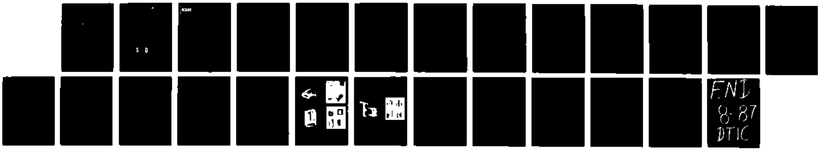
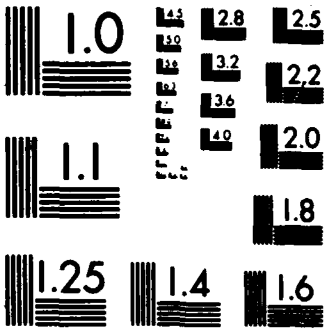
AD-A182 736 AUTOMATION RECOGNITION OF TRACKING OF OBJECTS(U) TEXAS 1/1
UNIV AT AUSTIN COMPUTER AND VISION RESEARCH CENTER
J K AGGARWAL 31 DEC 86 AFOSR-TR-87-0893
UNCLASSIFIED F49620-85-K-0007 F/G 17/5 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

②

AFOSR-TR- 87-0893

Final Scientific Research Report

AFOSR Contract
No. Contract F49620-85-K-0007

January 1, 1985 - December 31, 1986

Computer and Vision Research Center
The University of Texas at Austin
Austin, TX. 78712

Principal Investigator: Professor J. K. Aggarwal

87 7

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE

3. DISTRIBUTION/AVAILABILITY OF REPORT
Approved for public release, distribution unlimited

4. PERFORMING ORGANIZATION REPORT NUMBER(S)

5. MONITORING ORGANIZATION REPORT NUMBER(S)
AFOSR-TR- 87-0893

6a. NAME OF PERFORMING ORGANIZATION
Univ of Texas at Austin

6b. OFFICE SYMBOL *(If applicable)*

7a. NAME OF MONITORING ORGANIZATION
AFOSR/NE

6c. ADDRESS *(City, State and ZIP Code)*
Austin, TX 78712

7b. ADDRESS *(City, State and ZIP Code)*
Bldg 410
Bolling AFB, DC 20332-6448

8a. NAME OF FUNDING/SPONSORING ORGANIZATION
AFOSR

8b. OFFICE SYMBOL *(If applicable)*
NE

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
F49620-85-K-0007

8c. ADDRESS *(City, State and ZIP Code)*
Bldg 410
Bolling AFB, DC 20332-6448

10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.
61102F	2305	B3	

11. TITLE *(Include Security Classification)*
Automation Recognition of Tracking of objects

12. PERSONAL AUTHOR(S)
Professor Aggarwal

13a. TYPE OF REPORT
Final

13b. TIME COVERED
FROM 01 Jan 85 TO 31 Dec 85

14. DATE OF REPORT *(Yr., Mo., Day)*

15. PAGE COUNT
22

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB. GR.

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

The Computer and Vision Research Center conducts a broad program of research in computer vision, image processing, and architectures for image processing. During the period of this report, several projects were pursued including those on positioning and tracking of objects moving in space, parallel image processing, and 3-D representation and recognition.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT
UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL
Dr C Lee Giles

22b. TELEPHONE NUMBER *(Include Area Code)*
202-767-4931

22c. OFFICE SYMBOL
NE

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE.

Final Scientific Report

AFOSR Contract No. F49620-85-K-0007

Abstract

The Computer and Vision Research Center conducts a ~~borad~~ broad program of research in computer vision, image processing, and architectures for image processing. During the period of this report, several projects were pursued including those on positioning and tracking of objects moving in space, parallel image processing, and 3-D representation and recognition. The results on these six completed projects are briefly presented in this report:

~~A.~~ Representation of Objects from Range Maps;

~~B.~~ Hierarchical Data Structures for the Computation of 3-D Information from Multiple Views;

~~C.~~ Structure and Motion Computation from Point or Line Correspondences in Images;

~~D.~~ Surface Reconstruction and Representation of 3-D Scenes;

E. Parallel 2-D Convolution on a Mesh Connected Array Processor; and

F. A Parallel Processing Technique Exploiting Image Parallelism via the Hypercube .

Complete results have been presented at national/international conferences and published in refereed Journals.

A. Representation of Objects from Range Maps

Range (depth) data provides an important source of 3-D information. Range data implicitly contains information about the shape of the surface of objects because the coordinates of points on the surface of these objects can be easily recovered from them. Range data may be derived from intensity images or through direct measurement sensors.

The goal of this research is to develop algorithms for representation of objects from range data. The goal of building a representation is achieved in two stages namely, building an object description from range data acquired from a single view and integrating multiple views to construct models of objects.

In section 2 we will briefly discuss the representation of visible object surfaces from range data and section 3 will contain the integration of data or descriptions obtained from multiple views of an object to construct its model.

1. Representation of Visible 3-D Object Surfaces

There have been several studies on building object descriptions from range data. Most of the existing approaches make explicit assumptions about the underlying surfaces in the scene [1]-[7]. For example some of the early approaches assume the scene to be composed of planar objects [1]-[3]. Although techniques do exist for describing scenes composed of both planar and curved objects, they make explicit distinction in the reconstruction process of the aforementioned type of objects [5],[6]. Hence there is need for an algorithm that can treat planar and curved objects in a homogeneous fashion. In this research, we have developed an algorithm for building object representation based on regions that are a collection of surface patches homogeneous in certain intrinsic surface properties [8]. The algorithm is not restricted to polyhedral objects nor is it committed to particular type of approximating surface. The algorithm was tested on synthetic as well as real data with reasonable success. A brief discussion of the algorithm is given below (for an elaborate discussion the reader is refered to [8]).

An object is characterized in terms of its jump boundaries, internal edges (surface creases) and surface primitives. So the object representation problem deals with explicit identification and integration of these quantities. The input to the object description algorithm consists of a collection of 3-D points which need not be represented with one coordinate as a function of the other two. First, the two-dimensional arrays containing 3-D data are divided into overlapping windows of size L. Each (L x L) window of data is tested for occurence of a jump boundary where one surface occludes another. Therefore two adjacent points at a jump boundary will be separated by a significant distance. Thus jump boundaries are detected by looking for a significant range discontinuity between adjacent data points. If no jump boundary is present then a tension spline based surface is fitted to the data. This surface is refered to as a patch. The surface fitting algorithm is general, efficient (linear time [9]) and uses existing public domain numerical software. Following the surface fitting process, principal curvatures are computed and surface points are classified into one of the following types : elliptic, hyperbolic, parabolic, umbilic, and planar umbilic. Regions on the surface of an object are grown on the basis of this classification. The object description so obtained is view point independent and hence will prove to be useful in the context of object recognition. Our representation algorithm has been tested on real data obtained from a laser scanner [10] and the results obtained were most often in agreement with theoretical predictions.

2. Multiple View Integration and Model Construction

Constructing the 3-D model of an object involves integrating data or descriptions of an object obtained from multiple views and representing this integrated data or descriptions in a coherent manner. In this research we present a new technique for automatic construction of 3-D models of arbitrarily shaped objects, given range and intensity data acquired from multiple views. Our technique for integrating the information from multiple views does not require correspondence relationship between views to be determined, unlike most other approaches [11]-[15]. A brief description of the multiple view integration and model construction process is described here, for a detailed description the reader is refered to [16].

The object, for which the model is to be constructed, is assumed to rest on a plane (base plane). A pattern consisting of a single straight line is drawn on the base plane. The interframe transformation required to register any two views in a common reference coordinate system is derived by observing the orientation of the base plane pattern in the intensity images from multiple views. Once the interframe transformation for every view has been computed, the range data from different views are expressed in a common reference coordinate system and merged. A region description of the object model is obtained using the algorithm presented in Vemuri et al. [8]. Regions in this description are formed by a collection of surface patches that are homogeneous in intrinsic surface properties. Such a description is viewpoint independent, a property crucial for modeling. The present technique for 3-D model construction also demonstrates a way to combine multiple sources of information namely, range and intensity. information namely, range and intensity.

As a general technique for computing visible surface structure from range data, extracting viewer independent surface properties that are useful in recognizing objects, and integrating multiple views to construct models this research will be relevant in advancing the state of the art in robotics, graphics etc.

References

[1] P.M. Will and K.S. Pennington, "Grid Coding: A preprocessing technique for robot and machine vision," Artificial Intelligence, Vol.2, 1971, pp. 319-329.

[2] Y. Shirai & M. Suwa, " Recognition of Polyhedrons with a range finder ", Proc. of the 2nd Intl. Joint Conf. on Artificial Intelligence, Sept 1971, pp. 80-87.

[3] R.O. Duda, D. Nitzan & P. Barrett, " Use of range and reflectance data to find planar surfaces regions ", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol.1, No.3, July 1979, pp. 259-271.

[4] G.J. Agin & T.O. Binford, " Computer description of curved objects ", Proc. of the 3rd Intl. Joint Conf. on Artificial Intelligence, Aug. 1973, pp. 629-640.

[5] M. Oshima & Y. Shirai, " A scene description method using 3-D information ", Pattern Recognition, Vol.11, 1979, pp. 9-17.

[6] O.D. Faugeras et.al, "Segmentation of planar and quadratic patches from range data", IEEE Proc. Computer Vision and Pattern Recognition, 1983

[7] W.E.L. Grimson, "An implementation of computational theory for visual surface interpolation", Computer Vision, Graphics and Image Processing, 22, 1983, pp. 39-69.

[8] B. C. Vemuri, A. Mitiche and J. K. Aggarwal, "Curvature-based representation of objects from Range data," Image and Vision Computing, Vol. 4, Number 2, May 1986, pp. 107-114.

[9] A.K. Cline, "Surface smoothing by splines under tension," CNA-170, Univ. of Texas at Austin, 1981.

[10] "White Scanner 100a Users Manual," Technical Arts Corporation, Seattle, Washington, 1984.

[11] M. Potmesil, "Generating Models of 3-D Objects by Matching 3-D Surface Segments," Proceedings of the 8th Intl. Joint Conf. on Artificail Intelligence, Aug. 1983, Karlsruhe, West Germany, pp. 1089-1093.

[12] F.P. Ferrie and M.D. Levine, "Piecing Together the 3-D Shape of Moving Objects: An Overview," IEEE Proceedings Compu. Vision and Pattern Recognition, 1985, San Francisco, California, pp. 574-584.

[13] B. Bhanu, "Representation and Shape Matching of 3-D Objects," IEEE Transactions on Pattern Analysis and Machine Intelligence, May 1984, Vol. PAMI-6, No. 3, pp. 340-351.

[14] B.A. Boyter and J.K. Aggarwal, " Recognition with Range and Intensity Data," Proceedings of the Workshop on Computer Vision : Representation and Control, 1984, Annapolis, Maryland, pp. 112-117.

[15] B. Bhanu, T.C. Henderson and S. Thomas, "3-D Model Building Using CAGD Techniques," IEEE Proc. Compu. Vision and Pattern Recognition, 1985, San Francisco, California, pp. 234-239.

[16] B. C. Vemuri and J. K. Aggarwal, "3-D Model construction from multiple views using range and intensity data," IEEE Proc. on Computer Vision and Pattern Recognition, June 1986, pp. 435-437.

B. Hierarchical Data Structures for the Computation of 3-D Information from Multiple Views

Data acquisition and object representation are fundamental and crucial to research in computer vision. The 3-D data of an object may be acquired using an active sensing or a passive sensing approach. The active sensing approach includes direct and active range finding techniques (each of which involves a controlled energy beam and reflected energy detection), and contrived lighting techniques. The passive sensing approach includes monocular image-based range finding techniques (such as shape from shading, shape from texture, and shape from occluding contour), and multiple-view reconstruction techniques (such as stereo disparity, volume intersection, and structure from motion). Even though 3-D Information may be obtained from many sources, each of these cues is valid only for a particular class of situations. For example, determining shape from shading requires accurate modeling of the incident illumination and surface characteristics (e.g. reflectance) - which is difficult to achieve for most natural scenes. A detailed discussion of each of these technique can be found in [1].

Once the 3-D data of an object are acquired, they should be arranged in a particular format for ease of manipulation and analysis. A variety schemes have been proposed to describe 3-D objects [2]. These schemes may be broadly categorized as volumetric descriptions or surface descriptions. Advantages and disadvantages of each category are summarized in [3]. Most representations suffer from severe memory and processing requirements with increasing input sequence size. An octree representation scheme that uses efficient tree traversal algorithms overcomes these severe drawbacks.

Quadtrees and octrees are hierarchical data structures for the representations of 2-D silhouettes and 3-D objects, respectively. Both quadtrees and octrees are efficient data structures in terms of storage requirement, and are capable of retaining the detailed boundary information as well. These distinguished properties make them a natural choice for computing 3-D information from multiple silhouettes.

The silhouette of an object usually conveys insufficient 3-D information about the object. When the silhouette of an object is extended into 3-D space along the corresponding viewing direction to form a cylinder, one may only know that the object is bounded by the cylinder. This problem is resolved by intersecting the bounding cylinders from different views. In the past, octrees of 3-D Objects have been generated from multiple silhouettes using a technique known as volume intersection [4,5]. Further, a multi-level boundary search (MLBS) algorithm has been used to encode the surface information was called a volume/surface (VS) octree.

Although the MLBS algorithm is efficient in computing surface information from octrees, a further improvement in efficiency can be achieved if the octree generation algorithm computes the VS octrees of 3-D objects directly from the multiple silhouettes of the object. In this paper, we present a unified approach to compute the VS octree from occluding contours and silhouettes of multiple views of a 3-D object. The surface information is computed directly from the occluding contours of multiple views. The key idea is to encode contour information into the quadtrees of the associated silhouettes. The VS octree can then be generated directly from the contours and silhouettes of the multiple views. In order to obtain more accurate contour information, each occluding contour is fitted with a tension-spline [6]. This curve fitting process allows better approximations of the contour normals, and hence provides more accurate surface information. A modified octree refinement algorithm is also developed which updates both volumetric and surface information at the same time, as additional information is available. Experimental results show that the proposed approach for generating volume/surface octrees is more efficient than the one previously developed which employs the MLBS algorithm. In addition, the new algorithm provides a more detailed and accurate description of the object surface since the surface normals are not quantized to a set of twenty-six orientations (corresponding to the twenty-six neighbors of a cube) as in the previous approach.

It is known that in some cases a finite number of views is not enough to reconstruct the exact 3-D structure of an object. An object description scheme should be conducive to refinement as additional information is acquired. The octree structure allows subsequent refinement of the description of 3-D objects, which can be accomplished by an octree refinement algorithm as described in [5]. In the previous method, after the octree has been refined, updating the surface information in the octree requires that the MLBS be performed on the refined octree. The subsequent updating of a volume/surface octree thus requires alternating applications of the octree refinement algorithm and the MLBS algorithm. In this paper, the refinement algorithm is modified so that the updating of both volumetric and surface information is achieved in one pass. As a consequence, a significant speed-up in processing is obtained using this one-pass algorithm.

References

[1] R. A Jarvis, "A Perspective on Range Finding Techniques for Computer Vision," *IEEE Trans. on PAMI*, Vol. PAMI-5, No. 2, March 1983.

[2] J. K. Aggarwal, L. S. Davis, W. N. Martin, and J. W. Roach, "Survey: Representation Methods for Three-Dimensional Objects," in *Progress in Pattern Recognition*, Vol. 1, L. N. Kanal and A. Rosenfeld, Eds., North-Holland, New York, 1981, pp. 377-391.

[3] A. A. G. Requicha, "Representations for Rigid Solids: Theory, Methods, and Systems," *Computing Surveys*, 12, No. 4, 1980, pp. 437-464.

[4] C. H. Chien and J. K. Aggarwal, "Volume/Surface Octrees for the Representation of 3-D Objects," *Computer Vision, Graphics, and Image Processing*, 36, pp. 100-113, 1986.

[5] C. H. Chien and J. K. Aggarwal, "Identification of 3-D Objects from Multiple Silhouettes Using Quadtrees/Octrees," *Computer Vision, Graphics, and Image Processing*. 36, pp. 256-273, 1986.

[6] A. K. Cline, "Smoothing by Splines under Tension," CNA-168, Dept. of Computer Science, The Univ. of Texas at Austin, 1981.

c. STRUCTURE AND MOTION COMPUTATION FROM POINT OR LINE CORRESPONDENCES IN IMAGES

The world constantly evolves around us, events unfold and fold, objects appear and disappear, the scene we perceive continuously changes. Even stationary objects appear to have (relative) motion because of our own motion or the movement of the eyes. This observation confirms the belief that in the real world, a static pattern is a rarity -- continuous motion and change being the rule. The human eye and brain combination has an enormous capacity for efficiently and effectively processing and digesting this continuous flow of information. However, the development of this capability for computers has proven to be a difficult and a challenging task.

Several developments during the last decade have facilitated the computer analysis of time sequences of images. In particular, significant advances in the sensing, storing and processing technologies has facilitated the acquisition and storage of large amounts of data embedded in an image sequence. Also, the capacities of the human eye and the general principles of human eye functions have been and are being better understood through intensive efforts in psychological and physiological research. The evolution of VLSI technology has enabled cost effective implementation of special computer architectures dedicated to image processing and analysis. As a result, the problem of analyzing sequences of images and identifying the structure and motion of the imaged objects has attracted substantial attention from researchers.

The problem of computing structure and motion from images is important for both its theoretical challenge and its many practical applications. Theoretically, analyzing a sequence of images poses more problems than analyzing a single static image. Not only do separate pieces of information have to be extracted from each image frame, but they also have to be integrated and interpreted in a coherent manner. The analysis and interpretation of an image sequence have to account for the changing nature of the images between frames and still be able to build up a consistent and uniform interpretation. Although the task is difficult, a variety of applications including target tracking from video images, autonomous vehicle navigation, robot guidance, dynamic monitoring of production processes, and cloud tracking and weather forecasting have motivated and stimulated this research.

This article [1] provides an overview of the research on the problem of computing structure and motion from point [2] or line [3] correspondences in images. The emphasis is placed on the estimation of three-dimensional (3-D) surface structure and motion parameters from two-dimensional (2-D) projections. The issue of planar objects in 2-D motion is not addressed in this review. The paper by Aggarwal and Duda and the review by Martin and Aggarwal present the early work. In general, the recovery of 3-D structure and motion from images is difficult and complicated. Most of the approaches reported on 3-D structure and motion computation adopt the following steps: (1) compute observables in the images and (2) relate these observables to object structure and motion in space. Various observables have been considered: points, lines, optical flow, and range. These observables are usually extracted from visual images except in the case of range which may be sensed directly or computed from images. In this review, we focus on the work using points and lines as observables for computing structure and motion.

In principle, the observation of a number of points in two or more views can yield the position of these points in space and the relative displacement between the viewing systems. This line of reasoning using points as observables has been pursued by Roach and Aggarwal, Webb and Aggarwal, Nagel, Ullman, Tsai and Huang, Tsai, Huang and Zhu, Longuet-Higgins, and Mitiche, Seida and Aggarwal among many other researchers. The concensus is that the observation of five points in two views yields both structure and motion.

The use of line correspondences in the computation of structure and motion has been addressed by Yen and Huang and Aggarwal and Mitiche. Yen and Huang used seven line correspondences for solving structure and motion parameters and it was shown in that five lines in three views in general position can yield the orientation of the lines in space and the motion parameters. The use of line correspondences has the additional advantage over the point correspondences in that extraction of lines in images is less sensitive to noise than extraction of points.

Computation of structure and motion using optical flow generally involves estimating the perceived motion in the image plane and then computing the structure and motion from the projected point position and optical flow. Also, the availability of range data has greatly facilitated the computation of structure and motion since position and orientation information is directly available. Techniques which use optical flow and range data for structure and motion computation will be reviewed in a future paper.

The basic assumptions of the following analysis are that images have been properly segmented, the observables (points or lines) have been extracted from each image, and the correspondence of points or lines between images has been determined. As we shall see later, these assumptions are commonly made. These assumptions separate the structure and motion computation from many other peripheral processes such as scene segmentation and determination of the correspondence relationship. In the following, we review techniques based on points and lines. Finally, the importance and the impact of the fundamental assumption - point and line correspondences - are briefly discussed.

References

[1] J. K. Aggarwal and Y. F. Wang, "Structure and Motion Computation from Point or Line Correspondences in Images", in *Advances in Image Processing and Pattern Recognition*, edited by V. Cappellini and r. Marconi, Elsevier Science Publishers B. V., North-Holland, pp. 171-178, 1986.

[2] A. Mitiche, S. Seida, and J.K. Aggarwal, "Determining Position and Displacement in Space from Images," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, San Francisco, CA, pp. 504-509, June 1985.

[3] A. Mitiche, S. Seida, Y.F. Wang and J.K. Aggarwal, "Line-Based Computation of Structure and Motion Using Angular Invariance", submitted for publication in *Pattern Recognition*.

D. SURFACE RECONSTRUCTION AND REPRESENTATION OF 3-D SCENES

In image analysis and computer vision a considerable effort has been devoted to the development of representation schemes for both two-dimensional (2-D) and three-dimensional (3-D) objects. Various 2-D schemes have been developed to represent both interior regions and boundaries of objects. Popular techniques include quadtree, moments and medial axis transform for representing interior regions, and chain code, Ψ-s curve, and Fourier descriptors for representing boundaries. Most of the above representations have been generalized to represent 3-D objects. Other schemes are possible considering the additional degree of freedom in representing 3-D objects. Existing representation techniques can be broadly classified as volumetric, surface, line drawing or junction-labeling representations.

This paper [1] discusses the development of a versatile surface representation from a given volumetric scene description. The volumetric scene description scheme employed in this paper is the volume-segment structure developed in [2-3]. The volume-segment structure is obtained by integrating the information and constraints supplied from various 2-D projections using back projection with a volume intersection technique. The scene description is recorded as a hierarchical data structure which decomposes 3-D scene into a set of parallel planar slices; each slice is then characterized by a collection of 2-D shapes which defines the structure at that cross section. This construction process uses only silhouettes and is therefore more robust. The technique of back projection with volume intersection is easy to implement and is general enough to produce 3-D scene description from various 2-D projection structures. For example, this technique has been applied to generate octrees from three orthogonal quadtrees in [4] and to generate 3-D rectangular parallelpiped coding from 2-D rectangular-coded images in [5].

In previous work, a matching algorithm was developed for recognizing isolated 3-D objects [4]. It employed a volume-segment representation and used three-dimensional principal direction projection technique. However, Analyzing images from a general scene with multiple objects is complicated by missing data and occlusion. Hence, it is difficult to accomplish the recognition task by resorting to the 'global' analysis of [4]. Rather, the scene structure should be examined locally and evidence put together to achieve partial scene description and recognition. The strategy we use is to first construct a versatile representation that preserves local object structure and thus facilitates partial scene description and matching.

We discuss a technique to build an explicit surface representation from a general description of a scene containing several occluding objects. We do not require that separate object structures be extracted from the scene description (i.e. the scene description does not need to be completely segmented) prior to surface reconstruction. Incomplete object depictions due to missing data and occlusion are acceptable. To construct the surface description we need to first identify the 3-D object structures in the scene and then to extract the bounding surfaces of objects. A bottom up approach for surface construction is adopted here. First, we develop an algorithm for associating contours. Contours on pairs of consecutive slices are examined and associated based on the amount of overlap between the regions enclosed by the contours. Surface elements need to be fitted in between pairs of associated contours to establish the local surface structure. A relaxation and searching algorithm is introduced for surface triangulation. These surface elements are then coalesced to form larger object facets. The resulting surface structure is recorded in a polygon table which is the collection of the polygonal patches that forms the bounding surface description of the 3-D objects in the scene.

Some experimental results are shown below. Figures D.2.1, D.3.1 and D.4.1 show the wire frame 3-D structure of a bus, an object with a hole and scene with multiple objects, respectively. Figures D.2.2, D.3.2 and D.4.2 are the surface structures constructed for Figures D.2.1, D.3.1, D.4.1, respectively, as viewed from different angles.

References

[1] Y. F. Wang and J. K. Aggarwal, "Surface Reconstruction and Representation of 3-D Scenes", *Pattern Recognition*, Vol. 19, No. 3, 1986, pp. 197-207.

[2] W. N. Martin and J. K. Aggarwal, "Occluding Contours in Dynamic Scenes", *Proceedings of Conference on Pattern Recognition and Image Processing*, Aug. 1981, pp. 189-192.

[3] W. N. Martin and J. K. Aggarwal, "Volumetric Description of Objects from Multiple Views", *IEEE Transactions on PAMI*, Vol. PAMI-5, No. 2, Mar. 1983, pp. 150-158.

[4] Y. F. Wang, M. J. Magee and J. K. Aggarwal, "Matching Three-Dimensional Objects Using Silhouettes", *IEEE Transactions on PAMI*, Vol. PAMI-6, No. 4, Jul. 1984, pp. 513-518.

[5] Y. C. Kim and J. K. Aggarwal, "Rectangular Parallelpiped Coding: A Volumetric Representation of Three-Dimensional Objects", *IEEE Journal of Robotics and Automation*, Vol. RA-2, No. 3, Sept. 1986, pp. 127-134.

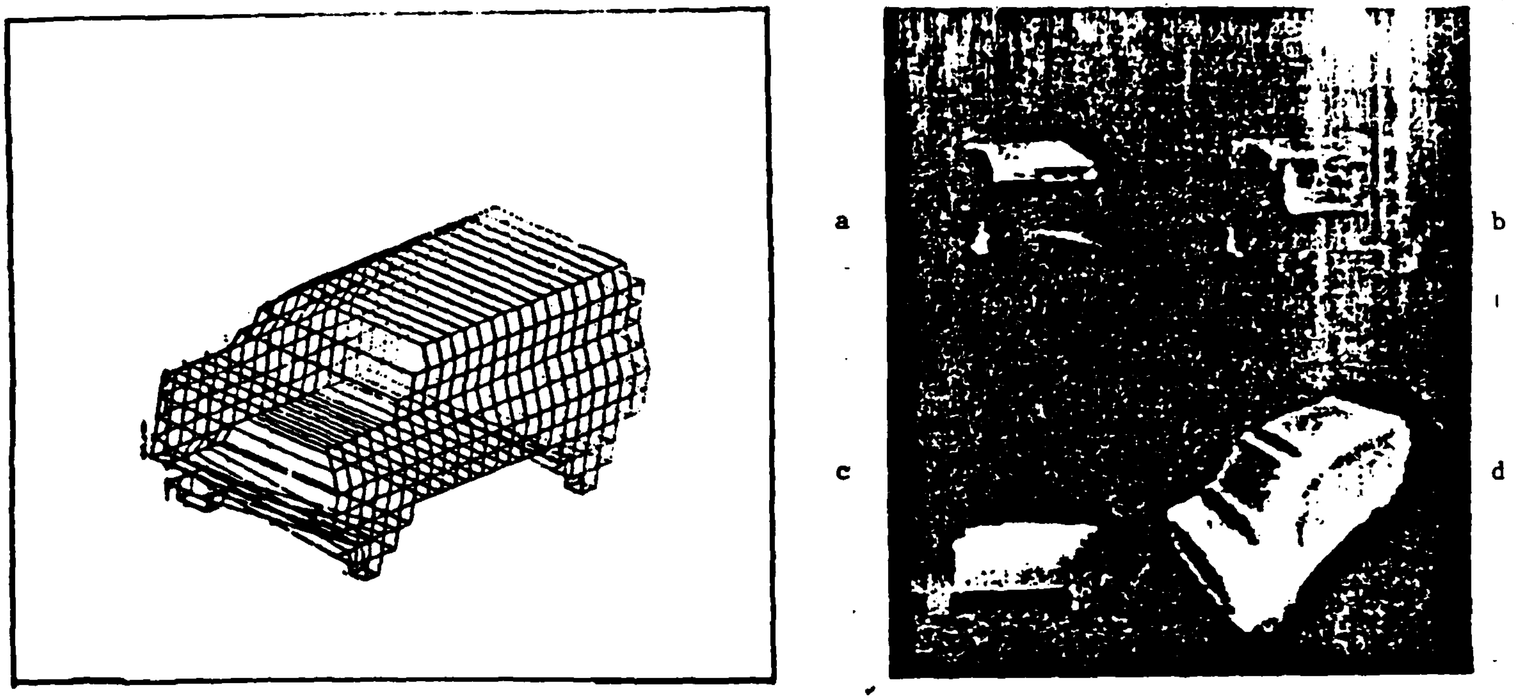


Figure D.2. (1). Wire frame 3-D structure of a bus, (2). Surface structure of the bus as viewed from the back (a), front (b), side (c), and top (d).

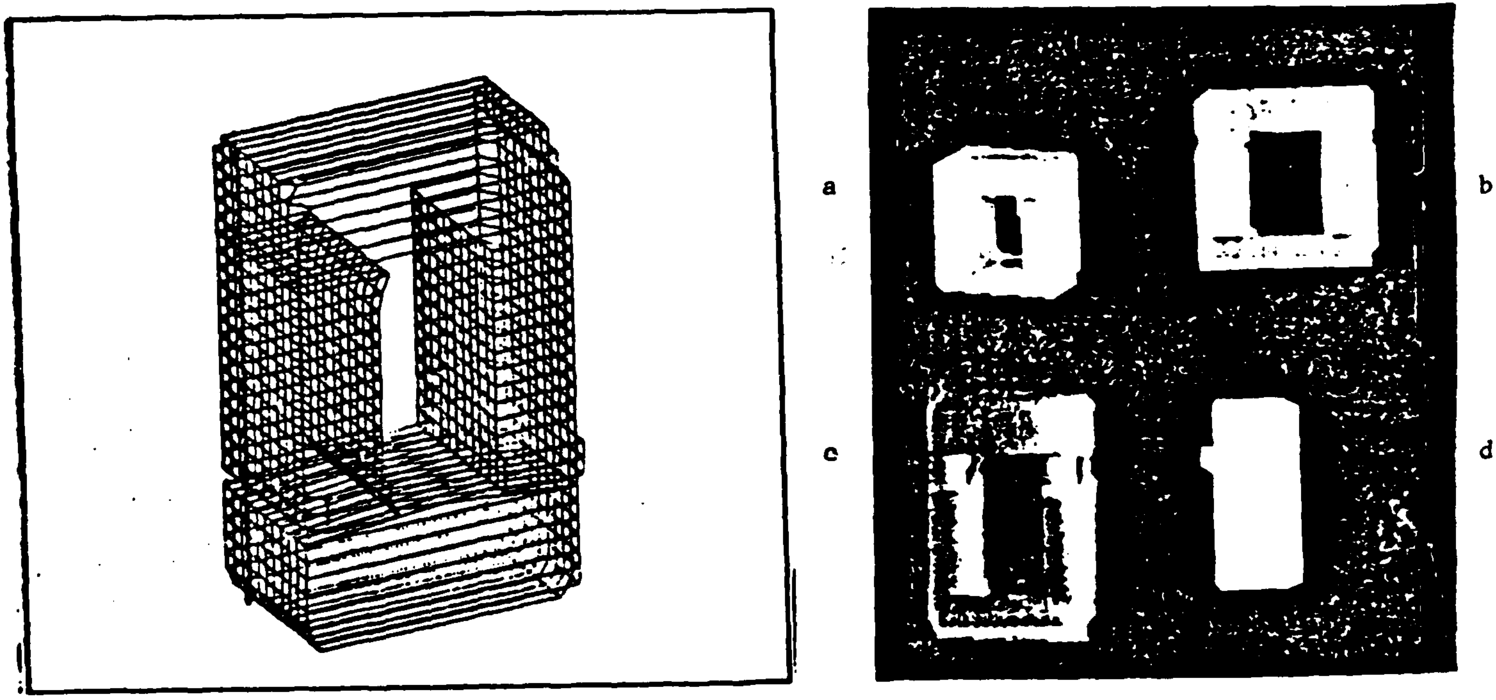


Figure D.3. (1). Wire frame 3-D structure of object with a hole, (2). Surface structure of object with a hole as viewed from side (a), (c), and (d), and front (b).

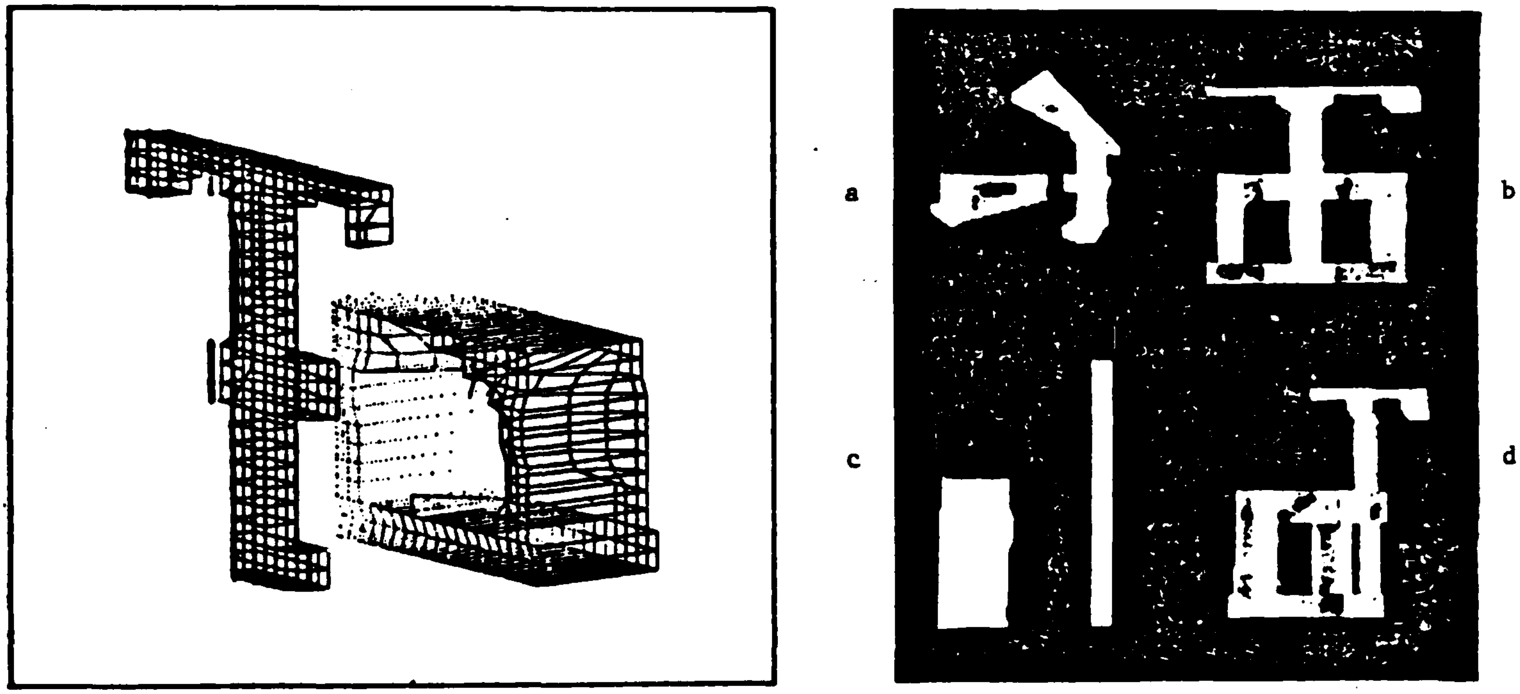


Figure D.4. (1). Wire frame 3-D structure of the scene with multiple objects, (2). Surface structure of the scene with multiple objects as viewed from side (a), (c), and (d), and front (b).

E. Parallel 2-D Convolution on a Mesh Connected Array Processor

2-D convolution is a basic operation frequently used in image processing. Conventional filtering in the spatial domain, template matching, various feature extraction schemes (gradient, Laplacian, Edge detector, etc.), and correlation techniques [1],[2] all use convolution. However, convolution is a computationally intensive task. Most processing structures proposed for 2-D convolution are in the form of systolic arrays. This approach matches the computation speed to the I/O speed [3],[4],[5] has described 2-D convolution methods using systolic arrays. The processing time for the convolution of an image in this approach is proportional to the number of pixels in an image, usually using the same number of processing elements (or cells) as that of the coefficients of a convolution window. Therefore, its computation power may be said to be somewhat limited in that the whole image cannot be processed simultaneously. When a high-speed parallel I/O mechanism is available or an image already resides in a processing structure after some preprocessing, the convolution must be compute-bound. Moreover, systolic array convolvers have been proposed for only square windows. We notice that other types of windows (circular, rectangular, diamond, etc.) as well as square windows are frequently employed in image processing tasks.[1],[6].

We have proposed a parallel algorithm for 2-D convolution for a mesh-connected array processor.[7] The convolution time for an image is proportional to the number of window coefficients. A mesh structure can provide high local communication throughput, especially near neighbor communication [8]. This characteristic is exploited in our approach. The basic idea is that a mesh-connected array having a dimension equal to the size of the image is considered as superimposed sub-arrays of convolution window size and the convolutions for all pixels are carried out in parallel, fully utilizing the entire processing structure. This idea was also briefly mentioned by Young [9]. However, neither a detailed description nor a quantitative analysis was given. Moreover, only a square window was mentioned. Here, we generalize the idea to windows of various shapes and arbitrary size, and give a quantitative analysis on the number of computation steps required for each window.

It is observed that a 4-neighbor-connected mesh structure gives an ideal convolution path for most square and rectangular windows. Therefore, for square and rectangular windows, the 4-neighbor-connected mesh would be preferred. The convolution paths of diamond and circular windows are considerably shortened by using a 6-neighbor-connected mesh, i.e., in the worst case, the path length is greater than the ideal path length, by one. And a 8-neighbor-connected mesh structure can provide an ideal convolution path for all windows except Mx1 rectangular windows. From the above discussion, it may be said that the 6-neighbor-connected mesh array is the best compromise for 2-D convolution of the window types considered here in terms of speed (path length) and hardware cost.

One of the characteristics of our scheme is that few registers are required in each PE and a simple control method is employed as in a systolic array. If more registers which may temporarily stack the partial results are available, the convolution path can be optimized further at the expense of relatively complicated control.

We have proposed a parallel algorithm and processing structure for 2-D convolution. The key idea is to exploit the systolic processing concept on a mesh-connected array. For most windows considered, optimal (shortest possible) paths have been found. This scheme provides a fast 2-D convolution which requires nearly the same number of steps as that of window coefficients for most types of windows. The simple processing element (cells) makes it possible to map the proposed algorithm onto VLSI chips easily. One of the advantages of the proposed parallel convolution scheme is that it may be extended to windows of arbitrary shapes for which a conventional systolic array may not be easily devised, and to 3-D convolution.

References

[1] A. Rosenfeld and A. C. Kak, "Digital Picture Processing," Academic Press, 1982.

[2] E. L. Hall, "computer Image Processing and Recognition," Academic Press, 1979.

[3] H. T. Kung, "why Systolic Architectures?," *IEEE Computer*, pp. 37-46, January 1982.

[4] H. T. Kung and S. W. Song, "A systolic 2-D Convolution Chip," *IEEE Computer Society Workshop on Computer Architecture for Pattern Analysis and Image Database Management*, pp. 159-160, November 1981.

[5] H. T. Kung and R. L. Picard, "Hardware Pipelines for Multi-Dimensional Convolution and Resampling," *IEEE Computer Society Workshop on Computer Architecture for Pattern Analysis and Image Database Management*, pp. 273-278, November 1981.

[6] D. Marr and E. Hildreth, "Theory of Edge Detection," *Proc. Royal Society London*, vol. B207, pp. 187-217, 1980.

[7] S.-Y. Lee and J. K. Aggarwal, "Parallel 2-D Convolution on a Mesh Connected Array Processor," *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Miami, Fl. June 1986.

[8] M. J. B. Duff, "Review of the CLIP Image Processing System," *Proceedings of the National Computer Conference*, pp. 1055-1060, 1978.

[9] T. Y. Young, P. S. Liu, and Y. Gao, "Reconfigurable VLSI Arrays for Pattern Analysis and Image Processing," *IEEE Computer Society Workshop on Computer Architecture for Pattern Analysis and Image Database Management*, pp. 118-124, October 1983.

F. A PARALLEL PROCESSING TECHNIQUE EXPLOITING IMAGE PARALLELISM VIA THE HYPERCUBE

Recently, active research on parallel processing has been carried out in many applications to overcome the limitation of conventional sequential machines. Image processing is a field in which such parallel processing is inevitably required. The time-consuming nature of image processing comes from the fact that it requires several levels of repetitive operations on each pixel, subregion, or some other data structure and also that an enormous amount of data needs to be processed. A number of parallel architectures have been general purpose architectures and the functionally dedicated architectures. However, it appears that we do not know much about how to efficiently use these parallel processing systems yet. This problem is as important as the design of a parallel architecture.

Basically, parallel image processing exploits the two fundamental modes of parallelism in image processing tasks: image parallelism and function parallelism. [2] Image parallelism is a kind of spatial parallelism, i.e., the same operation is repeated on each pixel or subregion so that an image frame may be partitioned into a set of subimages which can be processed by multiple processing elements (PEs) for speed-up. On the other hand, function parallelism is a temporal parallelism, i.e., an image processing task (function) consists of several levels of processing. Here we divide an image processing function into subfunctions and utilize the scheme of pipelining. This method is useful when a sequence of images needs to be processed.

The efficiency of exploiting image parallelism is determined by communication overhead. This overhead is mainly due to the distribution of subimages to a set of PEs, data exchange during computation, and the collection of local results. We cannot just keep partitioning an image since as we divide it further the communication overhead increases in general. After some point, the communication overhead might wipe out the advantage of parallel processing and, therefore, the processing time becomes even longer if we employ more PEs. Therefore, it is important to determine the optimum partitioning in terms of the number of PEs employed. In this paper, we address this problem when image parallelism is exploited on the hypercube structure.

The hypercube is a multiprocessor system structure in which various topologies for parallel processing can be imbedded. [3] These include the inherent multidimensional meshes, a ring, a pipeline, a tree, etc. We imbed in the hypercube a pseudo binary tree which is an efficient topology to combine the local results. It requires $\log_2^N$ steps of communication to combine N local results. Additionally, every communication is carried out along a single link, in other words, the messages are not routed through several nodes so that the communication overhead can be minimized. We also examine fast schemes for distribution of subimages. As a whole, the exploitation of image parallelism on the hypercube is modelled by a set of parameters which point where the processing time is minimized. [4]

We have considered schemes to exploit image parallelism using the hypercube structure. First, we proposed a pseudo binary tree imbedded in the hypercube, which is an efficient topology to collect local results. It takes $\log_2^N$ steps to combine N results just like a binary tree. But a pseudo binary tree requires a hypercube of smaller dimension than its corresponding binary tree. Moreover, all PEs in a hypercube can be utilized for a pseudo binary implementation while only at most half of PEs for a binary tree implementation. A computational model which can abstract a physical system (hypercube) closely enough by a set of parameters was built. Then we derived the formulas of the processing time per output for three processing schemes: the broadcast, singlecast and modified singlecast schemes. The model with the formulas may be used to find the optimum number of PEs to employ for the tasks of the type we considered here.

When the communication time between PEs in the cube is much smaller than that between a PE and the controller, the modified singlecast scheme can give a performance close to that of the broadcast which is an optimum scheme. The performance of the singlecast scheme becomes quite comparable to that of the broadcast scheme if the set-up time is negligible compared to the transmission time.

References

[1] S. Yalamanchili, K. V. Palem, L. S. Davis, A. J. Welch, and J. K. Aggarwal, "Image Processing Architecture: A Taxonomy and Survey," *Progress in Pattern Recognition*, vol. II, pp. 1-37, North Holland, 1985.

[2] S.-Y. Lee and J. K. Aggarwal, "A Problem-Driven Approach to Parallel Image Processing: System Design and Scheduling," *Proceedings of the IEEE Internaitonal Conference on Systems, Man and Cybernetics*, pp. 680-686, Tucson, AZ, November 1985.

[3] Y. Saad and M. H. Schultz, "topological Properties of Hypercubes," *Research Report UALEU/DCS/RR-389, Yale University*, June 1985.

[4] S.-Y. Lee and J. K. Aggarwal, "Exploitation of Image Parallelism Via The Hypercube," *Proceedings of the 2nd Conference on Hypercube Multiprocessors*, Knoxville, TN., September 1986.

END

8-87

DTIC