BIOINFORMATICS

Vol. 00 no. 00 2008
Pages 1–2

CellTrack: An Open-Source Software for Cell Tracking and Motility Analysis*

Ahmet Sacan[1,2,†], Hakan Ferhatosmanoglu[2,3], and Huseyin Coskun[3]

[1]Computer Engineering Department, Middle East Technical University, Ankara, TURKEY
[2]Dept. of Computer Science and Engineering, The Ohio State University, Columbus, OH, USA
[3]Biomedical Informatics Department, The Ohio State University, Columbus, OH, USA

Received on ; revised on ; accepted on

Associate Editor:

ABSTRACT

Motivation: Cell motility is a critical part of many important biological processes. Automated and sensitive cell tracking is essential to cell motility studies where the tracking results can be used for diagnostic or curative decisions and where mathematical models can be developed to deepen the understanding mechanisms underlying the cell motility.

Results: We propose a novel edge-based method for sensitive tracking of cells, and propose an ensemble of methods that achieves refined tracking results even under large displacements or deformations of the cells. The proposed methods along with other general purpose image enhancement methods are implemented in CellTrack, a self-contained, extensible, and cross-platform software package.

Availability: CellTrack is an Open Source project and is freely available at http://db.cse.ohio-state.edu/CellTrack

Contact: sacan@cse.ohio-state.edu

1 INTRODUCTION

Cell motility is an essential part of many biological processes that are necessary for the sustenance of an organism. Free-living unicellular organisms move to avoid toxic substances or to approach nutrients. Tissue cells of multicellular organisms mobilize during embryologic development (morphogenesis), wound healing, maintenance of tissues, generation of new blood vessels, cancer metastasis, and immune response. Understanding the underlying mechanisms of cell motility is crucial for curative or preventative treatments to many diseases that are caused by abnormalities in cell locomotion.

Accurate segmentation and tracking of cells in microscopic imaging is becoming to be an important step in cell-motility studies. For instance, tracking number and velocity of rolling leukocytes is essential to understanding and successfully treating inflammatory diseases (Ray et al., 2002). Mathematical modeling of cell locomotion also requires sensitive tracking of moving cells. In (Coskun et al., 2007), live cell imaging data is used to solve the inverse modeling problem in order to determine the material properties of the cells.

Manual processing of cell locomotion data is labor-intensive and error-prone. Automated cell tracking methods developed so far mainly focus on obtaining the overall velocity of a moving cell and do not provide a sensitive tracking of the cell as it deforms during its locomotion. Moreover, these methods are usually developed in isolation, addressing a specific problem, but not as part of an integrated software environment. The available software packages that do incorporate some of these methods are either proprietary or closed-source, which makes them inapplicable and non-extendible to different research problems.

We have developed a new edge-based tracking method that is based on *active snakes* (Kass et al.) and relies on a new energy functional that can accurately track changes in the shape of a moving cell. We also show that combining various tracking methods can yield more robust tracking results. These methods are implemented as part of CellTrack, an integrated and extensible software environment for tracking cells.

2 METHODS

Active snakes are elastic curves that evolve on the image plane to capture object boundaries (Kass et al.). Snake evolution is based on an energy minimization procedure that improves the coordinates of the snake. The snake energy is generally defined as a weighted sum of internal (E_{int}, e.g., continuity and curvature) and external (E_{ext}, e.g., image gradient) energy terms over the snake's elements (snaxels). Using active snakes for object tracking involves initializing a snake to its configuration from previous frame and evolving the snake again for the new frame.

We propose to use the configuration of the snake from the previous frame not just for initialization, but also as a constraint in the snake energy term, such that the snake in the new frame would effectively match up with the fitness of its previous configuration to the image plane. Our proposed energy functional is as follows:

$$E_{snake} = w_1 E_{int} + w_2 E_{ext} + w_3 E_{fit}$$

where E_{fit} is defined as a weighted sum of fitness terms:

$$E_{fit} = \sum_x w_x |E_x - E'_x|$$

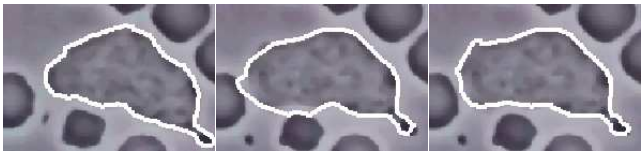
where x denotes the type of constraint being matched and E'_x is the energy term for x from the previous frame. The obvious type of such constraints are

*This research is partially supported by US National Science Foundation (NSF) Grant IIS-0546713 and Turkish Scientific and Research Council (TUBITAK) Grant 107E173.

†to whom correspondence should be addressed

the internal and external energy terms used in the evolution of the snake itself. However, other descriptors which are not appropriate for boundary detection and not included in the original snake energy can also be used here as constraints to improve the sensitivity of the tracking. We have identified three such descriptors: image intensity, signed image gradient and arc length of each snaxel. Our model can easily be extended to other color or texture features.

Figure 1 illustrates the success of our energy functional. Without the E_{fit} terms, the snake fails to capture the boundary of the tracked object but falsely deforms to the edges of the neighboring objects. Inclusion of E_{fit} terms causes the snake to maintain its internal and image-relative configuration from the previous frame to correctly capture the boundaries of the same object in the new frame. Note that our approach also has the unique benefit of providing a sensitive association of the snaxels across frames (see the inset of Figure 3 for an illustration). This is especially useful in cell-motility modeling and analysis studies, where the tracking of the bulk of the object is not sufficient and such a sensitive association is required (Coskun et al., 2007).



(a) previous frame (b) without E_{fit} (c) with E_{fit}

Fig. 1: Active snake tracking with and without the E_{fit} energy terms.

Note that tracking using active snakes relies on the assumption that the movement and deformation of the tracked object is small between consecutive frames. In order to relax this assumption and obtain more robust tracking results, we developed an ensemble of different tracking methods in a coarse to refined fashion. We refer the reader to Yilmaz et al. (2006) for a general survey of the object tracking methods.

Our combined method performs the following steps: The overall displacement and rotation of the object is first determined using a template matching method (Figure 2b). The resulting contour is used as the initial state to optical flow based deformation (Figure 2c); we use statistical outlier detection and local interpolation to achieve resistance against errors in the optical flow evaluation. Our extended active snake method is then applied to obtain the final snake configuration (Figure 2d). The combined method achieves accurate tracking even for large displacements or deformations of the objects between frames.

(a) prev. frame (b) templ. match (c) optic flow (d) active snake

Fig. 2: Tracking moving-camera capture using combination of methods.

3 THE SOFTWARE

The methods outlined above are implemented in CellTrack, a software package that aims to automate the cell tracking process. The simple and intuitive user interface allows easy navigation and analysis of image frames (see Figure 3 for a snapshot). In addition to the tracking methods, CellTrack provides general image processing and enhancement functions such as smoothing, background subtraction, and object segmentation. For each of these methods, the user can change the default parameters to meet the discrepancies and demands of different tracking applications. The ability of immediate previewing of results makes it easy to investigate the effect of each parameter. Besides automated cell detection and tracking capability, the interface also allows manual editing to initialize or modify the tracking data. CellTrack can be used to work with movie or image files of a variety of file formats. The tracking results can be exported either as raw text data for further numerical analysis, or as movie or image files for visualization, sharing, and publishing.

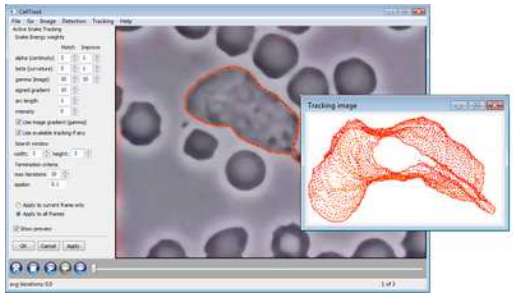


Fig. 3: Snapshot of CellTrack graphical user interface.

CellTrack is developed in C++ in order to avoid dependence on commercial development and deployment products and to avoid the computational overhead of higher-level languages. The image processing functions are based on OpenCV library (Bradski, 2000) and the graphical user interface is implemented with wxWidgets (Smart et al., 2005). Both of these libraries are open source and available for a broad range of platforms including Linux, Windows, and Mac OS, making CellTrack also a cross-platform software. CellTrack has been developed in a modular design pattern where the processing logic is separated from the user interface details and implemented as separate plugins. This design pattern makes CellTrack easily extendable to incorporate new methods. We believe that CellTrack will be useful not only as a cell tracking application for analysis of cell motility but also as a platform for development of new cell tracking methods. Finally, we wish to note that CellTrack is by no means a final solution to all cell tracking problems, but a good platform for continuing development. Please see the user manual of the software distribution for the current limitations of CellTrack.

REFERENCES

Bradski, G. (2000). The opencv library. *Dr. Dobb's Software Tools for the Professional Programmer*.

Coskun, H., Li, Y., and Mackey, M. A. (2007). Ameboid cell motility: A model and inverse problem, with an application to live cell imaging data. *Journal of Theoretical Biology*, 244(2):169–179.

Kass, M., Witkin, A., and Terzopoulos, D. Snakes: Active contour models. *International Journal of Computer Vision*, pages 321–331.

Ray, N., Acton, S., and Ley, K. (2002). Tracking leukocytes in vivo with shape and size constrained active contours. 21(10):1222–1235.

Smart, J., Hock, K., and Csomor, S. (2005). *Cross-Platform GUI Programming with wxWidgets*. Prentice Hall PTR.

Yilmaz, A., Javed, O., and Shah, M. (2006). Object tracking: A survey. *ACM Computing Surveys*, 38(4).