A Parallel Environment for Structural Analysis of Range Imagery*

A. Pérez¹, M. A. Abidi¹, and R. C. Gonzalez¹,²

¹Department of Electrical and Computer Engineering
University of Tennessee
Knoxville, TN 37996-2100

²Percepts Corporation
Pellissippi Center
Knoxville, TN 37932

Abstract

Rigid objects can be modeled by quadric or planar polyhedra. This representation allows low-level data driven algorithms to determine the identity and pose of an object. Before an object can be matched with an internal model, however, its image must be first segmented into basic components, such as smooth patches. This paper describes a low-level parallel range data analysis system that uses input range data to characterize the underlying scene in terms of the visible surfaces, using simple, invariant features to generate a view independent description of the scene. The central components of the system are a parallel segmenter and a parallel surface fitter. The segmentation algorithms are based on the estimation of surface curvatures. Surface fitting allows for pose estimation and recognition of objects in range scenes. Mathematically, surface fitting can be formulated as an overdetermined linear system which can be solved in the least-squares sense. Because of numerical stability and ease of implementation, the QR-factorization using the Givens transformation is suited for the parallel solution of overdetermined systems. We discuss the implementation of the range analysis system on a distributed-memory hypercube parallel computer.

I Introduction

In this paper we discuss the development of a parallel environment to determine the identity and pose of objects, depicted as regions in a range image. Rigid objects can be modeled by quadric or planar polyhedra. This representation allows low-level data driven algorithms to determine the identity and pose of an object. Before objects can be matched with internal models, a range image must be first segmented into smooth patches and quadric surfaces fit to each of them, such that some error criterion is minimized, e.g., least squares fit. This low-level stage is computationally intensive, and in general dominates the time complexity of a single object search. Therefore, it is essential that these tasks be performed as efficiently as possible.

One approach to improve the performance of range analysis systems is the use of parallel processing, since its availability is rapidly increasing, making it a cost-effective solution [1]. Parallel processor have the potential of offering performance several orders of magnitude higher than sequential machines. Hence, parallelism permits the efficient use of global operators and simultaneous testing of different hypotheses. Thus, parallelization makes feasible new approaches that are conceptually very simple, but computational or memory intensive.

Parallelism may be exploited in two basic ways, either by having the different processing elements work on the same problem using different data, or by executing different tasks on the same data set. The first approach will work best on a fine-grain multiprocessor, while the second one requires a coarse system with powerful processing elements. Efficiently loading concurrent programs onto multiprocessor architectures is a weighted graph partitioning problem. The corresponding optimization problem is NP. Near-optimal solutions can be found by using heuristic algorithms such as iterative improvement and simulated annealing [2,3]. Hence, the use of parallelism raises the question of how to map an algorithm into a particular parallel architecture. Analyzing the performance of multiprocessor systems is a complex task, since numerous factors jointly determine system performance.

One way to categorize parallel machines is based on whether they have global or local memory. Interprocessor communication in local memory machines is handled by a communication network using message passing protocols, whereas in global or shared memory machines all the processors have access to the same common memory. In a shared memory system the elements are tightly coupled. In a message passing system, each node has its own local memory and communication of results takes place through internode message passing. These systems are loosely coupled and the effectiveness of their intercommunication

*This work was performed under contract ORA-EO1291-09 with the Advanced Automation Technology Group, Martin Marietta Aerospace, Denver, Colorado, and contract DE-FG02-86NE37968 with the Department of Energy.

88 / SPIE Vol. 938 Digital and Optical Shape Representation and Pattern Recognition (1988)
is mostly dependent, from a hardware point of view, on the topology of the interconnection network. One of the more popular message-passing multiprocessor architectures currently available is the hypercube [4]. In this paper, we discuss the application of a hypercube-connected message-passing computer to parallel segmentation, edge detection, and least-square surface fitting of range data.

The first step in range data segmentation consists of the evaluation of curvature invariants, e.g., the Gaussian and Mean curvatures. A range image can be segmented in three basic steps. First a smoothing filter is applied to eliminate sparse high amplitude noise. Second, partial derivatives and curvature invariants are computed. Finally, least-square fitting is used to determine the scene structure and generate recognition primitives. Once a rough segmentation is made, the surface fitting of different patches allows for pose estimation. This step yields a rough segmentation of the scene. In order to determine the exact pose of objects in the scene, this step is followed by an accurate surface fitting of the patches generated in the previous step. The surface fitting is set as an overdetermined linear system over the polynomial coefficients of the quadric form describing the surface patches, and is solved using the QR decomposition. To implement the QR decomposition in a parallel setting we make use of a variation of the Givens transform, known as the square-root-free Givens transformation. This method has been implemented in a shared-memory parallel computer and in a message-passing parallel computer [5]. In section II, we review some characteristics of range data, as well as some properties of quadrics since they are used as primitives for segmentation. Then, in section III we discuss the problem of range data segmentation in the context of parallel processing.

The surface fitting problem is solved in parallel using two QR-factorization algorithms. To implement the QR decomposition in a parallel setting, we make use of a variation of the Givens transform [6], known as the square-root-free Givens transformation. A parallel scheme for Givens rotations was first proposed by Sameh and Kuck [7]. They considered a Single Instruction Multiple Instruction (SIMD) computer and their algorithm is as stable as its sequential counterpart. Recently, several studies have appeared regarding modifications to the Sameh and Kuck basic algorithm [8,9,10]. Cosnard and Robert [8] make a complexity analysis of the parallel QR-factorization, while Modi and Clarke [10] consider explicitly the case of overdetermined systems. In section IV the parallel solution of the surface fitting problem is discussed in detail. The experimental results are given in section V.

II Characteristics of Range Data

In order for a machine to perform in a 3D world it should be able to determine the pose and identity of objects in the field of view. Range data provides a sampling of the spatial location of points within the field of view. A range sensor measures the distance from a reference point to objects in the field of operation of the sensor. There are two basic approaches to range sensing: triangulation, either passive(stereo) or active (structured lighting), and time-of-flight methods [11]. Ideally, scenes should be described using representations that are pose independent, i.e., in order to efficiently process range data it is necessary to establish object representations that are independent of the relative pose of the object and the sensor. Some of the most commonly used low-level processing tools are smoothing filters (linear and nonlinear) and curvature evaluation. Figure 1 shows a block of of the low-level end of a general purpose range data analyzer. The objective of this system is to generate descriptions that are independent of object position, starting from raw range data. Because of the low resolution of actual range sensors, it is essential to start any processing with a filtering processing to reduce the effect of quantization and sensor noise. This is accomplished with median and gaussian filters. The aim of the curvature analysis is to determine regions of concavity and convexity, and the location of step and roof edges. Once a rough segmentation is made, surface fitting of the different patches allows for pose estimation. We look at some properties of general polynomial surfaces of second order, known as quadrics, in order to generate primitives for object identification and pose estimation and model matching. Since quadrics can always be expressed in terms of their principal axes, they are a natural way for representing the pose of an object. The surface fitting is set as an overdetermined linear system over the polynomial coefficients, and is solved using the QR decomposition, where Q is orthogonal, and R upper-triangular. The QR-factorization provides algorithms for parallel solution of least-square problems that are numerically stable and efficient.

The general equation of a quadric is

$$a_{mn}z^n + b_{mn}z^m + c = 0,$$

where $a_{mn}$ is a double symmetric tensor, $b_{mn}$ is a vector and $c$ is an invariant, all having constant values. A classification of quadric types may be done as follows [5,12]:

1. a central quadric;
2. a paraboloid
3. a proper cone;
4. a cylinder (elliptic, hyperbolic, or parabolic);
5. a pair of planes;

Given a number of points belonging to a patch of a range image, the fitting of a quadric to these points allows us to use the previous classification for labeling and recognition. Furthermore, the pose of a given object can be estimated from the principal axes of the fitted quadric [5].

III Parallel Segmentation

The segmentation and processing of range data requires the formulation of methodologies involving geometric and physical objects in a way that is independent of the underlying arbitrarily chosen coordinate system. A segmentation or labeling is a connected component analysis of an image. A segment is a maximal connected component. Particularly relevant to range analysis are the concepts of space invariants and surface curvatures since most range segmentation techniques make use of curvature invariants [13]. The estimation of these parameters, however, is generally noise-sensitive and difficult to handle at surface boundaries. Moreover, the objective of the curvature analysis is the location of step and roof edges, where these measurements are less reliable. Hence, the computation of curvature-related measurements must be carried out using operators that are scalable, so that the range image can be analyzed at different resolution levels, depending on the amount of relevant detail. In a general purpose system it is also necessary for the operator to be symmetric with respect to the z and y axes, so that it would be insensitive to object orientation. Furthermore, in order to facilitate its parallel implementation, an ideal operator should not have any data dependencies. An operator that provides a parameter for scale–tuning and directional symmetry is the laplacian–of–a–gaussian. In the context of range images, the sign of the laplacian is associated with regions of concavity and convexity. Hence, it seems natural to look at operators of the form

\[
f(\nabla^2 G_\sigma)
\]

where \(G_\sigma\) is a gaussian distribution with a variance of \(\sigma^2\).
IV Parallelization of Surface Fitting

It is now widely accepted that the best available parallel algorithm to solve dense systems of linear equations is the QR-decomposition using square-root-free Givens transformations for reasons of stability and simplicity [8]. Givens transformations lend themselves to parallel implementations because they allow the selective introduction of zeroes in the matrix and the fact that only two columns of the matrix are used at each step [14,15]. The Givens method is particularly useful in parallel computing because pivoting, which can dominate parallel Gaussian elimination algorithms, is not required [16,9]. A parallel implementation of the Givens method must take advantage of the fact that a single rotation affects only two rows. Thus, an appropriate sequence must be found so that parallelism is fully exploited. A Givens sequence is any sequence of Givens rotations in which zeros once created are preserved. If the matrix has \( m \) rows, then a maximum of \( m/2 \) rotations can be performed at once.

In a recent paper [5], we introduced a new Givens transformation that minimizes communication in a distributed-memory parallel computer by using a ring structure that allows message passing only between neighboring processors. In the following, we present a new algorithm that carries out a Givens sequence on one of the most prominent architectures of parallel computer topologies, a distributed-memory MIMD hypercube parallel computer. In a distributed-memory system, communication delays are typically one order of magnitude higher than computational delays. Furthermore, messages are usually buffered. Hence, an efficient distributed-memory should maximize the computation/communication ratio and use a few large messages, rather than many small ones. In this algorithm, parallelism is fully exploited because of the two following major reasons:

1. Because of the size and symmetry of the implementation of the Givens transformation, load balancing among the processors is intrinsic (except for the 2 processors that are handling the top and bottom of the matrix).
2. Because of the ring structure, chosen here, exchanged messages travel only one link of the hypercube, making the communication very efficient.

A time-efficiency study of this implementations is given in section V.

In their general form, quadrics in three dimensions are uniquely defined by 10 independent parameters. Hence, the fitting of \( m \) points \((m > 10)\) results into an \( m \times 10 \) overdetermined system that needs to be solved in the \( L_p \) sense. On the other hand, \( \|Ax - b\|_2 \) is a continuously differentiable function of \( x \). The least-square (LS) problem

\[
\min_x \|Ax - b\|_2,
\]

where the vector \( Ax_{LS} \) is the orthogonal projection of \( b \) onto the range of \( A \), has the added attraction that it can be converted to the following equivalent problem

\[
\min_x \| (Q^TA)x - (Q^Tb) \|_2 \quad \quad Q^TQ = I_m,
\]

by premultiplying both \( A \) and \( b \) by an orthogonal matrix \( Q \) [17]. The basic idea is to use orthogonal transformations to compute the factorization \( A = QR \), where \( Q \) is orthogonal and \( R \) is upper-triangular. This is mathematically equivalent to applying the Gram-Schmidt method to the columns of \( A \) and taking advantage of the intrinsic optimization properties of orthogonal projections [18].

Givens rotations can be used to compute the QR-decomposition [6]. The idea is to construct a matrix \( M \in \mathbb{R}^{m \times n} \) such that \( MA = S \) is upper-triangular and \( MM^T = D = \text{diag}(d_1, \ldots, d_m) \).

\( x_{LS} \) is obtained by solving the nonsingular upper-triangular system \( S_1x = v_1 \), since

\[
\|Ax - b\|_2 = \|D^{-1/2}(MAx = Mb)\|_2,
\]

for any \( x \in \mathbb{R}^n \).

A detailed description of algorithms implementing this transformation in shared-memory and message-passing systems were introduced by Pérez et al. [5].

V Experimental Results

The distributed memory algorithm for the solution of overdetermined system of equations using Givens transforms introduced by Pérez et al. [5] was implemented and tested in an Intel IPSC hypercube parallel computer. The parallelism is best exploited in systems that are both large and square. As shown in Fig. 2, the speed-up is linear on the number of processor. There are, however, three distinct regions in the graph. In the low end of the graph, for cubes of dimension 1 and 2, the bookkeeping overhead associated with the synchronization of concurrent tasks is significant and limits somehow the performance of the
algorithm. In the mid section of the graph, depicting the performance of cubes of dimension 3, 4, and 5, the algorithm runs at its best for this size problem because the setup time and communication overhead are no longer significant with respect to the processing time. Finally, when considering the performance of a cube of dimension 6, we notice that the speed-up factor actually decreased. This is due to the fact that the communication cost is significantly higher than the computation cost, and the ratio of communication to computation has increased beyond the point of optimal operation. For a bigger size matrix this point will, of course, move to a higher dimension cube.

The segmentation algorithms are implemented in an NCUBE/4 parallel computer with 8 nodes. Figures 3 and 4 are an actual range and its fully registered reflectance image. Figure 5 shows the output of the $\nabla^4 G_x$ operator. This is a presegmentation of the given scene. This operation did not perform well because of the low dynamic range of the image around the objects of interest (about 5% of the total range). This caused the blurring of the roof edges. This problem is less severe in Fig. 6 and 7, where the dynamic range of the image is about 15%. This resulted in smoother edges, hence a better
Figure 4: Registered reflectance of scene shown in Fig. 3 (courtesy of R. E. Carlton, Odetics Co., Anaheim, CA).

Figure 5: Output of the $\nabla^4$ operator.
Figure 6: Range image of a scene generated by an ERIM sensor.

Figure 7: Output of the $\nabla^4$ operator.
presegmentation is achieved. In both cases, false contouring is highly noticeable, however, this can be removed during during the surface fitting phase.

VI Conclusions

In this paper, we expanded on the results presented in [5] where we showed that the solution of the surface fitting problem is a crucial step for range data analysis. We showed the interpretation of some timing results for the implementation on a MIMD distributed-memory parallel computer of a new Givens sequence for a QR-factorization. We also discussed the implementation of segmenting algorithms on a distributed memory parallel computer.

References


