The goal of this project was to develop fast algorithms for performing the computations required by the various early vision modules that we have been developing. The early vision modules require intensive computation and are a major problem in system development since they consume a distractingly large amount of real time during the development process. The goal of the work therefore was higher computational efficiency, to be accomplished both by improving the basic efficiency of the algorithms as well as multiprocessing, so that the development of larger systems that use the early vision modules as components would not be adversely affected by slow response times of the modules.
Final Report to the Air Force Office of Scientific Research
for Grant AFOSR-88-0220

SOFTWARE MODULES FOR STEREO, TEXTURE AND PERCEPTUAL GROUPING IN EARLY VISION

Narendra Ahuja
University of Illinois
Coordinated Science Laboratory
1101 W. Springfield
Urbana, IL 61801

The goal of this project was to develop fast algorithms for performing the computations required by the various early vision modules that we have been developing. The early vision modules require intensive computation and are a major problem in system development since they consume a distractingly large amount of real time during the development process. The goal of the work therefore was higher computational efficiency, to be accomplished both by improving the basic efficiency of the algorithms as well as multiprocessing, so that the development of larger systems that use the early vision modules as components would not be adversely affected by slow response times of the modules.

We have achieved these objectives for several algorithms used in surface extraction from stereo images, texture analysis and shape from texture, and perceptual grouping. These algorithms include: two-dimensional Fast Fourier Transform, edge detection, feature matching, histogram computation, and surface fitting. We have developed multiprocessor algorithms for these tasks using dynamic scheduling and load balancing. As a result, tasks are assigned to processors as soon the processors become available. The use of dynamic load balancing leads to an improvement in performance by a factor of larger than two over the case in which uniform partitioning is used. The parallelization of the computation is accomplished by dividing the image into parts with each part processed separately and their results combined after the individual computations have been completed. We have achieved almost a linear speed up in the computation time with increase in the number of processors. The communication overhead is relatively small and increases very slowly as the number processors increases.

In addition to the above modules, we have also developed parallel, high speed modules used by algorithms that estimate scene structure and three-dimensional object motion. All modules we have developed have been tested on an Intel iPSC/2 hypercube multiprocessor.