Journal of Microscopy, Vol. 234, Pt 1 2009, pp. 62–79
Received 13 May 2008; accepted 15 November 2008

Automated tracking of migrating cells in phase-contrast video microscopy sequences using image registration

A.J. HAND*, T. SUN*, D.C. BARBER†, D.R. HOSE†
& S. MACNEIL*
*Kroto Research Institute, North Campus, University of Sheffield, Broad Lane, Sheffield, S3 7HQ, United Kingdom
†Medical Physics, Royal Hallamshire Hospital, University of Sheffield, Glossop Road, Sheffield, S10 2JF, United Kingdom

Key words. Cell motility, cell tracking, image registration.

Summary

Analysis of *in vitro* cell motility is a useful tool for assessing cellular response to a range of factors. However, the majority of cell-tracking systems available are designed primarily for use with fluorescently labelled images. In this paper, five commonly used tracking systems are examined for their performance compared with the use of a novel in-house cell-tracking system based on the principles of image registration and optical flow. Image registration is a tool commonly used in medical imaging to correct for the effects of patient motion during imaging procedures and works well on low-contrast images, such as those found in bright-field and phase-contrast microscopy. The five cell-tracking systems examined were Retrac, a manual tracking system used as the gold standard; CellTrack, a recently released freely downloadable software system that uses a combination of tracking methods; ImageJ, which is a freely available piece of software with a plug-in for automated tracking (MTrack2) and Imaris and Volocity, both commercially available automated tracking systems. All systems were used to track migration of human epithelial cells over ten frames of a phase-contrast time-lapse microscopy sequence. This showed that the in-house image-registration system was the most effective of those tested when tracking non-dividing epithelial cells in low-contrast images, with a successful tracking rate of 95%. The performance of the tracking systems was also evaluated by tracking fluorescently labelled epithelial cells imaged with both phase-contrast and confocal microscopy techniques. The results showed that using fluorescence microscopy instead of phase contrast does improve the tracking efficiency for each of the tested systems. For the in-house software, this improvement was relatively small ($< 5\%$ difference in tracking success rate), whereas much greater improvements in performance were seen when using fluorescence microscopy with Volocity and ImageJ.

Correspondence to: Prof. S. MacNeil. Tel: +44 114 222 5993; fax: +44 114 222 5945; e-mail: s.macneil@sheffield.ac.uk

Introduction

The analysis of *in vitro* cell motility and morphology is an area that is gaining increasing importance with recent advances in computing and imaging technology. Accurate knowledge about changes in cell behaviour will allow more in-depth comparisons between the effects of different drugs for the treatment of diseases, as well as enabling more thorough testing of suitable scaffolds for cell culture in the area of tissue engineering. It is also a vital tool for the validation of computational models of cell behaviour, such as the Epitheliome project (Walker *et al.*, 2004), which aims to model the social interactions of epithelial cells during wound healing and is currently in progress at the Universities of Sheffield and York.

The traditional way to track cells is via manual marking of time-lapse microscopy images. Although this is the 'gold standard' method of cell tracking, it is very time-consuming and can be tiring for the operator when tracking large numbers of cells over long time periods. This in itself can lead to errors in the tracking process. For this reason, a significant amount of research is currently underway to develop automated cell-tracking systems that are capable of tracking cells in time-lapse sequences with a minimum of user interaction (see, e.g. Zimmer *et al.*, 2002; Rabut & Ellenberg, 2004; Debeir *et al.*, 2005; Gor *et al.*, 2005).

However, many of these systems are being developed primarily for use with fluorescently labelled cells, which produce high-contrast images. Although phase-contrast microscopy produces images with bright or dark rings around the cells and may be considered to be a high-contrast imaging

technique, the intensity of this ring varies with the position of the cell relative to the focal point. This means that the ring often appears broken and will vary in intensity over the course of a time-lapse sequence. The success rate of automated systems for use with low-contrast bright-field or phase-contrast microscopy images therefore varies considerably depending on the specific aims of the software in question.

In this paper, a range of image segmentation and object-tracking techniques are briefly described and an overview of a selection of existing cell-tracking systems is given. The development and validation of an in-house cell-tracking system, based on image-registration methods and suitable for simultaneously tracking a large number of cells in both high- and low-contrast images, is then discussed.

Definition of terms

The process of identifying and following the motion of cells is often referred to in the literature as 'particle detection' and 'particle linking', respectively. In this paper, we refer to the identification of cells as 'segmentation' of the image into cells and background, as this is a more generalized term referring to the separation of an image into its component objects. In addition, the technique we describe for the in-house software does not require the identification of cells to be performed explicitly for each frame of an image sequence. As such the term *particle linking* would be misleading in this context. For this reason, we use the term *tracking* to describe the process of following the motion of the cells. Later in the paper, we also use the term *mapping* to refer to the matrix of vectors describing the motion of the individual pixels in an image sequence.

Common segmentation methods

In order to track the cells in a microscope image, the cells must first be located. To do this, the cells must be separated from the background and each other in a process known as segmentation. There is a wide range of image segmentation techniques, and in-depth reviews can be found in Pal & Pal (1993) and Althoff (2005). Some of the more common techniques currently employed in cell tracking include thresholding, edge detection and watershed segmentation.

The simplest segmentation method is thresholding, in which only pixels in the image above a pre-defined intensity threshold are classed as objects, whereas those below the threshold are classed as background. This method is discussed in Pal & Pal (1993) and Cheriet *et al.* (1998). Unfortunately, this method requires a high signal-to-noise ratio in order to work and as such will generally only work well on images acquired using fluorescence microscopy with labelled cells (e.g. Rabut & Ellenberg, 2004).

Using a single threshold for a whole image can also cause problems in non-uniformly illuminated samples where the intensity may vary across the image. In these cases, it is more appropriate to use local adaptive thresholding, in which small regions of the image are examined and a different threshold is chosen for each pixel based on the intensity information contained in the local region. There are a variety of methods for finding the optimum local thresholds, a comprehensive survey of which can be found in Sezgin & Sankur (2004). For the application of adaptive thresholding techniques to cell biology, see for example Neumann *et al.* (2006).

Edge detection is another common method used to segment objects from images. This method makes use of the intensity gradients in images to locate the 'edges' of objects such as cells. The method is based on the assumption that what a human perceives as an edge or boundary in an image is related to a large change in image intensity at that point (Gonzalez & Woods, 2002). Although this is not necessarily the actual physical edge of the cell, it is the most appropriate definition of the cell edge in terms of image segmentation, as these large changes in intensity will be represented by large gradient values, whereas the gradients across pixels in areas of constant intensity (i.e. not at boundaries) will be small. Edge-detection methods are more robust than simple thresholding; however, they still often fail in low-contrast images, where fluctuations in intensity due to image noise produce similar gradients to those found at the actual cell boundaries (Pal & Pal, 1993; Gonzalez & Woods, 2002).

One problem that both thresholding and edge detection have in common is that they are unable to distinguish between two objects of the same intensity that share a boundary, for example, two cells that are touching or overlapping. In this case, the two cells would most likely be identified by the computer to be a single object. A method has been devised to overcome this problem, which is now known as 'the watershed method' (Beucher & Lantuejoul, 1979). This method again makes use of the intensity gradients in the image.

The method assumes that there will be minima in the intensity gradients at the centres of the cells and large gradients at the edges. If this is visualized as a topological surface, with peaks or mountains representing the high gradients at the edges of the objects, the low gradients in the centres of the cells will be represented by troughs or valleys of varying heights. If 'holes' are created at each of the minima, and the surface is immersed into water, the water will eventually start to flood through these holes and form lakes in between the mountains. At any point where two separate lakes would meet, a dam or watershed is built between them to keep them apart. When the water finally reaches the top of the highest peak, only these dams should be visible. These should represent the edges of the different objects in the image.

Common tracking methods

Once an image has been segmented and the location of the cells identified, the tracking process can begin. One of the simplest methods of tracking is nearest-neighbour tracking, in which each frame of the time-lapse sequence is segmented

individually and then each cell in one frame is assumed to be matched to the nearest cell to that position in the next frame. Obviously, this has a tendency to misidentify cells when they are densely packed or when cells pass and overlap.

Feature matching is a method used by many tracking systems, as it is more robust than nearest-neighbour tracking yet still fairly simple to implement. This technique uses a computer algorithm to search for objects with similar features, such as surface area or cell diameter, within a given distance of the cell position on the previous frame. Information about the maximum distance the cell can travel between subsequent frames must be provided by the user, either as a specified range or by specifying the type of motion, for example, auto-regressive or random walk.

Feature matching relies on an effective prior segmentation of the image, as mistakes in segmentation can cause the cells to be incorrectly matched. As such, this technique is again most effective with high-contrast images such as those obtained via fluorescence microscopy. Furthermore, any large changes in morphology between subsequent frames may cause the cells in the second frame to be identified as new cells and result in the tracks becoming broken (De Hauwer *et al.*, 1999). For a more comprehensive discussion of common tracking methods and their suitability to tracking different features, see Miura (2005).

Existing systems

The cell-tracking systems that are available at the moment can be separated into two broad categories: commercial systems and open source systems; the latter are freely available for download from the Internet. The available commercial systems include Imaris (produced by Bitplane, Zurich, Switzerland) and Volocity (produced by Improvision, Coventry, UK). As these systems are produced for sale, they tend to include more features and technical support than the open-source software, including advanced image-processing techniques and three-dimensional (3D) visualization techniques. However, there is a wide range of open-source software available, ranging from more basic software such as Cell_Motility (Martens *et al.*, 2006), which is designed to analyze numerical data about cells that has already been extracted from the microscope images, to more complex software such as Cell Tracker (Shen *et al.*, 2006), which includes many of the features included in commercial software. A summary of some of the available systems and the features they offer is provided in Tables 1(a) and 1(b). A score of 1–5 has been given for ease of use, where 1 implies that the software is very difficult to use without prior training, and 5 implies that the software is easy to use and intuitive.

Development of in-house cell-tracking system

For the purposes of validation of the Epitheliome computational models, a cell-tracking system was required that would be able to simultaneously track a large number of cells (>100) in time-lapse phase-contrast microscopy images. There is a significant amount of research underway to develop systems capable of achieving this (see, e.g. Debeir *et al.*, 2005; Li *et al.*, 2006; Li & Kanade, 2007), but we have not yet found any available software that entirely fulfils our tracking requirements. CellTrack (Sacan *et al.*, 2008) uses a combination of tracking methods including active contours and optical flow estimation to track cell boundaries and intracellular points for single cells with high success; however, the published data currently only give qualitative results for a small number of well-defined cells, and a more in-depth evaluation of the software by the authors is yet to be performed. In order to meet our needs, we decided to develop an in-house software system that was designed specifically for this purpose.

Image registration – general principles

Image registration is a concept that has been applied to a wide range of problems, including motion correction in medical imaging, motion detection in computer vision and decompression of compressed video images. A wide range of image-registration techniques have been developed and several good review papers exist, including Maintz & Viergever (1998) and Lester & Arridge (1999). However, the basic concept is the same in all cases, that is, to find the optimum mapping function that aligns the objects present in one 'source' image to those in a target image. Although aligning cell images to each other is not in itself advantageous for cell biologists, by determining the mapping necessary to align a cell with its position in a previous image, it is possible to determine how far, in terms of pixels, the cell has moved from one image to the next and hence to quantify its motion.

The most basic method of achieving image alignment is feature- or landmark-based registration, where a selection of points are identified as landmarks in both the source and reference image. The source image is then moved, rotated or stretched until these points are aligned with those in the reference image. However, a feature-based method would have very limited success with microscopy images of large populations of cells, in which there are few identifiable 'landmarks' and the cell features that could potentially be registered all appear on a very small scale.

Image registration based on optical flow

In a paper by Barber (1999), an image-registration method is introduced that is based on vector field techniques relating to the optical flow field of the images being registered. This method does not require landmarks to be identified in the images, but instead registers each individual pixel (rather than each individual object) based on the flow of the pixel intensities between the two images. By constraining the

Table 1(a). Overview of existing cell-tracking systems (part 1).

Software name	Company/Author	Website	Proprietary?	Segmentation methods	Segmented features	Tracking methods	Multiple cell?
Volocity 4	Improvision	http://www.improvision.com	Yes	Thresholding, Edge Detection, Watersheds	Centroid, Outline	Feature Matching	Yes
Braincells	Gabor Ivancsy	http://pearl.elte.hu/~kyd/	No	Manual	Centroid	Manual	Yes
ImageJ (with Mtrack2 plugin)	Wayne Rasband	http://rsb.info.nih.gov/ij/	No	Thresholding, Edge Detection, Watersheds	Centroid, Outline	Feature Matching	Yes
Retrac	Molecular Motors Group	http://mc11.mcri.ac.uk/retrac.html	No	Manual	Centroid	Manual	Yes
CellTrack	Database Research Group, The Ohio State University	http://db.cse.ohio-state.edu/CellTrack/	No	Thresholding, Edge Detection	Centroid, Outline	Feature Matching, Mean-Shifts, Optical Flow, Active Contours	Yes
Imaris 5.5	Bitplane	http://www.bitplane.com/	Yes	Thresholding, Spot Detection	Centroid, Outline	Feature Matching	Yes
Cell Tracker	Bioanalytical Sciences Group, University of Manchester	http://dbkgroup.org/celltracker/	No	Thresholding, Level Sets, Active Contours	Centroid, Nuclear Outline, Cytoplasm Outline	Edge Detection, Active Contours	Yes

 Journal of Microscopy, **234**, 62–79

Table 1(b). Overview of existing cell-tracking systems (part 2) – NB. In 'Ease of use' column, software is rated from 1–5 where 1 means that the software is difficult to use without prior training and 5 means it is easy to use without any prior training.

Software name	Operating system	Additional Software requirements	Image Formats accepted	Extracted data	Data output format	Cell preparation required	Other details	Ease of use
Volocity 4	Cross-platform	None	Includes BMP, JPEG, TIFF, LIFF, Raw, PICT, GIF, LSM etc.	Number of cells, cell centroid positions, cell surface areas (2D), cell volumes (3D), cell displacements, cell velocities.	Tab Delimited Text	Fluorescently-labelled cells preferred	Range of packages available, Volocity Quantitation deals with cell tracking	3
Braincells	Windows	None	JPEG, BMP only	Number of cells, cell centroid positions, cell displacements, cell velocities.	MSExcel, Tab Delimited Text	None	Software is partly in Hungarian, translations in Help file	3
ImageJ (with Mtrack2 plugin)	Cross-platform	Java	Includes bitmap, JPEG, TIFF, QuickTime Movie, AVI, PICT, GIF, LSM etc.	Number of cells, cell centroid positions, cell surface areas, cell displacements, cell velocities.	Tab Delimited Text	Fluorescently-labelled cells preferred	Cells can also be tracked manually if required	2
Retrac	Windows	None	JPEG, BMP, TIFF, GIF	Number of cells, cell centroid positions, cell displacements, cell velocities.	Tab Delimited Text	None		5
CellTrack	Cross-platform	None	Includes bitmap, JPEG, TIFF, AVI, PNG, PBM, EXR etc.	Number of cells, boundary positions, cell speeds, path length, cell area, cell deformation.	Tab Delimited Text	None	Includes option to track intracellular points as well as boundary position	5
Imaris 5.5	Cross-platform	None	Includes PCDO, SOFTIMAGE, NITF, BMP, JPEG, Raw, TIFF, QuickTime etc.	Number of cells, cell centroid positions, cell surface areas (2D), cell volumes (3D), cell displacements, cell velocities, path curvature, number of fusions of tracks, number of divisions of cell tracks.	MSExcel, Comma-Separated Values	Fluorescently-labelled cells preferred	Range of packages available, ImarisTrack deals with cell tracking, MeasurementPro and FilamentTracer include other quantification features	2
Cell Tracker	Windows	Matlab Component Runtime (MCR)	LSM, TIFF, JPEG, BMP, Matlab .mat file	Number of cells, cell centroid positions, cell surface areas (cytoplasm and nucleus individually), cell displacements, cell velocities.	MSExcel, XML	None (see Other Details)	Nucleus and cytoplasm of cells to be tracked must be clearly visible as both are tracked concurrently	2

allowed movements of the pixels within the image to non-linear elastic motion, it is possible to register moving objects even if those objects are subjected to changes in morphology during the motion. This makes this registration technique particularly suited to images of biological cells.

Barber & Hose (2005) describe how such an image-registration algorithm can be used to automatically segment medical images such as an MRI image of the knee. This is achieved by registering the acquired patient image with a pre-segmented reference image. Once a mapping has been found that will align the patient image and the reference image, the same mapping can be applied to the segmentation image. This means that the segmentation itself is warped so that it fits the patient image instead of the reference image. It is this process in particular that will enable the motion of cells to be tracked.

Materials and methods

Cell culture

HaCat cells (Keratinocyte cell line) were cultured in defined keratinocyte serum-free medium (SFM) (Invitrogen, Paisley, U.K.; calcium concentration 0.09 mM) supplemented with 100 IU mL^{-1} penicillin and 100 mg mL^{-1} streptomycin in this research.

Cell labelling using Cell Tracker™

HaCat cells were labelled with Cell Tracker™ Red CMTPX (C34552) (Invitrogen) for time-lapse fluorescent microscopic study. The cells were incubated in fluorescent probe Cell Tracker™ reagent (0.5–25 μM, in SFM) for 15–45 min at 37°C and subsequently in normal SFM for 30 min at 37°C. The cell culture was then continued, and the fluorescently labelled cells were ready for time-lapse fluorescence video microscopy.

Time-lapse video microscopy

For the purposes of this experiment, two different time-lapse microscopy sequences were acquired using an ORCA ER CCD digital camera (Hamamatsu Photonics, Hamamatsu, Japan) mounted on a Leica DM-IRB inverted microscope (Leica, Wetzlar, Germany) with a 20× objective (Leica, NA 0.4). The cells were situated in a specifically designed cell culture chamber (37°C and 5% CO_2). For the first sequence, HaCat cells were cultured in SFM, and phase-contrast images were taken every 5 min over a certain period of time (hours).

For the second sequence, HaCat cells were first labelled with Cell Tracker™ Red CMTPX (C34552). Then both phase-contrast and fluorescence (λ_{ex} = 580 nm, λ_{em} = 650 nm for TRITC / Cell Tracker™ visualization) images of the cells were acquired at 10-min intervals over a period of 3.5 h using the same microscope, camera and objective as for the first sequence at each time point. Thus, a phase-contrast and a fluorescence time-lapse video were obtained from the same cell population for the evaluation of various cell-tracking methods.

Background removal and segmentation

In order to locate the cells within the phase-contrast microscopy images, it was necessary to segment the first frame of the time-lapse sequence. The method used to achieve this was based on a segmentation technique described in Zimmer *et al.* (2002). This technique uses the local average deviation in intensity as a measure to effectively distinguish pixels within the cells from background pixels in low-contrast images while also enhancing the boundaries between adjoining cells. With the background removed and the boundaries enhanced, it is then a simple procedure to perform a watershed segmentation on the resulting image. Figure 1 shows the results of using this technique on a phase-contrast microscopy image of HaCats cultured in SFM.

The registration equation

In order to find the pixel mappings, a registration equation is used. If **f** is a matrix containing the intensity values for the 'fixed' (or target) image and **m** is a matrix containing the intensity values of the pixels in the 'moved' (or source) image, the difference between the two images is related to the mapping functions ($\Delta x(x, y, z)$ and $\Delta y(x, y, z)$) and the intensity gradients across **f** and **m** as follows:

$$\mathbf{f} - \mathbf{m} = \frac{\Delta x}{2}\left(\frac{\partial \mathbf{f}}{\partial x} + \frac{\partial \mathbf{m}}{\partial x}\right) + \frac{\Delta y}{2}\left(\frac{\partial \mathbf{f}}{\partial y} + \frac{\partial \mathbf{m}}{\partial y}\right) - \frac{\Delta \alpha}{2}(\mathbf{f} + \mathbf{m}) - \Delta\beta. \quad (1)$$

$\Delta\alpha$ and $\Delta\beta$ are intensity mappings that are included to compensate for differences in overall intensity between the two images. This equation may easily be expanded for use with 3D images by adding a Δz term. It should be noted that traditional optical flow algorithms, such as the Lucas and Kanade method used in CellTrack, do not include the $\Delta\alpha$ and $\Delta\beta$ terms to compensate for intensity differences. Furthermore, traditional algorithms are based only on the intensity gradients in one of the images being considered, whereas this equation uses the average of the gradients in both the fixed and target images. We believe this makes the algorithm more robust than traditional methods.

The registration equation can be applied to each pixel (or voxel) within the image set, producing an under-determined set of equations with four unknown values (five in the case of 3D images). In order to find the solutions (i.e. the mapping functions), the equation is expanded in terms of local tri-linear basis functions centred on regularly placed nodes within the image. The resolution of the mapping is determined by the

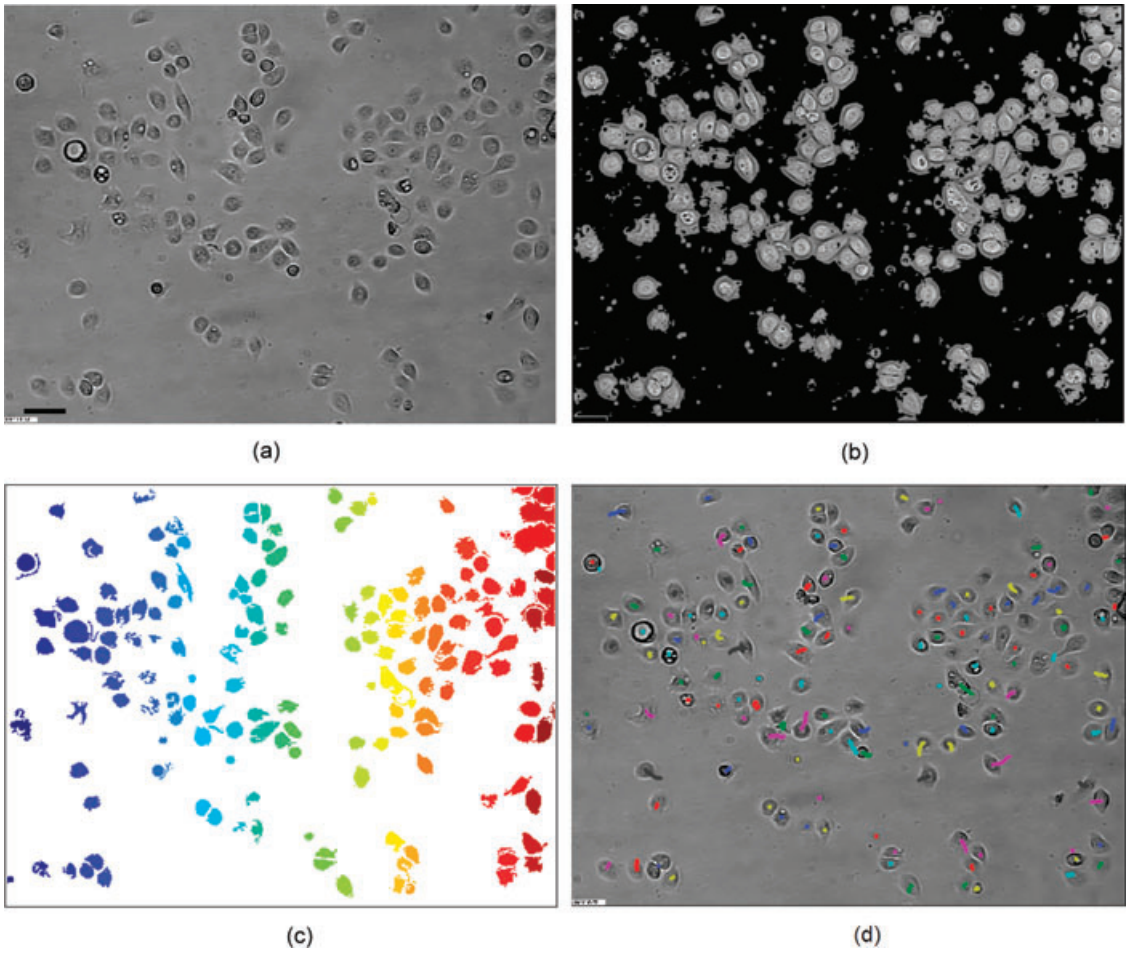

Fig. 1. (a) Original phase-contrast microscopy image of human epithelial cells. Scale bar length = 100 pixels or 32.5 μm; (b) Same image with background removed; (c) Final segmented image with each cell represented by a different colour for ease of identification (153 different shades are present in the image; these are not easily distinguishable by eye but can be identified by computer); (d) First frame of the phase-contrast sequence with tracks found by registration software overlaid.

spacing between the nodes, with the vector solutions for pixels between the nodal points determined by interpolation of the vectors found for the nearest nodal points. The algorithm is described in more depth in the appendix. For further information on the registration algorithm, see Barber *et al.* (2007).

Tracking method

In order to track cells in a time-lapse movie, the first frame of the movie must be segmented using any available method. This initial segmentation should be as accurate as possible to enable robust tracking of the cells. Following segmentation, the first frame of the time-lapse sequence is then registered to the subsequent frame. The obtained pixel-by-pixel mapping will show how the pixels in the first frame of the sequence should be moved to transform the image in the first frame of the video into the image in the second frame. The initial segmentation tells us which pixels in the first frame are part of each cell that has been identified. Hence, by using the mapping to identify where these particular pixels have moved, we are able to automatically follow the motion of the cells. This process can then be repeated for the entire time-lapse sequence, with each frame being registered to the subsequent frame and the pre-segmented cells moved according to the mapping obtained at each stage.

Validation

In order to validate the in-house software, ten frames of the first phase-contrast video microscopy sequence of HaCat cells were analyzed. This sequence showed the unlabelled HaCat cells, cultured in low physiological calcium at 37°C. Using the in-house software, the first frame of the sequence

(Fig. 1(a)) was segmented using the background removal method described earlier (Fig. 1(b)), followed by simple thresholding of the resulting image. This resulted in the segmentation image shown in Fig. 1(c). Each frame of the original image sequence was then registered to the following one and the obtained mapping was used to warp the segmentation image in the same way.

The solution to the registration algorithm is only calculated explicitly using the values from the nodal points. For pixels between the nodes, the mapping function is obtained by interpolation between the nodal values. This interpolation inevitably causes blurring of the image when it is warped and makes the cells appear to spread over a larger area than they actually do in the original image. To reduce the likelihood of errors appearing in the warped segmentation as a result of this, a sharpening filter was used to post-process each mapped segmentation image. The number of cells identified and the positions of the cell centroids were then extracted from this image for the first and last frame of the sequence.

To test the effectiveness of this method, our in-house software was compared with five existing cell-tracking systems. The first of these, Retrac (http://mc11.mcri.ac.uk/Retrac/index.html), is designed for manual tracking and requires a human operator to click on the centre of each cell for each frame in the sequence. As manual tracking is currently considered the 'gold-standard' in cell tracking, the results obtained via Retrac were taken to be the actual cell positions against which the other software systems would be compared. To reduce the uncertainty in these positions due to human error, manual tracking was performed three times, and the averages of these three attempts were used to define the actual cell positions. The other five software systems being considered were Imaris (www.bitplane.com) and Volocity (www.improvision.com), which are available commercially; CellTrack, which is open-source software recently released by the Database Research Group at the Ohio State University (http://db.cse.ohio-state.edu/CellTrack/) and the freely downloadable ImageJ software combined with the MTrack2 plug-in. Each of these systems includes its own method of locating the cells (i.e. a form of particle detection) for each frame and then links the cells found in each frame to those found in the previous one to track the motion. The original images were optimized for tracking using the tools available with each system. CellTrack offers a range of tracking options including combinations of several methods. For the purposes of this test, we decided to use the option of an initial basic tracking using template matching, combined with optical flow to improve the tracked boundary positions. Alternate tracking plug-ins are available for ImageJ, including a more robust plug-in (ParticleTracker) for use with fluorescent images. However after testing these, MTrack2 seemed to be the most suited to the images we were using.

Performance criteria

To enable comparison between the different systems, a range of performance criteria were defined. The first set of criteria was related to the initial segmentation of the images and were defined as follows:

1. Number of cell centres correctly located (i.e. within ten pixels of position marked manually via Retrac).
2. Number of cell centres incorrectly located (i.e. within correct cell but not within ten pixels of actual centre marked using Retrac).
3. Number of cells missed by software (i.e. no cell centre was identified that could be associated with that cell).
4. Number of 'false' cells found by software (i.e. cell centre was identified in area that contains only background noise or multiple cell centres had been marked in a single cell).

The value of ten pixels as the threshold for cell location was chosen as this was approximately equal to the standard deviation in the marked cell position when manual tracking was performed.

The remaining criterion measured the success of the tracking process by determining the percentage of cells that had been correctly located on the first frame of the sequence that were still being located correctly by the last frame. If the cell was located at a distance greater than ten pixels from the manually marked position on the last frame of the sequence, or not located at all, the track was considered to be lost. Only the cells that were correctly located on the first frame were included in these results so that an objective comparison, not dependant on the initial segmentation, could be obtained.

Comparison between phase-contrast and fluorescence microscopy

Once the in-house software had been validated for phase-contrast images, it was necessary to verify that consistent results would be obtained regardless of the imaging technique used. To achieve this, the phase-contrast and fluorescence microscopy sequences obtained using the fluorescently labelled sample were used. Several of the cells in this sample were in colonies and as such the boundaries between cells were difficult to identify automatically in both the image sequences. Hence, to ensure an effective comparison between the tracking abilities of the system for the two different imaging techniques, an automated segmentation was first performed on the fluorescence microscopy sequence. The cell boundaries were then edited with reference to the phase-contrast images to identify the individual cells in the colonies and any cells that left the field of view during the sequence were removed. This edited segmentation image was then used to initialize the tracking procedure for both image sequences. Comparison between tracking performance using phase-contrast and fluorescence images were also made using Imaris, Volocity, CellTrack and ImageJ.

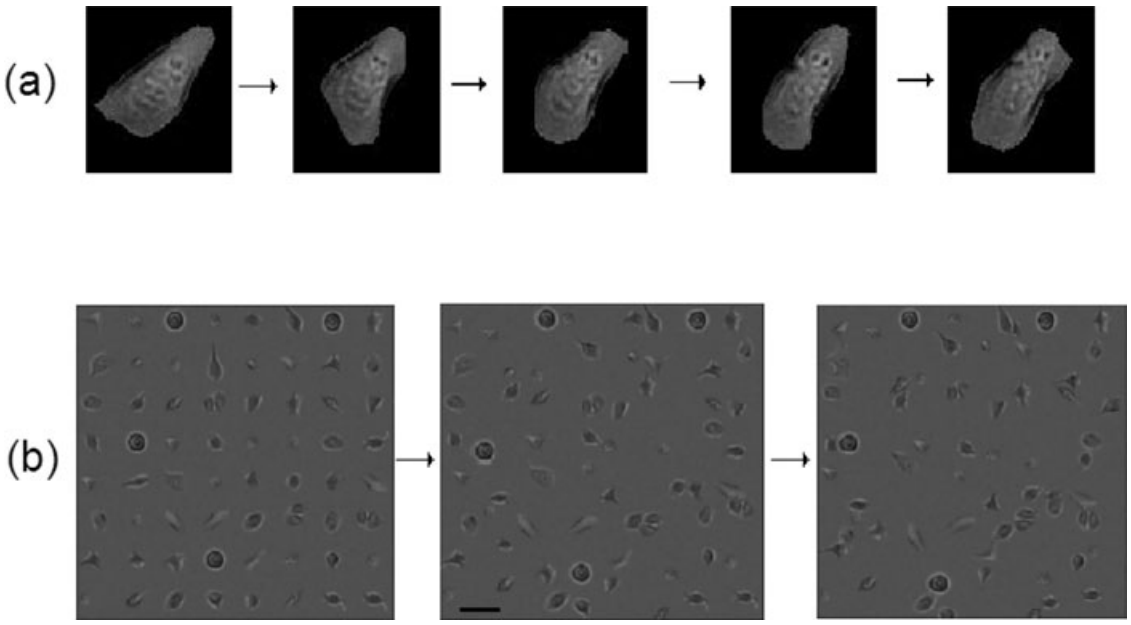

Fig. 2. (a) Example of a cell sequence extracted from real time-lapse images; (b) First three frames of an artificially generated cell sequence containing 64 cells allowed to move up to 60 pixels per frame. Scale bar length = 100 pixels.

Comparison between tracking systems using phase-contrast images at different cell densities and time intervals

In order to determine how the performances of the tracking systems vary when different cell densities and time intervals are used, a series of artificial cell images were created. This was achieved by first extracting a background region from the original phase-contrast image sequence and creating a new 800 × 800 pixel background image containing pixels with intensity varying from the minimum to the maximum intensity values from this region. A number of cells were then extracted from the original image by drawing around the boundary of each cell and extracting the region within the boundary from the image, which was then stored as a matrix of pixel intensity values. This was repeated for each of the selected cells for each frame of the sequence so that the changing shape of the cell for each frame was recorded. An example of an extracted cell sequence is shown in Fig. 2(a).

Once the cells had been extracted from the original image, it was necessary to re-insert them into the new background image at the required cell densities. This was achieved by initially dividing the image into equal-sized sections and selecting the required number of cells at random from those extracted from the original selection. The orientation of each cell was also chosen randomly, and the extracted cell image was rotated to match this. For each cell to be inserted, an area of the background image matching the cell shape was removed around the centre of one of the sections of the image and replaced with the image of the cell. For the next image in the sequence, the centre position of the cell was shifted by a random distance up to five pixels in any direction. The next frame of each cell sequence was then used and the cells were inserted at the new position as above. This was repeated until ten images showing the cells moving around and changing shape as in a real time-lapse sequence were produced.

The number of cells and densities used were as follows:

4 cells in image = 1 cell/160 000 pixels^2
16 cells in image = 1 cell/40 000 pixels^2
64 cells in image = 1 cell/10 000 pixels^2
144 cells in image = 1 cell/4444 pixels^2
256 cells in image = 1 cell/2500 pixels^2
400 cells in image = 1 cell/1600 pixels^2

Further image sequences were produced for each cell density with the cells being allowed to move by distances of 10, 15, 20, 30, 40, 60, 80, 100 and 120 pixels. Figure 2(b) shows three frames of the sequence containing 64 cells where the cells were allowed to move by up to 60 pixels. The average diameter of a cell in the sequence was approximately 40 pixels, so a cell moving 120 pixels between images will have moved by approximately three times its diameter. The average area of one of the extracted cells was approximately 1600 pixels^2, meaning that with 400 cells in the image the cells were all touching or overlapping.

We decided to use these artificially generated images rather than actual cell images seeded at different densities because it allows us to know the exact position of each cell centre at each time point, without requiring manual tracking to be performed. This was necessary as it becomes very difficult to track cells manually when the density is high and the cells

are overlapping and passing over each other, especially if the cells move large distances between images. The cells were then tracked using the software as for the real cell images and the cell centroids output by the software were compared with the known positions.

Change in tracking efficiency with number of frames in sequence

In the previous experiments described, only ten frames of the full image sequence were used as the manual tracking process also makes tracking over long time-lapse sequences difficult. Using artificially generated images as described earlier, it was possible to test how the tracking performance varies with the number of frames over which the cell is tracked. An image sequence containing 100 frames showing four cells moving up to 15 pixels per frame was constructed, and the cells were tracked using the in-house software. The deviation of the output centroid positions from the actual cell positions were then determined for each frame.

Results

Performance criteria – phase contrast only

The results obtained for the performance criteria are shown in Fig. 3. A total of 156 cells were found during manual marking using Retrac, and this value remained constant throughout the sequence. Of the five alternate systems tested, the in-house software and Imaris had the greatest success in identifying the cell locations on the first frame, with the in-house software locating 135 cells correctly and Imaris locating 133 cells. In terms of tracking the cells, the in-house software proved most effective, with a successful tracking rate of 95%. The most successful of the existing systems was CellTrack, tracking approximately 89% of the cells it located correctly. However, in our hands the initial segmentation using CellTrack was less good as we only located 42 of the 156 cells to within ten pixels of the correct position. (This software does allow manual editing and drawing of cell boundaries to initiate the tracking process, which were not explored in depth in this study.) Imaris was the next most successful of the existing systems, tracking just over 81% of the cells it located correctly.

Figure 1(d) shows the paths found by the in-house software overlaid on the first frame of the sequence. Figure 4 shows the mean deviation of the cell positions found using the in-house software from those found using Retrac for each of the ten frames of the sequence. This shows that a similar level of tracking accuracy was achieved for each of the intermediate frames, although tracking performance does degrade as the image sequence progresses. This is due to an accumulation of minor errors in the registered cell positions at each time point.

Speed

An important consideration when choosing a cell-tracking system is the speed at which it is able to track the cells in a time-lapse sequence. A balance must be achieved between the accuracy of the results needed to satisfy the requirements of the experiment and the time taken to obtain those results. For the ten frames containing 156 cells tracked for during validation, the software systems, approximate times taken to complete the segmentation and tracking process (including configuring the tracking options) are given below:

1. Retrac – approximately 2 h
2. Imaris – approximately 10 min
3. Volocity – approximately 10 min
4. ImageJ – approximately 5 min
5. CellTrack – approximately 20 min
6. In-house software – approximately 20 min

These times were obtained using heterogeneous systems without a benchmark and as such should be taken as a guide only.

Phase-contrast versus fluorescence microscopy results

Retrac was used to track the cells manually to provide a standard for comparison once again. A total of 58 cells were identified using Retrac. Using the segmentation technique described, the in-house software identified all 58 of the cells to within ten pixels of the correct position in the first frame. The methods used by these systems do not allow an initialization image to be used in this way and as such segmentation was performed using the available tools in the software. The number of cells identified correctly in the first frame by each of the cell-tracking systems are shown in Fig. 5.

Figure 5(c) shows the percentage of cells tracked correctly over the ten frames of the two sequences for each of the systems. Of the cells that were initially identified in the correct position, the in-house system was able to track 91% of the cells successfully (i.e. to within ten pixels of the actual cell position) across the whole sequence for the phase-contrast image set and tracked 95% of the cells successfully in the fluorescence microscopy image set. In each of the alternate systems, an improvement in overall performance was also seen when fluorescent images were used. The greatest improvement in performance was seen with Volocity, which was able to track only 38% of the cells correctly for the phase-contrast sequence and 100% for the fluorescent sequence. CellTrack achieved the same tracking rate (85%) with both imaging modalities; however, it was able to segment a greater number of cells correctly in the fluorescent images than in the phase-contrast images.

Results of tracking at different cell densities and time intervals

Figure 6 is a chart showing approximately how the tracking performance of the in-house software varied with cell density

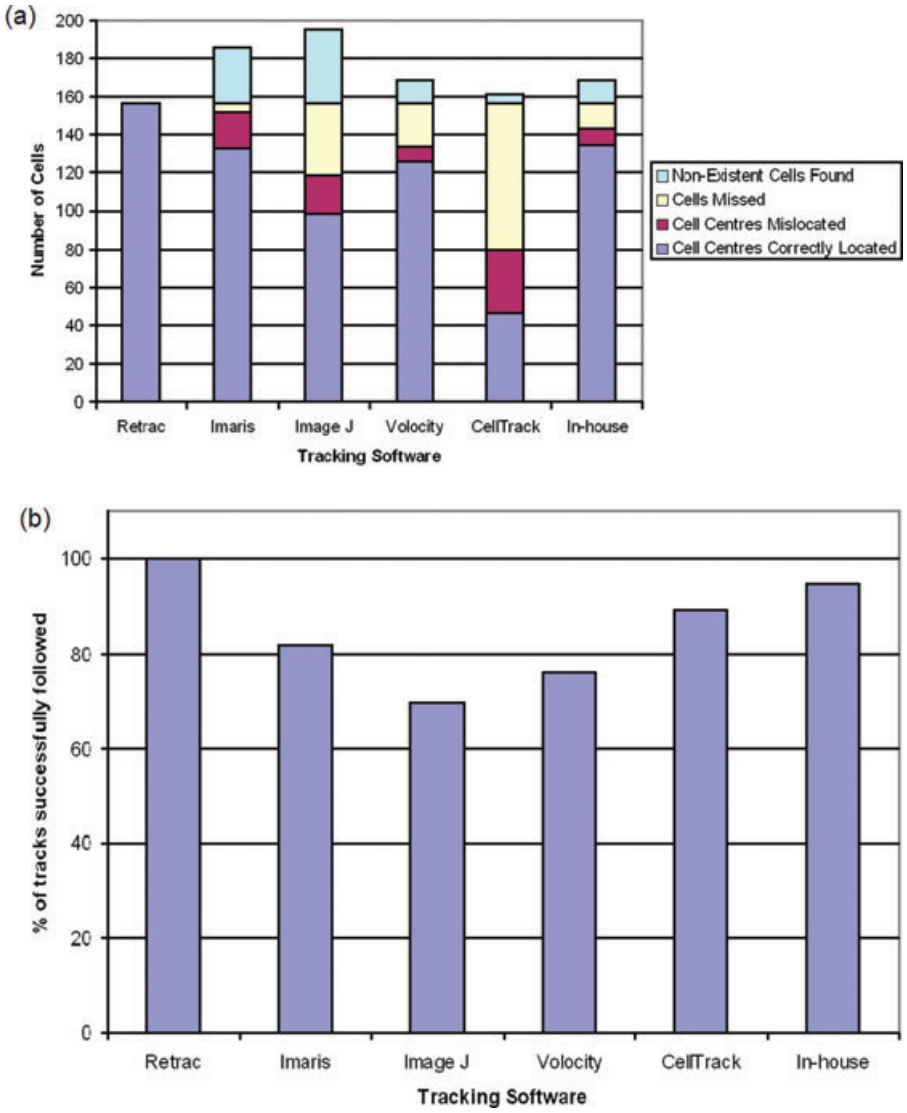

Fig. 3. (a) Stacked bar graph showing the results obtained by each of the tested software systems for the performance criteria relating to the segmentation of the first frame of the first phase-contrast time-lapse sequence. Ideally, the dark blue sections should be as large as possible and the remaining sections as small as possible; (b) Graph showing the percentage of cells that were successfully tracked by each of the tested software systems over all ten frames of the first phase-contrast time-lapse sequence.

and time interval. Performance degrades as cell density and time interval increase as would be expected.

Results of tracking over a large number of frames

Figure 7 shows the mean deviation of the registered cell centres from the actual cell centres against the number of frames being registered. The results show that tracking performance degrades as the number of frames in the sequence increases. This is due to the accumulation of errors mentioned earlier. The figure shows that reasonable tracking performance is obtained up to approximately 50 frames, after which the mean deviation increases steadily.

Discussion

Although the in-house software appeared to perform better than the commercial systems tested when using the phase-contrast images, it is important to remember that these systems are primarily designed to track cells in fluorescently labelled images, as are the vast majority of automated cell-tracking systems currently in existence. As shown in the results, the performance of these systems therefore increases significantly when fluorescence microscopy images are used. A major advantage of an image-registration/optical flow–based system is that the actual image acquisition process used should not significantly affect the tracking performance of the software, so long as the average image intensity remains consistent over

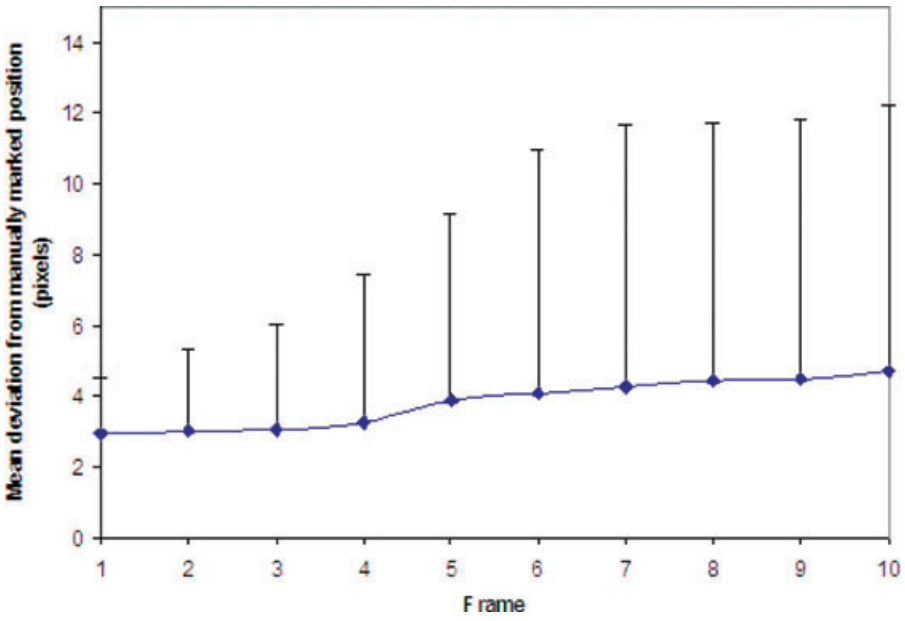

Fig. 4. Graph showing mean deviation of cell positions obtained by the registration software from those obtained using Retrac against the frame number for ten frames of the first phase-contrast microscopy sequence. Error bars show +2 standard deviations.

the entire time-lapse sequence. This is demonstrated by both the in-house software and CellTrack, which showed similar tracking rates regardless of the imaging modality. However, it is also important to ensure that images are taken at small enough intervals to ensure that the cells do not move too great a distance between frames (less than approximately 1 cell diameter between frames is preferred, as the results show that tracking performance degrades rapidly after this point). This problem can often be overcome either by increasing the node spacing used in the registration algorithm or by scaling the images to a smaller size before registration so that the number of pixels moved by the cells per frame is reduced. Using fluorescence microscopy and/or smoothing the images prior to registration can also improve the tracking performance.

The relatively low speed of the in-house software compared with the existing systems is due to the fact that the warping of the segmented cells is currently dealt with in series, meaning that each cell is warped individually. Parallel processing of the cells will reduce the computation time significantly and should be introduced to the software in the near future.

At the present time, a disadvantage of using an image registration–based method, as opposed to a feature-matching method of cell tracking, is that the system will not inherently detect new cells appearing in the field of view. This is due to the fact that, following segmentation of the initial frame of the sequence, no further image segmentation is performed. Hence, the in-house software is currently unable to automatically track new cells following divisions, unlike the existing systems tested here. The fact that only one image is segmented also means that any errors in the initial segmentation, or in the registration process, accumulate as registration of the sequence progresses. When this occurs, the cell appears to become 'smeared', as some of the pixels within the segmented cell are moved, whereas others remain stationary (see Fig. 8(a)). This effect was the cause of the miscalculation of the centroid positions for six of the seven cells that were tracked incorrectly by the in-house software. The seventh cell that was tracked incorrectly was a very small cell that moved a distance greater than its diameter between frames. As such, the tracking algorithm failed in this case and lost the cell completely.

Although it is possible that the smearing effect may actually be due to small fragments of the cell being left behind when the cell moves, such as the trailing adhesions described in Webb *et al.* (2002) and Rid *et al.* (2005), there is no clear evidence of this in the image sequence we used, as a zoomed view of one of these cells does not show any visible debris (see Figs 8(b) and (c)).

In order to deal with the problems mentioned earlier, it will be necessary to find a method of identifying new cells appearing in the image. This may be achieved by detecting patterns in the optical flow fields of the time-lapse sequence. Germain *et al.* (1999) described how specific optical flow patterns are seen when a cell enters the mitotic phase. Searching for these patterns during registration may allow dividing cells to be identified and localized segmentation to be performed to locate the new cell when it appears. This could also allow automatic linking of parent and daughter cells.

A method must also be found to reduce the likelihood of cumulative errors in the registration process. This could possibly be achieved by using active contours (Ray & Acton, 2002) initialized at the boundary positions suggested by

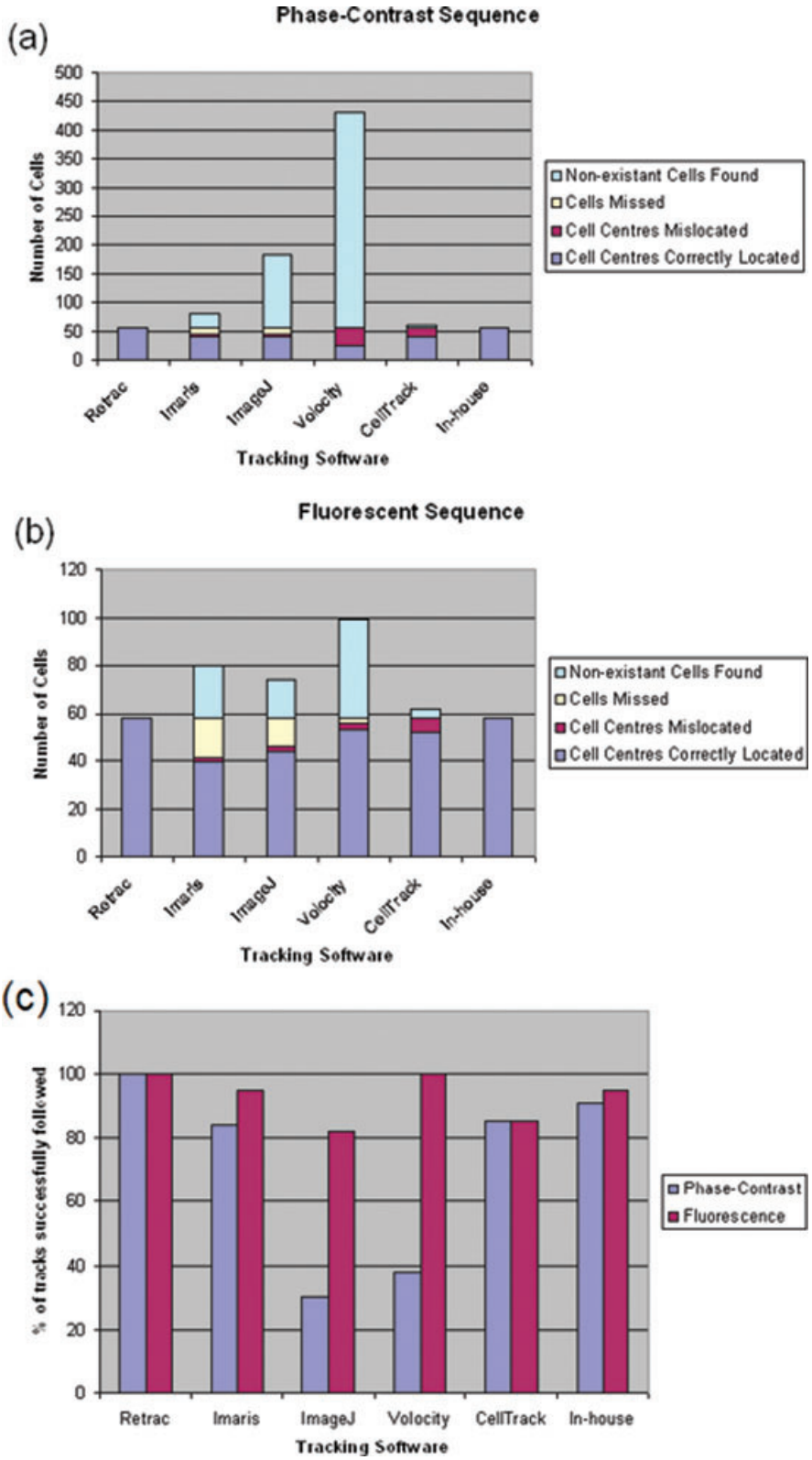

Fig. 5. Stacked bar graphs showing the results obtained by each of the tested software systems for the performance criteria relating to the segmentation of (a) the first frames of the phase-contrast and (b) fluorescence time-lapse sequences. Ideally, the dark blue sections should be as large as possible and the remaining sections as small as possible. (c) Graph showing the percentage of cells that were successfully tracked by each of the tested software systems over all ten frames of the phase-contrast and fluorescence time-lapse sequences.

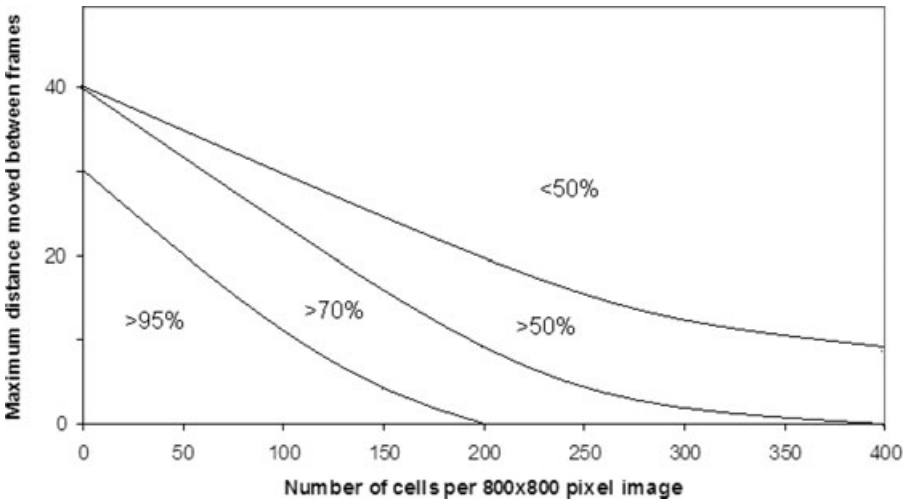

Fig. 6. Diagram showing how the percentage of cells that were successfully tracked over all ten frames of an artificial phase-contrast time-lapse sequence varies with both cell density and distance moved between images.

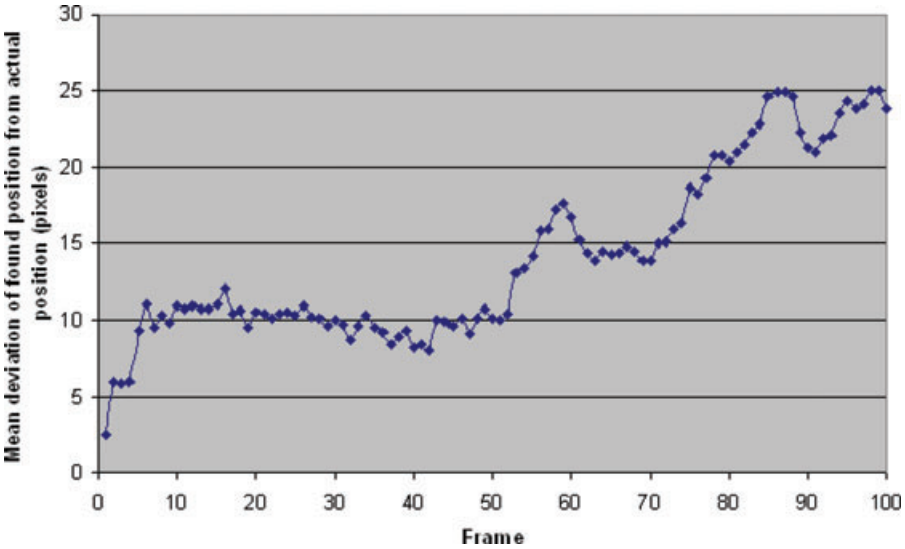

Fig. 7. Graph showing mean deviation of cell positions obtained by the registration software from the actual positions against the frame number for 100 frames of an artificially generated phase-contrast microscopy sequence.

the registration process, as this could be used to search for the nearest cell 'edge' in the image and realign the registered cell boundaries with the detected edges for each frame. Active contours methods have already been used to successfully track small numbers of cells in phase-contrast images (see Shen *et al.* [2006] and Sacan *et al.* [2008]). However, this is computationally demanding and would increase the processing time significantly when dealing with large numbers of cells. It would also require prior knowledge of the cell shape in order to determine the appropriate constraints and weightings to be used in the algorithm. An alternate approach may be to use the 'smearing' effect seen when the cumulative errors occur to identify that the cell may not have been registered correctly and to perform a further localized segmentation. If this can be achieved, the tracking success rate could be increased even further. Until this problem is resolved, the smeared cells are easily identified and can simply be removed from the results before further analysis is performed, as this will further improve the accuracy of the migration statistics extracted from the data.

A further issue is that the software does not currently have a method to deal explicitly with overlapping cells. When two cells pass over each other in the image, the algorithm is not able to determine which cell is moving in which direction when the cells separate and as such it may either continue to track the wrong cell or attempt to track both as though they

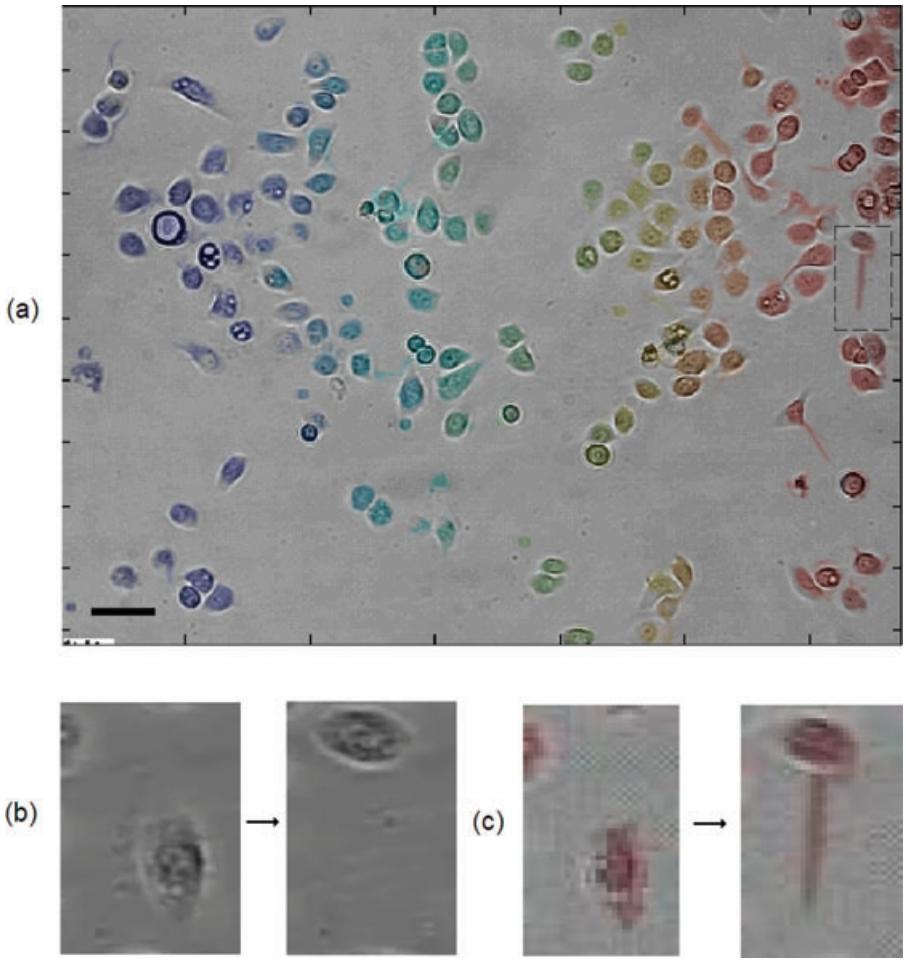

Fig. 8. (a) Final frame of the first phase-contrast time-lapse sequence with locations of mapped cells overlaid (coloured). Scale bar length = 100 pixels or 32.5 μm. Note 'smearing' effect where cumulative errors have caused some of the segmented cells to develop tails (marked by dotted line); (b) Zoomed images of the first and final frames of the sequence showing the area outlined in (a); (c) Zoomed images with registered cell position overlaid in red.

were a single cell. We will be looking into finding a solution to this problem in the near future.

Conclusions

The results show that image registration can be used to effectively track cells in phase-contrast time-lapse microscopy images within a reasonable time frame. Under the specific tracking conditions tested here, the in-house software developed using registration achieved a higher tracking success rate than the currently available systems that were also tested when phase-contrast images were used; however, the processing time was approximately double that of the commercial software. The results also show that low-contrast cell images can be segmented quite successfully using the background removal method described, although there are other established methods for segmentation that could be used. The results also demonstrate that cells can be tracked with a greater efficiency by using fluorescence microscopy than by using phase contrast. For the in-house software, the difference in tracking success was small (<5%), whereas the improvement in performance was much larger for the alternate systems. When using fluorescence imaging, Volocity proved to be the most successful, tracking 100% of the cells correctly.

Software availability

The registration algorithm upon which the in-house system is based is written in C. An executable file containing the registration algorithm will be made publicly available via the Virtual Physiological Human Network of Excellence (http://www.vph-noe.eu/). The cell-tracking application itself is written in MATLAB. An executable file of the cell-tracking application is not currently available for download, but the source code may be made available on request.

Acknowledgements

We thank EPSRC for providing a PhD studentship for A.J. Hand. The Wellcome Trust (grant no. GR077544AIA) is acknowledged for support of the University of Sheffield Light Microscopy Facility.

References

Althoff, K. (2005) *Segmentation and tracking algorithms for in vitro cell migration analysis*. Doctoral Thesis, Department of Signals and Systems, Chalmers University of Technology, Gothenberg, Sweden.

Barber, D.C. (1999) Efficient nonlinear registration of 3D images using high order co-ordinate transfer functions. *J. Med. Eng. Technol.* **23**, 157–168.

Barber, D.C. & Hose, D.R. (2005) Automatic segmentation of medical images using image registration: diagnostic and simulation applications. *J. Med. Eng. Technol.* **29**, 53–63.

Barber, D.C., Oubel, E., Frangi, A.F. & Hose, D.R. (2007) Efficient computational fluid dynamics mesh generation by image registration. *Med. Image Anal.* **11**, 648–662.

Beucher, S. & Lantuejoul, C. (1979) Use of watersheds in contour detection. *Proceedings of the International Workshop on Image Processing, Real-Time Edge and Motion Detection/Estimation*, Rennes, France, pp. 17–21.

Cheriet, M., Said, J. & Suen, C. (1998) A recursive thresholding technique for image segmentation. *IEEE Trans. Image Process.* **7**, 918–921.

De Hauwer, C., Darro, F., Camby, I., *et al.* (1999) In vitro motility evaluation of aggregated cancer cells by means of automatic image processing. *Cytometry* **36**, 1–10.

Debeir, O., Milojevic, D., Leloup, T., *et al.* (2005) Mitotic tree construction by computer in vitro cell tracking: a tool for proliferation and motility features extraction. *Computer as a Tool, 2005. EUROCON 2005.The International Conference* **2**, 951–954.

Germain, F., Doisy, A., Ronot, X. & Tracqui, P. (1999) Characterization of cell deformation and migration using a parametric estimation of image motion. *IEEE Trans. Biomed. Eng.* **46**, 584–600.

Gonzalez, R. & Woods, R. (2002) *Digital Image Processing*, 2nd edn. Publishing House of Electronics Industry, Beijing.

Gor, V., Bacarian, T., Elowitz, M. & Mjolsness, E. (2005) Tracking cell signals in fluorescent images. *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) – Workshops* **3**, 142.

Lester, H. (1999) *Non-linear registration of medical images*. PhD Thesis, University of London.

Lester, H. & Arridge, S. (1999) A survey of hierarchical nonlinear medical image registration. *Pattern Recognit.* **32**, 129–149.

Li, K., Miller, E., Weiss, L., Campbell, P. & Kanade, T. (2006) .Online tracking of migrating and proliferating cells imaged with phase contrast microscopy. *Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW '06)*, pp. 65–72.

Li, K. & Kanade, T. (2007) Cell population tracking and lineage construction using multiple-model dynamics filters and spatiotemporal optimization. *Proceedings of the 2nd International Workshop on Microscopic Image Analysis with Applications in Biology (MIAAB)*. Piscatauray, NJ, USA.

Maintz, J.B. & Viergever, M.A. (1998) A survey of medical image registration. *Med. Image Anal.* **2**, 1–36.

Martens, L., Monsieur, G., Ampe, C., Gevaert, K. & Vandekerckhove, J. (2006) Cell motility: a cross-platform, open source application for the study of cell motion paths. *BMC Bioinformatics* **7**, 289.

Miura, K. (2005) Tracking movement in cell biology. *Adv. Biochem. Eng. Biotechnol.* **95**, 267–295.

Neumann, B., Held, M., Liebel, U., *et al.* (2006) High-throughput RNAi screening by time-lapse imaging of live human cells. *Nat. Methods* **3**, 385–390.

Pal, N. & Pal, S. (1993) A review on image segmentation techniques. *Pattern Recognit.* **26**, 1277–1294.

Rabut, G. & Ellenberg, J. (2004) Automatic real-time three-dimensional cell tracking by fluorescence microscopy. *J. Microsc.* **216**, 131–137.

Ray, N. & Acton, S. (2002) Active contours for cell tracking. Image Analysis and Interpretation, 2002. *Proceedings of the Fifth IEEE Southwest Symposium on*, pp. 274–278.

Rid, R., Schiefermeier, N., Grigoriev, I., Small, J.V. & Kaverina, I. (2005) The last but not the least: the origin and significance of trailing adhesions in fibroblastic cells. *Cell Motil Cytoskeleton* **61**, 161–171.

Sacan, A., Ferhatosmanoglu, H. & Coskun, H. (2008) Cell Track: an open-source software for cell tracking and motility analysis. *Bioinformatics* **24**, 1647–1649.

Sezgin, M. & Sankur, B. (2004) Survey over image thresholding techniques and quantitative performance evaluation. *J. Electron. Imaging* **13**, 146–168.

Shen, H., Nelson, G., Kennedy, S., *et al.* (2006) Automatic tracking of biological cells and compartments using particle filters and active contours. *Chemometrics and Intelligent Laboratory Systems* **82**, 276–282.

Walker, D.C., Southgate, J., Hill, G., *et al.* (2004) The epitheliome: agent-based modelling of the social behaviour of cells. *Biosystems* **76**, 89–100.

Webb, D.J., Parsons, J.T. & Horwitz, A.F. (2002) Adhesion assembly, disassembly and turnover in migrating cells – over and over and over again. *Nat. Cell Biol.* **4**, E97–E100.

Zimmer, C., Labruyere, E., Meas-Yedid, V., Guillen, N. & Olivo-Marin, J. (2002) Segmentation and tracking of migrating cells in videomicroscopy with parametric active contours: a tool for cell-based drug testing. *IEEE Trans. Med. Imaging* **21**, 1212–1221.

Appendix: The registration algorithm

All image-registration algorithms start from the basis of a measure of the quality of registration between the two images. Mappings between the images are invariably characterized by a vector of mapping parameters **a**. The goal of the image-registration algorithm is to find **a** such that the registration quality measure is maximized or minimized. Various measures have been proposed, from distances between anatomical landmarks to mutual information. Identification of landmarks or other anatomical features is not trivial and in general requires pre-processing of the image data. Although there will always be cases where this is the only solution that works, the generally preferred solution is to use a measure based on pixel intensities. The simplest measure is based on the sum-of-squared differences between the images. If $f(x,y)$ is the fixed (or destination) image and $m(x,y)$ is the moved (or source) image, then the quality measure Q is given as a function of **a** by

$$Q = \sum_{\text{alt voxels}} (\mathbf{f} - \mathbf{r}(\mathbf{a}))^2, \qquad \text{(A1)}$$

where **r(a)** is the moved image **m** mapped with the mapping defined by the elements of **a**. If **m** can be completely mapped to **f**, then Q will go to zero (apart from the effects of noise on the images). However, a weakness of this quality measure is that if there are significant intensity differences between the images that cannot be eliminated by the mapping, then these differences may drive the registration algorithm to an incorrect solution. A quality measure that appears to avoid the above problem is the Mutual Information measure and its variants, and this is currently taken to be the gold standard approach. Unfortunately, it is computationally very expensive, especially for non-linear mappings. In the present work, the sum-of-squares measure is used, although mutual information will appear again later. However, a modification to the theory is made that makes the use of this measure more acceptable and less sensitive to un-mappable variations in intensity.

Note first that the sum-of-squares measure is completely appropriate for mapping binary images, that is, images whose voxel values are either 0 or 1. Note also that any intensity image can be converted into a binary image by increasing its dimensionality by 1. For example, a two-dimensional (2D) intensity image $f(x, y)$ can be converted to a 3D binary image $f_b(x, y, z)$ by replacing the value $f(x, y)$ by a stack of $n = f(x, y)$ voxels of unit intensity. Similarly a 3D intensity image can be converted to a four-dimensional (4D) binary image. The approach used in this work is therefore to convert an intensity image to a binary image and then use a registration algorithm based on minimizing the sum-of-squares measure. The mapping between two binary images is non-unique and to achieve a unique mapping additional constraints are needed. Actually this is often true for intensity images even if there is in fact a one-to-one mapping between them, but it is more obviously true for binary images.

The relationship between $\mathbf{f_b}$ and $\mathbf{m_b}$ for images of any dimensionality can be written in terms of the mapping function $\mathbf{\Delta r}$ as

$$\mathbf{f_b}(\mathbf{r}) = \mathbf{m_b}(\mathbf{r} + \Delta\mathbf{r}), \tag{A2}$$

where **r** is the position vector. As noted earlier, this relationship will only hold provided that the intensity distribution of $\mathbf{m_b}$ can be mapped to $\mathbf{f_b}$ but as the images are binary this condition can be met. For small displacements, the right-hand side can be expanded using a Taylor expansion. It can be shown with appropriate approximations that

$$\mathbf{f_b}(\mathbf{r}) - \mathbf{m_b}(\mathbf{r}) = \frac{1}{2}\Delta\mathbf{r}(\mathbf{r})\left[\frac{\partial \mathbf{f_b}(\mathbf{r})}{\partial \mathbf{r}} + \frac{\partial \mathbf{m_b}(\mathbf{r})}{\partial \mathbf{r}}\right]. \tag{A3}$$

The definition of spatial differentials for binary images is awkward since these differentials become infinite. However, we can resolve this problem theoretically by dealing with smoothed versions of the binary images, in the limit matching the degree of smoothing to the magnitude of the mapping between the two images.

For 2D images, the binary development is a 3D image (4D for 3D images). However, they are not completely general 3D binary images. In particular, they are not re-entrant along the 'intensity' dimension. For a general binary image, the intensities along a line drawn parallel to any of the axes can change from 0 to 1 and back again any number of times as the line is traversed. For a binary image constructed from an intensity image, the transition from 1 to 0 along the 'intensity' axis can only occur once. For this class of binary images, the 'intensity' dimension is clearly different from the other dimensions. This limitation allows the 3D problem to be returned to a 2D one. The mapping function $\Delta\mathbf{r}$ is a function of x,y and the intensity dimension w. The three components of this are given by $\Delta x(x, y, w)$, $\Delta y(x, y, w)$, and $\Delta w(x, y, w)$. Since the image is not re-entrant in the w direction, we can make the first two components of the mapping function independent of w. The same could be done for the intensity function. In fact it is more useful to make it more generally linearly dependent in s. Then the mapping function becomes $\Delta x(x,y, w) = \Delta x(x, y)$, $\Delta y(x, y, w) = \Delta y(x, y)$ and $\Delta w(x, y, w)^* = \Delta s(x, y)^* w + \Delta t(x, y)$. If these are inserted into Eq. (A.3) and integrated over s, the result is

$$\begin{aligned} f(\mathbf{r}) - m(\mathbf{r}) = \frac{1}{2}\Bigg[& \Delta x(\mathbf{r})\left[\frac{\partial \mathrm{f}(\mathbf{r})}{\partial x} + \frac{\partial \mathrm{m}(\mathbf{r})}{\partial x}\right] \\ & + \Delta y(\mathbf{r})\left[\frac{\partial \mathrm{f}(\mathbf{r})}{\partial y} + \frac{\partial \mathrm{m}(\mathbf{r})}{\partial y}\right] \\ & - \Delta s(\mathbf{r})[\mathrm{f}(\mathbf{r}) + \mathrm{m}(\mathbf{r})] - 2\Delta t(\mathbf{r})\Bigg]. \end{aligned} \tag{A4}$$

In this case, the functions **f** and **m** are the original intensity functions, and the registration is reduced to a 2D problem again.

The last two terms on the right-hand side of this equation are clearly terms that can deal with un-mappable (in 2D) intensity differences between the two images. As with the binary images, these terms make the mapping non-unique. However, the mapping can be made unique by adding further constraints, such as smoothness, and this is done here.

In order to compute the mapping, the mapping function is expanded in terms of a set of basis functions.

$$\begin{aligned} & f(x, y) - m(x, y) = \\ & \frac{1}{2}\left[\begin{array}{l} \sum\limits_{\text{all } i} a_{xi}\phi_i\left[\frac{\partial f}{\partial X} + \frac{\partial m}{\partial X}\right] + \sum\limits_{\text{all } i} a_{yi}\phi_i(x, y)\left[\frac{\partial f}{\partial y} + \frac{\partial m}{\partial y}\right] + \\ -\sum\limits_{\text{all } i} a_{si}\phi_i(x, y)[f(x, y) + m(x, y)] - 2\sum\limits_{\text{all } i} a_{ti}\phi_i(x, y) \end{array}\right]. \end{aligned} \tag{A5}$$

In the present work, the basis functions are local. The parameters of the mapping, the a_{ji}, represent the values of the mapping at nodes on a square grid. The intermediate points on the mapping are obtained by bi-linear interpolation between these points. The basis functions ϕ_i are therefore the familiar bi-linear interpolation functions. Formally, the basis function

centred on the grid point x_i, y_i can be written as

$$\phi i(x, y) = \left(1 - \left|\frac{x - x_i}{d}\right|\right)\left(1 - \left|\frac{y - y_i}{d}\right|\right) \times \left\{-1 <= \left(\frac{x - x_i}{d}, \frac{y - y_i}{d}\right) <= 1\right\}, \tag{A6}$$

where d is the spacing between the nodes of the registration grid.

Equation (A5) can be reduced to matrix form as

$$\mathbf{f} - \mathbf{m} = \mathbf{Ta} \tag{A7}$$

Finding a vector **a** that satisfies this equation finds the mapping which maps **m** to **f**. However, as Eqs (A4) and (A5) were reached through a Taylor expansion that dropped higher-order terms they are only accurate for small values of **a**, that is, small displacements of **m** relative to **f**. The correct values of **a** need to be found using an iterative approach. Following Lester (1999), it can easily be shown that if $\mathbf{a}_n$ is the current estimate of **a**, then $\mathbf{a}_{n+1} = \mathbf{a}_n + \Delta\mathbf{a}$ where

$$\Delta\mathbf{a} = [\mathbf{T}^t\mathbf{T} + \lambda\mathbf{L}^t\mathbf{L}]^{-1}(\mathbf{T}^t(\mathbf{f} - \mathbf{m}(\mathbf{a_n})) - \lambda\mathbf{L}^t\mathbf{L}\mathbf{a_n}). \tag{A8}$$

The matrix **L** is a Laplacian operator and $\mathbf{L}^t\mathbf{L}$ imposes a smoothing constraint on the solution for $\Delta\mathbf{a}$. $\mathbf{m}(\mathbf{a}_n)$ is the image **m** mapped with the mapping defined by $\mathbf{a}_n$. The parameter λ controls the degree of smoothing imposed on the solution.

As registration proceeds, $\mathbf{m}(\mathbf{a}_n)$ approaches **f**. **T** is a function of $\mathbf{m}(\mathbf{a}_n)$ and hence of $\mathbf{a}_n$. Since local basis functions are used, this matrix is sparse and Eq. (A7) can be solved efficiently by conjugate gradient methods.

As with many problems of this sort, there is no clear basis for the determination of λ. In practice, $\mathbf{T}^t\mathbf{T}$ is very ill conditioned (and in some cases, may be singular) and that $\mathbf{L}^t\mathbf{L}$ is also singular. However, the combination of the two matrices is in general not singular. The value of λ is chosen to be the value that minimizes the condition number of $\mathbf{T}^t\mathbf{T} + \lambda\mathbf{L}^t\mathbf{L}$ and hence makes the solution of Eq. (A8) most robust.

Starting with an initial value of $\mathbf{a}_0 = 0$, this gives an iterative method by which **a** can be calculated.

Updating the displacement vector

In principle, each $\Delta\mathbf{a}$ is used to map the image $\mathbf{r}(\mathbf{a})$ closer to the fixed image **f**. However, interpolation smoothes the image data, and many iterative steps would degrade the registered image. For this reason, it is appropriate to store a mapping function that is updated at each iteration, and this function is applied to the original image **m.** However, updating the mapping function is not a simple additive process. After the first $\Delta\mathbf{a}$ is calculated (and added to zero to produce the first value of **a**), the image **m** is moved to a new position nearer **f** using **a**. At the next iteration a new $\Delta\mathbf{a}$ is calculated, but it is incorrect to directly add this to the current value of **a** to try and generate the next mapping from **m** to **f**. The increment that should be added is $\Delta\mathbf{a}$ transformed back to the space of **m**. This is easily done.

If the corners of the square grid are (x_1, y_1), (x_2, y_2) etc. and the corners of the transformed grid are (u_1, v_1), (u_2, v_2) etc., then these are related through

$$\begin{bmatrix} u_1 & u_2 & u_3 & u_4 \\ v_1 & v_2 & v_3 & v_4 \end{bmatrix} = A \begin{bmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ x_1y_1 & x_2y_2 & x_3y_3 & x_4y_4 \end{bmatrix} = \boldsymbol{U} = \boldsymbol{A}X, \tag{A9}$$

where A is the transformation matrix. Then we can compute **A** from

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{X}^t[\boldsymbol{X}^t\boldsymbol{X}]^{-1}. \tag{A10}$$

At each stage in the iteration, we have a current 'registered' image and calculate the displacement vector to move this image further towards the fixed image. However, as a running total of the overall displacement is being kept, this displacement vector must be transformed into the space of the total transform. This is done, updating as we go along, by recognizing that if the displacement vector for the current iteration is $(u = x + \Delta u, v = y + \Delta v)$, the total displacement vector in the space of the totalized transform is this value transformed by A.