, "Shortest connection networks and some generalizations," Bell Systems Technology Journal 36, pp. 1389-1401, 1957.
[7] M. Frigo, and S.G. Johnson, "The Fastest Fourier Transform in the West," Technical Report, UMI Order Number: TR-728, Massachusetts Institute of Technology, 1997.
[8] H. Peng, et al. "V3D enables real-time 3D visualization and quantitative analysis of large-scale biological image data sets," Nature Biotechnology 28, pp. 348-353, 2010.