

Syracuse University

SURFACE

Dissertations - ALL

SURFACE

January 2015

Comparison of Cost Function Against Positional Analysis for Automated Tracking of Three-Cell Interactions

Ye He

Syracuse University

Follow this and additional works at: <https://surface.syr.edu/etd>



Part of the [Engineering Commons](#)

Recommended Citation

He, Ye, "Comparison of Cost Function Against Positional Analysis for Automated Tracking of Three-Cell Interactions" (2015). *Dissertations - ALL*. 344.

<https://surface.syr.edu/etd/344>

This Thesis is brought to you for free and open access by the SURFACE at SURFACE. It has been accepted for inclusion in Dissertations - ALL by an authorized administrator of SURFACE. For more information, please contact surface@syr.edu.

Abstract

Localization and tracking of cells generates raw digital information from microscopy images, including images of stained nuclei and highly precise determination of central positions of cells, which can be analyzed for investigation of cell motility. In a previous study by this group, an algorithm termed automated contour-based tracking for *in vitro* environments (*ACTIVE*) was established for tracking large cell populations for long periods of time. For the two-cell interaction events on which *ACTIVE* was initially focused, error rate was reduced as much as 43% compared to a traditional positional analysis algorithm by Kilfoil and colleagues. In the present thesis, we investigated whether the *ACTIVE* algorithm could be improved when applied to a more complicated condition: three-cell interactions. To determine whether modification of the *ACTIVE* algorithms could allow *ACTIVE* to outperform the Kilfoil benchmark method when applied not only to two-cell interaction cases but also to three-cell interaction cases, two approaches were developed and studied: 1) optimization of the existing *ACTIVE* cost-function weighting factors by orthogonal design with addition of two new factors, velocity and directionality, and detection of ranges and effects for all factors, and 2) modification of the circumstances under which the Kilfoil approach and the cost function approach were executed. We found the position factor to be the most important and accurate among all the factors, and optimized all factors. What is more, the directionality was determined to be the second most significant factor of the cost function for correctly tracking cells. However, modification of neither the position nor directionality factor could achieve higher accuracy than the Kilfoil method. Having evaluated the new strategy that combines both the cost function and the Kilfoil method, we found that

the new strategy did not result in higher accuracy for three-cell interactions, as compared to the pure Kilfoil benchmark method. The accuracy of the new strategy was 6% lower on average than the Kilfoil method. Although the results of the present work do not yet achieve a method for analysis of three-cell interactions that outperforms purely positional analysis, the work provides a method for optimization of the cost function and new understanding of characteristics of three-cell interactions that lead to reduced accuracy in the cost function and/or positional (Kilfoil) approaches.

Key words: Automated cell tracking, Orthogonal design, Weighting factor optimization, MATLAB

Comparison of Cost Function Against Positional Analysis for Automated Tracking of Three-Cell Interactions

By

Ye He

Nankai University, P. R. China, 2013

THESIS

Submitted in partial fulfillment of the requirement for the

Degree of Master of Science in Bioengineering

Syracuse University

August 2015

Copyright 2015 Ye He

All Rights Reserved

Acknowledgements

I am very grateful to have Dr. Henderson as my advisor, for supporting and guiding me as I learned a lot from this research field, as well as for his inspiration and encouragement during my thesis work. When I came across the issues of searching new factors that were able to describe cell motility, Dr. Henderson proposed many ideas that truly opened my mind. I sincerely appreciate all of his help so that I could mark this thesis work as a perfect ending of my graduate study.

I would also like to thank Megan Brasch, who comprehensively explained this research background to me and patiently answered my various questions. I would not get a better understanding of the work without her help. What's more, based on previous work completed by her and her colleagues in our lab, I could be able to go further. I also appreciate her great patience for showing me basic cell culture work and providing me amounts of data sets.

Last, I should be thankful to have my family for supporting me throughout my education. They taught me how to live, how to recognize my inside value and chase after my goals.

Table of Contents

Abstract.....	i
Acknowledgements.....	v
Introduction.....	1
Methods.....	7
1. Cell Experiments	8
1.1 Cell Culture.....	8
1.2 Cell Staining and Imaging	9
2. Cell Segmentation and Particle Identification.....	9
3. Interacting Cells Definition and Particle Tracking.....	10
4. Determination for Accuracy of the Kilfoil Benchmark Method	13
5. Cost Function Definition and Update.....	16
6. Cost Function Optimization	20
6.1 Application of Orthogonal Design.....	20
6.2 Velocity and Directionality Incorporation	23
7. Examination of Each Factor in the Cost Function	24
8. Condition Optimization.....	25
9. Method Validation.....	26
Results.....	27
1. Accuracy of the Kilfoil Benchmark Method.....	27
2. Orthogonal Test Results	32
3. Examination of Factors in the Cost Function.....	34
3.1 Determination of Position Weighting Factor Range.....	34
3.2 Determination of Each Factor.....	35
4. Modification of <i>ACTIVE</i> Execution Condition	37
5. Method Validation.....	40
5.1 Test on Data Sets of Same Density.....	41
5.2 Test on Data Sets of A Higher Density.....	41
Discussion.....	43
Appendices.....	49
1. Script: Three-cell Interaction Cases and Frame Information Extraction	49
2. Script: Plot Cells in Each Frame	50
3. Script: Collection of Each Cell Information for 1 Interacting Frame	51

4. Script: Collection of Each Cell Information for More Than 1 Interacting Frames	52
5. Cost Function for 1 Interacting Frame ^[12]	54
6. Cost Function for More Than 1 Interacting Frame ^[12]	61
7. Category Function: Determine Which Method Shall Be Used	68
8. Script: Use Directionality to Correct Cell Tracking.....	69
Reference	71
Biographic Data	75

Introduction

Analysis of cell behaviors in large population is of great importance to current bioengineering and pharmaceutical research. In many cell therapies, the location, distribution, and long-term viability of cells should be evaluated in a noninvasive manner ^[1]. In order to better understanding living cell behavior in model environments, an analysis of both anatomic and dynamic properties are often required ^[2].

Cell tracking refers to a process in which the behaviors of single cells can be monitored by modern digital imaging techniques ^[3]. The behaviors can include cell migration, cell cycle, and cell signaling process ^[3-5]. With a better understanding of those cell behaviors, such as movements of cell skeleton and prediction of cell fates, production of extracellular matrix, and contacts between transient cells ^[6], significant insights into complex multi-cellular processes can be achieved, including processes of wound healing, regeneration and ontogenesis ^[7-9], host defense mechanisms, and tumor cell metastasis and invasion ^[10]. Under both normal and pathological conditions, cells can grow and migrate extensively ^[10]. A deep understanding of the cellular movements, interactions, and the reaction to stimuli, are essential to effectively manage the environments and materials in which the cells are living ^[11]. The development of specific biomaterial scaffolds also requires the spatio-temporal measurements of cell behaviors, which is significant in tissue engineering ^[7, 11].

Therefore, the accuracy and efficiency of the methods for cell tracking also become challenging and worth studying in depth. Manual tracking relying on visual detection by a trained operator is widely used, especially for small population of cells ^[12], and still can achieve the most reliable results ^[13] but is extremely time-consuming ^[14] for a large cell

population and challenging for longer durations and higher-level dimension tracking. What is more, manual tracking is more subject to observer variance and bias ^[14], which prevents manual tracking from improving cell detection rates and becoming a standardized tool for the investigation of cell behavior ^[3].

To overcome the limitations of manual tracking, many automated tracking tools and methods have been developed ^[2], in which the characteristics of dynamic cells could be evaluated at the single-cell level ^[15]. Modern cell tracking methods include computer-controlled stage positioning and digital image acquisition systems, which enable sequences of cell images to be processed as time-lapse ^[10], in order to efficiently obtain quantitative and accurate measurements of cell behaviors, whose positive effect could enhance results of analysis ^[7]. For example, Padfield's tracking algorithms examined cell mitosis and apoptosis from fluorescent images, as well as protein translocation, from which the generated measurements are especially required in cancer research, immunology and developmental biology ^[13]. In House's work, they computed shape, orientation and movement characteristics to summarize and reason cell behavior not only in single-cell level, but also in groups ^[11]. The quantitative measurements included centroid displacement, path length, average speed, and changes and persistence in movement direction, which considerably described cell morphology and trajectories ^[11]. Rabut and Ellenberg introduced a 3-D fluorescent imaging method that could be used as a reliable microscopy-screening assay in cell cycle and migration, as well as development of loss-of-function phenotypes in RNAi experiments ^[4]. What is more, automated computational cell tracking methods complying with standardized characterization are commonly applied to *in vitro* models

established in various complex environments ^[12], such as hydrogel substrates with different stiffness properties ^[11], ordered topographies ^[16] and microchannels ^[17], which have specific biochemical and biophysical properties.

In most studies, prior to imaging, cells should initially be pre-processed by a proper labeling method according to different cell types and methods, such as nuclear DNA dye for cell nuclei ^[13], lipophilic carbocyanine derivative dye for red blood cells ^[17], fluorescent probe tagged protein attached to cells ^[18], and stem cells encapsulated by polysaccharide-coated iron oxide nanoparticles for MRI detection ^[1]. Then the motion of the cells could be captured by microscope. Because all objects in real world (including cells in the present case) extends in both space and time, time-lapse images are able to provide all information we need to track cells and analyze cell migration ^[2]. To date, most tracking algorithms consist of two key steps: 1) a segmentation step to recognize relevant cells by grey level intensity, color, or texture ^[10], and separate them from background in every frame, 2) a linking step to identify and connect segmented cells from frame to frame ^[2], thereby reconstructing their temporal continuity ^[3].

In the segmentation step, the ability to distinguish cells and their surroundings is crucial in many image-processing methods. In order to identify cells and background noise, an intensity threshold ^[2], level set ^[18,19], wavelet ^[20] and contour-based methods ^[21,22] are commonly used. Thus the cells can be identified by a particular measurement of respective method, then, all remaining parts that do not meet the requirement are filtered as background noise ^[12, 2]. Once the cells are detected, their information will be reserved for further cell tracking. Then, in the linking step, it is challenging to identify a same cell that is segmented in consecutive frames.

The simplest approach is identifying the nearest cell in the next frame, which refers to not only the spatial distance, but also to difference in intensity, volume, and other features^[2] specified by algorithms. In more complex conditions, the similarity of a same cell in consecutive frames can refer to image registration^[23] or smooth shape transitions (i.e. contours)^[5, 24]. Overall, the linking methods are based on the assumption that most cell behaviors meet a smooth and continuous process^[3].

In previous work by this group, an algorithm termed automated contour-based tracking for *in vitro* environments (*ACTIVE*) was established. Two significant innovations were achieved: the first one was that cell interaction events could be identified by specific contour-based profile; the second one was based on contour-based information, the Kilfoil benchmark linking was modified to link cells frame by frame^[12], so that cell interaction events could be analyzed and classified as divisions, merging or special events, and then a customized cost function was applied for cell tracking correction^[12]. The characteristics used for cost determination include five factors in total: integrated intensity, normalized intensity, area, aspect ratio and position. By execution of the equation, the minimum cost could be determined and then could be used to identify each cell in consecutive frames.

Previous work of the *ACTIVE* method was set up for two-cell interactions and the algorithm worked well in those cases, in which it could reduce error by as much as 43% in comparison with the Kilfoil method^[12], demonstrating the algorithm was fit for simple two-cell conditions. However, previous work of the *ACTIVE* also indicated that most errors detected were prone to be in higher-level (three or more) cell interactions^[12], since in multi-cell interaction events, cells were more difficult to be identified in a cluster while they

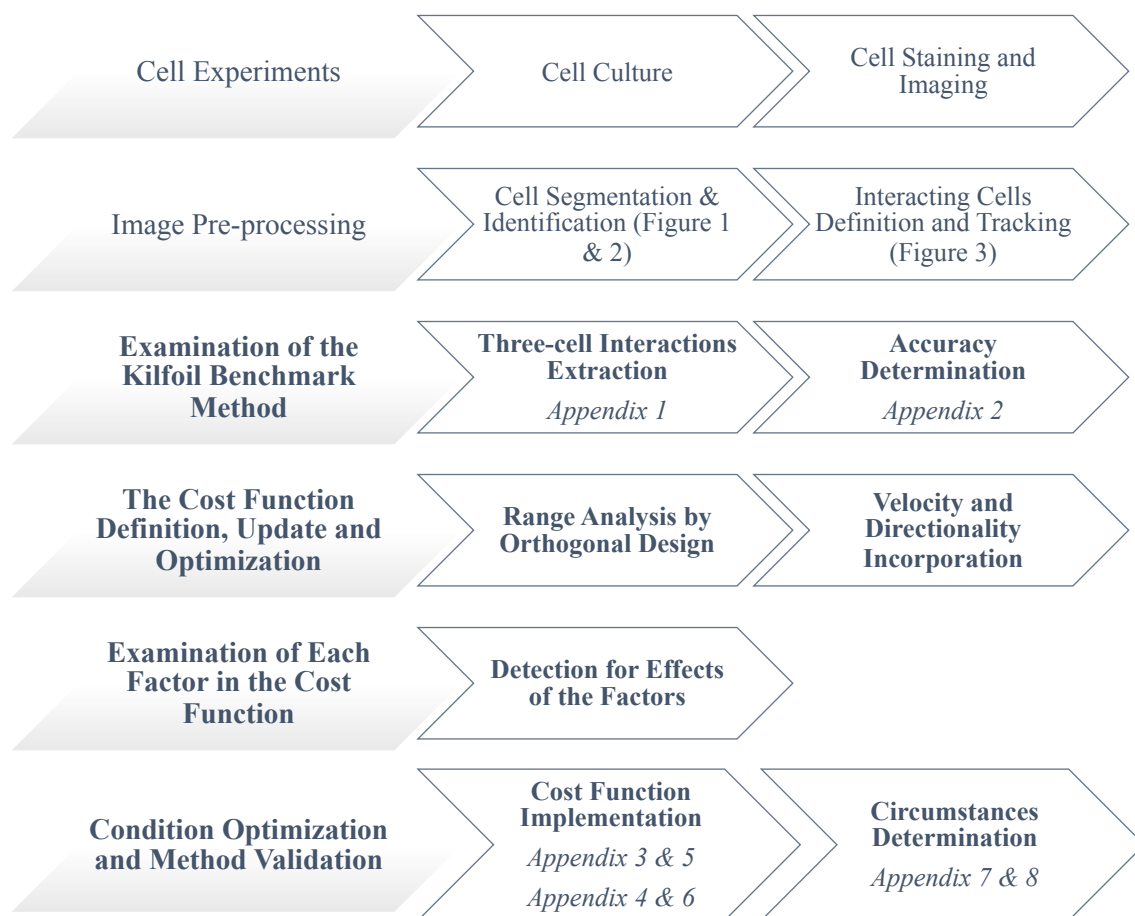
are interacting. Recognizing a population of cells requires more precise method to track each cell and longer time to process, making it challenging to achieve further improvement to identify and correct cell tracking. In contrast to the cost-function-based approach employed for two-cell interactions in *ACTIVE*, the Kilfoil tracking method relies only on positional information, which in theory might be relatively inaccurate for evaluating cell migration, since there are additional features that can be evaluated during cell migration, including cell nuclei intensity, shape, and direction of motion. Those valuable features can provide comprehensive analysis for cell identification during tracking. In contrast to the positional analysis introduced by the Kilfoil method, our goal was to investigate alternative approaches that might provide a more precise method to comprehensively assess cell migration based on the current *ACTIVE* method, including additional accessible cell information, not only intensity, area, aspect ratio, and position, but also velocity and directionality, and selected circumstances to determine which method was optimal. In our work, we chose three-cell interactions to analyze and aimed at the improvement of the cost function. Cell tracking accuracy was chosen as a crucial reference for evaluating the efficiency of cell tracking, and was determined by the percentage of how many cases in which cell identifications (IDs) had been correctly designated after either the Kilfoil linking method or the cost function was executed. The accuracy of Kilfoil linking was viewed as a benchmark method for cell tracking. Then the optimized *ACTIVE* and its cost function were applied to investigate the question of how accurate it would perform better than the benchmark method.

In order to determine whether modification of the *ACTIVE* algorithm could allow the *ACTIVE* algorithm to perform better than traditional Kilfoil benchmark method in analysis of

three-cell interactions, two approaches were studied: 1) optimization of the existing *ACTIVE* cost function weighting factors by orthogonal design with addition of two new factors, velocity and directionality, and detection of ranges and effects for all factors, and 2) modification of the circumstances under which the Kilfoil and the cost function were executed, based on a duration in which cells were occluded so that their information was absent. Thus, our study is an investigation of two approaches to determine whether morphometric and motility characteristics (*ACTIVE* and its cost function), or a combination of morphometric and motility characteristics with the Kilfoil method could outperform purely positional information (the Kilfoil method).

Methods

In order to clearly explain the overall strategy and set of tasks pursued herein, a flow chart is presented as Scheme 1. Cell experiments, image pre-processing and the cost function definition were developed by previous *ACTIVE* work^[12,25]. In this thesis work, examination of the Kilfoil benchmark method, the cost function update and optimization, examination of each factor in the cost function, and condition optimization and method validation were the focus and are highlighted in the chart, and the coding that formed the basis and product of the thesis was mainly created or modified to work for these purposes. The final version of code that was created for each of these corresponding parts are listed in Appendix 1 through Appendix 8, which are also marked in this flow chart, and the explanation of each is presented in the Results.



Scheme 1: Overall process, sections in normal text were established and completed by previous ACTIVE work, sections in bold text were completed by the present work

1. Cell Experiments

Cell culture, cell staining and imaging information were acquired from previous ACTIVE work completed by Baker, et al ^[25]. The following subsections briefly summarized the methods used in that previous work.

1.1 Cell Culture

In the previous ACTIVE work, C3H10T1/2 mouse fibroblasts (ATCC) were first expanded

in complete growth medium with seeding at 5 000, 10 000 or 20 000 cells/cm². Then, after expansion, cells at passage 13-15 were used for experiments. Samples were prepared for cell tracking experiments by seeding cells on static wrinkled substrate. After cell attachment, cells were stained and prepared for live cell imaging. The resulting experimental densities demonstrated nuclear area densities, quantified as the percentage of total image area occupied by cell nuclei, of 3.12%. [25]

1.2 Cell Staining and Imaging

In the previous *ACTIVE* work, cells were stained with Hoechst 33342 nuclear dye and image over 24 h in order to image cell nuclei for tracking analysis. Substrates with attached cells were placed in LabTek glass-bottom chamber slides and staining solution was added. Then the chamber slide was placed in a 37 °C incubator for 20 min to allow for the staining of cell nuclei. Afterwards, the chamber slides were placed in a live cell incubator, a Leica DMI 6000B inverted microscope was used to imaging, which was conducted at 37 °C for 24 h under 5 % CO₂, with images captured every 3 min by a camera. [25]

2. Cell Segmentation and Particle Identification

The *ACTIVE* was developed and implemented in MATLAB. In the present work, our main cost function code was inherited from the previous *ACTIVE* work. For the *ACTIVE* cell segmentation and particle identification, cell staining images were first processed by a Kilfoil band-pass filter to remove background noise, as shown in Figure 1, and then an intensity map was created to a generate contour profile by MATLAB built-in function, based on nuclear

intensity fluctuation ^[25]. Correlated contours helped to define individual cells ^[25], as shown in Figure 2.

3. Interacting Cells Definition and Particle Tracking

As the previous work set, contour profiles were established to recognize different intensities. When cells came close to each other, interaction events were established, where each cell taking part in the event should share at least one parent contour (lower level contour) ^[25]. Interacting duration was defined as how many frames in which the parent contours exist. Each frame was taken by 3 minutes duration. In order to distinguish multiple contour peaks that represent each cell in one parent contour, an ellipse was fitted to define a single cell based on a particular fit height, which a contour level of 15 was selected in current work ^[25], as shown in Figure 3. According to the contour-based segmentation established by the *ACTIVE* method, the Kilfoil linking system was applied for cell tags identification (IDs) and linked them in consecutive frames ^[25]. Thus, in general, a particular cell with a unique ID could be tracked through frames. Then in post-processing, the customized cost function could be used to check the validity of cell identification and correct cell tracking. Figure 4 shows an example of a three-cell interaction, where the paths of three cells come close to each other and become a qualified interaction event, and presented correct cell identification before and after their interaction, according to the cost function correction ^[12].

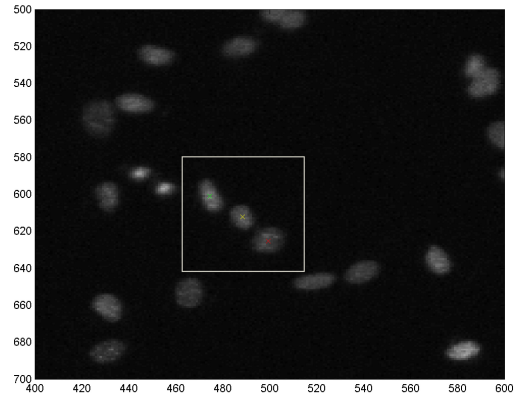


Figure 1: The interacting frame in a three-cell interaction event, which processed with the Kilfoil bandpass

filter to remove local noise

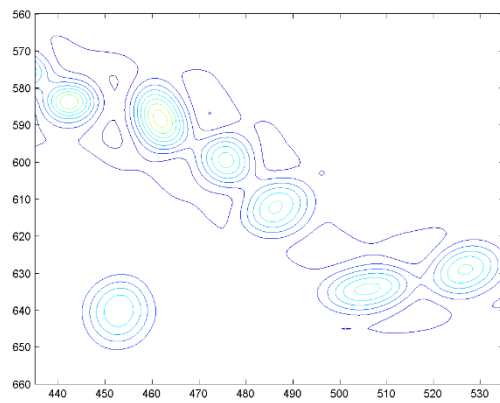


Figure 2: Contour image of Figure 1, the contour profiles were established by various nuclear intensities. A

three-cell interaction event was defined as any three cells that share at least one parent contour (lower level

contour)

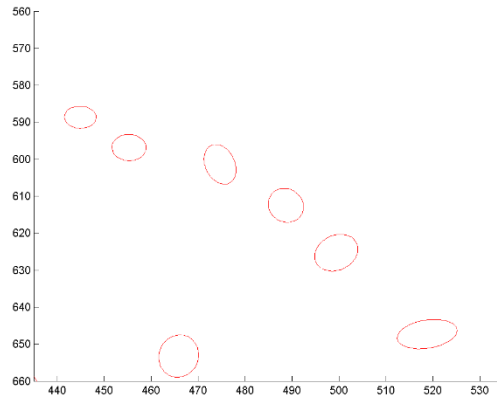


Figure 3: Each cell contour shown in Figure 2 was fitted by an ellipse to identify a single cell

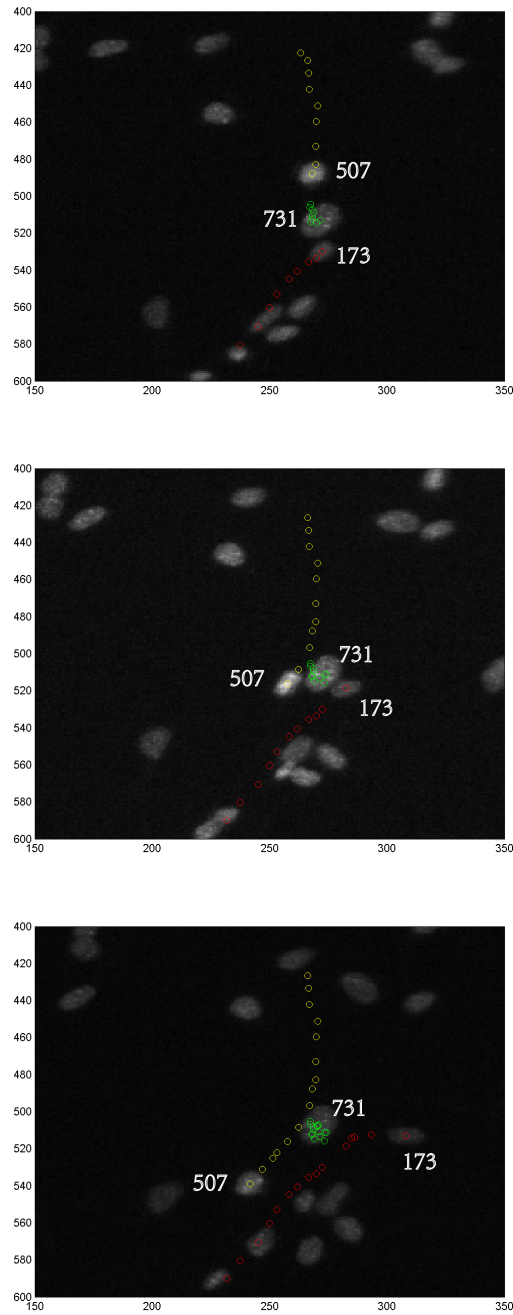


Figure 4: An example of three-cell interaction event with cell moving paths shows cell identifications according to the cost function correction in post-processing, when $T=81$ min, 99 min, and 123 min, from top to bottom

4. Determination for Accuracy of the Kilfoil Benchmark Method

Prior to post-processing that the cost function was applying, we first examined the Kilfoil

linking method to obtain the benchmark accuracy. Three-cell interaction cases were extracted from a low cell density (5,000 cells/cm²) data set: *7Percent_MedDensity_Sample5*, where all required information was included from previous work: intensity, cell area, major and minor axis, cell centroid position, etc. The code was implemented in MATLAB, as shown in Appendix 1. From the script in Appendix 1, we were able to acquire a list of three-cell interactions *three_cell_event_matrix* and all required information of each cell stored in *matrix* for plotting and tracking. Then each case was manually examined to check whether the tagged cell IDs were correct before and after interaction: we tracked each cell in filtered images and identified whether each cell ID matches before and after interaction, respectively. The code shown in Appendix 2 plots each cell according to their cell identification (IDs) and clearly labels them in each particular frame, which provided a reliable visual tracking. Thus, the accuracy of the Kilfoil benchmark method was determined according to total number of correct events and incorrect events.

For instance, Figure 5 shows an example for an event of correct cell identification that cells were correctly labeled based on their information, in which 99 tagged, 266 tagged and 740 tagged cell were successfully identified after interaction. Figure 6 shows an example for an event of incorrect cell identification, in which 151 tagged cell and 269 tagged cell wrongly switched after interaction. Both of the two events were identified by the Kilfoil linking.

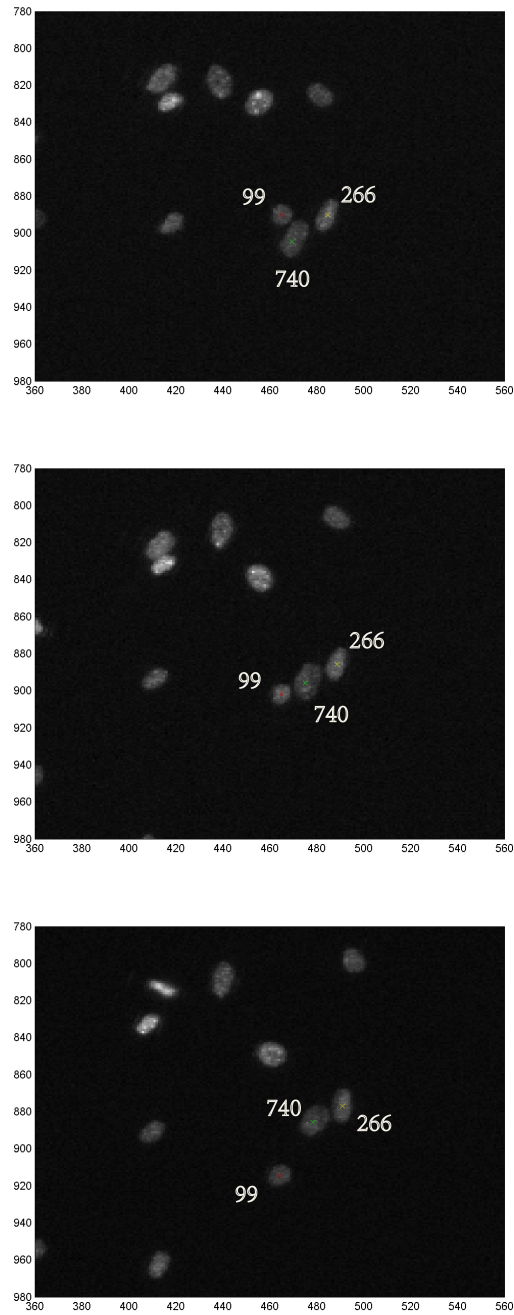


Figure 5: An example of three-cell interaction that has correct cell identification, $T=18$ min, 27 min, and 36 min, that before interaction, interacting and after interaction, from top to bottom

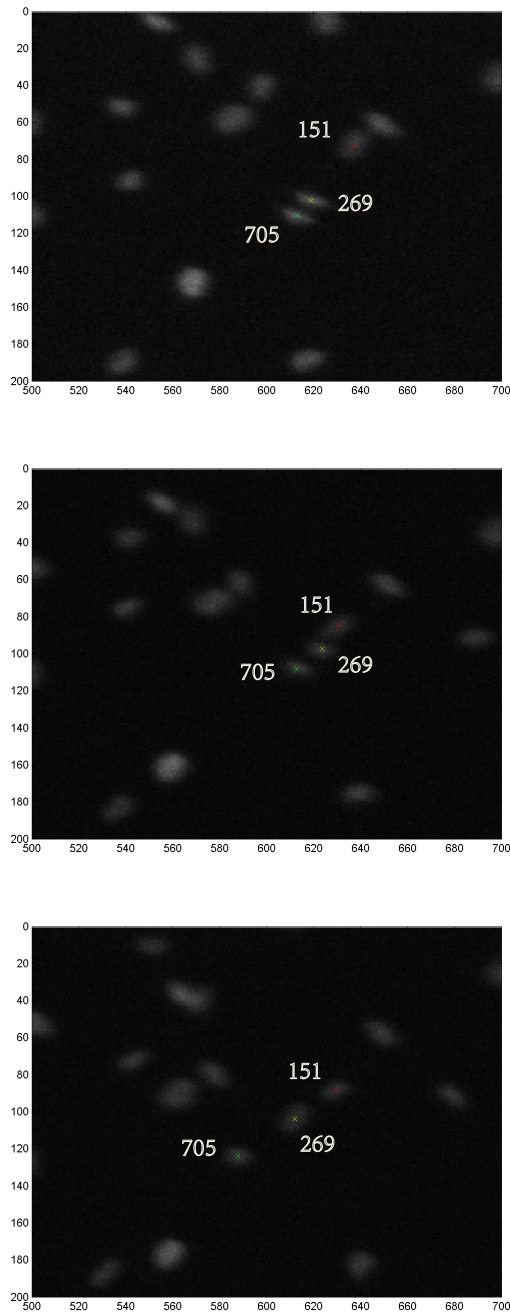


Figure 6: An example of three-cell interaction that has incorrect cell identification, T=93 min, 108 min, and 123 min, that before interaction, interacting and after interaction, from top to bottom

5. Cost Function Definition and Update

As a key innovation in the original *ACTIVE* method, a cost function was established in post-processing for cell tracking correction ^[12,25]. Since cells come close to each other, it is

challenging for previously employed linking system to distinguish them during the interaction, which would result in failure of cell identification (ID) ^[25], as showed in Figure 4. Based on cell ID history and specific kinds of interaction events, in previous work, a multi-frame positional or two-frame fingerprint analysis was employed conditionally to accurately identify cell IDs ^[25].

The general equation of positional analysis was defined as:

$$\text{cost} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

where (x_1, y_1) and (x_2, y_2) represent the center of masses for a cell in frames 1 and 2 respectively ^[25]. Thus, all possible combinations for the two IDs were tested (there were four combinations in two-cell interaction cases), and the minimum cost combination was selected as correct cell IDs in that particular case ^[25].

However, in certain cases where a single cell was occluded for several consecutive frames, of which cell information was absent during that interval, a fingerprint analysis was an alternative to improve the accuracy for cell tracking, while positional analysis would lead to reduced accuracy ^[25]. The general equation for fingerprint analysis was defined as:

$$\begin{aligned} \text{FP cost} = & w_{II} * \left(\frac{II_1}{II_1} - \frac{II_2}{II_2} \right)^2 + w_{NI} * \left(\frac{NI_1}{NI_1} - \frac{NI_2}{NI_2} \right)^2 + w_A * \left(\frac{A_1}{A_1} - \frac{A_2}{A_2} \right)^2 + w_{AR} \\ & * \left(\frac{AR_1}{AR_1} - \frac{AR_2}{AR_2} \right)^2 + w_P * \left(\frac{[\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}]}{\left[\frac{d_1 + d_2}{2} \right]} \right)^2 \end{aligned}$$

where II is the integrated intensity value for a cell, NI is the normalized intensity value for a cell, A is the area of a cell, AR is the aspect ratio of a cell, d is the diameter for a cell, each w represents a weight value for the subscripted variables previously defined and numerical subscripts denote frame numbers ^[25]. And w_{II} , w_{NI} , w_A and w_{AR} were set to 1 and w_P

was set to 0^[25]. Same as positional analysis, all possible combinations for the two IDs were tested (there were four combinations in two-cell interaction cases), and the minimum cost combination was selected as correct cell IDs in that particular case^[25]. With either positional analysis or fingerprint analysis, minimum cost was selected as correct cell IDs, and then, no further process needed to be done^[25]. However, if the combinations were incorrect, then *ACTIVE* would be taking further steps to update cell IDs^[25].

In the present work, our goal is to optimize the cost function. To achieve this goal, we first updated the setup of the cost function. While analyzing the process of a cell-cell interaction, a frame before interaction and a frame after interaction were the two crucial time points we should marked. Then, we would have three cells that captured in two frames, respectively, which generated 9 combinations that would need to be tested in comparison with 4 combinations in two-cell interactions. For a straightforward expression of the combination setup in the cost function, specifically, we denote A, B and C as each cell, 1 and 2 as the frame before interaction and after interaction, respectively. For example, A1 represents cell A in the frame before interaction, B2 represents cell B in the frame after interaction. An example event is shown in Figure 7. Thus, the 9 combinations could be divided into 3 groups, as presented in Table 1.

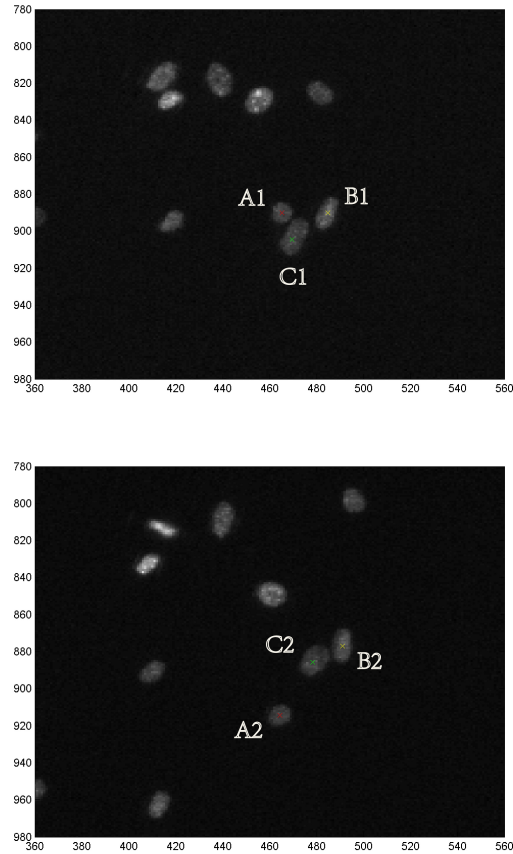


Figure 7: An example event for cell combinations before and after interaction, T=18 min, and 36 min, from left to right

Table 1: Combinations of Three-cell interactions

Group 1			Group 2			Group 3		
A1-A2	A1-B2	A1-C2	B1-A2	B1-B2	B1-C2	C1-A2	C1-B2	C1-C2

In Figure 7, presented is how cells were designated according to the cost function setup. In Table 1, group 1 denoted as the three possible combinations of the first cell, group 2 denoted as the three possible combinations of the second cell, group 3 denoted as the three possible combinations of the third cell. According to our expectations for correct cases, A1-A2, B1-B2 and C1-C2 should be the minimum in each group, which provided an index for cell tracking

correction of the cost function. Also, if a particular cell tracking was correct, cell identifications should be consistent before and after interaction. If a particular cell tracking was incorrect (cell identification did not match), the cell was wrongly retagged then resulted in a wrong case and contributed to an error rate.

6. Cost Function Optimization

6.1 Application of Orthogonal Design

6.1.1 Orthogonal Design

Due to multiple factors existing in the cost function, a widely used mathematical method was introduced in the present work to optimize the factors. An orthogonal design (OD, n, s^m) with experimental runs, m factors and s levels is denoted by an $n * m$ matrix, where columns and rows are identified as factors and experimental runs, respectively [26, 27]. In an orthogonal design, all levels are symmetrically placed in each column to keep the levels uniformly for each factor, which is seen as a key construction to mate the levels for different factors [27]. In current research in many subjects as computer science, chemistry, and biology, experiments should be designed comprehensively and outcomes should also be analyzed with all influent facets. Thus, powerful and cost-efficient methods are significantly worth selecting to optimize experimental conditions, which would lead to better outcomes we desire. As a stepwise approach, orthogonal array designs can be found in many subjects [26-29]. In most experiment settings with multi factors and levels effects, it is extremely time-consuming if we try all possible combinations [29]. While in orthogonal design, only particular selected combinations need to be conducted as representative instead of a full-scale test, since they are

uniformly distributed in the research field and can greatly reflect a situation of the whole examined field ^[29]. The term of orthogonality also means each involved variable can be evaluated independently of one another ^[28]. With implementing complex factors with multiple levels, orthogonal design provides a rigorous and simultaneous opportunity to test all involved factors ^[30].

Hence, as a developed method, orthogonal design can be introduced to establish an overall view of optimizing factors in the cost function for better fitting in three-cell interactions, in which we can efficiently examine the particular ways each factor independently effects and results in an overall contribution to final outcomes, as well as determine ranges and values of the weighting factors to improve the cost function fitting in three-cell interaction events. Since the range and exact value of all weighting factors were not known initially, there would be infinite combinations for the potential to test. In order to determine the existing factors (integrated intensity, normality intensity, area, aspect ratio and position) with current measure scale, has the most impact on final cost results, we first assigned the five weighting factors with four levels of each, as an initial estimation. By using an orthogonal design, we can efficiently try most possibilities of the values with the least tests and avoid redundant tests. With five factors and four levels of each, a $L_{16}(4^5)$ orthogonal array is defined in Table 2 ^[31], so that only 16 tests were sufficiently able to provide an overall perception of ranges. For convenient calculation, the four levels of each weighting factor were designated as 0.25, 0.5, 0.75 and 1.

Table 2: $L_{16} (4^5)$ for Values of Weighting Factors Designation

TRIAL NO.	WEIGHTING FACTORS				
	w_{II}	w_{NI}	w_A	w_{AP}	w_P
1	0.25	0.25	0.25	0.25	0.25
2	0.25	0.5	0.5	0.5	0.5
3	0.25	0.75	0.75	0.75	0.75
4	0.25	1	1	1	1
5	0.5	0.25	0.5	0.75	1
6	0.5	0.5	0.25	1	0.75
7	0.5	0.75	1	0.25	0.5
8	0.5	1	0.75	0.5	0.25
9	0.75	0.25	0.75	1	0.5
10	0.75	0.5	1	0.75	0.25
11	0.75	0.75	0.25	0.5	1
12	0.75	1	0.5	0.25	0.75
13	1	0.25	1	0.5	0.75
14	1	0.5	0.75	0.25	1
15	1	0.75	0.5	1	0.25
16	1	1	0.25	0.75	0.5

6.1.2 Range Analysis

There are two significant parameters in the range analysis: K_{ji} and R_j . K_{ji} is defined as the sum of the evaluation indexes of all levels ($i, i = 1,2,3,4$) in each factor (j), and $\overline{K_{ji}}$ (mean value of K_{ji}) is used to determine the optimal level and optimal combination of factors: larger $\overline{K_{ji}}$, better level of respective factor is ^[31]. R_j is defined as the range between the maximum and minimum value of $\overline{K_{ji}}$, and is used for evaluating the importance of the factor: larger R_j , greater importance of the factor is ^[31].

However, in most research, they require highest yield of an outcome, so that both $\overline{K_{j_i}}$ and R_j are crucial for factor optimization. In contrast, we purely need minimum cost in the present work, so that the $\overline{K_{j_i}}$ should be set as small as possible (it seems conflicting that K should equal to 0). As a result, $\overline{K_{j_i}}$ analysis was not as meaningful as R_j , and only R_j was qualified for factor analysis in our work so that a most influential factor could be revealed.

6.1.3 Orthogonal Design Implementation

By applying an orthogonal design in three-cell interaction events, according to the results of range analysis (R_j), the result showed that positional information weights most among all five factors. Thus, we concluded that the positional part contributes most than other factors when they are all on a same level. In other words, by equivalently varying a value to all weighting factors, the greatest change in magnitude is observed in the positional part. The results of orthogonal test indicated a rough range of weighting factors, and relationship between each weighting part: when assign an equal value to each weighting factor, final cost is mostly contributed by the positional part; when assign different values to the weighting factors, a raw factor range can be achieved.

6.2 Velocity and Directionality Incorporation

Existing factors in the cost function include integrated intensity, normalized intensity, area, aspect ratio and position, which were used conditionally for cell tracking correction in previous two-cell interactions ^[25]. Apart from existing factors, we intended to find other distinct characteristics that could describe and measure cell motility clearly. Thus, based on

observation of cell motional trajectories, we incorporated two new factors, velocity and directionality, then the cost function can be improved as:

$$\begin{aligned} \text{FP cost} = & w_{II} * \left(\frac{II_1}{II_1} - \frac{II_2}{II_2} \right)^2 + w_{NI} * \left(\frac{NI_1}{NI_1} - \frac{NI_2}{NI_2} \right)^2 + w_A * \left(\frac{A_1}{A_1} - \frac{A_2}{A_2} \right)^2 + \\ & w_{AR} * \left(\frac{AR_1}{AR_1} - \frac{AR_2}{AR_2} \right)^2 + w_P * \left(\frac{[\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}]}{[\frac{d_1 + d_2}{2}]} \right)^2 + \\ & w_V * \left(\frac{V_1 - V_2}{\bar{V}} \right)^2 + w_D * (D_1 - D_2)^2 \end{aligned}$$

where w_V is the weighting factor for velocity of a cell, w_D is the weighting factor for moving directionality of a cell. To optimize the cost function, each value assigned for weighting factor can be varied. In our work, velocity was defined as the difference in velocity between post-interaction and pre-interaction. Directionality was defined as the difference in direction between post-interaction and pre-interaction, and was implemented by MATLAB built-in arctangent function: $P = \text{atan2}(Y, X)$ ^[32]. Then the two expression parts were incorporated to update the cost function.

7. Examination of Each Factor in the Cost Function

We designed different combinations of weighting factors in the cost function, in order to examine the independent effect that a factor would contribute to final outcome. The tests were conducted dependently in two parts: 1) keep other weighting factors as 1, vary position weighting factor from 0 to 2 with 0.25 interval (the larger position weighting factor, the larger cost values are, then when position weighting factor =2, final cost results are initially relying on positional part), as shown in Table 3; 2) determine each factor accuracy: keep just one weighting factor as 1 at a time, the others are 0, as shown in Table 4. All 89 cases were

comprehensively tested to obtain all cost results and corresponding accuracies.

Table 3: Design for Variation of Position Weighting Factor

TRIAL NO.	w_{II}	w_{NI}	w_A	w_{AP}	w_P	w_V	w_D
1	1	1	1	1	0	1	1
2	1	1	1	1	0.25	1	1
3	1	1	1	1	0.5	1	1
4	1	1	1	1	0.75	1	1
5	1	1	1	1	1	1	1
6	1	1	1	1	1.25	1	1
7	1	1	1	1	1.5	1	1
8	1	1	1	1	1.75	1	1
9	1	1	1	1	2	1	1

Table 4: Design for Variation of Each Weighting Factor

TRIAL NO.	w_{II}	w_{NI}	w_A	w_{AP}	w_P	w_V	w_D
1	1	0	0	0	0	0	0
2	0	1	0	0	0	0	0
3	0	0	1	0	0	0	0
4	0	0	0	1	0	0	0
5	0	0	0	0	1	0	0
6	0	0	0	0	0	1	0
7	0	0	0	0	0	0	1

8. Condition Optimization

The accuracies of each test obtained from Table 3 and Table 4 respectively are designed to determine how change in position weighting factor affects accuracy with all other factors

held constant, and how each factor independently affects accuracy. The results thus should reveal the factors for optimal performance of the cost function and how that performance compares with the Kilfoil benchmark method.

9. Method Validation

We examined 5 new data sets to validate the method and examined corresponding accuracies of the three-cell interactions in the new data sets: *7Percent_MedDensity_Sample1*, *7Percent_MedDensity_Sample2* and *7Percent_MedDensity_Sample3*, in which the cell density were as a same density of 5 000 cells/cm², and *7Percent_MedHighDensity_Sample1*, *7Percent_MedHighDensity_Sample2*, in which the cell density were 10 000 cells/cm². In order to get a better calculation, the events were classified into two conditions that will be elaborated in the Results, and their code was shown in Appendix 3 to Appendix 6, respectively. By detecting the accuracies, we could analyze whether the optimized cost function and its executed condition could lead to increasing in accuracy.

Results

1. Accuracy of the Kilfoil Benchmark Method

By analyzing all three-cell interaction events in the data set of *7Percent_MedDensity_Sample5*, excluding the cases where cell information were incomplete and tagged cells which did not interact or other abnormal conditions, there were 89 events which could be well analyzed for accuracy determination and cost function optimization. As a result, there were 75 events where all the three cells were correctly identified after cell interactions. In contrast, there were 14 events where cells were incorrectly identified after interactions, showing that the accuracy of three-cell interactions that could be achieved was 84.27% by the Kilfoil benchmark method.

The error events were mainly observed to be when two cells occluded for several consecutive frames during interaction and the linking system could not distinguish them, thus causing their information, including tagged IDs, to be missing. As the interaction completed, their IDs reappeared but were incorrectly switched.

By analyzing all the three-cell interaction events, we observed two characteristics: a) interacting duration during which the event occurred, measured by how many frames that the three cells were sharing at least one parent contour. An example event is shown in Figure 8. b) a frame gap in which one or two involved cells were occluded and lost information, including identification for consecutive frames, since the linking system could not distinguish occluded cells in that particular interval. An example event is shown in Figure 9. The frame gap determination has been addressed in the corresponding code. For instance, 1-frame gap indicates all cell information that could be tracked consecutively by a single frame before and

a single frame after the interaction, 2-frame gap indicates that all cell information completed two frames before and two frames after the interaction, etc. Detailed data was obtained and shown in Table 5 and Table 6.

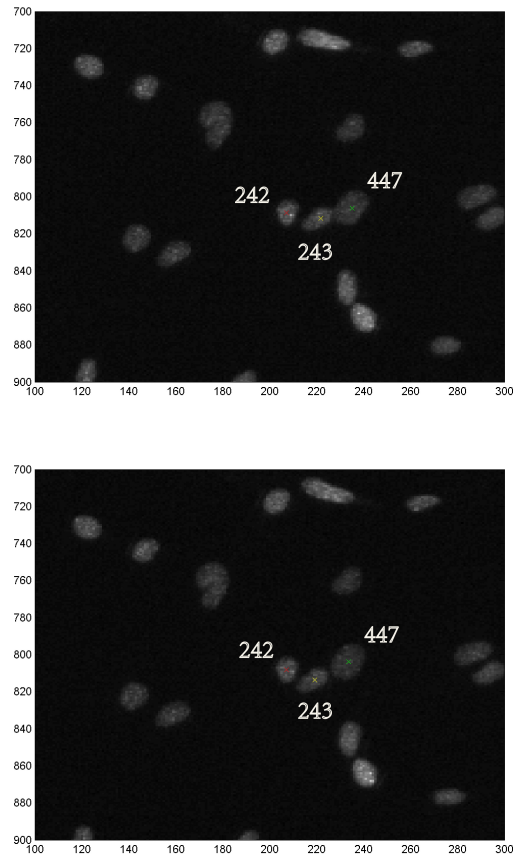


Figure 8: A three-cell interaction event that has two-interacting frames, T=138 min and 141 min, from top to bottom

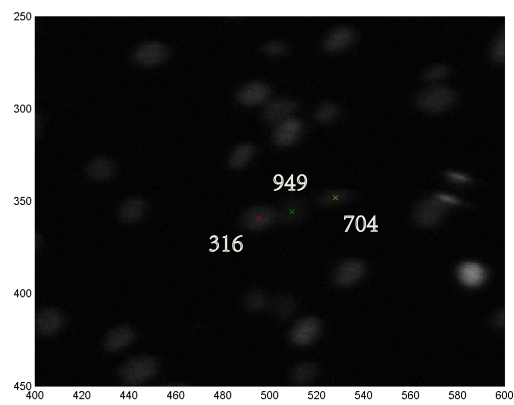
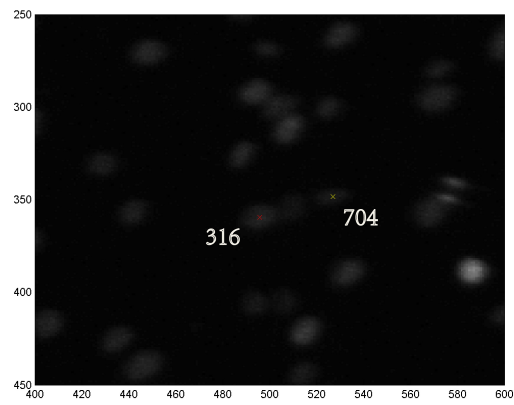
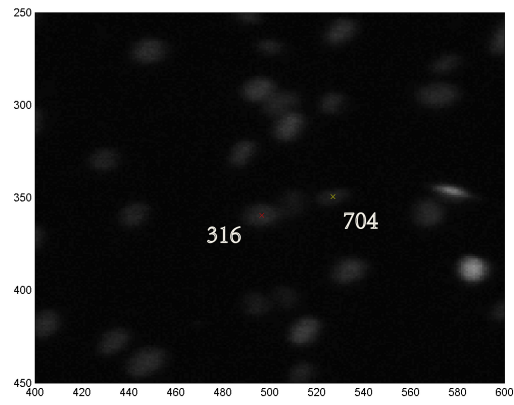
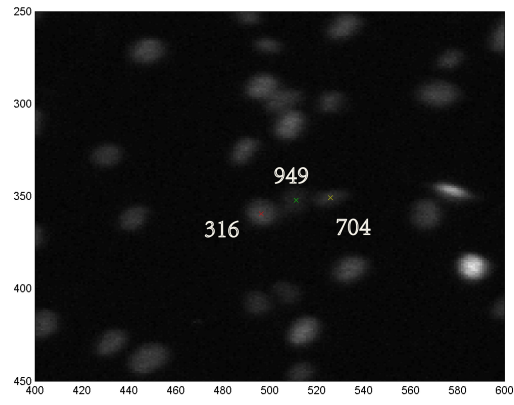


Figure 9: A three-cell interaction event of three-frame gaps, T=858 min, 861 min, 864 min and 867 min, from top to bottom. When T=861 min and 864 min, the tagged 949 cell could not be identified

Table 5(A): Sum of Correct Cases for Different Interacting Frames

TOTAL NUMBER OF INTERACTING FRAME	TOTAL NUMBER OF CORRECT CASE	PERCENTAGE
1	28	37.33%
2	19	25.33%
3	10	13.33%
4	4	5.33%
5	5	6.67%
6	3	4.00%
7	1	1.33%
8	3	4.00%
9	1	1.33%
10	1	1.33%
SUM	75	1

Table 5(B): Sum of Incorrect Cases for Different Interacting Frames

TOTAL NUMBER OF INTERACTING FRAME	TOTAL NUMBER OF INCORRECT CASE	PERCENTAGE
1	10	71.43%
2	2	14.29%
3	1	7.14%
9	1	7.14%
SUM	14	1

Table 6(A): Sum of Correct Cases for Different Frame Gaps

TOTAL NUMBER OF FRAME	TOTAL NUMBER OF CORRECT	PERCENTAGE
GAP	CASE	
1	59	78.67%
2	1	1.33%
3	4	5.33%
4	1	1.33%
5	1	1.33%
6	4	5.33%
7	1	1.33%
9	1	1.33%
10	1	1.33%
11	2	2.67%
SUM	75	1

Table 6(B): Sum of Incorrect Cases for Different Frame Gaps

TOTAL NUMBER OF FRAME	TOTAL NUMBER OF INCORRECT	PERCENTAGE
GAP	CASE	
1	1	7.14%
4	3	21.43%
5	2	14.29%
6	1	7.14%
7	3	21.43%
8	1	7.14%
10	1	7.14%
11	1	7.14%
14	1	7.14%
SUM	14	1

Figure 8 shows a two-interacting frame event presented by filtered images. The tagged 242, 243 and 447 cells were sharing one parent contour during these two frames in this particular event. Figure 9 shows a three-frame gaps event, where begins with the cells that were interacting when $T=858$ and captured consecutively, afterward tagged 949 cell lost its identification in consecutive two frames, then showed up when $T=867$. Whereby the frame gap was 3 as defined above.

In regards to various interacting frames of all the events from Table 5, 1-interacting frame cases (38 out of 89, 42.7%) and 2-interacting frames cases (21 out of 89, 23.6%) account for more than half of the total cases. However, we could not arrive at a clear conclusion about an obvious tendency to indicate whether shorter or longer interacting frames are preferable for correct cell identifications in the Kilfoil benchmark method.

In regards to various frame gaps of the events from Table 6, 1-frame gap cases (59 out of 89, 66.3%) shows all cell information were existing during the whole process of an event, also, suggests that all cell information integrated in a single frame before and a single frame after the interaction were more likely to have correct cell identifications. As for more than 1-frame gap events, the pattern was not clear.

Thus, the analysis of duration and frame gap could provide a general condition of correct events and incorrect events, including potential strategy for future cost function optimization.

2. Orthogonal Test Results

From all the Kilfoil correct cases, we chose 40 cases to determine the potential ranges and

relationships among the existing factors. The values were designated as Table 3 shown above, and then the cost function was applied to calculate orthogonal results. Due to each event having individual fluctuation in measurements, it was more reliable to present the order of R_j s of the factors instead of exact value. In each case, we denoted digit 1 as the most important factor that weighted most among all the factors, then digit 2 represented the second important factor, and so on, digit 5 represented the least important factor. Then we tested all cases and obtained mean values of the order. The results are shown in Table 7.

Table 7: Results of Orthogonal Test

	Integrated Intensity	Normalized Intensity	Area	Aspect Ratio	Position
No.	3.17	4.73	3.09	2.87	1.12

Thus, we can conclude from Table 7 that the positional part would weigh most among all the factors, which means that by assigning a same value to each weighting factor, the positional part contributed most to the final cost result, while integrated intensity part, cell area part and aspect ratio part had approximate importance, and normalized intensity part weighted least. In other words, by increasing the value of position weighting factor, the cost results would be more influenced by the positional part, until thoroughly relied on the positional part.

The results indicated a future perception of weighting factor optimization. We should examine a minimum value of the position weighting factor that co-effect with other factors, which could lead to a same accuracy compared to the marginal case where only position factor could have effect on cost results. Thus, any values assigned to the position weighting factor above this minimum value would achieve the same impact on the final cost results. As

a result, this particular value would be the maximum of the range of the position weighting factor. Thus, we should examine the effect and accuracy of each factor since they were all involved in the cost determination, and apart from the weighting values, it was also necessary to select the factors themselves.

3. Examination of Factors in the Cost Function

We first incorporated velocity and directionality parts to update the cost function. Second, we examined how each factor effected and contributed to the final cost results. Thus, we could obtain an overall view of how each factor performs.

3.1 Determination of Position Weighting Factor Range

As shown by the design outlined in Table 3, we tested 9 combinations for each three-interaction event and obtained corresponding accuracy that in how many cases, cells could be correctly identified among all cases. The results are shown in Table 8.

Table 8: Results for Variation of Position Weighting Factor

TRIAL NO.	w_{II}	w_{NI}	w_A	w_{AP}	w_P	w_V	w_D	ACCURACY
1	1	1	1	1	0	1	1	44.94%
2	1	1	1	1	0.25	1	1	51.69%
3	1	1	1	1	0.5	1	1	51.69%
4	1	1	1	1	0.75	1	1	55.06%
5	1	1	1	1	1	1	1	56.18%
6	1	1	1	1	1.25	1	1	58.43%
7	1	1	1	1	1.5	1	1	58.43%
8	1	1	1	1	1.75	1	1	59.55%
9	1	1	1	1	2	1	1	60.67%

From Table 8, we learned that in the range from 0 to 2, the positional part had slightly increasing impact to the final cost results, as well as accuracy. When position weighting factors reached 2 with a combination where other weighting factors equaled to 1, the accuracy achieved as same as a cost result contributed by the pure positional part, which is addressed below.

Thus, regardless of other factors, positional analysis would be able to achieve a highest accuracy of 60.67% for three-cell interactions. On the other hand, the positional part had been further shown to be the most weighted part; hence, when co-effects with other weighting parts, the range could be narrowed as 0 to 2 with the simplest scale level.

3.2 Determination of Each Factor

As shown by the design outlined in Table 5, we tested 7 combinations for each three-cell interaction event, under the condition that only one weighting part exists at a time to

contribute to final cost. We obtained corresponding accuracy of each weighting part, which demonstrated the ability of correctly identifying cells. The results are shown in Table 9.

Table 9: Results for Variation of Each Weighting Factor

TRIAL NO.	w_{II}	w_{NI}	w_A	w_{AP}	w_P	w_V	w_D	ACCURACY
1	1	0	0	0	0	0	0	20.22%
2	0	1	0	0	0	0	0	22.47%
3	0	0	1	0	0	0	0	21.35%
4	0	0	0	1	0	0	0	12.36%
5	0	0	0	0	1	0	0	60.67%
6	0	0	0	0	0	1	0	6.74%
7	0	0	0	0	0	0	1	38.20%

From Table 9, we found that characteristic of position could achieve the highest accuracy in cell tracking identification, which means positional analysis could describe cell moving trajectories better and serve in identifying cells during three-cell interactions. The pure positional part achieved the same accuracy comparing to the design that when other weighting factors were set to 1 and position weighting factor was set to 2, as No. 9 trail showed in Table 8. However, the accuracy was not as high as the Kilfoil benchmark method, in which tracking cells by positional information.

The two results described above demonstrate that the two new factors that were systematically studied the accuracies were not as high as we imagined before, that modification of the *ACTIVE* algorithms could allow *ACTIVE* to outperform the Kilfoil benchmark method. The reason might be that the conditions were more complicated to

analyze in three-cell interactions. First, the cost combinations increase exponentially for higher-level cell interactions. For example, the combinations would be 4 (2^2) in two-cell interactions, 9 (3^2) in three-cell interactions, 16 (4^2) in four-cell interactions, etc., which result in more cell pairs to compare. Second, apart from measurements for position, the measuring scale of other weighting features were mostly ranging from infinitesimal to 1, where even a small fluctuation in measurements would be difficult to distinguish the minimum cost. Third, as we found in Table 10, not every factor could be fitting for higher-level cell interactions, accuracy may decrease by improper factors if we incorporate all of them.

4. Modification of *ACTIVE* Execution Condition

From Table 9, we found that directionality was able to achieve a second highest accuracy among all factors. We also found that for the 14 incorrect cases, pure directionality factor could be able to correct 11 cases, which achieved an accuracy of 78.57% for incorrect cases, while higher than other factors. The relationship between the number of frame gaps and validity of each incorrect case are shown in Table 10.

Table 10: Frame Gap Pattern of Incorrect Three-cell Interaction Cases

EVENT NO.	TOTAL NUMBER OF FRAME GAP	WHETHER CORRECT OR NOT
258	14	Correct
297	11	Correct
247	10	Correct
129	8	Correct
86	7	Correct
92	7	Correct
110	7	Correct
75	6	Incorrect
20	5	Incorrect
64	5	Correct
62	4	Incorrect
80	4	Correct
161	4	Correct
151	1	Correct

We already learned from Table 6 that 1-frame gap events were more likely to have correct tracking based on the Kilfoil benchmark method. While we learnt from Table 10 that incorrect events were more likely to have more than a single frame gap. Thus, we counted a total number of correct cases determined by directionality and compared it with correct cases determined by the Kilfoil benchmark method, according to different frame gaps. The results are shown in Table 11.

Table 11: Conclusion for Total Number of Correct Cases Determined by Directionality and the Kilfoil

Benchmark Method		
TOTAL NUMBER OF FRAME GAP	TOTAL NUMBER OF CORRECT CASE DETERMINED BY DIRECTIONALITY	TOTAL NUMBER OF KILFOIL CORRECT CASE
1	16	59
2	1	1
3	2	4
4	2	1
5	2	1
6	2	4
7	3	1
8	1	1
9	1	0
10	1	1
11	2	2
12	0	0
13	0	0
14	1	0

From Table 11, we see that a highest accurate strategy could be tested, when frame gap was no less than 3 frames, directionality could be used to determine cell identification after three-cell interactions, when frame gap was less than 3 frames, the Kilfoil benchmark method could be an alternative. By this combined method, we could achieve a new accuracy of 88.76%, which was 4.5% higher than the Kilfoil benchmark method (84.27%).

5. Method Validation

The code for event information and cost function implementation was optimized for validation. Overall, the main highlighted code was shown in Appendix 5, Appendix 6, Appendix 7 and Appendix 8. Appendix 5 and Appendix 6 mainly included the implementation of cost function for different conditions, Appendix 7 and Appendix 8 were circumstances determination sections.

The code for events having only one interacting frame was shown in Appendix 3 and Appendix 5. In Appendix 3, the code worked for one-interacting frame events and cell information, whereby before and after interaction were collected and stored in *xyzs_info*. Then the variables and assigned weighting factors were called in the function shown in Appendix 5. By execution of the function, the outputs would be two variables stored in *cost*, as *cost_vector* and *framegap*. All nine possible combinations of three-cell interactions were stored in *cost_vector*, and it would be used in circumstances determination as well as *framegap*.

The code for the events having at least two interacting frames is shown in Appendix 4 and Appendix 6. All meanings and purposes of variables were identical with one-interacting frame events as addressed above. The only difference in the information collection of multi-interacting frame events was that we both recorded the first interacting frame and the last interacting frame, during which period the cell behavior could be omitted since the condition during interaction would be difficult and unnecessary to clarify. This setting was mainly used for velocity and directionality cost calculation. Identically, *cost_vector* and *framegap* would be the two outputs for circumstances determination.

The code for circumstances determination was shown in Appendix 7 and Appendix 8. With the input variables of *cost_vector* and *framegap*, the code shown in Appendix 7 was able to implement the strategy that when frame gap was less than 3 frames, the Kilfoil benchmark method remained, when frame gap was no less than 3 frames, directionality would be selected after three-cell interactions. Then if the respective combination was a minimum in each cell pair, the cell identification would keep the same. If not, then the cost results would be used in the script shown in Appendix 8, a minimum combination would be viewed as correct cell identification and the other two would be switched, and relabeled in the corresponding figure. Thus we could identify whether each cell was correctly tracked.

5.1 Test on Data Sets of Same Density

We first tested the strategy for three new data sets: *7Percent_MedDensity_Sample1*, *7Percent_MedDensity_Sample2* and *7Percent_MedDensity_Sample3*, in which the cell seeding density was 5 000 cells/cm² in static wrinkle substrates. We examined 57 three-cell interaction cases in total and the new strategy was able to obtain an accuracy of 85.96% (49 out of 57). While the Kilfoil benchmark method could achieve an accuracy of 92.98% (53 out of 57).

5.2 Test on Data Sets of A Higher Density

Then we tested the strategy for another two data sets: *7Percent_MedHighDensity_Sample1*, *7Percent_MedHighDensity_Sample2*, in which the cell seeding density was 10 000 cells/cm² in static wrinkle substrates. We examined 80 three-cell interaction cases total and the new

strategy was able to obtain an accuracy of 86.25% accuracy (69 out of 80). While the Kilfoil benchmark method could achieve an accuracy of 91.25% (73 out of 80).

Discussion

In this work, two strategies for improving accuracy during tracking of cells during three-cell events were studied. The results show that neither morphological nor mobility data could outperform a purely positional approach for cell tracking under these particular circumstances. The results further provide valuable insight that we still could obtain meaningful evaluation from the work and predict future improvements for higher level of cell interactions. Not only the factors in the cost function can be selected, but also the algorithm of *ACTIVE* or the expression of cost function can be optimized in future work.

By reviewing the whole work, we first extracted the three-cell interactions from the data set and analyzed its tracking accuracy by the Kilfoil benchmark method. The result showed 84.27% accuracy that there might be a potential capacity for enhancing cell tracking correction by the cost function. In our further analysis of the interacting events, we counted the total number of interacting frame and frame gap for correct cases and incorrect cases, respectively. Since these two indexes elaborated a general interacting duration based on a contour-defined pre-process and a time gap that a cell could not be traced, respectively. In previous work of *ACTIVE* in two-cell interactions, it has been reported that more frame gaps would result in higher possibility, where the Kilfoil method incorrectly identifies the cells^[25], due to positional analysis would no longer reliable. In our work, in contrast with interacting frame, Table 7(A) shows that completed cell information in consecutive frames were prone to be correctly tracked by the Kilfoil method, which was similar with the conclusion before. Thus, the frame gap could be mainly taken into account in our analysis, by aiming at how to correct cell tracking with those cases with more frame gaps.

Prior to improving the cost function, we modified the implementation of the code and had it adapted for three-cell interactions in our work. The increase of potential combinations for three cells became the first challenge in our method, which is shown in Table 1. In two-cell interactions, there were only four potential combinations needed for comparison. While in three-cell interactions, the potential combinations increased to nine, resulted in a more complicated condition that each cell needed to be compared three times. Then we modified the code of cost function for accurately applying for three-cell interactions. In this process, however, much higher accurate assessments for the characteristics of the cells were required. Once the cells had nearly equal measurements in a particular magnitude, it was challenging to tell the difference between each of them, whereby we tried to distinguish their identification before and after interaction. Even a tiny fluctuation could result in a wrong combination value, and the final cost and cell identification. Thus, based on current measured magnitudes, we had to accurately examine the factors that contributed to the final cost, including the ranges and relationships of them, as well as whether they could effect as we expected and search for other potential features expressed during cell migration.

From the first approach, with the optimization of existing factors, including integrated intensity, normal intensity, area, aspect ratio and position, we tried to testify potential ranges and relationships among them by a introduced mathematical method: orthogonal design, based on the particular measurement scale. With a comprehensive and efficient examination (shown in Table 3), orthogonal testing provided a reliable result, showing that the position factor weighted and contributed more than other factors to the cost. Which means, we could vary the values of each weighting factors and balance them in a reasonable range in order to

acquire desirable results. In this particular range, the positional part could act as a maximum limit. In other words, generally, while in certain circumstance where all the five factors were co-effecting, once a particular value of the position weighting factor has been set, the final cost would not have obvious changes when other weighting factors were set on any values within the range. So that the final cost would rely on the position part, and then applied in the selection of correct cell combination. As a widely used statistical methodology, orthogonal design can be used in many research fields. In our work, it was also fitting for range determination of various factors on various levels.

What is more, based on carefully observation of the cell migration, we introduced two new factors into the cost function: velocity and directionality. Due to the particular static wrinkled substrate that was used for cell culture, it was indicated that cells preferred to migrate parallel to the wrinkle direction ^[12], which means that the directionality could be evaluated as a meaningful factor in our work. Then we assessed the relationships among all the factors and individual effect. With a co-effect with other factors, the position weighting factor could be set in a range from 0 to 2, while it equaled to 2, the accuracy could be achieved as highest as 60.67%. On the other hand, we also examined the effect of each factor, which was necessary because, though they were raised and worked well in two-cell interactions, they might not fit for more complicated three-cell interactions, due to the selection of the correct cost combination being more restricted as explained above. The dependent highest accuracy was achieved by positional part, while others were too inaccurate to be viewed as efficient factors.

We next considered a second approach: concluded an alternative approach: modification of the circumstances under which the Kilfoil and the cost function were executed according to

the frame gap. Since less frame gap cases prone had correct cell tracking by the Kilfoil benchmark method, while more frame gap cases could use directionality to correct cell tracking due to their slight higher accuracy shown in Table 10. Thus, a new strategy was established, as directionality could be applied to determine cell identification when frame gap was no less than 3, and the Kilfoil benchmark method could be an alternative when the frame gap was less than 3. In this particular strategy, we could achieve a new accuracy with the combined method with 88.76%, which was 4.5% higher accuracy than the Kilfoil benchmark method (84.27%). First, cells preferred to move along the wrinkle on the substrate so that directionality could be able to describe cell migration, in which cells were more likely to keep their moving angles during the interactions. Second, positional information could hardly identify cells if they occluded for several frames after interactions as previous *ACTIVE* work had testified, in addition where directionality also behaved well in this particular condition, so that we took an example that the new strategy was based on a frame gap.

Based on the new strategy, we completed the final code of cost function implementation, as shown in Appendix 5 and Appendix 6. The code was divided into two versions that fit for two interacting conditions respectively. There were two innovations of the code, update for three-cell interactions, and velocity and directionality incorporation. Then with the calculated cost results, the code in Appendix 7 and Appendix 8 could be able to finalize the strategy according to frame gap.

However, though we had prepared sufficient factors work for the cost function by examining the range of weighting factors and individual impact of each factor, the new strategy was not found to reproducibly increase cell tracking accuracy when applied to

multiple data sets. Thus, the original and improved cost function could not further improve correct cell tracking for three-cell interactions, and still needs further investigation and modification.

In future improvement of *ACTIVE* algorithm, first, the modification of cost function should be aimed. We illustrated the principle of combinations for different cell pairs, the condition was simplest in two-cell interactions, for each case had only four combinations to compare. However, in higher level cell interactions, the combinations increase exponentially, for example, there were 9 combinations for three-cell interactions, there should be 16 for four-cell interactions, 25 for five-cell interactions, etc., which resulted in a more and more complex condition where the comparisons are confused to identify. Without an extreme precise measurement of the factors, it is difficult to correctly identify respective cells during the tracking process. What is more, the algorithm of the cost function can be modified to decrease comparisons for higher level of cell interactions, for instance, each cell can be compared only once in the cost calculation. Since more cell pair combinations will result in duplicated comparisons and reducing accuracy. Hence, in future work, based on the established innovative cell interaction events in *ACTIVE*, i.e. cell division and merging, the cost function can be potentially altered to another form, expression, or algorithm.

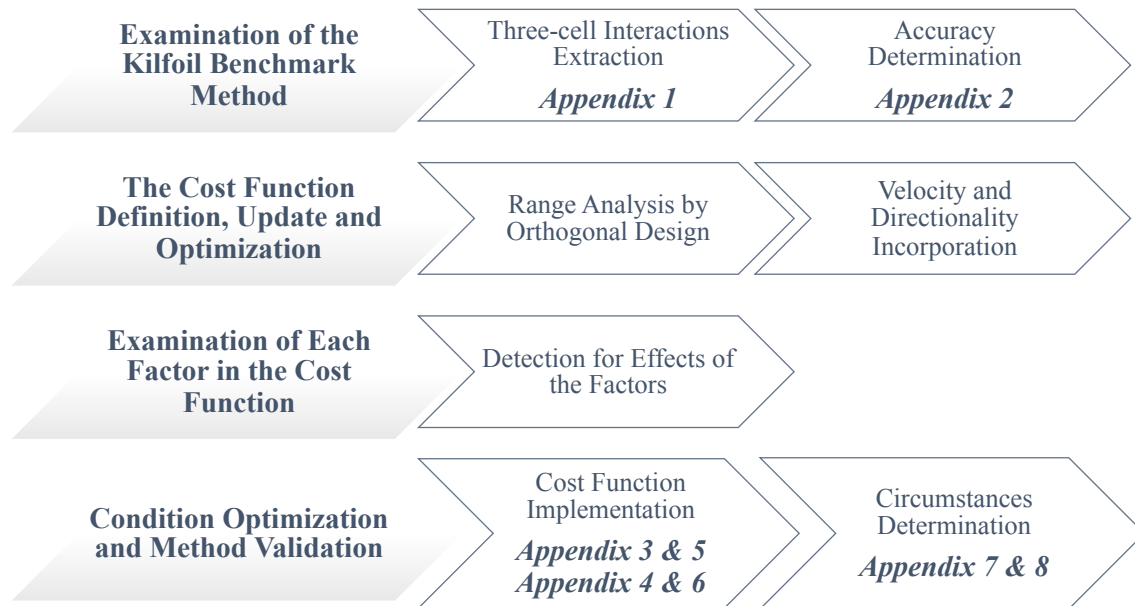
Second, factors selection is still worth studying. Since in three-cell interactions, the results suggested that the five original factors had various low accuracies in identifying cells after interactions, due to the complex cost combinations. So that it is significant to select proper factors that could be able to accurately identify the identical cells after interactions. What is more, specific features of different substrates, which also are worth further investigation in a

way, can influence cell behavior. Such as directionality in our work, it could predict cell migrate direction to some extent. In brief, the characteristics for analyzing cell behaviors over long time not only require the properties of cells themselves, but also the properties of specific substrates that how the cells prefer to migrate on those substrates.

In conclusion, though positional analysis appears to be a more accurate method than morphological and mobility analysis for cell tracking under the current circumstances, as we did not achieve higher accuracy by the cost function inherited from previous *ACTIVE* work for three-cell interactions, further improvement is worth studying. Not only is it possible to select more accurate factors to assess cell migration, but also to optimize the algorithm of *ACTIVE* and its cost function for higher level of cell-cell interactions.

Appendices

We attached all necessary code in this Appendices part. In order to have a brief overlook, a modified flow chart was presented below in Scheme 2, which derived from the one presented in Methods. In Scheme 2, we highlighted the crucial code in corresponding step.



Scheme 2: Main sections completed by the present work, attaches with bold text that indicates corresponding Appendix

1. Script: Three-cell Interaction Cases and Frame Information Extraction

```
cell_id_matrix=mult_array(:,2:4);
n=1;
% select and sort three cell event
for i=1:length(cell_id_matrix);
    if length(cell_id_matrix{i})==3;
        three_cell_event_matrix{n,1}=cell_id_matrix{i,1}';
        three_cell_event_matrix{n,2}=cell_id_matrix{i,2};
        three_cell_event_matrix{n,3}=cell_id_matrix{i,3};
        n=n+1;
    end
end
```

```

end
% sort cell id to a single column
p=1;
for m=2:n-1;

cell_id_array(p:p+5,1)=[three_cell_event_matrix{m-1},three_cell_event_matri
x{m}]';
    p=p+3;
end
% check whether a cell ID is unique
m=1;
n=1;
for i=1:length(cell_id_array);
    if cell_id_array(n)==cell_id_array(n+1);
        check(m)=cell_id_array(n+1);
        n=n+1;
        m=m+1;
    end
end
end
% sort cell information(output:matrix)
sum=0;
for j=1:length(cell_id_array);
    bool_id=xyzs_id(:,13)==cell_id_array(j);
    cell_info=xyzs_id(bool_id);
    matrix(sum+1:sum+length(cell_info),1:13)=xyzs_id(bool_id,1:13);
    sum=sum+length(cell_info);
end
end
% get all frame information
image_info = imfinfo('7Percent_MedDensity_Sample1.tif');

```

2. Script: Plot Cells in Each Frame

```

j=47; % event no.
m=3*(j-1)+1; % cell id order

% particular frame
I = imread('7Percent_MedDensity_Sample3.tif', 477, 'Info', image_info);
figure;
imagesc(I)
colormap('gray')
axis([800 1000 400 600])
hold on

```

```

% plot each cell
A=find(matrix(:,13)==cell_id_array(m));
B=find(matrix(:,12)==three_cell_event_matrix{j,2});
C=intersect(A,B,'rows');
plot(matrix(C(3),1)',matrix(C(3),2)', 'rx')

m=m+1;
A=find(matrix(:,13)==cell_id_array(m));
B=find(matrix(:,12)==three_cell_event_matrix{j,2});
C=intersect(A,B,'rows');
plot(matrix(C(1),1)',matrix(C(1),2)', 'yx')

m=m+1;
A=find(matrix(:,13)==cell_id_array(m));
B=find(matrix(:,12)==three_cell_event_matrix{j,2});
C=intersect(A,B,'rows');
plot(matrix(C(1),1)',matrix(C(1),2)', 'gx')

saveas(gcf, '477.tiff')

```

3. Script: Collection of Each Cell Information for 1 Interacting Frame

```

% 1 interacting frame
% obtain cell id information
event_no=11;
id=3*(event_no-1)+1; % first cell id no.

% first cell
A=find(matrix(:,13)==cell_id_array(id));
B=find(matrix(:,12)==three_cell_event_matrix{event_no,2});
C=intersect(A,B,'rows');
xyzs_id1_f1=matrix(C(1)-1,:);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
xyzs_id1_fi=matrix(C(1),:); % interact frame
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
xyzs_id1_f2=matrix(C(1)+6,:);

% second cell
id=id+1;
A=find(matrix(:,13)==cell_id_array(id));
B=find(matrix(:,12)==three_cell_event_matrix{event_no,2});
C=intersect(A,B,'rows');
xyzs_id2_f1=matrix(C(1)-6,:);

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
xyzs_id2_fi=matrix(C(1),:); % interact frame
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
xyzs_id2_f2=matrix(C(1)+6,:);

% third cell
id=id+1;
A=find(matrix(:,13)==cell_id_array(id));
B=find(matrix(:,12)==three_cell_event_matrix{event_no,2});
C=intersect(A,B,'rows');
xyzs_id3_f1=matrix(C(1)-6,:);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
xyzs_id3_fi=matrix(C(1),:); % interact frame
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
xyzs_id3_f2=matrix(C(1)+6,:);

xyzs_info(1,:)=xyzs_id1_f1;
xyzs_info(2,:)=xyzs_id1_fi;
xyzs_info(3,:)=xyzs_id1_f2;
xyzs_info(4,:)=xyzs_id2_f1;
xyzs_info(5,:)=xyzs_id2_fi;
xyzs_info(6,:)=xyzs_id2_f2;
xyzs_info(7,:)=xyzs_id3_f1;
xyzs_info(8,:)=xyzs_id3_fi;
xyzs_info(9,:)=xyzs_id3_f2;

% call the cost function to calculate cost
cost_1(w_int_intensity, w_norm_intensity, w_area, w_aspect, w_position,
w_velocity, w_direction, xyzs_info, frame_avg, xyzs_id_columns);
% call the category function to implement the strategy
category(cost_vector, framegap);

```

4. Script: Collection of Each Cell Information for More Than 1 Interacting

Frames

```

% more than 1 interacting frame
% obtain cell id information
event_no=44;
id=3*(event_no-1)+1; % first cell id no.

% first cell
A=find(matrix(:,13)==cell_id_array(id));

```

```

B=find(matrix(:,12)==three_cell_event_matrix(event_no,2));
C=intersect(A,B,'rows');
xyzs_id1_f1=matrix(C(2)-7,:);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
xyzs_id1_fi1=matrix(C(2),:); % 1st interact frame
xyzs_id1_fi2=matrix(C(2)+9,:); % 2nd interact frame
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
xyzs_id1_f2=matrix(C(2)+16,:);

% second cell
id=id+1;
A=find(matrix(:,13)==cell_id_array(id));
B=find(matrix(:,12)==three_cell_event_matrix(event_no,2));
C=intersect(A,B,'rows');
xyzs_id2_f1=matrix(C(1)-6,:);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
xyzs_id2_fi1=matrix(C(1),:); % 1st interact frame
xyzs_id2_fi2=matrix(C(1)+9,:); % 2nd interact frame
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
xyzs_id2_f2=matrix(C(1)+16,:);

% third cell
id=id+1;
A=find(matrix(:,13)==cell_id_array(id));
B=find(matrix(:,12)==three_cell_event_matrix(event_no,2));
C=intersect(A,B,'rows');
xyzs_id3_f1=matrix(C(1)-7,:);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
xyzs_id3_fi1=matrix(C(1),:); % 1st interact frame
xyzs_id3_fi2=matrix(C(1)+6,:); % 2nd interact frame
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
xyzs_id3_f2=matrix(C(1)+7,:);

xyzs_info(1,:)=xyzs_id1_f1;
xyzs_info(2,:)=xyzs_id1_fi1;
xyzs_info(3,:)=xyzs_id1_fi2;
xyzs_info(4,:)=xyzs_id1_f2;

xyzs_info(5,:)=xyzs_id2_f1;
xyzs_info(6,:)=xyzs_id2_fi1;
xyzs_info(7,:)=xyzs_id2_fi2;
xyzs_info(8,:)=xyzs_id2_f2;

xyzs_info(9,:)=xyzs_id3_f1;

```

```

xyzs_info(10,:)=xyzs_id3_f1;
xyzs_info(11,:)=xyzs_id3_fi2;
xyzs_info(12,:)=xyzs_id3_f2;

% call the cost function to calculate cost
cost_2(w_int_intensity, w_norm_intensity, w_area, w_aspect, w_position,
w_velocity, w_direction, xyzs_info, frame_avg, xyzs_id_columns);
% call the category function to implement the strategy
category(cost_vector, framegap);

```

5. Cost Function for 1 Interacting Frame ^[12]

```

function [cost_vector, framegap]=cost_1(w_int_intensity, w_norm_intensity,
w_area, w_aspect, w_position, w_velocity, w_direction, xyzs_info, frame_avg,
xyzs_id_columns)
% 1 interacting frame

cost_vector=zeros(8,9);
xyzs_id1_f1=xyzs_info(1,:);
xyzs_id1_fi=xyzs_info(2,:);
xyzs_id1_f2=xyzs_info(3,:);
xyzs_id2_f1=xyzs_info(4,:);
xyzs_id2_fi=xyzs_info(5,:);
xyzs_id2_f2=xyzs_info(6,:);
xyzs_id3_f1=xyzs_info(7,:);
xyzs_id3_fi=xyzs_info(8,:);
xyzs_id3_f2=xyzs_info(9,:);

framegap=xyzs_id1_fi(xyzs_id_columns-1)-xyzs_id1_f1(xyzs_id_columns-1);

id1_frame1=xyzs_id1_f1(xyzs_id_columns-1);
% id2_frame1=xyzs_id2_f1(xyzs_id_columns-1);
% id3_frame1=xyzs_id3_f1(xyzs_id_columns-1);
id1_frame2=xyzs_id1_f2(xyzs_id_columns-1);
% id2_frame2=xyzs_id2_f2(xyzs_id_columns-1);
% id3_frame2=xyzs_id3_f2(xyzs_id_columns-1);

% Extract frame averages from frame_avg
avg_norm_intensity1 = frame_avg(7,id1_frame1); %average intensity/area
avg_norm_intensity2 = frame_avg(7,id1_frame2);
avg_area1 = frame_avg(5,id1_frame1); %average area
avg_area2 = frame_avg(5,id1_frame2);
avg_int_intensity1 = frame_avg(6,id1_frame1); %average integrated intensity
avg_int_intensity2 = frame_avg(6,id1_frame2);

```

```

avg_diameter1 = (2*frame_avg(1,id1_frame1) + 2*frame_avg(2,id1_frame1))/2; %
average of major and minor diameter
avg_diameter2 = (2*frame_avg(1,id1_frame2) + 2*frame_avg(2,id1_frame2))/2;

% Extracts the intensity
int_intensity1=xyzs_id1_f1(7);
int_intensity2=xyzs_id2_f1(7);
int_intensity3=xyzs_id3_f1(7);
int_intensity4=xyzs_id1_f2(7);
int_intensity5=xyzs_id2_f2(7);
int_intensity6=xyzs_id3_f2(7);

% Extracts the areas
area1=xyzs_id1_f1(6);
area2=xyzs_id2_f1(6);
area3=xyzs_id3_f1(6);
area4=xyzs_id1_f2(6);
area5=xyzs_id2_f2(6);
area6=xyzs_id3_f2(6);

% Calculate the average intensity
norm_intensity1=int_intensity1/area1;
norm_intensity2=int_intensity2/area2;
norm_intensity3=int_intensity3/area3;
norm_intensity4=int_intensity4/area4;
norm_intensity5=int_intensity5/area5;
norm_intensity6=int_intensity6/area6;

% Calculate the aspect ratio
aspect1=xyzs_id1_f1(3)/xyzs_id1_f1(4);
aspect2=xyzs_id2_f1(3)/xyzs_id2_f1(4);
aspect3=xyzs_id3_f1(3)/xyzs_id3_f1(4);
aspect4=xyzs_id1_f2(3)/xyzs_id1_f2(4);
aspect5=xyzs_id2_f2(3)/xyzs_id2_f2(4);
aspect6=xyzs_id3_f2(3)/xyzs_id3_f2(4);

% Calculate the distance between centroids
centroid_distance_1=((xyzs_id1_f1(1)-xyzs_id1_f2(1))^2+(xyzs_id1_f1(2)-xyzs
_id1_f2(2))^2)^(1/2);
centroid_distance_2=((xyzs_id1_f1(1)-xyzs_id2_f2(1))^2+(xyzs_id1_f1(2)-xyzs
_id2_f2(2))^2)^(1/2);
centroid_distance_3=((xyzs_id1_f1(1)-xyzs_id3_f2(1))^2+(xyzs_id1_f1(2)-xyzs
_id3_f2(2))^2)^(1/2);
centroid_distance_4=((xyzs_id2_f1(1)-xyzs_id1_f2(1))^2+(xyzs_id2_f1(2)-xyzs

```

```

_id1_f2(2))^2)^(1/2);
centroid_distance_5=((xyzs_id2_f1(1)-xyzs_id2_f2(1))^2+(xyzs_id2_f1(2)-xyzs
_id2_f2(2))^2)^(1/2);
centroid_distance_6=((xyzs_id2_f1(1)-xyzs_id3_f2(1))^2+(xyzs_id2_f1(2)-xyzs
_id3_f2(2))^2)^(1/2);
centroid_distance_7=((xyzs_id3_f1(1)-xyzs_id1_f2(1))^2+(xyzs_id3_f1(2)-xyzs
_id1_f2(2))^2)^(1/2);
centroid_distance_8=((xyzs_id3_f1(1)-xyzs_id2_f2(1))^2+(xyzs_id3_f1(2)-xyzs
_id2_f2(2))^2)^(1/2);
centroid_distance_9=((xyzs_id3_f1(1)-xyzs_id3_f2(1))^2+(xyzs_id3_f1(2)-xyzs
_id3_f2(2))^2)^(1/2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% VELOCITY BEFORE INTERACTION %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
displacement_A1=((xyzs_id1_f1(1)-xyzs_id1_fi(1))^2+(xyzs_id1_f1(2)-xyzs_id1
_fi(2))^2)^(1/2);
displacement_B1=((xyzs_id2_f1(1)-xyzs_id2_fi(1))^2+(xyzs_id2_f1(2)-xyzs_id2
_fi(2))^2)^(1/2);
displacement_C1=((xyzs_id3_f1(1)-xyzs_id3_fi(1))^2+(xyzs_id3_f1(2)-xyzs_id3
_fi(2))^2)^(1/2);
velocity_A1=displacement_A1/framegap;
velocity_B1=displacement_B1/framegap;
velocity_C1=displacement_C1/framegap;
avg_velocity=(velocity_A1+velocity_B1+velocity_C1)/3;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% VELOCITY AFTER INTERACTION %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
displacement_A2=((xyzs_id1_f2(1)-xyzs_id1_fi(1))^2+(xyzs_id1_f2(2)-xyzs_id1
_fi(2))^2)^(1/2);
displacement_B2=((xyzs_id2_f2(1)-xyzs_id2_fi(1))^2+(xyzs_id2_f2(2)-xyzs_id2
_fi(2))^2)^(1/2);
displacement_C2=((xyzs_id3_f2(1)-xyzs_id3_fi(1))^2+(xyzs_id3_f2(2)-xyzs_id3
_fi(2))^2)^(1/2);
velocity_A2=displacement_A2/framegap;
velocity_B2=displacement_B2/framegap;
velocity_C2=displacement_C2/framegap;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% DIRECTION BEFORE INTERACTION %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% fi-f1
rad_A1=atan2(xyzs_id1_fi(2)-xyzs_id1_f1(2),xyzs_id1_fi(1)-xyzs_id1_f1(1))+p
i; % atan2(Y,X) value in (-pi,pi), assign to (0,2pi)
rad_B1=atan2(xyzs_id2_fi(2)-xyzs_id2_f1(2),xyzs_id2_fi(1)-xyzs_id2_f1(1))+p
i;
rad_C1=atan2(xyzs_id3_fi(2)-xyzs_id3_f1(2),xyzs_id3_fi(1)-xyzs_id3_f1(1))+p
i;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% DIRECTION AFTER INTERACTION %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% f2-f1

```



```

rad_A2=atan2(xyzs_id1_f2(2)-xyzs_id1_f1(2),xyzs_id1_f2(1)-xyzs_id1_f1(1))+p
i;
rad_B2=atan2(xyzs_id2_f2(2)-xyzs_id2_f1(2),xyzs_id2_f2(1)-xyzs_id2_f1(1))+p
i;
rad_C2=atan2(xyzs_id3_f2(2)-xyzs_id3_f1(2),xyzs_id3_f2(1)-xyzs_id3_f1(1))+p
i;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Calculate the total cost based on weights and differences
% 1
cost_vector(1,1) =
w_int_intensity*(int_intensity1/avg_int_intensity1-int_intensity4/avg_int_i
ntensity2)^2;
cost_vector(2,1) =
w_norm_intensity*(norm_intensity1/avg_norm_intensity1-norm_intensity4/avg_n
orm_intensity2)^2;
cost_vector(3,1) = w_area*(area1/avg_area1 - area4/avg_area2)^2;
cost_vector(4,1) = w_aspect*(aspect1 - aspect4)^2;
cost_vector(5,1) = w_position*(centroid_distance_1/(avg_diameter1/2 +
avg_diameter2/2))^2;
cost_vector(6,1) = w_velocity*((velocity_A1-velocity_A2)/avg_velocity)^2;
if abs(rad_A1-rad_A2) > pi
    cost_vector(7,1) = w_direction*(abs(rad_A1-rad_A2)-pi)^2;
else
    cost_vector(7,1) = w_direction*(rad_A1-rad_A2)^2;
end
cost_vector(8,1) =
cost_vector(1,1)+cost_vector(2,1)+cost_vector(3,1)+cost_vector(4,1)+cost_ve
ctor(5,1)+cost_vector(6,1)+cost_vector(7,1);
% 2
cost_vector(1,2) =
w_int_intensity*(int_intensity1/avg_int_intensity1-int_intensity5/avg_int_i
ntensity2)^2;
cost_vector(2,2) =
w_norm_intensity*(norm_intensity1/avg_norm_intensity1-norm_intensity5/avg_n
orm_intensity2)^2;
cost_vector(3,2) = w_area*(area1/avg_area1 - area5/avg_area2)^2;
cost_vector(4,2) = w_aspect*(aspect1 - aspect5)^2;
cost_vector(5,2) = w_position*(centroid_distance_2/(avg_diameter1/2 +
avg_diameter2/2))^2;
cost_vector(6,2) = w_velocity*((velocity_A1-velocity_B2)/avg_velocity)^2;
if abs(rad_A1-rad_B2) > pi
    cost_vector(7,2) = w_direction*(abs(rad_A1-rad_B2)-pi)^2;
else

```

```

    cost_vector(7,2) = w_direction*(rad_A1-rad_B2)^2;
end
cost_vector(8,2) =
cost_vector(1,2)+cost_vector(2,2)+cost_vector(3,2)+cost_vector(4,2)+cost_
vector(5,2)+cost_vector(6,2)+cost_vector(7,2);
% 3
cost_vector(1,3) =
w_int_intensity*(int_intensity1/avg_int_intensity1-int_intensity6/avg_int_i
ntensity2)^2;
cost_vector(2,3) =
w_norm_intensity*(norm_intensity1/avg_norm_intensity1-norm_intensity6/avg_n
orm_intensity2)^2;
cost_vector(3,3) = w_area*(area1/avg_area1 - area6/avg_area2)^2;
cost_vector(4,3) = w_aspect*(aspect1 - aspect6)^2;
cost_vector(5,3) = w_position*(centroid_distance_3/(avg_diameter1/2 +
avg_diameter2/2))^2;
cost_vector(6,3) = w_velocity*((velocity_A1-velocity_C2)/avg_velocity)^2;
if abs(rad_A1-rad_C2) > pi
    cost_vector(7,3) = w_direction*(abs(rad_A1-rad_C2)-pi)^2;
else
    cost_vector(7,3) = w_direction*(rad_A1-rad_C2)^2;
end
cost_vector(8,3) =
cost_vector(1,3)+cost_vector(2,3)+cost_vector(3,3)+cost_vector(4,3)+cost_
vector(5,3)+cost_vector(6,3)+cost_vector(7,3);
% 4
cost_vector(1,4) =
w_int_intensity*(int_intensity2/avg_int_intensity1-int_intensity4/avg_int_i
ntensity2)^2;
cost_vector(2,4) =
w_norm_intensity*(norm_intensity2/avg_norm_intensity1-norm_intensity4/avg_n
orm_intensity2)^2;
cost_vector(3,4) = w_area*(area2/avg_area1 - area4/avg_area2)^2;
cost_vector(4,4) = w_aspect*(aspect2 - aspect4)^2;
cost_vector(5,4) = w_position*(centroid_distance_4/(avg_diameter1/2 +
avg_diameter2/2))^2;
cost_vector(6,4) = w_velocity*((velocity_B1-velocity_A2)/avg_velocity)^2;
if abs(rad_B1-rad_A2) > pi
    cost_vector(7,4) = w_direction*(abs(rad_B1-rad_A2)-pi)^2;
else
    cost_vector(7,4) = w_direction*(rad_B1-rad_A2)^2;
end
cost_vector(8,4) =
cost_vector(1,4)+cost_vector(2,4)+cost_vector(3,4)+cost_vector(4,4)+cost_

```

```

ctor(5,4)+cost_vector(6,4)+cost_vector(7,4);
% 5
cost_vector(1,5) =
w_int_intensity*(int_intensity2/avg_int_intensity1-int_intensity5/avg_int_i
ntensity2)^2;
cost_vector(2,5) =
w_norm_intensity*(norm_intensity2/avg_norm_intensity1-norm_intensity5/avg_n
orm_intensity2)^2;
cost_vector(3,5) = w_area*(area2/avg_area1 - area5/avg_area2)^2;
cost_vector(4,5) = w_aspect*(aspect2 - aspect5)^2;
cost_vector(5,5) = w_position*(centroid_distance_5/(avg_diameter1/2 +
avg_diameter2/2))^2;
cost_vector(6,5) = w_velocity*((velocity_B1-velocity_B2)/avg_velocity)^2;
if abs(rad_B1-rad_B2) > pi
    cost_vector(7,5) = w_direction*(abs(rad_B1-rad_B2)-pi)^2;
else
    cost_vector(7,5) = w_direction*(rad_B1-rad_B2)^2;
end
cost_vector(8,5) =
cost_vector(1,5)+cost_vector(2,5)+cost_vector(3,5)+cost_vector(4,5)+cost_ve
ctor(5,5)+cost_vector(6,5)+cost_vector(7,5);
% 6
cost_vector(1,6) =
w_int_intensity*(int_intensity2/avg_int_intensity1-int_intensity6/avg_int_i
ntensity2)^2;
cost_vector(2,6) =
w_norm_intensity*(norm_intensity2/avg_norm_intensity1-norm_intensity6/avg_n
orm_intensity2)^2;
cost_vector(3,6) = w_area*(area2/avg_area1 - area6/avg_area2)^2;
cost_vector(4,6) = w_aspect*(aspect2 - aspect6)^2;
cost_vector(5,6) = w_position*(centroid_distance_6/(avg_diameter1/2 +
avg_diameter2/2))^2;
cost_vector(6,6) = w_velocity*((velocity_B1-velocity_C2)/avg_velocity)^2;
if abs(rad_B1-rad_C2) > pi
    cost_vector(7,6) = w_direction*(abs(rad_B1-rad_C2)-pi)^2;
else
    cost_vector(7,6) = w_direction*(rad_B1-rad_C2)^2;
end
cost_vector(8,6) =
cost_vector(1,6)+cost_vector(2,6)+cost_vector(3,6)+cost_vector(4,6)+cost_ve
ctor(5,6)+cost_vector(6,6)+cost_vector(7,6);
% 7
cost_vector(1,7) =
w_int_intensity*(int_intensity3/avg_int_intensity1-int_intensity4/avg_int_i

```

```

ntensity2)^2;
cost_vector(2,7) =
w_norm_intensity*(norm_intensity3/avg_norm_intensity1-norm_intensity4/avg_n
orm_intensity2)^2;
cost_vector(3,7) = w_area*(area3/avg_area1 - area4/avg_area2)^2;
cost_vector(4,7) = w_aspect*(aspect3 - aspect4)^2;
cost_vector(5,7) = w_position*(centroid_distance_7/(avg_diameter1/2 +
avg_diameter2/2))^2;
cost_vector(6,7) = w_velocity*((velocity_C1-velocity_A2)/avg_velocity)^2;
if abs(rad_C1-rad_A2) > pi
    cost_vector(7,7) = w_direction*(abs(rad_C1-rad_A2)-pi)^2;
else
    cost_vector(7,7) = w_direction*(rad_C1-rad_A2)^2;
end
cost_vector(8,7) =
cost_vector(1,7)+cost_vector(2,7)+cost_vector(3,7)+cost_vector(4,7)+cost_ve
ctor(5,7)+cost_vector(6,7)+cost_vector(7,7);
% 8
cost_vector(1,8) =
w_int_intensity*(int_intensity3/avg_int_intensity1-int_intensity5/avg_int_i
ntensity2)^2;
cost_vector(2,8) =
w_norm_intensity*(norm_intensity3/avg_norm_intensity1-norm_intensity5/avg_n
orm_intensity2)^2;
cost_vector(3,8) = w_area*(area3/avg_area1 - area5/avg_area2)^2;
cost_vector(4,8) = w_aspect*(aspect3 - aspect5)^2;
cost_vector(5,8) = w_position*(centroid_distance_8/(avg_diameter1/2 +
avg_diameter2/2))^2;
cost_vector(6,8) = w_velocity*((velocity_C1-velocity_B2)/avg_velocity)^2;
if abs(rad_C1-rad_B2) > pi
    cost_vector(7,8) = w_direction*(abs(rad_C1-rad_B2)-pi)^2;
else
    cost_vector(7,8) = w_direction*(rad_C1-rad_B2)^2;
end
cost_vector(8,8) =
cost_vector(1,8)+cost_vector(2,8)+cost_vector(3,8)+cost_vector(4,8)+cost_ve
ctor(5,8)+cost_vector(6,8)+cost_vector(7,8);
% 9
cost_vector(1,9) =
w_int_intensity*(int_intensity3/avg_int_intensity1-int_intensity6/avg_int_i
ntensity2)^2;
cost_vector(2,9) =
w_norm_intensity*(norm_intensity3/avg_norm_intensity1-norm_intensity6/avg_n
orm_intensity2)^2;

```

```

cost_vector(3,9) = w_area*(area3/avg_area1 - area6/avg_area2)^2;
cost_vector(4,9) = w_aspect*(aspect3 - aspect6)^2;
cost_vector(5,9) = w_position*(centroid_distance_9/(avg_diameter1/2 +
avg_diameter2/2))^2;
cost_vector(6,9) = w_velocity*((velocity_C1-velocity_C2)/avg_velocity)^2;
if abs(rad_C1-rad_C2) > pi
    cost_vector(7,9) = w_direction*(abs(rad_C1-rad_C2)-pi)^2;
else
    cost_vector(7,9) = w_direction*(rad_C1-rad_C2)^2;
end
cost_vector(8,9) =
cost_vector(1,9)+cost_vector(2,9)+cost_vector(3,9)+cost_vector(4,9)+cost_ve
ctor(5,9)+cost_vector(6,9)+cost_vector(7,9);

save cost cost_vector framegap

category(cost_vector, framegap);

end

```

6. Cost Function for More Than 1 Interacting Frame ^[12]

```

function [cost_vector, framegap]=cost_2(w_int_intensity, w_norm_intensity,
w_area, w_aspect, w_position, w_velocity, w_direction, xyzs_info, frame_avg,
xyzs_id_columns)
% more than 1 interacting frame

cost_vector=zeros(8,9);
xyzs_id1_f1=xyzs_info(1,:);
xyzs_id1_fi1=xyzs_info(2,:);
xyzs_id1_fi2=xyzs_info(3,:);
xyzs_id1_f2=xyzs_info(4,:);
xyzs_id2_f1=xyzs_info(5,:);
xyzs_id2_fi1=xyzs_info(6,:);
xyzs_id2_fi2=xyzs_info(7,:);
xyzs_id2_f2=xyzs_info(8,:);
xyzs_id3_f1=xyzs_info(9,:);
xyzs_id3_fi1=xyzs_info(10,:);
xyzs_id3_fi2=xyzs_info(11,:);
xyzs_id3_f2=xyzs_info(12,:);

framegap=xyzs_id1_fi1(xyzs_id_columns-1)-xyzs_id1_f1(xyzs_id_columns-1);

```

```

id1_frame1=xyzs_info(1,xyzs_id_columns-1);
% id2_frame1=xyzs_id2_f1(xyzs_id_columns-1);
% id3_frame1=xyzs_id3_f1(xyzs_id_columns-1);
id1_frame2=xyzs_info(9,xyzs_id_columns-1);
% id2_frame2=xyzs_id2_f2(xyzs_id_columns-1);
% id3_frame2=xyzs_id3_f2(xyzs_id_columns-1);

% Extract frame averages from frame_avg
avg_norm_intensity1 = frame_avg(7,id1_frame1); %average intensity/area
avg_norm_intensity2 = frame_avg(7,id1_frame2);
avg_area1 = frame_avg(5,id1_frame1); %average area
avg_area2 = frame_avg(5,id1_frame2);
avg_int_intensity1 = frame_avg(6,id1_frame1); %average integrated intensity
avg_int_intensity2 = frame_avg(6,id1_frame2);
avg_diameter1 = (2*frame_avg(1,id1_frame1) + 2*frame_avg(2,id1_frame1))/2; %
average of major and minor diameter
avg_diameter2 = (2*frame_avg(1,id1_frame2) + 2*frame_avg(2,id1_frame2))/2;

% Extracts the intensity
int_intensity1=xyzs_id1_f1(7);
int_intensity2=xyzs_id2_f1(7);
int_intensity3=xyzs_id3_f1(7);
int_intensity4=xyzs_id1_f2(7);
int_intensity5=xyzs_id2_f2(7);
int_intensity6=xyzs_id3_f2(7);

% Extracts the areas
area1=xyzs_id1_f1(6);
area2=xyzs_id2_f1(6);
area3=xyzs_id3_f1(6);
area4=xyzs_id1_f2(6);
area5=xyzs_id2_f2(6);
area6=xyzs_id3_f2(6);

% Calculate the average intensity
norm_intensity1=int_intensity1/area1;
norm_intensity2=int_intensity2/area2;
norm_intensity3=int_intensity3/area3;
norm_intensity4=int_intensity4/area4;
norm_intensity5=int_intensity5/area5;
norm_intensity6=int_intensity6/area6;

% Calculate the aspect ratio
aspect1=xyzs_id1_f1(3)/xyzs_id1_f1(4);

```

```

aspect2=xyzs_id2_f1(3)/xyzs_id2_f1(4);
aspect3=xyzs_id3_f1(3)/xyzs_id3_f1(4);
aspect4=xyzs_id1_f2(3)/xyzs_id1_f2(4);
aspect5=xyzs_id2_f2(3)/xyzs_id2_f2(4);
aspect6=xyzs_id3_f2(3)/xyzs_id3_f2(4);

% Calculate the distance between centroids
centroid_distance_1=((xyzs_id1_f1(1)-xyzs_id1_f2(1))^2+(xyzs_id1_f1(2)-xyzs_id1_f2(2))^2)^(1/2);
centroid_distance_2=((xyzs_id1_f1(1)-xyzs_id2_f2(1))^2+(xyzs_id1_f1(2)-xyzs_id2_f2(2))^2)^(1/2);
centroid_distance_3=((xyzs_id1_f1(1)-xyzs_id3_f2(1))^2+(xyzs_id1_f1(2)-xyzs_id3_f2(2))^2)^(1/2);
centroid_distance_4=((xyzs_id2_f1(1)-xyzs_id1_f2(1))^2+(xyzs_id2_f1(2)-xyzs_id1_f2(2))^2)^(1/2);
centroid_distance_5=((xyzs_id2_f1(1)-xyzs_id2_f2(1))^2+(xyzs_id2_f1(2)-xyzs_id2_f2(2))^2)^(1/2);
centroid_distance_6=((xyzs_id2_f1(1)-xyzs_id3_f2(1))^2+(xyzs_id2_f1(2)-xyzs_id3_f2(2))^2)^(1/2);
centroid_distance_7=((xyzs_id3_f1(1)-xyzs_id1_f2(1))^2+(xyzs_id3_f1(2)-xyzs_id1_f2(2))^2)^(1/2);
centroid_distance_8=((xyzs_id3_f1(1)-xyzs_id2_f2(1))^2+(xyzs_id3_f1(2)-xyzs_id2_f2(2))^2)^(1/2);
centroid_distance_9=((xyzs_id3_f1(1)-xyzs_id3_f2(1))^2+(xyzs_id3_f1(2)-xyzs_id3_f2(2))^2)^(1/2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% VELOCITY BEFORE INTERACTION %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
displacement_A1=((xyzs_id1_f1(1)-xyzs_id1_fi1(1))^2+(xyzs_id1_f1(2)-xyzs_id1_fi1(2))^2)^(1/2);
displacement_B1=((xyzs_id2_f1(1)-xyzs_id2_fi1(1))^2+(xyzs_id2_f1(2)-xyzs_id2_fi1(2))^2)^(1/2);
displacement_C1=((xyzs_id3_f1(1)-xyzs_id3_fi1(1))^2+(xyzs_id3_f1(2)-xyzs_id3_fi1(2))^2)^(1/2);
velocity_A1=displacement_A1/framegap;
velocity_B1=displacement_B1/framegap;
velocity_C1=displacement_C1/framegap;
avg_velocity=(velocity_A1+velocity_B1+velocity_C1)/3;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% VELOCITY AFTER INTERACTION %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
displacement_A2=((xyzs_id1_f2(1)-xyzs_id1_fi2(1))^2+(xyzs_id1_f2(2)-xyzs_id1_fi2(2))^2)^(1/2);
displacement_B2=((xyzs_id2_f2(1)-xyzs_id2_fi2(1))^2+(xyzs_id2_f2(2)-xyzs_id2_fi2(2))^2)^(1/2);
displacement_C2=((xyzs_id3_f2(1)-xyzs_id3_fi2(1))^2+(xyzs_id3_f2(2)-xyzs_id

```

```

3_fi2(2))^2)^(1/2);
velocity_A2=displacement_A2/framegap;
velocity_B2=displacement_B2/framegap;
velocity_C2=displacement_C2/framegap;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% DIRECTION BEFORE INTERACTION %%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% f1-f1
rad_A1=atan2(xyzs_id1_fil(2)-xyzs_id1_f1(2),xyzs_id1_fil(1)-xyzs_id1_f1(1))
+pi; % atan2(Y,X) value in (-pi,pi), assign to (0,2pi)
rad_B1=atan2(xyzs_id2_fil(2)-xyzs_id2_f1(2),xyzs_id2_fil(1)-xyzs_id2_f1(1))
+pi;
rad_C1=atan2(xyzs_id3_fil(2)-xyzs_id3_f1(2),xyzs_id3_fil(1)-xyzs_id3_f1(1))
+pi;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% DIRECTION AFTER INTERACTION %%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% f2-f1
rad_A2=atan2(xyzs_id1_f2(2)-xyzs_id1_f1(2),xyzs_id1_f2(1)-xyzs_id1_f1(1))+p
i;
rad_B2=atan2(xyzs_id2_f2(2)-xyzs_id2_f1(2),xyzs_id2_f2(1)-xyzs_id2_f1(1))+p
i;
rad_C2=atan2(xyzs_id3_f2(2)-xyzs_id3_f1(2),xyzs_id3_f2(1)-xyzs_id3_f1(1))+p
i;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Calculate the total cost based on weights and differences
% 1
cost_vector(1,1) =
w_int_intensity*(int_intensity1/avg_int_intensity1-int_intensity4/avg_int_i
ntensity2)^2;
cost_vector(2,1) =
w_norm_intensity*(norm_intensity1/avg_norm_intensity1-norm_intensity4/avg_n
orm_intensity2)^2;
cost_vector(3,1) = w_area*(area1/avg_area1 - area4/avg_area2)^2;
cost_vector(4,1) = w_aspect*(aspect1 - aspect4)^2;
cost_vector(5,1) = w_position*(centroid_distance_1/(avg_diameter1/2 +
avg_diameter2/2))^2;
cost_vector(6,1) = w_velocity*((velocity_A1-velocity_A2)/avg_velocity)^2;
if abs(rad_A1-rad_A2) > pi
    cost_vector(7,1) = w_direction*(abs(rad_A1-rad_A2)-pi)^2;
else
    cost_vector(7,1) = w_direction*(rad_A1-rad_A2)^2;
end
cost_vector(8,1) =
cost_vector(1,1)+cost_vector(2,1)+cost_vector(3,1)+cost_vector(4,1)+cost_ve
ctor(5,1)+cost_vector(6,1)+cost_vector(7,1);
% 2

```



```

cost_vector(1,2) =
w_int_intensity*(int_intensity1/avg_int_intensity1-int_intensity5/avg_int_i
ntensity2)^2;
cost_vector(2,2) =
w_norm_intensity*(norm_intensity1/avg_norm_intensity1-norm_intensity5/avg_n
orm_intensity2)^2;
cost_vector(3,2) = w_area*(area1/avg_area1 - area5/avg_area2)^2;
cost_vector(4,2) = w_aspect*(aspect1 - aspect5)^2;
cost_vector(5,2) = w_position*(centroid_distance_2/(avg_diameter1/2 +
avg_diameter2/2))^2;
cost_vector(6,2) = w_velocity*((velocity_A1-velocity_B2)/avg_velocity)^2;
if abs(rad_A1-rad_B2) > pi
    cost_vector(7,2) = w_direction*(abs(rad_A1-rad_B2)-pi)^2;
else
    cost_vector(7,2) = w_direction*(rad_A1-rad_B2)^2;
end
cost_vector(8,2) =
cost_vector(1,2)+cost_vector(2,2)+cost_vector(3,2)+cost_vector(4,2)+cost_ve
ctor(5,2)+cost_vector(6,2)+cost_vector(7,2);
% 3
cost_vector(1,3) =
w_int_intensity*(int_intensity1/avg_int_intensity1-int_intensity6/avg_int_i
ntensity2)^2;
cost_vector(2,3) =
w_norm_intensity*(norm_intensity1/avg_norm_intensity1-norm_intensity6/avg_n
orm_intensity2)^2;
cost_vector(3,3) = w_area*(area1/avg_area1 - area6/avg_area2)^2;
cost_vector(4,3) = w_aspect*(aspect1 - aspect6)^2;
cost_vector(5,3) = w_position*(centroid_distance_3/(avg_diameter1/2 +
avg_diameter2/2))^2;
cost_vector(6,3) = w_velocity*((velocity_A1-velocity_C2)/avg_velocity)^2;
if abs(rad_A1-rad_C2) > pi
    cost_vector(7,3) = w_direction*(abs(rad_A1-rad_C2)-pi)^2;
else
    cost_vector(7,3) = w_direction*(rad_A1-rad_C2)^2;
end
cost_vector(8,3) =
cost_vector(1,3)+cost_vector(2,3)+cost_vector(3,3)+cost_vector(4,3)+cost_ve
ctor(5,3)+cost_vector(6,3)+cost_vector(7,3);
% 4
cost_vector(1,4) =
w_int_intensity*(int_intensity2/avg_int_intensity1-int_intensity4/avg_int_i
ntensity2)^2;
cost_vector(2,4) =

```

```

w_norm_intensity*(norm_intensity2/avg_norm_intensity1-norm_intensity4/avg_norm_intensity2)^2;
cost_vector(3,4) = w_area*(area2/avg_area1 - area4/avg_area2)^2;
cost_vector(4,4) = w_aspect*(aspect2 - aspect4)^2;
cost_vector(5,4) = w_position*(centroid_distance_4/(avg_diameter1/2 + avg_diameter2/2))^2;
cost_vector(6,4) = w_velocity*((velocity_B1-velocity_A2)/avg_velocity)^2;
if abs(rad_B1-rad_A2) > pi
    cost_vector(7,4) = w_direction*(abs(rad_B1-rad_A2)-pi)^2;
else
    cost_vector(7,4) = w_direction*(rad_B1-rad_A2)^2;
end
cost_vector(8,4) =
cost_vector(1,4)+cost_vector(2,4)+cost_vector(3,4)+cost_vector(4,4)+cost_vector(5,4)+cost_vector(6,4)+cost_vector(7,4);
% 5
cost_vector(1,5) =
w_int_intensity*(int_intensity2/avg_int_intensity1-int_intensity5/avg_int_intensity2)^2;
cost_vector(2,5) =
w_norm_intensity*(norm_intensity2/avg_norm_intensity1-norm_intensity5/avg_norm_intensity2)^2;
cost_vector(3,5) = w_area*(area2/avg_area1 - area5/avg_area2)^2;
cost_vector(4,5) = w_aspect*(aspect2 - aspect5)^2;
cost_vector(5,5) = w_position*(centroid_distance_5/(avg_diameter1/2 + avg_diameter2/2))^2;
cost_vector(6,5) = w_velocity*((velocity_B1-velocity_B2)/avg_velocity)^2;
if abs(rad_B1-rad_B2) > pi
    cost_vector(7,5) = w_direction*(abs(rad_B1-rad_B2)-pi)^2;
else
    cost_vector(7,5) = w_direction*(rad_B1-rad_B2)^2;
end
cost_vector(8,5) =
cost_vector(1,5)+cost_vector(2,5)+cost_vector(3,5)+cost_vector(4,5)+cost_vector(5,5)+cost_vector(6,5)+cost_vector(7,5);
% 6
cost_vector(1,6) =
w_int_intensity*(int_intensity2/avg_int_intensity1-int_intensity6/avg_int_intensity2)^2;
cost_vector(2,6) =
w_norm_intensity*(norm_intensity2/avg_norm_intensity1-norm_intensity6/avg_norm_intensity2)^2;
cost_vector(3,6) = w_area*(area2/avg_area1 - area6/avg_area2)^2;
cost_vector(4,6) = w_aspect*(aspect2 - aspect6)^2;

```

```

cost_vector(5,6) = w_position*(centroid_distance_6/(avg_diameter1/2 +
avg_diameter2/2))^2;
cost_vector(6,6) = w_velocity*((velocity_B1-velocity_C2)/avg_velocity)^2;
if abs(rad_B1-rad_C2) > pi
    cost_vector(7,6) = w_direction*(abs(rad_B1-rad_C2)-pi)^2;
else
    cost_vector(7,6) = w_direction*(rad_B1-rad_C2)^2;
end
cost_vector(8,6) =
cost_vector(1,6)+cost_vector(2,6)+cost_vector(3,6)+cost_vector(4,6)+cost_ve
ctor(5,6)+cost_vector(6,6)+cost_vector(7,6);
% 7
cost_vector(1,7) =
w_int_intensity*(int_intensity3/avg_int_intensity1-int_intensity4/avg_int_i
ntensity2)^2;
cost_vector(2,7) =
w_norm_intensity*(norm_intensity3/avg_norm_intensity1-norm_intensity4/avg_n
orm_intensity2)^2;
cost_vector(3,7) = w_area*(area3/avg_area1 - area4/avg_area2)^2;
cost_vector(4,7) = w_aspect*(aspect3 - aspect4)^2;
cost_vector(5,7) = w_position*(centroid_distance_7/(avg_diameter1/2 +
avg_diameter2/2))^2;
cost_vector(6,7) = w_velocity*((velocity_C1-velocity_A2)/avg_velocity)^2;
if abs(rad_C1-rad_A2) > pi
    cost_vector(7,7) = w_direction*(abs(rad_C1-rad_A2)-pi)^2;
else
    cost_vector(7,7) = w_direction*(rad_C1-rad_A2)^2;
end
cost_vector(8,7) =
cost_vector(1,7)+cost_vector(2,7)+cost_vector(3,7)+cost_vector(4,7)+cost_ve
ctor(5,7)+cost_vector(6,7)+cost_vector(7,7);
% 8
cost_vector(1,8) =
w_int_intensity*(int_intensity3/avg_int_intensity1-int_intensity5/avg_int_i
ntensity2)^2;
cost_vector(2,8) =
w_norm_intensity*(norm_intensity3/avg_norm_intensity1-norm_intensity5/avg_n
orm_intensity2)^2;
cost_vector(3,8) = w_area*(area3/avg_area1 - area5/avg_area2)^2;
cost_vector(4,8) = w_aspect*(aspect3 - aspect5)^2;
cost_vector(5,8) = w_position*(centroid_distance_8/(avg_diameter1/2 +
avg_diameter2/2))^2;
cost_vector(6,8) = w_velocity*((velocity_C1-velocity_B2)/avg_velocity)^2;
if abs(rad_C1-rad_B2) > pi

```

```

    cost_vector(7,8) = w_direction*(abs(rad_C1-rad_B2)-pi)^2;
else
    cost_vector(7,8) = w_direction*(rad_C1-rad_B2)^2;
end
cost_vector(8,8) =
cost_vector(1,8)+cost_vector(2,8)+cost_vector(3,8)+cost_vector(4,8)+cost_ve
ctor(5,8)+cost_vector(6,8)+cost_vector(7,8);
% 9
cost_vector(1,9) =
w_int_intensity*(int_intensity3/avg_int_intensity1-int_intensity6/avg_int_i
ntensity2)^2;
cost_vector(2,9) =
w_norm_intensity*(norm_intensity3/avg_norm_intensity1-norm_intensity6/avg_n
orm_intensity2)^2;
cost_vector(3,9) = w_area*(area3/avg_area1 - area6/avg_area2)^2;
cost_vector(4,9) = w_aspect*(aspect3 - aspect6)^2;
cost_vector(5,9) = w_position*(centroid_distance_9/(avg_diameter1/2 +
avg_diameter2/2))^2;
cost_vector(6,9) = w_velocity*((velocity_C1-velocity_C2)/avg_velocity)^2;
if abs(rad_C1-rad_C2) > pi
    cost_vector(7,9) = w_direction*(abs(rad_C1-rad_C2)-pi)^2;
else
    cost_vector(7,9) = w_direction*(rad_C1-rad_C2)^2;
end
cost_vector(8,9) =
cost_vector(1,9)+cost_vector(2,9)+cost_vector(3,9)+cost_vector(4,9)+cost_ve
ctor(5,9)+cost_vector(6,9)+cost_vector(7,9);

save cost cost_vector framegap

category(cost_vector, framegap);

end

```

7. Category Function: Determine Which Method Shall Be Used

```

function []=category(cost_vector, framegap)

A_A=min(min(cost_vector(7,1), cost_vector(7,2)), cost_vector(7,3)); % minimum
AA
B_B=min(min(cost_vector(7,4), cost_vector(7,5)), cost_vector(7,6)); % minimum
BB
C_C=min(min(cost_vector(7,7), cost_vector(7,8)), cost_vector(7,9)); % minimum
CC

```

```

if framegap <= 3
    disp('Keep Kilfoil Method');

else framegap > 3
    disp('Use Directionality to Determine Cost Results');
    if
(A_A==cost_vector(7,1))&&(B_B==cost_vector(7,5))&&(C_C==cost_vector(7,9))
        disp('Cell ID correct');
    else disp('Cell ID incorrect, need to be switched');
    end
end
end
end

```

8. Script: Use Directionality to Correct Cell Tracking

```

% directionality to tag cells
AA=min(min(cost_vector(7,1), cost_vector(7,2)), cost_vector(7,3)); % minimum
AA
BB=min(min(cost_vector(7,4), cost_vector(7,5)), cost_vector(7,6)); % minimum
BB
CC=min(min(cost_vector(7,7), cost_vector(7,8)), cost_vector(7,9)); % minimum
CC

XX=min(min(AA,BB),CC); % min value among AA,BB,CC, then select the other two
to switch

I = imread('7Percent_MedDensity_Sample1.tif', 91, 'Info', image_info);
figure;
imagesc(I)
colormap('gray')
axis([350 550 0 200])
hold on

if XX==AA;
    plot(xyzs_id1_f2(1,1)',xyzs_id1_f2(1,2)', 'rx') %% remain same
    plot(xyzs_id2_f2(1,1)',xyzs_id2_f2(1,2)', 'gx')
    plot(xyzs_id3_f2(1,1)',xyzs_id3_f2(1,2)', 'yx')
    hold on
end

if XX==BB;
    plot(xyzs_id1_f2(1,1)',xyzs_id1_f2(1,2)', 'gx')

```

```
plot(xyzs_id2_f2(1,1)',xyzs_id2_f2(1,2)', 'yx') %% remain same
plot(xyzs_id3_f2(1,1)',xyzs_id3_f2(1,2)', 'rx')
hold on
end

if XX==CC;
plot(xyzs_id1_f2(1,1)',xyzs_id1_f2(1,2)', 'yx')
plot(xyzs_id2_f2(1,1)',xyzs_id2_f2(1,2)', 'rx')
plot(xyzs_id3_f2(1,1)',xyzs_id3_f2(1,2)', 'gx') %% remain same
hold on
end
```

Reference

- [1] Rogers, Walter J., Craig H. Meyer, and Christopher M. Kramer. "Technology insight: in vivo cell tracking by use of MRI." *Nature Clinical Practice Cardiovascular Medicine* 3.10 (2006): 554-562.
- [2] Meijering, Erik, Oleh Dzyubachyk, and Ihor Smal. "Methods for cell and particle tracking." *Methods Enzymol* 504.9 (2012): 183-200.
- [3] Rapoport, Daniel H., et al. "A novel validation algorithm allows for automated cell tracking and the extraction of biologically meaningful parameters." *PloS one* 6.11 (2011): e27315.
- [4] Rabut, Gwénaél, and Jan Ellenberg. "Automatic real - time three-dimensional cell tracking by fluorescence microscopy." *Journal of microscopy* 216.2 (2004): 131-137.
- [5] Shen, Hailin, et al. "Automatic tracking of biological cells and compartments using particle filters and active contours." *Chemometrics and Intelligent Laboratory Systems* 82.1 (2006): 276-282.
- [6] Cohen, Andrew R., et al. "Computational prediction of neural progenitor cell fates." *Nature methods* 7.3 (2010): 213-218.
- [7] Bise, Ryoma, et al. "Automatic cell tracking applied to analysis of cell migration in wound healing assay." *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*. IEEE, 2011.
- [8] Kragl, Martin, et al. "Cells keep a memory of their tissue origin during axolotl limb regeneration." *Nature* 460.7251 (2009): 60-65.
- [9] Sulston, John E., and H. Robert Horvitz. "Post-embryonic cell lineages of the nematode, *Caenorhabditis elegans*." *Developmental biology* 56.1 (1977): 110-156.
- [10] Anoraganingrum, Dwi. "Cell segmentation with median filter and mathematical morphology operation." *Image Analysis and Processing, 1999. Proceedings. International Conference on*. IEEE, 1999.

- [11] House, Dwight, et al. "Tracking of cell populations to understand their spatio-temporal behavior in response to physical stimuli." *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on*. IEEE, 2009.
- [12] Baker, Richard M., et al. "Automated, contour-based tracking and analysis of cell behaviour over long time scales in environments of varying complexity and cell density." *Journal of The Royal Society Interface* 11.97 (2014): 20140386.
- [13] Padfield, Dirk, Jens Rittscher, and Badrinath Roysam. "Quantitative biological studies enabled by robust cell tracking." *Biomedical Imaging: From Nano to Macro, 2011 IEEE International Symposium on*. IEEE, 2011.
- [14] Chen, Xiaowei, Xiaobo Zhou, and Stephen TC Wong. "Automated segmentation, classification, and tracking of cancer cell nuclei in time-lapse microscopy." *Biomedical Engineering, IEEE Transactions on* 53.4 (2006): 762-766.
- [15] Eilken, Hanna M., Shin-Ichi Nishikawa, and Timm Schroeder. "Continuous single-cell imaging of blood generation from haemogenic endothelium." *Nature* 457.7231 (2009): 896-900.
- [16] Yim, Evelyn KF, et al. "Nanopattern-induced changes in morphology and motility of smooth muscle cells." *Biomaterials* 26.26 (2005): 5405-5413.
- [17] Pinho, Diana, et al. "Automatic tracking of labeled red blood cells in microchannels." *International journal for numerical methods in biomedical engineering* 29.9 (2013): 977-987.
- [18] Leung, Nat, and Justin WL Wan. "Reconstruction of missing cells in fluorescent microscopy." *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*. IEEE, 2012.
- [19] Degerman, J., et al. "A comparative study between level set and watershed image segmentation for

tracking stem cells in time-lapse microscopy." *Proc. Microscopic Image Analysis with Applications in Biology (MIAAB 06)* (2006).

[20] Padfield, Dirk, Jens Rittscher, and Badrinath Roysam. "Spatio-temporal cell segmentation and tracking for automated screening." *Biomedical Imaging: From Nano to Macro, 2008. ISBI 2008. 5th IEEE International Symposium on*. IEEE, 2008.

[21] Yang, Fuxing, et al. "Cell segmentation, tracking, and mitosis detection using temporal context." *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2005*. Springer Berlin Heidelberg, 2005. 302-309.

[22] Pectreaux, Jacques, Christopher Zimmer, and Jean-Christophe Olivo-Marin. "Biophysical active contours for cell tracking I: Tension and bending." *Image Processing, 2006 IEEE International Conference on*. IEEE, 2006.

[23] Hand, A. J., et al. "Automated tracking of migrating cells in phase - contrast video microscopy sequences using image registration." *Journal of microscopy* 234.1 (2009): 62-79.

[24] Ray, Nilanjan, and Scott T. Acton. "Active contours for cell tracking." *Image Analysis and Interpretation, 2002. Proceedings. Fifth IEEE Southwest Symposium on*. IEEE, 2002.

[25] ACTIVE electronic supplementary material: <http://rsif.royalsocietypublishing.org/>

[26] Georgiou, S. D., et al. "Construction of orthogonal and nearly orthogonal designs for computer experiments." *Biometrika* (2014): asu021.

[27] Georgiou, S. D. "Orthogonal designs for computer experiments." *Journal of Statistical Planning and Inference* 141.4 (2011): 1519-1525.

[28] Wu, Xuan, and Dennis YC Leung. "Optimization of biodiesel production from camelina oil using orthogonal experiment." *Applied Energy* 88.11 (2011): 3615-3624.

- [29] Wu, Jingming, and Hian Kee Lee. "Orthogonal array designs for the optimization of liquid–liquid–liquid microextraction of nonsteroidal anti-inflammatory drugs combined with high-performance liquid chromatography-ultraviolet detection." *Journal of Chromatography A* 1092.2 (2005): 182-190.
- [30] Zurovac, Jelena, and Randy Brown. "Orthogonal design: A powerful method for comparative effectiveness research with multiple interventions." *Center on Healthcare Effectiveness, Mathematica Policy Research* (2012).
- [31] Alsaran, Akgün, Ayhan Çelik, and Cafer Çelik. "Determination of the optimum conditions for ion nitriding of AISI 5140 steel." *Surface and Coatings Technology* 160.2 (2002): 219-226.
- [32] <http://www.mathworks.com/help/matlab/ref/atan2.html>

Biographic Data

Author Name: Ye He

EDUCATION

M.S., Bioengineering, August 2015 *Syracuse University, College of Engineering and Computer Science, Syracuse, NY*

B.S., Pharmacy, May 2013 *Nankai University, College of Pharmacy, Tianjin, China*

PROFESSIONAL RESEARCH

Thesis: Comparison of Cost Function Against Positional Analysis for Automated Tracking of Three-cell Interactions, *Syracuse University, NY (June 2014 to August 2015)*

- Be familiar with animal cell culture, passage and staining, imaged cells by microscope to automatically tracked cells migration over long time
- Applied mathematical algorithm and optimized MATLAB code to improve cell tracking accuracy

FRET Based OGT Activity Assay, *Nankai University, China (November 2013 to May 2014)*

- Cultured E.Coli strain, over-expressed and purified the O-GlcNAc transferase (OGT)
- Catalyzed glycosyl transferring reaction from UDP-GlcNAz to acceptor peptide
- Detected enzyme activity of OGT by HPLC, screened optimal glycerol concentration for enzyme conservation
- Conducted the click ligation reaction, tested various influential factors that affected the enzyme activity in the system

The Cloning Expression and Application of Arginine Deiminase, *Nankai University, China (April 2011 to March 2012)*

- Designed primers of AD genetic sequences, cloned genes and constructed recombinant plasmids

- Transformed and constructed recombinant strain, induced expression
- Explored optimal density and inducer concentration of expression, optimized transformation conditions of the system
- Detected a maximum conversion ratio of Arginine Deiminase and prepared Citrulline

ACADEMIC PROJECTS

Regression Analysis - Case Studies, August 2014 to December 2014

- Developed various regression models by using SAS to analyze data from biology research etc., improved model fitting and performance to predict tendency, summarize relative statistical results

Preparation of Various Drugs in Different Dosage Forms, March 2012 to December 2012

- Designed corresponding ingredients and optimized excipients, prepared normal injection, emulsifier, tablet, suppository, etc., morphological indicators were tested qualified

Pharmaceutical Analysis, March 2012 to June 2012

- Comprehensive Applied chemical principles and reactions, tested impurities to guarantee drug quality, according to the Pharmacopoeia regulations

Pharmacological Experiments, September 2011 to January 2012

- Animal anesthesia (mice and rabbit), intravenous and intramuscular injection methods, etc., studied various drug effect models and pharmacology