Automatic Summarization of Changes in Biological Image Sequences Using Algorithmic Information Theory

Andrew R. Cohen, *Member*, *IEEE*, Christopher S. Bjornsson, Sally Temple,
Gary Banker, and Badrinath Roysam, *Senior Member*, *IEEE*

Abstract—An algorithmic information-theoretic method is presented for object-level summarization of meaningful changes in image sequences. Object extraction and tracking data are represented as an attributed tracking graph (ATG). Time courses of object states are compared using an adaptive information distance measure, aided by a closed-form multidimensional quantization. The notion of meaningful summarization is captured by using the gap statistic to estimate the randomness deficiency from algorithmic statistics. The summary is the clustering result and feature subset that maximize the gap statistic. This approach was validated on four bioimaging applications: 1) It was applied to a synthetic data set containing two populations of cells differing in the rate of growth, for which it correctly identified the two populations and the single feature out of 23 that separated them; 2) it was applied to 59 movies of three types of neuroprosthetic devices being inserted in the brain tissue at three speeds each, for which it correctly identified insertion speed as the primary factor affecting tissue strain; 3) when applied to movies of cultured neural progenitor cells, it correctly distinguished neurons from progenitors without requiring the use of a fixative stain; and 4) when analyzing intracellular molecular transport in cultured neurons undergoing axon specification, it automatically confirmed the role of kinesins in axon specification.

Index Terms—Image sequence analysis, algorithmic information theory, algorithmic statistics, information distance, gap statistic, clustering.

1 Introduction

We are given a set of S image sequence(s), denoted

$$\{\{I_t^s(x)\}_{t=1}^{T_s}\}_{s=1}^{S} = \{I_1^1(x), \dots, I_{T_1}^1(x), \dots, I_1^S(x), \dots, I_{T_S}^S(x)\},$$

where x is the spatial coordinate of a 2D/3D pixel and t represents a temporal sampling point. The lengths of the image sequence(s) T_S may vary. We are interested in unsupervised algorithms that can compute a *concise* and *meaningful* summary of the changes. These two terms will be defined precisely in terms of algorithmic information theory [1], [2], [3] and algorithmic statistics [4], [5], [6].

The goal of developing automated object-level algorithms that can analyze multiple forms of time-lapse image data is ambitious. Our strategy is to consider modular approaches that isolate the unavoidably application-specific aspects (prior work), while allowing the development of general-purpose tools for the remaining aspects (this paper). Application-specific aspects include the imaging physics and geometry that determine low-level segmentation and change detection algorithms [7], object models and object extraction algorithms, models of object behaviors, and algorithms for tracking objects over time. Fortunately, usable (albeit not perfect) object extraction and tracking algorithms are available for several interesting applications, especially biological image sequences of interest to us [8], [9], [10], [11], [12], [13], [14]. We focus on biological image sequences to illustrate our methods in this paper, although the methods are more general.

We propose algorithmic information theory and algorithmic statistics as a basis for summarizing the object extraction and tracking results in a general manner. These are nonprobabilistic approaches to information theory and statistics that can quantify relationships between individual digital objects (algorithmic information theory) and between a specific digital object and a model (algorithmic statistics) more precisely than is possible using classical (probabilistic) information theory and statistics. Unlike Shannon's entropy, which describes an ensemble of objects [15], algorithmic information theory is concerned with absolute information content in individual objects. Importantly, these approaches allow us to capture the notion of a *concise and meaningful summary* of the changes within and across image sequences.

The absolutely most concise description of a digitally represented object (bit string) is given by its Kolmogorov complexity [16]. For our purposes, the objects and tracking results can all be represented ultimately as bit strings. The

- *A.R. Cohen is with the University of Wisconsin, PO Box 784, Milwaukee, WI 53201. E-mail: cohena@uwm.edu.*
- *C. Bjornsson and B. Roysam are with Rensselaer Polytechnic Institute, 110 8th Street, Troy, NY 12180-3590. E-mail: bjornc@rpi.edu, roysam@ecse.rpi.edu.*
- *S. Temple is with the New York Neural Stem Cell Institute, One Discovery Drive, Rensselaer, NY 12144. E-mail: sallytemple@nynsci.org.*
- *G. Banker is with the Oregon Health and Science University, 3181 SW Sam Jackson Park Road, L606, Portland, OR 97239. E-mail: bankerg@ohsu.edu.*

Manuscript received 14 Sept. 2007; revised 13 Feb. 2008; accepted 28 May 2008; published online 11 June 2008.
Recommended for acceptance by S. Sclaroff.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2007-09-0602.
Digital Object Identifier no. 10.1109/TPAMI.2008.162.

0162-8828/09/$25.00 © 2009 IEEE Published by the IEEE Computer Society

Kolmogorov complexity of a string x, denoted $K(x)$, is the length $l(p)$ of the shortest binary program p that runs on a universal computing device φ (a universal Turing machine) and produces the string x as output, i.e., $\varphi(p) = x$. Mathematically, this is stated as follows [16]:

$$K(x) = \min_{\{p \mid \phi(p)=x\}} l(p). \tag{1}$$

Intuitively, the above equation describes a "competitive selection" of the shortest program (algorithmic description), denoted p^*, from an unbounded set of competing programs $\{p_0, p_1, \ldots\}$, each capable of producing the desired output x. The unbounded nature of this competition guarantees that the winning program can most efficiently model and exploit *any and all* structural regularities in the specific string x. Unfortunately, it also implies the lack of a guarantee that this competition will achieve closure every time. In other words, the Kolmogorov complexity is uncomputable in the general case (as defined by Li and Vitanyi [16], it is just upper semicomputable). The uncomputability of Kolmogorov complexity has motivated several authors to seek useful approximations. Indeed, the present work is inspired and enabled by the trilogy of papers published recently by Vitanyi et al. [1], [2], [3] in which they describe practical approaches to approximate Kolmogorov complexity and the related notion of algorithmic information distance using common data compression algorithms such as bzip. In particular, the *normalized compression distance* (NCD) measure has been shown to be a versatile and broadly applicable tool for pattern analysis [17]. Problem formulations based on NCD can be very general, parameter free, robust to noise, and independent of applications and data formats [2], [17], [18].[1] They can also overcome the main practical limitation of Minimum Description Length (MDL) [19], [20] based techniques for approximating Kolmogorov complexity—the need for "tightly tuned" application-specific models and data representations.

The notion of meaningful information has been defined in the field of algorithmic statistics in terms of the notion of *randomness deficiency*. In this paper, we present practical methods to approximate this concept in the context of image sequence analysis.

2 Related Literature

This paper draws upon techniques from the fields of object detection, multitarget tracking, symbolic representation, object-level change detection, algorithmic information theory, and algorithmic statistics.

The idea of object-level change description is not new. An extensive body of literature exists in the area of video surveillance. In the biological arena, Al-Kofahi et al. described changes in cultured neurons [10] and progenitor-cell cultures [9]. Narasimha-Iyer et al. have described diverse changes in human retinas imaged over time [12]. Bao et al. have described the development of worm embryos [21]. Roussel et al. have described movement patterns of worm populations [13]. While this list is by no means comprehensive, it illustrates an important point—most systems described to date are specialized to their respective application domains. In biological images, many objects of interest are blob-like (e.g., cells) or tube-like (e.g., neurites, vessels, etc.). For blob detection, we use mathematical morphology [22] in combination with a fast radial transform [23]. For tube detection, we use a method by Tyrrell et al. [14] that models tubular structures using superellipsoids. For handling less well-defined objects, we resort to generic features such as interest/key points [24].

Multiple-hypothesis tracking [25] is commonly used to establish temporal correspondences between objects over time. There are two main aspects to this approach: prediction and assignment. In the assignment problem, hypotheses are formed by associating sensor measurements with predictions. Hypotheses can include false alarms or new tracks and may include a priori knowledge. The Kuhn-Munkres or Hungarian algorithm [26] was one of the early solutions to the assignment problem. Our tracking implementation is based on the work of Al-Kofahi et al. [9]. The Viterbi algorithm [27] provides a maximum a posteriori estimate of the optimal set of states in a sequence of multiple assignment problems. Pitié et al. [28] use this algorithm for offline multiple-object tracking. Their approach, using the Viterbi trellis for tracking, inspired our method of generating a summary from an attributed tracking graph (ATG). Pradeep and Whelan [29] also use a Viterbi algorithm for tracking facial features through an image sequence. The choice of object extraction and tracking algorithms impacts our methodology only in the accuracy and variety of the extracted features and the accuracy of the estimated tracking assignments.

Stauffer and Grimson [30] presented a system that establishes patterns of activity from tracking results using a codebook of representations to classify higher level behaviors from tracking. The goal of their system is to classify an object given one or more observations. Katz et al. [31] build on a similar approach and integrate with natural language processing. Héas and Datcu [32] present another similar approach, using clustering on spatiotemporal data to classify behaviors. Medioni et al. [33] proposed a system that generates scenarios from tracking information. They use an attributed graph structure to represent object, feature, and tracking information. Their goal is to generate application-specific AI-driven scenarios, e.g., "the car is avoiding the checkpoint." Our approach builds upon the above body of pioneering efforts.

The information-theoretic background to this work is the classic "Vitanyi trilogy" composed of three related papers describing an absolute information-theoretic distance between bit strings, its practical approximation, and applications to real-world data [1], [2], [3]. Unlike the Shannon entropy notion, which applies to an ensemble of objects [15], algorithmic information theory is concerned with absolute information content in individual objects. A method of clustering based on classical (probabilistic) information theory can be found in [34]. A rigorous treatment of algorithmic information theory can be found in the book by Li and Vitanyi [16]; a more accessible treatment can be found

[1] In theory, the choice of compression algorithm and/or its settings are adjustable parameters. However, we (and other authors [17]) do not vary these and simply use the bzip compressor with the default (maximum memory) settings.

in Chaitin's book [35]. Our work is inspired by effective practical approximations and distance measures described by Vitanyi et al. [1], [2], [3] to the otherwise uncomputable notions in algorithmic information theory. Specifically, the NCD measure approximates the idealized normalized information distance (NID) using generic data compression algorithms. In [17], Keogh et al. apply the NCD to classic data mining problems including anomaly detection, classification, and clustering. Recently, Cebrián et al. [18] have shown the NCD to be robust to noise.

Our enthusiasm for the NCD measure is based on its impressive performance in diverse pattern analysis applications studied by other authors [17] and in our work. Cilibrasi and Vitanyi [2] demonstrated the effectiveness of NCD across diverse applications in genomics, virology, languages, literature, music, handwritten digits, and astronomy. In genomics, the mitochondrial genomes of 24 mammalian species, each around 17,000 bases, were downloaded from the GenBank database on the World Wide Web. Comparing these genomes using the NCD correctly reproduced the evolutionary tree for these 24 species. The analysis of the mitochondrial genome of eight different fungi supported the current hypothesis among domain experts that classifies the fungi into two separate groups. In virology, the sequenced genome of the SARS Virus was compared to potential similar viruses, establishing relationships among the viruses very similar to the definitive tree based on medical-macrobiogenomics analysis, appearing later in the *New England Journal of Medicine*. In the field of language classification, the analysis of a single document, "The Universal Declaration of Human Rights," in 52 languages correctly grouped the languages by Native-American, Native-African, and Native-European origin. In analyzing text from Russian literature from five different authors and three or four texts per author, the NCD in conjunction with a hierarchical clustering correctly grouped the texts by author. Interestingly, clustering English translations produced errors; it was found that the clustering was based partly on the translator. In other experiments, authors are correctly separated by gender and period. In music analysis, using a custom quantized MIDI representation, the analysis using the NCD correctly distinguished classical versus rock versus jazz by genre. In a separate experiment, they were not able to reliably classify classical pieces by composer; this may have reflected a shortcoming of the MIDI representation used or it may be indicative of some previously unknown underlying similarity of the works. For handwriting recognition, analyzing handwritten digits consisting of 30 128×128 binary images of the numbers "4," "5," and "6" resulted in a classification accuracy of 93 percent. The current state of the art for this problem is in the upper 90 percent level. Finally, in the field of astronomy, analyzing time series data from 12 different galactic objects comprising four categories of object type corresponded precisely with accepted classification of these objects. In [17], Keogh et al. analyzed 36 1D time series from the UCR Time Series Archive using the NCD and 51 other distance measures. Because most distance measures, with the exception of longest common subsequence (LCSS) and dynamic time warping (DTW), do not support comparing time series of different lengths, the time series were all truncated to be the same length. For all of the time series data, the NCD was found to outperform every other distance measure that was analyzed on each of the 36 time series. Finally, in [18], symmetric channel noise was used to corrupt individual bytes of genomic, text, music, and image files. It was found that, even in the presence of very high quantities of noise, the NCD continues to perform robustly. The NCD was applied to the problem of detecting anomalies in image sequence data in [36]. In [37], the NCD was applied to the problem of image pair registration. This work obtained better results from a method based on Shannon entropy than from the NCD; it is possible that the use of quantization with the NCD, as described in Section 8, might have improved the performance of the NCD for this application.

A second trilogy of papers on the subject of algorithmic statistics [4], [5], [6] has inspired us to develop practical methods to capture the notion of *meaningful information*. Specifically, we describe methods for using the gap statistic introduced by Tibshirani et al. [38] to estimate the randomness deficiency, a notion that underlies the concept of meaningful information.

MDL is related to algorithmic information theory [19], [20]. Leclerc [39] uses MDL to construct the shortest description of an image. The connection between MDL and algorithmic information theory is explored in more detail in [40]. Yi et al. [41] have used a greedy descent to minimize a code length subject to a distortion, yielding a lossy approach to MDL, and applied the technique to segmenting multivariate data using a mixture of Gaussians model.

Portions of our work have been inspired by Symbolic Aggregate Approximation (SAX) [17], [42], a technique used in the data mining community to assign a symbolic value to numerical data. It assumes a Gaussian distribution of the numerical values and generates a "codebook" using the equiprobable regions of the Gaussian distribution. Megalooikonomou et al. [43] present an alternative to SAX based on a multiresolution vector quantizer (VQ). A comprehensive description of quantization theory can be found in [44]. Current implementations of SAX use a lookup table to find breakpoints (up to a maximum of 10) in 1D data. As part of the automatic quantization presented in this paper, we describe a closed-form solution for finding an arbitrary number of breakpoints for data of any dimension.

3 Overview of Method

Our problem formulation is based on the relative Kolmogorov complexity (captured in the form of an algorithmic information-theoretic distance) that is capable of quantifying *any and all differences* between digital objects, rather than the Kolmogorov complexity of a single digital object. The main advantage of the relative-complexity-based formulation is the existence of effective yet practical compression-based approximations to the algorithmic information distance. This enables our method to be applied to a broad class of applications in a manner that is parameter free and not requiring training data. The only requirement is the availability of object segmentation and tracking algorithms for the chosen application.

3.1 Object Extraction and Tracking

Starting with the given set of image sequences $\{\{I_t^s(x)\}_{t=1}^{T_s}\}_{s=1}^{S}$, we run application-specific algorithms to extract a set of objects and a corresponding vector of features/attributes in each image. We also use application-specific tracking algorithms to estimate temporal correspondences between objects over time. Then, we obtain a set of tracked objects denoted $\{\mathrm{O}_i^t\}$. We use the symbol Ω_i to denote the time course of feature values for an individual object. Each object time course has a feature vector of dimension $T_i \times F_i$, where T_i is the number of time samples in Ω_i and F_i is the number of features in Ω_i. The number of time samples can be different for each object. In principle, the number of features can also be different, although, for all data sets analyzed to date by us, F_i is equal for all objects. For convenience, we abbreviate F_i as F.

3.2 Attributed Tracking Graph (ATG)

We construct a data structure named the ATG, bringing together objects extracted from the image sequence(s), together with their time course of feature values and their spatiotemporal associations. A node in the ATG represents an object at a particular time instance. Each node has a vector of attributes representing object features (some of which may be specified in relation to other objects). Links between nodes represent temporal associations estimated by the tracking algorithm. If our ATG consists of M objects (extracted from one or more image sequences), we use the symbol Ω to denote the set of time courses of feature values for all M objects $\{\Omega_1, \ldots, \Omega_M\}$. The rest of our analysis is conducted entirely on the ATG. Specifically, we use an information-theoretic distance measure, aided by an adaptive multidimensional quantization algorithm, to compare object time courses in a general manner. Typically, the time courses are of different lengths within an application and the features can vary across applications. This results in a distance matrix that is analyzed further as follows.

3.3 Analyzing the Distance Matrix

We analyze the distance matrix to estimate jointly an optimal (in the sense to be defined next) combination of the following parameters:

1. a subset of relevant features $\hat{f} \subseteq \{1, \ldots, F\}$,
2. the number of clusters among object time courses $\hat{k}$,
3. cluster assignments $[l_1, \ldots, l_M]$ specifying the membership of each object time course to a group, and
4. a quantization level $\hat{N}$.

We seek the combination of the above parameters that maximize the meaningful information captured by the clustering as measured by the randomness deficiency. Randomness deficiency is a concept from algorithmic statistics that quantifies how well a model captures the meaningful information in a specific digital object. Algorithmic statistics has until now been a theoretical concept quantifying the relationship between an individual digital object and a model, based on Kolmogorov complexity. In this paper, we show that the gap statistic—a concept developed in the statistics community [38]—can be used to approximate the randomness deficiency [4] for clustering models. With this in mind, our automatic summarization is designed to maximize a measure based on the gap statistic. Fig. 1 illustrates a sample image sequence (Fig. 1a), three different views of the ATG (Figs. 1b, 1c, and 1d), the features stored at a single node (Fig. 1e), and the ATG colored with summary results (Fig. 1f). The following sections elaborate upon the above ideas.

4 Algorithmic Information Distance

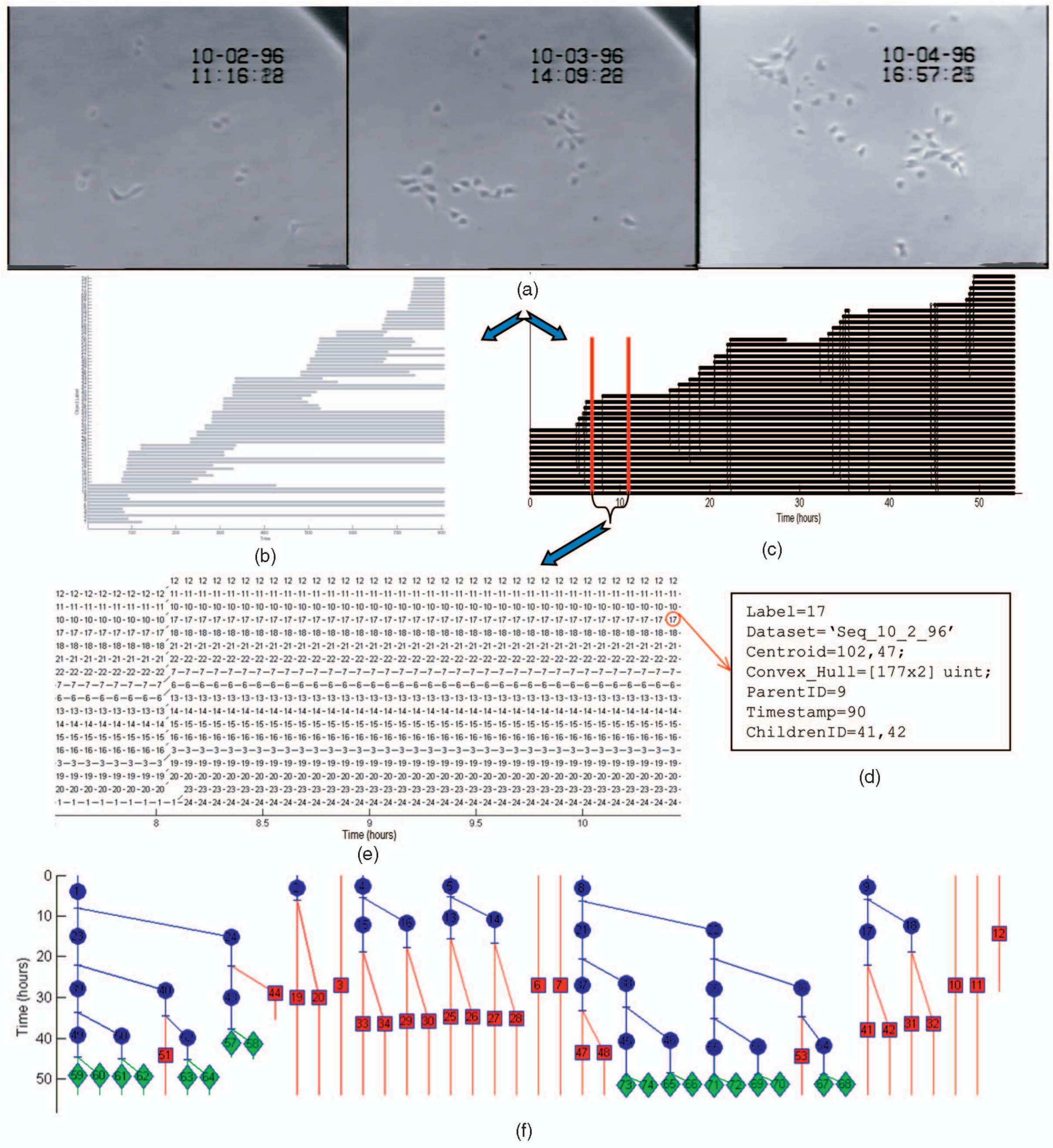
As noted above, our formulation is based on the relative Kolmogorov complexity between digitally represented objects (strings), rather than the Kolmogorov complexity of individual objects. Of all of the distance measures that can be defined between a pair of objects, the universal measure based on algorithmic information theory has been shown to be the lowest up to a constant precision [1]. Bennett et al. [1] define the "absolute information distance" between two strings x and y, denoted $E(x,y)$, as $E(x,y) = \max\{K(x|y), K(y|x)\}$, where $K(x|y)$ is the conditional Kolmogorov complexity of a string x relative to string y defined as the length of the shortest program to compute x if string y is provided to the universal computer as an auxiliary input. This program must efficiently take advantage of *any and all* regularities or redundancies, regardless of the amount of computation required. Although $K(x)$ and $K(x|y)$ are defined in terms of a particular universal computer, they are machine independent up to an additive constant by way of Church's thesis [45]. $E(x,y)$ is the length of the shortest binary program that computes y from x, as well as x from y, while remaining unchanged itself, to within an additive logarithmic constant $O(\log \max\{K(y|x), K(x|y)\})$. Importantly, the lengths of the two strings need not be the same. Bennett et al. [1] show that $E(x,y)$ satisfies metric properties up to an additive fixed constant.

Although many distances are innately absolute, the image analysis problems of interest to us only require a relative or normalized distance metric. This requirement is met by the universal similarity metric defined by Li et al. [3], known as the "NID," and given as follows:

$$NID(x,y) = \frac{\max\{K(x|y), K(y|x)\}}{\max\{K(x), K(y)\}}. \qquad (2)$$

The NID is symmetric and assumes a value of 0 when the two objects are maximally similar or identical and a value of 1 when they are maximally dissimilar. It is approximately a metric—it satisfies the identity axiom (to within $O(1/K(x))$), the triangle inequality (to within $O(1/\max\{K(x), K(y), K(z)\})$), and the symmetry axiom. The reader is referred to [3] for a detailed justification of NID, especially the choice of the correct denominator term. Finally, NID has been shown to be "universal" in the sense that $NID(x,y)$ is at least as small as any normalized distance between objects x and y in the class of upper semicomputable normalized distances [1].

By itself, $NID(x,y)$ is an interesting theoretical concept with little practical value due to the noncomputability of its



```
Label=17
Dataset='Seq_10_2_96'
Centroid=102,47;
Convex_Hull=[177x2] uint;
ParentID=9
Timestamp=90
ChildrenID=41,42
```

Fig. 1. Illustrating the methodology for analyzing a single 800-frame video of cultured neural progenitor cells. (a) Frames 1, 400, and 800. (b) A rendering of the ATG showing object time courses for the 74 cells extracted from the image sequence with each of the 20,000+ vertices representing the state of an object at a single time instance and links established by the tracking algorithm. (c) A different rendering of the ATG using a priori knowledge of the parent-child (transmitotic) relationship established by the tracking algorithm. (d) Enlarged view of the ATG section indicated in red in (c). (e) Attributes at one of the nodes in the ATG. (f) ATG colored according to summary. Note that, although our method is very powerful at finding any differences among the cells, it does not tell us what the differences are. In this case, subsequent molecular marker studies confirmed that cells labeled in blue were neuroprogenitors and cells labeled in red were differentiated neurons. Cells labeled in green were short lived because of either apoptosis or the termination of the video recording.

constituent Kolmogorov complexity terms. We can identify two basic approaches to overcome this limitation. The first, which we term "design based," is to design application-specific strategies to arrive at (or approximate) the shortest (winning) program p^*. In other words, it is possible, in principle, to identify the essential characteristics of a winning program and carefully engineer such a program. The second approach, which we term "compression based," is to use generic algorithms that can capture patterns in bit strings. In [2], Cilibrasi and Vitányi presented a method for approximating the NID using the NCD, a similarity metric between strings that is computed using "off-the-shelf" lossless compression programs such as zip, gzip, bzip2, etc. Compression algorithms excel at identifying and exploiting patterns in the data and the NCD measure exploits this ability. Intuitively, strings with similar patterns will compress better together versus when compressed separately. The NCD is computed as follows: Let $C(x)$

denote the size of the compressed version of string x and $C(x;y)$ be the size of the compressed version of the concatenation of x and y. Then,

$$NCD(x,y) = \frac{C(x;y) - \min(C(x),C(y))}{\max(C(x),C(y))}. \tag{3}$$

There are no parameters needed to compute the NCD, except for the choice of compression algorithm and its settings. As shown by Vitanyi et al., the choice of compression algorithm has a negligible impact on the final analysis [2]. In our work, we use the `bzip2` algorithm with default settings. The strings being compared do not even have to be of the same size or dimension. As shown in [3], the NCD is a reasonable approximation to NID in that it is a nonnegative number in the range 0 and $(1+\varepsilon)$, where the ε arises from imperfections in real-world compression algorithms and is typically less than 0.1. It is also approximately a metric and its deviations from metric properties depend upon the performance of the compression algorithm. Finally, `bzip2` uses a block-based encoding that results in nearly symmetrical compression, $C(x;y) \sim C(y;x)$ [3]. The apparent simplicity of the NCD belies its power. The value of the NCD lies in its multiple advantages and demonstrated practical effectiveness [17]. First, its computation does not require any specific background knowledge about the data. Second, it can be computed meaningfully when x and y are of different lengths. Practically speaking, the NCD can be computed for any set of image sequences in much the same manner, regardless of application details. Different compression algorithms may capture more or less of the structure in a string and exhibit different levels of compression performance. Happily, the normalization in (3) above ensures that differences in NCD values resulting from the choice of different compression algorithms (or settings) are modest. In the present work, we developed two ways to further enhance the NCD measure. The first enhancement (described below) improves its handling of data that is not a time series. The second enhancement (described in Section 8) improves its handling of numerical time series data using an automatic multidimensional quantization algorithm, denoted $Q_N(\cdot)$, to achieve multiprecision analysis. Here, N denotes the number of quantization levels. The above improvements, in combination with the notion of meaningful summarization provided by our use of gap spectral clustering (described in Section 6), provide significant advantages compared to the prior applications that simply use the NCD in conjunction with hierarchical clustering.

Our first enhancement to the NCD is intended to address the fact that some of the data in the ATG are not time series of measurements, for example, scalars. For these quantities, there are an insufficient number of bits for quantization and/or compression algorithms to accurately assess similarities. To overcome this practical limitation, we use the normalized euclidean distance (NED) that was defined and shown to be a metric in [46]. With this in mind, we propose the following hybrid distance measure that represents a reasonable first approximation to the true information distance: Given two inputs, we examine their format and choose to compute the NCD for time series data and the NED for others (usually, scalars). We term this measure the *normalized adaptive information distance* (NAID), defined as

$$NAID(x,y) = \begin{cases} NED(x,y), \text{ if } x,y \text{ are not time-series \& } \dim(x)=\dim(y), \\ NCD(x,y) \text{ otherwise,} \end{cases} \tag{4a}$$

where

$$NED(x,y) = \frac{\|x-y\|}{\|x\| + \|y\| + eps}, \tag{4b}$$

where $eps = 2^{-52}$, the floating-point resolution, and $\|x\|$ denotes the L_2 norm of vector x. The $NAID$ measure is used throughout our work, with the additional enhancement of multidimensional quantization discussed in Section 8.

5 Comparing Object Time Courses

The information distance measure defined above provides us with the ability to compare object time courses Ω_i. In this regard, the need arises to perform feature selection to choose a subset of only the relevant features $f \subseteq \{1..F\}$ rather than the full set. We use the symbol Ω^f to denote the time course of this subset of features for all M objects. In practice, we also adaptively quantize the data using the function $Q_N(\cdot)$ described in Section 8 below. The resulting distance matrix is given by

$$NAID\big(Q_N(\Omega^f)\big) = \begin{bmatrix} NAID\big(Q_N(\Omega_1^f), Q_N(\Omega_1^f)\big) & \cdots & NAID\big(Q_N(\Omega_1^f), Q_N(\Omega_M^f)\big) \\ \vdots & \ddots & \vdots \\ NAID\big(Q_N(\Omega_M^f), Q_N(\Omega_1^f)\big) & \cdots & NAID\big(Q_N(\Omega_M^f), Q_N(\Omega_M^f)\big) \end{bmatrix}. \tag{5}$$

Unsupervised analysis of the above distance matrix provides the basis for automatic summarization of the events occurring in the set of underlying image sequences. We propose clustering as a means for unsupervised analysis. Specifically, the spectral clustering algorithm by Ng et al. [47], in conjunction with the Gap statistic [38], offers important advantages for this work and is discussed further in the next section.

6 The Gap Statistic

Tibshirani et al. [38], proposed the gap statistic as an effective tool for automatically estimating the number of clusters in the data that outperforms several alternative methods. It offers several advantages for our work. First, unlike statistical-test-based measures such as g-means [48], it works well with a small number of data points. Second, it is able to handle the case where the data is best represented by a single cluster. Third, it is very easy to implement. Most importantly, however, it can be adapted to estimate the randomness deficiency, as discussed in Section 7. Overall, we have found the gap statistic to be a broadly applicable tool in our work. The gap statistic compares the clustering of the data to an ensemble of clustering results of random data generated by a uniform distribution. The optimal number of clusters is defined as the smallest number of clusters for which adding an additional cluster does not improve the magnitude of the gap value defined below.

Specifically, define D_r to be the sum of the distances between points in cluster r:

$$D_r = \sum_{i,j \in C_r} d_{i,j}. \tag{6}$$

We define W_k as the intracluster distance summed across all k clusters, where n_r is the number of points in cluster r:

$$W_k = \sum_{r=1}^{k} \frac{1}{2n_r} D_r. \tag{7}$$

The gap statistic can now be calculated as the difference between the intracluster distances of our data and the intracluster distances of B randomly generated uniformly distributed data sets of the same dimension as our data:

$$Gap(k) = \Gamma_k = \frac{1}{B} \sum_{b=1}^{B} \log(W_{kb}) - \log(W_k). \tag{8}$$

Here, W_{kb} is calculated as in (7) for each of the B randomly generated uniformly distributed data sets. Given the standard deviation σ_k of our B randomly generated uniformly distributed data sets, we define the following factor s_k to account for the simulation error:

$$s_k = \sigma_k \sqrt{1 + \frac{1}{B}}. \tag{9}$$

Finally, we choose k as the smallest value of k for which

$$Gap(k) \geq Gap(k+1) - s_{k+1}. \tag{10}$$

Typically, when the data is not clearly separated, the gap plot will exhibit multiple local maxima. We have observed that the use of adaptive quantization (described in Section 8 below) also reduces the appearance of local maxima by eliminating spurious similarities between continuous-valued time series data points. The magnitude of the gap statistic Γ_k for a given number of clusters k is a measure of how much difference in intracluster dispersion we see in our data versus the randomly generated data and is used here as a criterion function for selecting the clustering that captures the most similarities in the data. The next section describes another use of the gap statistic.

7 Using the Gap Statistic to Estimate Randomness Deficiency

The concept of randomness deficiency captures the notion of a meaningful summary using concepts from algorithmic statistics [4], [5], [6], a branch of algorithmic information theory that quantifies the relationship between an individual digital object, referred to as the *data* denoted x, and a *model* of the data, denoted M. In this approach, the data is represented using a two-part code. The first part describes the model itself. The second part describes how to generate the data from the model. The two-part code can be specified for any class of models. Now, the most compact representation of data x with respect to model M is given by the program whose length is the Kolmogorov complexity of the two-part code, i.e., $K(M) + K(x|M)$. When the model has captured all possible regularities in the data, the first term in the two-part code represents the *meaningful information* and the second term describes only the random portion of the data and is referred to as the *data-to-model code*. As an example, consider the case where we wish to model x using a finite set $S = \{x_1, \ldots, x_n\}$. The most compact representation of x with respect to S is $K(S) + K(x|S)$. When there are no regularities available to shorten our description of x given S, $K(x|S) = \log|S|$ [5]. In this case, x can be considered a typical member of set S and $\log|S|$ is the data-to-model code for a finite set model—the number of bits required to describe an object x when model S has captured all meaningful information.

The *randomness deficiency* $\delta(x|M)$ of data x with respect to model M is defined as the difference between the complexity of describing an object for which the model has captured all meaningful information and the complexity of describing x given the model. For a finite set model,

$$\delta(x|S) = \log|S| - K(x|S). \tag{11}$$

Randomness deficiency measures how well the model captures the meaningful information in the data. When randomness deficiency is ≈ 0, the model has captured all regularities or meaningful information in the data, and the data can be considered "typical" for the model. For a more detailed discussion of typicality, see [4].

A variety of models have been described in the literature, including Turing machines, probability mass functions, finite set models, and recursive functions [6]. For this work, we present a two-part code representation using cluster models. This representation is based on the algorithmic information-theoretic distance measure presented in Vitanyi's trilogy on algorithmic statistics [4], [5], [6]. This enables our implementation to use the powerful approximation of the NCD rather than constructing application-specific MDL representations, allowing our method to be formulated in an application-independent manner.

We proceed to show that the gap statistic can be adapted and used in combination with the NAID to estimate the randomness deficiency for clustering models. As noted earlier, the gap statistic [38] compares the clustering of the data to an ensemble of clustering results of random data generated by a uniform distribution (8). As noted by Vitanyi, when using a finite set to model our data, there should be "no simple special properties" [6] that differentiate our data from any of the elements of the set. If there were such properties, they could be used to form a more compact representation than our original finite set model. Now, consider the case of a clustering model. It is true that any two data points belonging to the same cluster are more similar than any two data points in different clusters. This is the type of "special property" that would make a finite set model inappropriate when the data naturally forms clusters.

To construct the data-to-model code for a clustering model, we consider the case where the clustering has captured all regularities or meaningful information in the data. In this case, all data in the cluster are equally well represented by the cluster centroid. Any differences between data in the same cluster or between data and the cluster centroid would by definition be purely random. Since our clustering is based on an algorithmic information-theoretic distance measure, rather than a measure such as the euclidean, we cannot compute the centroid of the points in a cluster for use as a cluster representative. Instead, a "representative point" x^* within the cluster is chosen as described below. In principle, any point within the cluster can be used as the representative since the differences between the points of a cluster are known to be purely random. By the symmetry of algorithmic information [3],

$$K(x) + K(y|x) = K(y) + K(x|y) + c, \tag{12}$$

where c is a constant independent of x and y; any point within the cluster can be chosen as x^* without appreciably increasing the amount of information required to specify the other points. Points that belong to that cluster are then specified using a program of length $K(x|x^*)$ bits to compute x^* from x. The two-part code length for a clustering model can now be written as $K(x^*) + K(x|x^*)$. Next, we can write the data-to-model code for an object d for which the clustering model has captured all regular information as follows:

$$K_{DM}(d|M) = K(x^*) + K(d|x^*). \tag{13}$$

If we choose x^* as the point in cluster C_r with maximal complexity, i.e.,

$$K(x^*) \geq K(x) \quad \forall x \in C_r, \tag{14}$$

the maximal complexity of x^* together with the definition of the NID (2) and the symmetry of information (12) allow the NID to be reduced to

$$NID(x, x^*) = \frac{K(x^*|x)}{K(x^*)}. \tag{15}$$

Because points x and x^* are in the same cluster, there can be no simple special properties that differentiate any of the objects in the cluster:

$$K(x_i|x_j) \cong K(x_j|x_i) \quad \forall x_i, x_j \in C_r. \tag{16}$$

This allows us to rewrite (15) as follows:

$$NID(x, x^*) = \frac{K(x|x^*)}{K(x^*)}. \tag{17}$$

Now, the randomness deficiency for a clustering model can be defined as

$$\begin{aligned}\delta(x) &= K(x^*) + K(d|x^*) - (K(x^*) + K(x|x^*)) \\ &= K(d|x^*) - K(x|x^*).\end{aligned} \tag{18}$$

Substituting (17), we get

$$\delta(x) = NID(d, x^*)K(x^*) - NID(x, x^*)K(x^*). \tag{19}$$

Recalling that W_k is the intracluster NID, we substitute $W_k(d)$, the randomly generated uniformly distributed data intracluster NID for the first term, and $W_k(x)$, the intracluster NID for our data for the second term. Therefore,

$$\delta(x) = K(x^*) \times (W_k(d) - W_k(x)). \tag{20}$$

Since we are seeking a solution where $\delta(x)$ reaches a maximum or minimum value while $K(x^*)$ remains constant, we can rewrite (20) as follows:

$$\delta(x) = W_k(d) - W_k(x). \tag{21}$$

Comparing (21) and (8), we see that the gap statistic and the randomness deficiency are identical except for the logarithms of the two terms. The logarithms in (8) can be interpreted as expressing a conventional distance measure (e.g., euclidean) as a number of bits. When using the gap statistic with an NID-based distance measure, the formulation in (21) should be used.

One final observation is that the randomness deficiency measures how well a model captures the meaningful information in the data. In general, the goal is to select a model where $\delta = 0$. However, in the analysis here, the uniformly distributed randomly generated data used by the gap statistic represents the case where there is no structure or clusters in the data. We maximize the gap statistic in this case because we seek a representation that is as atypical as possible compared to the unstructured data. In Section 8 below, we describe methods for using the gap statistic to minimize any information in the quantization residuals, looking for a randomness deficiency approximately equal to zero, an approach that is more consistent with Vitanyi's second trilogy on algorithmic information statistics [4], [5], [6].

8 Quantization and the NCD

For numeric time series, the performance of the NCD, as measured by the magnitude of the gap statistic, can be improved by using quantization [44] as a preprocessing step. In recent papers [17], quantization was shown to improve the accuracy of the NCD on a wide variety of machine learning problems.

Numeric time series data can be quantized to a required level of precision by histogramming the data [49]. Similar numeric values are assigned to the same histogram bin and a representative value for that bin, its *symbol*, is used to represent all numeric values assigned to it. Placing the bins in the histogram such that each bin contains an equal number of data points (each bin has roughly equal probability) maximizes the entropy of the quantization, thereby maximizing the amount of information conveyed by each symbol [28]. For this, optimization techniques can be used to partition the histogram or an underlying distribution for the data can be assumed, allowing equiprobable regions to be estimated analytically. In the SAX approach [50] and in [49], the data is assumed to have a normal distribution. Their implementation is limited by their use of a lookup table that defines the locations of a maximum of 10 bins and is limited to one dimension. We propose a method that allows quantization of data of any dimension, to any number of symbols, under the assumption of normally distributed data.

If $x = [x_1, \ldots, x_p]$ is a p-dimensional normally distributed random variable with mean μ and covariance Σ, then the equiprobable regions of x are ellipsoids. These ellipsoids may be calculated as follows: First, define the quadratic form of x as

$$Q(x) = (x - \mu)^T \Sigma^{-1} (x - \mu), \tag{22}$$

where $Q(x)$ is a chi-square-distributed random variable with p degrees of freedom [51]. The hyperellipsoid given by $Q(x) = \chi^2(1 - \alpha, p)$ is the boundary of the $100\alpha\%$ confidence ellipse. The method proceeds as follows: The breakpoints or boundary points separating symbols are linearly spaced on the interval [0, 1]. The chi-square inverse cumulative distribution function (CDF) is used to find the value whose cumulative probability corresponds to each breakpoint. A symbol is assigned to x based on which region of the chi-square CDF (as defined by the breakpoints) that $Q(x)$ falls into. As a simple example, consider the standard univariate normal distribution $(\mu = 0, \Sigma = 1)$. From (22), $Q(x) = x^2$. The inverse chi-square CDF computes $Q(x) = x^2 = \chi^2(1 - 0.5, 1) = 0.4549$ and, thus, $x = \pm 0.6745$. The two equiprobable regions for this example are the inside or outside regions of the interval $[-0.6745, 0.6745)$, respectively. Note that, although there are other possible regions that are equiprobable for this case, e.g., $x \leq \mu$ and $x > \mu$, our closed-form approach works for any number of regions in any dimension.

The information that is discarded as the number of symbols is reduced is termed the residual. *We require that the residuals contain no meaningful information*. By removing residuals that contain random noise, we improve the ability of the NCD to detect similarities between objects. Our approach is to choose the optimal number of bins that maximize the value of the gap statistic. For a given number of symbols N, the time sequence $x_i = S_i^N + R_i^N$, where S_i^N represents the quantized version of x_i using N symbols. An optimal quantizer places the representative points (code vectors) for a symbol at the centroid of that symbols' corresponding region in the histogram [52]. The residuals R_i^N are obtained by replacing each symbol with a numerical value equal to the representative point for that symbol and subtracting the series of representative points from the original time series. As the number of symbols is reduced, the residual must be verified to contain no meaningful information. For this, we propose the following two steps.

Step 1. Compute a matrix of NCD matrix $\mathrm{D}(x_i)$ on the numeric time series data in which the $(i, j)\mathrm{th}$ entry is given by $\mathrm{NCD}(x_i, x_j)$. We refer to the distance matrix obtained from computing NCDs between the raw numeric data $\mathrm{D}(x_i)$ as $\mathrm{D}(S^1)$, where the superscript indicates the quantization levels. There is no residual associated with the raw numeric data, so we define $\Gamma(\mathrm{D}(R^1)) = 0$. Next, we compute the distance matrix $\mathrm{D}(S^N)$ on the quantized representation for each number of symbols N (the range of N is given at the end of this section).

Step 2. The NCD-based distance matrix $\mathrm{D}(R^N)$ is computed over the residuals R_i^N for each number of symbols N. This enables us to compute the gap statistic for the corresponding distance matrix. If the residuals consist of random noise, the gap statistic should find one cluster showing little or no improvement over the uniformly distributed randomly generated data, i.e.,

$$\Gamma\big(D(R^N)\big) \leq 0^+, \tag{23}$$

and observe that if any R^N violates (23), we cannot consider a representation of size N. We can now formulate the problem of finding the optimal quantization level as follows:

$$N^* = \arg\max_N \Gamma\big(D(S^N)\big) \;\text{ such that }\; \Gamma\big(D(R^N)\big) \leq 0^+. \tag{24}$$

Two additional bounds can be set on the number of symbols that must be considered. First, the number of atomic (single) symbols utilized by the compression algorithm is typically limited to the conceptually arbitrary but practically convenient 93 printable characters from the 128-symbol ASCII code. We limit ourselves to atomic symbols to ensure that the symbols themselves do not induce any similarities in the data. Second, we can use the identity axiom from the definition of a distance metric, $d_{i,j} = 0$ iff $i = j$, and observe that if, for a certain number of symbols n, any $\mathrm{NCD}(S_i^n, S_j^n)$ violates this, then we can ignore all $N \leq n$. Intuitively, this represents "zooming out" too far, causing objects to merge. In summary, the constraints on the number of symbols N in the quantized representation are $\Gamma(\mathrm{D}(R^N)) \leq 0^+$ and $\mathrm{NCD}(S_i^N, S_j^N) > 0$, $i \neq j$. Finding the optimal number of symbols requires an exhaustive search over N, subject to these constraints.

9 Automatic Feature Subset Selection

We wish to find the subset of features that maximizes our criterion function, the gap value Γ. In a p-dimensional feature set, there are $2^p - 1$ possible combinations of features to search, a task that becomes prohibitive for a large p. Techniques including implicit exhaustive (e.g., branch-and-bound) methods, floating search, or evolutionary programming can be used to reduce the complexity of the search [53], [54]. The gap statistic is a natural criterion for evaluating the quality of a feature subset. Using the theoretically optimal NID, it would be expected that the gap function would monotonically increase, and irrelevant features would be ignored by the NID. However, this is not the case when using the NCD. Irrelevant or noisy features may impact the performance of the compression algorithm, although in a limited manner [18]. Therefore, feature subset selection cannot use a branch-and-bound approach. This makes a floating search method the logical choice for feature subset selection in our applications of interest.

Sequential search methods [53], [54] are suboptimal approaches designed for computational efficiency and for dealing with nonmonotonic objective functions. Sequential forward selection adds the best feature at each iteration, while sequential backward selection removes the worst feature at each iteration. Given the currently selected feature subset x_i, we pick the unselected feature that best improves $\Gamma(x_i)$, the gap value for feature subset x_i:

$$\alpha^+ = \arg\max_{\alpha \notin x_i} \Gamma(x_i + \alpha). \tag{25}$$

In order to reduce the impact of local optima, the floating forward search algorithm adds a backtracking step after each forward step:

$$\beta^- = \arg\min_{\beta \in x_i} \Gamma(x_i - \beta) \;\text{ such that }\; \Gamma(x_i - \beta) \leq \Gamma(x_i), \tag{26}$$

iteratively removing any features that do not improve the outcome.

10 Generating the Summary

The above sections described the key concepts. This section summarizes the computational steps starting with the image sequences and ending with the summary.

1. We are given a set of S image sequence(s), denoted $\{\{I_t^s(x)\}_{t=1}^{T_s}\}_{s=1}^{S}$. We run application-specific algorithms to extract objects and their features in each image and track the objects over time through the image sequence(s). Our methodology allows objects to come from a single image sequence, from multiple image sequences, or from a combination thereof. We denote the time course of feature values for an individual object as Ω_i. Each object time course has a feature vector of dimension $T_i \times F_i$, where T_i represents the number of time samples for Ω_i and F_i represents the number of features for Ω_i. The number of time samples can be different for each object. In principle, the number of features can also be different, although in practice, for all data sets analyzed to date by us, F_i is equal for all objects. For convenience, we abbreviate F_i as F.
2. Create a data structure named the ATG bringing together objects extracted from the image sequence(s), together with their time course of feature values. If our ATG consists of M objects (extracted from one or more image sequences), we use the symbol Ω to denote the set of time courses of feature values for all M objects $\{\Omega_1, \dots, \Omega_M\}$.
3. Using the algorithm described in Section 9 or otherwise choose a subset of features, denoted $f \subseteq \{1, \dots, F\}$. We use the symbol Ω^f to denote the time course of this subset of features for all M objects:

   a. Using the multidimensional quantization algorithm described in Section 8, quantize the numeric values in Ω^f to assign one of N symbols. We write $Q_N(\Omega^f)$ to represent the N-symbol quantization of Ω^f. The special value of $N = 1$ represents the case where quantization is not used. In our examples, $N_{\max} = 93$, corresponding to the 93 printable characters from the 128-character ASCII code. We limit ourselves to atomic symbols to ensure that the symbols themselves do not inflict any artifactual similarities in the data. For the chosen feature subset, repeat the following steps by varying the quantization level N from 1 to $N_{\max}$:

      i. The NAID (described in Section 4) is used to calculate the $M \times M$ element pairwise distance matrix (5).

      ii. The Gap spectral clustering algorithm (described in Section 7) is run on the above distance matrix. This estimates the number of clusters k and the gap value Γ_k for the k clusters. In the absence of a priori knowledge of the maximum number of groups in the data, k is bounded by the number of object time courses in the ATG.

      iii. Retain the values $\hat{f}$, $\hat{N}$, and $\hat{k}$ that maximize the gap statistic value.

   The values $\hat{f}$, $\hat{N}$, and $\hat{k}$ that maximize the gap statistic overall are used to generate the summary:

$$\hat{f}, \hat{N}, \hat{k} = \underset{f,N,k}{\arg\max}\{\Gamma_k\big(NAID\big(Q_N(\Omega^f)\big)\big)\}. \qquad (27)$$

4. Using $\hat{f}$, $\hat{N}$, and $\hat{k}$, compute the list of cluster assignments $[l_1, \dots, l_M]$ $l_i \in 1, \dots, \hat{k}$ for each of the M object time courses in Ω^f. This is accomplished using a spectral clustering algorithm [47] Ψ that takes the NAID distance matrix and the number of clusters. This is expressed mathematically as follows:

$$[l_1 \cdots l_m] = \Psi(NAID(Q_{\hat{N}}(\Omega^{\hat{f}}), \hat{k}) \quad l_i \in 1 \dots \hat{k}. \qquad (28)$$

The list of cluster assignments $[l_1, \dots, l_M]$ $l_i \in 1, \dots, \hat{k}$ along with $\hat{f}$ constitute our summary. At first glance, the summary appears quite abstract. We developed a graphical tool that allows a user to inspect the summary overlaid on the ATG at any chosen level of detail, view the original image sequence data when appropriate, and display various computed quantities, all in context. Once the summary is run, this tool displays the ATG with each object time course color coded by the cluster assignments $[l_1, \dots, l_M]$. It also displays a plot of the gap statistic as a function of k. The gap statistic value is displayed for the user alongside the summary. Based on the visualization, the user can choose to either accept the summary or rerun the above analysis using a different f, N, and k.

11 Experimental Results

Results are presented from four different application domains. A synthetic or simulated data set was analyzed, allowing precise control over differences between objects within and across image sequences. The features for the synthetic data set were taken from [55] and consist of a 23-dimensional feature vector. The seven features relating to 3D cell motion and growth were modeled as described below, the remaining 15 features were set to random values. Gamma-distributed random variables [56] were used in the modeling and are denoted here as $\gamma(\kappa, \theta)$, where $\kappa \times \theta$ is the mean of the distribution and $\kappa \times \theta^2$ is the variance. When $\kappa \gg \theta$, the gamma distribution resembles a normal distribution but with the special property of taking on only positive values. Cell motility is based on a "run-and-tumble" model [57] similar to the motion of bacteria. This consists of periods of rapid directed movement followed by a period of random undirected motion. This was not intended to be a comprehensive model of cell motion; it specifically ignores cell-cell interactions and other factors affecting motility. Cell lifespan is also modeled as a gamma-distributed random variable. Once a cell reaches its lifespan, it undergoes cell division, producing two new cells, or, if a predetermined population limit has been reached, the cell undergoes apoptosis, or dies. The final

TABLE 1
Parameters of Synthetic Data Set

- Run: length of run interval = $\gamma(5,3)$
- Tumble: length of tumble interval = $\gamma(60,5)$
- Speed: velocity (pixels per frame) during run = $\gamma(8,2)$
- Lifespan: time between cell division = $\gamma(50,10)$
- Initial radius $r_0 = \gamma(200, 0.05)$
- 3000 frames, 138 cells

aspect of the model is cell size. The initial cell radius, denoted r_0, is a gamma-distributed random variable. The cells growth rate is labeled υ. At the end of its lifespan, the cell doubles its radius. The radius at time t is given by

$$r(t) = r_0 + r_0 \cdot \left(\frac{t - t_0}{lifespan}\right)^{\upsilon}. \tag{29}$$

Table 1[2] summarizes the parameters used to define the behavior of the synthetic data set. Two different populations were induced in the data by using different values for υ in (29). For population 1, υ was set to 0.9. For population 2, υ was set to 3. An automatic summary for this data set was generated. The automatic feature subset selection correctly identified "volume" as the sole differentiating feature. The summary consisted of two groups, each corresponding exclusively to the two populations. Fig. 2 shows a sample frame from the sequence (Fig. 2a), the full ATG (Fig. 2b), a zoomed in view of the ATG section shown in red (Fig. 2d) with the attributes from each ATG vertex (Fig. 2c), and, finally, the ATG colored with the summary results (Fig. 2e).

The controlled nature of the synthetic data set allows us to compare the NAID to other distance measures. Given the common occurrence of multidimensional time series, it is surprising to note that there are very few distance measures that allow comparing them. DTW [58] and LCSS [59] are two widely used distance measures. However, both methods suffer from significant limitations. DTW performs poorly in the presence of noisy or missing data. For biological applications, noisy and missing data are the rule rather than the exception. LCSS analysis requires a symbolic representation, which conflicts with our requirement of analyzing mixtures of numeric and quantized data (see Section 8). In order to benchmark our choice of the NAID measure, we used the Bhattacharya distance measure. It has been used successfully in similar other work. For example, Comaniciu et al. [60] chose the Bhattacharya distance for mean-shift tracking because of its near optimality in a Bayes' error sense and also because of its straightforward implementation, especially for 1D data. As surveyed by Zhou and Chellappa, several other probabilistic distance measures [61] work well with multidimensional time series, but our experiments focused on the Bhattacharya distance. Fig. 3a shows the average per-class error rate using the Bhattacharya distance and the NAID, with and without quantization for various lengths of time series, based on the synthetic experiment. We conducted an additional experiment to verify the robustness of NAID to noise. We considered two scenarios. First, to test the robustness of NAID to perturbation, we generated four sequences of pseudorandom values $\{X_1, X_2, X_3, X_4 = X_1 + \eta \times X_3\}$, where $\{X_1, X_2, X_3\}$ are sequences of pseudorandom values uniformly distributed on [0, 1] of length N (chosen here as 2^{14}). We varied η from 0.05 to 2 and computed $NAID(X_1, X_2)$, $NAID(X_1, X_3)$, and $NAID(X_2, X_3)$, selecting the quantization level that maximizes the NAID. The resulting plot is shown in Fig. 3b. For this scenario, our use of quantization enables the NAID to correctly recognize that the perturbed time series X_3 is more closely related to X_1 than to X_2.

We next conducted a simple experiment to evaluate the effect of permuting the values in a time series. We generated two sequences of pseudorandom values X_1 and X_2 uniformly distributed on [0, 1], of length N (chosen here as 2^{10}). From these, we computed a group of D (chosen here as 24) sequences labeled $X_1^{(i)}$ that are random permutations of X_1 by swapping each element with another element at random. The same process was repeated for X_2. This generated a total of 50 sequences. The resulting summary found two groups in the data. The first consisted exclusively of X_1 and its D permutations. The second consisted exclusively of X_2 and its D permutations. For this example, our method proved robust to permutation of the time series data.

We also consider the performance our methodology in the presence of object extraction and tracking errors. For this, we randomly introduced errors into the object extraction or tracking results and measured the accuracy of the summary on the synthetic data set, for which the correct result is known. This analysis was performed using the object time course data for the single "volume" feature previously selected by the feature subset selection algorithm. Errors are introduced as follows: A single object extraction error is introduced by substituting one value from each object time course with a random number generated over the range of values observed over all object time courses for the given feature. The specified numbers of object extraction errors are induced in each object time course. A single tracking error is introduced by randomly selecting two objects at some time and switching their time courses from that time onward. The specified number of tracking errors is induced over all object time courses. This is repeated using 0-20 errors. Fig. 4 shows the accuracy of summarization in the presence of object extraction and tracking errors. Fig. 4a represents object extraction errors. Fig. 4b represents tracking errors. It is intuitive that tracking errors, which can impact a greater portion of the time sequence data than isolated object extraction errors, will result in more errors in the summarization. Fig. 4 supports this intuition, showing summarization to be more robust to object extraction errors than to tracking errors.

The second data set analyzed was from a neuroprosthetic device application. As described in [62]. an ex vivo preparation captured real-time images of tissue deformation during neuroprosthetic device insertion using thick tissue slices from rat brains prepared with fluorescently labeled vasculature. Three different device shapes were analyzed (sharp, medium, and blunt). Each of the devices was inserted at three different speeds (slow, medium, and fast). The most important factor in determining tissue damage was found to be

[2] A Matlab program for generating the synthetic data may be downloaded from www.ecse.rpi.edu/~roysam/PAMI2008.

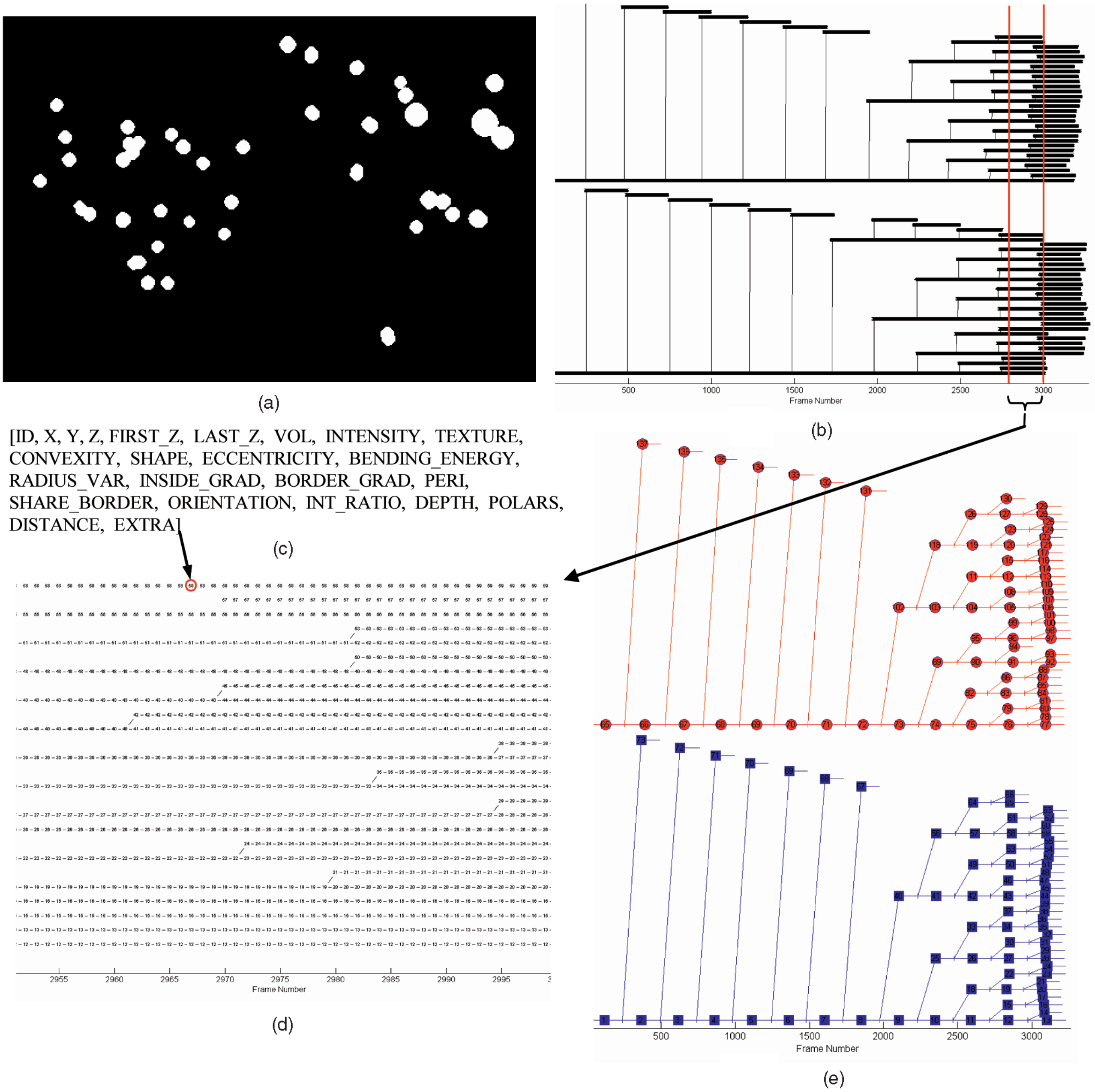


Fig. 2. Analysis of a synthetic data set. (a) Last frame in the image sequence. (b) ATG. (c) List of features stored at each ATG node. (d) Magnified view of a part of the ATG, showing details of the area inside the red lines. (e) ATG colored with summary results. The seven features related to movement and growth were modeled; the remaining 15 were set to random values. For this experiment, the radius state variable for one group of cells grew at a rate of $r^{0.9}$ and, for a second group of cells, it grew at a rate of r^{3}. The initial and final radii of both groups, as well as their motion, were drawn from identical distributions. The summary identified "volume" as the differentiating feature and correctly identified the cells by group.

the speed at which the neuroprosthetic device was inserted. For each of the 59 insertion movies, 100 interest points were identified and tracked. Our analysis of this data set began with the tracking results. The sequences did not all have the same number of images. The 2D time sequence of horizontal and vertical distance traveled by each interest point was the only feature considered in this analysis. Fig. 5 shows two images from this sequence, with tracking results overlaid as green lines (Fig. 5a), the 2D motion of each of the 100 control points used in the tracking (Fig. 5b), and the summary results overlaid on the ATG (Fig. 5c). The automatically generated summary found two groups in the data. The first group consisted exclusively of slow-insertion-speed movies; the second group consisted exclusively of fast and medium insertion speed movies. *This result agrees completely with the considerably more effort-intensive multidisciplinary analysis published in* [62], *culminating in the conclusion that insertion speed was the dominant factor affecting tissue strain.*

The third data set analyzed was time-lapse sequences of murine neural progenitor cells cultured in vitro. In [9], Al-Kofahi et al. presented a method for automatic tracking and lineage tree construction from these image sequences. Neural progenitor cells differentiate into either glial cells or

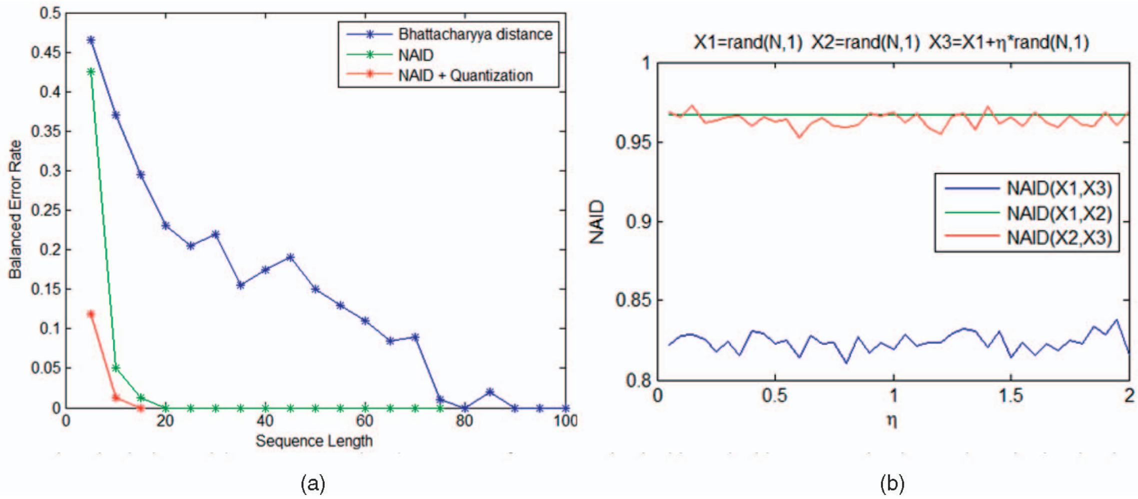


Fig. 3. (a) Comparing the balanced (average per class) error rate for our method with and without quantization to that obtained using the Bhattacharya distance. (b) For small sample sizes, the proposed NAID clearly outperforms the Bhattacharya measure, especially when quantization is used. The NAID used in conjunction with quantization is robust to additive random noise.

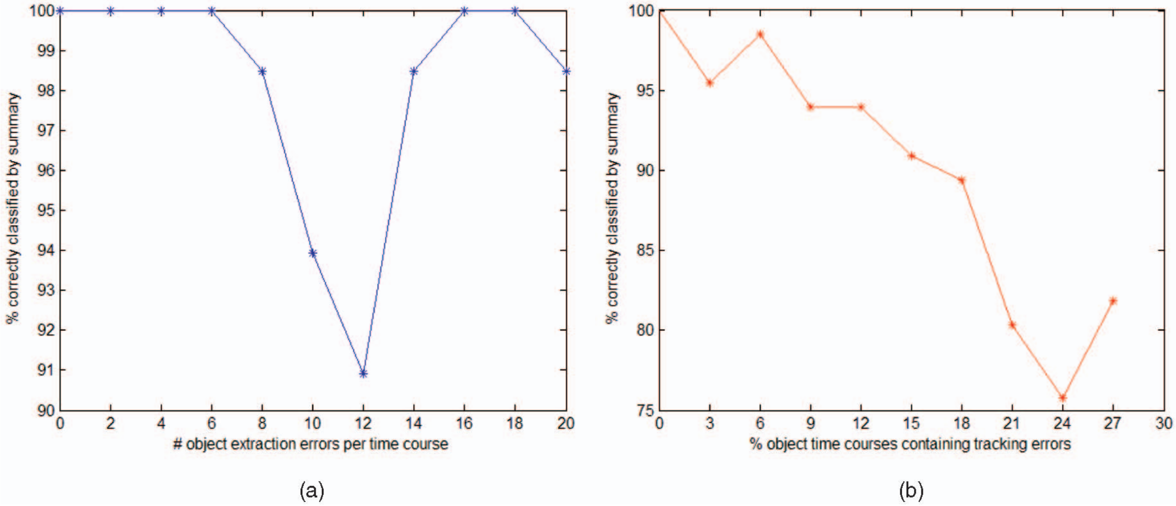


Fig. 4. Summarization of the synthetic data set containing 66 simulated cells of two types is more robust to (a) object extraction errors than (b) tracking errors. A single object extraction error is induced by substituting one value from each object time course with a random number generated over the range of values observed over all object time courses for the given feature. A single tracking error is induced by randomly selecting two objects at some time and switching their time courses from that time onward. When 30 percent of object time courses were affected by tracking errors, the summary was no longer able to find two groups in the data and the experiment was stopped.

neurons in vitro. Currently, the definitive classification of neurons versus glial and progenitor cells requires staining the culture for the molecule β-tubulin III that selectively labels neurons. The staining process is fatal to all cells in the culture, so it can only be done after the image sequence recording is complete. Fig. 1 shows the ATG and results for this sequence. The final application involves a dual-mode phase and a fluorescence time-lapse image sequence showing a live neuron and a fluorescently labeled kinesin protein believed to play a role in axonal specification, the process whereby one of a developing cell's neurites is chosen by the cell as the axon [63]. The data for this analysis was based on nine cell cultures. For each culture, two image sequences were obtained. The first image sequence contains a fluorescence image of the kinesin protein. The second image sequence contains a phase contrast image of the neuron. The image sequences contain between 30 and 72 frames. Since this image data contains more than one channel of information and the associations between information in channels are of considerable scientific interest in their own right, we add features to the ATG that quantify associations. Specifically, at each time slice, we find the percentage of the kinesin protein closest to the distal (growing) end of each neurite. This associative feature was included in the summarization analysis along with all other features. The resulting summary found two groups in the data. Fig. 6 shows the ATG colored with the summary results. All of the neurons that had not undergone axonal specification were assigned to one group, shown colored red. All of the neurons that had undergone axonal specification were assigned to the second group, shown colored blue. Vertices on the ATG correspond to individual neurites and are colored green when the kinesin protein is at the distal end of that neurite. Fig. 6b shows images from two sequences in which axonal specification occurred (sequence D and sequence H) and from sequences in which

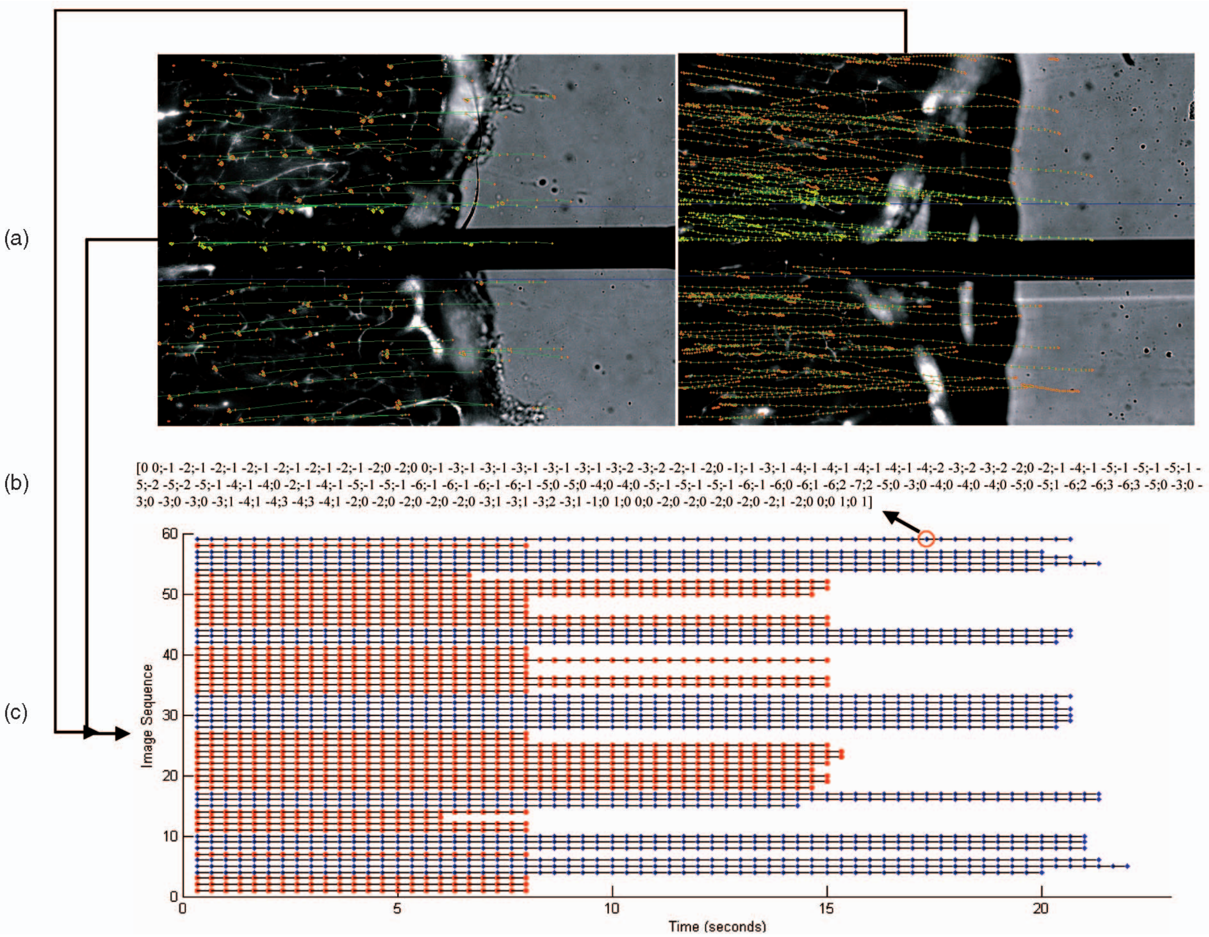


Fig. 5. (a) Two frames from a 59-frame series capturing the insertion of a $100-\mu\mathrm{m}$ wide silicon neuroprosthetic device into a coronal slice of rat brain with fluorescently labeled vasculature. Devices with three different tip shapes were inserted at three different speeds. (b) Attributes from one of the 2,600 ATG vertices, showing the 2D movement of each of the 100 control points per movie. (c) ATG colored with the summary results. Sequences colored in blue correspond to slow insertions; sequences in red correspond to medium and fast insertions. These fully automated results concord with previously published findings based on sophisticated biomechanical analysis concluding that insertion speed was the primary factor affecting tissue strain.

it did not occur (sequence B and sequence E). *The automated summary correctly identified all the neurons that underwent axon specification for these nine movies.*

We have also used the proposed method to analyze cultured retinal progenitor cells (data not shown here but being presented in a biological journal). Our automatic summary discovered previously unknown biologically significant behavioral differences among progenitors that produce different types of neurons. This discovery is particularly exciting as it will enable homogeneous populations of progenitors to be isolated, enabling the search for a genetic basis for determining a progenitors' outcome, which in turn could potentially lead to therapeutic applications for some forms of blindness using retinal progenitor cells. Other applications studied to date include the viability of mouse oocytes stained for mitochondria (preliminary data not shown here). Our method correctly identified differences in the movement patterns of mitochondria for oocytes undergoing two different forms of cell death—apoptosis and necrosis.

Overall, our experiments indicate the practicality of applying a common set of tools across multiple image sequence analysis applications. Our interests lie in the realm of biological images, so our experiments were limited to that domain. The conceptual generality of our approach indicates broader applicability to nonbiological image sequence data, but that remains to be investigated

12 Conclusions and Discussion

Our work demonstrates the practicality of using nonprobabilistic concepts from algorithmic information theory and algorithmic statistics to summarize changes within and across image sequences. Our use of the NID [1] and of its practical approximations, the NCD [3] and the NAID, allows us to compare a variety of data types including scalars, vectors, and multidimensional time series that are otherwise awkward (e.g., different lengths and mixed data types) in a manner that is theoretically optimal and powerful yet straightforward in implementation. This paper presents the first technique that we are aware of for approximating the notion of *meaningful information* from algorithmic statistics. The gap statistic, a widely used technique for estimating the number of clusters in a data set, is shown to be an effective tool for estimating the randomness deficiency for clustering models. Randomness

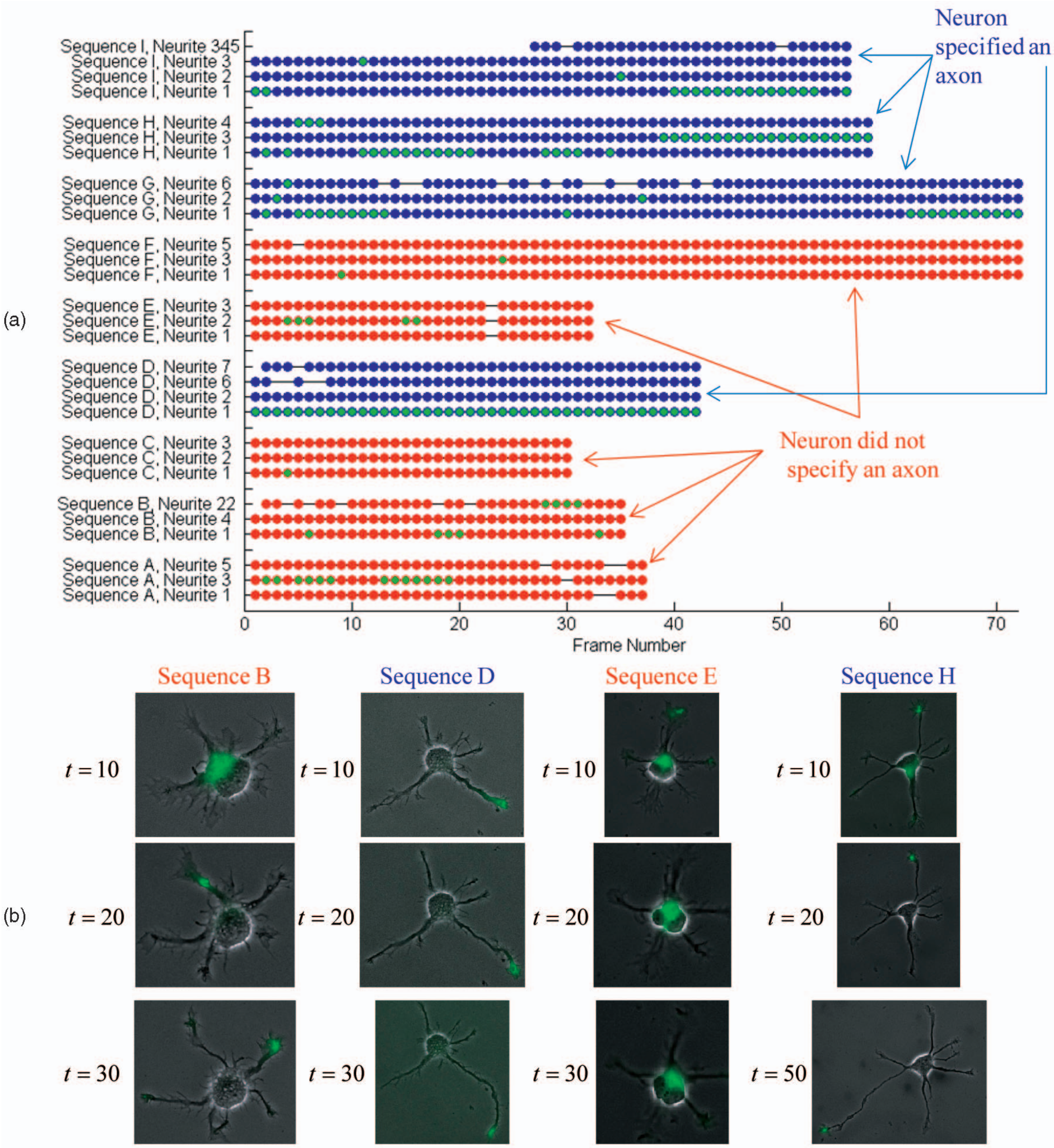


Fig. 6. (a) ATG with vertices colored according to the summary for the protein association application. Two groups were found in the summary; these are colored red and blue. A neurite vertex is shaded green when the kinesin protein is closest to the distal end of that neurite. (b) Example images from four of the sequences showing the fluorescently labeled kinesin protein in green overlayed on the phase contrast imaged neuron. For this application, our automatically generated summary correctly separated cells that had undergone axonal specification (colored blue) from those that had not (colored red).

deficiency is a concept from algorithmic statistics that quantifies how well a model captures the meaningful information in a specific digital object. Algorithmic statistics has until now been a theoretical concept quantifying the relationship between an individual digital object and a model, based on Kolmogorov complexity. The lack of practical approximations for randomness deficiency have to date prevented any practical applications of this otherwise powerful theory. Our work, in effect, connects the trilogy of papers describing a practical approximation to the algorithmic information-theoretic distance measure [1], [2], [3] to the second trilogy describing the theoretical field of algorithmic statistics [4], [5], [6].

Our approach provides a novel methodology for analysis of image sequence data at multiple levels of precision. It achieves an automatic closed-form multidimensional quantization using the gap statistic as a guide. The optimal quantization level is selected jointly with other problem parameters such as the number of groups in the summary and the relevant feature subset. Our method also analyzes the quantization residuals to ensure that meaningful information is not being discarded by the quantization process. Previous work using MDL for a multiresolution analysis was limited by the need to construct tightly tuned application specific MDL representations [63].

This paper also contributes to the data mining literature. We have presented here a closed-form method for multidimensional quantization of numeric data capable of assigning any number of symbols to data of any dimension such that the symbols are equiprobable under the assumption of normally distributed data. This dramatically improves upon the previously published method of SAX, a widely used quantization technique developed in the data mining community. Currently, SAX is only defined for up to 10 symbols and for 1D data [50]. The data is also analyzed in numeric (unquantized) form and other types of quantization could potentially be included.

There are limitations to our approach. Generating a pairwise distance matrix is computationally expensive, requiring $(M^2 - M)/2$ comparison operations for an M-object ATG. For a Matlab implementation on a standard desktop PC, generating a summary for a 50-object ATG takes approximately 4 hours. Fortunately, the methodology is inherently parallelizable; a preliminary implementation on a Blue Gene supercomputer achieved a performance improvement of 700 percent. Also, because our approach is based on algorithmic, rather than classical statistics, there are no analogs to measures such as p values or confidence intervals. The development of such measures remains a topic for future research.

The four different application domains that we considered are actual problems of interest but have in the past been treated with separate algorithm development efforts. *We were gratified to see that our experimental studies revealed that summaries that were meaningful in the sense of the gap statistic were also biologically meaningful.* For the synthetic data set, our method identified a visually subtle difference in growth rate among two populations of cells and identified the single feature "volume" out of 23 that best differentiated the populations. For the neural prosthesis movies, it analyzed the 2D movement of interest points within the tissue and automatically grouped movies on the basis of insertion speed. This was in agreement with a very detailed and application-specific analysis that found insertion speed to be the dominant factor affecting tissue strain [62]. For the cultured neural progenitor cell data sets, our method not only reproduced lineage trees that were computed using a highly application-specific set of algorithms [9] but also differentiated neurons from glia and progenitor cells from phase contrast data alone, without the use of fixative stains. Indeed, the automatic summary grouped these cell types differently based on their ATG features. Subsequently, the cultures were fixed and stained fluorescently for cell-type-specific markers. These experiments confirmed the validity of the automated analysis without the benefit of fluorescent staining. [9]. Finally, analyzing the association between cells' neurites and a kinesin protein enabled the automatic identification of cells that had undergone axonal specification [64]. *All four of these applications were analyzed using a single implementation that requires neither parameters nor application-specific knowledge.*

In conclusion, we feel that this paper provides valuable insights on the design of a new generation of image sequence analysis approaches. Although many other summarization approaches can be imagined, our clustering-based strategy has the advantage of using well-studied methods and the advantage of meaningful summarization. When considering other summarization strategies, some of the concepts and components from our work can be reused broadly.

Acknowledgments

This work was performed while A. Cohen was with the Department of Electrical, Computer and Systems Engineering, at the Rensselaer Polytechnic Institute. A preliminary version of this work appeared as a part of a conference proceeding [65]. Various portions of this research were supported by the Center for Subsurface Sensing and Imaging Systems under the Engineering Research Centers Program of the US National Science Foundation (EEC-9986821) and by Rensselaer Polytechnic Institute. The authors would like to thank Dr. William Shain at the Wadsworth Center and Dr. S.J. Kim at Seoul National University for the neuroprosthetics data, Yousef Al-Kofahi and Omar Al-Kofahi for help with tracking data, Susan Goderie for her help with the imaging of neural progenitor cells and advice, Dr. Stefanie Kaech at Oregon Health and Sciences University for time-lapse neuronal data, Hussein Sharafeddin for suggestions on the display of ATG, and Professor Richard Radke for insightful comments on the role of quantization in algorithmic information theory and automatic feature selection.

References

[1] C.H. Bennett, P. Gacs, L. Ming, M.B. Vitanyi, and W.H. Zurek, "Information Distance," *IEEE Trans. Information Theory,* vol. 44, pp. 1407-1423, 1998.

[2] R. Cilibrasi and P.M.B. Vitanyi, "Clustering by Compression," *IEEE Trans. Information Theory,* vol. 51, pp. 1523-1545, 2005.

[3] M. Li, X. Chen, X. Li, B. Ma, and P.M.B. Vitanyi, "The Similarity Metric," *IEEE Trans. Information Theory,* vol. 50, pp. 3250-3264, 2004.

[4] P. Gacs, J. Tromp, and P. Vitanyi, "Algorithmic Statistics," *IEEE Trans. Information Theory,* vol. 47, pp. 2443-2463, 2001.

[5] N.K. Vereshchagin and P.M.B. Vitanyi, "Kolmogorov's Structure Functions and Model Selection," *IEEE Trans. Information Theory,* vol. 50, pp. 3265-3290, 2004.

[6] P. Vitanyi, "Meaningful Information," *IEEE Trans. Information Theory,* vol. 52, pp. 4617-4626, 2006.

[7] R.J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam, "Image Change Detection Algorithms: A Systematic Survey," *IEEE Trans. Image Processing,* vol. 14, p. 294, 2005.

[8] K.A. Al-Kofahi, S. Lasek, D.H. Szarowski, C.J. Pace, G. Nagy, J.N. Turner, and B. Roysam, "Rapid Automated Three-Dimensional Tracing of Neurons from Confocal Image Stacks," *IEEE Trans. Information Technology in Biomedicine,* vol. 6, p. 171, 2002.

[9] O. Al-Kofahi, R.J. Radke, S.K. Goderie, Q. Shen, S. Temple, and B. Roysam, "Automated Cell Lineage Tracing: A High-Throughput Method to Analyze Cell Proliferative Behavior Developed Using Mouse Neural Stem Cells," *Cell Cycle,* vol. 5, pp. 327-335, Feb. 2006.

[10] O. Al-Kofahi, R.J. Radke, B. Roysam, and G. Banker, "Automated Semantic Analysis of Changes in Image Sequences of Neurons in Culture," *IEEE Trans. Biomedical Eng.,* vol. 53, pp. 1109-1123, 2006.

[11] O. Debeir, P. Van Ham, R. Kiss, and C. Decaestecker, "Tracking of Migrating Cells under Phase-Contrast Video Microscopy with Combined Mean-Shift Processes," *IEEE Trans. Medical Imaging,* vol. 24, pp. 697-711, 2005.

[12] H. Narasimha-Iyer, A. Can, B. Roysam, H.L. Tanenbaum, and A. Majerovics, "Integrated Analysis of Vascular and Non-Vascular Changes from Color Retinal Fundus Image Sequences," *IEEE Trans. Biomedical Eng.,* vol. 54, pp. 1436-1445, Aug. 2007.

[13] N. Roussel, C.A. Morton, F.P. Finger, and B. Roysam, "A Computational Model for *C. elegans* Locomotory Behavior: Application to Multi-Worm Tracking," *IEEE Trans. Biomedical Eng.,* vol. 54, pp. 1786-1797, Oct. 2007.

[14] J.A. Tyrrell, E. di Tomaso, D. Fuja, R. Tong, K. Kozak, R.K. Jain, and B. Roysam, "Robust Modeling of 2-D/3-D Microvasculature Imagery Using Super-Gaussians," *IEEE Trans. Medical Imaging,* vol. 26, pp. 223-237, Feb. 2007.

[15] T.M. Cover and J.A. Thomas, *Elements of Information Theory.* John Wiley & Sons, 1991.

[16] M. Li and P.M.B. Vitanyi, *An Introduction to Kolmogorov Complexity and Its Applications,* second ed. Springer, 1997.

[17] E. Keogh, S. Lonardi, and C.A. Ratanamahatana, "Towards Parameter-Free Data Mining," *Proc. ACM SIGKDD,* 2004.

[18] M. Cebrian, M. Alfonseca, and A. Ortega, "The Normalized Compression Distance Is Resistant to Noise," *IEEE Trans. Information Theory,* vol. 53, pp. 1895-1900, May 2007.

[19] P. Grünwald, I.J. Myung, and M. Pitt, *Advances in Minimum Description Length: Theory and Applications.* MIT Press, 2005.

[20] J. Rissanen, *Stochastic Complexity in Statistical Inquiry.* World Scientific, 1989.

[21] Z. Bao, J.I. Murray, T. Boyle, S.L. Ooi, M.J. Sandel, and R.H. Waterston, "Automated Cell Lineage Tracing in *Caenorhabditis elegans,*" *Proc. Nat'l Academy Sciences USA,* vol. 103, pp. 2707-2712, 2006.

[22] C. Ronse, L. Najman, and E. Decenciere, "Mathematical Morphology: 40 Years On," *Computational Imaging and Vision 30.* Springer, 2005.

[23] G. Loy and A. Zelinsky, "Fast Radial Symmetry for Detecting Points of Interest," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, no. 8, pp. 959-993, Aug. 2003.

[24] D.G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int'l J. Computer Vision,* vol. 60, pp. 91-110, 2004.

[25] D. Reid, "An Algorithm for Tracking Multiple Targets," *IEEE Trans. Automatic Control,* vol. 24, p. 843, 1979.

[26] J. Munkres, "Algorithms for the Assignment and Transportation Problems," *J. SIAM,* vol. 5, pp. 32-38, Mar. 1957.

[27] A. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," *IEEE Trans. Information Theory,* vol. 13, pp. 260-269, Apr. 1967.

[28] F. Pitie, S.A. Berrani, A. Kokaram, and R. Dahyot, "Off-Line Multiple Object Tracking Using Candidate Selection and the Viterbi Algorithm," *Proc. IEEE Int'l Conf. Image Processing,* vol. 3, pp. 109-112, 2005.

[29] P.P. Pradeep and P.F. Whelan, "Tracking of Facial Features Using Deformable Triangles," *Proc. SPIE,* vol. 4877, S. Andrew, D.M. Fionn, M. James, and F.W. Paul, eds., pp. 138-143, 2003.

[30] C. Stauffer and W.E.L. Grimson, "Learning Patterns of Activity Using Real-Time Tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 8, pp. 747-757, Aug. 2000.

[31] B. Katz, J. Lin, C. Stauffer, and E. Grimson, "Answering Questions about Moving Objects in Surveillance Videos," *Proc. AAAI Spring Symp. New Directions in Question Answering,* Mar. 2003.

[32] P. Heas and M. Datcu, "Modeling Trajectory of Dynamic Clusters in Image Time-Series for Spatio-Temporal Reasoning," *IEEE Trans. Geoscience and Remote Sensing,* vol. 43, pp. 1635-1647, 2005.

[33] G. Medioni, I. Cohen, F. Bremond, S. Hongeng, and R. Nevatia, "Event Detection and Analysis from Video Streams," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 8, pp. 873-889, Aug. 2001.

[34] E. Gokcay, E. Gokcay, and J.C. Principe, "Information Theoretic Clustering," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 2, pp. 158-171, Feb. 2002.

[35] G.J. Chaitin, *Information, Randomness and Incompleteness: Papers on Algorithmic Information Theory.* World Scientific, 1990.

[36] C.E. Au, S. Skaff, and J.J. Clark, "Anomaly Detection for Video Surveillance Applications," *Proc. 18th Int'l Conf. Pattern Recognition,* pp. 888-891, 2006.

[37] B. Anton, F. Miquel, B. Imma, and M. Sbert, "Compression-Based Image Registration," *Proc. IEEE Int'l Symp. Information Theory,* pp. 436-440, 2006.

[38] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the Number of Clusters in a Dataset via the Gap Statistic," *J. Royal Statistical Soc.,* vol. 63, pp. 411-423, 2001.

[39] Y.G. Leclerc, "Constructing Simple Stable Descriptions for Image Partitioning," *Int'l J. Computer Vision,* vol. 3, pp. 73-102, May 1989.

[40] P. Adriaans and P.M.B. Vitanyi, "The Power and Perils of MDL," *Proc. IEEE Int'l Symp. Information Theory,* 2007.

[41] M. Yi, D. Harm, and H. Wei, "Segmentation of Multivariate Mixed Data via Lossy Data Coding and Compression," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 29, no. 9, pp. 1547-1562, Sept. 2007.

[42] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, "A Symbolic Representation of Time Series, with Implications for Streaming Algorithms," *Proc. Eighth ACM SIGMOD Workshop Research Issues in Data Mining and Knowledge Discovery* June 2003.

[43] V. Megalooikonomou, Q. Wang, G. Li, and C. Faloutsos, "A Multiresolution Symbolic Representation of Time Series," *Proc. 21st Int'l Conf. Data Eng.,* pp. 668-679, 2005.

[44] R.M. Gray and D.L. Neuhoff, "Quantization," *IEEE Trans. Information Theory,* vol. 44, pp. 2325-2383, 1998.

[45] D. Wood, *Theory of Computation.* Harper & Row, 1987.

[46] P.N. Yianilos, "Normalized Forms for Two Common Metrics," technical report, NEC Research Inst., Princeton, N.J., Dec. 1991.

[47] A.Y. Ng, M. Jordan, and Y. Weiss, "On Spectral Clustering: Analysis and an Algorithm," *Advances in Neural Information Processing Systems,* vol. 14, 2002.

[48] G. Hamerly and C. Elkan, "Learning the k in kmeans," *Advances in Neural Information Processing Systems,* vol. 17, 2003.

[49] L. Chen and M.T. Ozsu, "Multi-Scale Histograms for Answering Queries over Time Series Data," *Proc. 20th Int'l Conf. Data Eng.,* p. 838, 2004.

[50] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, "A Symbolic Representation of Time Series, with Implications for Streaming Algorithms," *Data Mining and Knowledge Discovery J.,* vol. 15, pp. 107-144, Oct. 2007.

[51] V. Chew, "Confidence, Prediction, and Tolerance Regions for the Multivariate Normal Distribution," *J. Am. Statistical Assoc.,* vol. 61, pp. 605-617, Sept. 1966.

[52] *Vector Quantization.* IEEE Press, 1990.

[53] L. Huan and Y. Lei, "Toward Integrating Feature Selection Algorithms for Classification and Clustering," *IEEE Trans. Knowledge and Data Eng.,* vol. 17, no. 4, pp. 491-502, Apr. 2005.

[54] P. Pudil, F.J. Ferri, J. Novovicova, and J. Kittler, "Floating Search Methods for Feature Selection with Nonmonotonic Criterion Functions," *Pattern Recognition,* vol. 2, pp. 279-283, 1994.

[55] G. Lin, M.K. Chawla, K. Olson, C.A. Barnes, J.F. Guzowski, and B. Roysam, "A Multi-Model Approach to Simultaneous Segmentation and Classification of Heterogeneous Populations of Cell Nuclei in 3D Confocal Microscope Images," *Cytometry,* vol. 71A, 2007.

[56] L.J. Bain and M. Engelhardt, *Introduction to Probability and Mathematical Statistics.* Duxbury, 1992.

[57] J.G. Mitchell and K. Kogure, "Bacterial Motility: Links to the Environment and a Driving Force for Microbial Physics," *FEMS Microbiology Ecology,* vol. 55, pp. 3-16, 2006.

[58] C.A. Ratanamahatana and E. Keogh, "Three Myths about Dynamic Time Warping," *Proc. SIAM Int'l Conf. Data Mining,* 2005.

[59] M. Vlachos, M. Hadjieleftheriou, D. Gunopulos, and E. Keogh, "Indexing Multi-Dimensional Time-Series with Support for Multiple Distance Measures," *Proc. ACM SIGKDD '03,* pp. 216-225, 2003.

[60] D. Comaniciu, V. Ramesh, and P. Meer, "Real-Time Tracking of Non-Rigid Objects Using Mean Shift," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 2, pp. 142-149, 2000.

[61] S.K. Zhou and R. Chellappa, "From Sample Similarity to Ensemble Similarity: Probabilistic Distance Measures in Reproducing Kernel Hilbert Space," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 28, no. 6, p. 917-929, June 2006.

[62] C. Bjornsson, S.J. Oh, Y.A. Al-Kofahi, Y.J. Lim, K.L. Smith, J.N. Turner, S. De, B. Roysam, W. Shain, and S.J. Kim, "Shape- and Insertion Rate Dependent Tissue Damage Due to Neuroprosthetic Device Insertion," *J. Neural Eng.,* vol. 3, pp. 196-207, 2006.

[63] Q. Gao, M. Li, and P.M.B. Vitanyi, "Applying MDL to Learning Best Model Granularity," *Artificial Intelligence,* vol. 121, pp. 1-29, 2000.

[64] G. Jacobson, B. Schnapp, and G.A. Banker, "A Change in the Selective Translocation of the Kinesin-1 Motor Domain Marks the Initial Specification of the Axon," *Neuron,* vol. 49, pp. 797-804, 2006.

[65] A.R. Cohen, C. Bjornsson, S. Temple, G. Banker, and B. Roysam, "Automatic Summarization of Changes in Image Sequences using Algorithmic Information Theory," *Proc. Fifth IEEE Int'l Symp. Biomedical Imaging,* 2008.

Andrew R. Cohen received the BSEE, MSCE, and PhD degrees from Rensselaer Polytechnic Institute in 1989, 1991, and 2008, respectively. He is currently an assistant professor in the Department of Electrical Engineering and Computer Science at the University of Wisconsin, Milwaukee. His research interests include image sequence analysis, algorithmic information theory, spectral methods, multisensor fusion, and supercomputer applications. He is a member of the IEEE and the IEEE Computer Society.

Christopher S. Bjornsson received the BSc degree in genetics and the PhD degree in developmental cell biology from the University of Manitoba, Winnipeg, Manitoba, Canada, in 1995 and 2003, respectively. He was a postdoctoral fellow in neurobiology in William Shain's Laboratory, Wadsworth Center, New York State Department of Health, Albany. He is currently the director of the Microscopy and Imaging Core Facility at the Center for Biotechnology and Interdisciplinary Studies, Rensselaer Polytechnic Institute, where he is also a research assistant professor in biology. His research is focused on describing neurovascular responses to neural prosthetic devices and evaluating the efficacy of local drug delivery in treating the reactive responses to these devices. He received Canadian NSERC Postgraduate Scholarships, as well as departmental and faculty awards from the University of Manitoba for excellence in teaching. He is a member of the Societies for Neuroscience and Cell Biology.

Sally Temple received the BA degree in zoology from the University of Cambridge and the PhD degree in developmental neurobiology from the University College London, United Kingdom. She is the scientific director of the New York Neural Stem Cell Institute and a professor of neuropharmacology and neuroscience at Albany Medical College, Albany, New York. Her laboratory is currently exploring how mouse stem cells produce diverse neurons in temporal order, mechanisms of neural stem cell self-renewal and asymmetric cell division, and the role of endothelial-derived bioactive factors in the stem cell niche. She is a board member of the International Society for Stem Cell Research and is on the editorial board for *Cell and Stem Cell*. She is the recipient of the Schaffer Award (Albany Medical College), Klingstein Award in the Neurosciences, and Jacob Javits Merit Award.

Gary Banker received the BS degree in mathematics and psychology from the University of Washington, Seattle, in 1968 and the PhD in neuroscience from the University of California, Irvine, in 1973. He held faculty positions at Albany Medical College and the University of Virginia before joining Oregon Health and Science University (OHSU) in 1998. In 2008, he was appointed as the senior scientist in the Jungers Center for Neuroscience Research, a new institute devoted to research on neurodegenerative disease and neural regeneration. He directs OHSU's Multidisciplinary Neuroscience Training Program, serves on the executive committee for the Nanobiotechnology Center, Cornell University, and is a member of the editorial board of *Brain Cell Biology*. His research focuses on the development and maintenance of neuronal polarity, using novel methods for visualizing neuronal development and protein trafficking in living neurons in culture. His current projects aim to elucidate the molecular machinery that underlies the fidelity of protein targeting and its role in the specification of polarity during neuronal development. He is a recipient of a Javitz Award for his research in neuroscience. He is a member of the Society for Neuroscience and the American Society for Cell Biology.

Badrinath Roysam received the BTech degree in electronics engineering from the Indian Institute of Technology, Madras, India, in 1984 and the MS and DSc degrees from Washington University, St. Louis, in 1987 and 1989, respectively. He has been with Rensselaer Polytechnic Institute, Troy, New York, since 1989, where he is currently a professor in the Electrical, Computer, and Systems Engineering Department. He is an associate director of the Center for Subsurface Sensing and Imaging Systems (CenSSIS)—a multiuniversity NSF-sponsored engineering research center—and a codirector of the Rensselaer Center for Open Source Software. He is an associate editor for the *IEEE Transactions on Biomedical Engineering* and the *IEEE Transactions on Information Technology in Biomedicine*. He is a senior member of the IEEE and a member of the IEEE Computer Society, the Microscopy Society of America, and the Society for Neuroscience.