



City Research Online

City, University of London Institutional Repository

Citation: Ulman, V., Maska, M., Magnusson, K. E. G., Ronneberger, O., Haubold, C., Harder, N., Matula, P., Matula, P., Svoboda, D., Radojevic, M., et al (2017). An objective comparison of cell-tracking algorithms. *Nature Methods*, 14(12), pp. 1141-1152. doi: 10.1038/nmeth.4473

This is the supplemental version of the paper.

This version of the publication may differ from the final published version.

Permanent repository link: <https://openaccess.city.ac.uk/id/eprint/18857/>

Link to published version: <https://doi.org/10.1038/nmeth.4473>

Copyright: City Research Online aims to make research outputs of City, University of London available to a wider audience. Copyright and Moral Rights remain with the author(s) and/or copyright holders. URLs from City Research Online may be freely distributed and linked to.

Reuse: Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge. Provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way.

City Research Online:

<http://openaccess.city.ac.uk/>

publications@city.ac.uk

Supplementary Material: Figures

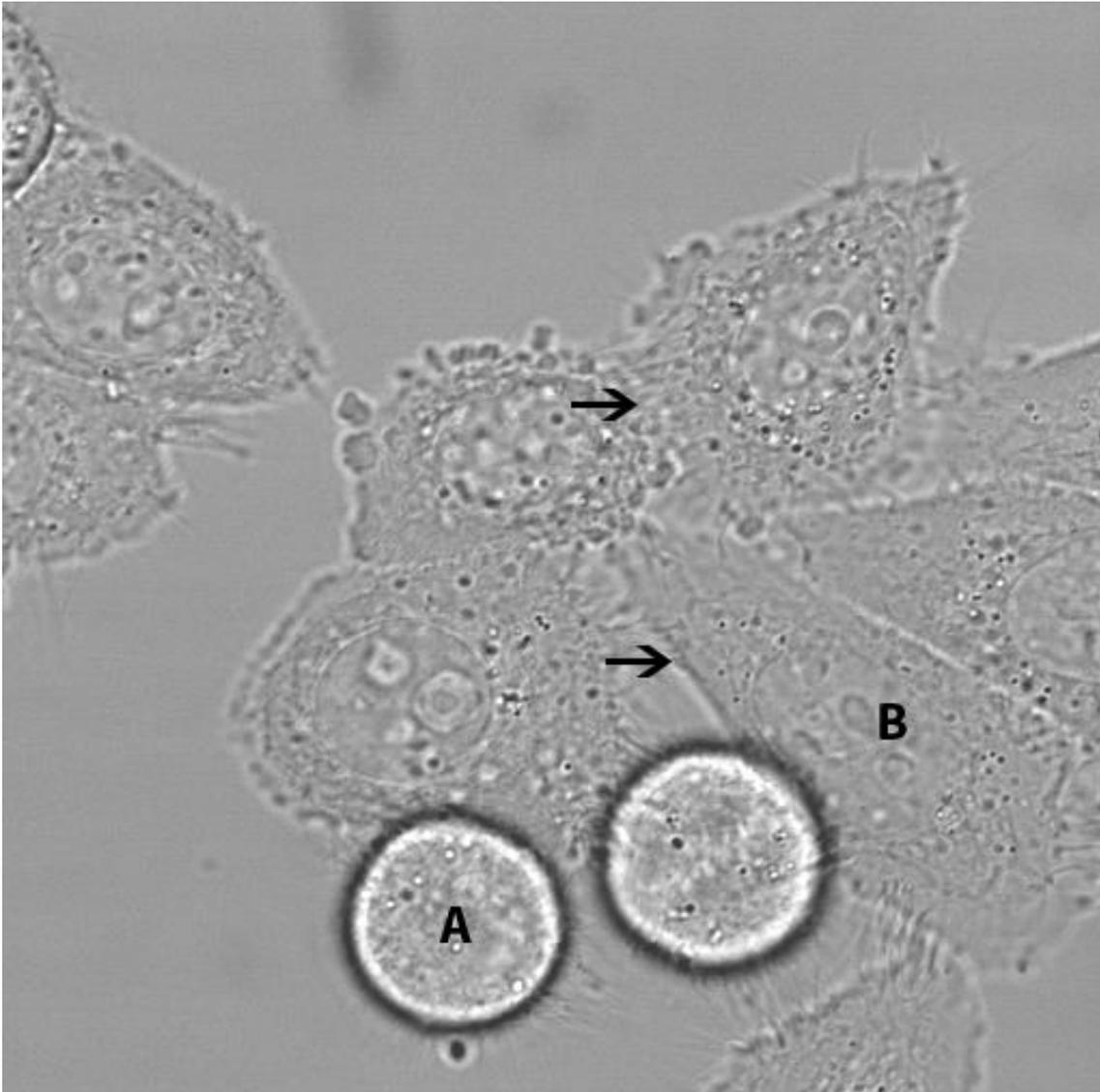


Figure 1. Representative frame of DIC-C2DH-HeLa videos. This dataset presents low *SNR* and *CR* values characteristic of phase-enhancement microscopy techniques mostly because the average intensity of the cells is very similar to the intensity of the background. The signal inside the cells is highly heterogeneous due to DIC-highlighted internal structures and organelles. The cells are heterogeneous in intensity and shape since in most frames co-exist well spread, cuboidal, low intensity interphase cells (B) with rounded, bright cells undergoing mitosis. The cells are highly packed and show very low intensity changes between neighboring cells (black arrows).

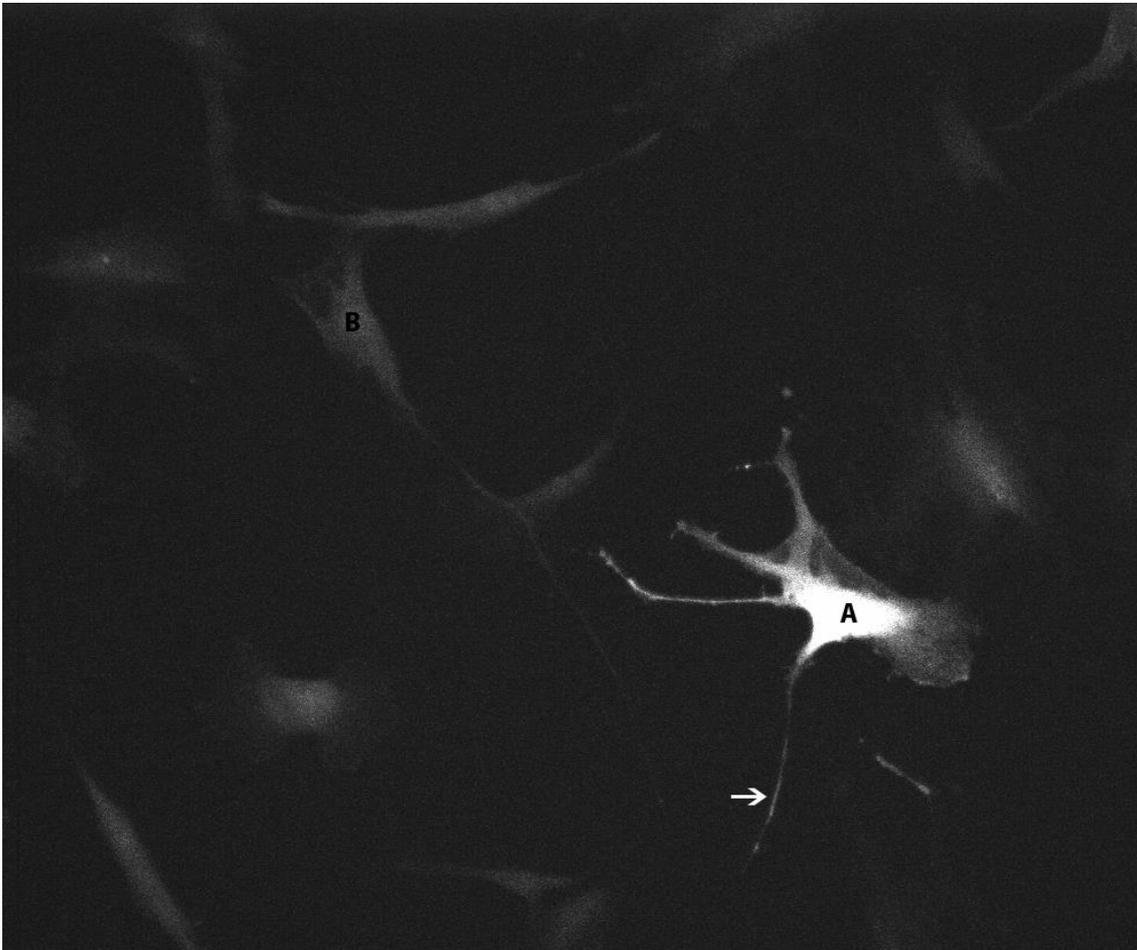


Figure 2. Representative frame of Fluo-C2DL-MSV videos. The *SNR* and *CR* values are low, due to the low level of emission of the fluorescent cytoplasmic reporter, especially in the long, thin filopodial extensions of the cell (white arrow). The intensity is also quite variable in different parts of the cell (see the different intensity of the nucleus and cytoplasm in cell A), and the cells present different levels of intensity (compare cells A and B), possibly due to different expression of the transfected fluorescent reporter. The shape of the cells is highly irregular due to the long filopodial extensions (e.g. white arrow). The cells show a significant degree of bleaching and move fast, causing low overlap of the cells between consecutive frames.

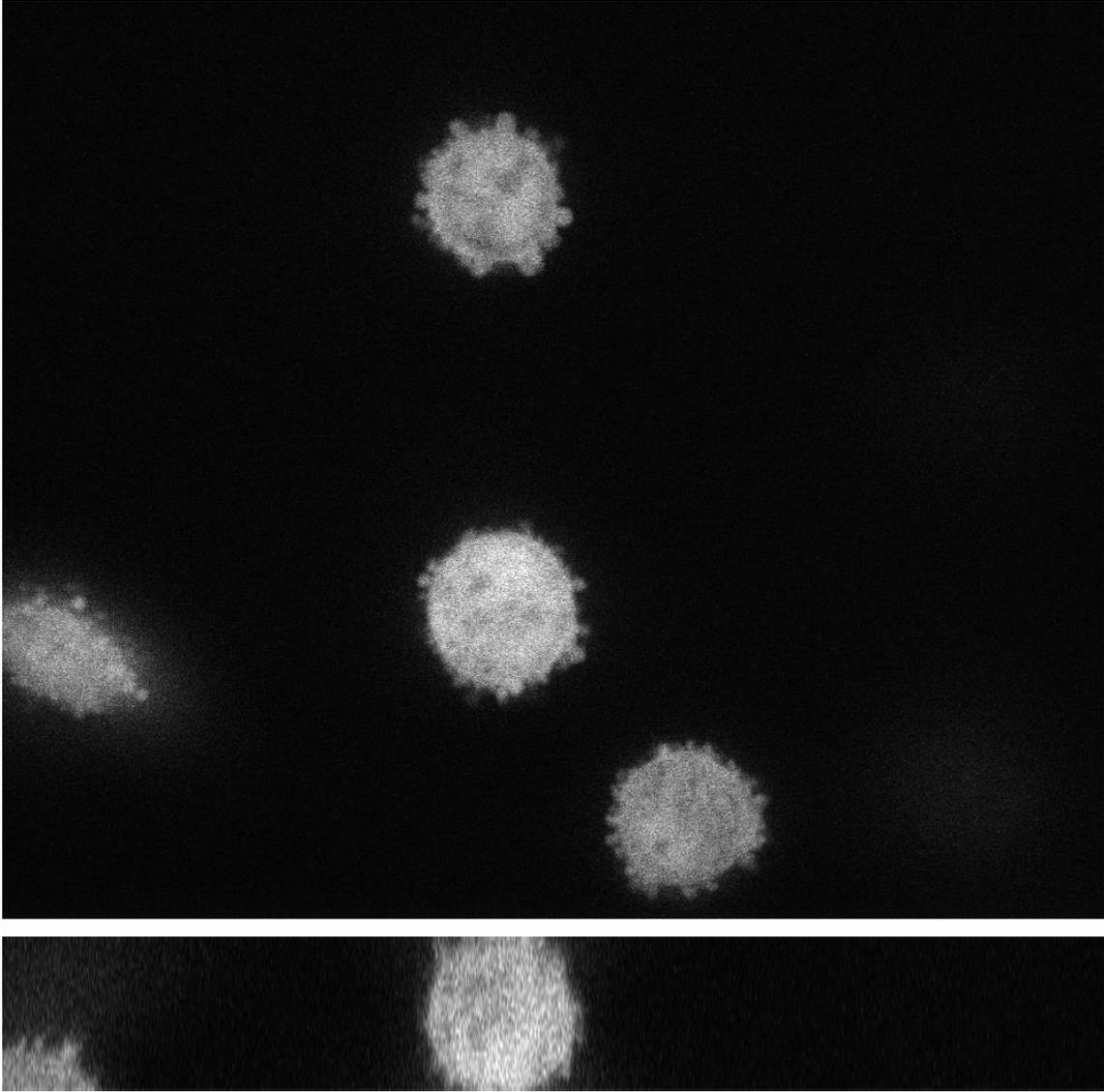


Figure 3. Representative xy (top) and xz (bottom) slices of a frame of Fluo-C3DH-H157 videos. This dataset displays reasonably good values for most properties, with the exception of some signal decay due to photobleaching. The presence of prominent blebs, and some heterogeneity between cell intensities can also complicate accurately segmenting and delineating the cell boundaries.

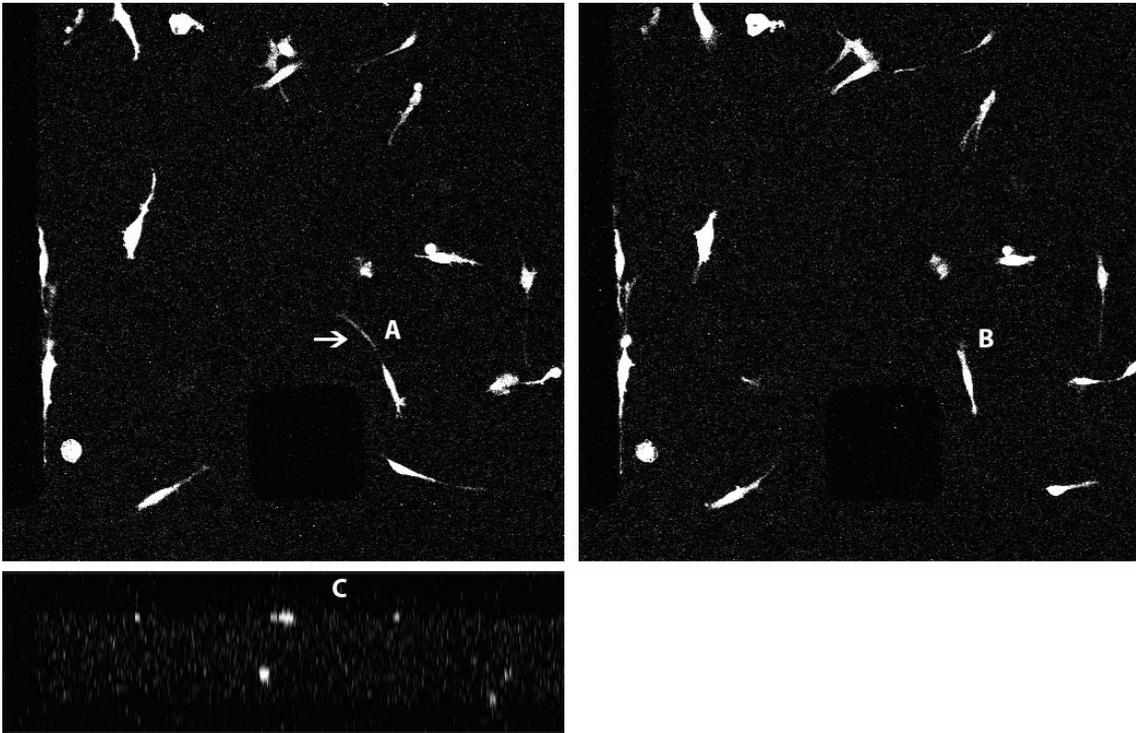


Figure 4. xy (top left and right) slices from two consecutive frames and one xz (bottom) slice of a frame of Fluo-C3DL-MDA321 videos. The *SNR* and *CR* of this dataset are relatively low due to both low signal intensity and increased background, which affects the quality of the signal especially in the long migration-related filopodial extensions (e.g. white arrow). This noisy signal efficiency causes high internal heterogeneity. To complicate the segmentation and tracking even further, the images are acquired at low resolution, especially in the axial direction (see bottom panel), and also in the temporal dimension (compare the same cell, A and B in two consecutive frames) and suffer from significant photobleaching.

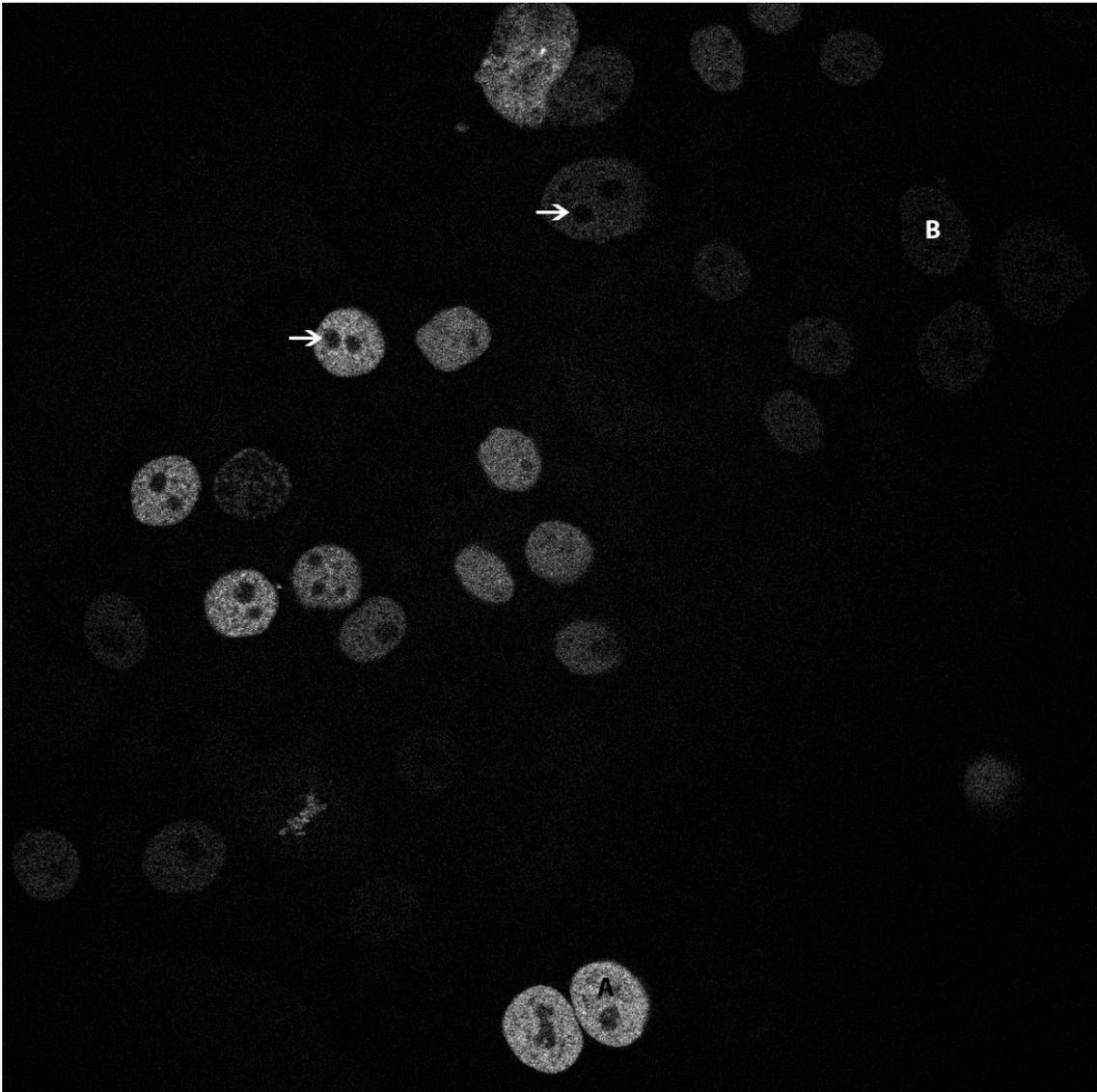


Figure 5. Representative frame of Fluo-N2DH-GOWT1 videos. The signal inside the cell nuclei varies due to the existence of prominent, unlabeled nucleoli (white arrows) and the heterogeneity of the average cell intensities (see for instance the difference between the cells labeled A and B).

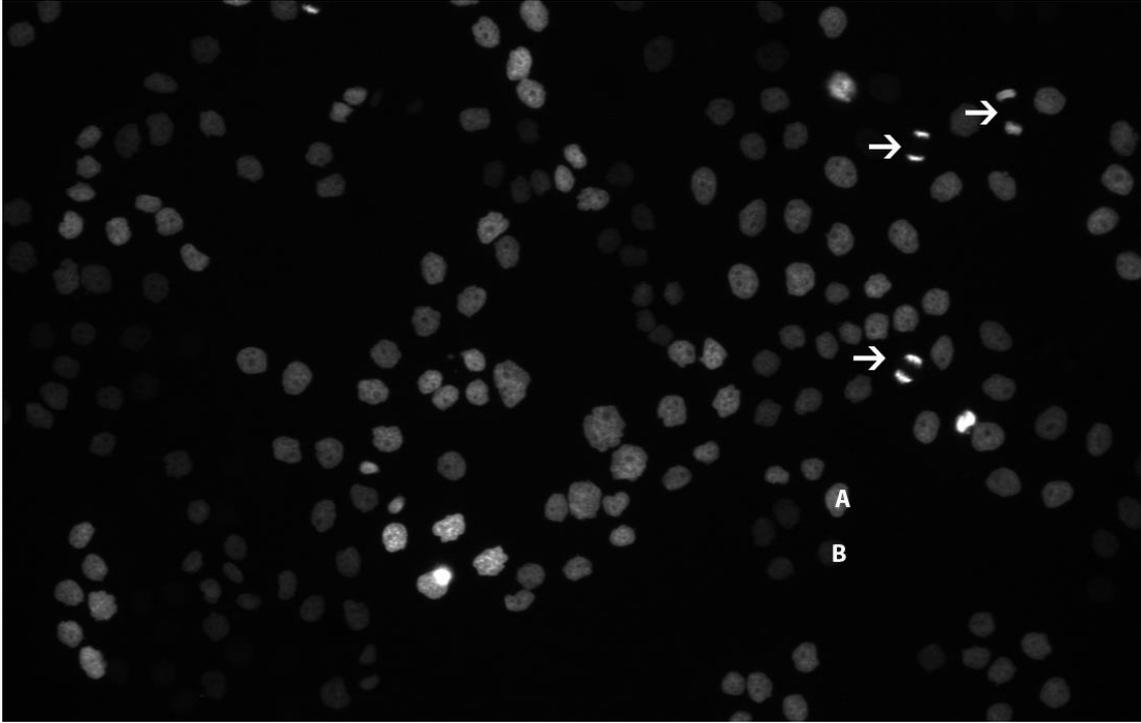


Figure 6. Representative frame of Fluo-N2DL-HeLa videos. The image shows some of the challenges posed by this dataset, including signal heterogeneity between cells as shown by the presence of a large range of nuclear intensity, the low spatial resolution, high cell density, and the presence of division events highlighted by white arrows.

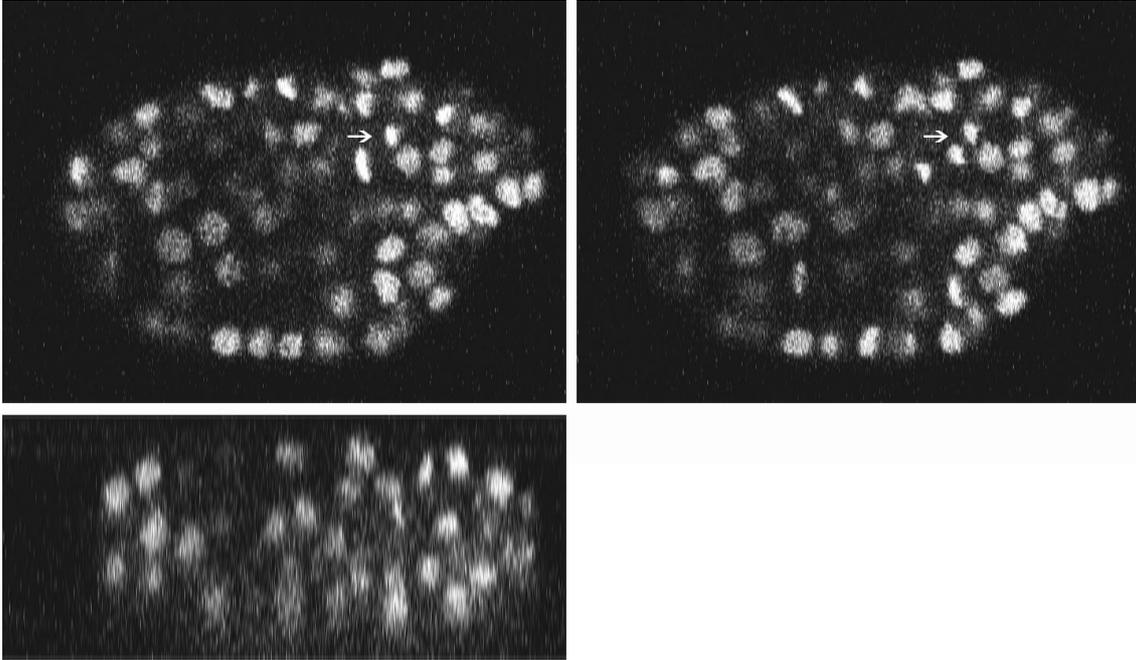


Figure 7. xy (top left and right) slices from two consecutive frames and one xz (bottom) slice of a frame of Fluo-N3DH-CE videos. The most significant problems of this dataset are high cell density, the low resolution in the axial direction (see xz bottom slice), low cell overlap between frames caused by large temporal acquisition step and the abundance of mitotic cells typical of a developing embryo (white arrows).

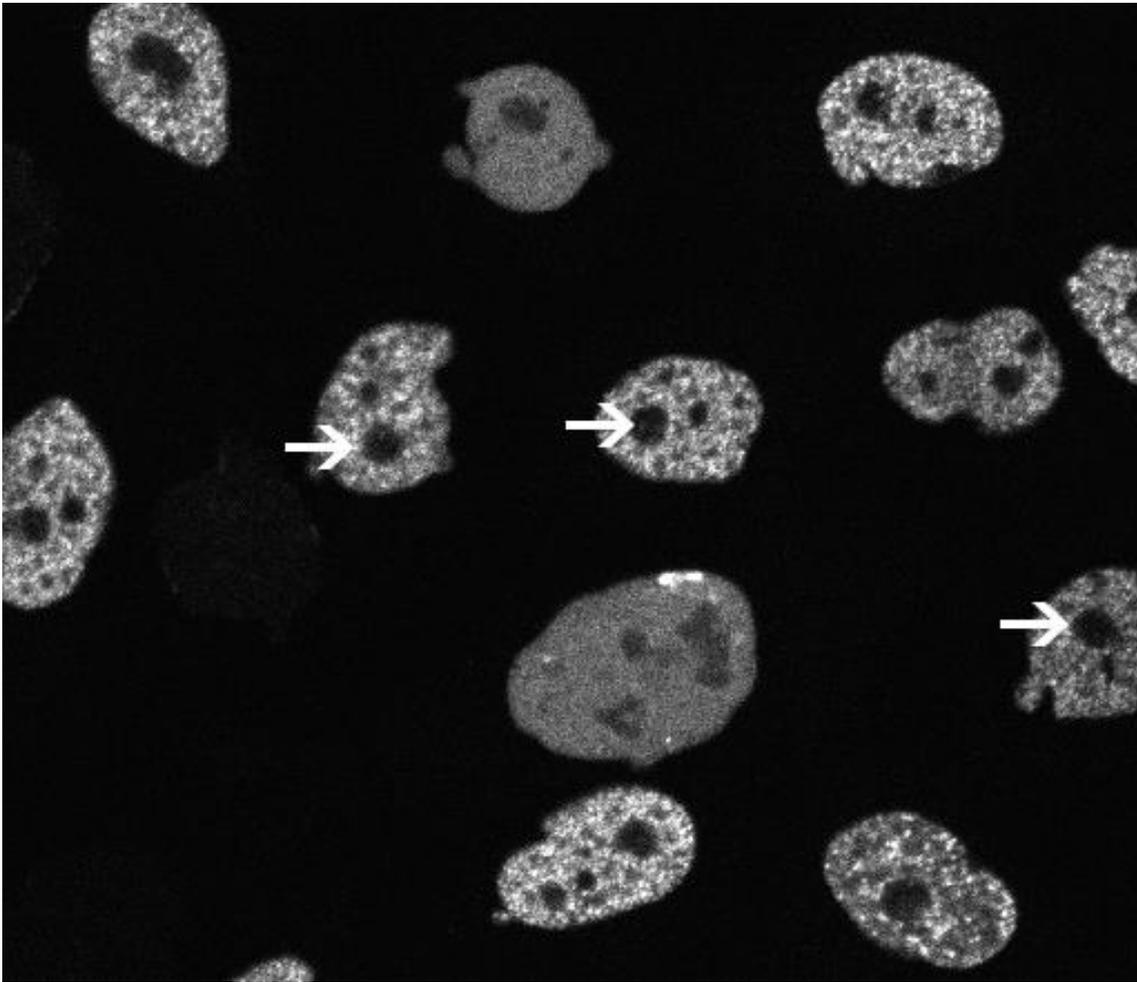


Figure 8. Representative frame of Fluo-N3DH-CHO videos. The main challenges of this dataset are the internal heterogeneity of the staining, clearly visible in the images and caused by the fact that the nuclear staining does not label the nucleoli of the cells (white arrows), and a relatively high cell density.

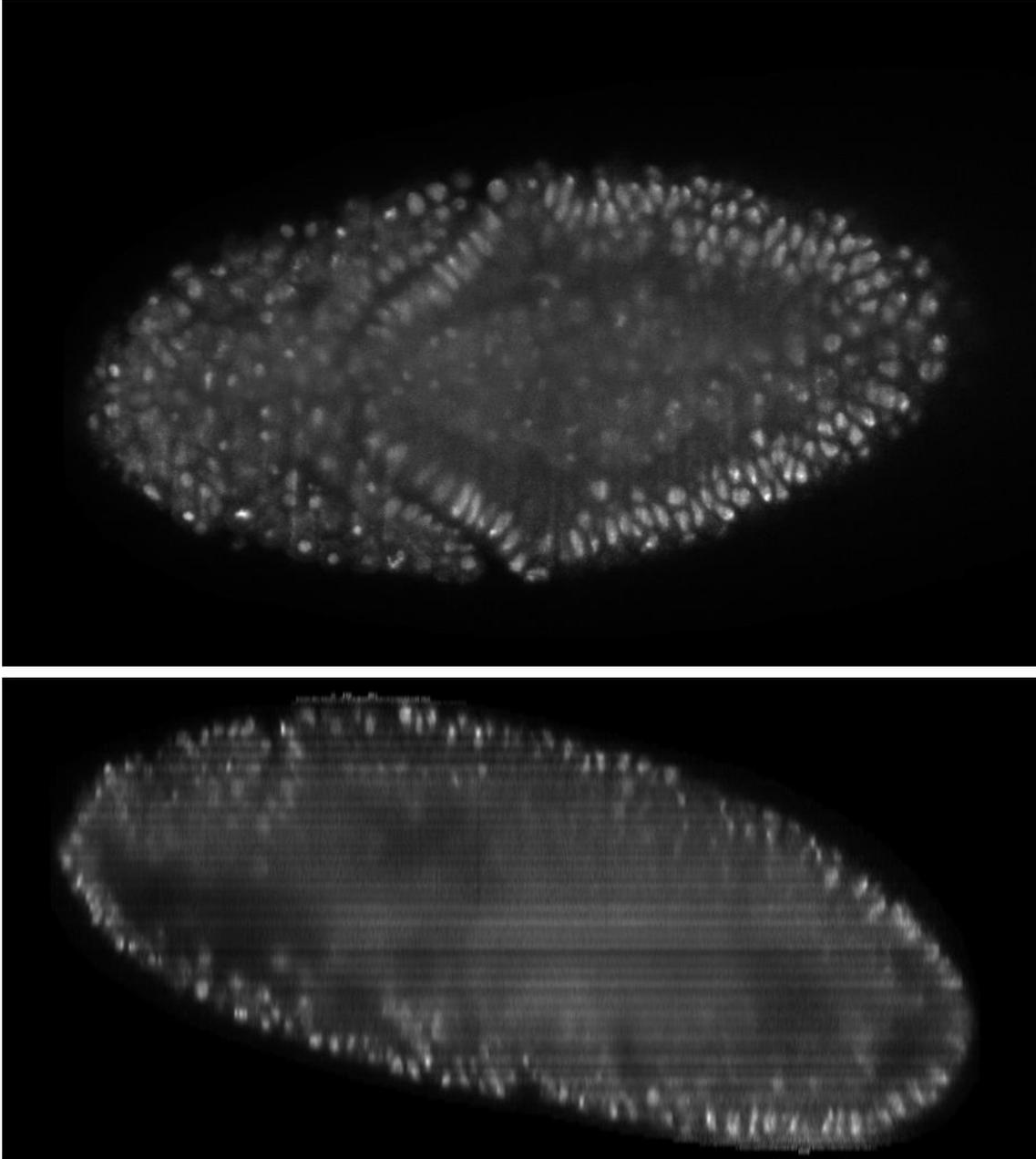


Figure 9. Representative xy (top) and xz (bottom) slices of a frame of Fluo-N3DH-CE videos. The low spatial (visible in this image) and temporal resolution characteristic of light sheet microscopy, and the presence of frequent mitoses typical of a developing embryo render this dataset the most difficult of the challenge datasets.



Figure 10. Representative frame of PhC-C2DH-U373 videos. At this level of resolution the SNR , CR , Het_i and Het_b values are deficient, as expected for a contrast enhancement microscopy modality. All other values are either average or good, which seems to compensate the deficient values for the segmentation and tracking task. Especially beneficial seems to be a high spatial resolution, and a relatively low cells density (visible in the image) and a high temporal resolution.



Figure 11. Representative frame of PhC-C2DL-PSC videos. Most of the parameters are in the average to low range, especially those already mentioned for brightfield modalities. The very low spatial resolution (visible in the image), to some extent compensated by a high temporal resolution, and significant number of division events (black arrows) make the cells in this dataset difficult to segment and track.

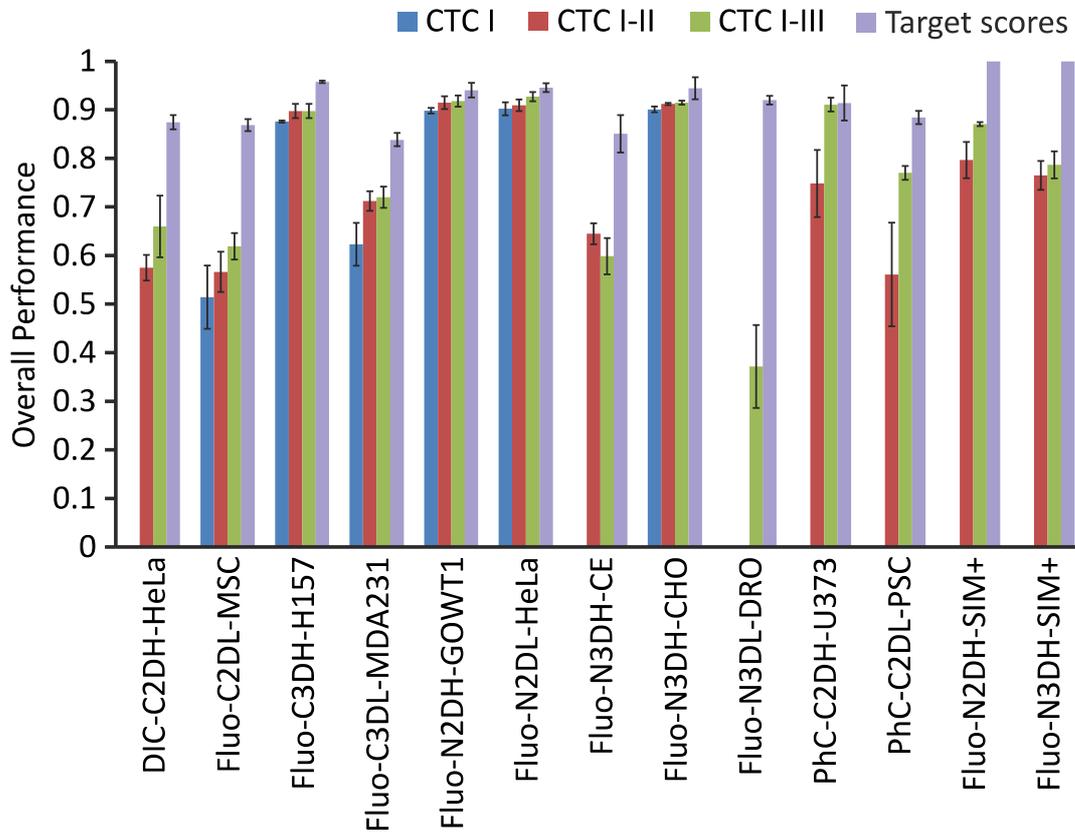


Figure 12. Average overall performance (OP) of the top-3 performing algorithms, if available, binned per dataset across CTC editions. For all datasets except for the synthetic datasets (Fluo-N2DH-SIM+ and Fluo-N3DH-SIM+), the target scores correspond to the average OP scores over the three individual manual annotations. For the two embryonic datasets (Fluo-N3DH-CE and Fluo-N3DL-DRO) there was only one tracking annotation available. Accordingly, the target scores for those datasets were calculated by considering 1.0 (i.e., perfect match to the provided tracking ground truth) as the tracking scores of the three individual manual annotations. Note that missing bars correspond to datasets not offered at a particular CTC edition. A decreasing trend for Fluo-N3DH-CE is caused by the fact that only two submissions were received for this dataset in CTC II, not allowing one to average the OP scores over the three best performing algorithms as in the case of the third CTC edition.

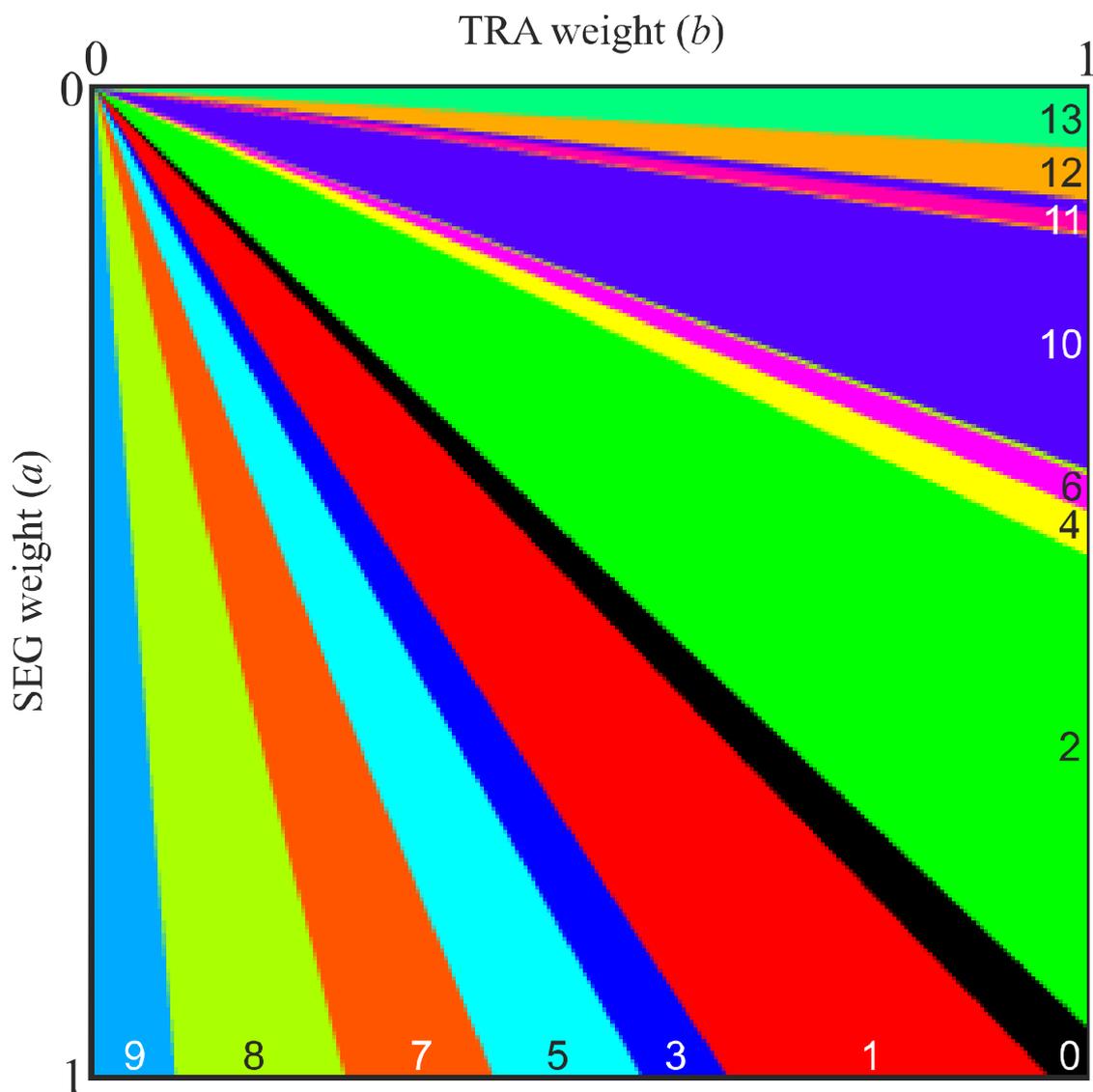


Figure 1213. Robustness of the weighting used for **SEG** and **TRA**. The image shows the number of rank changes in the top-three ranked methods, as we change the weight given to **SEG** and **TRA** from 0 to 1, in 0.001 steps.

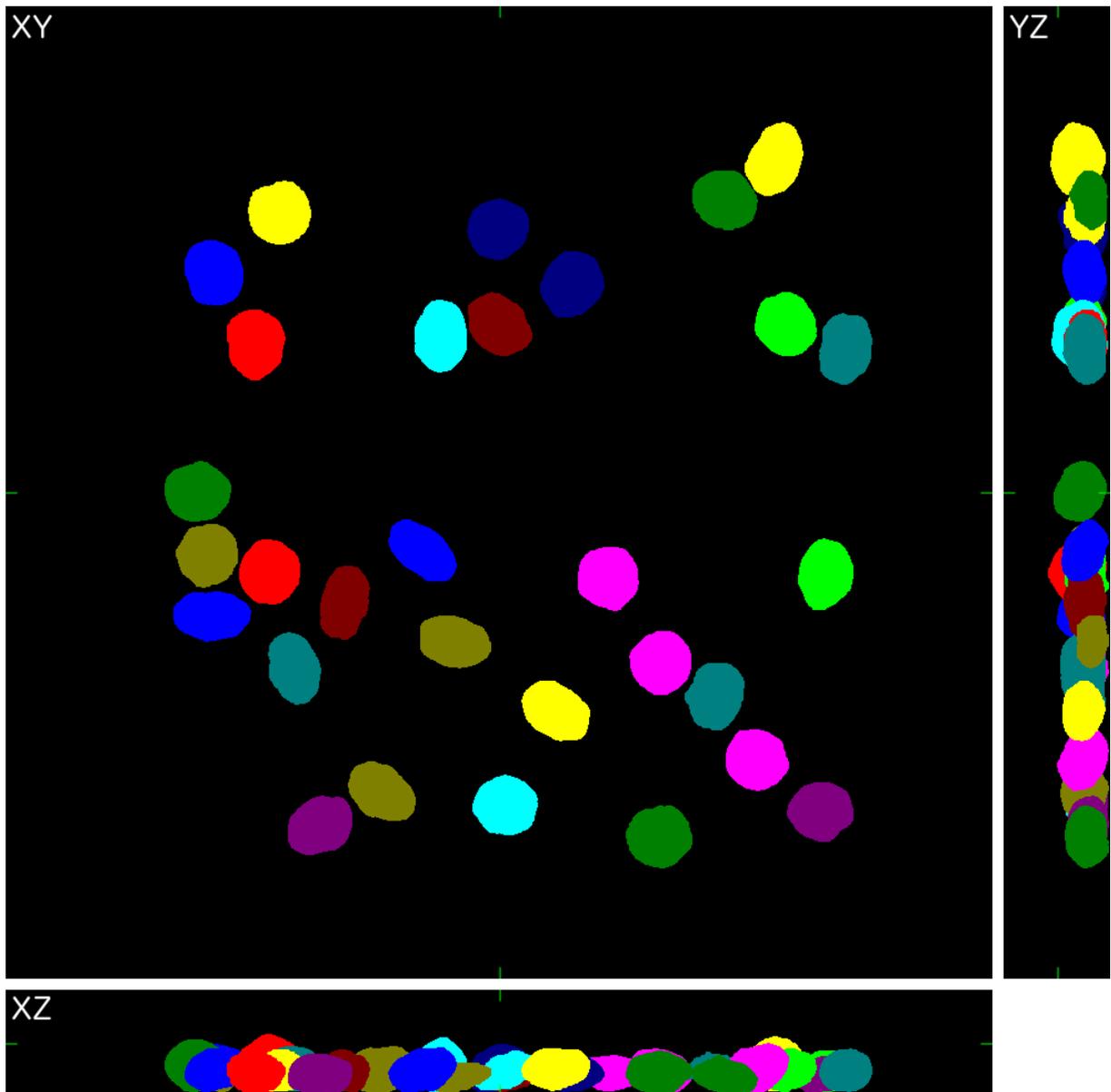


Figure 1314. Example of the initial image (technically a labeled mask) used for one of the Fluo-N3DH-SIM+ competition videos. The image is displayed using maximum intensity (label) projection. Note the presence of an extended boundary to allow cells to move away from the cell population. Cutting away this border introduces the effect that cells are leaving and entering in the final image sequence.

Supplementary Material: Tables

Name Released edition	Objects of interest	Modality/Magnification /Bit depth	Frame size [grid points]	No. of frames	Spatial resolution [μm]	Temporal resolution [min]	No. of tracks	No. of objects
DIC-C2DH-HeLa CTC II	HeLa cells	DIC/63 \times /8	512 \times 512	115	0.19 \times 0.19	10	17 (37)	1366 (1157)
Fluo-C2DL-MSC CTC I	Rat mesenchymal stem cells	SDC/20 \times /16	992 \times 832 (1200 \times 782)	48	0.3 \times 0.3 (0.398 \times 0.398)	20 (30)	20 (15)	657 (258)
Fluo-C3DH-H157 CTC I	H157 lung cancer cells	SDC/63 \times /16	992 \times 832 \times 35 (992 \times 832 \times 80)	60	0.12 \times 0.12 \times 0.5	1 (2)	6 (10)	347 (143)
Fluo-C3DL-MDA231 CTC I	MDA231 human breast carcinoma cells	LSC/20 \times /16	512 \times 512 \times 30	12	1.24 \times 1.24 \times 6	80	46 (43)	452 (394)
Fluo-N2DH-GOWT1 CTC I	Nuclei of GOWT1 mouse stem cells	LSC/63 \times /8	1024 \times 1024	92	0.24 \times 0.24	5	56 (51)	3014 (2418)
Fluo-N2DL-HeLa CTC I	Nuclei of HeLa cells	WF/10 \times /16	1100 \times 700	92	0.645 \times 0.645	30	400 (406)	13836 (16538)
Fluo-N3DH-CE CTC II	Early <i>C. elegans</i> developing embryo	LSC/63 \times /8	712 \times 512 \times 31	190	0.09 \times 0.09 \times 1	1	370 (1167)	10817 (37682)
Fluo-N3DH-CHO CTC I	Nuclei of Chinese hamster ovarian cells	LSC/63 \times /8	512 \times 443 \times 5	92	0.2 \times 0.2 \times 1	9.5	27 (30)	1008 (985)
Fluo-N3DL-DRO CTC III	Developing <i>Drosophila</i> <i>melanogaster</i> embryo	LS/16 \times /16	1272 \times 603 \times 125	50	0.4 \times 0.4 \times 2.03	0.5	203 (417)	9973 (12002)
PhC-C2DH-U373 CTC II	Glioblastoma- astrocytoma U373 cells	PhC/20 \times /8	696 \times 520	115	0.65 \times 0.65	15	7 (11)	531 (679)
PhC-C2DL-PSC CTC II	Pancreatic stem cells	PhC/4 \times /8	720 \times 576	300	1.6 \times 1.6	10	1299 (1404)	70268 (76372)
Fluo-N2DH-SIM+ CTC II	Synthetic nuclei of HL60 cells	WF/40 \times /16	660 \times 718 (664 \times 790)	110 (138)	0.12 \times 0.12	28.8	155 (141)	5463 (4767)
Fluo-N3DH-SIM+ CTC II	Synthetic nuclei of HL60 cells	SDC/40 \times /16	720 \times 728 \times 59 (702 \times 716 \times 59)	150 (110)	0.12 \times 0.12 \times 0.2	28.8	136 (161)	4535 (5410)

Table 1. Technical features of the competition datasets used in the three editions of the Cell Tracking Challenge **Note:** spinning-disk confocal (SDC); laser-scanning confocal (LSC); widefield fluorescence (WF); phase contrast (PhC); differential interference contrast (DIC); light-sheet (LS); The numbers in parentheses indicate particular values for the second video in a given dataset.

Name	Affiliation(s)
Participating editions	
COM-US CTC I	Compunetix Inc. Monroeville, PA, USA
CUL-UK CTC III	Research Centre in Biomedical Engineering, School of Mathematics, Computer Science and Engineering, City University of London, United Kingdom
CUNI-CZ CTC I	Institute of Cellular Biology and Pathology, First Faculty of Medicine, Charles University in Prague, Prague, Czech Republic
FR-Be-GE CTC III	Computer Science Department and BIOSS Centre for Biological Signaling Studies, University of Freiburg, Germany
FR-Ro-GE CTC III	Computer Science Department and BIOSS Centre for Biological Signaling Studies, University of Freiburg, Germany
HD-Har-GE CTC I, II	Biomedical Computer Vision Group, Dept. Bioinformatics and Functional Genomics, BIOQUANT, IPMB, University of Heidelberg and DKFZ, Heidelberg, Germany
HD-Hau-GE CTC III	Heidelberg Collaboratory for Image Processing, IWR, University of Heidelberg, Germany
IMCB-SG (1-2) CTC II	Institute of Molecular and Cell Biology, A*Star, Singapore
KIT-GE CTC III	Group for Automated Image and Data Analysis, Institute for Applied Computer Science, Karlsruhe Institute of Technology, Eggenstein-Leopoldshafen, Germany
KTH-SE (1-4) CTC I, II, III	ACCESS Linnaeus Centre, KTH Royal Institute of Technology, Stockholm, Sweden Baxter Laboratory for Stem Cell Biology, Stanford University School of Medicine, Stanford, USA
LEID-NL CTC I, II, III	Division of Image Processing, Department of Radiology, Leiden University Medical Center, Leiden, the Netherlands Erasmus MC — University Medical Center Rotterdam, Rotterdam, the Netherlands Intelligent Systems Department, Delft University of Technology, Delft, the Netherlands
MU-CZ CTC II	Centre for Biomedical Image Analysis, Masaryk University, Brno, Czech Republic Center for Applied Medical Research, University of Navarra, Pamplona, Spain
NOTT-UK CTC II	Department of Engineering, University of Nottingham, United Kingdom Department of Engineering, University of Nottingham, Ningbo, China
PAST-FR CTC II, III	BioImage Analysis Unit, Institut Pasteur, Paris, France
UP-PT CTC III	I3S - Instituto de Investigação e Inovação em Saúde, Universidade do Porto, Portugal Faculdade de Engenharia, Universidade do Porto, Porto, Portugal Freiburg Institute for Advanced Studies, University of Freiburg, Germany
UPM-ES CTC I	Biomedical Image Technologies, ETSI Telecomunicación, Universidad Politécnica de Madrid and Ciber-BBN, Madrid, Spain
UZH-CH CTC III	S3IT, University of Zurich, Switzerland

Table 2. Cell Tracking Challenge participants across the three challenge editions.

Software Description	Link
COM-US algorithm	
CUL-UK algorithm	
CUNI-CZ algorithm	
FR-Be-GE algorithm	
FR-Ro-GE algorithm	
HD-Har-GE algorithm	
HD-Hau-GE algorithm	
IMCB-SG (1) algorithm	
IMCB-SG (2) algorithm	
KIT-GE algorithm	
KTH-SE algorithm (1)	
KTH-SE algorithm (2)	
KTH-SE algorithm (3)	
KTH-SE algorithm (4)	
LEID-NL algorithm	
MU-CZ algorithm	
NOTT-UK algorithm	
PAST-FR algorithm	
UP-PT algorithm	
UPM-ES algorithm	
UZH-CH algorithm	
Evaluation framework	
Image Parameter evaluation	

Table 3. Available software packages.

Dataset	τ (Kendall)
DIC-C2DH-HeLa	1.00
Fluo-C2DL-MSK	0.76
Fluo-C3DH-H157	0.64
Fluo-C3DL-MDA231	0.50
Fluo-N2DH-GOWT1	0.62
Fluo-N2DL-HeLa	0.49
Fluo-N3DH-CE	1.00
Fluo-N3DH-CHO	0.31
Fluo-N3DL-DRO	0.33
PhC-C2DH-U373	0.60
PhC-C2DL-PSC	0.79
Fluo-N2DH-SIM+	0.74
Fluo-N3DH-SIM+	0.52
Global	0.55

Table 4. Kendall's τ correlation coefficient calculated from the SEG and TRA scores.

Supplementary Material: Cell Tracking Algorithms

This supplementary document provides a description of all cell tracking methods evaluated in this study. In total, 17 teams participated with their own, custom developed methods. Generally speaking, cell tracking methods involve several steps: 1) preprocessing, in which the quality of raw image data is enhanced to facilitate further analysis; 2) cell segmentation (the spatial aspect), in which regions satisfying certain criteria are identified in every frame of the video; 3) cell tracking (the temporal aspect), in which the segmented cells are associated between frames based on another set of criteria to form tracks; and 4) post-processing, which allows better adaptation of a whole image analysis pipeline to a particular dataset; that are either performed sequentially or simultaneously [1-4]. Here, we introduce the principles underlying each step of each participating method. A complete list of parameters required by each method is given **Supplementary Data 4**. Either Windows or Linux command line executable files of the methods are or will be publicly available through the challenge website. They can require other software to be pre-installed on your machine before running them.

COM-US

Authors: Craig Carthel, Stefano Coraluppi

Email: stefano.coraluppi@compunetix.com

Platform: Windows

Prerequisites: None

COM-US: SUMMARY

Our approach includes distinct detection and tracking stages. The detection stage includes image pre-processing (convolution-based low-pass filtering), image segmentation (with a computationally-efficient iterative histogram-processing approach), and object extraction (baricenters of all large-enough contiguous object regions). The automatic tracking employs the multiple-hypothesis tracking paradigm, and includes data association (with small hypothesis-tree depth), sequential track extraction (with feedback to data association processing to favor confirmed tracks), and track post-processing (to identify parent links). Tracking is based on a nearly constant position object motion model. We have not yet introduced feature-aided tracking for these datasets. Further refinement to track post-processing will be required for improved performance.

COM-US: PREPROCESSING

Each 2D or 3D frame is smoothed by image convolution with a unity-valued disk (2D) or ellipsoid (3D) followed by normalization. This removes spurious spikes from the image data.

COM-US: SEGMENTATION

The image segmentation process is motivated by the heavy computational burden associated with processing of the fairly large 2D or 3D images. Thus, we start by forming an intensity histogram for each image. Next, we identify the best-fit N -point approximation to the histogram. This is defined as a set of N intensity values, chosen such that the average displacement from an intensity value to the closest of the N points is minimized. The pixels that are mapped to the highest and lowest of the N values are retained and discarded, respectively. The remaining pixels are then approximated again with an N -point approximation. The procedure continues until the remaining histogram contains N or fewer values. The middle value defines the detection threshold. All pixel values that match or exceed the detection threshold define the object region; all other pixel values define the background region. The object region

leads immediately to object detections, as each set of contiguous pixels defines an object. The barycenter of each contiguous object region is an object detected. Small objects are discarded.

COM-US: TRACKING

The automatic tracker takes as input the sequence of detection sets that result from the detection processing described above. The automatic tracker is based on a *multiple-hypothesis tracking* (MHT) paradigm that we have enhanced over the years, principally for defense surveillance applications; see [5] and references therein. Our approach relies on (1) sequential data-association and (2) track-extraction processing, with some feedback from the track-extraction module to the data-association module as described below. In the data-association module, all detections are accounted for in multiple-hypothesis processing that partitions the set of all detections. Each subset includes at most one detection per time point. Association decisions are based on identifying the *maximum a posteriori* (MAP) global hypothesis with a fixed delay; all competing global hypotheses that conflict with the MAP solution far enough in the past are discarded, while hypothesis diversity in the recent past is maintained. Track-oriented MHT does not require explicit enumeration of global hypotheses. Hypothesis management logic limits the hypothesis space by disallowing sufficiently unlikely associations and considering either a missed detection or an object death hypothesis in the absence of a measurement update. Hypothesis selection relies on local (track) hypothesis scores that in turn, depend on birth and death statistics, detection statistics, and object motion and localization statistics that are used in recursive Kalman filtering. Spurious false alarms are removed in the subsequent track-extraction stage. The processing entails a sliding-window *M-of-N* test that promotes tentative tracks to nearly-confirmed and, ultimately, to confirmed track status. This impacts track scoring in the data-association stage, thus introducing a feedback mechanism that enhances overall tracking performance by favoring nearly-confirmed and confirmed tracks over competing ones. At most K missed detections are allowed before track termination.

COM-US: POST-PROCESSING

Although tracker processing does not rely on the detailed shape characteristics of object detections, it is important for these to be available at the tracker output for **SEG** performance evaluation. Thus, when a detection is available, the object state estimate is replaced by the latest detection. When a detection is not available (i.e., track coast), the object state estimate defines the single pixel that is labeled in the output file. While the single-pixel objects represent a node mismatch with respect to **TRA** processing, it is

necessary to ensure proper individual and parentage link identification in our current processing scheme. Indeed, the final processing stage considers each object track in turn. For each track, we identify the closest potential sibling at the time of object birth. If the potential sibling is close enough, a parent link to the track at the previous time is introduced, if the track exists. Additionally, a parent link between the track at the previous time and the identified sibling replaces the individual link. A more effective track post-processing scheme that avoids the need for single-pixel coast-track objects could be considered as a future improvement.

CUL-UK

Authors: José Alonso Solís-Lemus, Constantino Carlos Reyes-Aldasoro

Email: Constantino.Reyes-Aldasoro.1@city.ac.uk

Platform: Linux

Prerequisites: MATLAB Compiler Runtime 2013a (x64)

CUL-UK: SUMMARY

Our algorithm is based on *PhagoSight*, an open-source MATLAB package for the analysis of fluorescent neutrophil and macrophage migration in a zebrafish model [32]. The package consists of pre-processing, segmentation, tracking and, if necessary, post-processing and visualization steps.

CUL-UK: PREPROCESSING

Pre-processing steps are optional and can be determined by the user. The datasets can be reduced in size with a construction of a Gaussian pyramid where neighboring elements are averaged to form a new pixels, this process reduces the uncertainty of the intensity at the expense of the spatial resolution, and in addition can reduce the computational complexity of large datasets. A 3×3 Gaussian filter was applied to those datasets that were not reduced. The datasets which presented an uneven background intensity were applied a retrospective shading correction algorithm [33].

CUL-UK: SEGMENTATION

Segmentation was performed with a hysteresis thresholding that generated three regions: voxels below a lower threshold (*thresL*) were classified as background, and those above a higher threshold (*thresH*) were classified directly as cells. The remaining voxels between these two levels were then classified as cells if they are in contact with voxels above the higher threshold, or as background otherwise. Both thresholds were automatically determined using Otsu's algorithm. A minimum size value (*minSize*) was selected to discard segmented elements that were small and thus considered as noise. It should be noted that the segmentation algorithm was designed to have an automatic value selection without user input, through an optimization of the threshold and minimum size selection. Those parameters are underlined in **Supplementary Data 4**. For other datasets, we selected the parameters manually through a particle swarm optimization [34].

CUL-UK: TRACKING

The segmented cells were tracked with a model-based tracking algorithm adapted from the keyhole tracking algorithm [35]. The algorithm links the objects in contiguous time points to form the tracks by means of a keyhole model, which predicts the most probable landing position of a cell at time $t+1$ (“child”), from the position in times t (“parent”), and $t-1$ (“grandparent”). The most probable step for a cell that is moving from time $t-1$ to time t , is to follow the direction of the previous steps with the same velocity to time $t+1$. Assuming that a child (cell at time $t+1$) would move with similar direction and velocity as its parent (cell at time t), its landing position can be predicted. To consider for changes in speed, turns or random walk-like movements, two regions of probability where the child cell is most likely to land were defined: a narrow wedge (60° wide) oriented towards the predicted landing position, for straight-moving displacements, and a truncated circle (300°) that complements the wedge, for random-moving displacements, which together resemble a keyhole. The size of the keyhole at $t+1$ is determined by the distance between times $t-1$ and t .

CUL-UK: POST-PROCESSING

Temporal variation of intensity was analyzed as cells that disappear from their tracks and then re-appear a few points later. Collisions of cells were analyzed by measuring the volume of cells in time and splitting cells whose volume increased considerably. Finally, as the lack of proofreading and editing tools has been one of the main barriers in adopting automated and semi-automated methods, *PhagoSight* provide such tools, through which users can evaluate the output of algorithms and correct mistakes that can be visually detected.

CUNI-CZ

Authors: Pavel Křížek, Guy M. Hagen

Email: Pavel.Krizek@lf1.cuni.cz

Platform: Windows

Prerequisites: MATLAB Compiler Runtime 2009a (x86)

CUNI-CZ: SUMMARY

Automated segmentation of objects like cells or cell nuclei and their tracking in time is an important procedure for biological research. For segmentation, we propose a simple method based on the k -means threshold selection algorithm combined with a sliding neighborhood approach. For tracking of segmented objects, we applied a simple method based on frame-by-frame association of nearest neighbors. The algorithm was originally designed for data from our Leica SP5 confocal and Andor spinning disk microscopes and is based on the experience the authors developed in previous projects.

CUNI-CZ: PREPROCESSING

Each raw camera image is slightly blurred by a Gaussian kernel with a user-defined standard deviation σ . We perform this step in order to reduce noise and to smooth out the boundaries of segmented objects. If one z-stack consists of more images, then blurring is performed for each image in the z-stack separately and a maximal intensity projection is used as the final image.

CUNI-CZ: SEGMENTATION

The segmentation algorithm starts with threshold estimation. We applied an iterative threshold selection method based on the k -means algorithm [11] to the blurred camera image. In order to cover large intensity variations of cells and background, the threshold value is appropriately adopted in different parts of the image. We did this using a sliding neighborhood approach with a window size (*box*) given by the user and with a step of a half of the window size. The threshold value is estimated in each of the sliding windows. In this way, we obtain a discrete map of thresholds for different parts of the image. A smooth threshold map is obtained by bi-cubic interpolation of the discrete map.

A binary image with masks of the cells is obtained by thresholding the blurred camera image using the generated smooth threshold map. After thresholding, we apply a fill hole transform [12] on the binary

image to fill regions with holes. Next, to split masks of cells belonging to one connected component (i.e., to separate two touching cells), we perform a watershed transform [13] computed on images obtained by one of the following two strategies. Either morphological opening by reconstruction [14] of the blurred camera image; or morphological opening by reconstruction of a distance transform [15] of the thresholded binary image. The size of the structuring element for the morphological opening operation is a user-defined parameter. Finally, cells are detected as connected components in the final binary image. Only connected components larger than a given minimum area (*minArea*) are taken into account.

CUNI-CZ: TRACKING

The tracking algorithm is based on the nearest neighbor approach. As a reference point for each segmented object, we used its center of gravity. We measured the pair-wise distances between the positions of the objects in the current frame and the positions of the objects in the previous frames. All distances are transformed to probabilities, with higher values corresponding to better position matches. With each object, we also associated a weight w such that $w = 1$ for an object which is still in the field of view, $w = 0$ for an object that divided into more objects so we do not need to track it further, and $w \in (0,1)$ if the object disappears from the field of view. Combining probabilities with weights (*distNorm*) and the user-defined threshold (*distThres*), we can distinguish the following situations:

- Many-to-one case (multiple cells overlap): Object in the current frame is marked with the label of the closest object from the previous frame. For other objects in the neighborhood we follow the one-to-none case.
- One-to-one case (object migrates to a new position): We keep the track of the object and update its position.
- One-to-none case (object disappears from the field of view): The position cannot be updated, but the weight of the object is decreased by 5%.
- None-to-one case (object enters the field of view): A new object is created.
- One-to-many case (object divides into two or more objects): Two or more new objects are created, and for the original object we set the weight to zero.

CUNI-CZ: POST-PROCESSING

No post-processing step is performed.

FR-Be-GE

Authors: Robert Bensch, Olaf Ronneberger

Email: bensch@cs.uni-freiburg.de

Platform: Linux

Prerequisites: MATLAB 2014b (x64)

FR-Be-GE: SUMMARY

Our approach [44] for cell tracking focuses on the segmentation of cells in phase contrast microscopy images. The key feature of our algorithm is that it strongly favors dark-to-bright transitions at the boundaries of the (arbitrarily shaped) segmentation mask, which is effectively found by a fast min-cut approach. The small but essential difference to standard min-cut based approaches is that our graph contains directed edges with asymmetric edge weights. The tracking algorithm applies segmentation propagation to promote temporal consistency. Label propagation is performed by a greedy association of segments between two consecutive frames. We provide an open-source implementation for Matlab on our homepage [43].

FR-Be-GE: PREPROCESSING

Image intensities are normalized to the interval $[0, 1]$ first. Then, images are background corrected by subtracting the smoothed image (large Gaussian kernel with standard deviation σ_{bgr} of 20 pixels) from the original image.

FR-Be-GE: SEGMENTATION

Our segmentation approach exploits the fact that the true cell borders in positive phase contrast microscopy always appear as a dark-to-bright transition in outwards direction. It means that all borders with an inverse transition (bright-to-dark) are definitely not the sought cell borders. We set up a segmentation energy functional for a mask $M:\Omega\rightarrow\{0,1\}$ with $\Omega\subset\mathbb{R}^2$ and the given image $I:\Omega\rightarrow\mathbb{R}$. The functional contains a data cost $C_{\text{obj}}:\mathbb{R}\rightarrow\mathbb{R}$ that depends on the intensity, and an edge cost $C_{\text{edge}}:\mathbb{R}\rightarrow\mathbb{R}$ that depends on the intensity gradient at the mask border in outwards direction

$$E(M) = \lambda \int_{\Omega} M(x) \cdot C_{\text{obj}}(I(x)) dx + \int_{\Omega} C_{\text{edge}}(\langle \nabla M(x), -\nabla I(x) \rangle) dx$$

where we define ∇M to be a unit normal vector on the mask boundary and $\mathbf{0}$ elsewhere. The data cost for a gray value v is derived from the foreground intensity histogram $P(v|O)$ and background intensity histogram $P(v|B)$ from training regions. We define it as $C_{\text{obj}}(v) = (P(v|B) - P(v|O)) / (P(v|O) + P(v|B))$. The

edge cost for the intensity derivative d is computed as

$$C_{\text{edge}}(d) = \begin{cases} \exp\left(-\frac{d^2}{2\sigma^2}\right) & \text{if } d > 0, \\ 1 & \text{otherwise.} \end{cases}$$

This asymmetric edge term favors dark-to-bright transitions at the mask borders. To optimize the energy in (1), we discretize the edge term into eight directions (8 pixel neighborhood) and solve it by a min-cut [45] using the maxflow algorithm MATLAB-interface [46]. The min-cut parameters λ and σ are estimated by the best performing parameters on the training data found by grid-search. Region histograms (with $N_{\text{bins}} = 120$ bins) for computing the data costs in (1) are obtained from manual foreground and background scribbles drawn into a single frame of a training sequence (that is not contained in the segmentation ground truth). We use both training sequences (2 frames in total) for dataset **PhC-C2DH-U373** and only one training sequence (1 frame in total) for dataset **PhC-C2DL-PSC**. Finally, we optionally apply a hole-filling algorithm (*holeFilling*) and discard small segments below a pixel area threshold a_{min} .

FR-Be-GE: TRACKING

Our tracking algorithm consists of two parts. Segmentation propagation promotes temporally consistent segmentation by propagating segmentation information to subsequent frames. Label propagation transfers the label of each segment to subsequent frames using a greedy association.

Segmentation propagation. Min-cut segmentation yields a binary segmentation mask. Segmentation information is propagated from frame t to frame $t + 1$ in two fashions:

- Foreground propagation (FP): The eroded mask is set as hard foreground constraint for the min-cut segmentation in the next frame. This adds robustness to the region term in case of insufficient foreground evidence. The size of erosion must be chosen at least as large as the expected motion of the object boundary pixels between frames. The erosion is computed using a disk-shaped structuring element (radius s_{erosion}).
- Non-merging constraint (NM): If it can be assumed that cells do not merge, it is reasonable to prevent separate objects from merging in the next frame. We achieve this by computing a distance transform on the segmentation mask and applying watershed transform seeded at the object locations. The boundaries of the herewith computed “support regions” of each object are set as hard background constraint.

Label propagation. For propagating labels we use a greedy algorithm. Each segment in frame t propagates its label to the segment in frame $t + 1$ with the highest overlap (measured as intersection over union). If a segment in frame $t + 1$ receives multiple labels, it prefers the segment in frame t with the highest overlap and discards the other labels. If a segment receives no label, a new label is assigned. Additionally, the provided field of interest (FOI) specification is used to discard segments that lie completely outside the FOI (specified by the value E). However, we still use the tracking information from the full view to add parent links in case segments, which are tracked on the full view, reenter the FOI.

FR-Be-GE: POST-PROCESSING

No post-processing is carried out after tracking.

FR-Ro-GE

Authors: Olaf Ronneberger, Robert Bensch, Philipp Fischer, Thomas Brox

Email: ronneber@informatik.uni-freiburg.de

Platform: Linux

Prerequisites: MATLAB 2014b (x64)

FR-Ro-GE: SUMMARY

Our approach for cell tracking is based on a deep convolutional neural network for segmentation and a greedy label propagation for tracking. The convolutional network takes the raw images as input and provides the final segmentation masks as output. The network architecture consists of a contracting path to capture context and a symmetric expanding path that enables precise localization. The network can be trained end-to-end from the few annotated images provided for the training datasets using extensive data augmentation with elastic deformations.

FR-Ro-GE: PREPROCESSING

In all datasets (except for **PhC-C2DH-U373**), the gray values are image-wise normalized (*normInt*) to unit range and zero median to compensate the changes in illumination during the recording. Two datasets (**DIC-C2DH-HeLa** and **Fluo-C2DL-MSK**) are downsampled (*scaleFactor*) to achieve a larger field of view and to compensate different recording resolutions.

FR-Ro-GE: SEGMENTATION

The segmentation is performed by a u-shaped deep convolutional network. It consists of a contracting path with a series of convolution, ReLU and max-pooling layers and an expansion path with a series of up-convolution, ReLU and convolution layers. In the expansion path, feature maps from the contracting path with the same resolution are copied (see [39] for the detailed architecture). The architecture of the network is an extension of the “fully convolutional network” [38]. One important modification in our architecture is that the upsampling part has a large number of feature maps, which allows the network to propagate context information to higher resolution layers. As a consequence, the expansive path is more or less symmetric to the contracting path, and yields the u-shaped architecture. The network does not have any fully connected layers and only uses the valid part of each convolution, i.e., the segmentation map only contains the pixels, for which the full context is available in the input image. This strategy allows the seamless segmentation of arbitrarily large images by an overlap-tile strategy. To

predict the pixels in the border region of the image, the missing context is extrapolated by mirroring the input image. The segmentation of a 512x512 pixel image takes less than a second on a standard laptop (equipped with a NVidia GTX 980m GPU). For the challenge contribution, we averaged the predicted segmentation maps of the input image and its mirrored versions.

Training the Segmentation Network. The loss function for training is computed by a pixel-wise soft-max over the final feature maps combined with a weighted cross entropy loss function [39]. We pre-compute the loss-weight map for each ground truth segmentation map to compensate the unbalanced class-frequency in the training data set, and to force the network to learn the small separation borders, that we introduce between touching cells. The loss-weight map is then computed as

$$w(x) = w_c(x) + w_0 \cdot \exp(-(d_1(x) + d_2(x))^2 / (2\sigma^2)),$$

where w_c is the weight map to balance the class frequencies, $d_1: \Omega \rightarrow \mathbb{R}$ denotes the distance to the border of the nearest cell and $d_2: \Omega \rightarrow \mathbb{R}$ the distance to the border of the second nearest cell. In our experiments, we set $w_0 = 10$ and $\sigma \approx 5$ pixels. The provided manual segmentation masks in the training data set do not cover all visible cells. To obtain a consistent background training set, we manually created “ignore” - regions that cover all unlabeled cells, and set the loss-weights to zero within these regions. In deep networks with many convolutional layers and different paths through the network, a good initialization of the weights is extremely important. Otherwise, parts of the network might give excessive activations, while other parts never contribute. For a network with our architecture (alternating convolution and ReLU layers), a good initialization is achieved by drawing the initial weights from a Gaussian distribution with zero mean and a standard deviation of $\sqrt{2/N}$, where N denotes the number of incoming nodes of one neuron [36]. Data augmentation is essential to teach the network the desired invariance and robustness properties, when only few training samples are available. For microscopic images, we primarily need shift and rotation invariance and robustness to deformations and gray value variations. Especially, random elastic deformations of the training samples seem to be the key concept to train a segmentation network with a very low number of annotated images. We generate smooth deformations using random displacement vectors on a coarse 3 by 3 grid. The displacements are sampled from a Gaussian distribution with 10 pixels standard deviation. Per-pixel displacements are then computed using bicubic interpolation. Gray values of the input images are randomly scaled with a factor drawn from a Gaussian distribution with mean 1 and standard deviation 0.1. Drop-out layers at the end of the contracting path perform further implicit data augmentation. The augmented input images and their corresponding segmentation maps are used to train the network with the stochastic gradient descent

implementation of Caffe [37]. Due to the unpadded convolutions, the input image is larger than the output by a constant border width. To minimize the overhead and make maximum use of the GPU memory, we favor large input tiles over a large batch size and hence reduce the batch to a single image. To compensate for instable gradients, we accordingly set a high momentum (0.99) such that a large number of the previously seen training samples determine the current optimization step. We start the training with an initial learning rate of 0.001 which is decreased by a factor of 10 every 20,000 iterations. After 60,000 iterations (approx. 10 hours on an NVidia Titan GPU) the training is finished.

FR-Ro-GE: TRACKING

For tracking, we use a greedy algorithm. Each segment in frame t propagates its label to that segment in frame $t + 1$ with the highest overlap (measured as intersection over union). If a segment in frame $t + 1$ receives multiple labels, it prefers the segment in frame t with the highest overlap and discards the other labels. If a segment receives no label, a new label is assigned. For the dataset **PhC-C2DH-U373**, we applied two further processing steps that improved the results in our other submission. In each frame small segments below pixel area a_{\min} are discarded. Additionally, the provided field of interest (FOI) specification is used to discard segments that lie completely outside the FOI (specified by the value E). However, we still use the tracking information from the full view to add parent links in case segments, which are tracked in the full view, reenter the FOI.

FR-Ro-GE: POST-PROCESSING

No post-processing is carried out after tracking.

HD-Har-GE^a

Authors: Nathalie Harder, Karl Rohr

Email: k.rohr@dkfz.de

Platform: Linux

Prerequisites: None

HD-Har-GE: SUMMARY

Our approach for cell tracking in different types of 2D and 3D microscopy image sequences combines segmentation and tracking methods. Segmentation comprises filtering for noise reduction, region-adaptive thresholding, and watershed transformation for splitting cell clusters. Tracking is based on local optimization using a cost function within spatio-temporal regions. The tracking algorithm first determines one-to-one correspondences, second, detects mitosis events based on a likelihood measure, and subsequently combines mitotic tracks.

HD-Har-GE: PREPROCESSING

To reduce the image noise, Gaussian filtering (standard deviation σ_F) or median filtering (radius r_{m1}) was applied. In the case of bright speckles within the cell nuclei or strongly varying contrasts of neighboring objects, intensity clipping was performed where the optimal clipping value was determined using Otsu or Renyi entropy thresholding.

HD-Har-GE: SEGMENTATION

After preprocessing, region-adaptive thresholding was applied to obtain an initial segmentation. The approach uses small image regions (radius r_{inner}) and computes local thresholds in enclosing larger regions with a size similar to the average object diameter (r_{outer}). The sensitivity of the approach can be adapted to the image data based on the minimum intensity variance within image regions (σ_{min}^2). Namely, local threshold values are computed if the variance within the respective region exceeds the minimum value, otherwise a global (usually higher) threshold is used. The segmentation result is further enhanced by median filtering (r_{m2}) and hole filling. To split up cell clusters, a watershed transform after Euclidean distance transform is used. This approach is well suited for splitting up clusters of circular or spherical objects, however, for more elongated shapes it tends to result in over-segmentation. Thus, for data sets with elongated objects (e.g., complete cells) we usually skipped this step to avoid wrong splits.

^a In CTC I the method was denoted as HEID-GE.

HD-Har-GE: TRACKING

Our tracking approach exploits the information from cell segmentation and consists of three main steps: (1) Determination of one-to-one correspondences, (2) mitosis detection and establishment of one-to-many correspondences [6], and (3) detection and merging of trajectories that do not cover all frames of an image sequence. In the first step, we use a local optimization procedure, where for each object, hypotheses are generated, namely triplets with all potential predecessors and successors within a limited Euclidean distance (expected maximum displacement d_{\max}). These triplets are ranked and compared with other possible triplets. The rank of a hypothesis is determined based on a cost function, which includes the Euclidean distance, the morphological similarity, as well as the trajectory smoothness. The weights for these three components (w_1, w_2) are specified based on the properties of the image type. In the second step, appearing objects are investigated to detect mitosis events. Depending on the type of staining we exploit different measures (*mitMeasure*) for mitosis detection: The overlap-distance ratio, an object morphology-based likelihood measure, and a combination of both measures. The overlap-distance ratio (ODR) determines the ratio between the area overlap of potential mother and daughter cell objects and their centroid distances [7]. The object morphology-based likelihood measure (Likelih) takes into account the sizes and the mean intensities of potential mother and daughter cells normalized to the average values of the whole population. The different terms can be weighted and adapted according to the image data, for example, to define whether mitotic cells are darker or brighter than the average object intensity value (extension of the likelihood measure for mitosis detection in [8]). The combined measure (Combi) exploits both the overlap-distance ratio and the object morphology-based likelihood measure. If a mitosis event is detected, the trajectories of mother and new daughter cell are merged.

HD-Har-GE: POST-PROCESSING

In the final step, trajectories that do not cover all frames of an image sequence are considered and merged if they are in close spatio-temporal vicinity.

HD-Hau-GE

Authors: Carsten Haubold, Martin Schiegg, David Stöckel, Steffen Wolf, Fred A. Hamprecht

Email: carsten.haubold@iwr.uni-heidelberg.de

Platform: Linux

Prerequisites: None

HD-Hau-GE: SUMMARY

Our tracking-by-assignment approach, *Conservation Tracking* [40], incorporates a pixel-wise classification into foreground and background, followed by connected component extraction to obtain the segmentation. To solve the assignment problem, for each detection we build a graphical model containing random variables, which are connected to possible successors via transition nodes. The most probable configuration is found by running a global optimization. To cope with under-segmentation, a detection is allowed to contain more than one object. Two classifiers are trained to support the optimization in deciding how many cells each detection contains, and which cells possibly divide. The training of all these classifiers was facilitated by the open source software framework *ilastik* [41]. All three classifiers are Random Forests, trained by manual annotation of the Training datasets with interactive feedback in *ilastik*. Only 5 to 10 annotated examples for every case were required to reach the submitted quality.

HD-Hau-GE: PREPROCESSING

No preprocessing is carried out before segmentation.

HD-Hau-GE: SEGMENTATION

To extract detections from the raw data, we interactively train a random forest by sparsely annotating pixels (or voxels), and let it predict probabilities for all pixels indicating whether they belong to the foreground or background class. The annotations are drawn by the user with a brush tool, while the predictions are presented as interactive feedback. The annotations are refined by drawing additional strokes until the predictions yield properly segmented cells or nuclei. The random forest works with a set of features computed from the raw images (e.g., Gaussian gradient magnitude or the structure tensor eigenvalues). We threshold the pixel-wise probability for the foreground class at θ_{thresh} , and extract the connected components to get the segmentation. We also filter detections that are too small and comprise less pixels than s_{min} , or are larger than s_{max} .

HD-Hau-GE: TRACKING

The tracking routine [40] builds a graphical model of the detections and the possible transitions. Conservation Tracking is not restricted to tracking cells or nuclei, but can track arbitrary dividing or non-dividing objects. It allows objects to merge, to split after a merge, and to divide. To be able to distinguish the very similar events of splitting and dividing, it uses two classifiers: one for object count per detection, and one for divisions. The first classifier is trained to detect how many cells or nuclei are combined into a segment. The division classifier learns, based on a mother cell and two children candidates in the next time frame, which features indicate a division. The classifiers are trained in a similarly interactive fashion as the pixel classification mentioned above: a user assigns the correct class to some detections in the GUI, and the current predictions for all other detections are presented as colored overlay over the data. By annotating more detections, the user can improve the classifiers to his/her satisfaction. The optimization procedure uses those classifiers as a guide, but can contradict them to conserve objects in the global temporal context. The classifier influence can be controlled using parameters as described below.

Graphical Model and Global Optimization. As mentioned before, we build a probabilistic graphical model and add random variables for all detections and possible transitions. Every detection can contain between 0 and m objects. Each of these configurations denotes a state in the optimization, and has an energy value attached. This energy is the negative logarithm of the probability for this state as it was predicted by the object count classifier. It can be adjusted with the factor w_{det} (we use the same notation for parameters as in the original paper [40]). The energy of an appearing cell can be controlled via w_{app} , and disappearances by w_{van} . The energy decreases linearly from a distance of d_{border} to the border. Cells that have their center closer than d_{crop} to their border, are discarded. A transition node is created for every detection-pair in consecutive temporal frames, if the Euclidean distance d between their centers is less than d_{max} pixels. Transition probabilities are defined relative to the Euclidean distance d as

$$p_{\text{tr}}(d) = \begin{cases} \exp\left(-\frac{d}{\alpha}\right) & \text{if this transition is active,} \\ 1 - \exp\left(-\frac{d}{\alpha}\right) & \text{otherwise.} \end{cases} \quad (1)$$

Transition utility energies are thus $-w_{\text{tr}} \cdot \log(p_{\text{tr}}(d))$. Each detection node in the graph that is connected to two outgoing transition nodes, meaning there are two possible fates in $t + 1$, could possibly divide. Then,

we add an extra random variable that captures whether this division is active. The negative logarithm of the division probability predicted by the classifier is used as energy there, scaled by w_{div} . The optimal parameters for tracking (w_{app} , w_{van} , w_{div} , w_{det} , and w_{tr}) were obtained through structured learning on the training datasets, the other parameters were found by a grid search. We find the tracking solution with minimal energy by transforming the graph described above to an integer linear program with indicator variables for each detection-, transition- and division hypothesis. We integrate conservation constraints, and optimize using the commercial CPLEX solver [42].

HD-Hau-GE: POST-PROCESSING

For the **PhC-C2DL-PSC** dataset, our segmentation was trained such that it separates most objects, and thus the detections did not cover the complete cells in the raw data. We applied a dilation operation to the segmentation with a structured disc element of radius $r_{\text{dilate}} = 2$. We did not dilate the results of any other dataset.

IMCB-SG

Authors: Pengdong Xiao, Weimiao Yu

Email: xiaopd@hotmail.com

Platform: Windows

Prerequisites: MATLAB Compiler Runtime 2012a (x64)

IMCB-SG: SUMMARY

We developed two different algorithms for two different categories of datasets, namely two-dimensional spatio-temporal (2D+t) and three dimensional spatio-temporal (3D+t). For 2D+t case, we further classified the data sets into fluorescent and phase contrast types. For 3D+t case, we treat low signal to noise ratio (SNR) and highly non-uniform intensity data sets separately from normal data sets while conducting segmentation. It is performed using a seed controlled watershed approach. Tracking for 2D+t is realized by directly tracing the seeds; and tracking for 3D+t is conducted by frame-to-frame data association according to the minimization of a cost function using the Hungarian algorithm.

IMCB-SG (1): 2D+t datasets

For 2D+t cell tracking, we use an approach of performing tracking first then followed by segmentation.

Tracking. First, a Gaussian filter is used to smooth the images in spatial domain for two purposes. One is to remove noise and the other is to highlight the region of the interested objects. Window size and standard deviation are two parameters of the Gaussian filter that are set based on the structure of cells or nuclei in the images. Morphology operations are used to further remove background noise. For low contrast images, we use histogram equalization and contrast adjustment techniques to enhance the images. Second, we align the 2D images into 3D stacks such that the third dimension represents time. After the alignment, we use a convolution based Gaussian filter to smooth the images along time. The purpose we perform this step is to recover missing objects caused by low temporal resolution, weak or corrupted signal, or fast movement of cells or nuclei. After the first two steps, we use extended maxima transform to locate seeds based on image gradient. The extended maxima transform is a local maxima searching algorithm which can have different image features as input. We have tried both the distance

function based on obtained binary images and image gradient as input, and the result shows that image gradient performs better on most data sets. In the last step of the tracking, the trace of the above-mentioned 2D seeds forms 3D segments which are our sought tracking trajectories. We apply a morphological opening operation to remove some small 3D segments that are considered as noise or artifacts caused by the processing. Tracking is realized by tracing the trajectories of the remaining 3D segments. Cell movement and mitotic events are all incorporated in the trajectories.

Segmentation. In the segmentation stage, a thresholding technique is applied to obtain binary images for each 2D time frame. A global image threshold value by Otsu's method is used to conduct thresholding. After thresholding, we use morphological opening to remove spurious objects. Based on the binary images obtained, we combine 2D seeds in each time frame and image gradient after Gaussian smoothing in both spatial and temporal domains to use the watershed algorithm to obtain final segmentation. The main purpose of using the seed controlled watershed algorithm is to solve the touching cell segmentation problem. In addition, for phase contrast images we use a linear imaging model to restore artifact free images by removing halos and shade-off from phase contrast images [28]. After the restoration, the artifact free images are used as input of our algorithms and the rest of the processing for segmentation and tracking is the same.

IMCB-SG (2): 3D+t datasets

For 3D+t data sets, we perform segmentation first and then follow by tracking.

Segmentation. Segmentation is performed differently for normal datasets and for low signal to noise ratio (SNR) datasets with highly non-uniform intensity. During data preparation stage, isotropic 3D image stacks based on linear interpolation are prepared first before segmentation. Severe artifacts appear in data sets with low SNR and highly non-uniform intensity. Therefore, the proposed segmentation is performed separately for normal data sets and artifact data sets. For normal data sets, we apply a 3D Gaussian kernel to smooth the images and parameters are set according to the size of cells or nuclei. After that, we generate a histogram to determine a threshold value and apply thresholding to obtain 3D binary images. Morphological opening is used to remove small objects from the binary images. Furthermore, we use distance transform to compute Euclidean distance. By using the distance function as input, we apply extended maxima transform to find seeds in 3D. According to the seeds and the binary images obtained, the watershed algorithm is used to perform final segmentation. For data sets

with low SNR and highly non-uniform intensity, we first use linear interpolation to down-sample image stacks. The main reason for conducting this step is that the following processing steps are very computationally expensive with original resolution. With reduced image resolution, we can perform Gaussian smoothing in 3D spatial domain efficiently. At the same time, we can also perform background removal by using morphological operation and contrast adjustment to enhance the images. After these preprocessing steps, we interpolate the down-sampled images back to their original size. Then, we conduct thresholding to obtain binary images, and use distance function or image intensity as input to detect seeds by using extended maxima transform. Final segmentation is realized by using a seed controlled watershed algorithm based on distance function and image intensity.

Tracking. For tracking of 3D+t data sets, we have used two approaches to tackling the problem. The first approach is a nearest neighbor search based algorithm. We first define a region corresponding to a cell in the first time frame. According to the number of potential objects in the region of the next time frame, we determine whether it is a migration or mitosis. A recursive tracing technique is used to construct lineage trees of cell mitoses. The second approach is based on global matching using the Hungarian algorithm, which combines both Euclidean distance and overlapping volume. We deal with mapping cases of one-to-one, one-to-many, and many-to-one. For one-to-one case, we consider it as a migration. If it is a many-to-one case, we choose one with the least cost and the rest are considered as termination in this time frame. One-to-many case is considered as a mitotic event and if potential matches are bigger than two then new cell emerging situation is considered.

KIT-GE

Authors: Johannes Stegmaier and Ralf Mikut

Email: johannes.stegmaier@kit.edu

Platform: Windows

Prerequisites: MATLAB Compiler Runtime 2014a (x64)

KIT-GE: SUMMARY

Our framework is based on the TWANG segmentation algorithm as described in [25, 26]. In particular, the algorithm performs a seeded segmentation of the provided image data that is capable of extracting fluorescently labeled objects from 2D or 3D images reliably and fast. Temporal associations were identified using a straightforward nearest neighbor matching as implemented in our open-source MATLAB toolbox Gait-CAD [27].

KIT-GE: PREPROCESSING

Noise reduction of the input images was performed using a 2D median filter (*medianRad*) and to smooth the segmentation results, an additional Gaussian filtering was applied (*gaussVar*). In the case of 3D images the median filtering was performed individually for each of the slices. For the **Fluo-N2DH-GOWT1** dataset an additional morphological closing was used to avoid holes in the segmentation (*closingRad*). Using a down-sampled version of the input image (width and height scaled by 0.25, depth unchanged for 3D images), seed points were detected by identifying local maxima in the 8-neighborhood (2D)/26-neighborhood (3D) of each pixel within a Laplacian-of-Gaussian (LoG) space-scale maximum projection, which was iteratively calculated using LoG filtered images of different discrete scales (*LoGMin*, *LoGMax*). We used a non-strict maximum detection to avoid misdetections caused by intensity plateaus and merged redundant seed points based on a minimum expected distance criterion. At each identified location, the mean intensity of a 7x7 window was calculated and used to discard low intensity seed points with a semi-automatically optimized binary threshold (*seedThres*).

KIT-GE: SEGMENTATION

The actual segmentation of the spherical objects was performed on each image individually using the TWANG segmentation algorithm [25]. For every detected seed point, a cube with side lengths proportional to the radius of the respective blob was cropped from the preprocessed images and the regions were processed in parallel [26]. With the goal of a fast approximate segmentation of (hyper-) spherical objects, the cropped image regions were transformed to a representation that could be segmented by a simple adaptive thresholding. Therefore, a new image was formed based on a Gaussian weighted dot product of the seed point normal (a normalized direction pointing away from the seed point) with the normalized intensity gradient vector at that each pixel (pointing in the direction of the steepest intensity change) as described in [25] (*gradStd*, *kernelSizeMult*, and *kernelStd*). In this transformed image, the transition regions between individual nuclei obtained low intensity values, whereas pixels belonging to the currently considered nucleus obtained high intensity values that could be easily separated from the background using an adaptive binary threshold (Otsu's method).

KIT-GE: TRACKING

The identification of temporal associations of the detected nuclei was derived with the tracking toolbox contained in the open-source MATLAB toolbox Gait-CAD [27]. Essentially, the centroids of identified segments were tracked by identifying nearest neighbors in subsequent frames. Matches were only considered as valid if maximum distance was not exceeded (*maxDist*). If the distance ratio of the closest and the second closest nearest neighbor was sufficiently small, the nucleus was considered a potential cell division candidate (*neighDistRatio*). Post processing routines for cell division detection and the fusion of fragmented tracks, however, are still under development and were disabled for the submitted tracking results. The tracking results were subsequently linked back to the segmentation images, i.e., the intensity values of all segmented regions within each image were set to the assigned tracking ID.

KIT-GE: POST-PROCESSING

No post-processing is carried out after tracking.

KTH-SE

Authors: Klas E. G. Magnusson, Joakim Jaldén, Helen M. Blau

Email: klasma@kth.se

Platform: Windows

Prerequisites: MATLAB 2012b (x64) or higher

KTH-SE: SUMMARY

We have used a tracking by detection framework with three separate segmentation algorithms and a track linking algorithm based on the Viterbi algorithm. For the **Fluo-N3DL-DRO** dataset, we also used a detection pre-processing algorithm based on GM-PHD filtering, which allows us to use dynamic motion models in the track linking step. When possible, we used a search algorithm to optimize the segmentation parameters, but in some cases we got better results by optimizing the parameters manually in a graphical user interface. Automatically optimized parameter values are underlined in the text below as well as in the parameter configurations listed in **Supplementary Data 4**.

KTH-SE: PREPROCESSING

All images were converted to 64-bit double images with a saturation intensity of 1.

KTH-SE: SEGMENTATION

We used three different segmentation algorithms to generate the binary segmentation masks, which we then post-processed to extract cell regions. When possible, we used a search algorithm to optimize the segmentation parameters, but in some cases we got better results by optimizing the parameters manually in a graphical user interface. Automatically optimized parameter values are underlined in the descriptions below.

Bandpass segmentation. To segment all of the **Fluo** datasets and **PhC-C2DL-PSC**, we used the bandpass filtering based segmentation algorithm that we presented in [47]. We performed the filtering by convolving the original image I with two different Gaussian filters G_S and G_B , with covariance matrices

$\Sigma_S = \sigma_S^2 \Sigma$ and $\Sigma_B = \sigma_B^2 \Sigma$. In two dimensions, Σ is the 2×2 identity matrix, and in three dimensions $\Sigma = \text{diag}(1, 1, 1/r^2)$, where r is the ratio between the voxel height and the voxel width. The two filtered images are given by $I_S = I * G_S$ and $I_B = I * G_B$, and the bandpass filtered image is computed as $I_{BP} = I_S - \alpha I_B$, where α is a free parameter. The binary segmentation mask is obtained by applying the threshold τ to I_{BP} . To avoid under-segmentation of dim objects that are close to bright objects, we preprocessed some of the datasets using intensity clipping, where all pixel values above I_{\max} are set to I_{\max} . In **Fluo-C2DL-MS** there was a bright background region in the upper part of the image, which was not removed entirely by the bandpass filter. To deal with this, we computed a background image by taking the minimum intensity for each pixel position, through the time dimension of the image sequence. We then subtracted this background image. In **Fluo-N3DH-CE** the noise properties are different in the different image dimensions, and therefore we used a $5 \times 1 \times 3$ median filter to reduce the noise before we applied the bandpass filter. In **Fluo-N3DH-CE** and **PhC-C2DL-PSC**, we used different values for σ_S and σ_B for the first and the last image of the sequence, and used linear interpolation to compute different values for each image in between.

Variance segmentation. To segment the cells in **PhC-C2DH-U373**, we computed a texture image representing the intensity variance in a region around each pixel in the original image. This technique has been used previously to segment cells in transmission microscopy images [48, 49]. But instead of just computing the variance in a neighborhood of each pixel, we weighted the surrounding pixels using a Gaussian kernel G with covariance matrix $\Sigma_{\text{var}} = \sigma_{\text{var}}^2 \mathbf{I}_2$, where \mathbf{I}_2 is the 2×2 identity matrix. We computed the weighted local variance image V as $(G * I^2) / (G * \mathbf{1}) - (G * I)^2 / (G * \mathbf{1})^2$, where I^2 is an image with the squared pixel intensities and $\mathbf{1}$ is an image with all ones. The obtained variance image was thresholded using a threshold τ_{var} to give a binary segmentation mask. We used the parameter values $\sigma_{\text{var}} = \underline{1.88}$ and $\tau_{\text{var}} = \underline{5.57E-5}$. To get rid of background features, we first subtracted a background image. The background image was computed as the median intensity for each pixel position, taken over the time dimension of the sequence.

Ridge segmentation. To segment cells in **DIC-C2DH-HeLa**, we developed an algorithm inspired by the algorithm used to segment muscle fibers in [50]. We first applied a ridge detection filter similar to the filter described in [50], to highlight the boundaries between the cells. The ridge detection was done by smoothing the image with Gaussian kernels with standard deviations σ of 5, 6, 7, 8, 9, and 10 pixels and

computing the Hessian at each pixel of the six resulting images. The ridge image $v_0(\sigma)$ at the scale σ was then computed from the eigenvalues λ_1 and λ_2 , where $\lambda_1 \leq \lambda_2$, of the corresponding Hessians as

$$v_0(\sigma) = \begin{cases} 0 & \text{if } \lambda_1 > 0, \\ \exp\left(\frac{-R_B}{\gamma^2}\right) \left(1 - \exp\left(\frac{-S}{\beta^2}\right)\right) & \text{otherwise,} \end{cases}$$

where $R_B = |\lambda_2|/|\lambda_1|$ and $S = (\lambda_1)^2 + (\lambda_2)^2$. We used $\gamma=1$ and $\beta=10$. The final ridge image was obtained by taking the pixel-wise maximum of $v_0(\sigma)$ over all σ and smoothing using a Gaussian filter with a standard deviation of 1 pixel. Once we had the ridge image, we transformed the intensities using the function $f(x) = \text{asinh}(20x)$, to enhance dim ridges, and divided by the mean intensity of the transformed image. Then, we thresholded the ridge image at $T=0.75$, and skeletonized the resulting binary mask to extract cell boundaries. To determine which of the resulting regions were cells and which were background, we computed a local variance image where each pixel value represented the sample variance in a 9×9 neighborhood of the corresponding pixel in the original image. Regions with an average local variance above 0.0005 were considered to be cell regions. To fill in gaps in the skeletonized boundaries, we detected all end points of the skeleton and connected pairs of them by straight lines. End points were connected if they were no more than 50 pixels apart, and if the added line cut through a single segment, without generating a fragment smaller than 7500 pixels. If one of the new regions would become a background region, the size threshold was instead set to 200 pixels, as the operation would not split a cell in two. After joining end points, we removed cracks in regions by erasing all boundary pixels, which were bordering a single region. Then we merged the background regions and the border pixels into a single background region. Finally, we merged cell regions with less than 7500 pixels into adjacent cell regions until all cell regions either had at least 7500 pixels or were surrounded by background pixels.

Post-processing. To break regions with multiple cells into individual cell regions, we applied a seeded watershed transform (*watersheds*) to the image intensity (wI), the bandpass filtered image (wB), or the distance transform (wS) of the binary segmentation mask. The pixel values in the distance transform are the Euclidean distances to the closest background pixels. For z-stacks, where the voxel height was different from the voxel width, we used the anisotropic distance transform [51], where the distance between z-planes is different from the distance between neighboring voxels in the same plane. In **Fluo-N3DH-CE** and **Fluo-N3DL-DRO**, this did however give poor separation boundaries between the watersheds, as the distance between z-planes was too large. To avoid these problems, we inserted virtual z-planes between adjacent z-planes in the distance transform. We assigned values to the virtual z-planes using linear interpolation, ran the watershed transform and then removed the virtual planes. We

used nine virtual z-planes for **Fluo-N3DH-CE** and two for **Fluo-N3DL-DRO**. For all datasets, the watershed transform was constrained to the foreground pixels of the binary segmentation mask, to speed up the computation, and to avoid getting watersheds, which overlap with multiple cell regions. To avoid over-segmentation, we applied Gaussian smoothing with a standard deviation of σ_w , and/or an h -minima transform with an h -value of H_{\min} . In **Fluo-N2DH-GOWT1**, we also removed watershed seeds with a distance transform value below 10 pixels, to further reduce over-segmentation. In **Fluo-C2DL-MS** and **PhC-C2DL-PSC**, we applied an additional watershed transform, after the first one, to break even more clusters into individual cells. To get rid of regions without cells, we removed regions with fewer than A_{\min} voxels, and regions where the summed voxel intensity was below S_{\min} . To compute the summed voxel intensity, we subtracted the minimum value of the image, and summed all voxels inside the segmented region. In **Fluo-N3DL-DRO**, we also removed regions larger than 10000 voxels. For some datasets, we applied morphological operators to the extracted cell regions. We filled in holes in the segments of all datasets. In the **Fluo-N2DH-SIM+-02** image sequence and in **FluoN3DH-SIM+**, we added all pixels inside the convex hulls of the original regions. Whenever a pixel was in the convex hull of multiple regions, we did not add it to any of them. In **Fluo-N2DH-GOWT1** there were also pieces missing from the segments, but the true regions were not always convex, so to fill in missing parts, we instead applied morphological closing with a circular structuring element with a radius of 12.2 pixels. The variance-based segmentation of **PhC-C2DH-U373** tends to give too large regions, due to the large kernel size used to compute the variance. To overcome this problem, we applied morphological erosion with a circular structuring element with a radius of 8.31 pixels. In **DIC-C2DH-HeLa** there was quite a lot of over-segmentation, but in many cases over-segmented regions were correctly segmented in adjacent images. We therefore tried to reduce the over-segmentation by looking for cases where multiple cells overlapped with the same region in an adjacent image. If the fragments were smaller than 15000 pixels and had at least 60 % of their pixels in common with the region in the adjacent image, they were merged into a single region.

Parameter optimization. For many of the datasets we used an automated search algorithm to optimize the segmentation parameters. The search algorithm used a type of coordinate ascent with variable step length to optimize the individual parameters one at a time. The parameters were initialized using manual tweaking, and the step lengths were set to 10 % of the initial values. In each optimization iteration of the optimization, the algorithm goes through the parameters one at a time and tries both increasing and decreasing them by the corresponding step lengths. The parameters are adjusted to the best value if either of the options gives a better result. If a better segmentation is found, the step length is increased

by 20 % and otherwise it is decreased by 20 %. We used **SEG** as utility function for the optimization and ran it for 25 iterations. For **Fluo-C2DL-MSC**, **Fluo-N2DH-SIM+**, and **Fluo-N3DH-SIM+**, the parameters were optimized separately for each image sequence, but for all other datasets, the optimization was performed over all image sequences simultaneously, on the average **SEG**.

KTH-SE: TRACKING

For all datasets except **Fluo-N3DL-DRO**, we applied our global track linking algorithm [49] directly to the detected cell regions. For **Fluo-N3DL-DRO** we used a newly developed detection preprocessing algorithm [52], which takes advantage of the dynamic nature of the nuclei motion by preprocessing the detected locations using a Gaussian Mixture Probability Hypothesis Density (GM-PHD) filter [53]. Once we had preprocessed the locations, we linked them using the track linking algorithm presented in [49].

Global track linking. Our track linking algorithm is global in the sense that it considers all images of the sequence simultaneously when tracks are generated. The algorithm optimizes a probabilistically motivated scoring function by iteratively adding cell tracks to the image sequence. This is done by constructing a state space diagram representing all possible ways in which an additional cell track can be added to the image sequence [49]. The arcs of the state space diagram have scores associated with them, so that we can find the track that increases the scoring function the most by finding the highest scoring path through the state space diagram. Given that the state space diagram is a trellis graph, the highest scoring path can be found by solving a shortest path problem using the Viterbi algorithm. To prevent incorrectly created tracks from blocking the creation of correct tracks in subsequent iterations, the preexisting tracks can be edited using so called swap operations, when new tracks are created [49]. The scoring function is a sum of logarithmic probabilities of tracking events, which describe migration, mitosis, appearance, disappearance, and the number of cells in each detection. The probabilities of migration events are computed as described in [49], using a Brownian motion model where the location of a cell in one image is assumed to follow a Gaussian distribution with covariance matrix $\sigma_V^2 \Sigma$, centered around the location of the cell in the previous image. We used the same Σ as in *Bandpass Segmentation*, except for **Fluo-N3DH-CE**, where we used $\Sigma = \text{diag}(1, 1, 1/(4r)^2)$, as there was significantly less motion in the z-dimension than in the other dimensions. The values for σ_V were set manually for all datasets. In **Fluo-N3DH-CE**, we used different values for σ_V for the first and the last image of the sequence, and used linear interpolation to compute a different value for each image in between. The prior probabilities that the segmented regions contain zero, one, or more than one cell are denoted p_0 , p_1 , and p_2 . The

probability that a cell undergoes mitosis in a region is denoted p_s , and the probability that a cell appears or disappears randomly in a region is denoted p_A . All of these probabilities were set manually. Once the Viterbi algorithm has finalized generating tracks, the segmented regions with multiple cells are separated using k-means clustering of the pixel coordinates as described in [49], so that each cell gets a region of its own. Then the track linking is updated, to account for the new centroid positions of the individual cells, by solving an assignment problem that maximizes the scoring function. For the image sequences which have $FP="yes"$, we included segmented regions in the results even if the track linking algorithm found them to be false positives. This was done to maximize the **TRA** and **SEG** measures, which penalize false negatives more than false positives.

Global track linking with detection preprocessing. The cells in **Fluo-N3DL-DRO** form a tissue which deforms as the embryo develops. Because of this, the nuclei follow smooth and predictable trajectories. The track linking procedure described in the previous section assumes that the nuclei follow Brownian motion, and can therefore not take the velocities of the nuclei into account when it predicts where they are going to be in the next frame. To enable tracking of fast moving nuclei, we therefore used the algorithm described in [52]. It first runs a GM-PHD filter on the centroids of the nuclei and then links the Gaussian components (which include velocity states) of the computed hypothesis densities into tracks using the track linking algorithm in [49]. For the GM-PHD we used the directed linear motion model previously used by us to track simulated microtubules in [52], with a scale factor $q = 0.5$ for the process noise, and an observation noise covariance of $R = 4\Sigma$. For the remaining parameters described in [52] we used the following values: $p_s = 0.9999$, $p_D = 0.999$, $\kappa = 4E-6$, $w_{\min} = 0.001$, $KLD_{\min} = 1$, $J_{\max} = 10000$, and $\sigma_v = 2$. We first tracked all the nuclei using the algorithm described above, and then we selected the tracks that overlapped with one of the manually marked nervous system nuclei in the first image. For nervous system nuclei, which had no overlapping tracks, we selected the closest non-overlapping track. We also had the a priori information that all of the selected tracks should reach the end of the video. We therefore extended broken selected tracks by linking them to fragments of unselected tracks. This was done by propagating the state of the broken track to the frame after the break, using the directed linear motion model, and then linking it to the closest unselected track in that frame. This was done iteratively until all selected tracks reached the end of the image sequence. The extension of broken tracks was expected to improve the results significantly as the rate of false positive tracks was found to be very low on the training data before this step. The performance increases on the training data was however small. Tracking all cells is computationally more demanding than tracking just the cells that are counted

towards the final score, but we wanted to take the opportunity to push the boundaries of our tracking solution using the **Fluo-N3DL-DRO** dataset.

KTH-SE: POST-PROCESSING

To remove segmentation errors that were due to over-segmentation in the watershed transforms, we iteratively merged region fragments without cells into adjacent regions with cells, after the tracking had been completed. We took this idea one step further in the image sequence **FluoC2DL-MS-01**, where we also merged region fragments without cells in image t , into cells with which they overlapped in one of the images $t - 3$, $t - 2$, $t - 1$, $t + 1$, $t + 2$, and $t + 3$. The fragments were merged with the regions of the cells in image t , provided that the cells were present in that image. The merging was done iteratively until no more fragments could be merged.

LEID-NL

Authors: Oleh Dzyubachyk, Erik Meijering

Email: o.m.dzyubachyk@gmail.com

Platform: Windows

Prerequisites: MATLAB Compiler Runtime 2012a (x64)

LEID-NL: SUMMARY

The developed method is based on the model-evolution approach, where the segmentation and tracking tasks are performed simultaneously. The level sets are chosen as the model due to their ability to provide high-quality segmentation on different types of images and natural ability to handle topological changes. In general, the framework follows that described in our earlier publication [10]. The algorithm was tailored to each particular data sequence by selecting a suitable set of parameters. In addition, several additional features were implemented and used in different combinations for each particular data sequence. For example, one of such additional built-in routines helps overcoming the situation when the level set segments two just divided nuclei as one object. Also special attention has been paid to segmenting the nuclei entering the field of view, which were detected by producing a rough estimate of the image foreground at each time step. Several methods, including various thresholds and other fast binary segmentation methods, were added to the algorithm for this purpose. Selection of the most appropriate binary segmentation method for the initial binary segmentation and detection of nuclei entering the field of view was performed empirically.

LEID-NL: PREPROCESSING

No data preprocessing is performed.

LEID-NL: SEGMENTATION AND TRACKING

The presented cell tracking and segmentation algorithm is based on the method described in our earlier publication [10]. Within this approach, segmentation and tracking are encapsulated into a single task by the model-evolution approach. More precisely, each object being tracked is represented by a model, which is fitted to the data at every time point. During such fitting, information of different nature, like image data or regularity information, can be combined into a single energy functional, which is subsequently minimized via the gradient descent method. Level-sets were chosen as the model for their ability to provide high-quality segmentations for wide range of microscopy data and natural topology change, which is important for handling cell division events.

In the current version of the algorithm, several modifications have been introduced in comparison with the method presented in [10]. The most important modifications are described below (see also “Post-processing”).

Separation of touching cells. In our method, the position of the optimal separation plane between two touching cells is detected based on the minimum of the Radon transform of the intensity image in the local window containing both objects. For large objects, this operation can be rather computationally expensive. To make the separation more efficient, in this implementation of our method, we reduced the space of possible locations of the dividing plane by forcing it to pass through a preselected point. This point is calculated as the one corresponding to the intensity minimum on the segment connecting the markers of the objects being separated.

Initial segmentation and detection of entering cells. In our method, both the initial segmentation and the detection of objects entering the field of view, is performed based on a rough estimation of the image foreground. This is achieved by binary segmentation of the image, which can be done in many different ways. To improve the flexibility of the method, we made this step dependent on the type of the data being processed. Thus, we have extended our algorithm with a wide range of binary segmentation methods, in particular various thresholds. The most appropriate method for each data sequence is selected empirically.

Inter-scan cell motion compensation. The model-evolution approach, that our method is based on, assumes sufficient overlap between the two regions occupied by an object at every two consecutive time points. However, in some cases, for example when the objects are relatively small and fast moving, and/or the temporal resolution is low, this condition is violated. In order to compensate for the inter-scan motion, we have developed a scheme that estimates the new position of each object at the next time point based on its current position. This operation is performed before the evolution of each contour, and consists of two main steps: 1) Marker detection, and 2) Joint estimation of the new position of all the objects. In a similar way, we deal with the cases when in the data sets with DNA staining, e.g. **Fluo-N2DL-HeLa**, two daughter cells move far away from the final position of the mother cell. For detection of such events, we supply our algorithm with different prior information, in particular, that the daughter cells typically move in the direction perpendicular to the major axis of the mother cell.

LEID-NL: POST-PROCESSING

Even though level-sets can naturally change their topology, splitting or not-splitting of the contour in each particular case depends on several factors, in particular image data and energy weights. In some cases, too strong regularization can preclude the contour from splitting. This happens either in the cases when the energy minimum corresponds to the situation when one level-set engulfs two proximate objects, or when the desired energy minimum is not reached due to the contour evolution getting stuck in a local energy minimum. In cell segmentation, such cases are quite often encountered when two daughters do not move far apart from each other. To better handle such cases, we have extended our algorithm with an additional post-processing step. The level-sets that might be engulfing two separate cells after the convergence of the contour evolution are detected by low solidity, defined as the ratio of the total volume to the volume of the convex hull of their shapes. Subsequently, an additional round of contour evolution is performed on these objects with relaxed regularity constraints, starting from the current state. The contours that do split into two disjoint parts as result of such additional contour evolution are labeled as belonging to divided objects, otherwise the old contour state, before the additional evolution round, is kept.

MU-CZ

Authors: Martin Maška, Arrate Muñoz-Barrutia, Carlos Ortiz-de-Solórzano, Michal Kozubek

Email: xmaska@fi.muni.cz

Platform: Windows (x64)

Prerequisites: None

MU-CZ: SUMMARY

The method follows the tracking by model evolution paradigm. It is based on the minimization the Chan-Vese model in a fast level set-like framework that is integrated with a topological prior allowing simultaneous tracking of multiple cells over time.

MU-CZ: PREPROCESSING

To reduce the amount of noise in the analyzed 2D and 3D time-lapse sequences, every single frame was preprocessed using a Gaussian filter with the standard deviation of one pixel in each direction.

MU-CZ: SEGMENTATION AND TRACKING

The method used for analyzing some of the competition datasets is built on our previously published cell tracking scheme [18, 19]. It follows the tracking by model evolution paradigm, in which the segmentation and tracking steps are solved simultaneously, exploiting final results of individual frames as initial conditions for the analysis of the following frames. The approach is based on the minimization of the Chan-Vese model in a fast level set-like framework that is integrated with a topological prior allowing simultaneous tracking of multiple cells over time. In comparison to the published approach [18, 19], several modifications have been introduced to deal with specific features of the analyzed competition datasets. They are described in more detail in the rest of this section.

Clustered cells in the first frame. One of the main limitations of the original tracking scheme is its inability to correctly track cells being clustered in the first frame, calling for manual interaction with the user. To fully automatize the whole process, a cluster separation routine based on the evolution of topologically inflexible implicit active contours has been employed [20].

Capturing entering cells. The original approach assumes that new cells entering the field of view are border components disjoint with the existing cells. However, due to low temporal resolution and various experimental setups, entering cells do not necessarily touch the image border when they first appear in

the field of view in some of the analyzed competition datasets. Therefore, the condition for selecting entering cell candidates has been slightly modified omitting the necessity of being border components in a binary mask obtained using a weighted 2-means clustering.

Compensation for time-variant fluorescence intensity. In the analyzed simulated datasets, we have often observed a phenomenon of time-variant average fluorescence intensity within individual nuclei. This especially holds shortly before and after division events, when the amounts of fluorescently stained DNA materials dramatically change. To partly compensate for this phenomenon, the background weight in the Chan-Vese model (λ_2) is temporarily multiplied by a user-defined constant (*mult*).

MU-CZ: POST-PROCESSING

Based on the instructions from the challenge organizers, all tracked objects outside a specified field of interest (*borderSize*) are systematically discarded from the tracking results, not to be penalized during the TRA measure evaluation.

NOTT-UK

Authors: Yuexiang Li, Siu-Yeung Cho, John Crowe

Email: yuexiang.li@szu.edu.cn

Platform: Windows 7

Prerequisites: None

NOTT-UK: SUMMARY

This algorithm is designed to automatically track the changes in cell shape and position in a time-lapse video. The segmentation is based on local thresholding and has an excellent performance in low quality images. To track the detected cells in different frames, a frame-by-frame association using Euclidean distance as criterion is implemented.

NOTT-UK: PREPROCESSING

Due to the low quality conditions of the cell images, an existing enhancement method based on morphological operations [21], Top-Hat and Bottom-Hat, is performed before segmentation. The method can enhance the image contrast, which leads to better segmentation outcomes.

NOTT-UK: SEGMENTATION

Our segmentation algorithm is designed using local thresholding technique. The enhanced cell image will be firstly separated into sub-images. Then, the method will detect if there are cells or parts of cells in each of the sub-images by comparing their standard deviations with the one from the whole image. If there are objects detected in the processing sub-image, the method will perform the same enhancement approach used in the pre-processing step to deeply improve image quality. Followed that, Otsu thresholding technique is used to segment the sub-image. After all of them are processed, the outcomes will be placed together to generate the result. For some of the databases, such as **Fluo-N2DH-GOWT1**, even with enhancements, there are still cells in fragmentation. In that case, the proposed method detects the fragmented cells by calculating their roundness. If this parameter is lower than a pre-set value, the detecting area will be further enhanced and segmented to regenerate the shape of the cell being processed. Differently from other databases, the segmentation approach for **Fluo-C2DL-MSC** is established on the global thresholding technique. Because the gray scale distribution of the images in this database is relatively uniform compared to the others, it is more computationally efficient to use

global thresholding, instead of the local one. The pre-processing step is the same for this database. After that, Otsu thresholding is used to each of the frames to generate segmentation results.

NOTT-UK: TRACKING

Cell tracking is achieved by a frame-by-frame association. A segmented time-lapse sequence, which cells are all detected in each frame, is the input for the proposed approach. Center of gravity of the cell is chosen to represent cell's position. Hence, the method is designed to firstly calculate the centers of cells in the processing frame and the ones in the previous frame. Then, for every cell in the current frame, the approach calculates the Euclidean distances from the cell's center to the ones in the previous frame, and links the nearest ones to it. Obviously, this method may lead to many issues, such as one-to-many correspondence, in experiments. Therefore, related judgment criteria are designed for it. After cell linking in neighbor frames, if two or more cells are pointing to the same one in the previous frame, a division of the cell is deemed to happen. New labels are assigned to each of the cells in the processing image (one-to-many). In addition, if the minimum distance exceeds a pre-set threshold, which is the cell radius in our experiments, the processing cell is regarded as a new one appeared in the current frame and marked with new label (none-to-one). Conversely, cells in the previous frame will be marked as disappeared, if there is no cell in the current frame linked to them (one-to-none). Apart from frame-by-frame tracking, for the 3D database, like **Fluo-N3DH-CHO**, links between different layers are also needed. To achieve that, our proposed method starts with segmentations of the different layers. Then, it sums up the outcomes into one image, and performs cell tracking with the neighbor frame. According to the obtained labeled graph, the method finally marks the cells in different layers with corresponding labels.

NOTT-UK: POST-PROCESSING

No post-processing step is performed.

PAST-FR

Authors: Alexandre Dufour, Jean-Christophe Olivo-Marin

Email: adufour@pasteur.fr

Platform: Linux

Prerequisites: None

PAST-FR: SUMMARY

Our approach to cell tracking is based on the theory of deformable models. It permits a precise analysis of the cell shape, in addition to keeping cell identity and determining its center of mass. Each cell contour is pre-detected automatically in the first image of each video, then optimized by minimizing an energy-functional composed of various terms related either to the image (homogeneous regions, image gradients, etc.) and regularization terms (contour smoothing, shape constraints, etc.). This process is then repeated for every subsequent frame, while the pre-detection step is re-used to detect objects entering the field of view. Therefore, our method does require that a minimal overlap exists between the successive positions of each cell. While active contours are known to be computationally demanding, we implement a discrete formalism, whereby the contour is defined as a closed polygonal line to enable fast computations and small memory footprint [29, 30].

PAST-FR: PREPROCESSING

First, the high intensity variations in the image data are reduced by capping the lower and upper $n\%$ pixel values. Second, median filtering within a neighborhood of radius rM is performed.

PAST-FR: SEGMENTATION AND TRACKING

The analysis of each frame involves three steps. First, the cells are pre-detected using a hierarchical k-means approach [31] that involves smoothing using a Gaussian filter with a large radius rG , quantization of image intensities into 10 classes, and extraction of connected components in each class in the ascending order, the size of which falls within a pre-defined size range $[aMin, aMax]$. Next, a contour of spatial sampling s for each previously unknown cell is initialized. Finally, all contours are deformed according to image gradients (weighted by wg), image homogeneity (weighted by wh) and regularization (weighted by wr), while division events are automatically detected and tracked when a contour splits.

PAST-FR: POST-PROCESSING

No post-processing step is performed.

UP-PT

Authors: Tiago Esteves, Maja Temerinac-Ott, Pedro Quelhas

Email: dee11017@fe.up.pt

Platform: Windows

Prerequisites: MATLAB Compiler Runtime 2012b (x86)

UP-PT: SUMMARY

We track cells in 2D and 3D microscopy image sequences based on Laplacian of Gaussians (LoG) local detection and detection-association tracking. The detection-association is based on a Euclidean distance nearest neighbor search for detections between consecutive frames. The cells in all data modalities are detected using LoG filtering which enhances the image's blob like structure which corresponds to cell locations [23]. To improve the cell's shape estimation after Log detection we resort to automatic image threshold or to the Sliding Band Filter (SBF) as better suited to the data modality under analysis [22]. The SBF is a local convergence filter, capturing edge convergence on a band around a certain location, which can also detect blob like structure, but which can adapt to a wider range of shapes than the LoG filter.

UP-PT: PREPROCESSING

We rescale each image (*imScale*) for speed purposes. We reduce image noise by applying a Gaussian filter with a specific standard deviation (*LOGsigma*).

UP-PT: SEGMENTATION

We perform detection using scale non-maxima suppression, where we vary the scale of the LoG filter between the expected range of the cell radius (*Rmin*, *Rmax*) [22, 23]. In the first frame, we specify the cells to track by considering local maxima from the filter response above a specific threshold (*initDetectTH*). In the remaining frames, we consider all the detections with a filter response above a lower threshold (*detectTH*) to ensure the detection of all the cells being tracked. Given all detections, we then perform cell shape estimation based on a local convergence filter (SBF option) [22, 23] or based on image thresholding (Otsu option) specifying the method to use in the parameter *getCellShape*. For SBF parametrization, we set the range of filter scales to be the same as in the LoG (*Rmin* and *Rmax*), setting the bandwidth of the filter (*q*) and the number of orientations (*N*) to values verified to be adequate to the data [22, 23]. The SBF is applied in the image only to the locations of LoG detection for a better cell shape estimation for each detection. When a good shape estimation can be obtained using automatic

thresholding, we apply Otsu's method, in which we decide if we want to smooth the image again or not (*filterShape*) previous to image thresholding. For each LoG detection, we select a region (ROI) around the specific cell location where the size of this region is given by multiplying the estimated cell size by a specific factor (*windowSize*). Finally, within the selected ROI we focus on the bigger segmented region (in terms of area) and we get the boundary information, which corresponds to the boundary estimate for the detected cell. In case of 3D data, the detection and shape estimation is performed in all the slices of each data volume that has image entropy higher than a predefined value (*entropyE*). Detections that appear near the same location (x,y) within z-axis are considered imaging slices of the same cell. Cells that are detected in less than a certain number of slices (*minSlices*) are discarded. During the process of saving the results, we also have the possibility of increasing the estimated cell size by dilating *sizeComp* times the segmentation result.

UP-PT: TRACKING

Our tracking is based on a detection-association approach [24]. We perform the association of the closest detections in consecutive frames based on the Euclidean distance. If the detection does not have a neighbor within a minimum distance ($4 * \text{detection_radius}$), we stop tracking it.

UP-PT: POST-PROCESSING

Based on the detection-association result, we get the final cell tracking information by performing the next steps: merging of incomplete tracks that are in close spatio-temporal vicinity; removal of tracks that have less than four frames; merging each track that does not start in the first frame with the closest track that starts in a previous time point (mitosis event). For the **Fluo-N3DL-DRO** dataset, we only consider tracks that start from the first frame and we remove all the others.

UPM-ES

Authors: David Pastor-Escuredo, Daniel Jiménez-Carretero, María J. Ledesma-Carbayo

Email: david.pas@gmail.com

Platform: Linux, Windows

Prerequisites: MATLAB Compiler Runtime 2008a (x64)

UPM-ES: SUMMARY

We propose an algorithm exploiting general and robust features of cell and cell nuclei data of fluorescence time-lapse microscopy. To achieve that we apply spatio-temporal mathematical morphology [16], [17] to exploit the characteristic size range of cells and the redundancy and coherence of trajectories at the same time, just by working with spatio-temporal structuring elements (SEs) in the extended domain –particularly to 2D+t challenge datasets. With this strategy, filtering, segmentation and tracking are intrinsically coupled. Filtering uses the spatial information through grayscale area openings that limit the range of characteristic sizes within the image. On the other hand, the temporal consistency of the trajectories is exploited using a novel proposed stochastic spatio-temporal morphological reconstruction that enhances coherent structures through time to improve the “trajectory-to-noise” ratio. After this filtering, a simple threshold is enough to perform an initial segmentation that is refined through a “merge or split” filtering with binary dilations and erosions that allow us to keep the characteristic size range. Trajectories are found by labeling the binary image in 2D+t with spatio-temporal connectivity. Each trajectory is identified with a label making it possible to analyze different spatial frames of the image to detect divisions and check that the topology is consistent. By isolating each label we can impose constraints and generate the graph. This is an effective method to perform tracking, keeping coherence of trajectories.

UPM-ES: PREPROCESSING

The characteristic noise of microscopy images is a highly non-linear salt and pepper like noise. We apply a non-linear median filter of radius r_M to reduce this effect. The main point of this work is to use spatio-temporal coherence of data to improve the image quality to perform the tracking. Spatial characterization is exploited using a used gray-scale spatial area opening of radius r_{AO} to remove small flat regions that could generate over-segmentation. The temporal coherence is exploited with a novel strategy: “stochastic spatio-temporal morphological reconstruction”. This process provides a “trajectory-

to-noise” ratio enhancement as coherent spatio-temporal structures are highlighted over those that do not keep that kind of coherence such as noise:

- Extract regional maxima (already pruned with the residue)
- Perform 2D+t morphological reconstruction having the regional maxima of one frame as markers for the reconstruction
- Sum up the reconstructed images from each step in the 2D+t domain

The resulting images weight the spatio-temporal connectivity of the structures in the image. Thus, trajectories are enhanced over noise. This method is especially effective for images with high temporal resolution or featuring smooth migration dynamics.

UPM-ES: SEGMENTATION

After the trajectory-to-noise ratio enhancement, we simply apply thresholding to binarize obtaining the subsequent candidate segmentations by finding connected components in 2D. Threshold T is set at the level that provides a more concentrated histogram of segmentations areas (in 2D). The initial segmentation is polished with a morphological closing of radius r_C and filling holes. Then, we perform a step of “merge or split” using binary dilation and erosion as a hierarchical clustering according to the size. A maximum size ($MSize$) is set for objects to be eroded r_{EP} and a minimum size ($mSize$) for objects to be dilated r_{DP} , a residue size ($rSize$) is used for a final binary area opening to avoid over-detection, so segmentations keep the size constrains in the data while keeping intrinsic consistency temporally.

UPM-ES: TRACKING

As explained, both filtering and segmentation are performed intrinsically contributing to the tracking, as alternative to decoupled segmentation and tracking. We use spatio-temporal SEs to find and label connected components in 2D+t as trajectories in the segmented image after erosion for safe labeling r_{SL} . Labels are re-dilated r_{SL} and isolated by thresholding being able to check the 2D objects in each frame and refine the trajectories to create the graph:

- No object found:
 - If no previous detection, the trajectory has not started yet
 - If trajectory was registered before, then it is over and we close the graph
- One object found:
 - If no registered before, the graph branch starts here
 - If registered, the trajectory keeps on and we update the branch

- More than one object found. We perform another 2D+t labeling forward from the current time point. This is done to re-evaluate the topology of the trajectory for the current point:
 - If one trajectory (2D+t object) is detected, it means that it reconnects after this step, so it is not robust to annotate mitosis and we update the branch.
 - If more than one trajectory is detected, it means that indeed there is a division and new branches are opened in the graph. We update the initial 2D+t label images with the new labels got by the forward labeling and keep on the loop.

For now, the method cannot identify over division detection. On the other side, the method is really robust to not generate cross-links or consider false positive detections generating undesired links in the graph. In general, this approach is a straightforward and effective way to perform cell tracking with datasets with a suitable time-step that keeps spatio-temporal coherence of cell trajectories. The extension to 3D+t depends only on computation capabilities and efficient implementation of the algorithm.

UPM-ES: POST-PROCESSING

No post-processing step is performed.

UZH-CH

Author: Ömer Demirel

Email: oemer.demirel@uzh.ch

Platform: Linux

Prerequisites: MATLAB R2014b (x64)

UZH-CH: SUMMARY

Our cell tracking algorithm consists of two traditional steps: Segmentation and tracking. In segmentation, we combine watershed segmentation with Otsu thresholding. Later, the identified objects are tracked in a frame-by-frame fashion. There, two consecutive frames are taken into account and the objects are matched to each other via a modified nearest-neighbor algorithm. No post-processing step is executed.

UZH-CH: PREPROCESSING

To increase the signal-to-noise ratio (SNR), we first adjust the contrast of the image by mapping the intensity values in the input image frame to new values in the modified image such that 1% of data is saturated at low and high intensities. Following that, we blur the resulting image with a Gaussian filter of width w . Lastly, we compute the mean intensity and subtract that from the image. Where negative intensities occur, we take their absolute values.

UZH-CH: SEGMENTATION

Our segmentation is based on seeded watershed segmentation. To find the seed locations, the image is first complemented and its negative distance transform is calculated. Using this, we compute a seed mask by applying an extended-minima transform with the depth level of h . This gives us regional minima, which are connected components of pixels that have constant intensity. These minima are used as seed points for individual watershed basins. We use this mask to start the watershed transform and obtain the segmented image. After that, we apply a downscaled Otsu thresholding with the downscaling factor of d . To clean up artifacts, we neglect objects that are smaller than 10 pixels and fill segmented objects to avoid holes in them.

UZH-CH: TRACKING

Our tracking is based on a nearest-neighbor search algorithm where the nearest neighbors are chosen in a spatiotemporal manner (i.e., closest objects in the next time frame). We compute all-to-all distances

between the objects in frame t and $t+1$. A maximum distance (*maxDist*) thresholding of 50 pixels is applied and for each object O_t found in t , we get a first set of possible candidates in $t+1$. Here, we limit the maximum number of neighbors (*maxNeigh*) to five in 2D. Depending on the number of candidates and their relative sizes compared to the object, we formulate three main scenarios: (i) disappearing, (ii) self-tracking, and (iii) division. If no spatiotemporal neighbors are found or if a single candidate is found, but it is more than twice the size of O_t , then the object is disappearing. If only a single candidate with an acceptable size (i.e., less than $2 \cdot \text{object size}$), is linked to O_t , we call this case self-tracking. Whenever we are left with more than two possible candidates, we create all possible two-pairings and check which pair's mean centroid is closest to the O_t . If this closest pair's total size is not larger than twice the size of O_t , we accept this cell division.

UZH-CH: POST-PROCESSING

No post-processing is carried out after tracking.

REFERENCES

1. Zimmer C, Zhang B , Dufour A, Thébaud A, Berlemont S, Meas-Yedid V, Olivo-Marin J-C. On the digital trail of mobile cells. *IEEE Signal Processing Magazine* **23**, 54-62 (2006).
2. Meijering E, Dzyubachyk O, Smal I, van Cappellen WA. Tracking in cell and developmental biology. *Seminars in Cell and Developmental Biology* **20**, 894-902 (2009).
3. Rohr K, Godinez WJ, Harder N, Wörz S, Mattes J, Tvaruskó W, Eils R. Tracking and quantitative analysis of dynamic movements of cells and particles. *Cold Spring Harbor Protocols* **2010**, pdb.top80 (2010).
4. Meijering E, Dzyubachyk O, Smal I. Methods for cell and particle tracking. *Methods in Enzymology* **504**, 183-200 (2012).
5. Coraluppi S, Carthel C. Modified scoring in multiple-hypothesis tracking, *ISIF Journal of Advances in Information Fusion* **7**, 153-164 (2012).
6. Harder N, Mora-Bermúdez F, Godinez WJ, Ellenberg J, Eils R, Rohr K. Automated analysis of the mitotic phases of human cells in 3D fluorescence microscopy image sequences. In *Proceedings of the 9th International Conference on Medical Image Computing and Computer-Assisted Intervention*, 840-848 (2006).
7. Withers JA, Robbins KA. Tracking cell splits and merges. In *Proceedings of IEEE Southwest Symposium on Image Analysis and Interpretation*, 117-122 (1996).
8. Harder N, Mora-Bermúdez F, Godinez WJ, Wünsche A, Eils R, Ellenberg J, Rohr K. Automatic analysis of dividing cells in live cell movies to detect mitotic delays and correlate phenotypes in time. *Genome Research* **19**, 2113–2124 (2009).
9. Magnusson KEG, Jaldén J. A batch algorithm using iterative application of the Viterbi algorithm to track cells and construct cell lineages. In *Proceedings of the 9th IEEE International Symposium on Biomedical Imaging*, 382–385 (2012).
10. Dzyubachyk O, van Cappellen WA, Essers J, Niessen WJ, Meijering E. Advanced level-set based cell tracking in time-lapse fluorescence microscopy. *IEEE Transactions on Medical Imaging* **29**, 852–867 (2010).
11. Šonka M, Hlaváč V, Boyle R. *Image Processing, Analysis, and Machine Vision*. London, U.K.: Chapman & Hall, 1999.
12. Soille P. *Morphological Image Analysis: Principles and Applications*, Berlin, Germany: Springer-Verlag, 2003.
13. Meyer F. Topographic distance and watershed lines. *Signal Processing* **38**, 113-125 (1994).

14. Vincent L. Morphological grayscale reconstruction in image analysis: Applications and efficient algorithms. *IEEE Transactions on Image Processing* **2**, 176-201 (1993)
15. Breu H, Gil J, Kirkpatrick D, Werman M. Linear time Euclidean distance transform algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **17**, 529-533 (1995).
16. Pastor-Escuredo D, Luengo-Oroz MA, Duloquin L, Lombardot B, Ladesma-Carbayo MJ, Bourguine P, Peyrieras N, Santos A. Spatio-temporal filtering with morphological operators for robust cell migration estimation in "in-vivo" images. In *Proceedings of the 9th IEEE International Symposium on Biomedical Imaging*, 1312-1315 (2012).
17. Luengo-Oroz MA, Pastor-Escuredo D, Castro-Gonzalez C, Faure E, Savy T, Lombardot B, Rubio-Guivernau JL, Duloquin L, Ledesma-Carbayo MJ, Bourguine P, Peyrieras N, Santos A. 3D+t morphological processing: Applications to embryogenesis image analysis. *IEEE Transactions on Image Processing* **21**, 3518-3530 (2012).
18. Maška M, Daněk O, Garasa S, Rouzaut A, Muñoz-Barrutia A, Ortiz-de-Solórzano C. Segmentation and shape tracking of whole fluorescent cells based on the Chan-Vese model. *IEEE Transactions on Medical Imaging* **32**, 995-1006 (2013).
19. Maška M, Muñoz-Barrutia A, Ortiz-de-Solórzano C. Fast tracking of fluorescent cells based on the Chan-Vese model. In *Proceedings of the 9th IEEE International Symposium on Biomedical Imaging*, 1316-1319 (2012).
20. Maška M, Daněk O, Ortiz-de-Solórzano C, Muñoz-Barrutia A, Kozubek M, García IF. A two-phase segmentation of cell nuclei using fast level set-like algorithms. In *Proceedings of the 16th Scandinavian Conference on Image Analysis*, 390-399 (2009).
21. Ye D, Zhao Y, Li D. The study of local contrast enhancement for MR images. 2005. p. 6-7.
22. Esteves T, Quelhas P, Mendonça AM, Campilho A. Gradient convergence filters and a phase congruency approach for in vivo cell nuclei detection. *Machine Vision and Applications* **23**, 623-638 (2012).
23. Esteves T, Oliveira MJ, Quelhas P. Cancer cell detection and morphology analysis based on local interest point detectors. In *Proceedings of the 6th Iberian Conference on Pattern Recognition and Image Analysis*, 624-631 (2013).
24. Esteves T, Oliveira MJ, Quelhas P. Cancer cell detection and tracking based on local interest point detectors. In *Proceedings of the 10th International Conference on Image Analysis and Recognition*, 434-441 (2013).

25. Stegmaier J, Otte J, Kobitski A, Bartschat A, Garcia A, Nienhaus GU, Strähle U, Mikut R. Fast segmentation of stained nuclei in terabyte-scale, time resolved 3D microscopy image stacks. *PLoS ONE* **9**, e90036 (2014).
26. Kobitski A, Otte JC, Takamiya M, Schäfer B, Mertes J, Stegmaier J, Rastegar S, Rindone F, Hartmann V, Stotzka R, García A, Wezel JV, Mikut R, Strähle U, Nienhaus GU. An ensemble-averaged, cell density-based digital model of zebrafish embryo development derived from light-sheet microscopy data with single-cell resolution. *Scientific Reports* **5**, 1–10 (2015).
27. Stegmaier J, Alshut R, Reischl M, Mikut R. Information fusion of image analysis, video object tracking, and data mining of biological images using the open source MATLAB toolbox Gait-CAD. *Biomedizinische Technik (Biomedical Engineering)* **57**, 458–461 (2012).
28. Yin Z, Kanade T, Chen M. Understanding the phase contrast optics to restore artifact-free microscopy images for segmentation. *Medical Image Analysis* **16**, 1047–1062 (2012).
29. Zimmer C, Olivo-Marin JC. Coupled parametric active contours, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **27**, 1838-1842 (2005).
30. Dufour A, Thibeaux R, Labruyère E, Guillén N, Olivo-Marin JC. 3D active meshes: Fast discrete deformable models for cell tracking in 3D time-lapse microscopy. *IEEE Transactions on Image Processing* **20**, 1925-1937 (2011).
31. Dufour A, Meas-Yedid V, Grassard A, Olivo-Marin JC. Automated quantification of cell endocytosis using active contours and wavelets, In *Proceedings of the 19th International Conference on Pattern Recognition* (2008).
32. Henry KM, Pase L, Ramos-Lopez CF, Lieschke GJ, Renshaw SA, Reyes-Aldasoro CC. PhagoSight: an open-source MATLAB package for the analysis of fluorescent neutrophil and macrophage migration in a zebrafish model. *PLoS ONE* **8**, e72636 (2013).
33. Reyes-Aldasoro CC. Retrospective shading correction algorithm based on signal envelope estimation, *Electronics Letters* **45**, 454-456 (2009).
34. Kennedy J, Eberhart R. Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks*, 1942–1948 (1995).
35. Reyes-Aldasoro CC, Akerman S, Tozer GM. Measuring the velocity of fluorescently labelled red blood cells with a keyhole tracking algorithm. *Journal of Microscopy* **229**, 162–173 (2008).
36. He K, Zhang X, Ren S, Sun J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *arXiv [cs.CV]*, page arXiv:1502.01852, 2015.

37. Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, Guadarrama S, Darrell T. Caffe: Convolutional architecture for fast feature embedding. arXiv preprint arXiv:1408.5093, 2014.
38. Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. arXiv [cs.CV], page arXiv:1411.4038, 2014.
39. Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. In *Proceedings of Medical Image Computing and Computer-Assisted Intervention*, 234-241(2015).
40. Schiegg M, Hanslovsky P, Kausler BX, Hufnagel L, Hamprecht FA. Conservation tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, 2928-2935 (2013).
41. Sommer C, Strähle C, Köthe U, Hamprecht FA. Ilastik: Interactive learning and segmentation toolkit. In *Proceedings of the 8th IEEE International Symposium on Biomedical Imaging*, 230-233 (2011).
42. <http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/>
43. <http://lmb.informatik.uni-freiburg.de/resources/opensource/CellTracking/>
44. Bensch R, Ronneberger O. Cell segmentation and tracking in phase contrast images using graph cut with asymmetric boundary costs. In *Proceedings of the 12th IEEE International Symposium on Biomedical Imaging*, 1220-1223 (2015).
45. Boykov Y, Funka-Lea G. Graph cuts and efficient n-d image segmentation. *International Journal of Computer Vision* **70**, 109–131 (2006).
46. Boykov Y, Kolmogorov V. An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26**, 1124–1137 (2004).
47. Maška M, Ulman V, Svoboda D, Matula P, Matula P, Ederra C, Urbiola A, Espana T, Venkatesan S, Balak DMW, Karas P, Bolcková T, Štreitová M, Carthel C, Coraluppi S, Harder N, Rohr K, Magnusson KEG, Jaldén J, Blau HM, Dzyubachyk O, Křížek P, Hagen GM, Pastor-Escuredo D, Jimenez-Carretero D, Ledesma-Carbayo MJ, Munoz-Barrutia A, Meijering E, Kozubek M, Ortiz-de-Solórzano C. A benchmark for comparison of cell tracking algorithms. *Bioinformatics* **30**, 1609–1617 (2014).
48. Wu K, Gauthier D, Levine MD. Live cell image segmentation. *IEEE Transactions on Biomedical Engineering* **42**, 1–12 (1995).
49. Magnusson KEG, Jaldén J, Gilbert PM, Blau HM. Global linking of cell tracks using the Viterbi algorithm. *IEEE Transactions on Medical Imaging* **34**, 1–19 (2015).
50. Sertel O, Dogdas B, Chiu CS, Gurcan MN. Microscopic image analysis for quantitative characterization of muscle fiber type composition. *Computerized Medical Imaging and Graphics* **35**, 616–628 (2011).

51. Mishchenko Y. A fast algorithm for computation of discrete Euclidean distance transform in three or more dimensions on vector processing architectures. *Signal, Image and Video Processing* **9**, 1–9 (2012).
52. Magnusson KEG, Jaldén J. Tracking of non-Brownian particles using the Viterbi algorithm. In *Proceedings of the 12th IEEE International Symposium on Biomedical Imaging*, 380-384 (2015).
53. Vo BN, Ma WK. The Gaussian mixture probability hypothesis density filter. *IEEE Transactions on Signal Processing* **54**, 4091-4104 (2006).