Spatio-Temporal Cell Cycle Phase Analysis Using Level Sets and Fast Marching Methods

Dirk Padfield\textsuperscript{a,c,*}, Jens Rittscher\textsuperscript{a}, Nick Thomas\textsuperscript{b}, Badrinath Roysam\textsuperscript{c}

\textsuperscript{a}GE Global Research, One Research Circle, Niskayuna, NY, 12309.
\textsuperscript{b}GE Healthcare, Maynard Centre, Cardiff CF147YT, UK.
\textsuperscript{c}Rensselaer Polytechnic Institute, 110 8th St., Troy, NY 12180.

Abstract

Enabled by novel molecular markers, fluorescence microscopy enables the monitoring of multiple cellular functions using live cell assays. Automated image analysis is necessary to monitor such model systems in a high-throughput and high-content environment. Here we demonstrate the ability to simultaneously track cell cycle phase and cell motion at the single cell level. Using a recently introduced cell cycle marker, we present a set of image analysis tools for automated cell phase analysis of live cells over extended time periods. Our model-based approach enables the characterization of the four phases of the cell cycle G1, S, G2, and M, which enables the study of the effect of inhibitor compounds that are designed to block the replication of cancerous cells in any of the phases. We approach the tracking problem as a spatio-temporal volume segmentation task, where the 2D slices are stacked into a volume with time as the z dimension. The segmentation of the G2 and S phases is accomplished using level sets, and we designed a model-based shape/size constraint to control the evolution of the level set. Our main contribution is the design of a speed function coupled with a fast marching path planning approach for tracking cells across the G1 phase based on the appearance change of the nuclei. The viability of our approach is demonstrated by presenting quantitative results on both controls and cases in which cells are treated with a cell cycle inhibitor.

Key words: Cell cycle phase, segmentation, tracking, level sets, fast marching, path planning, model-based analysis, shape and size constraint, automated image analysis, cell cycle phase marker, high throughput, high content, confocal fluorescence imaging

* Corresponding author. Phone: 1-518-387-4149. Fax: 1-518-387-6981.
Email addresses: padfield@research.ge.com (Dirk Padfield), rittsche@crd.ge.com (Jens Rittscher), Nick.Thomas@ge.com (Nick Thomas), roysam@ecse.rpi.edu (Badrinath Roysam).
1 Introduction

The development of automatic methods for studying cell proliferation and viability in cell populations is a challenging biomedical image analysis problem. Building on a rich body of visual tracking research, a number of very promising cell tracking algorithms have been presented. These algorithms automatically determine biologically relevant measurements such as cell motility and the rate of mitosis or apoptosis. The field of cell cycle control and its links with cancer and biological responses to DNA damage represent one of the most dynamic areas of contemporary biomedical research [1]. The focus of this paper is to enable the study of the cell cycle in greater detail by automatically extracting information about the four phases of each cell. We present an approach that makes use of a particular molecular marker to extract such information.

1.1 Cell Cycle Phase

When eukaryotic cells replicate, they pass through a tightly regulated series of events known as the cell cycle. The cell cycle is comprised of four phases, G1, S, G2, and M, each of which is marked by distinctive characteristics as the DNA is replicated and the cell splits into two daughter cells. In G1, the cell carries out any necessary DNA repair and prepares for S phase. In S, it unwinds its DNA strand, replicates its DNA, and rewinds the strand. In G2, all of the other elements of the cells are replicated. Collectively, the G1, S, and G2 phases comprise interphase. The M phase is a mechanical process broken into prophase, metaphase, anaphase, telophase, and finally cytokinesis during which the nuclear envelope breaks down, chromosomes are separated, the nuclear walls of daughter nuclei are formed, and finally cytoplasmic components are segregated to complete the formation of two identical daughter cells.

Cell cycle checkpoints exist at specific points in the cell cycle in eukaryotic cells to prevent them from progressing to the next phase of the cell cycle in the event of DNA damage or other conditions that would make cell division dangerous for the cell [2]. Study of the G1, S, and G2 phases is necessary for diagnosis and treatment strategies that target the cell cycle checkpoints. Discoveries related to the study of the eukaryotic cell cycle have the potential to be used in the treatment of cancer, and the cell cycle mechanisms can serve as targets in drug discovery [1].

Blocking a cell from passing through any of its phases will stop the cell from replicating. This is important in the study and treatment of cancer since it is desirable to find ways to block the replication of cancer cells without perturbing normal cells. Many compounds could potentially meet these requirements, and it is necessary to study their effectiveness and the mechanisms by which they operate. In addition,
Cytokinesis
(Cell Division)
Mitosis
(Nuclear Division)
DNA Replication

Fig. 1. Phases of the cell cycle indicated by a cell cycle phase marker. The figure shows a schematic of the cell cycle along with a representation of the relative brightness of the nucleus (inner oval) and cytoplasm (outer oval) in each phase. N represents the nuclear brightness, and C represents the cytoplasm brightness. The surrounding four images show real examples of each of these phases, where the bottom two both show the two daughter cells resulting from mitosis. The phases are color-coded as G1 (cyan), S (green), G2 (purple), and M (red). This color-coding scheme is used throughout the paper.

meaningful representations of the data need to be extracted to facilitate the effective screening of different cell cycle inhibitors.

Image analysis of fluorescent cellular images typically requires staining with a fluorescent dye to identify cell nuclei. This requirement presents a problem for long term cellular imaging since fluorescent DNA binding dyes have toxic side effects. An alternative method to mark nuclei and keep cells alive through more than one cell cycle is to engineer the cell to express a fluorescent protein such as Green Fluorescent Protein (GFP) coupled to a nuclear localization sequence or as a histone protein so the protein is resident in the nuclei of all cells as was done, for example, by Kanda et al. [3]. Although the nuclei are marked, nuclear fluorescence is not related to DNA content and hence the G1, S, and G2 phases of interphase cannot be distinguished (as is the case for DNA binding dyes), and the various sub-phases of mitosis can only be distinguished by morphological analysis. For the study of cell cycle inhibitors including, for example, Roscovitine and Nocodazole, which block the cell in the cell checkpoints in the G1 and G2 phase, respectively, it is necessary to image those phases as well.

In this work, a non-toxic dynamic Cell Cycle Phase Marker (CCPM) [4,5] is used that follows the cell cycle progression in living cells based on expression of a fluorescent protein linked to key cell cycle control elements. A CCPM enables imaging live cell assays for time periods lasting as long as several days. Sensor distribution through the cell cycle for a CCPM dataset is illustrated in Figure 1, where a cell is shown in each of the phases. To date, this cell cycle sensor is the only available technology that enables the identification of the four main phases of the cell cycle for live cells. It is currently the only method for studying the effect in live cells of inhibitor compounds that block the replication of cancer cells by stopping them
Fig. 2. **Representation of cell cycle phase as a 3D color-coded tree.** A normal cell cycle is shown on the left, and an inhibited cell cycle where the cell is arrested in mitosis (red) is shown on the right. The figures are schematic representations of the phases that the cell undergoes over time, where the colors correspond to the phases shown at the right of each figure.

in one of the phases of the cell cycle. Thus, the analysis of this type of data can contribute significantly to the study of cell cycle inhibitors and the diagnosis and treatment of cancer.

1.2 **Problem Statement**

Each CCPM cell assay well (dish imaged by a microscope) has several hundred cells, and they need to be monitored several times per hour over several days. In addition, for each compound studied, several wells need to be prepared to give statistically significant results. When the number of compounds is large, it becomes evident that manually outlining every cell in each well for all time frames for many compounds is intractable. Therefore, it is necessary to employ automatic analysis algorithms.

Several factors make this problem particularly challenging. The cell cycle phase marker changes the appearance of each individual cell as a function of the cell phase as illustrated in Figure 1. A large number of cells can be packed very densely, which makes even visual tracking difficult. Additional challenges include changes in staining intensity over time and varying cell velocity.

Analysis methods should be capable of automatically segmenting the various phases of the nuclei and generating simple and meaningful representations of the results such as that shown schematically in Figure 2. The images show the cell splitting over time, where time is the vertical axis and the phases are color-coded according to the color scheme in Figure 1.
Fig. 3. **Segmentation of nuclear tunnels in a 3D spatio-temporal volume.** The 97 images of size $1280 \times 1280$ were taken 30 minutes apart. The G2 and S phase nuclei can be seen as dark tunnels that carve through the volume. The algorithms first segment the G2, S, and M phases and then link across the G1 phase.

### 1.3 Approach

Since all temporal information is available at the time of analysis, we formulate this problem as a spatio-temporal volume segmentation task as opposed to a 2D tracking task. The 2D images taken over time are stacked to form a 3D spatio-temporal volume as shown in the left image of Figure 3. One particular advantage of this approach is that different phases can be segmented independently in a 3D volume and then linked together to form 3D tracks. It also enables the possibility of dividing the segmentation task into different steps, which is a key aspect of our approach. The appearance of each cell depends on its phase in the cell cycle, and, as a result, the difficulty of the segmentation task varies drastically. During the G2 and S phases, nuclei carve our dark “tunnels” of the 3D volume; the nuclei are dark while the surrounding cytoplasm is relatively bright. The M phase cells are almost uniformly bright. The G1 phase is particularly difficult to segment because the fluorescence is evenly distributed between the nucleus and the cytoplasm, and no clear boundaries are present.

We first apply segmentation techniques to extract all cells in the G2 and S phases, separately segment the M phase cells, and then use a track linking approach to track the cells through the G1 phase. Our main contribution is the design of a model-based speed function coupled with a fast marching path planning approach that enables the tracking of cells across the difficult to segment G1 phase. To constrain the evolution of a level set for segmenting the S and G2 phase cells, we designed a shape and size constraint that exploits the location of the seeds and models the cell characteristics. We also introduce a seed placement algorithm based on a model of S and G2 nuclei. These seeds are used as inputs to the level set segmentation algorithm for segmenting the nuclei. This approach is visualized in the right image of Figure 3.
In Section 2, we describe some of the related work. In Section 3, we present our approach to the problem of segmenting and tracking cells in CCPM datasets. Our results are given in Section 4, and we state our conclusions in Section 5.

2 Related Work

Some of the basic biological processes studied by previous methods include cell growth, fertilization, embryonic development, tissue repair and wound healing, differentiation, immune response, cancer metastasis, chemotaxis, and improving drug discovery. These goals are met through characterizing cell motility, migration, deformation, cell population dynamics, and other such measures. However, image analysis tools have not previously been developed for CCPM datasets, and thus tools are lacking for measuring the G1, S, and G2 phases for live cells. Most previous methods rely on a nuclear fluorescent stain, which highlights the nucleus to be segmented and simplifies the segmentation process.

Applying commonly known visual tracking methods [6] to CCPM datasets would require a stochastic model that has the power to predict the changes of the phases. Standard tracking approaches generally assume that either the appearance of the objects or the trajectory are not changing. In the case of CCPM images, both the appearance and the trajectory of the objects change with time. Therefore, it is difficult to approach the problem using standard tracking techniques.

Many researchers have used level sets for segmenting images of cells. Solorzano et al. [7] use competing 2D level sets to segment multiple cells. Mukherjee, Ray, and Acton [8] use 2D level sets to track leukocyte cells in transillumination images over time, and, for validating these tracks, Ray and Acton [9] use 2D level sets in 2D spatio-temporal images, which are defined by Sato et al. in [10]. Feghali [11] directly uses 3D level sets for segmenting the tracks in 2D slices taken over time as a spatio-temporal volume (although not for the application of cell segmentation). 3D level sets are also used by Sarti et al. [12], in this case applied to 3D volumes of nuclei in confocal microscopy images.

Coupled level sets and level sets based on shapes also present powerful segmentation approaches. Zeng et al. [13] present a method of coupled surface propagation via level set methods, which can capture a representation of the two bounding surfaces leading to automatic segmentation. Zhao et al. [14] use multiple level set functions with a coupling term to segment multiple objects, which prevents regions from overlapping and the creation of vacuums. Leventon et al. [15] evolve the level set with the assumption that the structure has a shape that belongs to a shape distribution function, which is computed from a given set of manually generated training instances. This enables the incorporation of shape constraints in the segmentation process.
Existing cell tracking methods can roughly be divided into two main classes. The first is composed of independent frame cell detection followed by linking based on minimizing the distances between detected cells according to some criteria [16,17,18,19]. The second is based on first detecting the cells in one frame and then evolving them in time to keep track of cell movements and deformations [20,21,22]. Some approaches combine both detection and contour evolution methods [23,24,25,26]. Many cell tracking approaches are not concerned with the actual segmentation of the cells, only the tracking. As such, although the cell is identified and tracked, the actual cell border is not accurately segmented. The cell segmentation, however, can yield useful information about the change in cell size, shape, and morphology that can lead to useful biological insights.


Identifying the various phases of individual cells is a task that has also been addressed using automatic analysis algorithms. Classifying images of live cells into the four phases has not previously been attempted, but researchers have developed analysis algorithms for studying the phases of mitosis that are characterized by different morphological properties. Harder et al. [29] classify cells with fluorescently marked nuclei into interphase and several phases of mitosis using feature extraction and feature classification. Chen et al. [30] and Yang et al. [31] segment and track large numbers of cells marked with a nuclear stain and classify them into several stages of mitosis.

The related literature demonstrates that a wide range of effective approaches have been applied to the problem of cell segmentation and tracking. We expand upon the state of the art by approaching the individual study of the four phases of the cell cycle.
Fig. 4. **Ring kernel for one scale and example G2 cell.** The ring kernel approximates the appearance of the cell and aids in the detection of the cell center. The detections at various kernel scales are combined to detect cells of different sizes. Note that the two images are not shown on the same scale.

3 Spatio-Temporal Cell Cycle Phase Analysis

In order to automatically generate a meaningful description of the cell cycle in images with a CCPM, it is necessary to track nuclei over time. We have developed a set of algorithms for automatically segmenting and tracking cells in CCPM datasets. The images are first pre-processed using a smoothing algorithm [32] and employing flat-field correction, which iteratively fits a quadratic surface to the image similar to [33,34,35,36]. Using the pre-processed images, we introduce a seed placement and classification method, a shape and size constraint for level set evolution, and a tracking method based on fast marching path planning. Many of the algorithms were developed using algorithms from the Insight Toolkit [37].

3.1 Automatic Seed Placement

In order to evolve a level set, initial seeds need first to be defined. We take a two-step approach to seed placement within nuclei: multi-resolution normalized cross correlation with a ring-like kernel followed by point classification.

We designed a set of ring kernels of different sizes that are positive inside the ring band, and zero inside and outside the band. To illustrate, an example kernel for one size is given in Figure 4(a), and an example G2 cell is given in Figure 4(b). This yields high correlation values for dark nuclei surrounded by bright cytoplasm. We run the correlation step with different sized kernels to detect various nucleus sizes. Non-maximum suppression of the correlation image can thus effectively detect nuclei.
Fig. 5. Positive and negative edges under a mask around the nucleus. The left figure shows the image from Figure 4(b) with the edges in the region overlaid. The middle figure shows the magnitude of the edges, with brighter values representing larger magnitudes. The right image shows the positive curvature edges in white, the negative curvature edges in black, and the zero curvature edges in gray. It is clear from this illustration that the cell has more positive curvature edges than negative and with larger magnitude.

This step also yields several detections in the background since background segments surrounded by the cytoplasm of nearby cells look similar to nuclei. To reduce the number of detections in the background, the seed placement step is followed by a seed classification step using a trained classifier. Three features of the seed neighborhood are particularly effective at discriminating between nuclei and background: 1) the image intensity, 2) the number of edges with positive curvature, and 3) the gradient magnitude of those edges. To measure the latter two, the Canny edges are separated into those having positive curvature, negative curvature, and zero curvature, where curvature $\kappa$ is defined as the divergence of the normalized image gradient

$$\kappa = \nabla \cdot \frac{\nabla f}{|\nabla f|} = \frac{f_{xx}f_y^2 - 2f_xf_yf_{xy} + f_{yy}f_x^2}{(f_x^2 + f_y^2)^{3/2}}$$

where $f$ is the image, $f_x$, $f_y$ are the first derivatives, and $f_{xx}$ and $f_{yy}$ are the second derivatives. These measures are good discriminators because the nuclei are generally brighter than the background, and the nuclei are surrounded by many strong edges with positive curvature. On the other hand, edges around the background are generally fewer in number and magnitude. Figure 5 shows a visual representation of the classification using the edges around a nucleus.

A Fisher linear discriminant [38] classifier was found to be sufficient for separating the seed points into the classes of nucleus versus background. A training phase is employed to determine what feature values characterize a cell using a set of seed points that are manually classified into the categories of nucleus or background from a representative image. Figure 6 demonstrates the classification of manually placed seeds. Around each training point, a region of interest representative of the typical size of a nucleus is considered, and the three features are measured. Using the classification surface, the automatically generated seeds from the normalized
Fig. 6. **Manual training points.** Each of these points was manually classified into either dark nucleus (green), bright nucleus (yellow), or background (red).

Fig. 7. **Example result of the seed classification step.** These seeds for G2 phase nuclei were automatically placed using the method of normalized correlation followed by seed classification. These seeds serve as the initial level set for the level set segmentation of the nuclei. Most of the nuclei contain seeds and that there are few seeds in the background.

cross-correlation step for testing images are classified. An example result is shown in Figure 7, where the detected seeds are shown as bright red dots overlayed on the cell image. The signed distance of these seeds are used as the initial level set for the segmentation step.
3.2 Shape/Size Constraint for Level Set Segmentation

Using the pre-processed images and the seeds, we segment the cells using an active contour model solved with level sets based on edges. Level sets can include several energy terms such as propagation terms that move the curve outwards, curvature terms that impose regularization, and advection terms that attract the curve to edges; these terms are minimized by the contour evolution. One challenge with edge-based level sets is that, if the contour leaks out of weak-edge nuclei, it will proceed rapidly in the background, and, if allowed to propagate, the contours from these weak-edge nuclei can eventually join with other nuclei and result in unusable segmentations. One approach to stop the leaking is to increase the attraction to edges. However, the constantly changing intensity profile of the dynamic cell cycle sensor frequently yields gradients inside the nucleus, so increasing an edge-attracting force can cause the level set to become stuck in local minima inside of the nucleus. In addition, there is often no edge present at the border of the nucleus. An alternative is the implementation of a size constraint, but size alone is not sufficient since the resulting constrained level set will not necessarily correspond to the shape of the cell. Finally, another problem exists with respect to the curvature term. When starting from small seeds, the curvature term has a tendency to dominate because the curvature measure is relatively high. This can cause the level sets to shrink to a point and disappear if the propagation term is not large enough. As the level set grows, this is less of an issue since the curvature term is weakened by the smaller curvature measures.

We have developed a shape/size constraint that addresses these problems by constraining the evolution of the level set based on a model of the cell. Next, we describe the level set formulation followed by our shape/size constraint.

Since original work by Kass [39], extensive research has been carried out on “snakes” or active contour models for boundary detection. The classical approach is based on deforming an initial contour $C_0$ towards the boundary of the object to be detected. The deformation is obtained by minimizing a functional designed so that its minimum is obtained at the boundary of the object. The classical snakes approach associates the curve $C$ with an energy given by

$$E(C) = \alpha \int_0^1 |C'(q)|^2 dq + \beta \int_0^1 |C''(q)|^2 dq - \lambda \int_0^1 |\nabla I(C(q))| dq$$

where $\alpha$, $\beta$, and $\lambda$ are real positive constants. The first two terms control the smoothness of the contours to be detected, and the third term is responsible for attracting the contour towards the object in the image (external energy). This approach is non-intrinsic since the energy depends on the parameterization of the curve and is not directly related to the object geometry. The model is also not capable of easily handling changes in topology.
To avoid the limitations inherent in the snakes model, level sets can be used, which give an implicit representation of the curve. The main advantages of using level sets are that arbitrarily complex shapes can be modeled and topological changes such as merging and splitting are handled implicitly. They provide a powerful mathematical framework for image segmentation based on a balance of curve-based and image-based forces. Level set methods were first described in [40], and the theory and many examples of applications are given in [41,42]. They were proposed for segmentation in [43], [44], and [45] leading to the geodesic active contour equation formulated in [45] for the update of the level set function

$$\frac{\partial \phi}{\partial t} = -g |\nabla \phi| + \gamma g \kappa |\nabla \phi| - \eta \nabla g \cdot \nabla \phi$$

(3)

The first term on the right acts as a propagation term, the second as a regularization term, and the third as an edge attracting term. The scalars $\gamma$ and $\eta$ give relative weights to the terms of the equation. $\kappa$ is the mean curvature as defined in Equation 1 with $f$ replaced by $\phi$. The function $g$ is a monotonically decreasing function from 1 to 0 of the image gradient. We set the function $g$ based on a sigmoid function to give control over the mapping of the edge strength.

$$g(|\nabla I|) = \frac{1}{1 + e^{-\frac{|\nabla G_{\sigma} * I| - \beta_e}{\alpha_e}}}$$

(4)

Here $|\nabla G_{\sigma} * I|$ represents the gradient magnitude of the convolution of the image $I$ with a Gaussian function parameterized by $\sigma$. The variable $\alpha_e$ controls the slope of the curve and should be between 0 and $\infty$ to map the low gradient magnitudes to high speeds and vice versa. The variable $\beta_e$ shifts the curve left to right and should be between the min and max of $|\nabla G_{\sigma} * I|$. The $e$ subscripts of $\alpha_e$ and $\beta_e$ signify that these parameters are related to edges.

These level sets have a tendency to leak in the presence of low-contrast edges, so we developed a shape/size constraint to control this. Figure 8 illustrates that weak nuclei gradients are particularly problematic for contour leaking. As opposed to using a functional with fixed parametrization, we incorporate a probabilistic prior based on the expected cell size. The seed placement step ensures that the seed points are placed near the medial axis of a cell (see Figure 7). Based on this observation, we can construct a model of the location of the nuclear edge from a mapping of the distance function from the seeds that will affect the speed of the level set function. This mapping should give positive speed for distances less than the expected cell size and negative speed for values outside.

We designed a function mapping the distance to the seeds $\tau$ to the speed of the
Fig. 8. **Examples of leaking and constrained level sets and shape/size function.** The blue contours indicate the seeds, and the red contours indicate the evolving level sets. In the image on the left several contours initialized in nuclei with low gradient edges such as that in the middle left grow without bound and eventually merge with the contours of other nuclei. The figure in the middle shows results using the shape/size constraint. In this case, the contours inside nuclei with strong gradient edges are still able to move to these edges while those in nuclei with weak gradient edges are constrained not to leak into the background. The figure on the right shows the constraint function. Since this distance function is based on the medial axis characteristic of the seeds and the expected cell size, it takes into account both size and shape. The function yields a positive speed close to the seeds, a negative speed far away, and a speed of zero at the distance corresponding to the expected cell radius.

The function $s(\tau)$ that results in smoothly varying speed values,

$$s(\tau) = 2 \left( \frac{1}{1 + \exp \left( -\frac{\tau - \beta_s}{\alpha_s} \right)} \right) - 1. \quad (5)$$

Here $s(\tau)$ is the speed that depends on the distance $\tau$, $\beta_s$ is related to the expected cell size, and $\alpha_s$ controls the slope of the curve. The $s$ subscripts of $\alpha_s$ and $\beta_s$ signify that these parameters are related to the speed.

The scalar multiplication and subtraction factors lead to a function that yields speeds from $s(0) = 1$ to $s(\infty) = -1$ as shown on the right in Figure 8. This ensures that the function evolves with a positive speed inside of the cell and a negative speed outside of the cell. The parameters $\alpha_s$ and $\beta_s$ should be set to ensure that 1) the speed is 1 at the center of the cell and 2) the speed is 0 at the expected cell radius. Therefore, we use the following constraints to solve for these parameters

$$s(0) = 1 \approx 1 - \varepsilon \quad (6)$$
$$s(R) = 0 \approx \varepsilon \quad (7)$$

where $R$ is the expected cell radius and $\varepsilon = 2e^{-16}$ is a small constant used to avoid numerical instability and computer round-off error.
Setting $\tau = 0$, Equation 5 becomes

$$s(0) = \frac{2}{1 + \exp\left(\frac{\beta_s}{\alpha_s}\right)} - 1. \quad (8)$$

Setting the right-hand side of this equation to the right-hand size of the constraint in Equation 6, and solving for $\alpha_s$ in terms of $\beta_s$, we find that

$$\alpha_s \approx \frac{\beta_s}{\ln(\varepsilon) - \ln(2)}. \quad (9)$$

Setting $\tau = R$, Equation 5 becomes

$$s(R) = \frac{2}{1 + \exp\left(-\frac{R - \beta_s}{\alpha_s}\right)} - 1 \quad (10)$$

Setting the right-hand side of this equation to the right-hand side of the constraint in Equation 7, and solving for $\beta_s$ in terms of $\alpha_s$ and $R$, we find that

$$\beta_s \approx R + \alpha_s \ln(1 - \varepsilon). \quad (11)$$

Finally, substituting $\alpha_s$ from Equation 9 in Equation 11, we solve for $\beta_s$ and $\alpha_s$ in terms of $R$

$$\beta_s \approx R$$

$$\alpha_s \approx \frac{R}{\ln(\varepsilon) - \ln(2)} \approx -0.0272R. \quad (12)$$

From these results, we see that $\beta_s$ should be set to the expected cell radius and $\alpha_s$ to $-0.0272$ of the cell radius.

Note that the function in Equation 5 is calculated only once for every pixel in the image. Thus, rather than updating the model at every iteration as some other methods do, this method is computationally efficient.

The $s(\tau)$ function can be applied to the propagation term of the level set equation to slow down this term as the level set expands. To incorporate this constraint into the level set equation, we balance it against the $g$ edge function that is already multiplied by the propagation term. The level set equation updated from Equation 3 is thus represented as

$$\frac{\partial \phi}{\partial t} = -\left(\lambda g + (1 - \lambda)s(\tau)\right)|\nabla \phi| + \gamma g \kappa |\nabla \phi| - \eta \nabla g \cdot \nabla \phi \quad (13)$$
where we use a variable $\lambda$ that controls the relative weight of these edge and distance forces.

Since the curvature and advection terms have not been modified, they are still free to continue to regularize the curve and attract it to edges, respectively. Thus, in the presence of strong gradients, the advection force will take over as the level set evolves and lock onto the edges as before. However, in the presence of weak edges, the shape/size constraint will keep the level set from growing without bound. The constraint also addresses the problem of small seeds disappearing due to the curvature term. Since the constraint is applied to the propagation term and decreases with distance from the seeds, it can overcome the curvature term when the seeds are small and gradually decrease to balance the decreasing curvature strength.

The image in the middle of Figure 8 shows the same level set evolution as on the left but using the shape/size constraint. In this case, the level sets inside nuclei with strong gradients are still able to move to these edges while those in nuclei with weak gradients are constrained not to leak into the background. The evolution has already reached steady state and stopped at the configuration shown in this figure as opposed to the evolution on the left, which continues to evolve outside of the cells.

The result of the level set segmentation step is a set of 3D tubes corresponding to the S and G2 phase of the cells. These cells can then be associated with the corresponding M phase cells, which are round and bright and can be segmented using hysteresis thresholding combined with marker-controlled watershed segmentation. To obtain a full 3D tree structure, a tracking step is needed to connect across the G1 phase.

### 3.3 Model-Based Geodesic Cell Phase Tracking

We here introduce a novel method for tracking across the G1 phase of the cell cycle using the segmentations of the G2, S, and M phases. Our approach uses the fast marching algorithm as a path planning tool where we define a speed map based on a model of the phase transition. Our model is based on the appearance change of the nuclei from internally bright in the M phase to gradually dark at the end of the S phase and into the G2 phase (see Figure 1). This can be viewed as a path planning problem where the cost function is defined to vary according to the model, and it can be solved using the fast marching method [46]. The fast marching method, described in [41], is a method for solving a time arrival task described by the Eikonal equation

$$ | \nabla T | F = \sqrt{\left(\frac{\partial T}{\partial x} \right)^2 + \left(\frac{\partial T}{\partial y} \right)^2 + \left(\frac{\partial T}{\partial z} \right)^2} F(x, y, z) = 1 $$

(14)
in 3D where \( F(x, y, z) \) is the speed map or cost function and \( T \) is the resulting time-crossing map for every point in the domain. Given a starting point, the fast marching method builds the solution outwards throughout the domain choosing the smallest time at each step of its evolution. Then, given any point in the domain, the shortest path is constructed by propagating from that point to the fast marching starting point using gradient descent by solving the ordinary differential equation \( \frac{dx}{dt} = -\nabla T \).

The formulation of an appropriate cost function \( F \) plays a central role in the path-planning problem. In our model, the cost function switches from emphasizing bright nuclei to dark nuclei along the tracks. In other words, the cost is low (fast moving) in bright nuclei and high in dark areas at the beginning of the track and gradually switches to a low cost for dark nuclei and high cost for bright areas. Because the function should range between 0 and 1, we define it by

\[
F(x, y, n) = f(I(x, y, n)) = \frac{1}{1 + \exp\left(-\frac{I(x, y, n) - \beta}{\alpha(n)}\right)}
\]

where the speed function \( F \) is a function of the intensity \( I(x, y, n) \) at a given image point \((x, y, n)\), and \( z \) has been substituted with \( n \) to represent the \( z \) distance relative to the starting point. \( \beta \) controls the shift of the function, which can be set experimentally and remain fixed. \( \alpha(n) \) controls the slope of the curve, which varies depending on the distance from the starting point. Changing the sign of \( \alpha \) mirrors the function around the horizontal axis at \( 1/2 \). To generate a cost function to move quickly in bright nuclei, \( \alpha \) should be positive, to move quickly in dark nuclei, it should be negative, and it should change gradually between these stages.

To generate a smooth transition from bright to dark nuclei, we solve for \( \alpha \) for varying speed given a fixed \( \beta \) and a fixed starting intensity. The speed should vary in equal increments between 0 and 1, and the number of steps \( N \) depends on the expected amount of time taken for the transition. The fixed minimum intensity \( I_t \) is set to be the lowest intensity corresponding to a cell, or, equivalently, the background threshold. Solving for \( \alpha \) in Equation 15 for a speed varying from 0 to 1 gives

\[
\alpha(n) = -\frac{I_t - \beta}{\ln\left(\frac{1}{F_t(n)} - 1\right)}
\]

\[
n = 0, ..., N - 1
\]

\[
F_t(0) = \epsilon, F_t(N - 1) = 1 - \epsilon
\]

Here \( F_t(n) \) is the speed corresponding to the minimum intensity \( I_t \) at position \( n \), and \( \epsilon \) is a small number that avoids computational errors at speeds of 0 and 1.

To implement this model, a starting point is chosen at the end of the M phase of
Fig. 9. **Changing α to give speeds that vary in equal increments.** Each curve represents a function from Equation 15 generated using a different α. The αs in Equation 16 are solved to give speeds that vary in equal increments.

Table 1

**Numerical example of solution of α.** In this example, α is solved according to equation 16 using the following values: β = 300, It = 200, N = 15, and ε = 2e^{-16}.

<table>
<thead>
<tr>
<th>n</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td>α</td>
<td>0</td>
<td>39</td>
<td>56</td>
<td>77</td>
<td>109</td>
<td>170</td>
<td>348</td>
<td>∞,−∞</td>
<td>-348</td>
<td>-170</td>
<td>-109</td>
<td>-77</td>
<td>-56</td>
<td>-39</td>
<td>0</td>
</tr>
</tbody>
</table>

In this example, each parent cell. A time-dimension distance function is generated from this point, where all points in the same x-y plane as the starting point have distance zero, all those at the next time plane have distance 1, and so on for the entire volume. These distances correspond to n in Equation 16, and an α can be calculated for each one (note that if the distance is larger than N − 1, the α(N − 1) value is used). Using these α values that vary throughout the volume along with the intensity at each voxel, Equation 15 can be used to calculate the speed mapping for each point in the image. The result of changing α in this way is illustrated in Figure 9.

Table 1 gives a numerical example of solving α for varying speed given a fixed β and starting intensity It. For this example, β = 300, It = 200, N = 15, and ε = 2e^{-16}.

In Figure 10, some example original images are shown in the top row, and images of the resulting speed function are shown in the bottom row. These cost functions are used for the fast marching algorithm.

This fast marching tracking algorithm is run on all parent cells to find their daughter cells. The overall algorithm is given in Algorithm 1. Note that for each parent, the fast marching algorithm is run for the entire domain, which can be computationally expensive if the volume is large and there are many parent cells. We obtain a significant speedup by enabling the setting of targets which, once reached, cause the fast marching algorithm to terminate. The center of the cell on the earliest time slice of the daughter cells are specified as the target points. Once two of the targets have been reached, the algorithm terminates since it has found the daughters.
\[ \alpha(0) = 0 \quad \alpha(3) = 77 \quad \alpha(6) = 348 \quad \alpha(8) = -348 \quad \alpha(11) = -77 \quad \alpha(14) = 0 \]

Fig. 10. **Speed images for the fast marching algorithm.** The top row shows the original images, and the bottom row shows the cost function calculated for each image. The cost function is built separately for each parent cell with the end of the M phase as the starting point. In this example, the tracked cell is the circled one undergoing mitosis. With increasing n, or distance, from the starting point, the \( \alpha \) increases to move the speed from 0 towards \( \frac{1}{2} \) and then flips sign and decreases as the speed moves towards 1. A nonlinear \( \alpha \) is required to give a smooth speed transition in equal increments according to Equation 16. This gives more speed to bright nuclei in the beginning and changes to gradually more speed for dark nuclei towards the end, where brighter parts of the cost function image indicate higher speeds.

**Algorithm 1** Fast Marching Tracking Algorithm.

1: for each parent cell do
2: Using Equations 15 and 16, find the speed image \( F(x, y, n) \) from the parent cell.
3: Solve \( |\nabla T| F = 1 \) to find \( T \).
4: Choose as the two daughters the two targets that have the smallest \( T \).
5: Extract the paths using gradient descent \( \frac{dx}{dt} = -\nabla T \) from the daughters to the parent.
6: end for

4 Results

The algorithms were tested on 22 sets of series of 97 images of size 1280 x 1280 with pixels representing 750\( \mu \)m x 750\( \mu \)m on the specimen. The algorithms were run on 400 x 400 x 97 pixel regions, which were cropped out of the same location in each of the full volumes for computational efficiency. Images were acquired on an IN Cell Analyzer 3000 laser line scanning confocal imager (GE Healthcare, Milwaukee, WI) using 488nm excitation and a 535nm-545nm emission filter. The IN Cell 3000 uses a 40x objective with an effective magnification at the camera of 32x. The cells were grown in a well plate and maintained on the instrument throughout the duration of imaging under standard culture conditions (37°C / 5% CO\(_2\)). Each of the 97 images was taken 30 minutes apart for a total time of 48 hours.
Fig. 11. **3D Tree Representation of Results.** The left figure shows a tree of a cell from a control well with random colors corresponding to different cells. The middle figure shows the same tree color-coded by cell phase, and the right figure shows the tree of an inhibited cell that is arrested while trying to undergo mitosis. The various phases of the cell cycle are shown with the same color-coding scheme as in Figures 1 and 2: G1(cyan), S(green), G2(purple), M(red).

The 22 wells consist of 6 control wells and 16 wells treated with varying concentrations of cell cycle inhibitors: 8 with Roscovitine and 8 with Nocodazole. The biological expectation for the treated wells is that, for larger concentrations, the cell cycle will be arrested earlier and the cells will either die or remain in the final phase until the end of the experiment.

To compare the duration of the various cell phases, the durations were averaged across the wells. This found that the M phase in inhibited cells was 2-3 times longer than for control cells, indicating that cells had become arrested at the checkpoint. At the same time, the G1, S, and G2 phases were approximately the same duration in the inhibited datasets as in the control. In each of these experiments, there were approximately 400 cells per image. This showed that both Roscovitine and Nocodazole suppress proliferation compared to the control datasets. To visualize this, Figure 11 shows a representative control cell tree and Nocodazole cell tree. The control tree splits as usual whereas the inhibited cell attempts to split but gets arrested in mitosis. Compare this with Figure 2. Figure 12 shows a surface rendering of a set of cells color-coded by phase passing through an image plane.

To show the G2, S, and M phase cell segmentation results, Figure 13 gives samples for a cropped portion of a control dataset and a Nocodazole dataset. Here the G2, S, and M phases are effectively segmented.

To evaluate the proposed tracking approach, a number of cells were tracked using
Fig. 12. **Surface rendering of segmented cells through the image plane.** Three cells are shown undergoing mitosis, where the cell phases are color-coded as in Figure 1. Displaying the rendered cells in relation to the image plane is a useful way for biologists to visualize the cell phases.

![Surface rendering of segmented cells](image)

(a) Control dataset segmentation  
(b) Inhibited dataset segmentation

Fig. 13. **Sample segmentation results.** The figure on the left shows a sample segmentation result for a control dataset, and the figure on the right shows a sample for an inhibited dataset. Since only the G2 and M phase cells are identified in the segmentation step, only purple (G2) and red (M) phases are present.

![Sample segmentation results](image)

Both an Euclidean distance metric as well as the fast marching method. A selection of the tracking results is given in Table 2, where green indicates that the event was correctly tracked and red indicates errors. The numbers in the table indicate the arbitrary labels of the parents and daughters and help illustrate whether the daughters were correctly identified with respect to the ground truth given in the second two columns. The table shows that the fast marching method performs better than the Euclidean distance method. What follows is a detailed analysis of the table.

**Correct parent-daughter association for both methods.** Parent 1, 4, and 6 are
examples where both the Euclidean method and the fast marching method are able to correctly track the cell and link from the parent to the daughters. However, while the Euclidean distance generates a linear interpolation between the points, the fast marching method extracts a path that more accurately characterizes the track of the cell. This is shown in Figure 14, where the track followed by the Euclidean method is shown as a white circle, and the track followed by the fast marching method is shown as a red circle. The underlying cost function of the fast marching method is that shown in the bottom row of Figure 10. Because the fast marching method models cell transition, it moves quickly in the beginning to follow the bright M phase and follows much more closely the center of the cell throughout the transition. To illustrate this point further, we show spatio-temporal images [10] in Figure 15 of the tracks in the horizontal direction of these images for the rightmost tracks (the left tracks do not move very far, so they are not as illustrative). To briefly describe spatio-temporal images, the idea is for every 2D plane of the original volume to extract the line that passes through the track. The figure shows that, while both methods end up at the same place, the fast marching method, starting from the top of the spatio-temporal image, moves quickly in the bright regions and then gradually moves to the end point, whereas the Euclidean method follows a straight line between the points.

Correct parent-daughter association only through fast marching method. Parents 2 and 5 show examples where the Euclidean method fails and the fast marching method succeeds. In both these cases, a nearby cell underwent mitosis at roughly the same time, and the Euclidean method chose one of those daughters as the daughter. However, the fast marching method successfully tracked the cell.

Incorrect parent-daughter association for both methods. Parent 3 is a special case. As indicated in the “Ground Truth” column, this cell only has one daughter. This is because the cell split near the end of the time sequence, and the second daughter did not appear before the end of the experiment. Since the tracking methods search for two daughters, they report another nearby cell as the second daughter. This error can be corrected in several ways including a daughter competition method where statistical measures are employed to determine which daughters belong to which parents. Since the fast marching method also includes a time reaching component, this could be used as a measure of how likely the track is to be valid.

4.1 Algorithmic Validation

The seed placement and classification steps, which are necessary to initialize the level set segmentation, occupy an important role in the processing because seeds should be placed in the nuclei and not in the background to enable the correct nuclear segmentation. To validate these algorithms, a dataset was used that includes the CCPM along with a nuclear stain. This nuclear stain cannot be used for the time-
Fig. 14. Visual comparison of the Euclidean method versus the fast marching method.
The white circles show the track of the Euclidean method, and the red circles show the track extracted using the fast marching method. For the fast marching algorithm, the paths correspond to the calculated optimal path from the parent to the two daughters using the speed images from the bottom row of Figure 10. This example shows how the fast marching algorithm closely follows the movement of the cell center whereas the Euclidean method simply interpolates between the starting and ending points.

Fig. 15. Spatio-temporal images for the Euclidean and fast marching methods. The left image shows the spatio-temporal image [10] of the Euclidean method and the right shows the spatio-temporal image for the fast marching method of the right tracks shown in Figure 14. This representation shows that the fast marching method better tracks the actual trajectory of the cell whereas the Euclidean method simply interpolates. The parent cell is shown at the start of the track at the top of the figure, and the two daughter cells show up as the two bright regions separated by a darker background region moving vertically through the image. By the nature of spatio-temporal images, only one track can be shown per image, and these figures show the track on the right in Figure 14.
Table 2
Comparison of the Euclidean method to the fast marching method. The $P$ stands for parent cell, and the $D_1$ and $D_2$ stand for the two daughter cells. The label numbers are arbitrary and are used simply to show whether the daughters are correct. The first column gives the parent labels, the next two columns give the ground truth labels of the corresponding daughters, the next two columns give the daughters determined by the Euclidean method, and the last two columns give the daughters found by the fast marching method. Green labels indicate correctly determined daughters, and red labels indicate errors. The table shows that the fast marching method is better able to find the correct daughters, and Figures 14 and 15 give examples of how it also better captures the path of the cell. The text gives a detailed analysis of this table.

<table>
<thead>
<tr>
<th>$P$</th>
<th>$D_1$</th>
<th>$D_2$</th>
<th>$D_1$</th>
<th>$D_2$</th>
<th>$D_1$</th>
<th>$D_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9</td>
<td>10</td>
<td>9</td>
<td>10</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>11</td>
<td>12</td>
<td>11</td>
<td>13</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>3</td>
<td>14</td>
<td>-</td>
<td>14</td>
<td>15</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>17</td>
<td>16</td>
<td>17</td>
<td>16</td>
<td>17</td>
</tr>
<tr>
<td>5</td>
<td>18</td>
<td>19</td>
<td>18</td>
<td>20</td>
<td>18</td>
<td>19</td>
</tr>
<tr>
<td>6</td>
<td>21</td>
<td>22</td>
<td>21</td>
<td>22</td>
<td>21</td>
<td>22</td>
</tr>
</tbody>
</table>

Table experiments because it is toxic and kills the cells over time. But it can be used to yield a ground truth map for the location of nuclei at one time frame.

Our approach is to segment the nuclei in the nuclear channel and use this as a mask for determining whether the seeds were placed correctly in the CCPM channel. This segmentation consists of adaptive thresholding followed by watershed segmentation from the extended maxima of the binary image distance map to separate the touching nuclei. The seed placement algorithms are run on the CCPM channel as usual, and each of the seeds that were placed is compared to the nuclear mask.

To validate the seed placement and seed classification steps, images from 96 wells with nuclear stains were used. For the initial seeds and the classified seeds, the accuracy score was calculated as the ratio of seeds placed in nuclei to all seeds, and the classification step demonstrated an increase in accuracy from 20% to 60%. This number indicates that some seeds are still placed in the background, but many such seeds are removed by the segmentation evolution since seeds in the nuclei are merged together due to spatial connectivity, while those in the background tend to shrink. This number also appears low because the space occupied by the nuclei is much less than the background. It does, however, provide a quantitative measure that can be used to improve the seed placement since new algorithms can be easily tested and compared against previous methods.

To validate the tracks and phases, a system was developed for facilitating manual
generation of ground truth data. This system loads a series of 2D images taken over time and allows the user to track a particular cell through mitosis by marking the centroid of the cell on each image by its phase (G1, S, G2, or M). Each cell cycle is sequentially numbered beginning at the G1 stage after mitosis, and the parent and child relationships are stored for each tracked cell cycle. This ground truth system was used to generate tracks and phases for over 20 cell cycles, and the accuracy of this ground truth data was confirmed by an expert biologist. The automatic tracking results were automatically compared with the manual results, and the cells demonstrated correct phase tracking, meaning that the parents were associated with the correct daughters. The next step is to validate the boundaries of the segmentation of the cells and generate ground truth for more cells to further validate the automatic results.

4.2 Failure Modes

Here we present some of the failure modes that we observed along with suggestions for addressing them. Figure 16 shows image sequences representing three different failure modes. The first row shows a failure resulting from occlusion by another cell. This can be corrected by merging close tracks in a post-processing step. The second row demonstrates two cells that are merged because they move too close. This can also be addressed in a post-processing step by measuring the shape of the tracks and separating tracks that merge. The last row shows a track that is lost because the cell appears to prepare for mitosis and then relapses back into the G2 phase. This may not be a failure since it is a result of abnormal biological behavior, but it could be tracked and then measured by also linking across such states. These types of phenomena are also of interest to biologists because they represent abnormal behavior that is not usually seen; automatic methods could identify such anomalies and enable the biologists to more easily find such regions of interest.

Note that since the tracking step seeks to link together parent cells with their daughters, segmentation failures resulting in missed objects or extra objects will lead to incorrect linking of those cells. As the segmentation algorithm improves, this will also lead to fewer tracking errors. However, the fast marching algorithm finds the optimal path taken by the cells and finds the best linking from parents to daughters, which reduces the influence of segmentation errors.

5 Conclusions and Future Work

This paper introduces a framework for tracking cells in space and time in conjunction with measuring the output from a Cell Cycle Phase Marker using segmentation and tracking methods. These methods facilitate the automatic analysis of the phases
Fig. 16. **Failure Modes.** The first row represents a track lost due to occlusion between the second and third frame. The second row shows two cells that merge because they move too close together in the fourth frame. The last row shows a track lost in frames 3 and 4 because the cell moves towards mitosis and then moves back again into the G2 phase. It can be argued that this is not a failure mode since it is a result of abnormal biological behavior. In all of the figures, only the cells of interest are marked to make it easier to see the failure modes.

of the cell cycle over extended periods of time without manipulating the genetic makeup of the cells.

We introduced a speed function that models the appearance change of the cells through the phases coupled with the use of fast marching for tracking that enables accurate tracking of the parent cells to daughter cells and closely follows the path of the cell. The experimental results indicate that this model-based tracking method holds the promise of reliably tracking cells through mitosis events. We introduced a model-based shape/size constraint for the level set segmentation step that avoids the problem of leaking as well as methods for automatic seed placement. We also introduced methods for validation and visualization of the data for effective analysis of the results and use by biologists.

These methods enable the automatic computer analysis of cell cycle for many datasets over extended periods of time without necessitating the staining of the nucleus. Abstracting the type of information in Figures 2 and 11 from time-lapse image sequences could give researchers great insight into what is happening to cells when they are treated with a drug or when a gene is knocked out or over-expressed. This representation of data would allow them to determine many aspects of the cell characteristics such as whether particular cell cycle phases were extended or shortened, whether this effect was the same or different across the cell population, whether shortening of one phase was compensated for by extension in another or not, whether cells were more or less motile during these changes, and whether mitosis occurred correctly and on time or not.

Future work includes using the output of these methods to measure such quan-
titative cell characteristics on large control and toxicological datasets to generate statistically meaningful results that correspond with and yield insight into biologically relevant phenomena. Since the manual validation of these algorithms is cumbersome, an edit-based validation framework is being developed that will provide a semi-automatic method of analyzing the results. In addition, we plan to refine the breakdown of the tracks into all of the phases using Markov Chains with an ergodic model of the 4 phases where the probabilities of transitioning from one to the next will represent the expected duration of the cell in each phase as given by training data.

6 Acknowledgments

We would like to thank Simon Stubbs for his development of the Cell Cycle Phase Marker with Nick Thomas. We would also like to thank Daniel Renzi for his assistance with understanding the theory and methods of level sets and their application to cellular images.

References


