

UNIVERSITY OF MANITOBA

An Interactive Computer Graphics Based Method
for the Direct Machine Encoding of Building Descriptions.

by

Harold E. Jackson

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE
OF MASTER OF ARCHITECTURE

DEPARTMENT OF ARCHITECTURE
FACULTY OF ARCHITECTURE
UNIVERSITY OF MANITOBA

WINNIPEG, MANITOBA

OCTOBER 1979

AN INTERACTIVE COMPUTER GRAPHICS BASED METHOD
FOR THE DIRECT MACHINE ENCODING OF BUILDING DESCRIPTIONS

BY

HAROLD EDWARD JACKSON

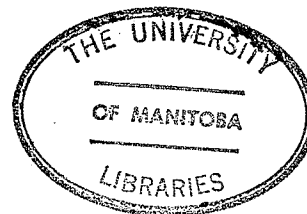
A dissertation submitted to the Faculty of Graduate Studies of
the University of Manitoba in partial fulfillment of the requirements
of the degree of

MASTER OF ARCHITECTURE

✓
© 1979

Permission has been granted to the LIBRARY OF THE UNIVER-
SITY OF MANITOBA to lend or sell copies of this dissertation, to
the NATIONAL LIBRARY OF CANADA to microfilm this
dissertation and to lend or sell copies of the film, and UNIVERSITY
MICROFILMS to publish an abstract of this dissertation.

The author reserves other publication rights, and neither the
dissertation nor extensive extracts from it may be printed or other-
wise reproduced without the author's written permission.



ABSTRACT

This thesis presents an operative architectural experiment in interactive computer graphics and computer aided building design. An interactive computer graphics based method is described for the direct machine encoding of building descriptions. The method is designed to permit an architect, while creating a series of plan section drawings of a building directly on the screen of a computer graphics terminal, to automatically generate a digital data base within a computer that describes the geometry, topology, and attributes of that building's enclosures and enclosing surfaces. The method makes use of a planar graph analogy, in which the drawings are treated as planar graphs and the building as a three dimensional network overlay of these graphs. The building data base so generated by the method may be subsequently used in conjunction with other programmed methods to algorithmically examine aspects of that building's design. An interactive computer graphic system, known as GRAPHIX, is experimentally implemented based on this method and its operation illustrated in conjunction with an existing thermal analysis program, as an architectural design aid.

The GRAPHIX program was written in PDP 11 assembler code. The code was written expressly for execution in a PDP 11/20 mini-computer equipped with 16K words of core memory, terminal, floppy disk mass storage, Tektronix 4002A graphics display terminal, sonic pen (attached about the 4002A screen), and a Tektronix 1610 hardcopy unit.

ACKNOWLEDGEMENTS

This thesis was executed under the joint direction of members of the Department of Architecture, Faculty of Architecture, University of Manitoba, and the Department of Architecture, School of Architecture, Carleton University, Ottawa. To all those whose interest and encouragement made possible my graduate study in these Departments and its culmination in this document, I am grateful. In particular, however, I would like to express my gratitude to the following:

To my Thesis Advisor, Prof. Ben Harnish, Dept. of Environmental Studies, Faculty of Architecture, University of Manitoba, for his friendship, guidance and patience in the execution of this work,

To my Thesis Examiner, Dean Jack Anderson, Faculty of Architecture, University of Manitoba, for his knowledge, patience and direction,

To my Thesis Examiner, Professor Jean Marie Comeau, School of Architecture, Carleton University, for his unparalleled excellence as an educator and his patient assistance in the development of the planar graph methodology outlined herein,

To my Thesis Examiner, Professor Jim Wells, Dept. of Computer Science, University of Manitoba, for his encouragement and timely assistance in the execution of this work,

To Dean D. Shadbolt, School of Architecture, Carleton University, for his assistance in arranging my studies within his School,

To Mr. Bill Cope, former Professor, School of Architecture, Carleton University, for initiating my study at Carleton University,

To the former Directors, Mr. Paul Dirkson and Dr. M.S. Doyle, of the University of Manitoba Computer Services, for the provision of the computer equipment upon which the GRAPHIX program was implemented,

To Mr. Larry McNish, Systems Programmer, University of Manitoba Computer Centre, for his untiring personal assistance, programmatic and system expertise, and exciting insights into the world of computer systems,

To Dr. R.C. Read, Mathematics Department, University of Waterloo, for his suggestions pertaining to the resolution of the critical planar graph inspection algorithm,

To Mr. R. Alan Dakin, Systems Analyst, for his generous assistance and suggestions in the preparation of this work,

and, To Mrs. Emily O'Seasain, who typed this work using I.B.M.'s TEXT360 system.

In closing, I would like to thank my wife, Carole, whose assistance can never be adequately acknowledged.

Harold Jackson

Carole Jackson
Amanda Jackson
and, Squeaky.

TABLE OF CONTENTS

Abstract	11
Acknowledgements	111
Table of Contents	v
Table of Figures	vii
Preface	ix
CHAPTER 1: INTRODUCTION	page 2
History	page 2
Intent	page 3
The Method	page 5
CHAPTER 2: EXPLICIT METHODS OF REPRESENTATION	page 15
Introduction	page 15
Section 2.1 Drawing Input, Storage, and Retrieval	page 17
The Graphical Data Structure	page 17
The GRAPHIX Graphical Data Structure	page 18
The Graphical Display Sub-set	page 20
The Secondary Component Sub-set	page 27
The LIBRARY Sub-set	page 28
Section 2.2 Assembling and Enumerating the Planar Graph	page 32
Introduction	page 32
Procedural Summary	page 32
Algorithm 2.1 Construct the planar graph G	page 34
Algorithm 2.2 Enumerate the surface boundaries of the graph	page 41
Algorithm 2.3 Enumerate the regions of the graph	page 54
Algorithm 2.4 Attach attribute labels to the regions	page 63
Algorithm 2.5 Construct the dual graph G(d)	page 68
Section 2.3 Assembling and Enumerating the overlay graph G _O	page 72
Introduction	page 73
Procedural Summary	page 74
Algorithm 2.6 Construct the overlay graph G _O	page 76
Algorithm 2.7 Attach adjacent region labels to graph G _O	page 97
Algorithm 2.8 Enumerate all vertically contiguous regions	page 103
Summary	page 108
CHAPTER 3: THE GRAPHIX SYSTEM	page 111
Introduction	page 111
The GRAPHIX System for Building Description	page 111
The GRAPHIX Sub-systems	page 113
The GRAPHIX MONITOR Sub-system	page 113
The BUILDING COMPONENT Sub-system	page 119
The PLAN SECTION DRAWING Sub-system	page 124
The Building MODEL GENERATOR Sub-system	page 132
Conclusion	page 133

CHAPTER 4: CONCLUSION	page 137
Introduction	page 137
Observations	page 138
1.0) The Explicit Methods.	page 138
2.0) The GRAPHIX System.	page 140
Conclusions	page 143
APPENDICIES	page 148
APPENDIX A: AN INTRODUCTION TO GRAPH THEORY	page 150
Introduction	page 150
What is a Graph	page 150
Connectedness	page 154
Walks	page 156
Planarity	page 158
APPENDIX B: THE RESULTS OF A BUILDING THERMAL ANALYSIS	page 162
APPENDIX C: A VIDEO TAPE OF THE GRAPHIX SYSTEM	page 189
BIBLIOGRAPHY	page 191

TABLE OF FIGURES

Figure 1.1	A NESTING SYSTEM OF RECTANGULAR POLYHEDRA. . . .	page 7
Figure 1.2	AN ALTERNATING SERIES OF PLAN SECTION DRAWINGS. . . .	page 8
Figure 1.3	THE GRAPH REPRESENTATIONS FOR A BUILDING. . . .	page 10
Figure 2.1	THE GRAPHICAL DATA STRUCTURE.	page 19
Figure 2.2	THE GRAPHIX GRAPHICAL DATA STRUCTURE.	page 20
Figure 2.3	THE SUB-PICTURE DISPLAY CODE.	page 22
Figure 2.4	DRAWING SYMBOLS: LINEAR CLASS.	page 24
Figure 2.5	DRAWING SYMBOLS: LABEL CLASS.	page 25
Figure 2.6	DRAWING SYMBOLS: AREAL CLASS.	page 26
Figure 2.7	THE SECONDARY COMPONENT SUB-SET.	page 28
Figure 2.8	THE LIBRARY SUB-SET.	page 29
Figure 2.9	A GRAPHIX GRAPHICAL DATA STRUCTURE.	page 31
Figure 2.10	SURFACE BOUNDARY PARSING.	page 37
Figure 2.11	THE BASIC P,D,V NOTATION SYSTEM.	page 39
Figure 2.12	A BASIC PDV DESCRIPTION OF A DRAWING.	page 41
Figure 2.13	COUNTER-CLOCKWISE ORDER OF VERTICES.	page 42
Figure 2.14	SOURCE BEARING' ANGLE.	page 43
Figure 2.15	THE FIRST AVAILABLE RIGHT HAND TURN.	page 45
Figure 2.16	THE CIRCUMSCRIBING CIRCUITS OF A GRAPH.	page 46
Figure 2.17	BEGIN TRAVERSE OF EXTERIOR BOUNDARY CIRCUIT. . . .	page 49
Figure 2.18	COMPLETION OF BOUNDARY CIRCUIT TRAVERSE.	page 51
Figure 2.19	THE BOUNDARY CIRCUITS OF THE GRAPH.	page 53
Figure 2.20	PARENT PARTS OF THE GRAPH.	page 55
Figure 2.21	POINT LYING IN A REGION OF THE GRAPH.	page 56
Figure 2.22	THE BOUNDARY CIRCUIT(S) OF A REGION.	page 57
Figure 2.23	THE ENUMERATED REGIONS OF A GRAPH.	page 63
Figure 2.24	THE TARGET FOR AN AREAL CLASS SUB-PICTURE. . . .	page 65
Figure 2.25	A TARGET WITHIN A COMPLEX REGION.	page 66
Figure 2.26	THE LABELLED REGIONS OF THE GRAPH.	page 68
Figure 2.27	THE DUAL GRAPH $G(d)$ OF THE PLANAR GRAPH G	page 69
Figure 2.28	ADJACENT BOUNDARY CIRCUITS, REGIONS AND COLOURS. .	page 70
Figure 2.29	THE ADJACENT REGIONS OF THE GRAPH.	page 72
Figure 2.30	THE UPPER AND LOWER GRAPHS $G(u)$ AND $G(l)$	page 77
Figure 2.31	THE PARENT, TEMPORARY, AND OVERLAY GRAPHS. . . .	page 78
Figure 2.32	FIRST AND SECOND OCCURRENCES IN THE TAG LIST. . .	page 79
Figure 2.33	THE GORU AND GORL LISTS OF THE OVERLAY GRAPH. . .	page 81
Figure 2.34	THE RESEQUENCING OF EDGE INTERSECTS IN VPL. . .	page 84
Figure 2.35	INTERSECTS ON SECOND OCCURRENCE EDGES.	page 87
Figure 2.36	ENUMERATE THE INTERSECT VERTICES OF GRAPH GT . .	page 92
Figure 2.37	THE OVERLAY GRAPH GO	page 97
Figure 2.38	COLOURING THE OVERLAY GRAPH GO	page 99
Figure 2.39	VERTICAL NETWORK LINKAGE LISTS: GRAPH GO	page 100
Figure 2.40	THE COLOURED OVERLAY GRAPH GO	page 103
Figure 2.41	VERTICAL NETWORK LINKAGE LISTS: $G(U)$ AND $G(L)$. .	page 105
Figure 2.42	VERTICALLY CONTIGUOUS REGIONS OF THE GRAPH G . .	page 106
Figure 2.43	THE BUILDING NETWORK CONSTRUCTION PROCESS. . . .	page 108

Figure 3.1	INTERACTIVE COMPUTER GRAPHICS WORKSTATION. . . .	page 112
Figure 3.2	THE GRAPHIX SYSTEM.	page 114
Figure 3.3	THE GRAPHIX SUB-SYSTEM DISPLAY.	page 116
Figure 3.4	THE BUILDING COMPONENT LIBRARY DATA BASE. . . .	page 119
Figure 3.5	THE BUILDING SUB-SYSTEM DISPLAY.	page 121
Figure 3.6	THE PLAN SECTION DRAWING SUB-SYSTEM.	page 125
Figure 3.7	THE STOREY DRAW AND NAME MODULE DISPLAYS. . . .	page 127
Figure 3.8	ENTRY AND ERASURE OF SYMBOLS FROM THE DISPLAY. .	page 128
Figure 3.9	PLAN SECTION DRAWING ILLUSTRATIONS (1 of 3). . .	page 130
Figure 3.10	PLAN SECTION DRAWING ILLUSTRATIONS (2 of 3). . .	page 131
Figure 3.11	PLAN SECTION DRAWING ILLUSTRATIONS (3 of 3). . .	page 132
Figure 3.12	THE THERMAL ANALYSIS OF A BUILDING DESIGN. . . .	page 134
Figure A.1	DIFFERENT MAPPINGS OF A GRAPH.	page 151
Figure A.2	A SUBGRAPH.	page 153
Figure A.3	A DISCONNECTED GRAPH.	page 155
Figure A.4	CUT-VERTICES AND BRIDGES.	page 157
Figure A.5	WALKS, TRAILS, PATHS AND CYCLES.	page 158
Figure A.6	A PLANAR GRAPH.	page 159

PREFACE

This thesis was initiated in order to address a situation in the field of Computer Aided Building Design that has remained relatively unchanged for the past ten years. This situation has three major aspects to it.

Firstly, there is an increasing demand that sophisticated computer methods of numeric analysis, particularly environmental design analysis methods, be applied within the building design process.

Secondly, there are currently no effective methods for entering into the computer the detailed three dimensional building design descriptions which are required by these analytic methods, other than the prototypical methods which employ the interactive computer graphics technology.

And thirdly, while prototypical systems do exist, there is no readily available or generally accepted interactive

computer graphics method for entering three dimensional building descriptions into a computer.

The work contained in this thesis has attempted to address each aspect of the foregoing situation. It describes an interactive computer graphic method to enter building descriptions into the computer. It describes the implementation of that method as an operational system. And it describes the use of that system, in conjunction with an existing thermal heat loss-gain program, to demonstrate its potential as a Building Design Aid.

The method developed for entering building descriptions into a computer permits the designer to 'draft' the description directly on the screen of an interactive computer graphics terminal. The method permits the designer to 'draft' the building design as a series of plan section drawings, a mode of graphic communication familiar to most designers. The method arms the computer with the logic necessary for it to automatically extract the information needed to represent the building within the computer in its three dimensions. Modifications to the design, and the internal digital representation of the building design, are simply accomplished by the the designer making alterations to the plan section drawings. The building description held in the computer will support any one of a variety of numerical analysis procedures

and it is particularly amenable to most forms of environmental design analysis.

The body of the thesis describes the fundamentals of the method developed for building description. The specific logics of the method are then presented in detail, supported by a glossary of terms in the appendix. This algorithmic description will be of interest to anyone involved in the technics of Computer Aided Building Design.

A description of an operational computer system for building description, known as GRAPHIX, as implemented on a specific computer system is presented. This implementation and the resulting 'GRAPHIX' system served two purposes. It provided a vehicle for investigating some of the issues that surround the implementation of any interactive computer graphics method for building description. And, it provided a vehicle to demonstrate and investigate some issues surrounding the actual use of interactive computer graphics based systems for Computer Aided Building Design. A single storey dwelling was 'drafted' into the computer in order to facilitate the demonstration and the building subsequently analysed for thermal heat loss/gain using an existing building analysis computer package.

The conclusions to the work are certainly not exhaustive, nor are they intended to be so. However they do itemize some of the difficulties and development opportunities in this work that researchers active in the field might find pertinent to future studies.

It is hoped that the work will in some way add to the literature of computer Aided Building Design, as well as other related fields. It is hoped that the work will indicate to other C.A.B.D. researchers a way of emulating the current work methods employed by architectural designers and planners. And lastly, it is hoped that this work will stimulate other researchers to approach Computer Aided Building Design from the architectural designers viewpoint.

INTRODUCTION

CHAPTER 1: INTRODUCTION

"Computer: [1] A data processor that can perform substantial computation, including numerous arithmetic or logic operations, without intervention by a human operator during the process. [2] A device capable of solving problems by accepting data, performing described operations on the data, and supplying the results of these operations."(1)

History

The objective of Computer Aided Building Design is to develop procedures that will permit designers to apply more science and technology, in a cost effective manner, to the design process for the built environment.

When architects first began researching how they might effectively employ the information processing facilities of the digital computer , it was expected that once 'state of the art' building design analysis and evaluation procedures were encoded in computer programs, that both the minimal cost of executing these programs and the detailed information supplied by these programs would assure their general application. However, as researchers continued to gain practical experience with the technology, and as the body of existing Computer Aided Building Design procedures continued to grow, it became increasingly apparent that neither the state of the computer technology , nor the cost of executing the

design procedures, nor the information derived from the execution of the procedures, could provide sufficient economic incentive to justify the general acceptance of Computer Aided Building Design. For although the technology of the time provided the means necessary to effectively program and run a wide range of potentially useful design operations, it provided no economic means for generating and entering those digital models of the building design that were prerequisite to the use of those programmed procedures(2,3).

Economically viable computer systems for the comprehensive design analysis of buildings did not begin to appear until the late 1960's and early 1970's when, just prior to the commencement of this work, the critical building description encoding problem was first broached. Equipped with the newly introduced interactive computer graphics technology(4), several architectural researchers developed the first graphic methods(5,6,7,8) "for the direct machine encoding of building descriptions"(9). These methods not only permitted the architect to 'draught' the design drawings for a building directly and first hand on the screen of a graphics terminal, but permitted the computer, using the information communicated through the drawings, to automatically and economically construct a digital data base containing a three dimensional description of the fabric of a building design.

Intent

The development of interactive computer graphics based methods for the direct machine encoding of building descriptions and the subsequent implementation of such methods in comprehensive Computer Aided Building Design systems, like ARK-2(10) and PACE1(11), has demonstrated that computer technology can be economically employed to apply more science and technology in the architectural design process. If further advances in the development of economic Computer Aided Building Design systems are to be achieved, knowledge of interactive computer graphic based methods for the direct machine encoding of building descriptions is important.

This thesis presents an operative experiment in interactive computer graphics and Computer Aided Building Design in which a specific interactive computer graphics based method for the direct machine encoding of building descriptions is developed and documented.

The method permits an architect, while creating a series of plan section drawings of a building directly on the screen of a computer graphics terminal, to automatically generate a digital data base within the computer that describes the geometry, topology, and attributes of that building's enclosures and enclosing surfaces. The method makes use of a planar graph analogy in which the drawings are treated as planar graphs and the building as a three-dimensional network overlay of these graphs. The building data base generated using this method may be subsequently used in conjunction with other programmed computer analyses methods to

algorithmically examine various aspects of that building design.

An interactive computer graphics system, known as GRAPHIX, is subsequently implemented based on the previous method. The operation of the system is then illustrated, in conjunction with an existing thermal analysis program, to demonstrate the relevance of the method as an architectural design aid in the building design process.

The Method

Any three-dimensional real world object (eg. a nail) can be represented as an 'element' of some homogeneous character (eg. steel) that occupies a contiguous 'region' of three-dimensional Euclidean space(12) and any complex real world object, such as a building, can be represented as a nesting system of such 'elements' (eg. a nail, embedded in a wood stud, embedded in a partition wall, etc.). In the most direct sense, any object or nesting system of objects can be completely defined when the geometric location and attribute characteristics are specified for each element, whether explicitly in terms of the points of the region that belong to the element or implicitly in terms of the surface boundaries that circumscribe and separate the elements. Once these data are specified, any point, line, surface or element can be identified and its intrinsic geometric, attribute and topologic (both internal and external) properties determined.

Development of an interactive computer graphics based method for

the direct machine encoding of building descriptions involves the creation of some suitable procedures whereby:

- 1) the constituent material and spatial components of a building can be represented as a finite nesting system of discrete three-dimensional 'elements';
- 2) the description of the attribute and three-dimensional geometry for each building 'element' can be graphically entered into the computer, via the two-dimensional communication medium of an interactive computer graphics terminal; and,
- 3) the two-dimensional graphic imagery used to describe the individual building 'elements' can be algorithmically inspected, the fundamental geometric, topologic and attribute space defining information algorithmically extracted, and the description of the building as a finite three-dimensional system of discrete 'elements' constructed and stored.

The method for the direct machine encoding of building descriptions developed in this thesis proposes that the above three procedures be implemented in a manner analogous in convention to that traditionally employed in the preparation of standard architectural orthogonal drawings, such that:

- 1) The constituent materials and spatial components of a building

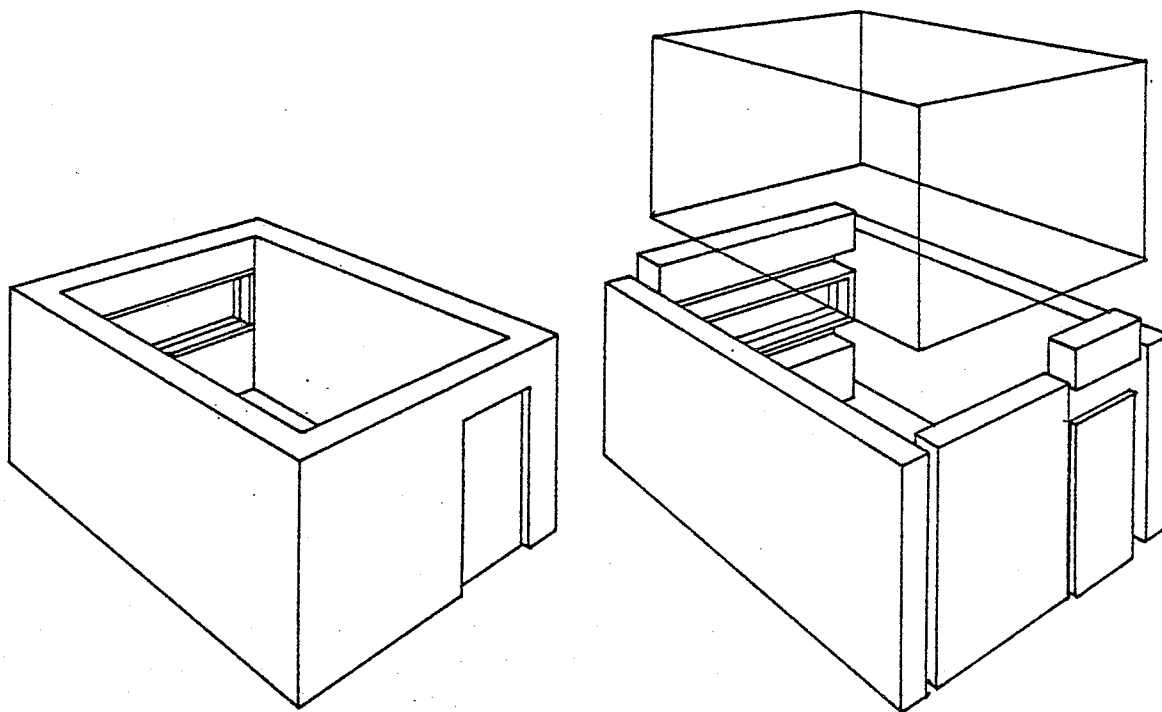


Figure 1.1 A NESTING SYSTEM OF RECTANGULAR POLYHEDRA.

are represented as a nesting system of rectangular polyhedra, called 'containers', possessing homogeneous physical attributes, where (see Figure 1.2)

a material 'container' is used to represent a standard building component assembly such as a door, window, partition wall or floor/ceiling assembly, and

a spatial 'container' is used to represent a volumetric

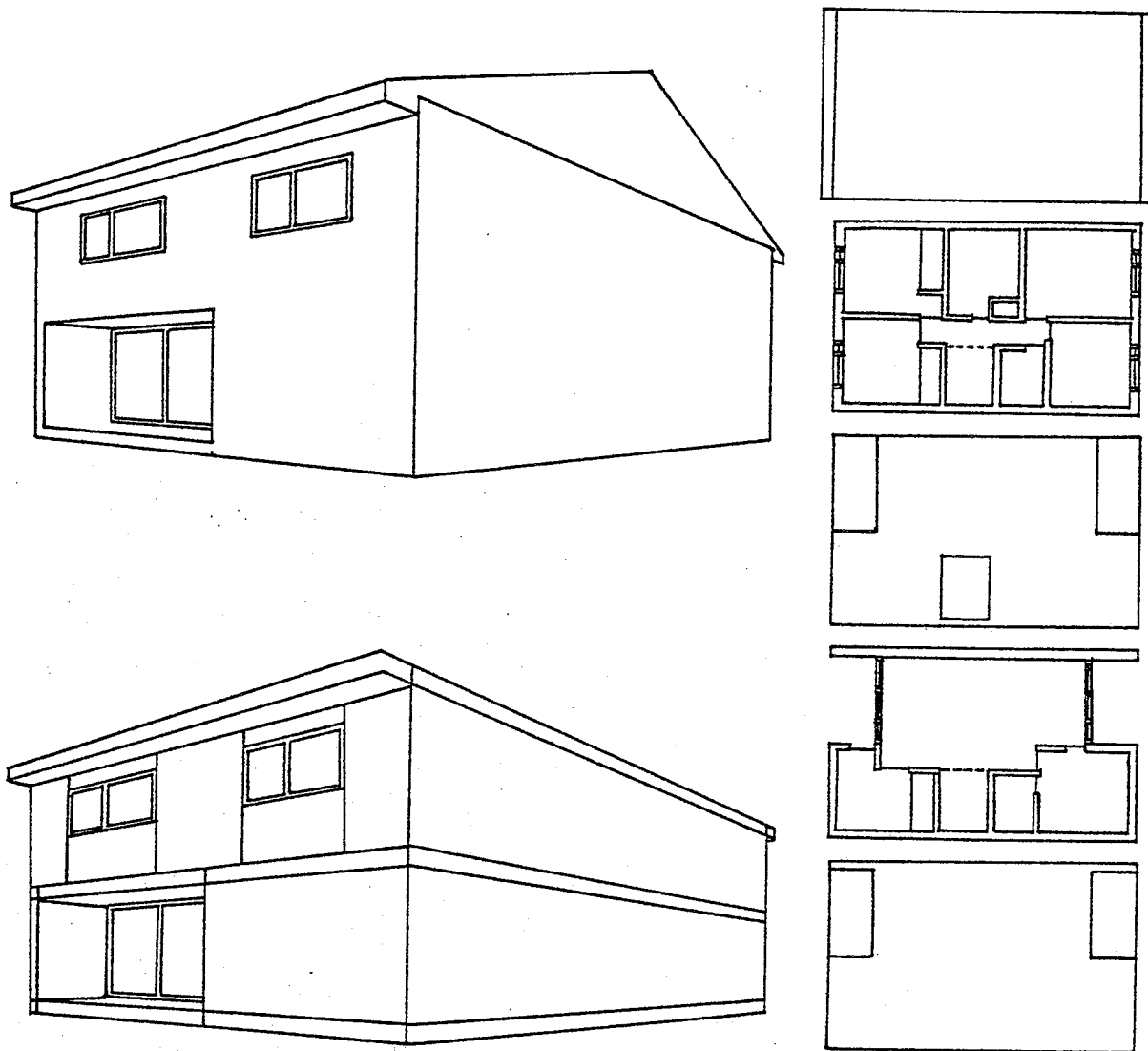


Figure 1.2 AN ALTERNATING SERIES OF PLAN SECTION DRAWINGS.

enclosure such as a room, corridor or the exterior environment to a building.

- 2) The building is graphically described by a series of alternating horizontal storey and floor/ceiling zones, where

- a) each storey and floor/ceiling zone is graphically input to the computer as a plan section drawing (see Figure 1.2),
 - b) the vertical surfaces enclosing each 'container' are mapped as straight line segments in the plan section drawing,
 - c) the contents of each 'container' are mapped as line symbol patterns or as textual labels,
 - d) each zone occupies a uniform vertical dimension,
 - e) each 'container' is considered to extend vertically and continuously through the complete zone height, and
 - f) the vertical extremes of each zone map the horizontal surfaces enclosing the 'containers' of that plan section.
- 3) The fundamental geometric, topologic and attribute information is extracted from the plan section drawings and the final nesting system of 'containers' constructed and stored by treating each plan section drawing as a planar graph (see Figure 1.3(a)) in which;
- each region of the graph will identify a specific 'container',
 - each edge of the graph will identify a vertical surface boundary separating two juxtaposed 'containers', and
 - each vertex will identify the juncture of two vertical

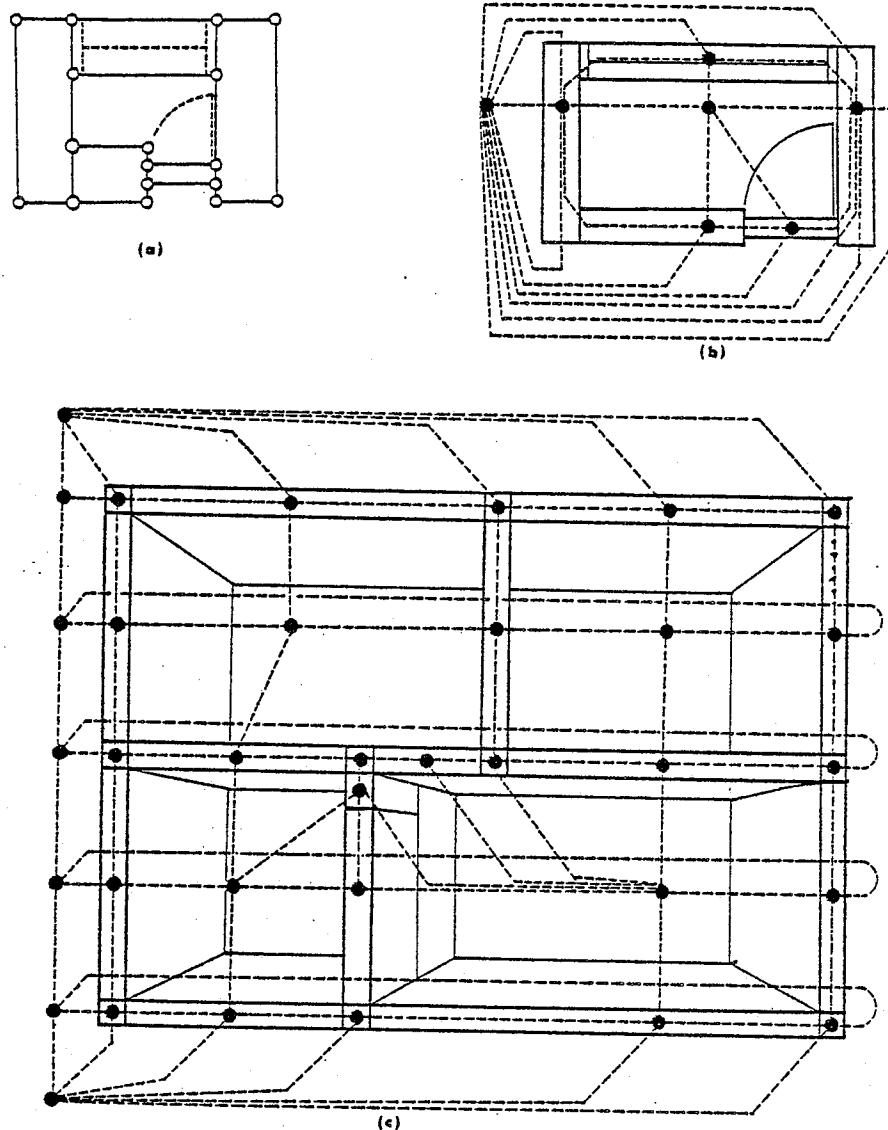


Figure 1.3 THE GRAPH REPRESENTATIONS FOR A BUILDING.

surface boundaries (see Figure 1.3(b)); and,

treating the nesting system of 'containers' as a three-dimensional dual graph network in which;

each vertex of the dual graph network will identify a

'container', and

each horizontal edge will identify a vertical surface boundary separating two horizontally juxtaposed 'containers' and

each vertical edge will identify a horizontal surface boundary separating two vertically juxtaposed 'containers' (see Figure 1.3(b,c)).

The remainder of this thesis expands further upon the method proposed above. Chapter 2 will document in detail the explicit procedures and data structures that were devised to realize the method. Chapter 3 will present a description of an interactive computer graphics system for the direct machine encoding of building description, known as GRAPHIX, that operationally implements the method. And, finally, Chapter 4 will present some observations and conclusions resulting from this work.

References: Chapter 1

- 1 Dorf, Richard C. Computers and Man. California: Boyd and Fraser Publishing Company, 1974. p. 450.
- 2 Teague, Lavette C. Jr. The Representation of Spatial Relationships In A Computer System For Building Design. Research Report R68-25. Massachusetts Institute of Technology, Cambridge, Massachusetts, June 1968. p. 96.
- 3 Spatial Synthesis in Computer Aided Building Design, ed. Charles Eastman. London: Applied Science Publishers Ltd., 1975. p. 14.
- 4 Sutherland, Ivan E. SKETCHPAD: a man-machine graphical communication system. Lexington Massachusetts: Massachusetts Institute of Technology, Lincoln Laboratory, 1963.
- 5 Parlslow, R.D., Prowse, R.W., and Green, R.E. Computer Graphics, Techniques and Applications. New York: Plenum Press, 1969.
- 6 Negroponte, Nicholas, "Toward a Theory of Architecture Machines". A.I.A. Journal, 51, March 1969, No. 3, pp.71-74.
- 7 Roberts, Lawrence A. Machine Perception of Three Dimensional Solids. Research Report TR 315. Massachusetts Institute of Technology, Lincoln Laboratory, 1965.
- 8 Wood, R.C. and Hendren, P.H., "A Flexible Computer Graphic System for Architectural Design". Information Display, March-April 1968, pp. 35-40.
- 9 op. cit. 3. p. 17.
- 10 Dean, Perry, and Stewart, "The ARK 2 System". Progressive

Architecture, July 1971.

11 Maver, Thomas W., "PACE1: Computer Aided Building Appraisal".

The Architects' Journal, July 28, 1971.

12 Carr, C. Stephen. Geometric Modelling. Ph.D. Thesis, University of Utah, 1969. p. 3.

2

THE EXPLICIT METHODS

CHAPTER 2: EXPLICIT METHODS OF REPRESENTATIONIntroduction

Following the selection of a basic building description method, a series of data structures and algorithms were developed. There were two functions to be fulfilled by these. They should support the interactive designer-computer communication of the information describing the plan section drawings of building. And, once in the computer, they should support the transformation of the information into the planar graph and three dimensional dual graph network representations of the building. These data structures and algorithms provided the information input, output, storage and processing framework that was a prerequisite to the subsequent experimental implementation of the method on and as an interactive computer graphics system.

This chapter documents those algorithms and data structures. It is organized into three sections. Each describes a discrete stage in the assembly of the three dimensional network description of the building. Section 2.1 sets out those procedures developed to input, store and display the plan section drawings describing a building. Section 2.2 describes how each of the plan section drawings for a building is first converted into a two dimensional planar graph representation and subsequently enumerated to establish the horizontal spatial relationships required to describe the building.

Finally, in Section 2.3, procedures are described for overlaying vertically adjacent plan section drawings and enumerating the vertical spatial relationships needed to conclude the assembly of the three dimensional network description of the building.

It should be noted that the work reported on within this chapter was carried out under the direction of Prof. J.M.Comeau, School of Architecture at Carleton University. Prof. Comeau not only assisted in the general and detailed formulation of the planar graph representation method for building description, but he also arranged for the timely assistance of Prof. R.C. Read, Department of Mathematics, University of Waterloo, in our effort to implement a circumscribing circuit threading routine (Algorithm 2.2) which proved to be the central problem of this work. It is intended that this chapter might, indirectly, serve to document the invaluable assistance provided to the author by Prof. Comeau.

Section 2.1 Drawing Input, Storage, and Retrieval

This section describes the GRAPHIX Graphical Data Structure developed to input, store and display the plan section drawings for a building. The data structure, together with two ancillary list structures, provides a capture mechanism for the geometric and attribute information used by procedures described in subsequent sections to assemble, first, the two-dimensional and, then, the three-dimensional graph representations of the building.

There are two subsections to this section. A brief description of a typical Graphical Data Structure is presented in the first sub-section to illustrate the general representation stratagem which characterizes the structure. This is then followed by a description of the GRAPHIX Graphical Data Structure in the second sub-section.

The Graphical Data Structure

The term Graphical Data Structure, used by Newman and Sproull in Principles of Interactive Computer Graphics(1), defines a specific method of graphic representation(2) characterized by the use of sub-picture display subroutines. In this method, the graphic display (ie. picture) is treated as an assembly of discrete drawing symbols or sub-pictures. The display commands for each unique sub-picture are defined in separate display subroutines. In these subroutines all the graphic display commands are defined relative

to some arbitrarily selected local origin. Each sub-picture occurring in the drawing display is, in turn, defined using a list of instance transforms. Each instance transform element contains both a pointer referencing the appropriate display subroutine and a set of geometric data used by the display subroutine to position the drawing symbol in the display.

A typical Graphical Data Structure representing a simple modular building plan section drawing is illustrated in Figure 2.1. As can be seen from Figure 2.1(a), the building storey section is constructed of five separate components: two window units, two wall units, and one door unit. Figure 2.1(b) shows a conceptual dissaggregation of the plan section drawing and its representation as a series of sub-picture instances, with each sub-picture instance defined according to its type and geometric location. Figure 2.1(c) shows an equivalent Graphical Data Structure with each sub-picture instance represented by a co-ordinate pair and scale value, and by a call to the subroutine that draws that particular component symbol.

The GRAPHIX Graphical Data Structure

The organization and sub-element composition of each element in the instance transform list of the GRAPHIX Graphical Data Structure is illustrated in Figure 2.2. The information in each element is divided into two functional sub-sets (Graphical Display and Non-Graphical Display) which are sub-divided in turn into various

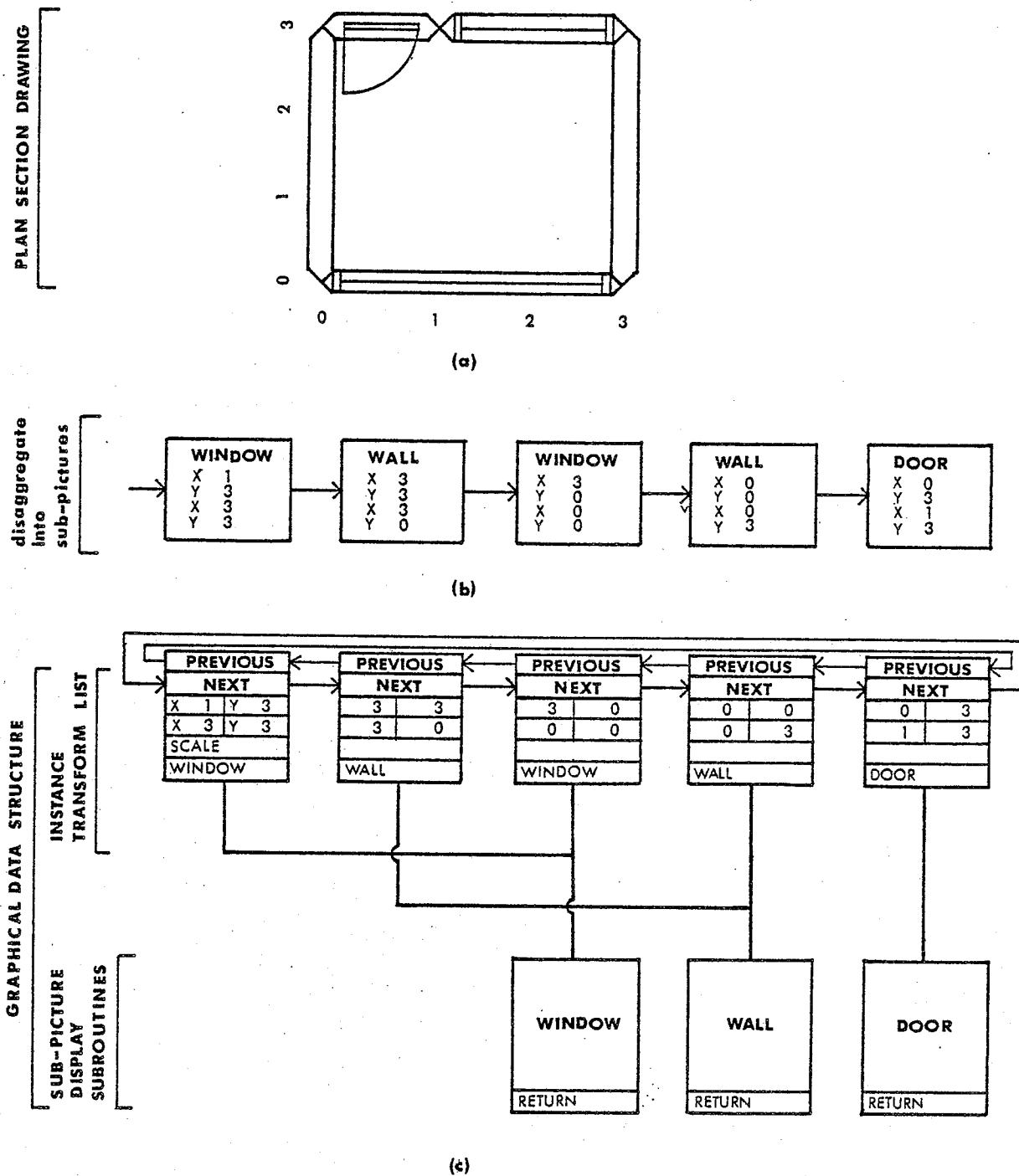


Figure 2.1 THE GRAPHICAL DATA STRUCTURE.

Adapted from a diagram by, Newman, W.M., and Sproull, R.F.,
Principles of Interactive Computer Graphics, 1973. p. 107.

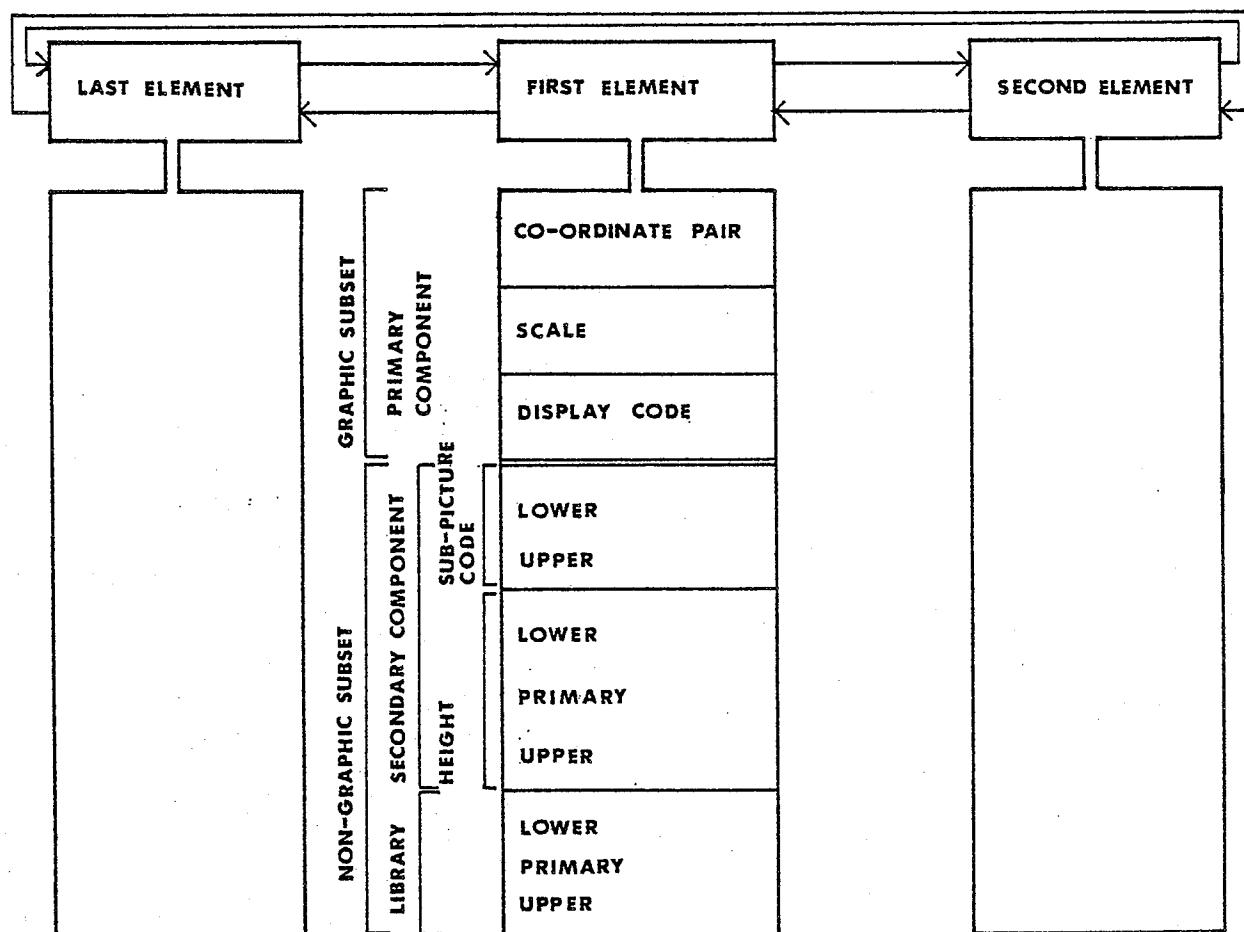


Figure 2.2 THE GRAPHIX GRAPHICAL DATA STRUCTURE.

sub-element components. The remainder of this section illustrates* and describes the functions served by each of these two subsets and their constituents.

* Any illustrations depicting the logical organization of a data structure do not represent the actual physical arrangement of the data nor show all the information stored for use by the processing routines.

The Graphical Display Sub-set

The GRAPHICAL DISPLAY sub-set is used to store information describing a symbol instance in the plan section drawing display. The information contained in the coordinate and scale data items is passed by the picture display processor to the sub-picture display subroutine which generates the symbol display. As illustrated in Figure 2.3, the display code may also be used to reference specific list entries in the Building Component Library. Entries in the Building Component Library store material and functional descriptions particular to an individual building component (thermal conductivity, reflectance, absorbance, cost) which may subsequently be used in the computerized analysis of a building.

Certain arbitrary conventions have been adopted for the purposes of simplicity in sub-picture positioning and sub-picture type. Essentially these conventions require that the building described be modular and that the plan section drawing of the building be defined using three separate classes of sub-picture. Table 2.1 contains a brief summary of the conventions established to deal with sub-picture placement. The following paragraphs describe the three classes of sub-picture type.

Three classes of sub-picture are defined for use in the plan section drawing and building description process. Each of these classes communicates a specific unit of geometric or geometric and attribute information describing building containers. The three

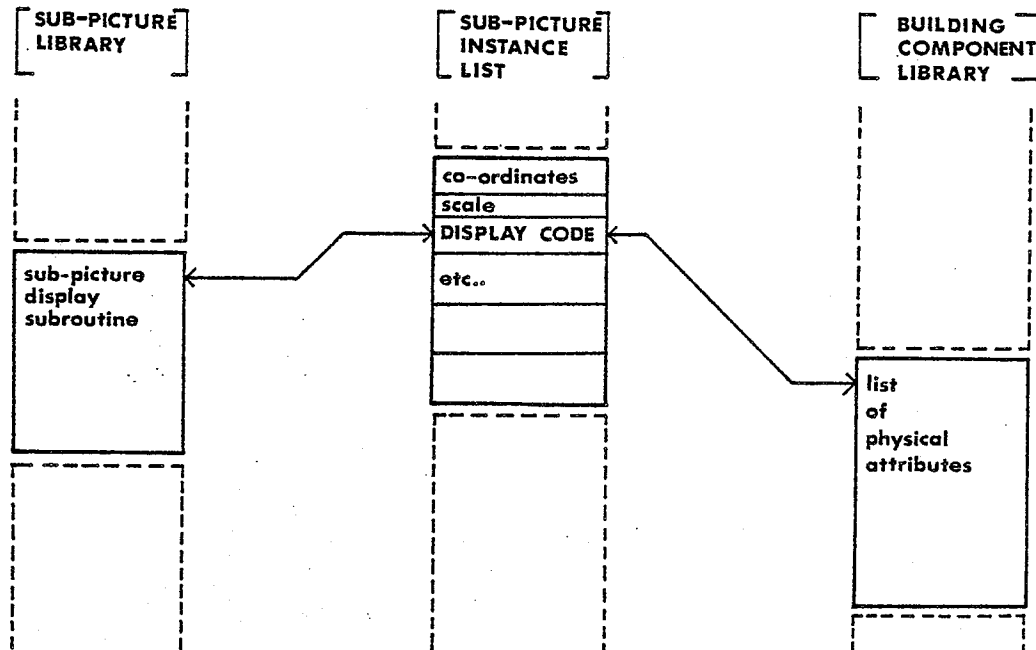


Figure 2.3 THE SUB-PICTURE DISPLAY CODE.

classes are:

LINEAR

LABEL

AREAL

The LINEAR class of sub-picture (simple straight line segments) geometrically locate individual surface boundaries separating adjacent containers. It may be used to define any n-sided container ($n > 3$) -see Figure 2.4(b)- or to subdivide existing

- 1) All containers shall be situated according to a uniform rectangular grid.
- 2) All surface boundaries shall be located along this grid.
- 3) All symbols depicting surface boundaries shall be permitted to be coincident and intersect any other surface boundary, but shall NOT be permitted to cross any other surface boundary.

Table 2.1 GRAPHIC CONVENTIONS.

spatial containers -see Figure 2.4(a). However, as no attribute information identifying the contents of the containers bounding the surface is implied by this class of sub-picture, it must be used in conjunction with the LABEL class of sub-picture which is provided for that purpose.

The LABEL class of sub-picture (eg. room names) geometrically locate attribute labels identifying the contents of a single container. This class of symbol is considered to be composed of two elements; a point element and an alphanumeric string element (see Figure 2.5). The point element defines a location in the plane which lies within the container being identified. The alphanumeric string categorizes the physical characteristics of the container. This class of sub-picture must be used to define the contents of

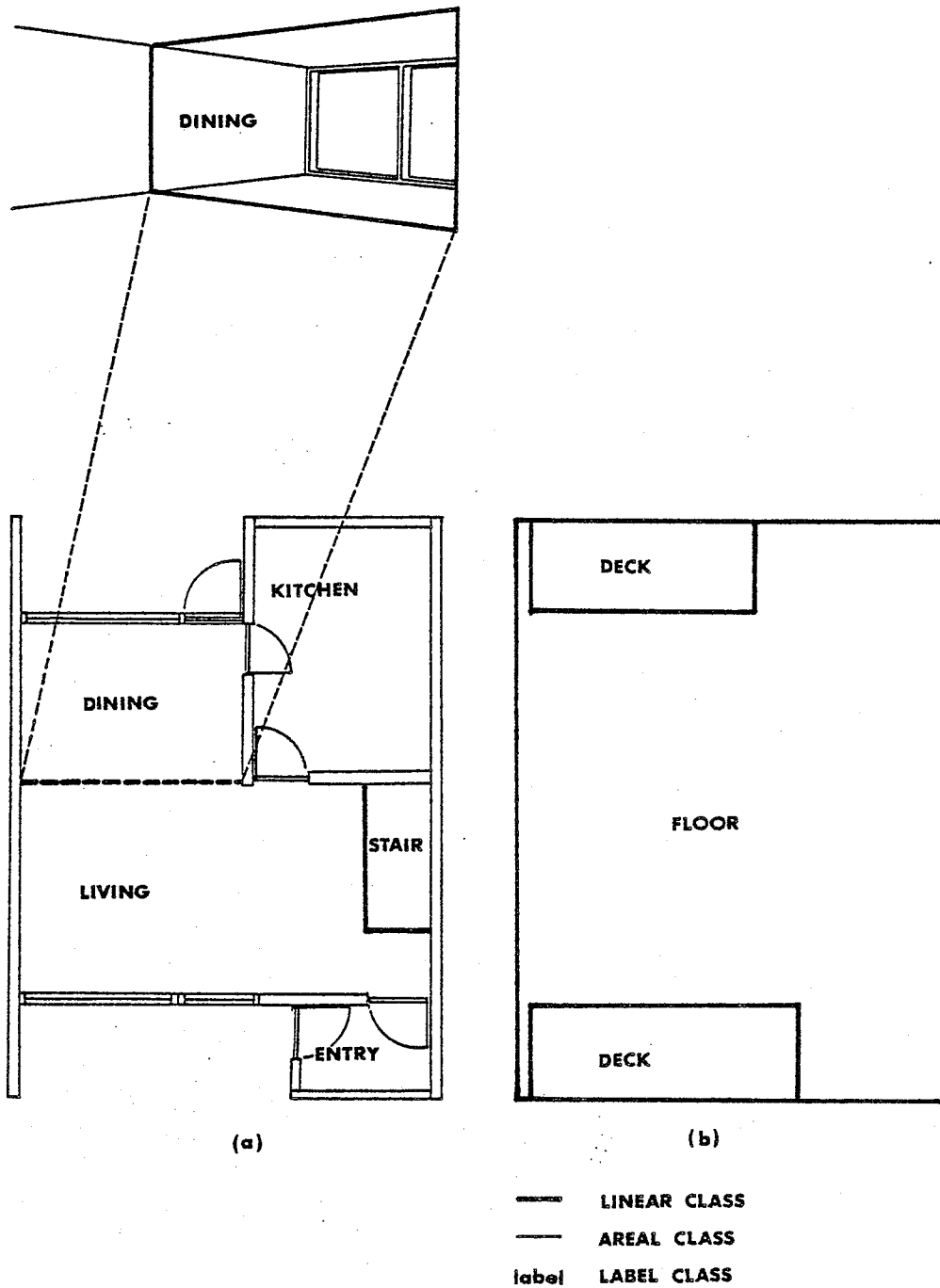


Figure 2.4 DRAWING SYMBOLS: LINEAR CLASS.

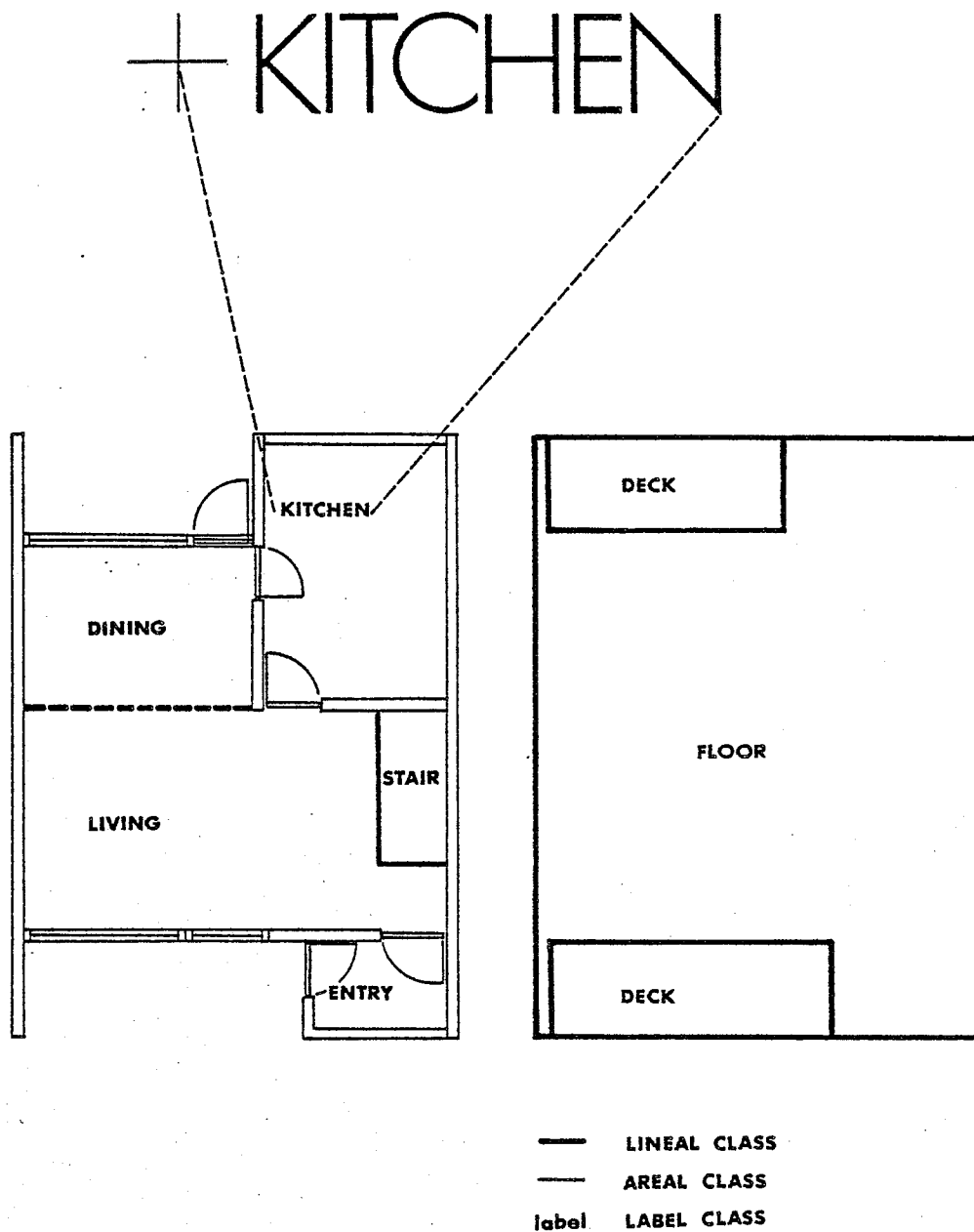


Figure 2.5 DRAWING SYMBOLS: LABEL CLASS.

any n-sided spatial or material container that has been geometrically defined in whole or in part with LINEAL class sub-pictures.

The AREAL class of sub-picture (eg. wall or window component symbol) defines both the surface boundaries and the attribute

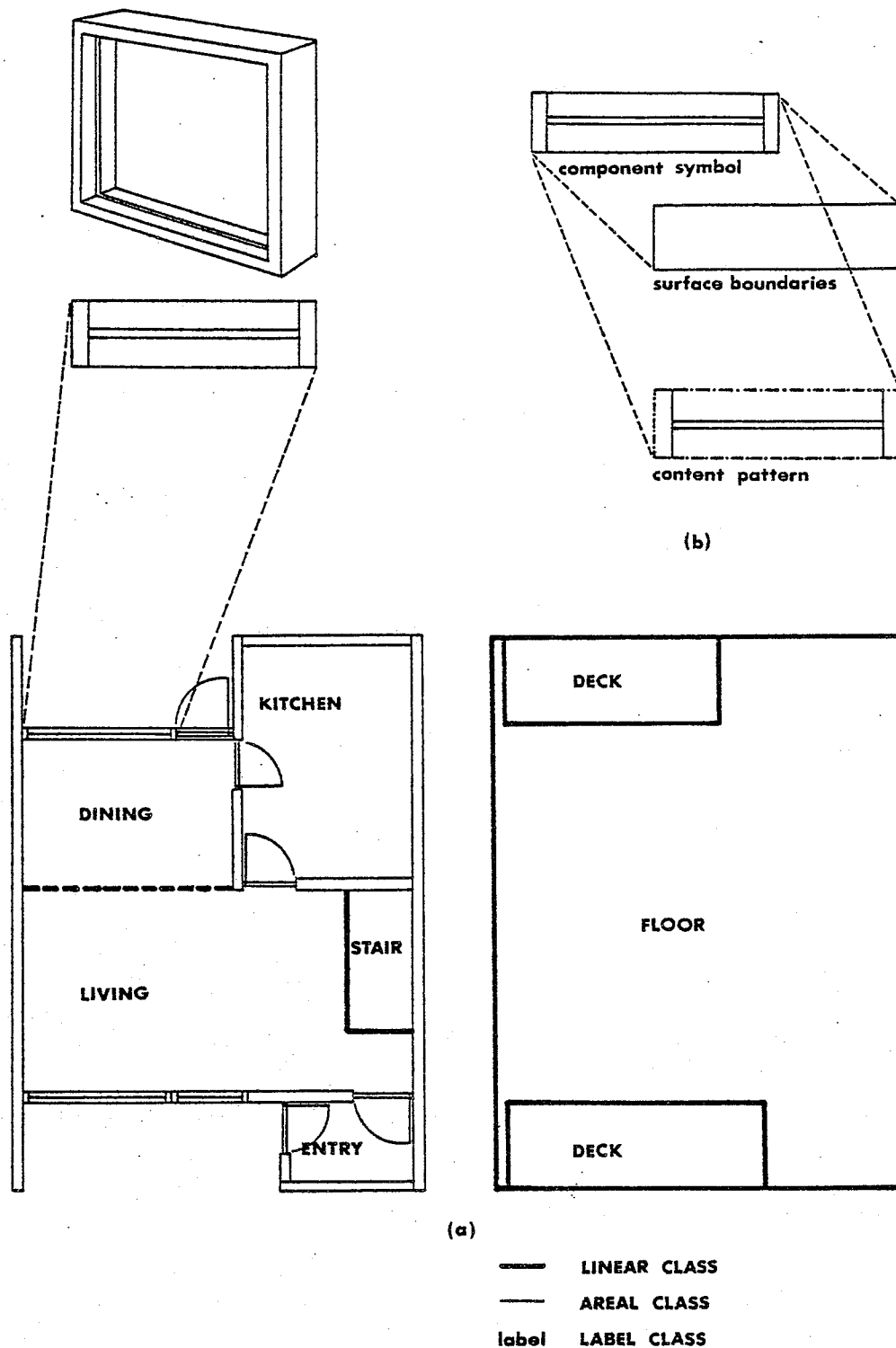


Figure 2.6 DRAWING SYMBOLS: AREAL CLASS.

information necessary to identify a complete 'material container'. As depicted in Figure 2.6(b), each sub-picture is deemed to be composed of a set of four straight line segments, forming a rectangle representing the surface boundaries of the container and a set of n straight line segments, patterned within the container, representing the contents of the container.

The Secondary Component Sub-set

The SECONDARY COMPONENT sub-set (see Figure 2.2) is used to store information describing the vertical sub-division of containers depicted in a building plan section drawing. Using the data items of this sub-set it is possible to locate and identify building components within a container other than the primary building component depicted in the drawing display. For study purposes, these secondary components are permitted to be defined either above and/or below the primary building component (Figure 2.7). By definition, this sub-set may only be used in conjunction with AREAL and LABEL sub-picture and not the LINEAR class of sub-pictures.

Each Secondary Component is identified and geometrically located by a sub-picture code and a height value. The sub-picture code identifies the type of building component. The height value identifies the difference in vertical elevation between the

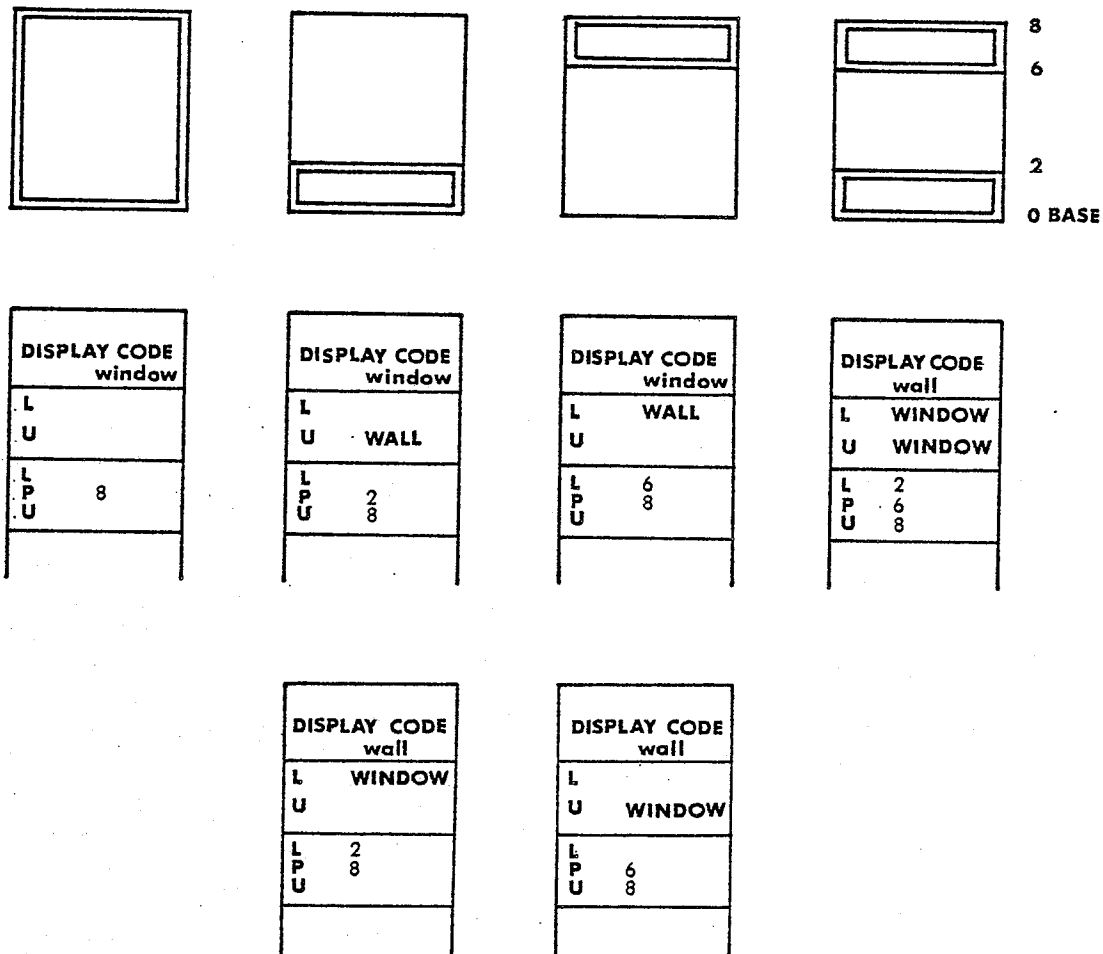


Figure 2.7 THE SECONDARY COMPONENT SUB-SET.

uppermost extreme of each component and the base height established for the plan section.

The LIBRARY Sub-set

The LIBRARY sub-set is designed to overcome some of the descriptive rigidity imposed by the Building Component Library. It should be used in conjunction with a file known as the Instance

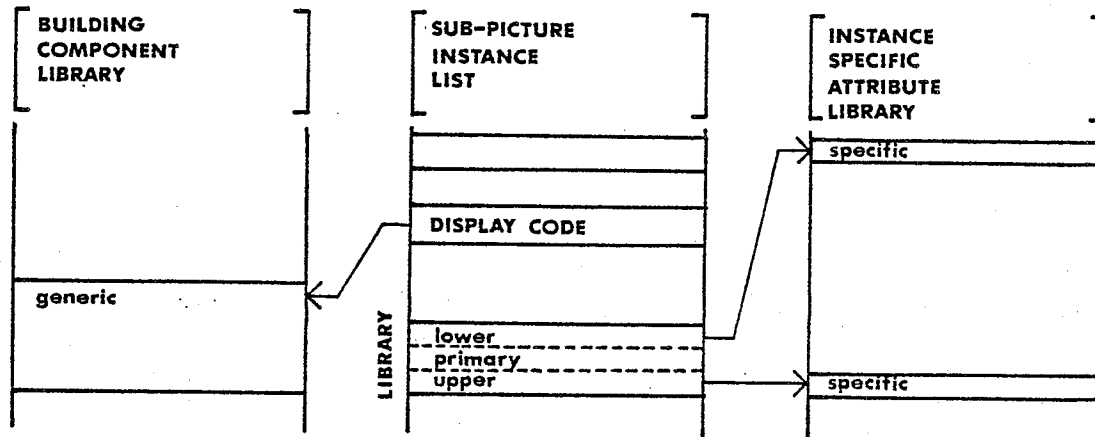


Figure 2.8 THE LIBRARY SUB-SET.

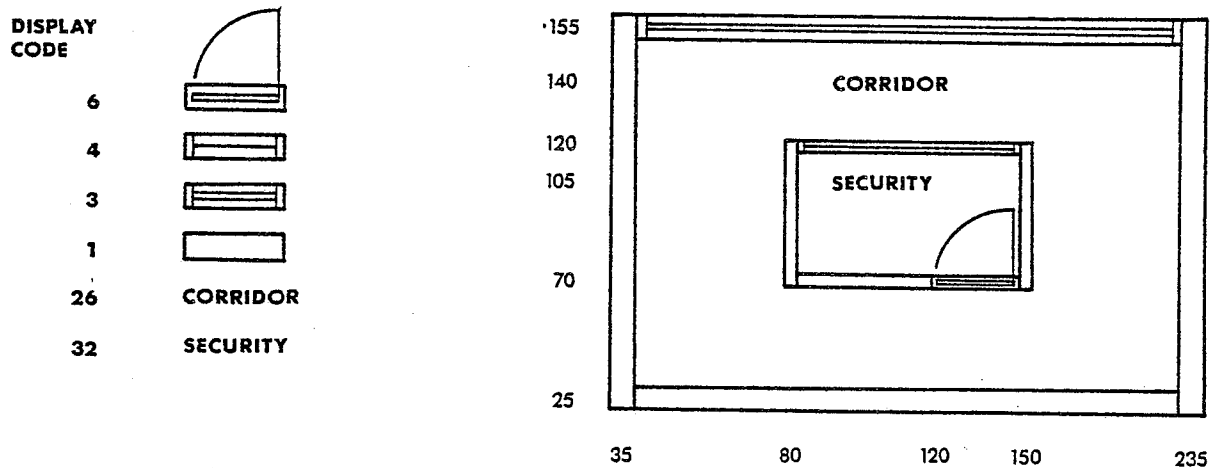
Specific Attribute Library. Using both this sub-set and the Instance Specific Attribute Library, it is possible to assign attributes to a specific container instance that supercede those defined for the parent component in the Building Component Library (Figure 2.8). The data items of this sub-set (one for each of the three possible components permitted to be located in a container) are used to point to user defined entries in the Instance Specific Attribute Library, should such entries be made. Like the Secondary Component sub-element, this sub-set can only be employed in conjunction with the AREAL and LABEL classes of sub-picture.

A GRAPHIX Graphical Data Structure is presented in Figure 2.9 to conclude this section. Although the plan section drawing is

October 1979

EXPLICIT METHODS

simple, its' representation does illustrate the use of the data structure.



GRAPHIX GRAPHICAL DATA STRUCTURE																
	OBJECT	NEXT	PREVIOUS	GRAPHIC SUBSET					NON-GRAPHIC SUBSET							
				PRIMARY COMPONENT					SECONDARY COMPONENT					LIBRARY		
				CO-ORDINATES				SCALE	DISPLAY CODE	SUB-PICTURE CODE		HEIGHT				
										display code						
				x1	y1	x2	y2			lower	upper	lower	prime	upper	l	p u
graphic symbols	0	1	14	35	25	35	155	10	1	0	0	0	0	0	0	0
	1	2	0	45	25	225	25	10	1	0	0	0	0	0	0	0
	2	4	1	80	70	80	120	5	1	0	0	0	0	0	0	0
	3	5	15													
	4	9	2	45	155	225	155	10	3	0	0	0	0	0	0	0
	5	7	3													
	6	12	9	235	25	235	155	10	1	1	0	2	8	0	0	0
	7	8	5													
	8	10	7													
	9	6	4	85	120	145	120	5	4	0	0	0	0	0	0	0
	10	11	8													
	11	15	10													
	12	13	6	150	120	150	70	5	1	0	0	0	0	0	0	0
	13	14	12	145	70	125	70	5	6	0	1	0	7	8	0	0
	14	0	13	120	70	85	70	5	1	0	0	0	0	0	0	0
	15	3	11													
textual symbols	16	18	18	90	105	0	0	0	32	0	0	0	0	0	0	0
	17	19	23													
	18	16	16	90	140	0	0	0	26	0	0	0	0	0	0	0
	19	20	17													
	20	21	19													
	21	22	20													
	22	23	21													
	23	17	22													

Figure 2.9 A GRAPHIX GRAPHICAL DATA STRUCTURE.

Section 2.2 Assembling and Enumerating the Planar Graph

Introduction

Once the plan section drawings are input, each drawing must be converted from its GRAPHIX Graphical Data Structure representation into the Planar Graph Representation before the construction of the three dimensional description for the building can be completed. This section documents both the algorithms developed to assemble the Planar Graph Representation for each plan section drawing and the extended PDV(4)* data structure developed to store the description of the drawings.

Five algorithms are used to perform the GRAPHIX Graphical Data Structure to Planar Graph Representation conversion. The function of each is summarized below. Detailed operational descriptions for the algorithms follow later. The description for the extended PDV data structure used to store the Planar Graph Representation of each plan section drawing is incorporated within the description of the algorithms generating the information base.

Procedural Summary

The two dimensional Planar Graph Representation of each plan section drawing is constructed using the following algorithm sequence:

* This acronym is derived from the terms Point, Degree and Vector.

Algorithm 2.1 constructs a planar graph description for a plan section drawing. The GRAPHIX Graphical Data Structure describing the plan section drawing is inspected and a planar graph G is constructed in which: the vertices are isomorphic to the end points of the vertical surface boundaries described in the drawing; the edges are isomorphic to the vertical surface boundaries described in the drawing; the regions are isomorphic to the containers described in the drawing; and, the circuits circumscribing each region of the graph are isomorphic to the vertical surface boundaries of each container described in the drawing.

Algorithm 2.2 identifies each region (container) of the graph G and enumerates the circuits circumscribing each region of the graph. The basic PDV description of the graph G (constructed in Algorithm 2.1) is inspected and the circuits circumscribing the exterior region and all circuits circumscribing the interior regions of each Part of the graph G are enumerated.

Algorithm 2.3 identifies the surface boundaries (the circuit or circuits) that vertically circumscribe each container (region). The Part and circumscribing circuit information generated in the previous algorithm is inspected and the circuit(s) circumscribing each region of the graph are labelled as belonging to that region.

Algorithm 2.4 Having assembled a topological and geometric description of the surface boundaries of each container, this

algorithm proceeds to attach the attribute information identifying the contents of each container to each region and its constituent graph elements (its circumscribing circuits and edges). This is accomplished by inspecting each sub-picture instance contained in the GRAPHIX Graphical Data Structure and attaching to each region (and its circumscribing circuits and edges) of the graph G , a pointer to the AREAL or LABEL sub-picture instance which identifies the contents of that region (container).

Algorithm 2.5 assembles a description of the graph G in which all the containers adjacent to each and every container depicted in the plan section drawing are identified. The circumscribing circuits of each region are traversed in turn and a list of all adjacent regions is constructed. This description is stored as a dual graph representation in which the vertices of the dual graph $G(d)$ represent the regions of the graph G , and the edges of $G(d)$ represent the adjacency of the regions of the graph G .

The Algorithms:

Algorithm 2.1 Construct the planar graph G

TO CONSTRUCT A PLANAR GRAPH DESCRIPTION REPRESENTING THE SURFACE BOUNDARIES DEPICTED IN A PLAN SECTION DRAWING AND DESCRIBED IN THE GRAPHIX GRAPHICAL DATA STRUCTURE.

GIVEN: 1) A planar graph G consisting of a finite edge set E of q edge elements, each element e of E being defined by an

unordered pair of vertices $v(a)$, $v(b)$ such that

- a) each pair of vertices for e defines a distinct position in a cartesian plane, and
- b) the geometric realization of each edge element e delineates a straight line segment joining its vertices.

2) A surface boundary input list S is defined, where each element s of S is defined by an unordered pair of cartesian coordinates $c(a)$, $c(b)$, such that

- a) each pair of cartesian coordinates for s defines a straight line segment contiguous to a surface boundary described in the GRAPHIX Graphical Data Structure, and
- b) each and every surface boundary defined for each sub-picture is defined once and only once in the surface boundary input list S .

and ASSUMING a labeling convention for the entry of discrete boundary edges from S into E , such that each edge $e(i)$ is labeled with a successive odd positive integer value greater than or equal to 1, but such that i at no time exceeds $2q$.

STEP 1: ASSEMBLE EDGE DESCRIPTION FOR THE PLANAR GRAPH.

1.1) Add, in turn, surface boundary elements from S to the edge list E describing the planar graph G , examining each edge before it's addition to the list to ensure that the current edge:

- A) does not define a loop: the two vertices for the current

edge do not occupy the same position in the plane;

B) does not cross over (to be distinguished from overlap) any other edge in the edge set;

and, that any other edge element in S partially or fully overlapping the current edge is;

C) in the case of complete overlap (Figure 2.10(a)) deleted from the surface boundary list S;

D) in the case of partial overlap (Figure 2.10(b)), parsed in such a manner as to remove the overlapping portion of the line segment (excluding the junction termini) from the list S; and,

E) in the case of intersection (Figure 2.10(c)) are parsed into two separate line segments about the loci of intersection.

Once all edges have been examined and parsed where necessary, the graph G can be embedded in the plane such that G is isomorphic to the geometric realization of G in the drawing.

STEP 2: ASSEMBLE PDV DESCRIPTION FOR THE PLANAR GRAPH.

2.0) Before the graph can be subjected to a maze-threading procedure to enumerate the surface boundaries and the regions of the graph, the information in the edge set E must be restructured. The description of the graph is to be stated in terms of;

- a) a set of labeled and unique vertices, and
- b) a set of directed edges describing adjacency between pairs of labeled vertices.

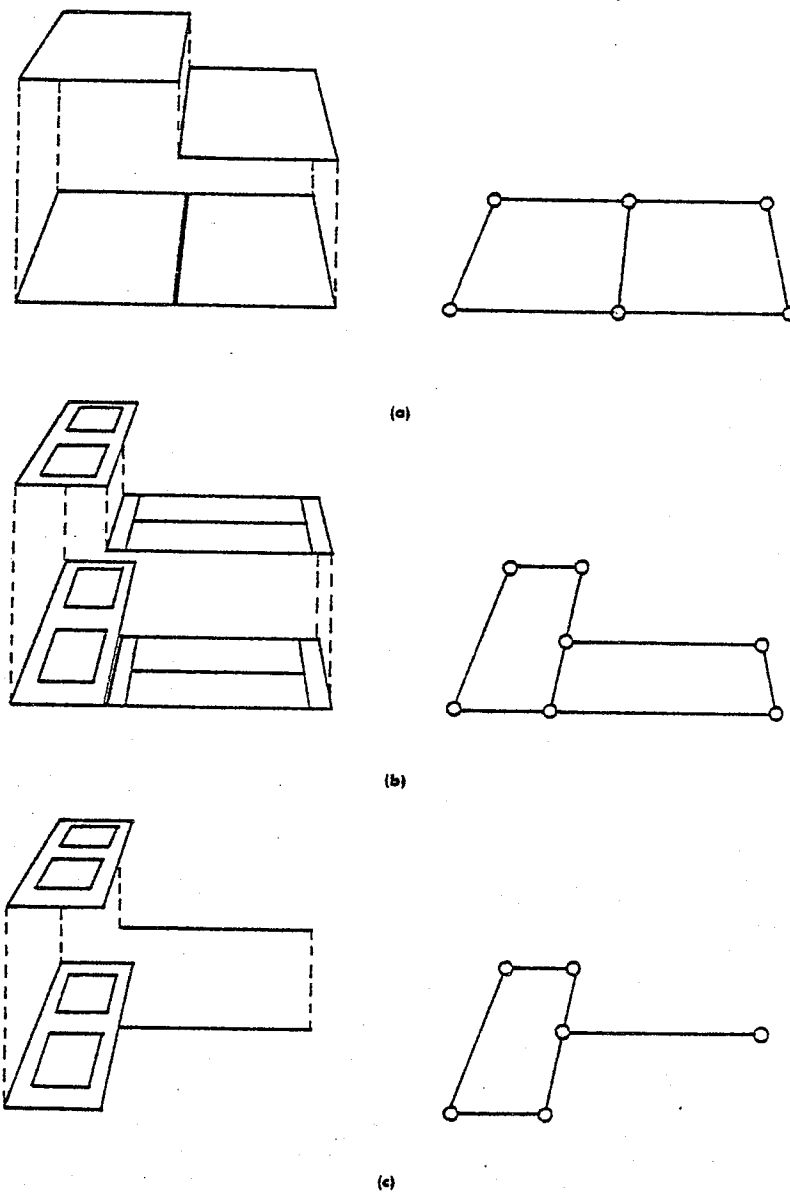


Figure 2.10 SURFACE BOUNDARY PARSING.

Re-organize the information in the edge set E , specific properties of the graph G and its geometric realization G' , and restructure that information into five lists, P , P' , D , V , and B , where;

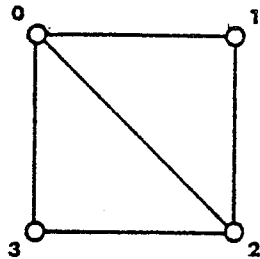
- 1) P defines, for each distinct vertex in the graph, the origin of it's vertex adjacency list in V;
- 2) P' defines the coordinate position for each vertex labelled in P;
- 3) D defines the degree of each vertex in P, the degree of a vertex reflecting the number of elements occupied by its vertex adjacency list in V;
- 4) V defines the adjacency of vertices in the graph (the edges); for each vertex in P, V provides a list of all vertices adjacent to that vertex;
- 5) B defines the bearing angle for each edge described from P to V, where the angle measured is that of the edge entrant to the vertex referenced in V;

A working list TAG and a counter L are used in the execution of the algorithm:

- 6) TAG, of length $2(q)$, provides work space for the identification of distinct vertices in the edge list E, and is empty; and,
- 7) L, a counter, is used to label distinct vertices.

2.1) To construct the TAG vector: select successive elements from the edge vector E for examination, naming the current edge the SUBJECT EDGE, $e(s)$, and creating a subset of E called the OBJECT VECTOR, $E(o)$, such that $E(o) = E - e(s)$.

- A) Examine each of the two vertices defining the subject edge; vertices occupying a common position in the plane are to be



	P	P'	D	V	B
0	0	0,1	3	1	90
1	3	1,1	2	2	135
2	5	1,0	3	3	180
3	8	0,0	2	2	180
4				0	270
5				3	270
6				0	315
7				1	360
8				0	360
9				2	90

Figure 2.11 THE BASIC P,D,V NOTATION SYSTEM.

assigned an identical vertex label (see Figure 2.12), and;

- 1) if the current subject vertex has no been assigned a label in TAG;

A) assign the next sequential value for label L to the vertex and enter the co-ordinate values for the vertex in P' at the position indicated by the label;
and

B) examine each vertex in the object vector and label, as above with L, vertices occupying a position in the plane identical to that of the current subject vertex.

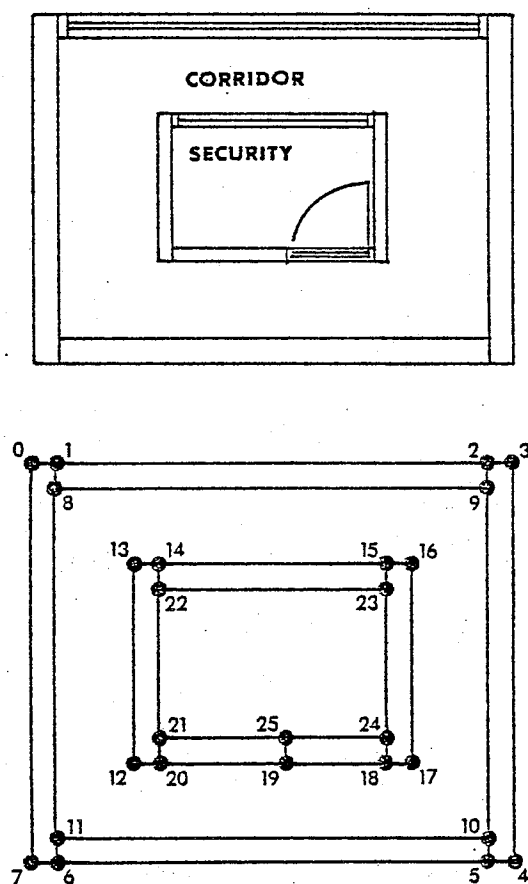
- 2) if the current vertex has been assigned a label in TAG;

then proceed to examine another subject vertex, until all elements of E have been exhausted.

2.2) Construct the P, D, V, and B vectors: upon completion of the TAG labelling routine, L shows the number of distinct vertices in the graph and TAG associates each vertex in E with one of the L distinct vertices. Using TAG to identify all instances of a given vertex in E, and the paired vertices of E to identify all vertices adjacent to a given vertex, scan successively through TAG in ascending order from 1 to L for each of the L distinct vertices of the graph as follows:

- A) Enter in P(i) the address of the next available storage location in V; this location in V is the start of the adjacency list for vertex i.
- B) Count the number of occurrences of vertex i in TAG, (the degree of vertex i) and for each occurrence:
 - 1) Enter into the next available location in V, the label of the vertex adjacent to the vertex i; compute the Bearing Angle of the edge from the vertex i to its adjacent vertex and enter it into the next available location of B.
- C) Upon conclusion of the search for vertex i, the degree of vertex i is entered into the next available element of D and the search procedure is initiated for the next search vertex.

The PDV data structure describing the surface boundaries depicted in building plan section is illustrated in Figure 2.12.



P	P'		D	V	S
0	30	160	2	1	90
1	40	160	3	7	180
2	230	160	3	2	90
3	240	160	2	8	180
4	240	20	2	0	270
5	230	20	3	3	90
6	40	20	3	9	180
7	30	20	2	1	270
8	40	150	3	4	180
9	230	150	3	2	270
10	230	30	3	5	270
11	40	30	3	3	360
12	77.5	67.5	2	6	270
13	77.5	122.5	2	10	360
14	82.5	122.5	3	4	90
15	147.5	122.5	3	7	270
16	152.5	122.5	2	11	360
17	152.5	67.5	2	5	90
18	147.5	67.5	3	0	360
19	122.5	67.5	3	6	90
20	82.5	67.5	3	9	90
21	82.5	72.5	3	11	180
22	82.5	117.5	3	1	360
23	147.5	117.5	3	8	270
24	147.5	72.5	3	2	360
25	122.5	72.5	3	10	180
26				9	360
27				5	180
28				11	270
29				10	90
30				6	180
31				8	360
32				13	360
33				20	90
34				12	180
35				14	90
36				13	270
37				15	90
38				22	180
39				14	270
40				16	90
41				23	180
42				15	270
43				17	180
44				16	360
45				18	270
46				17	90
47				19	270
48				24	360
49				18	90
50				20	270
51				25	360
52				19	90
53				12	270
54				21	360
55				20	180
56				22	360
57				25	90
58				21	180
59				14	360
60				23	90
61				22	270
62				15	360
63				24	180
64				23	360
65				18	180
66				25	270
67				24	90
68				19	180
69				21	270

Figure 2.12 A BASIC PDV DESCRIPTION OF A DRAWING.

Algorithm 2.2 Enumerate the surface boundaries of the graph

TO ENUMERATE FOR EACH PART OF A PLANAR GRAPH THOSE CIRCUITS
AND THEIR CONSTITUENT EDGE ELEMENTS THAT CIRCUMSCRIBE
FIRSTLY THE EXTERIOR REGION AND, SUBSEQUENTLY, ALL INTERIOR
REGIONS OF THAT GRAPH.

GIVEN: 1) a planar graph G , the edges of which represent the
vertical surface boundaries depicted in a plan section
drawing, which is described by the lists;

	P	D	V	B
P_i	n	7	d	225
			e	180
			f	135
			g	90
			a	0
			b	315
			c	270

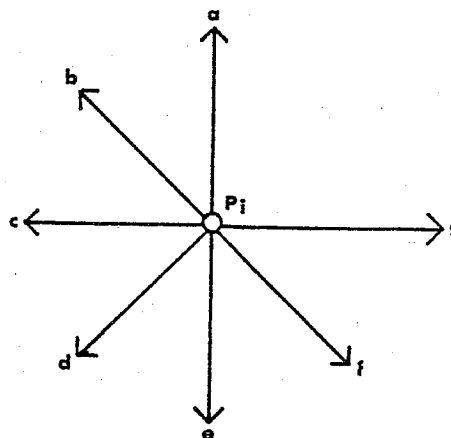


Figure 2.13 COUNTER-CLOCKWISE ORDER OF VERTICES.



P, whose elements label each vertex;

D, whose elements describe the degree for each vertex
labelled in P ;

V, whose elements identify those vertices adjacent to

each vertex labelled in P ; and

B , whose elements describe the bearing angle for each element in V , where for the list V , for each vertex of the graph the vertices adjacent to that vertex are listed in counter-clockwise order (see Figure 2.13).

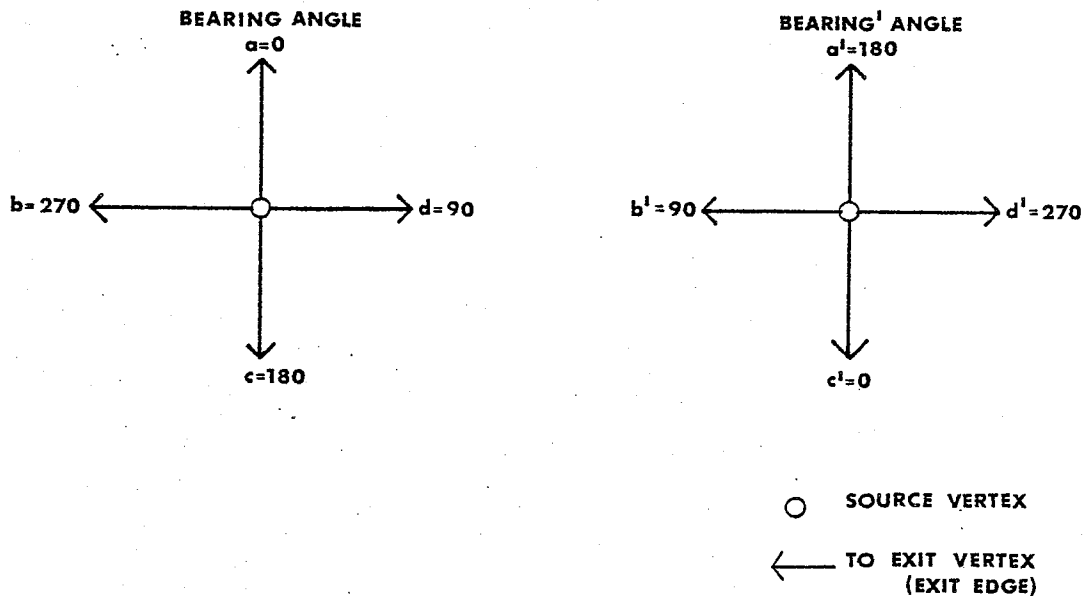


Figure 2.14 SOURCE BEARING' ANGLE.

and ASSUMING the following terminology:

- 1) SOURCE VERTEX refers to a vertex of the graph that is to be or is being exited;
- 2) EXIT VERTEX refers to a vertex of the graph that is to be or is being entered along an edge extending from the SOURCE VERTEX;
- 3) SOURCE BEARING' ANGLE refers to the inverse of the

Bearing Angle (as depicted in Figure 2.14) for the edge that was traversed in order to enter the current SOURCE VERTEX;

- 4) START VERTEX refers to the first vertex of a circuit circumscribing a region of the graph.
- 5) THE FIRST AVAILABLE RIGHT HAND TURN refers to that edge emanating from the current SOURCE VERTEX (all the edges emanating from the SOURCE VERTEX are identified via their vertex label in the list V) which;
 - a) has not been previously traversed, and
 - b) lies immediately adjacent, in a counter-clock-wise direction, to that edge (see Figure 2.15) which was used to enter the current SOURCE VERTEX.

The FIRST AVAILABLE RIGHT HAND TURN is identified from the set of available 'exit vertices' (ie. available exit edges) for the current SOURCE VERTEX by inspecting the Bearing Angle for each available exit vertex and selecting,

if the SOURCE BEARING' ANGLE $> 0^\circ$, that vertex with the largest Bearing Angle which is less than the current SOURCE BEARING' ANGLE or, if no such vertex can be identified or,

if the SOURCE BEARING' ANGLE $= 0^\circ$, that vertex which has the largest Bearing Angle.

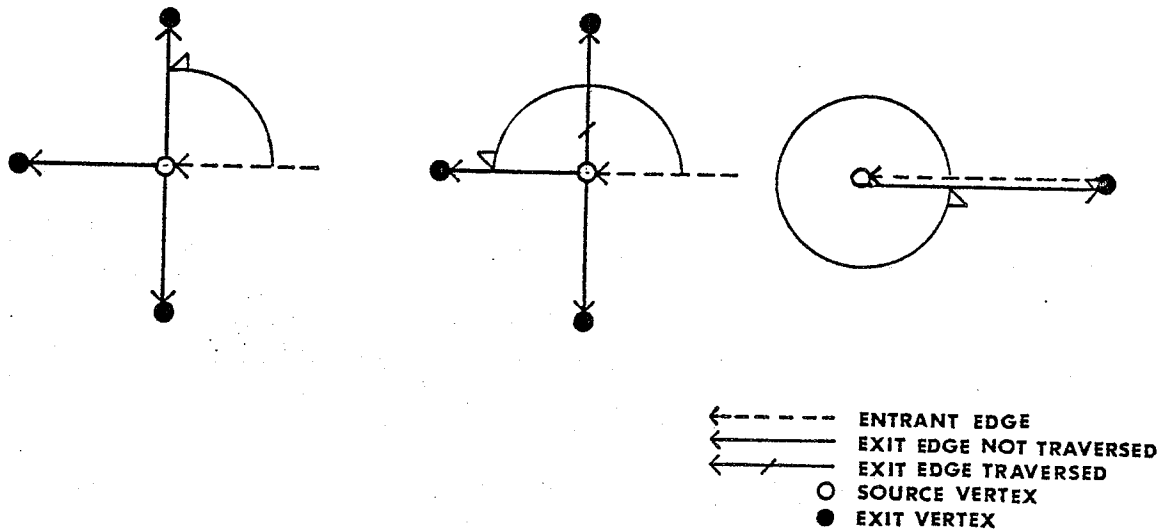


Figure 2.15 THE FIRST AVAILABLE RIGHT HAND TURN.

- 6) BOUNDARY CIRCUIT refers to a circuit of the graph that circumscribes a region of the graph; and,
- 7) EXTERIOR BOUNDARY CIRCUIT and INTERIOR BOUNDARY CIRCUITS geometrically identify that circumscribing circuit which circumscribes the region outside a Part of the graph and those circuits (or circuit) which circumscribe the regions inside a Part of the graph, respectively.

and DEFINING the lists VTAG, VTRAV, VCLR, PPL, CL and PL, such that:

- 1) VTAG defines, for each corresponding EXIT VERTEX in V,

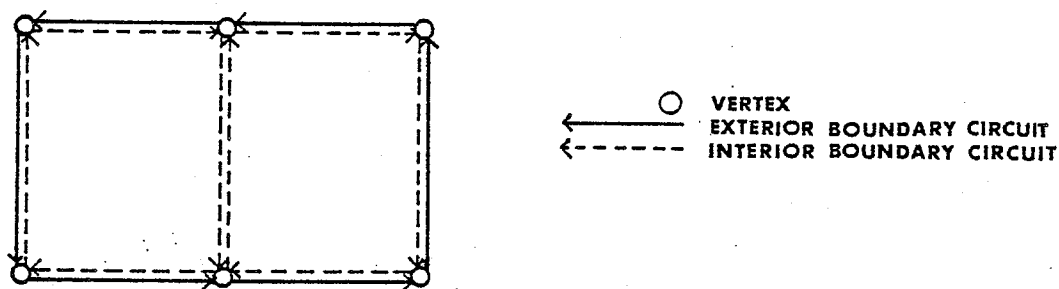


Figure 2.16 THE CIRCUMSCRIBING CIRCUITS OF A GRAPH.

- the SOURCE VERTEX in P for that EXIT VERTEX;
- 2) VTRAV defines, for each corresponding EXIT VERTEX in V and the entrant edge implied by that vertex entry in V , whether or not that entrant edge has been marked as being traversed in the enumeration process;
 - 3) VCLR defines, for each corresponding EXIT VERTEX in V and the entrant edge implied by that vertex entry in V , the boundary circuit to which that edge belongs;
 - 4) PPL defines, for each vertex of the graph, the Part of the graph to which that vertex belongs;
 - 5) CL indicates which circuit labels have been assigned and for those that have been assigned, to identify an EXIT VERTEX in the list V and, by implication, the entrant

edge to that vertex that lies along and partially comprises that boundary circuit; and,

- 6) PL indicates which Part labels have been assigned and, for those that have been assigned, to identify the circuit labels assigned to the first boundary circuit (exterior circuit) of that Part of the graph.

STEP 1: LOCATE A START VERTEX LYING ALONG THE EXTERIOR BOUNDARY CIRCUIT OF A PART OF THE GRAPH.

1.1) Search the elements in the coordinate list P' associated with each vertex of the graph, examining the x coordinate for those vertices that have not been assigned a Part Label (as indicated in the PL list), and attempt to nominate that vertex with the largest x coordinate value as the START VERTEX.

- A) If no such START VERTEX is identified then either all the circuits of all Parts of the graph have been enumerated or the graph is empty. In either case the enumeration procedure has been completed and this algorithm should be exited.
- B) If such a START VERTEX is identified, that START VERTEX lying along the rightmost portion of the exterior boundary circuit of an as yet non-enumerated Part of the graph, then proceed to Step 2.0.

STEP 2: ENTER A PART OF THE GRAPH AND INITIATE A TRAVERSE OF ITS EXTERIOR BOUNDARY CIRCUIT.

2.1) Select the next available Part Label j from the Part Label

list PL, nominating this label as the current PART LABEL; mark that Part Label in PL as being assigned, entering the vertex label i , identifying the START VERTEX in P, into the j th element of PL; and, enter the Part Label j , that will be used to label all vertices of this Part, into that element of the Vertex Part list PPL that is associated with the START VERTEX.

2.2) Define a psuedo edge, having a Bearing Angle of 270° , entrant to the START VERTEX. This psuedo edge will be used to enter this Part of the graph but does not formally belong to the graph (see Figure 2.17); record the START VERTEX label as the current SOURCE VERTEX; enter the START VERTEX traversing along the psuedo edge and record the SOURCE BEARING' ANGLE for the psuedo edge used to enter the START VERTEX (ie.. $B=270^\circ$ and $B'=90^\circ$).

2.3) Set the CIRCUIT LABEL REQUEST FLAG, which will cause the next Circuit Label to be fetched from the Circuit Label list in Step 3.

STEP 3: TRAVERSE AND ENUMERATE THE EXTERIOR AND THEN THE INTERIOR BOUNDARY CIRCUITS OF THIS PART OF THE GRAPH.

3.1) Examine the exit vertices in V, for the current SOURCE VERTEX, that are marked as not having been previously traversed. Attempt to identify THE FIRST AVAILABLE RIGHT HAND TURN, recording its vertex label b (in the b th element of V) as the EXIT VERTEX and;

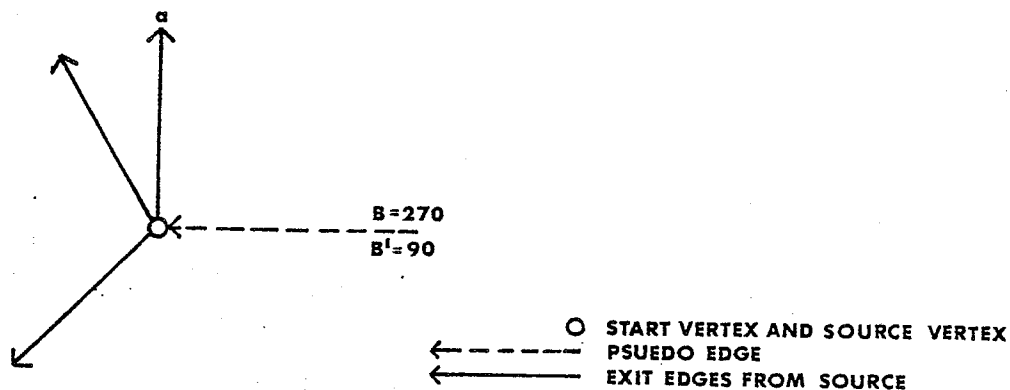


Figure 2.17 BEGIN TRAVERSE OF EXTERIOR BOUNDARY CIRCUIT.

A) If such a RIGHT HAND TURN is identified:

- 1) Determine if the CIRCUIT LABEL REQUEST FLAG is set, and;
 - a) If not set, then proceed to A.2 of this step.
 - b) If set, then proceed to select the next available Circuit Label m from the Circuit Label list CL , nominating this label as the current CIRCUIT LABEL; mark this Circuit Label in CL as being assigned, entering the element address k for the EXIT VERTEX b into the m th element of CL ; and, clear the CIRCUIT REQUEST FLAG.
- 2) Enter the current CIRCUIT LABEL into the m th element of $VCLR$, so as to identify the boundary circuit to which

this edge belongs and mark this edge as having been traversed.

- 3) Set the m th element in VTRAV, marking this edge as having been traversed.
 - 4) Assign the current PART LABEL to the b th element in PPL, associating this vertex with the current Part.
 - 5) Nominate the EXIT VERTEX as the current SOURCE VERTEX and determine the SOURCE BEARING' ANGLE for the new SOURCE VERTEX.
 - 6) Compare the START VERTEX for the boundary circuit currently being traversed with the current SOURCE VERTEX, and;
 - a) if not equal, return to 3.1 of this step in order to continue the traverse of this boundary circuit.
 - b) if the START VERTEX and SOURCE VERTEX are one in the same (see Figure 2.18), then the traverse of the current boundary circuit has been completed; set the CIRCUIT LABEL REQUEST FLAG; and, return to 3.1 of this step in order to seek out another boundary circuit, if it exists, in this Part of the graph.
- B) If such a RIGHT HAND TURN is not identified: Examine successive vertices in P until that list is exhausted and, for every vertex in PPL that is labelled as belonging to the current Part of the graph (ie. PPL label and PART LABEL are one in the same) perform the following;
- 1) Define the vertex selected above as the SOURCE VERTEX

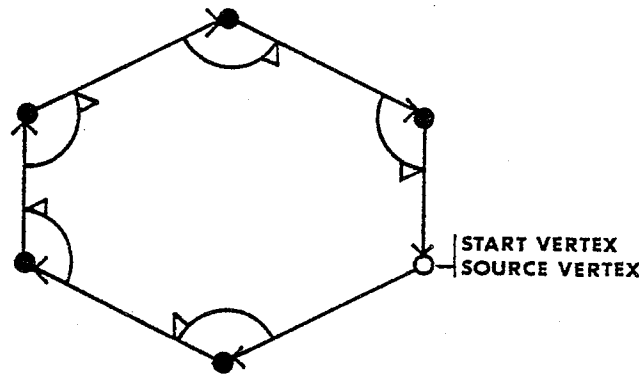


Figure 2.18 COMPLETION OF BOUNDARY CIRCUIT TRAVERSE.

and attempt to identify an available (ie. non-traversed) exit vertex in V for the current SOURCE VERTEX and;

- a) If such an exit vertex is identified, define this vertex as the current EXIT VERTEX; nominate the current SOURCE VERTEX as the START VERTEX; set the CIRCUIT LABEL REQUEST FLAG; and return to A.1 of this step in order to traverse this edge and the boundary circuit to which it belongs.
- b) If no such exit vertex is identified, then all the edges in this Part of the graph have been traversed and all the boundary circuits enumerated (see Figure 2.19). As there may be other Parts of the graph yet to be located and traversed, return to 1.1 of step 1 to continue the traverse and enumeration process for this graph.

Leaf blank to correct
numbering

EXPLICIT METHODS

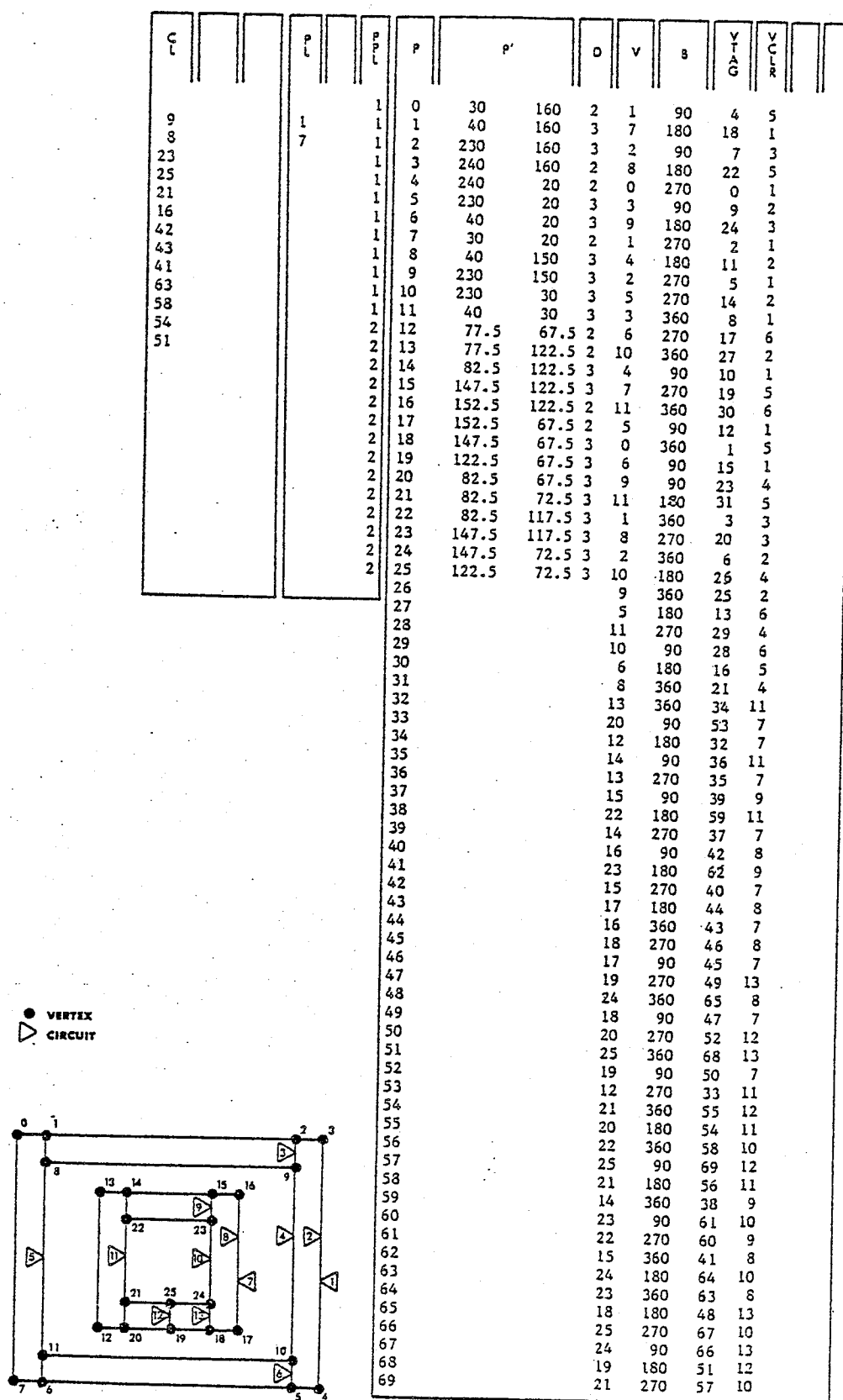


Figure 2.19 THE BOUNDARY CIRCUITS OF THE GRAPH.

Algorithm 2.3 Enumerate the regions of the graph

TO ENUMERATE THE BOUNDARY CIRCUITS CIRCUMSCRIBING EACH REGION OF THE GRAPH.

GIVEN: 1) The extended PDV description for the planar graph, as developed in the previous algorithm, in which each edge is assigned a Part and Circuit label,

and, ASSUMING the following terms and enumerative conventions, such that:

1) The term PARENT is used to refer to that Part of the graph within which another Part of the graph is directly embedded. (eg. see Figure 2.20, where Part 1 is the PARENT Part of Part 2 and Part 2 is the PARENT Part of Part 3 but Part 1 is not the PARENT Part of Part 3);

2) The task of determining whether or not a point lies within any simple region of the graph can be performed by summing the angles (as depicted in Figure 2.21) subtended between a point (referred to as the TARGET) and successive vertices lying along the boundary circuit of the region in question. If the point lies within the region (Figure 2.21(a)) the sum of these angles will equal 360 degrees. On the other hand, if the point lies outside the region in question (Figure 2.21(b)), the sum of these angles will equal 0 degrees.

and, DEFINING the lists PTAG, NP, R, RC, CR and VR, such that:

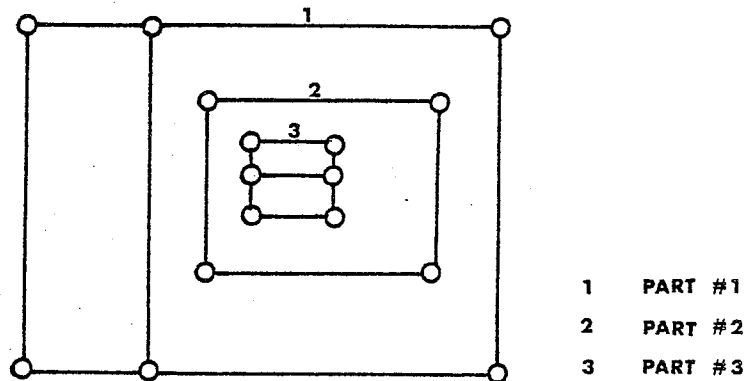


Figure 2.20 PARENT PARTS OF THE GRAPH.

- 1) PTAG defines, for each Part of the graph, whether the Part i is available ($PTAG(i)=0$) or not available ($PTAG(i)=1$) for examination/selection;
- 2) NP defines, for each Part of the graph, whether that Part has a PARENT Part, and if so, the Part Label used to reference that PARENT Part;
- 3) R defines, for each region label, whether that label has been or has not been assigned and if assigned, the number of boundary circuits circumscribing that region;
- 4) RC defines, for each region of the graph, the circuit label(s) used to identify the boundary circuit(s) of that region. If the region is simple (ie. circumscribed

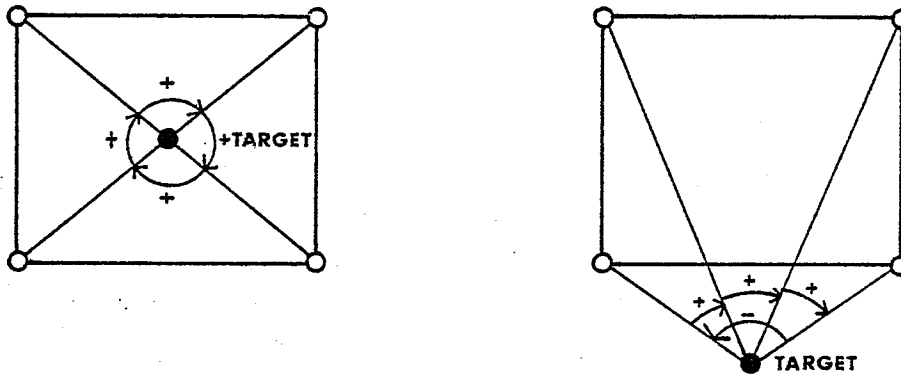
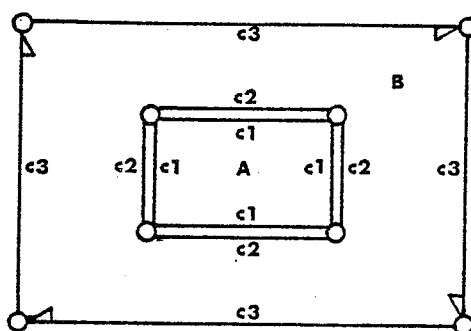


Figure 2.21 POINT LYING IN A REGION OF THE GRAPH.

by only one boundary circuit) the region will have only one entry in this list. However, if the region is complex (ie. circumscribed by more than one boundary circuit) the first entry in the list for that region will identify the exterior boundary circuit for the region;

- 5) CR defines, for each circuit label, the region to which that boundary circuit belongs; and,
- 6) VR defines, for each edge of the graph, the region to which that edge belongs.



	R	RC
region A	C1	
region B	C3,C2	

Figure 2.22 THE BOUNDARY CIRCUIT(S) OF A REGION.

STEP 1: IDENTIFY ALL NESTINGS OF PARTS OF THE GRAPH.

1.1) Select in turn each Part of the graph, as identified in PL and nominate the selected Part as the current SOURCE Part. If all the Parts of the graph have been previously selected, proceed to step 2.

1.2) Obtain the x and y coordinates for a vertex lying along the exterior boundary circuit of the SOURCE Part and define the coordinates of this vertex as the current TARGET.

1.3) Successively select Parts from those remaining in PL (all Parts of the graph except for the current SOURCE Part) and, nominating the Part selected as the current OBJECT part, determine if the TARGET lies within the exterior boundary

circuit for the current OBJECT Part.

A) if the TARGET does not lie within the exterior boundary circuit of the current OBJECT Part (ie. $\text{ANGLE}=0$) then the SOURCE Part is not nested within the current OBJECT Part. Return to 1.3 in order to examine the next Part of the graph.

B) If the TARGET does lie within the exterior boundary circuit of the current OBJECT Part (ie. $\text{ANGLE}=360$), then the current SOURCE Part is nested within the current OBJECT Part. Enter the Part label identifying the SOURCE Part into that element of NP, associated with the current OBJECT Part, and then return to 1.3 in order to select another Part of the graph for examination.

STEP 2: HIERARCHIALLY RELATE THE NESTED PARTS OF THE GRAPH.

2.1) Examine the entries for each Part element in NP in turn and PTAG that element as being not available (ie. for Part i $\text{PTAG}(i)=1$), if there is no PARENT identified in NP for that Part element.

2.2) Examine in turn the entries for each Part element in NP and if the PARENT list for each Part is either

empty, indicating that this Part element is embedded in the exterior region of the graph, or
contains only one PARENT entry, indicating that this part is nested directly within an interior region of the PARENT Part,

then proceed to step 3; all the parts of the graph have been hierarchically related. Otherwise, continue processing in this step, proceeding to 2.3.

2.3) Examine the entries in NP for each Part element in succession and for each part that is marked in the PTAG list as being available (ie. PTAG=0) and that has only one PARENT perform the following;

- A) Nominate the single PARENT Part selected as the current OBJECT.
- B) Nominate the PARENT of the current OBJECT as the TARGET.
- C) Examine in succession the entries in NP for each Part element that has more than one PARENT and remove any occurrences of the TARGET from that Parts NP list.
- D) PTAG the current OBJECT as being not available.

Upon completion return to 2.2 in order to determine whether or not all Parts of the graph has been directly related to a single Parent or the exterior region of the graph.

STEP 3: ENUMERATE THE BOUNDARY CIRCUITS CIRCUMSCRIBING EACH
 REGION OF THE GRAPH.

3.1) Enumerate the boundary circuit(s) circumscribing the exterior region of the graph: initialize the PTAG and R lists as being 'available'; select the next available region label (eg. the first region label being 1) from R and nominate this label as the current REGION NAME; search NP, identifying and counting each Parentless Part of graph; enter the number of Parentless

Parts into the R for the current REGION NAME, marking that label as being assigned; enter the circuit labels, identifying the exterior boundary circuit for each Parentless part into the element(s) of RC associated with the current REGION NAME; enter the REGION NAME into the elements of the CR and VR associated with the exterior boundary circuit for the Parentless Parts of the graph; and, PTAG the Parentless parts of the graph as being 'not available' (ie. PTAG (i)=1).

3.2) Enumerate the boundary circuits circumscribing those regions, other than the exterior region, within which the Parts of the graph are embedded: select in turn any available nested Part from NP (ie. PTAG=0), and nominate the Part selected as the current OBJECT Part and;

A) Identify the boundary circuit of the Parent Part (using the angle summation algorithm) in which the current OBJECT Part is embedded: enter the circuit label, identifying this circuit of the PARENT Part, into that element of NPC associated with the current OBJECT Part; select the next available region label from R and nominate this label as the current REGION NAME; initialize the circuit counter and count the boundary circuits belonging to the current REGION NAME, as they are identified; enter the circuit labels, for both the boundary circuit of the PARENT Part and then the exterior boundary circuit of the current OBJECT Part into RC for the current REGION Name; enter the current REGION NAME into the CR and VR elements associated with the boundary

circuit of the PARENT Part and the exterior boundary circuit of the current OBJECT Part; PTAG the OBJECT Part as being 'not available'; and,

B) Examine NP and determine if any other available Parts of the graph are embedded within the current PARENT Part;

1) If another Part or Parts is embedded within the current PARENT PART, then nominate each such Part in turn as the current OBJECT Part and perform the following: enter the circuit label identifying the exterior boundary circuit for the current OBJECT Part into the RC element associated with that Part of the graph; increment the circuit counter; enter the current REGION NAME into the CR and VR elements associated with the exterior boundary circuit of the current OBJECT Part; PTAG the current OBJECT Part as being 'not available'; and, proceed to 1 to examine the next available nested Part of the graph, if it exists; otherwise proceed to 2.

2) If no other Part of the graph is embedded within the current PARENT Part, then perform the following; enter the circuit counter into R for the current REGION NAME; and, return to 3.2 in order to enumerate another complex region of the graph, if it exists. Otherwise, proceed to 3.3 of this step.

3.3) Enumerate the boundary circuits circumscribing all remaining simple regions of the graph: examine VCLR and for each circuit not as yet assigned a region label in VR, perform the

following; initialize and increment the circuit counter; select the next available region label from R and nominate this label as the current REGION NAME; enter the circuit counter into R for the current REGION NAME; and, enter the REGION NAME into the CR and VR elements associated with the current unassigned boundary circuit.

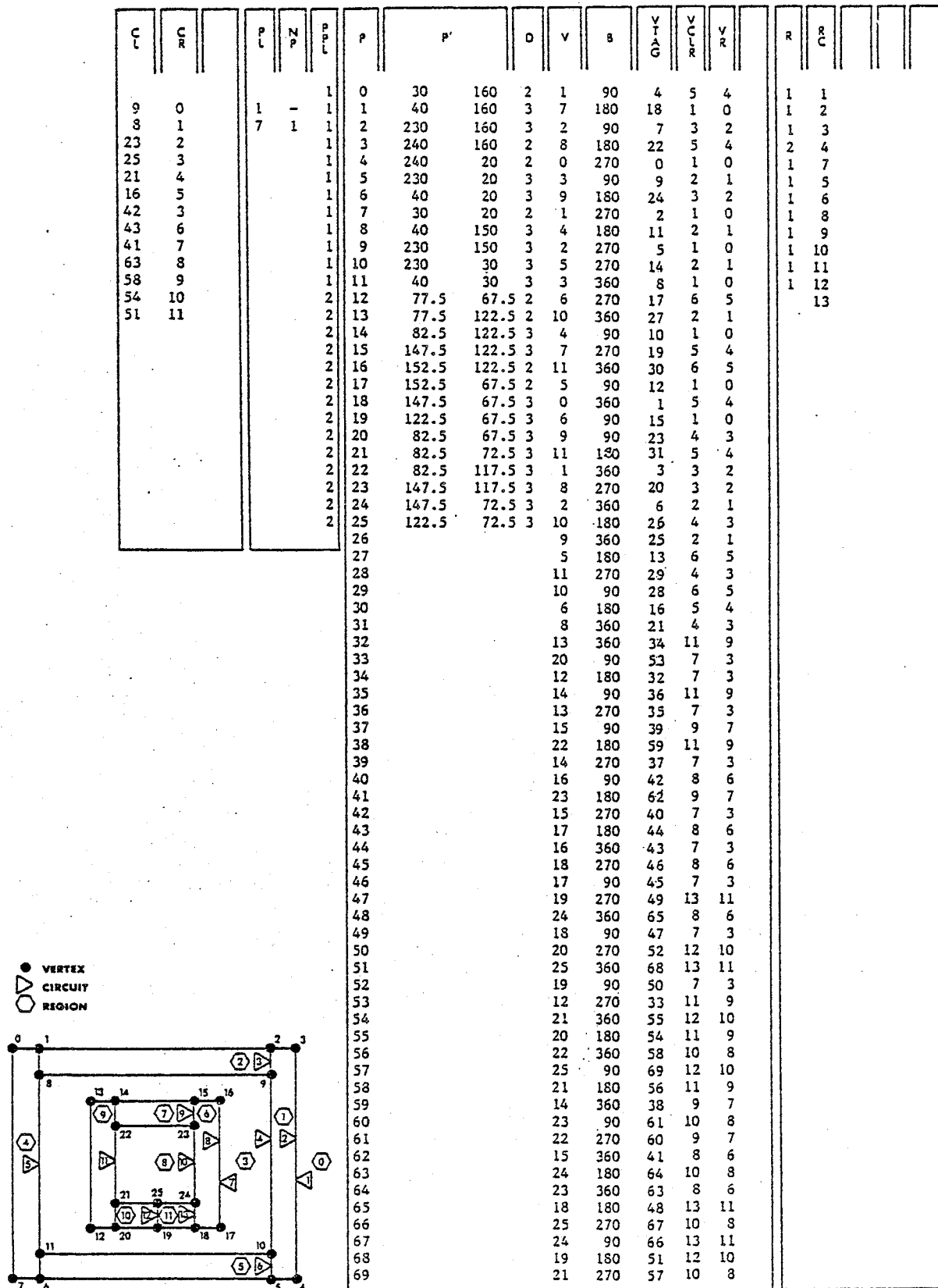


Figure 2.23 THE ENUMERATED REGIONS OF A GRAPH.

Algorithm 2.4 Attach attribute labels to the regions

TO ATTACH THE ATTRIBUTE LABELS* CONTAINED IN THE GRAPHIX GRAPHICAL DATA STRUCTURE TO THE REGIONS OF THE GRAPH G.

GIVEN: 1) The extended PDV description for the planar graph as developed in the previous algorithm, in which all the boundary circuits, regions and Parts of the graph have been identified and,

2) the AREAL and NAME class of sub-pictures contained in the GRAPHIX Graphical Data Structure for the plan section drawing currently under assembly.

and, DEFINING the lists RCOL, CCOL, and VCOL, such that:

- 1) RCOL defines the attribute identity of each region of the graph;
- 2) CCOL defines the attribute identity of each boundary circuit of the graph; and
- 3) VCOL defines the attribute identity of each edge of the graph.

1.1) Attach the 'exterior interface'** label to the exterior region (ie. region 0 in the lists R, RC, and RV) of the graph: enter the exterior interface code into those elements of RCOL, CCOL, and VCOL that are labelled in R, CR, and VR, respectively, as belonging to region 0. (Region 0 is identified by a 'E' in all Figures.)

* The attribute labels assigned to identify regions are identical to the sub-picture drawing codes used to describe the region in the GRAPHIX Graphical Data Structure.

** The 'interior interface' code labels those graph elements that are associated with the exterior of a building.

1.2) Identify and attach the 'attribute labels' to all remaining regions of the graph: examine each sub-picture instance in the GRAPHIX Graphical Data Structure (GGDS) for the plan section drawing currently under assembly and, for each instance, perform the following;

A) Acquire the attribute information from the GGDS;

- 1) if the sub-picture instance is of the LINEAR class, return to examine the next instance as no attribute information is conveyed by this class of sub-picture.

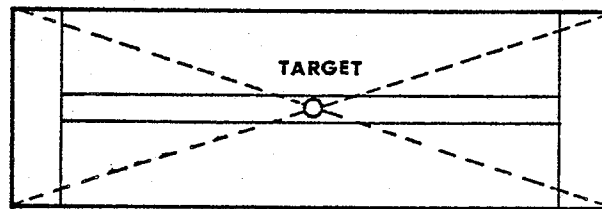


Figure 2.24 THE TARGET FOR AN AREAL CLASS SUB-PICTURE.

- 2) if the sub-picture instance is of the AREAL class; using the information defining the instance in the GGDS, calculate the coordinates for a TARGET point lying within (the centroid of the surface boundary rectangle - see Figure 2.24) the confines of that sub-picture surface boundaries; nominate the address of this AREAL

instance in the GGDS as the current COLOUR; and, proceed to 'B'.

- 3) if the sub-picture instance is of the NAME class; nominate the first coordinate for the NAME instance as the TARGET and the sub-picture display code for this instance as the current COLOUR; and proceed to 'B'.

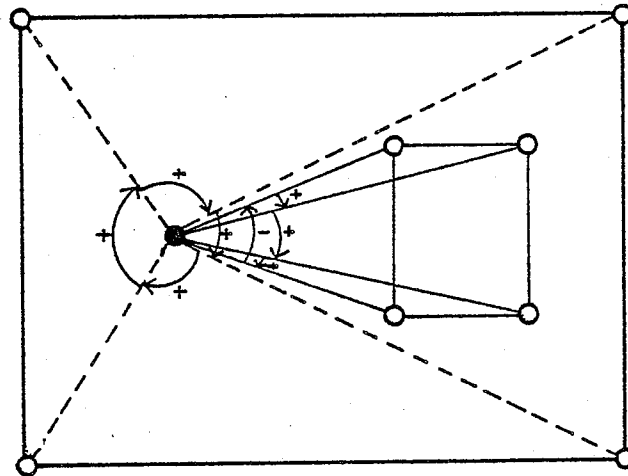


Figure 2.25 A TARGET WITHIN A COMPLEX REGION.

B) Identify the region of the graph within which the current TARGET lies:

- 1) search in turn all the complex regions, if present, of the graph that are identified in NP and NPC (ie. those

regions defined by more than one boundary circuit), and determine if the current TARGET lies inside the region (ie. inside the exterior boundary circuit of the region labelled in NPC for this region, and exterior to the remaining boundary circuits circumscribing this region - see Figure 2.25). If the TARGET does not lie within this or any other complex region of the graph, proceed to '2'. If the TARGET does lie within this complex region, perform the following; enter the current COLOUR into the elements of RCOL, CCOL, and VCOL that are labelled in the R, CR, and VR lists, respectively, as belonging to this region of the graph.

- 2) Otherwise, search in turn all simple regions of the graph identified in R (ie. those regions for which only one boundary circuit is defined), and determine which region the current TARGET lies within; enter the current COLOUR into the elements of RCOL, CCOL, and VCOL that are labelled in R, CR, and VR, respectively, as belonging to this region of the graph.

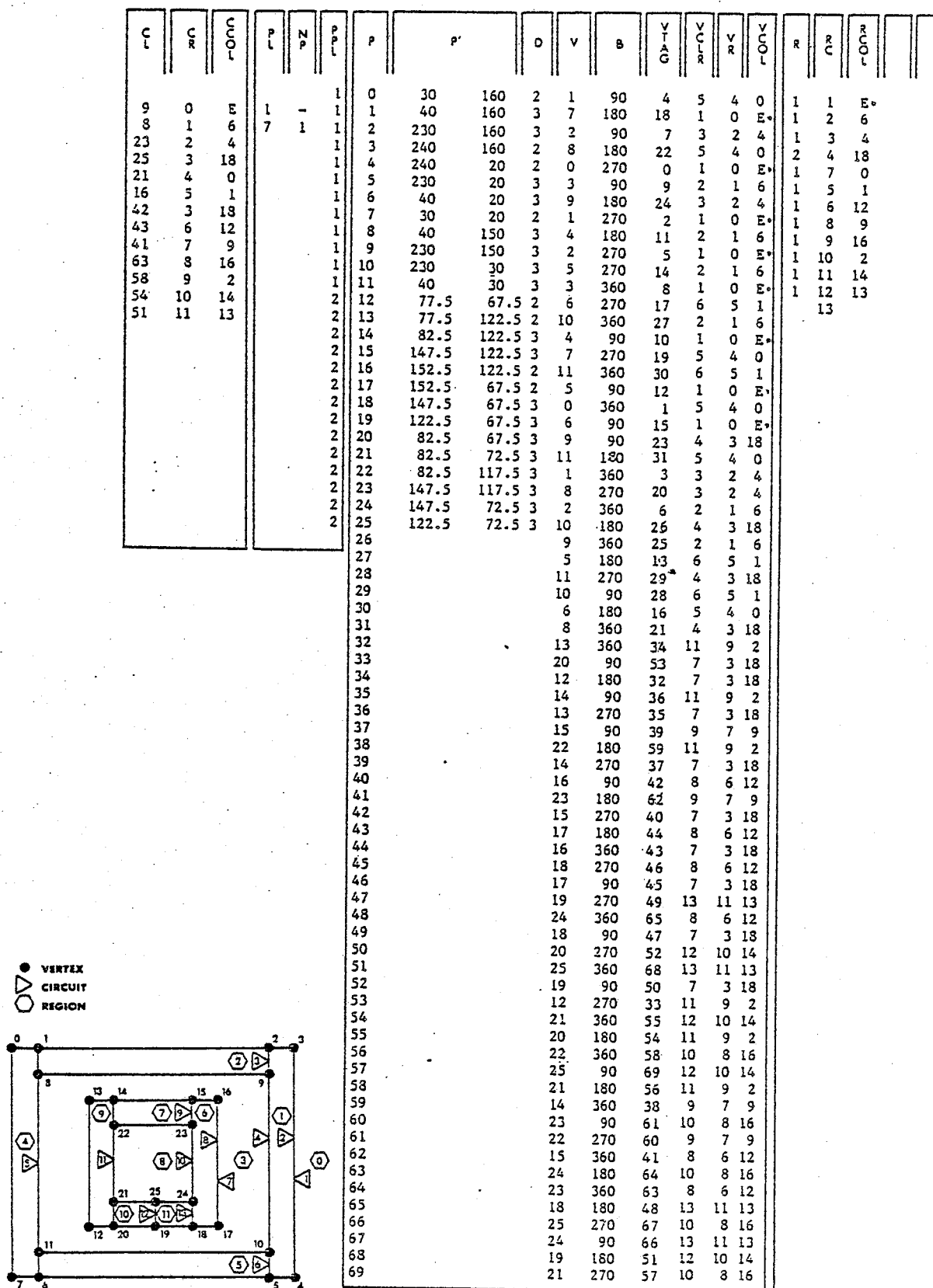


Figure 2.26 THE LABELLED REGIONS OF THE GRAPH.

Algorithm 2.5 Construct the dual graph $G(d)$

TO CONSTRUCT A DUAL GRAPH REPRESENTATION $G(d)$ OF THE PLANAR GRAPH G , where

each vertex of the dual graph $G(d)$ represents a region (container) of the graph, and each edge of the dual graph $G(d)$ represents the adjacency of two regions (a shared surface boundary) of the graph G .

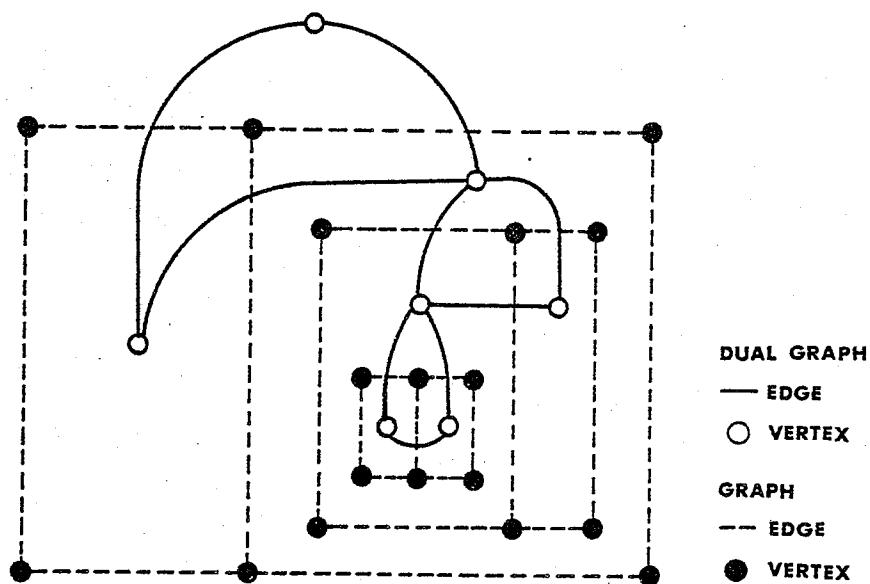


Figure 2.27 THE DUAL GRAPH $G(d)$ OF THE PLANAR GRAPH G .

GIVEN 1) The extended PDV description for the planar graph G , as developed to this juncture; where

- a) all the regions of the graph G are defined in R ; and
- b) the identity of the adjacent boundary circuit, region, and colour for each edge exiting each vertex of the graph is identified (see Figure 2.28) in the elements of VC , VR , and $VCOL$ that are associated with the exit edge immediately counterclockwise to the edge in V (as noted in algorithm 2.2, the exit edges for each vertex of the graph are listed in clockwise order in V).

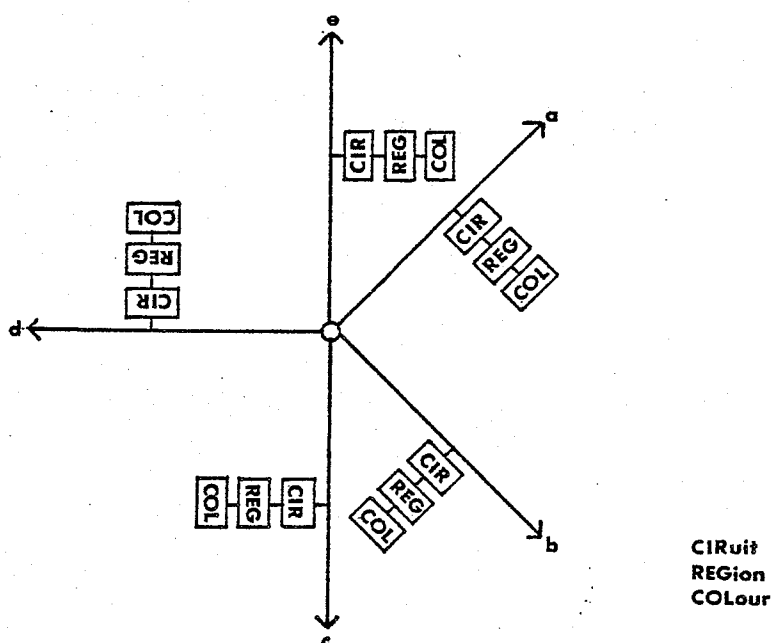


Figure 2.28 ADJACENT BOUNDARY CIRCUITS, REGIONS AND COLOURS.

and DEFINING the lists RAN and $RDUAL$ to be used, in conjunction

with the previously defined list R , to store the description of the dual graph $G(d)$, such that;

- 1) R , previously used to define the regions of the graph G , will identify the vertices of the dual graph $G(d)$;
- 2) RAN defines, for each vertex in the dual graph $G(d)$ (ie. each region of the graph G), the number of adjacent vertices (ie. regions); and,
- 3) $RDUAL$ defines, for each vertex of the dual graph $G(d)$, the identity (ie. region labels) of all adjacent vertices (ie. regions).

STEP 1: CONSTRUCT A DUAL GRAPH $G(d)$ OF THE PLANAR GRAPH G .

1.1) Select in turn each region of the graph G identified in the list R , nominating the region selected as the current OBJECT region and, selecting the boundary circuit(s) from RC that are contiguous to the current OBJECT region, perform the following;

- A) Obtain the address in V for an exit edge lying along the boundary circuit contiguous to the current OBJECT region, and
- B) Traverse the exit edges of the boundary circuit, counting and identifying each region that is adjacent to the current OBJECT; enter the count for the number of adjacent regions into that element of RAN that is associated with the current OBJECT region; and, successively enter each of the region labels identifying the regions adjacent to the current OBJECT region into the next available elements of $RDUAL$.

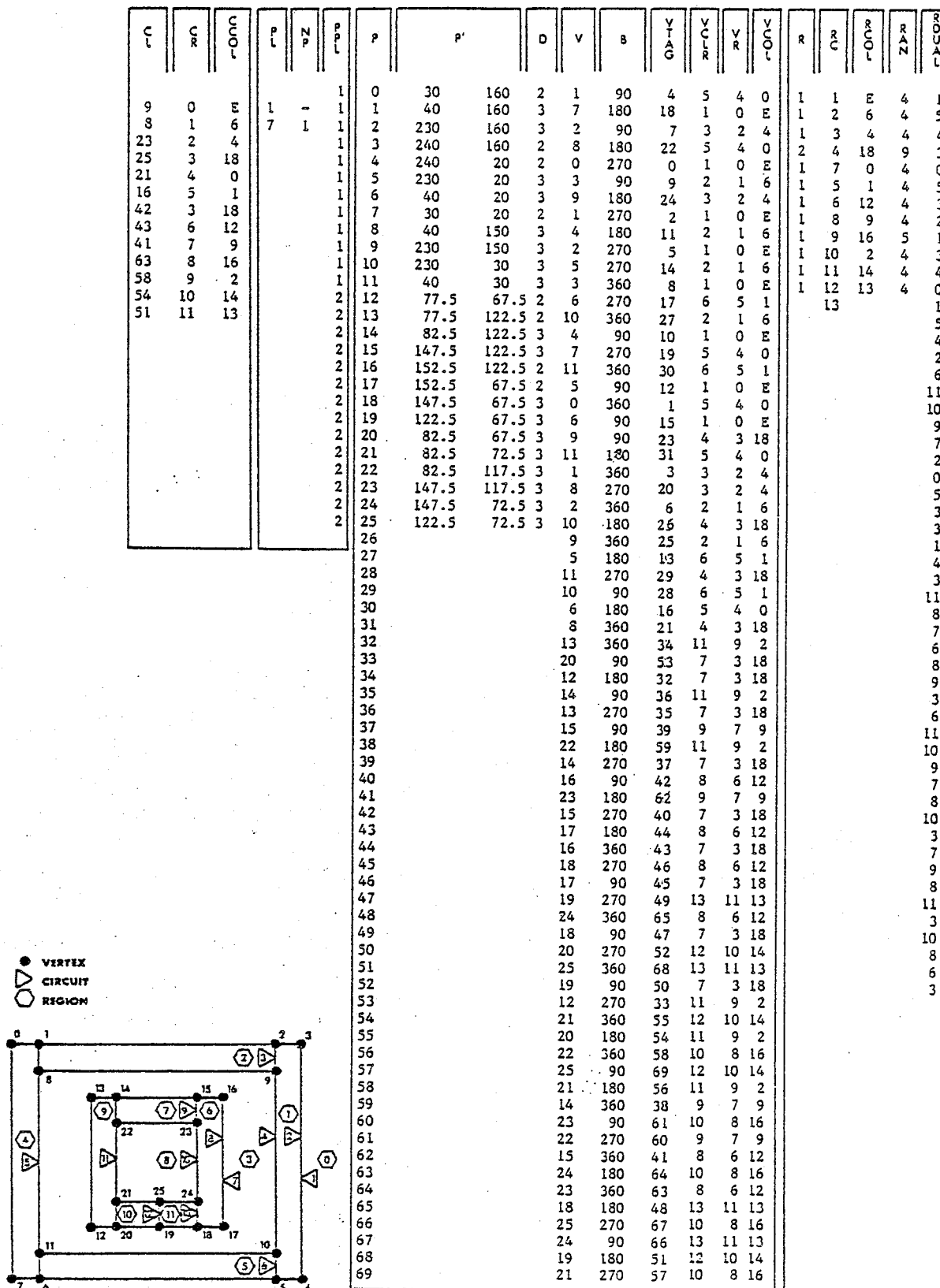


Figure 2.29 THE ADJACENT REGIONS OF THE GRAPH.

Section 2.3 Assembling and Enumerating the overlay graph GOIntroduction

Once the Planar Graph Representation of each plan section drawing has been assembled, the construction process for the three dimensional network representation of the building can be initiated. This process requires that the Planar Graph Representation of vertically adjacent building sections be pairwise overlayed and the planar graphs resulting from that union procedure be inspected. This section documents the algorithms and the extensions to the PDV data structure that are used to perform the graph overlay and inspection procedures which result in the generation of the network representation for a building.

A series of five algorithms is used in the construction process for the three dimensional network representation of a building. Three of the algorithms in this series were developed specifically for the task at hand. Two of the algorithms used in this process are also used in the assembly process for the two dimensional Planar Graph Representation. A functional description for each of the five algorithms is presented below in order to summarize the sequence of events followed in the construction process for the three dimensional network representation. Detailed operational descriptions for each of the three algorithms specifically developed for the network representation construction process follow later. As in the previous section, descriptions of the extensions to the PDV data structure are incorporated within the

description for the algorithms generating the information base.

Procedural Summary

The overlay and inspection procedure applied to the Planar Graph Representation of each pair of vertically adjacent building sections is executed in the following manner:

Algorithm 2.6 constructs an overlay planar graph G_0 that describes the union of the Planar Graph Representations of two vertically adjacent building sections. The regions of the overlay graph G_0 resulting from this union map the vertical juxtaposition of individual containers delineated in the adjacent building sections.

Algorithm 2.2 enumerates the individual circuits of the graph G_0 that are coincident to each vertically adjacent region of the graph. The basic PDV description of the graph G_0 (constructed in Algorithm 2.6) is inspected and the circuits circumscribing the exterior region and all circuits circumscribing the interior regions of each Part of the graph G_0 are enumerated.

Algorithm 2.3 identifies the surface boundaries (ie. boundary circuits) that map each region of the graph G_0 . The Part and boundary circuit information generated in the previous algorithm is inspected and the boundary circuit(s) circumscribing each region are labelled as belonging to that region of the graph.

Algorithm 2.7 Each region in the graph G_0 maps the juxtaposition of two vertically adjacent building components. Each of these building components is identified by a unique region label in their respective horizontal planar graph representations ($G(u)$ and $G(l)$). This algorithm attaches two data items to each region in the overlay graph G_0 ; one identifies the region label for the building component juxtaposed immediately below (in $G(l)$) and the other identifies the region label for the building component juxtaposed immediately above (in $G(u)$). Each of these data items describes a vertical (directed) edge in the building NETWORK representation.

Algorithm 2.8 inspects the vertical adjacency linkage lists previously established for the regions of the graph G_0 and creates two similar and reciprocal vertical adjacency linkage lists for the regions of both the lower and upper graphs, $G(l)$ and $G(u)$, respectively. Once this procedure has been executed for each of the plan section drawings, each mode (container) and each edge (shared container surface boundary) of the building network will have been described and the construction process for the NETWORK representation of the building completed.

The AlgorithmsAlgorithm 2.6 Construct the overlay graph G_0

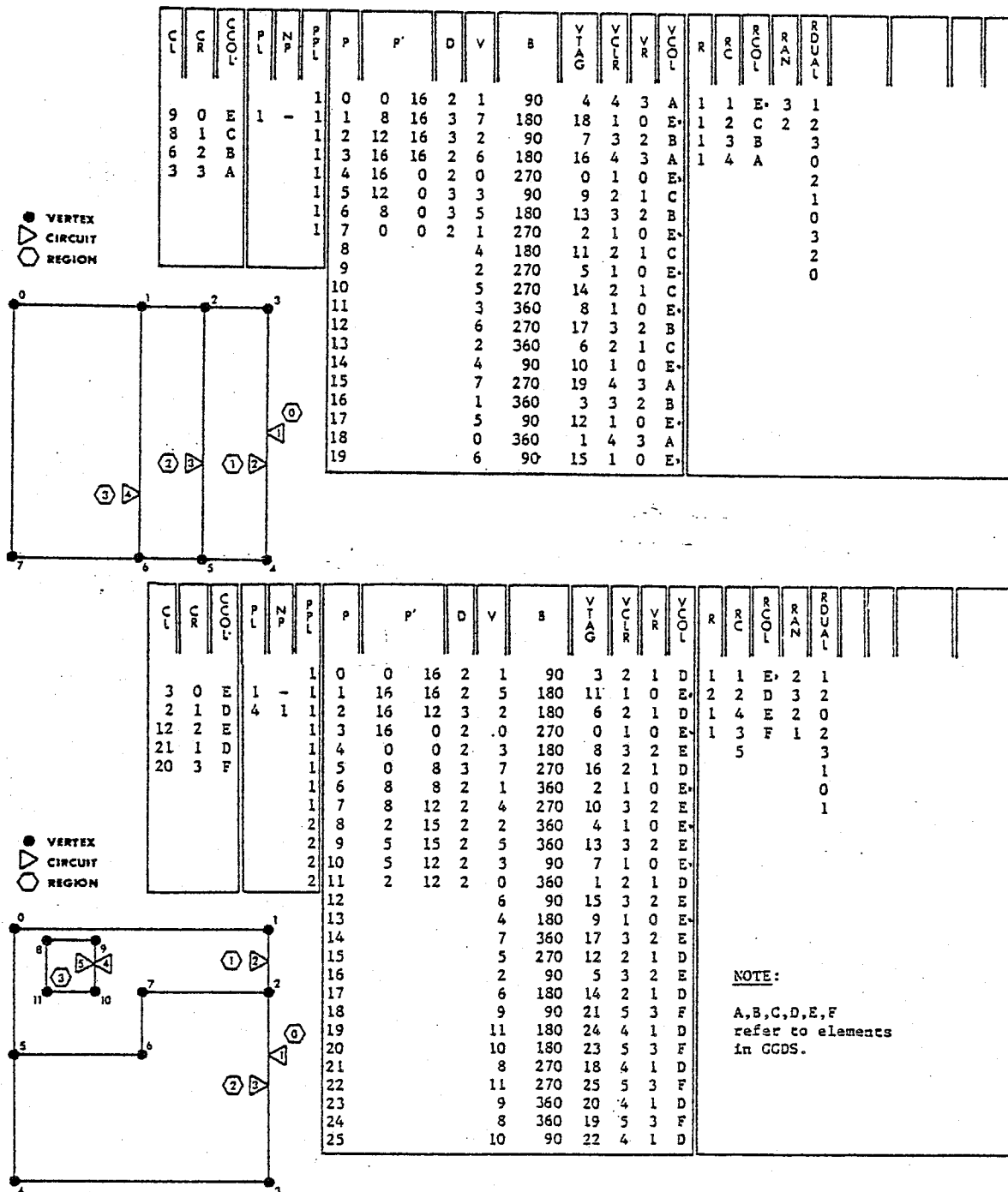
TO OVERLAY TWO VERTICALLY ADJACENT PLANAR GRAPHS OF THE SERIES DESCRIBING A BUILDING AND TO CONSTRUCT AN EXTENDED PDV DESCRIPTION OF THE PLANAR GRAPH RESULTING FROM THAT UNION.

GIVEN: 1) The extended PDV description (as developed in section 2 of this chapter) of two vertically adjacent planar graphs, referred to as the upper graph $G(u)$ and the lower graph $G(l)$, that describe vertically adjacent building sections.

and DEFINING the following graphs;

- 1) two temporary graphs $GT(u)$ and $GT(l)$ describing the parent graphs $G(u)$ and $G(l)$ respectively, where
 - A) the vertices of the temporary graphs map: the vertices of its parent graph; the intersection of edges of the parent graph with the edges of the adjacent parent graph; and, the coincidence of edges of the parent graph with vertices of the adjacent parent graph; and,
 - B) the edges of the temporary graphs map the adjacency of the vertices of a temporary graph as defined in its parent graph.
- 2) the overlay planar graph G_0 describing the union of the graphs $G(u)$ and $G(l)$

and, DEFINING the following for use in the inspection, construction

Figure 2.30 THE UPPER AND LOWER GRAPHS $G(u)$ AND $G(1)$.

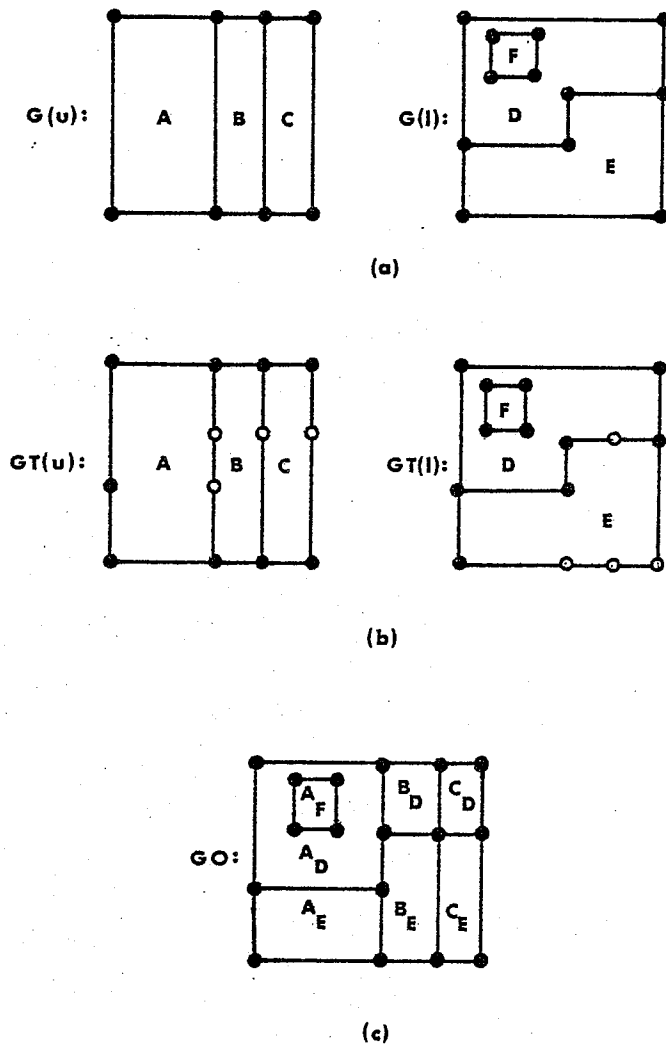


Figure 2.31 THE PARENT, TEMPORARY, AND OVERLAY GRAPHS.

and storage of the Parent, Temporary and Overlay graphs:

- 3) the lists TAG, VPI, VPD, and VP are used to enumerate the points of intersection and junction of the temporary

graphs, where;

- A) TAG defines, for each exit edge in V of the parent graph, whether that exit edge is the 'first occurrence' in V of the description of that edge or the 'second occurrence', and if the 'second occurrence', the address in V of the 'first occurrence' (see Figure 2.32);

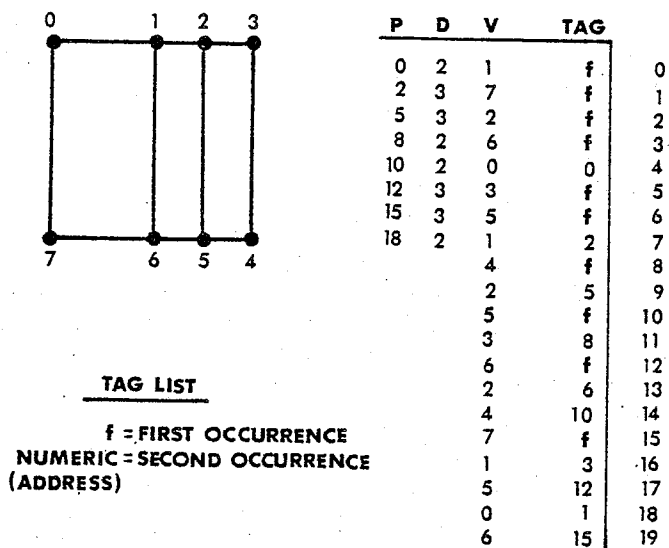


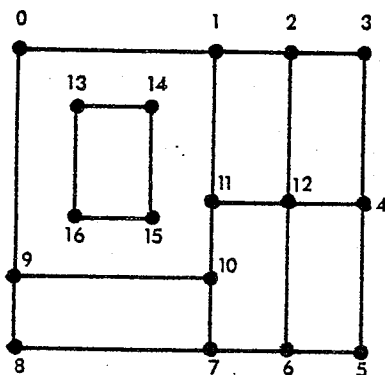
Figure 2.32 FIRST AND SECOND OCCURRENCES IN THE TAG LIST.

- B) VPI defines, for each exit edge of the parent graph, the point(s) of intersection and/or coincidence between the adjacent Parent and that exit edge;
- C) VPD defines, for each exit edge of the parent graph,

the number of points of intersection and/or coincidence between the adjacent Parent graph and that exit edge; and,

- D) VP defines, for each exit edge of the parent graph, the start address in VPI of those points of intersection/coincidence for that exit edge;
- 4) the lists GTP, GTP', GTD, GTV, GTB, and GTR are used to store an extended PDV description for each Temporary graph (the prefix GT referencing the list that is used for the Temporary graph and the suffix referencing the standard list elements of the extended PDV description, as defined in Algorithm 2.1);
- 5) the lists GOP, GOP', GOD, GOV, GOB and the lists GORU, GORL and ALIAS are used to store an extended PDV description of the overlay graph resulting from the union of the Parent graphs $G(u)$ and $G(l)$ (via their Temporary graphs $GT(u)$ and $GT(l)$) where;
- A) GORU defines, for each exit edge of the overlay graph, the region of the upper parent graph $G(u)$ lying immediately to the right of that exit edge (see Figure 2.33);
- B) GORL defines, for each exit edge of the overlay graph, the region of the lower parent graph $G(l)$ lying immediately to the right of that exit edge (see Figure 2.33);
- C) ALIAS defines, for each vertex of the Temporary graph

GT(1), the vertex label that will be assigned to that vertex in the overlay graph G₀; and,



P	V	GORU	GORL
10 → 11		B	E
11 → 1		B	not assigned
11 → 12		not assigned	E

Figure 2.33 THE GORU AND GORL LISTS OF THE OVERLAY GRAPH.

D) GOP, GOP', GOD, GOV, and GOB define the vertices, position, degree, exit edges, and bearing angle of the exit edges for the overlay graph G₀.

STEP 1: IDENTIFY ALL 'SECOND OCCURRENCES' IN THE V LIST.

1.1) Identify and tag the 'second occurrence' of the two exit edges that are listed in V of the graph G(u) for each surface boundary (see Figure 2.32): initialize all the elements of TAG(u) as being the 'first occurrence'; select 'first occur-

rence' exit edges in turn from V of the upper graph $G(u)$, nominating the selected exit edge as the current SUBJECT EDGE, and perform the following for each SUBJECT EDGE;

- A) Identify both the SOURCE VERTEX being exited by the current SUBJECT EDGE and the OBJECT VERTEX being entered by the current SUBJECT EDGE.
 - B) Enter P of the graph $G(u)$ at the OBJECT VERTEX and, examining the list of exit edges for that vertex, locate the exit edge entrant to the current SOURCE VERTEX and nominate that edge as the current OBJECT EDGE; enter the address in V of the current SUBJECT EDGE into that element of $TAG(u)$ that is associated with the current OBJECT EDGE (marking this OBJECT EDGE as the 'second occurrence' of the SUBJECT EDGE).
- 1.2) Identify and tag the 'second occurrence' of the two exit edges that are listed in V of the graph $G(l)$ for each surface; perform those operations described in 1.1 above, substituting $TAG(l)$ and $G(l)$ for the terms $TAG(u)$ and $G(u)$, respectively.

STEP 2: IDENTIFY ALL THE VERTICES OF THE TEMPORARY GRAPH $GT(u)$ THAT WILL RESULT FROM THE UNION OF THE GRAPHS $G(u)$ and $G(l)$.

- 2.1) Identify all the vertices of the graph $GT(u)$: select in turn each vertex of the upper graph $G(u)$, nominating the selected vertex as the current SUBJECT VERTEX; for each SUBJECT VERTEX, select in turn 'first occurrence' exit edges of the current SUBJECT VERTEX, nominating the selected exit as the

SUBJECT EDGE, and perform the following;

- A) Enter the address of the next available element in VPI into that element in VP that is associated with the current SUBJECT EDGE and initialize the intersect counter.
- B) Examine in turn the 'first occurrence' (TAG(1)=0) exit edges of the lower graph G(1), nominating the selected exit edge as the current OBJECT EDGE, and perform the following for each OBJECT EDGE;
 - 1) Determine if the current OBJECT EDGE (including its incident vertices) is coincident to the current SUBJECT EDGE and
 - a) if the two edges do not intersect; return to B if there are exit edges of the lower graph G(1) yet to be examined; otherwise proceed to B.3.
 - b) if the two edges do intersect; enter the coordinate pair locating the point of intersection into the next available element of the VPI list; increment the intersect counter and return to B if there are exit edges of the lower graph G(1) yet to be examined; otherwise, proceed to B.3.
 - 2) Enter the intersect counter (identifying the number of points of intersection occurring along the current SUBJECT EDGE) into that element of VPD that is associated with the current SUBJECT EDGE.
 - 3) Examine the intersection coordinates in VPI for the current SUBJECT EDGE and re-sequence the coordinate

pairs in this portion of the list such that the elements map, from top to bottom, the order of intersection along the current SUBJECT EDGE from its SOURCE VERTEX to its OBJECT VERTEX (see Figure 2.34); return to B if there are exit edges of $G(u)$ that have yet to be examined for the current SUBJECT VERTEX or if there are yet to be examined vertices in $G(u)$; otherwise, proceed to Step 3.

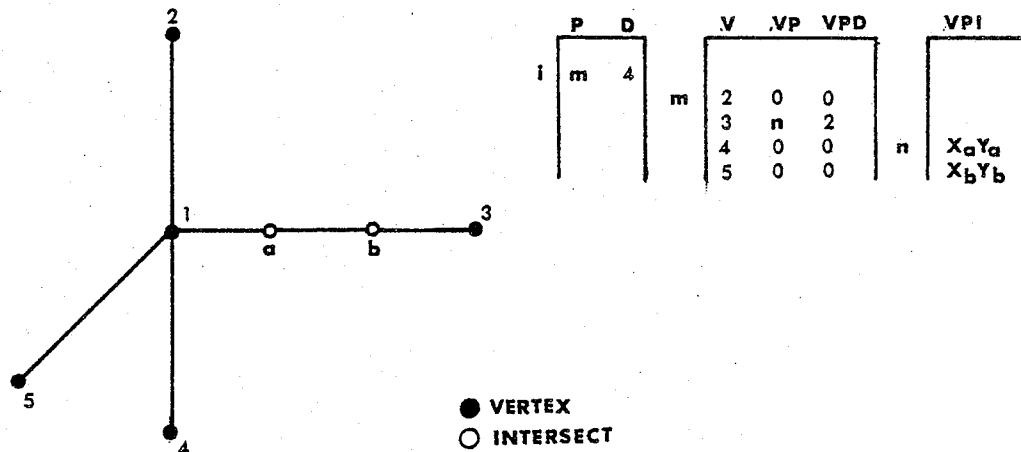


Figure 2.34 THE RESEQUENCING OF EDGE INTERSECTS IN VPI.

STEP 3: CONSTRUCT AN EXTENDED PDV DESCRIPTION OF THE TEMPORARY GRAPH $GT(U)$

3.1) Enumerate the exit edges of the non-intersect vertices of

the temporary graph $GT(u)$: count the number of vertices in the graph $G(u)$ and assign the graph vertex counter this value; set the vertex counter equal to the graph vertex vertex and increment the vertex counter; select in turn each vertex of the graph $G(u)$, nominating the selected vertex as the current SUBJECT VERTEX, and perform the following for each SUBJECT VERTEX;

- A) Enter the address of the next available element in $GTV(u)$ into the next available element of $GTP(u)$; obtain the coordinate and degree of the current SUBJECT VERTEX from P' and D of the graph $G(u)$ and enter the values into the next available element of $GTP'(u)$ and $GTD(u)$, respectively.
- B) Examine in turn the exit edges of the current SUBJECT VERTEX, nominating the selected exit edge as the current SUBJECT EDGE and the vertex being entered as the current OBJECT VERTEX, and perform the following for each SUBJECT EDGE;
 - 1) Obtain the bearing angle for the current SUBJECT EDGE from $B(u)$ and enter this angle into the next available element of $GTB(u)$.
 - 2) Examine the $TAG(u)$ list element (indicating whether or not this edge is the first or second occurrence of this edge in $V(u)$) associated with the current SUBJECT EDGE and;
 - a) if the SUBJECT EDGE is the 'first occurrence' ($TAG=0$); obtain, from $VPD(u)$, the number of

intersect points occurring along the SUBJECT EDGE,
and;

- i) if there are no edge intersects ($VPD=0$), enter the current OBJECT VERTEX into the next available element of $GTV(u)$.
- 11) if there is one or more edge intersects ($VPD(u) > 0$); obtain the first (uppermost) intersect coordinate pair from $VPI(u)$ that is associated with the current SUBJECT EDGE and enter that coordinate pair into that element of $GTP'(u)$ that is addressed by the vertex counter; enter the vertex counter (ie. the vertex label for the coincident point in the graph $G(u)$) into the first element of $VPI(u)$ for the current SUBJECT EDGE (replacing the coordinate of the intersect vertex with the vertex label to be used to refer to that intersect vertex); enter the vertex counter into the next available element of $GTV(u)$; and, add the number of intersect edges associated with the current SUBJECT EDGE to the vertex counter (see Figure 2.35).
- b) if the SUBJECT EDGE is the 'second occurrence' (ie. $TAG =$ address in $V(u)$ of the first edge occurrence); obtain from the $VPD(u)$ element associated with the 'first occurrence' of the SUBJECT EDGE, the number

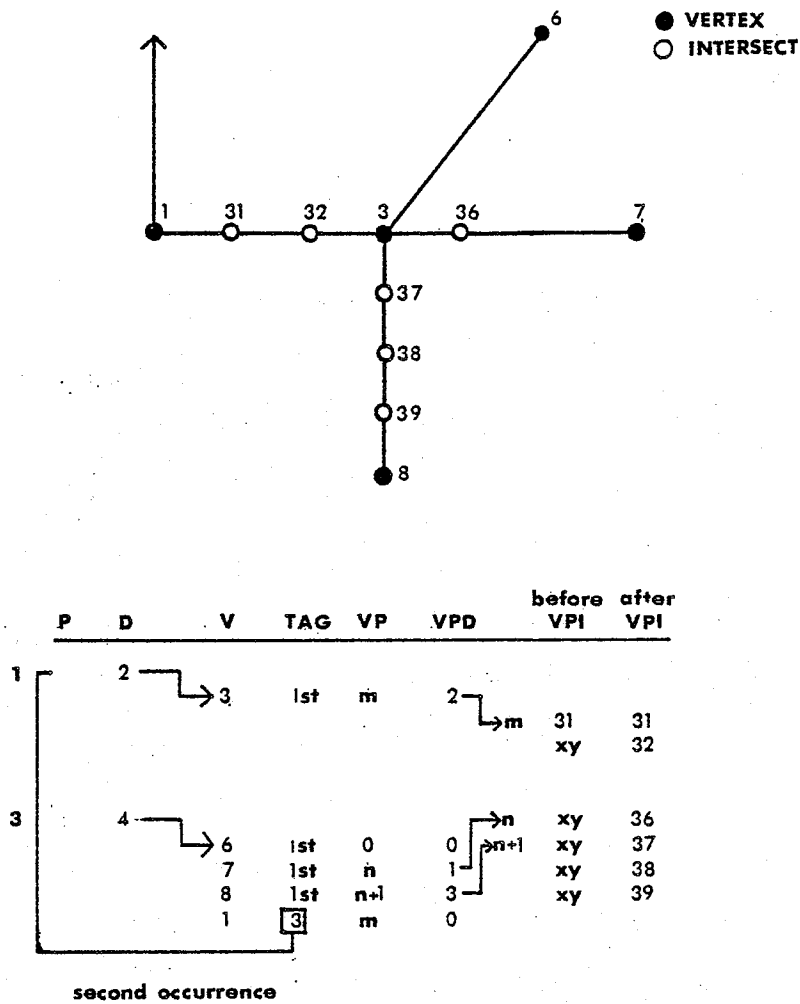


Figure 2.35 INTERSECTS ON SECOND OCCURRENCE EDGES.

of edge intersects occurring along the current
SUBJECT EDGE and;

i) if there are no edge intersects; enter the

current OBJECT VERTEX into the next available element of GTV(u).

- ii) if there is only one edge intersect; obtain the vertex label resident in the first (uppermost) element of VPI(u) that is associated with the 'first occurrence' of the current SUBJECT EDGE and enter it into the next available element of GTV(u).
- iii) if there is more than one edge intersect; obtain both the vertex label (from VPI(u)) assigned to the first edge intersect and the number of edge intersects (from VPD(u)) for the 'first occurrence' of the current SUBJECT EDGE and, decrementing the number of intersects by one, add the two values and nominate the result as the current intersect vertex label; obtain the coordinates of the last (lowermost) edge intersect lying along the 'first occurrence' of the current SUBJECT EDGE and enter its coordinates into that element of GTP'(u) that is addressed by the intersect vertex value; and, enter the intersect vertex label into the last element of VPI(u) that is associated with the 'first occurrence' of the current SUBJECT EDGE.

3.2) Enumerate the exit edges of the intersect vertices of the

temporary graph $GT(u)$ (see Figure 2.36): set the vertex counter equal to the graph vertex counter and increment the vertex counter; select in turn each vertex of the graph $G(u)$, nominating the selected vertex as the current SUBJECT VERTEX, and perform the following for each SUBJECT VERTEX;

- A) Examine in turn the 'first occurrence' exit edges of the current SUBJECT VERTEX, nominating the selected exit edge as the current SUBJECT EDGE; nominate the vertex being entered by the current SUBJECT EDGE as the current OBJECT VERTEX; obtain from R the region label identifying the region lying to the right of the current SUBJECT EDGE, nominating this region label as the current RIGHT LABEL; obtain from R the region lying immediately to the left of the current SUBJECT EDGE (the region label affixed to that edge listed immediately counter-clockwise in V to the current SUBJECT EDGE), nominating this region label as the current LEFT LABEL; obtain the bearing angle for the current SUBJECT EDGE from B and nominate this angle as the current RIGHT ANGLE; compute the inverse bearing angle for the current SUBJECT EDGE and nominate it as the current LEFT ANGLE; and, obtain the number of intersect points for the current SUBJECT EDGE from that element of $VPD(u)$ that is associated with the current SUBJECT EDGE and;
- 1) If there are no edge coincident points ($VPD=0$), return to 3.2 to examine the next exit edge.
 - 2) If there is only one intersect ($VPD=1$); enter the

address of the next available element in $GTV(u)$ into that element of $GTP(u)$ addressed by the vertex counter; enter a degree of two (2) into that element of $GTD(u)$ addressed by the vertex counter; enter the OBJECT VERTEX into the next available element of $GTV(u)$ and enter the current RIGHT LABEL into the next available element of $GTR(u)$; enter the current RIGHT ANGLE into the next available element of $GTB(u)$; enter the SUBJECT VERTEX into the next available element of $GTV(u)$ and enter the current LEFT LABEL into the next available element of $GTR(u)$; enter the LEFT ANGLE into the next available element of $GTB(u)$; and increment the vertex counter.

- 3) If there are two or more edge intersects ($VPD(u) \geq 1$); perform the following for each edge intersect in order of occurrence (ie. from the SUBJECT VERTEX to the OBJECT VERTEX) along the current SUBJECT EDGE;

a) enter the address of the next available element in $GTV(u)$ into that element of $GTP(u)$ addressed by the vertex counter; enter a degree of two (2) into that element of $GTD(u)$ addressed by the vertex counter; and, depending on the position of the current edge intersect in the sequence occurring along the current SUBJECT EDGE, perform one of the following;

- i) if the current edge intersect is the first in the sequence; enter the current SUBJECT VERTEX into the next available element of $GTV(u)$;

enter the current LEFT LABEL into the next available element of GTR(u); enter the LEFT ANGLE into the next available element of GTB(u); enter the vertex counter plus one into the next available element of GTV(u); enter the current RIGHT LABEL into the next available element of GTR(u); enter the RIGHT ANGLE into the next available element of GTB(u); and, increment the vertex counter.

ii) if the current edge intersect is the last in the sequence; enter the vertex counter minus one into the next available element of GTV(u); enter the current LEFT LABEL into the next available element of GTR(u); enter the LEFT ANGLE into the next available element of GTB(u); enter the current OBJECT VERTEX into the next available element of GTV(u); enter the current RIGHT LABEL into the next available element of GTR(u); enter the RIGHT ANGLE into the next available element of GTB(u); and, increment the vertex counter.

iii) if the current edge intersect is neither the first nor the last in the sequence; enter the vertex counter minus one into the next available element of GTV(u); enter the LEFT LABEL into the next available element of

GTR(u); enter the vertex counter plus one into the next available element of GTV(u); enter the RIGHT LABEL into the next available element of GTR(u); enter the RIGHT ANGLE into the next available element of GTB(u); and, increment the vertex counter.

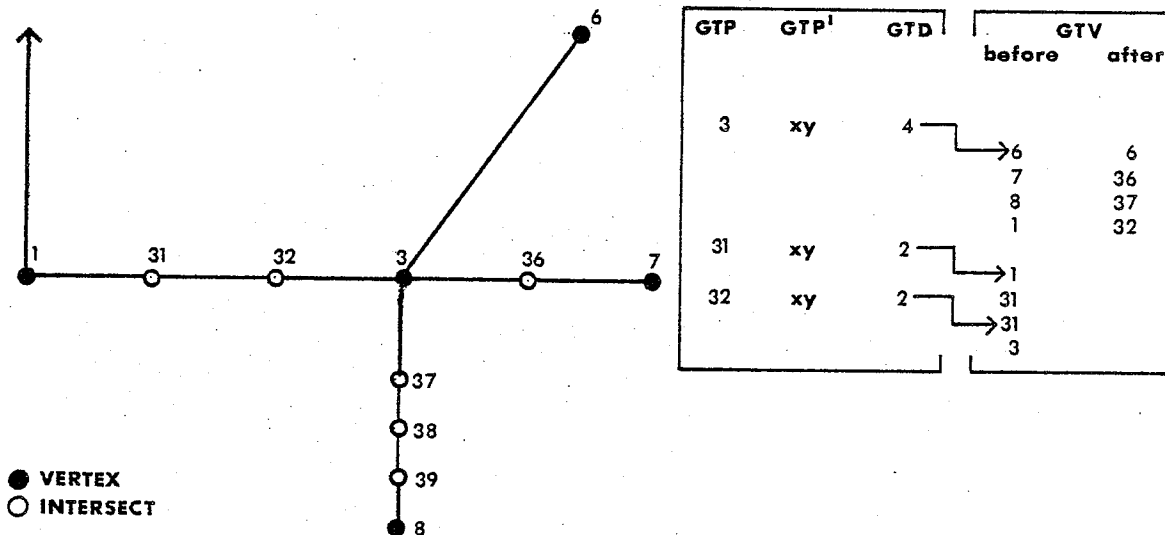


Figure 2.36 ENUMERATE THE INTERSECT VERTICES OF GRAPH GT.

STEP 4: CONSTRUCT AN EXTENDED P,D,V DESCRIPTION OF THE TEMPORARY GRAPH GT(1).

4.1) Perform those operations described in steps 2 and 3 above, reversing and substituting those terms referring to the graphs

$G(u)$ and $G(l)$, and construct the description for the temporary graph $GT(l)$.

STEP 5: CONSTRUCT A DESCRIPTION OF THE OVERLAY GRAPH G_0 RESULTING FROM THE UNION OF THE GRAPHS $G(u)$ AND $G(l)$.

5.1) Construct the ALIAS vertex label list for the vertices of the graph $GT(l)$, assigning to each vertex of $GT(l)$ a label which will identify that vertex in the graph G_0 : count the vertices of the graph $GT(u)$ and assign the vertex counter that value; select in turn each vertex of the graph $GT(l)$, nominating the selected vertex as the current SUBJECT VERTEX, and perform the following for each SUBJECT VERTEX;

A) Obtain the coordinate value for the current SUBJECT VERTEX from GTP' of the graph $GT(l)$ and nominate that coordinate as the current TARGET.

B) Select in turn each vertex of the graph $GT(u)$ and nominate the vertex selected as the current OBJECT VERTEX; assign FOUND the value 'not found'; obtain the coordinate for the current OBJECT VERTEX and compare this coordinate with the current TARGET coordinate, and;

1) If the coordinates are not equal; return to B to select the next vertex of the graph $GT(u)$.

2) If the coordinates are equal; enter the OBJECT VERTEX into that element of ALIAS that is associated with the current SUBJECT VERTEX; assign FOUND the value 'found'; and, proceed to C immediately below.

C) Examine the value of FOUND and if a coordinate equal to the TARGET was 'not found' in B above; increment the vertex counter and enter that value into that element of ALIAS that is associated with the current SUBJECT VERTEX.

5.2) Enter the vertices of the graph GT(u) and all the exit edges of these vertices (as listed in GT(u) and, through the ALIAS list, in GT(1)) into the description of the graph GO: initialize the elements of TAG as being 'available'; select in turn each vertex of the graph GT(u), nominating the selected vertex as the current SUBJECT VERTEX, and perform the following for each SUBJECT VERTEX;

A) Obtain the coordinate location for the current SUBJECT VERTEX from GTP' and enter that coordinate into the next available element of GOP'; enter the address of the next available element of GOV into the next available element of GOP; selecting in turn each exit edge of the current SUBJECT VERTEX and obtaining the label identifying the vertex being entered, the bearing angle of that exit edge, and the label identifying the region lying to the right of that exit edge (from GTV(u), GTB(u) and GTR(u) respectively), enter each value into the next available element of GOV, GOB, and GORU, respectively; and, obtain the degree of the current SUBJECT VERTEX from GTD(u), and assign that value to the degree counter.

B) Examine the ALIAS vertex label assigned to each vertex of the graph GT(1) and if the ALIAS label equals the current

SUBJECT VERTEX, nominate this vertex as the current OBJECT VERTEX and perform the following;

- 1) TAG the current OBJECT VERTEX of the graph GT(1) as being 'not available'.

- 2) Examine in turn each exit edge of the current OBJECT VERTEX, nominating the exit edge selected as the current OBJECT EDGE, and perform the following for each OBJECT EDGE;

- a) examine in turn each exit edge of the current SUBJECT VERTEX and;

- i) if the OBJECT EDGE is coincident to the SUBJECT EDGE (ie. the coordinate values for the vertices being entered by each edge are identical); enter the region label for the current OBJECT EDGE into that element of the GORL list that is associated with the current SUBJECT EDGE.

- ii) if the OBJECT EDGE is not coincident to the SUBJECT EDGE; enter the bearing angle and the region label of the OBJECT EDGE of the graph GT(1) into the next available element in their respective lists GOB and GORL; obtain the ALIAS vertex label for the vertex being entered along the current OBJECT EDGE and enter that ALIAS into the next available element of GOV; and, increment the degree counter.

C) Enter the degree counter into the next available element of GOD.

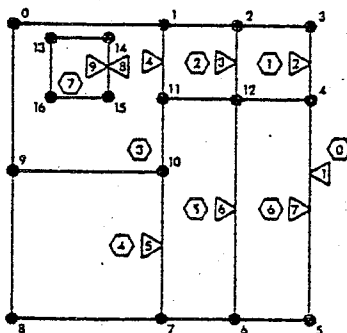
5.3) Enter the remaining 'available' vertices of the graph GT(1) into the description of the graph GO: select in turn each 'available' vertex of the graph GT(1), nominating the selected vertex as the current SUBJECT VERTEX, and perform the following for each SUBJECT VERTEX;

A) Enter the address of the next available element of GOV into the next available element of GOP (ie. that element addressed by the current SUBJECT VERTEX); obtain and enter the degree of the SUBJECT VERTEX into the next available element of GOD; select in turn each exit edge of the current SUBJECT VERTEX and obtaining the ALIAS vertex label identifying the vertex being entered along the selected exit edge, the bearing angle of the selected exit edge, and the label identifying the region lying to the right of the selected exit edge, enter each value into the next available element in GOV, GOB, and GORL, respectively.

5.4) Re-organize the exit edge, bearing, and region lists for each vertex of the graph GO such that the sequence of exit edges in the V list maps the clockwise occurrence of the exit edges about the vertex.

C	C	N	P	P	P	P	O	V	S	V	V	G	G	G	G	V	R	R	R
L	R	P	L	L	L	L				A	C	U	R	R	R	R	C	C	C
9	0	1	-	1	0	0	16	2	1	90	4	4	A	D		3	1	1	
8	1	8	1	1	1	8	16	3	9	180	23	1	E	E		0	1	2	
6	2			1	2	12	16	3	2	90	7	3	B	D		2	1	3	
3	3			1	3	16	16	2	11	180	29	4	A			3	2	4	
24	4			1	4	16	12	3	0	270	0	1	E	E		0	1	8	
19	5			1	5	16	0	2	3	90	8	2	C	D		1	1	5	
16	6			1	6	12	0	3	12	180	32	3	B			2	1	6	
39	3			1	7	8	0	3	1	270	2	1	E	E		0	1	7	
38	7			1	8	0	0	2	4	180	12	2	C	D		1		9	
				1	9	0	8	3	2	270	5	1	E	E		0			
				1	10	8	8	3	5	180	14	7	C	E		6			
				1	11	8	12	3	12	270	33	2		D		1			
				1	12	12	4	3	3	360	8	1	E	E		0			
				2	13	2	15	2	6	270	17	7	C	E		6			
				2	14	5	15	2	4	360	10	1	E	E		0			
				2	15	5	12	2	7	270	20	6	B	E		5			
				2	16	2	12	2	12	360	34	7	C			6			
					17				5	90	13	1	E	E		0			
					18				8	270	22	5	A	E		4			
					19				10	360	27	6	B			5			
					20				6	90	15	1	E	E		0			
					21				9	360	25	5	A	E		4			
					22				7	90	18	1	E	E		0			
					23				0	360	1	4	A	D		3			
					24				10	90	28	5		E		4			
					25				8	180	21	1	E	E		0			
					26				11	360	31	6	B	E		5			
					27				7	180	19	5	A			4			
					28				9	270	24	4		D		3			
					29				1	360	3	3	B			2			
					30				12	90	35	6		E		5			
					31				10	180	26	4	A	E		3			
					32				2	360	6	2	C			1			
					33				4	90	11	7		E		6			
					34				6	180	16	6	B			5			
					35				11	270	30	3		D		2			
					36				14	90	39	9		F		7			
					37				16	180	42	8		D		3			
					38				15	180	41	9		F		7			
					39				13	270	36	8		D		3			
					40				16	270	43	9		F		7			
					41				14	360	38	8		D		3			
					42				13	360	37	9		F		7			
					43				15	90	40	8		D		3			

● VERTEX
 ▽ CIRCUIT
 ○ REGION



REGION
 LABELS

G(U):

E = 0
 A = 3
 B = 2
 C = 1

G(L):

E = 0
 D = 1
 E = 2
 F = 3

Figure 2.37 THE OVERLAY GRAPH G0.

Algorithm 2.7 Attach adjacent region labels to graph G_0 .

TO ASSIGN UPPER AND LOWER GRAPH REGION LABELS TO ALL NON COLOURED EDGES AND ALL REGIONS OF THE OVERLAY GRAPH G_0 .

GIVEN: 1) The extended PDV description of two vertically adjacent planar graphs, referred to as the upper graph $G(u)$ and the lower graph $G(l)$, that describe vertically adjacent building sections, and

2) an extended PDV description (including those lists describing the graphs' boundary circuits, regions and Parts, as well as those lists described in the previous algorithm) of the graph G_0 , describing the union of the graphs $G(u)$ and $G(l)$, where, those elements of the colour list $GORU$ (identifying the region of the upper graph $G(u)$ that lies immediately to the right of an exit edge of the graph G_0) OR those elements of the colour list $GORL$ (identifying the region of the lower graph $G(l)$ that lies immediately to the right of an exit edge of the graph G_0), which are associated with the set of exit edges of the boundary circuit(s) of each region of the graph G_0 , WILL on input to this algorithm be either; FULLY COLOURED every exit edge of the regions' boundary circuit(s) is assigned a colour label (the list elements of $GORL$ of the exit edges of region A in Figure 2.38); PARTIALLY COLOURED at least one but not every exit edge of the regions' boundary circuit(s) is assigned a colour label (the elements of $GORU$ and $GORL$ of the edges of

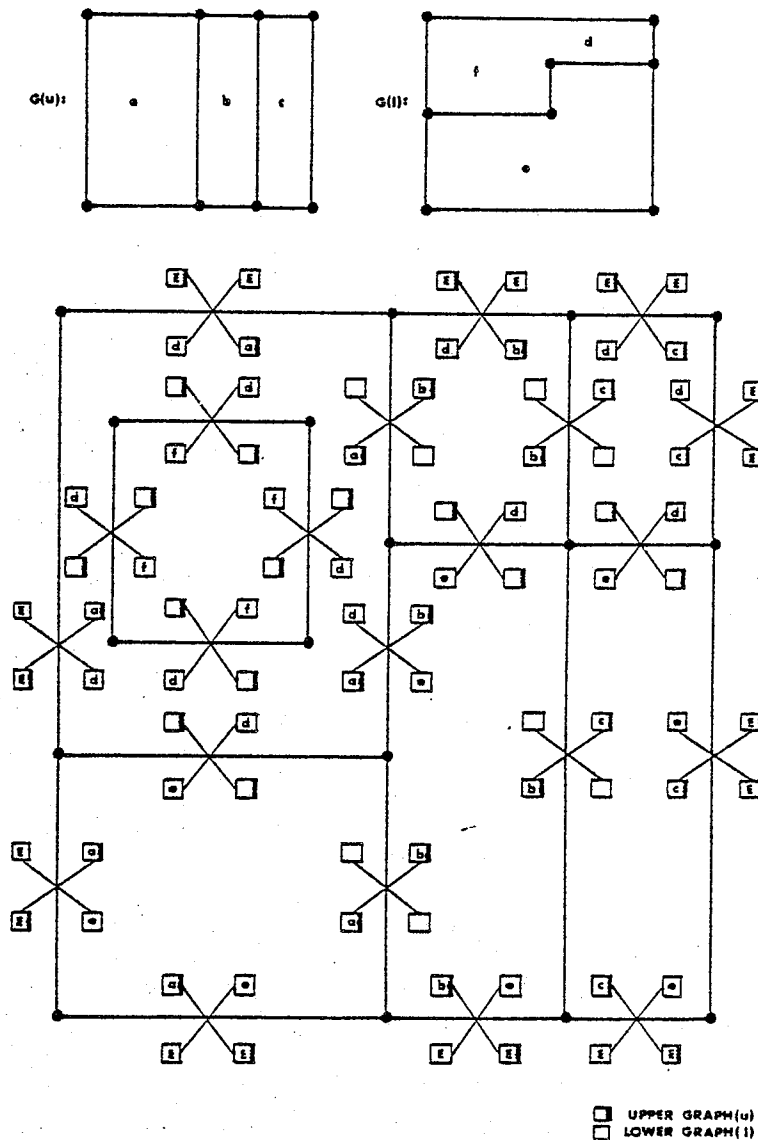


Figure 2.38 COLOURING THE OVERLAY GRAPH G_O .

regions B and C, respectively, in Figure 2.38); or,
NOT COLOURED no exit edge of the regions' boundary
circuit(s) is assigned a colour label (the elements of

GORU of the edges of region A in Figure 2.38);

and, upon completion of this algorithm, will all be FULLY COLOURED.

and DEFINING the lists RULINK and RLLINK, where;

- 1) the lists RULINK and RLLINK define the region of the upper and lower parent graphs $G(u)$ and $G(l)$, respectively, that is contiguous vertically to each region of the graph G_0 .

NOTE. THESE TWO LISTS DESCRIBE THE VERTICAL EDGES IN THE NETWORK REPRESENTATION THAT IS USED TO DESCRIBE THE BUILDING (see Figure 2.39).

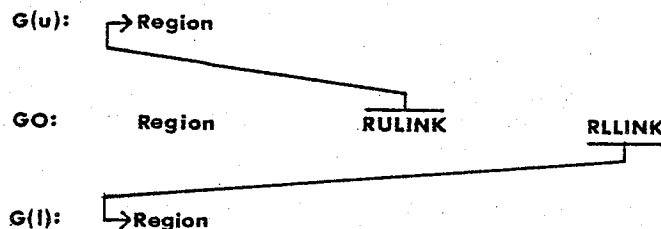


Figure 2.39 VERTICAL NETWORK LINKAGE LISTS: GRAPH G_0 .

STEP 1: ENUMERATE THE UPPER COLOUR OF THE REGIONS OF THE GRAPH G_0 .

- 1.1) Select in turn each region from R of the graph G_0 ,

nominating the region selected as the current REGION, and perform the following for the current REGION;

A) Determine whether the current REGION is FULLY, PARTIALLY or NOT COLOURED: initialize the values COUNT and STATUS to 0 and COLOUR to 'not assigned'; traverse in turn each exit edge of each boundary circuit of the current REGION, nominating the selected edge as the current SUBJECT EDGE, and perform the following for each SUBJECT EDGE;

1) Examine GORU element associated with the current SUBJECT EDGE and,

a) if the exit edge is coloured; increment COUNT and STATUS; and, if COLOUR is 'not assigned', assign to COLOUR the colour of the current SUBJECT EDGE.

b) if the exit edge is not coloured; increment COUNT.

B) Determine the colour of the current REGION if that region is NOT COLOURED: examine the value STATUS, and if STATUS is equal to zero (ie. none of the exit edges of the current REGION are assigned a colour in their GORU list), perform the following;

1) Arbitrarily select one vertex of the current REGION and nominate the selected vertex as the current TARGET; examine in turn each region of the graph $G(u)$, nominating the region selected as the current OBJECT REGION, and determine whether or not the current TARGET lies wholly within the current OBJECT REGION; and,

a) if the TARGET does not reside wholly within the

current OBJECT REGION; return to B.1 to examine the next region of the graph $G(u)$.

b) if the TARGET does reside wholly within the current OBJECT REGION; assign COLOUR the region label identifying the current OBJECT REGION of the graph $G(u)$; and, proceed to C.

C) Assign the current COLOUR to the current REGION; enter the current COLOUR into that element of RULINK that is associated with the current REGION; compare the values COUNT and STATUS and,

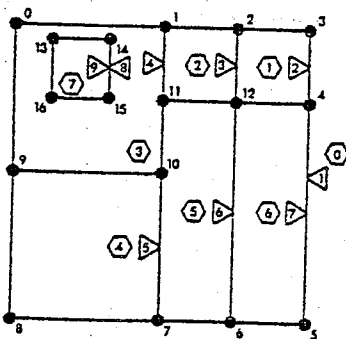
1) if the values of COUNT and STATUS are equal (ie. the current REGION is FULLY COLOURED), return to 1.1 to examine the next region of the graph G_0 .

2) if the values of COUNT and STATUS are not equal (ie. the current REGION is either PARTIALLY or NOT COLOURED), traverse each exit edge of each boundary circuit of the current REGION and, while traversing each edge, enter the current COLOUR into that element of RULINK that is associated with the exit edge; and, return to 1.1 to examine the next region of the graph G_0 .

STEP 2: ENUMERATE THE LOWER COLOUR OF THE REGIONS OF THE GRAPH G_0 .

2.1) Perform the operations specified in step 1 substituting the terms $G(1)$, GORL, and RLLINK for the terms $G(u)$, GORU, and RULINK, respectively.

C	D	Z	P	P'	D	V	B	V T A G	V C L R			G O R U	G O R L	V R	R	R U	R U I N K	R U I N K
9	0	1	1	0	2	1	90	4	4			A	D	3	1	1	0	0
8	1	1	1	8	3	9	180	23	1			E.	E.	0	1	2	1	1
6	2	1	1	16	3	2	90	7	3			B	D	2	2	3	1	3
3	3	1	1	16	2	11	180	29	4			A	D	3	2	3	2	2
24	4	1	1	12	3	0	270	0	1			E.	D	0	1	3	2	2
19	5	1	1	0	2	3	90	8	2			C	D	1	1	3	2	2
16	6	1	1	12	3	12	180	32	3			B	D	2	1	1	3	3
39	7	1	1	0	3	1	270	2	1			E.	E.	0	1	1	1	3
38	8	1	1	0	2	4	180	12	2			C	D	1	1	1	1	3
	9	1	1	8	3	2	270	5	1			E.	E.	0	9			
	10	1	1	8	3	5	180	14	7			C	E.	6				
	11	1	1	12	3	12	270	33	2			C	D	1				
	12	2	2	12	4	3	360	8	1			E.	E.	0				
	13	2	2	15	2	6	270	17	7			C	E.	6				
	14	2	2	5	2	4	360	10	1			E.	E.	0				
	15	2	2	5	2	7	270	20	6			B	E.	5				
	16	2	2	2	2	12	360	34	7			C	E.	6				
	17					5	90	13	1			E.	E.	0				
	18					8	270	22	5			A	E.	4				
	19					10	360	27	6			B	E.	5				
	20					6	90	15	1			E.	E.	0				
	21					9	360	25	5			A	E.	4				
	22					7	90	18	1			E.	E.	0				
	23					0	360	1	4			A	D	3				
	24					10	90	28	5			A	E.	4				
	25					8	180	21	1			E.	E.	0				
	26					11	360	31	6			B	E.	5				
	27					7	180	19	5			A	E.	4				
	28					9	270	24	4			A	D	3				
	29					1	360	3	3			B	D	2				
	30					12	90	35	6			B	E.	5				
	31					10	180	26	4			A	E.	3				
	32					2	360	6	2			C	D	1				
	33					4	90	11	7			C	E.	6				
	34					6	180	16	6			B	E.	5				
	35					11	270	30	3			B	D	2				
	36					14	90	39	9			A	F	7				
	37					16	180	42	8			A	D	3				
	38					15	180	41	9			A	F	7				
	39					13	270	36	8			A	D	3				
	40					16	270	43	9			A	F	7				
	41					14	360	38	8			A	D	3				
	42					13	360	37	9			A	F	7				
	43					15	90	40	8			A	D	3				



REGION
LABELS

G(U) :

E = 0
A = 3
B = 2
C = 1

G(L):

E = 0
D = 1
E = 2
F = 3

Figure 2.40 THE COLOURED OVERLAY GRAPH G₀.

Algorithm 2.8 Enumerate all vertically contiguous regions

TO ENUMERATE THOSE REGIONS OF THE OVERLAY GRAPH G_0 THAT ARE CONTIGUOUS VERTICALLY TO EACH REGION OF THE UPPER AND LOWER GRAPHS $G(u)$ and $G(l)$, RESPECTIVELY.

- GIVEN: 1) The extended PDV description of the overlay graph G_0 , describing the union of two vertically adjacent building plan sections, and
- 2) the extended PDV graph description (as developed to this juncture) for those two vertically adjacent building sections, referred to as the upper graph $G(u)$ and the lower graph $G(l)$.

and DEFINING the following lists, where;

- 1) RULINK and RLLINK define the region(s) of the overlay graph G_0 that are contiguous vertically to each region of the adjacent plan section graphs $G(u)$ and $G(l)$.
 - 2) RUN and RLN define the number of regions of the overlay graph G_0 that are contiguous vertically to each region of the graphs $G(u)$ and $G(l)$, respectively. (see Figure 2.42)
- 1.1) Enumerate those regions of the overlay graph G_0 that are contiguous vertically to each region of the graph $G(u)$: select in turn each region from R of the graph $G(u)$ and nominate the regions as selected as the current SUBJECT REGION; set COUNT equal to zero and;
- A) Select in turn each region from R of the overlay graph G_0 , until all elements of that list are exhausted, and nominate

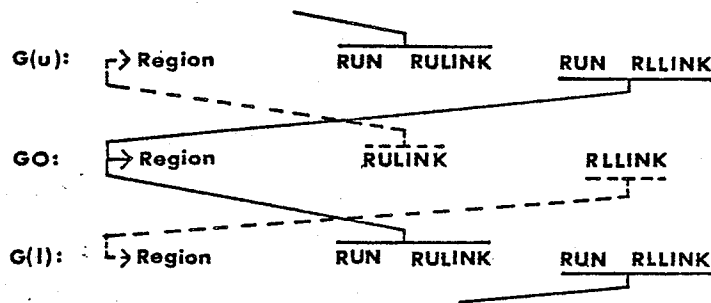


Figure 2.41 VERTICAL NETWORK LINKAGE LISTS: $G(U)$ AND $G(L)$.

the regions as selected as the current OBJECT REGION; obtain the region label from the element of RULINK of the graph G_O that identifies the region of the graph G that is directly above the current OBJECT REGION and nominate that region label as the current COLOUR; compare the current COLOUR and the current SUBJECT REGION and;

- 1) if COLOUR and SUBJECT REGION are not equal (ie. the current SUBJECT REGION and OBJECT REGION are not vertically contiguous); return to select the next region of the graph $G_O(u)$.
- 2) if COLOUR and SUBJECT REGION are equal (ie. the current SUBJECT REGION and OBJECT REGION are vertically contiguous); enter the OBJECT REGION into the next

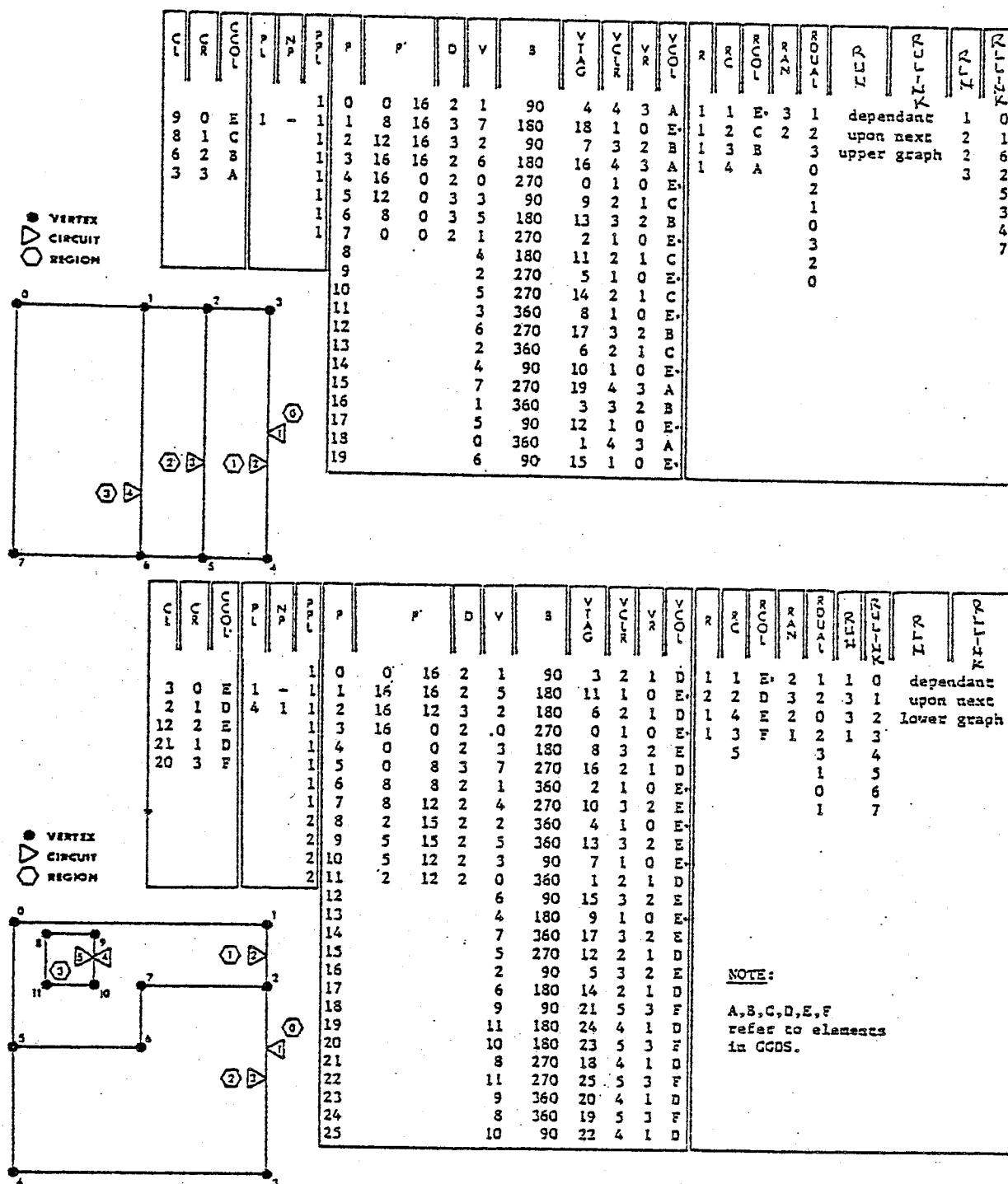


Figure 2.42 VERTICALLY CONTIGUOUS REGIONS OF THE GRAPH G.

available element of RLLINK of the graph $G(u)$; increment the value COUNT; and, return to select the next region of the graph G_0 .

B) Enter the value COUNT into the next available element of RUN of the graph G and return to select the next region of the graph G .

1.2) Enumerate those regions of the lower graph $G(1)$ that are vertically contiguous to each region of the overlay graph G_0 : perform those operations specified in 1.1 substituting the terms $G(1)$, RULINK, and RUN for the terms $G(u)$, RULINK, and RLN, respectively, wherever they may occur.

Summary

The construction of the NETWORK representation for a building may be executed by inputting a series of plan section drawings depicting the Storey and Floor/Ceiling sections of a building and a plan section drawing depicting the immediate physical environs of that build design. This is accomplished, by converting each of these plan section drawings into a coloured planar graph representation, and by recursively applying the above described planar graph and planar graph overlay enumeration procedures, as depicted below in Figure 2.43.

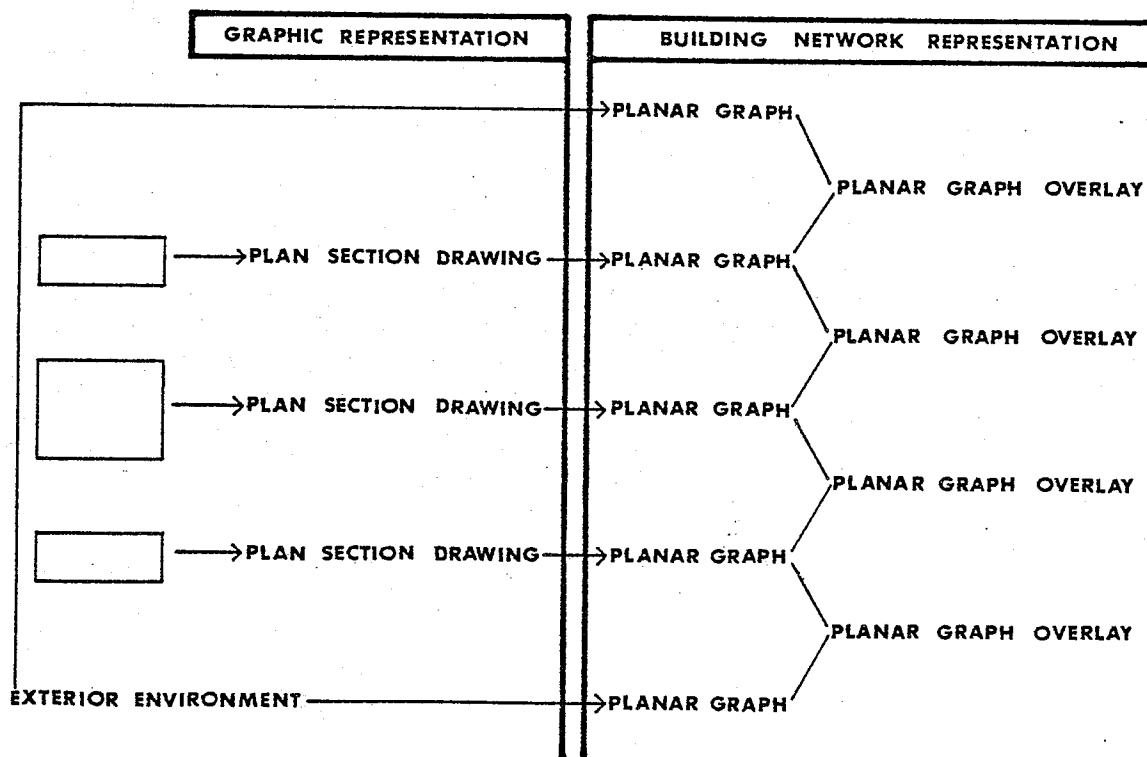


Figure 2.43 THE BUILDING NETWORK CONSTRUCTION PROCESS.

References: Chapter 2

- 1 Newman, William M., and Sproull, Robert F.. Principles of Interactive Computer Graphics. New York: McGraw-Hill Book Company, 1973. p.p. 607.
- 2 ibid. p.p. 105-108..
- 3 Symons, D. C.. A Parcel Geocoding System for Urban and Rural Information. National Capital Commission, Ottawa, September 23, 1970. p.p. 253-269.
- 4 Paton, K., "An Algorithm for the Blocks and Cutnodes of a Graph". Communications of the A.C.M.: vol. 14, no. 7, July 1971. p.471.

October 1979

3

THE GRAPHIX SYSTEM

CHAPTER 3: THE GRAPHIX SYSTEM

Introduction

In the previous chapter explicit methods were developed to assemble and store the network representation of a building in the computer. This chapter describes the GRAPHIX interactive computer graphics system for building description that experimentally implemented those explicit methods and, in conclusion, briefly illustrates the subsequent thermal analysis of a specific building design that was digitally described using the GRAPHIX system.

The GRAPHIX System for Building Description

The GRAPHIX system for building description provides a single designer, working directly on the screen of a suitably configured interactive computer graphics workstation, with the tools necessary to construct a series of plan section drawings describing the design of a building and its immediate physical surroundings. Using the information supplied to it directly and indirectly through these plan section drawings, the system will automatically generate a three-dimensional network model of the building design. This network model of the building may be subsequently used by various other computer aided design procedures to inspect or

otherwise evaluate that buildings design.

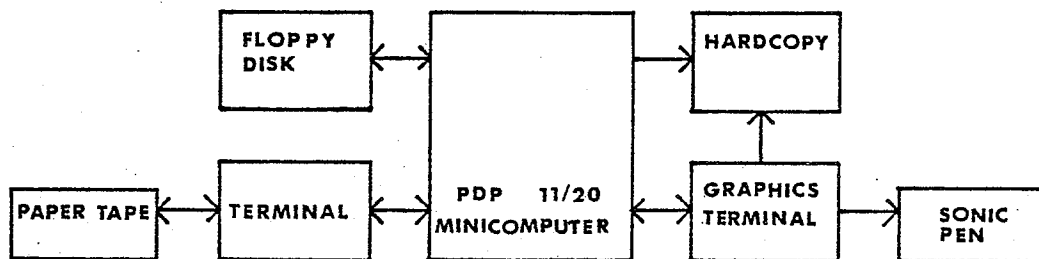


Figure 3.1 INTERACTIVE COMPUTER GRAPHICS WORKSTATION.

The GRAPHIX program was written entirely in PDP 11 assembler code. The code was written expressly for execution on an interactive computer graphics workstation that was configured from equipment made available by the Computer Services Centre, University of Manitoba. This interactive computer graphics workstation was configured (Figure 3.1) using a PDP 11/20 mini-computer with 16K words of core memory, a single floppy disk mass storage unit, a Tektronix 4002A graphics display terminal, a

sonic pen (affixed about the 4002A screen*) and a Tektronix 1610 hardcopy unit.

The GRAPHIX Sub-systems

The GRAPHIX system is composed of four (4) distinct sub-systems; the GRAPHIX MONITOR Sub-system, the BUILDING COMPONENT Sub-system, the PLAN SECTION DRAWING Sub-system (composed of three drawing sub-systems, the Storey, the Horizontal, and the Exterior Obstruction sub-systems), and the BUILDING MODEL GENERATOR Sub-system. Each of these sub-systems is designed to serve a specific information acquisition task and to operate on one or more of the three major data bases used to store the description for the building design (Figure 3.2). Descriptions for each of these Sub-systems are presented below.

The GRAPHIX MONITOR Sub-system

The GRAPHIX MONITOR Sub-system provides basic data management services to the GRAPHIX system. These services include initialization of all building data bases during system start-up, management of all sub-system state changes, and basic file management facilities for operating on the GRAPHIX Graphical Data Base. Entry

* The microphones of the sonic pen were affixed about the 4002A screen to simulate the operation of a 'light pen'. This arrangement allows the designer to point directly at specific elements in the display and to communicate information to the GRAPHIX system.

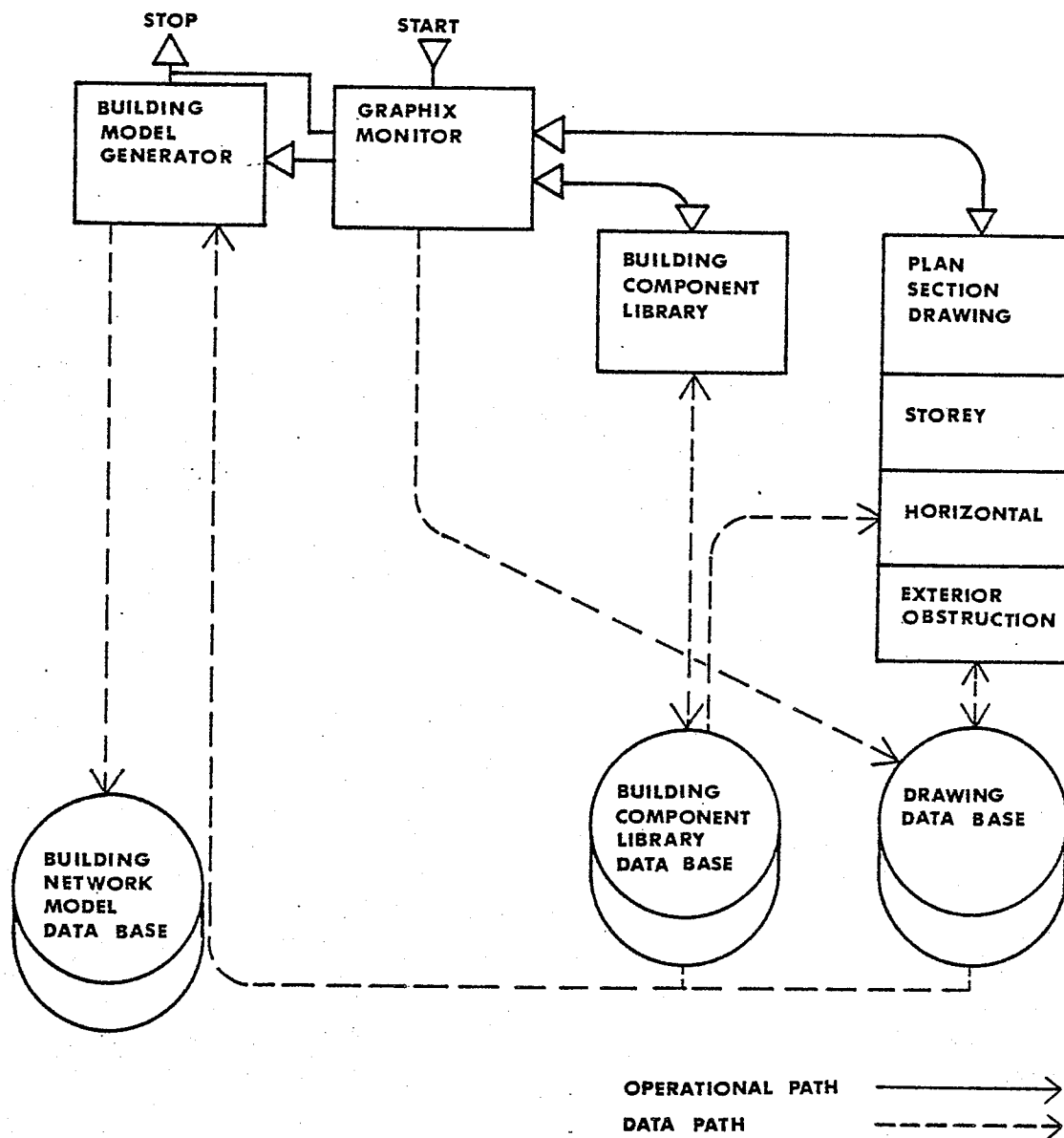


Figure 3.2 THE GRAPHIX SYSTEM.

to the MONITOR is achieved automatically upon system start-up, and upon exit from the Building Component and Plan Section Drawing sub-systems. Residence in the MONITOR is signified by the display

depicted in Figure 3.3*. When the MONITOR sub-system is exited the designer will be asked if the Building Model Generator sub-system is to be invoked. Should the response to this query be affirmative, the network model for the building design will first be assembled and the GRAPHIX system then shut down. Should the response be negative, the exiting of the MONITOR sub-system will lead to the immediate shut down of the GRAPHIX system.

Operation

Once start-up procedures for the system have been executed, the MONITOR will begin the initialization process, interrogating both the floppy disk and designer regarding the job name, job number, and status of the current description task (ie. whether the resident floppy disk has or has not been used previously to store a description of the current building design). Should the information resident on the disk differ from that received from the designer, the MONITOR will request that the designer reload the disk cartridge with the appropriate diskette. Otherwise, the MONITOR will proceed to initialize all building data bases, if the description is 'new', and await further instructions from the designer.

Following the above described initialization process, either of the remaining two management services provided by the MONITOR may

* It should be noted that all the GRAPHIX System display figures presented in this chapter are derived directly from photographs taken while the Graphix System was in operation. For presentation purposes the polarity of the black and white fields of the screen display have been reversed.

be invoked as required by the designer.

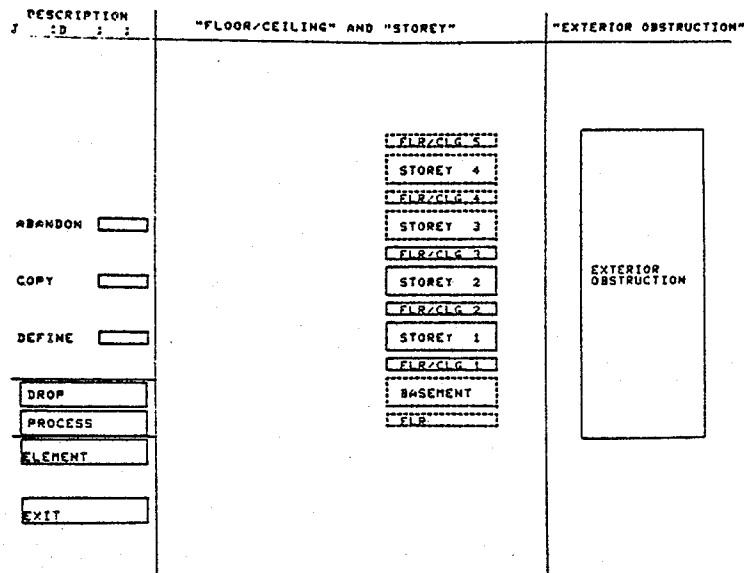


Figure 3.3 THE GRAPHIX SUB-SYSTEM DISPLAY.

Should the designer choose to change the sub-system state in order to proceed with the building description process, three (3) options present themselves.

OPTION 1: Should the designer wish to enter, modify, or inspect the Building Component Data Base, the Building Component Sub-system should be accessed. Entry to this sub-system is gained simply by pointing to the ELEMENT and then the PROCESS command of the menu list located

at the left of the MONITOR display (see Figure 3.3). The MONITOR will mark all commands with an X as they are input and then initiate the requested change in system state.

OPTION 2: Should the designer wish to create a new, or modify or inspect an existing, plan section drawing of the building design*, then one of the drawing sub-systems of the Plan Section Drawing sub-system should be accessed. Entry to either the Storey, Horizontal, or Exterior Obstruction drawing modules of this sub-system is simply gained by again pointing at the appropriate module box (six boxes representing the six available drawing files for the Horizontal module, five boxes representing the five available files for the Storey module, and a single box representing the only file available for the Exterior Obstruction module) and then the DEFINE and PROCESS commands. Should the particular drawing file being accessed not contain any drawing information, the MONITOR will query the designer regarding the vertical dimensions of the section being entered, by displaying a request for section height at the bottom of the screen. Once the designer has typed

* The lines forming the boxes representing each drawing file will be solid if the file has been previously entered and it contains drawing information, or the box will be formed by dashed lines indicating that no drawing information has yet been entered into that file of the GRAPHIX Graphical Data Base.

in and verified the vertical dimension for the section, the MONITOR will initiate the requested change in sub-system state.

OPTION 3: Should the designer wish to discontinue the description process for the current building design, he or she need only point at the EXIT command at the bottom of the menu list. At this juncture the MONITOR will query the designer to ascertain whether or not the Building Model Generator is to be invoked. Should the designer not wish to construct the network model of the building - the building description may not be complete - a NO response will result in the immediate shut-down of the GRAPHIX system. If a YES response is typed on the graphics terminal in response to the query, then the MONITOR will proceed to initiate the Model Generator Sub-system. Once the three-dimensional digital description of the building is assembled the GRAPHIX system will terminate operation.

Finally, should the designer choose to abandon (delete) a particular drawing file from the Drawing Data Base, or copy one drawing file into another empty drawing file, then the designer should invoke one of the two basic file management commands provided at the top of the Monitor's menu list. To invoke the ABANDON command the designer must point at that command, the appropriate drawing file, the PROCESS command, and then respond affirmatively to a MONITOR request to verify the command action. To

invoke the COPY command, the designer must first point at the file to be copied from, then the file into which the first file will be copied (this file must be empty), and the PROCESS command. Again, upon receipt of an affirmative response to a verification request, the appropriate copy will be executed by the MONITOR. Non-affirmative responses to the verification request will, in both of the above cases, cause the command to be ignored.

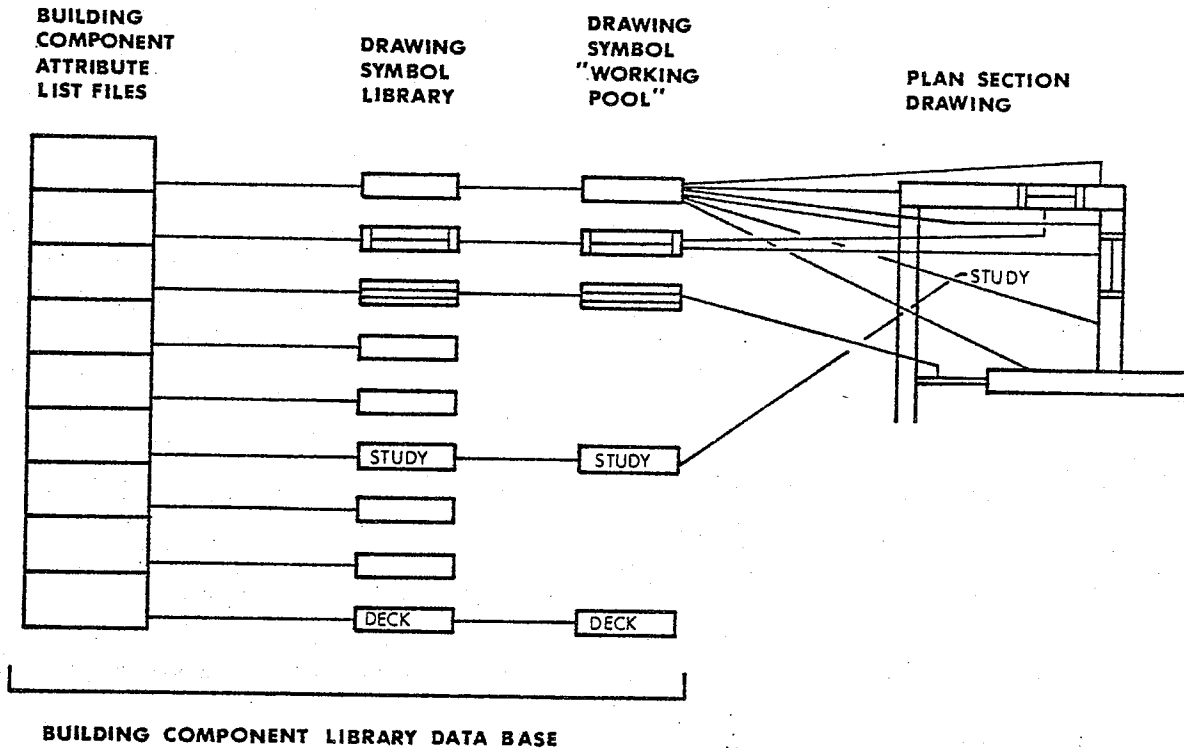


Figure 3.4 THE BUILDING COMPONENT DATA BASE.

The BUILDING COMPONENT Sub-system

Instances of standard building components (ie. door, windows, wall constructions, etc.) are often repeatedly used at different locations in a building design. Because it would be redundant and clearly uneconomical to store attribute information for each standard component with every sub-picture instance, it is preferable to group the attribute information for each component in one location (data base) and to provide cross-references to that information from each sub-picture instance.

In the GRAPHIX system for building description (see Figure 3.4), the attribute information for each building component is stored in the BUILDING COMPONENT LIBRARY data base under separate file and the attribute information contained in each of these files is cross-referenced to each instance in the building design through the plan section drawing symbols provided to delineate and represent those building components.

The BUILDING COMPONENT Sub-system is used 1) to delete drawing symbols from and, to re-instate drawing symbols into, the working pool of drawings symbols available to the Plan Section Drawing sub-system and, 2) to edit the attribute files, describing the physical properties of each building component, that are stored in the Building Component Library data base for each of the drawing symbols. Entry to this sub-system is executed via the Monitor sub-system and residency in the sub-system is signified by the display depicted in Figure 3.5. The exiting of this sub-system will

COMPONENT TABLE : : D : :	SUB-PICTURES	12" SCALE	8" SCALE	4" SCALE
ABANDON		WALL * 1	PART	PART
		WALL * 2	STUD	
COPY		WALL * 3	WALL * 3	
		WALL * 4		
DEFINE		WALL * 5		
		SGL PANE	SGL PANE	
		DBL PANE	DBL PANE	
DROP		TPL PANE	TPL PANE	
PROCESS		DOOR	DOOR	DOOR
EXIT				

Figure 3.5 THE BUILDING SUB-SYSTEM DISPLAY.

cause program control to revert back to the GRAPHIX Monitor sub-system.

Operation

When the description process for a building description is initiated, the GRAPHIX system will automatically: assign all drawing symbols available in the Plan Section Drawing sub-system to the 'working pool' of drawing symbols (the twenty-seven material component graphic symbols depicted in the COMPONENT sub-system display, twelve material component textual symbols such as ROOF, EXTERIOR DECK and CONIFEROUS TREE, and twenty-three spatial

component textual symbols such as KITCHEN, PANTRY and BEDROOM); and, enter into the Building Component Library data base a predefined set of attribute values (ie. thermal conductivity, reflectance and absorbancy for the material component symbols) for each of the symbols in that 'working pool' (Figure 3.5). To modify either the drawing elements contained in the working pool, and in so doing modify the drawing symbols available to the Plan Section Drawing sub-system, or to modify the values contained in any attribute list, one or more of the following command sequences must be performed.

COMMAND SEQUENCE 1: To abandon a drawing element from the current Plan Section Drawing symbol 'working pool'.

- 1) Point at the ABANDON command of the menu list, point at the appropriate drawing symbol on the screen if the symbol to be deleted is not textual, and then point at the PROCESS command.
- 2) If the symbol to be abandoned is textual, the system will display a request at the bottom of the screen, asking that the designer type in the textual name of the symbol.
- 3) The system will now check the Building Component Library data base to ensure that the selected drawing symbol is not currently being used in any plan section drawing for the building. If the symbol has been located in a plan section drawing the symbol will not be abandoned by the system.
- 4) The system will finally request that the designer verify that this ABANDON command is to be executed. Once verification is received, the system will perform the requested deletion and

exit this command sequence.

COMMAND SEQUENCE 2: To re-instate a drawing symbol into the Plan Section Drawing sub-systems 'working pool' of symbols, and/or to define attribute values for any of the building components.

- 1) Point at the DEFINE command of the menu list, at the appropriate drawing symbol on the screen if the symbol to be defined is not textual, and then at the PROCESS command.
- 2) The system will request that the user type in the name of the symbol to be defined. If a graphic symbol is being defined and the designer types in a name rather than a null response, then that name will be displayed in conjunction with the symbol in the Plan Section Drawing menu list.
- 3) At this juncture the system will ask the designer if a listing of the existing attribute values for the selected drawing symbol is required. If such a request is made, the system will display this list at the terminal.
- 4) The system will now query the designer in order to determine if new attribute values are to be entered.
- 5) If the designer chooses to edit any of the existing values, the system will query the designer regarding the sequence number of the data item to be changed and the value to be substituted in its place. Otherwise the command sequence is exited.
- 6) Once the data item number and substitute value have been entered on the terminal keyboard and the system has executed the edit, the designer will be asked if any other attribute value changes are to be made.

7) The system will iterate through this editing sequence until the designer indicates that no further changes are to be made. At this juncture the command sequence will be exited.

COMMAND SEQUENCE 3: To copy the attribute file for one building component into the attribute file of another building component.

1) Point at the COPY command of the menu list, and if the building component is depicted on the display, at the building component to be copied from and then at the building component to be copied into, and finally at the PROCESS command.

2) If no graphic symbols were selected prior to entry of the PROCESS command, the system will request that the designer type in the name of the textual symbol to be copied from and the name of the textual symbol to be copied into.

3) The system will finally request that the designer verify that this copy is to be executed. Once verification is received, the system will perform the requested copy and exit this command sequence.

The PLAN SECTION DRAWING Sub-system

The PLAN SECTION DRAWING Sub-system is used to 'draught' an alternating series of Storey and Ceiling/Floor plan section drawings depicting the building design itself and an Exterior Obstruction plan section drawing depicting the immediate physical environs surrounding the building design. These plan section drawings are subsequently used by the Building Model Generator sub-system to construct the network representation of the building

design.

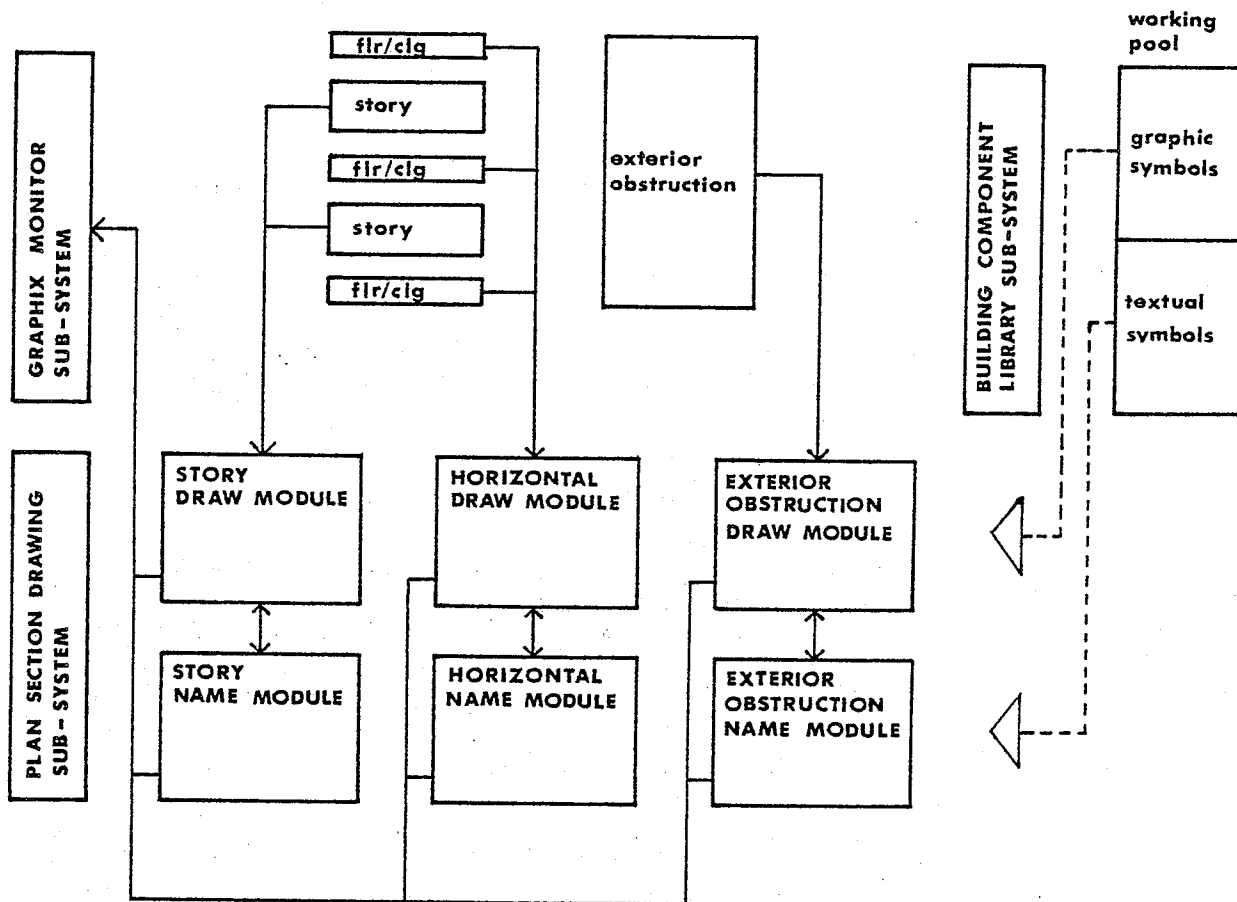


Figure 3.6 THE PLAN SECTION DRAWING SUB-SYSTEM.

As depicted in Figure 3.6, this Sub-system is comprised of three separate drawing sub-systems (the STOREY, the HORIZONTAL and the EXTERIOR OBSTRUCTION drawing sub-systems), and each of these drawing sub-systems is in turn sub-divided into a draw and a name module. Each of these module pairs is used to draught either the graphic (draw) or textual (name) symbols that are both appropriate

to the type of plan section drawing and are assigned by the designer to the 'working pool' of elements in the Building Component Library.

Entry into any one of the three available PLAN SECTION DRAWING sub-systems is achieved directly through the GRAPHIX Monitor, when the designer explicitly defines the drawing file to be worked on (STOREY 1). Entry will always be made into the draw module of the selected sub-system, from which the name module may be subsequently entered. Exiting from any of this sub-systems' draw or name modules will cause program control to revert back to the GRAPHIX Monitor sub-system.

Operation

Once residency in any one of the PLAN SECTION DRAWING sub-systems draw or name modules is established, the designer will be presented with an appropriate display similar in structure to that of either the STOREY DRAW module or the STOREY NAME module, as depicted in Figure 3.7. This display will sub-divide the working area of the graphics terminal into two functional regions, a menu list area to the left and a draughting area to the right. The menu list provides the designer with a symbol list of some of the building components available for the current drawing type and a simple set of draughting commands to assist in the manipulation and placement of those symbols in the draughting area.

The representation of a building plan section drawing is achieved by systematically entering drawing symbols into the draughting area of the display (Figure 3.8 (a)) and erasing

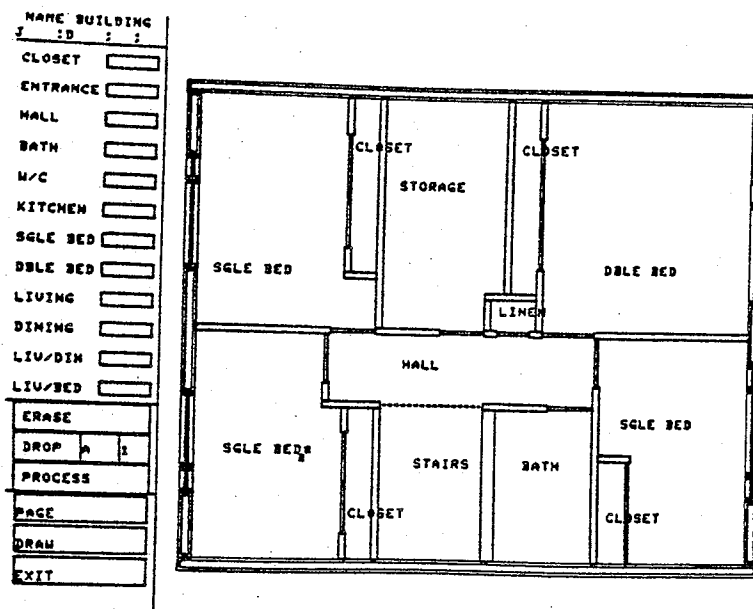
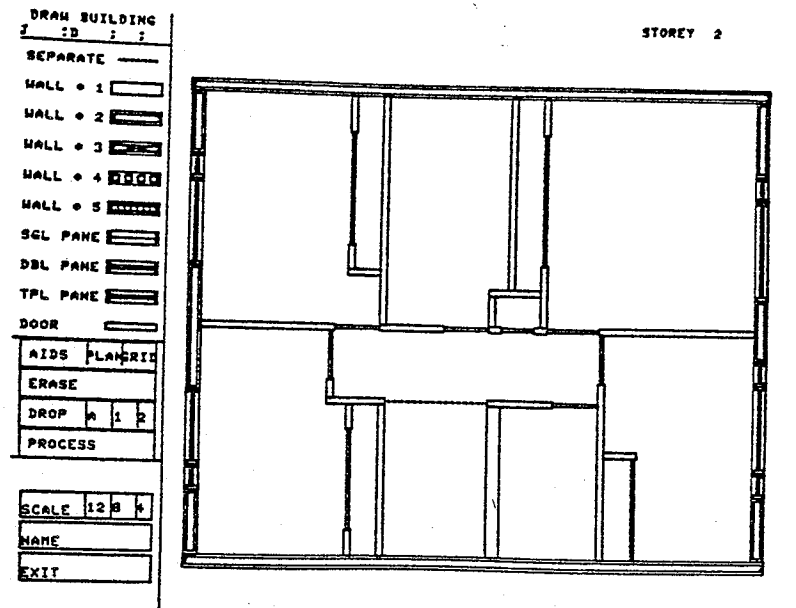


Figure 3.7 THE STOREY DRAW AND NAME MODULE DISPLAYS.

existing symbols from the display (Figure 3.8 (b)), until the desired plan section drawing is finally produced. The 'draughting' environment provides for the further sub-division of overall height

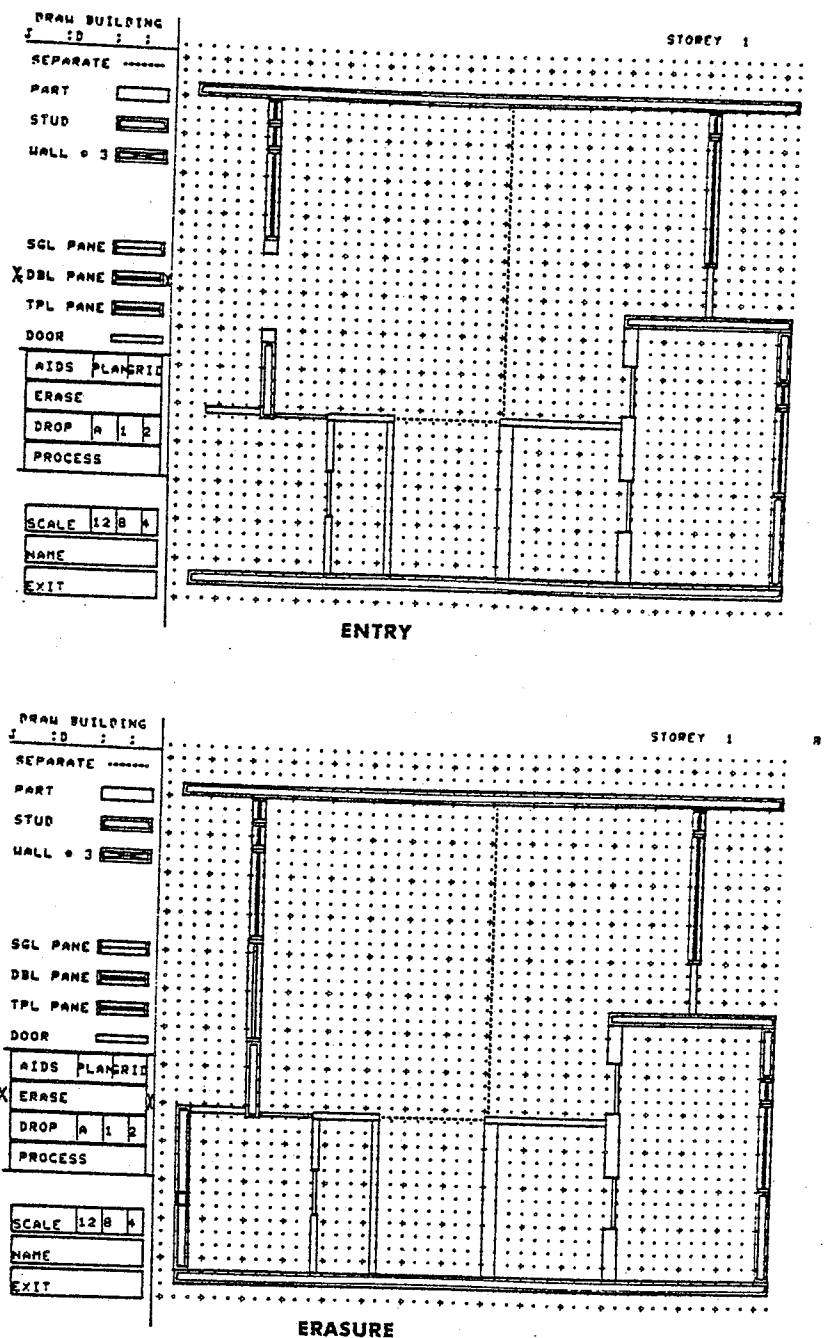


Figure 3.8 ENTRY AND ERASURE OF SYMBOLS FROM THE DISPLAY.

for building components such as doors and windows (sill and head heights) and the definition of the components situated below and/or above those elements.

Figure 3.9 through to 3.11 illustrate both the general sequence followed in the preparation of the plan section drawings describing a two-storey residence and some of the display options offered in the PLAN SECTION DRAWING sub-system modules.

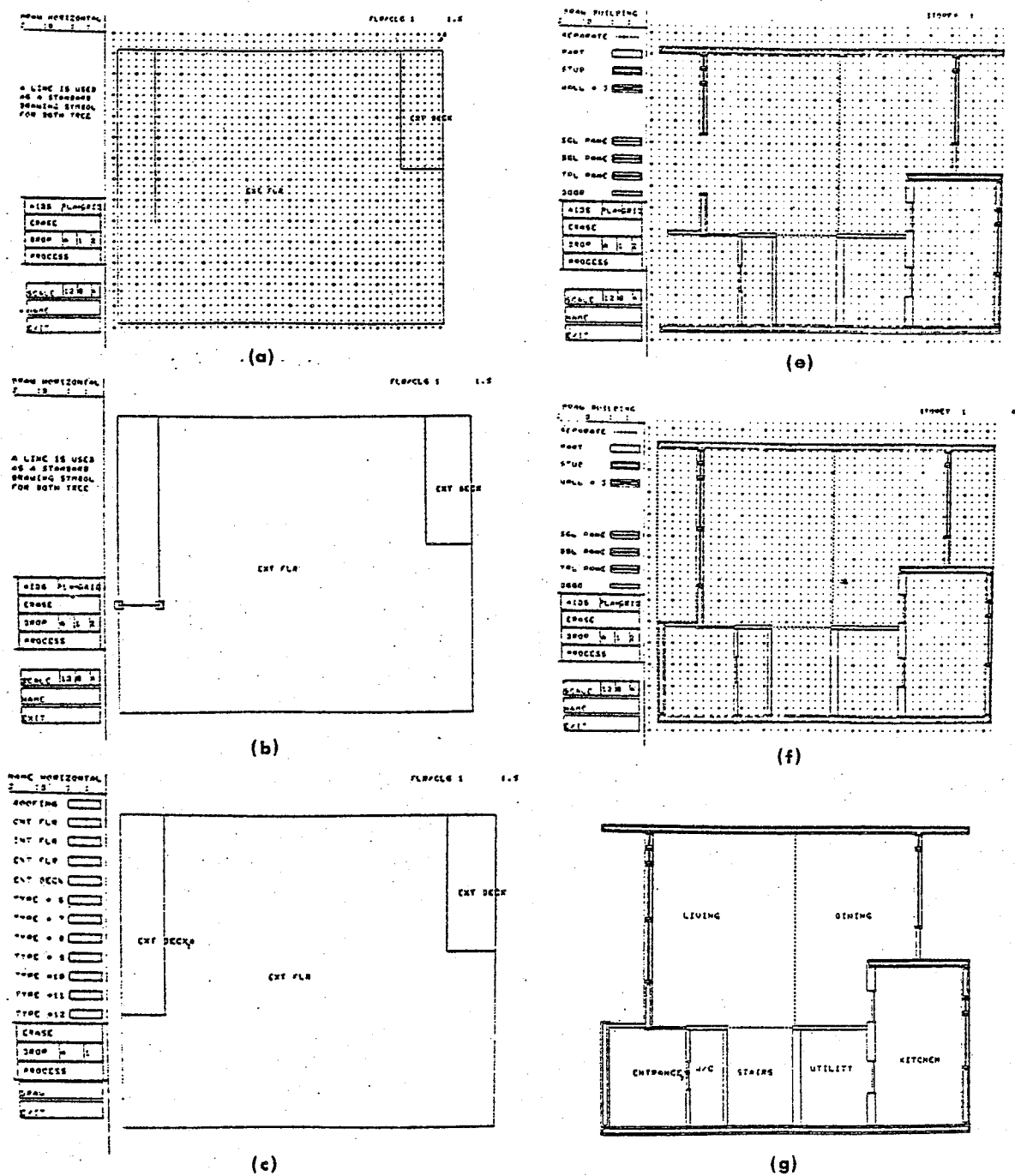


Figure 3.9 PLAN SECTION DRAWING ILLUSTRATIONS (1 of 3).

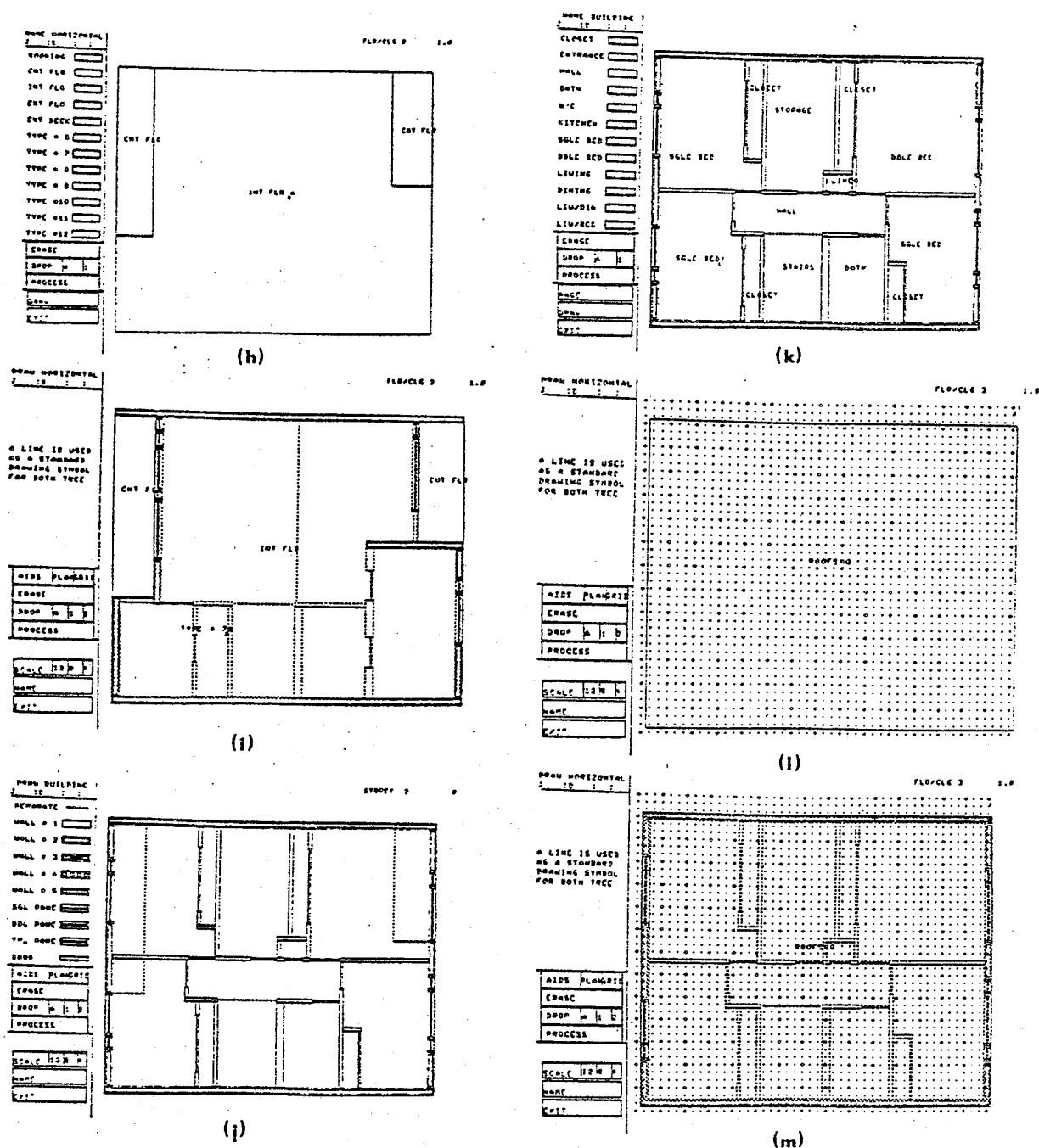
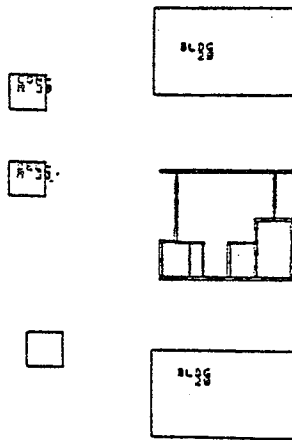


Figure 3.10 PLAN SECTION DRAWING ILLUSTRATIONS (2 of 3).

NAME OBSTRUCTION	
1	2
BUILDING	
TREE	DENSITY
CON. LIGHT	<input type="checkbox"/>
CON. MEDIUM	<input type="checkbox"/>
CON. HEAVY	<input type="checkbox"/>
DEC. LIGHT	<input type="checkbox"/>
DEC. MEDIUM	<input type="checkbox"/>
DEC. HEAVY	<input type="checkbox"/>
50 FT	<input type="checkbox"/>
40 FT	<input type="checkbox"/>
30 FT	<input type="checkbox"/>
20 FT	<input type="checkbox"/>
10 FT	<input type="checkbox"/>
ERASE	
DROP	A 1
PROCESS	
DRAW	
EXIT	

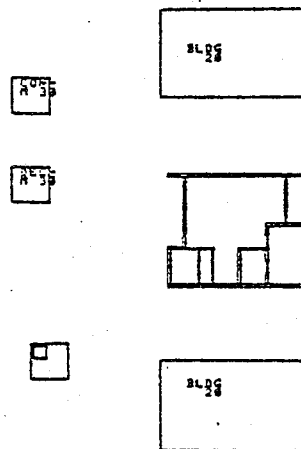
EXT. 0897



(n)

NAME OBSTRUCTION	
1	2
BUILDING	
TREE	DENSITY
CON. LIGHT	<input type="checkbox"/>
CON. MEDIUM	<input type="checkbox"/>
CON. HEAVY	<input type="checkbox"/>
DEC. LIGHT	<input type="checkbox"/>
DEC. MEDIUM	<input type="checkbox"/>
DEC. HEAVY	<input type="checkbox"/>
50 FT	<input type="checkbox"/>
40 FT	<input type="checkbox"/>
30 FT	<input type="checkbox"/>
20 FT	<input type="checkbox"/>
10 FT	<input type="checkbox"/>
ERASE	
DROP	A 1
PROCESS	
DRAW	
EXIT	

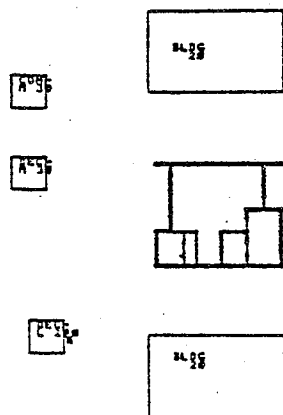
EXT. 0897



(o)

NAME OBSTRUCTION	
1	2
BUILDING	
TREE	DENSITY
CON. LIGHT	<input type="checkbox"/>
CON. MEDIUM	<input type="checkbox"/>
CON. HEAVY	<input type="checkbox"/>
DEC. LIGHT	<input type="checkbox"/>
DEC. MEDIUM	<input type="checkbox"/>
DEC. HEAVY	<input type="checkbox"/>
50 FT	<input type="checkbox"/>
40 FT	<input type="checkbox"/>
30 FT	<input type="checkbox"/>
20 FT	<input type="checkbox"/>
10 FT	<input type="checkbox"/>
ERASE	
DROP	A 1
PROCESS	
DRAW	
EXIT	

EXT. 0897



(p)

Figure 3.11 PLAN SECTION DRAWING ILLUSTRATIONS (3 of 3).

The Building MODEL GENERATOR Sub-system

The Building MODEL GENERATOR Sub-system is used to invoke a batch program that will construct the three dimensional network representation for the building design as developed in Chapter 2. Entry into this sub-system is executed via the Monitor sub-system. Once the batch network assembly program has ceased executing a message will be displayed at the terminal indicating the success or, alas, the failure of the assembly task. Upon the display of this task status message, the GRAPHIX system will cease operations.

Conclusion

Once the GRAPHIX system for building description had been implemented as described above, a simple* building design was entered into the computer using the GRAPHIX system and that building design analysed using an existing building thermal heat loss/gain computer program written by Mr. Alan Dakin (M. Arch. Thesis, University of Manitoba, 1973(1)). The building design used in this analysis and some of the results (see Appendix B for complete result listings) derived from this analysis are depicted together in Figure 3.12.

*This building heat loss/gain analysis program requires that the building be of a convex, rectangular, polyhedral form and that the polyhedral form possess but one interior 'zone'.

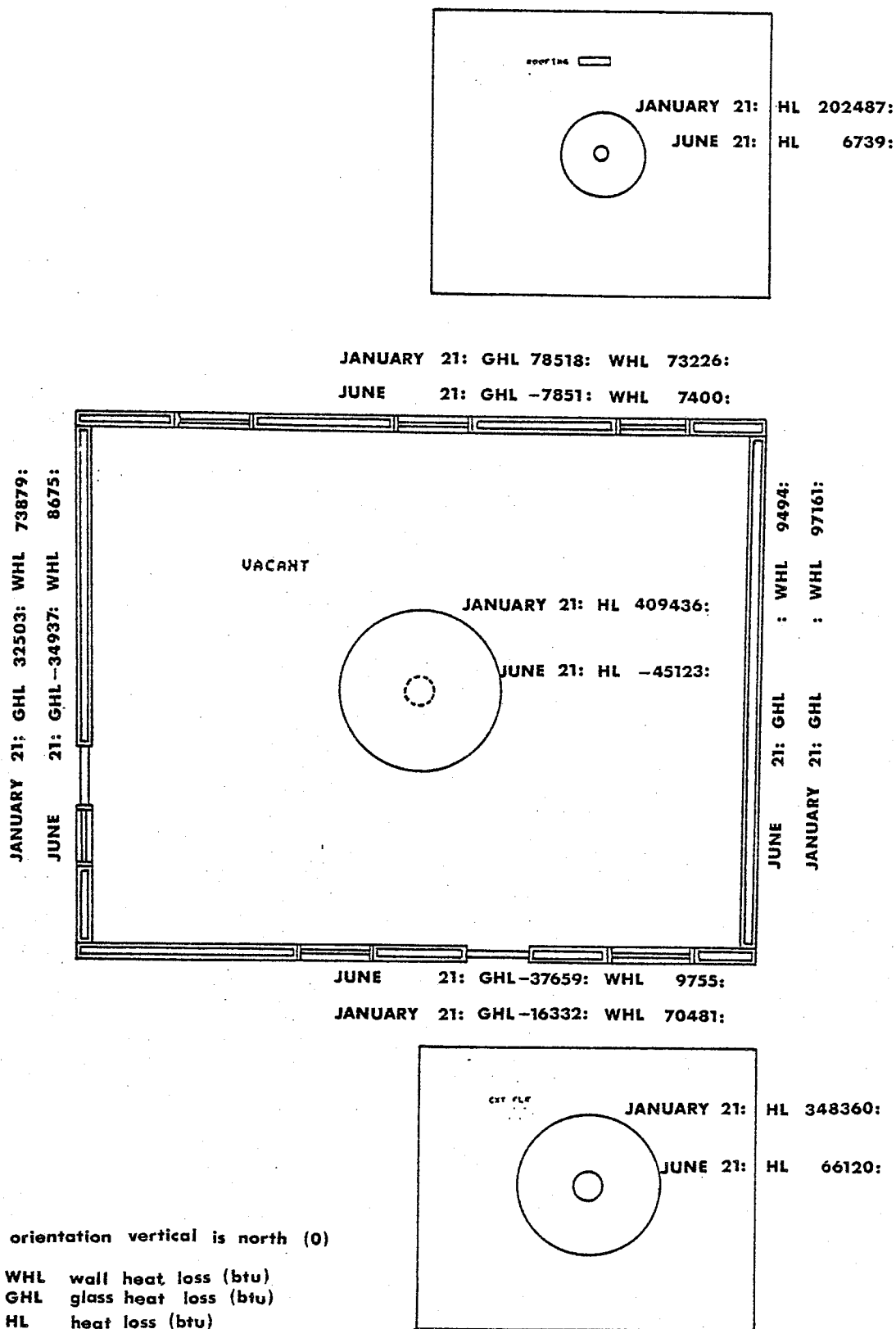


Figure 3.12 THE THERMAL ANALYSIS OF A BUILDING DESIGN.

References: Chapter 3

- 1 Dakin, R.A. Some Micro-climatic Effects on the Siting and Orientation of Buildings in a Northern Environment. M. Architecture Thesis, University of Manitoba, 1973.

4

THE CONCLUSIONS

CHAPTER 4: CONCLUSIONIntroduction

The objective of Computer Aided Building Design is to develop computerized procedures that will permit designers to apply more science and technology in a cost effective manner in the building design process. Once an initial design for a building is input within the computer a designer can quickly analyse and evaluate a variety of alternatives and thus make better design decisions. The primary justification for interactive computer graphics development is to provide a direct and economic means to input that building data which is required to operate computer aided building analysis and evaluation procedures.

The Explicit Methods and the GRAPHIX interactive computer graphics system presented in the previous chapters of this thesis describe one means that can be used to generate three dimensional digital models of building designs which can in turn be used to support the application of computer aided building analysis and evaluation procedures. This chapter presents some critical observations and conclusions regarding that means of building description.

Observations

The development of the Explicit Methods and the GRAPHIX System provided the author with an invaluable opportunity to acquire and exercise some of the theoretic knowledge and some of the technical skills requisite to the development of an useful interactive computer graphics based system for the direct machine encoding of building descriptions. Based upon the experience gained through the execution of this work, there are a number of points that can be made regarding both the Explicit Methods and the GRAPHIX System.

1.0) The Explicit Methods.

1.1) The planar graph, planar graph overlay, and dual graph network methods provide a very flexible and powerful system of representation for buildings composed of rectangular polyhedra. The system can be used not only to describe both the physical (material component) elements of a building such as window, wall, door and ceiling/floor constructions and the spatial (spatial component) elements such as rooms and the exterior environment but, most importantly, it will derive the geometric description for the latter from that of the former.

1.2) Because the system of representation closely parallels that traditionally employed in the preparation of standard architectural orthographic projection drawings, translation of the representation into familiar architectonic graphic

displays can be readily accomplished.

- 1.3) The system of representation is applicable to any scale of description and may be employed for any purpose which requires a complete and accurate geometric model of rectangular polyhedra.
- 1.4) The system of representation is restricted to the description of rectangular polyhedra located in continuous horizontal building sections and it cannot, at least as presently developed, be used to generate descriptions for important, non-rectangular, planar forms such as sloped or pitched roofs. Although this limitation poses few restrictions to the description of most commercial and industrial building types, it does restrict the system's application to the description of most types of residential building.

If this system of representation were to be further developed, serious consideration should be given to the development of special input, planar graph, planar graph overlay, and dual graph network methods which would accommodate the description of such building forms.

- 1.5) The system of representation was developed to operate as a 'batch' procedure. While this strategy facilitated the execution of this work, it ensured that no information describing the spatial elements nor any topological information describing the relationship between physical elements of a building design would be available to the

designer until the description input process was completed. This restriction is neither desirable nor theoretically inherent to the method. Methods for the interactive construction and editing of the network representation were examined and the development of an 'interactive' version of the 'batch' construction methods was found to be feasible. As detailed development of this approach was not considered to lie within the scope of this work, formal development of an 'interactive' method has been left to future study.

- 1.6) The extended PDV data structure provides a clear and seemingly efficient framework for storing and operating on the planar and dual graph representations employed to describe a building.
- 1.7) The algorithms used to construct the network representation described make use of moderately complex mathematical concepts drawn from the field of Graph Theory. Information retrieval from the building network representation data base requires the application of concepts drawn from the same field of mathematical theory and the development of data base interrogation methods which are similar in complexity to those used in the data base construction process.

2.0) The GRAPHIX System.

- 2.1) The GRAPHIX System provides a crude but operative mechanism for the communication of building information and the construction of the final dual graph network representation of a building design.
- 2.2) The Tektronix 4002A interactive computer graphics terminal used in the implementation of the GRAPHIX System is a storage tube display device. To delete imagery from the display, the display screen must first be erased and the edited image completely re-displayed. Although image erasure and re-display using the GRAPHIX System was normally executed in from two to five (2-5) seconds, it was found that even this short time interval could be extremely frustrating to the designer if multiple deletion operations were executed in quick succession.

The GRAPHIX System made too much use of screen erasure and re-display. If the System were to be further developed, the occurrence of this type of screen manipulation should be reduced significantly.

- 2.3) The screen of an interactive computer graphics terminal is an extremely valuable resource. This resource was not well utilized in the Plan Section Drawing Sub-system of the GRAPHIX System. Too much screen display area (approximately one-fifth (1/5) of the active display area) was dedicated to support menu list functions and, conversely, too little

was dedicated to support the primary plan section drawing 'drafting' function.

The screen display area might have been better allocated had the author chosen to invoke all the drawing commands via the graphics terminal keyboard rather than via the sonic pen and display menu list, and if he had chosen to display more abbreviated versions of the drawing symbols used in the 'drafting' process. These measures would have permitted the designer to communicate with both hands and provided more effective use of the graphics display screen.

- 2.4) The Tektronix 4002A used in the implementation of the GRAPHIX System was not equipped with a cross-hair screen co-ordinate input device. In order to circumvent this shortcoming, the author affixed the microphones of a sonic pen tablet input device about the screen of the Tektronix display screen and programmed an input interpolation routine that interrelated the location of a sonic pen strike on the display screens' outer surface with a screen address on the graphics display.

This data input arrangement provided an acceptable means for communicating display screen coordinate information to the GRAPHIX System, although some diminution in accuracy was apparent when pointing at locations at the screen's periphery.

- 2.5) The Plan Section Drawing Sub-system of the GRAPHIX System

did not display any information on the display identifying or describing the height of any secondary component, such as the wall element above the head of a door. The display processor of the GRAPHIX System should have compiled textual listings describing each such secondary component and output this information in conjunction with the appropriate primary building component symbol in the plan section drawing display.

- 2.6) The 'PLAN' drawing aid provided in the Plan Section Drawing Sub-system permitted the designer to display the plan section drawing lying immediately below the building plan section currently being 'drafted'. Such a facility is essential to interactive computer graphics applications, such as that considered within this work. Ideally, this drawing aid should have permitted the designer to superimpose any of the other plan section drawings used to depict the building design.

Conclusions

This thesis has involved architecture, mathematics and information processing, and dealt with one interactive computer graphics based method for inputting and representing geometric models of building designs in a digital computer. Working within the conceptual framework of an existing building description method, the thesis has explored how that concept might be

algorithmically realized, operationally implemented and practically employed in the building design process.

The following conclusions have been drawn as a result of this study;

- 1) The planar graph, planar graph overlay, and dual graph network methods described in this work provide a sound theoretic and operative basis to the development of useful interactive computer graphics based systems for the direct machine encoding of building descriptions.
- 2) The GRAPHIX System, as implemented, is useful only as a Computer Aided Building Design research and teaching tool. However, if further developed as observed above, the GRAPHIX System could provide a useable and practical graphical tool for directly inputting three-dimensional digital models of a building into the computer.
- 3) If the GRAPHIX System were to be further developed, every attempt should be made to acquire working versions of the special purpose GRAPPLE* and GLIDE(1) program language compilers.

The GRAPPLE language has been successfully employed by Public Works Canada to implement their comprehensive building drawing system, DRAW1(2), and is particularly suited to the representation of building floor plans such as those created in the GRAPHIX system.

*GRAPPLE was developed by Bell-Northern Research ltd., Ottawa.

The GLIDE language, developed by Eastman and Henrion, provides facilities to perform geometric and topological operations upon polyhedra.

- 4) A working knowledge of Planar Graph Theory is not only essential to the study of interactive computer based methods for the direct machine encoding of building descriptions but an increasing number of computer aided design and analysis methods employed in the architecture(3), interior design(4), landscape architecture(5,6), and city planning professions(7,8).
- 5) The development of useful interactive computer graphics systems for the direct machine encoding of building description is neither a trivial nor a simple task. Intensive reseach has been conducted in this area for at least 10 years and, while some very powerful systems are now operational, there still remain numerous problems without satisfactory solution. Until a recognized solution to this problem is found, the problem area will remain an active and viable field of endeavor for architectural and other Computer Aided Building Design researchers.

References: Chapter 4.

- 1 Mitchell, William John. Computer-aided architectural design. New York: Petrocelli/Charter, 1977. p. 200.
- 2 Computer-Aided Design Weds Scientific Progress to Designers' Art. Public Works Canada, Computer-Aided Design Centre, Ottawa, August, 1977.
- 3 Grason, John. Methods for the Computer-Implemented Solution of a Class of "Floor Plan" Design Problems. Phd. Thesis, Carnegie-Mellon University, 1970.
- 4 Pfefferkorn, Charles E.. "The Design Problem Solver: A System for Designing Equipment and Furniture Layouts" in Charles Eastmans ed., Spatial Synthesis in Computer-Aided Building Design. London: Applied Science Publishers Ltd., 1975. p.p. 230-286.
- 5 Canada Geographic Information System: Graphics Sub-system Commands. Canada Land Data Systems Division, Lands Directorate, Environment Canada, Ottawa, October, 1977.
- 6 Jackson, Harold E.. PLUS: an evaluation. Department of Landscape Architecture, University of Manitoba, 1973.
- 7 Symons, D. C.. A Parcel Geocoding System for Urban and Rural Information. National Capital Commission, Ottawa, September 23, 1970.
- 8 Harper, Robert H.. "An Administrative Organization for Data Management and Information System Development" in the Proceedings of the Twelfth Annual Conference of the Urban and Regional Information Systems Association. Montreal, Quebec, August,

October 1979

CONCLUSION

1974.

APPENDICES

APPENDIX A

APPENDIX A: AN INTRODUCTION TO GRAPH THEORYIntroduction

In order to familiarize the reader with the graph terminology and symbolism employed in this thesis, some general remarks and definitions pertaining to graph theory and graphs per se are presented in this section. As there currently is no vocabulary universally accepted in the field of graph theory, the reader is urged to study the conventions set out in this section before proceeding to examine the contents of Chapter 2.

What is a Graph

A graph can be visually depicted as a collection, or set, of very simple objects: a set of points, or vertices, joined by a prescribed set of line segments, or edges(1). These sets of edges and vertices are the fundamental objects of concern in the graph and possess no properties other than their visual objectivity. Neither the position of the vertices and edges, nor the curvature, nor the length of the edges bear any significance in graph theory. Rather, it is the study of the manner in which these edges and vertices can be inter-related which constitutes graph theory. For example, the graphs G1 and G2 of Figure A.1 have the same vertex and edge content and therefore are the same graphs(2).

Definition VERTEX: A vertex is the only significant joining of line

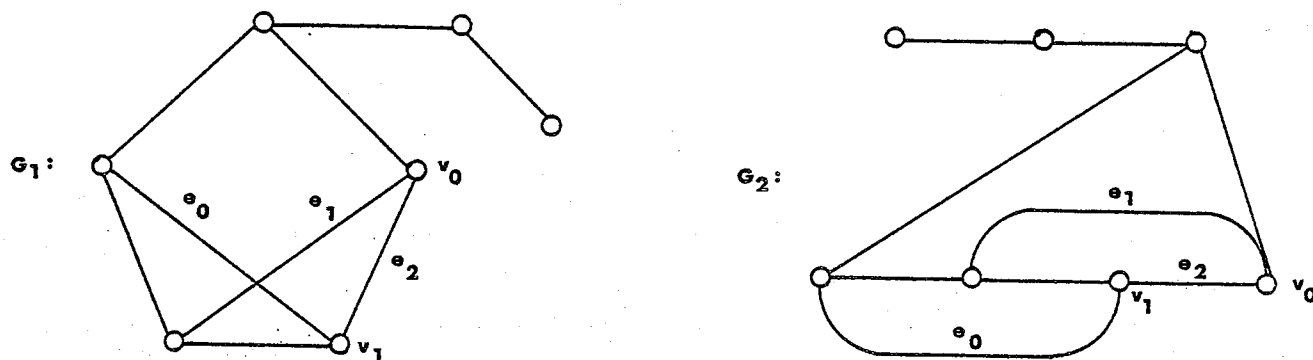


Figure A.1 DIFFERENT MAPPINGS OF A GRAPH.

segments(3).

In Figure A.1 the vertex v_0 of graph G_1 does join the edges labelled e_1 and e_2 , whereas there is no significant joining of the edges e_0 and e_1 in that graph.

Definition EDGE: An edge denotes the joining of two vertices as shown by e_1 and e_2 of graph $G_1(4)$.

Definition GRAPH: A graph G consists of a finite non empty set V of n vertices together with a prescribed set E of q unordered pairs of distinct vertices of V . Each pair $e=(v_1, v_2)$ of vertices in E is an edge of G , and e is said to join v_1 and $v_2(5)$. However, an edge may not join a vertex to itself (a loop). In a multigraph, a pair of vertices may be joined by more than one edge (multiple or parallel edges), but in a simple graph a pair of vertices may be joined by

at most one edge(6). In this thesis I shall use the term graph to refer only to finite simple graphs without loops.

Developing a vocabulary to describe and identify properties for distinct vertices and edges of a graph, the following definitions are presented with examples illustrated in Figure A.2.

Definition INCIDENCE: An edge is incident to a vertex, and a vertex is incident to an edge, if the vertex is a vertex of the edge(7).

The vertex v_0 is incident to edges e_0 , e_1 , and e_2 . The edge e_0 is incident to vertices v_0 and v_5 .

Definition ADJACENT EDGES: Two edges are adjacent if the edges are incident to the same vertex(8).

The edges e_0 and e_5 are adjacent, being incident to the common vertex v_5 .

Definition ADJACENT VERTICES: Two vertices are adjacent if the vertices are incident to the same edge(9).

The vertices v_0 and v_4 are adjacent, being incident to the common edge e_2 .

Definition DEGREE: The degree of a vertex is the number of edges incident to the vertex(10).

The vertices v_6 , v_7 , and v_8 are of degree one, the vertex v_5 of degree two, and the remaining vertices of the graph, namely v_0 , v_1 , v_2 , v_3 , and v_4 , of degree three.

Definition END EDGE: An end edge is an edge incident to at least

one vertex of degree one(11).

The edges e_3 and e_9 are end edges.

Definition END VERTEX: An end vertex is a vertex of degree one(12).

The vertices v_6 , v_7 , and v_8 are end vertices.

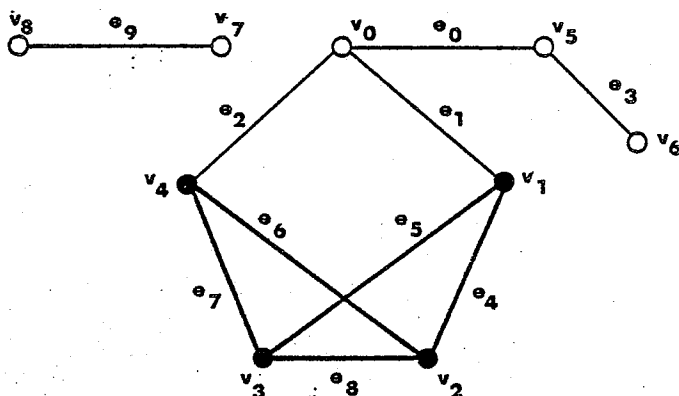


Figure A.2 A SUBGRAPH.

Definition SUBGRAPH: A subgraph $G(s)$ of a graph is a graph having all of its vertices and edges in $G(13)$. The subgraph $G(s)$ is a subset of the elements of G .

Because a subgraph is a graph and a graph is a subgraph of itself, it may seem that the two words 'graph' and 'subgraph' are synonymous. Although this is true in some respects, the word 'graph' is used to denote the complete set of elements upon which attention is focused. The term 'subgraph' is used to focus attention on a particular subset of the complete set(14).

To assist in the designation and discussion of subgraphs of a graph, the elements of the subgraph are visually enhanced in the diagrams, illustrating subgraph edges with heavy lines, and subgraph vertices with solid dots. The subgraphs $G(s)$ of graph G in Figure A.2 contains the edge subset $E(s)=(e_4, e_5, e_6, e_7, e_8)$ and the vertex subset $V(s)=(v_1, v_2, v_3, v_4)$.

Three topics, namely CONNECTEDNESS, WALKS, and PLANARITY, are now introduced which describe specific graph concepts pertinent to this thesis.

Connectedness

A graph may be said to be either disconnected or connected. A graph G , or subgraph $G(s)$, is disconnected if it is possible to divide its set of vertices V , or $V(s)$, into two distinct subsets, V_1 and V_2 , such that there are no edges joining any vertex in V_1 to any vertex in V_2 ; otherwise, the graph is said to be connected(15).

The following algorithm may be used to determine a graphs' or subgraphs' state of connectedness.

Select any vertex v_0 of a non-null graph G (or non-null subgraph $G(s)$). Label all vertices adjacent to v_0 as v_1 . Except for those vertices previously labelled, label all vertices adjacent to vertices labelled v_1 as v_2 . Continue this vertex labelling process until it terminates, as it will for a finite graph. If, when the labelling process terminates, all the vertices have not been labelled, $G(G(s))$ is disconnected(16).

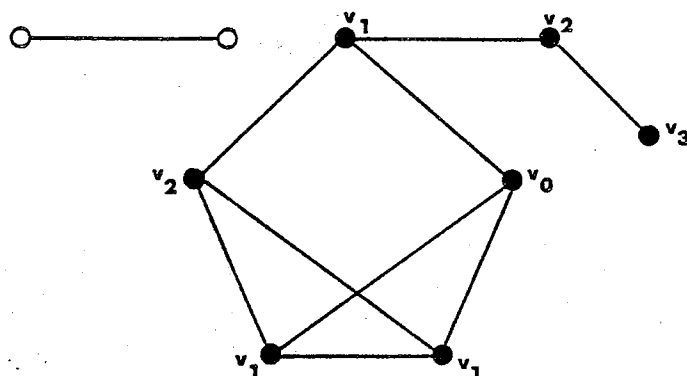


Figure A.3 A DISCONNECTED GRAPH.

Because all the vertices of graph G in Figure A.3 are not labelled at the conclusion of the labelling process, the Graph G is disconnected.

Very closely related yet significantly different from the concept of connectedness is the concept of a PART. The word 'Part' is used to describe a 'whole piece' of a graph or of a subgraph. For example, the above graph G has two Parts. A Part is a particular kind of subgraph of a graph (or of a subgraph) and can be algorithmically defined as;

Definition PART: Select any vertex v_0 of a non-null graph G (or a non-null subgraph $G(s)$). Apply the same vertex labelling process to the graph as is described above. The set of all edges of G ($G(s)$) incident to all vertices which have been labelled constitutes a

subgraph of G ($G(s)$) called a Part P of G ($G(s)$)(17).

As the term Part can be used to imply properties possessed by either a subgraph or graph, its use is always relative to the universe of objects under consideration. When a reference is made to a particular Part P , the graph or subgraph of which P is a Part must then always be made explicit, i.e. P of G or P of $G(s)$. However, in order to reduce the constant definition of Part in this thesis, the Part P should always be interpreted to refer to the graph G under consideration at the time unless stipulated to the contrary.

Definition CUT-VERTEX: A vertex belonging to a connected graph G the removal of which, together with the edges incident to it, produces a disconnected graph is referred to as a cut-vertex.

Definition BRIDGE: An edge belonging to a connected graph G , the removal of which produces a disconnected graph, is referred to as a bridge.

In Figure A.4, vertex v_3 of the graph G_1 is a cut-vertex, and the edge e_4 of the graph G_2 is a bridge. Thus, if v_3 is a cut-vertex of the connected graph G_1 , then $G_1 - v_3$ is a disconnected graph. Likewise, if e_4 is a bridge of the connected graph G_2 , then $G_2 - e_4$ is a disconnected graph. It should be noted that the removal of either v_3 from G_1 or e_4 from G_2 will increase the number of Parts in each graph.

Walks

Definition WALK: A walk of a graph G is an alternating sequence of

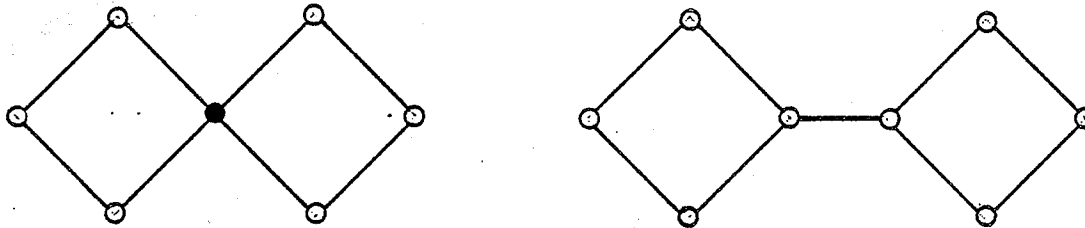


Figure A.4 CUT-VERTICES AND BRIDGES.

vertices and edges $v_0, e_0, v_1, e_1, v_2, \dots, v_{(n-1)}, e_{(n)}, v_{(n)}$, beginning and ending at vertices immediately preceding and following it. A walk joining v_0 and $v_{(n)}$ may be denoted by a sequence of vertex labels, $v_0, v_1, v_2, \dots, v_{(n)}$, where any vertex in the series infers vertex adjacency in the graph. A walk is considered to be closed if $v_0 = v_{(n)}$ and is open otherwise(18).

Placing restrictions on the sequence of vertices and edges contained in any walk sequence, the following terms are used to define specific walk subsets:

Definition TRAIL: A trail is a walk where all the edges are distinct.

Definition PATH: A path is a walk where all the vertices (and thus necessarily all the edges) are distinct.

Definition CYCLE: A cycle is a closed walk in which its n vertices

are distinct and n is greater than 3.

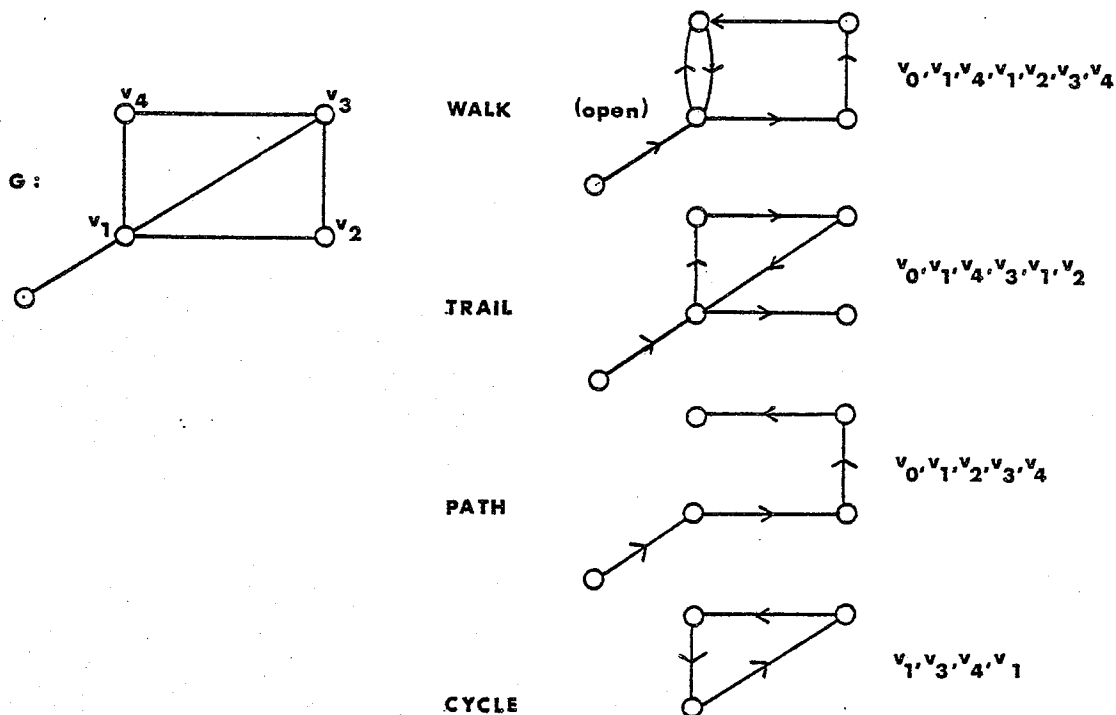


Figure A.5 WALKS, TRAILS, PATHS AND CYCLES.

Planarity

The discussion so far has been entirely in terms of the abstract graph where its geometric diagram has served only for illustrative purpose. At this juncture the planar graph is introduced and unlike the abstract graph, is defined in terms of its geometric diagram.

Definition PLANAR GRAPH: Note that a planar graph is a non-null graph which can be drawn on a plane so that it has no edges intersecting or crossing (ie. no two edges share a common point in the plane which is not a vertex).

In topology, the plane and the sphere are classified as surfaces

of genus zero. The genus of a surface is the largest number of closed polygons that can be drawn on the surface without separating it. A closed polygon drawn on either a plane or a sphere separates the surface into distinct areas(19).

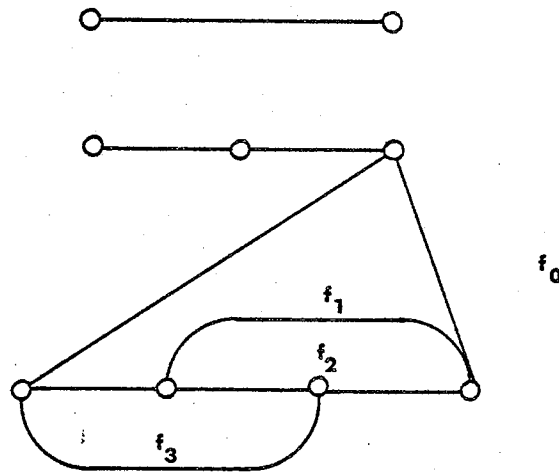


Figure A.6 A PLANAR GRAPH.

Definition REGION: The separate areas of a plane identified by a planar graph are referred to as its regions. The unbounded region f_0 is called the exterior region, and the bounded areas f_1 , f_2 , and f_3 will be referred to as the interior regions of the graph.

References: Appendix A

- 1 Maxwell, L.M., and Reed, M.B., Applied Graph Theory. New York:
American Elsevier Publishing Company, 1971. p. 1.
- 2 ibid. p. 2.
- 3 ibid. p. 13.
- 4 ibid. p. 14.
- 5 Harary, F., Graph Theory. Don Mills, Ontario: Addison-Wesley
Publishing Company, 1969. p. 9.
- 6 Read, Ronald C., Graph Theory and Computing. New York: Academic
Press, 1972. p. 49.
- 7 op. cit. 1. p. 14.
- 8 ibid. p. 14.
- 9 ibid. p. 14.
- 10 ibid. p. 14.
- 11 ibid. p. 14.
- 12 ibid. p. 14.
- 13 ibid. p. 14.
- 14 op. cit. 5. p. 11.
- 15 op. cit. 1. p. 16.
- 16 op. cit. 6. p. 49.
- 17 op. cit. 1. p. 25.
- 18 ibid. p. 26.
- 19 op. cit. 5. p. 25.
- 20 op. cit. 1. p.p. 34-35.

APPENDIX B

APPENDIX B: THE RESULTS OF A BUILDING THERMAL ANALYSIS

This appendix documents the output produced by Mr. A. Dakins' thermal heat loss/gain building analysis computer program. The output describes some of the calculations that were performed by this program for a single storey house design which was digitally described using the GRAPHIX interactive system. These results were used to generate the data presented in Figure 3.12.

October 1979

APPENDIX B

HAROLD JACKSON
 LATITUDE= 49.44333 DEGREES
 LONGITUDE= 100.00000 DEGREES

SURFACE ORIENTATION	SURFACE SLOPE	SURFACE AREA	U VALUE (WALL)	SURFACE ABSORB.	GLASS AREA	U VALUE GLASS	CRACK LENGTH	SURFACE CONDUCT.
180.00	90.00	144.00	.25	.50	80.00	.49	0.00	6.00
270.00	90.00	192.00	.25	.50	0.00	.49	0.00	6.00
360.00	90.00	144.00	.25	.50	80.00	.49	0.00	6.00
90.00	90.00	149.28	.25	.50	42.72	.49	0.00	6.00
180.00	0.00	672.00	.15	.50	0.00	.49	0.00	4.00
0.00	0.00	672.00	.25	0.00	0.00	0.00	0.00	.01

VALUES FOR JAN. 21 ST.

POLAR DECLINATION = -20.0000 DEGREES

TIME (HRS)	ALTITUDE (DEGREES)	AZIMUTH (DEGREES)	RON (BTU/HR)	DIFFUSE (BTU/HR)
1	-57.823	27.175	0.000	0.000
2	-51.806	49.453	0.000	0.000
3	-43.606	66.584	0.000	0.000
4	-34.354	80.312	0.000	0.000
5	-24.725	92.170	0.000	0.000
6	-15.162	103.199	0.000	0.000
7	-6.018	114.118	0.000	0.000
8	2.361	125.463	12.412	.360
9	9.590	137.633	166.294	4.823
10	15.237	150.859	227.202	6.589
11	18.863	165.107	251.371	7.290
12	20.117	180.000	258.082	7.484
13	18.863	194.893	251.371	7.290
14	15.237	209.141	227.202	6.589
15	9.590	222.367	166.293	4.823
16	2.361	234.537	12.412	.360
17	-6.018	245.802	0.000	0.000
18	-15.162	256.801	0.000	0.000
19	-24.725	267.830	0.000	0.000
20	-34.354	279.688	0.000	0.000
21	-43.606	293.416	0.000	0.000
22	-51.806	310.547	0.000	0.000
23	-57.823	332.825	0.000	0.000
24	-60.117	360.000	0.000	0.000

October 1979

APPENDIX B

WALL SURFACE ONF ~ 0.00 DEGREES

	1	2	3	4	5	6	7	8	9	10	11	12
SURFACE 1												
WHL	3110.	3110.	3110.	3110.	3110.	3110.	3110.	3108.	3013.	2911.	2845.	2822.
GHL	3387.	3387.	3387.	3387.	3387.	3387.	3387.	3364.	3086.	2986.	2941.	2941.
THL	6497.	6497.	6497.	6497.	6497.	6497.	6497.	6471.	6099.	5897.	5795.	5762.
SURFACE 2												
WHL	4147.	4147.	4147.	4147.	4147.	4147.	4147.	4105.	3686.	3694.	3873.	3762.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	4147.	4147.	4147.	4147.	4147.	4147.	4147.	4105.	3686.	3694.	3873.	3762.
SURFACE 3												
WHL	3110.	3110.	3110.	3110.	3110.	3110.	3110.	3088.	2732.	2516.	2309.	2361.
GHL	3387.	3387.	3387.	3387.	3387.	3387.	3387.	2880.	-5331.	-10546.	-13411.	-14321.
THL	6497.	6497.	6497.	6497.	6497.	6497.	6497.	5968.	-2599.	-8030.	-11013.	-11960.
SURFACE 4												
WHL	3224.	3224.	3224.	3224.	3224.	3224.	3224.	3201.	2833.	2608.	2487.	2448.
GHL	1809.	1809.	1809.	1809.	1809.	1809.	1809.	1798.	1612.	1627.	1612.	1608.
THL	5033.	5033.	5033.	5033.	5033.	5033.	5033.	4999.	4504.	4235.	4099.	4056.
SURFACE 5												
WHL	8709.	8709.	8709.	8709.	8709.	8709.	8709.	8694.	8239.	7791.	7501.	7402.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	8709.	8709.	8709.	8709.	8709.	8709.	8709.	8694.	8239.	7791.	7501.	7402.
SURFACE 6												
WHL	14515.	14515.	14515.	14515.	14515.	14515.	14515.	14515.	14515.	14515.	14515.	14515.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	14515.	14515.	14515.	14515.	14515.	14515.	14515.	14515.	14515.	14515.	14515.	14515.
GRAND TOT	45399.	45399.	45399.	45399.	45399.	45399.	45399.	44753.	34443.	28103.	24771.	23537.

	13	14	15	16	17	18	19	20	21	22	23	24
SURFACE 1												
WHL	2845.	2911.	3013.	3108.	3110.	3110.	3110.	3110.	3110.	3110.	3110.	3110.
GHL	2950.	2986.	3086.	3364.	3387.	3387.	3387.	3387.	3387.	3387.	3387.	3387.
THL	5795.	5897.	6099.	6471.	6497.	6497.	6497.	6497.	6497.	6497.	6497.	6497.
SURFACE 2												
WHL	3793.	3882.	4017.	4144.	4147.	4147.	4147.	4147.	4147.	4147.	4147.	4147.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	3793.	3882.	4017.	4144.	4147.	4147.	4147.	4147.	4147.	4147.	4147.	4147.
SURFACE 3												
WHL	2399.	2516.	2732.	3088.	3110.	3110.	3110.	3110.	3110.	3110.	3110.	3110.
GHL	-13411.	-10546.	-5331.	2880.	3387.	3387.	3387.	3387.	3387.	3387.	3387.	3387.
THL	-11013.	-8030.	-2599.	5968.	6497.	6497.	6497.	6497.	6497.	6497.	6497.	6497.
SURFACE 4												
WHL	3012.	2872.	2866.	3192.	3224.	3224.	3224.	3224.	3224.	3224.	3224.	3224.
GHL	92.	-2051.	-2406.	1417.	1809.	1809.	1809.	1809.	1809.	1809.	1809.	1809.
THL	3104.	821.	460.	4609.	5033.	5033.	5033.	5033.	5033.	5033.	5033.	5033.
SURFACE 5												
WHL	7501.	7791.	8239.	8694.	8709.	8709.	8709.	8709.	8709.	8709.	8709.	8709.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	7501.	7791.	8239.	8694.	8709.	8709.	8709.	8709.	8709.	8709.	8709.	8709.
SURFACE 6												
WHL	14515.	14515.	14515.	14515.	14515.	14515.	14515.	14515.	14515.	14515.	14515.	14515.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	14515.	14515.	14515.	14515.	14515.	14515.	14515.	14515.	14515.	14515.	14515.	14515.
GRAND TOT	23696.	24877.	30731.	44401.	45399.	45399.	45399.	45399.	45399.	45399.	45399.	45399.

VALUES FOR FEB. 21 ST.

POLAR DECLINATION = -10.0000 DEGREES

TIME (HRS)	ALTITUDE (DEGREES)	AZIMUTH (DEGREES)	RDN (BTU/HR)	DIFFUSE (BTU/HR)
1	-48.996	22.798	0.000	0.000
2	-43.744	42.831	0.000	0.000
3	-36.217	59.421	0.000	0.000
4	-27.372	73.322	0.000	0.000
5	-17.885	85.559	0.000	0.000
6	-8.239	97.007	0.000	0.000
7	1.176	108.375	.345	.010
8	9.972	120.261	167.618	5.029
9	17.713	133.185	239.836	7.195
10	23.881	147.512	269.764	8.093
11	27.909	163.280	283.042	8.491
12	29.317	180.000	286.904	8.607
13	27.909	196.720	283.042	8.491
14	23.881	212.488	269.764	8.093
15	17.713	226.815	239.836	7.195
16	9.972	239.739	167.618	5.029
17	1.176	251.626	.345	.010
18	-8.239	262.993	0.000	0.000
19	-17.885	274.441	0.000	0.000
20	-27.372	286.678	0.000	0.000
21	-36.217	300.579	0.000	0.000
22	-43.744	317.169	0.000	0.000
23	-48.996	337.202	0.000	0.000
24	-50.917	360.000	0.000	0.000

October 1979

APPENDIX B

WALL SURFACE ONE ~ 0.00 DEGREES

	1	2	3	4	5	6	7	8	9	10	11	12
SURFACE 1												
WHL	2747.	2747.	2747.	2747.	2747.	2747.	2747.	2747.	2506.	2395.	2324.	2300.
GHL	2991.	2991.	2991.	2991.	2991.	2991.	2990.	2677.	2506.	2395.	2324.	2300.
THL	5738.	5738.	5738.	5738.	5738.	5738.	5737.	5322.	5063.	4910.	4823.	4795.
SURFACE 2												
WHL	3662.	3662.	3662.	3662.	3662.	3662.	3661.	3072.	2967.	3100.	3341.	3066.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	3662.	3662.	3662.	3662.	3662.	3662.	3661.	3072.	2967.	3100.	3341.	3066.
SURFACE 3												
WHL	2747.	2747.	2747.	2747.	2747.	2747.	2746.	2482.	2256.	2098.	2003.	1971.
GHL	2991.	2991.	2991.	2991.	2991.	2991.	2985.	-2723.	-8174.	-12065.	-14464.	-15272.
THL	5738.	5738.	5738.	5738.	5738.	5738.	5731.	-241.	-5918.	-9967.	-12461.	-13302.
SURFACE 4												
WHL	2848.	2848.	2848.	2848.	2848.	2848.	2847.	2573.	2339.	2175.	2076.	2043.
GHL	1597.	1597.	1597.	1597.	1597.	1597.	1597.	1443.	1386.	1367.	1361.	1360.
THL	4445.	4445.	4445.	4445.	4445.	4445.	4444.	4016.	3725.	3542.	3437.	3402.
SURFACE 5												
WHL	7691.	7691.	7691.	7691.	7691.	7691.	7691.	7199.	6590.	6111.	5808.	5704.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	7691.	7691.	7691.	7691.	7691.	7691.	7691.	7199.	6590.	6111.	5808.	5704.
SURFACE 6												
WHL	12818.	12818.	12818.	12818.	12818.	12818.	12818.	12818.	12818.	12818.	12818.	12818.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	12818.	12818.	12818.	12818.	12818.	12818.	12818.	12818.	12818.	12818.	12818.	12818.
GRAND TOT	40092.	40092.	40092.	40092.	40092.	40092.	40082.	32185.	25246.	20515.	17766.	16484.

	13	14	15	16	17	18	19	20	21	22	23	24
SURFACE 1												
WHL	2324.	2395.	2506.	2645.	2747.	2747.	2747.	2747.	2747.	2747.	2747.	2747.
GHL	2499.	2515.	2556.	2677.	2990.	2991.	2991.	2991.	2991.	2991.	2991.	2991.
THL	4823.	4910.	5063.	5322.	5737.	5738.	5738.	5738.	5738.	5738.	5738.	5738.
SURFACE 2												
WHL	3099.	3193.	3342.	3526.	3662.	3662.	3662.	3662.	3662.	3662.	3662.	3662.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	3099.	3193.	3342.	3526.	3662.	3662.	3662.	3662.	3662.	3662.	3662.	3662.
SURFACE 3												
WHL	2003.	2098.	2256.	2482.	2746.	2747.	2747.	2747.	2747.	2747.	2747.	2747.
GHL	-14464.	-12065.	-8174.	-2723.	2985.	2991.	2991.	2991.	2991.	2991.	2991.	2991.
THL	-12461.	-9967.	-5918.	-241.	5731.	5738.	5738.	5738.	5738.	5738.	5738.	5738.
SURFACE 4												
WHL	2597.	2410.	2307.	2388.	2846.	2848.	2848.	2848.	2848.	2848.	2848.	2848.
GHL	-474.	-3252.	-4789.	-3953.	1584.	1597.	1597.	1597.	1597.	1597.	1597.	1597.
THL	2123.	-841.	-2482.	-1565.	4431.	4445.	4445.	4445.	4445.	4445.	4445.	4445.
SURFACE 5												
WHL	5808.	6111.	6590.	7199.	7691.	7691.	7691.	7691.	7691.	7691.	7691.	7691.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	5808.	6111.	6590.	7199.	7691.	7691.	7691.	7691.	7691.	7691.	7691.	7691.
SURFACE 6												
WHL	12818.	12818.	12818.	12818.	12818.	12818.	12818.	12818.	12818.	12818.	12818.	12818.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	12818.	12818.	12818.	12818.	12818.	12818.	12818.	12818.	12818.	12818.	12818.	12818.
GRAND TOT	16210.	16224.	19414.	27059.	40070.	40092.	40092.	40092.	40092.	40092.	40092.	40092.

VALUES FOR MAR. 21 ST.				
POLAR DECLINATION =		0.0000	DEGREES	
TIME (HRS)	ALTITUDE (DEGREES)	AZIMUTH (DEGREES)	PON (BTU/HR)	DIFUSE (BTU/HR)
1	-38.491	19.310	0.000	0.000
2	-33.919	37.052	0.000	0.000
3	-27.105	52.594	0.000	0.000
4	-18.794	66.178	0.000	0.000
5	-9.600	78.420	0.000	0.000
6	.000	90.000	0.000	0.000
7	9.600	101.580	147.550	5.238
8	18.794	113.822	231.685	8.225
9	27.105	127.406	266.987	9.478
10	33.919	142.948	284.300	10.093
11	38.491	160.690	292.640	10.389
12	40.117	180.000	295.150	10.478
13	38.491	199.310	292.640	10.389
14	33.919	217.052	284.300	10.093
15	27.105	232.594	266.987	9.478
16	18.794	246.178	231.685	8.225
17	9.600	258.420	147.549	5.238
18	-.000	270.000	0.000	0.000
19	-9.600	281.580	0.000	0.000
20	-18.794	293.822	0.000	0.000
21	-27.105	307.406	0.000	0.000
22	-33.919	322.948	0.000	0.000
23	-38.491	340.690	0.000	0.000
24	-40.117	360.000	0.000	0.000

October 1979

APPENDIX B

WALL SURFACE ONT ~ 0.00 DEGREES

	1	2	3	4	5	6	7	8	9	10	11	12
SURFACE 1												
WHL	2372.	2372.	2372.	2372.	2372.	2372.	2283.	2124.	1979.	1866.	1795.	1770.
GHL	2583.	2583.	2583.	2583.	2583.	2583.	2254.	2082.	2023.	2000.	1991.	1989.
THL	4956.	4956.	4956.	4956.	4956.	4956.	4537.	4206.	4002.	3866.	3786.	3760.
SURFACE 2												
WHL	3163.	3163.	3163.	3163.	3163.	3163.	2572.	2328.	2370.	2554.	2819.	2361.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	3163.	3163.	3163.	3163.	3163.	3163.	2572.	2328.	2370.	2554.	2819.	2361.
SURFACE 3												
WHL	2372.	2372.	2372.	2372.	2372.	2372.	2269.	2082.	1911.	1777.	1693.	1664.
GHL	2583.	2583.	2583.	2583.	2583.	2583.	1121.	-3154.	-7589.	-10982.	-13113.	-13845.
THL	4956.	4956.	4956.	4956.	4956.	4956.	3390.	-1072.	-5678.	-9205.	-11420.	-12181.
SURFACE 4												
WHL	2459.	2459.	2459.	2459.	2459.	2459.	2352.	2158.	1981.	1842.	1755.	1725.
GHL	1379.	1379.	1379.	1379.	1379.	1379.	1205.	1115.	1086.	1075.	1072.	1071.
THL	3839.	3839.	3839.	3839.	3839.	3839.	3557.	3274.	3067.	2918.	2826.	2796.
SURFACE 5												
WHL	6643.	6643.	6643.	6643.	6643.	6643.	6201.	5495.	4871.	4389.	4086.	3982.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	6643.	6643.	6643.	6643.	6643.	6643.	6201.	5495.	4871.	4389.	4086.	3982.
SURFACE 6												
WHL	11071.	11071.	11071.	11071.	11071.	11071.	11071.	11071.	11071.	11071.	11071.	11071.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	11071.	11071.	11071.	11071.	11071.	11071.	11071.	11071.	11071.	11071.	11071.	11071.
GRAND TOT	34627.	34627.	34627.	34627.	34627.	34627.	31327.	29302.	19703.	15593.	13169.	11789.

	13	14	15	16	17	18	19	20	21	22	23	24
SURFACE 1												
WHL	1795.	1866.	1979.	2124.	2283.	2372.	2372.	2372.	2372.	2372.	2372.	2372.
GHL	1991.	2000.	2023.	2082.	2254.	2583.	2583.	2583.	2583.	2583.	2583.	2583.
THL	3786.	3866.	4002.	4206.	4537.	4956.	4956.	4956.	4956.	4956.	4956.	4956.
SURFACE 2												
WHL	2393.	2488.	2639.	2832.	3044.	3163.	3163.	3163.	3163.	3163.	3163.	3163.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	2393.	2488.	2639.	2832.	3044.	3163.	3163.	3163.	3163.	3163.	3163.	3163.
SURFACE 3												
WHL	1693.	1777.	1911.	2082.	2269.	2372.	2372.	2372.	2372.	2372.	2372.	2372.
GHL	-13113.	-10982.	-7589.	-3154.	1121.	2583.	2583.	2583.	2583.	2583.	2583.	2583.
THL	-11420.	-9205.	-5678.	-1072.	3390.	4956.	4956.	4956.	4956.	4956.	4956.	4956.
SURFACE 4												
WHL	2192.	1986.	1843.	1810.	2000.	2459.	2459.	2459.	2459.	2459.	2459.	2459.
GHL	-874.	-3897.	-5919.	-6484.	-4207.	1379.	1379.	1379.	1379.	1379.	1379.	1379.
THL	1317.	-1911.	-4076.	-4674.	-2207.	3839.	3839.	3839.	3839.	3839.	3839.	3839.
SURFACE 5												
WHL	4086.	4389.	4871.	5495.	6201.	6643.	6643.	6643.	6643.	6643.	6643.	6643.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	4086.	4389.	4871.	5495.	6201.	6643.	6643.	6643.	6643.	6643.	6643.	6643.
SURFACE 6												
WHL	11071.	11071.	11071.	11071.	11071.	11071.	11071.	11071.	11071.	11071.	11071.	11071.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	11071.	11071.	11071.	11071.	11071.	11071.	11071.	11071.	11071.	11071.	11071.	11071.
GRAND TOT	11734.	10699.	12829.	17858.	26035.	34627.	34627.	34627.	34627.	34627.	34627.	34627.

October 1979

APPENDIX B

VALUES FOR APR. 21 ST.

POLAR DECLINATION = 11.6000 DEGREES

TIME (HRS)	ALTITUDE (DEGREES)	AZIMUTH (DEGREES)	RDN (BTU/HR)	DIFFUSE (BTU/HR)
1	-27.123	16.550	0.000	0.000
2	-23.132	32.182	0.000	0.000
3	-17.010	46.415	0.000	0.000
4	-9.313	59.279	0.000	0.000
5	-.550	71.127	0.000	0.000
6	8.846	82.466	111.668	5.416
7	18.490	93.889	204.082	9.898
8	27.593	106.108	245.331	11.899
9	36.876	120.014	266.706	12.935
10	44.459	136.669	278.413	13.503
11	49.769	156.887	284.386	13.793
12	51.717	180.000	286.229	13.882
13	49.769	203.113	284.386	13.793
14	44.459	223.331	278.413	13.503
15	36.876	239.986	266.706	12.935
16	27.593	253.892	245.330	11.899
17	18.490	266.111	204.082	9.898
18	8.846	277.535	111.668	5.416
19	-.550	288.873	0.000	0.000
20	-9.313	300.721	0.000	0.000
21	-17.010	313.585	0.000	0.000
22	-23.132	327.818	0.000	0.000
23	-27.123	343.450	0.000	0.000
24	-28.517	360.000	0.000	0.000

October 1979

APPENDIX B

WALL SURFACE ONF ^ 0.00 DEGREES

	1	2	3	4	5	6	7	8	9	10	11	12
SURFACE 1												
WHL	1631.	1631.	1631.	1631.	1641.	1571.	1407.	1250.	1112.	1005.	938.	915.
GHL	1776.	1776.	1776.	1776.	1776.	1034.	1161.	1056.	1008.	987.	978.	976.
THL	3407.	3407.	3407.	3407.	3407.	2605.	2568.	2304.	2120.	1992.	1916.	1891.
SURFACE 2												
WHL	2174.	2174.	2174.	2174.	2174.	1715.	1362.	1294.	1384.	1575.	1831.	1220.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	2174.	2174.	2174.	2174.	2174.	1715.	1362.	1294.	1384.	1575.	1831.	1220.
SURFACE 3												
WHL	1631.	1631.	1631.	1631.	1631.	1285.	1562.	1415.	1272.	1157.	1083.	1057.
GHL	1776.	1776.	1776.	1776.	1776.	1478.	949.	-1706.	-5462.	-8577.	-10514.	-11170.
THL	3407.	3407.	3407.	3407.	3407.	2760.	2511.	-291.	-4190.	-7420.	-9432.	-10113.
SURFACE 4												
WHL	1691.	1691.	1691.	1691.	1691.	1334.	1619.	1467.	1318.	1199.	1122.	1096.
GHL	948.	948.	948.	948.	948.	787.	607.	550.	525.	515.	511.	509.
THL	2639.	2639.	2639.	2639.	2619.	2121.	2226.	2016.	1843.	1714.	1633.	1605.
SURFACE 5												
WHL	4566.	4566.	4566.	4566.	4566.	4213.	3501.	2816.	2224.	1769.	1483.	1385.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	4566.	4566.	4566.	4566.	4566.	4213.	3501.	2816.	2224.	1769.	1483.	1385.
SURFACE 6												
WHL	7610.	7610.	7610.	7610.	7610.	7610.	7610.	7610.	7610.	7610.	7610.	7610.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	7610.	7610.	7610.	7610.	7610.	7610.	7610.	7610.	7610.	7610.	7610.	7610.
GRAND TOT	23803.	23803.	23803.	23803.	23803.	21025.	19780.	15749.	10991.	7240.	5042.	3600.

	13	14	15	16	17	18	19	20	21	22	23	24
SURFACE 1												
WHL	938.	1005.	1112.	1250.	1407.	1571.	1631.	1631.	1631.	1631.	1631.	1631.
GHL	978.	987.	1008.	1055.	1161.	1034.	1776.	1776.	1776.	1776.	1776.	1776.
THL	1916.	1992.	2120.	2304.	2568.	2605.	3407.	3407.	3407.	3407.	3407.	3407.
SURFACE 2												
WHL	1251.	1340.	1482.	1666.	1876.	2095.	2174.	2174.	2174.	2174.	2174.	2174.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	1251.	1340.	1482.	1666.	1876.	2095.	2174.	2174.	2174.	2174.	2174.	2174.
SURFACE 3												
WHL	1083.	1157.	1272.	1415.	1562.	1571.	1631.	1631.	1631.	1631.	1631.	1631.
GHL	-10514.	-8577.	-5462.	-1706.	949.	1430.	1776.	1776.	1776.	1776.	1776.	1776.
THL	-9432.	-7420.	-4190.	-291.	2511.	3001.	3407.	3407.	3407.	3407.	3407.	3407.
SURFACE 4												
WHL	1423.	1225.	1076.	1006.	1059.	1334.	1691.	1691.	1691.	1691.	1691.	1691.
GHL	-1308.	-4218.	-6303.	-7309.	-6723.	-3383.	948.	948.	948.	948.	948.	948.
THL	116.	-2994.	-5228.	-6383.	-5664.	-2049.	2639.	2639.	2639.	2639.	2639.	2639.
SURFACE 5												
WHL	1483.	1769.	2224.	2816.	3501.	4213.	4566.	4566.	4566.	4566.	4566.	4566.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	1483.	1769.	2224.	2816.	3501.	4213.	4566.	4566.	4566.	4566.	4566.	4566.
SURFACE 6												
WHL	7610.	7610.	7610.	7610.	7610.	7610.	7610.	7610.	7610.	7610.	7610.	7610.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	7610.	7610.	7610.	7610.	7610.	7610.	7610.	7610.	7610.	7610.	7610.	7610.
GRAND TOT	2945.	2298.	4019.	7802.	12403.	17476.	23803.	23803.	23803.	23803.	23803.	23803.

October 1979

APPENDIX B

VALUES FOR MAY 21 ST.				
POLAR DECLINATION = 20.0000 DEGREES				
TIME (HRS)	ALTITUDE (DEGREES)	AZIMUTH (DEGREES)	RDN (BTU/HR)	DIFFUSE (BTU/HR)
1	-18.863	14.893	0.000	0.000
2	-15.237	29.141	0.000	0.000
3	-9.590	42.367	0.000	0.000
4	-2.361	54.537	0.000	0.000
5	6.018	65.882	53.972	3.265
6	15.162	76.801	165.433	10.009
7	24.725	87.830	219.057	13.253
8	34.354	99.688	247.299	14.962
9	43.606	113.416	263.420	15.937
10	51.806	130.547	272.747	16.501
11	57.823	152.825	277.652	16.798
12	60.117	180.000	279.186	16.891
13	57.823	207.175	277.652	16.798
14	51.806	229.453	272.747	16.501
15	43.606	246.584	263.420	15.937
16	34.354	260.312	247.299	14.962
17	24.725	272.170	219.057	13.253
18	15.162	283.199	165.433	10.009
19	6.018	294.118	53.972	3.265
20	-2.361	305.463	0.000	0.000
21	-9.590	317.633	0.000	0.000
22	-15.237	330.859	0.000	0.000
23	-18.863	345.107	0.000	0.000
24	-20.117	360.000	0.000	0.000

October 1979

APPENDIX B

WALL SURFACE ONT ^ 0.00 DEGREES

	1	2	3	4	5	6	7	8	9	10	11	12
SURFACE 1												
WHL	1008.	1009.	1008.	1008.	932.	869.	946.	544.	415.	315.	253.	231.
GHL	1098.	1098.	1098.	1098.	-438.	-1074.	179.	188.	144.	435.	113.	110.
THL	2106.	2106.	2106.	2106.	494.	-206.	1125.	733.	559.	438.	365.	341.
SURFACE 2												
WHL	1344.	1344.	1344.	1344.	1135.	682.	496.	479.	580.	765.	1007.	308.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	1344.	1344.	1344.	1344.	1135.	682.	496.	479.	580.	765.	1007.	308.
SURFACE 3												
WHL	1008.	1008.	1008.	1008.	851.	512.	372.	860.	733.	630.	563.	540.
GHL	1098.	1098.	1098.	1098.	908.	518.	327.	-858.	-3743.	-6560.	-8395.	-9018.
THL	2106.	2106.	2106.	2106.	1759.	1030.	699.	2.	-3011.	-5930.	-7832.	-8478.
SURFACE 4												
WHL	1045.	1045.	1045.	1045.	882.	530.	385.	892.	760.	653.	584.	560.
GHL	586.	586.	586.	586.	485.	277.	385.	75.	51.	40.	35.	33.
THL	1631.	1631.	1631.	1631.	1367.	807.	568.	966.	811.	692.	618.	593.
SURFACE 5												
WHL	2822.	2822.	2822.	2822.	2669.	2025.	1334.	687.	132.	-294.	-562.	-653.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	2822.	2822.	2822.	2822.	2669.	2025.	1334.	687.	132.	-294.	-562.	-653.
SURFACE 6												
WHL	4704.	4704.	4704.	4704.	4704.	4704.	4704.	4704.	4704.	4704.	4704.	4704.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	4704.	4704.	4704.	4704.	4704.	4704.	4704.	4704.	4704.	4704.	4704.	4704.
GRAND TOT	14713.	14713.	14713.	14713.	12128.	9042.	8917.	7571.	3775.	375.	-1699.	-3185.

	13	14	15	16	17	18	19	20	21	22	23	24
SURFACE 1												
WHL	253.	315.	435.	544.	946.	869.	932.	1008.	1008.	1008.	1008.	1008.
GHL	113.	123.	144.	188.	179.	-1074.	-438.	1098.	1098.	1098.	1098.	1098.
THL	365.	438.	559.	733.	1125.	-206.	494.	2106.	2106.	2106.	2106.	2106.
SURFACE 2												
WHL	337.	421.	554.	726.	1261.	1158.	1243.	1344.	1344.	1344.	1344.	1344.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	337.	421.	554.	726.	1261.	1158.	1243.	1344.	1344.	1344.	1344.	1344.
SURFACE 3												
WHL	563.	630.	733.	860.	946.	869.	932.	1008.	1008.	1008.	1008.	1008.
GHL	-8395.	-6560.	-3743.	-858.	239.	463.	895.	1098.	1098.	1098.	1098.	1098.
THL	-7832.	-5930.	-3011.	2.	1184.	1332.	1828.	2106.	2106.	2106.	2106.	2106.
SURFACE 4												
WHL	783.	595.	451.	373.	385.	530.	882.	1045.	1045.	1045.	1045.	1045.
GHL	-1614.	-4345.	-6379.	-7485.	-7411.	-5661.	-1384.	586.	586.	586.	586.	586.
THL	-831.	-3750.	-5928.	-7112.	-7026.	-5130.	-502.	1631.	1631.	1631.	1631.	1631.
SURFACE 5												
WHL	-562.	-294.	132.	687.	1334.	2025.	2669.	2822.	2822.	2822.	2822.	2822.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	-562.	-294.	132.	687.	1334.	2025.	2669.	2822.	2822.	2822.	2822.	2822.
SURFACE 6												
WHL	4704.	4704.	4704.	4704.	4704.	4704.	4704.	4704.	4704.	4704.	4704.	4704.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	4704.	4704.	4704.	4704.	4704.	4704.	4704.	4704.	4704.	4704.	4704.	4704.
GRAND TOT	-3819.	-4412.	-2990.	-261.	2582.	3883.	10436.	14713.	14713.	14713.	14713.	14713.

October 1979

APPENDIX B

VALUES FOR JUNE 21 ST.

POLAR DECLINATION = 23.4500 DEGREES

TIME (HRS)	ALTITUDE (DEGREES)	AZIMUTH (DEGREES)	RON (BTU/HR)	DIFFUSE (BTU/HR)
1	-15.466	14.262	0.000	0.000
2	-11.982	27.964	0.000	0.000
3	-6.527	40.763	0.000	0.000
4	.502	52.611	.000	.000
5	8.704	63.698	89.022	5.964
6	17.718	74.384	175.901	11.785
7	27.214	85.173	220.364	14.764
8	36.862	96.774	245.136	16.424
9	46.246	110.277	259.754	17.404
10	54.712	127.436	268.378	17.981
11	61.082	150.592	272.965	18.289
12	63.567	180.000	274.406	18.385
13	61.082	209.408	272.965	18.289
14	54.712	232.564	268.378	17.981
15	46.246	249.723	259.754	17.404
16	36.862	263.226	245.136	16.424
17	27.214	274.827	220.364	14.764
18	17.718	285.616	175.901	11.785
19	8.704	296.302	89.022	5.964
20	.502	307.389	.000	.000
21	-6.527	319.237	0.000	0.000
22	-11.982	332.036	0.000	0.000
23	-15.466	345.738	0.000	0.000
24	-16.667	360.000	0.000	0.000

October 1979

APPENDIX B

WALL SURFACE ONE ~ 0.00 DEGREES

	1	2	3	4	5	6	7	8	9	10	11	12
SURFACE 1												
WHL	590.	590.	590.	590.	456.	420.	497.	100.	-25.	-121.	-181.	-202.
GHL	643.	643.	643.	643.	-2169.	-2222.	-584.	-358.	-403.	-426.	-437.	-440.
THL	1233.	1233.	1233.	1233.	-1713.	-1802.	-87.	-258.	-428.	-547.	-618.	-642.
SURFACE 2												
WHL	787.	787.	787.	787.	448.	95.	-53.	-58.	44.	223.	455.	-269.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	787.	787.	787.	787.	448.	95.	-53.	-58.	44.	223.	455.	-269.
SURFACE 3												
WHL	590.	590.	590.	590.	336.	71.	-40.	472.	351.	254.	191.	169.
GHL	643.	643.	643.	643.	291.	-50.	-228.	-891.	-3251.	-5834.	-7588.	-8191.
THL	1233.	1233.	1233.	1233.	627.	21.	-268.	-420.	-2899.	-5580.	-7397.	-8022.
SURFACE 4												
WHL	612.	612.	612.	612.	348.	74.	-41.	489.	364.	263.	198.	175.
GHL	343.	343.	343.	343.	155.	-27.	-122.	-222.	-246.	-258.	-264.	-266.
THL	955.	955.	955.	955.	504.	47.	-163.	267.	118.	5.	-66.	-91.
SURFACE 5												
WHL	1653.	1653.	1653.	1653.	1333.	682.	11.	-614.	-1150.	-1560.	-1818.	-1906.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	1653.	1653.	1653.	1653.	1333.	682.	11.	-614.	-1150.	-1560.	-1818.	-1906.
SURFACE 6												
WHL	2755.	2755.	2755.	2755.	2755.	2755.	2755.	2755.	2755.	2755.	2755.	2755.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	2755.	2755.	2755.	2755.	2755.	2755.	2755.	2755.	2755.	2755.	2755.	2755.
GRAND TOT	8617.	8617.	8617.	8617.	3953.	1790.	2196.	1673.	-1560.	-4705.	-6690.	-8175.

	13	14	15	16	17	18	19	20	21	22	23	24
SURFACE 1												
WHL	-181.	-121.	-25.	100.	497.	420.	456.	590.	590.	590.	590.	590.
GHL	-437.	-426.	-403.	-358.	-584.	-2222.	-2169.	643.	643.	643.	643.	643.
THL	-618.	-547.	-428.	-258.	-87.	-1802.	-1713.	1233.	1233.	1233.	1233.	1233.
SURFACE 2												
WHL	-242.	-161.	-33.	133.	662.	560.	607.	787.	787.	787.	787.	787.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	-242.	-161.	-33.	133.	662.	560.	607.	787.	787.	787.	787.	787.
SURFACE 3												
WHL	191.	254.	351.	472.	497.	420.	456.	590.	590.	590.	590.	590.
GHL	-7588.	-5834.	-3251.	-891.	-310.	-103.	273.	643.	643.	643.	643.	643.
THL	-7397.	-5580.	-2899.	-420.	136.	316.	728.	1233.	1233.	1233.	1233.	1233.
SURFACE 4												
WHL	354.	173.	34.	-45.	-41.	74.	348.	612.	612.	612.	612.	612.
GHL	-1817.	-4435.	-6413.	-7515.	-7559.	-6186.	-2849.	343.	343.	343.	343.	343.
THL	-1463.	-4262.	-6379.	-7568.	-7601.	-6113.	-2501.	955.	955.	955.	955.	955.
SURFACE 5												
WHL	-1818.	-1560.	-1150.	-614.	11.	682.	1333.	1653.	1653.	1653.	1653.	1653.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	-1818.	-1560.	-1150.	-614.	11.	682.	1333.	1653.	1653.	1653.	1653.	1653.
SURFACE 6												
WHL	2755.	2755.	2755.	2755.	2755.	2755.	2755.	2755.	2755.	2755.	2755.	2755.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	2755.	2755.	2755.	2755.	2755.	2755.	2755.	2755.	2755.	2755.	2755.	2755.
GRAND TOT	-8784.	-9355.	-8134.	-5963.	-4073.	-3602.	1210.	8617.	8617.	8617.	8617.	8617.

VALUES FOR JULY 21 ST.

POLAR DECLINATION = 20.6000 DEGREES

TIME (HRS)	ALTITUDE (DEGREES)	AZIMUTH (DEGREES)	RDN (UTU/HR)	DIFFUSE (UTU/HR)
1	-18.272	14.782	0.000	0.000
2	-14.671	28.934	0.000	0.000
3	-9.057	42.086	0.000	0.000
4	-1.863	54.202	0.000	0.000
5	6.486	65.504	55.044	3.743
6	15.609	76.385	159.383	10.838
7	25.161	87.376	211.406	14.376
8	34.795	99.194	239.341	16.275
9	44.071	112.890	255.453	17.371
10	52.317	130.036	264.828	18.008
11	58.393	152.467	269.775	18.345
12	60.717	180.000	271.324	18.450
13	58.393	207.533	269.775	18.345
14	52.317	229.964	264.828	18.008
15	44.071	247.110	255.453	17.371
16	34.795	260.806	239.341	16.275
17	25.161	272.624	211.406	14.376
18	15.609	283.615	159.383	10.838
19	6.486	294.496	55.043	3.743
20	-1.863	305.798	0.000	0.000
21	-9.057	317.914	0.000	0.000
22	-14.671	331.066	0.000	0.000
23	-18.272	345.218	0.000	0.000
24	-19.517	360.000	0.000	0.000

October 1979

APPENDIX B

WALL SURFACE ONF ~ 0.00 DEGREES

	1	2	3	4	5	6	7	8	9	10	11	12
SURFACE 1												
WHL	353.	353.	353.	353.	274.	212.	283.	-106.	-232.	-330.	-392.	-413.
GHL	384.	384.	384.	384.	-1235.	-1860.	-630.	-612.	-664.	-691.	-704.	-707.
THL	737.	737.	737.	737.	-952.	-1648.	-347.	-718.	-897.	-1021.	-1095.	-1120.
SURFACE 2												
WHL	470.	470.	470.	470.	256.	-170.	-352.	-371.	-275.	-97.	136.	-550.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	470.	470.	470.	470.	256.	-170.	-352.	-371.	-275.	-97.	136.	-550.
SURFACE 3												
WHL	353.	353.	353.	353.	192.	-127.	-264.	210.	87.	-14.	-78.	-101.
GHL	384.	384.	384.	384.	164.	-252.	-463.	-1529.	-4262.	-6981.	-8766.	-9374.
THL	737.	737.	737.	737.	356.	-380.	-727.	-1319.	-4176.	-6395.	-8844.	-9875.
SURFACE 4												
WHL	366.	366.	366.	366.	199.	-132.	-273.	217.	90.	-14.	-81.	-104.
GHL	205.	205.	205.	205.	87.	-135.	-247.	-353.	-381.	-395.	-402.	-403.
THL	571.	571.	571.	571.	287.	-267.	-521.	-135.	-291.	-409.	-483.	-508.
SURFACE 5												
WHL	988.	988.	988.	988.	815.	174.	-507.	-1143.	-1689.	-2107.	-2369.	-2459.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	988.	988.	988.	988.	815.	174.	-507.	-1143.	-1689.	-2107.	-2369.	-2459.
SURFACE 6												
WHL	1646.	1646.	1646.	1646.	1646.	1646.	1646.	1646.	1646.	1646.	1646.	1646.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	1646.	1646.	1646.	1646.	1646.	1646.	1646.	1646.	1646.	1646.	1646.	1646.
GRAND TOT	5149.	5149.	5149.	5149.	2399.	-644.	-807.	-2040.	-5682.	-8983.	-11010.	-12465.

	13	14	15	16	17	18	19	20	21	22	23	24
SURFACE 1												
WHL	-392.	-310.	-232.	-106.	283.	212.	274.	353.	353.	353.	353.	353.
GHL	-704.	-691.	-668.	-612.	-630.	-1860.	-1235.	384.	384.	384.	384.	384.
THL	-1095.	-1021.	-897.	-718.	-347.	-1648.	-962.	737.	737.	737.	737.	737.
SURFACE 2												
WHL	-522.	-440.	-310.	-141.	378.	283.	365.	470.	470.	470.	470.	470.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	-522.	-440.	-310.	-141.	378.	283.	365.	470.	470.	470.	470.	470.
SURFACE 3												
WHL	-78.	-14.	87.	210.	283.	212.	274.	353.	353.	353.	353.	353.
GHL	-8766.	-6981.	-4262.	-1529.	-507.	-305.	151.	384.	384.	384.	384.	384.
THL	-8844.	-6995.	-4176.	-1319.	-284.	-93.	425.	737.	737.	737.	737.	737.
SURFACE 4												
WHL	105.	-76.	-214.	-288.	-273.	-132.	199.	366.	366.	366.	366.	366.
GHL	-1992.	-4631.	-6588.	-7637.	-7518.	-5833.	-1811.	205.	205.	205.	205.	205.
THL	-1887.	-4706.	-6802.	-7925.	-7812.	-5965.	-1612.	571.	571.	571.	571.	571.
SURFACE 5												
WHL	-2369.	-2107.	-1689.	-1143.	-507.	174.	815.	988.	988.	988.	988.	988.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	-2369.	-2107.	-1689.	-1143.	-507.	174.	815.	988.	988.	988.	988.	988.
SURFACE 6												
WHL	1646.	1646.	1646.	1646.	1646.	1646.	1646.	1646.	1646.	1646.	1646.	1646.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	1646.	1646.	1646.	1646.	1646.	1646.	1646.	1646.	1646.	1646.	1646.	1646.
GRAND TOT	-13071.	-13623.	-12227.	-9600.	-6905.	-5602.	677.	5149.	5149.	5149.	5149.	5149.

VALUES FOR AUG. 21 ST.

POLAR DECLINATION = 12.3000 DEGREES

TIME (HRS)	ALTITUDE (DEGREES)	AZIMUTH (DEGREES)	RON (BTU/HR)	DIFFUSE (BTU/HR)
1	-26.436	16.404	0.000	0.000
2	-22.476	31.916	0.000	0.000
3	-16.395	46.067	0.000	0.000
4	-8.735	58.879	0.000	0.000
5	-0.002	70.692	0.000	0.000
6	9.376	82.003	102.206	6.235
7	19.017	93.403	189.415	11.554
8	28.535	105.605	230.445	14.057
9	37.450	119.512	252.203	15.384
10	45.082	136.222	264.261	16.120
11	50.445	156.603	270.451	16.498
12	52.417	180.000	272.366	16.614
13	50.445	203.397	270.451	16.498
14	45.082	223.778	264.261	16.120
15	37.450	240.488	252.203	15.384
16	28.535	254.395	230.445	14.057
17	19.017	266.597	189.415	11.554
18	9.376	277.997	102.205	6.235
19	-0.002	289.308	0.000	0.000
20	-8.735	301.121	0.000	0.000
21	-16.395	313.933	0.000	0.000
22	-22.476	328.084	0.000	0.000
23	-26.436	343.596	0.000	0.000
24	-27.817	360.000	0.000	0.000

October 1979

APPENDIX B

WALL SURFACE ONF ^ 0.00 DEGREES

	1	2	3	4	5	6	7	8	9	10	11	12
SURFACE 1												
WHL	475.	475.	475.	475.	475.	414.	255.	103.	-31.	-135.	-200.	-222.
GHL	517.	517.	517.	517.	517.	-285.	-206.	-347.	-413.	-445.	-468.	-464.
THL	993.	993.	993.	993.	993.	129.	49.	-244.	-444.	-580.	-660.	-686.
SURFACE 2												
WHL	634.	634.	634.	634.	634.	209.	-128.	-203.	-125.	53.	294.	-296.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	634.	634.	634.	634.	634.	209.	-128.	-203.	-125.	53.	294.	-296.
SURFACE 3												
WHL	475.	475.	475.	475.	475.	157.	409.	270.	133.	23.	-48.	-73.
GHL	517.	517.	517.	517.	517.	158.	-364.	-2788.	-6335.	-9320.	-11185.	-11817.
THL	993.	993.	993.	993.	993.	315.	45.	-2519.	-6202.	-9297.	-11234.	-11890.
SURFACE 4												
WHL	493.	493.	493.	493.	493.	163.	424.	280.	138.	24.	-50.	-76.
GHL	776.	776.	776.	776.	776.	84.	-123.	-199.	-234.	-251.	-258.	-260.
THL	769.	769.	769.	769.	769.	247.	301.	81.	-96.	-227.	-308.	-336.
SURFACE 5												
WHL	1331.	1331.	1331.	1331.	1331.	964.	262.	-411.	-989.	-1433.	-1713.	-1808.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	1331.	1331.	1331.	1331.	1331.	964.	262.	-411.	-989.	-1433.	-1713.	-1808.
SURFACE 6												
WHL	2218.	2218.	2218.	2218.	2218.	2218.	2218.	2218.	2218.	2218.	2218.	2218.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	2218.	2218.	2218.	2218.	2218.	2218.	2218.	2218.	2218.	2218.	2218.	2218.
GRAND TOT	6936.	6936.	6936.	6936.	6936.	4081.	2747.	-1078.	-5639.	-9267.	-11403.	-12798.

	13	14	15	16	17	18	19	20	21	22	23	24
SURFACE 1												
WHL	-200.	-135.	-31.	103.	255.	414.	475.	475.	475.	475.	475.	475.
GHL	-460.	-445.	-413.	-347.	-206.	-285.	517.	517.	517.	517.	517.	517.
THL	-660.	-580.	-444.	-244.	49.	129.	993.	993.	993.	993.	993.	993.
SURFACE 2												
WHL	-266.	-179.	-41.	137.	340.	553.	634.	634.	634.	634.	634.	634.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	-266.	-179.	-41.	137.	340.	553.	634.	634.	634.	634.	634.	634.
SURFACE 3												
WHL	-48.	23.	133.	270.	409.	414.	475.	475.	475.	475.	475.	475.
GHL	-11105.	-9320.	-6335.	-2788.	-364.	118.	517.	517.	517.	517.	517.	517.
THL	-11234.	-9297.	-6202.	-2519.	45.	533.	993.	993.	993.	993.	993.	993.
SURFACE 4												
WHL	229.	41.	-97.	-158.	-99.	163.	493.	493.	493.	493.	493.	493.
GHL	-1979.	-4727.	-6072.	-7559.	-6909.	-3723.	276.	276.	276.	276.	276.	276.
THL	-1750.	-4686.	-6769.	-7717.	-7008.	-3561.	769.	769.	769.	769.	769.	769.
SURFACE 5												
WHL	-1713.	-1433.	-989.	-411.	262.	964.	1331.	1331.	1331.	1331.	1331.	1331.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	-1713.	-1433.	-989.	-411.	262.	964.	1331.	1331.	1331.	1331.	1331.	1331.
SURFACE 6												
WHL	2218.	2218.	2218.	2218.	2218.	2218.	2218.	2218.	2218.	2218.	2218.	2218.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	2218.	2218.	2218.	2218.	2218.	2218.	2218.	2218.	2218.	2218.	2218.	2218.
GRAND TOT	-13405.	-13959.	-12228.	-8535.	-4094.	835.	6936.	6936.	6936.	6936.	6936.	6936.

VALUES FOR SEPT 21 ST.

POLAR DECLINATION = 0.000 DEGREES			
TIME (HRS)	ALTITUDE (DEGREES)	AZIMUTH (DEGREES)	RDN (BTU/HR)
1	-38.491	19.310	0.000
2	-33.919	37.052	0.000
3	-27.105	52.594	0.000
4	-18.794	66.178	0.000
5	-9.600	78.420	0.000
6	.000	90.000	0.000
7	9.600	101.580	126.286
8	18.794	113.822	210.714
9	27.105	127.406	247.502
10	33.919	142.948	265.789
11	38.491	160.690	274.653
12	40.117	180.000	277.328
13	38.491	199.310	274.653
14	33.919	217.052	265.789
15	27.105	232.594	247.502
16	18.794	246.178	210.714
17	9.600	258.420	126.286
18	-.000	270.000	0.000
19	-9.600	281.580	0.000
20	-18.794	293.822	0.000
21	-27.105	307.406	0.000
22	-33.919	322.948	0.000
23	-38.491	340.690	0.000
24	-40.117	360.000	0.000

DIFFUSE
(BTU/HR)

October 1979

APPENDIX B

HALL SURFACE ONT ^ 0.00 OFGRFES

	1	2	3	4	5	6	7	8	9	10	11	12
SURFACE 1												
WHL	922.	922.	922.	922.	922.	922.	841.	689.	549.	440.	371.	347.
GHL	1004.	1004.	1004.	1004.	1004.	1004.	635.	404.	315.	276.	260.	256.
THL	1925.	1925.	1925.	1925.	1925.	1925.	1476.	1093.	864.	716.	631.	603.
SURFACE 2												
WHL	1229.	1229.	1229.	1229.	1229.	1229.	718.	460.	483.	648.	894.	463.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	1229.	1229.	1229.	1229.	1229.	1229.	718.	460.	483.	648.	894.	463.
SURFACE 3												
WHL	922.	922.	922.	922.	922.	922.	829.	651.	486.	357.	275.	247.
GHL	1004.	1004.	1004.	1004.	1004.	1004.	-334.	-4358.	-8595.	-11860.	-13915.	-14622.
THL	1925.	1925.	1925.	1925.	1925.	1925.	495.	-3707.	-8109.	-11584.	-13648.	-14375.
SURFACE 4												
WHL	955.	955.	955.	955.	955.	955.	860.	675.	504.	370.	285.	256.
GHL	536.	536.	536.	536.	536.	536.	340.	219.	173.	154.	147.	145.
THL	1491.	1491.	1491.	1491.	1491.	1491.	1200.	894.	677.	524.	432.	401.
SURFACE 5												
WHL	2580.	2580.	2580.	2580.	2580.	2580.	2169.	1481.	873.	404.	108.	7.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	2580.	2580.	2580.	2580.	2580.	2580.	2169.	1481.	873.	404.	108.	7.
SURFACE 6												
WHL	4301.	4301.	4301.	4301.	4301.	4301.	4301.	4301.	4301.	4301.	4301.	4301.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	4301.	4301.	4301.	4301.	4301.	4301.	4301.	4301.	4301.	4301.	4301.	4301.
GRAND TOT	13452.	13452.	13452.	13452.	13452.	13452.	10358.	4521.	-912.	-4910.	-7274.	-8500.

	13	14	15	16	17	18	19	20	21	22	23	24
SURFACE 1												
WHL	371.	440.	549.	689.	841.	922.	922.	922.	922.	922.	922.	922.
GHL	260.	276.	315.	404.	635.	1004.	1004.	1004.	1004.	1004.	1004.	1004.
THL	631.	716.	864.	1093.	1476.	1925.	1925.	1925.	1925.	1925.	1925.	1925.
SURFACE 2												
WHL	494.	587.	732.	918.	1121.	1229.	1229.	1229.	1229.	1229.	1229.	1229.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	494.	587.	732.	918.	1121.	1229.	1229.	1229.	1229.	1229.	1229.	1229.
SURFACE 3												
WHL	275.	357.	486.	651.	829.	922.	922.	922.	922.	922.	922.	922.
GHL	-13915.	-11860.	-8595.	-4358.	-334.	1004.	1004.	1004.	1004.	1004.	1004.	1004.
THL	-13640.	-11504.	-8109.	-3707.	495.	1925.	1925.	1925.	1925.	1925.	1925.	1925.
SURFACE 4												
WHL	695.	504.	376.	358.	558.	955.	955.	955.	955.	955.	955.	955.
GHL	-1680.	-4494.	-6320.	-6693.	-4291.	536.	536.	536.	536.	536.	536.	536.
THL	-985.	-3990.	-5944.	-6335.	-3733.	1491.	1491.	1491.	1491.	1491.	1491.	1491.
SURFACE 5												
WHL	108.	404.	873.	1481.	2169.	2580.	2580.	2580.	2580.	2580.	2580.	2580.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	108.	404.	873.	1481.	2169.	2580.	2580.	2580.	2580.	2580.	2580.	2580.
SURFACE 6												
WHL	4301.	4301.	4301.	4301.	4301.	4301.	4301.	4301.	4301.	4301.	4301.	4301.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	4301.	4301.	4301.	4301.	4301.	4301.	4301.	4301.	4301.	4301.	4301.	4301.
GRAND TOT	-9090.	-9486.	-7284.	-2249.	5828.	13452.	13452.	13452.	13452.	13452.	13452.	13452.

VALUES FOR OCT. 21 ST.

POLAR DECLINATION = -10.5000 DEGREES

TIME (HRS)	ALTITUDE (DEGREES)	AZIMUTH (DEGREES)	RDN (UTU/HR)	DIFFUSE (UTU/HR)
1	-48.706	22.683	0.000	0.000
2	-43.475	42.647	0.000	0.000
3	-35.969	59.212	0.000	0.000
4	-27.138	73.111	0.000	0.000
5	-17.658	85.354	0.000	0.000
6	-8.011	96.810	0.000	0.000
7	1.410	108.188	.568	.021
8	10.219	120.089	153.402	5.599
9	17.977	133.034	225.084	8.216
10	24.162	147.396	255.698	9.333
11	28.204	163.216	269.440	9.835
12	29.617	180.000	273.455	9.981
13	28.204	196.784	269.440	9.835
14	24.162	212.604	255.698	9.333
15	17.977	226.966	225.084	8.216
16	10.219	239.911	153.402	5.599
17	1.410	251.812	.568	.021
18	-8.011	263.190	0.000	0.000
19	-17.658	274.647	0.000	0.000
20	-27.138	286.889	0.000	0.000
21	-35.969	300.788	0.000	0.000
22	-43.475	317.353	0.000	0.000
23	-48.706	337.317	0.000	0.000
24	-50.617	360.000	0.000	0.000

October 1979

APPENDIX B

WALL SURFACE ONF ~ 0.00 DEGREES

	1	2	3	4	5	6	7	8	9	10	11	12
SURFACE 1												
WHL	1318.	1318.	1318.	1318.	1318.	1318.	1317.	1219.	1085.	976.	906.	882.
GHL	1435.	1435.	1435.	1435.	1435.	1435.	1433.	1083.	932.	876.	853.	847.
THL	2752.	2752.	2752.	2752.	2752.	2752.	2751.	2302.	2017.	1851.	1760.	1730.
SURFACE 2												
WHL	1757.	1757.	1757.	1757.	1757.	1757.	1755.	1212.	1098.	1217.	1443.	1176.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	1757.	1757.	1757.	1757.	1757.	1757.	1755.	1212.	1098.	1217.	1443.	1176.
SURFACE 3												
WHL	1318.	1318.	1318.	1318.	1318.	1318.	1317.	1074.	855.	700.	606.	574.
GHL	1435.	1435.	1435.	1435.	1435.	1435.	1434.	3822.	-9090.	-12891.	-15241.	-16034.
THL	2752.	2752.	2752.	2752.	2752.	2752.	2741.	-2748.	-8235.	-12191.	-14635.	-15459.
SURFACE 4												
WHL	1366.	1366.	1366.	1366.	1366.	1366.	1365.	1113.	886.	726.	628.	596.
GHL	766.	766.	766.	766.	766.	766.	765.	590.	517.	490.	480.	478.
THL	2132.	2132.	2132.	2132.	2132.	2132.	2131.	1703.	1403.	1216.	1109.	1073.
SURFACE 5												
WHL	3689.	3689.	3689.	3689.	3689.	3689.	3689.	3205.	2607.	2135.	1837.	1735.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	3689.	3689.	3689.	3689.	3689.	3689.	3689.	3205.	2607.	2135.	1837.	1735.
SURFACE 6												
WHL	6149.	6149.	6149.	6149.	6149.	6149.	6149.	6149.	6149.	6149.	6149.	6149.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	6149.	6149.	6149.	6149.	6149.	6149.	6149.	6149.	6149.	6149.	6149.	6149.
GRAND TOT	19232.	19232.	19232.	19232.	19232.	19232.	19215.	11823.	5038.	377.	-2338.	-3596.

	13	14	15	16	17	18	19	20	21	22	23	24
SURFACE 1												
WHL	906.	976.	1085.	1219.	1317.	1318.	1318.	1318.	1318.	1318.	1318.	1318.
GHL	853.	976.	932.	1083.	1433.	1435.	1435.	1435.	1435.	1435.	1435.	1435.
THL	1760.	1851.	2017.	2302.	2751.	2752.	2752.	2752.	2752.	2752.	2752.	2752.
SURFACE 2												
WHL	1208.	1301.	1446.	1626.	1757.	1757.	1757.	1757.	1757.	1757.	1757.	1757.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	1208.	1301.	1446.	1626.	1757.	1757.	1757.	1757.	1757.	1757.	1757.	1757.
SURFACE 3												
WHL	606.	700.	855.	1074.	1317.	1318.	1318.	1318.	1318.	1318.	1318.	1318.
GHL	-15241.	-12891.	-9050.	-3822.	1424.	1435.	1435.	1435.	1435.	1435.	1435.	1435.
THL	-14635.	-12191.	-8235.	-2748.	2741.	2752.	2752.	2752.	2752.	2752.	2752.	2752.
SURFACE 4												
WHL	1122.	946.	854.	942.	1364.	1366.	1366.	1366.	1366.	1366.	1366.	1366.
GHL	-1269.	-3893.	-5284.	-4353.	745.	766.	766.	766.	766.	766.	766.	766.
THL	-147.	-2947.	-4431.	-3411.	2109.	2132.	2132.	2132.	2132.	2132.	2132.	2132.
SURFACE 5												
WHL	1837.	2135.	2607.	3205.	3689.	3689.	3689.	3689.	3689.	3689.	3689.	3689.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	1837.	2135.	2607.	3205.	3689.	3689.	3689.	3689.	3689.	3689.	3689.	3689.
SURFACE 6												
WHL	6149.	6149.	6149.	6149.	6149.	6149.	6149.	6149.	6149.	6149.	6149.	6149.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	6149.	6149.	6149.	6149.	6149.	6149.	6149.	6149.	6149.	6149.	6149.	6149.
GRAND TOT	-3829.	-3702.	-447.	7122.	19195.	19232.	19232.	19232.	19232.	19232.	19232.	19232.

VALUES FOR NOV. 21 ST.

POLAR DECLINATION = -19.8000 DEGREES

TIME (HRS)	ALTITUDE (DEGREES)	AZIMUTH (DEGREES)	RON (BTU/HR)	DIFFUSE (BTU/HR)
1	-57.633	27.058	0.000	0.000
2	-51.635	49.285	0.000	0.000
3	-43.451	66.411	0.000	0.000
4	-34.206	80.149	0.000	0.000
5	-24.579	92.019	0.000	0.000
6	-15.013	103.060	0.000	0.000
7	-5.862	113.993	0.000	0.000
8	2.527	125.351	13.176	.415
9	9.767	137.539	160.790	5.065
10	15.426	150.789	221.018	6.962
11	19.060	165.070	245.216	7.724
12	20.317	180.000	251.964	7.937
13	19.060	194.930	245.216	7.724
14	15.426	209.211	221.018	6.962
15	9.767	222.461	160.790	5.065
16	2.527	234.649	13.175	.415
17	-5.862	246.008	0.000	0.000
18	-15.013	256.940	0.000	0.000
19	-24.579	267.981	0.000	0.000
20	-34.206	279.851	0.000	0.000
21	-43.451	293.589	0.000	0.000
22	-51.635	310.715	0.000	0.000
23	-57.633	332.942	0.000	0.000
24	-59.917	360.000	0.000	0.000

October 1979

APPENDIX B

WALL SURFACE ONF ^ 0.00 DEGREES

	1	2	3	4	5	6	7	8	9	10	11	12
SURFACE 1												
WHL	2131.	2131.	2131.	2131.	2131.	2131.	2131.	2128.	2034.	1934.	1868.	1845.
GHL	2321.	2321.	2321.	2321.	2321.	2321.	2321.	2294.	2004.	1895.	1855.	1844.
THL	4452.	4452.	4452.	4452.	4452.	4452.	4452.	4422.	4038.	3829.	3723.	3689.
SURFACE 2												
WHL	2842.	2842.	2842.	2842.	2842.	2842.	2842.	2797.	2393.	2398.	2572.	2460.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	2842.	2842.	2842.	2842.	2842.	2842.	2842.	2797.	2393.	2398.	2572.	2460.
SURFACE 3												
WHL	2131.	2131.	2131.	2131.	2131.	2131.	2131.	2107.	1765.	1552.	1436.	1399.
GHL	2321.	2321.	2321.	2321.	2321.	2321.	2321.	1783.	-6117.	-11246.	-14084.	-14987.
THL	4452.	4452.	4452.	4452.	4452.	4452.	4452.	3890.	-4351.	-9693.	-12648.	-13588.
SURFACE 4												
WHL	2209.	2209.	2209.	2209.	2209.	2209.	2209.	2184.	1830.	1609.	1489.	1450.
GHL	1239.	1239.	1239.	1239.	1239.	1239.	1239.	1027.	1092.	1043.	1026.	1022.
THL	3449.	3449.	3449.	3449.	3449.	3449.	3449.	3411.	2922.	2652.	2515.	2471.
SURFACE 5												
WHL	5967.	5967.	5967.	5967.	5967.	5967.	5967.	5950.	5496.	5051.	4764.	4665.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	5967.	5967.	5967.	5967.	5967.	5967.	5967.	5950.	5496.	5051.	4764.	4665.
SURFACE 6												
WHL	9946.	9946.	9946.	9946.	9946.	9946.	9946.	9946.	9946.	9946.	9946.	9946.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	9946.	9946.	9946.	9946.	9946.	9946.	9946.	9946.	9946.	9946.	9946.	9946.
GRAND TOT	11107.	31107.	31107.	31107.	31107.	31107.	31107.	30415.	20444.	14182.	10871.	9643.

	13	14	15	16	17	18	19	20	21	22	23	24
SURFACE 1												
WHL	1868.	1934.	2034.	2128.	2131.	2131.	2131.	2131.	2131.	2131.	2131.	2131.
GHL	1855.	1895.	2004.	2294.	2321.	2321.	2321.	2321.	2321.	2321.	2321.	2321.
THL	3723.	3829.	4038.	4422.	4452.	4452.	4452.	4452.	4452.	4452.	4452.	4452.
SURFACE 2												
WHL	2490.	2579.	2712.	2838.	2842.	2842.	2842.	2842.	2842.	2842.	2842.	2842.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	2490.	2579.	2712.	2838.	2842.	2842.	2842.	2842.	2842.	2842.	2842.	2842.
SURFACE 3												
WHL	1436.	1552.	1765.	2107.	2131.	2131.	2131.	2131.	2131.	2131.	2131.	2131.
GHL	-14084.	-11246.	-6117.	1783.	2321.	2321.	2321.	2321.	2321.	2321.	2321.	2321.
THL	-12648.	-9693.	-4351.	3890.	4452.	4452.	4452.	4452.	4452.	4452.	4452.	4452.
SURFACE 4												
WHL	2000.	1864.	1861.	2175.	2209.	2209.	2209.	2209.	2209.	2209.	2209.	2209.
GHL	-460.	-2540.	-2856.	822.	1239.	1239.	1239.	1239.	1239.	1239.	1239.	1239.
THL	1540.	-676.	-995.	2997.	3449.	3449.	3449.	3449.	3449.	3449.	3449.	3449.
SURFACE 5												
WHL	4764.	5051.	5496.	5950.	5967.	5967.	5967.	5967.	5967.	5967.	5967.	5967.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	4764.	5051.	5496.	5950.	5967.	5967.	5967.	5967.	5967.	5967.	5967.	5967.
SURFACE 6												
WHL	9946.	9946.	9946.	9946.	9946.	9946.	9946.	9946.	9946.	9946.	9946.	9946.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	9946.	9946.	9946.	9946.	9946.	9946.	9946.	9946.	9946.	9946.	9946.	9946.
GRAND TOT	9815.	11035.	16846.	30041.	31107.	31107.	31107.	31107.	31107.	31107.	31107.	31107.

VALUES FOR DEC. 21 ST.

POLAR DECLINATION = -23.4500 DEGREES

TIME (HRS)	ALTITUDE (DEGREES)	AZIMUTH (DEGREES)	RDN (BTU/HR)	DIFFUSE (BTU/HR)
1	-61.082	29.408	0.000	0.000
2	-54.712	52.564	0.000	0.000
3	-46.246	69.723	0.000	0.000
4	-36.862	83.226	0.000	0.000
5	-27.214	94.827	0.000	0.000
6	-17.718	105.616	0.000	0.000
7	-8.704	116.302	0.000	0.000
8	-502	127.389	0.000	0.000
9	6.527	139.237	112.105	3.195
10	11.982	152.036	197.300	5.623
11	15.466	165.738	229.567	6.543
12	16.667	180.000	238.315	6.792
13	15.466	194.262	229.567	6.543
14	11.982	207.964	197.300	5.623
15	6.527	220.763	112.104	3.195
16	-502	232.611	0.000	0.000
17	-8.704	243.698	0.000	0.000
18	-17.718	254.384	0.000	0.000
19	-27.214	265.173	0.000	0.000
20	-36.862	276.774	0.000	0.000
21	-46.246	290.277	0.000	0.000
22	-54.712	307.436	0.000	0.000
23	-61.082	330.592	0.000	0.000
24	-63.567	360.000	0.000	0.000

October 1979

APPENDIX B

HALL SURFACE ONF ^ 0.00 DEGREES

	1	2	3	4	5	6	7	8	9	10	11	12
SURFACE 1												
WHL	2707.	2707.	2707.	2707.	2707.	2707.	2707.	2707.	2659.	2567.	2504.	2482.
GHL	2948.	2948.	2948.	2948.	2948.	2948.	2948.	2948.	2746.	2601.	2550.	2537.
THL	5655.	5655.	5655.	5655.	5655.	5655.	5655.	5655.	5405.	5168.	5054.	5019.
SURFACE 2												
WHL	3610.	3610.	3610.	3610.	3610.	3610.	3610.	3610.	3306.	3225.	3365.	3309.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	3610.	3610.	3610.	3610.	3610.	3610.	3610.	3610.	3306.	3225.	3365.	3309.
SURFACE 3												
WHL	2707.	2707.	2707.	2707.	2707.	2707.	2707.	2707.	2445.	2179.	2044.	2002.
GHL	2948.	2948.	2948.	2948.	2948.	2948.	2948.	2948.	-3134.	-9471.	-12715.	-13713.
THL	5655.	5655.	5655.	5655.	5655.	5655.	5655.	5655.	-690.	-7292.	-10671.	-11711.
SURFACE 4												
WHL	2806.	2806.	2806.	2806.	2806.	2806.	2806.	2806.	2534.	2259.	2119.	2075.
GHL	1574.	1574.	1574.	1574.	1574.	1574.	1574.	1574.	1484.	1421.	1400.	1394.
THL	4381.	4381.	4381.	4381.	4381.	4381.	4381.	4381.	4018.	3680.	3519.	3470.
SURFACE 5												
WHL	7580.	7580.	7580.	7580.	7580.	7580.	7580.	7580.	7339.	6922.	6644.	6548.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	7580.	7580.	7580.	7580.	7580.	7580.	7580.	7580.	7339.	6922.	6644.	6548.
SURFACE 6												
WHL	12634.	12634.	12634.	12634.	12634.	12634.	12634.	12634.	12634.	12634.	12634.	12634.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	12634.	12634.	12634.	12634.	12634.	12634.	12634.	12634.	12634.	12634.	12634.	12634.
GRAND TOT	39514.	39514.	39514.	39514.	39514.	39514.	39514.	39514.	32012.	24337.	20545.	19269.

	13	14	15	16	17	18	19	20	21	22	23	24
SURFACE 1												
WHL	2504.	2567.	2659.	2707.	2707.	2707.	2707.	2707.	2707.	2707.	2707.	2707.
GHL	2550.	2601.	2746.	2948.	2948.	2948.	2948.	2948.	2948.	2948.	2948.	2948.
THL	5054.	5168.	5405.	5655.	5655.	5655.	5655.	5655.	5655.	5655.	5655.	5655.
SURFACE 2												
WHL	3339.	3423.	3546.	3610.	3610.	3610.	3610.	3610.	3610.	3610.	3610.	3610.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	3339.	3423.	3546.	3610.	3610.	3610.	3610.	3610.	3610.	3610.	3610.	3610.
SURFACE 3												
WHL	2044.	2179.	2445.	2707.	2707.	2707.	2707.	2707.	2707.	2707.	2707.	2707.
GHL	-12715.	-9471.	-3134.	2948.	2948.	2948.	2948.	2948.	2948.	2948.	2948.	2948.
THL	-10671.	-7292.	-690.	5655.	5655.	5655.	5655.	5655.	5655.	5655.	5655.	5655.
SURFACE 4												
WHL	2617.	2508.	2570.	2806.	2806.	2806.	2806.	2806.	2806.	2806.	2806.	2806.
GHL	64.	-1675.	-1192.	1574.	1574.	1574.	1574.	1574.	1574.	1574.	1574.	1574.
THL	2681.	833.	1379.	4381.	4381.	4381.	4381.	4381.	4381.	4381.	4381.	4381.
SURFACE 5												
WHL	6644.	6922.	7339.	7580.	7580.	7580.	7580.	7580.	7580.	7580.	7580.	7580.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	6644.	6922.	7339.	7580.	7580.	7580.	7580.	7580.	7580.	7580.	7580.	7580.
SURFACE 6												
WHL	12634.	12634.	12634.	12634.	12634.	12634.	12634.	12634.	12634.	12634.	12634.	12634.
GHL	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
THL	12634.	12634.	12634.	12634.	12634.	12634.	12634.	12634.	12634.	12634.	12634.	12634.
GRAND TOT	19680.	21688.	29613.	39514.	39514.	39514.	39514.	39514.	39514.	39514.	39514.	39514.

APPENDIX C

APPENDIX C: A VIDEO TAPE OF THE GRAPHIX SYSTEM

A black and white video tape of the GRAPHIX system is appended to this thesis. The video tape depicts a working session at a 4002A graphics terminal, when inputting a building description into the computer using the GRAPHIX system. It is hoped that by this device the reader may gain an overview of the work conducted in this thesis.

BIBLIOGRAPHY

BOOKS

- [1] Arbib, Michael A. Theories of Abstract Automata.
Prentice-Hall Inc., 1969.

- [2] *Carr, C. Stephen. Geometric Modelling. Ph.D. Thesis,
University of Utah, 1969.

- [3] *Canada Geographic Information System: Graphics Sub-system
Commands. Canada Land Data Systems Division, Lands
Directorate, Environment Canada, Ottawa, October, 1977.

- [4] Chen, Wai-Kai. Applied Graph Theory. New York: American
Elsevier Publishing Company, 1971.

- [5] Codd, E. F. Cellular Automata. Academic Press Inc., 1968.

- [6] *Computer-Aided Design Weds Scientific Progress to
Designer's Art. Public Works Canada, Computer-Aided
Design Centre, Ottawa. August, 1977.

* An asterix marks those texts/articles that are
directly referenced in the body of the thesis.

- [7] Computer Applications in Architecture and Engineering, ed. G.N. Harper. New York:McGraw-Hill Book Company, 1968.
- [8] Computer Graphics in Architecture and Design, ed. M. Milne. New Haven: Yale School of Architecture, 1969.
- [9] *Dakin, R. A. Some Micro-climatic Effects on the Siting and Orientation of Buildings in a Northern Environment. M. Architecture Thesis, University of Manitoba, 1973.
- [10]*Dorf, Richard C. Computers and Man. California: Boyd and Fraser Publishing Company, 1974.
- [11] Emerging Methods In Environmental Design And Planning, ed. Gary T. Moore. M.I.T. Press, 1970.
- [12] Environmental Design: Research and Practice, ed. W. Mitchell. Proceedings of the edra 3/ar 8 conference . Vol. 2, January 1972.
- [13]*Grason, John. Methods for the Computer-implemented Solution of a Class of "Floor Plan" Design Problems. Phd. Thesis, Carnagie-Mellon University, 1970.

* An asterix marks those texts/articles that are directly referenced in the body of the thesis.

- [14]*Harary, F. Graph Theory. Don Mills, Ontario: Addison-Wesley Publishing Company, 1969.

- [15] Hendrin, Philip. Experiments In Form Using Computer Graphics. University of Texas Press, 1969.

- [16]*Harper, Robert H. "An Administrative Organization for Data Management and Information System Development" in the Proceedings of the Twelfth Annual Conference of the Urban and Regional Information Systems Association. Montreal, Quebec, August, 1974.

- [17] Hopcraft, J. and Ullman, J. Formal Languages And Their Relations To Automata. Addison-Wesley Publishing Co., 1969.

- [18]*Jackson, Harold E. PLUS: an evaluation. Department of Landscape Architecture, Faculty of Architecture, University of Manitoba, 1973.

- [19] Joanson, T., Perkins, J. and Weinzapel, G. Image: An Interactive Computer System For Multi-constrained Spatial Synthesis. Proceedings of Design Automation Workshop. Atlantic City, New Jersey. N.O.

* An asterix marks those texts/articles that are directly referenced in the body of the thesis.

- [20] Kentucky Workshop on Computer Applications to Environmental Design, ed. Michael Kennedy. University of Kentucky, Lexington, 1970.
- [21] Leowy and Siekeritz, Cell Structure And Function. Holt, Rinehart and Winston Inc., 1969.
- [22] March, Lionel, and Steadman, Philip. The Geometry of Environment. London: RIBA Publications Limited, 1971.
- [23]*Maxwell, L.M., and Reed, M.B. Applied Graph Theory. New York: American Elsevier Publishing Company, 1971.
- [24]*Mitchell, William John. Computer-aided Architectural Design. New York: Petrocelli/Charter, 1977.
- [25] Minsky, Marvin L. Computation: Finite And Infinite Machines. Prentice-Hall Inc., 1967.
- [26]*Newman, William M., and Sproull, Robert F. Principles of Interactive Computer Graphics. New York: McGraw-Hill Book Company, 1973.

* An asterix marks those texts/articles that are directly referenced in the body of the thesis.

- [27]*Parslow, R.D., Prowse, R.W., and Green, R.E. Computer Graphics, Techniques and Applications. New York: Plenum Press, 1969.
- [28]*Read, Ronald C. Graph Theory and Computing. New York: Academic Press, 1972.
- [29]*Roberts, Lawrence A. Machine Perception of Three Dimensional Solids, Research Report TR 315. Massachusetts Institute of Technology, Lincoln Laboratory, 1965.
- [30]*Spatial Synthesis in Computer Aided Building Design, ed. Charles Eastman. London: Applied Science Publishers Ltd., 1975.
- [31]*Sutherland, Ivan E. SKETCHPAD: a man-machine graphical communication system. Lexington Massachusetts: Massachusetts Institute of Technology, Lincoln Laboratory, 1963.
- [32]*Symons, D.C. A Parcel Geocoding System for Urban and Rural Information. National Capital Commission, Ottawa, Ontario, September 23, 1970.

* An asterix marks those texts/articles that are directly referenced in the body of the thesis.

- [33] *Teague, Lavette C. Jr. The Representation of Spatial Relationships In A Computer System For Building Design, Research Report R68-25. Massachusetts Institute of Technology, Cambridge, Massachusetts, June 1968.
- [34] Thomas, R.K. Three-Dimensional Design A Cellular Approach. Reinhold, 1969.
- [35] Wells, Mark B. Elements of Combinatorial Programming. Pergammon Press, 1970.

PERIODICALS.

- [36] Agra, O.M., and Whitehead, B. "A Study of Movement in School Building". Building Science: vol. 2, 1968. pp. 279-289.
- [37] Cousin, Jean. "Topological Organization of Architectural Space". Architectural Design: October 1970.
- [38] Davidson, Colin H., and Roberts, John, "An operating system of building information". BUILD International: May/June 1972.

* An asterix marks those texts/articles that are directly referenced in the body of the thesis.

- [39]*Dean, Perry, and Stewart, "The ARK 2 System". Progressive Architecture, July 1971.

- [40] Eastman, Charles. "Representations for Space Planning". Communications of the A.C.M.: vol. 13. No. 4. April 1970.

- [41] Hafez, E., Agraa, O.M., and Whitehead, B., "Automation of Data Preparation in Computer Program for the Planning of Single-storey Layouts". Building Science: Vol. 2, 1967. pp. 83-88.

- [42] Levin, P.H. "The Use of Graphs to Decide the Optimum Layout of Buildings". Architects Journal: 7th October 1964. pp. 809-815.

- [43]*Maver, Thomas W., "PACE1: Computer Aided Building Appraisal". The Architects' Journal: July 28, 1971.

- [44]*Negroponte, Nicholas, "Toward a Theory of Architecture Machines". A.I.A. Journal: 51, March 1969, No. 3, pp. 71-74.

* An asterix marks those texts/articles that are directly referenced in the body of the thesis.

- [45]*Paton, K. "An Algorithm for the Blocks and Cutnodes of a Graph". Communications of the A.C.M.: vol. 14, no. 7, July 1971.
- [46] "Research in action: an integrated design study applied to schools development". Architectural Research and Teaching: vol. 1, No. 2, November 1970.
- [47]*Wood, R.C. and Hendren, P.H., "A Flexible Computer Graphic System for Architectural Design". Information Display: March-April 1968. pp. 35-40.

* An asterix marks those texts/articles that are directly referenced in the body of the thesis.