Title
Dimensionality-Reduction and Constraint in Later Vision

Journal
Proceedings of the Annual Meeting of the Cognitive Science Society, 9(0)

Author
Saund, Eric

Publication Date
1987

Peer reviewed

Dimensionality-Reduction and Constraint in Later Vision

Eric Saund
Department of Brain and Cognitive Sciences and
the Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

Keywords: computational vision, dimensionality-reduction, connectionist networks

Abstract

A computational tool is presented for maintaining and accessing knowledge of certain types of constraint in data: when data samples in an n-dimensional feature space are all constrained to lie on an m-dimensional surface, $m < n$, they can be encoded more concisely and economically in terms of location on the m-dimensional surface than in terms of the n feature coordinates. The recoding of data in this way is called *dimensionality-reduction*. Dimensionality-reduction may prove a useful computational tool relevant to later visual processing. Examples are presented from shape analysis.

1 Introduction

It is commonly understood that vision involves the interaction of incoming data with a priori knowledge about the world. In Early Vision, constraint due to the physics of image formation, rules of imaging geometry, and statistical properties of the world can be analyzed mathematically to support inferences about primitive scene properties based on image measurements. Problems of later vision, however, involving the interpretation of image data in terms of task goals, object models, and meaningful world events, do not offer straightforward mathematical characterizations of the more abstract constraints that give rise to comprehensible images. For example, it is difficult to characterize the constraints on a shape profile that might qualify it to be called a "scissors" shape.

One type of constraint that arises in shape analysis can be cast in terms of surfaces embedded in high-dimensional feature spaces. This paper suggests that the computational tool of *dimensionality-reduction* can be used to maintain and access knowledge of complex constraints of this form.

2 Dimensionality-Reduction

An idealized view of early visual processing is that certain image features are extracted from raw images. For example, figure 1 shows primitive features extracted from an image of a pair of scissors, making explicit the x-location, y-location, and orientations of edges and corners. Slightly more complex features may be constructed by combining primitives, say, by measuring the distances and relative orientations between pairs of corners. We might imagine that one aspect of early visual processing involves the delivery of image descriptions in terms of data points in a huge multidimensional feature space whose coordinate axes are defined by the features measured. The job of later visual processing is to make sense out of the feature descriptions delivered it.

Image feature data can only be meaningfully interpreted by appeal to knowledge of constraints governing the world giving rise to images. These constraints are reflected in the behavior of extracted features for given classes of shapes. For example, features associated with a pair of scissors

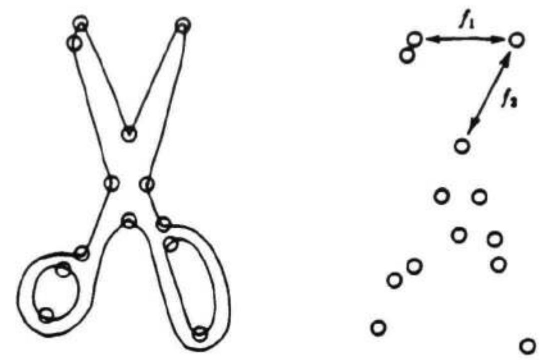

Figure 1. Hypothetical corner primitives extracted from the image of a pair of scissors. These can be combined into more complex features such as the distances between pairs of corners.

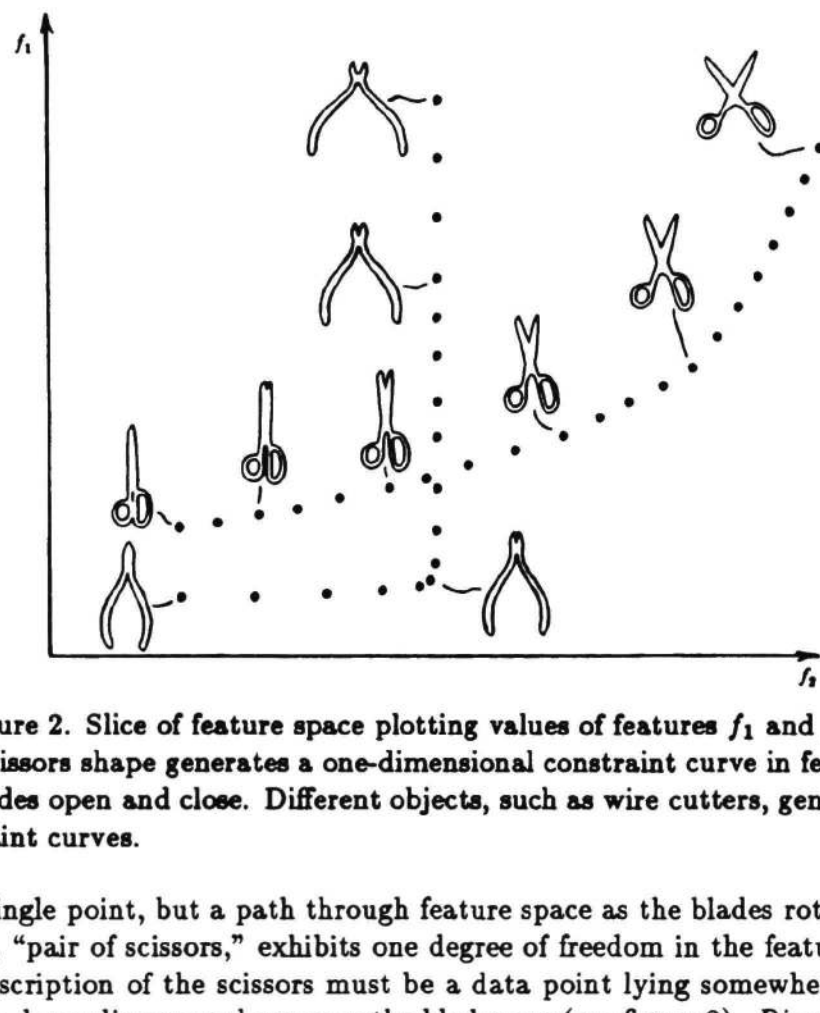

Figure 2. Slice of feature space plotting values of features f_1 and f_2 from figure 1. A scissors shape generates a one-dimensional constraint curve in feature space as the blades open and close. Different objects, such as wire cutters, generate different constraint curves.

generate not a single point, but a path through feature space as the blades rotate about the pivot. The shape class, "pair of scissors," exhibits one degree of freedom in the feature space, and every feature space description of the scissors must be a data point lying somewhere on the constraint curve, just where depending upon how open the blades are (see figure 2). *Dimensionality-reduction* [Krishnaiah and Kanal, 1982; Kohonen, 1984] is the computational mapping between the description of data expressed in terms of its location in a high-dimensional feature space, and in terms of its location on a lower-dimensional *constraint surface*. Figure 3 illustrates. This mapping employs knowledge of the lower-dimensional constraint surface, embedded in the high-dimensional feature

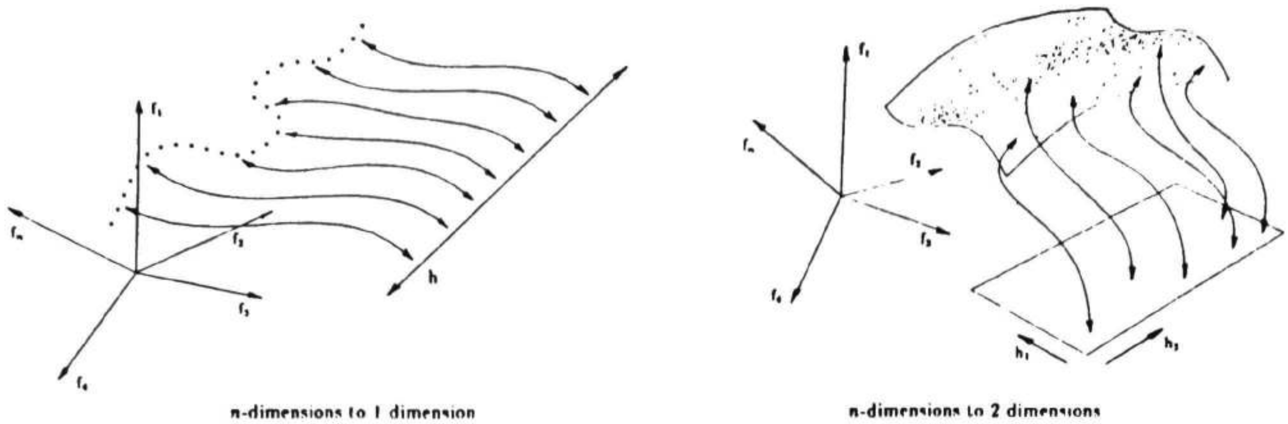

Figure 3. Dimensionality-reduction is a mapping between points described in a high-dimensional feature space, and points described in terms of location on a lower-dimensional constraint surface.

space, that is generated by a physical process or class of data. Different objects generate different constraint surfaces in the feature space, as seen in figure 2.

The purpose served by dimensionality-reduction is abstraction and simplification in the description of data. A description consisting of a symbolic token indicating that an input datum lies on the "scissors" constraint surface in the feature space, plus one parameter indicating where it lies on the surface (how open the scissors are), is certainly preferable to a listing of the input's coordinate location along each of the n feature dimensions. In general, a dimensionality-reduced representation can be expected to make explicit descriptive parameters capturing the natural degrees of variability inherent to a class of data, while it factors out redundancy latent among the original measured features.

3 Achieving Dimensionality-Reduction

A black box depiction of a dimensionality-reducer is presented in figure 4. Each such box contains knowledge of one constraint surface, such as, for example, might characterize one class of object shapes. At the bottom of the box enters the description of an image in terms of an n-dimensional feature vector. Out the top emerge n lines, and out the side, m more, where m is the dimensionality of the constraint surface. Each line can represent a bounded real number; for convenience suppose that the feature coordinates are normalized so that all features take values between 0 and 1.

The dimensionality-reducer operates as follows. If the numbers coming out the top of the black box match those coming in the bottom, then it is determined that the data point whose feature vector is given does in fact lie on the constraint surface, and its location on the constraint surface may be read on the m lines coming out the side (the dimensionality-reducer implicitly creates a coordinate system for the constraint surface). If the numbers coming out the top do not match the input feature vector, then it is determined that the data point specified at the input does not lie on the constraint surface.

This black-box dimensionality-reducer may also be used in the opposite direction. That is, an m-dimensional vector specifying a location on the constraint surface may be placed on the side lines as input, and the dimensionality-reducer will then compute, and output at the top, the coordinates of this data point in the n-dimensional feature space.

Several alternative implementations of such a black box dimensionality-reducer are possible.

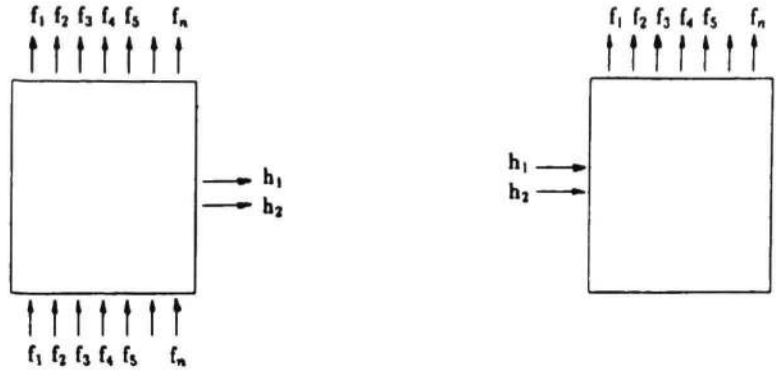

Figure 4. Black box dimensionality-reducer.

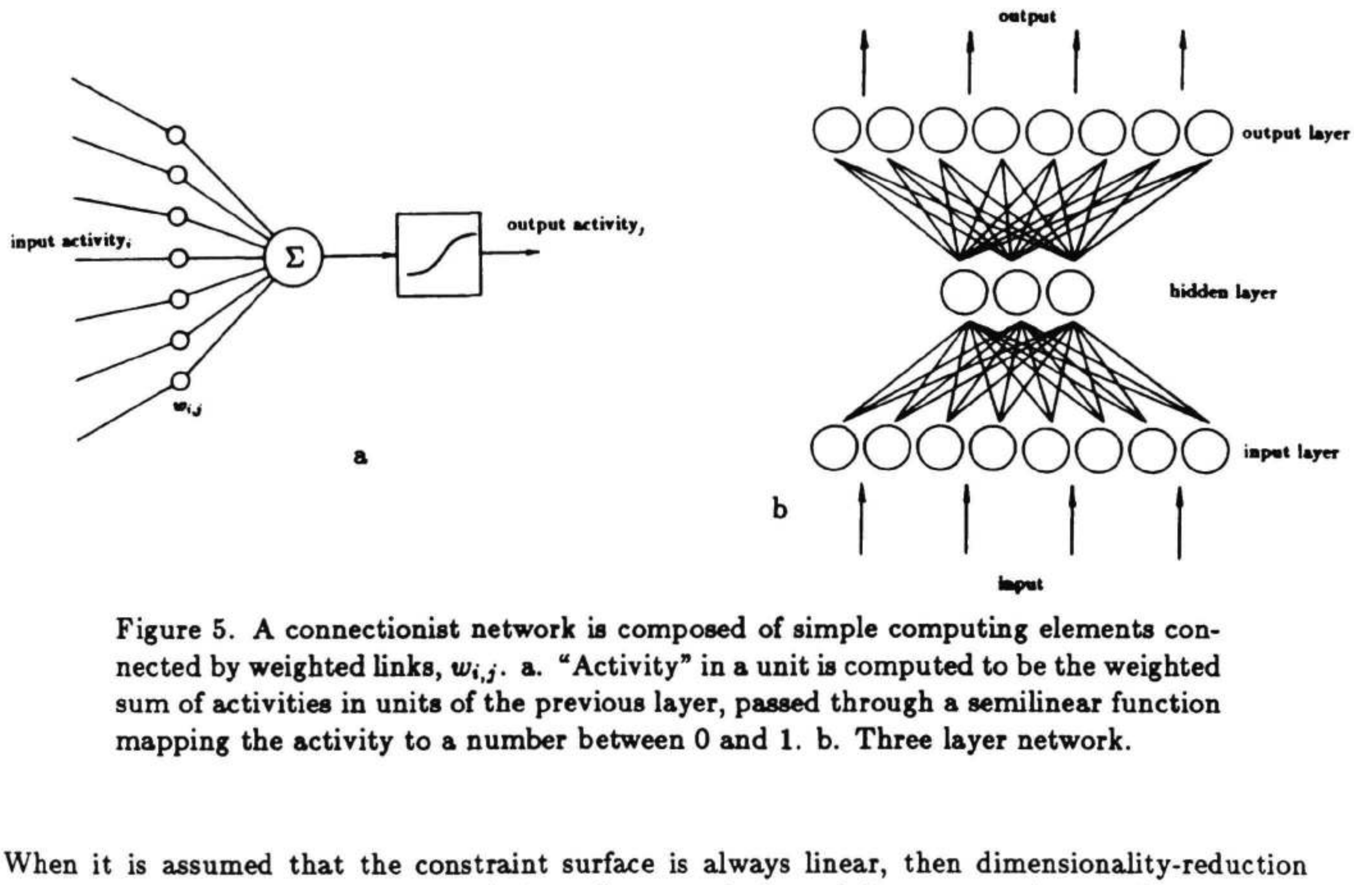

Figure 5. A connectionist network is composed of simple computing elements connected by weighted links, $w_{i,j}$. a. "Activity" in a unit is computed to be the weighted sum of activities in units of the previous layer, passed through a semilinear function mapping the activity to a number between 0 and 1. b. Three layer network.

When it is assumed that the constraint surface is always linear, then dimensionality-reduction amounts to principle components analysis, or factor analysis, and the computation may be expressed as a matrix multiplication [Watanabe, 1965; Tou and Heydorn, 1967; Fukunaga and Koontz, 1970; Kittler and Young, 1973].

A more general implementation, in which the constraint surface may curve to a considerable degree, uses a connectionist network of simple computing elements. Figure 5 illustrates a three-layer network. Each unit takes an activity between 0 and 1. Activity is fixed at the bottom layer as input, then activity for each unit in the middle layer (called the "hidden" layer) is computed as

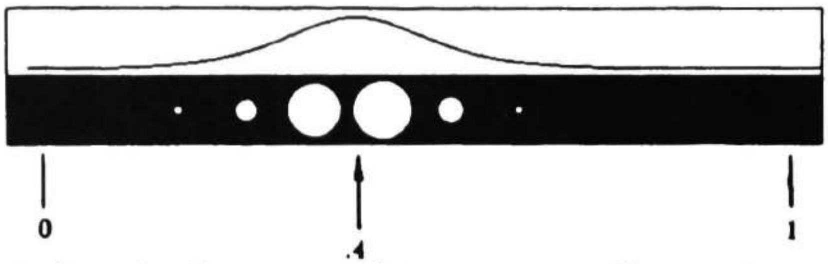

Figure 6. Scalar values between 0 and 1 are represented in sets of units, called *scalar sets*, whose activity takes a characteristic unimodal pattern. Activity of a unit is represented as size of circle. The activity pattern shown in this 12-unit scalar-set represents the number, .4.

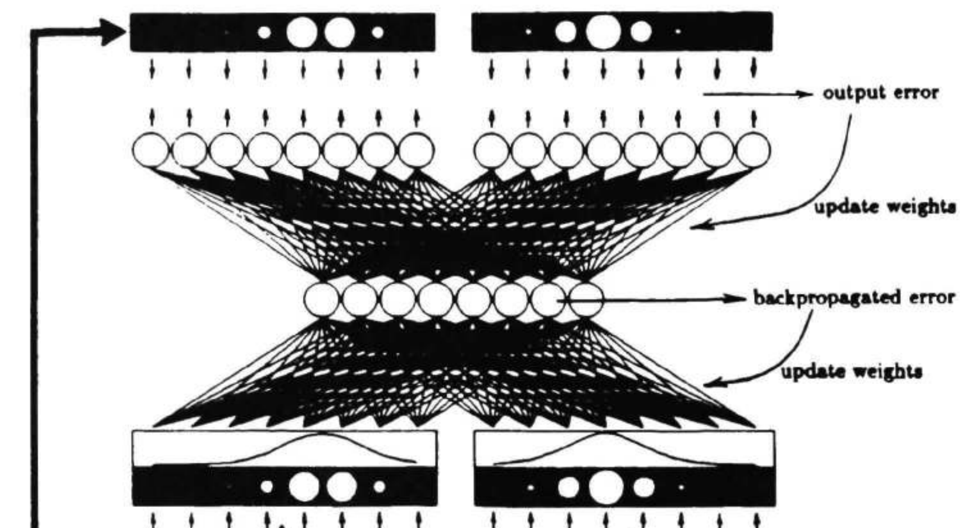

Figure 7. Connectionist dimensionality-reducer. In this case, two scalar-sets are provided at the input and output layers, and the hidden layer is a one-dimensional scalar set, therefore, this network can represent one-dimensional constraint curve in a two-dimensional feature-space. During training period, errors from desired activity are used to train network to reproduce input activity pattern at output.

a semilinear function of the weighted sum of activities on the input units. Output layer activity is computed from the hidden layer activity in a similar way.

Each scalar feature value is represented as the pattern of activity over a set of units, called a *scalar-set*, as illustrated in figure 6. A one-dimensional scalar-set is provided at the input layer and at the output layer for each dimension of the higher-dimensional feature space. The hidden layer is configured as a scalar-set whose dimensionality matches that of the embedded constraint surface. The input, hidden, and output layers of the network correspond to the bottom, side, and top of the black box dimensionality-reducer (see figure 7).

Dimensionality-reducing behavior is achieved by virtue of the link weights between successive layers of the network. These weights are established using the backpropagation training procedure [Rumemlart, Hinton and Williams, 1985; Rumelhart and McClelland, 1986], which furnishes crucial self-organizing properties during the training phase. Training consists of repeated presentation of input activity/desired output activity pairs, where the desired output is defined to be identical to the input activity. At each training trial, activity at the input layer is fixed according to the coordinates of a data sample expressed terms of the high-dimensional feature space. For example, each image of a pair of scissors, with blades open to some degree, generates one training sample, or data point in feature space. Activity propagates through the network, and the output layer activity is compared with that of the input, to create an output error vector. This error is used to incrementally update link weights between the hidden and output layers in such a way as to reduce the error. In addition, the output activity error is backpropagated through the hidden layer/output layer link weights to arrive at an equivalent error in activity at the hidden layer. This essentially amounts to analyzing how each unit at the hidden layer contributed to the error observed at the output layer. The hidden layer error is in turn used to update link weights between the input and hidden layers. In order to achieve patterns of activity at the hidden layer that are interpretable in terms of a location on the lower-dimensional constraint surface, an auxiliary error is introduced at the hidden layer to be added to the backpropagated error. This auxiliary error serves to pressure the input/hidden links into creating patterns of activity at the hidden layer scalar-set taking a characteristic unimodal form representing a location on the constraint surface. A more detailed discussion of the connectionist dimensionality-reducer is presented in [Saund, 1986] and [Saund, 1987].

4 Dimensionality-Reduction in Shape Description

The practical matter of using a dimensionality-reducer involves measuring feature parameters on images and plugging them into appropriate slots in the "black box" input. Each dimensionality-reducer possesses knowledge of legal configurations of features for a given class of data, such as a shape category. Tasks such as object recognition are in principle accomplished by testing feature vectors delivered by early vision against various objects' dimensionality-reducers, amounting to a sort of generalized template matching. Some means must be provided for determining just what measured features to pair with each input line; as the number of measured features increases there occurs a combinatoric explosion of possible feature/input matchings. Therefore, to be utilized by a visual system the dimensionality-reduction tool must be used within a computational shell controlling the mechanics of measuring features, selecting candidate object models, assigning measured feature values to appropriate input lines, evaluating the dimensionality-reducer's applicability to the feature vector, and reading the reduced description off the side output lines.

The choice of features defining a higher-dimensional feature space is important to achieving useful data abstraction through dimensionality-reduction. The example above illustrates that a pair of scissors generates a one-dimensional constraint surface in a feature space derived from simple edge and corner primitives. However, differently shaped scissors give rise to different constraint curves in the feature space because they all create somewhat different configurations of edge and corner features, and a separate dimensionality-reducer would be required for each constraint curve. Where dimensionality-reduction might more realistically come into play is in interpreting feature descriptions at a more abstract level, say, once edge and corner primitives have been grouped into parts. Then, a single dimensionality-reducer might suffice to capture the blade pivoting constraint inherent to all pairs of scissors.

Prior to training a dimensionality-reducer, it is important to select a dimensionality for the constraint surface to match the inherent dimensionality of the data. The connectionist dimensionality-reducer described above provides no means for doing this automatically. However, it is easy to detect whether the constraint surface is of inadequate dimensionality, because under this condition, during training, a network will converge to a state in which it does not correctly map activity at the input layer to (nearly) identical activity at the output layer in terms a unimodal pattern of

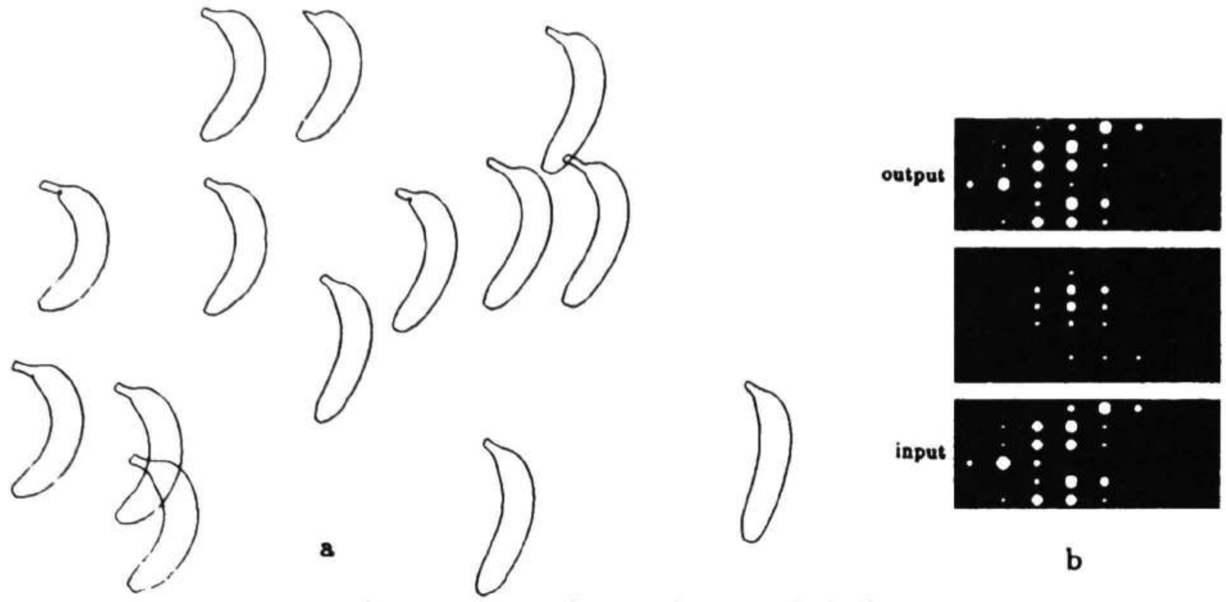

Figure 8. a. Banana shapes arranged according to their locations on a two-dimensional constraint surface found by a connectionist dimensionality-reducer. These were originally described in terms of six simple features such as distances between the ends and lengths and curvatures of various contour segments. The parameters of variability found to pertain among these bananas are roughly the curvature of the lower part of the banana (left/right), and the overall size of the banana (up/down). b. Activity of the dimensionality-reduction network for one banana. Note that output activity matches input activity, and that hidden layer activity is centered around one location in the two-dimensional scalar-set.

activity at the hidden layer.

It is desirable to seek constraint surfaces of low dimensionality for two reasons. First, limits may exist on the tractability of discovering many-dimensional constraint surfaces. The amount of data that must be analyzed in order to establish a constraint surface increases as the power of the surface's dimensionality. In the current computer implementation of the connectionist dimensionality-reducer, the cost in terms of network links and nodes appears to become prohibitive after m becomes three or four. Second, the data simplification afforded by dimensionality-reduction may lose some value when the underlying constraint surface is yet of high dimensionality. For example, it is not certain that much is gained by describing data in terms of a ten-dimensional coordinate system instead of a twenty-dimensional coordinate system. Dimensionality-reduction is perhaps most useful when the constraints operating in a given problem can be decomposed into systems of just a few inherent degrees of freedom each.

Constraint surfaces in multidimensional feature spaces arise in several different ways. The scissors example illustrates a situation in which a single object generates a one-dimensional parameterized class of shapes by virtue of physically constrained motion between parts. Constraint surfaces may also originate in classes of shape objects in which each individual object has a fixed shape, but one that can vary in only certain ways from the shapes of other objects in the class.

Figure 8 shows a set of bananas that were originally described in terms of six properties crudely measured on the banana shapes, such as the distance between the ends and average curvature of various contour segments. By training a connectionist dimensionality-reducer on these data samples, the bananas were found to lie on a two-dimensional constraint surface in the six-dimensional feature space. The organization of this constraint surface is illustrated in the figure; bananas are placed on a plane according to their respective two-dimensional coordinates. Note that banana shapes are organized on the basis of very subtle differences in their geometrical properties.

Although the reduced dimensionality representation concisely encodes the essential parameters of variability among members of the data class falling on a constraint surface, the lower-dimensional coordinate axes do not necessarily align with interpretations of these parameters preferred by human observers. For example, the horizontal and vertical axes of figure 8 roughly correspond to curvature of the lower part of a banana, and banana size, respectively, however, the dimensionality-reduction training procedure run again on the same banana data might rotate these axes an arbitrary amount in the plane.

5 Conclusion

Dimensionality-reduction is a computational tool for exploiting knowledge of constraint latent in a collection of data in order to achieve simpler and more perspicuous descriptions. It is applicable when data samples lie on a lower-dimensional constraint surface embedded in an initial, higher-dimensional, descriptive feature space. Unlike mathematical and physical model based procedures for capturing constraint in early vision, dimensionality-reduction is a quite general concept which holds the possibility for capturing the rather more complex and non-analytic nature of regularities inherent to later visual analysis. The ultimate utility of this tool in computational vision and Cognitive Science rests with the degree to which the visual world exhibits the appropriate type of constraint. This paper presents two rather simple examples of dimensionality-reduction at work in the analysis of visual shape.

References

Fukunaga, K., and Koontz, W.,[1970], "Application of the Karhunen-Loève Expansion to Feature Selection and Ordering," *IEEE Transactions on Computers*, C-19, 311-318.

Kittler, J., and Young, P., [1973], "A New Application to Feature Selection Based on the Karhunen-Loève Expansion," *Pattern Recognition*, 5, 335-352.

Kohonen, T.,[1984], *Self-Organization and Associative Memory*, Springer-Verlag, Berlin.

Krishnaiah and Kanal, eds., [1982], *Handbook on Statistics, Vol. 2: Classification, Pattern Recognition, and Reduction of Dimensionality*, North-Holland.

Rumelhart, D., Hinton, G., and Williams, R., [1985], "Learning Internal Representations by Error Propagation," ICS Report 8506, Institute for Cognitive Science, UC San Diego.

Rumelhart, D., McClelland, J, and the PDP Research Group, [1986], *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Bradford Books, Cambridge, Ma.

Saund, E., [1986], "Abstraction and Representation of Continuous Variables in Connectionist Networks," *Proceedings of the Fifth National Conference on Artificial Intelligence*, 638-644.

Saund, E., [1987], "Dimensionalty-Reduction Using Connectionist Networks," AI Memo 941, Massachusetts Institute of Technology.

Tou, J., and Heydorn, R., [1967], "Some Optimal Approaches to Feature Extraction," in Tou, ed., *Computer and Information Sciences II*, Academic Press.

Watanabe, S.,[1965], "Karhunen-Loève Expansion and Factor Analysis," *Trans. 4th Prague Conference on Information Theory.*