

# Hierarchical Partial Matching and Segmentation of Interacting Cells

Zheng Wu<sup>1</sup>, Danna Gurari<sup>1</sup>, Joyce Y. Wong<sup>2</sup>, and Margrit Betke<sup>1</sup>

<sup>1</sup> Department of Computer Science, Boston University, Boston, MA 02215, USA

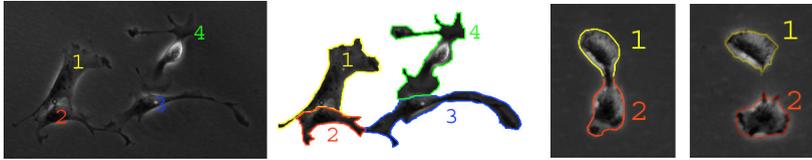
<sup>2</sup> Department of Biomedical Engineering, Boston University, Boston, MA 02215, USA

**Abstract.** We propose a method that automatically tracks and segments living cells in phase-contrast image sequences, especially for cells that deform and interact with each other or clutter. We formulate the problem as a many-to-one elastic partial matching problem between closed curves. We introduce Double Cyclic Dynamic Time Warping for the scenario where a collision event yields a single boundary that encloses multiple touching cells and that needs to be cut into separate cell boundaries. The resulting individual boundaries may consist of segments to be connected to produce closed curves that match well with the individual cell boundaries before the collision event. We show how to convert this partial-curve matching problem into a shortest path problem that we then solve efficiently by reusing the computed shortest path tree. We also use our shortest path algorithm to fill the gaps between the segments of the target curves. Quantitative results demonstrate the benefit of our method by showing maintained accurate recognition of individual cell boundaries across 8068 images containing multiple cell interactions.

## 1 Introduction

Cell morphology and behavior analysis has an important role for studying biological processes, developing biomaterials, and diagnosing and fighting diseases. Reliable automated analysis of cell morphology and behavior depends on accurately finding the contours of each cell in every image. This is challenging because many cells undergo significant appearance variation in short periods of time. Amplifying the difficulty of maintaining shape recognition is the fact that cells frequently approach or touch other cells and clutter for a variety of reasons, including dense clustering of cells on the substrate, imperfect segmentation of the cell boundaries, and insufficient image resolution. The key computer vision challenge addressed in this paper is how to obtain cell shape information as cells deform during interactions with multiple other cells.

General segmentation algorithms that address cell deformation were summarized thoroughly by Rittscher [1]. They are limited to scenarios without apparent cell-to-cell or cell-to-clutter contacts or collisions. When a contact event occurs, the image of the shared boundary typically does not have a sufficiently strong intensity gradient, as in Fig. 1. This makes it difficult to detect the boundary where cells are touching. Segmentation methods that distinguish between cells



**Fig. 1.** Matching cell boundaries through time-lapse sequences are challenging tasks for these cases of colliding, dividing, and otherwise interacting cells

within a cluster of cells include static image-based and temporal-based methods. Nath et al. [2] added constraints to the level set method using the assumption that cells in the image share similar characteristics. Liu and Sclaroff [3] noted segmentation success in the presence of small amounts of cell overlap. Heuristics about expected intensity changes along boundaries between cells, or cells and clutter, have also been adopted [4]. These static image-based methods limit the generalizability across imaging modalities, degree of overlap, or cell type. Object boundaries have been propagated between successive frames such that each pixel on the merged boundary is associated to a pixel on one of the cell boundaries in the previous frame [5]. However, to avoid the high computational cost associated with this partial matching problem, heuristics were adopted which limit segmentation accuracy. The heuristics also make it difficult to increase the scale of the proposed method and apply it successfully to scenarios where multiple cells are involved in an interaction.

In this paper, we treat the segmentation of interacting cells as a partial matching problem between cell boundaries obtained from consecutive frames. Methods addressing the issue of *partial* matching were either applied to rigid transformations only [6] or chose distance measures such as the *edit distance* [7] that are not suitable for modeling the effects of stretching and shrinking. To the best of our knowledge, we are not aware of any competing method that can be applied directly to our elastic partial matching problem. Our contributions are: 1) A general framework to address the task of many-to-one partial matching, as opposed to the traditional tasks of one-to-one or two-to-one matching; 2) A new Double Cyclic Dynamic Time Warping procedure with a mechanism that reuses shortest path trees to speed up computation; 3) A tracking system embedded with the matching scheme that achieves high segmentation accuracy for varying numbers of deforming cells over long durations of interactions.

## 2 Method

Our segmentation method is embedded in a tracking system, which performs initial segmentation, frame-by-frame data association, partial or complete boundary matching, and curve gap completion. During the preprocessing step, we followed the work by Theriault et al. [8] to obtain the initial segmentation. The Hungarian algorithm is adopted to solve the frame-by-frame data association

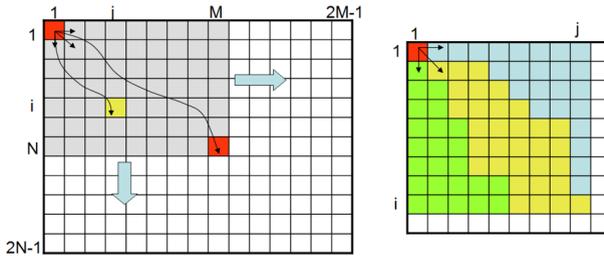
problem and identify merging and splitting events. Based on the relative positions of cells, the number and the order of cells involved in partial matching could be determined. In the following, the main effort is to address the problems of many-to-one partial matching and gap completion.

### 2.1 Partial Matching with Hierarchical Dynamic Programming

The shape of a deformable object is represented by its closed boundary curve  $c$  of length  $L_c$ . The curve is re-parameterized as a function  $C: s \rightarrow \mathbf{f}$ , where  $s \in [0, L]$  is the arc length and  $\mathbf{f}$  is a  $d$ -dimensional local feature descriptor. A matching or alignment between curves  $C$  and  $C'$  is a monotonic mapping function  $g: [0, L] \rightarrow [0, L'], g(s) = s'$ . Given a set of  $K$  candidate curves  $\{C_k\}$  representing individual cells and a target curve  $C_t$  extracted from a merged cluster, where every segment of the target curve must be uniquely matched to a segment of one of the candidate curves, our goal is to minimize the following functional:

$$E(g_1, g_2, \dots, g_K, l_1, l_2, \dots, l_{K+1}) = \sum_{k=1}^K \int_{l_k}^{l_{k+1}} \|C_k(g_k(s)) - C_t(s)\|^2 ds \quad (1)$$

Particularly, we set  $l_1 = 0, l_{K+1} = L$  so that the target curve  $C_t$  will be *completely* matched, while each segment of  $C_t$  will be *partially* matched to the candidate curve  $C_k$ . The search space of this functional includes all possible matching functions  $\{g_k\}$  and cutting points  $\{l_k\}$  along the target curve  $C_t$ . Note that, when  $K = 1$ , Eq. 1 is reduced to the functional commonly used for the *complete* alignment between two curves [9,10].



**Fig. 2.** Extended dynamic programming table for Double Cyclic Dynamic Time Warping. Left: Every  $M \times N$  block has to be computed to find the optimal alignment starting at pair  $(i, j)$  to any ending pair up to  $(i + N, j + M)$ . Shortest paths computed from adjacent blocks can be reused to speed up the computation. Right: After a block  $[1:i, 1:j]$  has been computed, all shortest paths going through  $(2,1)$  (green) remain the same in block  $[2:i+1, 1:j]$ ; Similarly, those going through  $(1,2)$  (blue) can be reused in block  $[1:i, 2:j+1]$ ; those going through  $(2,2)$  (yellow) can be reused in block  $[2:i+1, 2:j+1]$ .

We chose Dynamic Time Warping (DTW) to search for the matching function  $g$ . DTW allows us to capture the significant stretching and shrinking effects of the boundary of moving fibroblast cells. Given two sequences of *ordered* data points, the best alignment is computed with the following subproblem structure:

$$d(i, j) = c(i, j) + \min\{d(i - 1, j) + w_i, d(i, j - 1) + w_j, d(i - 1, j - 1) + w_{ij}\} \quad (2)$$

where  $c(i, j)$  measures the dissimilarity between data point  $i$  from one of the sequences and data point  $j$  from the other sequence and  $w_i, w_j, w_{ij}$  are penalty terms for different types of pairings.  $d(i, j)$  is the accumulated matching cost up to pair  $(i, j)$ . The whole process can be visualized by a dynamic programming (DP) table, as in Fig. 2. If the lengths of two sequences are  $M$  and  $N$  respectively, it takes  $O(MN)$  steps to compute the global optimal alignment for the *complete* matching, which corresponds to the shortest path from entry  $(1, 1)$  to entry  $(N, M)$  in the DP table. As a byproduct, every entry  $(i, j)$  in the DP table encodes the alignment from the pair  $(1, 1)$  to the pair  $(i, j)$ .

If a matching of *closed* curves is desired, as in our case, it is also required to search for the start pair. This has been accomplished with Cyclic Dynamic Time Warping (CDTW) [10]. For the problem of matching *complete* curves, only one of the two sequences has to be cyclically shifted in order to search for the correct start pair. A straightforward implementation of CDTW leads to  $O(MN \min(M, N))$  computations while the fastest algorithm known for this task can compute the optimal alignment of *complete* curves in  $O(MN \log \min(M, N))$  steps, which is based on the key observation that the optimally aligned curves corresponding to different start pairs cannot intersect in the DP table [10].

Standard DTW/CDTW methods have been mostly applied to solve the *complete matching problem*. They are essentially used to search for the alignment function  $g$  only in Eq. 1. The complexity of the *partial matching problem* comes from the need to search for “cutting points” of the target curve in the functional in Eq. 1. Such complexity inevitably leads to the need to search for a start pair by cyclically shifting **both** sequences. We call this extension “Double Cyclic Dynamic Time Warping” (DCDTW). Note that even if we tried every possible pair  $(n, j)$ , for  $n = 1, \dots, N$ , we still do not know if point  $j$  from the second sequence should be matched to any point from the first sequence in first place. At the same time, we also do not know the optimal length of a subsequence we should look for. To address all these difficulties, we here adopt a hierarchical dynamic programming approach to solve the partial matching problem optimally.

At the first stage of our proposed algorithm, we need to compute the optimal alignment for pairs  $(s_k, s_t)$  and  $(s'_k, s'_t)$ , where  $s_k, s'_k \in \{1, 2, \dots, N_k\}$  are the start and end points from the candidate curve  $C_k$  and, similarly,  $s_t, s'_t \in \{1, 2, \dots, M\}$  are the start and end points from the target curve  $C_t$ . To search for the start pair, we extend the DP table to size  $(2N_k - 1) \times (2M - 1)$  as shown in Fig. 2. In the extended table, the shortest path from cell  $(s_k, s_m)$  to cell  $(s'_k, s'_t)$  (such that  $|s_k - s'_k| \leq N, |s_t - s'_t| \leq M$ ) gives the optimal alignment between curve segment  $C_k^{s_k \rightarrow s'_k}$  and  $C_t^{s_t \rightarrow s'_t}$ . To compute all such shortest paths, we can apply DTW for every  $N \times M$  block from the extended DP table, which requires  $O(N^2 M^2)$

steps. However, the following proposition shows that approximately at least one third of this computation is not necessary.

**Proposition 1.** *When applying the standard DTW  $N \times M$  times for every  $N \times M$  block of the extended DP table,  $\frac{1}{3}O(M^2N^2)$  of the computation is redundant.*

The proposition uses the observation that the DP table for one block can be reused for the adjacent blocks because of the non-intersect property of shortest path. We find such reduction is usually much higher in practice.

We denote all shortest paths from DCDTW as the set of matching functions  $G$  with the matching cost  $\{c_{s_t \rightarrow s'_t}^{s_k \rightarrow s'_k}\}$ . Given a segment of the target curve  $C_t^{l_k \rightarrow l'_k}$  such that  $1 \leq l_k \leq l'_k \leq M$ , the optimal partial matching with the candidate curve  $C_k$  is chosen to be  $g_k^{k'} = \arg \min_{g \in G} c_{l_k \rightarrow l'_k}^{s_k \rightarrow s'_k}$  and the corresponding minimum cost is denoted as  $c_{l_k}^{l'_k}$ . There will be  $K \times M \times M$  matching functions selected, each of which represents the best alignment if we want to match one segment cut from the target curve to one of the candidate curves (possibly partially).

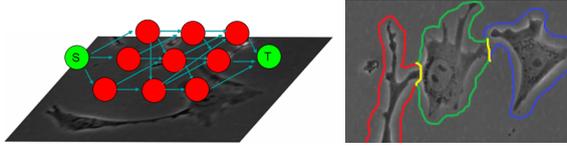
Multiple candidate curves compete for the matching points on the target curve, because every data point on the target curve can only be matched exactly once to one of the candidate curves. We find the best cutting points  $\{l_k\}$  on the target curve through the iterative steps:

$$\gamma(k, m) = \begin{cases} c_1^m, & \text{if } k = 1 \\ \min_{1 \leq l \leq m} \gamma(k-1, l) + c_l^m + w_m, & \text{otherwise.} \end{cases} \quad (3)$$

where  $\gamma(k, m)$  is the compositional cost for matching segment  $C_t^{1 \rightarrow m}$  to  $k$  candidate curves and  $w_m$  is the penalty for choosing point  $m$  as the cutting point. We here assume the relative order of candidate curves is known (through tracking) but the first curve to be matched has to be searched. The above computation takes  $O(KM^2)$  steps, and  $\gamma(K, M)$  is the minimum matching cost corresponding to the global minimum of the functional in Eq. 1. After the global minimum is found at  $\gamma(K, M)$ , the best cutting points can be traced back and the best alignment with each individual cell can be recovered from its corresponding matching function  $g_k^{k'}$ . We want to emphasize that there could be repetition in the  $K$  candidate curves in order to handle the situation where more than one segment from the target curve is matched to the same individual curve, as shown in Fig. 3.

## 2.2 Gap Completion with Shortest Path

To obtain a complete segmentation of interacting cells, we need to fill in the gaps between disconnected pieces of the boundaries of touching cells (Fig. 3). Similar to the path tracing for actin filament segmentation [11], we close the boundary by dynamic programming. For each gap, we create a directed graph that connects two points, the sink and source nodes, that delineate the gap. The pixels of the image patch between these points become nodes in the graph. Two nodes are connected if the corresponding pixels are neighboring pixels. A cost



**Fig. 3.** Left: To fill the gap between boundary pixels represented by  $S$  and  $T$ , the shortest path through a graph, which models the image properties of the gap, is computed, Right: Output result. Given the input of a single combined outline, our method cuts it into the separate boundaries of three cells (red, green, blue) and fills boundary gaps (yellow) where the cells touch.

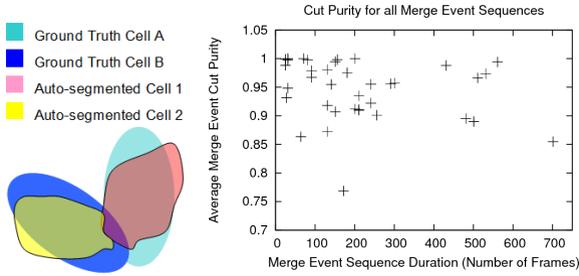
is associated with each node that captures the preference of attraction/repelling when a node is visited. In our application, we use as the node cost the minimum Euclidean distance to the predicted location of the boundary of the cell obtained via tracking. A transition cost is associated with each edge that penalizes moving into the interior of the cell, which is chosen to be the intensity gradient. Once the graph is constructed, the goal is to find the shortest path traveling from the source node to the sink node with minimum accumulated cost. Since the cost along a path is additive, it favors paths with short lengths and avoids crossing high gradient regions. The shortest path can be computed efficiently, again, by using dynamic programming.

### 3 Experiments and Results

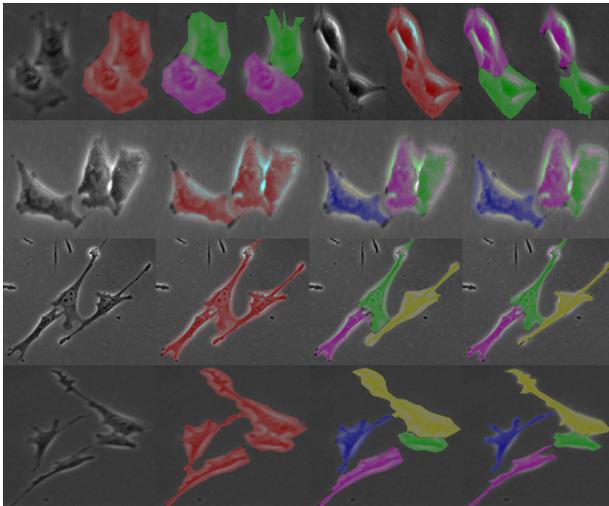
We collected a dataset including phase-contrast images of fibroblast cells of the Balb/c 3T3 mouse strain cultured at 37°C in 5% CO<sub>2</sub> observed with a Zeiss Axiovert S100 microscope. A library of merge events was extracted, which consisted of cell-to-cell/clutter interactions, mitosis, and apparent collisions due to over-segmentation. We collected 41 merge subsequences with 7052 images showing 2-to-1 merging and 975 frames showing 3-1 merging as exemplified in Fig. 5.

We used the precision measure to evaluate the initial outline of a segmented merged object. Given the ground truth  $A$  and segmentation result  $B$ , precision calculates the average overlap between the two regions as  $\frac{|A \cap B|}{|A \cup B|}$ . As a baseline, the average annotator agreement using this criterion is 0.58, which indicates the challenge of obtaining a reliable ground truth due to inter-annotator disagreement. The precision of the segmentation algorithm over 100 non-merged cells was 0.408 which is close to the agreement between annotators. The error is attributed in part to the algorithm incorporating the halos surrounding cells while annotators did not include halos in the cell boundaries when annotating.

To evaluate the performance of separating merged objects, we modified the accuracy measure [12]. Given the ground truth region  $A$  and the segmentation region result  $B$ , accuracy calculates the fraction of the true cell region captured by the segmented region as  $\frac{|A \cap B|}{|A|}$ . To capture the fact that one segmentation result may be better at the expense of another when separating merged objects,



**Fig. 4.** Performance of Boundary Cutting Algorithm. Left: the accuracy is the area of intersection between the ground truth and auto-segmented region normalized by the area of the ground truth. The cut purity is computed as the average of accuracy of all merged objects. Right: Each tick mark on the  $x$ -axis represents the number of frames in a unique merge-event subsequence. The mean values shown are measured using the scores from all cells observed across all images in the subsequence. Higher values indicate stronger results.



**Fig. 5.** Qualitative result of interacting cells. Four images for each merging event are shown, which are (from left to right) the original image, the initial segmentation, the final output and the manual segmentation.

we report a single score we call “cut purity” to indicate the average accuracy score for all objects involved in the merge event. The cut purity measure limits obtaining a good score to the situation when all cells in the merge event are accurately segmented. Fig. 4 shows results for all image sequences with each point indicating an average cut purity score across all images in the merge event. As is shown, our method remains strong for both short and long duration merge

events as fibroblast cells regularly undergo significant amounts of deformation during interactions. Our Matlab implementation can process each merging event in 5 secs on a Intel Xeon(R) 3.20GHz PC.

## 4 Conclusions

We introduced Double Cyclic Dynamic Time Warping to solve the problem of many-to-one partial matching of closed curves. A shortest path was chosen to fill the gaps between the segments of the target curve. The experiments show that our methods produce accurate cell boundaries by appropriately cutting the combined outline of touching cells. In future work, we plan to apply our methods to other types of cells with additional imaging modalities.

**Acknowledgement.** This work is supported by NSF grant 0910908. We thank Matthew Walker, Diane Theriault, Gordon Towne and Quan Fang for data collection.

## References

1. Rittscher, J.: Characterization of biological processes through automated image analysis. *Annual Review of Biomedical Engineering* 12, 315–344 (2010)
2. Nath, S.K., Palaniappan, K., Bunyak, F.: Cell Segmentation Using Coupled Level Sets and Graph-Vertex Coloring. In: Larsen, R., Nielsen, M., Sporring, J. (eds.) *MICCAI 2006, Part I. LNCS*, vol. 4190, pp. 101–108. Springer, Heidelberg (2006)
3. Liu, L., Sclaroff, S.: Medical image segmentation and retrieval via deformable models. In: *Proc. of Intl. Conf. on Image Processing (ICIP)*, pp. 1071–1074 (2001)
4. Pan, J., Kanade, T., Chen, M.: Learning to detect different types of cells under phase contrast microscopy. In: *Proc. of MIAAB*, pp. 49–52 (2009)
5. Bise, R., Li, K., Eom, S., Kanade, T.: Reliably tracking partially overlapping neural stem cells in dic microscopy image sequences. In: *MICCAI Workshop on Optical Tissue Image Analysis in Microscopy, Histopathology and Endoscopy, OPTMHIS* (2009)
6. Bruckstein, A., Katzir, N., Lindenbaum, M., Porat, M.: Similarity-invariant signatures for partially occluded planar shapes. *Int. J. Comp. Vis.* 7, 271–285 (1992)
7. Chen, L., Feris, R.S., Turk, M.: Efficient partial shape matching using smithwaterman algorithm. In: *NORDIA*, pp. 1–6 (2008)
8. Theriault, D., Walker, M., Wong, J., Betke, M.: Cell morphology classification and clutter mitigation in phase-contrast microscopy images using machine learning. In: *Machine Vision and Applications* (2011)
9. Sebastian, T.B., Klein, P.N., Kimia, B.B.: On aligning curves. *IEEE Trans. Pattern Anal. Mach. Intell.* 25(1), 116–125 (2003)
10. Schmidt, F.R., Farin, D., Cremers, D.: Fast matching of planar shapes in sub-cubic runtime. In: *Proc. of Intl. Conf. on Computer Vision (ICCV)*, pp. 1–6 (2007)
11. Li, H., Shen, T., Huang, X.: Actin Filament Segmentation Using Dynamic Programming. In: Székely, G., Hahn, H.K. (eds.) *IPMI 2011. LNCS*, vol. 6801, pp. 411–423. Springer, Heidelberg (2011)
12. Udupa, J.K., LeBlanc, V.R., Zhuge, Y., Imielinska, C., Schmidt, H., Currie, L.M., Hirsch, B.E., Woodburn, J.: A framework for evaluating image segmentation algorithms. *Comput. Med. Imaging Graph* 30(2), 75–87 (2006)