A HOMOGENEOUS COMPUTATIONAL MODEL FOR SPATIAL INFERENCE ON MASSIVELY-PARALLEL ARCHITECTURES

Mark J. Carlotto
The Analytic Sciences Corporation (TASC)
55 Walkers Brook Dr.
Reading, MA 01867

ABSTRACT

A computational model for 2-d spatial inference on massively parallel SIMD architectures is described. In the model, spatial information is represented by three basic types of parallel variables or pvars: label maps which assign unique numbers to sets of related processors (e.g., the largest cube address of the set of processors representing a connected region), feature maps which contain the property values of related sets of processors, and hypothesis maps which indicate the probability, membership, belief, etc. that a processor set belongs to a particular class. Spatial inference involves the application of parallel operators to pvars, e.g., labeling operators to assign unique labels to related groups of processors that belong to the same class, spatial operators to compute features of connected regions, and inference operators to assign classes to regions based on their properties. The application of the model to a geographic information retrieval problem is described.

1. INTRODUCTION

In geographic information systems, image understanding systems, and other systems that reason about spatial data, a variety of representations are used. Many employ some form of iconic representation (label maps, spatial occupancy arrays, quad trees, etc.) to explicitly delimit the spatial extent of regions in the image space. Iconic representations are usually complemented by some type of non-spatial or symbolic representation such as an attributed graph where the nodes in the graph correspond to regions in the image.The symbolic representation describes properties of and between regions, and provides a place to store hypotheses, and other summary information about the regions. Traditionally, spatial reasoning has been viewed as a process that involves the repeated transfer of information between spatial and symbolic representations. An alternate computational model is described here that is based on a uniform representation for all spatial information (iconic and symbolic) using parallel variables organized in a 2-d grid.

The organization of the paper is as follows. Section 2 summarizes the salient features of the Connection Machine and the *Lisp programming language. Section 3 presents a data-parallel model for spatial reasoning. Section 4 describes some of the operators that have been implemented to date. Application of the model to a geographic information retrieval problem is presented in Section 5.

2. THE CONNECTION MACHINE AND *LISP

The Connection Machine (CM) is a data-parallel computing system containing up to 64K physical processors which can act like millions of virtual processors. The CM, originally conceived by Hillis (Ref. 1), is built by Thinking Machines Corporation (TMC). A description of the CM system can be found in Ref. 2. The CM-2 contains 64K bits per physical processor and can perform 32 bit arithmetic at a rate of 2500 MIPs for a 64K system. The current system configuration at TASC is a 8096 processor CM-2 system with a Symbolics front-end processor and a frame buffer that allows the contents of the CM to be viewed at rates up to a gigabit per second.

*Lisp, a parallel dialect of Common Lisp, and PARIS, the assembly language of the CM are provided within the Symbolics software environment. *Lisp (Ref. 3) is based on objects known as parallel variables or pvars which we shall denote in uppercase Greek letters, e.g., A. Elements of pvars are processors that may be accessed by their cube address (i.e., relative to the hypercube) or their grid address, $\alpha(x,y)$. Elements of pvars may be signed and unsigned integers, variable precision floating point numbers, and booleans. The operation (!! α) returns a pvar in which the value of α has been broadcast to all processors in the currently selected set. Macros such as *when, *cond, and *if select subsets of processors. For example the form (*if (=!! A B) (!! 1) (!! 0)) returns a pvar that contains ones in those elements in which A and B are equal and zeros elsewhere. Functions and macros that operate on all selected processors in parallel are identified by !! suffixes, e.g., (+!! A B). Reducing operations are denoted by a * prefix and return a value from the currently selected set, e.g., (*min A). Relative addressing in the grid is also provided. The form (pref-grid-relative!! A (!! -1) (!! 0)) returns a pvar that is equal to A shifted one position to the left. The reader is referred to Ref. 3 for additional information on *Lisp.

3. DATA-PARALLEL MODEL FOR SPATIAL REASONING

Fig. 1 is the proposed computational model for spatial reasoning in 2-d domains that contain objects that may belong to K possible classes. Such a model is appropriate for many image understanding and geographic information processing applications. It involves 1), representing spatial data (label maps, features, and hypotheses) by 2-d pvars and 2), viewing the processes of labeling, segmentation, feature extraction, and spatial inference as data-parallel transformations between pvars.

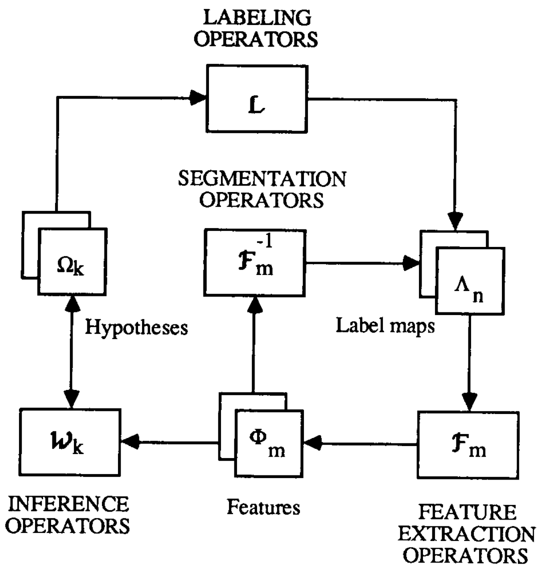

Fig. 1 Computational model

The organization and representation of spatial data using pvars is shown in Fig. 2. Label maps Λ are pvars that are used to explicitly delimit the spatial extent of regions (or edges) that may or may not be spatially connected. Labeling operators, $\mathcal{L}(\Omega) \rightarrow \Lambda$ assign unique numbers to sets of related processors (e.g., the largest cube address of the set of processors representing a connected region). Features Φ are pvars that are used in conjunction with label maps to store properties of regions (e.g., area, distance from) or simply by themselves to

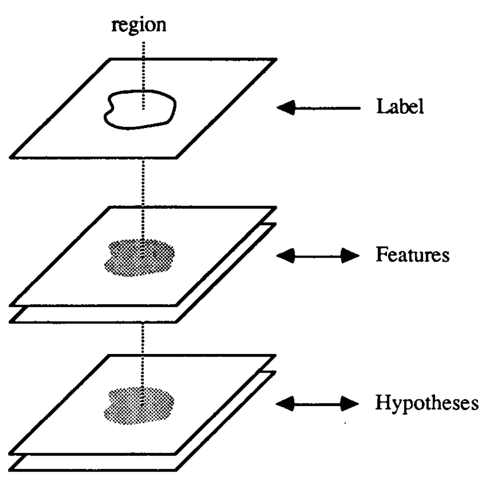

Fig. 2 Organization and representation of spatial data using parallel variables

store properties derived from other properties (e.g., the local mean of an image). In pvars that describe region properties, the value for each region is replicated in all the processors that belong to the region. Feature extraction operators $\mathcal{F}(\Lambda, \Phi') \rightarrow \Phi$ compute features from label maps and/or other features. Conversely, segmentation operators $\mathcal{F}^{-1}(\Phi) \rightarrow \Lambda$ can be viewed as inverse operations that compute label maps from features. Hypotheses Ω describe the degree to which regions belong to various classes. Inference operators $\mathcal{W}(\Phi) \rightarrow \Omega$ compute hypothesis maps from features.

Labeling and segmentation operators assign unique numbers in the range from one to the number of processors to each region in the image. All of the pixels in a region with label λ can then be easily accessed in unit time within forms such as

(*when (=!! Λ (!! λ)) &body).

Feature extraction and inference operators access regions by the label map. For example, the area ϕ of a region with label λ can be computed as

(*when (=!! Λ (!! λ)) (*set Φ (!! (*sum (!! 1))))

where the result is broadcast to each processor in the currently selected set. Inference operators can then compute hypotheses in parallel, e.g.,

(*set Ω (+!! Ω (abs!! (-!! (Φ (!! ϕ_0)))

in an amount of time proportional to the number of features.

To see how the above model applies to spatial reasoning consider the following examples. In a black-and-white image interpretation application, the input Φ is segmented into homogeneous connected regions by some type of region grower, $\mathcal{F}^{-1}(\Phi) \rightarrow \Lambda$. Various features related to the average brightness, texture, size, and shape of the regions and the spatial relationships between regions are computed from the label map and image, $\mathcal{F}_m(\Lambda, \Phi) \rightarrow \Phi_m$. These features are then evaluated against a set of constraints in order to accumulate evidence for candidate object categories such as buildings, roads, etc., $\mathcal{W}_k(\Phi_m) \rightarrow \Omega_k$.

In a geographic information system application, the input might be a database that provides certain kinds of information (surface material type, soil drainage characteristics, slope, etc.). The objective is to infer other kinds of spatial information, e.g., likely locations for a nuclear waste site based on constraints such as soil drainage characteristics, distance from populated areas or bodies of water, etc. The inputs, represented by a set of spatial occupancy arrays $\{\Omega_k\}$ are labeled $\mathcal{L}(\Omega_k) \rightarrow \Lambda_k$ and are used to compute properties such as the area, compactness, containment, and distance between regions, $\mathcal{F}_m(\Lambda_k) \rightarrow \Phi_{mk}$. A measure of the suitability of various areas $\mathcal{W}_{k'}(\Phi_{mk'}) \rightarrow \Omega_{k'}$ can be used to determine the best places (if any) to put the nuclear waste site. This second type of application is pursued further in Section 5.

4. DATA-PARALLEL OPERATORS

This section discusses some basic operators that have been implemented to date for the purpose of developing the ideas introduced in this paper. Additional operators are currently under development and improved algorithms, e.g., based on scanning (Refs. 4 and 5), will be added in the future.

CH2649-2/89/0000/0324$01.00 © 1988 IEEE

A connected components labeler based on the "brush fire" algorithm was implemented. Initially, each processor in the output pvar Λ is assigned its cube address, i.e., a number between one and the number of processors, N^2. Then, for all processors whose input pvar $\Omega = \{\omega(x,y)\}$ is non-zero, if $\omega(x,y) = \omega(x+u,y+v)$ where (x+u, y+v) are the addresses of the 4- or 8-nearest neighbors of (x, y), the output is updated as $\lambda(x,y) = \max \{ \lambda(x,y), \lambda(x+u,y+v)\}$. The process is repeated until $\Lambda^{t+1} = \Lambda^t$. The run time is proportional to the size of the largest connected region, which for small connected regions or large highly irregular regions is comparable to the more complex scan-based algorithms described in Refs. 4 and 5. For large regularly shaped regions, significant improvement can be obtained using scanning to propagate the maximum label up, down, left, right, and along diagonal connected segments.

Spatial operators include those that compute properties of individual regions (unary operators) and those that compute information about relationships between two or more regions (n-ary operators). Unary operators that compute geometrical properties such as the area, perimeter, and centroid of connected regions have been implemented using a counting approach. As an example, the area is computed by stepping through each unique label and adding up the number of processors in the currently selected set as described earlier.The complexity is thus of the order of the number of regions. Relational operations such as the minimum distance between two sets of connected regions are performed by computing the distance from any point in one set to all image pixels. The method involves propagating the label with the minimum distance and either the minimum distance or the address of the nearest processor. The minimum distance to each connected region in the other set is obtained by stepping through all labels and executing a *min over the minimum distances within the currently selected set. The complexity is of the order of the number of regions and the size of the largest region.

The set of feature pvars can be viewed as an image of feature vectors. This motivates an inference strategy based on a decision theoretic pattern classification approach. The inference operator implemented computes a similarity measure between a feature value or constraint and a feature pvar, and accumulates the similarity measure across all features. Constraints have the form ($\Phi \lozenge \phi_0$ w) where Φ is a feature pvar, $\lozenge$ is a parallel version of the standard Common Lisp predicates, ϕ_0 is a number, and w is a weighting factor. A constraint returns a pvar that contains zeros in those processors that satisfy the predicate and $w \mid \phi(x,y) - \phi_0 \mid$ in the others. The resultant pvar can be added to the results from other constraints to produce a score for the k^{th} class Ω_k. This is accomplished in an amount of time that is proportional to the number of features or constraints. For K classes, the process is repeated for each set of feature prototypes or constraints. The $\{\Omega_k\}$ can then be used as the basis for assigning the "best" class, in some sense, to each region.

5. CASE STUDY: GEOGRAPHIC INFORMATION SYSTEM

An example illustrating the application of our model to geographic information processing is shown in Fig. 3. The objective is to find regions that satisfy certain terrain constraints. The area of interest (a) is 512x512 pixels in size and contains the following categories: water, wetlands, coniferous and deciduous trees, bare soil, grass, agriculture, main, and secondary roads. The CM is configured as a 512x512 grid with a virtual/physical processor ratio of 32:1 thus providing up to 2048 bits per processor. First, coniferous and deciduous tree categories are merged and intersected with regions that are not main roads (b) and a label map computed. Information about tree regions such as the area, compactness, and distance between groups of trees can then be computed (c). Information about trees relative to other categories (e.g., distance from, containment, intersection, adjacency, etc.) is determined by marking those categories in working memory (d), computing a label map to uniquely identify each connected region, and applying the appropriate spatial operator, e.g., (e) is the minimum distance to main roads.

The result in (f) shows the five best forested areas given the constraints:

(area > 10000 0.5)
(compactness > 0.05 0.25)
(distance-from-water-or-wetlands > 5 0.75)
(distance-from-main-roads < 1 1.0)
(distance-from-secondary-roads < 1 0.5)

The result in (f) was obtained by ranking scores and selecting the top five areas (i.e., the five "closest" areas with respect to the decision region defined by the constraints).

6. SUMMARY

Massively parallel architectures motivate new approaches to old problems. While parallel processing solutions are almost always faster (they'd better be), in some cases they may even be simpler than those originally developed on serial machines. A homogeneous data-parallel model for 2-d spatial inference has been described that represents spatial information in a uniform manner by parallel variables organized in a 2-d grid. The model is simpler since it relies on a single representation as opposed to the heterogeneous (iconic and symbolic) representations used in more conventional systems. An initial application of the model to a geographic information processing problem was presented.

Future efforts will address other spatial reasoning tasks such as image interpretation and will involve developing additional data-parallel operators. We also plan to investigate the problem of handling images whose size exceeds the maximum number of virtual processors available in a given system.

REFERENCES

(1) W.D.Hillis, The Connection Machine, MIT Press, Cambridge, MA, 1985.

(2) L.W.Tucker and G.G.Robertson, "Architecture and applications of the Connection Machine," Computer, Vol. 21, No. 8, August, 1988.

(3) *Lisp Reference Manual (Version 4.0), Thinking Machines Corporation, Cambridge MA, 1987.

(4) J.Little, G.E.Blelloch, and T.Cass, "Parallel Algorithms for computer vision on the Connection Machine," Proc. International Conference on Computer Vision, pp 587-591, 1987

(5) G.E.Blelloch, "Scans as primitive parallel operations," Proc. International Conference on Parallel Processing, pp 355-362, August, 1986

Fig. 3a Thematic map showing trees, water and wetlands, roads, and open areas (bare soil, grass, and agriculture)

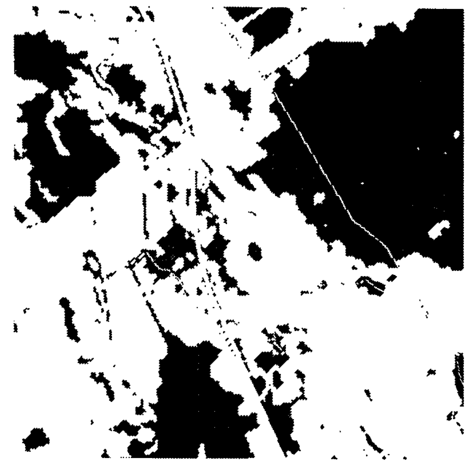

Fig. 3b Forested regions partitioned by main roads

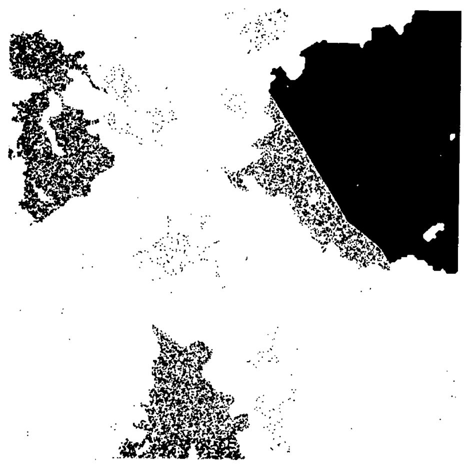

Fig. 3c Halftone rendition of the area of forested regions

Fig. 3d Main roads

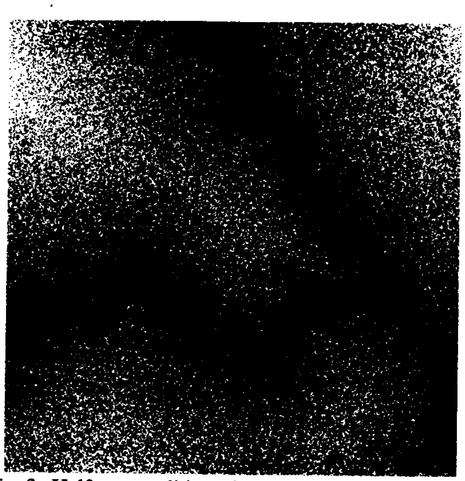

Fig. 3e Halftone rendition of the distance from main roads

Fig. 3f Five best forested regions for given constraints