

SHAPE STATISTICS FOR CELL DIVISION DETECTION IN TIME-LAPSE VIDEOS OF EARLY MOUSE EMBRYO

M. Cicconet, K. Gunsalus*

D. Geiger, M. Werman[†]

Center for Genomics and Systems Biology
New York University

Courant Institute of Mathematical Sciences
New York University

ABSTRACT

We describe a statistical approach to the problem of estimating the times of cell-division cycles in time-lapse movies of early mouse embryos. Our method is based on the likelihoods for cells of certain radii ranges to be in each frame—without actually locating or counting the cells. Computing the likelihoods consists of a voting scheme where votes come from quadruples of points in a way similar to the first step of the Randomized Hough Transform for ellipse detection. To locate divisions, we search for points of abrupt change in the matrix of likelihoods (built for all frames), and pick the two optimal division points using a dynamic programming algorithm. Our results for the first and second cell division cycles differ less than two frames from the medians of the annotated times in a database of 100 annotated videos, and outperform two other recent methods in the same set.

Index Terms— shape statistics, division detection, time lapse, video analysis, mouse embryo

1. INTRODUCTION

Several papers have been published in recent years on the topic of time-lapse imaging of early embryos [1, 2, 3, 4, 5, 6, 7, 8]. In the case *in vitro fertilized* (IVF) human embryos, certain measurements relating to cell division timing have been shown to correlate with embryonic viability in a clinical setting [2, 4, 7, 8]. For model organisms, such as mouse, statistics of cell duration are relevant in the study of the role of certain genes in the initial phases of life. The contribution of different genes to early developmental events can be studied by silencing gene activity using RNA interference (RNAi) and analyzing any resulting changes in cellular behavior (including cell cycle timing) in early embryos.

Two main approaches to cell division detection are found in the literature. In [4], the sum of absolute pixel-to-pixel differences between consecutive frames generates a curve whose

peaks ideally highlight cell divisions. The drawback is that it is agnostic to other events, such as abrupt cell movements. The second approach, as used in [2, 5, 9], consists of tracking each cell individually, and detect when they divide by observing their boundary. But the task is hard to accomplish, due to cell overlap and occlusions.

In this work we take a statistical approach to the problem. First, we compute the likelihoods for cells of certain radii ranges to be in each frame—without actually locating the cells. The computation consists of a voting scheme, where votes come from quadruples of points in a way similar to the first step of the Randomized Hough Transform (RHT) for ellipse detection [10]. Second, we search for points of abrupt change in the matrix of likelihoods (built for all frames), and pick the two optimal division points using a dynamic programming algorithm [11].

We test our method on a database of 100 annotated videos [12] and estimate the time points of first and second division cycles (from one to two cells, and from two to four cells, respectively). Our results outperform two other recent methods, and the medians of our estimated division times differ less than two frames from the medians of the annotated times.

2. SHAPE STATISTICS

Mouse-embryo cells are well approximated by ellipses—see Figure 1. In this section we describe how we look for evidence that there’s an ellipse of a certain radii range in an image. We call “radii range” the interval between the semi-minor and semi-major axis of an ellipse. Such evidence is captured by looking at pairs of points with their respective tangents. Points and tangents are themselves computed using Morlet wavelets [13].

2.1. Points and Tangents

Let I be our input grayscale image, of size $N_r \times N_c$. Let w and h be half-window and hop parameters, respectively. Let s be the scale of the wavelet, and N_θ the total number of orientations considered. We convolve I with N_θ wavelets, of scale s and orientations $\theta_k = \frac{k}{N_\theta} \cdot 2\pi$, for $k = 0, \dots, N_\theta - 1$, generating N_θ new images, I_k .

*This research is supported by NIH grants 5R01-GM085503 and 5R01-HD046236.

[†]The fourth author is a visiting scholar, from The Hebrew University of Jerusalem. His work was supported in part by the Israel Science Foundation and the Intel Collaborative Research Institute for Computational Intelligence (ICRI-CI).

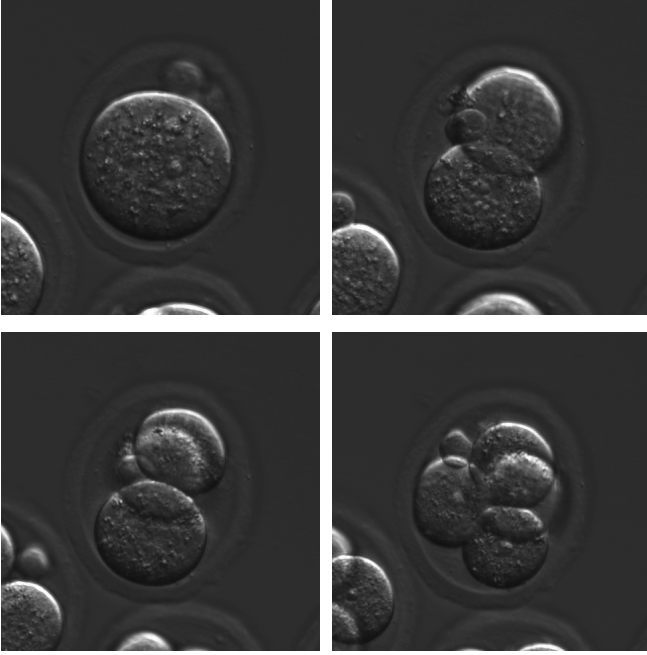


Fig. 1. Sample frames from a time-lapse movie of a particular mouse embryo.

Let $\tilde{N}_r = \lfloor (N_r - 2w)/h + 1 \rfloor$ and $\tilde{N}_c = \lfloor (N_c - 2w)/h + 1 \rfloor$. Then, for every $i = 1, \dots, \tilde{N}_r$ and $j = 1, \dots, \tilde{N}_c$, we find

$$\max_{k, \tilde{i}, \tilde{j}} I_k(w + (i - 1)h + \tilde{i}, w + (j - 1)h + \tilde{j}),$$

for $k = 1, \dots, N_\theta$ and $\tilde{i}, \tilde{j} = -w, \dots, w$.

Our method is based on the list of points of maxima for which the maximum satisfies certain local and global restrictions, along with the corresponding wavelet angles. We define this list as

$$L = \{(p(l), \tau(l)) : l = 1, \dots, N_p\},$$

with $\tau(l) = (\cos \theta(l), \sin \theta(l))$, where $\theta(l)$ is the tangent at the point of maximum $p(l)$. We will also use the notation τ_p to refer to the tangent at point p . Figure 2 shows an example of captured tangents for an image containing 4 cells.

The parameters s , w , h , and k are determined empirically, depending on the images. s is such that only edges of a target magnitude are considered. w and h must satisfy the condition that two neighbor windows do not overlap (otherwise there would be redundant pairs in the set L). The local restrictions enforce that all values in the window are above a global threshold, and coefficients are discarded if there's a stronger output nearby. N_θ is usually 16 or 32 (larger values do not improve accuracy significantly). In the results reported here we use $N_\theta = 16$, $h = 5$, and $w = 1$.

There are two main reasons for looking at the wavelet outputs on a sub-grid of the image, not on the entire image. First,

the smaller the the grid, the faster the method. Second, a window approach allows robustness to small perturbations.

2.2. Likelihoods

Let p, q be points in an ellipse, with tangents τ_p, τ_q . Let x_{pq} be the intersection of the lines passing through p with tangent τ_p and through q with tangent τ_q . Let m_{pq} be the midpoint of (p, q) . Then the following property holds: the line through x_{pq} and m_{pq} intersects the center of the ellipse. In fact, this trivially holds for a circle, and an ellipse is a center-preserving linear transformation of a circle. It then follows that two such lines intersect at the ellipse's center.

This property has been used for ellipse detection in the Randomized Hough Transform [10]. We implemented it with a slight modification: we look at pairs of pairs, instead of a triplet of points (a triplet defines two distinct pairs), since it makes the implementation easier, and lets one choose more robust, close to orthogonal, pairs of lines to intersect. Also as in the RHT, we reduce computational load by using a subset of points (actually a subset of pairs of points), chosen randomly.

In the videos we analyze there are mainly three stages, corresponding to the first three generations of cells. In the initial stage, there's one cell (of first generation), of radii range around r_1 . When this cell divides, it gives origin to two daughter cells (of second generation), each with radii range around r_2 , with $r_2 < r_1$. When these two cells divide, there will be four cells (of third generation), with radii range around r_3 , with $r_3 < r_2$. Ideally, the second generation cells divide at the same time. In practice, though, there's usually a separation in frames. Our method estimates the first division and the average location of 2nd and 3rd divisions (when the 3rd generation cells appear).

Let $R_i = [r_i^L, r_i^U]$ be the estimated interval of radii for a cell of generation i . Since we are interested in the first two rounds of divisions, we consider $i \in \{1, 2, 3\}$. Let c_{pqrs} be the center of the ellipse defined by points p, q, r, s and their respective tangents, as described above. We define a_i as the number of points c_{pqrs} for which the distances between c_{pqrs} and the boundary points p, q, r, s are in the range R_i . Ideally, dividing a_i by $((r_i^L + r_i^U)/2)^2$ (which is approximately the

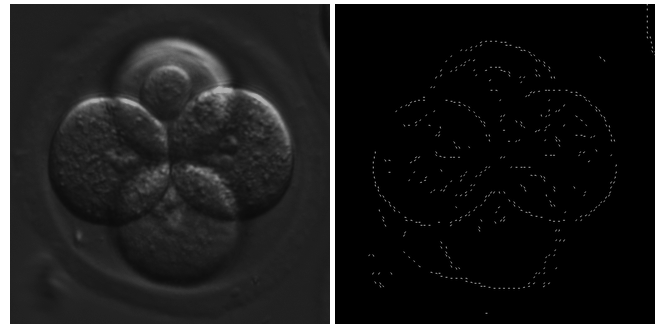


Fig. 2. Embryo image, and corresponding captured tangents.

area of the corresponding ellipse) would make the number of votes invariant to change in radii range. However there’s greater overlap for smaller cells, so we further multiply a_2 by 2 and a_3 by 4 (these factors chosen empirically).

We compute $a = \{a_3, a_2, a_1\}$ for every frame in the time-lapse video, normalize it so that $\sum_i a_i = 1$, and concatenate them in a matrix A (of size $3 \times n$, where n is the number of frames). A is further normalized to be in the range $[0, 1]$ (an example can be seen in Figure 3 (a)). Finally, we define $C = 1 - A$, and use C for division detection, as described in the next section.

3. DETECTING DIVISIONS VIA DYNAMIC PROGRAMMING

Divisions are obtained from a certain path of minimal cost, computed via dynamic programming, in the matrix C built in the previous section. DP is guaranteed to find the optimal solution, which is important because otherwise we can’t be sure if a bad solution is due to the model or to the method used to solve it. Some problems produce models that are NP hard to solve, in which case one has no choice but to work with approximation methods. But we were fortunate that in our problem DP finds a solution with satisfactory speed.

As input to the dynamic programming algorithm, we have a 3-rows image C similar to the one shown in Figure 3 (a) (where bright pixels should be interpreted as low values), in which the values range from 0 to 1. We look for the path of minimal cost going from left to right, starting at the third row, and not going downwards. That is, once the path is in row i , it can either remain on that row or go to row $i - 1$ (supposing $i - 1$ exists). Furthermore, there’s an additional cost related to not changing rows. The higher the change in cost within the row, the higher the cost of not moving upwards.

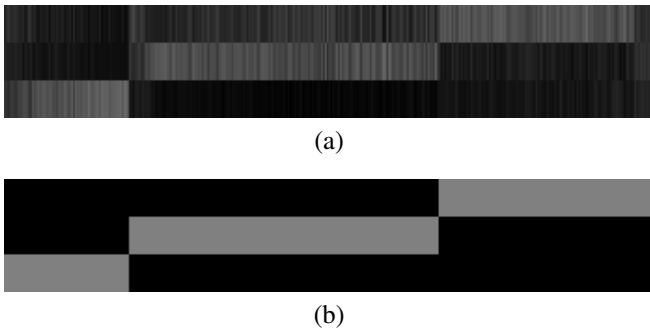


Fig. 3. (a) 3-rows matrix A ; rows represent different frames; columns represent radii ranges; the last row corresponds to the radii range of 1st-generation cells. (b) 3-rows matrix showing the solution (in gray), by dynamic programming, of the path of minimum cost from left to right in (a).

Dynamic programming involves building two matrices: Q , of accumulated path costs, and P , of predecessors (which allows recovering the path of minimal cost).

From C , we define a “windowed” derivative, D . Let w be a window-size parameter; and m, n the number of rows, columns in C , respectively. For $j = w + 1 \dots n - w + 1$, and $i = 1, 2, 3$, set

$$D_{i,j} = \frac{1}{w} \|C_{i,j \dots j+w-1} - C_{i,j-w \dots j-1}\|_1,$$

and the remaining columns of D by replicating the closest filled column.

Let β be a “decay” parameter. We define a “derivative strength,” S , as

$$S(i, j) = 1 / (1 + e^{-\beta D(i, j)}).$$

S ranges from $1/2$ to 1 when D goes from 0 to ∞ .

The first column of Q is set as the first column of C . For $j = 2, \dots, n$, we set

$$\begin{aligned} s &= \min\{Q_{i,j-1}, Q_{i+1,j-1} + \mu(1 - S_{i+1,j})\}, \\ s_a &= \arg \min\{Q_{i,j-1}, Q_{i+1,j-1} + \mu(1 - S_{i+1,j})\}, \\ P_{i,j} &= s_a + i - 1, \\ Q_{i,j} &= s + C_{i,j}, \text{ for } i = 1, 2, \text{ end} \\ P_{3,j} &= 3, \\ Q_{3,j} &= Q_{3,j-1} + C_{3,j}, \end{aligned}$$

where s_a assumes values 1 or 2 and μ controls the weight of the penalty S . For the example shown in Figure 3 we used $w = 5$, $\beta = 5$, and $\mu = 5$. These parameters are set empirically (they are equal by coincidence).

To recover the optimal path $p = \{p_1, \dots, p_n\}$, we compute $p_n = \arg \min\{Q_{1,n}, \dots, Q_{m,n}\}$, and for $j = n - 1, \dots, 1$, we set $p_j = P_{p_{j+1}, j+1}$. See Figure 3 (b) for an example.

4. RESULTS AND DISCUSSION

We tested our method in 100 time-lapse videos of mouse embryos, from the database in [12]. Statistical results are detailed in Figure 4 and Table 1, where we also show the performance of the methods described in [4] and [9].

The median of the true and computed times of first divisions differ by only “half” a frame (50.5 vs 51). The median of the average between the annotated (true) second and third division times differs 1.25 frames from the computed “mean” between second and third division. As can be seen in Table 1, our method provides medians that are closer to the true medians when compared with other methods. As per Figure 4, our method also provides the closest distribution.

Our experiments show that the dynamic programming approach on the matrices of shape likelihoods provides good statistical predictors for the times of first and second rounds of cell divisions. It outperforms not only a simple frame-difference method [4], but also a much more sophisticated algorithm that relies on cell tracking [9].

Regarding computational complexity, capturing wavelet coefficients is linear in the size of the image, and computing

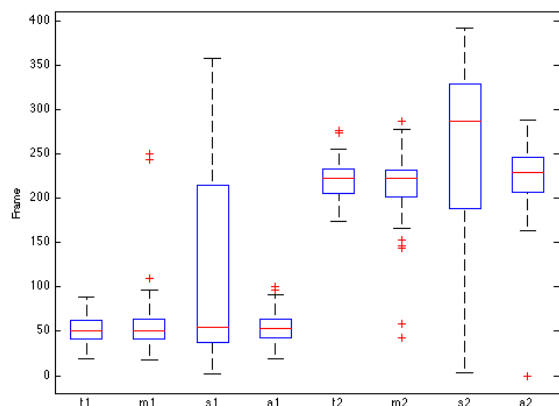


Fig. 4. Box plot of annotated and measured values for the times of first cell division ($\{t,m,s,a\}1$) and the mean between times of second and third divisions ($\{t,m,s,a\}2$). $t\{1,2\}$: true (annotated) distribution; $m\{1,2\}$: our method; $s\{1,2\}$: method suggested in [4]; $a\{1,2\}$: method described in [9]; Red lines within boxes are the corresponding medians—their values are shown in Table 1.

	t	m	s	a
1st division cycle	50.5	51	55	53
2nd division cycle	221.75	223	287	228.75

Table 1. Medians of the times of first and second rounds of divisions (corresponding to the red lines within boxes in Figure 4). t: true distribution; m: our method; s: [4]’s method; a: [9]’s method.

the likelihoods a_i is quadratic in the number of coefficients. Thus our method is quadratic in the number of image pixels. In practice, our Objective-C implementation runs in about 5 seconds per frame (each frame is 480 by 480 pixels).

A time lapse movie has about 300 frames. Dynamic programming was implemented in Matlab (R2011b, 64-bit), and runs in about 8 milliseconds per movie. All experiments were performed on an iMac with 2.8 HGz Intel Core i7 processor and 16 GB RAM, running Mac OS 10.9.

5. REFERENCES

- [1] Y. Mio and K. Maeda, “Time-lapse cinematography of dynamic changes occurring during in vitro development of human embryos,” *Am J Obstet Gynecol.*, 2008.
- [2] C. Wong, K. Loewke, N. Bossert, B. Behr, C. De Jonge, T. M Baer, and R. Pera, “Non-invasive imaging of human embryos before embryonic genome activation predicts development to the blastocyst stage,” *Nature Biotechnology*, vol. 28, pp. 1115–21, Oct 2010.
- [3] Tatsuo Nakahara, Akira Iwase, Maki Goto, Toko Harata, Miyabi Suzuki, Miki Ienaga, Harumi Kobayashi, Sachiko Takikawa, Shuichi Manabe, Fumitaka Kikkawa, and Hisao Ando, “Evaluation of the safety of time-lapse observations for human embryos,” *J Assist Reprod Genet.*, vol. 27, pp. 93–96, 2010.
- [4] M. Meseguer, J. Herrero, A. Tejera, K. Hilligsoe, N. Ramsing, and J. Remohi, “The use of morphokinetics as a predictor of embryo implantation,” *Human Reproduction*, vol. 26, no. 10, pp. 2658–71, Oct 2011.
- [5] M. Cicconet, D. Geiger, and K. Gunsalus, “Wavelet-based circular hough transform and its application in embryo development analysis,” in *VISAPP, 2013*, pp. 669–674.
- [6] SL Chavez, KE Loewke, J Han, F Moussavi, P Colls, S Munne, B Behr, and RA Reijo Pera, “Dynamic blastomere behaviour reflects human embryo ploidy by the four-cell stage,” *Nat. Commun.*, no. 3, pp. Article number: 1251, 2012.
- [7] C Wong, AA Chen, B Behr, and S Shen, “Time-lapse microscopy and image analysis in basic and clinical embryo development research,” *Reprod Biomed Online*, vol. 2, no. 26, pp. 120–129, Feb 2013.
- [8] J Conaghan, AA Chen, SP Willman, K Ivani, PE Chenette, R Boostanfar, VL Baker, GD Adamson, ME Abusief, M Gvakharia, KE Loewke, and S Shen, “Improving embryo selection using a computer-automated time-lapse image analysis test plus day 3 morphology: results from a prospective multicenter trial,” *Fertil Steril.*, vol. 2, no. 100, pp. 412–419, Aug 2013.
- [9] M. Cicconet, M. Gutwein, K. C. Gunsalus, and D. Geiger, “Label free cell-tracking and division detection based on 2d images for timing analysis of early embryo development,” Submitted to *Computers in Biology and Medicine*, 2014.
- [10] Robert A. Mclaughlin, “Randomized hough transform: Improved ellipse detection with comparison,” Tech. Rep., Univ. of Western Australia, 1998.
- [11] R. Bellman, *Dynamic Programming*, Dover Publications, Incorporated, Mineola, USA, 2003.
- [12] M. Cicconet and K. C. Gunsalus, “Mouse embryo tracking database,” <http://aquila.bio.nyu.edu/celltracking/>, 2014.
- [13] Joan Bruna and Stéphane Mallat, “Invariant scattering convolution networks,” *CoRR*, vol. abs/1203.1513, 2012.