Topology-based signal separation

V. Robins
Department of Applied Mathematics, Research School of Physical Sciences and Engineering, The Australian National University, Canberra, ACT 0200 Australia

N. Rooney and E. Bradley
Department of Computer Science, University of Colorado, Boulder, Colorado 80309-0430

(Received 8 April 2003; accepted 25 February 2004; published online 19 May 2004)

Traditional noise-filtering techniques are known to significantly alter features of chaotic data. In this paper, we present a noncausal topology-based filtering method for continuous-time dynamical systems that is effective in removing additive, uncorrelated noise from time-series data. Signal-to-noise ratios and Lyapunov exponent estimates are dramatically improved following the removal of the identified noisy points. © *2004 American Institute of Physics.*
[DOI: 10.1063/1.1705852]

Traditional linear or Fourier-based schemes for removing noise are unsuitable for filtering chaotic signals because they remove all modes in some interval of the frequency spectrum. Since chaos is characterized by a broad frequency spectrum, this process inevitably destroys part of the dynamically relevant signal. The dynamical systems community has developed a variety of nonlinear filtering methods that exploit the state-space *geometry* of these systems in order to remove noise without disturbing the signal. This paper proposes a nonlinear filtering method that is based on the characteristic *topology* of an attractor of a continuous-time dynamical system. In particular, such objects are known to be *perfect*, that is, they contain no *isolated points*. In practice, the numerical representation of an orbit fails to be perfect because of finite sampling and/or additive noise. We use a variable-resolution form of topological analysis to find and remove any noisy points. Experiments with clean Lorenz data contaminated with various forms of noise show highly encouraging results: the topology-based filter removes 96–100% of the noisy points, improving the signal-to-noise ratio (SNR) from ≈20 dB to more than 50 dB, with a false-positive rate of 1.9–2.5%. Estimates of the largest Lyapunov exponent in these data are dramatically improved following the removal of the identified noisy points: the noisy data have λs on the order of 10–100, while λs of the filtered data are 2–4. Data from a laboratory apparatus—a parametrically forced pendulum—showed equally encouraging results, though of course one cannot quantify percentages or SNR in real-world data. Importantly, the method proposed here is not intended to apply to, and does not work for, any kind of discrete-time dynamical system or *map*, and it requires that the time series be oversampled. These are not unrealistic assumptions; most real world experiments involve "a time series which is oversampled from a continuous flow but whose measurement is contaminated with uncorrelated additive noise...this situation is so common in physics experiments..." [Theiler and Eubank (1993)]. The key feature of our method is its identification of separation of scale. Since separation of scale is fundamental to many other forms of signal that one might be interested in untangling, this method is by no means limited to dynamical systems—or to filtering applications.

I. INTRODUCTION

Removing noise from chaotic data is highly problematic. Chaotic behavior is both broad band and sensitively dependent on system state, so traditional filtering schemes—which simply remove all signal in some band of the power spectrum—can alter important features of the dynamics, and in a significant manner. Many authors have recognized this, see Theiler and Eubank (1993) for a good synopsis. A variety of schemes have been proposed for working around this limitation; see Chap. 7 of Abarbanel (1995) for a review. A few of these rely on variations of traditional linear filters [Landa and Rozenblum (1989), Ortega (1995), Piccardi (1996), Pikovsky (1986), Schreiber (1993)], but the majority use nonlinear approaches. Specific techniques vary, depending on how much is known about the problem at hand, but the basic idea is to exploit the fact that deterministic dynamics evolves upon smooth submanifolds of state space. One family of noise-reduction methods is based on local approximations to these submanifolds. Farmer and Sidorowich (1988), for instance, use the stable and unstable manifolds of the dynamical system, via forward- and backward-time simulation and an averaging scheme, to reduce additive noise. Kostelich (1992) uses a different geometric property of the dynamics, linearizing around saddle points, where orbits are recurrent. There are many other approaches in this family; see, for example, Hammel (1990) and Kostelich and Yorke (1988). Another family of methods projects the noisy vectors onto carefully chosen subspaces, iterating the procedure until the results settle down to what is presumably the true dynamics [Cawley and Hsu (1992a, 1992b), Sauer (1992), Schreiber and Grassberger (1991)]. If measurements of more than one state variable are available, this process can be streamlined [Hegger and Schreiber (1992)] if one has the full state vectors and also knows the dynamics, one can avoid many of the

mathematical gymnastics described above, along with their attendant numerical sensitivities [Bröcker and Parlitz (2001) and Davies (1992)]. Probability and statistics can also be useful filtering techniques when one knows the dynamics, as described in Marteau and Abarbanel (1991).

The success of these methods amply demonstrates that the characteristic *geometry* of the state space of a dynamical system can be a useful basis for a dynamically meaningful noise-filtering scheme. Filtering approaches that rest upon *topology* can be equally powerful, but have seen far less investigation. State-space attractors (that is, ω-limit sets) of continuous-time dynamical systems are perfect sets [since a point in the attractor either lies on an orbit trajectory (and is therefore the limit of nearby points in that trajectory) or is the limit point of a sequence of trajectories]. A perfect set is one with no *isolated points*, i.e., no points x for which there is a $\delta>0$ such that no other point from the set lies in a ball of radius δ about x. In practice, the numerical representation of an orbit fails to be perfect because of finite sampling and/or additive noise. Provided that there is a separation of scale between the sampling effects and the additive noise, one can effectively identify isolated points and conclude that they are noisy. This idea is the topic of this paper.

II. APPROACH AND EVALUATION

The broad field of topology is generally concerned with the features of an object that are invariant under deformations that stretch and twist but never tear or glue the object. The ideas and techniques described in this paper are based in the subset of that field that addresses *connectedness*, the topological concept that captures the notion of continuity of a space. An object—more formally, a *subspace*—is connected if it cannot be decomposed into two nonempty open sets. Note that these two sets are open in the subspace topology (i.e., the intersection of an open set from the parent space with the subset of points in the subspace). Because experimental data are not infinite in quantity or precision, however, we cannot simply use the traditional topological definitions of connectedness to assess real-world data. Rather, we must reformulate those notions, as described in Robins (2000) and Robins *et al.* (1998, 2000), to fit the discrete nature of the underlying space (e.g., the space of floating-point numbers on a computer, or the space of measurements made by a sensor that has 1 mV of precision). The roots of this approach lie in Cantor's early work, which defines two points as *epsilon connected* if they are joined by an *epsilon chain*: a finite sequence of points $x_0 \cdots x_N$ that are separated by distances of epsilon or less: $|x_i - x_{i+1}| < \epsilon$. This provides an easy way to make explicit the finite precision of real data, to formulate useful definitions of topological properties that make sense at variable resolutions, and to deduce the topology of the underlying set from the limiting behavior of those properties.

In this paper, we use several of the fundamental quantities defined in Robins (2000) and Robins *et al.* (1998, 2000), most importantly the number $C(\epsilon)$ of the epsilon-connected components in a set and the number $I(\epsilon)$ of epsilon-isolated points. An epsilon-connected component is a maximal epsilon-connected subset; an epsilon-isolated point is an epsilon-component consisting of a single point. As established in Robins *et al.* (1998), one can compute C and I for a *range* of epsilon values, and deduce the topological properties of the underlying set—in this case, the true, underlying orbit of the dynamical system—from the patterns in the C and I curves. Figure 1 demonstrates the basic ideas. The point-set data shown in part (a) form a single ϵ-connected component for $\epsilon > \epsilon^*$, where ϵ^* is the largest interpoint spacing, as shown in Fig. 1(b). In this case, $C(\epsilon)=1$. If ϵ is slightly less than ϵ^*, $C(\epsilon)=2$; the corresponding ϵ-components are shown in part (c). As ϵ shrinks further, successively closer point pairs are resolved, and $C(\epsilon)$ increases, eventually flattening out at $C(\epsilon)=N$ for sufficiently small values of ϵ, where N is the number of points in the data set. The precise manner of that increase depends upon the connectedness properties and fractal dimension of the underlying set, as well as the distribution of data points over the object; this is described briefly in Robins (2000) and covered in more depth in Robins *et al.* (2000). Briefly speaking, $C(\epsilon)=1$ for $\epsilon > \epsilon^*$ if the underlying set is connected, then rises smoothly and sharply with decreasing ϵ because smaller ϵ values allow successively closer point pairs to be resolved. This behavior is shown in Fig. 1(d). If the underlying set is a totally disconnected fractal, like the set in part (e) of the figure, $C(\epsilon)$ rises in a stair-step fashion because of the scaling of the gaps in the set. The number of ϵ-isolated points is closely related to C. A point becomes ϵ-isolated when ϵ decreases past the distance to its nearest neighbor. If the data approximate a perfect set, $I(\epsilon)$ behaves in a similar manner to $C(\epsilon)$ for small ϵ; in particular $\lim_{\epsilon \to 0} C(\epsilon) = \lim_{\epsilon \to 0} I(\epsilon) = N$, where N is the number of points in the data set. This result was proven by Penrose and Yukich for the case $N \to \infty$ [Penrose and Yukich (2001)]. For larger values of ϵ, a perfect set will have $I(\epsilon)=0$. Note that $I(\epsilon) \leqslant C(\epsilon)$ for all ϵ. If $I(\epsilon)=0$ and $C(\epsilon)>0$, this implies the existence of some number of distinct connected components in the data.

The computer implementation of these calculations relies on constructs from discrete geometry called the minimal spanning tree (MST) and the nearest neighbor graph (NNG). The former is the tree of minimum total branch length that spans the data; see Fig. 1(b) for an example. To construct the MST, one starts with any point in the set and its nearest neighbor, adds the closest point, and repeats until all points are in the tree. {This is essentially Prim's algorithm [Cormen *et al.* (2001)]. More specifically, it begins with any vertex as the root and grows the MST in stages, adding at each stage an edge (x, y) and vertex y to the tree if (x, y) is minimal among all edges where x is in the tree and y is not.} The nearest-neighbor graph or NNG is a directed graph that has an edge from x_A to x_B if x_B is the nearest neighbor of x_A. To construct it, one starts with the MST and keeps the shortest edge emanating from each point. Both algorithms may be easily implemented in R^d; the computational complexity of the MST is $O(N^2)$ in general and $O(N \log N)$ in the plane, where N is the number of data points. Once these graphs are constructed, computing C and I is easy: one simply counts edges. $C(\epsilon)$, for example, is one more than the number of MST edges that are longer than ϵ, and $I(\epsilon)$ is the number of

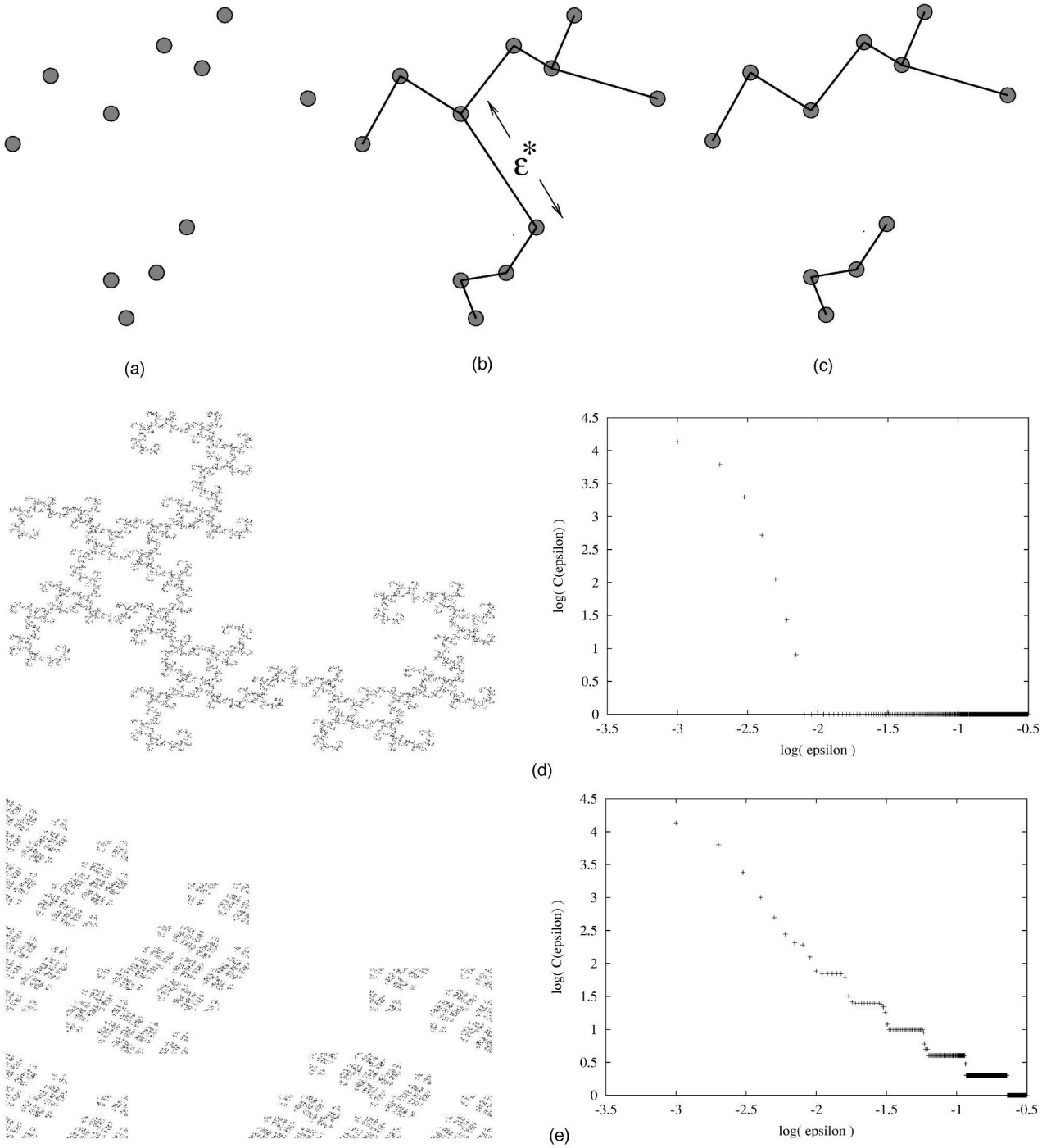

FIG. 1. Computing connectedness: (a) point-set data and (b) the *minimal spanning tree* whose edges connect nearest neighbors in that data. If $\epsilon > \epsilon^*$—the largest interpoint gap in the set—all of the points are ϵ-connected (that is, the number of connected components $C=1$); if ϵ is slightly less than ϵ^*, the set contains two connected components, as shown in part (c). The behavior of C as a function of ϵ reflects the topology of the underlying set—the object of which these points are samples. If that set is connected, $C(\epsilon)$ will fall off smoothly with decreasing ϵ, as successively closer point pairs are resolved; see part (d). If the set is a disconnected fractal, as in part (e), $C(\epsilon)$ falls off in a stair-step fashion because of the scaling of the gaps in the data.

NNG edges that are longer than ϵ. Note that one must count NNG edges with multiplicity, since x_A being x_B's nearest neighbor does not imply that x_B is x_A's nearest neighbor (i.e., if a third point x_C is even closer to x_A). Note, too, that the MST and NNG need only be constructed once; all of the C and I information for different ϵs is captured in their edge lengths.

These reformulations and algorithms allow one to assess the state-space topology of a dynamical system—a fundamental and meaningful property—even though the orbits involved are quantized in space and time by the finite resolution of sensors and computers. As discussed in Robins (2000), this can be useful in both discrete- and continuous-time systems; if the system is a flow, however, topology can

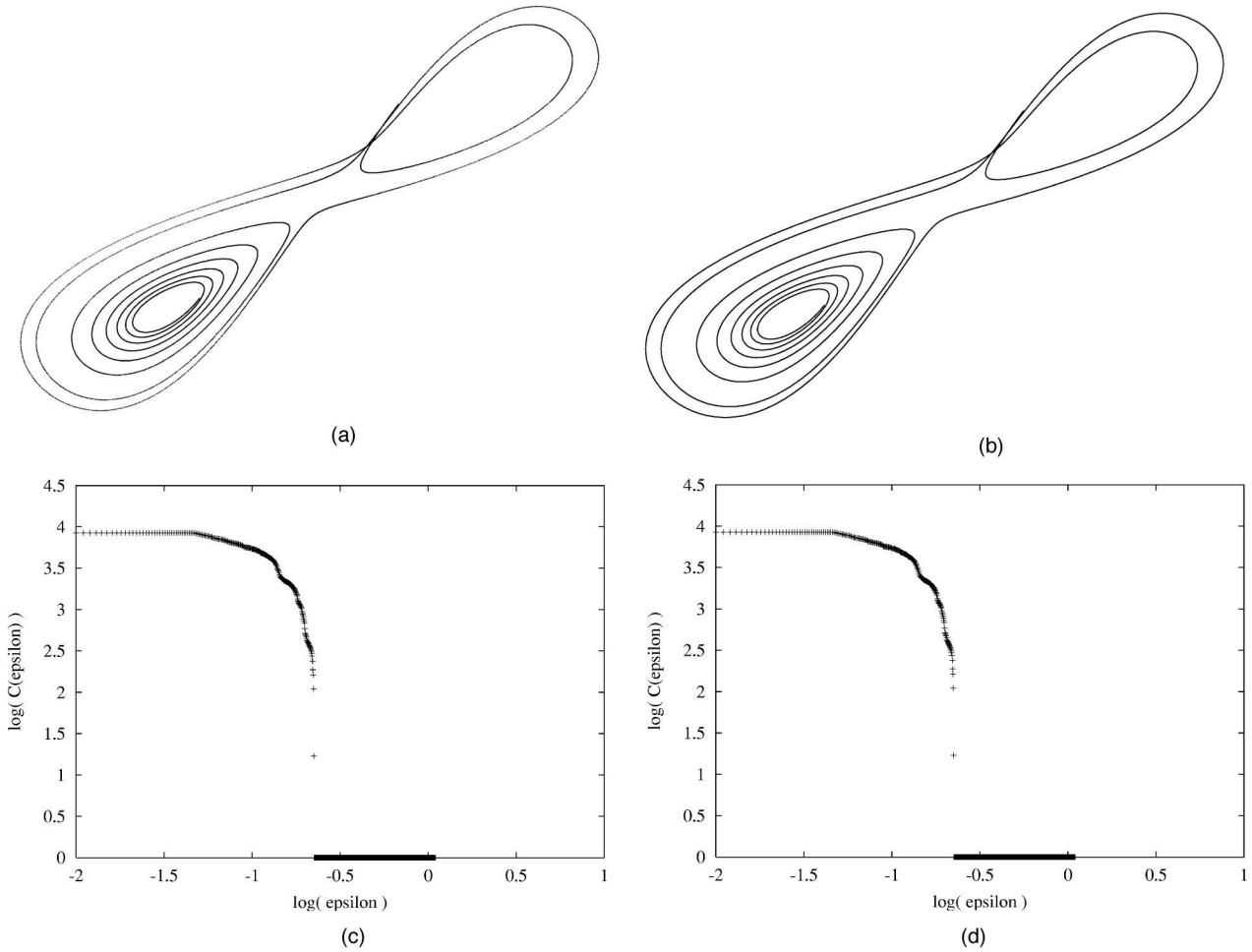

FIG. 2. Topology of the Lorenz attractor: (a) trajectory reconstructed via delay coordinate embedding from the x component of a 9000-step fourth order Runge–Kutta integration of the Lorenz equations, starting from the initial condition $(x,y,z)=(-11,-12,45)$, with $a=16$, $r=50$, $b=4$, and $\Delta t=0.0005$. Embedding parameters were $m=7$ and $\tau=0.05$; in this 2D rendering, $x(t+\tau)$ is plotted against $x(t)$. The MST of this trajectory, shown in part (b), is visually indistinguishable from the orbit shown in part (a) because of the small time step. The $C(\epsilon)$ and $I(\epsilon)$ plots in parts (c) and (d) indicate that the underlying orbit is connected and perfect.

be even more powerful, as the associated attractors are not only connected, but also perfect. Figure 2 shows MST-based connectedness results for the canonical Lorenz system. The C and I curves exhibit the classic smooth, sharp falloff that indicates connectedness and perfectness, respectively. This behavior is affected by the sampling: the falloff is less sharp for data that are sparsely sampled or nonuniform. As explained above, $C(\epsilon)$ and $I(\epsilon)$ results are virtually identical for the connected sets that we study in this paper. The latter carries the information that we want, as we are looking for lapses in perfectness, so we only plot $I(\epsilon)$ hereafter. Incidentally, the MST and NNG constructs contain other information that is useful for dynamical analysis. Their branching structure can be used to identify orbit types in dynamical systems [Yip (1991)] and discontinuities in bubble-chamber tracks [Zahn (1971)], as discussed further at the end of this paper, and they are widely used in the kinds of clustering tasks that arise in pattern recognition [Duda and Hart (1973)] and computer vision [Ballard and Brown (1982)].

Because orbits of continuous-time dynamical systems are, in theory, perfect sets, any isolated points on such an orbit are an aberration. Noise is one potential cause of this; consider an intermittent glitch in a sensor (or an error in an algorithm) that adds noise to some subset of points in the orbit—a situation, as indicated in the lead paragraph, that is quite common in physics experiments. In order to explore this effect, we added noise to the time-series data from Fig. 2, redid the embedding, repeated the topological analysis, and observed the effects on the $I(\epsilon)$ plots. Figure 3 shows a representative set of results. The spanning tree clearly brings out the displacement of the noisy points from the rest of the orbit: if the magnitude of the noise is large compared to the interpoint spacing, the edges joining the noisy points to the rest of the tree are longer than the original edges, which creates an extra shoulder on the I and C curves. Of course, if the magnitude of the noise is *small* compared to that spacing, the associated MST edges will not be unusually long, and so the I and C curves will not have an extra shoulder. Figure 4 demonstrates this, showing $I(\epsilon)$ plots for different magnitudes and types of noise. In particular, part (a) of the figure shows the effects of *constant* noise, where a fixed value $(\pm n)$ was added to roughly 1% of the original time-series

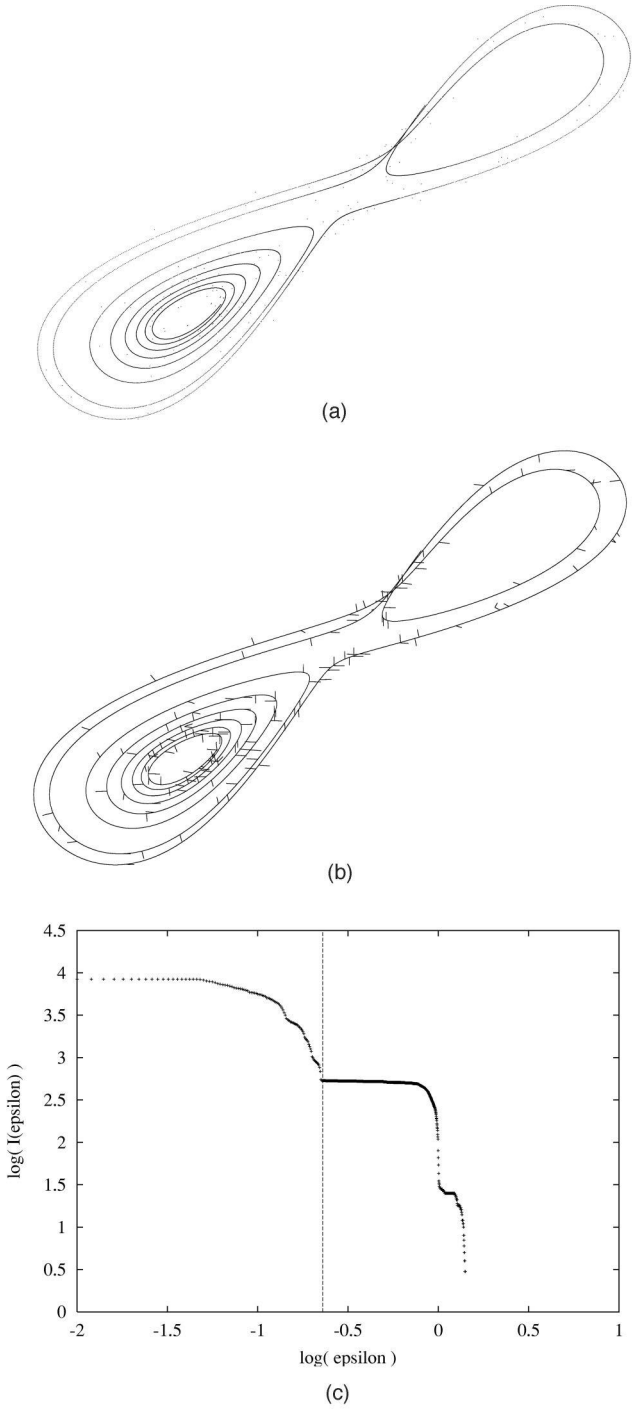

FIG. 3. The effects of noise upon attractor topology. Each point in the original Lorenz time series from the previous figure was perturbed, with probability 0.01, by a value of ± 1.0, which is roughly 2% of the overall width of the reconstructed attractor. These noise-added data were then embedded to obtain the trajectory shown in (a). The minimal spanning tree of this set, pictured in (b), clearly shows the noisy points, as does the $I(\epsilon)$ plot in part (c), where the noise adds a shoulder to the curve of Fig. 2(d). ϵ_n can be interpreted as approximating the maximum edge length of the MST of the non-noisy data.

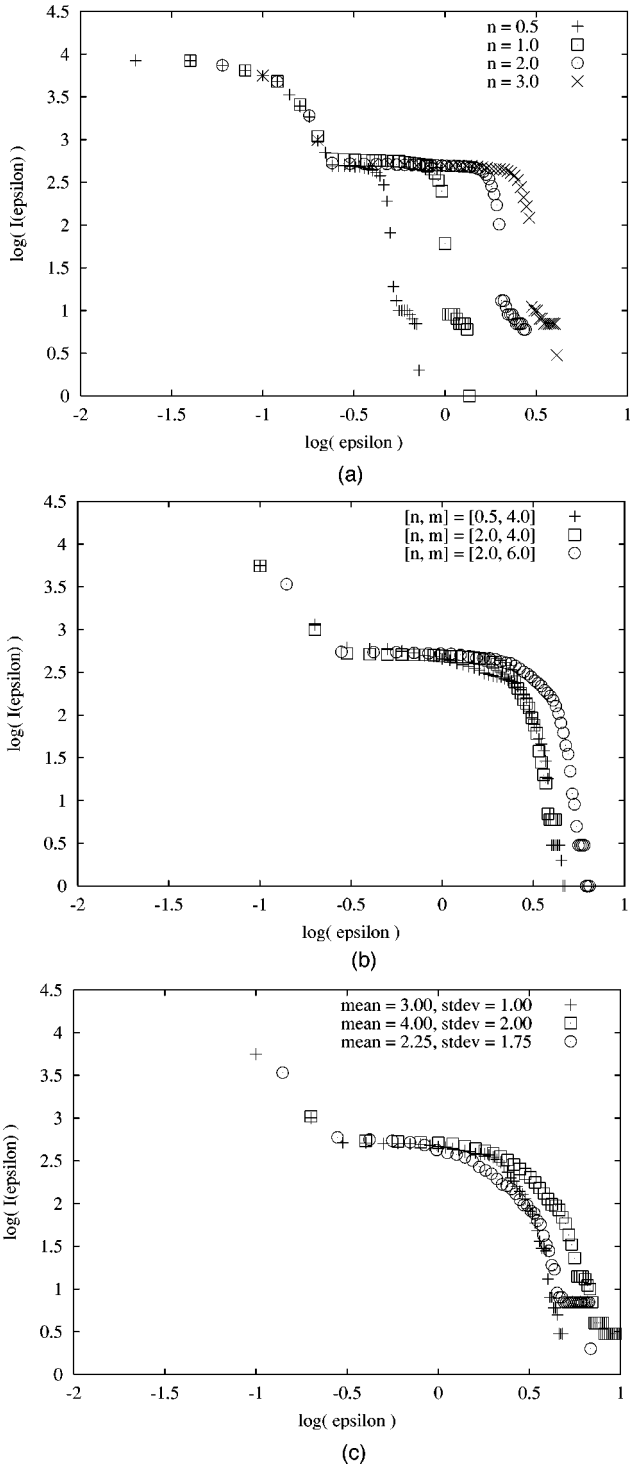

FIG. 4. Effects of noise distribution upon $I(\epsilon)$. (a) Constant noise: $\pm x_{\text{noise}} = n$ was added to each x in the original Lorenz time series, with probability 0.01, for various n. (b) Uniform random noise: $\pm x_{\text{noise}} \in [n, m]$ was added to each x, with probability 0.01, for various n and m. All values in $[n, m]$ were chosen with equal probability. (c) Gaussian noise: $\pm x_{\text{noise}}$ was added to each x, with probability 0.01. The mean and standard deviation of x_{noise} were varied as shown. In all cases, noise adds recognizable shoulders to the plots.

points. The width of the shoulder increases with n, reflecting the wider distribution of MST edge lengths that results when the noise-added points are embedded. (Note that any or all of the coordinates of a given embedded point may include noise, so points do not simply move $\pm n$ along one axis of the reconstruction space.) Different types of noise, e.g., *uniform*, where we added some number between n and m, for different $[n, m]$, or *Gaussian*, with different σs and $\overline{x}$s, make the shape of the falloff slightly more irregular, but do not

change the general shape of the shoulder. In all cases, the larger the noise, the wider the shoulder. Again, this makes sense; larger noise values push points further from their original position on the true orbit, which means that they will remain ϵ-isolated until ϵ is quite large. (This is equivalent to the claim above about the effects of ''small'' noise on the MST.) Of course, the *direction* of the noise vector also matters. If a noise-added point happens to fall near another point, e.g., if the noise vector is along the trajectory, rather than in a transverse direction, the associated MST edge will not distinguish it from the non-noisy points. There are some ways around this, as described later in this paper. This method's reliance on separation of scale makes it powerful, but does give it an Achilles heel: if there is a *continuous spread* of additive noise, there is no longer a separation of scale, and so the MST cannot identify noisy points. Nonetheless, the noisy data sets in Fig. 4 are representative of a broad class of potential disturbances in physical experiments; they contain instances of the pathological, difficult-to-detect cases mentioned above—small noise magnitudes and along-the-orbit noise—and so they provide a useful set of test cases for this paper. Incidentally, we use embedded data in these examples, rather than the full state-space trajectories produced by the integrator, because we were interested in evaluating the utility of these techniques for experimental data, and dynamical systems are rarely observable in practice. (In what control theory calls an *observable* system, one can either measure or deduce values for every state variable.)

The obvious differences between the $I(\epsilon)$ curves in Figs. 2 and 3 suggest a topology-based filtering scheme. Specifically, a breakpoint in the falloff region of the curve (indicated by ϵ_n in Fig. 3), followed by a second hump, indicates that there is a scale separation in the data set. That is, the MST edge length distribution has two peaks, one below ϵ_n and one above it. Such a separation of scale can arise if two processes are at work in the data—such as signal and noise. If the noise is large compared to the sample spacing of the data, we can take advantage of that scale separation in order to disentangle the two. The breakpoint ϵ_n, in this case, can be interpreted as approximating the maximum edge length of the MST of the non-noisy data. At that value, most of the noisy points—and few of the regular points—are ϵ-isolated. One can easily discard the noisy points by pruning the appropriate edges of the MST: those that are longer than the breakpoint value ϵ_n. Figure 5 shows the results of this filtering technique, as applied to the data from Fig. 3. Using $\epsilon = 0.3$ (slightly above the ϵ_n breakpoint from Fig. 3) to prune the minimal spanning tree, this method removed 534 of the 545 noisy points and 150 of the 7856 non-noisy points. This translates to 98.0% success with a 1.9% false positive rate, and a signal-to-noise ratio [SNR $= 20 \log_{10}(\text{signal/noise})$] reduction from 23.2 dB to 56.9 dB. These rates vary slightly for different types and amounts of noise, but the success and false-positive percentages remain close to 100% and 0%, respectively, and the SNR of the filtered data is substantially larger than that of the noisy data; see Table I. These are promising numbers—comparable to or better than the existing filtering schemes described in the Introduction. Increasing the pruning length, as one would predict, decreases the false-positive rate; somewhat less intuitively, though, larger pruning lengths do *not* appear to significantly affect the success rate—until they become comparable to the length scales of the noise. Incidentally, the exact details of the pruning algorithm are somewhat more subtle than is implied above because not all noisy points are terminal nodes of the spanning tree. Thus, an algorithm that simply deletes *all* points whose connections to the rest of the tree are longer than the pruning length can sever connections to other points, or clusters of points. This is an issue if one noisy point creates a ''bridge'' to another noisy point and only one of the associated MST edges is longer than ϵ_n. Lastly, note that while noise was added to each point in the scalar time-series data with probability 0.01, each of those data points is a coordinate of m points in an m-dimensional embedding, so roughly m% of the points in the *embedded* trajectory are noisy.

Another way to evaluate this filtering method is to use dynamical invariants, e.g., to compare the Lyapunov exponent λ of the original, noise-added, and filtered trajectories. Table II shows these results, calculated with an IDL implementation of Wolf's algorithm [Wolf (1986)]. As is well known, noise not only increases the dimension of an orbit, but also affects its λ. This effect, which is described nicely in Brown (1993) and Bryant and Brown (1990), is abundantly clear from the third column of the table. Ideally, filtering out the noise would reduce λ back to the original value. In practice, all of these calculations—and the comparison to the λ of the original trajectory— are somewhat problematic, as numerical algorithms for calculating λ are notoriously sensitive to orbit length, initial conditions, and other dynamical, algorithmic, and computational parameters. For instance, if the initial conditions of the trajectory used to generate Fig. 2(a) are varied slightly—± 0.1 on each of the three state variables x, y, and z—the calculated λ ranges from 0.125 to 0.422, in spite of the temporal averaging performed by Wolf's algorithm. If the initial condition is fixed at the values used in Fig. 2 and the orbit *length* is varied from 5000 to 30000, the calculated λ ranges from 0.11 to 0.86. The λ of the *embedded* trajectory—which should be equal to the original λ if the embedding dimension is adequate—varies as shown in Table III. Since the dimension of the Lorenz system is three, the Whitney/Mañé/Takens conditions require that $m \geqslant 7$, in the worst case, for a successful embedding, though Sauer *et al.* suggest that $m \leqslant d_A$, where d_A is the box-counting dimension, is sufficient [Sauer *et al.* (1991)]. The λ values in the table do indeed appear to settle out—around $m = 6$ or 7—but to a value that is significantly lower than those calculated for the full *xyz* trajectory. All of this variation makes it hard to know what to compare the values in the rightmost column of Table II against. Furthermore, *removing* points affects the λ calculation. Wolf's algorithm proceeds by identifying a point's nearest neighbor, then following the point pair until their spacing exceeds a heuristic threshold, then renormalizing by finding a new near neighbor (in the direction of the vector of the last separation), and repeating to the end of the trajectory. Removing points from that trajectory will necessarily force the near-neighbor search to return artificially distant results, with unpredictable effects upon the calculated λ.

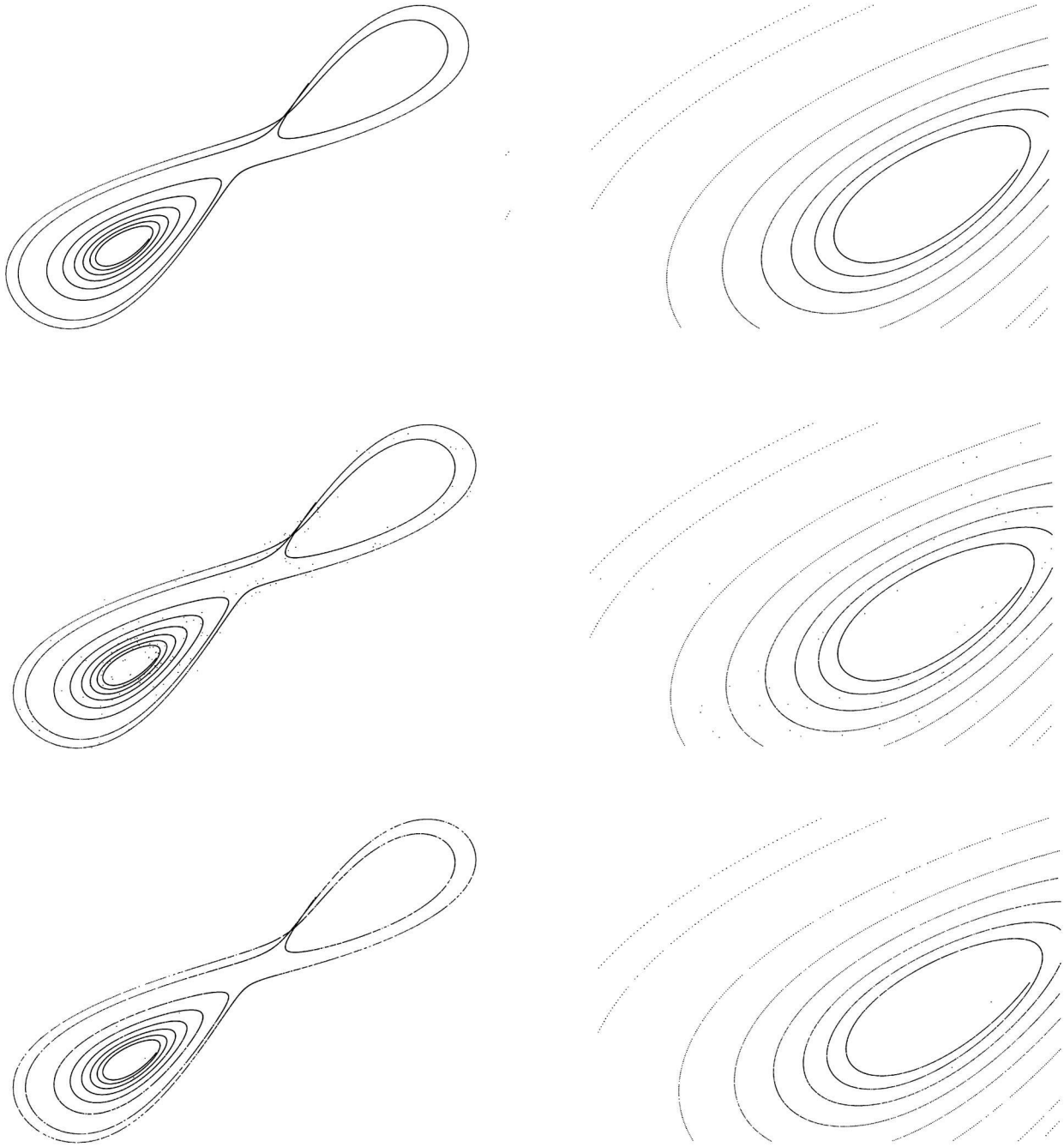

FIG. 5. Topology-based filtering of the Lorenz data: the original data [cf. Fig. 2(a)] are shown at the top and the noise-added version of that data from Fig. 3(a) appears in the middle. The filtered data in the bottom image were obtained by pruning the MST of Fig. 3(b) using $\epsilon = 0.3$—a value just above the ϵ_n breakpoint in part (c) of Fig. 3. Using this pruning value, the topology-based filtering algorithm removes 98% of the noisy points, with a false-positive rate of 1.9%.

To explore this effect, we randomly removed 700 of the points in the trajectory of Fig. 2—the same number that were removed, on the average, in our pruning experiments—and reran the calculation. Over ten such trials, λ ranged from 1.9 to 2.7, so it appears that removing points, in general, raises λ. All of these experiments make it clear that the error bars of Wolf's algorithm, especially as applied to data sets with pointwise gaps, are substantial. [This is not a new result, nor a unique feature of this algorithm; Brown (1993) discusses this issue, comparing and contrasting several λ-calculation algorithms.] In this context, it appears that our filtering method works very well, though not perfectly: the λ values for the filtered data are dramatically smaller than those for the noisy data, and not hugely different from the range of "true" λ values. This corroborates the success/false positive percentages given in the previous paragraph, and confirms that this topology-based filtering scheme is indeed effective at removing noise without disturbing the dynamics.

Removing noise that one has artificially added to a trajectory is a useful first test, but it is certainly not the intended

TABLE I. Effect of noise on the filtering algorithm. Noise parameters as in previous figure; pruning length $=0.3$ throughout. Signal-to-noise ratio $\mathrm{SNR}=20\log_{10}(\mathrm{signa/noise})$.

Noise type	Parameters	SNR of noisy data (dB)	% Noisy points removed	SNR of filtered data (dB)	% Non-noisy points removed
Constant	0.5	23.8	100	∞	2.1
Constant	1	23.2	98.0	56.9	1.9
Constant	2	22.1	97.7	54.7	2.1
Constant	3	23.7	99.4	68.2	1.9
Uniform	[0.5 4]	22.3	100	∞	2.5
Uniform	[2 4]	23.2	98.7	60.8	2.1
Uniform	[2 6]	23.7	100	∞	2.1
Gaussian	$\sigma=1$, $\bar{x}=3$	23.5	100	∞	2.1
Gaussian	$\sigma=1.75$, $\bar{x}=2.25$	23.6	96.0	51.3	2.0
Gaussian	$\sigma=2$, $\bar{x}=4$	23.5	97.9	56.9	2.0

use of this filtering method. The real applications are sensor data from physical experiments. (The problems caused by floating-point arithmetic limitations are comparatively minor on modern machines.) Figure 6 shows an instance of such an application: an experimental data set from a parametrically forced pendulum. The sampling rate in this data set was much faster than the highest frequency of the dynamics, but the angle sensor was intermittently noisy, and so the data set contains some outlying points. These are difficult to discern in the trajectory of part (a) because of the scales involved, but the minimal spanning tree in part (b) makes them clearly visible. The $I(\epsilon)$ curve is somewhat more complicated than in the Lorenz examples: it has a well-defined shoulder running from $\log\epsilon\approx-1.5$ to ≈-0.5, followed by a stair-step falloff for $\log\epsilon>-0.5$. As described earlier in this paper, this kind of pattern can be an indication that the set is a disconnected fractal. In this case, however, it is probably an artifact of the quantization of the angle sensor, which can only measure to 0.7°, coupled with the effects of embedding upon noisy data (i.e., that noise can appear in more than one component of the embedding vector).

Filtering these data as in the Lorenz example, we delete all points that are ϵ-isolated, using a $\log\epsilon$ value of -1.4, which is slightly above the breakpoint ϵ_n of the $I(\epsilon)$ curve shown in part (c). The results are shown in Fig. 7. Obviously, we cannot give percentage comparisons here, as we do not know which of the original points are noisy. A visual comparison of Figs. 6(b) and 7(b), however—particularly the MST closeups—suggests that most of the noisy points have been removed; note the absence of transverse ''hairs'' in the MST of the filtered data. [Again, comparing parts (a) of these figures requires good eyesight; these plots are shown here mainly for context for the MSTs.] Moreover, the absence of a shoulder on the $I(\epsilon)$ plot in Fig. 7(d) suggests that the filtered trajectory is continuous and perfect—as it should be, given that the pendulum is a continuous-time dynamical system, and that the sampling interval is small. All in all, these results are quite promising.

Oversampling is critical to the implementation of this approach for a variety of reasons, all of which stem from the mathematics of continuity. Undersampling destroys perfectness; in sparsely sampled data, MST edges not only skip over unsampled chunks of the attractor, but can even jump crosswise from one attractor thread to another instead of connecting points along the orbit. A related issue is our simplistic filtering model: we simply *remove* the point; we do not deduce where it ''should be'' and move it in that direction. This is obviously a bad idea if the data are not oversampled. One way to improve on this filtering model would be to exploit the characteristic geometric properties of dynamical systems, as in the schemes outlined in the first paragraph of this paper, e.g., to interpolate between the two points on

TABLE II. Effects of noise and filtering on the Lyapunov exponent, as calculated by Wolf's algorithm. Noise type and parameters as in Fig. 4. The λ of the original trajectory is discussed in the text.

Noise type	Parameters	λ of noisy data	λ of filtered data
Constant	0.5	8.7	3.2
Constant	1	35.5	3.2
Constant	2	82.2	3.4
Constant	3	81.3	2.9
Uniform	[0.5 4]	68.9	2.8
Uniform	[2 4]	81.9	4.7
Uniform	[2 6]	90.7	3.0
Gaussian	$\sigma=1$, $\bar{x}=3$	73.5	2.9
Gaussian	$\sigma=1.75$, $\bar{x}=2.25$	68.2	2.5
Gaussian	$\sigma=2$, $\bar{x}=4$	78.2	3.5

TABLE III. Lyapunov exponent of delay-coordinate reconstructions of the Lorenz data from Fig. 2 with different embedding dimensions.

Embedding dimension	λ
2	0.938
3	0.685
4	0.236
5	0.114
6	0.096
7	0.082
8	0.059
9	0.096
10	0.097

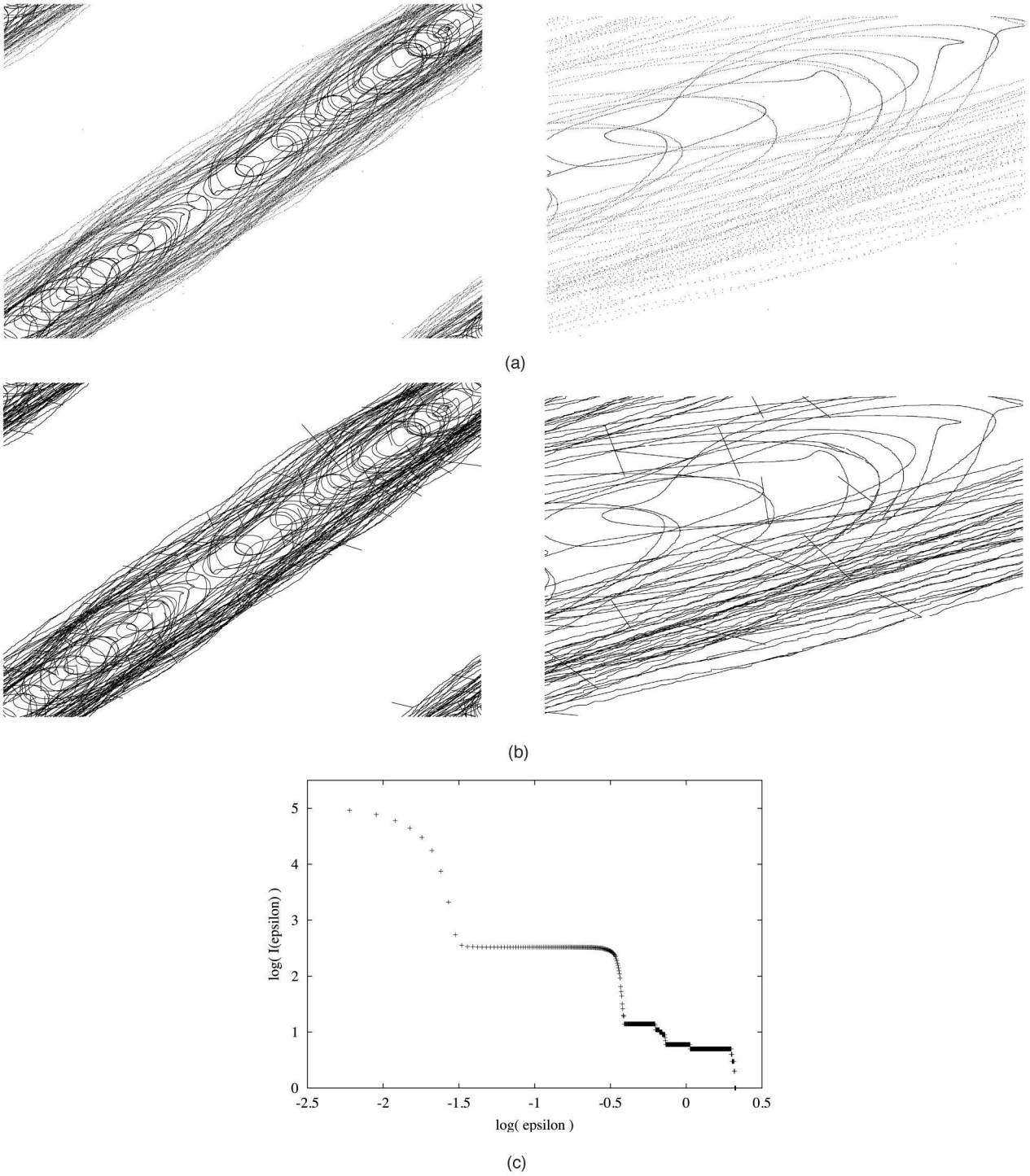

FIG. 6. Experimental driven pendulum data. (a) 99800 measurements of the bob angle θ, sampled every 265 microseconds by an optical encoder with a resolution of 0.7°, and embedded in $[0,2\pi)^7$ with $\tau=0.0265$ s. (b) MST and (c) $I(\epsilon)$ of the embedded pendulum trajectory. In these 2D renderings, $\theta(t+\tau)$ is plotted against $\theta(t)$. The stair-steps in the $I(\epsilon)$ plot for $\log \epsilon > -0.5$ are due in part to sensor quantization and in part to points where noise appears in more than one dimension of the embedding vector.

either side of the base of the edge that connects an isolated point to the rest of the trajectory. There are other ways to leverage continuity, both in identifying noisy points and reconstructing where they should have been. Any displacement transverse to the orbit, for example, is a clear suspect for noise. Zahn (1971) exploits this to prune noisy points from bubble chamber data, using the local structure of the minimal spanning tree to identify points that are not ''on the main tracks.'' In an even earlier application, Clark and Miller (1966) use MSTs to link sequences of spark-chamber images by iteratively removing the short, terminal ''hairs'' on the tree. Incorporating temporal information may also be help-

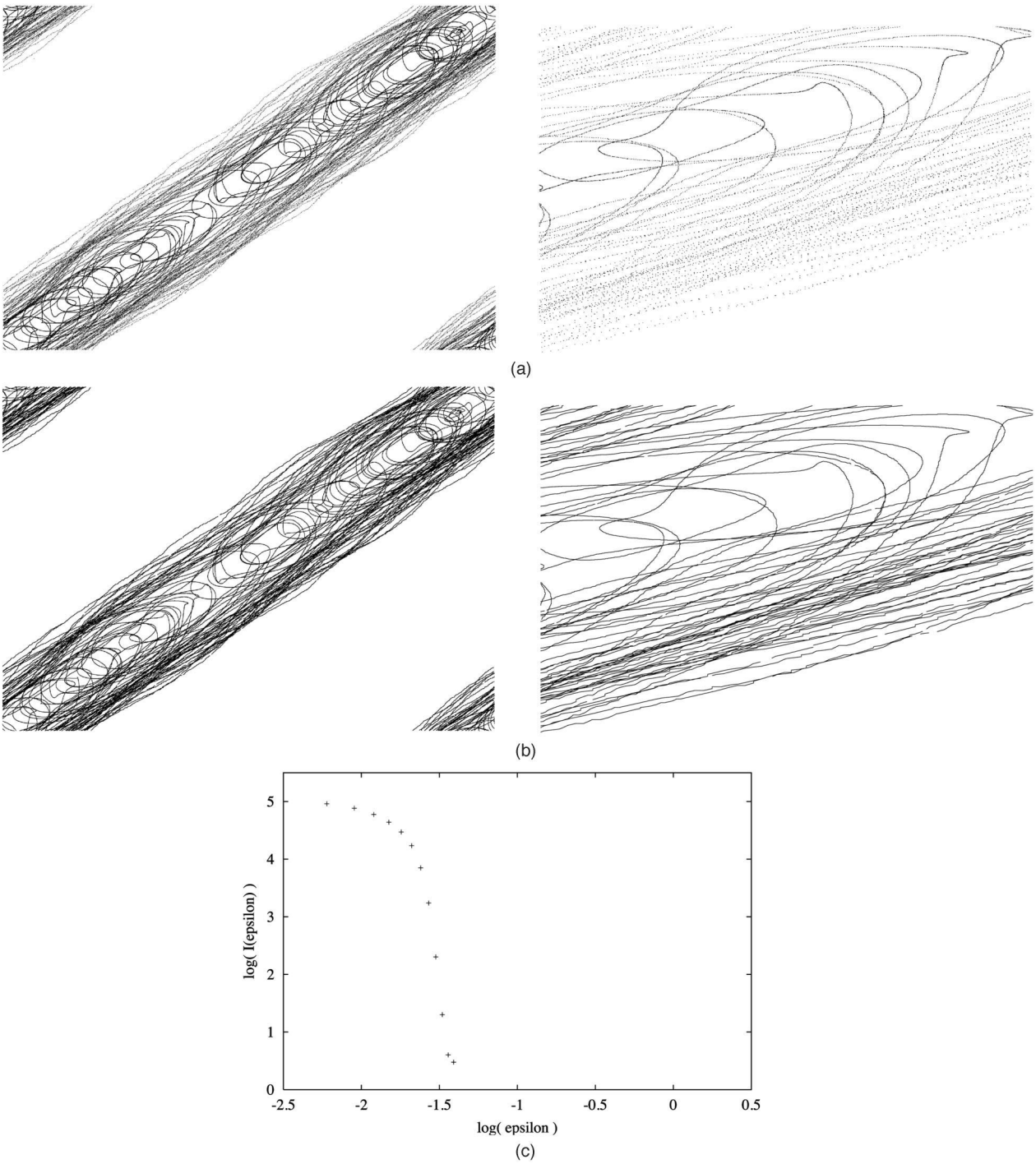

FIG. 7. Filtered pendulum data, obtained by removing all ϵ-isolated points from the data in the previous figure, with $\log \epsilon = -1.4$ chosen just above the first breakpoint of the associated $I(\epsilon)$ curve.

ful; our current scheme only uses the geometry and topology of the data, disregarding which points were close in time—which can be an effective indication of continuity (and violations thereof). Adding time as an additional dimension in the distance metric used in the MST would address this. We are working out a sensible formalization of all of these ideas within our computational topology framework, and will report upon their results in a future paper. In the meantime, this preliminary implementation works quite well, and it is generally far easier to oversample a physical system than to fix the source of the noise.

Quantization has complicated and interesting effects upon variable-resolution topology. Data quantization implies distance metric quantization, so MST edges can only take on discrete lengths. Moreover, in raster images, noise does not move points around, as it does in the examples in this paper; rather, it simply reshades pixels. The computer vision community [Ballard and Brown (1982)] distinguishes these two

kinds of noise as ''distortion'' and ''salt and pepper,'' respectively. Because the metric used in the MST captures distances between points, it is more effective at detecting the former than the latter.

III. CONCLUSION

The filtering method introduced in this paper exploits the fundamental topological properties of continuous-time dynamical systems in order to find and remove noisy points from a data set. ω-limit sets of these systems are, in theory, perfect sets, that is, they contain no isolated points. In practice, however, the numerical representation of an orbit fails to be perfect because of finite sampling and/or additive noise. The approach described here uses a variable-resolution approach to computational topology in order to work around the sampling effects and effectively identify the isolated, noisy points, and then remove them. (This only works if the underlying system is a flow, of course; orbits of maps need not be perfect.) Experiments with clean Lorenz data contaminated with various forms of noise showed encouraging results: the topology-based filter identified 96–100% of the noisy points, improving the signal-to-noise ratio from $\approx$20 dB to more than 50 dB, with a false-positive rate of 1.9–2.5%. Data from a laboratory apparatus—a parametrically forced pendulum—showed equally encouraging results, though of course one cannot quantify noise percentages or ratios in real-world data.

There have been a few other topology-based approaches to dynamical systems analysis. Our use of the minimal spanning tree as a data structure was inspired by Yip's work on automated phase-portrait analysis [Yip (1991)]. Mischaikow *et al.* (1999) use algebraic topology to construct a symbol dynamics from trajectory data. This neatly finesses the noise issue by using a coarser representation, and thus constitutes a form of filtering. Rather than use algebraic topology to construct a useful coarse-grained representation of the dynamics, our approach uses *geometric* topology to remove noisy points while working in the original space, which allows us to obtain much finer-grained results. Muldoon *et al.* compute the *homology* of embedded attractors [Muldoon *et al.* (1993)]. This approach does not address noise directly, but does describe how to build a triangulation from time-series data, and then computes many of the same topological quantities that are used in our broader work. Mees (1992) uses tesselations to find the dimension of the state space, along with information about folds and branches in the attractor, leading to a method for reconstruction of models of the dynamics. Mindlin and Gilmore also use topological techniques—templates and knots in particular—to model the stretching/squeezing dynamics on a chaotic attractor [Mindlin and Gilmore (1992)]. Lastly, approaches that are based on unstable periodic orbits [Auerbach *et al.* (1987) and Gunaratne *et al.* (1989)] could also be viewed as topological.

The key feature of the method proposed here is its identification of separation of scale: if the magnitude of the noise is greater than the interpoint spacing, the variable-resolution topological analysis described in this paper will bring that out. Conversely, noise that is of the order of the separation between the points in the data set, or less, does not produce an ϵ-isolated point, and thus cannot be identified or removed by this method. For these reasons, this method requires that the time series be oversampled, and it does not apply to discrete-time dynamical systems. Filtering schemes have different weaknesses, but all are vulnerable to small noise magnitudes, and the method described here is no exception. Like most others, too, topology-based filtering works only for *additive* noise, not dynamic noise, which is coupled back into system evolution (e.g., an error in an ODE solver, which affects the initial condition for the next step). Because of its reliance on separation of scale, too, this approach does not work if the additive noise has a continuous spread. Lastly, like most of the other methods described in the dynamical systems literature, this scheme is *noncausal*, in the sense that its algorithms require both past and future values of the trajectory in order to filter out a particular noisy point. As it is currently implemented, this method simply discards noisy points; we are working on modifications that remove that restriction by selectively adding geometric knowledge to the algorithm so it can be more intelligent about detecting and *adjusting* noisy points. Separation of scale is fundamental to many other forms of signal that one might be interested in untangling, so this method is by no means limited to dynamical systems—or to filtering applications.

ACKNOWLEDGMENTS

The IDL implementation of Wolf's algorithm was written by Ricardo Mantilla of the University of Colorado. Wheeler Ruml of PARC offered useful comments on a draft of this document.

Abarbanel, H., *Analysis of Observed Chaotic Data* (Springer, New York, 1995).

Auerbach, D., Cvitanovic, P., Eckmann, J. -P., Gunaratne, G., and Procaccia, I., ''Exploring chaotic motion through periodic orbits,'' Phys. Rev. Lett. **58**, 2387–2389 (1987).

Ballard, D. and Brown, C., *Computer Vision* (Prentice–Hall, Englewood Cliffs, NJ, 1982).

Bröcker, J. and Parlitz, U., ''Efficient noncausal noise reduction for deterministic time series,'' Chaos **11**, 319–326 (2001).

Brown, R., ''Calculating Lyapunov exponents for short and/or noisy data sets,'' Phys. Rev. E **47**, 3962–3969 (1993).

Bryant, P. and Brown, R., ''Lyapunov exponents from observed time series,'' Phys. Rev. Lett. **65**, 1523–1526 (1990).

Cawley, R. and Hsu, G., ''Local-geometric-projection method for noise reduction in chaotic maps and flows,'' Phys. Rev. A **46**, 3057–3082 (1992a).

Cawley, R. and Hsu, G., ''SNR performance of a noise reduction algorithm applied to coarsely sampled chaotic data,'' Phys. Lett. A **166**, 188–196 (1992b).

Clark, R. and Miller, W., ''Computer based data analysis systems at Argonne,'' Methods Comput. Phys. **5**, 47–98 (1966).

Cormen, T., Leiserson, C., Rivest, R., and Stein, C., *Introduction to Algorithms* (MIT Press, Cambridge, 2001), pp. 570–573.

Davies, M., ''Noise reduction by gradient descent,'' Int. J. Bifurcation Chaos Appl. Sci. Eng. **3**, 113–118 (1992).

Duda, R. and Hart, P., *Pattern Classification* (Wiley, New York, 1973).

Farmer, J. and Sidorowich, J., ''Exploiting chaos to predict the future and reduce noise,'' in *Evolution, Learning, and Cognition* (World Scientific, Singapore, 1988).

Gunaratne, G., Linsay, P., and Vinson, M., ''Chaos beyond onset: A comparison of theory and experiment,'' Phys. Rev. Lett. **63**, 1 (1989).

Hammel, M., ''A noise reduction method for chaotic systems,'' Phys. Lett. A **148**, 421–428 (1990).

Hegger, R. and Schreiber, T., ''A noise reduction method for multivariate time series,'' Phys. Lett. A **113**, 305 (1992).
Kostelich, E., ''Problems in estimating dynamics from data,'' Physica D **58**, 138–152 (1992).
Kostelich, E. and Yorke, J., ''Noise reduction in dynamical systems,'' Phys. Rev. A **38**, 1649–1652 (1988).
Landa, P. and Rozenblum, M., ''A comparison of methods for constructing a phase space and determining the dimension of an attractor,'' Sov. Phys. Tech. Phys. **34**, 1229–1232 (1989).
Marteau, P. and Abarbanel, H., ''Noise reduction in chaotic time series using scaled probabilistic methods,'' J. Nonlinear Sci. **1**, 313–343 (1991).
Mees, A., ''Tesselations and dynamical systems,'' in *Nonlinear Modeling and Forecasting*, edited by M. Casdagli and S. Eubank (Addison-Wesley, Reading, 1992).
Mindlin, G. and Gilmore, R., ''Topological analysis and synthesis of chaotic time series data,'' Physica D **58**, 229–242 (1992).
Mischaikow, K., Mrozek, M., Reiss, J., and Szymczak, A., ''Construction of symbolic dynamics from experimental time series,'' Phys. Rev. Lett. **82**, 1144–1147 (1999).
Muldoon, M., MacKay, R., Huke, J., and Broomhead, D., ''Topology from time series,'' Physica D **65**, 1–16 (1993).
Ortega, G., ''A new method to detect hidden frequencies in chaotic time series,'' Phys. Lett. A **209**, 351–355 (1995).
Penrose, M., and Yukich, J., ''Weak laws of large numbers in geometric probability,'' (preprint); math.PR/0107148, 2001.
Piccardi, C., ''On taming chaos using LTI filters,'' IEEE Trans. Circuits and Systems **43**, 431–432 (1996).
Pikovsky, A. S., ''Noise filtering in the discrete time dynamical systems,'' Sov. J. Commun. Technol. Electron. **31**, 911–914 (1986).
Robins, V., ''Computational topology at multiple resolutions,'' Ph.D. thesis, University of Colorado, June 2000.
Robins, V., Meiss, J., and Bradley, E., ''Computing connectedness: An exercise in computational topology,'' Nonlinearity **11**, 913–922 (1998).
Robins, V., Meiss, J., and Bradley, E., ''Computing connectedness: Disconnectedness and discreteness,'' Physica D **139**, 276–300 (2000).
Sauer, T., ''A noise reduction method for signals from nonlinear systems,'' Physica D **58**, 193–201 (1992).
Sauer, T., Yorke, J., and Casdagli, M., ''Embedology,'' J. Stat. Phys. **65**, 579–616 (1991).
Schreiber, T., ''Extremely simple nonlinear noise-reduction method,'' Phys. Rev. E **47**, 2401–2404 (1993).
Schreiber, T. and Grassberger, P., ''A simple noise-reduction method for real data,'' Phys. Lett. A **160**, 411–418 (1991).
Theiler, J. and Eubank, S., ''Don't bleach chaotic data,'' Chaos **3**, 771–782 (1993).
Wolf, A., ''Quantifying chaos with Lyapunov exponents,'' in *Chaos* (Princeton University Press, Princeton, 1986), pp. 273–290.
Yip, K., *KAM: A System for Intelligently Guiding Numerical Experimentation by Computer*, Artificial Intelligence Series (MIT Press, Cambridge, 1991).
Zahn, C., ''Graph-theoretical methods for detecting and describing Gestalt clusters,'' IEEE Trans. Comput. **C-20**, 68–86 (1971).