Eurographics Conference on Visualization (EuroVis) 2017
J. Heer, T. Ropinski and J. van Wijk
(Guest Editors)

Volume 36 (2017), Number 3

Uncertainty Footprint: Visualization of Nonuniform Behavior of Iterative Algorithms Applied to 4D Cell Tracking

Y. Wan[1] and C. Hansen[1]

[1]Scientific Computing and Imaging Institute, University of Utah, USA

Abstract

Research on microscopy data from developing biological samples usually requires tracking individual cells over time. When cells are three-dimensionally and densely packed in a time-dependent scan of volumes, tracking results can become unreliable and uncertain. Not only are cell segmentation results often inaccurate to start with, but it also lacks a simple method to evaluate the tracking outcome. Previous cell tracking methods have been validated against benchmark data from real scans or artificial data, whose ground truth results are established by manual work or simulation. However, the wide variety of real-world data makes an exhaustive validation impossible. Established cell tracking tools often fail on new data, whose issues are also difficult to diagnose with only manual examinations. Therefore, data-independent tracking evaluation methods are desired for an explosion of microscopy data with increasing scale and resolution. In this paper, we propose the uncertainty footprint, an uncertainty quantification and visualization technique that examines nonuniformity at local convergence for an iterative evaluation process on a spatial domain supported by partially overlapping bases. We demonstrate that the patterns revealed by the uncertainty footprint indicate data processing quality in two algorithms from a typical cell tracking workflow – cell identification and association. A detailed analysis of the patterns further allows us to diagnose issues and design methods for improvements. A 4D cell tracking workflow equipped with the uncertainty footprint is capable of self diagnosis and correction for a higher accuracy than previous methods whose evaluation is limited by manual examinations.

Categories and Subject Descriptors (according to ACM CCS): I.4.6 [Segmentation]: Pixel classification—I.4.8 [Scene Analysis]: Tracking—J.3 [Life and Medical Sciences]: Biology and Genetics—

1. Introduction

Tracking time-varying structures is an important data analysis task for image-based biology. In biological research, 4D (3D over time) fluorescence microscopy generates a temporal sequence of volumetric data via continuously scanning fluorescently stained living specimens. The capability to track moving structures from 4D scans is fundamental to answer many questions about development, functions, and diseases. However, there is no satisfactory solution, especially for densely packed structures with irregular and changing morphologies. We have been working on a real-world research project – segmenting and tracking cells from time varying (4D) confocal microscopy of zebrafish eye development [KOK*12]. It is a difficult tracking problem for two reasons. 1) The cells to be identified are densely populated three dimensionally in each time point. There are noisy signals to confuse the identification process, resulting many under- and over-segmented components. 2) Cells may emerge, disappear, split, merge, accelerate, and change shape over time. Combined with imperfectly segmented results, it is a challenge to establish the correct associations of cells between time points. We have tested several cell tracking methods and tools from recent publications, including graph-based maximum matching and probability association methods [CGCR13] [ALM*14]. No matter how their parameters were adjusted, we could not obtain satisfactory outcomes for our data sets. The tracking success rates were much lower than the claimed results. It indicates the limited scopes of existing tracking methods and tools. A method worked perfectly on one benchmark data set may fail completely on another. The same can be said for a single data set as well. Different regions exhibit localized characteristics, which may require different methods or adaptive parameter settings. We started working on our own tracking tool, where a data-independent metric to evaluate the tracking performance seemed necessary to overcome the issues in existing tools.

Therefore, we propose the uncertainty footprint, a technique developed to study the local convergence patterns in an iterative evaluation process, which can be used for both cell identification and association in a typical tracking workflow. We derived the uncertainty footprint from the observation that, for several optimiza-

tion processes in our study, the convergence rates at different locations over a spatial domain sometimes exhibit strong nonuniformity, which is introduced by conflicts or uncertainties from competitions between evaluations of neighboring basis functions. We leverage the uncertainties to detail and characterize a process's behavior. For example, we can infer whether a sequence of functions $\{f_t(x)\}$ obtained from an optimization process converges on global or local maxima from its convergence pattern. The uncertainty footprint accumulates local value changes over iterations, a simple calculation that can be incorporated into most iterative evaluation processes. It then visualizes the uncertainty patterns using a traditional toolset for volume rendering, such as color mapping and multidimensional histograms [KKH02]. Insights gained from the uncertainty footprint can further regulate an optimization process, including early iteration termination and parameter adjustments.

For 4D cell tracking, the benefits of the uncertainty footprint are twofold. 1) It allows evaluation and comparison of tracking results through visualizations of both the histograms and the volume rendering with color-mapped uncertainty foot print values. We can optimize initial conditions and algorithms based on feedback from the uncertainty footprint. For a tracking system, this was previously achieved by comparing with manually generated results, which was extremely time consuming and needed to be regenerated for every new data set. There were also simulated/benchmark data, which could hardly be representative for the complex cases in general applications, as seen in our test of existing methods. 2) It allows coordination and cooperation among different algorithms. Since segmentation errors contribute to most tracking issues, we can use the uncertainty footprint to locate the issues and apply a more effective algorithm than that is used for the initial segmentation. Therefore, segmentation and tracking algorithms can mutually support each other and progressively refine the result. It is also more economical at computing resources to allocate more expensive algorithms for only isolated subdomains.

2. Background

2.1. On convergence and uncertainty

Uniform convergence, which requires a sequence of functions $f_t(x)$ to converge at a uniform speed for each point on its domain X, is an important concept in mathematical analysis. Several properties of the sequence, such as continuity, integrability, and differentiability, can then be effectively discussed [Rud76]. The discussion of convergence and uniform convergence started with Cauchy's study on sequences of numbers and functions. A Cauchy sequence of real numbers $\{x_t | t \in \mathcal{N}\}$ is convergent, which requires that for any arbitrarily small positive real number ε, there is a positive integer N such that for all m and $n > N$, $|x_m - x_n| < \varepsilon$. In other words, the values of the sequence get arbitrarily close as the sequence progresses. Cauchy tried to use this definition in the study of function continuity and proved that the limit $f(x)$ of a series of continuous functions $f_t(x)$ is also continuous. Abel quickly found the flaw in the proof, realizing that $f_t(x)$ might not converge at the same rate for different x's [Sør05]. Later work on this topic shaped the modern definition of uniform convergence, which requires the same positive integer N for all x's of the function domain.

Convergence is used as one measure of uncertainty in a system [Liu15]. However, convergence is usually considered as a collective behavior over function domains for numerical analysis applications. This is because uniform convergence is often proved in the derivation from differential equations and functions under discussion are assumingly well-behaved. We feel the need to revisit the concept of convergence rate for certain optimization problems, where the function is defined by a series of partially overlapping bases. The iterative evaluation of the function at the borders between two bases may generate rapidly changing values, introduced by conflicts and competitions of the bases. Thus, convergence rates at different locations of the function domain can exhibit strong nonuniformity, revealing varying uncertainties. When convergence is only considered collectively, subdomains of high uncertainty overwhelm other low uncertainty ones. For example, an optimization may require many more iterations at only isolated locations when results have converged on most of its domain. On the other hand, slow local convergence may propagate and introduce a global impact. For example, convergence on local maxima can be considered as a less certain state, whose local convergence patterns may look distinctively different than that of convergence on global maxima. Therefore, it is necessary to design a tool for local convergence/uncertainty analysis.

2.2. On cell tracking

In biological research, structures to be tracked in a temporal data set are commonly modeled as particles, where their spatial locations are considered in tracking calculations [Sax08]. Another implication for particle tracking is that segmentation and tracking are considered as two separate procedures. Spatial locations are generally calculated as centroids of segmented components and input into a tracking system for evaluations. Because manually detecting and following large numbers of individual particles is not feasible, automated computational methods have been developed by many groups [MUS*14] [CSdC*14] [JD09]. For computational methods for particle tracking, we refer to a series of surveys from Meijering et al. [MSD06] [MDSC09] [MDS12]. They reviewed particle tracking methods including nearest-neighbor linking [HBK*10], spatiotemporal tracing [BDC05], probabilistic data association [GOM04] [GLE*11], and graph-based optimization approaches [SK05] [MJGB15]. They also listed and compared existing tracking tools.

To take advantage of the spatiotemporal relationship of segmented components, matching algorithms based on a graph are commonly used and relatively sophisticate for tracking in biological research. Jaqaman et al. [JLM*08] designed a tracking algorithm that addressed challenges in particle tracking, including high particle density, particle motion heterogeneity, temporary particle disappearance, and particle merging and splitting. They formulated the particle linking steps as global combinatorial optimization problems and identified the overall most likelihood of trajectories. Similar strategies are perhaps best modeled as the maximum bipartite matching problem in graph theory [TK06]. Segmented particles from two consecutive time points are naturally separated into two groups T and $T+1$; only two elements from different groups can be paired; the tracking problem is solved by finding the most pairs based on certain criteria, which are usually consolidated as

edge weights of a graph. The solution is equivalent to a maximum flow problem and similar to finding the shortest path on a graph [Sch02]. It repeatedly searches for augmenting paths with maximum weights. Mosig et al. [MJW*09] used overlap between time points for maximum bipartite matching and tracked moving cells from videos. Chatterjee et al. [CGCR13] used distances between particles to calculate weights, which were then minimized along with the cardinality (number of link pairs). Like the Dijkstra's shortest path algorithm [Dij59], single weight is desired for the maximum bipartite matching, while the multiweighted problem is NP-hard.

However, uncertainty analysis in cell tracking is overlooked, despite that data complexity often makes tracking results unreliable. Although research literature usually uses several benchmark data sets from real-world scans or simulations, the tracking validation and diagnosis in practice are still based on manual examinations, because no benchmark data can represent the variety/complexity in real applications and the choice of benchmark data is easily biased to favor a method under discussion. Uncertainty quantification and visualization have made substantial progress in other areas, such as MRI scans and simulations [vPOBB*10] [vPOBB*11], ultrasound data [BDH*15], and nonparametric model extraction [ASE16]. As cell tracking is performed on a complex 4D data set, the lack of quantified uncertainties at different locations of the data domain limits previous tracking methods. A generally applicable and intuitive uncertainty analysis method is thus desired.

3. Uncertainty footprint

3.1. Definition

For an iterative process over iterations $T = \{t | t \in \mathcal{N}\}$, its evaluation outcome is a sequence of functions $f_t(x)$, whose domain is X. The uncertainty footprint is defined as:

$$g_t(x) = \begin{cases} 0, & t = 1 \\ \sum_1^{t-1} |f_{\tau+1}(x) - f_\tau(x)|, & t > 1 \end{cases} \quad (1)$$

Here, $|\cdot|$ is the absolute value for real-valued functions, although other norms can be used, to measure the pointwise difference between two functions from consecutive iterations. Notice that the definition differs from a commonly used criterion for convergence, which sums over the function domain at each iteration, while our definition sums over iterations at each point of the domain. In plain language, the uncertainty footprint accumulates the pointwise difference between two iterations over all iterations. Intuitively, it leaves a "footprint" each time there is a value change at any location on the function domain. We can compare the footprints left from different local subdomains and measure uncertainty. The iterative process can be seen as making value adjustments based on its algorithm, without the explicit knowledge about the true value. The more the process is uncertain about an adjustment, the greater it changes the value, hence more footprints left in that area. Over iterations, we can observe the accumulation of the footprints and trace to the areas of high uncertainty. By analyzing the patterns of how uncertainty footprints accumulate, we can draw conclusion on the cause and design improved methods.

3.2. Characteristics

Several characteristics of the uncertainty footprint allow it to be easily analyzed.

Sharing of iteration numbers. The uncertainty footprint is a conjugal sequence to the original function sequence $f_t(x)$. There is one "snapshot" of uncertainty for each iteration. It also means that the uncertainty footprint must be computed within the iterations.

Sharing of function domain. The uncertainty footprint shares the function domain with the original sequence $f_t(x)$. It means that each point x on the domain has a sequence of values to represent its evolving uncertainty. Maintaining an uncertainty record at each point on the domain allows us to compare and differentiate regions.

Nonnegative and real-valued. The uncertainty footprint values are monotonically increasing from 0 at initial conditions. They are naturally mapped to a range from low uncertainty (close to 0) to high uncertainty (far from 0). We can compare and order uncertainty footprint values. They can also be color mapped and visualized using similar techniques for the presentation of the original sequence $f_t(x)$, such as volume rendering.

3.3. Histograms of uncertainty footprint values

For each iteration, we can study the uncertainty footprint value distribution using a 1D histogram. The histograms of all iterations are concatenated to form a 2D histogram. Several meaningful patterns in the 2D histogram may be observed for a detailed study of the behavior of the process. We made an illustration of the 2D histogram of uncertainty footprint values (Figure 1). For an iterative process, local features with nonuniform convergence are visualized as lines or curves in a 2D histogram. The meanings of the 2D histogram patterns are as follows. 1) Diagonal lines denote the aggregation of value adjustments. If values of a certain region on the function domain do not quickly converge, our evaluation of the uncertainty footprint accumulates the changes over time. When the value change of the region is at a relatively constant rate, a strong linear relationship between the uncertainty footprint values and iteration numbers can be observed. Since different regions may differ at the accumulation rate of their uncertainty footprint values, the diagonal lines split and have different slopes, signifying different rates. A high nonuniformity of the process is visualized as a wide range of the spread of lines. 2) Horizontal lines denote convergence. The uncertainty footprint values are fixed once convergence is reached. Some uncertainty footprint values stay at a low level, denoting a low uncertainty convergence, while others stay at high uncertainty levels. 3) Some diagonal lines turn into horizontal lines because of convergence. However, diagonal lines that do not turn denote oscillation. 4) A noisy 2D histogram not showing any of these patterns represents a highly nonuniform or divergent process.

The 2D histogram of uncertainty footprint values allows us to distinguish different patterns representing nonuniform behaviors and trends of local features, which can be obscured if the 1D histograms are examined individually. For example, we can locate the iteration number T where two lines separate. Then, we can find out the two different uncertainty levels at T and filter the result on the function domain. This allows pinpointing the two regions with different convergence rates. Since many optimization problems use a

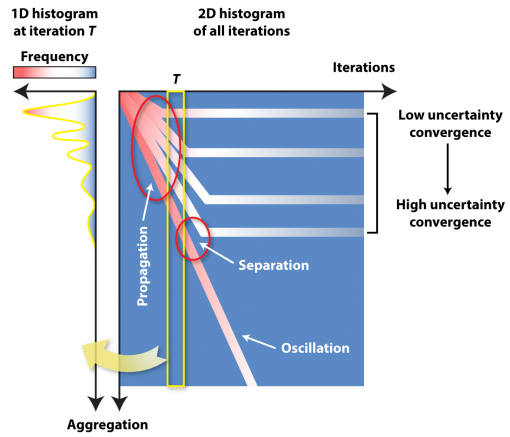

Figure 1: *An illustration of a typical 2D histogram of the uncertainty footprint values in an iterative process. The 2D histogram is a concatenation of a sequence of 1D histograms at consecutive iteration numbers. Each 1D histogram shows the frequency for different uncertainty footprint values. Frequency values are color mapped. Several patterns are interesting to the analysis of a 2D histogram. Lines or curves represent local features having similar convergence. Diagonal lines denote the aggregation of value changes. The steeper a diagonal line is, the faster the changes. Horizontal lines denote convergence. The locations of the horizontal lines determine the uncertainty at convergence. If a diagonal line does not turn to a horizontal line, it means oscillation. We can also find out when oscillation separates from convergence.*

mixture of bases with local support, disparity of convergence rates at different regions can be easily visualized. When a region of high uncertainty is isolated while most of the domain yield satisfactory results, we can apply a different setting or change the optimization method for the problematic region. The 2D histogram of uncertainty footprint values is also suitable for automated analysis. For example, we can use the Hough transform to detect lines and curves, the intersections of which are separation points [DH72].

4. Applications and results in 4D cell tracking

We designed uncertainty footprint and its histograms for the study of 4D cell tracking from live scans of zebrafish development [KOK*12]. For the uncertainty footprint values to be calculated and analyzed, two conditions need to be satisfied in an application. First, the computation is an iterative optimization process, generating a sequence of functions $\{f_t(x)|t \in \mathcal{N}\}$. Second, the functions $\{f(x)\}$ are defined on a domain supported by a mixture of local bases. The second condition is less obvious from the definition of the uncertainty footprint. We require local support based on the empirical observation that nonuniform convergence rates are more obvious and the analysis results have more clear meanings. As in the example applications, both conditions can be easily satisfied or constructed. Here, we adopt a common tracking workflow and discuss the applications of our method in two stages of the workflow – cell identification and association. For cell identification, or segmentation, we focus on the Expectation-Maximization (EM) algorithm for Gaussian mixtures [DLR77], which is demonstrated as an introductory example. For cell association, we use a graph-based optimization method, which is constructed from a maximum matching algorithm [Gib85] so that the two conditions for uncertainty footprint are satisfied.

4.1. EM for cell segmentation

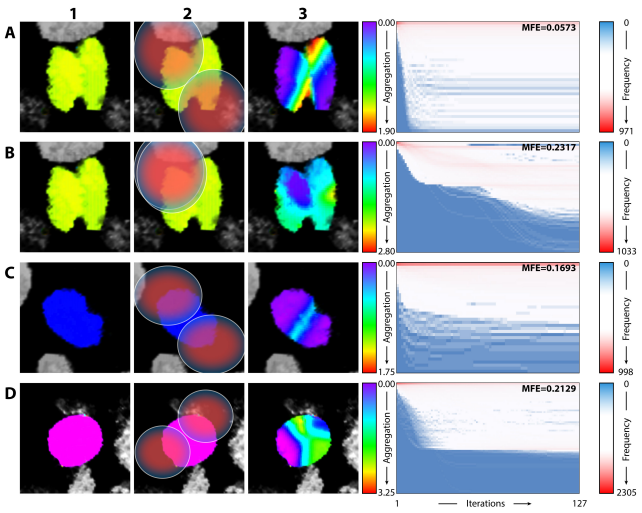

Figure 2: *Visualizations of the uncertainty footprint values from cell segmentation using the EM clustering method. (A) We use the EM clustering algorithm to segment two fused cells in a confocal microscopy scan of zebrafish eye development. The first column shows the two fused cells selected and colored. The second column shows the parameter estimation for two Gaussian distributions as the initial condition for the EM. The third column shows the uncertainty footprint result mapped to a purple-green-red lookup table. The fourth column is the 2D histogram of the uncertainty footprint values. (B) The same cells in A are segmented using the same EM process. However, the initial condition is different. The two initial Gaussian distributions are set to overlap each other. (C) We use the EM algorithm to segment a single cell with a concave shape. Despite that certain isolated parts exhibit higher uncertainty than others, we are able to detect a possibly acceptable segmentation at the dent on the cell. (D) We use the EM algorithm to segment a single cell with a round shape. The result exhibits high uncertainty and slow convergence. For all examples, we truncate or pad the 2D histograms to show the results from 127 iterations. Actual iteration numbers vary for different cases depending on convergence criteria. However, the trends to reach convergence can be clearly observed in the 2D histograms.*

Cell identification is commonly performed as a separate step in a tracking workflow. Depending on input data's complexity, several image segmentation methods are commonly employed. For example, when the cells in a volume data set are well separated by low intensity spaces, simple thresholding and the connected component analysis are sufficient for cell identification [GW07]. As input data become less ideal with the introduction of noise and blurriness,

more sophisticated methods are needed, such as Otsu thresholding [Ots79] and the watershed algorithm [BM93]. Here, we examine a specific clustering method, the EM algorithm for the Gaussian mixture [DLR77]. The EM algorithm outperforms many other methods for segmenting two or several fused cells. Fused cells are common in confocal microscopy, as diffraction of fluorescence emission makes it difficult to resolve adjacent fluorescently stained structures with fine details. Segmentation methods based on local intensity and gradient values have poor performance because there are usually no low-intensity borders between fused cells. Therefore, a shape-based method, such as the EM for the Gaussian mixture, is more effective, as some types of cells can be modeled using a Gaussian distribution [ALM*14].

To segment a group of $N(N \geq 2)$ fused cells, we strictly follow the EM clustering algorithm on a Gaussian mixture. The 3D coordinates of the voxels of the cell group form the sample set $X = \{x_j | j = 1..J \text{ voxels}\}$. We would like to estimate the parameters for a Gaussian mixture, $S = \{\{\tau_i\}, \{\mu_i\}, \{\Sigma_i\} | i = 1..N\}$ (τ_i, μ_i, and Σ_i are weight, centroid, and covariance matrix for each Gaussian distribution), which best approximates the sample set by maximizing the likelihood function $L(S;X) = \prod_1^J \sum_1^N Gauss(x_j;\mu_i,\Sigma_i)\tau_i$. Given an initial Gaussian mixture S_0, the EM algorithm is an iterative evaluation process on overlapping bases. For each iteration, we perform an E step, which updates the membership probability ($P_i(x_j) = (\tau_i Gauss(x_j;\mu_i,\Sigma_i))/(\sum_1^N \tau_k Gauss(x_j;\mu_k,\Sigma_k))$) for the sample set based on current parameter estimation S_t of iteration t, and an M step, which updates the parameter estimation S_{t+1} by maximizing the likelihood function L_t. The iteration terminates if the change of the likelihood function becomes sufficiently small ($|L_{t+1} - L_t| < \varepsilon$), or when a predefined maximum iteration number has been reached. The reaching of a certain maximum iteration number is usually considered a divergent result. However, strictly speaking, the EM convergence is guaranteed for basis functions of the exponential family with compact parameter space [Wu83], although it can converge to local maxima with undesired segmentation results.

For cell segmentation, there are three issues preventing the EM clustering algorithm to be directly applied in real-world data sets. 1) The number of clusters, or Gaussian distributions in the mixture, needs to be determined before applying the algorithm. It is a difficult task for a complex data set such as the confocal scan of cells of zebrafish eye development 2) The clustering results are dependent on the initial Gaussian mixture estimation. There is the possibility that a convergent result only achieves local maxima. We need a more detailed method to evaluate the clustering process than the simple convergence criterion, so that we may adjust the initial parameter estimation to improve the clustering result. 3) It is a computationally expensive process. One typical zebrafish eye scan contains several thousand cells. It would be more efficient and accurate if we could identify fused cells and only apply the EM algorithm locally. The evaluation of the uncertainty footprint is key to the solutions of these issues.

To evaluate the uncertainty footprint for the EM clustering algorithm, we define the membership probabilities $\{P_{i,t}(x_j)\}$ at iteration t as function $f_t(x)$ in Equation 1. Since $\{P_i\}$ is vector-valued, we use the L1 norm to evaluate Equation 1, i.e., summing up the membership probability changes for the Gaussian distributions. For each iteration, we first evaluate the standard EM steps, which are immediately followed by the calculation of the uncertainty footprint values for the membership probabilities via accumulating the changes at each sample point (voxel). Next, a 1D histogram is generated for the uncertainty footprint values. We terminate the iteration when the convergence condition is satisfied or a maximum iteration number reached. The 1D histograms from all iterations are concatenated to form a 2D histogram. Finally, we visualize both the 2D histogram and the volume rendering with color-mapped uncertainty footprint values.

Figure 2 demonstrates this process for several cell segmentation examples. We select two fused cells or a single cell to test our method and observe the visualized patterns from the uncertainty footprint. Figure 2A shows the segmentation of two fused cells. We are certain that there are two cells in the group only because the information can be inferred by examining the time sequence. Otherwise, this is considered as an ambiguous case for segmentation. We use a simple heuristic to initialize the EM process: the center of the first Gaussian is placed at the voxel with the highest intensity value; the center of the second Gaussian is placed at the voxel farthest from the first Gaussian; both covariance matrices are the same and estimated using all voxels from the group (Figure 2A2). A visualization of the color-mapped uncertainty footprint reveals that the two Gaussians "sweep" through the cell group pivoting at the two corners and settle down at the desired segmentation (Figure 2A3). The 2D histogram shows a fast convergence, as horizontal lines dominate the graph after about 40 iterations. The distribution of uncertainty is rather uniform (horizontal lines in the 2D histogram), which can be also confirmed by the regular gradient patterns in Figure 2A3. Figure 2B shows another result for the same group of fused cells. However, we use a different initial condition so that the two Gaussians are overlapping. Comparing the 2D histogram of Figure 2B to that of A, we observe slower overall convergence, higher aggregated changes, and less uniformly distributed lines. The patterns in the 2D histogram show that the Gaussians have difficulties at pathfinding to the global maximum, as local uncertainties (the isolated curves) propagate and prohibit convergence. The segmentation result is also undesired, which separates the group horizontally. Figure 2C shows how the EM clustering works for a single cell. The patterns in its 2D histogram are similar to Figure 2A. However, the convergence is slower as demonstrated by less uniform line distributions. Figure 2D is an extreme case, which usually would not have been segmented. Its result shows high aggregated changes, uneven propagation, and slow convergence.

For the EM algorithm, the 2D histogram provides a detailed map for local convergence analysis, which was not available when only one convergence criterion (e.g., the likelihood function L) was used. Instead of classifying a clustering result into convergent or divergent, we are now equipped with a tool to better distinguish the in-between cases. As shown in the examples, the 2D histogram of an effective EM clustering should show minimal isolated local uncertainty, and the overall value aggregation should be also low (Figure 2A). When local uncertainties become prominent in the 2D histogram (shown as the isolated curves in Figure 2B), and the local uncertainties propagate to influence other parts of the domain (shown as the steps at the border between the blue and white

regions in Figure 2B), a reconfiguration of the initial conditions should be considered. Otherwise, the convergence is likely on local maxima. If local uncertainties have a clear general trend to converge but with slow convergence rate, it is most likely to be a dubious result for a highly ambiguous data set (Figure 2C). Therefore, through the visualization of the 2D histograms of the uncertainty footprint, we can improve the EM clustering from several fronts. 1) A desired result shows prominent horizontal lines early in the iterations. We can terminate the iterations early to save computing time when agreggated values become horizontal lines. 2) We can reconfigure the initial conditions to improve the result when strong nonuniform patterns appear. 3) We can choose to accept or reject a clustering result based on 2D histogram analysis. We factor the patterns of a 2D histogram into aggregation, which measures how high the uncertainty footprint values reach, and nonuniformity, which describes how lines are distributed. The uncertainty value can be obtained from the height of a 2D histogram (Figure 2), whereas the nonuniformity is calculated as the mean filtered entropy (MFE) of the histogram [GW07]. The decision to accept an EM clustering result is made based on Figure 3. The green region represents acceptable results with low aggregation and nonuniformity.

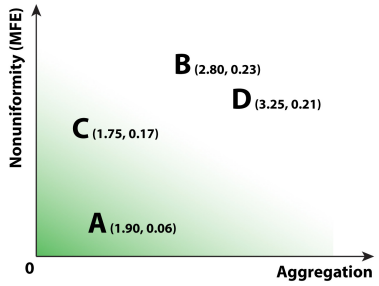

Figure 3: *A summary of EM result classification using the 2D histogram of uncertainty footprint. Two factors are examined. The horizontal axis of the graph represents the aggregation of local value changes. The vertical axis represents how much local uncertainty footprint value spreads. The more lines or curves are spread, the higher nonuniformity of local convergence. We place the four examples in Figure 2 on the graph. An acceptable result should have both values low, not just convergence, indicated as the green region.*

4.2. Graph-based optimization for cell association

Graph matching is a well-established method for cell association in a tracking workflow. Especially when cells have been segmented, the choice of graph matching to associate cells from two adjacent time points is intuitive: a graph $G = (V,E)$, where V and E are sets of vertices and edges respectively, defines the domain of an optimization problem; the vertex set V is a collection of all segmented cells from both time points t and $t+1$; the edge set E defines all plausible associations between cells from both time points; the optimization objective is to maximize the number of nonadjacent edges, called matched edges; otherwise, an edge is unmatched. As the vertex set is naturally divided into two disjoint sets for time points t and $t+1$, and an edge can only connect two vertices from different time points, the graph G is also bipartite and the optimization process is called the maximum bipartite matching. For cell tracking, a weighted graph is commonly used, where parameters quantifying the relationships between cells are added as edge weights, such as distance, overlapping size, size similarity, shape similarity, etc. The maximum matching on unweighted or weighted graphs can be constructed using the Edmonds's matching algorithm, which is a recursive algorithm exhaustively searching and comparing all augmenting path combinations to maximize the matches [Edm65]. An augmenting path is a graph path **a)** having alternatively matched and unmatched edges, and **b)** starting and ending at unmatched vertices, while a path satisfies **a)** is an alternating path. Therefore, an augmenting path is also an alternating path.

When a segmentation result is perfect and cells do not split, merge, emerge, or disappear over time, the bipartite maximum matching generates optimal results for the tracking problem. However, rarely do real-world data meet the perfect conditions. Uncertainties generated in scanning, preprocessing, and segmenting a data set can be problematic when applying the maximum matching algorithm directly. For a weighted bipartite graph, the Edmonds's maximum matching algorithm has three intrinsic issues for tracking imperfect data. 1) Two or even more matchings may have near identical weights. Simply discarding matchings with smaller weights does not consider the uncertainties in weights and is not robust. 2) For a densely-packed data set, a large number of vertices on a graph can be chained together on an augmenting path. A small disturbance on the weights can cause a chain reaction. This "butterfly effect" is undesired in tracking cells as cell movements are bounded within a neighborhood. The association of a pair of cells should not influence the results outside their neighborhood. 3) Multiple weights can be considered, such as distance, overlap size, similarity, etc. The original algorithm does not provide a solution to the multiweighted problem. Most of the issues are addressed by the introduction of the uncertainty footprint in graph matching.

For the uncertainty footprint to be evaluated properly, it requires an iterative process with progressively updated matching results. The idea behind our construction is simple: we limit the weighted maximum matching within a subgraph centered at vertex v ($v \in V$). The subgraph is extracted by including all vertices that can be visited from v via paths of length $l \leq L$, where L is a predefined maximum length. We visit and process all vertices on a graph in one iteration and then repeat. It resembles an iterative convolution process on an image. To leverage the uncertainties in edge weights, we introduce the concept of weight similarity, which is defined as $|w_1 - w_2|/(w_1 + w_2)$, where w_1 and w_2 are two weights to be compared. When comparing two locally computed matchings for a subgraph G_v at vertex v, we choose one over another only if their weights are not similar. Otherwise, both matching results are kept for v. Strictly speaking, the union of two matchings is not necessarily a matching. However, we still use the term a matched edge to denote a valid association between two vertices, and an unmatched edge for an invalid association that merely means neighborhood relationship. Therefore, unlike a standard matching, we allow a vertex to have multiple matched edges, where the number of matched edges is called the valence of the vertex. In the iterative process, a matched edge may change to an unmatched one, or vice versa,

to describe which we use the term edge flipping. Our constructed algorithm is detailed by the following pseudocode.

```
create graph G = (V,E) based on spatial relationship between
  time points t and t+1;
assign edge weights w's;
L = maximum length for subgraph;
Th_sim = similarity threshold;
initialize matching result M to {0};
for all iterations do
    for each v in G do
        if valence of v ≠ 1 then
            G_v = construct a subgraph using L;
            M = M \ G_v;
            {M_vi} = find all mutually exclusive matchings of
              G_v;
            Sort {M_vi} by weights;
            for all similarly weighted {M_vi} starting from the
              maximum weight do
                M = M ⋃ M_vi;
            end
        end
    end
end
```

The adoption of subgraphs and a similarity measure addresses the robustness issues of the original maximum matching algorithm. Multiple weights are also supported, as each iteration can be dedicated to one type of weight. The subgraphs act as partially overlapping basis functions; the iterative matching evaluations at where they overlap essentially generate inherent local ensembles for the uncertainty footprint analysis. Updating the values for one basis also changes its neighbors, creating conflicts and competitions. Greater conflict means greater uncertainty. To compute the uncertainty footprint, we define the valences of all vertices on graph G (G is the function domain) as the function sequence $f_t(x)$. Equation 1 is evaluated as the accumulated valence change for each vertex, which is equivalent to edge flipping counts. The meaning of the uncertainty footprint is intuitive: more conflicts present at one location induce greater valence change, and therefore greater uncertainty about the result.

We use the same visualization methods as in Section 4.1 to examine the uncertainty footprint in the cell association process. Figure 4 shows an example when we track cells between two time points of a scan of zebrafish eye development. We use the Synthetic Brainbows built within the FluoRender system to segment and color the volume data of both time points [WOH13] [WOCH09]. The result is then processed with our graph matching algorithm. In this example, we set the similarity threshold to 0.2 (two weights are considered similar when their difference to sum ratio is smaller than 0.2), and 2 for the maximum path length of subgraph extraction. We run 13 iterations and the 2D histogram of the uncertainty footprint is in Figure 4. We use two different representations of the 2D histogram, as one interesting feature is more obvious in the 3D ribbon graph: a low peak separates from the main peak and moves with a constant speed. The low peak must be related to the highly uncertain cells that our algorithm has difficulty to match. To study the nature of this phenomenon in detail, we filter the volume rendering result at time point t_1 by selecting a range only including the small peak (Figure 5A). We also play back the time sequence using the FluoRender system [WOCH12]. The result shows that most of these high uncertainty cells are generated from over-segmentation or noise data. Figure 6A and B show two such cases. Uncertainty in these cases increases because two components are competing to match with one cell over the iterations. The color-mapped 2D his-

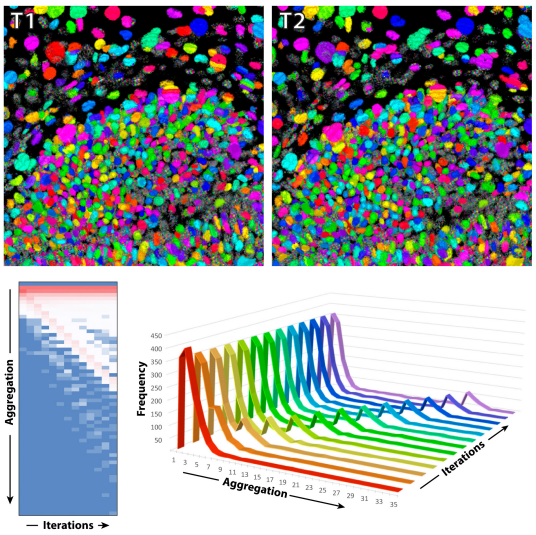

Figure 4: *Uncertainty footprint result for tracking two time points of a scan of zebrafish eye development. The top panels show the volume renderings of the two time points. The volume data are segmented and colored using the Synthetic Brainbows. The bottom panels show two representations of the 2D histogram of the uncertainty footprint, a 2D color-mapped image and a 3D ribbon graph. The 3D ribbon graph shows two prominent ridges, representing rapidly converged results and uncertain results respectively.*

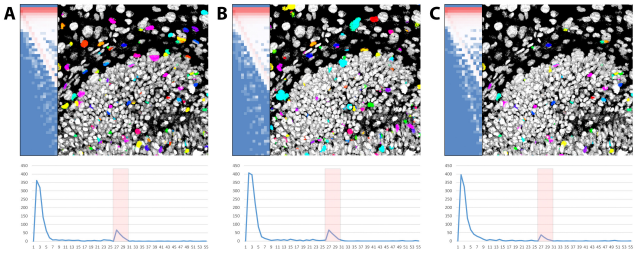

Figure 5: *Filtering the volume rendering results by selecting a range from the histogram of uncertainty footprint. Only cells within the selected range (red regions on histograms) are colored using the Synthetic Brainbows. (A) A result generated using a similarity threshold value of* 0.2. *(B) A result generated by adjusting the segmentation parameters so that fewer cases of over-segmentation are present. (C) A result generated from the same segmentation as in A, but using a similarity threshold value of* 0.85.

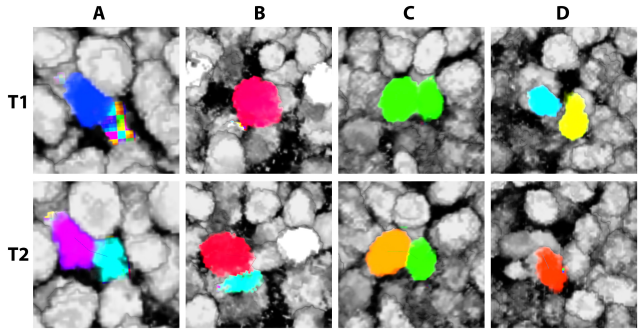

Figure 6: *Examples of high uncertainties in cell tracking. (A) Two cells are tracked over time. In T1, one cell with varied colors is identified as noise because of low intensity values, which should be matched to the cyan cell in T2. Instead, the blue cell in T1 is matched to both cells in T2. (B) One cell is tracked over time. In T2, the cell is over-segmented to two parts because of its shape. The red cell in T1 is matched to both parts in T2. (C) Two cells are tracked over time. In T1, the two cells are under-segmented and fused into one green component. In T2, the two cells are correctly segmented. The green component in T1 is tracked to both cells in T2. (D) The yellow cell in T1 disappears in T2. The matching of the cyan cell in T1 to the orange cell in T2 is correct.*

togram in Figure 4 also reveals uncertainties propagating at a higher speed than the low peak, which are lines with steeper slopes. Those are cases where three or more cells are competing for one component.

The uncertainty footprint and visualizations of its 2D histograms become an effective tool to diagnose and improve our cell tracking results, as manually checking each cell is time-consuming. In Figure 5A, the two peaks have clear meanings: the major peak represents cells that can be confidently tracked, and the minor peak consists of mostly segmentation problems. The size ratio between the two peaks represents the tracking quality, which we can improve in two ways. 1) We can improve the input data by adjusting segmentation parameters or using a different routine for cell identification. The uncertainty footprint is used to compare the quality of input data from different segmentation routines or settings. For example, we have determined that the high uncertainty of the result in Figure 5A are from over-segmentations. We adjust the settings of the Synthetic Brainbows so that the segmentation generates more under-segmentations. The same similarity threshold (0.2) is then used to match the two time points. The result is shown in Figure 5B. We consider the result a minor improvement, as the number of confidently tracked cells increases. Figure 6C and D show two examples when we filter the volume rendering results by uncertainty: Figure 6C an example for under-segmentation, and Figure 6D a case where one cell disappears. For under-segmented cells, we can leverage the result in Section 4.1, as the EM algorithm is shown to be effective to separate fused cells. 2) We can make improvements to the matching algorithm. Similar to the over-segmented case of Figure 6B, despite high uncertainty, many tracking results of over-segmented cells are in fact valid. In our matching algorithm, a mechanism to accommodate over-segmented results is adjusting the similarity threshold value. Figure 5C shows the result when we adjust the similarity threshold to 0.85 with the same segmentation input as for Figure 5A. Consider the case in Figure 6B, the two parts of a cell are considered similar in size; both are matched to the correct cell and this configuration no longer incurs conflicts. The histogram of Figure 5C reflects this improvement with a significant reduction of the uncertain peak comparing to Figure 5A, since many of the over-segmented cells in Figure 5A now have lower uncertainty values. To confirm the result, we manually examined 1177 segmented and identified components in the results of Figure 5C. When a similarity threshold of 0.85 is used, there are 75 cases with uncertainty values between 26 and 31 (the minor peak of Figure 5C), among which, 8 cases are due to under-segmentation, 37 cases due to noise, 19 cases due to over-segmentation, and 11 other cases including mitoses and disappearance of cells. Comparing to the similarity threshold of 0.2, there are 73 over-segmented cases reduced from the peak in Figure 5A and moved to lower uncertainties. There are 1030 cells fall under the major peak with uncertainty values between 1 and 10. We only detected two cases having problems, where the same under-segmentations were generated for both time points. There are 55 cells fall between the two peaks with uncertainty values between 11 and 25. They are generally considered as correct results, because most are cases similar to Figure 6B or influenced by noise. The remaining 17 cases are mostly noise insignificant to the analysis. For those under-segmented cells, the EM clustering method can be applied to refine the results. In the supplementary results, we include more EM examples to demonstrate how we evaluate resegmented results using the 2D histograms of the uncertainty footprint. The results from manual examinations demonstrate that a good correlation exists between the tracking/segmentation quality and the uncertainty footprint value distribution. It also shows that resegmentation can be incorporated into the cell matching process and guided by the uncertainty footprint evaluations.

5. Discussion

We designed the uncertainty footprint in our specific application of tracking cells in 4D confocal scans, as there has been a real need for an evaluation tool for such a complex task. Previously, only manual evaluation was available but not practical. The uncertainty footprint leverages the nonuniformity of local convergence in an iterative process and reveals trends of local convergence using histograms. When the disparity of convergence is shown as distinctive patterns in the 2D histogram of the uncertainty footprint, we can classify the result back on the function domain using a series of well-established methods already used in volume visualization. We have demonstrated the features of the uncertainty footprint for two different algorithms from our 4D tracking workflow. We can use the uncertainty footprint to quickly evaluate a tracking result and pinpoint local structures causing problems. The benefits are most obvious when data to be analyzed contain complex structures and noise.

Although a detailed presentation of our complete 4D tracking workflow and the resulting system is out of the scope, we would like to emphasize the importance of the uncertainty footprint with an outline of our tracking workflow. First, we use the Synthetic

Brainbows to segment and color an input 4D scan of cell development. The segmentation is fast and GPU-based, but many segmentation issues may occur. Next, we run the graph-based matching algorithm of Section 4.2. The uncertainty footprint of the matching results is generated on the fly. However, the matching iterations are controlled by a real-time analysis on the growing 2D histogram of the uncertainty footprint. The algorithm examines the histogram and searches for diagonal lines. When the peak of high uncertainty separates from the main peak of converged results, the iterations are interrupted to run a resegmentation. It automatically selects cells or components based on the high uncertainty values and uses the EM algorithm for the segmentation. Hence, the EM algorithm is only applied to local structures with a limited computational overhead. Furthermore, the cluster number for the EM clustering is also determined from the uncertainty footprint – the number of high uncertainty edges from one vertex, which can be derived from the line slope in the 2D histogram. Since high uncertainty cells or components are not necessarily under-segmented, we use the uncertainty footprint again to evaluate the EM results and only accept highly reliable ones, which are shown in Figure 2 and the supplementary results. Finally, we repeat the iterations until the user-defined criteria are satisfied.

Essentially, the uncertainty footprint is a binding force in our tracking workflow. The capability to examine and reevaluate the segmentation results within the matching iterations substantially improves the outcome. We made a comparison with the method of Amat et al. [ALM*14] Figure 7A1 and A2 show their result and ours respectively. We used an example data set downloaded from the link provided by their publication. We manually examined 1904 cells over 31 time points and counted incorrectly tracked cells. Despite an extremely clean data set, we counted 18 incorrect cell associations from Amat et al.'s result, which achieved a success rate of about 99%. The errors were mainly due to incorrectly linking partially segmented cells, as shown in Figure 7A1. Using our tracking workflow, a 100% success rate was achieved for the same sample data, as segmentation issues were detected and fixed automatically. The method of Amat et al. also adopts a tracking quality evaluation method similar to Kan et al. [KLB*13], which is based on track consistency, i.e., if one cell is tracked to two cells or none, the tracking needs a manual examination. Comparing the uncertainty footprint to their evaluation method, ours can provide more details and differentiate uncertain cases. As shown in Figure 7B1 and B2, a problem not involving track consistency cannot be detected by their method. We also experimented Amat et al.'s method on our zebrafish eye scan to test its performance on real-world data. However, no matter how we adjusted the parameters, a decent tracking result could not be obtained for comparison. As shown in Figure 7C1 and C2, only less than 20% of the cells were tracked. The reliably tracked cells were under 10%. The comparison results demonstrate that the commonly used benchmark method for cell tracking may be highly biased. A method with 99% success rate on one or several test data sets can completely fail on others. This is the reason that we need methods independent of benchmark data. We strongly believe that building uncertainty analysis into tracking algorithms is one viable solution and the uncertainty footprint is our contribution in this field.

Although the development of the uncertainty footprint is based on our specific application of 4D cell tracking, it is easily generalized for uncertainty analysis for most iterative processes. The histograms from the uncertainty footprint provide the visual analytic tools for human-in-the-loop interactions to regulate the iterative processes. Quantitative analyses can also be designed using the results from previous research on simple histograms and multidimensional histograms. We have demonstrated that, depending on the specific applications, different quantification methods are needed. For example, for the EM-based cell segmentation, we used the MFE and aggregation jointly; and for the graph-based cell tracking, we used the extracted 1D histogram, representing the distribution of the tracking quality. Our generalized method may also shed light on many other nonconvex optimization problems. Different from the conventional uncertainty quantification methods, which would have introduced perturbations and then quantify the outcome using statistical methods, our uncertainty quantification is indirect. Just like a detective using a person's footprints to analyze his/her behavior, we are deriving the uncertainty of an iterative process from its nonuniform behavior. By comparing the aggregated changes of different subdomains on a 2D histogram, we can understand confidence in convergence. When distinctive patterns appear for an uncertain convergence or diverging process, we can isolate the subdomains and perform a detailed study to resolve issues. The topology of the objective function over the spatial domain can influence how the uncertainty footprint results are interpreted. Intuitively, when

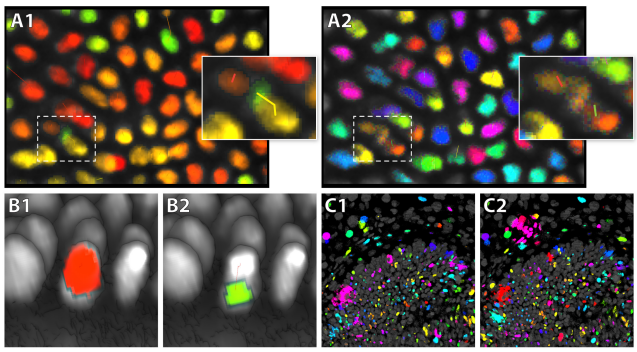

Figure 7: *A comparison of the cell tracking results between Amat et al.'s and ours. (A1) One time point from the tracking result using a sample data set provided by Amat et al. This time point contains several tracking issues. The inset magnifies one such issue that the green part is incorrectly tracked to the yellow cell due to an oversegmentation that separates the green from the orange. (A2) One time point from the tracking result using our workflow. The inset shows the correct tracking of the two cells. (B) Two time points of a magnified region from Amat et al.'s tracking result. The orange cell in the first time point is tracked to the green part of the cell in the second time point. Since the track is consistent, the issue cannot be detected using Amat et al.'s method without a detailed uncertainty analysis. (C) Two time points from the result by applying Amat et al.'s method to the 4D scan of zebrafish eye development. The colored cells are those could be tracked, only a portion of which are correct. All data and results are included in the supplementary materials and can be interactively viewed using FluoRender.*

many saddle points are in proximity, the process of getting trapped by one extremum would seem circuitous because of the mutual influence of subdomains. On the other hand, if a nonconvex objective function is topologically too simple, the result from the uncertainty footprint would seem uniform and confident. Our applications are focused on the problems in cell tracking, which have very distinguishable shapes but are also noisy. Therefore, it is possible for us to determine if segmentation or tracking results are correct. With an ever-increasing data size in scientific research, the simplicity of the uncertainty footprint also makes it suitable for real-time analysis. The computational overhead is nearly negligible. However, it requires doubling the storage for values from a previous iteration, if the evaluation of function $f_t(x)$ does not explicitly generate the differences. For memory-intensive applications, we need a subsampling method or a streamed processing.

6. Conclusion

In this paper, we present the uncertainty footprint, a method to quantify, visualize, and analyze uncertainties in an iterative process without explicit prior knowledge about its uncertainty. Our study shows a promising and potentially useful metric for objective and self-contained validation for a 4D cell tracking workflow. We also regard it a generally applicable method. In future work, we will concentrate our efforts on a comprehensive tracking system that can support a broad range of microscopy data. A self-contained validation method, such as the uncertainty footprint, will be used not only for evaluating results, but also for automated error corrections. Classification methods from the computer vision research will be used to enrich the knowledge from the uncertainty footprint analysis. The tracking system will also be able to accommodate multiple metrics so that more algorithms can be implemented to make automated and progressive improvements.

Acknowledgments

We would like to thank Kristen Kwan for providing the zebrafish eye development data, and Amat et al. for providing their cell tracking tool with the example data. This work was funded by the NIH P41 GM103545-18, NSF ACI-1339881, and NSF IIS-1162013.

References

[ALM*14] AMAT F., LEMON W., MOSSING D. P., MCDOLE K., WAN Y., BRANSON K., MYERS E. W., KELLER P. J.: Fast, accurate reconstruction of cell lineages from large-scale fluorescence microscopy data. *Nat. Methods 11*, 9 (2014), 951–958. 1, 5, 9

[ASE16] ATHAWALE T., SAKHAEE E., ENTEZARI A.: Isosurface visualization of data with nonparametric models for uncertainty. *IEEE Trans. Vis. Comput. Graphics 22*, 1 (2016), 777–786. 3

[BDC05] BONNEAU S., DAHAN M., COHEN L. D.: Single quantum dot tracking based on perceptual grouping using minimal paths in a spatiotemporal volume. *IEEE Trans. Image Process. 14*, 9 (2005), 1384–1395. 2

[BDH*15] BERGE C. S. Z., DECLARA D., HENNERSPERGER C., BAUST M., NAVAB N.: Real-time uncertainty visualization for b-mode ultrasound. In *IEEE Scientific Visualization Conference* (2015), pp. 33–40. 3

[BM93] BEUCHER S., MEYER F.: The morphological approach to segmentation: the watershed transformation. *Mathematical Morphology in Image Processing* (1993), 433–481. 5

[CGCR13] CHATTERJEE R., GHOSH M., CHOWDHURY A. S., RAY N.: Cell tracking in microscopic video using matching and linking of bipartite graphs. *Comput. Meth. Prog. Bio. 12*, 3 (2013), 422–431. 1, 3

[CSdC*14] CHENOUARD N., SMAL I., DE CHAUMONT F., MASKA M., SBALZARINI I. F., GONG Y., CARDINALE J., CARTHEL C., CORALUPPI S., WINTER M., COHEN A. R., GODINEZ W. J., ROHR K., KALAIDZIDIS Y., LIANG L., DUNCAN J., SHEN H., XU Y., MAGNUSSON K. E. G., JALDEN J., BLAU H. M., PAUL-GILLOTEAUX P., ROUDOT P., KERVRANN C., WAHARTE F., TINEVEZ J.-Y., SHORTE S. L., WILLEMSE J., CELLER K., VAN WEZEL G. P., DAN H.-W., TSAI Y.-S., DE SOLORZANO C. O., OLIVO-MARIN J.-C., MEIJERING E.: Objective comparison of particle tracking methods. *Nat. Methods 11*, 3 (2014), 281–289. 2

[DH72] DUDA R. O., HART P. E.: Use of the hough transform to detect lines and curves in pictures. *Comm. ACM 15* (1972), 11–15. 4

[Dij59] DIJKSTRA E. W.: A note on two problems in connexion with graphs. *Numerische Mathematik 1* (1959), 269–271. 3

[DLR77] DEMPSTER A. P., LAIRD N. M., RUBIN D. B.: Maximum likelihood from incomplete data via the em algorithm. *J. R. Stat. Soc. 39*, 1 (1977), 1–38. 4, 5

[Edm65] EDMONDS J.: Paths, trees, and flowers. *Canad. J. Math. 17* (1965), 449–467. 6

[Gib85] GIBBONS A.: *Algorithmic Graph Theory*. Cambridge University Press, 1985. 4

[GLE*11] GODINEZ W. J., LAMPE M., EILS R., MÃIJLLER B., ROHR K.: Tracking multiple particles in fluorescence microscopy images via probabilistic data association. In *IEEE International Symposium on Biomedical Imaging: From Nano to Macro* (2011), pp. 1925–1928. 2

[GOM04] GENOVESIO A., OLIVO-MARIN J.-C.: Split and merge data association filter for dense multi-target tracking. In *17th Interanational Conference on Pattern Recognition (ICPR)* (2004), pp. 677–680. 2

[GW07] GONZALEZ R. C., WOODS R. E.: *Digital Image Processing*. Pearson, 2007. 4, 6

[HBK*10] HUTH J., BUCHHOLZ M., KRAUS J. M., SCHMUCKER M., WICHERT G. V., KRNDIJA D., SEUFFERLEIN T., GRESS T. M., KESTLER H. A.: Significantly improved precision of cell migration analysis in time-lapse video microscopy through use of a fully automated tracking system. *BMC Cell Biol. 11*, 24 (2010). 2

[JD09] JAQAMAN K., DANUSER G.: Computational image analysis of cellular dynamics: a case study based on particle tracking. *Cold Spring Harbor Protocols* (2009), pdb.top65. 2

[JLM*08] JAQAMAN K., LOERKE D., METTLEN M., KUWATA H., GRINSTEIN S., SCHMID S. L., DANUSER G.: Robust single-particle tracking in live-cell time-lapse sequences. *Nat. Methods 5* (2008), 695–702. 2

[KKH02] KNISS J., KINDLEMANN G., HANSEN C.: Multidimensional transfer functions for interactive volume rendering. *IEEE Trans. Vis. Comput. Graphics 8*, 3 (2002), 207–285. 2

[KLB*13] KAN A., LECKIE C., BAILEY J., MARKHAM J., CHAKRAVORTY R.: Measures for ranking cell trackers without manual validation. *Pattern Recognition 46*, 11 (2013), 2849–2859. 9

[KOK*12] KWAN K. M., OTSUNA H., KIDOKORO H., CARNEY K. R., SAIJOH Y., CHIEN C.-B.: A complex choreography of cell movements shapes the vertebrate eye. *Development 139*, 2 (2012), 359–372. 1, 4

[Liu15] LIU B.: *Uncertainty Theory*. Springer-Verlag Berlin, 2015. 2

[MDS12] MEIJERING E., DZYUBACHYK O., SMAL I.: Methods for cell and particle tracking. *Methods Enzymol. 504* (2012), 183–200. 2

[MDSC09] MEIJERING E., DZYUBACHYK O., SMAL I., CAPPELLEN W. A. V.: Tracking in cell and developmental biology. *Semin. Cell Dev. Biol. 20* (2009), 894–902. 2

[MJGB15] MAGNUSSON K. E. G., JALDÉN J., GILBERT P. M., BLAU H. M.: Global linking of cell tracks using the viterbi algorithm. *IEEE Trans. Med. Imag. 34*, 4 (2015), 911–929. 2

[MJW*09] MOSIG A., JÄGER S., WANG C., NATH S., ERSOY I., PALANIAPPAN K.-P., CHEN S.-S.: Tracking cells in life cell imaging videos using topological alignments. *Algorithm Mol. Biol. 4*, 10 (2009). 3

[MSD06] MEIJERING E., SMAL I., DANUSER G.: Tracking in molecular bioimaging. *IEEE Signal Process. Mag. 23*, 3 (2006), 46–53. 2

[MUS*14] MAŠKA M., ULMAN V., SVOBODA D., MATULA P., MATULA P., EDERRA C., URBIOLA A., ESPAÑA T., VENKATESAN S., BALAK D. M., KARAS P., BOLCKOVÁ T., ŠTREITOVÁ M., CARTHEL C., CORALUPPI S., HARDER N., ROHR K., MAGNUSSON K. E. G., JALDÉN J., BLAU H. M., DZYUBACHYK O., KŘÍŽEK P., HAGEN G. M., PASTOR-ESCUREDO D., JIMENEZ-CARRETERO D., LEDESMA-CARBAYO M. J., MUÑOZ-BARRUTIA A., MEIJERING E., KOZUBEK M., , DE SOLORZANO C. O.: A benchmark for comparison of cell tracking algorithms. *Bioinformatics 30*, 11 (2014), 1609–1617. 2

[Ots79] OTSU N.: A threshold selection method from gray-level histograms. *IEEE Trans. Syst., Man, Cybern., Syst. 9*, 1 (1979), 62–66. 5

[Rud76] RUDIN W.: *Principles of mathematical Analysis*. McGraw-Hill, 1976. 2

[Sax08] SAXTON M. J.: Single-particle tracking: connecting the dots. *Nat. Methods 5* (2008), 671–672. 2

[Sch02] SCHRIJVER A.: On the history of the transportation and maximum flow problems. *Mathematical Programming 91*, 3 (2002), 437–445. 3

[SK05] SBALZARINI I. F., KOUMOUTSAKOS P.: Feature point tracking and trajectory analysis for video imaging in cell biology. *J. Struct. Biol. 151* (2005), 182–195. 2

[Sør05] SØRENSEN H. K.: Exceptions and counterexamples: understanding abel's comment on cauchy's theorem. *Historia Mathematica 32*, 4 (2005), 453–480. 2

[TK06] TARDOS E., KLEINBERG J.: *Algorithm Design*. Pearson Education, 2006. 2

[vPOBB*10] VAN PELT R., OLIVÁN BESCÓS J., BREEUWER M., CLOUGH R., GRÖLLER M., TER HAAR ROMENIJ B., VILANOVA A.: Exploration of 4d mri blood flow using stylistic visualization. *IEEE Trans. Vis. Comput. Graphics 16*, 6 (2010), 1339–1347. 3

[vPOBB*11] VAN PELT R., OLIVÁN BESCÓS J., BREEUWER M., CLOUGH R., GRÖLLER M., TER HAAR ROMENIJ B., VILANOVA A.: Interactive virtual probing of 4d mri blood-flow. *IEEE Trans. Vis. Comput. Graphics 17*, 12 (2011), 2153–2162. 3

[WOCH09] WAN Y., OTSUNA H., CHIEN C.-B., HANSEN C.: An interactive visualization tool for multi-channel confocal microscopy data in neurobiology research. *IEEE Trans. Vis. Comput. Graphics 15*, 6 (2009), 1489–1496. 7

[WOCH12] WAN Y., OTSUNA H., CHIEN C.-B., HANSEN C.: Fluorender: an application of 2d image space methods for 3d and 4d confocal microscopy data visualization in neurobiology research. In *IEEE Pacific Visualization Symposium* (2012), pp. 201–208. 7

[WOH13] WAN Y., OTSUNA H., HANSEN C.: Synthetic brainbows. *Comput. Graph. Forum 32*, 3 (2013), 471–480. 7

[Wu83] WU C. F. J.: On the convergence properties of the em algorithm. *Ann. Stat. 11*, 1 (1983), 95–103. 5