A Tracking Algorithm For Cell Motility Assays in CMOS Systems

C. Martínez-Gómez, A. Olmo, G. Huertas, P. Pérez, A. Maldonado-Jacobi and A. Yúfera *Member, IEEE*

***Abstract*—. This work proposes a method for the study and real-time monitoring of a single cell on a 2D electrode matrix, of great interest in cell motility assays and in the characterization of cancer cell metastasis. A CMOS system proposal for cell location based on occupation maps data generated from Electrical Cell-substrate Impedance Spectroscopy (ECIS) has been developed. From experimental assays data, an algorithm based on the analysis of the eight nearest neighbours has been implemented to find the cell center of mass. The path followed by a cell, proposing a Brownian route, has been simulated with the proposed algorithm. The presented results give an accuracy over 95% in the determination of the coordinates (x, y) from the expected cell center of mass.**

I. Introduction

Cell motility plays an important role in many biological processes, such as embryogenesis, wound cicatrization, immune response, and cancer evolution [1]. Tumour cell motility is directly related with the processes of cancer propagation, generating metastasis processes, which is one of the main reasons of death related with this injury. The in-vitro assays of cell motility represent an useful tool for the research on the regulation mechanism of the cancer cells migration, also to test the efficiency of alternative drugs to combat cancer at a cellular level. The most common methods for studying cell motility are optics, based on microscopy, and with fluorescence techniques. However, these well-established and referenced methods require fluorescence markers, which can interfere on the correct function of some proteins, modifying the normal cell evolution [2]. In addition, light application at high intensity levels required for exciting fluorescence compounds, can deliver or generate some toxics elements at cells. ECIS technique [3] is a well-established technique that allows the automatization in cell culture research based on impedance measurements done on the cell attachment performance, to obtain cell properties, cell index, etc. [4, 5]. ECIS techniques represent a non-invasive method for real time cell monitoring to study cell properties: cell adhesion, motility, drug assays, cell growing, etc. [6, 7].

Experimentally, ECIS technique requires of an excitation signal, current (or voltage), applied to obtain a voltage (or current) as response. The Bio-Impedance (BI) information due to cell attachment to the electrode is extracted from the signal response (real and imaginary components, or magnitude and phase [8]). The main problems to be solved to extract this information are two. Firstly, BI changes due to cell culture measurements must be performed with accuracy using adequate techniques and circuits with high performance [4] (frequency programmable voltage/current generators, amplifiers, demodulators, etc). Secondly, data obtained for BI of electrode-cell system must be decoded to rebuild the information sought, in general, number of cells.

The proposed system shown in Figure 1 can be implemented in CMOS technology. It is composed of a 2D matrix of electrodes, which act as "small sensors" of BI [9], integrated on the same or similar silicon substrate that is employed by the CMOS circuits for measuring and signal acquisition [8, 10]. The circuits allow row/file selection to drive the actual "pixel" under test.

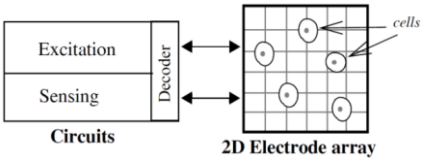


Figure 1: 2D electrode matrix and circuits for the excitation and acquisition of electrical signals for biompedance test of a cell culture.

The BI data obtained from cell cultures can be employed to model the 2D system proposed in Figure 1. It can be defined the fill-factor parameter (*ff*) as the proportional area filled by the cells to the total area of one electrode. This parameter oscillates from *ff=0*, when the electrode is totally empty of cells (on top), to *ff=1*, if the electrode is totally covered of cells. This system gives us a dimensional matrix of numbers, one for each pixel, in the range of 0 to 1, representative of a cell culture status. As it is illustrated in Figure 2, for a MCF7 cell line image, with an 8x8-electrode array, black and white images can be created from bio impedance measurements. The study proposed in this work focused on spatial-temporal location of a single cell inside an electrode matrix, using for that the information obtained from sensors (pixels, *ff* map). It has been developed a Location Algorithm implemented in Matlab to define the track followed by a single cell in a culture, determining the time evolution of its center of mass in a defined period of time.

This document is organized as follows. Section II describes the proposed system structure and the modelling of the cell under study. Section III details how the algorithm for locating a cell works, while section IV describes its program implementation, the simulations performed and the validation process. Finally, section V shows the results and conclusions.

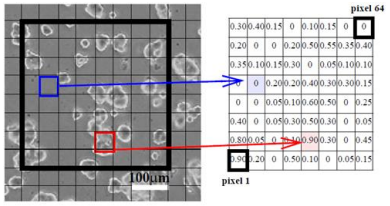


Figure 2: *Fill-Factor* map associated to each electrode.MCF7 cell line.

All authors are with Instituto de Microelectrónica de Sevilla,CSIC, U. de Sevilla, Av. Americo Vespucio sn,41092,Spain (yufera@imse-cnm.csic.es).

C. Martínez-Gómez, A. Olmo, P. Pérez, and A. Yúfera are with Dto. de Tec Electrónica, ETSII, U. de Sevilla, Av Reina Mercedes, 41010,Spain.

G. Huertas and A. Maldonado-Jacobi, are with the Dto de Electrónica y Electromag., F. Física, U. de Sevilla Av. Reina Mercedes, 41010, Spain.

II. System Modelling

The proposed system structure for the cell location algorithm is described. The first step is devoted to model the cell which will be used in the case study. A wide variety of cells with very different shapes and structures can be found. For the sake of simplicity, a circular cell is chosen in such a way that it is defined by both the location of the center of the circle (x, y) as well as the radius (r). It should be considered that circular cell modelling is an ideal model, while elliptic morphologies with variable radius could best conform to the reality. The second step is to define the bidimensional array of electrodes: an array *M* of *NxN* dimension, where each element *M(i,j)* includes an electrode of fixed area, being *i* the position of the row and *j* the position of the column. The array *M* stores in each element its corresponding *ff*. The electrodes in the array are considered squared and the side *l*, equal to the cellular diameter. The center of mass (*cm*) is defined as:

- Center of mass (*cm*): For a discrete set of masses system it is the weighted average, according to the individual mass, of the positions of all the particles that compose it. It can be calculated as:

$$\vec{r}_{cm} = \frac{\sum_i m_i \vec{r}_i}{\sum_i m_i} = \frac{1}{M}\sum_i m_i \vec{r}_i \qquad (1)$$

M, total mass; *m*, mass of the *i-th* particle; $\boldsymbol{r_i}$, position vector of the *i-th* mass.

III. Location Algorithm

The goal of the proposed algorithm is to obtain the center of mass (*cm*) of the cell, for a given and fixed occupation map, from *ff* of the electrode array elements defined, by measurements. An iterative algorithm has been developed, which assigns weights to each element of the array according to whether the 8 adjacent elements contain occupancy values. In the algorithm, it is defined:

- **The occupation array *M***, which includes the fill factor values from measurements.
- An empty **subdivision array $\boldsymbol{M_s}$** of *2Nx2N* dimensions, is also defined. It represents the subdivision of the occupation array, where each element *M (i, j)* is split into four. In each iteration, the M_s array is subdivided into 4 sub-elements and so on until an optimal result is reached. This array stores the weights indicating which elements of it are parts of the cell area under study (Figure 3).
- From proposed array model, the cell can occupy a maximum of four elements of the array *M*, i.e. there will be at most four non-zero fill factors in the array *M*. An **index vector *I*** is defined that contains the positions *(i.j)* of these four possible values of *M*.
- The **Array *P*** stores the central points with greater weight M_s elements, adjudicated by the algorithm described later.

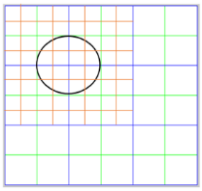


Figure 3. Array M of 3x3 dimension (blue), array Ms of 6x6 dimension (green) and the subdivision of Ms 12x12 (red).

The algorithm can be divided into three execution steps:

Step 1: Initialization: The occupation map elements of *M* have input values given by the *ff* resulting from the experiments. Firstly, an initialization process is performed, according to which the occupation map elements *M* are subdivided into 4 sub-elements and the weights are assigned, initializing the matrix *Ms*. These initial values are selected according to the algorithm proposed.

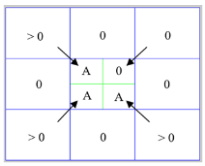
Step 2: Iteration: Secondly, an iterative process is developed where the subdivision array, which contains the weights, is subdivided into 4 sub-elements and so on, at each iteration. At each level of the iteration process, the current area resulting from the algorithm is calculated. The process ends when the areas obtained from the selected sub-elements, for a determined level of iteration, are the closest to the occupancy values obtained by the sensors *(ff)*.

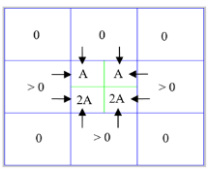
Step 3: Center of mass calculation: From the resulting center of mass, the ff_s corresponding to this point, called in the algorithm ff_{fb}, are calculated and compared with the real ff_s of the given occupation map. With this step, the system receives a feedback in such a way that the mass center is recalculated according to the difference obtained between the calculated and actual *ff*, causing a translation of the center of mass. This recalculation reduces the error in most of cases. The actions algorithm steps are detailed below.

First step begins by traversing the *M* array, which initially contains the values of the *ffs* resulting from biomedical experimentation. The goal is to assign values to the subdivision array M_s. Starting from each element *M(i,j)* with a non-zero value and smaller than *0.75*, weights are assigned to the four sub-elements of the array M_s which correspond to this element *M(i,j)*. The assigned weights are determinate by the values of the 8 adjacent elements of *M(i,j)*. In particular, the weights will depend on:

- If the neighbor of the diagonal contains a non-zero value, then a constant *A* is added to the element M_s adjacent to the diagonal (Figure 4a).

(a)





(b)

Figure 4. (a) It despicts the diagonal adjacency of the central element. (b) For each 4-adjacent neighbor of the array *M*.

- If the remaining neighbors, which do not conform the diagonal, contain a non-zero value, then a constant A is added to the two elements of M_s adjacent to the sides (Figure 4b).

With this in mind, an element of M_s will have at most a weight of $3A$. On the other hand, if the value is higher than 0.75, the maximum weight, $3A$, is straightaway assigned to the 4 sub-elements ($M_s(2i-1,2j-1)$, $M_s(2i-1,2j)$, $M_s(2i,2j-1)$, $M_s(2i,2j$. The M array is again examined and the following conditions are established:

- If $0 < M(i,j) \leq 0.25$, only the two larger weight sub-elements of the four possible sub-elements that would form $M(i,j)$ are stored in the array M_s.
- If $0.25 < M(i,j) \leq 0.50$, the three larger weight sub-elements are stored.
- If $M(i,j) > 0.50$, the four sub-elements of M_s are stored.

The iterative process tries to increase the resolution to obtain the area that most closely matches the real area of the cell under study. In each iteration, the M_s array increases its dimension as $2^{numIter+1}N$, being *numIter* the iteration number in which the process is and N the dimension of the M array.

As in step 1, the M_s array is examined and weights are assigned to the new subdivision array, M_{s_iter}, according to the values of the adjacent elements of M_s. Calculating the weights of the new array M_{s_iter} elements, those that contain the maximum weight with the same criteria established in step 1 are selected. With these elements the approached occupation area is calculated. As in each iteration the subdivision increases, the area that represents each element decreases and thus, the percentage of occupation area of each element will be given by: $4^{-1\text{-numIter}}$. At this point, the proposed occupation area by the algorithm is evaluated, and compared to the initial area, to which it must converge. If the estimated by the algorithm area is equal to the corresponding *ff* or approaches to a set range within error margins, the iterative process is terminated. Otherwise, step 2 is repeated to a maximum of 8 iterations. Once the iterative process is completed, the geometric centers of the higher-weight elements of the M_{s_iter} array are stored in the array P. And in turn, the mass center is calculated for each element of index I, this calculation is based on the points P contained within such elements. As discussed, there will be a maximum of four *ff* values and therefore four mass centers, calculated as:

$$\vec{r}_{cm_k} = \frac{1}{M} \sum_{i,j,k} ff_k P(i,j) = \sum_{i,j,k} ff_k P(i,j) \qquad (4)$$

where k defines the *k-th* value of the I vector and M defines the total system mass, in our case, is the sum of the *ff* whose value is always unitary. The iterative process results in the four mass centers related to each *ff*, allowing the calculation of the center of mass for the whole set corresponding to the *cm* of the cell.

$$\vec{r}_{cm_{cell}} = \sum_k ff_k \, \vec{r}_{cm_k} \qquad (5)$$

To verify that the result is correct, the system is feedback. The percentage of area occupied by the obtained cell (ff_{fb}) is calculated, and compared with the original ff_s. The *ff* and the mass center of the cell are recalculated:

$$\vec{r}_{cm_{cell\_fb}} = \sum_k ff_k + (ff_k - ff_{fd_k}) \vec{r}_{cm_k} \qquad (6)$$

IV. Software Implementation

The algorithm was implemented in Matlab. First, an example of the cellular localization is performed. Secondly, the study and simulation of a cell trajectory is described.

A: Cellular localization

An example with an array of 6x6 electrodes and a 10µm diameter cell is shown. To properly simulate the operation of cell cultures, the generation of the position that the cell occupies on the surface of the array is done in a random way. Once the mass center is defined, the occupation map to be used by the algorithm is calculated. Figure 5 shows the example of the map obtained from a cell with center cm_{real} *(13µm, 48µm)*.

0.02	0.23	0	0	0	0
0.12	0.63	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

Figure 5. Occupation map of a cell with center at $(x,y) = (13\mu m, 48\mu m)$.

After five iteration cycles, a set of points is obtained, from which the cell *cm* will be calculated (Figure 6). Specifically, two possible *cm* are obtained, corresponding to the algorithm execution without feedback (cm_{cell} ($12.63\mu m$, $47.99\mu m$)) and with feedback (cm_{cell_fb} ($12.87\mu m$, $47.96\mu m$)). The original cell is compared with the two generated results and the relative error is calculated. For Y axis, both results are approximated with an error lower than 0.05%. However, for X axis the error is reduced using feedback (from 2.8% to 1.0%). Figure 6 shows the matching between original and calculated cells.

B: Cell trajectory: Brownian motion

The ECIS technique opens the possibility of monitoring a cell culture in real time. In addition, to the estimation of the cellular location from a map of *ff* obtained with this technique, it is useful to have tools to analyze the cell time evolution. The tacks described by them could be analyzed. The mathematical model of the cell movement is of great relevance in the fields of biology and medicine, being the most commonly used based on the extensions of simple random motion processes. Assuming that motion is allowed in any direction, this process is essentially known as Brownian motion [11,13]. This phenomenon is based on the random motion of particles suspended in a fluid as a result of their collision with rapidly moving atoms or molecules. To generate a two dimensional random motion, two independent random paths are used. Instead of using random steps from a

Gaussian distribution, an approximation to Brownian motion can be constructed by taking random measurements of simple probability functions, such as a delta function or a constant probability density function [12]. A Brownian motion model is implemented, indicating the starting point for the cell and the number of time samples desired to obtain the trajectory. Random angles are generated and each *cm* is produced following a stochastic process:

$$cm_x(t) = cm_x(t-1) + \cos\sigma(t) \qquad (8)$$
$$cm_y(t) = cm_y(t-1) + \sin\sigma(t)$$

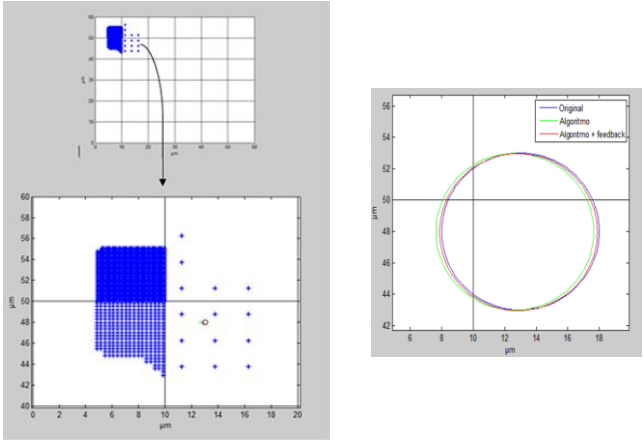


Figure 6. Resulting set of points for the elements with the higher weight, corresponding to the elements of the occupation map (left side). Real centre cm_{real} *(13µm,48µm)* (blue circle), obtained with the algorithm cm_{cell} *(12.63µm,47.99µm)* (green line) and the obtained with feedback cm_{cell_fb} *(12.87µm,47.96µm)* (red line). Right side: comparison of the original cell (blue line) with the calculated cell without feedback (green line) and with feedback (red line).

In addition, the generation of the trajectory is limited according to the size of the culture matrix. Figure 7 shows a possible trajectory generated by the cell under study. After obtaining the occupation map for each time instant, it is simulated the trajectory followed using the localization algorithm, previously implemented. To be specific, it is considered a cell with an average velocity of 0.1 µm/min. The example simulates 16 occupation maps for three hours. The measurements were obtained using an Intel® Core ™ i7-4501U processor at 2.60GHz and 11.9GB of RAM. The results obtained verify that in most cases the error decreases applying the feedback process, being position 2 and 3 the only points where the error is not improved. Even in these cases, errors are less than 3%. The majority of iterations required to obtain the position are two cycles. For iterations less than 8 cycles the time spent is less than 60 s. The highest errors obtained were located when the occupation map collects most of the area in a single element, but with connected elements with a very low value. In contrast, when the cell is more evenly divided into several elements, the error is very small. Finally, in the event that the cell is entirely in one element or divided exactly in two or four elements, the error obtained is null. Results show that the maximum error obtained is below 5% for the X axis and below 1% for the Y axis.

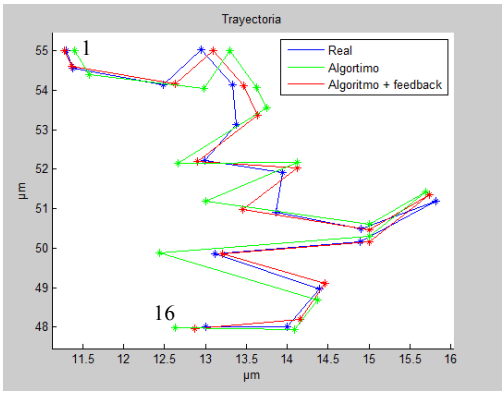


Figure 7. Real trajectory of a cell with radius *10µm* (blue), calculated with the localization algorithm (green), and with feedback (red).

V. Conclusion

A cellular localization system has been developed based on the occupation maps generated by electrical impedance spectroscopy. The localization system has been able to generate the approximated cell position in a culture, with a maximum relative error of 4.98%, and a typical error of 1%, when feedback is provided to the algorithm. The proposed tracking algorithm enables CMOS technologies for LoC systems in cell motility assays, particularly useful in cancer research and cell drugs assays.

Acknowledgment

This work was supported in part by the Spanish founded Project: TEC2013-46242-C3-1-P: Integrated Microsystem for Cell Culture Assays, co-financed with FEDER.

References

[1] R. Ananthakrishnan and A. Ehrlicher, "The Forces Behind Cell Movement," *Int J* Biol *Sci*, vol. 3, no. 5, 303–317. 2007.
[2] Z. Zhu, et al, "Time-lapse electrical impedance spectroscopy for monitoring the cell cycle of single immobilized S. pombe cells," Scientific Reports, vol. 5, 17180, Nov. 2015.
[3] I. Giaever et al.,"Use of Electric Fields to Monitor the Dynamical Aspect of Cell Behaviour in Tissue Culture," BME-33,242-247. 1986.
[4] Grimnes, S. and Martinsen, O., Bio-impedance and Bioelectricity Basics, 2nd edition. Academic Press, Elsevier. 2008.
[5] C. F. Yeh et al., "Towards an Endpoint Cell Motility Assay by a Microfluidic Platform," IEEE TNB, vol 14, nº 8, 835–840. 2015.
[6] A. J. Sinclair, et. al, "Bioimpedance analysis for the characterization of breast cancer cells in suspension," BME, 59, n 8, 2321–2329. 2012.
[7] D. Mondal and C. R. Chaudhuri, "Extended electrical model for impedance characterization of cultured HeLa cells in non-confluent state using ECIS electrodes," IEEE TNB, vol 12, nº 3, 239–246. 2013.
[8] A. F. M. Mansor, et al, "Cytotoxicity studies of lung cancer cells using impedance biosensor," (ICSSA),1–6. 2015.
[9] G. Huertas, A. Maldonado, A. Yúfera, et al, "The Bio-Oscillator: A Circuit for Cell-Culture Assays," IEEE TCASII, 62, 2,164–168. 2015.
[10] A. Yúfera and A. Rueda, "A CMOS bio-impedance measurement system," International Symposium on Design and Diagnostics of Electronic Circuits Systems, 252–257. 2009.
[11] A. Yúfera and A. Rueda, Design of a CMOS closed-loop system with applications to bioimpedance measurements, MJ. 41,231-239. 2010.
[12] P.-H. Wu, et al, "Three-dimensional cell migration does not follow a random walk," PNAS USA, vol. 111, n 11, 3949–3954, Mar. 2014.
[13] E. A. Codling et al, "Random walk models in biology," J R Soc Interface, vol. 5, no. 25,813–834, Aug. 2008.
[14] B. Qu and P. S. Addison, "Modelling Flow Trajectories Using Fractional Brownian Motion," International Workshop on Chaos-Fractals Theories and Applications, 420–424. 2010.