Anchoring Symbols by Mapping Sequences of Distance Measurements: Experimental Results

Erik Hoertnagl, Patrick M. Poelz, Erich Prem

Austrian Research Institute for Artificial Intelligence
Freyung 6/6, 1010 Wien, Austria
{erik, patrick, erich}@oefai.at

Abstract

In this paper we present an approach to symbol anchoring that is based on mapping sequences of distance measurements from simple sensors. The sensory space of the mobile robot is pre-structured according to its experiences when it first moves around in unexplored environments. Such pre-structuring depends not only on environmental features, but also on the type of behaviour the robot exhibits. Object representations correspond to streams of sensory signals that are mapped onto this sensory space and classified by a sequence detection mechanism. We report on novel experimental results using this technique where we compare variants of the approach and other more simple methods. We present data from experiments with varied parameters and input data types such as motor and distance sensor information. In our real mobile robot scenario the robot successfully discriminates a number of objects that can then be anchored in a second step using input from a human supervisor.

Introduction

Robot perception of the world

The world according to a mobile robot which moves around in pursuit of its tasks looks very different from the human observer's perception of the robot. While humans quickly recognize objects regardless from viewing angle and position and are able to correctly label such objects with names, all this is very hard for a robot. This is particularly true in cases where the robot is only equipped with simple distance sensors rather than a camera. However, making such a robot recognize and label objects which it encounters is an important goal, for example as a basis for planning components or simply for communication with the robot.

A considerable part of correctly connecting sensor streams and symbolic names for perceived objects is to determine which of the many object qualities are significant. In humans this process of learning about the features of the environment based on the sensory-motor inputs takes place in the first 5 years and even earlier in the womb. In this time span the human brain is highly dynamic and mouldable. Each sensory impression and each thought leaves its mark in the nervous system.

However, for robots it is not a trivial task to extract the important qualities of an object. Mobile robots today are often only equipped with relatively simple sensory systems such as infrared or sonar distance measurements and a few propriosensors e.g. for odometry. Recognition of objects for such a robot moving around in its environment brings up several problems including comparing multivariate time series of sensory experiences or which sensors to focus attention on. In the past, algorithms for multi-dimensional scaling (MDS) have been used for this purpose including PCA or SOM-based techniques. In [Prem *et al.* 2003] we presented first ideas using the Isomap approach for MDS. MDS-algorithms are useful in reducing the dimensions of the input space without loosing the neighbourhood relations of points in this input space. Reducing the dimensions should reduce the complexity of the recognition task without loosing important information or features of the sensory data.

Overview of the approach

In our approach the robot first shapes its perception of the world according to its behavioural and sensory experiences. In a second step, it learns to anchor sensor streams of objects based on its view of the world and in a third step to relate this view to our human conceptualisation of the world using symbols. This process is usually called symbol anchoring [Coradeschi & Saffioti 2003] and related to symbol grounding [Harnad 1990]. In this paper we focus on the first of these steps, i.e. processing sensory data as a basis for anchoring.

In a first step the robot explores its test environment and collects data until its experiences cover most of its environment and it encountered a majority of interesting situations, e.g. objects. Then, an Isomap (cf. section "The architecture") is computed from its sensor input streams and an experience-map is created. This map represents sequences of high-dimensional sensory data on a lower dimensional MDS-map. In this map, two experiences which are similar will also generally lie in the same region of the map and the trajectories of similar sensory experiences will generate sequences on the map that look similar. This map thus enables the robot to check whether an experience has already occurred and whether two experiences are similar. After the construction of the map and after the identification of objects, a teacher can now label these objects with symbols.

Anchoring in our case consists of two parts, first mapping the sensor and motor inputs into the low dimensional Isomap-space and then mapping these trajectories to named objects. Still the process of anchoring according to [Coradeschi & Saffioti 2003] involves one more issue, namely the process of reacquiring and tracking the found anchors. This aspect is only dealt with conceptionally in the current state of our work. Based on the sensors with which our robot is equipped, we are only able to deal with certain kinds of object representations and thus allow anchoring object classes rather than identifying single object entities.

Still there is a way of reacquiring single objects using semantic probability maps. I.e. the identified objects are linked together in a map in a way that allows the robot at any time to make predictions about what objects will follow depending on the object sequences it encountered in the near past. In a world where just a few number of similar or different objects exist, it is even possible to perform simple tracking based on such a semantic probability map.

Note the difference between the old "perceive environment – create model the world – develop plan" principle and what happens here: The robot learns through experience in its environment to interpret its sensors so as to detect similarities and differences. The map is not a model of the world, rather a model of how to perceive the world. Also, the map is not a topological map, but a topology-preserving record of the robot experiences. Learning to perceive similarities is a vital key for anchoring as well as recognition.

The test platform

In our experiments we use a wheeled mobile robot platform originally designed for sewage pipe inspection (KURT2). The robot is equipped with six wheels, twelve infrared, and two ultrasonic sensors as well as other sensors not used in the work described here. The robot carries a conventional laptop, which runs the control software.

Fig. 1. The circular test bed with several objects (triangles, cylinders and boxes).

The test environment of the robot consists of rooms at our offices bounded with small wooden boards. It contains a number of artificial and everyday objects such as boxes of different shapes, bins, etc. as obstacles and objects.

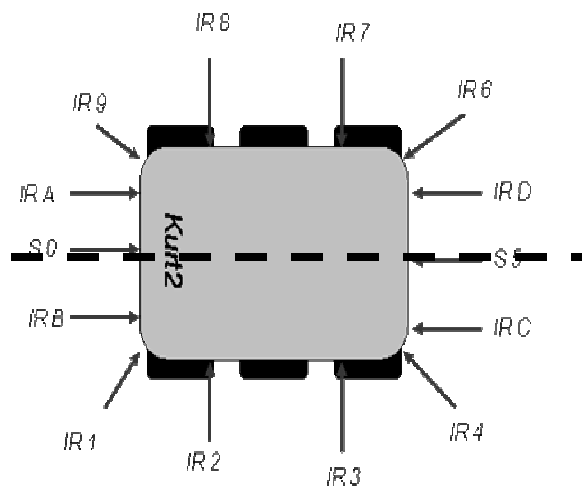

Fig. 2. The sensor space of KURT2 is topologically separated into two hemispheres and consists mainly of infrared distance sensors (IR) and sonar distance sensors (S)

For our experiments, the robot control software makes the robot drive along walls and from time to time randomly change direction to explore the whole space. Pre-wired wall-following behaviour ensures that the robot experiences the objects in similar ways whenever it passes them.

In the experiments described here, a teacher "pointed out" objects such as boxes, bins, triangles etc. to the robot. The teacher let the robot engage and then labelled the objects. The more often an object was seen by the robot, the more stable the object representation became through prototype creation.

In our test assembly, the robot perceived each object three times, but no prototypes were built. In this way, three different representations of each object should be created. The main purpose of the environment here was to prove that learned objects can be anchored and recognised. We used four different object types: a small cylinder, a large box, a medium sized box and a triangular-shaped cardboard. However, the two boxes were very similar for the human observer as well as for the robot. The circular shape of the environment as well as some characteristics of the surroundings, such as slippery ground etc., forced the robot to carry out many turns along its path to make sure that the robot never actually passed an object in exactly the same way twice.

Our mobile robot's sensors provide measurements at a frequency of 10 Hertz. We use an array of 16 sensors (twelve infrared, two ultrasonic and odometric data of the two motors) for further processing.

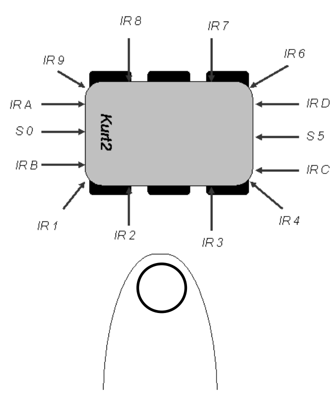

Fig. 3. The perception of an object produces a strongly object related echo in the sensor readings.

Sensor readings now represent the distance between the sensors' position on the robot and objects in the environment. To identify an object the robot needs to move, because only based on the stream of sensors the robot is able to separate objects from its surroundings. This fact, of course, results in a strong coupling between the robot's actions and its perceptions. Anytime the robot passes an object, the sensors show a strongly object related pattern, which is dependant on the robot's speed, the distance to the object, the objects shape, and (for some objects) the robot's heading direction. Figure 3 depicts an idealized representation of the sensor readings for a trash bin. The sensor stream indicates a tube-like shape (i.e. it represents also depth information about the perceived object.).

Problems arise if objects are positioned too close to each other, i.e. one object gets into the range of the front sensors while another one is perceived by the rear ones etc. This means, objects should be positioned isolated from each other or at least in a way so that none of them is located in front of each other.

In case that objects are to close to another to be perceived separately we have conceived two mechanisms to still enable object categorisation and anchoring.

On the one hand, sensor readings are weighted by their positions inside the object. We use a stretched sinus curve to provide higher weights to points that are near to the center of objects sequences and vice versa (Fig 4. depicts the weighting function on a typical sensor stream that results from perceiving a box).

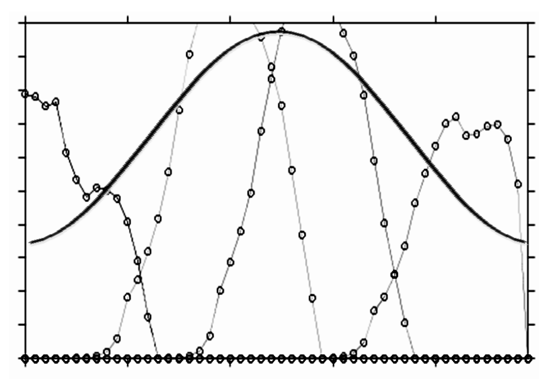

Fig. 4. Weighting of the compared object segments.

On the other hand we use features of the Isomap (cf. next section) to compute object probabilities. Based on the position on the sensor map we are able to make predictions about the certainty of perceiving one object against all others. This probability rises with the number of points that are detected by one specific object. Using this mechanism also enables us to separate objects from each other by having a point sequence that belongs to one object as well as to another by using their specific sequence probabilities.

The architecture

Isomap

The Isomap Algorithm was originally developed by Joshua Tenenbaum [Tenenbaum *et al*. 2000] and [Tenenbaum *et al*. 2002]. The underlying idea is to extract meaningful dimensions from multidimensional data based on measuring the distance between the data points within the geometric form that arises through the particular properties of the non-linear data. Basically, the algorithm captures the intrinsic geometry of the data surface based on this geodesic distance measurement. The transformation is then realised by combining a classical MDS-algorithm with this geodesic distance. Like in MDS, data points which are close in the original space will be close in the lower-dimensional embedding.

Measuring the intrinsic geometry would be relatively simple for linear relationships in the data, because linear data are a set of plains in the geometric space. Here a number of reliable methods such as PCA (principal component analysis) exist [Jollie, 1986]; non-linear data, however, generates more complex structures.

Most existing non-linear MDS algorithms are tuned to specific shapes. This is also true for Tenenbaum's algorithm which is best suited for data point shapes that can be flattened out, e.g. cylinders. The Isomap algorithm tries to capture the global structure of the data which makes it useful for our application. Nevertheless, in further work, other MDS algorithms should be tested and compared against the Isomap.

The underlying (or intrinsic) structure of the data can be assumed to be on a manifold, a non-linear lower dimensional subspace embedded in the input space. Using this assumption, [Tenenbaum *et al.*, 2002] defines the Isomap algorithm to perform non-linear data reduction and uses the following three main steps:

1. Construct spatially local neighbourhoods (using a radius threshold or nearest neighbours).
2. Compute a matrix of all-pairs shortest paths distances.
3. Perform classical MDS using the matrix.

Mapping algorithm

After the robot has built the Isomap and has learned an object representation, it starts moving around and tries to detect objects. Each data-point will be mapped onto the Isomap. Through this mapping the data becomes less complex and noisy and therefore easier to work on. The now acquired low-dimensional point will then be compared to the first points of the Isomap representation of the learned objects. If they are similar further data points will be compared to the following points of the affected objects.

If a sequence of points cannot be mapped onto the Isomap, because it was not seen before, then the sequence and its local environment are stored in a list. If this list of unmapped points grows too large the Isomap needs to be rebuilt.

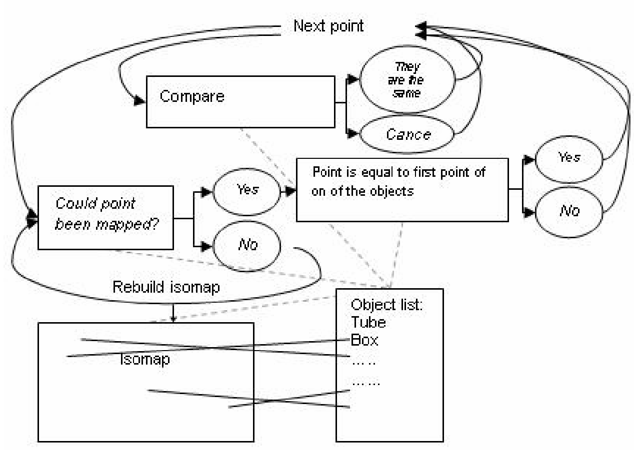

Fig. 5. The mapping algorithm maps sequences of high dimensional sensor streams onto the Isomap and compares the mapped representation with the objects witch have been learned.

Mapping is the process by which a data point (known or unknown) is projected on the map using the Isomap algorithm. The process should be able to map objects in the robot's environment onto the map in way that is robust with respect to interference. In this state objects are sequences of points that have been labelled by a teacher. Currently, for points in the Isomap their high-dimensional representation is still saved for practical reasons. Each point thus consists of its lower dimensional Isomap representation and the original data point. For any new point, the mapping algorithm now searches for the nearest data point in the original space among all stored points.

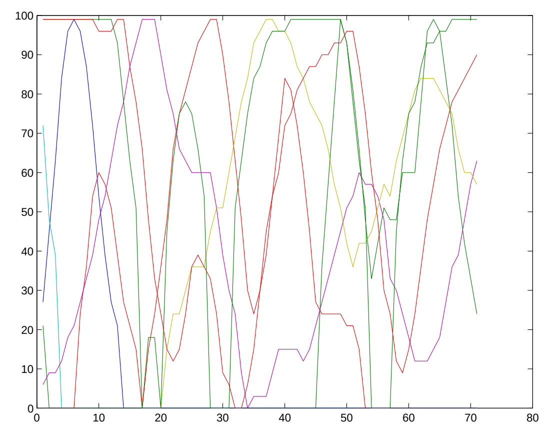

Fig. 6. Multivariate data stream (10 sensors)

If the distance to the nearest point is above a specific empirical threshold, this means that the point cannot be mapped.

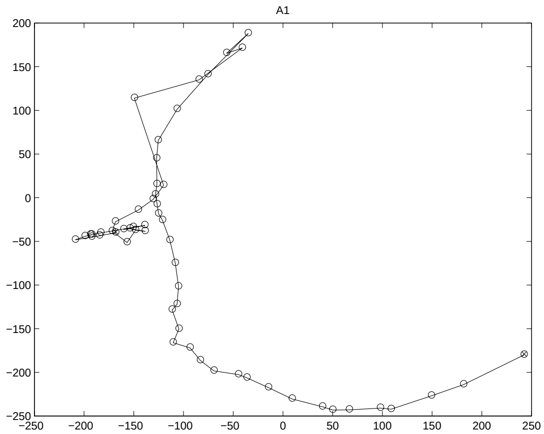

Fig. 7. Isomap (mapped stream, two dimensions) Preprocessed data stream of a perceived object and its representation on the Isomap.

Comparison of sensor segments

Objects are compared by incrementally comparing sequences of points. For sequence comparison, sequences of data points are mapped onto the Isomap; we then compare these resulting sequences.

Points that are generated when the robot passes similar objects will in many cases generate similar sequences of points on the Isomap. The advantage of the resulting sequence is their lower dimensionality and, of course, the possibility of an easy visualization of the mapped data. Since points are usually mapped on already existing points on the map, the mapping is quite robust as long as the robot's behaviour does not change significantly. For example, the sensor data generated as the robot passes a tube at two different levels of speed will generally look different from each other since it includes noise. In the Isomap, sequences will often look quite similar to each other. The mapping of single data points eliminates a great deal of noise; however, in some case the quantisation may actually produce additional errors. Such situations are depicted in Figure 8:

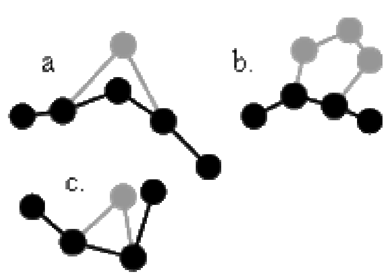

Fig. 8. Typical errors when comparing stored sequences of points on the Isomap for an object. a.) Mapping error: a point may come to lie on another point on the map through quantisation. b.) Objects result in similar but not the same sequences, e.g. when passing the object at a different angle. c.) Time warping error: after passing the same object at different speeds the segments are highly similar but differ in the number of points.

If we would take a simple point-wise Euclidean distance measure, these errors can severely affect recognition of sequences representing the same object. Such errors should be taken into account, but only to a limited extent. The algorithm for comparison must clearly discriminate between different objects which is simply not achievable by a point-to-point comparison. Instead, we use a simplified variant of dynamic time warping [Oates *et al*., 2000]. Sequences are compressed and dilated so as to minimize the errors between the compared sequences. The amount of compression and dilation must operate within certain plausible limits given by system properties such as typical speed of the robot, sensor properties, object size, etc.

In the following description of the algorithm there are two sensor segments A, B (point lists) which should be compared and for which the error should be returned. The algorithm works like a "pair-wise Euclidian distance" measure, but tries to minimize errors between small segments by skipping over parts in which segments are dissimilar.

```
While (there are some points left in A
and B)
{
   1. a = first point of A,
      b = first point of B
   2. d = D(a, b)
   3. error = error + d
   4. if ( D(first point of A, first
      point of B) > smooth threshold) )
         a. then search a similar point
            to a in the next 5 points
            in B,
         b. and search a similar point
            to b in the next 5 points
            in A
   5. errorT = 0, errorA=0, errorB=0;
   6. if a point is found in 4a then
      errorB = mean(D(ignored points,
      a))
   7. if a point is found in 4b then
      errorA = mean(D(ignored points,
      b))
   8. if errorA > errorB then
         a. errorT = errorB
            set point found in 4a, as
            current point in list B
         b. else errorT = errorA
            set point found in 4b, as
            current point in list A
   9.  error = error + errorT
```

```
    10.  A = A.next, B = B.next
}

Where D(x, y) denotes Euclidian
distance.
```

Results

As an evaluation of the performance of the algorithm we compared four different algorithms using four different objects: a box (a), a triangle (b), and two very similar tubes (c, d). The algorithm should discriminate the box, the triangle and the tubes, but the tubes should be recognized as the same object. In the test run, the robot performed three circles in the test bed. It perceived each object three times from different angles and therefore generated three different but highly similar representations of each of the objects. Each representation was then compared to all the other representations.

The contingency table shows a summary of the results: It depicts how often a representation of an object was correctly recognised. The diagonal line and the cells “d-c” (or “c-d”) should contain only values close to 9, meaning that all cases were classified correctly. All other cells should be close to zero, respectively.

We used 4 algorithms for comparison:

IsoDTW: Our pseudo DTW algorithm (descript in the section before) comparing the two representations mapped on to the Isomap (2 dimensional).

IsoECL: Euclidian distance calculated on the shapes of the Isomap representation of the object sensor streams.

DatDTW: Our pseudo DTW algorithm calculated on the object sensor streams (dimension of sensor space was 10).

DatECL : Euclidian distance calculated on the object sensor streams (dimension of sensor space was 10).

In the following we refer to the experimental setup where we used only ten of the robot’s sixteen sensors (the six infrared sensors of its right hemisphere, plus the front and rear ultra sonic and two motor values). All sensors have been pre-processed to minimize errors and noise and to map the sensors to the same value range using the following two methods:

- Normalisation: $z_i = (x_i - \mu_i) / \sigma_i$
- Low pass filter: Smoothen and cut low sensor values (set to zero)

For the computation of the error we normalise to the length of the shorter of the two compared sequences. If the resulting error falls below a certain threshold the two objects are regarded as similar. These thresholds have been chosen in a way that the number of wrongly classified objects of the most homogeneous objects is minimised and the number of correctly classified objects are maximised. This leads to a very high specificity.

IsoECL	b	c	a	d	***DatECL***	b	c	a	d
b	8	-	-	-	b	6	-	-	-
c	-	6	-	6	c	-	7	-	1
a	2	-	9	-	a	-	-	5	-
d	-	5	-	8	d	-	-	-	5

IsoDTW	b	c	a	d	***DatDTW***	b	c	a	d
b	8	-	-	-	b	6	-	-	-
c	-	7	-	6	c	-	8	-	1
a	1	-	8	-	a	-	-	5	-
d	-	5	-	8	d	-	-	-	5

Tab. 1. Confusion Matrix: Comparison of the four algorithms measuring the similarity of 4 objects. A minus sign ‘-‘ denotes no similarity.

The table clearly shows that by mapping onto the Isomap particularly the two similar objects (c and d), respectively the two cylinders are correctly classified. This classification is based on the characteristics of the Isomap to project similar sensor vectors to neighbouring points. Similar perceptions result in Isomap trajectories that have a small geodesic distance.

The mapping to the “grid” of known points, i.e. quantisation, amplifies similarities and dissimilarities of the categorised objects. However, quantisation also is a not neglectable source of error. Our pseudo dynamic time warping algorithm (see above) helps to minimise this error. The pseudo DTW raises specificity and at the same time reduces sensitivity of the categorisation process, as can be seen in the following table:

IsoECL	similarity	No similarity
No similarity	2	**88**
similarity	**42**	12

IsoDTW	Pos	No similarity
No similarity	1	**89**
similarity	**42**	12

Tab. 2a. This table shows the summarised confusion matrix for the two Isomap algorithms.

DatECL	similarity	No similarity
No similarity	0	**90**
similarity	**24**	30

DatDTW	similarity	No similarity
No similarity	0	**90**
Similarity	**25**	29

Tab. 3b. This table shows the summarized confusion matrix for the two algorithms that operate on the unmodified sensor streams.

IsoECL	
Specificity	0,98
Sensitivity	0,78

IsoDTW	
Specificity	0,99
Sensitivity	0,78

DatECL	
Specificity	1
Sensitivity	0,44

DatDTW	
Specificity	1
Sensitivity	0,46

Tab. 4. The algorithm on the Isomap representation exhibits higher sensitivity.

These results (Table 2a, 2b and Table 3) emphasize the features of our approach. The two algorithms (*IsoECL*, *IsoDTW*) for comparing the segment on the Isomap have both a very high sensitivity and also a high specificity. On the Isomap, the algorithms were able to detect similarities, which the algorithms do not detect in the multidimensional sensor stream.

Conclusion

We have presented an approach for anchoring multivariate motor and distance sensor data streams based on real-world robot experiments. The presented algorithm for sequence comparison promises to robustly group sequences for clustering similar objects. These clusters can be labelled with category names given from a teacher. We presented first evaluations of our algorithms in comparison to others.

Further work will extend the testing of our algorithm: it is necessary to study other MDS-algorithms to decide which suits best to our problem. More tests will also be necessary to investigate different environments and numbers of sensors and the way in which they affect the construction of the Isomap. Further work will also extend the method with a recombination algorithm so that it will be possible to construct prototypes which would make the recognition of objects more stable. Finally, Isomap could be combined with some sort of Markov model so as to predict experiences.

Acknowledgements

This research is supported by the European Commission's Information Society Technology Programme project SIGNAL, IST-2000-29225. Partners in this project are the University of Bonn (D), Napier University (UK), Istituto di Elettronica e di Ingegneria dell'Informazione e delle Telecomunicazioni at Genoa (I), and the Austrian Research Institute for AI, which is also supported by the Austrian Federal Ministry for Education, Science, and Culture and the Austrian Federal Ministry for Transport, Innovation, and Technology.

References

Coradeschi, S., Saffioti, A. 2003. Perceptual anchoring – anchoring symbols to sensor data in single and multiple robot systems. *Special issue of Robotics and Autonomous Systems,* Vol. 43 (1-2).

Harnad, S. 1990. The Symbol Grounding Problem, *Physica D*, 42, pp. 335-346.

Prem, E., Hoertnagl, E., Poelz, P.M. 2003. Anchoring symbols in hybrid autonomous systems using Isomap sequences. *Proc. Third Int. Conf. on Hybrid Intelligent Systems*, Melbourne.

Tenenbaum, J.B., de Silva, V., Langford, J.C. 2000. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319-2323.

Tenenbaum, J.B., de Silva, V., Langford, J.C. 2002. The Isomap Algorithm and Topological Stability. *Science*, 295(5500):7a.

Jollie, I.T. 1986. Principal Component Analysis. *Springer-Verlag*, New York.

Oates, T., Schmill, M., Cohen, P. 2000. A Method for Clustering the Experiences of a Mobile Robot that Accords with Human Judgments. University of Massachusetts, Amherst, MA, USA.