Efficient Serial Associative Memory

David Wilkes[1] and John K. Tsotsos
Department of Computer Science
University of Toronto
Toronto, Canada, M5S 1A4
E-mail: wilkes@vis.toronto.edu

Abstract

We present probabilistic algorithms for efficient storage and retrieval of sets of feature vectors, given a known error process operating on the query set, that perturbs the query set away from the corresponding stored set. The algorithms operate by mapping each set to a corresponding generalized indicator vector, then performing a pruned search of a tree containing stored indicator vectors. The pruning is based on the probability of the query, given the stored items below the current position in the tree. Analysis and trial results show that this approach requires less total computation than existing methods based on parallel architectures. The indicator vector retrieval method can also cope efficiently with query vectors of much higher dimensionality than existing serial algorithms for nearest-neighbour searches.

1 Introduction

Much current work in object recognition concerns generation of invariant or quasi-invariant scalars or vectors from image data, to be used to index into a database of models [13, 6, 10, 11, 12, 8, 3]. Given such feature vectors, the remaining problem is to use the feature vectors to avoid traversal of the entire database, by narrowing the database retrieval somehow. The typical approach is to do lookups of query vectors in a multidimensional grid or a one-dimensional hash table [6, 10, 11, 8, 4, 3].

Several authors have attempted to address the problem of error in the query feature vector by various means. Example methods are to store models in multiple neighbouring buckets in the feature or index space [13], or to query multiple buckets in a neighbourhood in the space [1], or to adjust bucket size to compromise between probability of correct retrieval and specificity of the index [10].

We attempt to achieve both specificity of indexing, in order to touch as little of the database as possible, and responsiveness to the possibility of significant errors in the query vectors. We assume that each object is characterized by a *set* of m-dimensional feature vectors that is stored for the object (e.g. a set of projective invariants or a set of line segment parameter-tuples). The vision system then generates a query set of feature vectors from sensor data, that will be a perturbation of a stored set. We assume that the system designer has a model of the error process that perturbs the query vector set away from the corresponding stored vector set. Our goal is then to retrieve the stored set maximizing the probability of the observed query set, under the model error process.

Our approach is to divide the problem into two subproblems. First, we map each feature vector set into a corresponding *generalized indicator vector* of high dimensionality. The vector produced is invariant to the order in which the set members are specified. This allows us to avoid the combinatorial difficulties associated with having to determine the correspondence between the individual feature vectors in the stored and query sets. Then, the remaining problem is the storage and retrieval of high-dimensional vectors, subject to the model error process.

Related work on *nearest-neighbour* and *near-neighbour* problems [2, 7, 1] in n-dimensional spaces proves to be inadequate for n larger than approximately 8. Work by Willshaw [15] has the disadvantage of a fixed storage capacity for a given error rate and vector length. Finally, Kanerva's parallel architecture [9] requires too many computations for each storage and retrieval operation.

2 Mapping feature vector sets to indicator vectors

We consider three techniques for producing a generalized indicator vector. Details are available in [14]. The first divides a normalized m-dimensional feature space into n hypercubic buckets. The n-dimensional indicator vector has a 1 at each position i if the i^{th} bucket contains a feature vector, and has a 0 at the i^{th} position otherwise. This straightforward tessellation of the feature space corresponds to replacing each feature vector in the set with a representative vector occupying a position in a hypercubic lattice.

An improvement to this scheme is instead to use a more tightly-packed lattice. Our analysis and experiments show that a *checkerboard* lattice (see, for example, [5]) results in fewer feature vectors being displaced into different cells given a certain error rate and indicator vector dimensionality n.

The improvement is illustrated in figure 1 where we show the probability of a query feature vector falling in a different cell than the corresponding stored vector, as a function of the indicator vector length. Our error model is the simplest capturing both measurement error and the possibility of missing features: we assume a uniform distribution for feature vector location within a hypersphere of radius r, mixed with a uniform distribution over the entire feature space

[1] also with Ontario Hydro Research Division

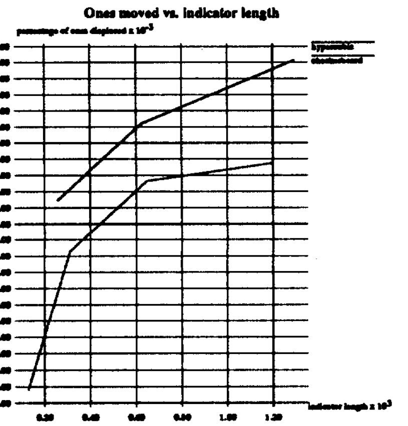

Figure 1: Probability of indicator vector 1 displacement vs. indicator vector size.

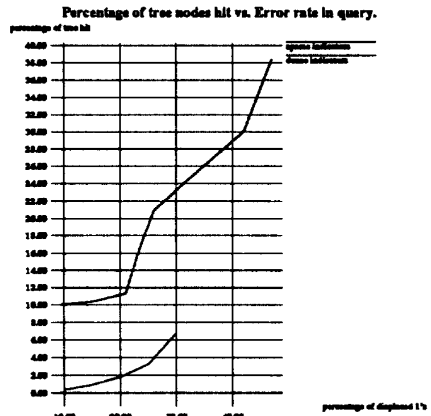

Figure 2: Percentage of index tree hit vs. percentage of indicator bits in error.

with probability f. In the figure, $r = 0.1$ in the feature space $[0, 1)^4$ and $f = 0.1$.

A variation on the above schemes is instead to have each stored indicator vector element give the *probability* that the given object will result in a feature vector occurrence in the relevant cell of the tessellation. This method may be applied if the error model is not known in advance, by collecting feature vector sets for multiple images of each object to be stored. We found the performance of such a scheme to be comparable to that of the binary scheme using a known error model.

3 Indicator vector storage and retrieval

Our approach to storage and retrieval of n-dimensional indicator vectors is to store the vectors in one of several standard search trees, depending on their sparsity and whether they have real or binary elements. Details are available in [14].

Given a query indicator vector $\mathbf{q}$, the retrieval algorithms are essentially best-first searches of the tree for the stored vector $\mathbf{u}$ maximizing the probability $p(\mathbf{q}|\mathbf{u})$ of the query vector, given the stored vector, under the assumed error model.

Figure 2 shows typical percentages of the index tree hit as a function of the percentage of displaced 1's, for both sparse (16 1's in a 648-D indicator) and dense indicator vectors (500 1's in a 1000-D indicator), with successful retrieval rates of 94% and 98% respectively. In each case, 10000 vectors were stored.

Acknowledgements The first author is indebted to Professor Geoffrey Hinton for suggesting the use of a real-number-based indicator vector to capture variation in feature vector position. The authors wish to thank the Natural Sciences and Engineering Research Council of Canada, the Information Technology Research Centre, and the Canadian Institute for Advanced Research for financial support. The authors are members of the Institute for Robotics and Intelligent Systems (IRIS) and wish to acknowledge the support of the Networks of Centres of Excellence Program of the Government of Canada and the participation of PRECARN Associates Inc. The second author is the CP (Unitel) Fellow of the Canadian Institute of Advanced Research. We would also like to thank the two anonymous reviewers for their valuable comments.

References

[1] Bentley, J.L., Weide, B.W., and Yao, A.C., "Optimal Expected-Time Algorithms for Closest Point Problems," *ACM Transactions on Mathematical Software*, Vol. 6, No. 4, ACM, December 1980.

[2] Bentley, J.L., "Multidimensional Divide-and-Conquer," *Communications of the ACM*, Vol. 23, No. 4, ACM, April 1980.

[3] Califano, A., and Mohan, R., "Multidimensional Indexing for Recognizing Visual Shapes," *Proc. Computer Vision and Pattern Recognition '91*, IEEE Computer Society Press, Washington, 1991.

[4] Clemens, D.T., and Jacobs, D.W., "Space and Time Bounds on Indexing 3-D Models from 2-D Images," *IEEE PAMI*, Vol. 13, No. 10, 1991.

[5] Conway, J.H., and Sloane, N.J.A., *Sphere Packings, Lattices and Groups*, Springer-Verlag, New York, 1988.

[6] Forsyth, D., Mundy, J.L., Zisserman, A., and Brown, C.M., "Invariance - A New Framework for Vision," Proc. Third ICCV, 1990.

[7] Friedman, J.H., Bentley, J.L., Finkel, R.A., "An Algorithm for Finding Best Matches in Logarithmic Expected Time," *ACM Transactions on Mathematical Software*, Vol. 3, No. 3, ACM, September 1977.

[8] Jacobs, D.W., "Space Efficient 3D Model Indexing," *Proc. CVPR '92*, 1992.

[9] Kanerva, P., *Sparse Distributed Memory*, MIT Press, Cambridge, 1988.

[10] Lamdan, Y., and Wolfson, H.J., "Geometric Hashing: A General and Efficient Model-Based Recognition Scheme," *Proc. Second International Conference on Computer Vision*, IEEE Computer Society Press, Washington 1988.

[11] Rothwell, C.A., Zisserman, A., Mundy, J.L., and Forsyth, D.A., "Efficient Model Library Access by Projectively Invariant Indexing Functions," *Proc. CVPR '92*, 1992.

[12] Stark, L. and Bowyer, K., "Indexing Function-Based Categories for Generic Recognition," *Proc. CVPR '92*, 1992.

[13] Stein, F. and Medioni, G., "Recognition of 3-D Objects from 2-D Groupings," *Proc. Image Understanding Workshop*, 1992.

[14] Wilkes, D., *Active Object Recognition*, Ph.D. Thesis, Department of Computer Science, University of Toronto. 1993 (Expected).

[15] Willshaw, D., "Holography, Associative Memory, and Inductive Generalization," in *Parallel Models of Associative Memory*, eds. Hinton and Anderson, Lawrence Erlbaum Associates, Hillsdale, 1981.