User-Friendly Semiautomated Assembly of Accurate Image Mosaics in Microscopy

PHILIPPE THÉVENAZ AND MICHAEL UNSER
École polytechnique fédérale de Lausanne (EPFL), Biomedical Imaging Group, Lausanne VD, Switzerland

KEY WORDS stitching; registration; spanning tree; blending; B-spline

ABSTRACT We present a semiautomated software solution to the problem of extending the lateral field of view of a classical microscope. The initial requirements are a set of overlapping images, along with their user-provided coarse mosaic. Our solution then refines this initial mosaic in a fully automatic fashion. We rely on a highly accurate registration engine to perform the pairwise registration of the individual images, and on an efficient strategy to minimize the amount of computations while maintaining the highest possible global quality. We describe these ingredients, which we make available as a free multiplatform user-friendly software package. We also highlight why and how the specific aspects of the present microscopy application differ from those encountered while creating more common mosaics such as panoramas. Finally, we present experimental results that illustrate and validate our method on a real biological sample. We conclude by showing that we are able to reach subpixel accuracy. *Microsc. Res. Tech. 70:135–146, 2007.* © 2006 Wiley-Liss, Inc.

INTRODUCTION

The purpose of mosaicing is to condense in a single image the contents of many. The individual images are called tiles, while the resulting collection of assembled tiles forms the mosaic which covers a larger field of view than any individual tile. Mosaicing can be desirable in various contexts. In this article, the application of interest is to assemble together partial views that a microscopist may have acquired during the unplanned exploration of a sample. We provide a free software called "MosaicJ" that addresses this specific task. We shall see shortly how our context differs from two other popular applications: panoramas, and scene "painting."

To create panoramas (often of a landscape), a wider scene results from stitching together several narrow photographs; these photographs need not have been taken in sequence, and their overlap is often minimal. To "paint" a scene, one uses a video camera, which yields many more tiles (typically, one tile per frame); any two consecutive tiles in the time sequence are likely to exhibit a high degree of overlap if the panning and tilting movements of the camera were sufficiently slow, and if the scene did not contain moving objects.

Contrarily to the microscopy context, publications abound that describe computerized techniques to solve the last two popular tasks above. Photogrammetric approaches are now mature (Triggs et al., 1999). They posit that salient features can be identified in the tiles; then, the tiles are warped according to some deformation model (often a homography, sometimes combined with a lens-distortion model) to enforce that the coordinates of matched features do coincide (Kang et al., 2000; Marzotto et al., 2004). One challenging aspect is to determine which salient features do match which; another is to choose a measure of coincidence that is robust in the presence of mismatched features.

In featureless approaches, intensity-based registration is used to drive the warping of each tile. As the criterion that decides whether the tiles have been correctly warped is now local, this offers the opportunity to incorporate the local value of a pixel of the resulting global mosaic in the criterion itself (Pires and Aguiar, 2005). If done so, the optimization process faces the heavy task of adjusting the geometry of every tile simultaneously, and the mosaic content at the same time.

In the absence of a priori knowledge, an interesting challenge faced by featureless and feature-based methods alike is to discover the global topology of the tiles—to determine on which two-dimensional manifold (e.g., a sphere, a cylinder, a plane) the tiles must be assembled to build a consistent mosaic. Several propositions to meet this challenge have been put forth; most approaches rely on the tiles being captured in a well-defined time sequence, so that consecutive tiles are known to overlap and can therefore be registered pairwise. This generates an initial chain-like topology which is then used to hypothesize the existence of additional potential overlaps between tiles that may be nonconsecutive (Kang et al., 2000; Sawhney et al., 1998). If verified, these overlaps create additional links that constrain or modify the topology. For instance, the initial one-dimensional chain might end up being folded and creating a mostly planar patch, or perhaps creating a circular band (a 360° panorama). Another

*Correspondence to: Philippe Thévenaz, École polytechnique fédérale de Lausanne (EPFL), Biomedical Imaging Group, Station 17, CH–1015 Lausanne VD, Switzerland. E-mail: philippe.thevenaz@epfl.ch

Received 28 June 2006; accepted in revised form 2 October 2006

Contract grant sponsors: Center for Biomedical Imaging (CIBM), Geneva-Lausanne Universities; EPFL; Hasler, Leenaards, and Louis-Jeantet Foundations.

DOI 10.1002/jemt.20393

Published online 28 November 2006 in Wiley InterScience (www.interscience.wiley.com).

approach is to register each new tile to the current state of the mosaic as it is built; then, motion-detection techniques can be used to filter out moving objects so as to retain only the background (Winkelman and Patras, 2004).

With the possible exception of (Pires and Aguiar, 2005), once their geometry is known, the warped tiles must be blended together to build the final mosaic. This blending mechanism is generally nontrivial as it is important that the seam between two tiles remains imperceptible. Known methods involve median filtering (Winkelman and Patras, 2004), a weighted average of pixels (Grana et al., 2006), and wavelets (Hasler and Süsstrunk, 2004). One critical aspect is maintaining the proper color balance over the whole mosaic, which requires compensating for the various adjustments of the in-camera gain that may have taken place. In particular, white balancing must be made to be consistent over the whole mosaic, and vignetting must also be corrected for.

Up to now, we have described the most esthetically seductive application of mosaicing: the creation of panoramas. However, mosaicing can also be put to good use for more utilitarian tasks, such as patching together retinal fundus images (Can et al., 2002), or microscope images (Nie and Si, 2005). In the present article, our intention is to focus on the specific application of synthesizing an *extended field of view for a light microscope*. This offers us the opportunity to safely ignore some of the traditional complications encountered while dealing with the creation of a panorama:

- As microscope coverslips are planar, so are microscope mosaics; therefore, it is unnecessary to consider topological issues such as: is the panorama open-ended or circular? if it closes on itself, is it a looped strip, a hemisphere, or a full sphere?
- Zoom capabilities are uncommon in microscopes; therefore, the magnification is constant for all tiles, and the geometric transformation is captured well by a simple rigid-body model (three parameters per tile). By contrast, a homography is generally necessary for panoramas (eight parameters).
- The objective of a microscope is carefully crafted to minimize distortions; therefore, correcting for lens distortion is unnecessary. By design anyway, the area imaged by a microscope is very close to its optical axis; this is a very different situation from panoramic images built of tiles acquired through, say, a fish-eye objective lens.
- The coverslip holder of the microscope very often constrains the movement to be essentially translational. In the absence of a *xy* table, some residual rotation cannot be disregarded but any large rotation would be unexpected.
- In addition to being planar, the imaged samples are thin; therefore, there is no parallax issues.
- The cases where an extended field of view is desired mostly involve nonliving material. Therefore, we take for granted that there is no motion within the imaged scene.
- The illumination of a laboratory microscope is well controlled, while that of a panorama is much less so. Moreover, the microscopist is likely to be able to act on the camera settings, particularly to disengage the automatic gain control. This situation is often more favorable than while attempting to "paint" a panorama using even the best consumer-product camera.

Despite all these simplifications (planar topology, rigid-body transformation with limited rotation, neither motion nor parallax issues, processing with constant illumination), creating a mosaic remains a nontrivial task. In a microscopy context, the purpose of the mosaic is less that of providing an entertaining display but more that of generating a technical document amenable to measurements. Therefore, high geometric accuracy is of particular concern. We propose in this article a solution where we pay special attention to maximizing the quality of the mosaic while minimizing the associated computational costs.

We describe in the subsection "Pairwise Registration" under Materials and Methods a highly accurate alignment algorithm that we employ as pairwise-registration engine in our mosaicing application. We explain in the subsection "Global Registration Strategy" under Materials and Methods how to best exploit the availability of this pairwise registration algorithm to build a consistent mosaic that may contain multiple overlaps. Together, our algorithms have been packaged in a freely available, multiplatform, public-domain software that we present in Results. In Discussion, we illustrate the creation of a mosaic, and conclude. We then provide in an extended appendix the mathematical details of our methods; these explanations are comprehensive enough to be suitable for implementation.

MATERIALS AND METHODS

Pairwise Registration

Four major ingredients are required to automatize the process of bringing two tiles into a common coordinate system: (1) a deformation model to manipulate the relative geometry of the tiles; (2) an image interpolation model to represent the tiles in their final orientation (and also to produce intermediate trial solutions in the course of the optimization); (3) an objective criterion to measure the degree of fit; and (4) an optimizer to determine the *best* fit. Here, we pick these four ingredients from among the most suitable found in the literature.

We restrict the deformation model to be rigid-body, with a rotation defined as in Eq. (A5) in the subsection "Registration Criterion" under Appendix. If desired, this model can be further constrained to a pure translation. Far from being a limiting factor, this choice promotes the robustness of our approach and, at the same time, is consistent with the optical properties of a microscope, and befits the conditions of its usage.

We choose to perform cubic-spline interpolation to create a continuously defined image out of a discrete set of pixels. This versatile model, explained in details in the subsection "Interpolation Model" under Appendix, offers an excellent tradeoff between quality and computational cost. As corroborated by the mathematical theory of approximation, this choice ensures that, in some precise sense, the amount of arbitrariness employed to fill the gaps between pixels is minimal for a given computational budget (Meijering, 2002; Thévenaz et al., 2000). Moreover, it allows us to take advantage from multiresolution image pyramids that are fully consistent with the data model and that minimize the loss of information between a given resolution level and any coarser one (Unser et al., 1993).

The registration criterion we select is least-squares. We compute it over the region of interest defined by the geodesic 5-pixel dilation of the initial common area between the two tiles we wish to register. In principle, we consider that the first tile of the pair remains immobile (we call it the target f_{T}), and that the second tile (we call it the source f_{S}) is rotated and translated relatively to the target. To estimate the registration criterion for a given rotation and translation, we scan the target in raster fashion and compute the difference between each target pixel and the corresponding interpolated value from the source. These differences are squared and summed, as captured by Eq. (A4) in the subsection "Registration Criterion" under Appendix. In practice, for efficiency reasons explained in the subsection "Registration Criterion" under Appendix, we permute the role of f_{S} and f_{T}. This criterion is optimally robust to white Gaussian noise under the hypothesis that the parts brought into correspondence have the same brightness and contrast within f_{S} and f_{T}, which is fortunately the case in a microscopy context (see Introduction).

To determine the best fit, we have chosen the classical Levenberg-Marquardt optimizer sketched in the subsection "Optimizer" under Appendix. This optimizer is reliable, fast, and has a manageable complexity. It has been developed specifically to solve least-squares problem. It requires the availability of partial derivatives, which happen to be readily at our disposal, thanks to the cubic-spline data model we are using. This iterative optimizer is particularly efficient since it is able to simultaneously adjust all parameters, from the very first trial onwards, and since it converges in quadratic fashion when sufficiently close from the solution. However, it is not very robust; this is dealt with by performing the optimization according to the coarse-to-fine strategy detailed in the subsection "Multiresolution" under Appendix. Clearly, it also helps if the initial condition to be refined by the optimizer is already close to the optimal solution.

In summary, we have combined some of the best solutions and algorithms available to date for subpixel rigid-body registration, and we are able to recover with great accuracy the relative orientation and position of a pair of overlapping tiles. The degree of accuracy depends in great part on the quality of the interpolation model, which removes as much arbitrariness as possible from the subpixel displacement of an image described in terms of discrete pixels. The robustness is dramatically improved by the use of a multiresolution approach which happens to be consistent with the interpolation model. The speed of convergence is fast because the optimizer is able to simultaneously process all registration parameters.

For fully overlapping images and in noiseless conditions, we showed in (Thévenaz et al., 1998) that it is possible to reach the accuracy of a thousandth of a pixel. Even when the data are corrupted by noise of equal power, which corresponds to 0 dB signal-to-noise ratio, about a tenth-pixel accuracy could be reached; these results were obtained over full-size (256 × 256) images. When the amount of overlap dwindles, however, the accuracy starts depending more on the data than on the algorithm itself: in the presence or absence of details, of structure, of texture, or of directionality of the image-defining elements, the accuracy can be locally excellent or terrible. To some degree, this accuracy may be monitored (Aguiar and Moura, 2001; Shi and Tomasi, 1994), but it is impossible to compensate for an absence of data. Nevertheless, as we minimize the reduction of overlap by computing the least-squares registration criterion over the largest-possible area (Pires and Aguiar, 2004), we have every reason to believe that our algorithm degrades gracefully with a gradual reduction in the amount of localizing features.

To obtain good registration results, we have observed in practice that it is favorable to explore at least one additional level of resolution beyond the finest one. We believe that the cause of the improvement is the smoothing stage implied by the coarsening of the resolution, since it offers the opportunity to remove noise and details, and greatly improves the performance. Therefore, it is essential that the size of the area of overlap be sufficiently large to be able to neglect boundary effects at the coarsest level. We suggest that the smallest dimension of the circumscribed rectangle of the region of interest be at least 20–30 pixels. This rule of thumb depends on geometry alone; it must be complemented by the conditions on the quality and number of features expressed earlier.

TABLE 1. All potentially available registration results

	A	B	C
A		$\mathbf{T}_{B\to A}$	$\mathbf{T}_{C\to A}$
B	$\mathbf{T}_{A\to B}$		$\mathbf{T}_{C\to B}$
C	$\mathbf{T}_{A\to C}$	$\mathbf{T}_{B\to C}$	

Global Registration Strategy

Given N tiles and our pairwise registration algorithm, we could initiate the construction of a mosaic by blindly attempting to register each tile to the $(N - 1)$ remaining ones, but there are several difficulties with this naive approach: (1) lack of any reasonable initial fit to refine; (2) high computational cost; and (3) potential absence of overlap. Even if these three obstacles are overcome, the resulting geometric relations between tiles will be noisy, so that it remains to see how to coalesce contradicting pairwise registration results into a consistent single mosaic.

We illustrate this by a simple thought experiment involving the three mutually overlapping tiles $\{A, B, C\}$. Registering the source image B with respect to the target image A results in the transformation matrix $\mathbf{T}_{B\to A}$ expressed in homogenous coordinates to accommodate for translation. Similar considerations are used to complete Table 1; unfortunately, because of the presence of noise in the data and because of other difficulties, any registration algorithm may experience lapses that may result in $\mathbf{T}_{B\to A}$ being not exactly equal to its inverse $\mathbf{T}_{A\to B}^{-1}$. Let us now choose A to anchor the mosaic. Then, the orientation of the tile B with respect to A can be set either by $\mathbf{T}_{B\to A}$, by $\mathbf{T}_{A\to B}^{-1}$, by $(\mathbf{T}_{C\to A}\, \mathbf{T}_{B\to C})$, and by a score of other combinations. Which one is best? How do we orient C in a consistent fashion once B is set up?

To resolve these issues, we assume that the quality of registering a pair of tiles varies like a measure of the surface of their initial overlap: even before actually registering them, we assume that a high initial overlap will result in a high final registration accuracy. Then, we simply do not undertake to register tiles with the least amount of overlap. Since the measure of overlap between two tiles does not depend on their order, we arbitrarily retain (compute) only one of $\{\mathbf{A}_{A\to B}, \mathbf{A}_{B\to A}\}$, and determine the discarded transformation as the inverse of the retained one. This strategy drastically reduces costs from $O(N^2)$ to $O(N)$, more precisely, from $(N^2 - N)$ to $(N - 1)$.

In doing so, we take for granted that some rough alignment of the tiles has already been established. This rough alignment provides two necessary ingredients to our method of building a mosaic: an initial fit to be refined, and an initial measure of overlap—which may default to 0 if, according to the rough mosaic, two tiles happen not to overlap at all. The measure of overlap can then be used to build a connectivity matrix, which in turn fully describes a graph in which the vertices represent the tiles, and where the presence of an edge between two vertices indicates that the two corresponding tiles do overlap.

By avoiding to register tiles with the least amount of overlap, in effect we remove edges from the graph. For the mosaic to remain well conditioned, however, we have to ensure that the graph remains connected; in other words, we have to ensure that two arbitrary tiles are always related by some transformation matrix $\mathbf{T}_0$, or at least by a chain of such transformations $\mathbf{T} = (\mathbf{T}_1\mathbf{T}_2 \ldots \mathbf{T}_K)$. Therefore, it is not advisable to remove some critical edges, even if they correspond to a small amount of overlap. Alternatively, we can also build the graph by progressively adding those edges that have the largest amount of overlap first; adding one edge corresponds to performing one pairwise registration operation. In this case, we must avoid creating loops (or cycles) in the graph, so that it never happens that two or more (possibly contradicting) transformations or chains of transformations can relate a pair of tiles.

A graph without cycles is called a tree; a tree that links every vertex is called a spanning tree. If we associate a weight (given by the measure of overlap) to each edge of an arbitrary graph (not necessarily a tree), then, among all possible trees that span the vertices of the graph, any one that collects the largest weights is called a maximum spanning tree (Marzotto et al., 2004). More than one may exist, for example when all the weights are equal; but the Kruskal's algorithm given in the subsection "Kruskal's Algorithm" under Appendix is sure to find one (Kruskal, 1956). Thus, the global registration strategy works as follows:

1. Obtain a rough mosaic;
2. Measure the overlaps of each pair of tiles and build a weighted graph G;
3. Register all pairs of tiles that are linked by an edge in the maximum spanning tree of G;
4. Choose some arbitrary tile as anchor for the mosaic;
5. For each vertex of G, determine the path that relates this vertex to the anchor and compute the corresponding chain of transformations to bring the tile into the coordinate system of the anchor tile.

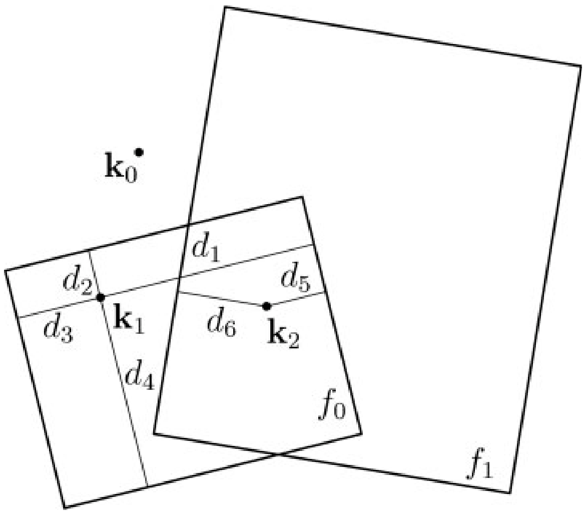

Fig. 1. Weights for two overlapping tiles.

The major benefit of this strategy is to maximize the registration accuracy while minimizing the computational burden, under the hypothesis that the degree of accuracy is driven by the amount of overlap. However, because there is no cycle in a maximum spanning tree, there is no redundancy of transformations either; therefore, we note that this strategy is vulnerable to a registration failure since the absence of redundancy does not allow us to check for failure, even less to correct for it. As the placement of each tile depends on a chain of transformation, no weak link is allowed; this is the reason why it is essential that our pairwise registration algorithm be of very high quality, such as is the case for the algorithm we presented in the subsection "Pairwise Registration" under Materials and Methods. We note that the propagation of errors in a chain of transformation has been studied in (Zagorodnov and Ramadge, 2000), which establishes that a few additional measurements (cycles) can greatly reduce the global error; unfortunately, this study considers translations only (which have a commutative structure), and it is unclear how to extend it to rigid-body transformations (which have a noncommutative structure).

Blending

Once the operations of the subsection "Global Registration Strategy" under Materials and Methods have been performed, we are ready to blend the tiles together. For each pixel of the final mosaic f, we compute a weighted average of the contributions of the appropriately transformed tiles f_n as given by

$$f(\mathbf{k}) = \frac{\sum_{n=0}^{N-1} w_n(\mathbf{k}) f_n(\mathbf{k})}{\sum_{n=0}^{N-1} w_n(\mathbf{k})}.$$

If the location $\mathbf{k}$ of the pixel happens to fall outside the support of a transformed given tile f_n, then we set the weight $w_n(\mathbf{k})$ to zero; else, we weigh the contribution by the distance to the closest boundary of the support of f_n. We illustrate in Figure 1 a case involving the two

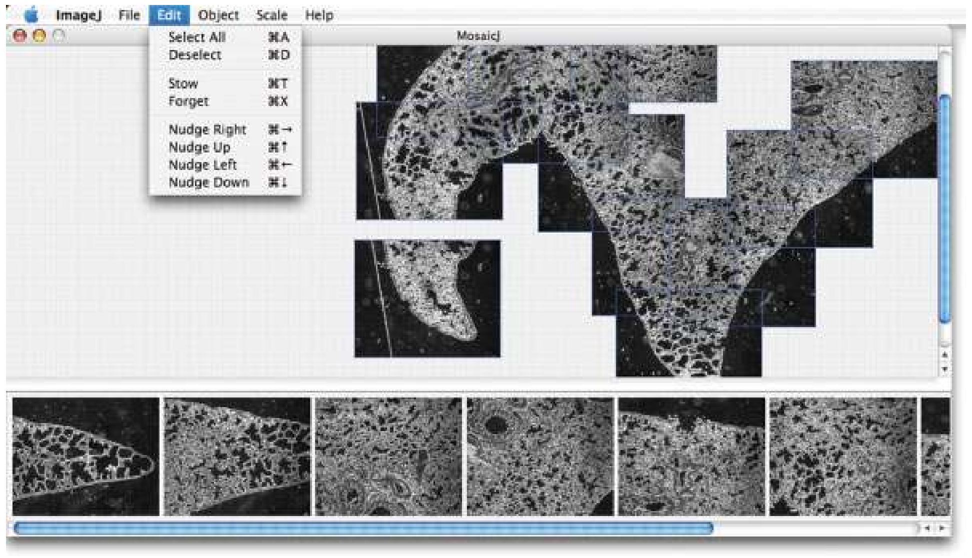

Fig. 2. Interface of MosaicJ. The workspace window contains two sections: the upper, and the lower one. The upper section is the gridded area where the user can interactively arrange tiles. (Here, several tiles have already been placed, and one additional is in the process of being placed, too. The tile boundaries have been outlined for clarity.) The lower section contains tiles that have not yet been arranged. The user interacts with MosaicJ by the way of menus and mouse-based operations. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

tiles f_0 and f_1. In this toy example, $\mathbf{k}_0$ does not belong to the support of either f_0 or f_1, so that $f(\mathbf{k}_0) = 0/0$ is undefined; when this happens, we set the mosaic to an arbitrary background value, typically $f(\mathbf{k}_0) = 0$. As $\mathbf{k}_1$ belongs to the support of f_0 but not to that of f_1, we set $w_0(\mathbf{k}_1) = \min(d_1,d_2,d_3,d_4)$ and $w_1(\mathbf{k}_1) = 0$, so that $f(\mathbf{k}_1) = d_2 f_0(\mathbf{k}_1)/d_2 = f_0(\mathbf{k}_1)$. Finally, we have that $w_0(\mathbf{k}_2) = \min(\ldots) = d_5$ and $w_1(\mathbf{k}_2) = d_6$, so that $f(\mathbf{k}_2) = (d_5 f_0(\mathbf{k}_2) + d_6 f_1(\mathbf{k}_2))/(d_5 + d_6)$. In this last example where $d_5 < d_6$, we see that f_0 contributes less than f_1 at the location $\mathbf{k}_2$ because of a greater proximity to the boundary of the tile.

To compute the weights $w_n(\mathbf{k})$, it is enough to perform a distance transform (Grana et al., 2006) of the support of the nth tile, which is straightforward if the tile is rectangular. Then, the geometric transformation $\mathbf{T}_n$ that has to be applied to the tile is applied to this distance transform too. We observe that, while a good interpolation model is required to build the mosaic, in practice nearest-neighbor interpolation is sufficient to deal with the weights themselves.

RESULTS

To assemble a mosaic out of overlapping individual tiles, we have developed a software called "MosaicJ" that implements the methods described in this article. We make this software freely available to the community (Thévenaz, 2006). It is provided in the form of a plugin for the public-domain general-purpose image-processing and analysis package called "ImageJ" (Rasband, 1997–2006). As it is written in the Java language, it is readily available for every popular operating system; unfortunately, it also inherits the memory limitations of some of the current Java virtual machines. In particular, because the treatment of each tile is individually disk-based, the limiting factor here is the final size of the mosaic rather than the number of tiles.

Our solution is semiautomated. The initial rough positioning of the tiles must be performed by the user, while MosaicJ provides the final delicate adjustments. The interactive stage is translational only; meanwhile, the automatized refinement stage allows for subpixel translations, and optional rotations.

MosaicJ has been written to be as user-friendly as possible and conceals every technical aspects and tuning parameters of the pairwise registration algorithm. As seen in Figure 2, it presents to the user a workspace over which tiles can be dragged, hierarchically grouped, and ungrouped, hidden or restored, at will. The whole documentation is available online (Thévenaz, 2006). The tiles are shown as thumbnails which are computed according to the precepts of the subsection "Multiresolution" under Appendix; the magnification factor can be modified to accommodate small and large tiles alike. The dimension of the workspace automatically adjusts itself to the user's needs.

The tiles can have independent sizes and types. Upon loading a color image, we convert it to grayscale by first computing the scatter matrix of the colors observed over the whole image, and then by performing a principal-component analysis to identify which linear combination of red, green, and blue components results in the largest contrast. We control the grayscale dynamics by ensuring that the weights of the linear combination add up to unity, and that the most important weight is positive. An independent scatter matrix is computed for each color image; should it possess no well-defined eigenvalues, we default to the color-to-grayscale conversion of the CCIR-709 standard. To accommodate the least-squares criterion of the subsection "Pairwise Registration" under Materials and Methods, we always perform registration on grayscale

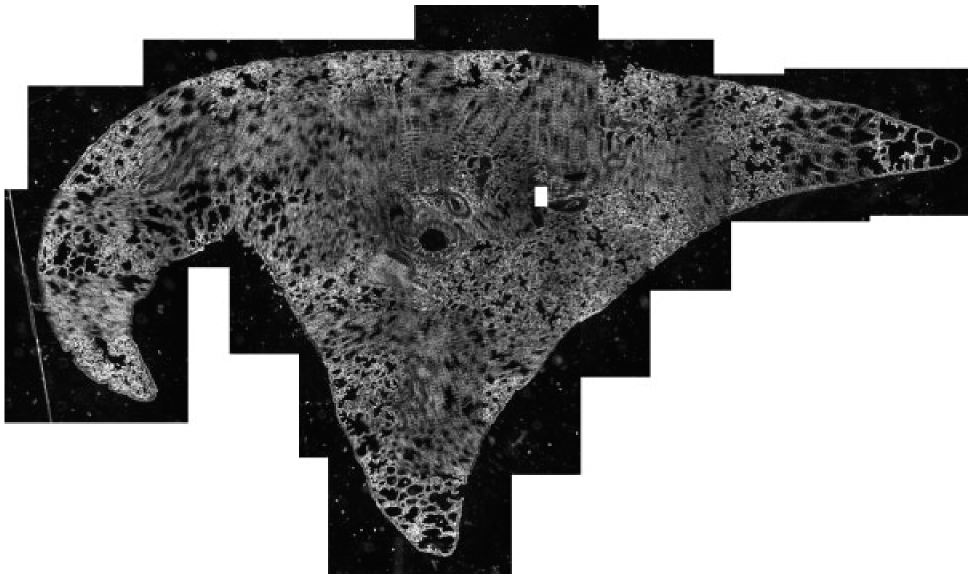

Fig. 3. Manual mosaic. Since no precise alignment is required at this stage, the puzzle of mosaic pieces can be quickly assembled thanks to the interface shown in Figure 2.

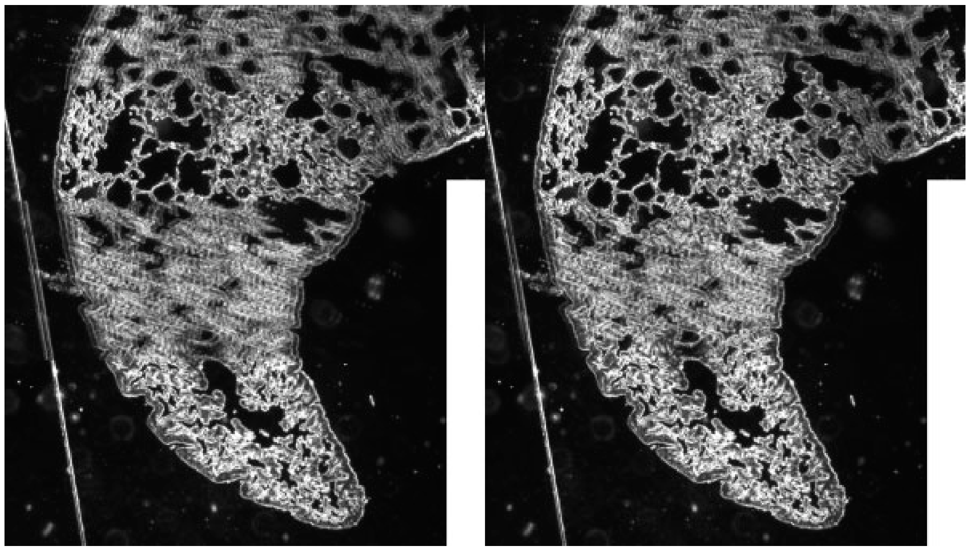

Fig. 4. Cut-out of the coarse mosaic of Figure 3. On the left side, to help discern the boundaries of the tiles, we performed simple averaging to blend them. On the right side, we applied the blending mechanism described in the subsection “Blending” under Materials and Methods.

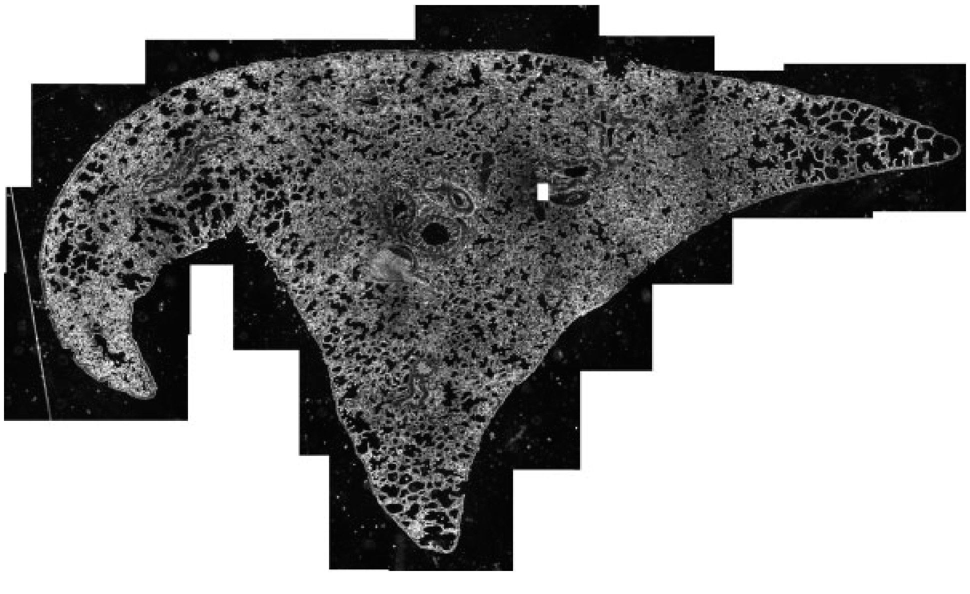

Fig. 5. Mosaic after automatic refinement. The twenty tiles of size (636 × 512) resulted in a mosaic of size (3,332 × 1,957).

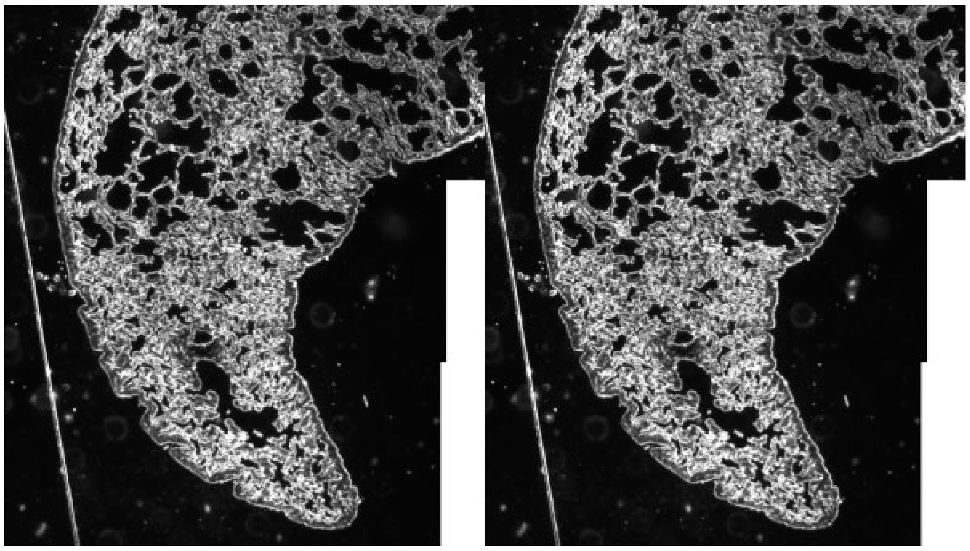
Fig. 6. Cut-out of the mosaic of Figure 5. On the left side, the boundaries of the tiles remain hard to discern even though we performed a simple averaging to blend them. On the right side, the blending mechanism described in the subsection "Blending" under Materials and Methods has been applied and the tiles connect seamlessly.

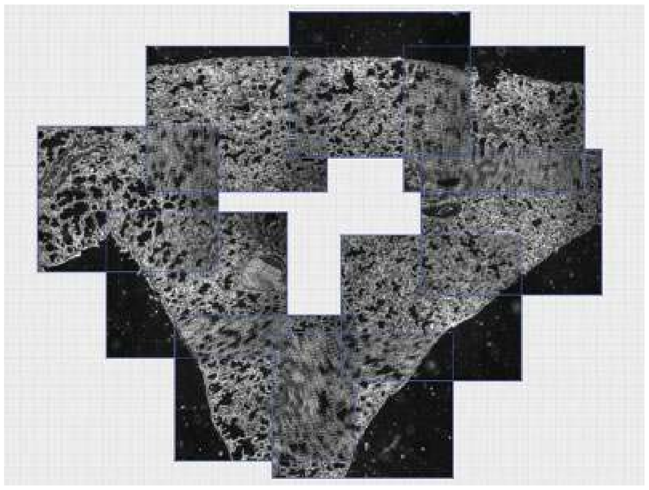
Fig. 7. Initial loop of nine tiles. The topmost tile has been duplicated, and the circular chain of tiles has been made linear by preventing the occurrence of a link between the two duplicates. The tile boundaries have been outlined for clarity. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

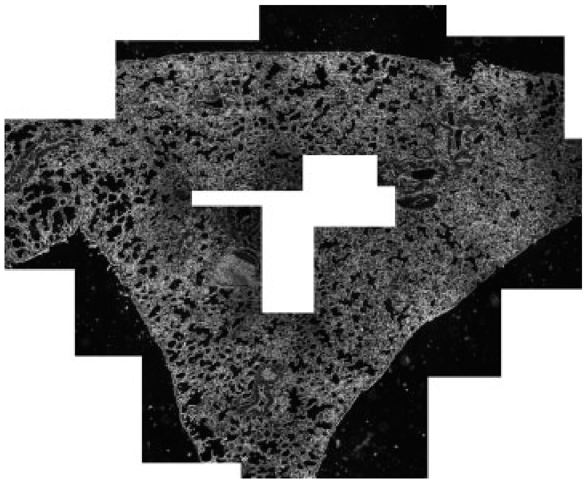
Fig. 8. Final loop. The two ends of the linear chain are in perfect match.

images; we also present these grayscale images to the user to provide feedback on the features really used for registration. If need be, however, colors are restored when the final mosaic is created.

DISCUSSION

Illustration

After being completed, the mosaic of Figure 2 consists of twenty tiles of size (636×512) each. They collectively represent a section of the lung of a cat. We show in Figure 3 the mosaic as it would appear if only the manual coarse registration would have been performed. We also present a cut-out of this coarse mosaic in Figure 4 to highlight the degree of initial misalignment of the tiles. One can observe that blending alone is already responsible for some increase in the perceptive quality of the mosaic, even though no geometric effort has taken place yet; this perceptual effect is an important ingredient of panoramas.

These results are to be contrasted with those shown in Figure 5, where the automatic refinement has been performed. There, although the position of the outer tiles can be easily inferred by how they contrast with the empty background, their edges cannot be prolonged inside the data where no seam can be discerned at all. Scrutinizing the cut-outs we present in Figure 6 reveals no subtler defects, even in close-up view. Additional examples can be found in (Thévenaz et al., 2006).

Validation

To check experimentally that the residual inaccuracies of our registration method do not grow unduly when they are propagated in a chain of transformations, we have removed some tiles from the illustration of the subsection "Illustration" under Discussion, so that only the loop of nine tiles shown in Figure 7 remains. Moreover, we have duplicated one tile, which

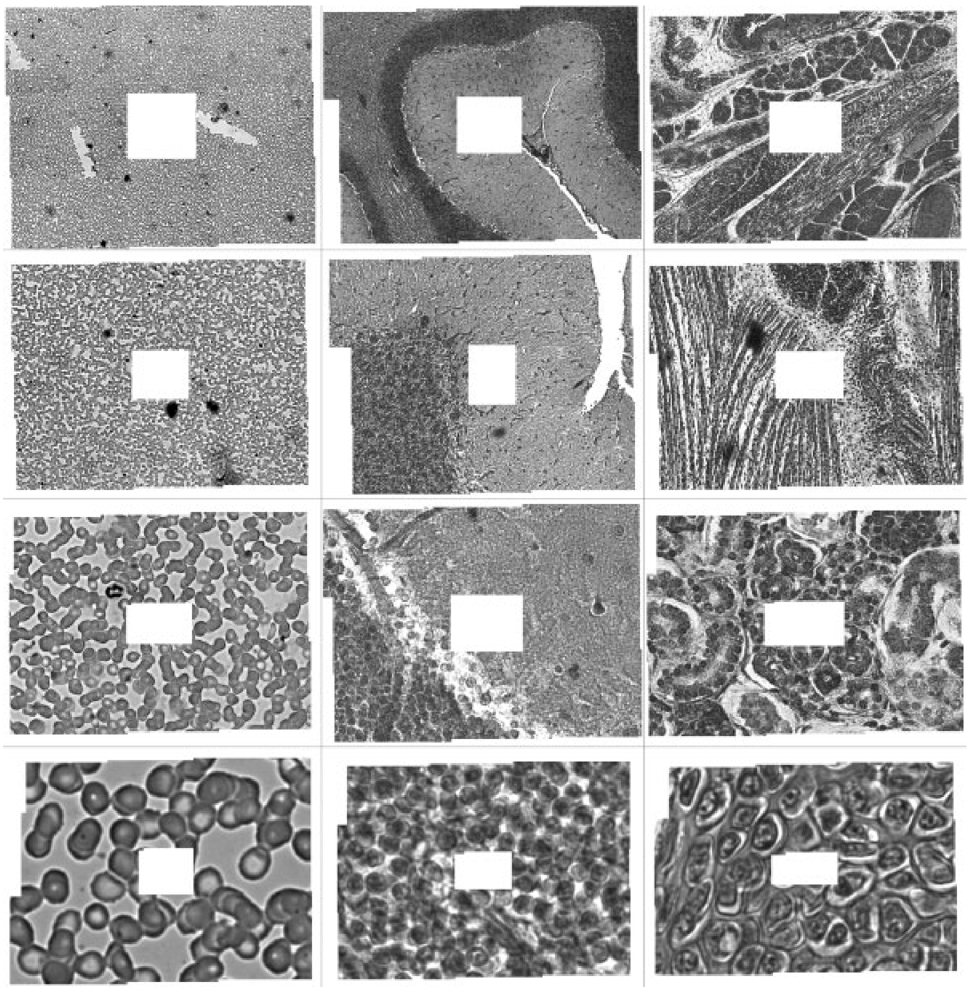

Fig. 9. Examples of mosaic with eight tiles in a loop. Each line from top to bottom corresponds to a microscope objective with a different magnification factor: 4×, 10×, 40×, and 100×. Each column from left to right corresponds to a different specimen: Blood smear, cerebellum, and mouse pup. Because the loop of tiles circles the central part but does not cover it, an empty region remains in the center of each mosaic.

results in a chain of tiles that can be described as $\{A',B,C,D,E,F,G,H,I,A''\}$, linked by the nine pairwise transformations $\{\mathbf{T}_{A'\to B}, \mathbf{T}_{B\to C}, \ldots, \mathbf{T}_{H\to I}, \mathbf{T}_{I\to A''}\}$. Since we represent these transformations by matrices in homogenous coordinates, translations are incorporated in the representation. Then, a signature of perfect registration accuracy would be recognized if the matrix product $\mathbf{T}_0 = (\mathbf{T}_{I\to A''}\,\mathbf{T}_{H\to I}\,\mathbf{T}_{G\to H}\cdots\mathbf{T}_{B\to C}\,\mathbf{T}_{A'\to B})$ would result in the homogenous identity matrix—with null translation. Here, the duplicated tile is $A' = A''$, and the loop of tiles is artificially cut between them.

With the data of Figure 7, we observe that the resulting $\mathbf{T}_0$ consists of a residual translation of norm $\|\mathbf{t}_0\| = 0.30$, in pixel units. (As our microscope possesses an *xy* table, we have ignored rotations.) We show in Figure 8 the mosaic that was built after automatic refinement of the initial situation of Figure 7. No seam can be discerned, and the registration looks perfect visually.

We have repeated this experiment for multiple slides. We have taken into account a combination of conditions, such as the material being imaged and the magnification factor; to this end, we have acquired four-hundred-eighty tiles that can be regrouped in different ways to build families of mosaics. Each individual mosaic consists of eight tiles $\{A',B,C,D,E,F,G,H,A''\}$, where $A' = A''$ as before. The dimension of each tile is now (640 × 480). We have considered groups of five mosaics for each specimen and magnification. We took advantage of three specimens: a blood smear, a slice of cerebellum, and a slice of a mouse pup. We had access to four microscope objectives: 4×, 10×, 40×, and 100×. We present in Figure 9 a few of the sixty mosaic loops that we built for this experiment.

TABLE 2. Accuracy (in pixel units) for mosaic loops regrouped by magnification factor

Magnification	Residual translation
4×	0.114 ± 0.086 (0.097)
10×	0.047 ± 0.033 (0.030)
40×	0.081 ± 0.053 (0.070)
100×	0.337 ± 0.217 (0.268)

We summarize in Tables 2 and 3 the observed accuracies. We report the precision as one standard deviation; the median error is given in parentheses. There are fifteen mosaics per magnification factor in Table 2, and twenty mosaics per specimen type in Table 3. Once pooled together, the sixty mosaics result in an average residual translation of 0.145 pixel units, and in a standard deviation of 0.164; the median is 0.089.

We see by all accounts that the accuracy remains subpixel, despite the fact that the reported error is

TABLE 3. Accuracy (in pixel units) for mosaic loops regrouped by specimen

Specimen	Residual translation
Blood smear	0.107 ± 0.109 (0.082)
Cerebellum	0.160 ± 0.193 (0.092)
Mouse pup	0.166 ± 0.180 (0.109)

compounded here from eight sequential registration operations. At the highest magnification, however, the field of depth is very small, and therefore some of the image registration assumptions do not hold as good. Tissue thickness nonuniformity, out-of-focus problems, slide surface nonuniformity, lens distortions, loss of resolution at the corners of the image, illumination uniformity, lens shading all become more significant at high magnification and numerical aperture and tend to reduce accuracy.

Obtaining comparable results by hand would be a daunting task that would require a lot of attention from the user, along with a nonstandard interface to deal with subpixel displacements. At the occasion of providing the initial rough alignment of tiles acquired through the 100× microscope objective, we experienced firsthand the human difficulty to be accurate: at this magnification, fuzziness prevails and tiles accommodate to a wide range of relative positions without impacting at all the visual quality of their match; yet, only one position is correct. MosaicJ proved able to find this position with much better accuracy than we could have ever done.

CONCLUSION

We have developed a semiautomated method that extends the field of view of microscope images by assembling partial views. Its intended audience is the microscopist who desires to scan a large area while acquiring a series of partial views, but who does not wish to—or cannot—plan the path of the scan. In a first stage, this freedom is dealt with by interactive manipulation of the resulting partial views. In a second stage, the position of the partial views is refined by a fully automatic pairwise registration process which has been developed while paying special attention to ensure the highest registration accuracy while minimizing the computational costs. Our solution is made freely available in the form of a plugin for ImageJ.

APPENDIX

In this section, we provide a formal description of our registration algorithm. To ensure universality, we give it in the form of a mathematical recipe that is fully consistent with the Java version (Thévenaz, 2006). The motivations and benefits of the various ingredients can be found in the relevant references: cubic-spline interpolation model (Thévenaz et al., 2000; Unser, 1999), quasi-Newton optimization strategy (Thévenaz et al., 1998), multiresolution image representation (Unser et al., 1993), and maximum spanning tree (Kruskal, 1956). These ingredients incorporate some of the best solutions available in the domain of image registration. The resulting global algorithm is optimized and streamlined for computational performance and accuracy. It is particularly appropriate in the context of a microscopy application.

Interpolation Model

Let f be a $(K_1 \times K_2)$ image given by its samples $f[k_1, k_2]$ with $k_1 \in [0 \ldots K_1 - 1]$ and $k_2 \in [0 \ldots K_2 - 1]$. The $(K_1 \times K_2)$ cubic-spline coefficients c of the image f are computed as follows (Thévenaz et al., 2000):

$$\forall n_2 \in [0 \ldots K_2 - 1]: \left[\begin{array}{ll} & c[0, n_2] \leftarrow \frac{(1-z_0)^4 \left(f[0,n_2] + \sum_{n_1=1}^{K_1-2} \left(z_0^{n_1} + z_0^{2K_1-n_1-2} \right) f[n_1,n_2] + z_0^{K_1-1} f[K_1-1,n_2] \right)}{z_0^2 - z_0^{2K_1}} \\ n_1 = 1 \rightarrow K_1 - 1 & c[n_1, n_2] \leftarrow \frac{(1-z_0)^4}{z_0^2} f[n_1, n_2] + z_0 c[n_1 - 1, n_2] \\ & c[K_1 - 1, n_2] \leftarrow \frac{-z_0}{1-z_0^2} (z_0 c[K_1 - 2, n_2] + c[K_1 - 1, n_2]) \\ n_1 = K_1 - 2 \rightarrow 0 & c[n_1, n_2] \leftarrow z_0 (c[n_1 + 1, n_2] - c[n_1, n_2]) \end{array}\right. \tag{A1}$$

$$\forall n_1 \in [0 \ldots K_1 - 1]: \left[\begin{array}{ll} & c[n_1, 0] \leftarrow \frac{c[n_1,0] + \sum_{n_2=1}^{K_2-2} \left(z_0^{n2} + z_0^{2K_2-n_2-2} \right) c[n_1,n_2] + z_0^{K_2-1} c[n_1,K_2-1]}{1 - z_0^{2K_2-2}} \\ n_2 = 1 \rightarrow K_2 - 1 & c[n_1, n_2] \leftarrow c[n_1, n_2] + z_0 c[n_1, n_2 - 1] \\ & c[n_1, K_2 - 1] \leftarrow \frac{-z_0}{1 - z_0^2} (z_0 c[n_1, K_2 - 2] + c[n_1, K_2 - 1]) \\ n_2 = K_2 - 2 \rightarrow 0 & c[n_1, n_2] \leftarrow z_0 (c[n_1, n_2 + 1] - c[n_1, n_2]), \end{array}\right. \tag{A2}$$

where $z_0 = \sqrt{3} - 2$. If desired, the sums that belong to the definition of $c[0, n_2]$ and $c[n_1, 0]$ can be truncated because z_0^n vanishes exponentially for increasing n. For example, $|z_0^7| < 10^{-4} < |z_0^6|$. Similarly, the terms $z_0^{2K_1-n_1-2}$ and $z_0^{2K_2-n_2-2}$ are often negligible for large K_1 and K_2. Collectively, the coefficients c then determine the continuously defined function

$$\forall x_1,x_2 \in \mathbb{R}^2 : f(x_1,x_2) = \sum_{n_2=\lfloor x_2\rfloor-1}^{\lceil x_2\rceil+1} \beta^3(x_2-n_2) \times \sum_{n_1=\lfloor x_1\rfloor-1}^{\lceil x_1\rceil+1} c[n_1,n_2]\beta^3(x_1-n_1), \tag{A3}$$

where the cubic B-spline β^3 is given by (Unser, 1999)

$$\beta^3(x) = \begin{cases} \frac{1}{2}x^2(|x|-2)+\frac{2}{3} & |x|<1 \\ \frac{1}{6}(2-|x|)^3 & |x|<2 \\ 0 & 2 \le x. \end{cases}$$

In practice, to determine Eq. (A3), it is less efficient to compute the expression above four times for the four arguments $\{x - \lfloor x\rfloor + 1, \ldots, x - \lceil x\rceil - 1\}$ than to simultaneously compute a collection of four values $\{\beta^3(\xi+1), \beta^3(\xi), \beta^3(\xi-1), \beta^3(\xi-2)\}$ for $\xi = x - \lfloor x\rfloor \in [0,1[$, according to the following algorithm:

$$\beta^3 : \left[\begin{aligned} \xi_1 &= 1-\xi \\ \xi_2 &= \xi^2 \\ \beta^3(\xi+1) &= \frac{1}{6}\xi_1\xi_1\xi_1 \\ \xi_1 &\leftarrow \xi_1+1 \\ \beta^3(\xi) &= \frac{2}{3}-\frac{1}{2}\xi_1\xi_2 \\ \beta^3(\xi-2) &= \frac{1}{6}\xi\xi_2 \\ \beta^3(\xi-1) &= 1-\beta^3(\xi+1)-\beta^3(\xi)-\beta^3(\xi-2). \end{aligned} \right.$$

The would-be out-of-bounds indices of c can be folded back to the domain $[0 \ldots K_1-1] \times [0 \ldots K_2-1]$ by the (perhaps repeated) application of

$$\forall n_2 \in \mathbb{Z} : \begin{cases} c[-n_1,n_2] = c[n_1,n_2] \\ c[K_1-1+n_1,n_2] = c[K_1-1-n_1,n_2] \end{cases}$$

$$\forall n_1 \in \mathbb{Z} : \begin{cases} c[n_1,-n_2] = c[n_1,n_2] \\ c[n_1,K_2-1+n_2] = c[n_1,K_2-1-n_2], \end{cases}$$

which can also be written

$$\left[\begin{aligned} p[k] &= \begin{cases} 0 & K=1 \\ |k|-(2K-2)(|k|/(2K-2)) & 1<K \end{cases} \\ q[k] &= \begin{cases} p[k] & 0 \le p[k] < K \\ 2K-2-p[k] & K \le p[k] < 2K-2 \end{cases} \\ c[k] &\leftarrow c[q[k]], \end{aligned} \right.$$

where the division $|k|/(2K-2)$ must be understood as an integer division. If the samples of f are extended likewise, then it can be observed that $\forall k_1,k_2 \in \mathbb{Z}$: $f[k_1,k_2] = f(k_1,k_2)$.

Registration Criterion

The criterion we wish to minimize is given by

$$\begin{aligned} J(\theta,\mathbf{t}) &= \frac{1}{2}\sum_{\mathbf{k}\in\mathbb{Z}^2}(f_S(\mathbf{R}_\theta(\mathbf{k}-\mathbf{x}_0)+\mathbf{x}_0+\mathbf{t})-f_T(\mathbf{k}))^2 \\ &\approx \frac{1}{2}\sum_{\mathbf{k}\in\mathbb{Z}^2}(f_S(\mathbf{k})-f_T(\mathbf{R}_{-\theta}(\mathbf{k}-\mathbf{x}_0)+\mathbf{x}_0-\mathbf{R}_{-\theta}\mathbf{t}))^2, \end{aligned} \tag{A4}$$

where θ is a rotation angle and where $\mathbf{t}$ is a translation. The source and target images are f_S and f_T, respectively. The rotation center is $\mathbf{x}_0$. The rotation matrix $\mathbf{R}_\theta$ is defined by

$$\mathbf{R}_\theta = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}. \tag{A5}$$

The gradient of J with respect to θ and $\mathbf{t}$ is given by

$$\begin{aligned} \frac{\partial J}{\partial\theta} &= \sum_{\mathbf{k}\in\mathbb{Z}^2}(f_S(\mathbf{R}_\theta(\mathbf{k}-\mathbf{x}_0)+\mathbf{x}_0+\mathbf{t})-f_T(\mathbf{k})) \\ &\quad\times(\nabla f_S(\mathbf{R}_\theta(\mathbf{k}-\mathbf{x}_0)+\mathbf{x}_0+\mathbf{t}))^\top\mathbf{R}_{\theta+\frac{\pi}{2}}(\mathbf{k}-\mathbf{x}_0) \\ &\approx \sum_{\mathbf{k}\in\mathbb{Z}^2}(f_S(\mathbf{k})-f_T(\mathbf{R}_{-\theta}(\mathbf{k}-\mathbf{x}_0)+\mathbf{x}_0-\mathbf{R}_{-\theta}\mathbf{t})) \\ &\quad\times(\nabla f_S(\mathbf{k}))^\top\mathbf{R}_{\frac{\pi}{2}}(\mathbf{k}-\mathbf{x}_0-\mathbf{t}) \\ \frac{\partial J}{\partial t_1} &= \sum_{\mathbf{k}\in\mathbb{Z}^2}(f_S(\mathbf{R}_\theta(\mathbf{k}-\mathbf{x}_0)+\mathbf{x}_0+\mathbf{t})-f_T(\mathbf{k})) \\ &\quad\times(\nabla f_S(\mathbf{R}_\theta(\mathbf{k}-\mathbf{x}_0)+\mathbf{x}_0+\mathbf{t}))^\top\begin{pmatrix}1\\0\end{pmatrix} \\ &\approx \sum_{\mathbf{k}\in\mathbb{Z}^2}(f_S(\mathbf{k})-f_T(\mathbf{R}_{-\theta}(\mathbf{k}-\mathbf{x}_0)+\mathbf{x}_0-\mathbf{R}_{-\theta}\mathbf{t})) \\ &\quad\times(\nabla f_S(\mathbf{k}))^\top\begin{pmatrix}1\\0\end{pmatrix} \\ \frac{\partial J}{\partial t_2} &= \sum_{\mathbf{k}\in\mathbb{Z}^2}(f_S(\mathbf{R}_\theta(\mathbf{k}-\mathbf{x}_0)+\mathbf{x}_0+\mathbf{t})-f_T(\mathbf{k})) \\ &\quad\times(\nabla f_S(\mathbf{R}_\theta(\mathbf{k}-\mathbf{x}_0)+\mathbf{x}_0+\mathbf{t}))^\top\begin{pmatrix}0\\1\end{pmatrix} \\ &\approx \sum_{\mathbf{k}\in\mathbb{Z}^2}(f_S(\mathbf{k})-f_T(\mathbf{R}_{-\theta}(\mathbf{k}-\mathbf{x}_0)+\mathbf{x}_0-\mathbf{R}_{-\theta}\mathbf{t})) \\ &\quad\times(\nabla f_S(\mathbf{k}))^\top\begin{pmatrix}0\\1\end{pmatrix}. \end{aligned}$$

TABLE A1. Tabulated values of a cubic B-spline and of its derivative at integer arguments

k	-1	0	1
$\beta^3(k)$	$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$
$\dot\beta^3(k)$	$\frac{1}{2}$	0	$-\frac{1}{2}$

The second equalities would be exact if integration over $\mathbb{R}^2$ would be used instead of summation over $\mathbb{Z}^2$. A remarkable property of choosing the second forms is that they use the spatial gradient of f_S without depending on the actual value of either θ or **t**; this gradient can therefore be precomputed (Thévenaz et al., 1998). Its expression is

$$\forall k_1 \in [0\ldots K_1-1], k_2 \in [0\ldots K_2-1] : \nabla f_S(\mathbf{k}) = \begin{pmatrix} \sum_{n_2=k_2-1}^{k_2+1} \beta^3(k_2-n_2) \sum_{n_1=k_1-1}^{k_1+1} c_S[n_1,n_2] \dot\beta^3(k_1-n_1) \\ \sum_{n_2=k_2-1}^{k_2+1} \dot\beta^3(k_2-n_2) \sum_{n_1=k_1-1}^{k_1+1} c_S[n_1,n_2] \beta^3(k_1-n_1) \end{pmatrix},$$

where $\dot\beta^3(x) = \frac{\mathrm{d}\beta^3(x)}{\mathrm{d}x}$. To further reduce the computational burden of ∇f_S, it is interesting to note that only integer arguments participate in the evaluation of β^3 and $\dot\beta^3$. Therefore, ∇f_S is no more than a filtered version of c_S. The relevant filter coefficients are given in Table A1.

Optimizer

Our optimization procedure is iterative. At each iteration, the goal is to find an update (Δθ, Δ**t**) that refines the initial solution (θ,**t**), in such a way that $J(\theta + \Delta\theta, \mathbf{t} + \Delta\mathbf{t}) < J(\theta, \mathbf{t})$. Given some tuning parameter λ, a trial update is obtained by

$$\begin{pmatrix} \Delta\theta \\ \Delta t_1 \\ \Delta t_2 \end{pmatrix} \leftarrow - \begin{pmatrix} (1+\lambda)(\frac{\partial J}{\partial\theta})^2 & \frac{\partial J}{\partial\theta}\frac{\partial J}{\partial t_1} & \frac{\partial J}{\partial\theta}\frac{\partial J}{\partial t_2} \\ \frac{\partial J}{\partial t_1}\frac{\partial J}{\partial\theta} & (1+\lambda)(\frac{\partial J}{\partial t_1})^2 & \frac{\partial J}{\partial t_1}\frac{\partial J}{\partial t_2} \\ \frac{\partial J}{\partial t_2}\frac{\partial J}{\partial\theta} & \frac{\partial J}{\partial t_2}\frac{\partial J}{\partial t_1} & (1+\lambda)(\frac{\partial J}{\partial t_2})^2 \end{pmatrix}^{-1} \times \begin{pmatrix} \frac{\partial J}{\partial\theta} \\ \frac{\partial J}{\partial t_1} \\ \frac{\partial J}{\partial t_2} \end{pmatrix}.$$

If the trial update fails to improve J, the tuning parameter λ is increased by some multiplicative factor κ and a new trial is produced. If the trial succeeds instead, then λ is decreased. Typically, we start the optimization with $\lambda = 1$ and set $\kappa = 4$. We use a disjunction of conditions to continue the optimization: the relative decrease $\Delta J/J$ must be sufficiently large, the relative change of parameters too, while λ must not grow unchecked, and the number of iterations performed so far does not exceed some arbitrary threshold that may be resolution-dependent (see the subsection "Multiresolution" under Appendix).

Multiresolution

We take advantage of multiresolution and perform the optimization according to a coarse-to-fine strategy. We first reduce the data in dyadic fashion with h as given in Table A2, which yields at each scale of increasing coarseness m a set of cubic-spline coefficients $c^{(m)}$, with $c^{(0)} = c$, that satisfy (Unser et al., 1993)

$$\forall k_1 \in \left[0\ldots\frac{K_1}{2^m}-1\right], \forall k_2 \in \left[0\ldots\frac{K_2}{2^m}-1\right] : p^{(m+1)}[k_1,k_2] = \sum_{n_2=-5}^{5} h[n_2] \sum_{n_1=-5}^{5} h[n_1] c^{(m)}[n_1-2k_1, n_2-2k_2],$$

with $c^{(m+1)}$ obtained in four (1 + 3) steps. Given $c^{(m)}$, we first compute $p^{(m+1)}$. We then apply Eqs. (A1) and (A2) with $p^{(m+1)}$ in the role of f, $q^{(m+1)}$ in the role of c, and z_1 in the role of z_0; we then re-apply Eqs. (A1) and (A2) with $q^{(m+1)}$, $r^{(m+1)}$, and z_2 in the role of f, c, and z_0, respectively; we finally re-apply a third and last time Eqs. (A1) and (A2) with $r^{(m+1)} \leftrightarrow f$, $c^{(m+1)} \leftrightarrow c$, and $z_3 \leftrightarrow z_0$. The values of z_1, z_2, and z_3 can be found in Table A3. Since $|z_1| > |z_0|$, more terms must be considered to maintain accuracy in case the sums that define $c[0,n_2]$ and $c[n_1,0]$ are truncated. For example, $|(z_1)^{15}| < 10^{-4} < |(z_1)^{14}|$. Conversely, less terms are required for z_2 and z_3. The poles z_1, z_2, and z_3 satisfy

$$\sum_{k=-3}^{3} \beta^7(k) z^{-k} = \frac{1}{5040 z^3}(z-z_1)\left(z-\frac{1}{z_1}\right)(z-z_2) \times \left(z-\frac{1}{z_2}\right)(z-z_3)\left(z-\frac{1}{z_3}\right).$$

Once the data have been adequately reduced, we proceed at the coarsest scale with the registration as previously described. After convergence at the mth level of resolution, we obtain a best translation $\mathbf{t}^{(m)}$ and a best rotation angle $\theta^{(m)}$ with respect to an arbitrary center $\mathbf{x}_0^{(m)}$, which determine the initial solution at the finer resolution $(m-1)$. More precisely, with **I** the identity matrix,

$$\left[\begin{array}{l} \theta^{(m-1)} \leftarrow \theta^{(m)} \\ \mathbf{t}^{(m-1)} \leftarrow 2\mathbf{t}^{(m)} + (\mathbf{R}_{\theta^{(m)}} - \mathbf{I})(\mathbf{x}_0^{(m-1)} - 2\mathbf{x}_0^{(m)}). \end{array}\right.$$

Kruskal's Algorithm

Let a weighted graph $G = \{V(G), E(G)\}$ consist of a set of vertices $V(G)$ and of a set of weighted edges $E(G)$, where each edge is assigned a weight $w_G(uv) \in \mathbb{R}$, with $u \in V(G)$, $v \in V(G)$, and $uv \in E(G)$. The number of vertices of G is $|G|$. The number of edges of G is $\varepsilon(G)$. A uv-path on G is a sequence of K distinct vertices $u_k \in V(G)$ such that $u_0 = u$, $u_{K-1} = v$, $u_k u_{k+1} \in E(G)$. The graph is said to be connected if at least one uv-path on G exists for all distinct u and v in $V(G)$. A cycle is a

TABLE A2. Coefficients of the antialiasing filter that is optimal (in the L_2 sense) for cubic splines

k	-5	-4	-3	-2	-1	0	1	2	3	4	5
$h[k]$	$\frac{1}{80640}$	$\frac{31}{20160}$	$\frac{559}{26880}$	$\frac{247}{2520}$	$\frac{9241}{40320}$	$\frac{337}{1120}$	$\frac{9241}{40320}$	$\frac{247}{2520}$	$\frac{559}{26880}$	$\frac{31}{20160}$	$\frac{1}{80640}$

TABLE A3. Poles of the septimic B-spline

$$
\begin{aligned}
\theta &= \tfrac{1}{3}\arccos\left(-\tfrac{738}{\sqrt{556549}}\right)\\
c &= \sqrt{301}\cos(\theta)\\
S &= \sqrt{903}\sin(\theta)\\
\lambda_1 &= 20 - 2c\\
\lambda_2 &= 20 + c - s\\
\lambda_3 &= 20 + c + s\\
z_1 &= \sqrt{\lambda_1^2 - 1} - \lambda_1\\
z_2 &= \sqrt{\lambda_2^2 - 1} - \lambda_2\\
z_3 &= -\left(\sqrt{\lambda_3^2 - 1} + \lambda_3\right)^{-1}
\end{aligned}
$$

uv-path such that $vu \in E(G)$. A tree is a connected graph that has no cycles. A forest is a disconnected graph that has no cycles.

Given a connected and weighted graph G with $|G| > 1$, the purpose of Kruskal's algorithm is to build a tree F that satisfies $V(F) = V(G)$ and that maximizes $\sum_{uv\in E(F)} w_G(uv)$. The procedure is to start with a forest F of $|G|$ trees that initially contain a distinct vertex only, and to gradually merge and link these trees in a greedy fashion until the forest F becomes a tree with $\varepsilon(F) + 1 = |F|$. This procedure can be proved to yield an optimal solution. Letting T and T_u be auxiliary trees, and letting H be a copy of G, the algorithm can be formally described by

```
$H \leftarrow G$
$F \leftarrow \{V(H), \emptyset\}$
$\forall u \in V(H) : T_u \leftarrow \{u, \emptyset\}$
while $\varepsilon(F) + 1 < |F|$ :
    $xy \leftarrow \arg\max_{uv\in E(H)} w_G(uv)$
    $H \leftarrow \{V(H), E(H)\backslash xy\}$
    $T_x \neq T_y \Rightarrow$
        $T \leftarrow T_x \cup T_y \cup \{\emptyset, xy\}$
        $T_x \leftarrow T$
        $T_y \leftarrow T$
        $F \leftarrow F \cup T.$
```

REFERENCES

Aguiar PMQ, Moura JMF. 2001. Image motion estimation—Convergence and error analysis. In: Proceedings of the 2001 IEEE International Conference on Image Processing (ICIP'01), Thessaloniki, Greece, October 7–10, Vol. II, pp. 937–940.

Can A, Stewart CV, Roysam B, Tanenbaum HL. 2002. A feature-based technique for joint, linear estimation of high-order image-to-mosaic transformations: Mosaicing the curved human retina. IEEE Trans Pattern Anal Mach Intell 24:412–419.

Grana C, Pellacani G, Seidenari S, Cucchiara R. 2006. Distance transform for automatic dermatologic images composition. In: Proceedings of the SPIE international symposium on medical imaging: Image processing (MI'06), San Diego, CA, February 11–16, Vol. 6144, pp. 61442U-1–61442U-9.

Hasler D, Süsstrunk S. 2004. Mapping colour in image stitching applications. J Visual Commun Image Represent 15:65–90.

Kang E-Y, Cohen I, Medioni G. 2000. A graph-based global registration for 2D mosaics. In: Proceedings of the fifteenth international conference on pattern recognition (ICPR'00), Barcelona, Spain, September 3–8, Vol. 1, pp. 257–260.

Kruskal JB, Jr. 1956. On the shortest spanning subtree of a graph and the traveling salesman problem. Proc Am Math Soc 7:48–50.

Marzotto R, Fusiello A, Murino V. 2004. High resolution video mosaicing with global alignment. In: Proceedings of the 2004 IEEE international conference on computer vision and pattern recognition (CVPR'04), Washington, DC, June 27–July 2, Vol. 1, pp. 692–698.

Meijering E. 2002. A chronology of interpolation: From ancient astronomy to modern signal and image processing. Proc IEEE 90:319–342.

Nie S-D, Si J-Y. 2005. Methodological study of automatically mosaicing for medical microscopic images. Chinese J Biomed Imaging 24: 173–178 (in Chinese).

Pires BE, Aguiar PMQ. 2004. Registration of images with small overlap. In: Proceedings of the 2004 IEEE sixth workshop on multimedia signal processing (MMSP'04), Siena, Italy, September 29–October 1, pp. 255–258.

Pires BE, Aguiar PMQ. 2005. Featureless global alignment of multiple images. In: Proceedings of the 2005 IEEE international conference on image processing (ICIP'05), Genova, Italy, September 11–14, Vol. 1, pp. 57–60.

Rasband WS. 1997–2006. ImageJ. U.S. National Institutes of Health, Bethesda, MD. Available at http://rsb.info.nih.gov/ij/

Sawhney HS, Hsu S, Kumar R. 1998. Robust video mosaicing through topology inference and local to global alignment. In: Proceedings of the fifth European conference on computer vision (ECCV'98), Freiburg im Breisgau, Germany, June 2–6, Vol. II, pp. 103–118.

Shi J, Tomasi C. 1994. Good features to track. In: Proceedings of the 1994 IEEE computer society conference on computer vision and pattern recognition (CVPR'94), Seattle, WA, June 21–23, pp. 593–600.

Thévenaz P. 2006. MosaicJ. École polytechnique fédérale de Lausanne (EPFL), Switzerland. Available at http://bigwww.epfl.ch/thevenaz/mosaicj/

Thévenaz P, Ruttimann UE, Unser M. 1998. A pyramid approach to sub-pixel registration based on intensity. IEEE Trans Image Process 7:27–41.

Thévenaz P, Blu T, Unser M. 2000. Interpolation revisited. IEEE Trans Med Imaging 19:739–758.

Thévenaz P, Lambiel D, Unser M. 2006. A strategy based on maximum spanning trees to stitch together microscope images. In: Proceedings of the SPIE International Symposium on Medical Imaging: Image Processing (MI'06), San Diego, CA, February 11–16, Vol. 6144, pp. 61442A-1–61442A-6.

Triggs B, McLauchlan P, Hartley R, Fitzgibbon A. 1999. Bundle adjustment—A modern synthesis. In: Proceedings of the International workshop on vision algorithms: Theory and practice (IWVA'99), Corfu, Greece, September 21–22. Lecture notes in computer science, Vol. 1883. Berlin-Heidelberg. pp. 298–372.

Unser M. 1999. Splines: A perfect fit for signal and image processing. IEEE Signal Process Mag 16:22–38.

Unser M, Aldroubi A, Eden M. 1993. The L_2-polynomial spline pyramid. IEEE Trans Pattern Anal Mach Intell 15:364–379.

Winkelman F, Patras I. 2004. Online globally consistent mosaicing using an efficient representation. In: Proceedings of the 2004 IEEE international conference on systems, man and cybernetics (ICSMC'04), The Hague, Netherlands, October 10–13, pp. 3116–3121.

Zagorodnov V, Ramadge P. 2000. Error stabilization in successive estimation of registration parameters. In: Proceedings of the Seventh IEEE International Conference on Image Processing (ICIP'00), Vancouver, BC, September 10–13, Vol. 1, pp. 228–231.