Similarity-Based Motion Tracking of Cells in Microscopic Images

Horst PERNER, Petra PERNER
Institute of Computer Vision and Applied Computer Sciences, Leipzig, Germany

Abstract. Live-cell assays are used to study the dynamic functional cellular processes in high-content screening of drug discovery processes. The large amount of image data created during the screening requires automatic image-analysis procedures that can describe these dynamic processes. One class of tasks in this application is the tracking of cells and the description of the events and the changes in the cell characteristics so that the desired information can be extracted from it based on data-mining and knowledge-discovery methods. In this paper, we propose a similarity-based approach for motion detection of the entire cell. Results are given on a test series from a real drug discovery process.

Keywords. cell tracking, similarity-based cell tracking, microscopic image analysis

1. Introduction

The utilization of dynamic high-content analysis approaches during preclinical drug research will permit to gain a more specified and detailed insight into complex subcellular processes by the use of living cell culture systems. This will effectively support future drug discoveries leading to a great outcome of highly specific and most effective drugs that come along with a well improved compliance. Live-cell assays are therefore used to study the dynamic functional cellular processes in high-content screening of drug-discovery processes. The large amount of image data created during the screening requires automatic image analysis procedures that can automatically describe these dynamic processes. One class of tasks in this application is the tracking of the cells, the description of the events and the changes in the cell characteristics, so that the desired information can be extracted from it based on data-mining and knowledge-discovery methods.

There are several tracking approaches known that track a cell by detection in each single frame and associate the detected cells in each frame by optimizing a certain probabilistic function [1]. Other approaches track cells by model evolution [2]. This approach seems to be able to handle touching and overlapping cells well but is computational expensive [3].

We propose in this paper a similarity-based approach for motion detection of cells. In Section 2 the material is given and the basic constraints of the algorithm resulting from the application are explained. The algorithm is described in Section 3. Results are
2. Material and Investigation of the Material

The live cell movies are comprised of a sequence of 113 images. Figure 1 shows images for six time frames. These images are taken in a time interval of thirty minutes. The whole experiment runs over two and a half days. The cells show no steady movement. They might suddenly jump from one direction to the opposite one. The cells might also turn around their own axis that will change their morphological appearance. They also might appear and disappear during the experiment as well as re-appear after having gone out of the focus of the microscope. It is also highly likely that the cells might touch or overlap. The tracking algorithm should run fast enough to produce the result in little computation time.

The following conditions have been decided for the algorithm: Only cells that are in focus should be tracked (background objects are not considered). Fragmented cells at the image borders are eliminated. Each detected cell gets an identity label and gets tracked. Disappearing cells are tracked until they disappear. Newly appearing cells are tracked upon their appearance and get a new identity label. Cells overlapping or touching each other are eliminated. They are considered as disappearing cells. Splitting cells are tracked after splitting and both get an identity label that refers to the mother cell. Cell fragments are eliminated by a heuristic less than 2xSigma of Cell Size. Note that we decided to exclude overlapping cells for the reason of the resulting higher computation time. If we want to consider this kind of cells, we can use our matching algorithm in [4] to identify the portion of touching cells belonging to the cell under consideration.

3. The Object-Tracking Algorithm

The unsteady movement of the cells and the movement of the cells around their own axis required special processing algorithms that are able to handle this. The successive estimation of the movement based on mean-shift or Kalman filters [5] would only make sense if we can expect the cells to have steady movement. Since it can happen that the cells jump from one side to another, we used a search window around each cell to check where the cell might be located. The size of the search window is estimated based on the maximal distance a cell can move between two time frames. The algorithm searches inside the window for similar cells. Since we cannot expect the cell
to have the same shape after turning around its own axis and as also the texture inside
the cell might change, we have chosen a similarity-based procedure for the detection of
the cell inside the window. The overall procedure is shown in Figure 2.

The image gets thresholded by Otsu’s well-known segmentation procedure. Afterwards
the morphological filter opening by a 3x3 window is applied to close the contour and the
inner holes. Fragmented cells at the image borders as well as small
remaining objects of a size of ten pixels are deleted. The cells at the image borders are
only tracked when they fully appear inside the image. Around the object is drawn the
convex hull and remaining holes or open areas inside the cell area are closed by the
operation flat-fill. The resulting images after these operations are shown in Figure 3 for
the six time frames. This resulting area is taken as the cell area and the area with its
grey levels is temporarily stored as template in the data base.

Then the center of gravity of the object is determined and the search window is
tentatively spanned around the object. A raw estimation of the cell’s movement
direction is calculated by the mean-shift filter over 3 frames. In the resulting direction
is started the search for similar cells. Cells fragmentated by the window border are not
considered for further calculation. Each cell inside the window is compared by the
similarity measure to the respective template of the cell under consideration. Before the
similarity is determined the cells are aligned, so that they have the same orientation.
The cell having the highest similarity score to the template is labeled as the same cell
moved to the position $x, y$ in the image $t+1$. The template is updated with the detected
cell area for the comparison in the next time frame. The position of the detected cell is
stored into the data base under the label of the template. Mytotic cells are detected by
classifying each cell based on the texture descriptor given in [6] and the decision tree
classifier.

Let $C_t$ be the cell at time-point $t$ and $C_{t+1}$ the same cell at time point $t+1$. Then the
rule for labeling a cell as “disappeared” is: IF $C_t$ has no $C_{t+1}$ THEN Disappearing_Cell.
Similarity determination between two cells. The algorithm computes the similarity between two image areas A and B; these images are in our case the bounding box around each cell (see Figure 4). According to the specified distance function, the proximity matrix is calculated, for one pixel at position $r, s$ in image A, to the pixel at the same position in image B and to the surrounding pixels within a predefined window. Then, the minimum distance between the compared pixels is computed. The same process is done for the pixel at position $r, s$ in image B. Afterwards, the average of the two minimal values is calculated. This process is repeated until all the pixels of both images have been processed. The final dissimilarity for the whole image is calculated from the average minimal pixel distance. The use of an appropriate window size should make this measure invariant to scaling, rotation, and translation.

Input: image A and image B

grey-level matrix
determination of the window size

Begin with first pixel: $r = s = 1$

computation of proximity matrix $d(a_{rs}, W_B)$ and $d(b_{rs}, W_A)$ based on $f_{pi} = d_{xy}$

search minimum:

$f_{pi}(a, B) = d_{pi}(a_{rs}, W_B)$

search minimum:

$f_{pi}(b, A) = d_{pi}(b_{rs}, W_A)$

$s = 1/2 (d_{pi}(a_{rs}, W_B) + d_{pi}(b_{rs}, W_A))$

Output: $D(A, B)$

$$D(A, B) = \sqrt{\sum f^2 (r, s)}$$

**Figure 4.** Flowchart of the algorithm for computing the similarity measure

<table>
<thead>
<tr>
<th>Cell_1 Image002</th>
<th>Cell_1 Image003</th>
<th>Cell_2 Image003</th>
<th>Cell_3 Image003</th>
</tr>
</thead>
<tbody>
<tr>
<td>Similarity value</td>
<td>0.0554918423</td>
<td>0.1190173551</td>
<td>0.0833996087</td>
</tr>
</tbody>
</table>

**Figure 5.** Cells used for comparison

The resulting similarity value based on this similarity measure for the pairwise comparison of cell_1 to cell_1, cell_1 to cell_2, and cell_1 to cell_3 in the preceding time frame is given in Figure 5. The lowest similarity value is obtained when comparing cell_1 to cell_1 in the proceeding time frame.
4. Results

Results in Figure 6 a–e show the tracking path of six different cells. We compared the manually determined path of a cell from an operator of the high-content screening process-line with the automatic determined path by the tool IBai-Track for 10 videos with 117 frames each. If both methods gave the same path we evaluated it as positive otherwise as negative. We observed a correspondence between these two descriptions of 98.2%. The computation time for a sequence of 117 images of 674x516 pixels each and on average 80 cells per image is 11 minutes 42 seconds on PC with 1.8 GHz.

5. Conclusions

We have presented our new cell-tracking algorithm for tracking cells in dynamic drug discovery experiments. The algorithm uses a window around the position of the cell in the preceding time frame for searching for the next position of the cell. This is necessary since the movement of the cells cannot be steady. The search inside the window is started in the expected direction of the cell. To calculate this direction, we use a mean-shift filter over 3 time frames. The detection of the cell is done based on a similarity determination of the grey-level profile between two cells. The cell giving the highest similarity value is selected as the same cell in the preceding time frame. A similarity measure that is invariant to translation is used for that reason. The resulting data are stored in a data file and can be used for further data-mining analysis.

References


