- The second challenge is associated with the lack of range information when camera measurements are the only measurements available. Camera's measurements consist only of bearings to specific feature points in images. The PFT data type is especially challenging inasmuch as recognized features do not necessarily represent known objects and do not contain location information.
- The third challenge is posed by the fact that computer vision information often relates to images taken in the past. For example, the PFT data type reports features that were recognized as being common to two images taken at earlier times. The need to update the current state estimate by use of information from the past presents a challenge because prior recursive state-estimating algorithms typically only propagate the current state.

The present algorithm addresses these challenges by incorporating the following innovations:

The first innovation is a preprocessing step, based on QR factorization (a particular matrix factorization, a description of which would exceed the scope of this article), that provides for optimal compression of LMT, PFT, and RPT updates that involve large numbers of recognized features. This compression eliminates the need for a considerable amount of real-time computation.

The second innovation is a mathematical annihilation method for forming a linear measurement equation from the PFT data. The annihilation method is equivalent to a mathematical projection that eliminates the dependence on the unknown scale factor.

The third innovation is a state-augmentation method for handling PFT and other data types that relate states from two or more past instants of time. The state-augmentation method stands in contrast to a prior stochastic cloning method. State augmentation provides an optimal solution to the state-estimation problem, while stochastic cloning can be shown to be suboptimal.

This work was done by David Bayard and Paul Brugarolas of Caltech for NASA's Jet Propulsion Laboratory.

The software used in this innovation is available for commercial licensing. Please contact Karina Edmonds of the California Institute of Technology at (626) 395-2322. Refer to NPO-41321

Representing Functions in *n* Dimensions to Arbitrary Accuracy

Computation can be simplified in cases in which data are noiseless.

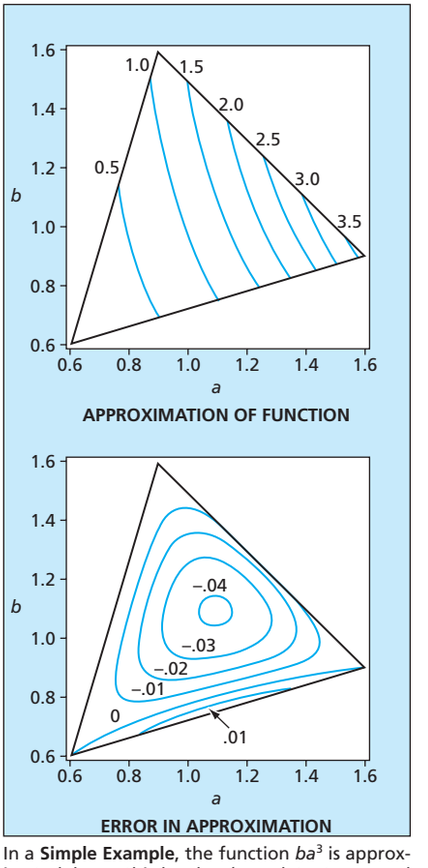
Langley Research Center, Hampton, Virginia

A method of approximating a scalar function of n independent variables (where n is a positive integer) to arbitrary accuracy has been developed. This method is expected to be attractive for use in engineering computations in which it is necessary to link global models with local ones or in which it is necessary to interpolate noiseless tabular data that have been computed from analytic functions or numerical models in n-dimensional spaces of design parameters.

This method is related to prior statistically based methods of fitting low-order approximate functional representations (response surfaces) to noisy experimental data. The prior methods are advantageous in situations in which large amounts of noisy data are available, but in situations in which the data and the functions that they represent are noiseless, it is computationally inefficient to generate the large quantities of data needed for fitting. Moreover, in the prior statistically based methods, the low-order functional representation cannot be defined to a specified degree of accuracy. The latter shortcoming limits the usefulness of response surfaces in design-optimization computations because (1) optimization calculations involve gradients of functions, which are approximated to orders lower than those of the functions themselves and, hence, can be so inaccurate as to yield poor results; and (2) the accuracy of a response surface can vary widely over its domain. The present method overcomes these shortcomings of the prior methods.

Increasingly, modern computational-simulation programs generate values of gradients of functions in addition to values of the functions themselves, in order to satisfy the need for accurate gradient as well as function values for optimizations. Taking advantage of this trend, the present method relies on the availability of both gradient and function data. In this method, the space of n independent variables is subdivided into an n-dimensional mesh of simplex elements (simplices) that amount to n-dimensional generalizations of modeling techniques used in the finite-element method. The exact values of the scalar function and its gradient, as generated by the applicable computational model, are specified at the simplex nodes, which are intersections of coordinate axes of the n-dimensional mesh. Within each simplex, the function and its gradient are interpolated approximately by a set of basis functions of the n coordinates.

In order to minimize the computational burden, one tries to use basis functions of order no higher than that needed to limit the error in the approximation to an acceptably low value. It would be preferable if, in a given case,



In a **Simple Example**, the function ba^3 is approximated by a third-order (complete to second order) polynomial on a triangular simplex. The error is zero at the nodes of the simplex and greatest near the middle.

one could obtain acceptable accuracy from polynomial functions of order no higher than third, complete to second order (see figure). The advantage of using such low-order polynomials is that the interpolation could be performed without need for matrix operations (which would, if needed, add to the computational burden). Approximate-error-indicator quantities, defined on the edges of the simplices, have been derived as guides to whether there is a need to refine the simplices to reduce the errors.

This work was done by Stephen J. Scotti of Langley Research Center. Further information is contained in a TSP (see page 1).
LAR-16297-1

Accumulate-Repeat-Accumulate-Accumulate Codes

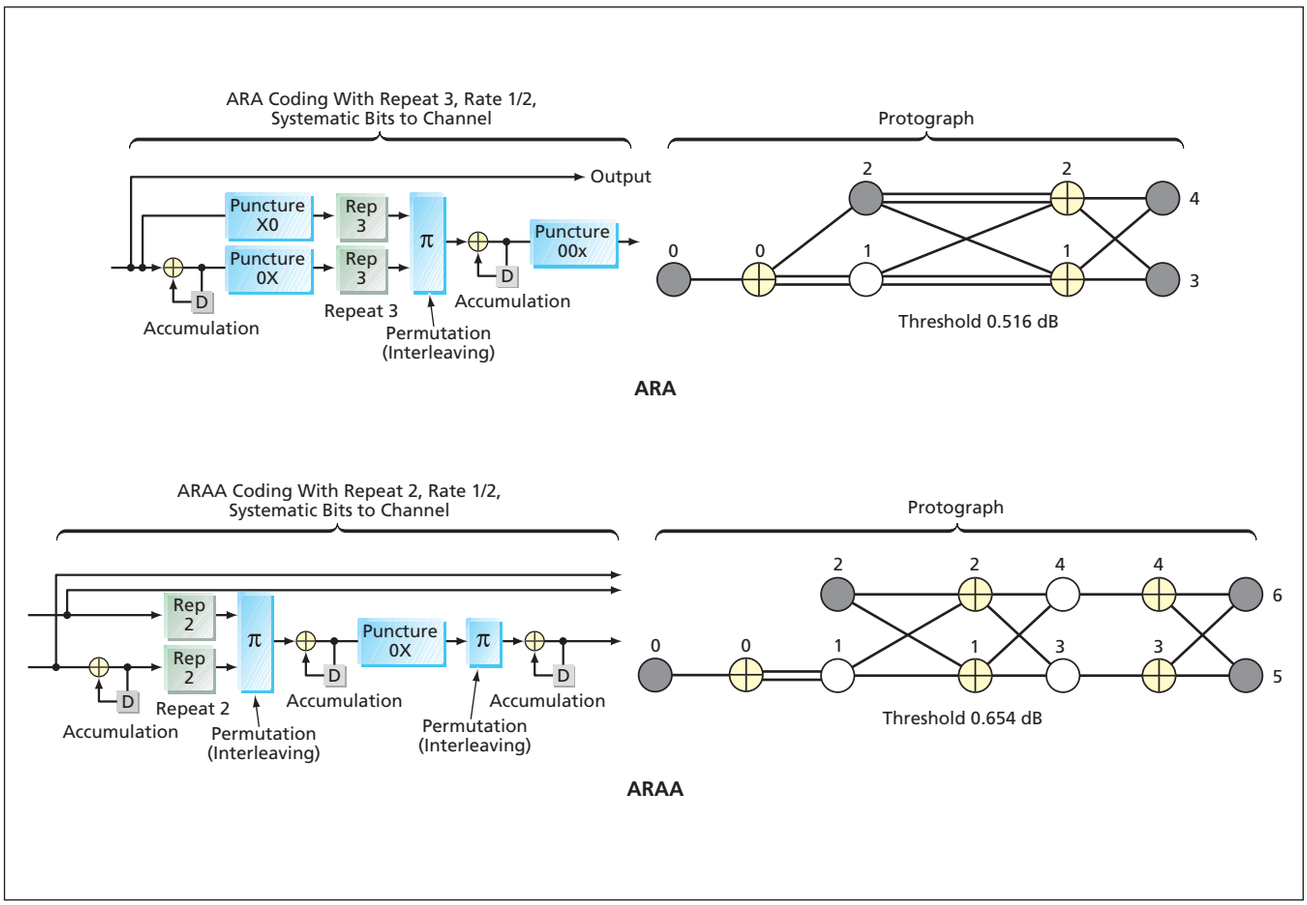
Fast, high-performance coders and decoders could be designed.

NASA's Jet Propulsion Laboratory, Pasadena, California

Accumulate-repeat-accumulate-accumulate (ARAA) codes have been proposed, inspired by the recently proposed accumulate-repeat-accumulate (ARA) codes. These are error-correcting codes suitable for use in a variety of wireless data-communication systems that include noisy channels. ARAA codes can be regarded as serial turbo-like codes or as a subclass of low-density parity-check (LDPC) codes, and, like ARA codes they have projected graph or protograph representations; these characteristics make it possible to design high-speed iterative decoders that utilize belief-propagation algorithms. The objective in proposing ARAA codes as a subclass of ARA codes was to enhance the error-floor performance of ARA codes while maintaining simple encoding structures and low maximum variable node degree.

A rate-1/2 classical repeat-and-accumulate (RA) code has a high threshold (3.01 dB). An ARAA code can be viewed as a preceded RA code with puncturing in concatenation with another accumulation, wherein the preceding is also simply an accumulation; these characteristics make it possible to design very fast encoders. The top part of the figure illustrates the simplest example of the encoding process for a rate-1/2 ARA code, its protograph (filled nodes correspond to transmitted code symbols), and the corresponding decoding threshold of 0.516 dB. Other rate-1/2 ARA examples with maximum variable node degree 5 have thresholds as low as 0.26 dB, which can be compared to the Shannon capacity limit of 0.19 dB.

The bottom part of the figure illustrates a simple example of the encoding process for a rate-1/2 ARAA code, its protograph, and the corresponding



These **Block Diagrams and Protographs** illustrate the similarities and differences between a simple rate-1/2 ARA code and a simple rate-1/2 ARAA code.