Computation and Parallel Implementation for Early Vision

J. Anthony Gualtieri

NASA Goddard Space Flight Center

The problem of early vision is to transform one or more retinal illuminance images—pixel arrays—to image representations built out of primitive visual features such as edges, regions, disparities, clusters, ... These transformed representations from the input to later vision stages that perform higher level vision tasks including matching and recognition. We have developed algorithms for: 1) edge finding in the scale space formulation; 2) correlation methods for computing matches between pairs of images; and 3) clustering of data by neural networks. These algorithms are formulated for parallel implementation of SIMD machines, such as the MPP (Massively Parallel Processor), a 128 × 128 array processor with 1024 bits of local memory per processor. For some cases we can show speedups of three orders of magnitude over serial implementations.

(1) Edge Detection in Scale Space [M. Manohar and J.A. Gualtieri, in preparation]

Edge focusing [Bergholm, PAMI-9, 726-741 (1987)] generalizes standard edge detection approaches by performing edge detection at a series of scales and tracks the edges from coarse to fine scales. We begin with the Canny edge detector [J. Canny, PAMI-8, 678-698 (1986)] which convolves the image, I, with the gradient of a Gaussian of size sigma to smooth the image and finds the direction and magnitude of the gradient of the image at each pixel. Non-maximum suppression then detects the edge pixels. We write E(I; sigma) to denote the binary image of pixels so found [an edge pixel is assigned the value 1 and all others are 0]. To this initial edge image we perform a "dilation," which generates a binary mask of pixels, D (E(I, sigma), t), that are within distance t of the already found edge pixels. We again apply the Canny edge detector, but now with a smaller sigma to obtain edges resolved to the smaller sigma, and only to those pixels in the mask D. Because Bergholm has proved that edges move at most 1 pixel if sigma changes by 0.5, no edges are lost. By repeating the dilation followed by edge detection until we reach a sigma of size 1 we obtain good edge localization. All computation is inherently local and it is straightforward to map the algorithm on the MPP.

(2) Correlation Methods for Computing Matching in Image Pairs [J.P. Strong, H.K. Ramapriyan, Second Symposium on Super Computers, Santa Barbara (1987)]

A generalization of the stereo problem is to compute local rotations and displacements for a pair of images—called the reference image and test image—that matches sub-regions in the image pairs. An application of this method is to obtain a vector flow field describing the motion of ice floes (the sub-regions) imaged by synthetic aperture radar from a spacecraft at different times. The algorithm contains an outer loop that steps through one-half-degree rotations, theta, of the test image relative to the reference image. For each rotation, the algorithm computes a global cross correlation of the reference image relative to the rotated test image as a function of translation in the horizontal and vertical directions of the test image. The cross correlation is then thresholded to select a set of displacements corresponding to the largest global matches in the image pairs. Next, for each such displacement the test image is shifted by that displacement to obtain a rough match of sub-regions in the image pairs. To find the actual

sub-regions matched, a local cross correlation between a pair of small windows, say 11 × 11 (in a 512 × 512 image), in the reference and test images is computed for each pixel in the reference image over a search area of a few pixels in the horizontal and vertical directions. Marking the pixels at which this "local correlation" exceeds a threshold generates a mask defining the matching sub-regions in the reference and test images and further gives for each pixel in the sub-region in the reference image the total displacement (one part from the rotation plus global cross-correlation displacement and a smaller displacement from the local cross correlation) the matching pixel in the sub-region in the test image. While this algorithm is computationally intensive, it has been successfully mapped to the MPP and runs in times of the order of a few minutes.

(3) Clustering of Point Sets by Neural Networks [B. Kamgar-Parsi, J.A. Gualtieri, J.E. Devaney, and B. Kamgar-Parsi, Proc. of the Second Symposium on the Frontiers of Massively Parallel Computing, George Mason Univ. (1988)]

An important problem for early vision is to be able to partition N visual features, such as points in a two-dimensional space, into K clusters—in a way that those in a given cluster are more similar to each other than to the rest of the clusters. As there are approximately K**N/K! ways of partitioning the points among the K clusters, finding the best solution is beyond exhaustive search when N is large (say 128). This problem can be formulated as an optimization problem for which very good, but not necessarily optimal solutions can be found using a neural network. We have constructed a cost function to be minimized that is composed of a "syntax" term that enforces the constraint that each point must belong to only one cluster, and a "goodness of fit" term that measures the quality of the solution. Though the problem involves a discrete optimization, by embedding it in the continuous space of an analog network we are able to perform a downhill search on the cost function that is more purposeful and effective than a search in a discrete space. Solutions are generated by starting the network from many randomly selected initial states and then taking the best solution from the ensemble of solutions found. The network is simulated on the MPP where we use the massive parallelism not only in solving the differential equations that govern the evolution of the network, but also in starting the network from many initial states at once thus obtaining many solutions in one run. We obtain speedups of two to three orders of magnitude over serial implementations and further we obtain better quality solutions than conventional techniques such as K-means clustering.

We see the neural network approach as being important for early vision in that 1) the methods of "programming" neural networks described here can be generalized to other problems such as determining cluster shape, fitting more general features such as lines and parametric curves to visual data, and extending these results to higher dimensional spaces, and, 2) with the advent in the next few years of Analog VLSI devices, these algorithms can be straightforwardly mapped to silicon with potential speedups of up "nine orders of magnitude" over conventional serial implementations.