HHS Public Access

Author manuscript

IEEE J Biomed Health Inform. Author manuscript; available in PMC 2017 May 01.

Published in final edited form as:

IEEE J Biomed Health Inform. 2016 November ; 20(6): 1575–1584. doi:10.1109/JBHI.2015.2480712.

MISTICA: Minimum Spanning Tree-based Coarse Image Alignment for Microscopy Image Sequences

Nilanjan Ray [Member IEEE],
Computing Science, University of Alberta, Canada

Sara McArdle,
La Jolla Institute for Allergy & Immunology, La Jolla, CA, USA and Department of Bioengineering, University of California, San Diego, La Jolla, CA, USA

Klaus Ley, and
La Jolla Institute for Allergy & Immunology, La Jolla, CA, USA

Scott T. Acton [Fellow IEEE]
Electrical & Computer Engineering and Biomedical Engineering, University of Virginia, Charlottesville, VA, USA

Abstract

Registration of an *in vivo* microscopy image sequence is necessary in many significant studies, including studies of atherosclerosis in large arteries and the heart. Significant cardiac and respiratory motion of the living subject, occasional spells of focal plane changes, drift in the field of view, and long image sequences are the principal roadblocks. The first step in such a registration process is the removal of translational and rotational motion. Next, a deformable registration can be performed. The focus of our study here is to remove the translation and/or rigid body motion that we refer to here as coarse alignment. The existing techniques for coarse alignment are unable to accommodate long sequences often consisting of periods of poor quality images (as quantified by a suitable perceptual measure). Many existing methods require the user to select an anchor image to which other images are registered. We propose a novel method for coarse image sequence alignment based on minimum weighted spanning trees (MISTICA) that overcomes these difficulties. The principal idea behind MISTICA is to re-order the images in shorter sequences, to demote nonconforming or poor quality images in the registration process, and to mitigate the error propagation. The anchor image is selected automatically making MISTICA completely automated. MISTICA is computationally efficient. It has a single tuning parameter that determines graph width, which can also be eliminated by way of additional computation. MISTICA outperforms existing alignment methods when applied to microscopy image sequences of mouse arteries.

Index Terms

microscopy image registration; image sequence alignment; minimum spanning trees

I. Introduction

Automated registration or alignment of time-sequenced images is an essential image processing task for microscopy image analysis. Due to various factors, time-sequenced microscopy images can be misaligned. For further studies, visualization, and measurements to be made on or from the images, all the images belonging to a sequence need to be registered in a common frame of reference. Automation is preferred for longer sequences, where frequent user interaction becomes extremely tedious.

In vivo microscopy images, which are fundamental to the study of significant research problems, such as the study of atherosclerosis [9], pose several challenges to image sequence alignment. These sequences are often long, exceeding 1000 images. The motion artifacts are significant due to the cardiac and the respiratory motions. Intravital image acquisition is subject to changes in focal plane and drifts in the field of view due to gross tissue movement. The exact cause of this motion is unknown, but it is likely due to the drift of the microscopy stage and displacement of fluid in the living tissue throughout the imaging time. These factors, mixed together, create considerable challenges for the existing techniques available for image sequence alignment.

In the literature, much of the attention on microscopy image registration has been placed on registration models and methods between two images (2D images or 3D volumes). Some papers focus on element matching algorithms, for example feature detection-based methods [13, 14], or intensity-based methods [18], while others focus on transformation models, including linear, elastic [15, 22], spline-based models [1, 17]. More robust methods use an intensity-based model for detection and a parametric motion model for transformation [11], or contour-based model for detection and both a linear and a Navier equation model for transformation [16].

The focus of this paper is on a complementary issue: the alignment of a *sequence* of images, which has received much less attention in the literature. In this respect, most of the existing body of work is based on an anchor image-based alignment that first lets the user select an anchor image, which does not change during the registration/alignment process. Few methods allow automatic anchor selection, with one such method found in [1, 14]. Others choose an anchor image randomly from the middle of a sequence [13]. Another approach finds an anchor frame that maximizes the correlation with the remaining frames [4, 6]. After anchor selection, the image frames are sequentially or recursively aligned to the anchor image, one after another.

The existing techniques of image sequence alignment often break down when the sequence contains an extended subsequence of poor quality images (as quantified by SSIM [21] or other suitable measures), such as those distorted by tissue motion. Also, these methods often do not work well on longer sequences, because the sequential alignment inherently accumulates registration errors. Our study here focuses on these issues.

Alternatives to the recursive alignment process have been proposed before by means of global alignment of an elastic volume [15]. For a set of time-sequenced images, the method simultaneously aligns all images to their few neighboring images (e.g., it considers 5

neighboring frames for each image frame) adjusting elastic parameters. The method is in principle a type of bundle adjustment [10], which tries to refine transformation parameters for a set of images simultaneously. Bundle adjustments are known to be slow in general and do not scale well, as computational cost becomes prohibitive for longer image sequences. Besides, there are a multitude of parameters (over 15) in the method [15] that need to be carefully tuned. The method [15] sometimes recommends a pre-alignment of the images sequence by rigid body transformation or translation for better results.

In this paper, we offer a completely automated and computationally efficient alternative, MISTICA, which overcomes the aforementioned roadblocks of the existing techniques. In any sequence of images, there are images that conform or register well with each other and there are nuisance, non-conforming images that form impediments to a registration method. MISTICA creates a number of subsequences from the original sequence, where each subsequence re-orders images starting with the most conforming to the least conforming images. Then, it registers the images in this new order in each subsequence. Registration error accumulation is reduced because substandard nonconforming images are registered at the end. In contrast, the elastic alignment method [15] attempts to overcome registration error accumulation by local adjustment of elastic parameters.

MISTICA selects an anchor image automatically, removing the burden of the user interaction. It has a single tuning parameter, graph width, measured in number of time-hops between two images, making it more user friendly than many existing techniques. Our experiments show that for a number of image sequences, the registration performance is robust to changes of graph width value within a wide range. Furthermore, we show that with some additional computational overhead, the requirement to input a graph width can be relaxed.

The concept used in MISTICA first appeared in a peer reviewed conference communication [8]. Here, we report the motivation and insight of the algorithm in detail with extensive experiments and comparisons. MISTICA uses weighted minimum spanning tree (MST) algorithm [3], which is to our knowledge the first to be deployed on the image sequence alignment problem.

II. Background and Motivation

We first make an observation that typical methods for aligning a sequence $S = \{I_1, I_2, \ldots, I_n\}$ of images follow a generic algorithm as outlined below.

Generic Image Sequence Alignment

Step 1 (Anchor selection)—Choose an anchor image. The anchor image does not change during registration. For many algorithms, the first image in the sequence is the anchor image. Alternatively, the user chooses an image I_i to be the anchor image. Let us denote the anchor image by I.

Step 2 (Subsequence generation)—Decompose the input image sequence S into a number of image sequences $S_1, S_2, \ldots, S_m$, so that $S = \{I\} \cup S_1 \cup S_2 \cup \ldots \cup S_m$. Informally, we

call these S_j's subsequences. For typical existing algorithms, these subsequences are non-overlapping and determined by the order of images in S. For example, if the i^{th} image is the anchor image, then two subsequences are created: $S_1 = \{I_{i-1}, I_{i-2}, \ldots, I_1\}$ and $S_2 = \{I_{i+1}, \ldots, I_n\}$.

Step 3 (Subsequence alignment)

```
For each subsequence Sj = {I1j, I2j, …, I|Sj|j}, j = 1,…,m, do:
  Set anchor image I as the target image: T ← I.
  For k = 1, …, |Sj|, do:
    Register Ikj to T. Call the registered image Rkj.
    Update target image: T ← Rkj.
  End For
End For
```

In this framework, probably the simplest possible algorithm is a fixed template matching [2] for registration that first decides on an anchor image, say, I_i, and considers the anchor as the fixed template. Then, such an algorithm registers every other image to the anchor image. In this case, $m = n$-1, and $S_1 = \{I_1\}$, .., $S_{i-1} = \{I_{i-1}\}$, $S_i = \{I_{i+1}\}$, …, $S_{n-1} = \{I_n\}$. Take another example, StackReg [19], which follows the generic algorithm with two subsequences $S_1 = \{I_{i-1}, I_{i-2}, \ldots, I_1\}$ and $S_2 = \{I_{i+1}, \ldots, I_n\}$.

Possibly the most severe limitation of the existing algorithms is their inability to correctly handle spells of images with poor quality or nuisance images. One such example image sequence is given in Fig. 1(a), which illustrates 6 images in a sequence $S = \{I_1, I_2, .., I_6\}$. Out of these images, I_2 and I_3 have poor quality, as depicted by shaded nodes in Fig. 1(a). If one chooses I_1 as the anchor image, we have only one subsequence (m=1) to register, i.e., $S_1 = \{I_2, \ldots, I_6\}$. However, the images I_2 and I_3 are poor in quality (by some measure), which leads to propagation of the registration error in S_1. On the other hand, if one chooses I_4 as the anchor image, we have two subsequences to register: $S_1 = \{I_3, I_2, I_1\}$ and $S_2 = \{I_5, I_6\}$. Now we have a slightly better situation that error propagation affects only S_1, but not S_2.

The next difficulty faced by many algorithms is the length of the subsequences. Many existing algorithms do not work well for longer subsequences, because of changing illumination, photobleaching of fluorescent dyes or drift in the field of view. This adverse effect is further compounded by the presence of poor quality images.

A third issue involves user interaction to choose an anchor image. In fact, a number of the existing algorithms require the user to choose an anchor image. Our example here shows that the registration error propagation can be mitigated with a careful selection of the anchor image. For example, the choice of I_4 over I_1 (in this example) would most likely yield a better registration. In practice, where the length of the image sequence may be in hundreds, browsing through images and looking for the anchor can be tedious and error-prone.

MISTICA overcomes these difficulties by creating a minimum weighted spanning tree, which is explained with the aforementioned example. We consider a graph on the input

image sequence, as shown in Fig. 1(b), where a graph of width 3 is constructed, considering the images as the nodes of the graph. Here, two nodes (i.e., images) are connected by an edge, if they are within a distance of 3 in the time axis on the sequence S (See Fig. 1(b)).

Our algorithm in MISTICA suitably constructs non-negative edge weights for the graph. For example, between good (i.e., conforming) and poor quality (i.e., nonconforming) images the edge weights would be large, whereas between good quality images, the weights would be smaller. Next, we compute a weighted minimum spanning tree (MST) on the graph shown in Fig. 1(b). An example MST is shown in Fig. 1(c). If one now considers I_4 as the anchor image, we obtain $m = 3$ subsequences: $S_1 = \{I_1, I_2\}$, $S_2 = \{I_3\}$, $S_3 = \{I_6, I_5\}$. Note that I_2 and I_3 appear at the ends of the subsequences S_1 and S_2, and therefore do not propagate registration errors. If we consider I_1 as the anchor image, we obtain $S_1 = \{I_2\}$, $S_2 = \{I_4, I_3\}$, $S_3 = \{I_4, I_6, I_5\}$. Once again, error propagation is prevented.

We note that the subsequences in the generic alignment algorithm can be overlapping. Under such circumstances, the generic algorithm remains well-defined if and only if a condition holds. To define this condition, we consider the notion of an *alignment path* for an image I_j, consisting of a sequence of images that starts at the anchor image I_i and ends at I_j. Suppose, I_j appears in two subsequences S_k and S_l. Then there are two alignment paths for I_j, one each from S_k and S_l. These two different paths lead to ambiguity, because it will yield two different transformations for I_j along these two paths. So, the algorithm is forced to consider only one alignment path, discarding all others, even if more than one path exists. We conclude that the alignment paths for all images *must be unique* for the generic algorithm to be well-defined and will yield a valid alignment result. The uniqueness of alignment paths has a natural algorithmic consequence that the subsequences must now form a tree rooted at the anchor image I_i. This is the reason we construct a spanning tree, viz., a MST to compute these subsequences. In the next section, we also show that when the alignment paths are unique (equivalently, when the subsequences form a tree), the alignment algorithm is the most efficient.

Note that by construction, MST tends to put the nonconforming images as leaf nodes in the tree. Otherwise, a nonconforming image would be an internal node in the tree and MST would have chosen more than one expensive edges coming out of this image– a situation that MST would avoid by all means. Thus, MISTICA decomposes the original image sequence into smaller subsequences and these smaller subsequences contain nonconforming images toward their end. This way, registration error propagation is minimized. Simple template matching method goes to one extreme in that it subdivides the input sequence into the largest number of subsequences, preventing any error propagation due to nonconforming images. Since the sequences are long, our experiences suggest that the anchor image in simple template matching ultimately cannot adapt to the illumination changes and drifts, even with measures such as normalized cross-correlation. The sequential methods, such as StackReg [19], resides on the other extreme end of the spectrum that it divides the input sequence, into at most two subsequences. So, they can adapt well to the drift and change of illumination; however, they fail to handle occurrences of nuisance images. MISTICA achieves a balance between these two extremities.

Automatic computation of the anchor image can be straightforward in MISTICA, which finds out a node midway between the first and the last node on the path in the MST. In the example provided by Fig. 1(c), I_1-I_4-I_6 is the path between I_1 and I_6. Thus, the anchor image would be I_4, by this computation. Since the nuisance images tend to occur toward the end of a subsequence S_k, automatic anchor computation never chooses such an image as the anchor image.

We note that alternatives to MST exists for a choice of a spanning tree, such as a shortest path spanning tree (SPST). However, automatic computation of an anchor image will be computationally more expensive, as a root image would be required to calculate a SPST. Thus, we chose a computationally attractive alternative, MST, in MISTICA.

III. Proposed MST-Based Alignment: MISTICA

The MISTICA algorithm steps are as follows:

Step 1—Construct a graph for the input image sequence (see below).

Step 2—Construct a minimum weighted spanning tree (MST) on the graph.

Step 3—Compute the path on the MST between the first and last images in the input image sequence. Find an image that is midway on this path. Set this image as the anchor image.

Step 4—Do a breadth (or depth) first traversal of the MST starting at the anchor image (root node). Store identity transformation matrix at the root node.

```
While there are nodes in MST yet to visit, do
  Get the next node C in the traversal order.
  Get the parent node P of C.
  Get the image I_C stored at C.
  Get the image I_P stored at P.
  Compute a transformation matrix T to register I_C to I_P.
  Retrieve the transformation matrix T_P stored at node P.
  Update T by multiplying with T_p: T ← T_pT.
  Store T at C.
End While
```

Step 5—Visit each node of the MST in any order to transform the image stored at the node by the corresponding transformation matrix stored at the same node.

The graph construction needs to be detailed. The edge weights are constructed as follows:

$$E_{exact}(I_i, I_j) = \begin{cases} ||I_i - R(I_i, I_j)||, & \text{if } i \neq j \text{ and} |i-j| \leq \delta \\ \infty, & \text{otherwise,} \end{cases} \quad (1)$$

where the function $R(X, Y)$ returns the registered source image Y with respect to the target image X. The norm (||.||) here denotes the sum of absolute differences of pixel values between two images. By convention, no edge with infinite weight is constructed. The sparsity of the graph depends on user-defined the graph width parameter δ. A smaller value of δ enforces greater sparsity. Note that for $\delta = n$-1, we construct a complete graph.

We call the graph constructed with edge weights (1) an *exact* graph, because the edge weights between two images are computed *after* the pair is registered with each other. The overhead with the exact method is dominated by this graph construction, where $O(\delta n)$ image registrations are performed.

We also consider another variant for graph construction, which is computationally more attractive. We call this variant an *approximate* graph, because no image pairs need to be registered for computing the edge weights. This is defined as follows:

$$E_{approx}(I_i, I_j) = \begin{cases} ||sub(I_i) - sub(I_j)||, & \text{if } i \neq j \text{ and } |i-j| \leq \delta \\ \infty, & \text{otherwise,} \end{cases} \quad (2)$$

where the function *sub*(.) downsamples an image, to mitigate the effect of motion in the pixel-wise image difference. For downsampling, we resize images by one fourth in both the horizontal and the vertical direction in all our experiments here. Then an anti-aliasing Gaussian smoothing is applied to the resized image. With the approximate edge weight construction (2), MISTICA achieves the computational lower bound for an image sequence registration, because only n-1 image registration operations are performed. This is the computational lower bound, because every image, except the anchor image, must be registered at least once.

IV. Results

A. Datasets

We performed registration experiments on six microscopy 2D data sets of mouse arteries. The details of the experimental setup and image acquisition have been reported in [8, 9]. These sequences vary in lengths from ~350 to ~1100 images. We denote these six datasets by (a) through (f), going from the top to the bottom, respectively, in Fig. 2, which also displays the first and the last images of these sequences. In addition, we have also performed registration on three 3D microscopy image sequences. 3D image acquisition has been detailed in [9]. For the 3D datasets, the stacks of 2D images form a time series. These 3D stacks are referred to as Z-stacks. Maximum intensity projections of the Z-stacks (also known as max Z projections) are a way to visualize the 3D data [9]. The 3D datasets are referred to as datasets (g), (h) and (i) and their max Z projections are shown in Fig. 3. Dataset (g) has 20 Z stacks with 40 images in each Z stack (a total of 800 images). Dataset (h) has 64 Z stacks with 44 images belonging to each Z stack; so, a total of 2816 images. The longest one is the (i) dataset with 89 Z stacks with 34 images in each Z stack (a total of 3026 images).

B. Competing Methods

For the 2D datasets, we consider a template matching (TM) [2], an adaptive version of the template matching (ATM), StackReg (SR) [19] and Register Virtual Stack Slices (RVS) [1, 14] and descriptor-based series registration (DBSR) [13] as competing methods to MISTICA. In TM, ATM and SR, the first image is chosen as an anchor image, while RVS and DBSR do not need a choice of an anchor image. Incidentally, MISTICA can also choose an anchor image automatically. In TM, the anchor image is registered to every other image and remains fixed during the registration process. In ATM, the anchor image is set as the target image at the beginning, then it changes as a pixel-by-pixel linear combination of the current registered source and itself. It is non-trivial to set this coefficient of linear combination. In all the experiments with ATM, we set this coefficient as 0.5, so that equal emphasis is given to the target and the source images. The frameworks of RVS and SR are similar in that they work on a forward or backward time-sequential order for alignment starting from the anchor image. DBSR uses a global alignment scheme for the images as in [15]. However, the DBSR plugin in Fiji [22] does not use elastic registration, unlike the method in [15]. While TM, ATM, SR, MISTICA all use the intensity-based method TurboReg plug-in [20] in ImageJ, RVS and DBSR use a feature detection-based technique. For the 3D datasets, we consider DBSR and the “Poorman’s 3D registration” method (P3D) of ImageJ, written by Michael Liebling of UCSB [12]. We listed all competing methods in Table I. MISTICA software can be downloaded from: http://webdocs.cs.ualberta.ca/~nray1/MyWebsite/Codes.htm.

C. Experimental Setup

We have considered two registration models in all the experiments: translation and rigid body transformation within the registration function $R(.,.)$. For TM, ATM, SR and MISTICA, we have used the same implementation, TurboReg plugin for $R(.,.)$ in ImageJ. We did not tune any parameters of the TurboReg plugin; we used the default parameter setting for all the methods. For the subsampling function $sub(.)$ in the approximate graph weight computation, we used a downsampling factor of four for both the horizontal and the vertical directions followed by an anti-aliasing Gaussian filter. For RVS method in Fiji, the default parameter setting has been used. For DBSR parameter choices in Fiji, we used brightness of detections as “medium”, approximate size of detection as “10 px”, type of detection as “maxima only”, number of neighbors as 10, redundancy for descriptor matching as 3. Other parameters in DBSR were set at default values.

D. Measuring Registration Accuracy

All the datasets described before are collected *in vivo*. Thus, it is practically impossible to obtain ground truth registration data for the sequences. We can only indirectly measure the registration accuracy. For this purpose, we use structural similarity index (SSIM) [21], which is a well-established quality metric to compare two images. For a fair comparison, for an image sequence $I_1, I_2, \ldots, I_n$, the first image I_1 is considered the anchor image for all the methods. All other images are compared to this anchor image by SSIM scores:

$$s_i = SSIM(I_1, I_i), \text{ for } i = 2, \ldots, n, \quad (3)$$

where, the function $SSIM(.,.)$ compares two images and produces a SSIM score between them. Next, we compute the mean μ and the standard deviation σ of s_i's. A higher value of μ implies a better alignment. In the error bar plots, the mean μ appears as a dot. Each vertical bar has a length of 2σ and is centered at μ.

E. Experiments

We use the abbreviations shown in Table I to denote the methods in comparisons.

Fig. 4 shows the error-bar plots for the six 2D datasets with all the methods. The registration model here is translation. We observe that all variations of MISTICA have produced better results than its competitors. Often MISTICA with approximate edge weights has exceeded MISTICA with exact edge weights for a graph width of 5. This behavior is consistent for all the datasets, because the approximate edge weight constructions had a chance to explore many more registration combinations. Note that we had limited the graph width to 5 for the exact method (1), because of practical computational consideration. We also note that the behavior of ATM is somewhat erratic across datasets, because the coefficient 0.5 for the linear combination may not have been ideal for all the sequences.

Fig. 5 shows results similar to those in Fig. 4, only with the registration model changed to rigid body transformation. This time, we observe that behavior of RVS is not as consistent as Next, we compare RVS method with a feature-based registration within MISTICA with approximate complete graph creation. Table II shows SSIM values for these two methods on the six 2D image sequences. For the feature-based registration in both RVS and MISTICA, SIFT features [7] were used. The transformation model is rigid body transformation here. The results demonstrate that MISTICA is better performing between the two methods. This once again reinforces our assertion that MST-based re-ordering of images in MISTICA helps the registration of the entire image sequence.

In another experiment, we added zero-mean Gaussian noise to images 101 through 200 of sequence (e). Fig. 6 shows two images - one without noise and another with noise. Thus, out of a total of 952 images, the sequence now contains 100 consecutive corrupted images. The mean and standard deviation of SSIM values of the unregistered noisy sequence and registered ones obtained by all the methods are reported in Table III. The transformation model for the registration considered here is translation.

From Table III, we notice that the feature detection-based RVS method could not generate any result for the sequence, because no reliable features could be detected for the corrupted images. Besides, within RVS there is no mechanism to reject these corrupted images. SR method gave poor registration performance, because the sequential nature of the algorithm accumulated excessive registration errors, throwing good images almost out of the field of view (also see supplementary videos provided at: http://webdocs.cs.ualberta.ca/~nray1/MyWebsite/Codes.htm). Being a global method on sequence alignment, the performance of

DBSR was only third after the two MISTICA methods (AMST100 and AMST200). AMT produced a performance that is close or similar to the unregistered input sequence (UR). Among the sequential registration methods other than MISTICA, TM produced the best performance, because it completely decoupled the input image sequence into individual subsequences, each consisting of a single image and was able to withstand the error propagation. However, as we discussed before, this extreme end of subsequence generation by TM does not work well, because, as we go further away (in time axis) from the anchor image, the intensity and field of view may also change considerably, making the registration process harder.

We considered three different parameter settings in MISTICA with approximate edge weights that produced SSIM values better than its competitors and the unregistered sequence. This experiment is in agreement with our assertion that MISTICA is able to handle spells of poor quality images quite well.

MISTICA includes a single parameter– the graph width, δ. An ideal graph would be a complete graph ($\delta = n$-1) with the exact edge weights. Because of the computational burden, choice of δ in practice is limited to a lower value, such as 5, for the exact graph edge weight construction. For the approximate method of edge weight construction, we can easily construct a complete graph even for a 1000-image long sequence. However, the approximate method does not guarantee that with a complete graph, we will obtain the best registration performance. To see the effect of δ on the registration performance, we plotted errors with error-bar plots in Fig. 7 for the image sequence (e) with translation used as transformation. From these error-bar plots, we observe that around δ =100, we obtain the best accuracy. However, the performance does not fall very quickly on larger values of δ. Also, for a wide range of δ (between 50 and 100), the registration performance is not as sensitive.

One can also consider a coarse-to-fine/pyramid approach with δ in MISTICA with approximate graph construction. For example, we can start the registration with a complete graph $\delta = n$-1. Next, on the registered sequence, MISTICA is run with $\delta = (n\text{-}1)/2$. This way the MISTICA can be run approximately $log_2(n)$ times. Thus, with the help of additional computations, one can waive the need for choosing a graph width parameter value. We ran this experiment on the 2D dataset (e) with translation as transformation. The second row of Table IV shows the registration performance by running the aforementioned pyramid scheme. The third row shows the performance of running exact MST5 after the pyramid AMST. The SSIM values show considerable gain compared to those shown in Fig. 4. As an explanation of the performance gain, we offer that with the pyramid approach we move from less restrictive to a more restrictive graph structure, progressively fine-tuning the registration.

Application of MISTICA to 3D datasets does not require any special extensions or constructions. However, in this case, MISTICA needs a complete and approximate graph for computational considerations. Thus, MISTICA selects an anchor image on the minimum spanning tree that is midway between the first image in the first Z-stack and the last image in the last Z-stack. Notice that this working principle of MISTICA applies in-plane 2D transformations to the 3D dataset for the course alignment, as opposed to a full 3D

transformation. After this alignment, an optional post processing can be made. Thus, we construct the max Z-projections and apply MISTICA with complete and approximate graph to this 2D sequence of max Z-projections. Next, the transformation for each max Z-projections is applied to all the 2D images belonging to the same Z-stack. This post processing closely resembles the working principle of Poorman's 3D registration (P3D) method [12] and that briefly mentioned in [11], which also apply in-plane transformations as opposed to full 3D transformations.

Tables V, VI and VII show the comparative results of these methods on the 3D datasets (g), (h) and (i), respectively. 2-photon 3D microscopy images only highlight a certain Z-section in the 3D. Thus, for numerical comparisons, SSIM values are computed on the Max Z projection images and reported in these tables. The symbols, "T" and "R," within parenthesis signifies translation and rigid body transformations, respectively. The symbol "+PP" signifies that the aforementioned MISTICA-based post processing has been applied. We observe that all the methods have been able to raise the SSIM values significantly compared to those of the unregistered sequences. Even though MISTICA did not apply full 3D transformations, its performance is often superior to that of the DBSR method, which is a global method applying full 3D transformations. The Tables V through VII also show that MISTICA-based post processing has been helpful. However, without the post processing MISTICA is still competitive.

Lastly, we examined the computation time required for each of these methods, as shown in Table VIII. All these methods ran ImageJ plugins on an Intel 3.4 GHz, 16GB Windows PC. We notice that MISTICA with approximate graph weight is 50% slower than TM, ATM and SR. The extra overhead has been spent in constructing a complete graph for the image sequence (e). DBSR, being a global method of sequence registration has been the slowest of all the methods reported here.

V. Conclusions

We have proposed and implemented a new algorithm, MISTICA, for coarse alignment of time sequenced images. MISTICA has been designed to handle occurrences of poor quality images in long sequences, including noisy or low-intensity data, as well as images with distortion due to tissue motion. We achieved this by taking advantage of the observation that an alignment algorithm does not have to work with the natural temporal order in which images were acquired. An algorithm can *re-order the images* suitably and align those. A computationally efficient framework with minimum spanning trees has been proposed here. In fact, the MST-based algorithm, MISTICA, achieves a computational lower bound for the image sequence alignment problem. We have shown experimentally that MISTICA works well in practice. MISTICA is straightforward in use, as it only has one parameter, the graph width, to choose; the graph width parameter is adequate for robust performance over a wide range of scenarios. Finally, the anchor selection is also completely automatic in MISTICA.

Acknowledgments

N. Ray acknowledges NSERC discovery grant as funding support.

Biographies

Nilanjan Ray received bachelor in mechanical engineering from Jadavpur University, Kolkata, India, in 1995, M.Tech. in computer science from the Indian Statistical Institute, Kolkata, India in 1997, and Ph.D. in electrical engineering from the University of Virginia, Charlottesville, USA in 2003. After having two years of postdoctoral research and a year of industrial work experience, Nilanjan joined the department of Computing Science, University of Alberta in 2006, where he is an Associate Professor. Nilanjan's research area is image and video analysis– analytics, segmentation, object detection, image classification, image registration, and object tracking. Nilanjan has over 100 publications in these research areas. Nilanjan serves as an Associate Editor for IEEE Transactions on Image Processing.

Sara McArdle received a BS in Biomedical Engineering in 2009 from Columbia University. She performed her doctoral research at the La Jolla Institute for Allergy and Immunology with Dr. Klaus Ley, and received her PhD in Bioengineering, with a specialization in multi-scale biology, from the University of California, San Diego in 2015.

Scott T. Acton (Fellow, 2013) is Professor of Electrical & Computer Engineering and of Biomedical Engineering at the University of Virginia. He received his M.S. and Ph.D. degrees at the University of Texas at Austin. He received his B.S. degree at Virginia Tech. He is a Fellow of the IEEE.

Professor Acton's laboratory at UVA is called VIVA - Virginia Image and Video Analysis. They specialize in biological and biomedical image analysis problems. The research emphases of VIVA include tracking, segmentation, representation, retrieval, classification and enhancement. Professor Acton has over 250 publications in the image analysis area including the books Biomedical Image Analysis: Tracking and Biomedical Image Analysis: Segmentation. Professor Acton is the Editor-in-Chief of the IEEE Transactions on Image Processing.

Dr. Klaus Ley, M.D. is Professor and Head of the Division of Inflammation Biology at the La Jolla Institute for Allergy and Immunology and Adjunct Professor of Bioengineering at the University of California, San Diego. He trained at Julius-Maximilians-Univer¬sität in Würzburg, Germany and was a Professor of Bioengineering at the University of Virginia and Director of their Robert M. Berne Cardiovascular Research until 2007. Klaus Ley studies chronic inflammatory diseases like atherosclerosis and inflammatory bowel disease with a focus on myeloid leukocytes (cells of the innate immune system). Recently, he started a large new research program with the goal of developing a vaccine to prevent atherosclerosis. He has 30 years of experience in inflammation research and published more than 300 papers in peer-reviewed journals (h-factor 109). He received more than $ 30 million in research funding from the National Institutes of Health. He is the recipient of the 2008 Bonazinga Award, the highest award of the Society for Leukocyte Biology and the 2010 Malpighi Award, the highest award of the European Society for Microcirculation. In 2014, he was president of the North American Vascular Biology Organization and has organized Keystone and Gordon Research Conferences.

References

1. Arganda-Carreras I, Sorzano COS, Marabini R, Carazo J-M, Ortiz-de-Solorzano C, Kybic J. Consistent and Elastic Registration of Histological Sections using Vector-Spline Regularization. LNCS, Comput Vis Approaches to Med Imag Anal. 2006; 4241:85–95.
2. Brunelli, R. Template Matching Techniques in Computer Vision: Theory and Practice. Wiley; 2009.
3. Cormen, TH., Leiserson, CE., Rivest, RL., Stein, C. Introduction to Algorithms. 2. MIT Press and McGraw-Hill; 2001.
4. Dunne J, Goobic AP, Acton ST, Ley K. A novel method to analyze leukocyte rolling behavior *in vivo* in the mouse cremaster muscle using intravital microscopy. Biological Procedures. 2004; 6:182–189.
5. [accessed on July 28, 2015] http://fiji.sc/Fiji
6. Goobic AP, Tang J, Acton ST. Image stabilization and registration for tracking cells in the microvasculature. IEEE Transactions on Biomedical Engineering. 2005; 52:287–289. [PubMed: 15709666]
7. Lowe DG. Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision. 2004; 60(2):91–110.

8. McArdle, S., Acton, ST., Ley, K., Ray, N. Registering sequences of in vivo microscopy images for cell tracking using dynamic programming and minimum spanning trees. IEEE International Conference on Image Processing; Paris, France. October 27–31, 2014;
9. McArdle S, Chodaczek G, Ray N, Ley K. Intravital live cell triggered imaging system (ILTIS) reveals monocyte patrolling and macrophage migration in atherosclerotic arteries. Journal of Biomedical Optics. 2015; 20(2):026005.
10. McLauchlan PF, Jaenicke A. Image mosaicing using sequential bundle adjustment. Image and vision computing. 2002; 20(9–10):751–759.
11. Ozeré, S., Bouthemy, P., Spindler, F., Paul-Gilloteaux, P., Kervrann, C. Robust parametric stabilization of moving cells with intensity correction in light microscopy image sequences. Proc. IEEE Int. Symp. on Biomedical Imaging (ISBI'13); San-Francisco, CA. April 2013;
12. [accessed on July 28, 2015] Poorman's 3D registration. http://sybil.ece.ucsb.edu/pages/poorman3dreg/index.html
13. Preibisch S, Saalfeld S, Schindelin J, Tomancak P. Software for bead-based registration of selective plane illumination microscopy data. Nature Methods. 2010; 7:418–419. [PubMed: 20508634]
14. [accessed January 22, 2015] Register Virtual Stack Slices. http://fiji.sc/wiki/index.php/Register_Virtual_Stack_Slices
15. Saalfeld S, Fetter R, Cardona A, Tomancak P. Elastic volume reconstruction from series of ultra-thin microscopy sections. Nature Methods. 2012; 9:717–720. [PubMed: 22688414]
16. Sorokin, D., Tektonidis, M., Rohr, K., Matula, P. Non-rigid countour-based temporal registration of 2D cell nuclei images using the Navier equation. IEEE 11th International Symposium of Biomedical Imaging (ISBI'14); Beijing, China. April 2014;
17. Sorzano C, Thevenaz P, Unser M. Elastic Registration of Biological Images using Vector-Spline Regularization. IEEE Trans Biomed Engin. 2005; 52:652–663.
18. Thévenaz P, Ruttimann UE, Unser M. A Pyramid Approach to Subpixel Registration Based on Intensity. IEEE Transactions on Image Processing. Jan; 1998 7(1):27–41. [PubMed: 18267377]
19. Thévenaz, P. [accessed on January 22, 2015] StackReg, An ImageJ plugin for the recursive alignment of a stack of images. http://bigwww.epfl.ch/thevenaz/stackreg/
20. Thévenaz, P. [accessed on January 24, 2015] TurboReg, An ImageJ plugin for the automatic alignment of a source image or a stack to a target image. http://bigwww.epfl.ch/thevenaz/turboreg/, distribution July 7, 2011
21. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP. Image quality assessment: From error visibility to structural similarity. IEEE Transactions on Image Processing. 2004; 13(4):600–612. [PubMed: 15376593]
22. Wang CW, Ka SM, Chen A. Robust image registration of biological microscopic images. Scientific Report. Aug 13.2014 4

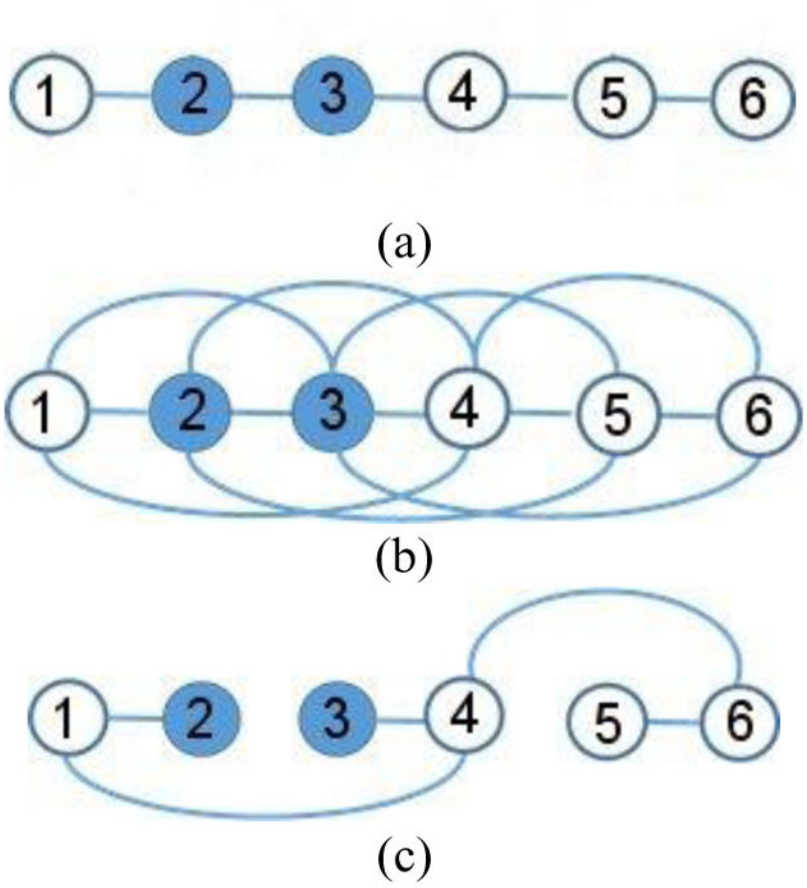

Fig. 1.

(a) A representation of a time-sequenced set of six images. Shaded circles depict poor quality or nonconforming images. (b) A graph constructed on the image sequence. (c) An example minimum weighted spanning tree.

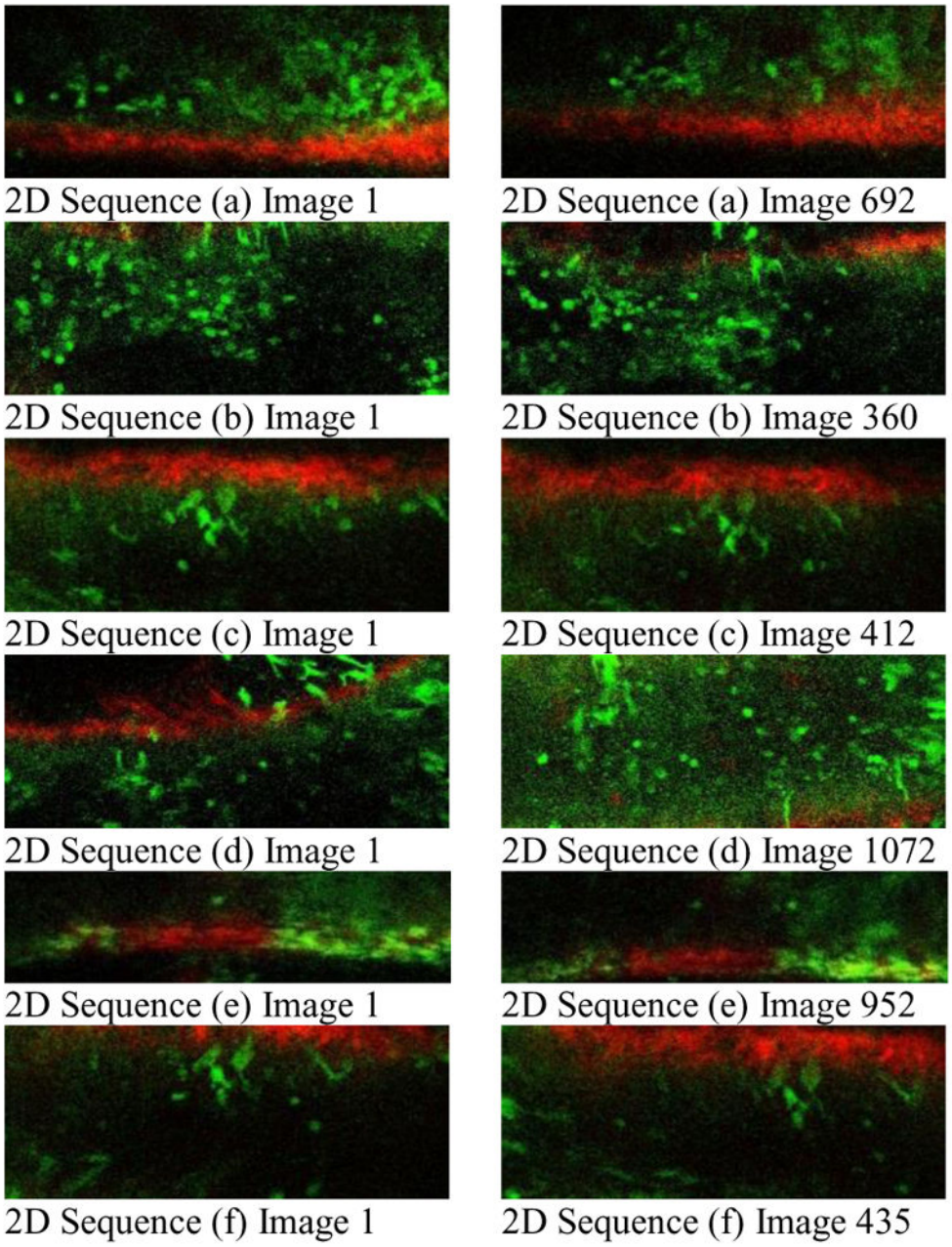

Fig. 2.

Six rows from top to bottom show 2D datasets (a) though (f), respectively. The first and the second columns show the first and the last images in the sequences, respectively. Consequently, the second column also shows the lengths of the sequences.

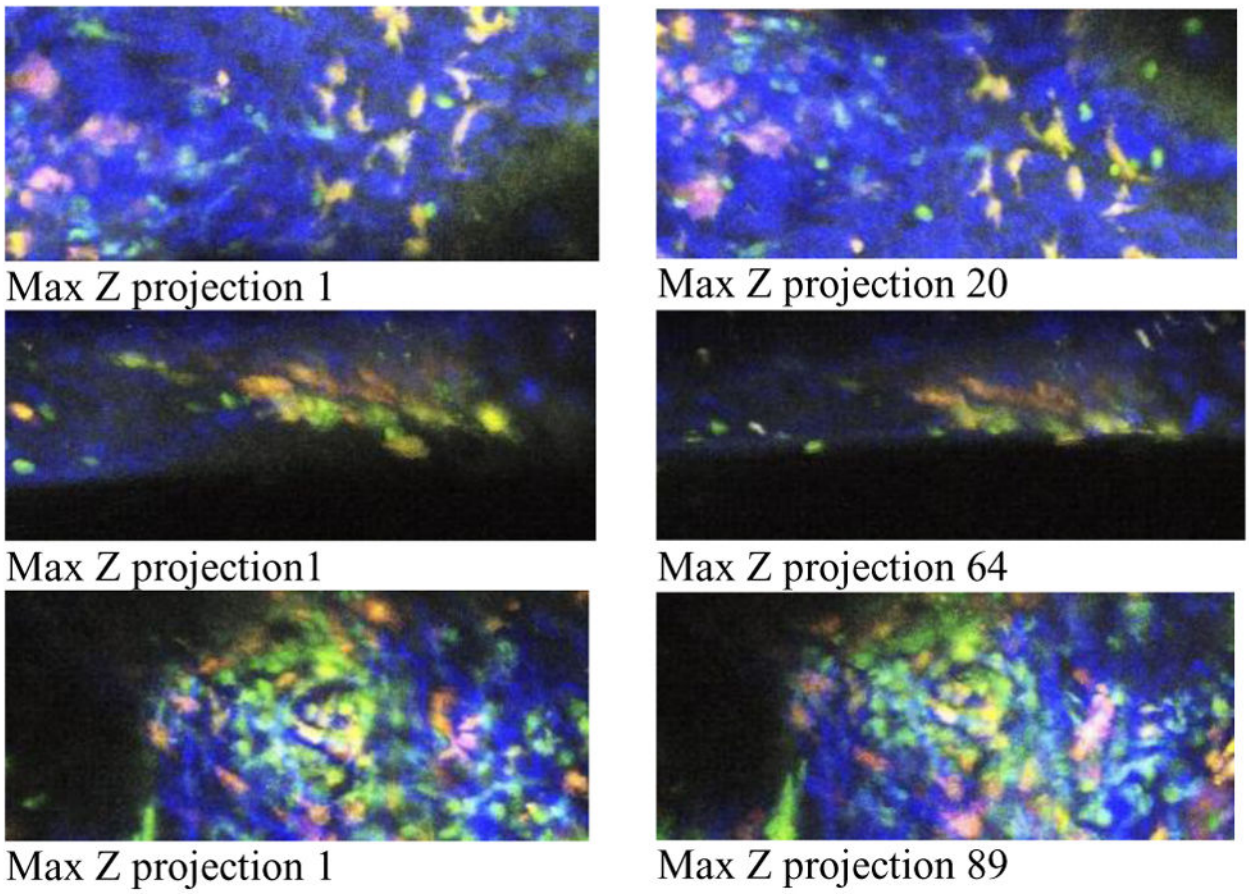

Fig. 3.

Three rows from top to the bottom show 3D datasets (g), (h) and (i), respectively. The first and the second columns show the first and the last Max Z-projections as a 2D way to visualize 3D image sequences from 2-photon microscopy [9]. These images are 512 pixels by 220 pixels.

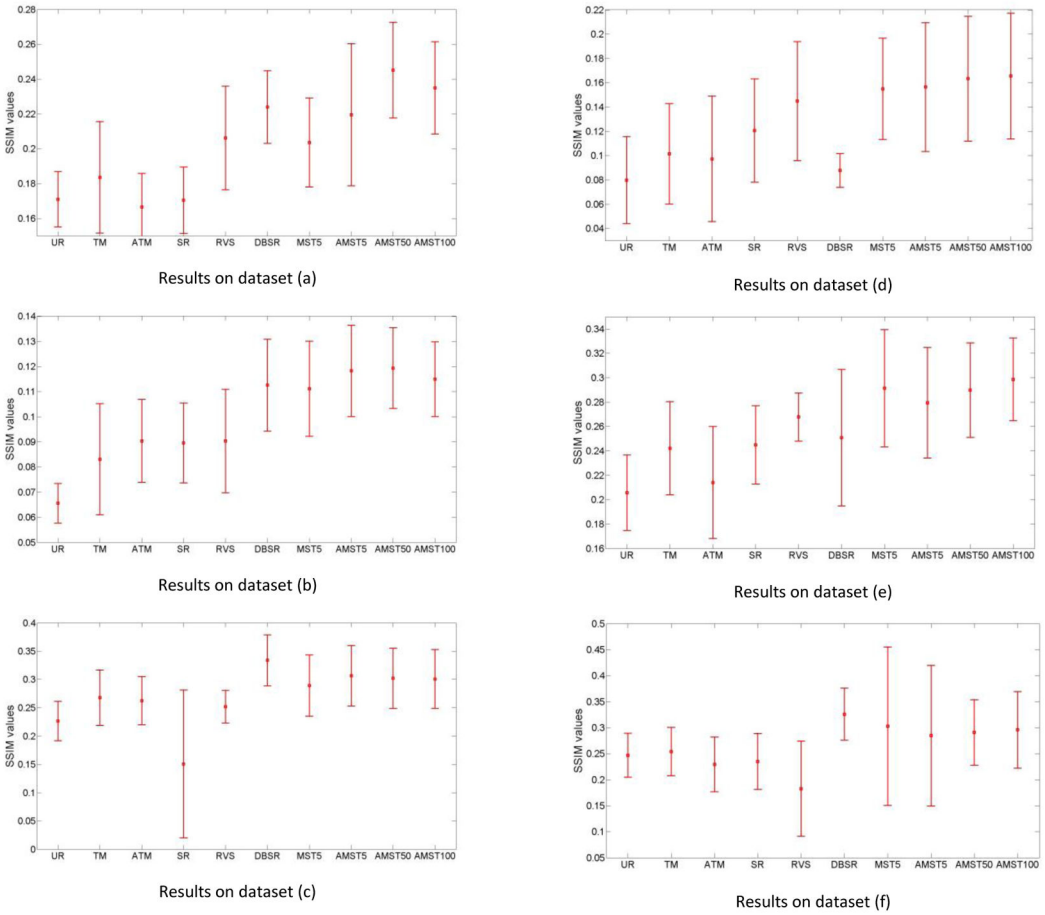

Fig. 4.

Mean SSIM values ± SD obtained for 2D sequences (a) through (f) registered with different methods. Translation is used for all the methods.

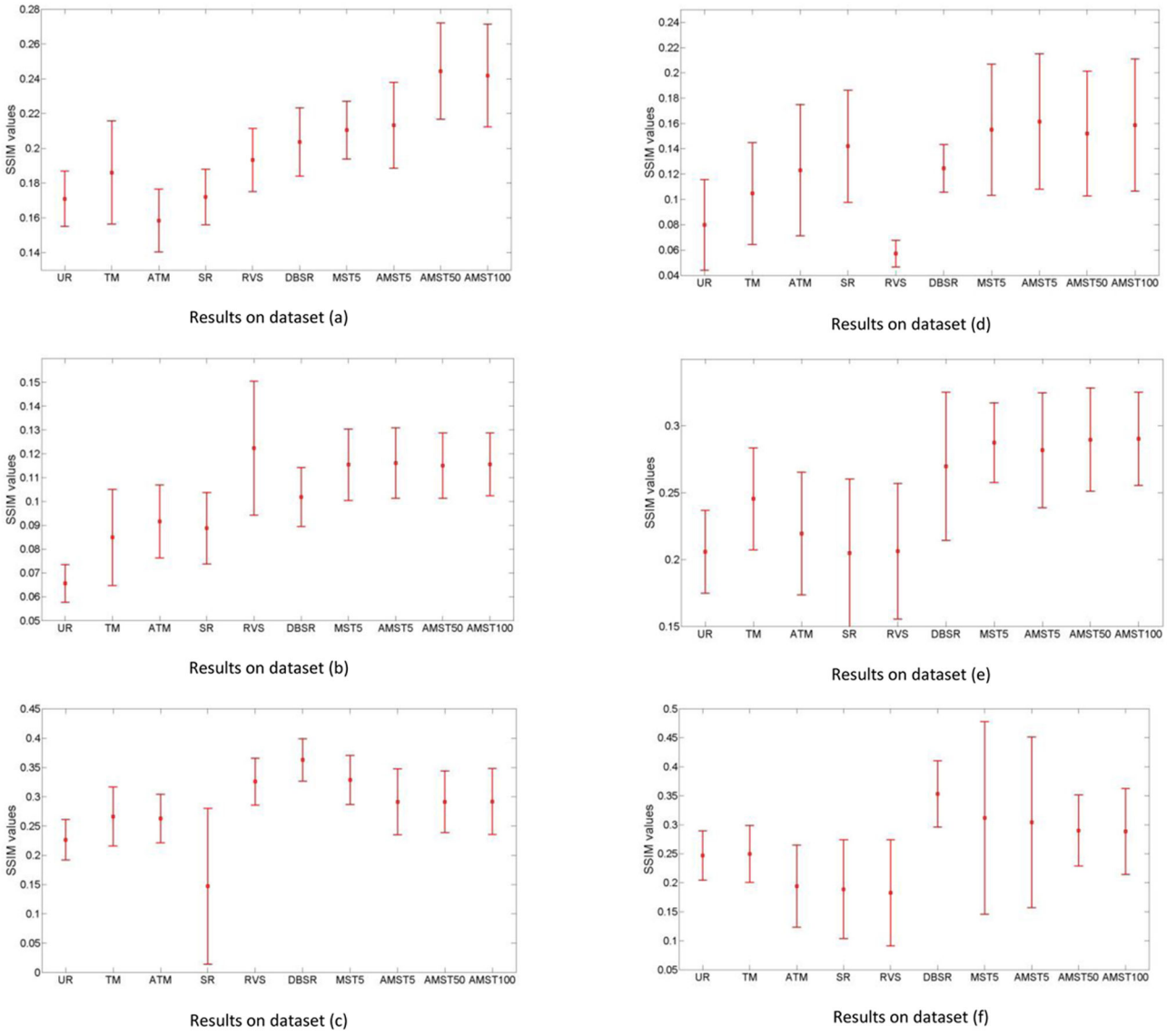

Fig. 5.

Mean SSIM values ± SD for 2D sequences (a) through (f). Rigid body transformation is used for all the methods.

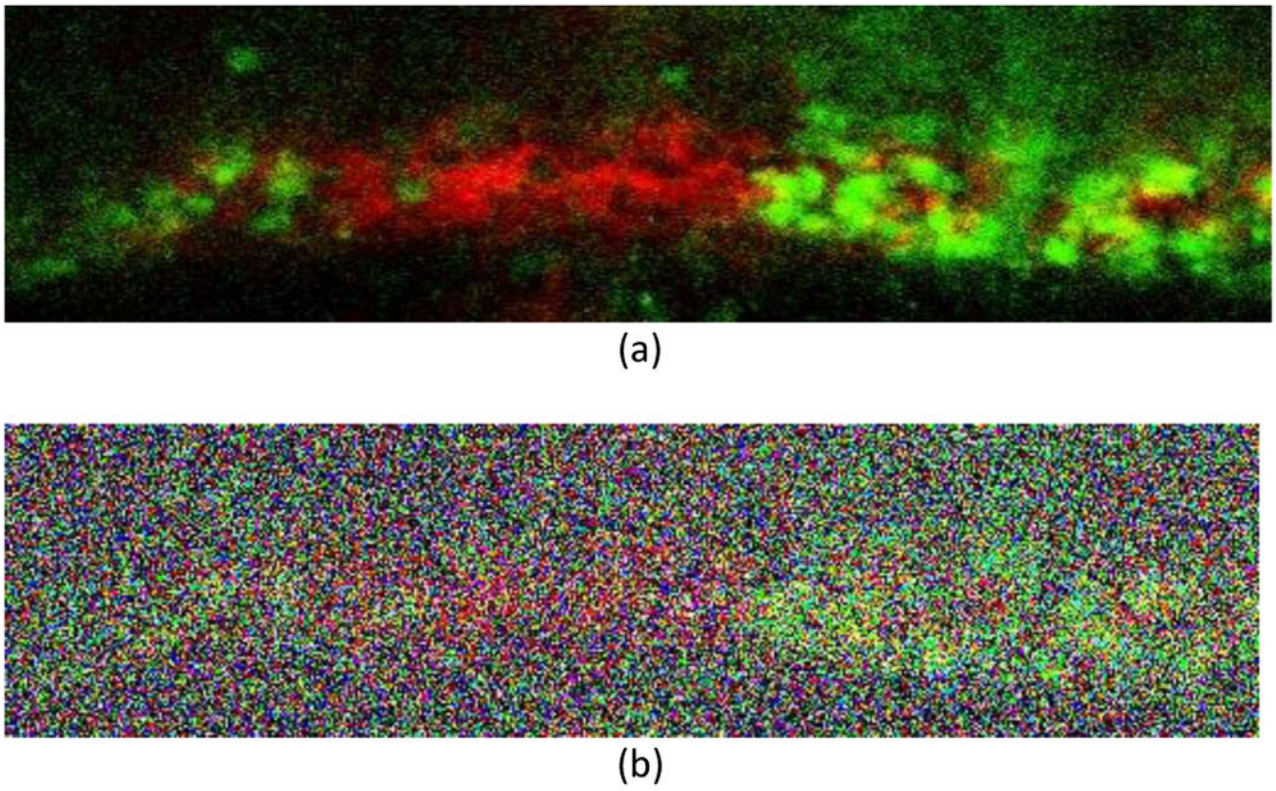

Fig. 6.
Adding noise to the dataset (e). (a) Image 150. (b) Image 150 after adding Gaussian noise.

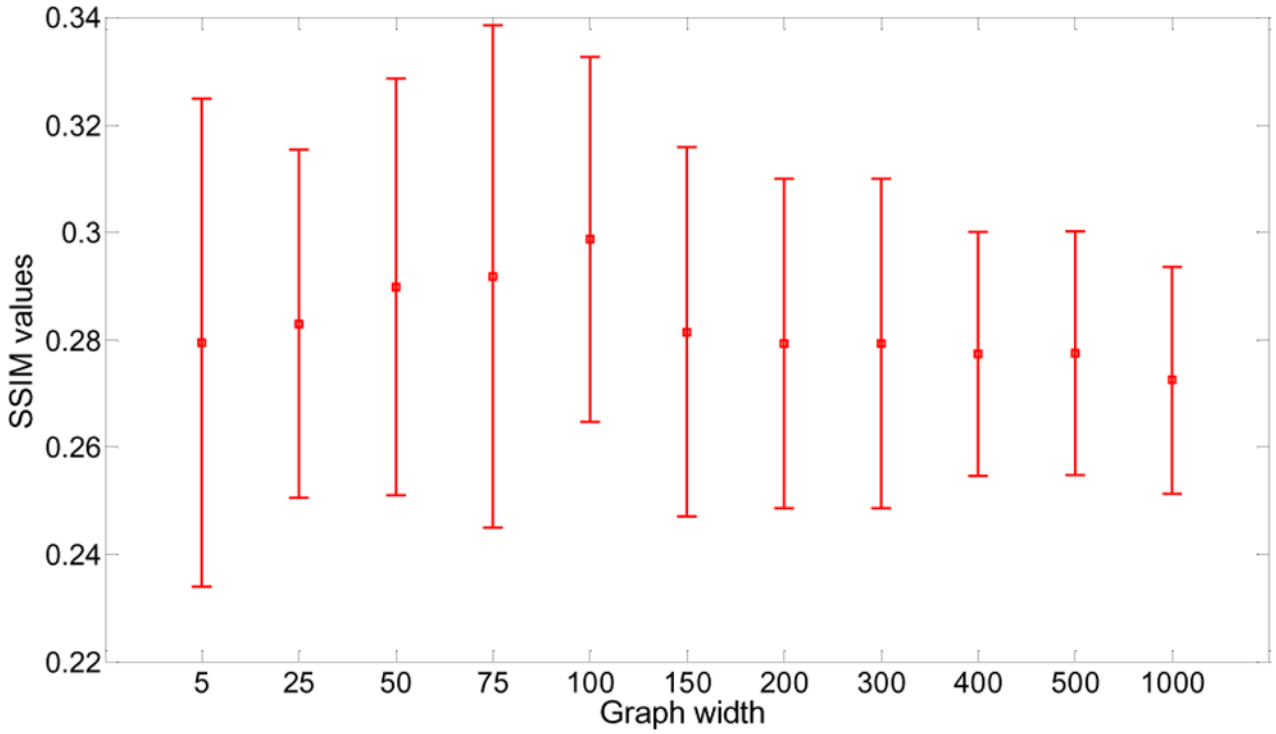

Fig. 7.

Mean SSIM values ± SD obtained by MISTICA with different graph widths on dataset (e). A graph width of 1000 meant a complete graph, because the length of the image sequence is 952 here. The transformation used was translation.

TABLE I

Abbreviation used for methods in comparison

Abbreviation	Name of Method
UR	Unregistered sequence
TM	Template matching
ATM	Adaptive template matching
SR	StackReg plugin in ImageJ [19]
RVS	Register virtual stack slices plugin in Fiji [14]
DBSR	Descriptor-based series registration in Fiji [13]
P3D	Poorman's 3D registration method of ImageJ [12]
MST*n*	MISTICA with exact graph weight construction (1); graph width = *n*
AMST*n*	MISTICA with approximate graph weight construction (2); graph width = *n*

TABLE II

FEATURE-BASED MISTICA VERSUS RVS

2D Sequence	RVS SSIM (mean ± std)	Feature-based MISTICA (mean ± std)
(a)	0.1933 ± 0.0182	0.2243 ± 0.0232
(b)	0.1224 ± 0.0281	0.1268 ± 0.0196
(c)	0.3259 ± 0.0402	0.3514 ± 0.0411
(d)	0.0571 ± 0.0105	0.0903 ± 0.0380
(e)	0.2062 ± 0.0508	0.2954 ± 0.0540
(f)	0.1825 ± 0.0916	0.3203 ± 0.0605

TABLE III

SSIM MEAN AND STANDARD DEVIATION VALUES OBTAINED ON REGISTERED NOISY IMAGE SEQUENCE

Method	Mean of SSIM Values (μ)	Std. Deviation of SSIM Values (σ)
UR	0.1817	0.0658
AMST50	0.2252	0.2132
AMST100	0.2594	0.0867
AMST200	0.2627	0.0809
SR	0.0783	0.0789
RVS	N/A	N/A
AMT	0.1858	0.0727
TM	0.2140	0.0761
DBSR	0.2390	0.1047

TABLE IV

SSIM MEAN AND STANDARD DEVIATION VALUES OBTAINED WITH PYRAMID SCHEMES

Method	Mean of SSIM Values (μ)	Standard Deviation of SSIM Values (σ)
Pyramid AMST	0.3602	0.0287
Pyramid AMST + MST5	0.3669	0.0286

TABLE V

SSIM MEAN AND STANDARD DEVIATION FOR 3D SEQUENCE (g)

Method	Mean of SSIM Values (μ)	Standard Deviation of SSIM Values (σ)
Unregistered	0.1170	0.0742
AMST (T)	0.2967	0.0554
AMST (T) + PP	0.3358	0.0578
P3D (T)	0.2612	0.0308
DBSR (T)	0.3064	0.0746
AMST (R)	0.3087	0.0537
AMST (R) + PP	0.3506	0.0583
P3D (R)	0.2643	0.0340
DBSR (R)	0.2154	0.0830

TABLE VI

SSIM MEAN AND STANDARD DEVIATION FOR 3D SEQUENCE (h)

Method	Mean of SSIM Values (μ)	Standard Deviation of SSIM Values (σ)
Unregistered	0.1331	0.0404
AMST (T)	0.2928	0.0831
AMST (T) + PP	0.3432	0.0807
P3D (T)	0.2349	0.0475
DBSR (T)	0.2770	0.0988
AMST (R)	0.3075	0.0800
AMST (R) + PP	0.3809	0.0665
P3D (R)	0.2229	0.0450
DBSR (R)	0.2535	0.0573

TABLE VII

SSIM MEAN AND STANDARD DEVIATION FOR 3D SEQUENCE (i)

Method	Mean of SSIM Values (μ)	Standard Deviation of SSIM Values (σ)
Unregistered	0.1258	0.0146
AMST (T)	0.3322	0.0532
AMST (T) + PP	0.3735	0.0471
P3D (T)	0.2825	0.0386
DBSR (T)	0.2433	0.1134
AMST (R)	0.2988	0.0426
AMST (R) + PP	0.4163	0.0455
P3D (R)	0.2947	0.0377
DBSR (R)	0.3641	0.0623

TABLE VIII

Computation time

Method	Average computation time per image of size 128-by-512
AMST – complete graph	0.15s
MST5	0.75s
SR	0.1s
RVS	0.34s
ATM	0.1s
TM	0.1s
DBSR	3.15s