UNIVERSITÉ PARIS DIDEROT
LABORATOIRE MATIÈRE ET SYSTÈMES COMPLEXES

GlobWatch - simple image segmentation tool for microfluidic image stacks

Nataša Puzović

February 25, 2019

Future users - please feel free to contact me at `natasa.puzovic@gmail.com` for any questions and unclarities.

GLOBWATCH_BLURFILTER - BLURRED IMAGE FILTER

Input: Raw TIF stacks of BF and RFP images as they are stored by the microscope camera.
Output:

- Stacks of sharp RFP channel images
- plots of blurriness measures of each frame in each stack
- Tab-separated table ("Blurred_Frames_table.txt") with 3 columns: ImageStackName; FrameInStack; blurriness (0 not blurred, 1 blurred).

Scripts in folder:

- GlobWatch_BlurFilter.m
  Main script. Adds paths and makes an empty folder called "Denoised_and_filtered_images". Writes table with the result of the blur detection called "Blurred_Frames_table.txt" at the end.
- Read_TIF_stack_composite.m
  Reads in a composite TIF stack of a timelapse experiment in which both BF and RFP images are taken once every 6min. Images from the two channels are alternatingly stored by the microscope (one BF, one RFP, one BF, etc.). They are read into MATLAB as a 3D matrix (Dimensions: ImageWidth x ImageHeight x Number of image in stack)

and BF and RFP images are split by taking the odd-numbered frames as a BF stack, and even-numbered frames as a RFP stack.

- Segment.m
  Reads in frame by frame, performing noise-reduction by substracting a disk-structured image from the raw image, segments denoised images using Otsu's method, and saves both denoised and segmented image stacks as 3D matrices.

- fmeasure.m
  Function to measure the relative degree of focus of an image, taken from a paper benchmarking 37 blur detection algorithms (S. Pertuz et al., Analysis of focus measure operators for shape-from-focus. Pattern Recognition, 46(5):1415:1432, 2013.)

- Detect_blurred_images.m
  Calculates the focus measure using the fmeasure function on each frame in the stack, then finds peaks of high focus measure values (high focus measure = blurred image) and removes them from the stack, the peak finding and removal process is done until no more blurred frames remain. For each image stack a plot of focus measure per frame is generated with detected blurred frames marked with asterisks.

- Output_segm_denoised_images.m
  Writes segmented and/or denoised images that are not blurred into a single TIF stack called "segmented_filtered_stack_%s.tiff" and/or "denoised_filtered_stack_%s.tiff" (s = name of the original image stack) into the "Denoised_and_filtered_images" folder.

GLOBWATCH - SEGMENTATION

Input: Raw TIF stacks of BF and RFP images as they are stored by the microscope camera.
Output:

- Stacks of sharp RFP channel images
- plots of blurriness measures of each frame in each stack
- Tab-separated table (Blurred_Frames_table.txt) with 3 columns: ImageStackName; FrameInStack; blurriness (0 not blurred, 1 blurred).

Scripts in folder:

- GlobWatch.m
  Main script. Adds paths and makes an empty folder called "Denoised_and_filtered_images". Writes table with the result of the blur detection called "Blurred_Frames_table.txt" after blur detection step. Calculates generation time

- Read_TIF_stack_composite.m
  Reads in a composite TIF stack of a timelapse experiment in which both BF and RFP images are taken once every 6min. Images from the two channels are alternatingly

stored by the microscope (one BF, one RFP, one BF, etc.). They are read into MATLAB as a 3D matrix (Dimensions: ImageWidth x ImageHeight x Number of image in stack) and BF and RFP images are split by taking the odd-numbered frames as a BF stack, and even-numbered frames as a RFP stack.

- Segment.m
  Reads in frame by frame, performing noise-reduction by substracting a disk-structured image from the raw image, segments denoised images using Otsu's method, and saves both denoised and segmented image stacks as 3D matrices.

- fmeasure.m
  Function to measure the relative degree of focus of an image, taken from a paper benchmarking 37 blur detection algorithms (S. Pertuz et al., Analysis of focus measure operators for shape-from-focus. Pattern Recognition, 46(5):1415:1432, 2013.)

- Detect_blurred_images.m
  Calculates the focus measure using the fmeasure function on each frame in the stack, then finds peaks of high focus measure values (high focus measure = blurred image) and removes them from the stack, the peak finding and removal process is done until no more blurred frames remain. For each image stack a plot of focus measure per frame is generated with detected blurred frames marked with asterisks.

- Output_segm_denoised_images.m
  Writes segmented and/or denoised images that are not blurred into a single TIF stack called "segmented_filtered_stack_%s.tiff" and/or "denoised_filtered_stack_%s.tiff" (s = name of the original image stack) into the "Denoised_and_filtered_images" folder.

- Get_blob_info.m
  Sets up a blob analysis object and runs blob analysis. For each segmented frame the output of the blob analysis is a 9xN matrix (N = number of detected blobs in the frame) and the columns are (Blob area in pixels; Centroid coordinate x; Centroid coordinate x; Bounding box coordinate x; Bounding box coordinate y; Bounding box width; Bounding box height; Major axis length in pixels; Minor axis length in pixels;) and a label matrix, which is a copy of the binary image with each blob enumerated. The 9xN result matrix is saved in a blob_stack list, and label matrix in the label_stack list. You may specify other statistics to be calculated in the blob analyzer object.

- Count_cells.m
  Counts the number of blobs in each frame and plots two plots with a different x axis: number of cells per frame and number of cells in minutes (where the time between two frames is taken to be 6min). **Nota bene** - This script counts all visible nuclei in the frame, which means it does not track which cells have entered or left the frame, and therefore is a rough estimate of the population growth.

A simple tracking method was also implemented, but yielded unreliable results. The following scripts are used for tracking:

- Track.m
  Main script.

- Correct_detection_errors.m
  Sometimes two nuclei appear to split and merge again. This script makes sure they are counted as two separate blobs in a frame where they appear merged after segmentation by checking if the bounding box of frame$_{n+1}$ overlaps with more than 1 centroid from frame$_n$. If it does it means there is a segmentation error and the blob information from the previous frame is used.

- Get_distances_between_centroids.m
  Calculates distances between centroids in two frames.

- Output_tagged_images.m
  Makes a folder called "Tagged_images" and saves all images from the RFP channel with a bounding box over detected cells and an ID in each box. The IDs may be individual cell IDs (ids in the code) or cell lineage IDs (lids in the code).

- Track_new.m
  New method of tracking via point set registration. Unfinished.