Automatic Tracking of Particles in Dynamic Fluorescence Microscopy

Daniel Sage[1], Florence Hediger[2], Susan M. Gasser[2], Michael Unser[1]

[1] *Biomedical Imaging Group*
Swiss Federal Institute of Technology Lausanne (EPFL), Switzerland

[2] *Department of Molecular Biology*
University of Geneva, Switzerland

daniel.sage@epfl.ch, florence.hediger@molbio.unige.ch,
susan.gasser@molbio.unige.ch, michael.unser@epfl.ch

Abstract

We present a new, robust algorithm for tracking fluorescent particles in dynamic image sequences obtained by brightfield or confocal microscopy. Specifically, we consider the problem of extracting the movement of chromosomal telomeres within the nucleus of a budding yeast cell. Our method has three components. The first is an alignment module that compensates for the movement of the biological structure under investigation. In our application, the images are aligned to the center of gravity of the nucleus which is detected by thresholding and fitted with an ellipse. The second step is a Mexican-hat filtering which we show to be optimally tailored to the detection of a Gaussian-like spot in fractal noise. The final component is a tracking algorithm that uses dynamic programming to extract the optimal (x,y,t) trajectory of a particle.

We have implemented the method as a Java Plugin for the public-domain ImageJ software. We have applied it to real data and have obtained results that are as good—if not better—as manual tracings. Our new algorithm reduces the analysis time of a 300 image sequence from 10 minutes, when it is done manually, to just a few seconds and offers the benefit of reproducibility.

1. Introduction

During the past decade, two important technological innovations have contributed to reshaping molecular biology research. The first was the development of fluorescent proteins that allow researchers to selectively label single proteins or DNA loci in vivo. The second is high resolution fluorescence imaging that is made possible by the new generation of brightfield and confocal microscopes. Thanks to these new tools biologists are able to observe gene expression and to study molecular dynamics within the living cell at sub-micron resolutions. They can acquire static images in two (XY) or three (XYZ) dimensions at one or several wavelengths (multispectral imaging) to localize the labeled structures of interest in a living specimen. They can also record dynamic sequences (time-lapse series) to study molecular transport or conformal changes within the cell and in the extracellular matrix.

While these methods offer an enormous potential for increasing our understanding of biology, they also constitute a real challenge for researchers in the field who have not yet devised efficient ways to exploit and quantitatively interpret this unprecedented flow of data. Currently, the large majority of data analysis and feature extraction is done manually, which is very time consuming. There are some commercial image analysis tools available but their capabilities for automatic feature extraction are rather limited. One of the difficulties is that the data are typically extremely noisy due to the weakness of the fluorescence signal and also because the instrumentation is pushed to its limits.

In this work, we are proposing a new algorithmic solution for the problem of tracking a fluorescent particle on a very noisy background. This problem, which is rather generic in fluorescence imaging, is motivated by an interesting application which is study of the dynamics of the chromosomes within the nucleus of the cells [1]. The ultimate goal is to obtain a complete description of the time trajectory of a particle. The results are then exploited for statistical analyses, model fitting, and the comparison of various conditions [2].

The algorithm that we propose has three primary components: 1) the detection of the nucleus and extraction of its center of gravity in order to provide a spatial reference; 2) the preprocessing by matched filtering to facilitate the detection of the particles, and 3) the search for the optimal time-trajectory which is solved most efficiently by dynamic programming. The system is capable of a fully automatic analysis but can also accept user-specified constraints. The operator can intervene by moving points on the computed trajectory, which is then updated accordingly by running the constrained optimization again. The method is extremely robust

because the tracking is done globally; in other words, the decision for a detection is based not only on the present and the recent past (which is the approach taken by most commercial software packages, such as Volocity 2 from Improvision [3]), but on the future as well, taking the whole data set into consideration.

2. Our motivation: study of the chromosome dynamics

To study the chromosome dynamics, specific chromosomal loci are labeled through the binding of a fluorescent protein (Green Fluorescent Protein or GFP). The behavior of this tagged locus can now be visualized *in vivo* through excitation of the GFP molecule and capture of its emission signal. Spatial information (XY) or (XYZ) are acquired at different time intervals giving rise to a dynamic analysis of a DNA molecule position [2].

Once the data are acquired, careful measurements have to be done in order to calculate biologically relevant properties such as rate of motion, spatial confinement and long-range interactions between independent particles. Our goal is to replace the laborious manual by an automated tracking that can handle the whole stack of images simultaneously. Automation also permits a reproducible analysis.

2.1. Biological background

The yeast *Saccharomyces cerevisiae*, known as the baker's yeast or budding yeast, is a single-celled organism. Although molecular processes are much simpler in this small eukaryotic cell, most of the essential biological pathways are conserved and knowledge gained from genetic studies is generally directly applicable to more complex organisms. This advantage and the availability of the complete yeast genome make yeast a very attractive biological model.

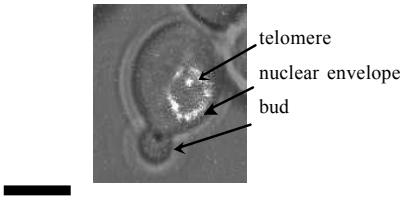

Figure 1. Image of a yeast cell expressing Nup49-GFP (nuclear envelope) in which Tel VI-R has been tagged. This cell is in a synthesis stage characterized by a small bud. Bar is 2μm.

Recently, increasing evidence suggests that the nucleus is not simply a membrane-bound vesicle enclosing massive amounts of nucleosomes-bound DNA, but that it is highly organized into various sub-compartments, probably to facilitate different nuclear functions [4]. The diploid yeast genome is composed of 32 linear chromosomes (64 telomeres).

Immunolocalization of telomeric proteins have shown that telomeres adopt a non-random configuration in the interphase nucleus, regrouping themselves into 8-10 foci that are preferentially located at the periphery of the nucleus near the envelope. This particular organization creates high local concentrations of silencing factors that bind telomeres and promote transcriptional silencing. These observations led to a growing number of questions about the dynamics of telomeres that could not be answered by classical immuno-microscopy of fixed cells. We analyzed by live microscopy the dynamic behavior of several specific telomeres in a dividing cell and their location in relation to the periphery of the nucleus.

2.2. Features of the acquisition

Images of a single yeast nucleus were taken every 1.5 seconds during 3-5 minutes with a confocal microscope. The capture rate is a compromise between scanning time, resolution and GFP bleaching [5]. To minimize bleaching, we do not take *z*-stacks over long periods of time, but instead we capture XY images and try to maintain the GFP spot in the focal plane by moving the objective up and down within a range of about 1 μm. This method allows a precise 2D + time analysis over relatively long periods of time (300-500 seconds).

2.3. Manual measurements

A manual analysis of these images requires that, for each frame, the biologist scores manually the coordinates of the center of the nucleus and the coordinates of the center of the GFP-tagged DNA spot by clicking on the corresponding pixel. These coordinates are reported in a spreadsheet for subsequent statistical analysis. It entails the entry of at least 600 mouse clicks per time lapse.

3. Prepocessing steps

3.1. Nucleus alignment

Since one is interested in characterizing the movement of the telomere inside the nucleus, the nuclear envelope has also been labeled fluorescently so that this structure remains visible. The practical difficulty is that the nucleus does not necessarily stay still during acquisition. The first step of our analysis is therefore to compensate for this movement. For this purpose, we smooth the image and apply a global threshold to detect the fluorescent structures which are predominantly located in the nuclear membrane. We then fit the thresholded data with an ellipse, using the least squares method of Fitzgibbon which works well for scattered data [6]. A typical result is shown in Figure 2.

The images are then realigned with respect to the center of these ellipses and also cropped to the largest radius.

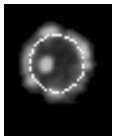

Figure 2. White overlay of the detected ellipse over the image.

3.2. Spot enhancing filter

To a first approximation, a typical spot profile can be represented by a two dimensional Gaussian function. There are two practical difficulties with our detection task. First, the spot does not only move laterally (x,y) but it can also move up or down along the z-axis outside the focal plan which tends to make it dimmer and more spread out. Second, the data is extremely noisy because we are operating at the detection limit to minimize the destructive effect of photo-bleaching. If we now make the assumption that the noise is fractal- like with a spectral power density that decays like $O(\omega^2)$ where ω is the radial spatial frequency, we can show that the optimal detector (whitened match filter) is given by the Laplacian of a Gaussian (Mexican hat filter). Prefiltering the images with this particular filter (bandpass) has the desirable effect of enhancing the spots while at the same time getting rid of some of the background structures. We note, however, that our tracking algorithm is so robust that this prefiltering is not always necessary.

4. Tracking using dynamic programming

The tracking conditions that are specific to our problem are the following:

1) There is one single spot to track over time; the difficulty is that it may be dimmed or even absent in some parts of the sequence.
2) There are as many as 300 to 500 images per sequence.
3) The structures of interest are small (about 30*30 pixels for the nucleus) and very noisy.
4) The fluorescence responses of the spot and of the nuclear proteins are similar; there is no way to distinguish them using the gray level information alone.
5) The movement of the spot is limited to few pixels from frame to frame.
6) The tracking can take place off line when the whole data has been acquired.

The detection task is difficult but sufficiently contrained to be formulated as a global optimization problem. In particular, we should take advantage of the strong dependency between the position of the spot in one image and the next one (chaining property). Here, we propose to solve the problem using dynamic programming (DP).

While the use of DP appears to be new in the biological context, it is a relatively standard technique for detecting contours or segmentating objects in images [7][8]. A few authors have also applied DP for target tracking in aero-space and military applications [9][10]. However, one should emphasize that the constraints there are different from ours: they must provide the best detection at a given time t based on the past only (on-line detection). In our case, we have the whole sequence available at once and we want to recover the trajectory from beginning to end. The key advantage of DP is that tracking and detection are performed simultaneously, which greatly improves the performance of the system.

4.1. Tracking problem specification

We are given a sequence $f(x,y,t)$ of N preprocessed images. The space (x,y) and time (t) indices take integer values with $x = 0,\cdots,N_x - 1$, $y = 0,\cdots,N_y - 1$ and $t = 0,\cdots,N-1$, respectively. The task is to find an optimal time-trajectory $\{\mathbf{x}_t\}_{t=0,\cdots,N-1}$ that describes the displacement of the particle.

A key constraint is that the maximum excursion from one frame to the next is limited:

$$\|\mathbf{x}_t - \mathbf{x}_{t+1}\| < \Delta$$

where Δ is a parameter of the algorithm. Since we are considering a discrete grid in space and time, it is possible, at least in principle, to enumerate all possible trajectories and to attempt to select the best possible one based on some objective criterion.

The cost function $\xi(\mathbf{x}_0,\mathbf{x}_1,\cdots\mathbf{x}_{N-1})$ must incorporate as much problem specific information as possible. In particular, we would like to:

1) favor positions where the intensity is bright or, alternatively, where the response to the spot-enhancing filter is strong;
2) favor smooth trajectories; this can be achieved by penalizing paths for which the average displacement from one frame to the next is large;
3) favor positions that are closer to the center of the nucleus in order to differentiate the spot from the nuclear membrane proteins that have a similar profile.

These various constraints can be incorporated by defining the following cost function:

$$\xi(\mathbf{x}_0,\mathbf{x}_1,\cdots\mathbf{x}_{N-1}) = \sum_{t=1}^{N-1}\left(f(\mathbf{x}_t,t) - \lambda_1\|\mathbf{x}_t - \mathbf{x}_{t-1}\| - \lambda_2\|\mathbf{x}_t - \mathbf{x}_c\|\right)$$

where λ_1 and λ_2 are weighting factors to be specified by the user and where $\mathbf{x}_c$ is the center of the nucleus.

4.2 Dynamic programming algorithm

An important property of the above cost function is that is satisfies the Bellman principle in that "*an optimal decision for each of the remaining states does not depend on previously reached states or previously chosen decisions*" [11]. In other words, for each point on the path at time t, there is only a cost contribution associated to its position $\mathbf{x}_t$ plus, eventually, a penalty due to the transition from the previous state; e.g, $\|\mathbf{x}_t - \mathbf{x}_{t-1}\|$. The implication is that the corresponding discrete optimization problem can be solved most efficiently by dynamic programming. The thrust of the algorithm is a main loop for $t = 0,\cdots,N-1$ whereby all potential (x,y,t) candidates are examined once only. At each position, the algorithm applies an iterative update formula for computing the optimal cost for reaching the current state. At the end of the process, the optimal solution is retrieved by back tracking. The algorithm is sufficient fast to provide interactive feedback.

In its standard formulation, the DP approach requires the specification of the starting point ($\mathbf{x}_0$) for the path. The algorithm is then capable of computing the optimal path to reach time $t = N-1$ with or without specification of the end point. It is also possible to consider the specification of the end point only, by running the algorithm backwards. Likewise, it is easy to introduce arbitrary constraints on the path by subdividing it into segments.

In our implementation we specify the first constraint automatically by selecting the brightest pixel in sequence which is very likely to belong to the particle of interest.

We also have the option of constraining the optimization further by specifying additional nodes. In this case, the optimal trajectory is recomputed only on the segments adjacent to the new node. In our implementation, the user can edit, move, add or delete nodes and the tracking is updated automatically with a response time that is typically less than a second. This another advantages of the DP procedure: the biologist expert can control the trajectory and work hand-in-hand with the algorithm when the quality of data is poor.

5. Implementation and results

5.1. Implementation

The presented algorithm of DP was implemented is Java as a plugin for ImageJ [12]. ImageJ is a public-domain software package for image processing. It is widely used by biologists and can process image sequences (or stacks). In addition to the DP tracking, our plugin contains the following features: 1) user interface, 2) nucleus alignment 3) preprocessing, 4) visualization of the results.

5.2. User interface

The program has a graphical user-interface with the following functionality: selection of the region of interest; parameter setting for the DP tracking (Δ, λ_1 and λ_2); and placement of the node positions by clicking on the 3D sectioning views (cf. Figure 3.).

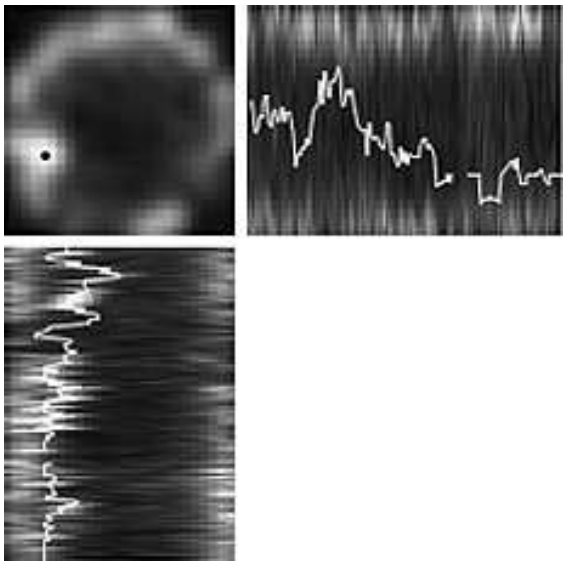

Figure 3. Three orthogonal sectioning views of the image and the path. 1) Upper left: section xy 2) Upper right section $y(t)$ with a trajectory in white overlay 3) Lower left section $x(t)$.

5.3. Result

Presently, we have applied the algorithm to more than 20 sets of real data. The center of the nucleus envelope was well located in all cases, and the detected trajectories were judged to be satisfactory, even when the images were extremely noisy (cf. Figure 4.). When the spot disappears completely for a short period of time—e.g. when it is out of focus—the algorithm is able to follow it nonetheless using the displacement constraint. On the other hand, it can fail when the spot disappears for too long a period. In this case, the user can easily enter new node points to further constrain the trajectory.

We found the algorithm to be reasonably fast in our application where the images are small. For a sequence of 200 images (50*50 pixels), the alignment of the whole sequence is done in 1.2 seconds on a Macintosh G4/800 MHz; the trajectory of the spot is tracked in 0.8 seconds when Δ=1 (resp. 3.0 seconds when Δ=5). This time has to be compared to the 10 minutes that are required for a manual extraction. Even though dynamic programming is a systematic approach that evaluates all allowable trajectories, it gives the answer quite rapidly. The most time-consuming part of the algorithm is the large number

of evaluations of the cost function for every transition. This number is directly proportional to the image size, the maximum gap Δ and the number of nodes.

We expect this program to be useful in practice. Indeed, the biologists appreciate the fact of being able to extract a complete and reproducible trajectory in just a few clicks.

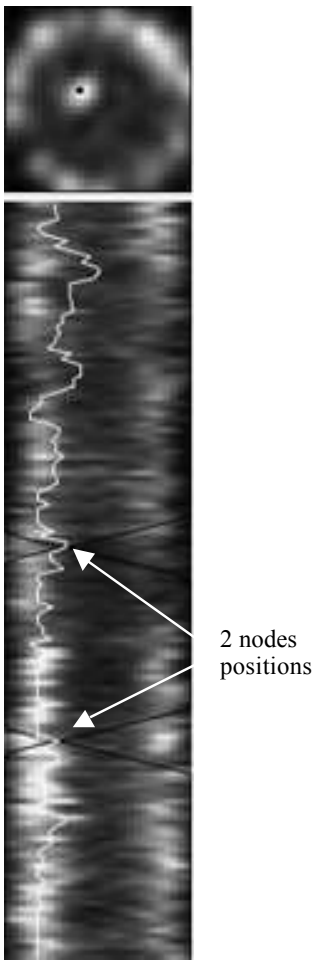

Figure 4. Result of a tracking over a sequence of 141 images (26*26 pixels). The upper image is the *xy* section at t=32. The lower image is a $x(t)$ section. The white overlay is the trajectory of the spot in the $x(t)$ section; the black overlay is the envelope of the searching space. Here, the maximum displacement is Δ=3.

6. Conclusion

We have presented a new algorithm for tracking the movement of a particle in a sequence of noisy images. The key feature that makes the method robust and relatively full-proof is that the detection task is formulated as a global optimization problem. The optimal solution is computed most efficiently by dynamic programming. The algorithm has been successfully applied to the analysis of the movement of chromosomal telomeres within the nucleus of a yeast cell. The results obtained are quite satisfactory suggesting that the dynamic programming approach has good potential for similar biological imaging problems. While the particle trajectories are usually extracted completely automatically, the software can accept hints or corrections provided by the biology expert. Its response time is sufficiently fast for it to operate in a semi-interactive mode with the priority given to the user input. In its present configuration, the system can track a spot over a sequence of 300-500 images in just a few seconds.

7. References

[1] S.M. Gasser, "Nuclear Architecture - Visualizing Chromatin Dynamics in Interphase Nuclei," *Science,* vol. 296 (5572), pp. 1412-1416, 2002.

[2] P. Heun, T. Laroche, K. Shimada, P. Furrer, S.M. Gasser, "Chromosome Dynamics in the Yeast Interphase Nucleus," *Science*, vol. 294 (5549), pp. 2181-2186, 2001.

[3] Web site of Volocity: http://www.improvision.com/

[4] S.M. Gasser, "Positions of Potential: Nuclear Organization and Gene Expression," *Cell*, vol. 104, pp. 639–642, 2001.

[5] F. Hediger, Neumann F.R., Van Houwe G., Dubrana K., S.M. Gasser, "Live Imaging of Telomeres: yKu and Sir Proteins Define Redundant Telomere-Anchoring Pathways in Yeast," *Current Biology*, vol. 12 (24), pp. 2076-2089, 2002.

[6] A. Fitzgibbon, M. Pilu, R.B. Fisher, "Direct Least Square Fitting of Ellipses," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, pp. 476-480, 1999.

[7] D. Geiger, A. Gupta, L. A. Costa, and J. Vlontzos, "Dynamic Programming for Detecting, Tracking, and Matching Deformable Contours", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 3, pp. 294-302, 1995.

[8] M. Unser, G. Pelle, P. Brun and M. Eden, "Automated Extraction of Serial Myocardial Borders from M-Mode Echocardiograms," *IEEE Transactions on Medical Imaging*, vol. 8, no. 1, pp. 96-103, 1989.

[9] Y. Barniv, "Dynamic programming solution for detecting dim moving targets," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 21, no. 1, pp. 144-156, 1985.

[10] S.M. Tonissen, R.J. Evans, "Peformance of dynamic programming techniques for Track-Before-Detect," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 32, no. 4 , pp. 1440 –1451, 1996.

[11] P. Lappas, J.N. Carter, R.I. Damper, "Robust Evidence-based object tracking," *Pattern Recognition Letters*, 23 (1-3), pp. 253-260, 2002.

[12] Web site of ImageJ: http://rsb.info.nih.gov/ij/