Fast Query-Optimized Kernel-Machine Classification

Computation is accelerated by an order of magnitude, without loss of accuracy.

NASA's Jet Propulsion Laboratory, Pasadena, California

A recently developed algorithm performs kernel-machine classification via incremental approximate nearest support vectors. The algorithm implements support-vector machines (SVMs) at speeds 10 to 100 times those attainable by use of conventional SVM algorithms. The algorithm offers potential benefits for classification of images, recognition of speech, recognition of handwriting, and diverse other applications in which there are requirements to discern patterns in large sets of data.

SVMs constitute a subset of kernel machines (KMs), which have become popular as models for machine learning and, more specifically, for automated classification of input data on the basis of labeled training data. While similar in many ways to *k*-nearest-neighbors (*k*-NN) models and artificial neural networks (ANNs), SVMs tend to be more accurate. Using representations that scale only linearly in the numbers of training examples, while exploring nonlinear (kernelized) feature spaces that are exponentially larger than the original input dimensionality, KMs elegantly and practically overcome the classic "curse of dimensionality." However, the price that one must pay for the power of KMs is that query-time complexity scales linearly with the number of training examples, making KMs often orders of magnitude more computationally expensive than are ANNs, decision trees, and other popular machine learning alternatives.

The present algorithm treats an SVM classifier as a special form of a *k*-NN. The algorithm is based partly on an empirical observation that one can often achieve the same classification as that of an exact KM by using only small fraction of the nearest support vectors (SVs) of a query.

The exact KM output is a weighted sum over the kernel values between the query and the SVs. In this algorithm, the KM output is approximated with a *k*-NN classifier, the output of which is a weighted sum only over the kernel values involving *k* selected SVs. Before query time, there are gathered statistics about how misleading the output of the *k*-NN model can be, relative to the outputs of the exact KM for a representative set of examples, for each possible *k* from 1 to the total number of SVs. From these statistics, there are derived upper and lower thresholds for each step *k*. These thresholds identify output levels for which the particular variant of the *k*-NN model already leans so strongly positively or negatively that a reversal in sign is unlikely, given the weaker SV neighbors still remaining.

At query time, the partial output of each query is incrementally updated, stopping as soon as it exceeds the predetermined statistical thresholds of the current step. For an easy query, stopping can occur as early as step $k = 1$. For more difficult queries, stopping might not occur until nearly all SVs are touched. A key empirical observation is that this approach can tolerate very approximate nearest-neighbor orderings. In experiments, SVs and queries were projected to a subspace comprising the top few principal-component dimensions and neighbor orderings were computed in that subspace. This approach ensured that the overhead of the nearest-neighbor computations was insignificant, relative to that of the exact KM computation.

This work was done by Dominic Mazzoni and Dennis DeCoste of Caltech for **NASA's jet Propulsion Laboratory**. *Further information is contained in a TSP (see page 1).*

The software used in this innovation is available for commercial licensing. Please contact Don Hart of the California Institute of Technology at (818) 393-3425. Refer to NPO-40441.

Indentured Parts List Maintenance and Part Assembly Capture Tool — IMPACT

Viewing and maintaining the complex assembly hierarchies of large databases is made easier.

Lyndon B. Johnson Space Center, Houston, Texas

Johnson Space Center's (JSC's) indentured parts list (IPL) maintenance and parts assembly capture tool (IMPACT) is an easy-to-use graphical interface for viewing and maintaining the complex assembly hierarchies of large databases. IMPACT, already in use at JSC to support the International Space Station (ISS), queries, updates, modifies, and views data in IPL and associated resource data, functions that it can also perform, with modification, for any large commercial database. By enabling its users to efficiently view and manipulate IPL hierarchical data, IMPACT performs a function unlike that of any other tool. Through IMPACT, users will achieve results quickly, efficiently, and cost effectively.

Speed, efficiency, and cost are critical issues in maintaining complex assembly hierarchies of large databases. IPLs consist of parts organized into such complex assembly hierarchies. The more complex the hierarchy, the more the associated list grows and the more difficult it becomes to locate a part to modify it. At JSC it was found that existing IPL manipulation methods were too complex, hard to use, and error-prone for time- and cost-sensitive ISS operations. IMPACT was therefore developed to address these drawbacks and to help users achieve results.

IMPACT uses a C++, X-Windows, and Motif application framework. At JSC, it operates with a PRO*C++ interface to an Oracle database. In this way, IMPACT can manipulate the vehicle master database IPL with PL/SQL packages. Since it was developed using object-oriented programming in a modular fashion, it has proved easy to maintain and its capabilities are as easily extended. IMPACT has shown a very high reliability factor as well.

IMPACT manages a rapidly changing flight sequence, manifests, and detailed parts list for ISS by featuring views of an ISS IPL based on flight phases (i.e., launch, on-orbit, and return) and flights. It also features resource data viewing for each part in the IPL and a hypertext-based help system. IMPACT can be started in "view only" as well as in "update modes." When in update mode, IMPACT supports the creation of database entries for new flights, elements, subelements, and parts as well as parts movement around assembly hierarchies, using menu-driven commands and buttons, and drag-and-drop technology. More than one IMPACT session can be brought up independent of another, and different views can be placed side-by-side on the same screen during the same session.

IMPACT is therefore a unique, flexible tool with an easy-to-use, highly intuitive graphical user interface. Its novelty lies in the fact that it allows users to view and manipulate IPL hierarchical data efficiently, something no other tool has allowed during the time of this reporting. Already in use on the ISS, IMPACT has proven to be flexible and can mature and grow with a system. As such, it is a valuable adjunct not only to the space industry for which it was developed but, with suitable modifications, to large commercial databases.

This work was done by Bobby Jain, Bill Morris, and Kelly Sharpe of Barrios Technology for **Johnson Space Center**. *For further information, contact*

Barrios Technology, Inc.
2525 Bay Area Blvd., Suite 300
Houston, TX 77058-1556
Phone: (281) 280-1900
Fax: (281) 280-1901

Refer to MSC 22915, volume and number of this NASA Tech Briefs *issue, and the page number.*

An Architecture for Controlling Multiple Robots

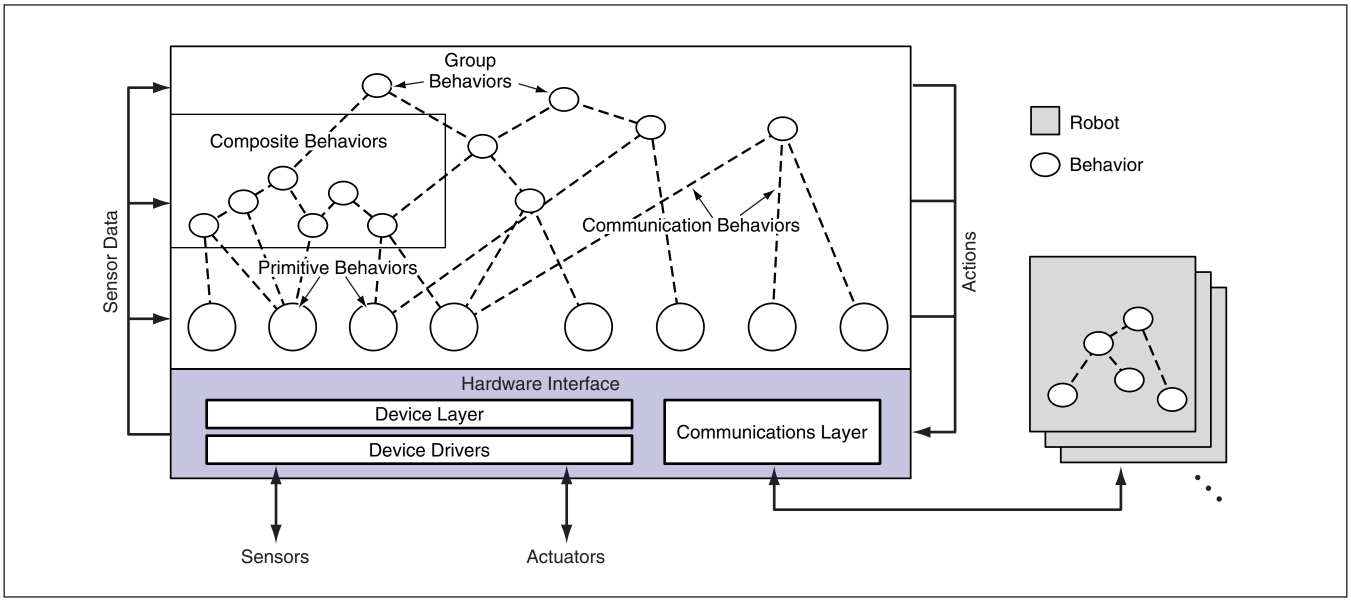
Hierarchies of behaviors can be constructed and coordinated with great versatility.

NASA's Jet Propulsion Laboratory, Pasadena, California

The Control Architecture for Multirobot Outpost (CAMPOUT) is a distributed-control architecture for coordinating the activities of multiple robots. In the CAMPOUT, multiple-agent activities and sensor-based controls are derived as group compositions and involve coordination of more basic controllers denoted, for present purposes, as behaviors.

The CAMPOUT provides basic mechanistic concepts for representation and execution of distributed group activities. One considers a network of nodes that comprise behaviors (self-contained controllers) augmented with hyper-links, which are used to exchange information between the nodes to achieve coordinated activities. Group behavior is guided by a scripted plan, which encodes a conditional sequence of single-agent activities. Thus, higher-level functionality is composed by coordination of more basic behaviors under the downward task decomposition of a multi-agent planner (see figure).

Robotics is a highly multidisciplinary field that requires efficient integration of many components (e.g., perception, mapping, localization, control, and learning) that involve diverse representations, frameworks, and paradigms (e.g., classical control theory, artificially intelligent planners, estimation theory, data fusion, computer vision, utility theory, decision theory, fuzzy logic, and multiple-objective decision making). The CAMPOUT provides a conceptual infrastructure for consolidating diverse techniques to enable the efficient use and integration of these components for meaningful interaction and operation.



The **CAMPOUT Provides for a Hierarchical Organization** of primitive behaviors, composite behaviors built from primitive behaviors, and groups composed from coordination of behaviors across multiple robots. Each robot runs an instance of this architecture and participates in coordination of activities through group behaviors. Coordination is facilitated through communication behaviors.