BMC Bioinformatics

RESEARCH ARTICLE Open Access

Visualization and correction of automated segmentation, tracking and lineaging from 5-D stem cell image sequences

Eric Wait[1], Mark Winter[1], Chris Bjornsson[2], Erzsebet Kokovay[3], Yue Wang[2], Susan Goderie[2], Sally Temple[2] and Andrew R Cohen[1*]

Abstract

Background: Neural stem cells are motile and proliferative cells that undergo mitosis, dividing to produce daughter cells and ultimately generating differentiated neurons and glia. Understanding the mechanisms controlling neural stem cell proliferation and differentiation will play a key role in the emerging fields of regenerative medicine and cancer therapeutics. Stem cell studies *in vitro* from 2-D image data are well established. Visualizing and analyzing large three dimensional images of intact tissue is a challenging task. It becomes more difficult as the dimensionality of the image data increases to include time and additional fluorescence channels. There is a pressing need for 5-D image analysis and visualization tools to study cellular dynamics in the intact niche and to quantify the role that environmental factors play in determining cell fate.

Results: We present an application that integrates visualization and quantitative analysis of 5-D (*x*, *y*, *z*, *t*, *channel*) and large montage confocal fluorescence microscopy images. The image sequences show stem cells together with blood vessels, enabling quantification of the dynamic behaviors of stem cells in relation to their vascular niche, with applications in developmental and cancer biology. Our application automatically segments, tracks, and lineages the image sequence data and then allows the user to view and edit the results of automated algorithms in a stereoscopic 3-D window while simultaneously viewing the stem cell lineage tree in a 2-D window. Using the GPU to store and render the image sequence data enables a hybrid computational approach. An inference-based approach utilizing user-provided edits to automatically correct related mistakes executes interactively on the system CPU while the GPU handles 3-D visualization tasks.

Conclusions: By exploiting commodity computer gaming hardware, we have developed an application that can be run in the laboratory to facilitate rapid iteration through biological experiments. We combine unsupervised image analysis algorithms with an interactive visualization of the results. Our validation interface allows for each data set to be corrected to 100% accuracy, ensuring that downstream data analysis is accurate and verifiable. Our tool is the first to combine all of these aspects, leveraging the synergies obtained by utilizing validation information from stereo visualization to improve the low level image processing tasks.

Keywords: Volumetric texture rendering, 3-D display, Stereoscopic 3-D, Stem cell, Time lapse, Lineaging, Validation and correction, Confocal microscopy, CUDA, Image montage reconstruction

*Correspondence: acohen@coe.drexel.edu
[1]Drexel University, 19104 Philadelphia, USA
Full list of author information is available at the end of the article

Background

Neural stem cells (NSCs) are motile and proliferative cells that undergo mitosis, dividing to produce daughter cells and ultimately producing differentiated neurons and glia. Understanding the mechanisms controlling NSC migration and proliferation will play a key role in the emerging field of regenerative medicine and in cancer therapeutics. All of the cells in a clone are genetic copies of the original stem cell. Image-based analysis of static 3-D images demonstrated the important relationship between neural stem cells and blood vessels, and the propensity of both adult and embryonic NSCs to seek out and maintain distinct spatial relationships with respect to vasculature known as their vascular niche [1-3]. Confocal and multiphoton microscopes that contain integrated incubation systems are able to image live NSCs together with blood vessels in intact tissue slices, with 5-D image stacks (x, y, z, t, λ) captured at intervals (*e.g.* 20 min.) over a period of 16–20 hours. Here, λ represents spectral information from a fluorescent label. By labeling the blood vessels and the NSCs with different fluorescent markers, these microscopes are able to capture image sequence data that show the dynamic behaviors of migrating proliferating NSCs while simultaneously capturing the relationship to other structures including blood vessels. We have developed an application that for the first time enables the use of time-lapse microscopy data to quantify the dynamic relationship between clones of mammalian NSCs and their niche in intact tissue containing vasculature and live proliferating cells.

The analysis of clones of migrating proliferating NSCs starts with *segmentation*, the delineation of individual cells in each image frame. *Tracking* then establishes temporal correspondences between segmentation results. Finally, *lineaging* establishes parent-daughter relationships across mitotic events. The analysis of stem cell clonal dynamics to date has consisted primarily of extracting and analyzing a *lineage tree* generated from cultured cells. A lineage tree is a graphical representation showing each cell's division time and the offspring it produces. Each daughter cell is a genetic copy of its parent cell. A lineage tree is often referred to as a *clone* of stem cells. Lineages also indicate the population dynamics of clones of stem cells, showing the lifespan and parentage of each cell in the clone, as well as indicating the phenotype of differentiated progeny. These trees summarize patterns of division (*i.e.* symmetric or asymmetric, cell cycle time, number of divisions, *etc.*) and differentiation in a single view. The lineage tree is a key tool in the analysis of stem cell clonal dynamics. In addition to tree level features, we can analyze cellular properties such as motion and morphology using tools such as Algorithmic Information Theoretic Prediction and Discovery (AITPD) [4,5]. AITPD analyzes the patterns of cellular dynamic behavior for individual cells established by segmentation, tracking and lineaging. It can accurately predict development potential for individual NSCs. Previously, we have shown that software in conjunction with AITPD enables the search for behavioral markers of different functional subtypes as well as the potential discovery of molecular mechanisms controlling stem cell proliferation [6]. There is a pressing need for new approaches to visualize and validate 5-D image sequence data of proliferating mammalian cells to enable quantitative analysis of the mechanisms controlling cellular proliferation and differentiation.

While it is possible to analyze the dynamic behaviors of stem cells in a manner that is robust to segmentation errors, ***any*** *errors in tracking or lineaging are likely to corrupt all subsequent analysis*. For *in vitro* phase contrast time lapse image sequence data (2-D) we recently developed a software tool called LEVER that allows a biologist to run automated segmentation, tracking, and lineaging on image sequence data in the laboratory [6]. LEVER displays the lineage tree in one window, while the image sequence data with segmentation and tracking results overlaid are displayed in a second window. Navigation and editing can be done on either window. The interface is designed so that users are able to easily identify and quickly correct any mistakes in the automated image analysis. We have found in this work and in previous studies that the vast majority of errors in tracking and lineaging are the result of segmentation errors [5,6]. These errors happen often when the number of cells in a given area have been incorrectly estimated. LEVER uses an inference-based learning approach, which propagates user-supplied corrections forward to reduce errors on later frames. Here we present an application called LEVER 3-D which displays image data in full stereoscopic 3-D and includes a utility to reconstruct 3-D image montages with the *intent to be run in the biology lab*. LEVER 3-D uses commodity gaming hardware to implement a high-performance interactive system for validating and correcting the automated image analysis results for 5-D stem cell data. This will allow the biologist to better understand stem cell dynamics and regulation within the neural stem cell micro-environment.

Methods

A total of 18 5-D image sequences showing adult mouse neural stem cells, ependymal cells, and vasculature were analyzed. The stem cells were imaged within a 3D wholemount explant culture of the subventricular zone (SVZ) of the brain. Each 5-D voxel location is specified as (x, y, z, t, λ) where λ refers to a multichannel fluorescence signal, one channel imaging NSCs, the second channel containing ependymal cells and vasculature. The movies were captured on two microscopes at two different laboratories. SVZ whole-

mounts were dissected under a dissection microscope as described previously [7,8] from transgenic mice that express green fluorescent or tomato red fluorescent protein in neural stem cells (FVB/N-Tg(GFAPGFP) 14Mes/J, the Jackson Laboratory; Ascl1$^{tm1.1(cre/ERT2)}$; B6.Cg-Gt(ROSA)$^{26Sortm9(CAG-tdTomato)}$, NSCI). Briefly, the brain was removed and halved and the cortex was peeled back to reveal the SVZ. A scalpel was used to make a 2–4 mm cut on the striatal side of the SVZ and watchmaker forceps were used to clip off the SVZ at the anterior and posterior sides and carefully transferred into phosphate buffered saline containing 5 μg/ml Alexa Fluor 647 conjugated Isolectin GS-IB4 (Life Technologies) on ice for 20 minutes to label the ependyma and blood vessels. SVZ wholemounts were transferred SVZ side down to a glass bottomed culture dish (MatTek Corporation) and immobilized by covering with cold (4°C) growth factor free Matrigel (BD Biosciences) and immediately transferred to an incubator set at 37°C with 5% CO2 for 20 minutes to allow the Matrigel to solidify. Freshly made slice culture medium was added to the culture dish and the dish was placed on a Zeiss LSM780 confocal microscope equipped with an environmental enclosure set at 37°C and 5% CO_2. The pinhole was set at 2 AU and Z stacks (25 1 μm steps) were collected every 20 minutes using a 20X objective for 16 hours. Spatial resolution was 512×512, at a pixel spacing of 0.8 μm for a total of 1.3 GBytes of image data at 32 bits per voxel (BPV). At this resolution, the entire image sequence can be downloaded to the video RAM on a 1.5 GB card, with room left for the frame buffer and back buffer. Images were also captured on a Zeiss LSM510 confocal microscope at a spatial resolution of 1024×1024 at a pixel spacing of 0.3 μm for up to 20 hours, resulting in as much as 5 GBytes of image data per sequence. These larger sequences require the image sequence data to be buffered in system memory, a task that is handled automatically by the display driver. Interestingly, less proliferation was observed in image sequences captured at the higher spatiotemporal resolution. Once the 5-D image sequence data has been captured it is imported directly from the microscope output file using the open source Bioformats application.

Processing of the 5-D image sequence data begins by using the open-source BioFormats [9] tool to convert the native microscope data (Zeiss LSM file) into intensity valued tiff images. The use of BioFormats enables LEVER 3D to work not only with the Zeiss specific file formats, but with file formats used by most of the major microscope manufacturers. In addition to the image data, BioFormats extracts the imaging settings including the spatiotemporal resolution used to account for image anisotropy, providing scaling for the tracking and distance-based calculations.

Figure 1 illustrates the flow of data and processing steps used to go from the raw input image sequence data to a fully validated and corrected clonal tracking and lineaging. Figure 1 also illustrates the main software components used including CUDA for efficient image processing from C++, MATLAB for 2-D visualization of the lineage tree and data analysis and export, and Direct 3D for 3-D rendering and visualization. Each of these steps is described in more detail throughout the remainder of this section.

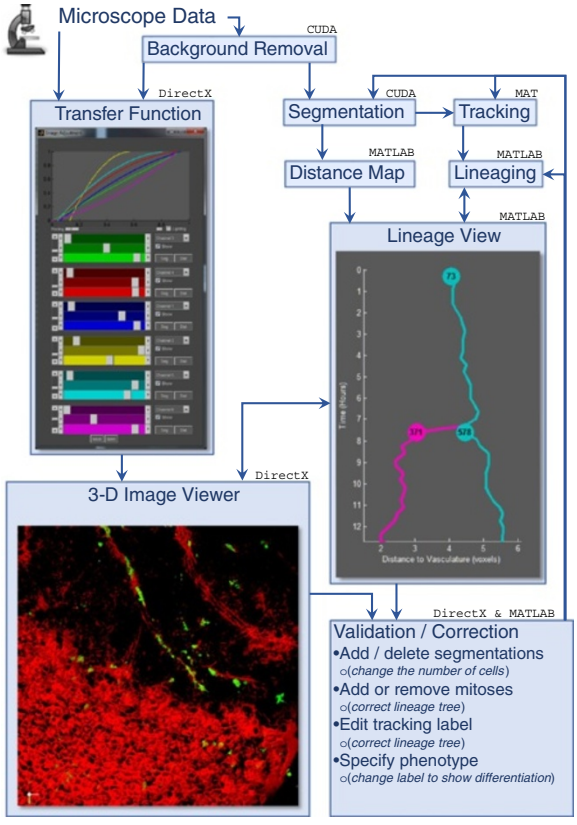

Figure 1 Schematic Diagram of LEVER 3-D. This flow chart shows the process in which LEVER 3-D uses automated algorithms along with user inputs to create a 100% corrected lineage tree. Note the feedback loop from the "*Validation and Correction*" section back into "*Segmentation*" and "*Tracking*".

Background noise removal

Confocal and multiphoton fluorescence microscopy of live stem cells is a battle between capturing enough signal and the disturbance of the specimen. Imaging is done in a manner that maximizes the signal-to-noise ratio (SNR) of the image data while being minimally invasive. Using less excitation energy causes less phototoxicity and disturbance to sensitive proliferating cells. This means that in practice, SNR ends up being quite low. One way to improve the analysis of this large volume of challenging image sequence data is to apply pre-processing techniques that model the underlying dynamic data and noise

processes. Here we use two different background noise removal techniques for the stem cell and the vasculature channel to better match the characteristics of the objects being imaged. These background noise removal algorithms provide the benefit both of removing noise and of providing adaptive contrast enhancement. This simplifies and improves the performance of the subsequent visualization transform as well as the cell and vessel segmentation algorithms.

For the stem cell channel, we adapt the background noise removal technique described by Michel *et al.* [10] that models the noise as a slow varying low-frequency background noise signal combined with a random (shot) noise process. Our approach differs slightly in that rather than using extreme value theory and peaks over thresholds approach to detecting fluorescent particles, we use a segmentation approach based on mathematical morphology combined with an adaptive Otsu transform [11] on the filtered image as described in the "*Segmentation*" section. This technique works as following: given the observed image ($\tilde{L}$) that we model as a combination of low-frequency background (B), random shot noise (R) and the original (denoised) image ($\hat{L}$) that we wish to recover,

$$\tilde{L} = B + R + \hat{L}. \tag{1}$$

The low-frequency background contribution *(B)* is estimated using a low pass (Gaussian) filter. The size of the neighborhood for the Gaussian filter can be set by the user, but is defaulted to 100 in each *(X,Y,Z)* dimension. When subtracting the results of this filter that had a large neighborhood, structure is preserved. Making the neighborhood too small will subtract away too much structure and the image will lose substantial overall energy. We have found that neighborhoods sizes in the range 75–250 give expectable results. However, going above 100 gives diminished returns. After subtracting the estimated background component from the observed image, the high frequency shot noise is removed using a median filter to produce the final denoised stem cell image used in the visualization transfer functions and the segmentation algorithm.

The above denoising approach works well on the stem cell channel where the foreground voxels (or pixels with a third dimension) are found in relatively small high frequency regions corresponding to cells. In the vasculature channel, foreground voxels can constitute large portions of the image corresponding to dense regions of blood vessels. For denoising the vasculature channel we have adopted a different denoising approach using Markov random fields [12] and a global estimate of noise variance rather than the local background estimate used in the stem cell channel. This is an iterative technique that first estimates the noise variance $\tilde{\sigma}$ for the original image I^0 by convolving it with a Laplacian operator. This noise variance estimation is used as the stopping condition for our iterative denoising where we keep the n^{th} image that is as different from the original image as our noise estimator predicted,

$$\|I^{n+1} - I^0\| \leq \tilde{\sigma}. \tag{2}$$

Each iteration changes a voxel by the minimum value Δ, where $\Delta = \min_{q,r \in i, q \neq r} |v_q - v_r|$ and v_i is the finite set of voxel values in the image. In other words, Δ is the minimum gap found on the current histogram. Each iteration adjusts the intensity of every voxel by a Δ depending on the gradient of the neighborhood, as defined as:

$$\begin{aligned} I^{n+1}_{i,j,k} = I^n_{i,j,k} + \Delta \times \operatorname{sgn}\Big[& \operatorname{sgn}\left(I^n_{i-1,j,k} - I^n_{i,j,k}\right) \\ & + \operatorname{sgn}\left(I^n_{i,j,k} - I^n_{i+1,j,k}\right) + \operatorname{sgn}\left(I^n_{i,j-1,k} - I^n_{i,j,k}\right) \\ & + \operatorname{sgn}\left(I^n_{i,j,k} - I^n_{i,j+1,k}\right) + \operatorname{sgn}\left(I^n_{i,j,k-1} - I^n_{i,j,k}\right) \\ & + \operatorname{sgn}\left(I^n_{i,j,k} - I^n_{i,j,k+1}\right)\Big]. \end{aligned} \tag{3}$$

This gradient decent/accent is continued until the stopping criterion $\tilde{\sigma}$ is met. The background denoising algorithms for both the vasculature and stem cell channels simplify segmentation, registration and visualization. The efficiency of all of the subsequent steps increases greatly if a "pure" signal can be extracted.

Registration

One of the next steps of preprocessing large static images is registration. One particularly interesting structure that produces stem cells throughout the life of the mammal is the subventricular zone (SVZ) in the brain. In previous studies, only subsections of the SVZ have been targeted for inspection [13]. Subsections were necessary due to the field of view being small given high magnification. We would rather be able to inspect the structure of the SVZ at the highest resolution possible. Having a sub-cellular resolution means that we can compare different populations with a higher level of precision. However, subsections may not contain corresponding structures that exactly line up between experiments. Our solution to this problem is to break the SVZ into a mosaic of high resolution overlapping subsections. These images are then reconstructed into an ultra high resolution large volume image of the entire structure.

Here we detail a novel optimization to this problem exploiting the constraints given by the structure of the specimen and the imaging modality. With modern microscopes, stage position is stored in the metadata of each image capture. Knowing the approximate stage position of the subsections with respect to one another, restricts which images can have adjacency. This information is

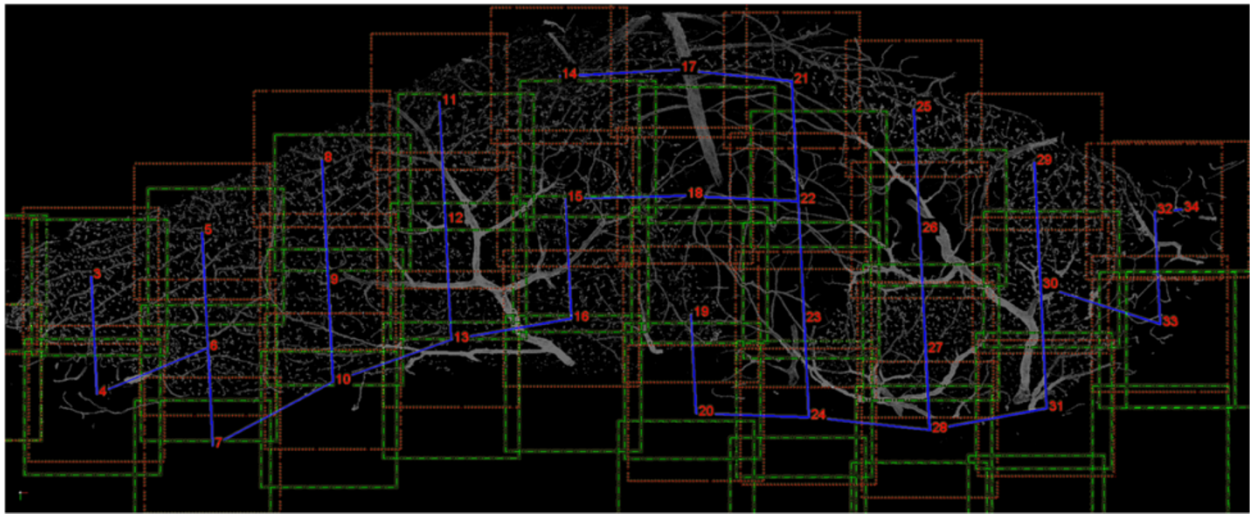

Figure 2 Reconstruction of an Entire SVZ. The image is the result of registering 34 subsections of a mouse subventricular zone. Registration using only microscope stage position data is indicated with a green dashed line. The blue solid lines represent a max spanning tree indicating which edge of the subsection was registered, *e.g.* subsection 22 was registered to 18, 21, and 23, where subsection 11 was only registered to 12. The red dashed lines indicate the final position of each subsection after registration. Registration happens in the *z* direction as well, not shown here.

enough to reconstruct the entire structure. However, the results leaves much to be desired. The initial reconstruction positions are displayed as green lines in Figure 2. During the imaging process, the specimen can shift relative to the stage, making stage position insufficient for registration. This can be caused by vibration of the stage, mechanical drift, dehydration or settling of the tissue slice, as well as removal of the slide between images. These position inaccuracies can be quickly overcome by using the overlapping image regions to find the relative offsets (register) of the mosaic of images.

The complexity of the registration problem can be reduced by the fact that images consist of a single timepoint and that each image in the montage is oriented orthogonal to the imaging stage. A montage with a large number of subsections, approaching 100, implies that the specimen has to be static in time. Imaging one subsection is time consuming and by the time another column or row is started, the overlap sections would have changed too much to be reconstructed. The next assumption comes from the movement of microscope stage itself. The stage moves in such a way that the subsections are all at right angles to one another in a checkerboard fashion. Our last assumption is that the volume will only deform in the direction of gravity. This is consistent with settling and dehydration. There should be only one subsection along this direction. This alleviates the need for transformations such as rotation, shearing, and deformation when registering. Based on these assumptions we can formulate an effective and accurate translational registration algorithm.

The following technique works best with a channel with unique semi-sparse structures that will span image subregions. The SVZ's vasculature channel is an ideal example of such a structure. The first step in our registration method is to create a maximum intensity image along the single subsection direction as stated in the paragraph above. The two overlapping maximum intensity projections are then shifted along the remaining two axes relative to each other as defined by a windowed area around the stage position data. Each position is evaluated to determine how well they fit together. We use the *normalized covariance* between the two overlapping volumes, normcov in Equation 4, to quantify similarity and select the maximum value. Once the best match is found, we use the original 3-D overlap volumes (no longer the maximum intensity images) to find the best registration along the remaining axis. The benefits to this method are that it is robust to variation in imaging parameters such as intensity (by subtracting the mean μ) and contrast (by normalizing over the variance σ), as well as being object agnostic. This technique also works directly on the images rather than requiring preprocessing to determine a set of feature points that are used in more complex registration schemes.

$$\text{normcov}(A, B) = \frac{\sum (A - \mu_A) \times (B - \mu_B)}{\sigma_A \times \sigma_B} \tag{4}$$

Each overlapping subsection is evaluated and stored in a graph structure. The nodes represent an offset from the original stage position. Each edge is the normalized covariance measure at the given offset. We then drop the lowest scoring edges until we are left with a max spanning tree, represented by the blue line in Figure 2. This

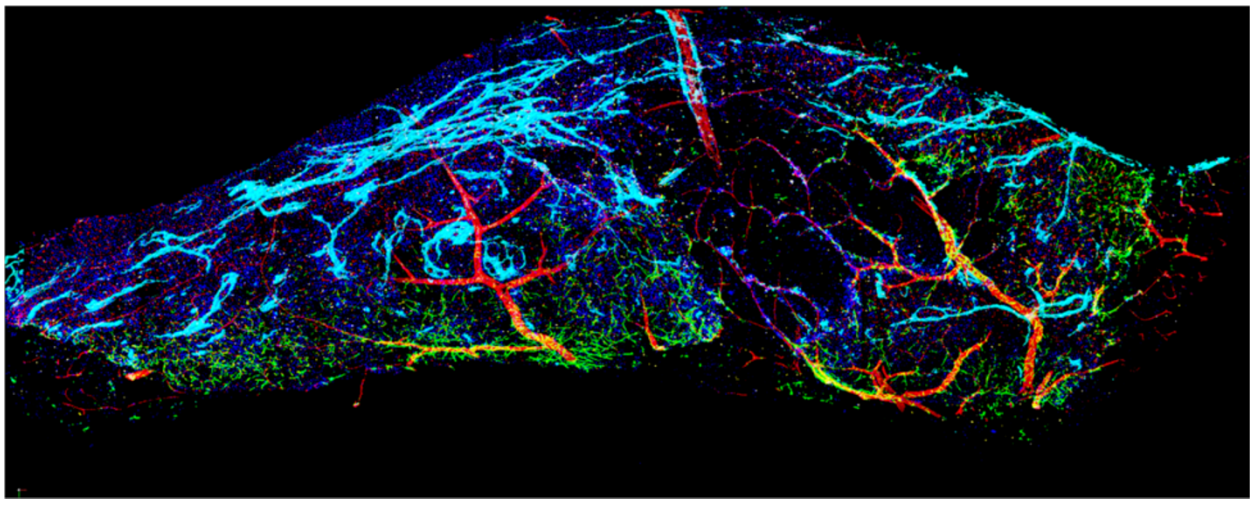

Figure 3 Fully Registered 3-D Montage with 5 Channels. This image has been reconstructed and rendered using the 3-D view window with adjustments made in the transfer function interface in Figure 5. The channels are: blood vessels (red), cell nuclei (dark blue), neural stem cells and astrocytes (green), oligodendrocytes (yellow), and migrating neuroblasts (cyan).

allows us to anchor a single sub-image and follow this max spanning tree assigning delta values relative to the change from the previous node in the graph. In other words, a node is chosen to be stationary (position based solely on stage position data). The delta for each node connected to this root node is based only on the registration position change. Each subsequent delta on a path of edges is calculated from the cumulative delta from the root and local registration delta. The final positions can be viewed as the red lines in Figure 2. Once deltas has been calculated for each sub-image, a final reconstructed image can be created. Given that each channel of a particular sub-image has been taken at the same time, we can register every channel using just one delta value. Figure 3 shows a fully registered SVZ containing five channels. Additional file 1 shows the same volume rotating and zooming into a region of interest. These images are then exported as large tiff files along with updated metadata files which contain the new dimensions. The image in Figure 3 has dimensions of 10,173 pixels in x, 3,858 in y, 74 in z, and five channels. These newly exported images are now ready to be processed just like any other images received from the microscope.

Segmentation

In previous applications involving stem cell segmentation and tracking in phase contrast 2-D image sequences [5,6,14] we have found that the most significant challenge to segmenting stem cells is to identify the correct number of cells in each connected component of foreground pixels. On a given frame, the numbers of cells in a given area may be ambiguous to even a domain expert. Ambiguity arises when the cells touch and most often occurs immediately following a mitosis event. This can also occur when there is a high density of cells. It is our experience that it is easier to resolve the correct segmentation in 3-D images as compared to 2-D images for both human and automated algorithms. The problem of estimating the number of cells in any connected component of foreground voxels is easier in 3-D due to the discriminative nature of the extra spatial channel. The output of the denoising algorithm improves segmentation accuracy by removing data that is not directly related to foreground voxels. Our denoiseing techniques are especially useful at preserving the gradient boundaries between cells and can remove ambiguity.

Our segmentation algorithm begins by applying an adaptive thresholding to all channels, using a CUDA Otsu filter [11]. This results in a binary image of foreground and background voxels. A morphological closing operator using a binary ball structuring element is applied to remove any erroneous holes in the structures. The stem cell channel is next processed with a connected component image filter and any connected components less than 19 μm^3 in volume are discarded. This value can be empirically set by the user prior to running the segmentation algorithm and is dependent only on cell type. Removing objects that fall below the smallest expected volume of a given cell type reduces spurious segmentations typically attributed to noise. Finally, the convex hull of the foreground voxels for each cell is computed using the open-source QHULLS package [15], creating facet and vertex lists for each cell. The convex hulls generated by QHULLS are then loaded into Direct 3D vertex and index buffers.

The vasculature channel is processed in a similar manner. Following the adaptive thresholding, a distance map

is computed using the MATLAB distance map filter. This provides the distance from each voxel to the nearest foreground voxel, and is used in the subsequent analysis to quantify the distance between each cell and its nearest vessel. The results of the stem cell segmentation are next passed to the tracking algorithm to establish temporal correspondences between segmentation results and assign tracking IDs to each cell.

Tracking

Once the cells have been segmented we use an approach called Multitemporal Association Tracking (MAT) [6,16] to establish temporal correspondences among segmentation results. MAT is a graph-based tracking approach which, for each frame, evaluates a multi-temporal cost function that approximates the Bayesian *a posteriori* association probability between the current set of tracks and all feasible track extensions out to a fixed window size W. In multiple hypothesis tracking, this data association problem is typically solved using bipartite or multidimensional assignment, which is NP-hard and requires explicit modeling of imaging specific conditions including occlusions, missed and extraneous detections. MAT instead uses a minimum spanning tree approach to solve the data association problem. It relies solely upon typical cell dynamic behavior of smooth motion and is independent of imaging conditions. In addition to the current results tracking stem cells in 3-D, MAT has achieved excellent results for hundreds of *in vitro* (2-D) image sequences of mouse adult and embryonic neural stem cells, as well as hematopoietic stem cells and rat retinal progenitor cells [6] and has been applied to tracking high density organelle transport along the axon [16]. All of the stem cell movies tracked by MAT in 2-D and 3-D were processed with the same implementation.

In order to extend tracks from frame t to $t+1$, we denote a partially constructed track terminating at the i^{th} detection in frame t as τ_i^t, and denote the set of all feasible extensions passing through the j^{th} detection in frame $t+1$ as ρ_j^{t+1}. The cost of edge c_{ij} in the tracking graph is assigned the minimum cost of extending partial track τ_i^t through the j^{th} detection in the next frame $c_{ij} = \min_j \left\{ C\left(\tau_i^t, \rho_j^{t+1}\right) \right\}$. Edges $c_{ij}^\star$ satisfying $c_{ij}^\star \leq c_{in}$ and $c_{ij}^\star \leq c_{mj}$ for any $m \neq i$ and $n \neq j$ are called matching edges. If it exists we extend each track τ_i^t along its matching edge $c_{ij}^\star$. For any detections in $t+1$ without a matched incoming edge $c_{mj}^\star$, we initialize a new track. Occlusions, where one cell is visually obscuring another, are handled by allowing tracks that are not extended to be considered in subsequent frames.

For tracking stem cells in the 5-D image sequences we used the same cost function used previously for tracking 2-D phase-contrast imaged NSCs [6], with the sole modification of using a Z value in the connected-component distance. We define the connected component distance between two detections as

$$d_{\text{CC}}(\alpha, \beta) = \min_{r_\alpha \in \alpha, r_\beta \in \beta} \left\{ \| r_\alpha - r_\beta \| \right\}, \tag{5}$$

where r_α, r_β are the scaled-voxel coordinates corresponding to the foreground-voxels of the segmentation detections and, respectively. We also define a detection size distance to preserve homogeneous sizes along a given track,

$$d_{\text{size}}(\alpha, \beta) = \frac{\max\{|\alpha|, |\beta|\} - \min\{|\alpha|, |\beta|\}}{\max\{|\alpha|, |\beta|\}}, \tag{6}$$

where $|\alpha|$ is the number of pixels in the foreground connected-component of detection α. For a given path extension we calculate the cost of the extended track (τ, ρ), as a weighted sum of the local connected component distances along the detections of (τ, ρ),

$$C(\tau, \rho) = (W - |\rho| + 1) \times \left(\sum_{i=1}^{W} w_i \left(d_{\text{CC}}\left(\rho_i, \rho_{i+1}\right) + d_{\text{size}}\left(\rho_i, \rho_{i+1}\right) \right) \right), \tag{7}$$

with ρ_i indicating the i^{th} detection on path (τ, ρ). We define $|\rho|$ as the minimum of the track length or window size W. By convention, if $i \leq 0$ we use $\rho_i = \tau_{\tau_{t+i}}$, which allows evaluation of the cost over the full-extended path. This cost reflects the expected behavior of neural stem cells, namely, the cells should not move far (small connected-component distance) and their size shouldn't vary greatly in adjacent frames. The multiplicative term discourages shorter tracks which would otherwise have lower cost due to fewer terms. We use a window size $W = 4$. This cost function has proven effective in tracking both 2-D and 3-D cells as shown in the current and previous applications.

Lineaging

Lineaging identifies parent-daughter relationships among the proliferating cells. Our lineaging approach uses the minimum cost path extensions discovered by the MAT algorithm and stored in a sparse graph structure during tracking, and is the same algorithm we have previously used for 2-D lineaging [5,6,14]. Cells that constitute viable tracks that appear after the first image frame are assigned parents based on these tracking results. Given a track τ_{newborn}, we identify its parent track τ_{parent}^* as,

$$\tau_{\text{parent}}^* = \arg\min \left\{ C(\tau_{\text{parent}}, \tau_{\text{newborn}}) \right\}. \tag{8}$$

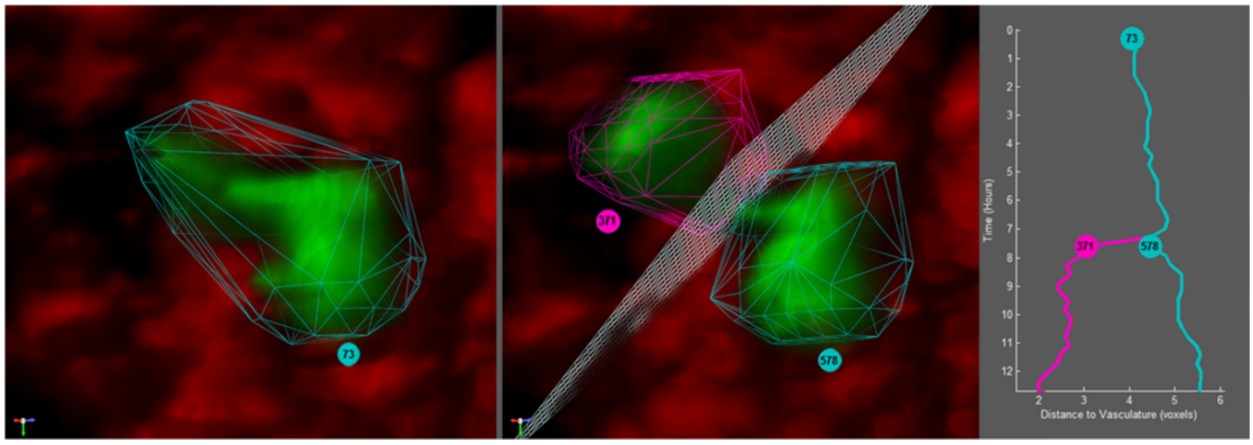

Figure 4 Mitosis event with lineage tree. The lineage tree in the right panel shows an entire clone starting with the progenitor cell 73 which divides into two daughter cells, 371 and 578. The *y*-axis represents time where the *x*-axis represents the cell's distance to its closest blood vessel. The left panel shows cell 73 in the frame prior to it undergoing mitosis. The center panel shows the frame in which cell 73 divides into cells 371 and 578. The cleavage plane is represented by a white mesh and shows the angle of cleavage relative to the vessel channel. Specimens that are imaged over time typically have fewer channels than static samples. Immunofluorescence can be detrimental to natural cell behavior and has to be used sparingly. Static images can be stained allowing for a larger number of channels given that the cells are already dead.

Once this has completed for all tracks in the image sequence, the largest (most nodes) lineage tree is presented to the user. This tree is the first one to be displayed with the expectation that it represents either the most interesting biological lineage or the lineage that needs the greatest number of user edits.

Traditional lineage trees communicate informative data such as cell cycle time, number of progeny, apoptosis, symmetry of subtrees, *etc*. In order to study cells in their niche, the spatial relationships between objects must be measured. Figure 4 and Additional file 2 shows a lineage tree that encodes the spatial distance between stem cells and the nearest vasculature. The vertical lines on the lineage tree in the direction of the x-axis have been modified to show the distance between a particular stem cell and its nearest vasculature voxel at each image frame. At the time of division, the lineage tree shows one daughter cell farther from the vascular structure while the other daughter cell maintains its distance. We also represent the angle in which the cells cleave during mitosis by a plane. This cleavage plane allows for qualitative inspection of how cells divide as well as an additional feature for quantitative analysis.

User interface

Automated analysis of image sequence data showing proliferating cells will inevitably make mistakes. In addition to low SNR, these image sequences often contain visual ambiguities, *e.g.* due to cells dividing, entering or exiting the imaging frame *etc.*, where it may be impossible even for a human domain expert to correctly identify the number of cells in a connected component of foreground voxels from a single time point. These segmentation errors cause tracking and lineaging errors which can then corrupt the ultimate analysis. Once all of the automated image analysis algorithms have completed, the data needs to be displayed to the user in such a way that it enables errors to be easily identified and quickly corrected. The approach adopted here is to display the volumetric image sequence data with segmentation and tracking results overlaid in a Direct 3D window and the lineage tree in a second MATLAB interactive figure window. The Direct 3D and MATLAB components run in a single memory process, launched from MATLAB and communicating with shared memory via the MATLAB Mex interface. This allows for fast prototyping and scripting right in the MATLAB integrated development environment. This enables programmers to extend the image processing and biologists to access their data directly (many users will fit both of these use cases).

Visualization of the volumetric data uses the notion of 3-D textures. The slices of each 3-D texture are projected onto planes which consist of two view aligned triangles. There are $\sqrt{2}$ times the maximum pixel dimension of these planes spanning the depth of the image volume. Direct 3D maps the image data onto the triangles using a custom shader. This shader incorporates parameters derived from the transfer function sliders in Figure 5. The transfer function was inspired by the work of Wan *et al.* [17]. Currently there are six unique colors that each channel can be assigned. As Wan *et al.* point out, it is difficult to render unique visually separable colors greater than three. When channels are allowed to alpha blend (transparency between channels)

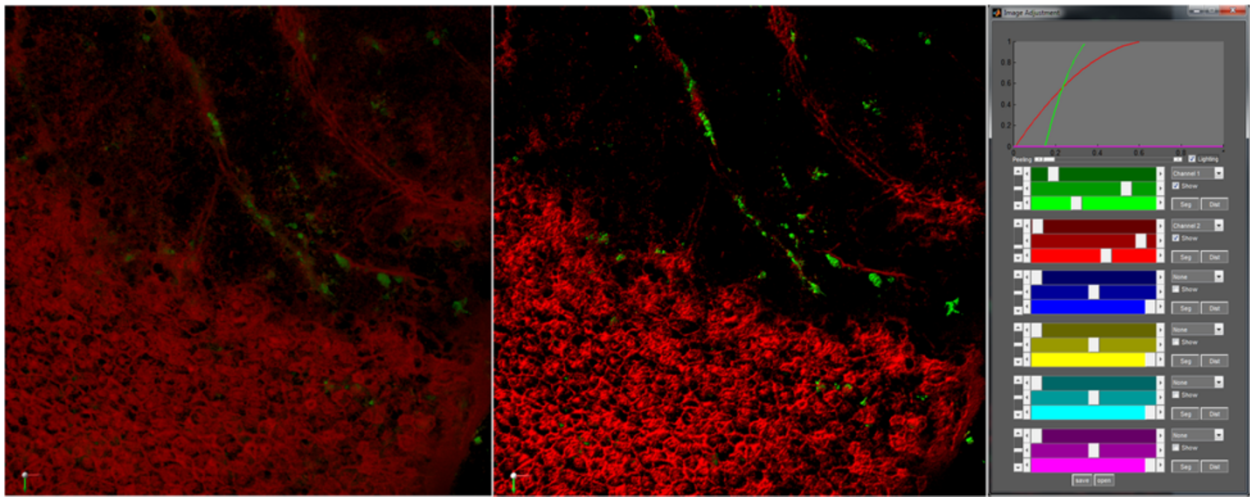

Figure 5 3-D Image viewer and transfer function windows. The right panel shows the controls to set a transfer function which maps the intensity values of the original images into values and colors in the view window. The left panel shows the original image data without any changes to the transfer function. The middle panel shows an image where the transfer function settings shown in the far right panel have been applied.

it becomes even more difficult to differentiate between colors that have been mixed. LEVER 3-D allows any of the colors to be assigned to a channel and its visibility toggled.

The colored sliders for an assigned channel are used to set a polynomial transfer function that will map the original image intensity values to an intensity value that will be colored and displayed in the 3-D window. The darker (top) slider for a given channel will set where the low values of the original image will be floored to zero. The brightest (bottom) slider for a given channel is used to set the threshold in which all larger values are mapped to the maximum value; in this case 255 for an 8-bit image. The middle slider changes the curve of the line between the max and min values. The slider on the left edge is used to set the multiplier on the base alpha value for a given channel, allowing a particular channel to be more or less transparent relative to the others. The base alpha for a given pixel is set to the maximum channel intensity at that voxel after the transfer function has been applied. This slider operates on the range of $[0, 2]$, where the center position equals a multiplier of 1. The last user interface control that pertains to the image data is whether to light the texture or not. The lighting check box turns on and off a global directional light. The image data is lit using the surface normal of each voxel. This normal is approximated directly in the shader by finding the gradient direction for each pixel based on a $3 \times 3 \times 3$ neighborhood. Lighting of this kind gives a more three dimensional feel to the image even when viewed on a two dimensional medium (see Figure 5). Once the images are set to the user's liking, they are able to integrate the segmentation results into the 3-D window.

Segmentation data is then loaded into video RAM to be overlaid onto the image volume to show tracking and lineaging results. The triangle meshes generated by QHULLS as the convex hull of each cell's voxels by the segmentation process are loaded into Direct 3D index and vertex buffers (triangle lists) and colors are assigned according to tracking and lineaging results. The segmentation buffer is rendered by a second custom shader that can be toggled between off, wire-frame, and solid. The default renderer draws the texture in its entirety and then draws the segmentation results on top of that. This means that the segmentation triangles will always be drawn on top of the image data. This does not show when segmentation is embedded into another structure; however, it renders at high frame rates. Objects obscure one another based solely on their placement in the render loop. Depth peeling gives more visual cues that one object is obscured by another. This is accomplished by layering or "peeling" from each structure in a depth sorted order and interlacing them in the render loop. That will draw the objects that are closer to the view after the further objects, obscuring more distant objects. This can slow the rendering of larger volume data down to non-interactive speeds. To mitigate this problem, a chunked depth peeling has been implemented [17]. There is a slider on the transfer function window (Figure 5) labeled "*Peeling*". This allows for $[1, N]$ chunks to be peeled, where N is the view aligned voxel count of the volume. This slider can be used to add a level of segmentation integration in the image data as the user's hardware will allow for and still be interactive. The segmentation results are quite often the reason for errors in the tracking and lineaging and are where the majority of the edits take place. This is why it is important to give as

many visual cues as possible to show where the segmentation is wrong. When it is wrong, LEVER 3-D allows the user to correct it manually.

Selecting a cell for visualization or editing in the 3-D volume with the use of two dimensional tools (*e.g.* mouse pointer) can be challenging. When the user clicks on the volume, an inverse projection is used to find the intended cell. The inverse projection consists of a ray starting at the view origin passing through the cursor's position on the projection plane and continuing through the volume space. The cell containing the first triangle that is intersected by this ray is then selected. With this selected cell, the user then can remove all other segmentations from the display to leave emphasis on the cell in question. In this view configuration, the user has the ability to play the sequence and follow the particular cell through the experiment.

Once a cell has been selected, the user then can correct the segmentation or tracking results for the cell. For correcting segmentation results, a cell can be split into n cells or can be deleted. To split a cell, we fit a mixture of Gaussians [18] on the foreground voxels of the cell. This is effective because a mixture of Gaussians decision boundaries favor ellipsoidal shapes that model 3-D NSCs more appropriately than k-means, which favor more spherical shapes. After a segmentation has been corrected, the tracking automatically reruns. The segmentation edit provided by the user can be "propagated" by inspecting tracking assignments for the original segmentation as potential automated correction candidates until the newly added segmentation establishes its own track. This is the same inference-based approach to learning from user provided edits used in our previous 2-D stem cell lineaging application [6].

The selection of a particular cell also selects the clone to show in the 2-D lineage tree window. The lineage tree is one of the easiest ways of identifying errors in the automated image analysis routines due to predictable qualities such as regularly spaced mitotic events, cells on the lineage tree existing until they reach the end of the sequence or a frame boundary, *etc*. The selection of a cell is translated through a Mex interface to MATLAB allowing the selected clone to be toggled. The shared memory Mex architecture enables all of the segmentation, tracking, and lineaging results to be accessed directly as MATLAB data structures and leverages the implementation from our previous 2-D stem cell lineaging application [6] for lineage tree manipulation and display. The Mex interface allows bidirectional communication between the MATLAB and the Direct 3D user interfaces allowing the two windows to be tightly coupled and enabling a high throughput approach to validating and correcting the automated image analysis algorithms.

The typical work flow of LEVER 3-D proceeds as follows. The user with access to the MATLAB program can launch LEVER 3-D from inside the MATLAB development environment. This allows the user to integrate LEVER 3-D visualization and analysis components with their own scripts. For user that do not have access to MATLAB, we provide a standalone program. The results from the stand alone program can be exported for analysis in other environments. The first step is to specify the location of the location of the raw data file from the microscope. The image data is then buffered onto the graphics card and is displayed in the image window. If the current dataset has been processed previously (from a previous session), the segmentation results are rendered in the image window. The lineage tree containing the most cells is shown in a second window. If the dataset is unprocessed, the user is able to specify a processing method on a particular channel. Now the user can explore the image data and validate the automated processing. Viewing of the image data is enhanced by the transfer function window. The transfer function is used to compensate for images with low amounts of fluorescence and uncalibrated monitors. Once the user is satisfied with data and the view settings, LEVER 3-D then can export image, movies, and metrics for external use. Additional file 3 provides a short video overview of the usage of the LEVER 3-D application from within the MATLAB environment. A similar usage scenario is also possible without requiring MATLAB by using our compiled executable.

Results and discussion

The analysis of the image sequence data proceeds as follows. All timing information is based on a Windows PC with dual Xeon X5570 processors (2.9 GHz), 24 GB of RAM and an Nvidia GTX 680 video card with 4 GB of video RAM. The automated image analysis routines were implemented in C++ using CUDA. Background noise removal, segmentation, tracking, vascular distance and lineaging are run off-line. This step requires seconds to only a few minutes using CUDA compared with up to 2 hours of processing time using the open source Insight Toolkit (ITK). These times are dependent on the size and dimensionality of image sequence data. The vast majority of this time is consumed by the background noise removal in ITK; in CUDA the task of noise removal, segmentation, tracking, and lineaging are more equally spaced. Background noise removal is a task that is only run once per image sequence and improves the results of the subsequent automatic segmentation algorithm, especially with suppressing image noise between closely adjacent cells improving the ability of the segmentation algorithm to separate nearby cells.

After the automated image analysis routines complete, the 3-D image data with segmentation and tracking results

overlaid are shown in the imaging window (Direct 3D connected via MATLAB's Mex interface) and the lineage tree is shown in a MATLAB figure window. The active shutter stereoscopic 3-D visualization glasses improve the visualization of the stem cell data, and especially the relationship between stem cells and vasculature. Stereoscopic rendering allows the viewer to disambiguate the relative distances between objects compared to a monoscopic viewing where additional visual cues are necessary, such as movement (rotation) or lighting.

Image sequence data displays at 60 frames per second, and manipulation of the 3-D volumetric data is fully interactive even with the 3D stereo vision hardware activated. Navigation can be done on either the 3-D window or on the 2-D lineage window. Clicking in the lineage tree window causes the frame to advance to the selected time point. The time can also be navigated on the Direct 3D window using the mouse wheel, and causes the time indicator on the lineage tree window to update. Users can edit the segmentation and tracking in the imaging window by splitting cells with the mouse or by typing tracking numbers directly onto a cell. The user has only had to correct the automated processes 7% of the time on average. Most of these errors were due to the cells not separating immediately after mitosis. We recently developed improved techniques for resolving visual ambiguity in 2-D image sequences of proliferating cells by incorporating tracking and lineaging information [19]. These methods offer a promising approach to reduce the number of errors in the 3-D segmentation and will be added in future versions. Given correct segmentation and the time resolution of the imaging is such that object overlap themselves by at least 50% between frames, there should be no tracking errors [6]. The tracking and lineaging algorithms are automatically executed in response to user provided segmentation edits in order to dynamically update the results and also to correct related segmentation errors in future frames. This process typically requires a few seconds to complete, making response to editing operations as well as the 3-D visualization fully interactive.

Once the tracking and lineaging for a clone of stem cells has been corrected, the data can be exported to MATLAB for further analysis. In order to explore the relationship between stem cells and their vascular niche, we used a distance map of the vascular channel. For each stem cell on the clone, we can use this distance map to instantly find the distance between the cell and the nearest blood vessel. In Figure 4 we plot this distance for the three cells on the lineage tree of the selected clone. Cell 73 is at a stable distance to the nearest blood vessel. When the cell divides one of the daughter cells moves into contact with the vessel while the second daughter continues on its parent's trajectory. This is a result of the cleavage plane, formed by the division between the two daughter cells, being oriented acutely toward the vessel. Interestingly, the daughter cell that is closer to the vasculature following division, cell 371, has a different pattern of motion than the daughter cell that is not in contact with vasculature. This may be indicative of a different sub type of stem cell or of the cell seeking to re-establish its location in the vascular niche following division. This is the first time, to our knowledge, that this relationship between a clone of mammalian NSC's and their vascular niche has been visualized and quantified dynamically in live cell and tissue image sequence data.

A key decision in the design of our application was the use of Direct 3D rather than OpenGL to provide 3-D rendering. In general, scientific visualization applications tend to use OpenGL while gaming applications tend to use Direct 3D. This decision was primarily based on the need to incorporate support for NVidia's 3-D Vision to utilize active shutter stereoscopic glasses into our application. Using Direct 3D enables the use of 3-D vision on less expensive NVidia GTX-class gaming cards, as well as on the more expensive Quadro cards. Additionally, automatic driver optimized support for stereo separation is available to Direct 3D applications only [20] eliminating coding overhead. The 3-D vision stereo glasses can be used from OpenGL, but that requires the use of the more expensive Quadro card and also requires explicit application support for stereo via quad buffering. Stereoscopic viewing enables a user to quickly identify and easily correct tracking and lineaging errors in a natural and highly interactive manner. Shortcomings of using Direct 3D are discussed in the "*Conclusions*" section.

Other applications have been developed for 3-D stem cell lineaging, notably by Murray *et al.* [21]. They developed an approach that does not however, include capabilities for learning from user supplied edits or 3-D visualization. In later work they incorporated a support vector machine to automatically identify segmentation errors [22], although we have found that segmentation errors occur primarily when there is visual ambiguity in the image data that the human eye is unable to resolve using only a single image frame. The current project is an extension of the LEVER application designed for 2-D phase contrast stem cell image sequences which uses a human observer to assist in correcting the visual ambiguity inherent in image sequences of live proliferating cells [6]. Aside from the 3-D rendering, one other difference of the current work is that the segmentation is implemented using CUDA rather than MATLAB or ITK. CUDA provides a significant performance improvement over MATLAB and ITK, making the 3-D noise and background removal and segmentation algorithms approximately 60 times faster.

In the area of biological image sequence data visualization there are a large number of commercial and open source products, as described in a review paper by Walter

et al. [23]. One thing that differentiates our work from the described approaches is the tight integration between the automated image analysis algorithms and the Direct 3D visualization. In contrast, most other applications utilize the Visualization Toolkit (VTK) an open source visualization library [24]. The 3-D rendering for our application was initially implemented using VTK, however VTK is only compatible with OpenGL and not with Direct 3D. Direct 3D has the benefit of using low-cost gaming hardware for stereoscopic visualization which is necessary for efficient validation of 3-D volumes. The open source ICY application [25] uses VTK for visualization and provides an extensible user interface for visualizing 2-D and 3-D images and incorporates segmentation and tracking algorithms, as well as editing of results and multiple linked views. Our work differs from ICY in supporting stem cell lineaging and using inference-based learning to propagate user-provided edits. A related application for visualizing multichannel fluorescence microscopy data for biological applications was presented by Wan *et al*. [17]. Their application provided more control over the viewing of the volumetric data including a user controllable 2-D transfer function for setting the rendering properties of the volumetric data. In contrast, the approach presented here uses the automatic image analysis algorithms to set the parameters on the visualization transfer function with the intention that our application will be used for quickly validating and correcting the clonal tracking and lineaging results prior to subsequent statistical and algorithmic information theoretic analyses.

There are a number of papers describing tools for visualizing and analyzing 3-D image sequence data [26-28]. Taken together, these show the power of combining image analysis and visualization tools. As Amat *et al*. [29] note, extending such techniques to "3D+t is not straight forward." The approach described here is novel in the ability to visualize 5-D image sequence data, utilizing automated tracking and lineaging algorithms to analyze the time course of the dynamic behaviors for all of the cells in a developing clone and incorporating user-provided edits to automatically correct related errors. This provides unprecedented functionality for working with complex live cell and tissue image sequence data and ensures that subsequent analysis starts with 100% corrected data.

Visualization comparison between 2-D projection and stereoscopic 3-D rendering is difficult to quantify. Confocal microscopes are able to capture true three-dimensional data and there are many tools that make two-dimensional projections of this data, such as that of Schmid *et al*. [30], and Peng *et al*. [26,27]. However, 2-D projection relies on visual cues to convey the relative depth between objects as explained by Wan *et al*. [17]. Using a stereoscopic projection allows our binocular vision to convey this information more precisely [31]. Even without depth peeling, lighting, or other cues necessary with monoscopic projection, the user can perceive depth between objects. We incorporate both depth peeling and lighting to make the scene look more natural. With the perceived depth from stereoscopic visualization, validation can be more accurate and efficient. Stereoscopic projection can also help earlier in the processing pipeline by expediting discovery. Interactions between structures in the SVZ are not fully known. Direct stereoscopic observation facilitates the identification of regions to quantify and determine their significance. This discovery phase optimizes the processing pipeline by identifying more precisely what models the analysis phase can emulate or exploit. We believe that there is enough qualitative benefit to the stereoscopic projection to base a large part of visualization decisions upon it.

Conclusions

We have developed a new application called LEVER 3-D for validating and correcting the automated segmentation, tracking and lineaging of stem cells from 5-D time lapse image sequence data. The segmentation and tracking results are overlaid on the image data in the 3-D rendering window. The lineage tree for the currently selected clone is shown in a MATLAB 2-D window. Navigation and editing can take place on either window; the MATLAB Mex interface is used to communicate between the C++, CUDA, DirectX, and MATLAB components. The ability to visualize the image data simultaneously with segmentation, tracking, and lineaging makes it possible to quickly identify and easily correct any errors in the automatic analysis. Direct 3D is used for 3-D rendering, providing active shutter stereoscopic visualization and interactive rendering on low-cost gaming hardware. We use the open-source Bioformats tool to read the image data directly from the microscope and CUDA kernels to implement the background removal and segmentation algorithms. The open-source MAT tracking algorithm developed previously for 2-D stem cell image sequences has been enhanced to work with 5-D stem cell data.

One drawback to our use of Direct 3D to enable active shutter stereoscopic is that our application is only available on the Windows operating system. The use of OpenGL would have required explicit application support for active shutter stereoscopic visualization and also the use of the more expensive Quadro-class video cards together with additional RAM for quad buffering, but would have allowed for true cross-platform support. Another drawback of Direct 3D is the lack of a web integration module such as WebGL, which makes it difficult to implement a web client for demonstrating the capabilities of the system or for implementing distributed applications for validating and correcting the 5-D image sequence data. We believe that these shortcomings are

offset by the improved visualization available for low-cost from DirectX, with active shutter stereoscopic visualization automatically in the display driver using Nvidia GTX class display cards.

Our goal is to develop an open source solution that allows biologists to process, validate and analyze 5-D stem cell image sequence data in the laboratory, increasing the pace of discovery by combining accurate unsupervised image analysis together with intuitive visualization and validation tools. The current version of the source code as well as video tutorials are available at (https://git-bioimage.coe.drexel.edu). We are including executables along with a complete 5-D dataset to enable readers to run the interface directly. Data collection has begun for a number of biological experiments that will utilize LEVER 3-D in a high throughput capacity to quantify dynamic behaviors and niche associations for clones of NSCs. The application described here represents a first step in disseminating widely applicable software tools for the analysis of proliferating cells and vasculature from 5-D image sequence data.

Availability of supporting data

Source code and test image data is available on our website http://bioimage.coe.drexel.edu in the *Software* section.

Additional files

Additional file 1: Video showing multi-resolution visualization of the 5 channel 3-D montage from Figure 3. Starting with the full resolution image a complete rotation of the volume is rendered using 300 intermediate frames. Following each revolution the width and height of the field of view are reduced by a factor of two. The white rectangle shows the location of the next field of view that will be rendered. The process is repeated five times, ultimately showing the image data at full resolution with no scaling.

Additional file 2: Video showing analysis results together with image data from 5-D confocal microscopy showing neural progenitors undergoing mitosis. The left panel shows the segmentation and tracking results overlaid on the image data. The right panel shows the lineage tree encoded with division time combined with the distance from each neural progenitor to the nearest vasculature at each image frame. The color of the segmentation on the left panel corresponds to the track shown on the lineage tree. The cleavage plane shows the orientation of the daughter cells at the time of separation relative to the surrounding vasculature. The ability to interactively explore complex spatiotemporal relationships in 5-D image data is an important prerequisite to quantitative analysis.

Additional file 3: A video demonstrating the use of LEVER 3-D from a MATLAB session. The control window provides access to the transfer functions with parameters controlling visualization. The image window shows the microscopy data together with the segmentation and tracking results. As the transfer functions are manipulated, the image display is updated immediately. The control window also provides access to the denoising and segmentation algorithms. All the data structures and functionality is accessible from MATLAB scripts. Stereoscopic 3-D requires a monitor and video card that supports Nvidia's 3-D vision.

Abbreviations

NSC: Neural stem cells; AITPD: Algorithmic Information Theoretic Prediction and Discovery; SVZ: Subventricular zone; SNR: Signal-to-noise ratio; MAT: Multitemporal Association Tracking.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

EW implemented algorithms, designed the user interface, and integrated the original LEVER program into the new imaging paradigm. MW was instrumental in the design of the shader paradigm, designing the MEX interface, and provided underling theory to the registration algorithm. CB and YW prepared the tissue samples and captured the subsequent images of both the time lapse and the montage. EK and YW prepared the tissue samples for the time lapse, SG captured the subsequent timelapse movies. ST was the principle investigator and provided oversight in the biological laboratory. EW and AC wrote the paper. All authors have read and approved the final manuscript.

Acknowledgments

Portions of this research were supported by Drexel University, by grant number R01NS076709 from the National Institute Of Neurological Disorders and Stroke, and by the National Institute On Aging of the National Institutes of Health under award number R01AG041861. The content is solely the responsibility of the authors and does not necessarily represent the official views of Drexel or the National Institutes of Health.

Author details

[1]Drexel University, 19104 Philadelphia, USA. [2]Neural Stem Cell Institute, 12144 Rensselaer, USA. [3]UT Health Science Center, 78229 San Antonio, USA.

Received: 22 April 2014 Accepted: 19 September 2014
Published: 3 October 2014

References

1. Shen Q, Wang Y, Kokovay E, Lin G, Chuang SM, Goderie SK, Roysam B, Temple S: **Adult svz stem cells lie in a vascular niche: a quantitative analysis of niche cell-cell interactions.** *Cell Stem Cell* 2008, **3**(3):289–300. doi:10.1016/j.stem.2008.07.026.
2. Kokovay E, Goderie S, Wang Y, Lotz S, Lin G, Sun Y, Roysam B, Shen Q, Temple S: **Adult svz lineage cells home to and leave the vascular niche via differential responses to sdf1/cxcr4 signaling.** *Cell Stem Cell* 2010, **7**(2):163–173. doi:10.1016/j.stem.2010.05.019.
3. Tavazoie M, Van der Veken L, Silva-Vargas V, Louissaint M, Colonna L, Zaidi B, Garcia-Verdugo JM, Doetsch F: **A specialized vascular niche for adult neural stem cells.** *Cell Stem Cell* 2008, **3**(3):279–288. doi:10.1016/j.stem.2008.07.025.
4. Cohen AR, Bjornsson C, Temple S, Banker G, Roysam B: **Automatic summarization of changes in biological image sequences using algorithmic information theory.** *IEEE Trans Pattern Anal Mach Intell* 2009, **31**(8):1386–1403.
5. Cohen AR, Gomes F, Roysam B, Cayouette M: **Computational prediction of neural progenitor cell fates.** *Nat Methods* 2010, **7**(3):213–218.
6. Winter M, Wait E, Roysam B, Goderie S, Kokovay E, Temple S, Cohen AR: **Vertebrate neural stem cell segmentation, tracking and lineaging with validation and editing.** *Nat Protoc* 2011, **6**(12):1942–1952. doi:10.1038/nprot.2011.422.
7. Doetsch F, Caille I, Lim DA, Garcia-Verdugo JM, Alvarez-Buylla A: **Subventricular zone astrocytes are neural stem cells in the adult mammalian brain.** *Cell* 1999, **97**(6):703–716. doi:S0092-8674(00)80783-7.
8. Ortega F, Costa MR, Simon-Ebert T, Schroeder T, Gotz M, Berninger B: **Using an adherent cell culture of the mouse subependymal zone to study the behavior of adult neural stem cells on a single-cell level.** *Nat Protoc* 2011, **6**(12):1847–1859. doi:10.1038/nprot.2011.404 nprot.2011.404.
9. **BioFormats** [http://loci.wisc.edu/software/bio-formats]
10. Michel R, Steinmeyer R, Falk M, Harms GS: **A new detection algorithm for image analysis of single, fluorescence-labeled proteins in living cells.** *Microsc Res Tech* 2007, **70**(9):763–770. doi:10.1002/jemt.20485.
11. Otsu N: **A threshold selection method from gray-level histograms.** *IEEE Trans Syst Man Cybern* 1979, **9**(1):62–66.
12. Ceccarelli M: **A finite markov random field approach to fast edge-preserving image recovery.** *Image Vis Comput* 2007, **25**(6):792–804. doi:10.1016/j.imavis.2006.05.021.
13. Narayanaswamy A, Dwarakapuram S, Bjornsson CS, Cutler BM, Shain W, Roysam B: **Robust adaptive 3-d segmentation of vessel laminae from**

fluorescence confocal microscope images and parallel gpu implementation. *IEEE Trans Med Imaging* 2010, **29**(3):583–597. doi:10.1109/TMI.2009.2022086.
14. Al-Kofahi O, Radke RJ, Goderie SK, Shen Q, Temple S, Roysam B: **Automated cell lineage tracing: a high-throughput method to analyze cell proliferative behavior developed using mouse neural stem cells.** *Cell Cycle* 2006, **5**(3):327–335.
15. **QHULL** [http://www.qhull.org/]
16. Winter M, Fang C, Banker G, Roysam B, Cohen A: **Axonal transport analysis using multitemporal association tracking.** *5* 2012, **1:**35–48.
17. Wan Y, Otsuna H, Chien CB, Hansen C: **An interactive visualization tool for multi-channel confocal microscopy data in neurobiology research.** *IEEE Trans Vis Comput Graph* 2009, **15**(6):1489–1496. doi:10.1109/TVCG.2009.118.
18. Theodoridis S, Koutroumbas K: *Pattern recognition*. San, Diego: CA: Academic Press; 2009.
19. Mankowski WC, Winter M, Wait E, Lodder MJ, Schumacher TN, Naik SH, Cohen AR: **Segmentation of occluded hematopoietic stem cells from tracking.** In *36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*: IEEE Journals & Magazines; 2014:5510–5513.
20. Corporation N: **Nvidia 3-d vision automatic best practices guide.** Report, July 2010. [http://developer.download.nvidia.com/whitepapers/2010/NVIDIA%203D%20Vision%20Automatic.pdf].
21. Murray JI, Bao Z, Boyle TJ, Waterston RH: **The lineaging of fluorescently-labeled caenorhabditis elegans embryos with starrynite and acetree.** *Nat Protoc* 2006, **1**(3):1468–1476. doi:10.1038/nprot.2006.222.
22. Aydin Z, Murray JI, Waterston RH, Noble WS: **Using machine learning to speed up manual image annotation: application to a 3d imaging protocol for measuring single cell gene expression in the developing c. elegans embryo.** *BMC Bioinformatics* 2010, **11**(1):84. doi:10.1186/1471-2105-11-84.
23. Walter T, Shattuck DW, Baldock R, Bastin ME, Carpenter AE, Duce S, Ellenberg J, Fraser A, Hamilton N, Pieper S, Ragan MA, Schneider JE, Tomancak P, Heriche JK: **Visualization of image data from cells to organisms.** *Nat Methods* 2010, **7**(3 Suppl):26–41. doi:10.1038/nmeth.1431.
24. Schroeder W, Martin K, Lorensen B: *The Visualization Toolkit, 2nd edn*. Upper Saddle River: Prentice Hall PTR; 1998. 645.
25. De Chaumont F, Dallongeville S, Olivo-Marin JC: **Icy: a new open-source community image processing software.** In *Biomedical Imaging: From Nano to Macro, 2011 IEEE International Symposium On*: IEEE Conference Publications; 2011:234–237. doi:10.1109/ISBI.2011.5872395.
26. Peng H, Ruan Z, Long F, Simpson JH, Myers EW: **V3d enables real-time 3d visualization and quantitative analysis of large-scale biological image data sets.** *Nat Biotech* 2010, **28**(4):348–353. http://www.nature.com/nbt/journal/v28/n4/abs/nbt.1612.html, doi:10.1038/nbt.1612.
27. Peng H, Bria A, Zhou Z, Iannello G, Long F: **Extensible visualization and analysis for multidimensional images using vaa3d.** *Nat Protoc* 2014, **9**(1):193–208. doi:10.1038/nprot.2014.011.
28. Jug F, Pietzsch T, Preibisch S, Tomancak P: **Bioimage informatics in the context of drosophila research.** *Methods* 2014, **68**(1):60–73. doi:10.1016/j.ymeth.2014.04.004.
29. Amat F, Keller PJ: **Towards comprehensive cell lineage reconstructions in complex organisms using light-sheet microscopy.** *Development, Growth & Differentiation* 2013, **55**(4):563–578. doi:10.1111/dgd.12063.
30. Schmid B, Schindelin J, Cardona A, Longair M, Heisenberg M: **A high-level 3d visualization api for java and imagej.** *BMC Bioinformatics* 2010, **11:**274. doi:10.1186/1471-2105-11-274.
31. Clements RJ, Mintz EM, Blank JL: **High resolution stereoscopic volume visualization of the mouse arginine vasopressin system.** *J Neurosci Methods* 2010, **187**(1):41–45. doi:10.1016/j.jneumeth.2009.12.011.

doi:10.1186/1471-2105-15-328
Cite this article as: Wait *et al.*: Visualization and correction of automated segmentation, tracking and lineaging from 5-D stem cell image sequences. *BMC Bioinformatics* 2014 **15**:328.