Stimulus representations that are invariant under invertible transformations of sensor data.

David N. Levin
Department of Radiology, University of Chicago

ABSTRACT

Humans have a remarkable ability to perceive the constancy of a stimulus even though its "appearance" has changed due to factors that are extrinsic to it. This paper shows how sensory devices can invariantly represent stimuli, even though their sensor states may have been transformed by factors extrinsic to the stimuli. Such transformations may be caused by changes of observational conditions such as: 1) alterations of the device's sensory apparatus, 2) changes in the observational environment external to the sensory device and the stimuli, and 3) modifications of the presentation of the stimuli themselves. The stimulus representations are invariant because they describe certain *relationships* of each sensor state to the time series of recently encountered sensor states, and these relationships are unchanged by any invertible transformation of sensor states. This paper describes three analytic methods of creating such representations, utilizing tensor calculus, affine-connected differential geometry, and Riemannian differential geometry, respectively. These techniques are illustrated by applying them to simulated data on two-dimensional sensor state manifolds. This approach may be useful for designing a representation "engine" that comprises the "front end" of an intelligent sensory device. It could create stimulus representations that are amenable to pattern analysis because they are not affected by many factors that are extrinsic to the stimuli.

Keywords: sensor, channel, distortion, calibration, pattern recognition, computer vision, speech recognition

1. INTRODUCTION

Most intelligent sensory devices contain pattern recognition software for analyzing the state of the sensors that detect stimuli in the device's environment. This software is usually "trained" to classify a set of sensor states that are representative of the "unknown" sensor states to be subsequently encountered. For instance, an optical character recognition (OCR) device might be trained on letters and numbers in images of printed pages. Or, a speech recognition device may be trained to recognize the words of a particular speaker. After these devices have been trained, their performance may be degraded if the correspondence between the stimuli and sensor states is altered by factors extrinsic to the stimuli of interest[1]. For example, the OCR device may be "confused" by image distortions due to derangement of the camera's optical/electronic path, or it may be misled by pixel intensities caused by altered illumination of the printed page. Such a perturbation changes the sensor state elicited by each stimulus and thereby defines a mapping of sensor states onto one another. If such transformations map one of the sensor states in the training set onto another one, the pattern recognition software may misclassify the corresponding stimulus.

These problems can be addressed by periodically recalibrating the device to account for changed conditions. For example, the device can be exposed to a test pattern that produces a known sensor state under "normal" conditions. The observed differences between the actual and ideal sensor states for this test stimulus can be used to correct subsequently encountered sensor states. This procedure must be implemented after each change in observational conditions in order to keep track of time-dependent distortions. Because the device may not be able to detect the presence of such a change, it may be necessary to recalibrate it at short fixed intervals. However, this will tend to decrease the device's duty cycle by frequently taking it "off-line". Furthermore, the recalibration process may be logistically impractical in some situations (e.g., computer vision and speech recognition devices at remote locations).

Human perception has the remarkable ability to adapt to sensory state transformations that are due to factors extrinsic to the stimuli of interest. This phenomenon has been the subject of philosophical discussion since the time of Plato, and it has also intrigued modern neuroscientists. It is dramatically illustrated by experiments[2] in which subjects wore goggles creating severe geometric distortions of the observed scene. Although the subjects initially noticed the distortion, their perceptions of the world returned to the pre-experimental baseline after several weeks of constant exposure to familiar stimuli seen through the goggles. For example, lines reported to be straight before the experiment were initially perceived to be warped, but these lines were once again reported to be straight after several weeks of viewing familiar scenes through the distorting lenses. There are many other examples of how our percepts are often

Medical Imaging 2001: Image Processing, Milan Sonka, Kenneth M. Hanson, Editors,
Proceedings of SPIE Vol. 4322 (2001) © 2001 SPIE · 1605-7422/01/$15.00

invariant under changed observational conditions. For instance, human observers are not usually confused by altered intensity of scene illumination or by changed acoustic properties of a listening environment. Although the raw sensory state of the observer is altered in each case, this is not usually attributed to changed intrinsic properties of the stimulus of interest (e.g., the scene, the music). Evidently, the perception of the stimulus remains invariant under these varying conditions. This ability to maintain constant percepts despite transformed sensory input may also help explain the fact that different persons tend to share the same perceptions of the world, despite significant differences in their sensory organs and processing pathways.

This report[3] describes how to build sensory devices that behave like the subjects of the above-mentioned goggle experiments. These devices generate stimulus representations that are initially affected by perturbed observational conditions, but they eventually return to their baseline forms. The stability of such representations may make it easier for the device's pattern analysis module to recognize and classify stimuli as conditions change. In order to see how this can be achieved, consider any perturbation that systematically alters the correspondence between the stimuli and the sensor states. Because of such changes, a stimulus that formerly resulted in sensor state x will now induce another sensor state x'. In general, x is an array of numbers that can be considered to be the coordinates of the sensor state on the manifold of all sensor states. In this language, the above-mentioned perturbations systematically transform the absolute coordinates of the sensor state associated with each stimulus. However, certain *relationships* between the coordinates of a collection of sensor states may remain invariant under such a transformation. This is analogous to the fact that the physical rotation or translation of a collection of particles in a plane does not affect the relationships among the members of the collection. For example, Euclidean coordinate geometry can be used to describe the *relative* positions of such particles in terms of a "natural" internal coordinate system that is rooted in the collection's intrinsic structure; i.e., the coordinate system that originates at the collection's center of "mass" and is oriented along its principal moments of "inertia". Such a self-referential description is invariant under global rotations and translations that change the absolute coordinates of each particle. This suggests the following strategy: if we describe stimuli in terms of the relationships among their sensor states, we may be able to represent them in a way that is not affected by the above-described perturbations of observational conditions. In this paper, we demonstrate the following realization of that approach: a sufficiently dense collection of recently encountered sensor states has a natural internal structure that endows the manifold with *local* coordinate systems, which are analogous to the global center-of-mass coordinate system of the particle collection. If the representation of each stimulus is referred to these local coordinate systems, it will be invariant under any invertible local transformations of sensor states. Now consider a device that uses this method to describe stimuli in terms of recently encountered stimuli. Suppose that: 1) prior to a perturbation, this method is used to describe a stimulus in terms of the relationship between its sensor state and the time series of recently encountered sensor states; 2) after the onset of the perturbation, the method is used to describe the same stimulus in terms of the relationship between its perturbed sensor state and a time series of recently encountered perturbed sensor states. This paper demonstrates that the two descriptions will be identical as long as the two time series of recently encountered sensor states (before and after the perturbation) were produced by the same sets of stimuli. In essence, the stability of this type of stimulus representation is due to the stability of the device's recent "experience" (i.e., the stability of the time series of recently encountered stimuli to which descriptions are referred). The representation of a stimulus will drift during a transitional period when the device is referring its description to a mixed time series of unperturbed and perturbed sensor states. However, as in the human case, the representation of each stimulus will eventually revert to its baseline form when the recently encountered states are entirely comprised of perturbed sensor states.

As suggested above, the task of creating invariant stimulus representations can be reduced to the mathematical task of describing sensor state relationships that are not affected by transformations of the sensor state manifold. Now, assume that the change in observational conditions defines a *one-to-one* transformation of the sensor states. This requirement of invertibility simply excludes changes in conditions (such as a change in the spectral content of scene illumination) that make it possible to distinguish previously indistinguishable stimuli or that obscure the difference between previously distinguishable stimuli. Such a perturbation has exactly the same effect on sensor state coordinates as a change of the sensor state coordinate system ($x \rightarrow x'$) in the absence of a perturbation. This is analogous to the fact that the physical rotation of an array of particles in a plane has the same effect on their coordinates as the inverse rotation of the axes of the coordinate system. Therefore, the task of finding sensor state relationships that are independent of perturbed observational conditions is mathematically equivalent to the task of finding sensor state relationships that are coordinate-independent. In other words, we need to find the relationships among the recently encountered sensor states that are independent of the coordinate system (x or x') used to label them.

2. THEORY

2.1. Coordinate-independent descriptions of sensor states

Consider a sensory device having an array of N detectors that are sensitive to various features of stimuli. These detectors may send their output to processing units that combine them in a possibly non-linear fashion. For example, in an imaging system, the processing units may compute the position of some image feature that is being tracked. Let the device's sensor state x denote the array of numbers x_k ($k = 1,..., N, N \geq 1$) that form the output of these processing units. Our goal is to create a description of the sensor states that is independent of the x coordinate system, which we happen to be using to label them. In other words, the same description must result if we used another (x') coordinate system to label the sensor states. In this paragraph, we demonstrate that such a coordinate-independent description can be created if we have a *coordinate-independent* way of identifying (Fig. 1): 1) a reference sensor state (x_0), 2) a path $x(u)$ ($0 \leq u \leq 1$) through the manifold of sensor states that connects the reference sensor state to the sensor state of interest ($x(0) = x_0, x(1) = x$), 3) N linearly-independent contravariant vectors h_a ($a = 1, ..., N$) at each point along the path. Here, a vector h is said to be contravariant if it transforms[1] as $h^k \rightarrow h'^k = \sum_{l=1,...,N} \frac{\partial x'^k}{\partial x_l} h^l$ under the change of coordinate systems $x \rightarrow x'$. If the foregoing conditions are met, each infinitesimal segment δx along the path can be decomposed into its components δs_a along the vectors h_a:

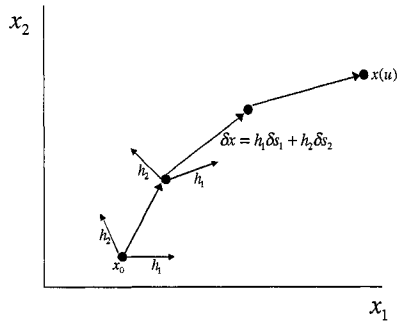

Figure 1. Consider a path $x(u)$ ($0 \leq u \leq 1$) between a reference sensor state x_0 and a sensor state of interest. If vectors h_a can be defined at each point along the path, each line segment δx can be decomposed into its components δs_a along the vectors at that point.

$$\delta x = \sum_{a=1,...,N} h_a \, \delta s_a \qquad (1)$$

The vectors h_a define local (generally non-orthonormal) axes that are independent of the global coordinate system (x or x') on the sensor state manifold, and δs_a is the description of δx relative to those axes. Note that δs_a is a coordinate-independent (scalar) quantity because δx and h_a are contravariant vectors[4, 5]. Therefore, if the components δs_a are integrated over the specified path connecting x_0 and x, the result[6] is a coordinate-independent description of the sensor state x:

$$s_a = \int_{x(0)}^{x} \delta s_a \qquad (2)$$

The next three subsections show how the coordinate-independent information required for this type of description (a reference state, paths connecting it to other sensor states, and the vectors h_a) can be derived from the local structure of the time series of recently encountered sensor states.

2.2 Sensor state manifolds having local directionality

In this subsection, we show that in some circumstances the vectors h_a can be directly derived from the time series of previously encountered sensor states and that these vectors can then be used to define a unique path between x_0 and any other sensor state. Suppose that the device has an accurate clock so that it can record $x(t)$, the time series of previously encountered sensor states. This function describes a trajectory that crosses the sensor state manifold. For example, Fig. 2a shows segments of such a trajectory for a system with two detectors. In this illustration, the trajectory segments tend to be along two primary directions (left side to right side, lower half-plane to upper half-plane), and points equally-spaced in time establish a scale along each one of them. We now show how these features can be used to define local vectors $h_a(x)$ in a manner that is independent of the coordinate system. Consider a point x that has multiple trajectory segments passing through it in N or more different directions, where N is the manifold's dimension. The time derivatives of the segments passing through x form a collection of contravariant vectors $\hat{h}_i$ at x:

---

[1] Except for dx_k, contravariant vector and tensor components are denoted by raised indices. Tensor components that transform covariantly (i.e., that transform by the matrix dx_k / dx'_l) are denoted by lowered indices.

$$\hat{h}_i = \left.\frac{dx}{dt}\right|_{t_i} \tag{3}$$

where t_i denotes the i^{th} time at which the trajectory passed through x. These vectors can be used to define N principal vectors at x if they tend to fall into clusters oriented along different directions in the manifold. To see this, pick an integer $C \geq N$ and partition the indices i of these vectors into C non-empty sets labeled S_c where $c=1, \ldots, C$. Next, compute the $N \times N$ covariance matrix M_c of the vectors corresponding to each set of indices:

$$M_c = \frac{1}{N_c} \sum_{i \in S_c} \hat{h}_i \hat{h}_i \tag{4}$$

where N_c is the number of indices in S_c. Each of these matrices transforms as a tensor with two contravariant indices, and the determinant of each matrix $|M_c|$ transforms as a scalar density of weight equal to minus two[4]; namely, if coordinates on the manifold are transformed as $x \rightarrow x'$, then

$$|M_c| \rightarrow |M_c'| = \left|\frac{\partial x'}{\partial x}\right|^2 |M_c| \tag{5}$$

Next, compute E, which is defined to be the sum of powers of these determinants:

$$E = \sum_c |M_c|^p \tag{6}$$

where p is some real positive number. Equation 5 implies that E transforms as a scalar density of weight $-2p$. Now tabulate the values of E for all possible ways of partitioning the set of vectors $\hat{h}_i$ into C non-empty sets, and find the partition that results in the smallest value of E. This partition will tend to group the vectors into subsets with minimal matrix determinants. Therefore, the vectors in each group will tend to be linearly dependent or nearly linearly dependent, and they will tend to form a cluster that is oriented in one direction. Next, compute the principal vectors h_c at x by finding the average vector in each part of the optimal partition:

$$h_c = \frac{1}{N_c} \sum_{i \in S_c} \hat{h}_i \tag{7}$$

Because the $\hat{h}_i$ are contravariant vectors, the h_c will also transform as contravariant vectors as long as the $\hat{h}_i$ are partitioned in the same manner in any coordinate system. However, because E transforms by a positive multiplicative factor, the same partition minimizes it in any coordinate system. Therefore, the optimal partition is independent of the coordinate system, and the h_c are indeed contravariant vectors. Finally, the indices of the h_c can be relabeled so that the corresponding determinants $|M_c|$ are in order of ascending magnitude. This ordering is also coordinate-independent because these determinants transform by a positive multiplicative factor (Eq.(5)). As a result, if the foregoing computations are done in any coordinate system, the same vectors h_c will be created, and these vectors provide a coordinate-independent characterization of the directionality of the trajectories passing through x.

The first N vectors that are linearly independent can be defined to be the h_a in Eq.(1). These can be used to compute δs_a, the coordinate-independent representation of any line element passing through x. Once we have specified a path connecting a reference state x_0 to any sensor state x, Eq.(2) can be integrated to create a coordinate-independent representation s_a of that state. The path must be completely specified because the integral in Eq.(2) may be path-dependent. To see this, note that Eq.(1) can be inverted to form:

$$\delta s_a = \tilde{h}_a \cdot \delta x \tag{8}$$

where the covariant vectors $\tilde{h}_a$ are found by solving $\sum_{a=1,\ldots,N} \tilde{h}_{ak} h_a^l = \delta_k^l$ and δ_k^l is the Kronecker delta function. It follows from Eq.(2) that each component of s_a is a line integral of $\tilde{h}_a$. Stoke's theorem shows that these line integrals will be path-dependent unless the "curl" of $\tilde{h}_a$ vanishes:

$$\frac{\partial \tilde{h}_{ak}}{\partial x_l} - \frac{\partial \tilde{h}_{al}}{\partial x_k} = 0 \tag{9}$$

Because this may not be true for some sensor state manifolds, we must create a coordinate-independent way of specifying a path from x_0 to any point x on the manifold. To see how this can be done, imagine generating a "type 1" trajectory through x_0 by moving along the local h_1 direction at x_0 and then moving along the h_1 direction at each subsequently encountered point. Next, generate a "type 2" trajectory through each point on the type 1 trajectory by

moving along the local h_2 direction at that point and at each subsequently-encountered point. Continue in this fashion until a type N trajectory has been generated through each point on every trajectory of type *N-1*. Because of the linear independence of the h_a at each point, the collection of points on type n trajectories ($1 \le n \le N$) comprises an n-dimensional subspace of the manifold. Therefore, each point on the manifold lies on a type N trajectory and can be reached from x_0 by traversing the following type of path: a segment of the type 1 trajectory, followed by a segment of a type 2 trajectory, ..., followed by a segment of a type N trajectory. This path specification is coordinate-independent because the quantities h_a transform as contravariant vectors. Therefore, if Eq.(2) is integrated along this "canonical" path, the resulting value of s_a provides a coordinate-independent description of the sensor state x in terms of the recently-encountered sensor states; i.e., a description that is invariant under perturbed observational conditions.

Some circumstances may prevent the derivation of the h_a at a sufficiently dense set of sample points on the manifold. For example, suppose that there is no unique way of partitioning the $\hat{h}_i$ in order to minimize E at some points, or suppose that the h_a (Eq.(7)) associated with the minimal value of E do not contain N linearly independent members. This means that the history of sensor states *x(t)* does not endow the manifold with sufficient directionality.

2.3 Sensor state manifolds that support parallel transport

In this subsection, we show how the past history of sensor states *x(t)* may have sufficient internal structure to define a method of moving vectors ("parallel transporting" them) across the manifold. It may be possible to define parallel transport rules on a manifold, even in the absence of local directionality at *every* point, the property that was required in the previous subsection. As long as a manifold supports parallel transport and has directionality *at a single point*, vectors h_a can be moved across the manifold from that point in order to define vectors h_a at all other points in a coordinate-independent manner. Then, Eqs.(1-2) generate coordinate-independent descriptions of sensor states.

As in the previous subsection, the time series of previously encountered sensor states *x(t)* describes a trajectory that crosses the sensor state manifold (e.g., Fig. 2a). According to the methods of affine-connected differential geometry[4,5], any vector can be moved across this manifold in a coordinate-independent manner if one can define a local affine connection $\Gamma^k_{lm}(x)$, which is a quantity transforming as:

$$\Gamma'^{k}_{lm} = \sum_{r,s,t=1,\ldots,N} \frac{\partial x'_k}{\partial x_r} \frac{\partial x_s}{\partial x'_l} \frac{\partial x_t}{\partial x'_m} \Gamma^r_{st} + \sum_{n=1,\ldots,N} \frac{\partial x'_k}{\partial x_n} \frac{\partial^2 x_n}{\partial x'_l \partial x'_m} \tag{10}$$

Specifically, given any contravariant vector V at x, consider the array of numbers $V + \delta V$ where:

$$\delta V^k = - \sum_{l,m=1,\ldots,N} \Gamma^k_{lm} V^l \delta x_m \tag{11}$$

It can be shown that $V + \delta V$ transforms as a contravariant vector at the point $x + \delta x$, as long as the affine connection transforms as shown in Eq.(10). The vector $V + \delta V$ at $x + \delta x$ is said to be the result of parallel transporting V along δx. Our task is to use the past history of sensor states *x(t)* to derive an affine connection on the sensor state manifold. Then, given a set of vectors h_a at just one point on the manifold (e.g., at the reference state x_0), we will be able to use the affine connection to populate the entire manifold with parallel-transported versions of those vectors. These can be used in Eqs.(1-2) to derive a coordinate-independent representation of any sensor state.

Consider a point x that is on at least *N(N+1)/2* trajectory segments. Each of these segments can be divided into infinitesimal line elements dx that correspond to equal infinitesimal time intervals. These line elements transform as contravariant vectors. Therefore, we can look for affine connections that parallel transport a given line element along itself into the next line element on the same trajectory segment. In other words, we can look for affine connections for which a given trajectory segment is locally geodesic[4,5]. Equation 11 implies that such an affine connection $\hat{\Gamma}^k_{lm}$ must satisfy the following N constraints:

$$\delta dx_k = - \sum_{l,m=1,\ldots,N} \hat{\Gamma}^k_{lm} dx_l dx_m \tag{12}$$

where $dx + \delta dx$ represents the trajectory's line element at $x + dx$. Now consider any collection of *N(N+1)/2* of the trajectory segments at x. An affine connection that makes all of these trajectory segments locally geodesic must satisfy $N^2(N+1)/2$ linear constraints like those in Eq.(12). Because a symmetric affine connection ($\Gamma^k_{lm} = \Gamma^k_{ml}$) has $N^2(N+1)/2$ components, one and only symmetric connection satisfies these equations unless they happen to be inconsistent (no solutions) or redundant (multiple solutions). Notice that if $\hat{\Gamma}^k_{lm}$ is a solution of these equations in one coordinate system, then $\hat{\Gamma}'^{k}_{lm}$ is a solution of the corresponding equations in any other coordinate system, where $\hat{\Gamma}^k_{lm}$ and $\hat{\Gamma}'^{k}_{lm}$ are related by Eq.(10). Therefore, if these equations have a unique solution in one coordinate system, there is a

unique solution of the corresponding equations in any other coordinate system, and these solutions are related by Eq.(10). Now, consider all collections of $N(N+1)/2$ trajectory segments that have a unique solution to these equations; i.e. all collections that are locally geodesic with respect to one and only one symmetric affine connection at x. Let Γ^k_{lm} be the average of the affine connections computed from these subsets of trajectory segments:

$$\Gamma^k_{lm} = \frac{1}{N_T} \sum_{i=1,\ldots,N_T} \hat{\Gamma}^k_{lm}(i), \tag{13}$$

where $\hat{\Gamma}^k_{lm}(i)$ is the symmetric affine connection that makes the i^{th} collection of trajectory segments locally geodesic and N_T is the number of such collections. The quantity Γ^k_{lm} transforms as shown by Eq.(10) because each contribution to the right side of Eq.(13) transforms in that way. Therefore, Γ^k_{lm} can be defined to be *the* affine connection at point x.

Now, suppose that N linearly independent vectors h_a can be defined at a point x_0 on the manifold. For example, the method in Section 2.2 might be used to derive these vectors from previously encountered sensor states in the vicinity of x_0. Alternatively, the device may have prior knowledge of certain vectors at x_0 that are known to be invariant under all relevant coordinate transformations (i.e., all relevant perturbations of observational conditions). The above-described affine connection can be used to parallel transport these vectors to any other point x on the manifold. The resulting vectors at x will depend on the path that was used to create them if the manifold has a non-zero curvature tensor[4,5]. Because this tensor will not vanish in many cases, the path connecting x_0 and x must be completely specified in a coordinate-independent manner. Such a path can be prescribed in the following fashion. Generate a trajectory through x_0 by repeatedly parallel transporting the vector h_1 along itself, and call this trajectory a type 1 geodesic. Next, parallel-transport all of the vectors h_a along this trajectory. Now, generate a type 2 geodesic through each point of this geodesic by repeatedly parallel transporting the vector h_2 along itself. Then, parallel-transport all of the vectors h_a along each of these geodesics, and generate a type 3 geodesic through each point on each type 2 geodesic by repeatedly parallel transporting the vector h_3 along itself. Continue in this manner until type N geodesics have been generated through each point on each type $N-1$ geodesic. Because of the linear independence of the vectors h_a at x_0, the parallel transported h_a will also be linearly independent. It follows that the collection of points on all trajectories of type n comprises an n-dimensional subspace of the manifold, and the type N trajectories will reach every point on the manifold. This means that any point x can be reached from x_0 by following a "canonical" path consisting of a segment of the type 1 geodesic, followed by a segment of a type 2 geodesic, ... followed by a segment of a type N geodesic. This path specification is coordinate-independent because it is defined in terms of a coordinate-independent operation: namely, the parallel transport of vectors. After the h_a have been "spread" to the rest of the manifold along these paths, a coordinate-independent representation s_a of any point x can be generated by integrating Eq.(2) along the "canonical" path between x_0 and x.

Notice that this method (as well as the methods in Sections 2.2 and 2.4) requires that a special sensor state (the reference state) is identifiable in every coordinate system. This reference state might be identified as a coordinate-independent feature of the history of sensor states. For example, x_0 could be the sensor state that is a nearby local maximum (or minimum) of the function defined by the number of times each sensor state has been encountered recently. The reference state could also be specified by some prior knowledge. For example, we might know *a priori* that: 1) x represents the intensity of a pixel in a digital image, 2) the device was originally calibrated so that $x=0$ corresponds to no light emission/reflection at the pixel's location, and 3) we are primarily concerned about sensor state transformations due to changes of illumination intensity. Because changes in illumination intensity are expected to map the point $x=0$ onto itself, we only need to be concerned about coordinate transformations with that fixed point. Therefore, in this case, prior knowledge tells us that the null point $x_0 = 0$ represents the same sensor state in every coordinate system of interest, and, therefore, it can be chosen to be the reference state. Alternatively, the system may be able to identify the reference state because the system's operator periodically "shows" it the reference stimulus, in the same manner as the leader of a choir or orchestra demonstrates the sound of a standard reference note to a musical ensemble prior to a performance.

2.4 Sensor state manifolds that support a metric

In this section, we show how the past history of sensor states $x(t)$ can impose a Riemannian metric on the manifold, even in the absence of local directionality at every point, the property that was required to implement the method in Section 2.2. This metric can then be used to define parallel transport rules. As long as the manifold has sufficient directionality to define vectors h_a at a single point, those vectors can be parallel transported in order to define vectors h_a at all other points. Then, Eqs.(1-2) can be used to create coordinate-independent descriptions of sensor states.

As in Sections 2.2 and 2.3, let $x(t)$ be the trajectory of previously encountered sensor states. Consider a point x that is on at least $N(N+1)/2$ trajectory segments. Each of these segments defines an infinitesimal line element $dx = x(t+dt) - x(t)$, where t is the time at which the trajectory segment passed through x and dt is an infinitesimal time interval. Now consider one of these line elements, and look for metrics that assign unit length to it. Such a metric $\hat{g}_{kl}$ must satisfy the following constraint:

$$\sum_{k,l=1,\ldots,N} \hat{g}_{kl} dx_k dx_l = 1 \tag{14}$$

Next, consider any collection containing $N(N+1)/2$ of the line elements at x. A metric that assigns unit length to all of these line elements must satisfy $N(N+1)/2$ linear constraints like the one in Eq.(14). Because a metric has $N(N+1)/2$ components, one and only one metric satisfies these equations unless they happen to be inconsistent (no solutions) or redundant (multiple solutions). If these equations have a unique solution in one coordinate system, there is a unique solution of the corresponding equations in any other coordinate system, and these solutions define the same covariant tensor in the different coordinate systems. This is a consequence of the fact that each line element dx transforms as a contravariant vector. Now, consider all collections of $N(N+1)/2$ line elements that have a unique solution to these equations. Let g_{kl} be the average of the metrics computed from these subsets of line elements:

$$g_{kl} = \frac{1}{N_L} \sum_{i=1,\ldots,N_L} \hat{g}_{kl}(i). \tag{15}$$

where $\hat{g}_{kl}(i)$ is the metric that assigns unit length to the i^{th} collection of line elements and N_L is the number of such collections. Note that sets of line elements for which Eq.(14) has no solution or multiple solutions do not contribute to Eq.(15). The quantity g_{kl} transforms as a covariant tensor because each contribution to the right side of Eq.(15) transforms in that way. Therefore, g_{kl} can be defined to be *the* metric at point x on the sensor state manifold.

There are several ways of using the above-defined metric to define parallel transport on the sensor state manifold. For example, the following quantity is a symmetric affine connection that preserves the metrically computed lengths of vectors during parallel transport:

$$\Gamma^k_{lm} = \frac{1}{2} \sum_{n=1,\ldots,N} g^{kn} \left(\frac{\partial g_{mn}}{\partial x_l} + \frac{\partial g_{nl}}{\partial x_m} - \frac{\partial g_{lm}}{\partial x_n} \right) \tag{16}$$

where g^{kl} is the contravariant tensor that is the inverse of g_{kl}. Other definitions of the affine connection are also possible[4]. Now, suppose that N linearly independent vectors h_a can be defined at a point x_0 on the manifold. For example, the method in Section 2.2 might be used to derive these vectors from previously encountered sensor states in the vicinity of x_0. The above-described affine connection can be used to parallel transport these vectors to any other point x on the manifold. The resulting vectors at x will depend on the path that was used to create them if the manifold has non-zero curvature. Therefore, in general, the path connecting x_0 and x must be completely specified in a coordinate-independent manner. Such a path can be prescribed exactly as it was in Section 2.3. Namely, we can define a "canonical" path to x that follows a specific sequence of geodesics, which were created by parallel transport of the vectors h_a at x_0. Then, a coordinate-independent representation s_a of any point x can be generated by integrating Eq.(2) along the canonical path between x_0 and x.

3. SIMULATED EXPERIMENTS

3.1 A sensor state manifold having directionality

In this section, we demonstrate the methodology in Section 2.2 by applying it to simulated data on a two-dimensional sensor state manifold. Let $x = (x_1, x_2)$ represent the state of the device's sensor. For example, these numbers might be the coordinates of a specific feature being tracked in a time series of digital images. Suppose that Fig. 2a represents the trajectories of the sensor states that were previously encountered by the system. Notice that these lines tend to be oriented in nearly horizontal or vertical directions, thereby endowing the manifold with directionality at each point. We used these data to compute the local vectors h_a on a uniform grid of sample points that was centered on the origin and had spacing equal to two units. To do this, we considered a small neighborhood of each sample point, and the time derivative of each trajectory segment traversing the neighborhood was computed at equal time intervals. Then, Eqs.(3-7) with $p=1$ were applied in order to derive local vectors from the collection of time derivatives at each sample point. The resulting vectors h_a, shown in Fig. 2b, were then interpolated in order to estimate the vectors at intervening points. As expected, these vectors reflect the horizontal and vertical orientations of the trajectories from which they were derived. Finally, Eqs.(1-2) were applied to these h_a in order to compute the coordinate-independent representation s_a

of each sensor state on the manifold, relative to the reference state which was chosen to be $x_0 = (0,0)$. The result is shown in Fig. 2c, which depicts the level sets of $s_a(x)$, the scale function that is intrinsic to the sensor state history in Fig. 2a. Figs. 2d and 2e show how some "images" in the x coordinate system are represented in the s coordinate system.

Next, we considered what would have happened if the same device had "experienced" sensor states shown in Fig. 3a. These trajectories are related to those in Fig. 2a by the following non-linear transformation:

$$x_1 \to 0.1 + x_1 + 0.1x_2 + 0.01x_1^2 - 0.02x_2^2 - 0.01x_1x_2 \qquad (17)$$
$$x_2 \to 0.2 - 0.2x_1 + x_2 - 0.01x_1^2 + 0.02x_2^2 + 0.01x_1x_2$$

which is invertible over the domain in Fig. 2a. For example, suppose that x is the location of a feature in a digital image. Equation (17) could represent the way the sensor states are transformed by a distortion of the optical/electronic path within the camera or by a distortion of the surface on which the camera is focused (e.g., distortion of a printed page). The procedure outlined above was used to compute the local vectors on a uniform grid of sample points. Fig. 3b shows the resulting vectors, which are oriented along the principal directions apparent in Fig. 3a. Next, interpolation was used to estimate the h_a at intervening points, and Eqs.(1-2) were used to compute the coordinate-independent representation s_a of each sensor state on the manifold, relative to the reference state which was chosen to be $x_0 = (0.1, 0.2)$. Notice that we have assumed prior knowledge of the transformed position of the reference state. In other words, we have assumed that we have the prior knowledge necessary to identify this state both before and after the onset of the perturbation. The result of this calculation is shown in Fig. 3c, which depicts the level sets of $s_a(x)$, the scale function inherent to the sensor state data in Fig. 3a. These functions were used to compute the s_a representation of the transformed versions of the "images" on the left sides of Figs. 2d and 2e. The transformed images and their s_a representations are shown in Figs. 3d and 3e. Comparison of Figs. 2d, e and Figs. 3d, e shows that the s_a representations of the untransformed and transformed images are nearly identical. Thus, the methodology in this paper makes it possible to maintain invariant representations of stimuli in the presence of unknown invertible transformations of sensor states, such as the one in Eq.(17). The tiny discrepancies between the right sides of Figs. 2d, e and Figs. 3d, e can be attributed to errors in the interpolation of the h_a, which are due to the coarseness of the grid on which h_a was sampled. This error can be reduced if the distance between sample points can be decreased. This is possible if the device is allowed to experience a denser set of sensor states (i.e., more trajectory segments than shown in Figs. 2a and 3a) so that even tiny neighborhoods contain enough data to compute the h_a.

3.2 A sensor state manifold supporting parallel transport

In this section, we demonstrate the methodology in Section 2.3 by applying it to simulated data on a two-dimensional sensor state manifold. Let $x = (x_1, x_2)$ represent the state of the device's sensor. Suppose that Fig. 4a represents the trajectory of sensor states previously encountered by the system. Notice that these happen to be straight lines that are traversed at constant speed. Equations 12-13 were used to compute the affine connection on a uniform grid of sample points that was centered on the origin and had spacing equal to two units. To do this, we considered a small square neighborhood of each sample point. Each trajectory segment that traversed the neighborhood was divided into line elements traversed in equal time intervals. Next, we considered any three pairs of such line elements, where each pair consisted of two adjacent line elements on a trajectory segment. Then, we asked if there was a unique affine connection $\hat{\Gamma}^k_{lm}$ that parallel transported each line element into the other line element of the same pair. The affine connection at the sample point was set equal to the average of the quantities $\hat{\Gamma}^k_{lm}$ that were derived for all possible triplets of paired line elements in the neighborhood (Eq.(13)). Triplets of paired line elements for which there was no unique solution (e.g., multiple solutions or no solution) did not contribute to this average. In this way, we derived an affine connection for which neighboring trajectory segments were geodesic in an average sense. In this particular case, all components of the resulting affine connection were computed to be zero; i.e., the x coordinate system is a geodesic coordinate system of a flat manifold. This result is expected because a vanishing affine connection is the only one that parallel transports equally long line elements of straight lines into one another.

The reference state was chosen to be the origin of the x coordinate system ($x_0 = 0$), and the method in Section 2.2 was used to compute local vectors from the directionality of trajectory segments near x_0. Specifically, we considered the trajectory segments passing through a small square neighborhood of x_0. Local vectors $\hat{h}_i$ were found by calculating the time derivatives along these trajectory segments at the equally spaced time points shown in Fig. 4a. We then looked at all possible ways of partitioning this collection of vectors into two subsets and found the partition with the minimal value of E (Eq.(6) with $p=1$). Finally, the average vector in each of these partitions was computed in order to

find the principal vectors h_c at x_0. As explained in Section 2.2, these are the directions in which the local trajectory segments tend to be oriented. In this case, these vectors were oriented in nearly horizontal and vertical directions.

The affine connection was used to "spread" these vectors throughout the manifold by parallel transporting them along type 1 and type 2 geodesics. Then, Eqs.(1-2) were used to compute the values of s_a that comprise the coordinate-independent representation of each sensor state x. The results are shown in Fig. 4b, which depicts the level sets of $s_a(x)$. Because of the flat nature of this particular manifold, the coordinate system s_a is related to the x coordinate system by an affine transformation. Figs. 4c, d show how some "images" in the x coordinate system are represented in the s coordinate system.

Next, we considered what would have happened if the same device had "experienced" sensor state trajectories in Fig. 5a, which are related to those in Fig. 4a by a quadratic transformation (similar to Eq.(17)) that was invertible over the domain in Fig. 4a. The procedure outlined above was used to compute the affine connection on a uniform grid of sample points. The resulting affine connection was non-vanishing at each sampled point, and smooth interpolation was used to estimate its values at intervening points. Next, the above-described procedure was used to compute the principal directions of the trajectories at the point $x_0 = (2.5, 0)$, which corresponds to the untransformed state $x_0 = 0$. In other words, we have assumed that we have the prior knowledge necessary to identify this state after the onset of perturbed observational conditions. The affine connection was used to "spread" these vectors throughout the manifold along type 1 and type 2 geodesics, and the s_a representation of each point in the manifold was computed by means of Eq.(1-2). Fig. 5b shows the level sets of the resulting function $s_a(x)$. Notice that the warping of the sensor states in Fig. 5a has led to the warping of the s_a coordinate system, which is the natural scale imposed on the manifold by the history of sensor states. The function $s_a(x)$ was used to compute the s_a representation of the transformed versions of the "images" on the left side of Figs. 4c, d. The transformed images and their s_a representations are shown in Figs. 5c, d. Comparison of Figs. 4c, d and 5c, d shows that the s_a representations of these images are nearly identical. In other words, these representations are invariant with respect to the non-linear transformation the sensor states, even though they were computed without using the functional form of that transformation. The tiny discrepancies between the right sides of Figs. 4c, d and Figs. 5c, d can be attributed to errors in the interpolation of the affine connection, which are due to the coarseness of the grid on which the affine connection was sampled. This error can be reduced if the distance between sample points can be decreased. This is possible if the device is allowed to experience a denser set of sensor states (i.e., more trajectory segments than shown in Fig. 4a and Fig. 5a) so that even tiny neighborhoods contain enough data to compute the affine connection.

4. DISCUSSION

This paper shows that the intrinsic structure of the time series of recently encountered sensor states imposes a "natural" scale $s_a(x)$ on the sensor state manifold of a sensory device. If the experienced sensor states are all subjected to an invertible transformation, this scale automatically adapts so that the representation (scale value) of each stimulus remains constant. Such transformations may be due to changes of observational conditions such as: 1) alterations of the internal characteristics of the device's sensory apparatus, 2) changes in the observational environment that is external to the sensory device and the stimuli, and 3) modifications of the presentation of the stimuli themselves. Unlike conventional sensory systems, a device with this type of "representation engine" in its "front end" need not be recalibrated with test patterns, and its pattern recognition software need not be retrained because of such perturbations[7]. This is advantageous because these procedures may be logistically impractical in some situations (remote, unsupervised devices), and, in any event, they reduce the device's "duty cycle" by taking it "off-line". This methodology of this paper may also make it easier to build multiple sensory devices (e.g., a fleet of robots) that have different sensors but use the same pattern analysis software. These different devices would represent stimuli in the same way as long as: 1) there is a one-to-one mapping between their sensor states corresponding to the same stimuli; 2) they have recently encountered stimuli with the same average properties.

This type of sensory device automatically "recalibrates" itself in a continuous fashion. It does this by using the time series of recently encountered sensor states (e.g., the sensor states encountered in the most recent ΔT time units) to derive a scale $s_a(x)$ at each time point. If ΔT or more time units have elapsed since the onset of a perturbation, the scale values of each subsequently encountered sensor state will be referred to a time series of perturbed sensor states. We have shown that this representation will be the same as that derived before the perturbation, as long as the time series of stimuli encountered before and after the perturbation have the same average local properties (i.e., the same local axes or h_a, up to the transformation caused by the perturbation). Thus, it is the stability of the device's "experience" that stabilizes its representation of individual stimuli. After the onset of perturbed observational conditions, the device's

stimulus representations will drift during the transitional period when those representations are referred to a mixture of unperturbed and perturbed sensor states. However, once the recent time series of sensor states is dominated by perturbed data (i.e., once ΔT time units have elapsed), the representation of each stimulus will return to its form before the perturbation. This means that the device can successfully adapt to a large change in observational conditions without errors, as long as the change occurs at a sufficiently slow rate. Specifically, if the change occurs in small steps separated by relatively long time intervals, each increment will cause a small distortion of the stimulus representations during a transitional period before the representations revert to their baseline forms. If the pattern analysis software can tolerate these small temporary distortions, it will continue to recognize stimuli correctly, even though the cumulative change of observational conditions may be large. In essence, the device is able to "keep up" with a slow pace of change by continually making the scale adjustments that are necessary to maintain invariant stimulus representations. Examples of all of these effects of time-dependent perturbations can be found in references 8 and 9. The time interval ΔT should be chosen to be long enough so that: 1) during each period of that length the sensor state manifold is populated with enough states to compute $s_a(x)$, 2) the average properties of stimuli encountered during any such period are stable enough to stabilize $s_a(x)$ during constant observational conditions, and 3) the number of sensor states encountered in any such period is sufficient to limit the noise propagating into the computation of $s_a(x)$. Within these limitations, ΔT should be chosen to be as short as possible so that the device rapidly adapts to changing observational conditions.

The behavior of such a device resembles that of human subjects during "goggle" experiments. Those experiments suggested that the properties of recently experienced sensory data strongly influence the way a subject's percepts are constructed from subsequent sensory data. Specifically, after a period of adaptation, subjects perceived stimuli just as they did before the onset of the distortion. The above-described devices have a similar ability to maintain constant stimulus representations in the presence of sensory input distortions, after a period of adaptation. Notice that the device may find it difficult to form stable internal representations of sensor states in a part of the manifold that is poorly sampled by recently encountered states. This is because the manifold will not be endowed with stable local structure (vectors h_a) if there is a paucity of sensor state trajectories near each relevant point. Like a human, the device must have sufficient "experience" in order to form stable stimulus representations. These behavioral similarities make it tempting to speculate that human perception is actually based on a mechanism analogous to the one described in this paper: namely, an ability to appreciate the coordinate-independent relationships among previously encountered sensory states. This could help to explain how humans perceive the constancy of visual stimuli in the presence of changing observational conditions and how they extract the same information content from a wide variety of speech signals that are distortions of one another[8,9].

REFERENCES

1. E. R. Davies, *Machine Vision: Theory, Algorithms, and Practicalities*, Academic Press, New York, 1990.
2. R. Held and R. Whitman, *Perception: Mechanisms and Models*, W. H. Freeman, San Francisco, 1992.
3. D. N. Levin, "Stimulus representations that are invariant under invertible transformations of sensor data. Part 1 and Part 2", submitted for publication (2000).
4. E. Schrodinger, *Space-Time Structure*, Cambridge University Press, Cambridge, UK, 1963.
5. S. Weinberg, *Gravitation and Cosmology: Principles and Applications of the General Theory of Relativity*, Wiley, New York, 1972.
6. D. N. Levin, "A differential geometric description of the relationships among perceptions", *J. Math. Psych.* **44**, 241-284 (2000).
7. D. N. Levin, *Self-referential method and apparatus for creating stimulus representations that are invariant under systematic transformations of sensor states*, Patents pending.
8. D. N. Levin, "Time-dependent signal representations that are independent of sensor calibration", *J. Acoust. Soc. Amer.* **108**, 2575 (2000). Posted at http://asa.aip.org/newport/information.html.
9. D. N. Levin, "Representing information in communications signals so that it is invariant under any invertible signal distortion", IEEE *Internat. Conf. on Acoustics, Speech, and Signal Proc.,* Salt Lake City, UT, 2001. Submitted for presentation.

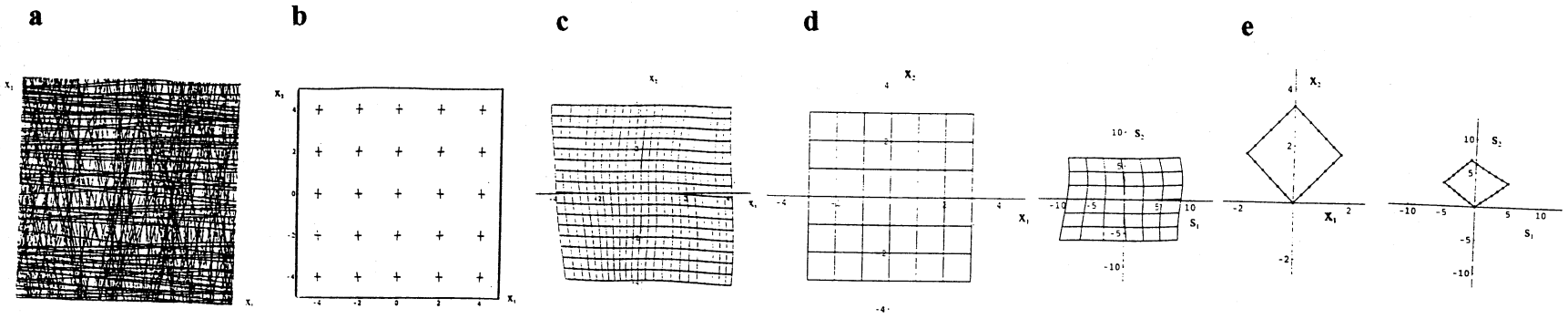

Figure 2. a) The simulated trajectory of recently encountered sensor states *x(t)*. The speed of traversal of each trajectory segment is indicated by the dots, which are separated by equal time intervals. The nearly horizontal and vertical segments are traversed in the left-to-right and bottom-to-top directions, respectively. The graph depicts the range $-5 \le x_k \le 5$. b) The local vectors h_a that were computed from the time derivatives of the trajectories in panel *a*. The nearly horizontal and vertical lines denote vectors that are oriented to the right and upward, respectively. c) The level sets of $s_a(x)$, which show the coordinate-independent stimulus representations derived by applying the method in Section 2.2 to the data in panel *a*. The nearly vertical curves are loci of constant s_1 for evenly spaced values between –11 (left) and 12 (right); the nearly horizontal curves are loci of constant s_2 for evenly spaced values between –8 (bottom) and 8 (top). d) and e) The device that encounters the sensor states in panel *a* will generate the figure on the right of each panel as the coordinate-independent representation of the collection of sensor states on the left.

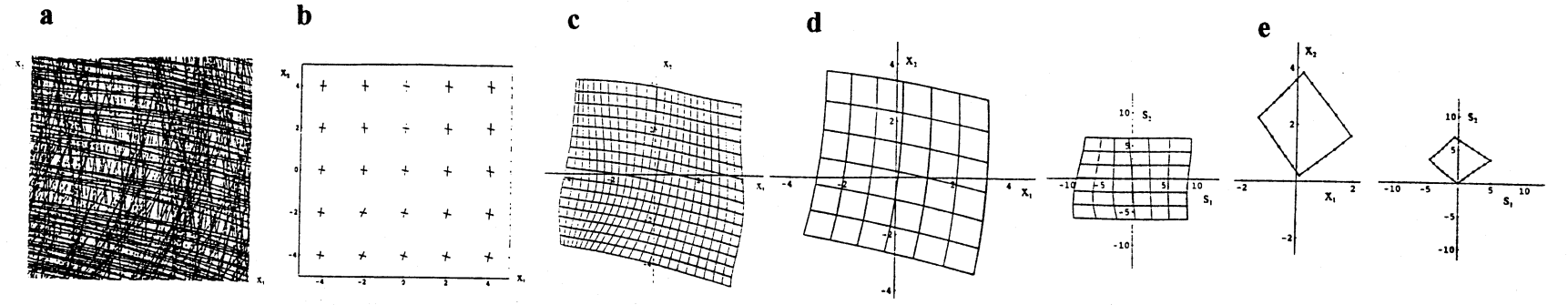

Figure 3. a) The simulated trajectory of recently encountered sensor states *x(t)* that is related to the trajectory in Fig. 2a by the transformation in Eq.(17). The speed of traversal of each trajectory segment is indicated by the dots, which are separated by equal time intervals. The nearly horizontal and vertical segments are traversed in the left-to-right and bottom-to-top directions, respectively. The graph depicts the range $-5 \le x_k \le 5$. b) The local vectors h_a that were derived from the sensor state data in panel *a*. The nearly horizontal and vertical lines denote vectors that are oriented to the right and upward, respectively. c) The level sets of $s_a(x)$, which show the coordinate-independent stimulus representations derived by applying the method in Section 2.2 to the data in panel *a*. The vertical curves are loci of constant s_1 for evenly spaced values between –12 (left) and 11 (right); the horizontal curves are loci of constant s_2 for evenly spaced values between –9 (bottom) and 7 (top). d) and e) The device that encounters the transformed sensor states in panel *a* will generate the right figure in each panel as the coordinate-independent representation of the collection of sensor states on the left. Each figure on the left was created by subjecting the corresponding left side of Figs. 2d and 2e to the transformation in Eq.(17). Notice that the right figures are nearly identical to those in Figs. 2d and 2e, thereby confirming the fact that these representations are invariant under the sensor state transformation in Eq.(17).

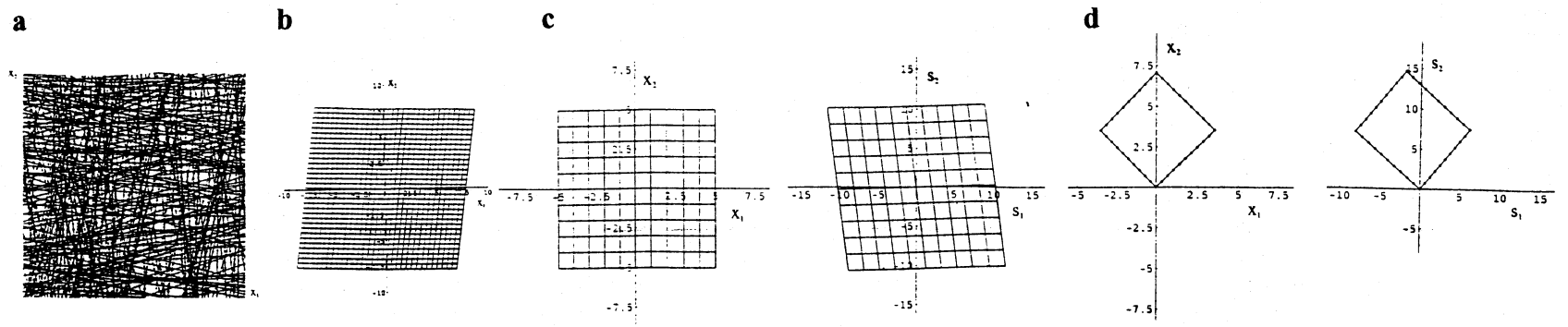

Figure 4. a) The simulated trajectory of recently encountered sensor states *x(t)*. The speed of traversal of each trajectory segment is indicated by the dots, which are separated by equal time intervals. The nearly horizontal and vertical segments are traversed in the left-to-right and bottom-to-top directions, respectively. The graph depicts the range $-10 \le x_k \le 10$. b) The level sets of $s_a(x)$, which show the coordinate-independent stimulus representations that were derived by applying the method in Section 2.3 to the data in panel *a*. The nearly vertical curves are loci of constant s_1 for evenly spaced values between –16 (left) and 16 (right); the nearly horizontal curves are loci of constant s_2 for evenly spaced values between –16 (bottom) and 16 (top). c) and d) The device that encounters the sensor states in panel *a* will generate the right figure of each panel as the coordinate-independent representation of the collection of sensor states on the left side of each panel.

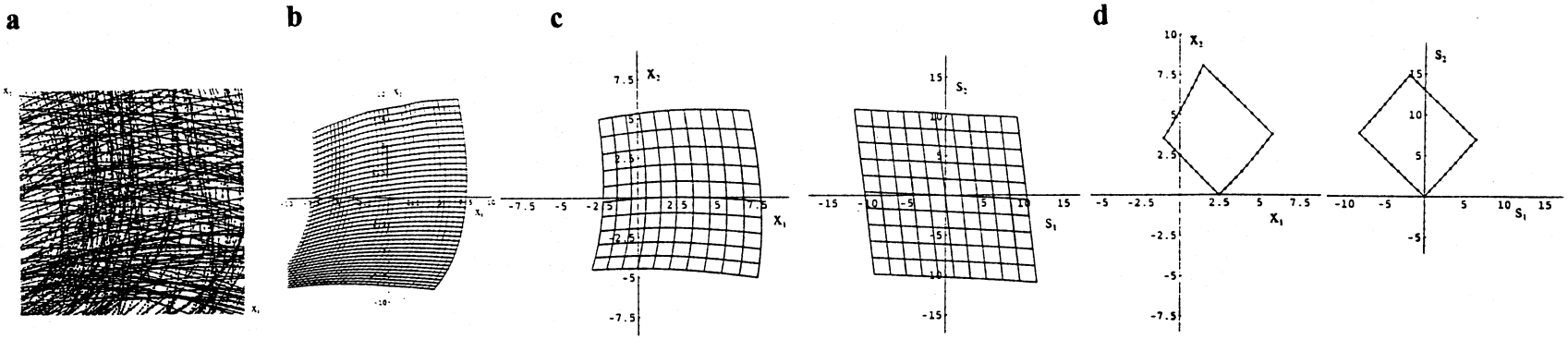

Figure 5. a) The simulated trajectory of recently encountered sensor states *x(t)* that is related to the trajectory in Fig. 4a by a quadratic coordinate transformation. The speed of traversal of each trajectory segment is indicated by the dots, which are separated by equal time intervals. The nearly horizontal and vertical segments are traversed in the left-to-right and bottom-to-top directions, respectively. The graph depicts the range $-10 \le x_k \le 10$. b) The level sets of $s_a(x)$, which show the coordinate-independent stimulus representations that were derived by applying the method in Section 2.3 to the data in panel *a*. The vertical curves are loci of constant s_1 for evenly spaced values between –24 (left) and 10 (right); the horizontal curves are loci of constant s_2 for evenly spaced values between –22 (bottom) and 16 (top). c) and d) The device that encounters the transformed sensor states in panel *a* will generate the right figures as coordinate-independent representations of the collections of sensor states on the left. Each figure on the left was created by subjecting the corresponding left figure in Figs. 4c, d to the quadratic transformation relating Figs. 4a and 5a. Notice that the right figures are nearly identical to those in Figs. 4c, d, thereby confirming the fact that these representations are invariant under the perturbation transforming Fig. 4a into Fig. 5a.