Efficient Serial Associative Memory

David Wilkes¹ 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 correspond-

¹also with Ontario Hydro Research Division
with probability $f$. In the figure, $r = 0.1$ in the feature space $[0,1]^d$ 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 $q$, the retrieval algorithms are essentially best-first searches of the tree for the stored vector $u$ maximizing the probability $p(q|u)$ 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 48-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 (Unidel) Fellow of the Canadian Institute of Advanced Research. We would also like to thank the two anonymous reviewers for their valuable comments.

References


