

SIMILARITY ASSESSMENT FOR CARDINAL DIRECTIONS BETWEEN EXTENDED SPATIAL OBJECTS

By

Roop K. Goyal

B.E. University of Roorkee, India, 1988

M. E. Nanyang Technological University, Singapore, 1995

A THESIS

Submitted in Partial Fulfillment of the
Requirements for the Degree of
Doctor of Philosophy
(in Spatial Information Science and Engineering)

The Graduate School
The University of Maine

May, 2000

Advisory Committee:

Max J. Egenhofer, Professor of Spatial Information Science and Engineering, Advisor

Peggy Agouris, Assistant Professor of Spatial Information Science and Engineering

M. Kate Beard-Tisdale, Associate Professor of Spatial Information Science and
Engineering

Robert D. Franzosa, Professor of Mathematics and Statistics

John R. Herring, Adjunct Professor of Spatial Information Science and Engineering

Library Rights Statements

In presenting this thesis in partial fulfillment of the requirements for a graduate degree at the University of Maine, I hereby agree that the Library has the right to make it freely available for inspection. I further agree that permission for “fair use” copying of this thesis for scholarly purposes may be granted by the Librarian. It is understood that any copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Roop K. Goyal

April 12, 2000

SIMILARITY ASSESSMENT FOR CARDINAL DIRECTIONS BETWEEN EXTENDED SPATIAL OBJECTS

By Roop K. Goyal

Thesis Advisor: Dr. Max J. Egenhofer

An Abstract of the Thesis Presented
in Partial Fulfillment of the Requirements for the
Degree of Doctor of Philosophy
(in Spatial Information Science and Engineering)

May, 2000

Cardinal directions are frequently used as selection criteria in spatial queries or for assessing similarities of spatial scenes. Current models for cardinal directions use crude approximations in the form of the objects' minimum bounding rectangles or their generalizations to points. To overcome the limitations of these models so that improved reasoning can be performed, the coarse direction-relation matrix is introduced. It partitions space around a reference object and records into which direction tiles an extended target object falls. The detailed direction-relation matrix captures more details by recording the ratio of the target object in each direction tile or the number of separations per tile. This multi-resolution model provides a better approximation for direction relations of complexly structured spatial objects than the approach with minimum bounding rectangles.

In order to record directions between arbitrary pairs of point, line, and region objects, the model based on the coarse direction-relation matrix is extended to the deep direction-relation matrix. It additionally records information about the intersection of the target object with the boundaries of direction tiles, if necessary. This thesis demonstrates that directions recorded at smaller scales using this model are compatible with the directions recorded at larger scales. The compatibility makes this model useful for direction-based queries in spatial databases over multiple scales.

To apply direction-relation matrices for the assessment of similarity between spatial scenes, this thesis develops a method to compute similarity between direction-relation matrices. The similarity between two direction-relation matrices depends on the distances between cardinal directions along a conceptual neighborhood graph, which has a node for each direction tile and edges connecting nodes corresponding to neighboring tiles. There are two types of graphs: the 4-neighborhood graph and the 8-neighborhood graph. The comparative study of the mappings from directions changes to similarity values provided by the graphs reveals that the 4-neighborhood graph provides a sounder mapping than the 8-neighborhood graph. The similarity assessment method gives cognitively plausible rankings of spatial scenes based on the cardinal direction between objects, and it is useful in retrieving spatially similar scenes in image databases, video databases, multimedia databases, and web databases.

Acknowledgments

I would like to express my sincere gratitude towards my thesis advisor Dr. Max Egenhofer for his support, guidance, and encouragement, and the members of my thesis advisory committee; Dr. Peggy Agouris, Dr. Kate Beard, Dr. Robert Franzosa, and Dr. John Herring, for their guidance and cooperation during this thesis.

I thank Dr. Walter Senus, National Imagery and Mapping Agency, and Professor George Markowsky, Department of Computer Science, for their hints that helped in solving the similarity assessment problem. I also thank Professor Fred Irons, Department of Electrical and Computer Engineering, and Professor V. K. Balakrishnan, Department of Mathematics and Statistics, for discussing with me the similarity assessment problem.

The Fogler Library proved to be an excellent source of references for this research, which was very helpful.

This research was partially supported by the National Imagery and Mapping Agency under grant number NMA202-97-1-1023 and a Massive Digital Data Systems contract sponsored by the Advanced Research and Development Committee of the Community Management Staff.

Dr. Mohan Kankanhalli, National University of Singapore encouraged me to accept the opportunity to do a Ph. D. in Spatial Information Science and Engineering at the University of Maine. I am thankful to Rashid Shariff and John Florence for their help during our settling at Orono, Maine.

I would like to thank my all friends in the Department of Spatial Information Science and Engineering for creating a conducive environment for work, especially Michela Bertolotto, Andreas Blaser, James Carswell, Kristin Eickhorst, Douglas Flewelling, Kathleen Hornsby, Craig Miller, Andrea Rodríguez, Paul Schroeder, Tony Stefanidis, and Boonsap Witchayangkoon.

To make us feel homely in the place far from our families, I thank Mahesh and Shashi Chaturvedi, Ramesh and Pushpa Gupta, and Dilip and Sukla Lakshman.

Special thanks are due to my parents and elder brothers for providing me excellent upbringing, education, and guidance, which helped me in reaching at this point in my life. My father in law encouraged me to complete the doctoral program.

My heartfelt thanks go to my wife Anju for her patience and the moral support she provided during my doctoral program. My daughters Deeksha and Saumya made my home an enjoyable place to live.

Table of Contents

Acknowledgments	ii
List of Tables	ix
List of Figures	xi
Chapter 1 Introduction.....	1
1.1 Spatial Reasoning.....	2
1.2 Direction Relations	4
1.3 Motivation for Formalizing Direction Relations	5
1.3.1 Directions in Queries over Spatial Databases.....	5
1.3.2 Directions in Spatial Reasoning	6
1.3.3 Directions between Geographic Objects	7
1.3.4 Directions in Multi-Resolution Geographic Databases	8
1.3.5 Directions in Content-Based Retrieval	9
1.4 Cardinal Directions between Extended Spatial Objects	11
1.4.1 Problem Statement.....	11
1.4.2 Goals and Hypothesis	12
1.4.3 Scope of the Study.....	12
1.4.4 Topics Excluded from the Present Investigation	13
1.5 Intended Audience	14
1.6 Major Results	14
1.7 Thesis Organization.....	15
Chapter 2 Models of Direction Relations	17
2.1 Reference Frames.....	18
2.2 A Model of Quantitative Directions.....	18

2.3	Models of Qualitative Directions for Points	19
2.3.1	Cone-Based Direction Models for Points	20
2.3.2	Projection-Based Direction Models for Points	21
2.3.3	Directions between a Line and a Point	22
2.4	Directions between Extended Objects	23
2.4.1	The Triangular Model.....	24
2.4.2	Interval Relations in Two-Dimensions	26
2.4.3	Qualitative Models for Space	27
2.4.4	Two-Dimensional Strings	28
2.4.5	Symbolic Arrays	30
2.4.6	Directions between Minimum Bounding Rectangles	31
2.5	Comparisons of the Direction Models	33
2.6	Summary.....	34
Chapter 3	Direction-Relation Matrix for Region Objects	36
3.1	Coarse Direction-Relation Matrix.....	37
3.2	Realizability of the Coarse Direction-Relation Matrix	40
3.2.1	Consistency Constraints	40
3.2.2	Iconic Representation of Direction-Relation Matrix	41
3.3	Effects of Shape, Size, and Distance.....	43
3.4	Detailed Direction-Relation Matrix	44
3.4.1	Areal Distribution.....	45
3.4.2	Component Distribution.....	46
3.4.3	Areal Distribution of Components	47
3.5	Comparison with MBR Direction Relations	48
3.6	Summary.....	49
Chapter 4	Deep Direction Model for Point, Line, and Region Objects.....	51
4.1	Applying the Coarse Direction-Relation Matrix to Lines and Points	52
4.1.1	Reference Object Considerations	52
4.1.2	Target Object Considerations	53

4.2	Deep Direction-Relation Matrix	55
4.3	Deep Direction-Relation Matrices for Various Types of Reference Objects	57
4.3.1	A Region Reference Object.....	58
4.3.2	A Linear Horizontal Reference Object.....	60
4.3.3	A Linear Vertical Reference Object.....	61
4.3.4	A Point Reference Object.....	62
4.4	Consistency Constraints Due to the Type of the Target Object.....	62
4.4.1	A Region Target Object.....	64
4.4.2	A Line Target Object.....	66
4.4.3	A Point Target Object.....	67
4.5	Directions at Multiple Scales.....	68
4.5.1	Compatibility for Deep Direction-Relation Matrices.....	70
4.5.2	Relations between Projections of Objects	71
4.5.3	Collapsing a Pair of Intervals into a Pair of Points.....	74
4.5.3.1	Intervals with relation <i>meets</i>	75
4.5.3.2	Intervals with relation <i>overlaps</i>	77
4.5.3.3	Intervals with relation <i>equal</i>	78
4.5.4	Collapsing a Pair of Intervals into a Point and an Interval.....	79
4.5.5	Collapsing a Pair of Intervals into an Interval and a Point	80
4.5.6	Collapsing an Interval and a Point into a Pair of Points	81
4.5.7	Collapsing a Point and an Interval into a Pair of Points	82
4.5.8	Compatible Directions at Smaller Scales	82
4.6	Analysis of the Deep Direction Model	85
4.6.1	Cognitively Plausible Values of Directions.....	85
4.6.2	Advantages of the Deep Direction Model Over Existing Models.....	86
4.7	Summary.....	88
Chapter 5 Similarity Between Cardinal Directions		89
5.1	The Distance between Two Cardinal Directions	90

5.1.1	Distance Between Two Single-Element Direction-Relation Matrices	91
5.1.2	Distance Between Two Multi-Element Direction-Relation Matrices	93
5.2	The Minimum Cost Solution for the Transformation Problem.....	100
5.2.1	A Basic Feasible Solution.....	103
5.2.2	Optimizing a Basic Feasible Solution	105
5.3	The Similarity Value	108
5.4	Summary.....	109
Chapter 6	Evaluation of the Similarity Assessment Method.....	110
6.1	Disha’s Architecture	111
6.1.1	Platform-Independent Classes	111
6.1.2	Platform-Dependent Classes for the User Interface	114
6.2	Implementation Issues.....	117
6.3	Disha’s Features	118
6.3.1	Checking Consistency for Direction-Relation Matrices.....	118
6.3.2	Determining the Number of Consistent Configurations	119
6.3.3	Direction-Similarity from a Direction-Difference Matrix.....	120
6.3.4	Direction-Similarity between Two Ordered Pairs of Extended Objects	121
6.4	Systematic Evaluation of the Method	123
6.4.1	Moving the Target Object Over the Reference Object.....	124
6.4.2	Rotating the Target Object Around the Reference Object	125
6.5	Eight-Neighbor Conceptual Graph	127
6.6	Comparison of the Soundness of the Mappings Provided by Four- Neighbor and Eight-Neighbor Graphs	128
6.6.1	Curved Movement of a Disjoint Target Object.....	129
6.6.2	Curved Movement of an Overlapping Target Object.....	131
6.6.3	Diagonal Movement of a Larger Target Object	132

6.6.4	Diagonal Movement of a Smaller Target Object.....	134
6.6.5	Vertical Movement of a Larger Target Object	135
6.6.6	Vertical Movement of a Smaller Target Object	136
6.6.7	Scaling up the Target Object in the Northwest Tile	138
6.6.8	Scaling up the Target Object in the North Tile	139
6.6.9	Rotation of the Target Object.....	140
6.6.10	Discussion.....	142
6.7	Summary.....	142
Chapter 7	Conclusions	144
7.1	Summary of the Thesis	144
7.2	Results and Major Findings	146
7.3	Future Work.....	149
7.3.1	Direction Reasoning using Direction-Relation Matrix.....	149
7.3.2	Deep Detailed Direction-Relation Matrix	150
7.3.3	Cognitive Evaluations.....	151
7.3.3.1	Cognitive Plausibility of the Direction-Relation Matrix.....	151
7.3.3.2	Distances between Single-Element Direction-Relation Matrices.....	151
7.3.3.3	Similarity Assessment.....	151
7.3.4	Mapping of Natural Language Direction Terms onto Direction- Relation Matrices.....	152
7.3.5	Detecting and Quantifying Change	152
7.3.6	Inferring the Type of Change from the Profile of Similarity Values	152
7.3.7	Extension to 3-Dimensional Space.....	153
7.3.8	Similarity between Raster Templates Using the Conceptual Neighborhood Graph	153
Bibliography	155
Biography of the Author	167

List of Tables

Table 2.1:	Evaluation of the directions models for extended objects in 2-D.	34
Table 4.1:	All possible values of the northwest element in the deep direction- relation matrix for the region reference. The remaining five bits (1-2 and 6-8) are always zero.	59
Table 4.2:	Significant scale reductions in the types of projections of reference target pair on the x -axis.	70
Table 4.3:	The conditions for the relations between two points on the x -axis; A is the reference point and B is the target point.	72
Table 4.4:	Conditions for relations of point B with respect to an interval A	72
Table 4.5:	Conditions for relations of interval B with respect to a point A	73
Table 4.6:	Conditions for 1-D interval relations.	74
Table 4.7:	An IPPP scale reduction maps relations between intervals onto the relations between points.	76
Table 4.8:	The scale reduction of the type IPII maps the relations between intervals onto the relations of point B with respect to interval A	80
Table 4.9:	The scale reduction of the type IIIP maps the relations between intervals into the relations of point B with respect to interval A	81
Table 4.10:	The scale reduction of the type IPPP maps the relations of point B with respect to interval A onto the relations between points.	81

Table 4.11: The scale reduction of the type PIPP maps the relations of interval B with respect to point A onto the relations between points.....	82
Table 4.12: The projections of the reference object A and the components of the target object B onto the x -axis, and their relations, where n is the component number.	84
Table 6.1: Similarities and ranks of Scenes 0-7 in Figure 6.24.	130
Table 6.2: Similarities and ranks of Scenes 0-7 in Figure 6.26.	131
Table 6.3: Similarities and ranks of Scenes 0-4 in Figure 6.28.	133
Table 6.4: Similarities and ranks of Scenes 0-4 in Figure 6.30.	134
Table 6.5: Similarities and ranks of Scenes 0-4 in Figure 6.32	136
Table 6.6: Similarities and ranks of Scenes 0-4 in Figure 6.34.	137
Table 6.7: Similarities and ranks of Scenes 0-4 in Figure 6.36.	138
Table 6.8: Similarities and ranks of Scenes 0-4 in Figure 6.38.	140
Table 6.9: Similarities and ranks of Scenes 0-7 in Figure 6.40.	141
Table 7.1: The types of projections of objects A , B , and C along an axis and the composition types. The codes P and I stand for point and interval types of projections, respectively.	150

List of Figures

Figure 1.1:	Three towns in the state of Maine.	4
Figure 1.2:	Various interpretations of the direction term north: (a) cone-shaped partition, (b) centroid-based direction line, and (c) rectangle-shaped partition.	6
Figure 1.3:	A feasible visualization of a description in the locality and elevation interpreter (Futch <i>et al.</i> 1992).	7
Figure 1.4:	Irregularly shaped geographic objects.	7
Figure 1.5:	(a)-(b) Object <i>B</i> is <i>north</i> of object <i>A</i> and (c) object <i>B</i> is <i>north</i> and <i>northeast</i> with respect to <i>A</i>	8
Figure 1.6:	Objects in multi-resolution geographic databases can be points, lines, and regions.	9
Figure 1.7:	Three images in a database.	10
Figure 2.1:	The azimuth angle.	19
Figure 2.2:	The cone-based model: (a) four-direction system and (b) eight-direction system.	21
Figure 2.3:	The projection-based direction model for point objects. The space is divided into (a) north-south half planes, (b) west-east half planes, and (c) nine directions.	22

Figure 2.4:	(a) Point C can be in 15 distinct qualitative locations with respect to the line AB , which forms the reference frame and (b) an iconic representations of these 15 relations.	23
Figure 2.5:	The centroid of B (c_B) is east of the centroid of A (c_A); therefore, $dir(A, B)$ is east.	24
Figure 2.6:	The triangular model for extended objects (a) B is visually east of A , but it does not fall in the east triangle, and (b) by adjusting the area of acceptance of east triangle, B is east of A	25
Figure 2.7:	The centroid of B is north of the centroid of A , but B is not north of A	25
Figure 2.8:	Allen's thirteen temporal interval relations in 1-dimension.	26
Figure 2.9:	Four basic relations in 2-D (Guesgen 1989).	27
Figure 2.10:	(a) Relations between an interval and a point and (b) relations between two extended objects at an arbitrary angle.	28
Figure 2.11:	(a) A scene and (b) its corresponding symbolic picture.	29
Figure 2.12:	A scene with extended objects and cutting lines.	30
Figure 2.13:	(a) A scene and (b) the corresponding symbolic array.	31
Figure 2.14:	Spatial relation between two minimum bounding rectangles.	32
Figure 2.15:	The spatial relations between minimum bounding rectangles (Papadias <i>et al.</i> 1995).	32
Figure 3.1:	Two objects with their minimum bounding rectangles.	37
Figure 3.2:	The nine tiles resulting from the partitioning of space around the reference object A	38

Figure 3.3:	Capturing the cardinal direction relation between two areal objects, A and B , through the projection-based partitions (a) around A as the reference object and (b) around B as the reference object.....	39
Figure 3.4:	The iconic representation of the direction-relation matrix: (a) a configuration with three non-empty cells and (b) a configuration with one non-empty cell and two cells that are either empty or non-empty.	41
Figure 3.5:	The interpretation of gray cells under the 4-connectivity constraint: (a) a configuration with gray cells and (b)-(e) the set of possible black-and-white configurations, with (d) being an illegal configuration because it is not 4-connected.	42
Figure 3.6:	An iconic representation of the 218 direction-relation matrices that can be realized between two regions.	42
Figure 3.7:	(a) The original configuration, (b) the size of the target object is increased, (c) the distance between the reference and target object is decreased, and (d) the target object is elongated.	44
Figure 3.8:	Three scenes for which the coarse direction-relation matrix serves as a cognitively inappropriate equivalence relation.	44
Figure 3.9:	Tile 0_A with (a) two strongly disconnected components of B and (b) two weakly disconnected components.	46
Figure 3.10:	A direction relation with four components in A 's Northern tile.....	47
Figure 3.11:	A configuration whose cardinal direction is better captured by the direction-relation matrix than by MBRs.	48
Figure 3.12:	Two configurations with the same direction-relation matrix, but different MBR relations.....	49

Figure 4.1:	Configurations with line and point objects: (a) the reference object is a region and the target object is a line, (b) a region and a point object, and (c) a line and a point object.	52
Figure 4.2:	Different references frames based on different types of objects: (a) for a point, (b) for a vertical line, (c) for a horizontal line, and (d) for other lines and all regions.	53
Figure 4.3:	A region reference A in a 2-dimensional embedding space: (a) nine direction partitions, (b) twelve boundary lines between direction partitions, and (c) four boundary points.	54
Figure 4.4:	The nine-bit field for a neighbor code in the deep direction-relation matrix.	56
Figure 4.5:	(a) The reference grid for a region object and (b) the patterns of neighbor codes for the northwest, north and same elements. A zero value for a bit in an element implies that the value of that bit is always zero.	59
Figure 4.6:	(a)-(i) Nine configurations corresponding to nine possible values of the northwest element in the deep direction-relation matrix for a region reference object.	60
Figure 4.7:	(a) The reference grid for a horizontal line and (b) the patterns of neighbor codes for the northwest, north, west, and same elements.	61
Figure 4.8:	(a) The reference grid for a vertical line and (b) the patterns of neighbor codes for the northwest, west, north, and same elements.	61
Figure 4.9:	(a) The reference grid for a point and (b) the patterns of neighbor codes for the northwest, north, and same element.	62

Figure 4.10: The direction of a point in specific situations with respect to (a) a vertical line having a row-sextuple and (b) a horizontal line having a column- sextuple.	64
Figure 4.11: Deep direction-relation matrices that are consistent for region targets with respect to region references.	65
Figure 4.12: Deep direction-relation matrices that are consistent for line targets with respect to (a) a region, (b) a horizontal line, and (c) a point.	67
Figure 4.13: Deep direction-relation matrices that are consistent for point targets with respect to (a) a point, (b) a horizontal line, (c) a vertical line, and (d) a region.	68
Figure 4.14: Objects A and B with their deep direction-relation matrices at (a) scale S^0 and (b)-(d) scale S^1	71
Figure 4.15: Relations in 1-D space for (a) point B with respect to interval A and (b) interval B with respect to point A	72
Figure 4.16: One-dimension interval relations, A is the reference interval and B is the target interval.	73
Figure 4.17: The relations between projections on to the x -axis: (a)-(b) interval relation is <i>meets</i> and point relations are (c)-(d) <i>before</i> , and (e) <i>equal</i>	77
Figure 4.18: (a) The region-region pair at scale S_0 and (b) region-line pair at scale S_1 after a zooming-out operation.	84
Figure 4.19: Pairs of region objects. (a) Object B is north of object A and (b) object B is north of A and intersects with the north-northeast line (N-NeL).	86

Figure 4.20: Region reference and line target. (a) object B is north of A , (b) B is north of A and intersects with the N-NeL, and (c) object B completely lies on the N-NeL.	86
Figure 4.21: Region reference and point target. (a) object B is north of A and (b) object B completely lies on the N-NeL.	86
Figure 4.22: Sixteen different types of object pairs.....	87
Figure 5.1: (a) The query scene and (b)-(d) Scenes 0-2 in a database.....	90
Figure 5.2: The shortest path to move the target object B from the northwest tile to the northeast tile is through the north tile, while considering only single-element directions.....	92
Figure 5.3: The conceptual neighborhood graph for nine cardinal directions based on the 4-neighborhood between tiles.	92
Figure 5.4: Four-neighbor distances between cardinal directions for regions.	93
Figure 5.5: The target object moves across single as well as multi-element cardinal directions from (a) northwest through (b) northwest and north, (c) north, (d) north and northeast, to (e) northeast.	94
Figure 5.6: A direction comparison example Scene 0 and Scene 1 with identical objects, but different directions D^0 and D^1 , respectively.....	96
Figure 5.7: The balanced transportation tableau.	101
Figure 5.8: The transportation tableau for the direction difference matrix of the example.....	103
Figure 5.9: A basic feasible solution of the transportation problem in the example using the northwest corner method.	104

Figure 5.10: An iteration of the transportation algorithm for the example.....	106
Figure 5.11: The second and last iteration of the transportation algorithm for the example.....	107
Figure 6.1: Architecture of the prototype Disha.	111
Figure 6.2: The object classes for (a) the direction-relation matrix and (b) the direction- difference matrix.	112
Figure 6.3: The aggregation hierarchy for the classes <i>CDrmGraph</i> and <i>CVertex</i>	113
Figure 6.4: The class <i>CBTableau</i> for the balanced transportation tableau.	113
Figure 6.5: Inheritance hierarchy for the classes corresponding to geometric elements.	115
Figure 6.6: The class <i>CObDrm</i> inherits from the classes <i>CDrm</i> and <i>CObject</i>	116
Figure 6.7: Inheritance hierarchy for the classes (a) <i>CObDeltaMatrix</i> and (b) <i>CObTableau</i>	116
Figure 6.8: Making a detailed direction-relation matrix sum-consistent.	118
Figure 6.9: The coarse matrix input dialog for checking the consistency of a coarse direction-relation matrix.	119
Figure 6.10: Counting consistent configurations.	120
Figure 6.11: Computing the distance from a direction-difference matrix.	120
Figure 6.12: A snapshot of Disha with two pairs of objects and their direction- relation matrices.	121
Figure 6.13: (a) The direction difference matrix and (b) the transportation tableau using the 4-neighbor graph.	122

Figure 6.14: The linear programming problem for the transportation problem in Figure 6.13b.	122
Figure 6.15: A solution for the linear programming problem in Figure 6.14 using LINDO (1999).....	122
Figure 6.16: A solution of the transportation problem in Figure 6.13b.	123
Figure: 6.17: Computing the similarity value from the transportation problem solution in Figure 6.16.....	123
Figure 6.18: Scenes generated by moving the target object over the reference object.	124
Figure 6.19: The pattern of similarity values for Scenes 0-7 in Figure 6.18.	125
Figure 6.20: Scenes generated by moving the target object around the reference object.	126
Figure 6.21: The pattern of similarity values for Scenes 0-16 in Figure 6.20.	127
Figure 6.22: The conceptual neighborhood graph for nine cardinal directions based on the 8-neighborhood between tiles.	127
Figure 6.23: Eight-neighbor distances between cardinal directions for regions.....	128
Figure 6.24: The scenes generated by curved movement of the disjoint target object.	129
Figure 6.25: The pattern of similarity values for Scenes 0-7 in Figure 6.24.	130
Figure 6.26: The scenes generated by curved movement of the overlapping target object.	131

Figure 6.27: The pattern of similarity values for Scenes 0-7 in Figure 6.26.	132
Figure 6.28: The scenes generated by diagonal movement of the larger target object.	133
Figure 6.29: The pattern of similarity values for Scenes 0-4 in Figure 6.28.	133
Figure 6.30: The scenes generated by diagonal movement of the smaller target object.	134
Figure 6.31: The pattern of similarity values for Scenes 0-4 in Figure 6.30.	135
Figure 6.32: The scenes generated by vertical movement of the larger target object.	135
Figure 6.33: The pattern of similarity values for Scenes 0-4 in Figure 6.32.	136
Figure 6.34: The scenes generated by vertical movement of the smaller target object.	137
Figure 6.35: The pattern of similarity values for Scenes 0-4 in Figure 6.34.	137
Figure 6.36: The scenes generated by enlarging the target object in the northwest tile.	138
Figure 6.37: The pattern of similarity values for Scenes 0-4 in Figure 6.36.	139
Figure 6.38: The scenes generated by enlarging the target object in the north tile.	139
Figure 6.39: The pattern of similarity values for Scenes 0-4 in Figure 6.38.	140
Figure 6.40: The scenes generated by rotating the target object.	141
Figure 6.41: The pattern of similarity values for Scenes 0-7 in Figure 6.40.	142

Chapter 1

Introduction

People communicate about geographic space often using highly generalized spatial concepts such as cardinal directions. For example, they typically refer to relationships between objects using such terms as north or northeast. These qualitative spatial concepts are sufficient for people to make inferences about the relationships and their combinations (Byrne and Johnson-Laird 1989), and qualitative spatial reasoning has been found to be fundamental to people's everyday activities (Riesbeck 1980). For instance, when people ponder about where to place a newly acquired table at home or when they decide where to construct a new building, they apply various forms of qualitative spatial reasoning. While people are good at making decisions based on spatial information for daily tasks, it is necessary to use information systems for complex and systematic spatial reasoning tasks involving a large number of constraints.

Geographic information systems (GISs) are built to aid people in making decisions for such complex tasks as land management, forest management, urban development, and hazardous waste management. A GIS's success depends on its ability to answer people's questions without making people learn about the internal data representation in the GIS. The GIS typically stores the geometry and other information about objects, and allows a user to perform such operations as retrieving stored information, analyzing the stored data, and reasoning to derive new spatial information. For instance, a state government's GIS may store the geometry and population of counties, so that a user can query and analyze the population of counties based on their spatial relationships. In order to query and analyze spatial data in a meaningful way, GISs need methods to formulate and process spatial relations.

Typically GISs use analytical techniques for extracting spatial information from quantitative representations, such as Euclidean geometry and Cartesian coordinate systems. The formalization of spatial relations and the explicit storage of spatial relations in a GIS is preferred over analytical methods, because spatial relations form the basis of most spatial queries and explicit storage makes this information available right away.

In order to make GISs respond to users' queries effectively, spatial concepts should be formalized in cognitively plausible ways and be incorporated into spatial query languages. The formalization of spatial concepts has been a research priority for more than a decade, as evident from the research agenda of the National Center for Geographic Information and Analysis (NCGIA) that identified the need for coherent mathematical theories of spatial relationships (Abler 1987). NCGIA's Varenus project identifies the computational implementation of geographic concepts as a strategic area of GIS research (NCGIA 1995). The extension of geographic representations is also a research priority according to the University Consortium of Geographic Information Science (UCGIS 1996). This thesis continues research on formalizing spatial concepts offering a new model for cardinal directions that is applicable to extended as well as point objects.

1.1 Spatial Reasoning

Reasoning is a fundamental logical operation that is often used to explain people's inferences. It can be performed on quantitative as well as qualitative information. Quantitative information is recorded using a predefined unit of a quantity such as meter for distances. For example, the distance between Boardman Hall and Fogler Library is 140 meters. Reasoning on quantitative information is performed by mathematical rules that are expressed as numeric operations. Qualitative reasoning, on the other hand, focuses on the essence of information using a small set of such symbols as {+, 0, -}. For example, if a faucet is discharging water in a bathtub and the rate of water entering the tub is more than the water leaving the tub, the tub will eventually overflow. To arrive at this conclusion, no elaborate equations were used. Qualitative reasoning allows us to make such inferences. Inferring new information using the available qualitative information about the space is called *spatial reasoning* (Hernández 1993; Cohn 1996). In

combination with reasoning about time, it is called spatio-temporal reasoning (Egenhofer and Golledge 1998).

Naive Geography (Egenhofer and Mark 1995b) is a modern concept in the area of spatial reasoning along the line of Naive Physics (Hayes 1978; Hayes 1985). Naive Geography attempts to make GISs behave close to the way people reason about space and time. Such systems are more likely to be accepted by people due to the expected ease of use. Naive Geography is a challenging task for spatio-temporal researchers and GIS designers, as the task requires models of spatial concepts that conform to people's way of reasoning. Formalization of spatial concepts has been a topic of active interest across such disciplines as GIS, geography, artificial intelligence, computer vision, cognitive science, and psychology. Vision systems perform spatial reasoning on objects extracted from digital imagery (Freeman 1975; Haar 1976).

Olivier and Gapp (1998) provide a collection of representation and processing techniques for spatial expression. Spatial relations capture knowledge about relative placement of spatial objects. In GISs, one typically distinguishes three types of spatial relations: topology, direction, and distance. Topological relations capture important knowledge about geometric relations between objects that are invariant under affine transformations such as translation, rotation, and scaling. For example, if two objects are disjoint, they continue to remain disjoint if the embedding space is rotated. The 4-intersection (Egenhofer and Franzosa 1991) distinguishes eight different topological relations between regions, which are called *disjoint*, *overlaps*, *meets*, *equals*, *inside*, *contains*, *covers*, and *covered-by*. Cardinal direction captures knowledge about relative location of an object with respect to another object in the embedding space such as *A* is north of *B*. Cardinal directions between objects do not depend on the objects' internal structures, the observer, and the scale of embedding space, but they depend on their relative positions with respect to each other and their shapes, and their extents. For example, if *A* is north of *B*, *A* continues to remain north of *B* if the embedding space is scaled. The distance between two objects depends on the efforts to move from the location of one object to the location of another. Distance is typically recorded using

Euclidean distance, which is also called quantitative distance. Hong (1994) developed a method to transform the quantitative distances into qualitative distances such as *near*, *medium*, *far*, and *very far*. This thesis focuses on models for cardinal directions.

1.2 Direction Relations

People perform qualitative spatial reasoning about directions in 2-dimensional space often using such concepts as *right*, *left*, *front*, and *back* and inference rules (Byrne and Johnson-Laird 1989). In the case of geographic objects, people use cardinal directions such as north and northeast. For example, Orono, Maine is northeast of Bangor, Maine (Figure 1.1). In this relation Bangor is the reference object, Orono is the target object, and northeast is the cardinal direction between this ordered pair of objects. People typically use eight cardinal directions: north, northeast, east, southeast, south, southwest, west, and northwest (Frank 1996). In direction reasoning, an additional identity direction *0* or *same* is also used, that gives nine cardinal directions.

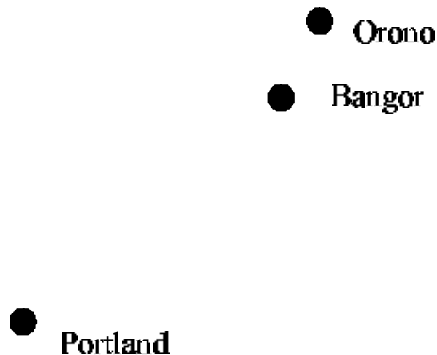


Figure 1.1: Three towns in the state of Maine.

Direction reasoning allows us to infer unknown directions from known directions. In databases, such inferences are very useful as they can be performed symbolically, which provides consistent information with minimal computing. Direction combined with distance is used for reasoning about locations (Hong 1994), and with topology for the reasoning about layouts of objects (Sharma 1996).

1.3 Motivation for Formalizing Direction Relations

While people commonly use cardinal directions in communication and for many inferences, the concepts are still too vague to be implemented consistently in GISs. There are several needs for formalized cardinal directions in GISs such as in query languages, in query processors, and in intelligent inference engines for spatial reasoning. The following motivations elaborate the properties necessary for a formalization of cardinal directions so that such a formalization has the widest applicability in GISs.

1.3.1 Directions in Queries over Spatial Databases

The term spatial database refers to a database that employs data structures that allow queries based on the spatial extent of objects (Günther and Buchmann 1990). Güting (1994) presents a survey of research in the area of data modeling, query languages, spatial data types, and spatial indexing. Quad-trees (Samet 1989a) and R-trees (Guttman 1984) are among the most popular data structures used in spatial databases. To retrieve the objects in a given direction, Papadias *et al.* (1994) used R-trees by defining a search rectangle with respect to a reference object based on the direction of interest, and returned objects that intersect with the search rectangle. This scheme involves an exhaustive search based on the geometry of objects. Formal models of directions allow direction relations to be explicitly stored in databases that facilitate the use of direction as a search criterion for such queries as, “Select all lakes that are northeast of Orono in Maine.”

A study conducted by Franklin *et al.* (1995) reveals that people parse their surrounding horizontal space into overlapping front, left, back, and right direction partitions; therefore, an observer may record a point that lies in an overlap of two directions as a point belonging to the neighboring directions, such as front and right. Similarly, cardinal direction terms such as north and east may also have overlapping regions in people’s minds, and these terms may have different meanings for different people. For example, some people may treat the north partition as a triangular area (Figure 1.2a), while others may treat it as a line passing through the centroid of the

reference object and going towards north (Figure 1.2b), and some others may treat it as a rectangular area that is north of the reference object (Figure 1.2c). People communicate with each other with direction terms that might have different meanings for different people, but query languages in GISs require formal definitions of direction terms. Therefore, the model of cardinal directions must be *formal*.

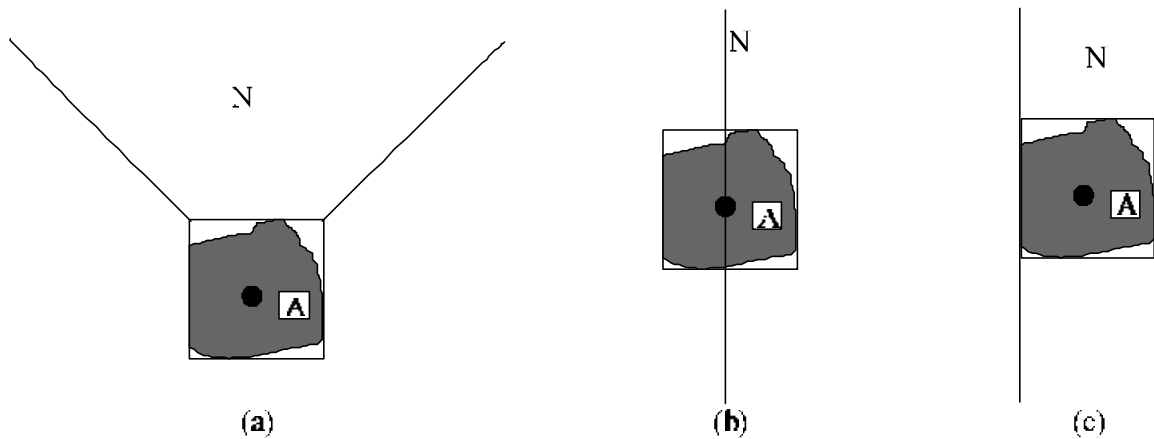


Figure 1.2: Various interpretations of the direction term north: (a) cone-shaped partition, (b) centroid-based direction line, and (c) rectangle-shaped partition.

1.3.2 Directions in Spatial Reasoning

Locations of places are often specified with respect to known places or landmarks using natural-language descriptions. The locality and elevation interpreter performed spatial reasoning on linguistic representations of geographic information (McGranaghan and Wester 1988; McGranaghan 1989; Futch *et al.* 1992). It was designed to convert the linguistic text described on herbarium specimens into geodetic coordinates in order to perform spatial reasoning about the locations of species of plants. An example of such a natural-language description is “Oahu. Palolo Valley. Along the stream and up the northeast bank at elevation of 1100-1500 feet.” Figure 1.3 shows a possible visualization of this description. The approximate location of a specimen in terms of geodetic coordinate is inferred using the geodetic coordinates of a known place such as the longitude and latitude of Oahu in the mentioned description. A model of cardinal

direction must support spatial reasoning operations to infer unknown directions from the known directions, that is, it must be *inferential*.

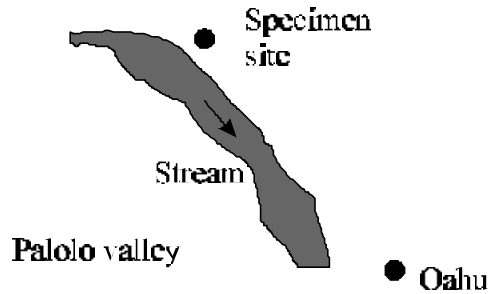


Figure 1.3: A feasible visualization of a description in the locality and elevation interpreter (Futch *et al.* 1992).

1.3.3 Directions between Geographic Objects

Shapes of objects can affect directions between them (Peuquet and Zhan 1987; Abdelmoty 1995). Geographic objects can occur in numerous shapes and can relate to each other in many possible ways. For instance, the convex hulls of objects can intersect (Figure 1.4a), an object can surround the other object (Figure 1.4b), and two objects can be overlapping and intertwined with each other (Figure 1.4c).

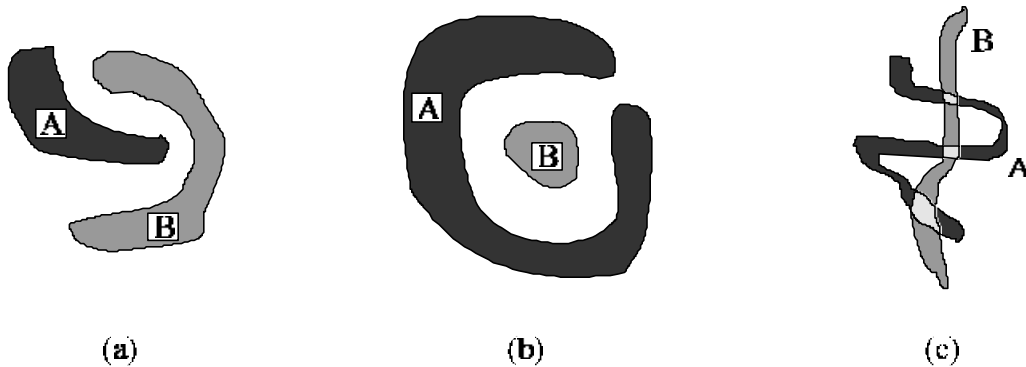


Figure 1.4: Irregularly shaped geographic objects.

Shapes of objects are of interest to people in various domains. For example, in computer graphics shapes such as triangles, rectangles, polygons, and circles are used frequently. In computer vision and image processing, shape parameters such as area,

compactness, elongation, directions of major and minor axes, and moments of objects are computed from the objects extracted from images to compare the shapes of objects.

Some changes in shapes, in specific situations, affect the direction of an object with respect to another object. Figure 1.5 gives an example where a change in the shape of target object *B* does not change its direction with respect to reference object *A* (Figures 1.5a-b), while another change in shape changes its direction (Figures 1.5a and 1.5c).

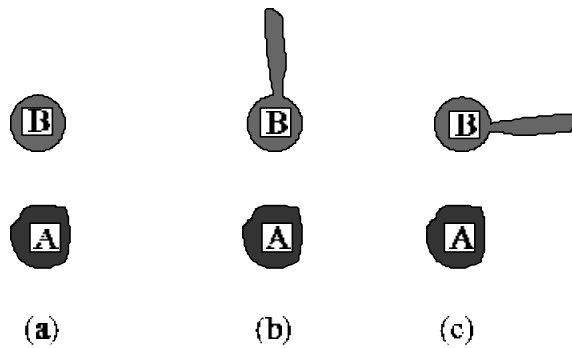


Figure 1.5: (a)-(b) Object *B* is *north* of object *A* and (c) object *B* is *north* and *northeast* with respect to *A*.

A direction model for geographic objects must be *sensitive* to those changes in shapes that affect cardinal directions between them. This thesis calls this property of direction models *shape-sensitiveness*. In order to record the direction that is sensitive to the shapes of the objects, a direction model must not approximate the shapes of the objects, but use the shapes of the objects as they are. Existing models of directions typically approximate geometries of objects by points or rectangles; therefore, they miss the effect of the shapes of objects on the directions between them and may yield misleading directions. A direction model must be able to represent directions between objects of all shapes without approximating their geometries; therefore, a direction model must be *shape-sensitive*.

1.3.4 Directions in Multi-Resolution Geographic Databases

Geographic databases record the geometry of spatial objects as points, lines, or regions depending on the level of detail considered (Goodchild and Proctor 1997). For example, a

city can be represented by a point or a region, and a road can be represented by a line or a region. A database may contain point, line, and region objects, which are the objects of different dimensions. Therefore, methods to determine directions between objects of different dimension are required. Figure 1.6 shows examples of directions between different dimensions. Directions recorded between different dimension representations of an ordered pair of objects in GISs must be cognitively equivalent. If a GIS records towns as points and another records them as regions, a query “Find towns that are north of Bangor in the state of Maine” should return the same result in both GISs.

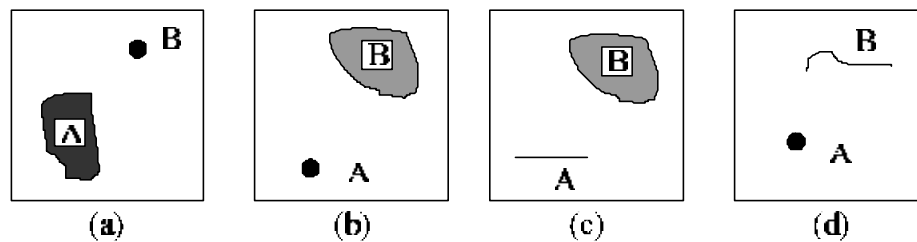


Figure 1.6: Objects in multi-resolution geographic databases can be points, lines, and regions.

Multi-representation geographic databases (Buttenfield 1989; Tryfona and Egenhofer 1996; Bertolotto 1998) record more than one approximation of geographic objects. In current GISs, a user needs to specify the dimension of an object in a query, as the object-oriented concepts of polymorphism and operator overloading that enable the use of same command for a semantically similar operation across different class-objects and arguments are new for GIS community (Newell 1992). Since objects are approximated at different dimensions, a user will have to make many queries to search for the objects of all dimensions. In order to let a user use one query for a search based on direction, direction between objects must not depend on the dimension of objects, that is, a model of directions must be *dimension-neutral*.

1.3.5 Directions in Content-Based Retrieval

Query by image content (Flickner *et al.* 1995) allows a user to retrieve images from a database based on the contents of images. Graphic representations of images store geometric and visual attributes of objects and spatial relations between them. The

geometric attribute of an object refers to its spatial extent, and visual attributes refer to color, shape, and texture (Gonzalez and Woods 1992). Geometric and visual attributes help in determining the presence of an object in a scene and spatial relations between objects distinguish relative placements of the objects in the embedding space. Combining object similarity and spatial relation similarity, one can make a query such as “Find scenes where object *A* and *B* are present, *B* is north of *A*, and *A* disjoint *B*.” Nabil *et al.* (1995) and Bruns and Egenhofer (1996) use spatial relation as a criterion to assess scene similarity. Gudivada and Raghavan (1995) assessed spatial similarity between scenes using quantitative directions between representative points of the objects. A query “Find scenes where *A* disjoint *B*” in a database containing scenes in Figure 1.7 would result in all three scenes in Figure 1.7. On the other hand, a query “Select all scenes where object *A* disjoint *B* and *B* north of *A*,” results in only the image in Figure 1.7c.

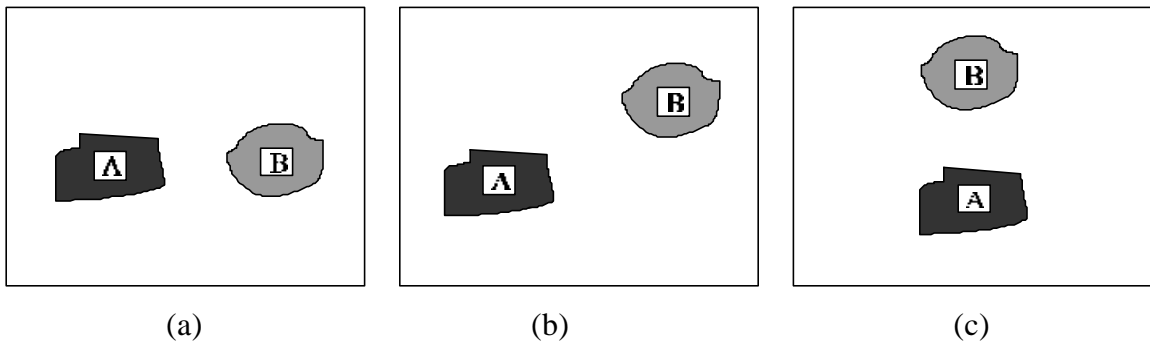


Figure 1.7: Three images in a database.

A user can also make a query to a geographic database such as, “Find scenes that are similar to the scene in Figure 1.7a,” and the system will rank similar scenes by their degrees of match based on similarities between objects and similarities between spatial relations. Results of this query are scenes in Figure 1.6b and 1.6c, and the degree of match for the image in Figure 1.7b will be higher than the degree of match for the image in Figure 1.7c. In order to use direction as a criterion to assess spatial similarity between scenes, the recorded directions must be *comparable*.

1.4 Cardinal Directions between Extended Spatial Objects

The concept of direction is well defined for point objects (Peuquet and Zhan 1987; Frank 1996), but people use this concept for extended objects as well. For example, Canada is north of the USA. The direction between extended objects is determined using crude approximations of the object geometries such as the approximation of objects by their centroids (Section 2.4.1) or by their minimum bounding rectangles (Section 2.4.6). Both models suffer from incorrect inferences, because the recorded directions are not the directions between the objects, but they are the directions between the approximations of the objects.

1.4.1 Problem Statement

In order to overcome limitations in direction representations due to approximations and to use them for applications mentioned in Section 1.3, a model to represent directions must have the following properties:

- The model must be *formal* so that it can be used as a basis for query processing involving directions in spatial databases (Section 1.3.1).
- The model must be *inferential* so that it can be used for deriving unknown directions from the known directions (Section 1.3.2).
- The model must be capable of representing direction between irregularly-shaped objects without approximating their geometries, that is, it must be *shape-sensitive* (Section 1.3.3).
- The model must be *dimension-neutral* so that it can be used in multi-resolution geographic databases (Section 1.3.4).
- The model must support similarity between directions, that is, the directions must be *comparable* (Section 1.3.5).

These five points form the foundation for the present investigations in the modeling and reasoning about cardinal directions.

1.4.2 Goals and Hypothesis

The goals of this thesis are (1) to develop a model for representing cardinal directions that has the properties outlined in the problem statement (Section 1.4.1) and (2) to develop a method for similarity assessment using this model of cardinal direction.

This thesis develops the desired model for similarity assessment using direction-relation matrices. The similarity between two directions depends on the distance between two direction-relation matrices along a conceptual neighborhood graph. The distance between two identical matrices is 0. If the distance between two directions D^0 and D^1 along the conceptual neighborhood graph is smaller than the distance between D^0 and D^2 , D^0 is more similar to D^1 than D^0 to D^2 . A larger distance gives smaller value of similarity and vice-versa.

A method of similarity assessment must provide a *sound mapping* of changes in directions onto the similarity values, such that it gives monotonically decreasing similarity values for increasingly larger changes in directions. The organization of cardinal directions in the direction-relation matrix forms the four-neighborhood and eight-neighborhood graphs.

The hypothesis of this thesis is:

“The four-neighborhood and eight-neighborhood graphs provide equally sound mappings of direction changes onto similarity values.”

Section 6.6 tests the hypothesis by comparing the rankings provided by both types of neighborhood graphs.

1.4.3 Scope of the Study

In order to improve the inference power of next-generation GISs and to facilitate the use of direction relations in spatial databases and content-based retrieval, this thesis uses

direction-relation matrices for direction representation. Direction-relation matrices record directions between objects, not their approximations.

This thesis builds the foundation for a new direction model using direction-relation matrices between region objects that provides a multi-representation view of cardinal directions and allows users to record increasingly more details if desired. It extends this model to a deep direction model that applies to arbitrary pairs of point, line, and region objects. Using direction-relation matrices, it develops a method to assess similarity between directions.

1.4.4 Topics Excluded from the Present Investigation

The theme of this thesis is cardinal directions, not direction giving. In direction giving and direction following (Riesbeck 1980), a subject checks the consistency of instructions and makes a mental map of the situation being described. The following aspects of direction relations are excluded from this investigation:

- This thesis does not formalize the orientation of objects. A quantitative orientation between two objects would be recorded as an angle between their major axes, and qualitative orientation as directions front, left, right, and back. It formalizes the qualitative cardinal directions between extended objects such as north, south, east, and west.
- The direction model presented in this thesis records direction between objects based on their geometries, and influences of their semantics such as figure and ground are not considered (Talmy 1983; Bittner 1997; Bittner 1999).
- The model in this thesis uses non-overlapping direction partitions, that is, the direction partitions are exclusive. This thesis does not study the effect of overlapping direction partitions (Franklin *et al.* 1995) on computational models of cardinal directions.
- The model in this thesis is tailored to 2-D embedding space, while it has potential of being extended to 3-D space (Fuhr *et al.* 1998). This aspect has been excluded from the current studies.

- This thesis does not develop methods to perform spatial reasoning using the presented model of direction.
- This thesis does not develop methods to record detailed direction-relation matrices for arbitrary pairs of different dimension objects and similarity assessment for directions between such arbitrary pairs.
- This thesis does not characterize the shape of objects. It uses the term *shapes-sensitive* for a property of direction models that allows recording of cardinal directions without approximating the objects by their minimum bounding rectangles or points thus preserves the effects of shape on cardinal directions.

1.5 Intended Audience

The intended audience of this thesis constitutes any person interested in spatial reasoning in general and design and development of geographic information systems in particular. This includes researchers from fields of spatial databases, digital libraries, artificial intelligence, content-based retrieval, and computer vision. The direction model and methods developed in this thesis are useful for designing next-generation spatial databases, spatial query languages, and spatial reasoning systems.

1.6 Major Results

The major finding of this thesis is a direction model based on direction-relation matrices that record directions between irregularly-shaped objects without approximating their geometries. The model has all five required properties: *formal*, *inferential*, *shape-sensitive*, *dimension-neutral*, and recorded directions are *comparable* (Section 1.4.1).

Major results of this thesis are:

- The coarse direction-relation matrix provides a knowledge structure to record multiple directions, such as {north, northeast, east} between regions. The model does not approximate the objects' geometries and it is sensitive to the shape of the objects.
- The detailed direction-relation matrix records the extent of the target object and further details about the target object in the framework that is similar to

the coarse direction-relation matrix. It can record directions at multiple resolutions and enhances the distinguishing capability of direction-relation matrices.

- The deep direction-relation matrix is capable of recording directions between arbitrary pairs of point, line, and region objects. The deep direction-relation matrix has additional expressive power to distinguish those directions that cannot be distinguished by the coarse direction-relation matrix, but the model based on minimum bounding rectangles can distinguish. The deep direction-relation matrices record identical values for cognitively equivalent directions.
- We demonstrate that directions recorded, using the deep direction-relation matrices, at smaller scales are compatible with directions recorded at larger scales, which makes this model useful for multi-resolution geographic databases.
- The method for similarity assessment between detailed direction-relation matrices makes the direction model useful for query by content in image databases, video databases, multimedia databases, and web databases.
- A major contribution of this thesis is the rejection of the hypothesis, which uncovers the following fact: the 4-neighborhood graph provides a sounder mapping than the 8-neighborhood graph.

1.7 Thesis Organization

The remainder of this thesis is organized into the following six chapters:

Chapter 2 reviews existing models of direction relations, including direction between points and minimum bounding rectangles. It assesses existing models of directions to check whether or not these models have the required properties for direction relations (Section 1.4.1).

Chapter 3 introduces the concepts of coarse and detailed direction-relation matrices. This chapter investigates how many valid directions can be distinguished using coarse direction-relation matrices. It discusses effects of shape, size, and distance on direction

relations. It also compares distinguishing capabilities of coarse direction-relation matrices and directions between minimum bounding rectangles.

In order to represent directions between points and lines, Chapter 4 develops the deep direction-relation matrix by extending the coarse direction-relation matrix. The deep direction-relation matrix additionally records information about the intersections of a target object with boundaries of direction tiles using neighbor codes, if necessary. This chapter also demonstrates that a direction recorded using this model at a smaller scale is compatible with a direction recorded at a larger scale. The compatibility makes this model useful for querying spatial databases at multiple scales.

Chapter 5 develops a method for similarity assessment using detailed direction-relation matrices for regions. The basis of the similarity assessment is the conceptual neighborhood graph of direction relations. The problem of similarity assessment is formulated as a balanced transportation problem, which is solved using the transportation algorithm.

Chapter 6 discusses the implementation of a direction comparison system. This system allows users to draw two ordered pairs of polygons, computes the direction in each pair, and computes the similarity between the directions. It evaluates the method of similarity assessment developed in Chapter 5, and compares the soundness of mappings provided by the 4-neighborhood and 8-neighborhood conceptual graphs.

Chapter 7 summarizes the thesis, identifies contributions, and highlights possible further research based on the findings of this thesis.

Chapter 2

Models of Direction Relations

Models of direction relations are critical in visual interfaces to geographic information systems (Mark 1992). In the future, geographic information systems (GISs) that provide multi-modal user interfaces with voice interaction will incorporate such terms into their query languages (Egenhofer 1996) and will require inference mechanisms to process such queries. For example, a user may ask a natural-language query such as, “Find lakes in the state of Maine that are northeast of Orono,” and a GIS will be powerful and intelligent enough to interpret the semantics of the spatial constraints. Such spatial-reasoning tasks are intuitive to human reasoning, but in order to use the processing power of computers to assist them in these tasks, formal models of spatial concepts are needed. This chapter reviews the existing models of direction relations.

Reference frames are important for specifying directions; therefore, Section 2.1 discusses reference frames. A model of quantitative directions using Cartesian coordinates is described in Section 2.2. Section 2.3 discusses the utility of qualitative directions in GISs and reviews the models of cardinal direction for points. Directions between extended objects are specified using crude approximations such as centroids and minimum bounding rectangles; Section 2.4 reviews models for extended objects. Section 2.5 compares the models of directions between extended objects and assesses their suitability for the five major tasks of direction relations in information systems (Sections 1.3.1-5). Section 2.6 summarizes the strength and deficiencies of the existing methods.

2.1 Reference Frames

Direction is a binary relation that is recorded for an ordered pair of objects A and B ; A is the reference object, while B is the target object. The third part of a direction is the reference frame that assigns direction symbols to partitions of space. If the reference frame changes, the direction also changes. There are three types of reference frames (Retz-Schmidt 1988):

- An *intrinsic* reference frame refers to an object's intrinsic front, back, left, and right. For example, front of a building is determined by the main side or main entrance, such as the entrance of a church.
- A *deictic* reference frame is based on the observer's point of view. An observer based on one's own front side divides the space into four direction regions: front, back, left, and right.
- An *extrinsic* reference frame on the Earth is defined by the location of poles. In 2-D, it leads to the system of the four cardinal directions north, west, east, and south.

In GISs, the relative orientation of geographic objects are typically described by cardinal directions; therefore, this thesis uses an extrinsic reference frame.

2.2 A Model of Quantitative Directions

In geographic applications, the direction of a target point is typically defined with respect to a reference point using the azimuth. The azimuth is an angle between the meridian line that passes through the reference point A and the geodesic line from A to the target point B (Figure 2.1). The value of directions for all points on the meridian towards north with respect to A is 0° . For all other points, the azimuth is measured counter-clockwise from the northern part of the meridian and lies in a semi-open interval $[0, 360)$ (Equation 2.1). Angles that are more than 360° are mapped onto the interval in a cyclic fashion. For example, the angle 360° is mapped onto 0° and the angle 450° is mapped onto 90° .

$$\mathbf{a} = \tan^{-1} \left(\frac{x_B - x_A}{y_B - y_A} \right) \quad (2.1)$$

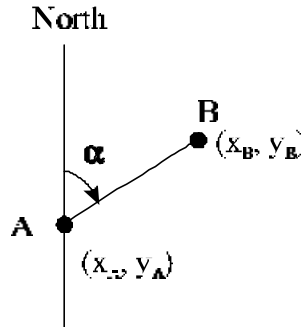


Figure 2.1: The azimuth angle.

Quantitative directions between two points are recorded using azimuth angles and their values can be approximated at a resolution suitable for their application. For example, the azimuth of *B* with respect to *A* may be $43^{\circ} 34' 45''$. Applications such as land surveying and mechanical design use quantitative directions. However, query languages in spatial databases use qualitative directions, not quantitative directions. Therefore, this thesis focuses on the formalization of qualitative directions.

2.3 Models of Qualitative Directions for Points

Qualitative directions are coarser approximations of directions than the quantitative direction and are described using a smaller set of symbols than the quantitative directions. Typically, the interval $[0^{\circ}, 360^{\circ})$ is divided into four or eight direction intervals, and an appropriate direction term is used for each interval (Frank 1996). A four-direction system uses the primary directions *north*, *south*, *east*, and *west* and an eight-direction system uses the primary directions and the secondary directions *northeast*, *southeast*, *southwest*, and *northwest*. People prefer qualitative directions over quantitative directions because often qualitative directions are all they need and most people are not able to compute trigonometric expression such as \tan^{-1} in their head. While querying a GIS, people are less likely to ask a question such as “Select lakes that have azimuth $43^{\circ} 34' 45''$ ” with

respect to Orono in Maine.” Instead, most people would like to formulate queries such as “Select lakes that are northeast of Orono in Maine.”

Models of cardinal directions use an extrinsic reference frame. Primarily, there are two types of models for the direction between two points: cone-based direction models (Section 2.3.1) and projection-based direction models (Section 2.3.2). Section 2.3.3 discusses a projection-based model that records directions of a point with respect to a line.

2.3.1 Cone-Based Direction Models for Points

The cone-based system partitions the space around a reference point into four (Figure 2.2a) or eight (Figure 2.2b) mutually exclusive partitions of 90° or 45° , respectively (Peuquet and Zhan 1987; Hong 1994; Abdelmoty 1995; Frank 1996; Shekhar and Liu 1998). The four-direction system uses the qualitative directions north (N), east (E), south (S), and west (W); whereas, the eight-direction system uses four additional directions: northeast (NE), southeast (SE), southwest (SW), and northwest (NW). A boundary between two direction partitions is assigned systematically, such as to the partition that is the clockwise neighbor of the boundary. For example, the boundary between the north and east partitions in the four-direction system is assigned to the east partition and the boundary between the east and south partitions is assigned to the south partition.

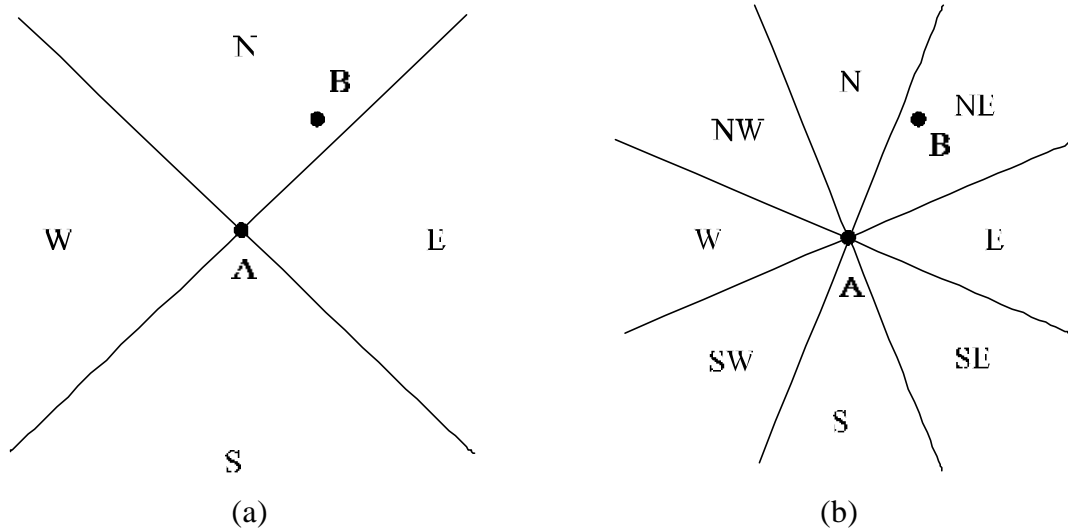


Figure 2.2: The cone-based model: (a) four-direction system and (b) eight-direction system.

If a target object coincides with the reference object, the direction between them is called *same*. Including the *same* direction, the four-direction system distinguishes among five directions and the eight-direction system distinguishes among nine directions. The direction of a target point with respect to a reference point is determined by the target's presence in a direction partition for the reference point. For example, in the four-direction system $dir(A, B)$ is *north* (Figure 2.2a), whereas in the eight-direction systems $dir(A, B)$ is *northeast* (Figure 2.2b).

2.3.2 Projection-Based Direction Models for Points

A projection-based direction model (Frank 1996) divides the space using horizontal and vertical lines passing through the reference point. A horizontal line divides the space into north and south half-planes (Figure 2.3a), whereas a vertical line divides the space into east and west half-planes (Figure 2.3b). Both horizontal and vertical lines together divide the space into four quadrants: northwest, northeast, southeast, and southwest (Figure 2.3c). The reference frame in the case of four quadrants consists of four direction regions, four lines, and a point. Direction regions are secondary directions *NW*, *NE*, *SW*, and *SE*; direction lines are primary directions *N*, *S*, *W*, and *E*; and *same* is the only point direction that coincides with the reference point.

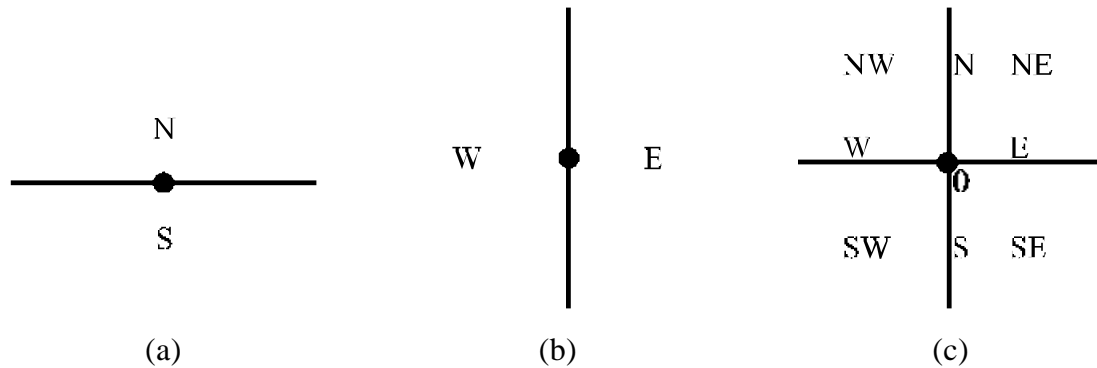


Figure 2.3: The projection-based direction model for point objects. The space is divided into (a) north-south half planes, (b) west-east half planes, and (c) nine directions.

The projection-based model (Figure 2.3c) has several advantages over the cone-based model (Figure 2.2b): (1) longitude and latitude parallels impose a structure on the globe that is identical to the projection-based model (Kulik and Klippel 1999), (2) the projection-based model results in a higher number of precise composition inferences compared to the cone-based model (Frank 1996), and (3) the projection-based model is easier to implement than the cone-based model in spatial databases due to the rectangular nature of the direction partitions. The cone-based model, on the other hand, is easier to scale up for higher number of qualitative directions, such as 16 or 32 directions than the projection-based model. However, the emphasis of this thesis is on computationally sound representations of directions and people rarely use more than nine qualitative directions; therefore, this thesis develops direction models based on projection-based partitions.

2.3.3 Directions between a Line and a Point

Freksa and Zimmermann developed a framework to represent directions of a point with respect to a line (Freksa 1992b; Freksa and Zimmermann 1992; Zimmermann 1993; Zimmermann and Freksa 1996). A point can be in any of the 15 qualitatively distinct locations with respect to a line (Figure 2.4a), and these relations can be represented distinctly using icons (Figure 2.4b).

This model is designed for intrinsic reference frames; therefore, it does not assume any external orientation and the line AB can be at any angle with respect to the x -axis. This model records the direction of a point with respect to a line without any approximations, which is a useful property of this model, however, it is not adequate for cardinal directions between extended objects due to its following shortcomings: (1) the line AB cannot always be parallel to a grid line in the extrinsic reference frame; therefore, it cannot always be used to record cardinal directions, (2) it does not apply to curved lines, (3) the target object must be approximated by a point, and (4) the method does not apply to regions. Section 2.4 discusses models of direction-relations for extended objects.

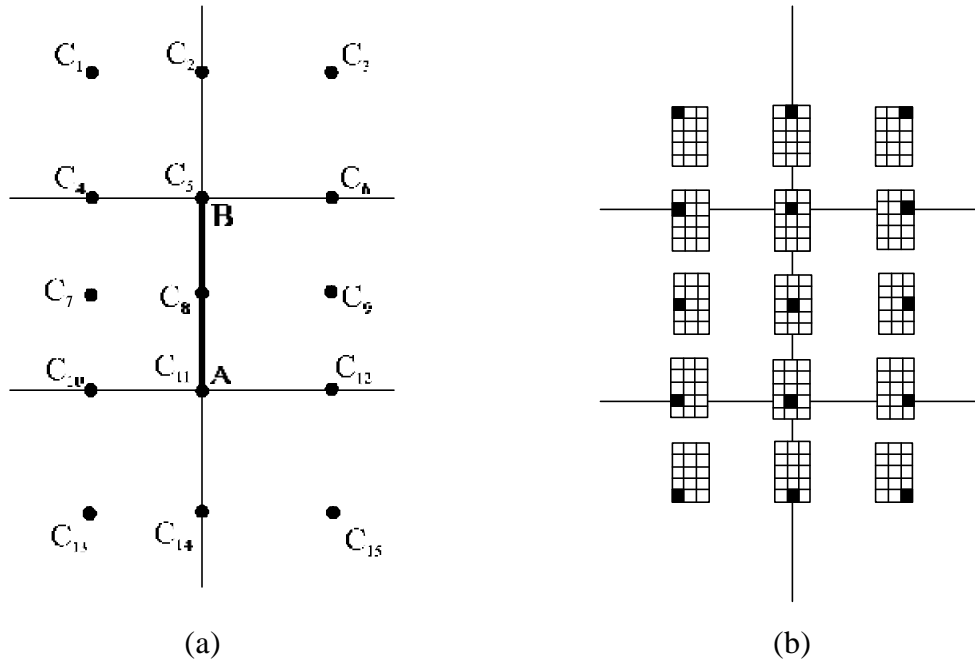


Figure 2.4: (a) Point C can be in 15 distinct qualitative locations with respect to the line AB , which forms the reference frame and (b) an iconic representations of these 15 relations.

2.4 Directions between Extended Objects

Although direction is well understood for point objects, people frequently specify directions between extended objects as well. For example, “Peeks-Kenny state park is

northwest of Orono, Maine.” This section reviews existing models for directions between extended objects.

2.4.1 The Triangular Model

The triangular model (Haar 1976) uses triangles of acceptance to determine a direction relation. Typically the object that is perceptually more prominent is used as the reference and the other object as the target. The embedding space is divided into four mutually exclusive cones or direction triangles of 90° each with respect to the reference object, such that all cones coincide with the centroid of the reference object. The centroid of a finite set of points p_1, \dots, p_N is their arithmetic mean $\left(\sum_{i=1}^N p_i \right) / N$ (Preparata and Shamos 1985). The area of acceptance of a direction triangle grows with increasing distance from the centroid. The target object is considered to be in the direction associated with a direction triangle in which its centroid is located (Peuquet and Zhan 1987). If the distance between objects is large compared to their sizes, the triangular model using centroids gives intuitive values for directions (Figure 2.5). However, the model is refined if the distance between objects is small compared to their sizes or if objects are overlapping, intertwined, or horseshoe-shaped.

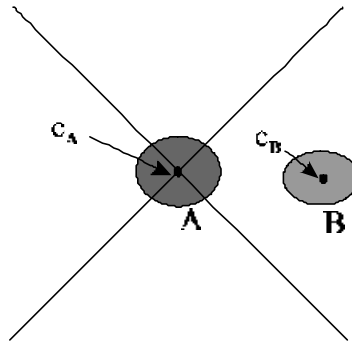


Figure 2.5: The centroid of B (c_B) is east of the centroid of A (c_A); therefore, $dir(A, B)$ is east.

If the distance between objects is small compared to their sizes, the centroid of a target object may not fall in the direction triangle to which it cognitively belongs. For example, in Figure 2.6a object B is visually east of object A , but B does not fall in the east triangle located at the centroid of A . In order to determine whether object B is in the east,

the area of acceptance of the east triangle is increased by moving the triangle's vertex backward (Figure 2.6b). The limit of this movement is a vertex point for the direction triangle such that the triangle lines pass through corners of the minimum bounding rectangle of the reference object. If the target object falls into the increased area of acceptance, the object is considered to be in the respective direction.

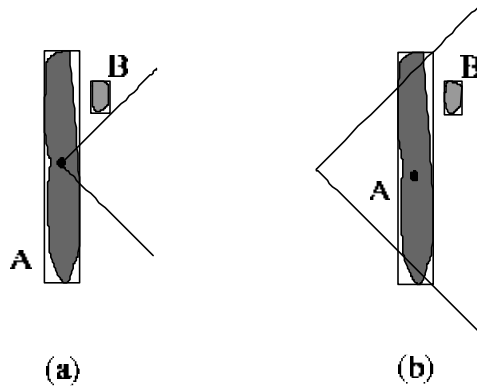


Figure 2.6: The triangular model for extended objects (a) B is visually east of A , but it does not fall in the east triangle, and (b) by adjusting the area of acceptance of east triangle, B is east of A .

If objects are overlapping, intertwined, or horseshoe-shaped, directions based on centroids can be misleading. For example, in Figure 2.7 the direction of object B 's centroid with respect to A 's centroid is north, even though no point of B is north of the complete object A . In such cases, a compound of four conditions determine whether an object is in a given direction or not.

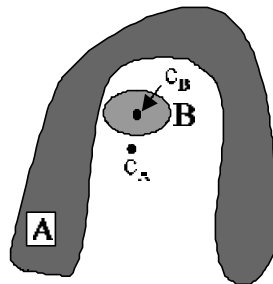


Figure 2.7: The centroid of B is north of the centroid of A , but B is not north of A .

In this model, there are special methods for computing directions for special conditions of objects, and there is no method that applies uniformly to all the cases. For example, the method that applies to the case in Figure 2.5 does not apply to the cases in

Figures 2.6 and 2.7. If objects are close to each other, triangles of two directions may intersect, for instance the north and east triangles, which may create confusion while specifying directions. This model is developed primarily to detect whether a target object exists in a given direction or not. If a target object is in multiple directions, such as {north, northeast, east}, this model does not provide a knowledge structure to represent such multiple directions. This model is computationally cumbersome and informal, due to overlapping direction triangles and special methods for computing directions for special cases; therefore, it is not suitable for direction queries in spatial databases.

2.4.2 Interval Relations in Two-Dimensions

Allen's interval relations (Allen 1983) represent spatial information in a 1-dimensional space. There are thirteen distinct relations (Figure 2.8), which are also called temporal interval relations or 1-dimension interval relations. Freksa (1992a) generalizes these interval relations to semi-intervals, where a semi-interval captures the information about either the beginning or the ending of an event, but not both.

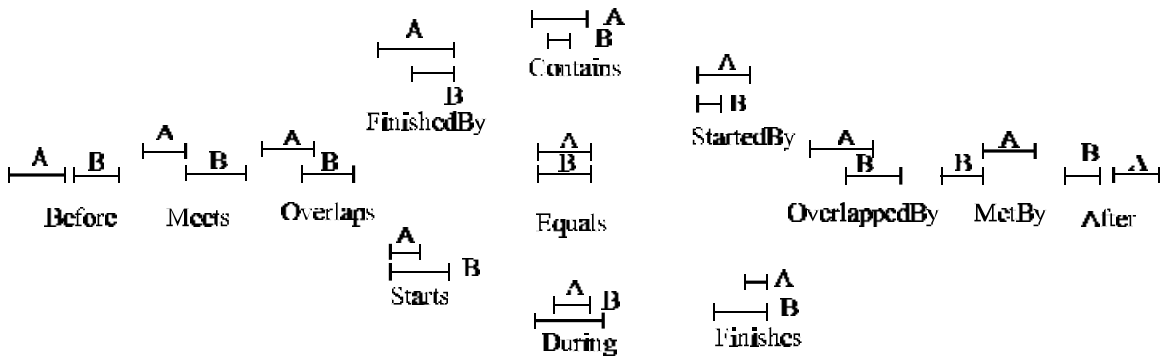


Figure 2.8: Allen's thirteen temporal interval relations in 1-dimension.

Guesgen (1989) extends 1-dimensional intervals to perform spatial reasoning by taking the projection of 2-dimensional objects onto the x - and y -axes. On an axis, he uses eight out of the thirteen interval relations and calls them as *left*, *attached*, *overlapping*, *inside*, and their converse relations (Figure 2.9). This model distinguishes $8 \times 8 = 64$ relations in 2-D. These relations are combinations of topological and directional relations in the rectangular world. This model approximates objects by their minimum bounding rectangles; therefore, the spatial relation may not necessarily be the same as the relation between exact representations of the objects. Sharma (1996) derives direction-relations

from thirteen 1-D interval relations between extended objects, and combines them with topological relations between objects to perform heterogeneous spatial reasoning. The direction model in Section 2.4.6 uses all thirteen relations on an axis.

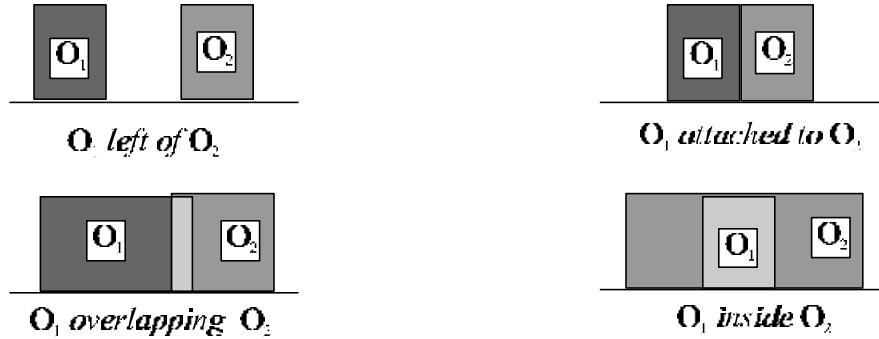


Figure 2.9: Four basic relations in 2-D (Guesgen 1989).

2.4.3 Qualitative Models for Space

Mukerjee and Joe (1990) and Mukerjee (1990) developed the qualitative models for space using Allen's interval relations. The basis for this model is a set of five relations of a point with respect to an interval: posterior (-), back (*b*), interior (*i*), front (*f*), and ahead (+) (Figure 2.10a). This model approximates objects by their minimum enclosing rectangles having sides parallel to an object's intrinsic front-back and left-right axes; therefore, it works best for the objects that have rectangular shapes. If these rectangles are projected onto the *x*- and *y*-axes, this model is similar to interval relations in 2-D (Guesgen 1989).

The qualitative model represents a relation between two intervals by a pair of interval-point relations. Each relation in a pair corresponds to the relation of an end point of the target interval with respect to the reference interval. This model generates a collision parallelogram by extending the support lines of both objects along their front-back axes (Figure 2.10b). The parallelogram has interval projections on the front-back axes of both objects. The relative position of an object with respect to another is represented by the relation of the object interval with respect to the interval that corresponds to the projection of the parallelogram on the object's axis. For example, *pos(A/B)*, the position of *A* with respect to *B* in Figure 2.10b is recorded as --. To record

the direction, both objects are approximated by points, and direction is recorded in the intrinsic reference frame as 1, 2, 3, and 4 for front-left, back-left, back-right, and front-right directions, respectively. For example, $dir(A/B)$ in Figure 2.10b is 2. Using this model, the spatial relation of A with respect to B in Figure 2.10b is recorded as $pos(A/B) = --$, $pos(B/A) = ++$, and $dir(A/B) = 2$.

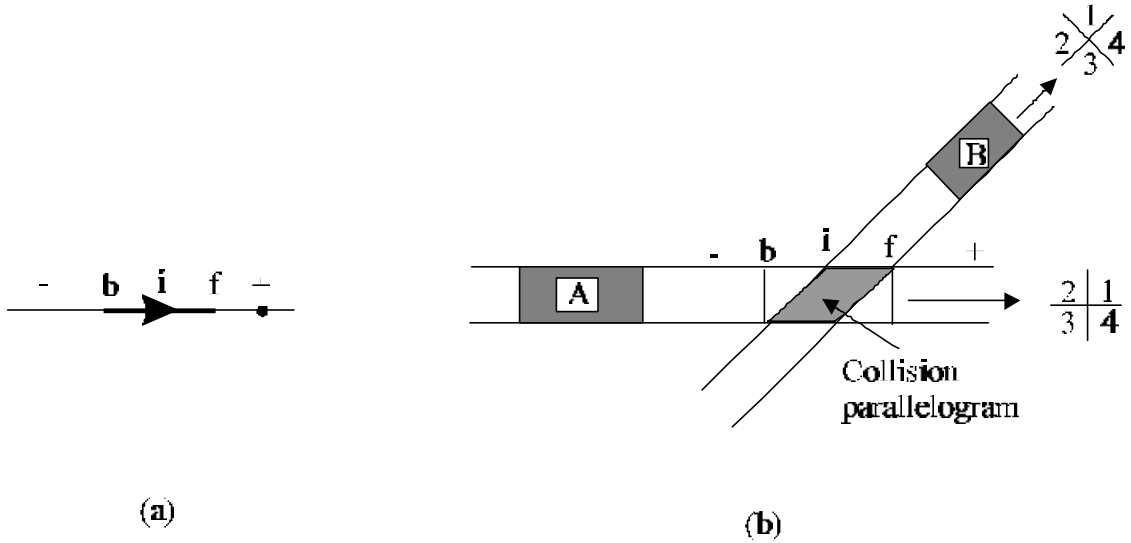


Figure 2.10: (a) Relations between an interval and a point and (b) relations between two extended objects at an arbitrary angle.

The primary difference between the qualitative model for space and Guesgen’s model is that the former records spatial relationships between two objects considering their intrinsic reference frames, whereas the latter records the projections on the x - and y -axes. The qualitative model also suffers from the approximation of the objects by their minimum enclosing rectangles, and does not address possibilities of a target object being in more than one quadrant, such as {front, front-right, right}.

2.4.4 Two-Dimensional Strings

Two-dimensional strings (Chang *et al.* 1987) are based on the projections of the objects on the x - and y -axes. A symbolic picture consists of a fixed size grid whose cells are filled by the empty space or symbols corresponding to physical objects in a scene. For example, the symbolic picture of the scene in Figure 2.11a is given in Figure 2.11b.

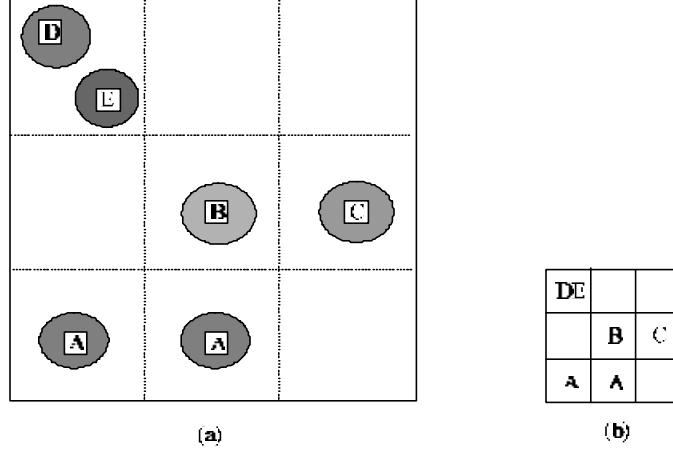


Figure 2.11: (a) A scene and (b) its corresponding symbolic picture.

Symbolic projections are the projections of a symbolic picture onto the x - and y -axes and they preserve the qualitative information along axes. Let Σ be a set of symbols representing the objects and A be the set $\{=, <, :\}$ such that the symbols “=”, “<”, and “:” do not belong to the set Σ . The symbols “=” and “<” specify the relations same and before, respectively between 1-D projections. The symbol “:” specifies “in the same set” in the 2-D symbolic picture. For example, the set Σ for the symbolic picture in Figure 2.11b is $\{a, b, c, d, e\}$. The 1-D strings on the x - and y -axes are $a=d:e<a=b<c$ and $a=a<b=c<d:e$, respectively. Thus the 2-D string for this symbolic picture is $(a=d:e<a=b<c, a=a<b=c<d:e)$.

Chang and Jungert (1996) used generalized 2-D strings to improve qualitative descriptions for extended objects over 2-D strings. Generalized 2-D strings use five-tuples $(\Sigma, C, E_{op}, e, \langle, \rangle)$, where Σ is a set of symbols representing the objects in the picture, C is the cutting line mechanism, $E_{op} \{<, =, |\}$ is the set of extended spatial operators, e is empty space of any size and shape, and “ \langle, \rangle ” is a pair of operators used to describe a local structure. For example, the generalized string on the x -axis for the picture in Figure 2.12 is $u=A/BeA/Be/CeB/C<D$, omitting e we get $u=A/BA/B/CB/C<D$. Similarly, the generalized string on the y -axis is $v=B/DeB/D<C/AeC/C$, omitting e we get $v=B/DB/D<C/AC/C$. The symbolic description of the picture using the generalized 2-D string is $\{A/BA/B/CB/C<D, B/DB/D<C/AC/C\}$.

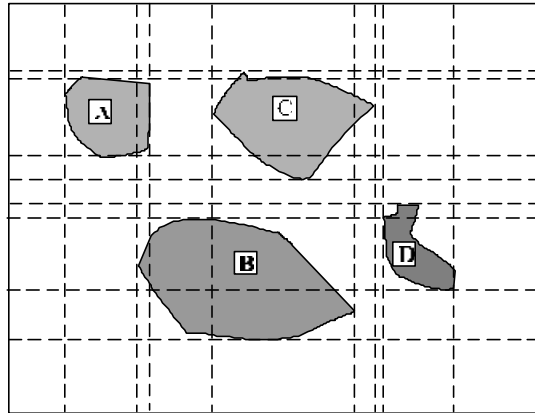


Figure 2.12: A scene with extended objects and cutting lines.

The 2-D C-string (Lee and Hsu 1990) is another string-based representation for extended objects using a larger set of operators $\{<, =, |, \%, [,], \setminus, /\}$. These operators are able to describe Allen's thirteen interval relations between 1-D projections. Chang and Jungert (1996) describe variations of 2-D strings.

Pictures can be indexed according to the 2D-strings of corresponding symbolic pictures and similar scenes from image databases can be retrieved by matching 2-D strings (Lee *et al.* 1989; Lee and Hsu 1990; Chang and Wu 1992; Lee *et al.* 1992). Models based on 2-D strings are formal and can be used across different dimension objects, but they too approximate extended objects. These models do not encode directions explicitly, and one will have to derive the direction information from symbols such as “<” and “=” in the 2-D strings. Models described in this section support similarity between 2-D strings, but have no provision for computing similarity between directions; therefore, they cannot be used for comparing directions.

2.4.5 Symbolic Arrays

Symbolic arrays (Glasgow and Papadias 1992) are based on the following cognitive factors: (1) hierarchical deep representations of the pictures are used in the long term memory, (2) visual and spatial representations are used in the working memory, and (3) topology and direction relations are preserved in mental representations. Symbolic arrays

preserve the order of an object's appearance along the x - and y -axes, and they can be used in a hierarchical fashion. An object at a higher resolution can be considered as consisting of many parts, and this information is represented using nested arrays. For example, in Figure 2.13a object A consists of two parts. At a higher-level A is shown as one symbol, but at a lower level it is shown as consisting of two parts in Figure 2.13b.

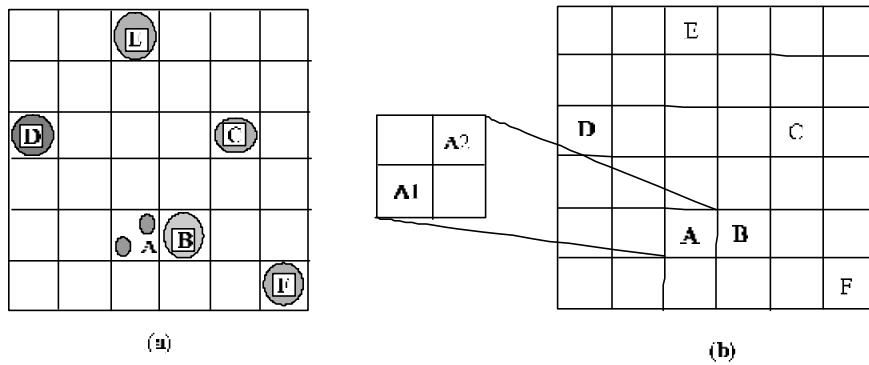


Figure 2.13: (a) A scene and (b) the corresponding symbolic array.

From the symbolic arrays, the information about the topological and directional relations between the objects is derived. The symbolic arrays are useful in visualizing the arrangement of the objects in the space. This arrangement can be used for retrieving the pictures that are topologically and directionally equivalent. Symbolic arrays also suffer from the approximation of objects and have no explicit representation of direction relations.

2.4.6 Directions between Minimum Bounding Rectangles

An approach for representing directions between extended objects is the spatial relation between minimum bounding rectangles (MBRs) of objects (Papadias *et al.* 1995). Reasoning between projections of MBRs on the x - and y -axes can be performed using 1-D interval relations. For example, in Figure 2.14, the projection of B on the x -axis ($proj_B^x$) is before $proj_A^x$ and $proj_B^y$ is before $proj_A^y$; therefore, the relation between MBRs of objects B and A is (*before, before*). Using this method, one can characterize relations between MBRs of objects uniquely. There are thirteen possible relations on an axis, therefore, this model distinguishes $13*13=169$ relations (Figure 2.15).

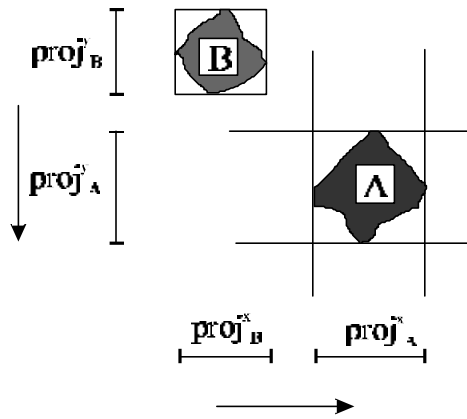


Figure 2.14: Spatial relation between two minimum bounding rectangles.

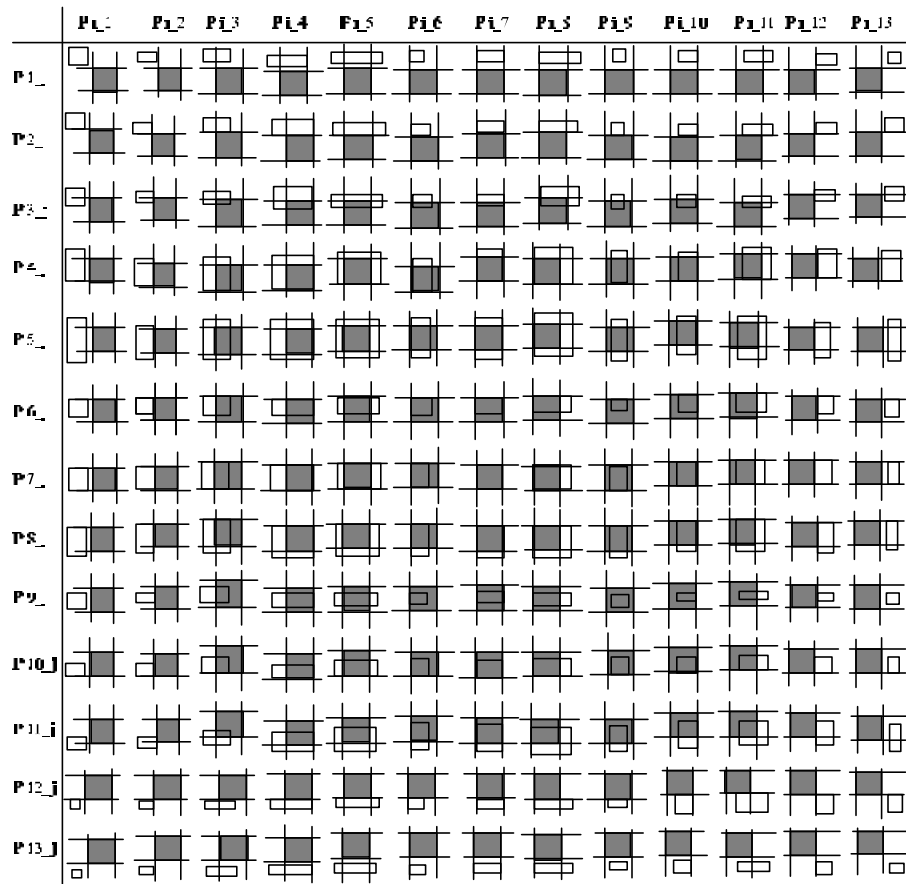


Figure 2.15: The spatial relations between minimum bounding rectangles (Papadias *et al.* 1995).

The basic difference between this method and Guesgen's method (Section 2.4.2) is that the latter uses 8 relations out of 13 interval relations on an axis and distinguishes only $8*8 = 64$ distinct spatial relations, whereas this model distinguishes 169 relations.

2.5 Comparisons of the Direction Models

The focus of this thesis is the representation of directions between extended objects; therefore, this section compares the models reviewed in Sections 2.4.1-6. The cone-based model for points (Section 2.3.1), the projection-based model for points (Section 2.3.2), and directions between a line and a point (Section 2.3.3) are not included in this comparative study, because they cannot represent directions between region objects. A direction model that has all required properties of a direction-relation system (Section 1.4.1) can be used for applications in Section 1.3. Therefore, models for direction between extended objects are evaluated for the following properties: *formal*, *inferential*, *shape-sensitive*, *dimension-neutral*, and *comparable* (Table 2.1). Shape-sensitiveness in this thesis refers to the use of the objects' geometries as they are without any approximation.

The triangular model (Section 2.4.1) approximates a target object's geometry by its centroid only when the distance between objects is large compared to their sizes. The models based on the 2-D strings and the symbolic arrays represent scenes at a high level of abstraction, and they can be used for the task of scene comparison. However, these models do not record direction relations; therefore, these models are not suitable for inferring and comparing directions. It is evident from Table 2.1 that there is no model that supports all five criteria; therefore, no model has the properties desired in the problem statement of this thesis.

Model	Properties of direction relations				
	Formal	Inferential	Shape-sensitive	Dimension-neutral	Comparable
Triangular model (Peuquet and Zhan 1987)	no	yes	sometimes	yes	no
Interval relations in 2-D (Guesgen 1989)	yes	yes	no	no	yes
Qualitative models for space (Mukerjee and Joe 1990)	yes	yes	no	no	no
2-D Strings (Chang <i>et al.</i> 1987)	yes	no	no	yes	no
Symbolic Arrays (Glasgow and Papadias 1992)	yes	no	no	yes	no
Direction between MBRs (Papadias <i>et al.</i> 1995)	yes	yes	no	no	yes

Table 2.1: Evaluation of the directions models for extended objects in 2-D.

2.6 Summary

In this chapter, we reviewed existing models of directions for points and extended objects. Models for directions between extended objects were evaluated to check whether or not these models have the five required properties (Section 1.4.1): *formal*, *inferential*, *shape-sensitive*, *dimension-neutral*, and *comparable*. The models for directions between extended objects were found to approximate objects by their centroids or minimum bounding rectangles; such approximations can yield misleading results. None of these models was found to be suitable for applications such as direction queries in spatial databases and content-based retrieval using directions. Chapter 3 develops a new model

based on direction-relation matrices to represent exact directions between extended objects that can be used for querying spatial databases and content-based retrieval, and this model addresses all the five requirements.

Chapter 3

Direction-Relation Matrix for Region Objects

Cardinal directions between points are fairly well understood, and they can be represented using the projection-based and cone-based models (Chapter 2). There is, however, no model for directions between extended objects that uses exact geometries of the objects (Table 2.1). Therefore, directions between extended objects have been represented using crude approximations of object geometries, such as centroids and minimum bounding rectangles (MBRs). The spatial relations between centroids of the objects often do not conform to the relations between the objects. The direction of B with respect to A in Figure 3.1 is {north, northeast, east}. If MBR approximations are used for recording direction, the direction of B with respect to A would be recorded wrongly as {north, northeast, east, same}.

It is clear that existing models of cardinal directions have serious limitations due to approximations of object geometries; therefore, a comprehensive model of cardinal directions that does not approximate the objects' geometries is needed. This chapter introduces an improved representation for cardinal directions between connected regions, which is compatible with directions described for point-like spatial objects. It is characterized by a cognitively plausible equivalence class for cardinal directions. This model partitions space around the reference object using the projection-based method (Sections 2.3.2 and 2.4.6) and the exact shape of the reference object. Coarse direction-relation matrices (Section 3.1) record a purely qualitative description (i.e., into which tile the target object falls). Since cardinal direction relations derived with this method do not necessarily imply the converse relation, the same method must be applied in the reverse direction (i.e., partitioning space around the target object and recording the distribution of the reference object across the tiles). Section 3.2 investigates how many direction

relations can be distinguished using coarse direction-relation matrices. Section 3.3 analyzes the effects that shape and size of objects, as well as the distance between them, have on direction relations. Detailed direction-relation matrices (Section 3.4) enhance qualitative description using quantitative values (i.e., how much of the target object falls into any particular tile). Descriptions based on coarse and detailed direction-relation matrices operate without metric values, such as radians or distances; therefore, they lead to broad equivalence classes (i.e., the same direction description applies to a series of conceptually similar configurations). Section 3.5 compares the expressive power of the coarse direction-relation matrix with the expressive power of MBR relations. Section 3.6 summarizes the results of this chapter.

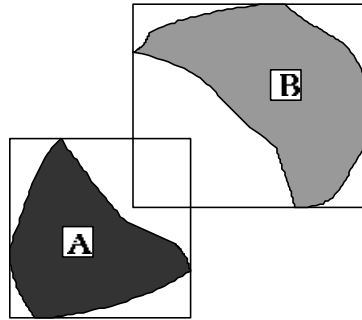


Figure 3.1: Two objects with their minimum bounding rectangles.

3.1 Coarse Direction-Relation Matrix

To overcome the limitations of MBR directions, we introduce a model that better captures the influence of the objects' shapes. It applies to direction relations between regions, that is, objects homeomorphic to connected 2-disks. This model uses the projection-based method around the reference object (i.e., the object *from* which the direction relation is described) and considers the exact representation of the target object (i.e., the object *to* which the direction relation is described). The projection-based method, applied around the reference object, partitions the embedding space into nine mutually exclusive regions, called the *direction tiles* (Figure 3.2), whose union forms a complete partition of space. The direction tiles at the periphery correspond to the eight cardinal directions—north (N_A), northeast (NE_A), east (E_A), southeast (SE_A), south (S_A), southwest (SW_A), west (W_A), and northwest (NW_A)—while the tile at the center, called

same (0_A), coincides with the minimum bounding rectangle of the reference object A (Sections 2.3.2 and 2.4.6). The boundaries between any neighboring tiles have no extent.

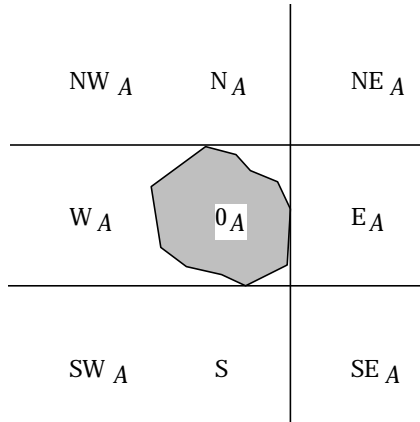


Figure 3.2: The nine tiles resulting from the partitioning of space around the reference object A .

We describe the cardinal direction from the reference object to a target by recording those tiles into which at least one part of the target object falls (Figure 3.3a). This method is more detailed than the MBR relations, particularly for non-convex shapes; however, it gives up converseness, an important property of cardinal-direction reasoning. For instance, if B is in (N_A, NE_A, E_A) (Figure 3.3a), then A is not necessarily (S_B, SW_B, WB) , but depending on A 's shape it may be $(S_B, 0_B, WB)$; $(SW_B, 0_B, WB)$; $(S_B, 0_B, SW_B)$; or $(S_B, 0_B, SW_B, WB)$ as well (Figure 3.3b); therefore, a scene description with coarse cardinal directions requires for each pair of objects the calculation and storage of two relations—one from A to B and another from B to A .

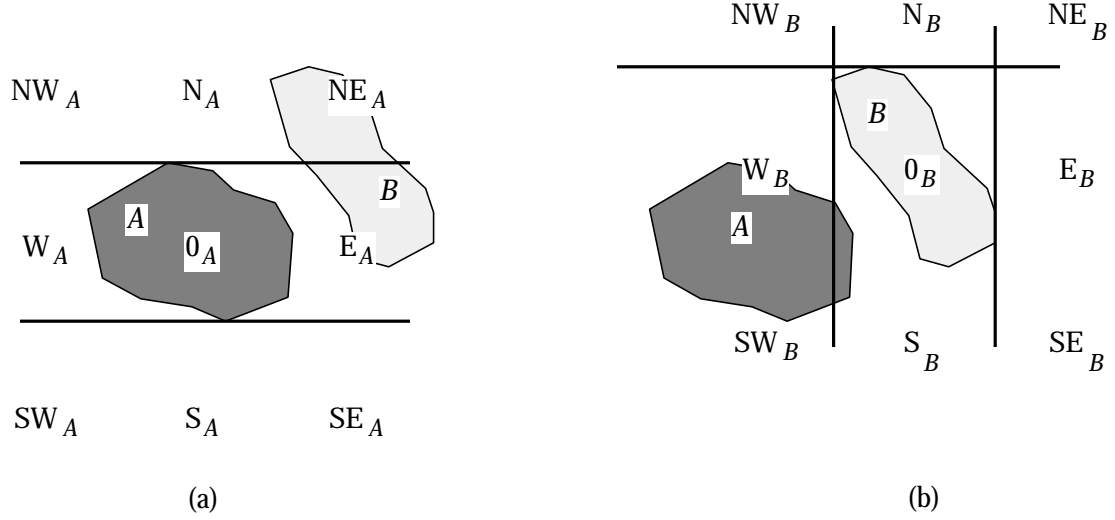


Figure 3.3: Capturing the cardinal direction relation between two areal objects, A and B , through the projection-based partitions (a) around A as the reference object and (b) around B as the reference object.

We capture *coarse cardinal directions* between two regions in the *direction-relation matrix* dir_{RR} , a 3×3 matrix that preserves the neighborhood of the partition around the reference object and registers the intersections between the target and the tiles around the reference object (Equation 3.1). The elements in the direction-relation matrix have the same topological organization as the partition around the reference object. Following the usual categorization of neighboring cells in square tessellations, two tiles are *4-neighbors* if they are horizontally or vertically adjacent (Samet 1989b).

$$dir_{RR}(A, B) = \begin{bmatrix} NW_A \cap B & N_A \cap B & NE_A \cap B \\ W_A \cap B & 0_A \cap B & E_A \cap B \\ SW_A \cap B & S_A \cap B & SE_A \cap B \end{bmatrix} \quad (3.1)$$

To describe coarse cardinal directions, we consider the emptiness and non-emptiness of the nine intersections. For example, the two direction-relation matrices for the two configurations in Figures 3.3a and 3.3b are given in Equations 3.2a and 3.2b, respectively. A direction relation is *4-connected* if all pairs of non-empty cells are transitively 4-neighbors.

$$dir_{RR}(A,B) = \begin{bmatrix} \emptyset & \neg\emptyset & \neg\emptyset \\ \emptyset & \emptyset & \neg\emptyset \\ \emptyset & \emptyset & \emptyset \end{bmatrix} \quad (3.2a)$$

$$dir_{RR}(B,A) = \begin{bmatrix} \emptyset & \emptyset & \emptyset \\ \neg\emptyset & \neg\emptyset & \emptyset \\ \neg\emptyset & \neg\emptyset & \emptyset \end{bmatrix} \quad (3.2b)$$

3.2 Realizability of the Coarse Direction-Relation Matrix

The coarse direction-relation matrix has nine elements, which yields $2^9=512$ possible distinct configurations. However, not all configurations are possible for direction relations. For example, a matrix with all elements having an empty value is impossible, as such a matrix would mean that the target object is absent. A matrix that has a non-empty value in the northwest and southeast elements, and an empty value in all other elements is also impossible, because this implies that the target object is disconnected. This section gives consistency constraints for regions, introduces icons to represent directions, and examines all possible directions.

3.2.1 Consistency Constraints

For any direction relation with a non-empty target object, at least one of the nine direction tiles must be non-empty. More than one intersection is non-empty if the target object extends through more than one direction tile. A direction-relation matrix with exactly one non-empty intersection is referred to as a *single-item* direction-relation matrix, whereas those matrices with more than one non-empty intersection are called *multi-item* direction-relation matrices. If the target object is a region, any two non-empty intersections have to be 4-connected; otherwise, the representation in a direction-relation matrix would be inconsistent. The examples in Equations 3.3a and 3.3b show inconsistent direction-relation matrices for region target objects, because the two non-empty intersections with the direction tiles are not 4-neighbors.

$$dir_{RR}(A, B) = \begin{bmatrix} \emptyset & \emptyset & \neg\emptyset \\ \neg\emptyset & \emptyset & \emptyset \\ \emptyset & \emptyset & \emptyset \end{bmatrix} \quad (3.3a)$$

$$dir_{RR}(A, B) = \begin{bmatrix} \emptyset & \emptyset & \neg\emptyset \\ \emptyset & \neg\emptyset & \emptyset \\ \emptyset & \emptyset & \emptyset \end{bmatrix} \quad (3.3b)$$

3.2.2 Iconic Representation of Direction-Relation Matrix

We use an iconic representation of the direction-relation matrix in the form of a 3×3 tessellation of square cells, where white cells stand for empty tile intersections, while black cells represent non-empty intersections (Figure 3.4a). If a value of a cell is unknown (i.e., the entry is empty or non-empty), that cell is shown in gray (Figure 3.4b).



Figure 3.4: The iconic representation of the direction-relation matrix: (a) a configuration with three non-empty cells and (b) a configuration with one non-empty cell and two cells that are either empty or non-empty.

A direction-relation icon with gray cells stands for the disjunction of all possible combinations with white and black cells, as long as at least one of the cells is black and all black cells are 4-connected. For example, the icon in Figure 3.5a with two gray cells would have 2^2 possible configurations (Figure 3.5b-e), three of which are legal, while one is not allowed since its black cells are not 4-connected (Figure 3.5d). Given the constraints about 4-connectedness, only a subset of direction-relation matrices may be realized—218 out of $2^9 = 512$ possible combinations (Figure 3.6).

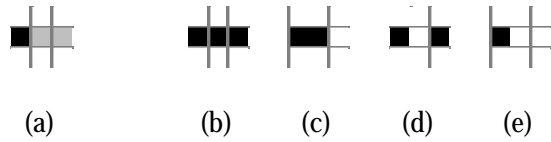


Figure 3.5: The interpretation of gray cells under the 4-connectivity constraint: (a) a configuration with gray cells and (b)-(e) the set of possible black-and-white configurations, with (d) being an illegal configuration because it is not 4-connected.

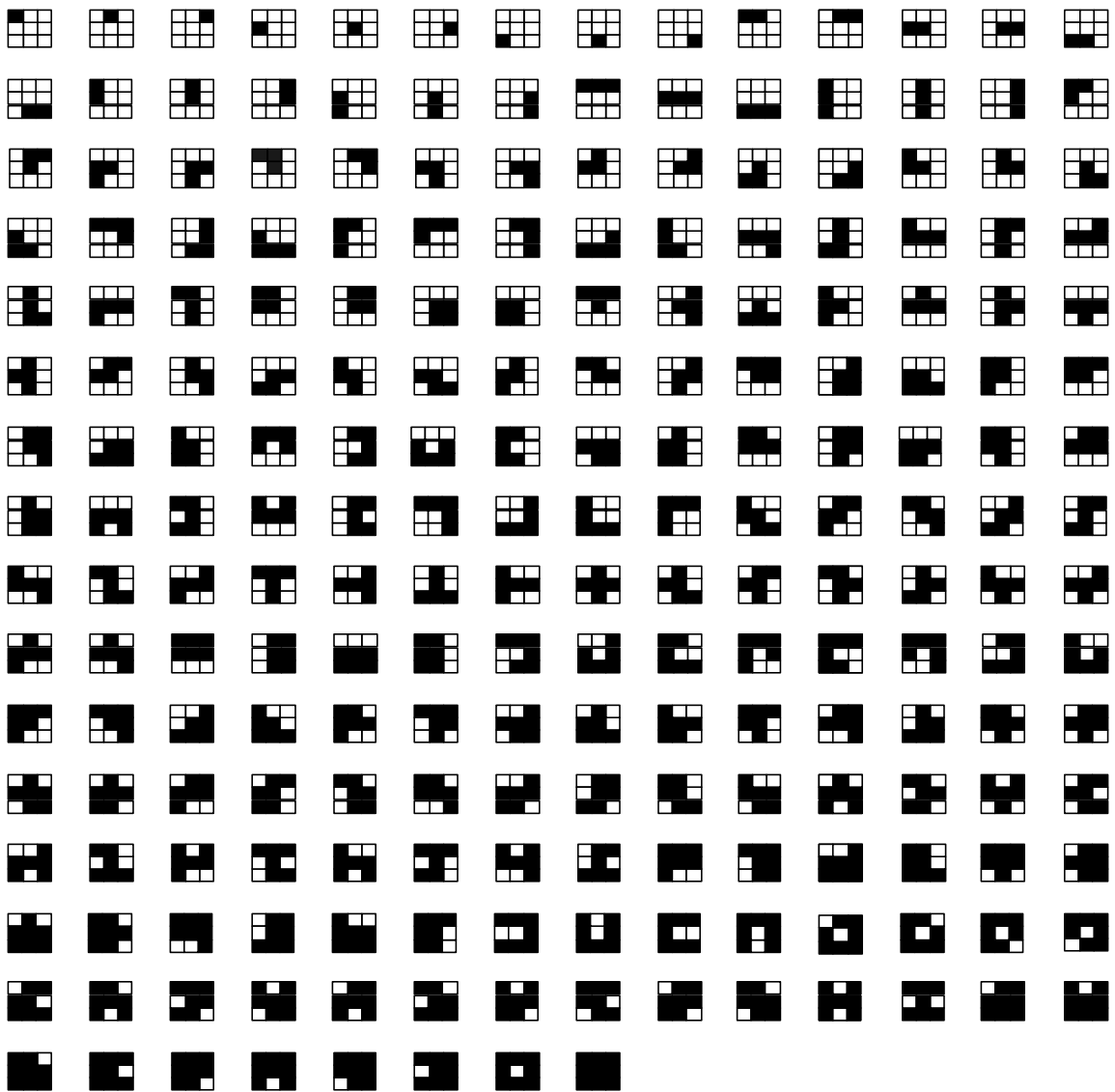


Figure 3.6: An iconic representation of the 218 direction-relation matrices that can be realized between two regions.

There are nine configurations with one non-empty tile; twelve configurations with two non-empty, 4-connected tiles; 22 configurations with three non-empty, 4-connected tiles; 36 configurations with four non-empty, 4-connected tiles; 49 configurations with five non-empty, 4-connected tiles; 48 configurations with six non-empty, 4-connected tiles; 32 configurations with seven non-empty, 4-connected tiles; nine configurations with eight non-empty tiles; and one configuration with nine non-empty tiles. These 218 situations can be recursively derived, starting with the first 9 single-item direction-relation matrices, by adding a 4-connected black cell into each possible location. With the help of a software prototype for direction-relation matrices, these configurations have been confirmed computationally (Section 6.3.2).

3.3 Effects of Shape, Size, and Distance

Properties of objects, such as shape and size and distance between them, affect their direction relations (Section 2.4.1). In this section, we study changes in directions between two objects, while the reference object is kept fixed and the target object is subjected to one of the following changes.

- For a given shape and distance, if the size of the target object is increased, the target object may intersect with more direction partitions (Figures 3.7a and 3.7b). If the size of the target object is decreased, it may intersect with fewer direction partitions.
- For a given shape and size, if the distance between the objects is reduced, the target object may intersect with more direction partitions (Figures 3.7a and 3.7c). If the distance between the objects is increased, the target object may intersect with fewer direction partitions.
- For a given distance and the size fixed by the area, a change in the shape of the target object may change its intersection with direction partitions. Shape is an attribute that is difficult to define, but, for the sake of this illustration, eccentricity (Gonzalez and Woods 1992) is used (Figures 3.7a and 3.7d).

Changes in shape and size of the reference object also affect direction relations, as these changes modify direction partitions.

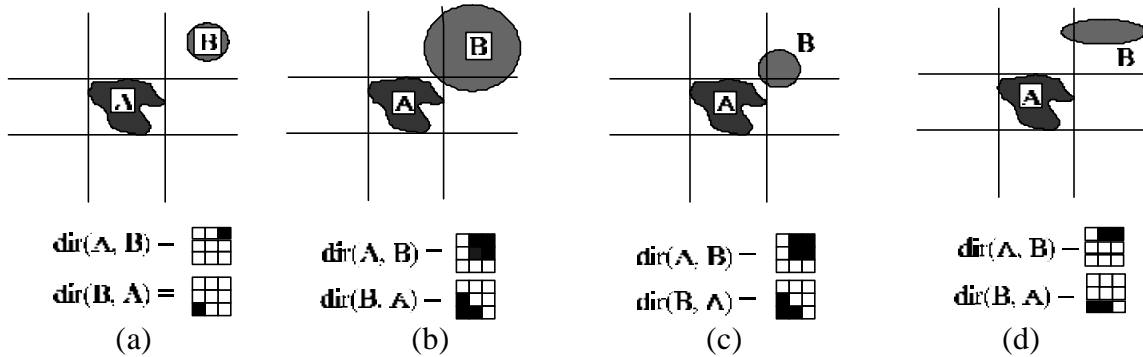


Figure 3.7: (a) The original configuration, (b) the size of the target object is increased, (c) the distance between the reference and target object is decreased, and (d) the target object is elongated.

3.4 Detailed Direction-Relation Matrix

The coarse direction-relation matrix records whether or not a target object falls in a direction tile or not. An interpretation of the mere fact that an object falls within some direction tile(s) of another object may be misleading or inappropriate. For example, the directions in Figures 3.8a and 3.8b may be considered to be more similar than those in 3.8b and 3.8c, because in 3.8a and 3.8b most or all of B is in the tile NE_A , while almost nothing of B is in tile NE_A in Figure 3.8c. The coarse direction-relation matrix, however, would identify 3.8b and 3.8c as equivalent, but 3.8a and 3.8b as different.

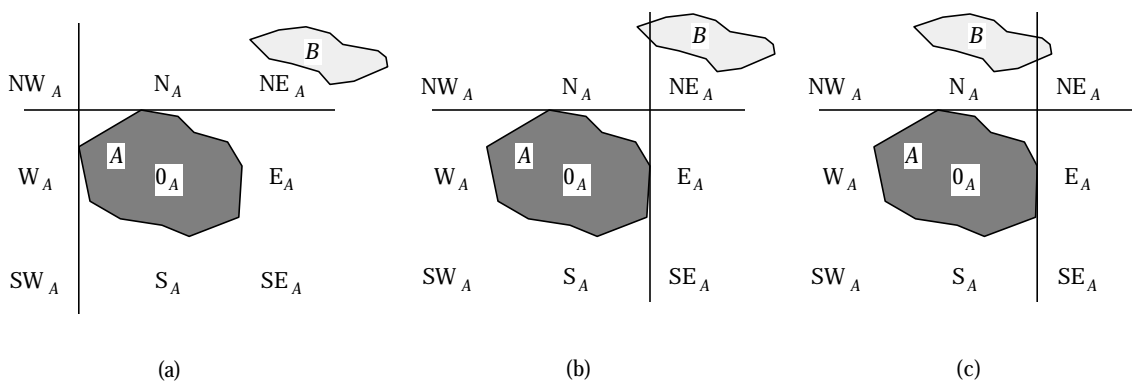


Figure 3.8: Three scenes for which the coarse direction-relation matrix serves as a cognitively inappropriate equivalence relation.

To provide more detail about directions among objects, we extend the cardinal-direction method describing refinements of the coarse cardinal directions by considering additional criteria for non-empty intersections. The refinements considered here are based on the areal distribution throughout the direction tiles, and if a tile contains more than one disconnected part of the target, the areal distribution within such tiles.

3.4.1 Areal Distribution

The *areal distribution* captures how much of the target object falls into each tile. For non-empty tiles, we record for each object that falls into more than one direction tile the percentage of the common intersection between a tile and the object. The refinement measure implies the value 0% for empty intersections with that tile and 100% if and only if the entire object falls into a single tile. Since detailed cardinal directions refine coarse cardinal directions, they inherit the property that the relations derived from this model are not necessarily converse; therefore, for each pair of objects two relations must be calculated and stored—from A to B and from B to A .

This refined model leads to a variation of the direction-relation matrix, recording normalized areas *in lieu* of empty and non-empty intersections between the tile and the target object (Equation 3.4). The range of each detailed cardinal direction x is $0 \leq x \leq 1.0$ and the sum of all ratios for an object with respect to the direction partition of another object must be 1.0.

$$dir_{RR}(A, B) = \begin{bmatrix} \frac{\text{area}(\text{NW}_A \cap B)}{\text{area}(B)} & \frac{\text{area}(\text{N}_A \cap B)}{\text{area}(B)} & \frac{\text{area}(\text{NE}_A \cap B)}{\text{area}(B)} \\ \frac{\text{area}(\text{W}_A \cap B)}{\text{area}(B)} & \frac{\text{area}(0_A \cap B)}{\text{area}(B)} & \frac{\text{area}(\text{E}_A \cap B)}{\text{area}(B)} \\ \frac{\text{area}(\text{SW}_A \cap B)}{\text{area}(B)} & \frac{\text{area}(\text{S}_A \cap B)}{\text{area}(B)} & \frac{\text{area}(\text{SE}_A \cap B)}{\text{area}(B)} \end{bmatrix} \quad (3.4)$$

Equations 3.5a and 3.5b show the detailed direction-relation matrices for the configurations shown in Figures 3.3a and 3.3b, respectively.

$$dir(A, B) = \begin{bmatrix} 0 & 0.05 & 0.45 \\ 0 & 0 & 0.50 \\ 0 & 0 & 0 \end{bmatrix} \quad (3.5a)$$

$$dir(B, A) = \begin{bmatrix} 0 & 0 & 0 \\ 0.63 & 0.05 & 0 \\ 0.29 & 0.03 & 0 \end{bmatrix} \quad (3.5b)$$

3.4.2 Component Distribution

As a target object's position is assessed with respect to the partitioning around the reference object, multiple topologically disconnected parts of the target object may exist for a single tile. Such disconnected parts within a tile are called *components*. Two types of disconnected parts may exist: components whose closures are disconnected—called strongly disconnected—(Figure 3.9a) and a weaker notion of components if they have connected closures, but disconnected interiors—called weakly disconnected (Figure 3.9b). In both cases, each non-empty tile has at least one component, while each empty tile has no components. The distribution of the components across the tiles is captured by recording the counts of components per tile. Equation 3.6 shows the component distribution for Figure 3.9a (and for Figure 3.9b under weakly-connected components).

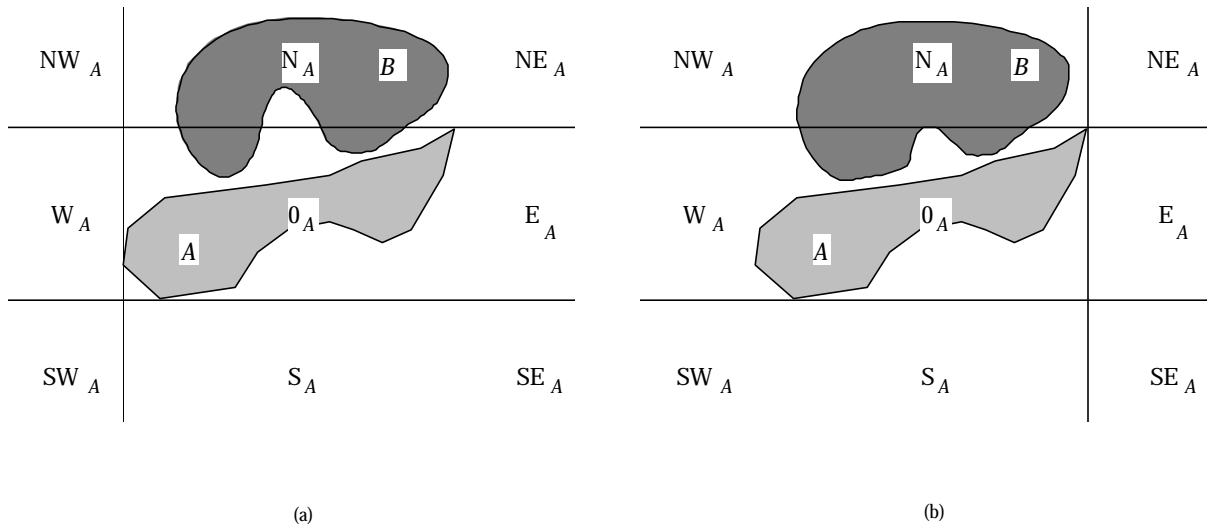


Figure 3.9: Tile O_A with (a) two strongly disconnected components of B and (b) two weakly disconnected components.

$$dir(A,B) = \begin{bmatrix} 0 & \#1 & 0 \\ 0 & \#2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (3.6)$$

3.4.3 Areal Distribution of Components

Whenever a tile contains more than one component, the components may vary in size. For example, in Figure 3.10 the Northern tile intersects with four components of B —two rather small ones and two larger ones. In order to distinguish such differences, it is necessary to capture the area of each component (Equation 3.7a). While the mere listing of such component areas provides information about what parts of B are located in that particular tile, it does not capture the distribution of the components throughout a tile. This can be achieved, however, by recording the sequence of the areas of the components, ordered along the boundary of the tile (Equation 3.7b).

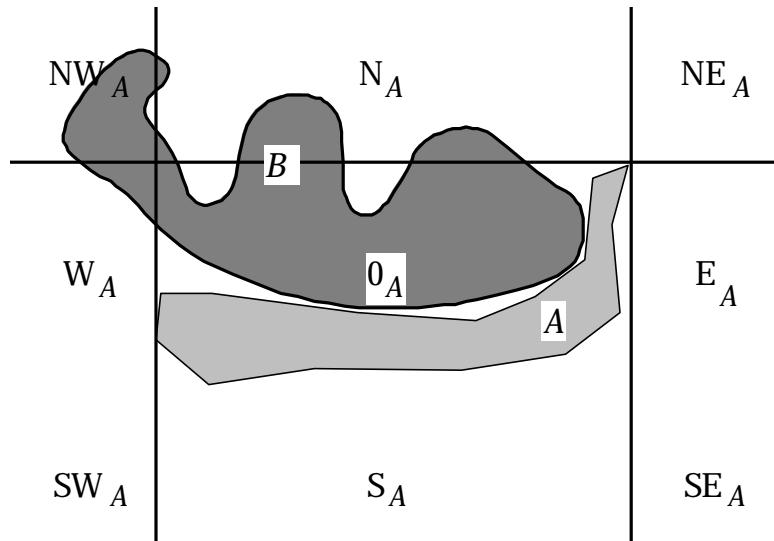


Figure 3.10: A direction relation with four components in A 's Northern tile.

$$dir(A,B) = \begin{bmatrix} 0.12 & (2 * 0.01, 0.04, 0.09) & 0 \\ 0.03 & 0.70 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (3.7a)$$

$$dir(A, B) = \begin{bmatrix} 0.12 & \begin{array}{|c|} \hline 0.01 & 0 \\ \hline \end{array} & 0 \\ 0.03 & \begin{array}{|c|} \hline 0.01 & (0.09, 0.04) & 0 \\ \hline \end{array} & 0 \\ 0 & 0.70 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (3.7b)$$

3.5 Comparison with MBR Direction Relations

Compared to the 169 MBR configurations (Figure 2.15), the direction-relation matrix provides more detail for concave shapes, but is less sensitive to the small changes along the boundaries of the partitions. For example, the direction-relation matrix may give a better assessment of the direction relation when one region surrounds another one than the MBR relations do (Figure 3.11); however, it does not capture subtle differences about alignments of the objects' minimal and maximal extents, while the MBR relations do distinguish these cases (Figure 3.12).

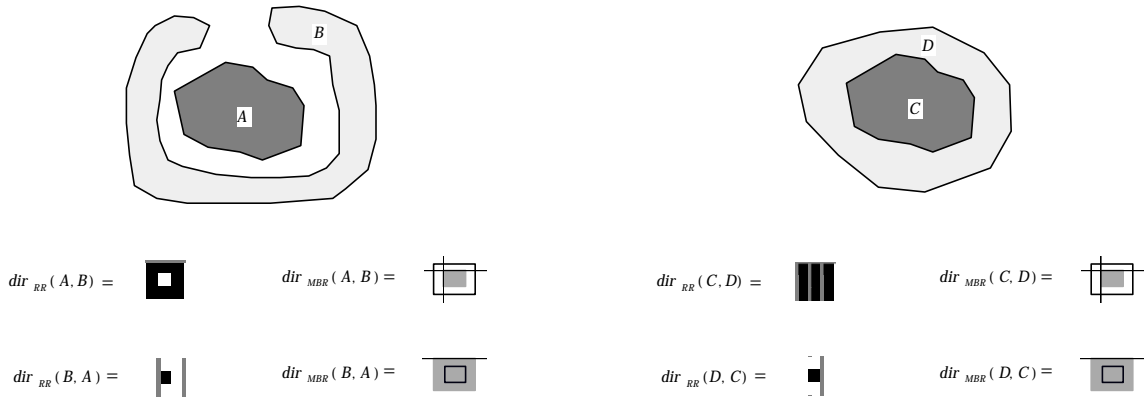


Figure 3.11: A configuration whose cardinal direction is better captured by the direction-relation matrix than by MBRs.

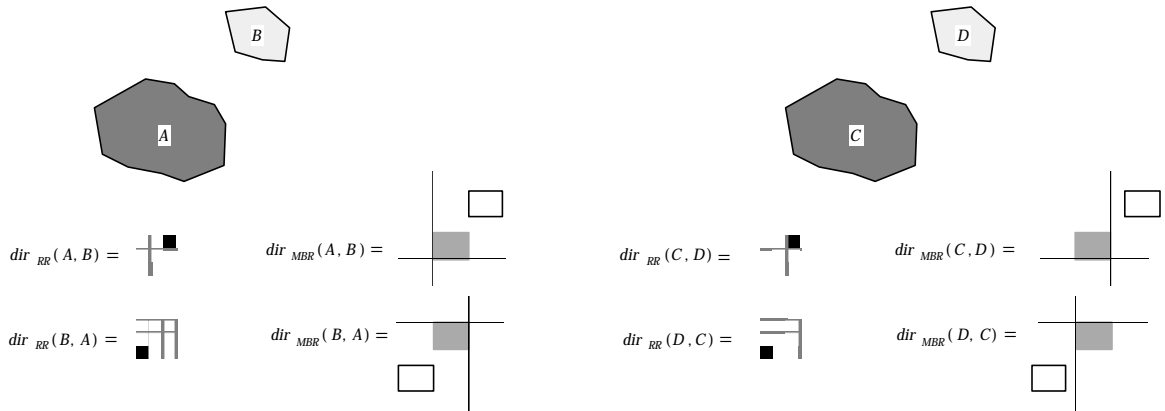


Figure 3.12: Two configurations with the same direction-relation matrix, but different MBR relations.

The common direction assessments of the direction-relation matrix and the MBR relations are the 36 convex configurations of the direction-relation matrix (Figure 3.6). The additional details captured by the remaining 133 MBR relations result from a refinement of the 36 configurations, distinguishing whether the boundaries of the MBRs are exactly aligned or not. On the other hand, the additional 182 direction-matrix relations provide refinements for non-convex shapes. A more detailed direction-relation matrix that captures the separations between the tiles in addition to the nine direction tiles would include all MBR cases. Such a 5×5 direction-relation matrix may be useful—and sometimes even necessary—for a small number of cases; however, most inferences will be sufficiently precise with the smaller and therefore, cognitively less straining 3×3 representation.

3.6 Summary

This chapter presented a new model for representing cardinal directions between extended spatial objects using direction-relation matrices. This model overcomes limitations of models that use approximations such as centroids and MBRs, as it captures directions more precisely if either of the objects is concave in shape. A particular feature of the direction-relation matrix is its ability to describe direction relations at multiple levels of detail. At a coarse level, the direction-relation matrix records into which partitions around the reference object the target object falls. At a finer level, it captures

how much of the target object falls into each partition; an even more detailed view is given if the direction-relation matrix records properties of the components in each partition. This multi-resolution model has significant implications for spatial query processing when a user, for instance, sketches the objects of interest. The coarse direction-relation may then act as a filter to quickly retrieve candidates, whereas the more detailed direction-relation matrices can be used to prioritize the candidates (Egenhofer 1997). The converse of a direction-relation cannot always be determined uniquely from a matrix; therefore, directions from A to B and from B to A are recorded explicitly. The comparison between the coarse direction-relation matrices and the MBR directions reveals that coarse direction-relation matrices capture the influence of objects' shapes on directions better than MBR relations.

The direction model presented in this chapter ignores boundaries between direction partitions, as the model focuses on the directions between regions. However, these boundaries should be taken into account while recording directions for lines and points. Chapter 4 presents a method to capture information about intersections between boundaries and the target object in direction-relation matrices, thereby enabling a unified method for direction relations between point, line, and region objects.

Chapter 4

Deep Direction Model for Point, Line, and Region Objects

The coarse direction model presented in the previous chapter is designed for region objects. This chapter extends the coarse direction model to the deep direction model, which applies to arbitrary pairs of point, line, and region objects. This extension enables the use of cardinal direction in a cognitively plausible way between different internal representations of the spatial objects. This extended model allows users to formulate queries such as “Find all towns in Maine that are *north* of Bangor” without pondering about the internal representations of the towns in the database. Such a model will facilitate queries based on the cardinal directions in multi-representation geographic databases (Buttenfield 1989; Puppo and Dettori 1995; Tryfona and Egenhofer 1996; Bertolotto 1998).

Section 4.1 discusses limitations of coarse direction-relation matrices when applied to point and line objects. Section 4.2 introduces the deep direction-relation matrix, which records the information about intersections of a target object with direction tiles and boundaries between tiles in a 3x3 matrix. Section 4.3 discusses the behavior of the deep direction-relation matrix for various types of reference objects. Section 4.4 describes consistency constraints imposed by the type of target object on the deep direction-relation matrix. Section 4.5 studies the compatibility of the directions recorded at multiple scales. Section 4.6 demonstrates that the deep direction-relation matrix records cognitively plausible values of directions, and discusses advantages of the deep direction-relation matrix over existing models. Section 4.7 summarizes the results of this chapter.

4.1 Applying the Coarse Direction-Relation Matrix to Lines and Points

The coarse direction-relation matrix (Section 3.1) successfully captures directions between two region objects, but lacks the expressive power to capture the information about line and point target objects in certain configurations. For example, coarse direction-relation matrices would not be able to record directions for the configurations in Figures 4.1a and 4.1b, because a coarse direction-relation matrix has no boundary elements and, therefore, all nine partition intersections with *B* would be empty. In Figure 4.1c, the partitions *west*, *same*, and *east* are lines, not regions; therefore, the partitions of space are different from the partitions used by the coarse direction-relation matrix.

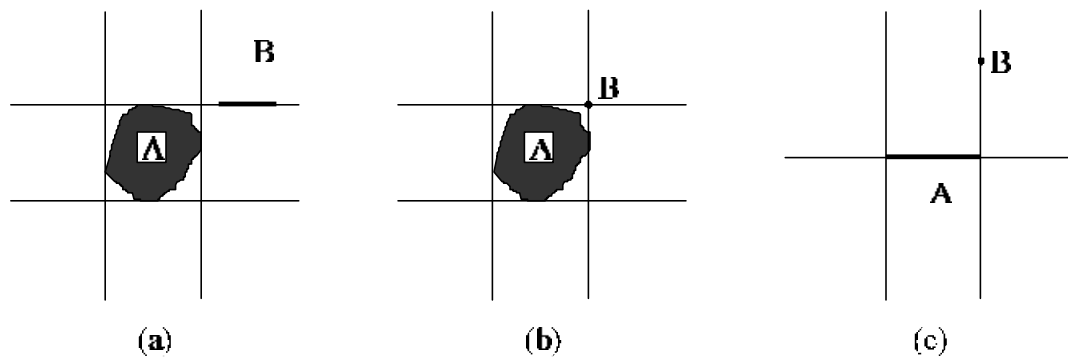


Figure 4.1: Configurations with line and point objects: (a) the reference object is a region and the target object is a line, (b) a region and a point object, and (c) a line and a point object.

Since the representation of a spatial object may change across different scales, it is necessary for a direction-relation model to be applicable and compatible across multiple representations of objects. In order to develop a deep direction model, two issues are considered: (1) the influence of the reference object on the construction of the reference grid and (2) the variations that arise due to different types of target objects.

4.1.1 Reference Object Considerations

The reference grid is based on the orientation and the extent of the reference object. The cardinal direction axes determine the orientation of the grid, and the extent of the

reference object determines the extent of partitions in the reference grid. At different scales, the reference object may be a polygon, a line, or a point. Since the direction-relation matrix forms the central tile around the reference object, the choice of a polygon, a line, or a point influences the construction of the reference grid. The projection of an object onto a grid axis results either in a line or a point. For a reference grid with a vertical and a horizontal axis, there are four possible combinations: (1) both projections are points (Figure 4.2a), (2) the projection onto the horizontal axis is a point and the projection onto the vertical axis is a line (Figure 4.2b), (3) the projection onto the horizontal axis is a line and the projection onto the vertical axis is a point (Figure 4.2c), and (4) both projections are lines (Figure 4.2d). Four pairs of projections yield four types of reference frames. The first three cases describe unique configurations when the object is a point, a horizontal line, and a vertical line, respectively. The last case occurs when the object is a region or a line that is neither strictly horizontal nor vertical.

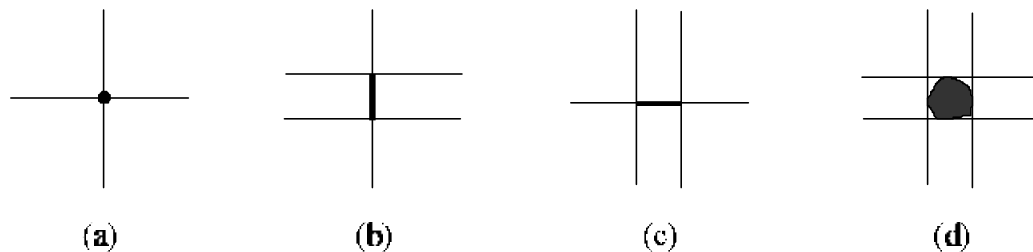


Figure 4.2: Different reference frames based on different types of objects: (a) for a point, (b) for a vertical line, (c) for a horizontal line, and (d) for other lines and all regions.

4.1.2 Target Object Considerations

In a similar way, the type of the target objects may influence the direction relation. While a target region must extend through at least one tile of the grid around the reference object, a line may fall “between the cracks,” that is, it may be located along the border between two neighboring tiles if the line is strictly horizontal or vertical and exactly aligned with the reference grid. For a point, additional alternatives exist as it may fall not only on the border between two neighboring tiles, but also coincide with the border of four tiles. A reference grid for the region reference that supports all cases would require

nine direction partitions (Figure 4.3a) with sixteen *boundary parts* for these nine partitions. Twelve of these boundary parts are lines (Figure 4.3b), and four parts are points (Figure 4.3c). The lines between the partitions are the north-northwest line (N-NwL), the north-northeast line (N-NeL), the east-northeast line (E-NeL), the east-southeast line (E-SeL), the south-southeast line (S-SeL), the south-southwest line (S-SwL), the west-southwest line (W-SwL), the west-northwest line (W-NwL), the north line (NL), the east line (EL), the south line (SL), and the west line (WL). The boundary points are the northwest point (NWP), the northeast point (NEP), the southeast point (SEP), and the southwest point (SWP). The coarse direction-relation matrix, however, captures only the intersections with nine direction partitions.

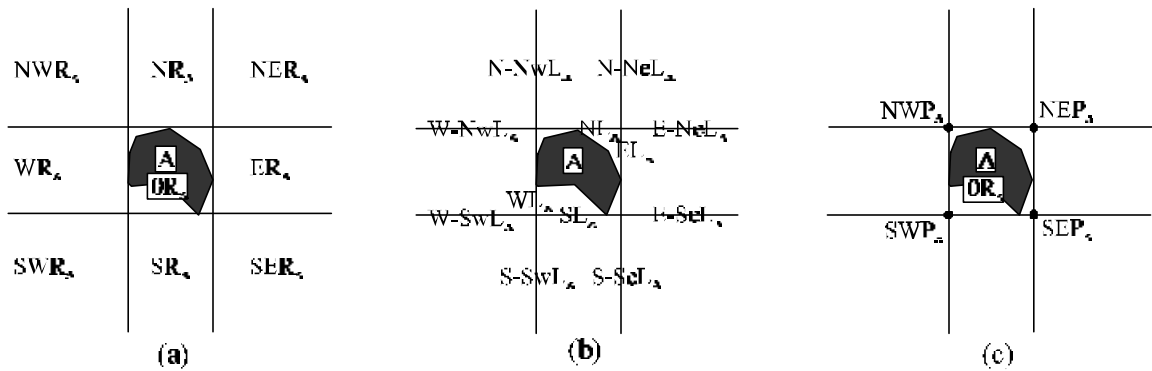


Figure 4.3: A region reference A in a 2-dimensional embedding space: (a) nine direction partitions, (b) twelve boundary lines between direction partitions, and (c) four boundary points.

An extension of the 3x3 direction-relation matrix would have to account for the nine partitions and all sixteen boundary parts. Such a 5x5 direction-relation matrix would have an element for each partition and each boundary part, and record the information about the intersection of the target object with all the 25 parts of the space. While it would address each possible part at which a point could be located with respect to a reference grid, it would increase the spatial resolution by 178% (from 9 to 25) for *all* types of spatial objects. People typically use the primary and secondary directions to communicate the knowledge about relative orientation of objects; therefore, the use of a 25-element matrix would be cognitively overwhelming, offering more elements than what people

handle easily (Miller 1956). In order to perform direction reasoning in a cognitively plausible way, it is desirable to keep the number of elements in a direction-relation matrix small.

A smaller number of elements also reduces the computational complexity for assessing similarity between cardinal directions (Chapter 5). Each of the additional sixteen elements would capture a very small part of the space, compared to the significant areas covered by the nine direction partitions. From a computational perspective, the higher resolution of the reference grid would introduce additional values that are often implied. For example, if a target polygon is northeast of a reference polygon and the coarse direction-relation matrix has non-empty values for the northeast and the north tiles only, it implies that the intersection of the target object with the line between the two tiles must be non-empty as well.

4.2 Deep Direction-Relation Matrix

Rather than increasing the spatial resolution, we increase the resolution of the values that each element in the 3x3-direction-relation matrix may have. In addition to the empty and non-empty values for the intersection of the target object with the direction partitions, we capture *for each empty direction partition* when the target object intersects with its boundaries. This information is encoded in the *neighbor code* of a direction. Such a direction-relation matrix is called a *deep direction-relation matrix*.

A neighbor code records information about intersections with the direction partition and the neighboring boundary parts using nine bits (x_0 - x_8) (Figure 4.4). Bit 0 (x_0) records the value of the intersection with the direction partition (DP); and bits 1-8 (x_1 - x_8) record the values of intersections with the left (L), bottom-left (BL), bottom (B), bottom-right (BR), right (R), top-right (TR), top (T), and top-left (TL) boundary parts, respectively.

x_8	x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0
TL	T	TR	R	BR	B	BL	L	DP
256	128	64	32	16	8	4	2	1

Figure 4.4: The nine-bit field for a neighbor code in the deep direction-relation matrix.

If the intersection of the target object with the direction partition (DP_A) is empty (\emptyset), we record a value 0 for x_0 ; if the intersection is non-empty ($\neg\emptyset$) we record a value 1 (Equation 4.1a). We denote the boundary of a direction partition that corresponds to bit i in the element by DP_A^i . For example, in the case of a region reference the direction partition for the northwest element is NWR_A (Figure 4.3a) and the boundary corresponding to bit 5 (i.e., DP_A^5) is $N-NwL_A$, that is, the right side boundary of NWR_A (Figure 4.3b). For $i=1$ to 8, the value of bit i is zero, if the target object intersects with the direction partition; it is 1 if the target object does not intersect with the direction partition, but intersects with the boundary corresponding to x_i (Equation 4.1b).

$$x_0 := \begin{cases} 0 & \text{if } (DP_A \cap B = \emptyset) \\ 1 & \text{if } (DP_A \cap B = \neg\emptyset) \end{cases} \quad (4.1a)$$

$$\text{for } i=1 \text{ to } 8: \quad x_i := \begin{cases} 0 & \text{if } (x_0 = 1) \\ 1 & \text{if } (x_0 = 0) \wedge (DP_A^i \cap B = \neg\emptyset) \end{cases} \quad (4.1b)$$

The value (X) of an element in a deep direction-relation matrix is the weighted sum of the neighbor code (Equation 4.2); the weighting by the powers of 2 enables a unique decoding of the neighbor code. The value of an element lies in the closed interval $[0, 510]$ depending upon the target object's intersection with the direction partition it represents and the boundaries of the partition. The value 0 for an element indicates that the target object neither intersects with the direction partition nor with the boundaries. The value 1 implies that the target object intersects with the respective direction partition, which is the only possible odd value for an element. All other possible values for an element are even, because a neighbor code records bits (x_1-x_8) as 0, when the intersection of the target object with the direction partition is non-empty.

$$X := \sum_{i=0}^8 2^i x_i \quad (4.2)$$

The deep direction-relation matrix for the direction of object B with respect to object A records nine neighbor codes corresponding to nine cardinal directions: north (X_N), northeast (X_{NE}), east (X_E), southeast (X_{SE}), south (X_S), southwest (X_{SW}), west (X_W), northwest (X_{NW}), and same (X_0) in the same topological organization as the coarse direction-relation matrix (Equation 4.3).

$$dir(A, B) := \begin{bmatrix} X_{NW} & X_N & X_{NE} \\ X_W & X_0 & X_E \\ X_{SW} & X_S & X_{SE} \end{bmatrix} \quad (4.3)$$

4.3 Deep Direction-Relation Matrices for Various Types of Reference Objects

A bit in the neighbor code's eight bits corresponding to boundaries of direction partition (x_1 – x_8) is recorded as 1 if (1) the boundary corresponding to the bit exists, (2) the target object's intersection with the respective partition is empty, and (3) the target object's intersection with the boundary is non-empty. If any one of these conditions is not met, the bit is marked zero. This section focuses on the first condition. We study the pattern of zero bits for the neighbor code due to the non-existence of certain boundaries for direction partitions, which occurs due to the type of a reference object and the location of a direction partition in the space.

The type of the reference object determines the types of partitions for *north*, *east*, *south*, *west*, and *same* directions. For example, the north partition for a region reference (Figure 4.2a) is a region, whereas for a point reference (Figure 4.2d) the north partition is a line. Therefore, a region reference has a boundary between its north and northwest partitions, whereas a point reference has no boundary between its north and northwest partitions. The location of a direction partition can also force some bits in the neighbor code for an element to zero. For example, the northwest partition has no boundary in its

top-left; therefore, bit 8 in the northwest element of the deep direction-relation matrix would always be 0. Sections 4.3.1-4 discuss neighbor code patterns for four types of reference object.

4.3.1 A Region Reference Object

A region object has a reference grid with nine direction partitions and sixteen boundaries between these partitions. The partitions northwest, northeast, southeast, and southwest have three boundaries each with other direction partitions (Figure 4.5a). For instance, the northwest region's boundaries are the west-northwest line, the northwest point, and the north-northwest line. Information about their intersections is recorded in bits 3–5 of the northwest element, respectively (Figure 4.5b). The remaining five bits (bits 1–2 and 6–8) in the neighbor code of the northwest element are always zero. If the intersection of the target object with the northwest region is non-empty, the neighbor code would record 0 for bits 1–8, regardless of the values of the target object's intersections with the boundaries of the northwest partition (Figure 4.6b). The value of the northwest element is 1 for this case (Table 4.1). If the intersection with the northwest region is empty, there are $2^3=8$ possibilities (Table 4.1; Figures 4.6a and 4.6c–i). For example, the value of the northwest element is 48 if the right and bottom-right boundaries intersect with the target object (Figure 4.6h).

The partitions north, east, south, and west have five boundaries each. The eight boundaries of the same partition are the west line, the southwest point, the south line, the southeast point, the east line, the northeast point, the north line, and the northwest point. The information about the intersections is recorded in bits 1–8 of the *same* (0) element. The maximum value (i.e., 510) is assumed only by the same element.

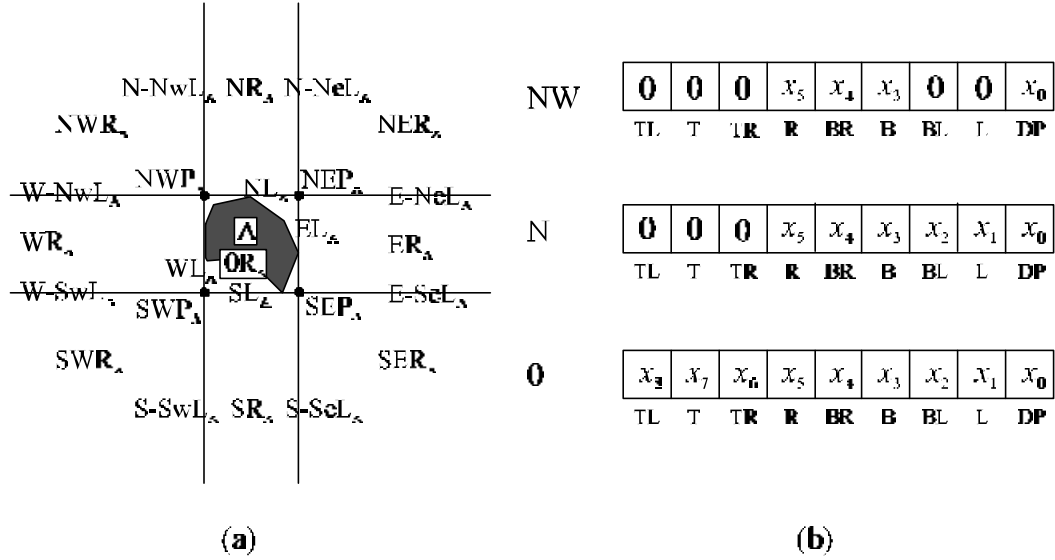


Figure 4.5: (a) The reference grid for a region object and (b) the patterns of neighbor codes for the northwest, north and same elements. A zero value for a bit in an element implies that the value of that bit is always zero.

Possible values for				Element value (X_{NW}) = $32x_5+16x_4+8x_3+x_0$	Illustration
x_5	x_4	x_3	x_0		
0	0	0	0	0	Figure 4.6a
0	0	0	1	1	Figure 4.6b
0	0	1	0	8	Figure 4.6c
0	1	0	0	16	Figure 4.6d
0	1	1	0	24	Figure 4.6e
1	0	0	0	32	Figure 4.6f
1	0	1	0	40	Figure 4.6g
1	1	0	0	48	Figure 4.6h
1	1	1	0	56	Figure 4.6i

Table 4.1: All possible values of the northwest element in the deep direction-relation matrix for the region reference. The remaining five bits (1-2 and 6-8) are always zero.

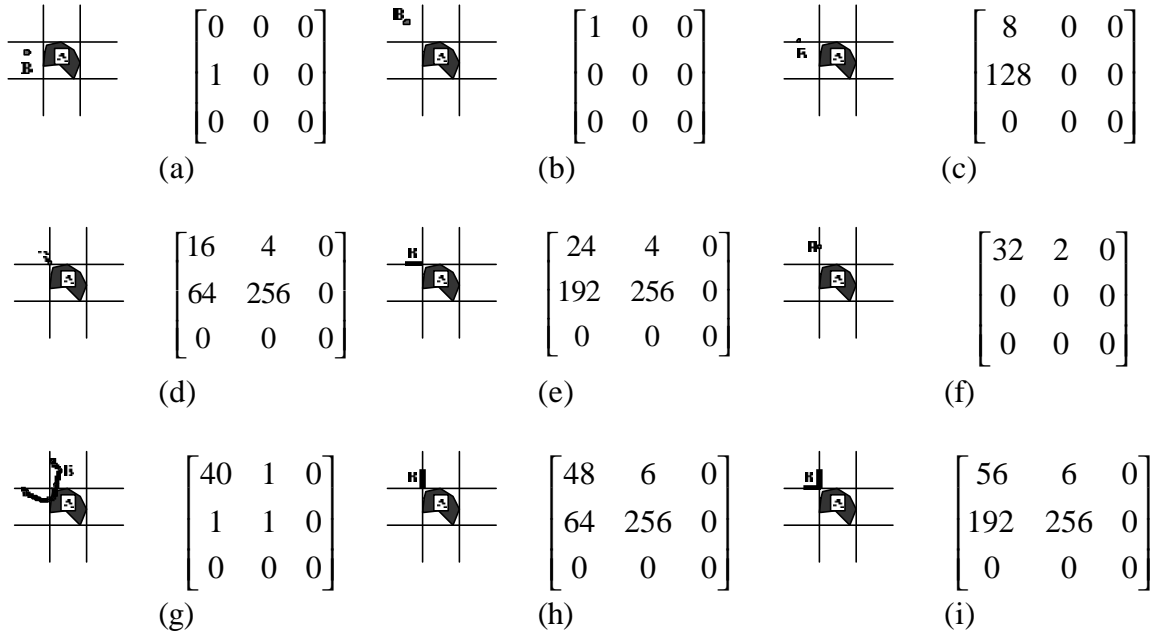


Figure 4.6: (a)-(i) Nine configurations corresponding to nine possible values of the northwest element in the deep direction-relation matrix for a region reference object.

4.3.2 A Linear Horizontal Reference Object

A linear horizontal reference object has nine direction partitions and six boundaries between these partitions. The direction partitions are north region, northeast region, east line, southeast region, south region, southwest region, west line, northwest region, and 0 line (Figure 4.7a). The boundaries of the nine partitions are the north-northwest line, the north-northeast line, the east point, the south-southeast line, the south-southwest line, and the west point.

The northwest partition has two boundaries, because the west-line at the bottom of the northwest-region is a direction partition on its own. Two boundaries for the northwest partition are the west point and the north-northwest line. The information about their intersections with the target object is recorded in bits 4–5, respectively (Figure 4.7b). The remaining six bits in the neighbor code are always zero. The north partition has four boundaries and the information about their intersections with the target object is recorded in bits 1–2 and bits 4–5. The west partition has only one boundary, and information about

the target object's intersection with the boundary is recorded in bit 5. The same partition has two boundary parts, and the information about the target object's intersections with these parts is recorded in bits 1 and 5.

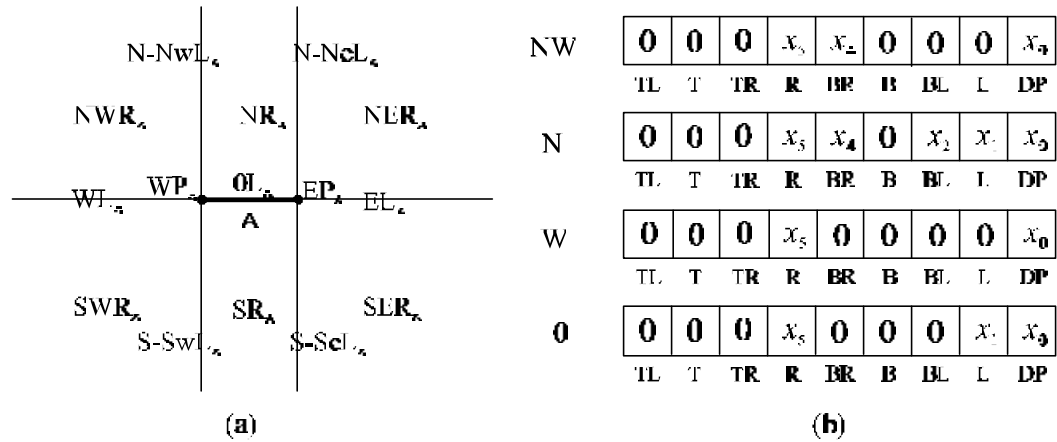


Figure 4.7: (a) The reference grid for a horizontal line and (b) the patterns of neighbor codes for the northwest, north, west, and same elements.

4.3.3 A Linear Vertical Reference Object

A vertical line, just like a horizontal line, has nine direction partitions and six boundary components (Figure 4.8a). Figure 4.8b shows four patterns of the neighbor codes.

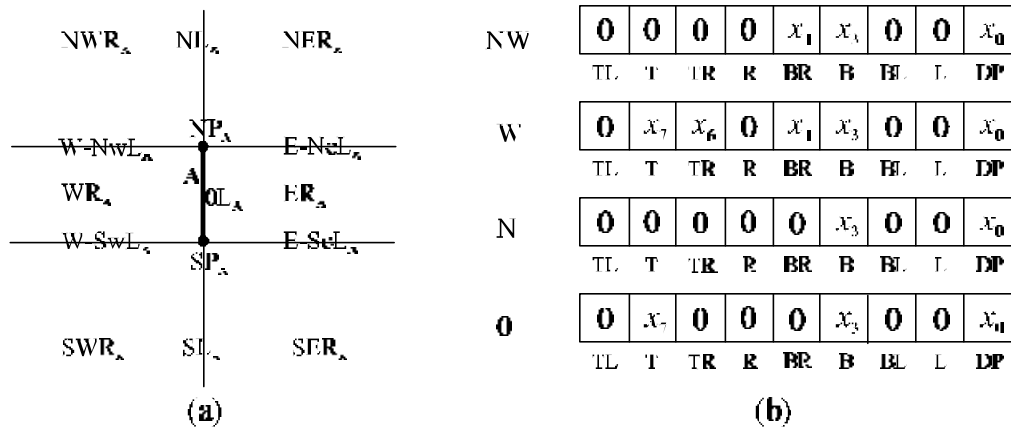


Figure 4.8: (a) The reference grid for a vertical line and (b) the patterns of neighbor codes for the northwest, west, north, and same elements.

4.3.4 A Point Reference Object

The reference frame for a point object has nine direction partitions and no boundaries between these partitions are needed (Figure 4.9a); therefore, the neighbor code is 0 for all direction partitions in Figure 4.9b.

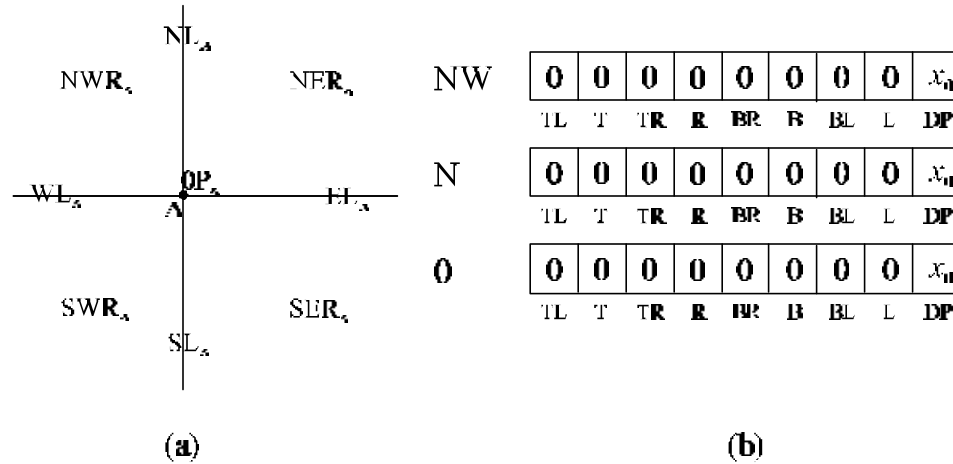


Figure 4.9: (a) The reference grid for a point and (b) the patterns of neighbor codes for the northwest, north, and same element.

4.4 Consistency Constraints Due to the Type of the Target Object

To analyze the consistency constraint for the target object, we distinguish three types of objects: regions (Section 4.4.1), lines (Section 4.4.2), and points (Section 4.4.3). Before describing these constraints, we need to define the terms used in explaining the consistency constraints.

Definition 4.1: A *non-zero element* in a deep direction-relation matrix is an element that has at least one of the nine bits (0-8) with a value 1.

Definition 4.2: An element with a *non-empty neighbor code* in a deep direction-relation matrix is an element that has at least one bit with a value 1 for the neighbor code in bits 1-8.

For example, the deep direction-relation matrix in Figure 4.6c has non-empty neighbor codes for the northwest and west elements. An element with a *non-empty neighbor code* is a *non-zero* element, but the reverse is not always true. For example, the deep direction-relation matrix in Figure 4.6b has a non-zero value for the northwest element, but its neighbor code is empty.

Definition 4.3: A *row-tuple* in a deep direction-relation matrix is a set of two adjacent elements with non-empty neighbor codes in a row, such that the left element has only its right bit with value 1 and the right element has only its left bit with value 1.

The value of the left element in a row-tuple is 2, while the value of the right element is 32. For example, the deep direction-relation matrix in Figure 4.6f has a row-tuple formed by the northwest and north elements.

Definition 4.4: A *column-tuple* in a deep direction-relation matrix is a set of two adjacent elements with non-empty neighbor codes in a column, such that the top element has only its bottom bit with value 1 and the bottom element has only its top with value 1.

Definition 4.5: A *quadruple* in a deep direction-relation matrix is a set of four elements with non-empty neighbor codes arranged in a rectangular fashion, such that only the bottom-right, bottom-left, top-left, and top-right bits have a value 1 for the top-left, top-right, bottom-right, and bottom-left elements, respectively.

A point target on a corner of the same direction partition for a region reference gives a deep direction-relation matrix with a quadruple (Figure 4.6d).

Definition 4.6: A *row-sextuple* in a deep direction-relation matrix is a set of six elements with non-empty neighbor codes arranged in two adjacent rows, such that (1) only the bottom-right, bottom, and bottom-left bits have a value 1 for the left, central, and right elements in the top row and (2) only the top-right, top, and top-left bits have a value 1 for the left, central, and right elements in the bottom row, respectively.

Definition 4.7: A *column-sextuple* in a deep direction-relation matrix is a set of six elements with non-empty neighbor codes arranged in two adjacent columns, such that (1) only the bottom-right, right, and top-right bits have a value 1 for the top, middle, and bottom elements in the left column and (2) only the bottom-left, left, and top-left bits have a value 1 for the top, middle, and bottom elements in the right column, respectively.

For example, a point target that coincides with the north point of a vertical line reference yields a deep direction-relation matrix with a row-sextuple (Figure 4.10a), whereas a point target that coincides with the west point of a horizontal line reference yields a deep direction-relation matrix with a column-sextuple (Figure 4.10b).

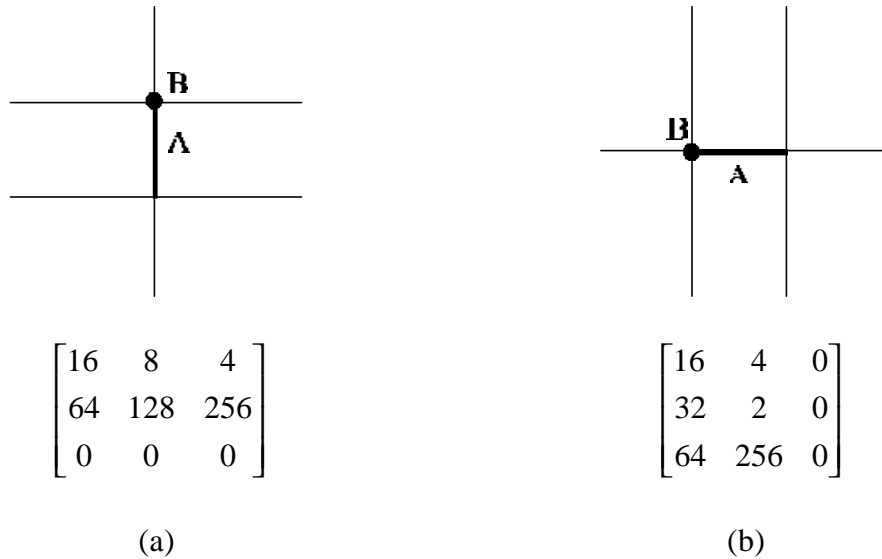


Figure 4.10: The direction of a point in specific situations with respect to (a) a vertical line having a row-sextuple and (b) a horizontal line having a column-sextuple.

4.4.1 A Region Target Object

A region object imposes the constraint of 4-connectedness on the deep direction-relation matrix, similar to the 4-consistency constraint for the coarse direction-relation matrix (Section 3.2.1). Additionally, neighbor codes in the deep direction-relation matrix must also follow constraints imposed by connected regions. A 4-consistent deep direction-relation matrix satisfies all of the following three constraints:

- at least one element of the matrix must have a value 1 (Figure 4.11a),
- all elements that have the value 1 must be 4-connected (Figure 4.11b), and
- all elements with non-empty neighbor codes have at least one 4-neighbor or 8-neighbor with a value 1 (Figure 4.11c).

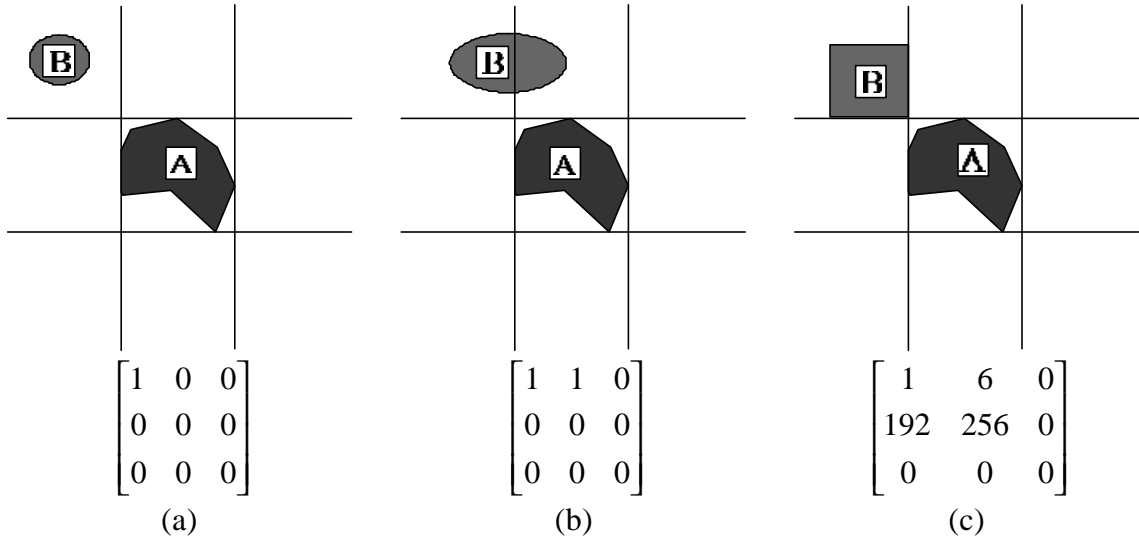


Figure 4.11: Deep direction-relation matrices that are consistent for region targets with respect to region references.

The first two constraints follow from constraints of the coarse direction-relation matrix (Section 3.2.1), while the third constraint is specific to deep direction-relation matrices. For example, the deep direction-relation matrix in Equation 4.4 satisfies the first two constraints, but does not satisfy the third constraint. This matrix refers to a target object that only intersects with the northwest-region (NWR) and the south-southwest line (S-SwL). It is impossible for a connected region to intersect with NWR and S-SwL without intersecting any other direction partitions; therefore, this deep direction-relation matrix is inconsistent for a region target.

$$dir(A, B) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 32 & 2 \end{bmatrix} \quad (4.4)$$

The three constraints apply to the deep direction-relation matrix recorded for a target object with respect to any of the four types of reference objects (Figure 4.2). For horizontal line, vertical line, and point reference objects, there is an additional constraint in each case.

- A horizontal line reference has three non-region partitions—the west line, the same point, and the east line. If a target object intersects with non-region direction partitions and does not intersect with any region partition, it cannot be a region. Therefore, in the case of a horizontal line reference, at least one element corresponding to its six region partitions—northwest, north, northeast, southeast, south, and southwest—must have the value 1.
- Similarly for a vertical line reference, at least one element corresponding to its six region partitions—northwest, northeast, west, east, southwest, and southeast—must have the value 1.
- For a point reference object, at least one element corresponding to its four region partitions—northwest, northeast, southeast, and southwest—must have the value 1.

4.4.2 A Line Target Object

In addition to 4-connected configurations, a line target generates 8-connected configurations. A line can fall completely on the boundaries of direction partitions, except the point boundaries. These two factors impose the following two constraints on the deep direction-relation matrix:

- all non-zero elements must be eight-connected (Figures 4.12a-c), and
- to exclude the possibility of a point fulfilling the eight-connectivity constraint, an additional constraint according to the type of the reference object must be fulfilled.
- if the reference object is a region, the matrix must not contain a quadruple.

- if the reference object is a vertical line, the matrix must not contain a row-sextuple.
- if the reference object is a horizontal line, the matrix must not contain a column-sextuple.
- if the reference object is a point and the value of the same element is 1, at least another element must be 1.

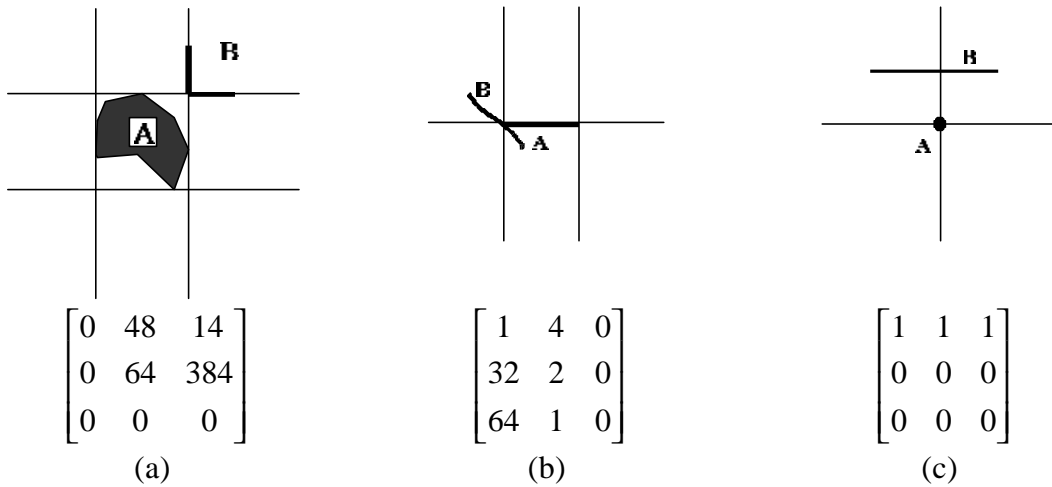


Figure 4.12: Deep direction-relation matrices that are consistent for line targets with respect to (a) a region, (b) a horizontal line, and (c) a point.

4.4.3 A Point Target Object

A deep direction-relation matrix is consistent for a point target object if it satisfies any and only one of the following six conditions:

- it has only one element with value 1 (Figure 4.13a),
- it has only one row-tuple (Figure 4.13b),
- it has only one column-tuple (Figure 4.13c),
- it has only one quadruple (Figure 4.13d),
- it has only one row-sextuple (Figure 4.10a), or
- it has only one column-sextuple (Figure 4.10b).

Satisfying more than one constraint would imply that the target object is not a point. For example, if a deep direction-relation matrix has more than one element with value 1, it would mean that the target object is either a line (Figure 4.12c) or a region (Figure 4.11b).

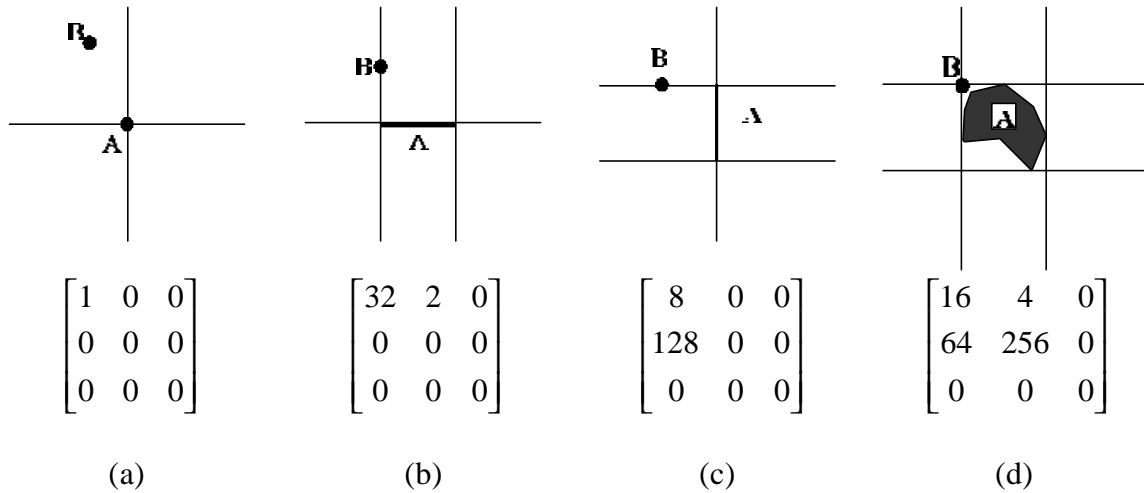


Figure 4.13: Deep direction-relation matrices that are consistent for point targets with respect to (a) a point, (b) a horizontal line, (c) a vertical line, and (d) a region.

4.5 Directions at Multiple Scales

Based on the expressive power and flexibility of the deep direction-relation matrix, a more complex problem can be addressed: the consistent modeling of direction relations across multiple representations. The use of multiple representations of spatial objects is an important issue in GISs, because they typically encode the geometry of spatial objects in terms of points, lines, and polygons. The choice of the encoding, however, is not necessarily unique and often times the same geographic object may be represented as a polygon or a point, or as a polygon or a line. The decision about a particular representation depends on the level of detail, often referred to as the scale of a data set (Goodchild and Proctor 1997). For example, a town may be a dot on a national map, but a polygon at a more detailed level. When using spatial data in a GIS, users typically have to know about the encoding of the data in order to apply appropriate operations. For instance, while it makes sense to calculate the area of a polygon, the same operation does

not apply if the object's geometry is represented as a line or a point. In this section, we study the directions recorded by deep direction-relation matrices at multiple scales.

We consider the directions of B with respect to A in a scene at the initial scale (S_0) and the scale after zooming-out (S_1). The zooming out operation in this chapter is a graphical zoom (Frank and Timpf 1994), which corresponds to a scale reduction in cartographic generalization (McMaster and Shea 1992). Zooming out is equivalent to observing the same scene from a more remote distance. A scale reduction can reduce the dimension of an object in a scene. "Reduction of the dimension of an object due to a scale reduction" is referred as the *collapse* operation in cartographic generalization (McMaster and Shea 1992). A collapse operation may change a region into a line or a point, or a line into a point. In the case of a graphical zoom, after zooming-out an object remains in the scaled range of the original object, which is not always the case for cartographic generalization. In this study, we consider only geometric aspects of the collapse operation; therefore, we assume that after a scale reduction, the object at a reduced scale remains within the scaled range of the original object.

We call a scale reduction that reduces the dimension of an object a *significant scale reduction*. This study considers significant scale reductions only, because only such changes make qualitative differences in a reference target pair of a direction relation. The effects of significant scale reductions are studied for the x -axis projections of the objects; the results apply to the y -axis projections correspondingly. There are five cases of significant scale reductions on the x -axis, where $\Pi_x^0(A)$ denotes the projection of object A on the x -axis at scale S_0 (Table 4.2). While generating the names of the cases, "I" denotes an interval, and "P" denotes a point. For example, the case IIPP (Table 4.2) refers to a scale reduction that collapses a pair of intervals (II) into a pair of points (PP).

Case	Before significant scale reduction		After reduction	
	$\Pi_x^0(A)$	$\Pi_x^0(B)$	$\Pi_x^1(A)$	$\Pi_x^1(B)$
IIPP	interval	interval	point	point
IPII	interval	interval	point	interval
IIIP	interval	interval	interval	point
IPPP	interval	point	point	point
PIPP	point	interval	point	point

Table 4.2: Significant scale reductions in the types of projections of reference target pair on the x -axis.

In order to study the compatibility of directions at multiple scales, Section 4.5.1 defines the term *compatible* for deep direction-relation matrices. We use the projections of the objects on the x -axis to study the deep direction-relation matrix at multiple scales. Section 4.5.2 discusses the relations between different types of projections of objects. Sections 4.5.3–7 analyze the five cases of significant scale reductions (Table 4.2). Section 4.5.8 summarizes the results of this study.

4.5.1 Compatibility for Deep Direction-Relation Matrices

We denote the deep direction-relation matrices for the directions of object B with respect to object A at scales S_0 and S_1 by D^0 and D^1 , respectively.

Definition 4.8: The direction D^1 is *compatible* with the direction D^0 if for each non-zero element in D^1 the corresponding elements in D^0 are non-zero (Equation 4.5).

$$compatible(D^0, D^1) := \forall_{i,j} : D_{i,j}^1 \neq 0 \Rightarrow D_{i,j}^0 \neq 0 \quad (4.5)$$

The number of non-zero elements in D^0 is either equal to or more than the number of non-zero elements in D^1 . For example, if the target object B intersects with the northwest and north partitions (Figure 4.14a) at scale S_0 , then at scale S_1 the target object can intersect with the northwest partition (Figure 4.14b), the north partition (Figure 4.14c), and the north and northwest partitions (Figure 4.14d). The D^1 matrices recorded in Figures 4.14b-d are *compatible* with D^0 in Figure 4.14a.

The binary relation *compatible* is *reflexive*, because D^0 is *compatible* with itself. If D^1 is *compatible* with D^0 and D^2 is *compatible* with D^1 , D^2 is *compatible* with D^0 ; therefore, the relation *compatible* is *transitive*. It is also *antisymmetric*, because D^1 *compatible* with D^0 and D^0 *compatible* with D^1 implies D^0 equals D^1 . The relation *compatible* is *reflexive*, *transitive*, and *antisymmetric*; therefore, it is a *partial order* relation.

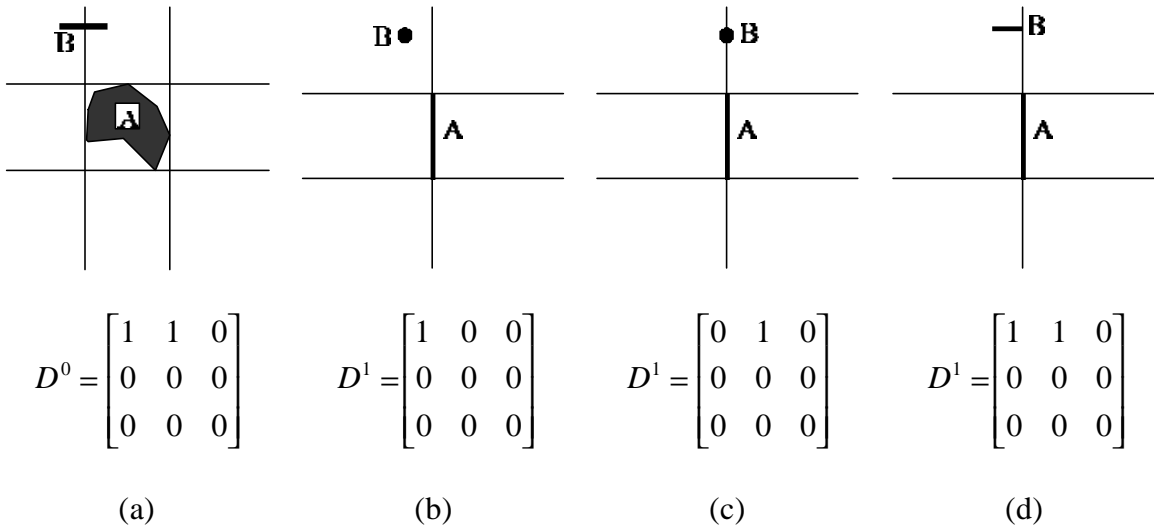


Figure 4.14: Objects A and B with their deep direction-relation matrices at (a) scale S^0 and (b)-(d) scale S^1 .

4.5.2 Relations between Projections of Objects

An object's projection onto an axis is either a point or an interval. The projection of an ordered pair of a reference object and a target object onto an axis can be (1) a pair of points, (2) an interval and a point, (3) a point and an interval, and (4) a pair of intervals. A point in a 1-dimensional space (along an axis) can be *before* ($<$), *equal to* ($=$), or *after* ($>$) another point (Table 4.3), where x_A is the location of object A on the x -axis.

Relation	Condition
<i>before</i> (<)	$x_B < x_A$
<i>equal</i> (=)	$x_B = x_A$
<i>after</i> (>)	$x_B > x_A$

Table 4.3: The conditions for the relations between two points on the x -axis; A is the reference point and B is the target point.

A point can be *before* (<), *at-start of* (*as*), *during* (*du*), *at-finish of* (*af*), or *after* (>) an interval on an axis (Figure 4.15a). The relations in Figure 4.15a correspond to the relations *posterior* (-), *back* (*b*), *interior* (*i*), *front* (*f*), and *ahead* (+) (Mukerjee and Joe 1990). We denote the left and right extremes of the interval A by x_{A-} and x_{A+} , respectively. The relations in Figure 4.15a can be expressed using the conditions between x_B and the extremes x_{A-} and x_{A+} (Table 4.4). An interval can have the following relations with respect to a point on an axis: *before* (<), *finishes-at* (*fa*), *contains* (*co*), *starts-at* (*sa*), and *after* (>) (Figure 4.15b; Table 4.5).

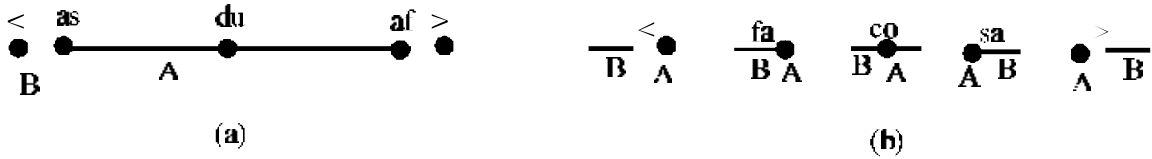


Figure 4.15: Relations in 1-D space for (a) point B with respect to interval A and (b) interval B with respect to point A .

Relation	Conditions
<i>before</i> (<)	$x_B < x_{A-}$
<i>at-start</i> (<i>as</i>)	$x_B = x_{A-}$
<i>during</i> (<i>du</i>)	$(x_B > x_{A-}) \wedge (x_B < x_{A+})$
<i>at-finish</i> (<i>af</i>)	$x_B = x_{A+}$
<i>after</i> (>)	$x_B > x_{A+}$

Table 4.4: Conditions for relations of point B with respect to an interval A .

Relation	Conditions
<i>before</i> (<)	$x_{B+} < x_A$
<i>finishes-at</i> (<i>fa</i>)	$x_{B+} = x_A$
<i>contains</i> (<i>co</i>)	$(x_{B-} < x_A) \wedge (x_{B+} > x_A)$
<i>starts-at</i> (<i>sa</i>)	$x_{B-} = x_A$
<i>after</i> (>)	$x_{B-} > x_A$

Table 4.5: Conditions for relations of interval *B* with respect to a point *A*.

An interval can have one of thirteen relations (Allen 1983) with respect to another interval (Figure 2.8). Figure 2.8 uses the symbol *B* for the reference interval and the symbol *A* for the target interval, whereas Figure 4.16 uses symbol *A* for the reference interval and symbol *B* for the target interval, because the direction-relation matrix uses symbols *A* and *B* for the reference and target objects, respectively. Each interval relation can also be described using conditions between the extremes of the intervals (Table 4.6) (Freksa 1992a).

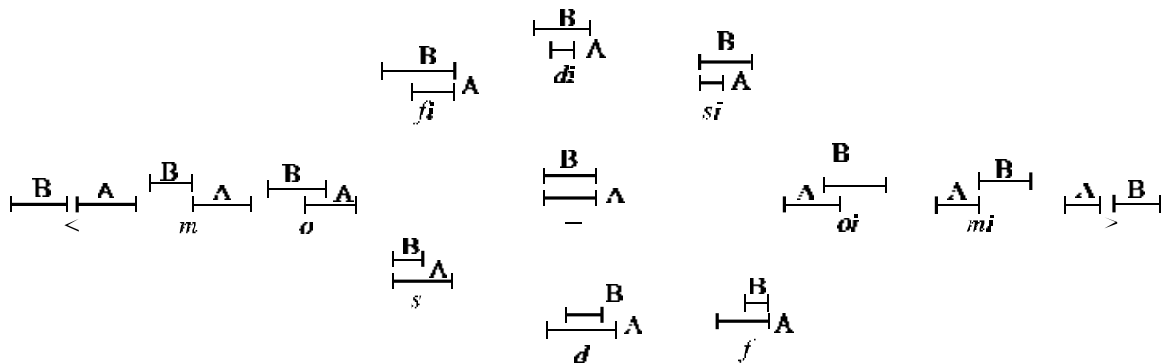


Figure 4.16: One-dimension interval relations, *A* is the reference interval and *B* is the target interval.

Relation	Conditions
<i>before</i> (<)	$x_{B+} < x_{A-}$
<i>meets</i> (m)	$x_{B+} = x_{A-}$
<i>overlaps</i> (o)	$(x_{B-} < x_{A-}) \wedge (x_{B+} > x_{A-}) \wedge (x_{B+} < x_{A+})$
<i>finishes-inverse</i> (fi)	$(x_{B-} < x_{A-}) \wedge (x_{B+} = x_{A+})$
<i>starts</i> (s)	$(x_{B-} = x_{A-}) \wedge (x_{B+} < x_{A+})$
<i>during</i> (d)	$(x_{B-} > x_{A-}) \wedge (x_{B+} < x_{A+})$
<i>equal</i> (=)	$(x_{B-} = x_{A-}) \wedge (x_{B+} = x_{A+})$
<i>during-inverse</i> (di)	$(x_{B-} < x_{A-}) \wedge (x_{B+} > x_{A+})$
<i>starts-inverse</i> (si)	$(x_{B-} = x_{A-}) \wedge (x_{B+} > x_{A+})$
<i>finishes</i> (f)	$(x_{B-} > x_{A-}) \wedge (x_{B+} = x_{A+})$
<i>overlap-inverse</i> (oi)	$(x_{B-} > x_{A-}) \wedge (x_{B-} < x_{A+}) \wedge (x_{B+} > x_{A+})$
<i>meets-inverse</i> (mi)	$x_{B-} = x_{A+}$
<i>after</i> (>)	$x_{B-} > x_{A+}$

Table 4.6: Conditions for 1-D interval relations.

To study the deep direction model at multiple scales, we use the relations between an ordered pair of points (Table 4.3), a point and an interval (Figure 4.15a), an interval and a point (Figure 4.15b), and intervals (Figure 4.16).

4.5.3 Collapsing a Pair of Intervals into a Pair of Points

In this section, we consider the case IIPP, where both the reference and target projections on the x -axis are intervals. After a scale reduction, the projection of an interval collapses into a point (Equation 4.6).

$$[x_{A-}^0, x_{A+}^0] \rightarrow x_A^1 \quad (4.6)$$

We assume that a zooming-out operation on a projection of an object on an axis yields the same result as zooming-out on a scene containing this object, followed by recording of the projection on an axis. The shrinking of a region reference (Figure 4.5a) into a vertical line reference object (Figure 4.8a) is an example of a significant scale reduction that collapses the reference interval into a point. In this example, the north region with its left and right boundaries collapses into a north line (Equation 4.7).

$$\{N-NwL, NR, N-NeL\}_0 \rightarrow \{NL\}_1 \quad (4.7)$$

The point x_A^1 is the result of collapsing the interval $[x_{A-}^0, x_{A+}^0]$, therefore, the point will lie within the scaled interval $\left[\frac{x_{A-}^1}{a}, \frac{x_{A+}^1}{a}\right]$ (Equation 4.8).

$$\frac{x_{A-}^0}{a} \leq x_A^1 \leq \frac{x_{A+}^0}{a} \quad (4.8)$$

We denote distances of the point x_A^1 from the left and right extremes of the scaled interval A by non-negative variables L and M , respectively, and the range of point x_A^1 in terms of the extremes of the scaled interval A (Equations 4.9a). Similarly, we describe the range of point x_B^1 with respect to the scaled interval for B using non-negative variables N and P (Equation 4.9b). Sections 4.5.3.1–3 use the ranges of points x_A^1 and x_B^1 (Equations 4.9a-b) to analyze the scale reduction for the interval relations: *meets*, *overlaps*, and *equal*, respectively.

$$\frac{x_{A-}^0}{a} + L = x_A^1 = \frac{x_{A+}^0}{a} - M, \quad L \geq 0, M \geq 0 \quad (4.9a)$$

$$\frac{x_{B-}^0}{a} + N = x_B^1 = \frac{x_{B+}^0}{a} - P, \quad N \geq 0, P \geq 0 \quad (4.9b)$$

4.5.3.1 Intervals with relation *meets*

The condition for the interval relation *meets* is $x_{B+}^0 = x_{A-}^0$ (Table 4.6), which holds for scaled intervals as well (Equation 4.10). To determine new relations for point x_B^1 with respect to the point x_A^1 , we combine the condition of *meets* relation (Equation 4.10) with the ranges of the points in the scaled intervals (Equations 4.9a-b), which gives the constraint between the points at scale S_1 (Equation 4.11). This constraint is rearranged as difference between two points (Equation 4.12). If both L and P are zero, $x_B^1 = x_A^1$. If $L > 0$ or $P > 0$, $x_B^1 < x_A^1$. Thus, a relation *meets* between an ordered pair of intervals can

transform to either “<” or “=”). The relation between an ordered pair of points is a set {<, =} (Table 4.7).

$$\frac{x_{B+}^0}{a} = \frac{x_{A-}^0}{a} \quad (4.10)$$

$$x_B^1 + P = x_A^1 - L \quad (4.11)$$

$$x_B^1 - x_A^1 = -(L + P) \quad (4.12)$$

$R_x^0(A_i, B_i)$	$R_x^1(A_p, B_p)$
<	<
<i>m</i>	{<, =}
<i>o</i>	{<, =, >}
<i>fi</i>	{<, =, >}
<i>s</i>	{<, =, >}
<i>d</i>	{<, =, >}
=	{<, =, >}
<i>di</i>	{<, =, >}
<i>si</i>	{<, =, >}
<i>f</i>	{<, =, >}
<i>oi</i>	{<, =, >}
<i>mi</i>	{=, >}
>	>

Table 4.7: An IIPP scale reduction maps relations between intervals onto the relations between points.

The reduction in the dimension of the objects’ projections is only due to a scaling operation; therefore, the order between projections along an axis does not change. We denote the relation of interval B with respect to interval A at scale S_0 on the x -axis by $R_x^0(A_i, B_i)$. If the value of $R_x^0(A_i, B_i)$ is {<}, that is, B west of A , after a case IIPP reduction, the value of new relation $R_x^1(A_p, B_p)$ is {<}, which is also B west of A . If $R_x^0(A_i, B_i) = \{m\}$ it means B is west of A and runs up to the west point of A ’s interval. After an IIPP reduction, the relation $R_x^1(A_p, B_p)$ is a relation from the set {<, =}. In a specific instance, the value of a relation from a set depends on the lengths of the intervals and their relative placement. For example, the scene in Figure 4.17a can transform into the scenes in Figure 4.17c or 4.17e after an IIPP scale reduction. Similarly, an IIPP

reduction can transform the scene in Figure 4.17b into the scenes in Figure 4.17d or Figure 4.17e. The values of the relation $R_x^1(A_p, B_p)$ in Figures 4.17c and 4.17d are qualitatively equivalent. The directions in Figures 4.17c-e are *compatible* with the directions in Figures 4.17a-b.

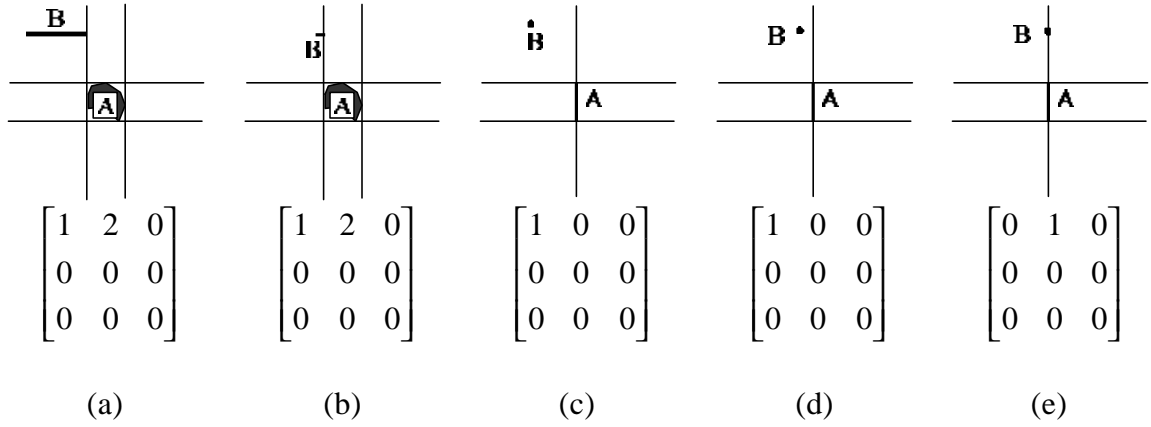


Figure 4.17: The relations between projections on to the x -axis: (a)-(b) interval relation is *meets* and point relations are (c)-(d) *before*, and (e) *equal*.

A query “Find all scenes where object B is *northwest* of A ” on Figure 4.17 would return the scenes in Figures 4.17a-d, while a query, “Find all scenes where object B intersects with the *north partition* of A ” would return the scenes in Figures 4.17a-b and 4.17e. This example illustrates that the directions recorded using the deep direction-relation matrices at a smaller scale are compatible with the directions recorded at a larger scale.

4.5.3.2 Intervals with relation *overlaps*

We apply the conditions of *overlaps* relation (Table 4.6) to the scaled intervals (Equations 4.13a-c).

$$\frac{x_{B-}^0}{a} < \frac{x_{A-}^0}{a} \quad (4.13a)$$

$$\frac{x_{B+}^0}{a} > \frac{x_{A-}^0}{a} \quad (4.13b)$$

$$\frac{x_{B+}^0}{a} < \frac{x_{A+}^0}{a} \quad (4.13c)$$

Combining the conditions between the extremes of scaled intervals for *overlaps* relation (Equations 4.13a-c) with the ranges of points x_A^1 and x_B^1 (Equations 4.9a-b), we get relations between the points in terms of inequality conditions (Equations 4.14a-c).

$$x_B^1 - N < x_A^1 - L \quad (4.14a)$$

$$x_B^1 + P > x_A^1 - L \quad (4.14b)$$

$$x_B^1 + P < x_A^1 + M \quad (4.14c)$$

The inequalities (Equations 4.14a-c) are expressed as the differences between points x_B^1 and x_A^1 using positive variables Q , R , and S , respectively (Equations 4.15a-c).

$$x_B^1 - x_A^1 = N - L - Q, \quad Q > 0 \quad (4.15a)$$

$$x_B^1 - x_A^1 = R - L - P, \quad R > 0 \quad (4.15b)$$

$$x_B^1 - x_A^1 = M - P - S, \quad S > 0 \quad (4.15c)$$

In Equation 4.15a, if N equals $L+Q$, $x_B^1 = x_A^1$; if N is more than $L+Q$, $x_B^1 > x_A^1$; and if N is less than $L+Q$, $x_B^1 < x_A^1$, which implies $x_B^1 \{<, =, >\} x_A^1$. Equations 4.15b-c also give the same result (Table 4.7).

4.5.3.3 Intervals with relation *equal*

We apply the conditions of *equal* relation (Table 4.6) to the scaled interval (Equations 4.16a-b).

$$\frac{x_{B-}^0}{a} = \frac{x_{A-}^0}{a} \quad (4.16a)$$

$$\frac{x_{B+}^0}{a} = \frac{x_{A+}^0}{a} \quad (4.16b)$$

We combine the conditions of the relation *equal* (Equations 4.16a-b) with the ranges of the points (Equations 4.9a-b), and obtain conditions for the points (Equations 4.17a-b).

$$x_B^1 - N = x_A^1 - L \quad (4.17a)$$

$$x_B^1 + P = x_A^1 + M \quad (4.17b)$$

The conditions between the points (Equations 4.17a-b) are arranged as the difference between the points (Equations 4.18a-b), which imply $x_B^1 < x_A^1$, $x_B^1 = x_A^1$, and $x_B^1 > x_A^1$ relations are possible; therefore, the resultant relation between the points is a set $\{<, =, >\}$ (Table 4.7).

$$x_B^1 - x_A^1 = N - L \quad (4.18a)$$

$$x_B^1 - x_A^1 = M - P \quad (4.18b)$$

This section derived relations between two points at scale S_1 if the relation between two intervals at scale S_0 is *meets*, *overlaps*, and *equal*. Similar derivation can be performed for the remaining ten relations between intervals (Table 4.7). For the interval relations *before* and *after*, the results for an IIPP reduction are unique. For all other relations, a scale reduction gives a set of relations. For a given pair of intervals, the actual relation between points is determined by the lengths of the intervals and their relative placements, as shown for the relation *meets* (Figure 4.17). However, values of new relations are compatible with the relations before scale reduction.

4.5.4 Collapsing a Pair of Intervals into a Point and an Interval

In the case IIPi, the reference object's interval projection collapses into a point, while the target object's projection continues to be an interval (Equations 4.19a-b). The resulting relations are given in Table 4.8.

$$\left[x_{A-}^0, x_{A+}^0 \right] \textcircled{R} x_A^1 \quad (4.19a)$$

$$[x_{B-}^0, x_{B+}^0] \textcircled{R} [x_{B-}^1, x_{B+}^1] \quad (4.19b)$$

If a particular relation does not fit into this case, the result is recorded as *impossible*. For example, if interval *B* equal interval *A*, after a significant scale reduction, both *A* and *B* must collapse into points. Due to the same length of both intervals, it is impossible for *B* to stay an interval, while *A* collapses into a point.

$R_x^0(A_i, B_i)$	$R_x^1(A_p, B_i)$
<	<
<i>m</i>	{<, <i>fa</i> }
<i>o</i>	{<, <i>fa</i> , <i>co</i> }
<i>fi</i>	{ <i>fa</i> , <i>co</i> }
<i>s</i>	<i>impossible</i>
<i>d</i>	<i>impossible</i>
=	<i>impossible</i>
<i>di</i>	{ <i>fa</i> , <i>co</i> , <i>sa</i> }
<i>si</i>	{ <i>co</i> , <i>sa</i> }
<i>f</i>	<i>impossible</i>
<i>oi</i>	{ <i>co</i> , <i>sa</i> , >}
<i>mi</i>	{ <i>sa</i> , >}
>	>

Table 4.8: The scale reduction of the type IIPI maps the relations between intervals onto the relations of point *B* with respect to interval *A*.

4.5.5 Collapsing a Pair of Intervals into an Interval and a Point

In the case IIIP, the reference object's projection continues to be an interval, while the target object's interval projection collapses into a point (Equations 4.20a-b). Table 4.9 gives the resulting relations after the change.

$$[x_{A-}^0, x_{A+}^0] \textcircled{R} [x_{A-}^1, x_{A+}^1] \quad (4.20a)$$

$$[x_{B-}^0, x_{B+}^0] \textcircled{R} x_B^1 \quad (4.20b)$$

$R_x^0(A_i, B_i)$	$R_x^1(A_i, B_p)$
<	<
<i>m</i>	{<, <i>as</i> }
<i>o</i>	{<, <i>as</i> , <i>du</i> }
<i>fī</i>	<i>impossible</i>
<i>s</i>	{ <i>as</i> , <i>du</i> }
<i>d</i>	<i>du</i>
=	<i>impossible</i>
<i>di</i>	<i>impossible</i>
<i>si</i>	<i>impossible</i>
<i>f</i>	{ <i>du</i> , <i>af</i> , >}
<i>oi</i>	{ <i>du</i> , <i>af</i> }
<i>mi</i>	{ <i>af</i> , >}
>	>

Table 4.9: The scale reduction of the type IIIP maps the relations between intervals into the relations of point *B* with respect to interval *A*.

4.5.6 Collapsing an Interval and a Point into a Pair of Points

In the case IPPP, the reference object's interval projection collapses into a point, while the target object's projection continues to be a point (Equations 4.21a-b). Table 4.10 gives the resulting relations.

$$[x_{A-}^0, x_{A+}^0] \textcircled{R} x_A^1 \quad (4.21a)$$

$$x_B^0 \textcircled{R} x_B^1 \quad (4.21b)$$

$R_x^0(A_i, B_p)$	$R_x^1(A_p, B_p)$
<	<
<i>as</i>	{<, =}
<i>du</i>	{<, =, >}
<i>af</i>	{=, >}
>	>

Table 4.10: The scale reduction of the type IPPP maps the relations of point *B* with respect to interval *A* onto the relations between points.

4.5.7 Collapsing a Point and an Interval into a Pair of Points

In the case PIPP, the reference object's projection continues to be a point, while target object's interval projection collapses into a point (Equations 4.22a-b). Table 4.11 gives possible relations after this change.

$$x_A^0 \textcircled{R} x_A^1 \tag{4.22a}$$

$$[x_{B-}^0, x_{B+}^0] \textcircled{R} x_B^1 \tag{4.22b}$$

$R_x^0(A_p, B_i)$	$R_x^1(A_p, B_p)$
<	<
<i>fa</i>	{<, =}
<i>co</i>	{<, =, >}
<i>sa</i>	{=, >}
>	>

Table 4.11: The scale reduction of the type PIPP maps the relations of interval B with respect to point A onto the relations between points.

4.5.8 Compatible Directions at Smaller Scales

The relations between projections at a smaller scale are compatible with the relations at a larger scale in the IIPP case of scale reduction (Section 4.5.3). Similar reasoning applies to the other four cases; therefore, the relations between the projections of objects on an axis, after a significant scale reduction, are compatible with the relations before zooming out. The region reference grid has nine direction partitions and sixteen boundary parts, which make 25 *parts of space*. Similarly, reference grids for a horizontal line and a vertical line have 15 parts of the space, and the reference grid for a point has 9 parts of space. We analyze the change of direction due to a significant scale reduction in a reference frame part by part.

The intersection of each part of space with a target object can have more than one separation (Figure 3.9). We call such a separation a *target component*. The union of all the target components yields the target object. The projection of a target component onto an axis can be a point or an interval. A significant scale reduction collapses at least one projection from all the target components and the reference object. A change in the type

of a pair of projections of the reference object and a target component on an axis due to zooming out falls in one of the five cases. For these cases, relations after a scale reduction on an axis are compatible with the relations before scale reduction. Combining all parts, we infer that the direction recorded by the deep direction-relation matrix at a smaller scale is compatible with, if not equal to, the direction recorded at a larger scale.

For example, at scale S_0 the target object B is a region (Figure 4.18a), and at scale S_1 it collapses into a line (Figure 4.18b), while the reference object A continues to remain a region. The target object B has five components B_1 , B_2 , B_3 , B_4 , and B_5 . At scale S_0 , the components B_1 , B_3 , and B_5 are regions, and B_2 and B_4 are lines. The components B_1 and B_5 intersect with the west partition, B_3 intersects with the northwest partition, and B_2 and B_4 intersect with the west-northwest line. At scale S_1 , the object B becomes a line. The components B_1 , B_3 , and B_5 collapse into lines and B_2 and B_4 collapse into points.

We analyze the relation of each component's x -axis projections with the reference object's x -axis projection for this example (Table 4.12). The changes in the types of projections occur for the components B_2 and B_4 only, and both the changes are of the same type IIIIP. The values of relations for the projections of B_2 and B_4 with respect to A 's projection after scale reduction is obtained from Table 4.9. The analysis for the y -axis projections can be performed similarly, but there is no change in the types of projections of the reference object and the target components on the y -axis in this example. At both scales, the directions of B with respect to A are recorded as the same value (Equation 4.23). The relations for each component's projections with respect to the reference projections at scale S_1 are compatible with the relations at S_0 ; therefore, the direction at scale S_1 in this example is compatible with the direction at scale S_0 .

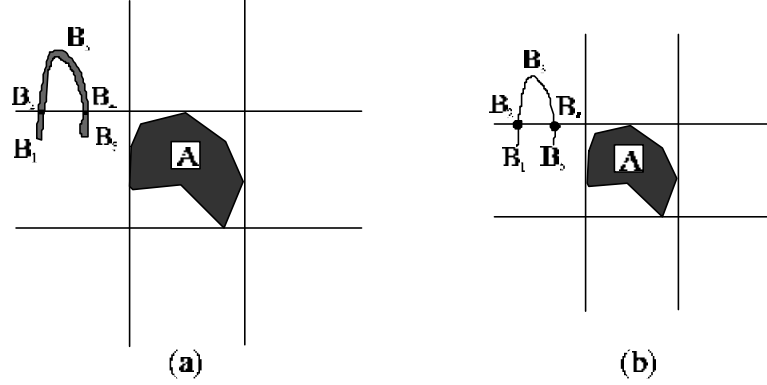


Figure 4.18: (a) The region-region pair at scale S_0 and (b) region-line pair at scale S_1 after a zooming-out operation.

n	At scale S_0			At scale S_1		
	$\Pi_x^0(A)$	$\Pi_x^0(B_n)$	$R_x^0(A, B_n)$	$\Pi_x^1(A)$	$\Pi_x^1(B_n)$	$R_x^1(A, B_n)$
1	interval	interval	<	interval	interval	<
2	interval	interval	<	interval	point	<
3	interval	interval	<	interval	interval	<
4	interval	interval	<	interval	point	<
5	interval	interval	<	interval	interval	<

Table 4.12: The projections of the reference object A and the components of the target object B onto the x -axis, and their relations, where n is the component number.

$$dir(A, B) = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (4.23)$$

The directions between a pair of objects recorded at two different scales may not always be equal, however. If they were always equal, the models of directions between points would be sufficient, and there would be no need for models of directions between extended objects. The study in this section shows that the deep direction-relation matrices record directions that are *compatible* with the directions recorded at larger scales, which makes this model useful for direction-based queries at multiple scales.

4.6 Analysis of the Deep Direction Model

The deep direction-relation matrix uses nine elements, and each element records information in nine bits; therefore, a deep direction-relation matrix needs 81 bits. This is 224% more than 25 bits in a 5x5 matrix. In the exchange of the larger number of bits, the deep direction-relation matrix has the following advantages over a 5x5 matrix: (1) it records nine elements, regardless of the types of the objects, (2) all elements across the matrix have the same structure, and (3) it facilitates the assessment of compatibility of directions at multiple scales (Section 4.5).

4.6.1 Cognitively Plausible Values of Directions

The examples in this section illustrate that the deep direction-relation matrix records the same value for equivalent directions, irrespective of the dimensions of objects. The directions of object B with respect to object A in Figures 4.19a, 4.20a, and 4.21a are equivalent, and in all these cases the value of recorded directions $dir(A, B)$ is the same (Equation 4.24a). Similarly, the directions of B with respect to A in Figures 4.19b and 4.20b are equivalent, and they are captured as the same value using neighbor codes (Equation 4.24b). The direction $dir(A, B)$ in Figures 4.20c and 4.21b are equivalent and recorded as the same value (Equation 4.24c). These examples illustrate that the deep direction-relation matrix is capable of capturing directions for point, line, and polygon target objects in a consistent and cognitively plausible way.

$$dir(A, B) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (4.24a)$$

$$dir(A, B) = \begin{bmatrix} 0 & 1 & 2 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (4.24b)$$

$$dir(A, B) = \begin{bmatrix} 0 & 32 & 2 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (4.24c)$$



Figure 4.19: Pairs of region objects. (a) Object *B* is north of object *A* and (b) object *B* is north of *A* and intersects with the north-northeast line (N-NeL).

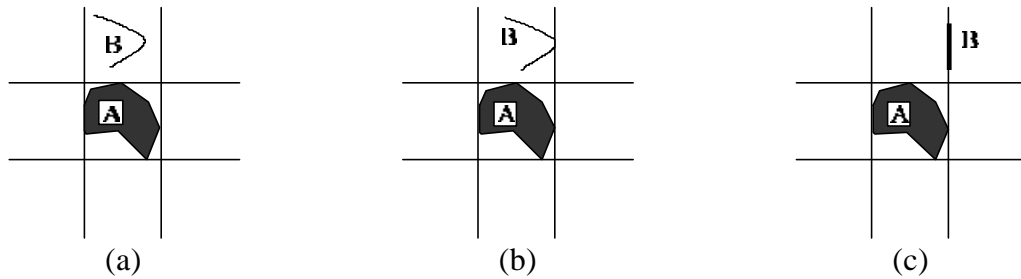


Figure 4.20: Region reference and line target. (a) object *B* is north of *A*, (b) *B* is north of *A* and intersects with the N-NeL, and (c) object *B* completely lies on the N-NeL.



Figure 4.21: Region reference and point target. (a) object *B* is north of *A* and (b) object *B* completely lies on the N-NeL.

4.6.2 Advantages of the Deep Direction Model Over Existing Models

This section compares the deep direction-relation matrix with the MBR-based model (Section 2.4.6) and the coarse direction-relation matrix (Section 3.1). The MBR-based model is adequate only for recording directions between rectangles, and the coarse

direction-relation matrix is adequate only for recording directions between regions. In the projection-based direction partitions there are four different types of objects (Figure 4.2). These four types of objects can be reference as well as target objects, which gives sixteen different types of reference-object pairs (Figure 4.22).

Reference object type	Target object type			
	Point	Vertical-line	Horizontal-line	Region or other lines
Point				
Vertical-line				
Horizontal-line				
Region or other lines				

Figure 4.22: Sixteen different types of object pairs.

Out of these sixteen types, the model based on MBRs and the coarse direction-relation matrix apply to only one type of pair (i.e., pair of regions). They do not apply to the remaining fifteen types of pairs. For example, if a horizontal line target object coincides with a grid line with respect to a region reference object, none of the 169 MBR relations can record this direction (Figure 2.15). Similarly, none of the 218 coarse direction-relation matrices can record this direction (Figure 3.6). The deep direction-

relation matrix, however, provides a unified framework to represent directions for all sixteen pairs.

4.7 Summary

This chapter extended the model based on the coarse direction-relation matrices to include information about the intersections of the target object with the boundaries of direction partitions. The new model is capable of recording directions between arbitrary pairs of point, line, and polygon objects. The deep direction-relation matrix always has nine elements, regardless of the dimensions of the objects. We demonstrated that deep direction-relation matrices record directions that are compatible with the directions recorded at larger scales, which makes the deep direction model useful for direction-based queries in multi-resolution spatial databases. Examples in this chapter showed that the deep direction-relation matrix records cognitively plausible values of directions; therefore, the deep direction model frees the user from pondering about the dimension of the objects.

Chapter 5

Similarity Between Cardinal Directions

People casually assess similarity between spatial scenes in their routine activities. Likewise, users of a pictorial database are often interested in retrieving scenes that are similar to a given scene, and ranking them according to degrees of their match. For example, a town architect would like to query a database for the towns that have a landscape similar to the landscape of the site of a planned town. Similarity is an intuitive and subjective judgment. It displays no strict mathematical models (Tversky 1977). Bruns and Egenhofer (1996) use spatial relations between objects for the assessment of scene similarity. Spatial relations are also used for similarity assessment in image databases (Chu *et al.* 1994; Bimbo *et al.* 1995; Bimbo and Pala 1997; Chu *et al.* 1998), multimedia databases (Al-Khatib *et al.* 1999; Yoshitaka and Ichikawa 1999), and video databases (Jiang and Elmagarmid 1998; Pissinou *et al.* 1998; Aslandogan and Yu 1999). In order to use spatial relations for similarity assessment, we need methods to assess similarity between spatial relations.

Cardinal directions can be used for defining the results of queries in spatial databases. For example, the query scene (Figure 5.1a) and scenes in the database (Figures 5.1b-d) contain objects *A* and *B*, and the value of the topological relations between objects *A* and *B* in all these scenes is *disjoint*. All scenes in this database are topologically equivalent to the query scene. However, when considering the cardinal direction as an additional search criterion, one can determine that Scene 0 is the most similar to the query scene. In order to make the direction-relation matrix applicable for assessing spatial similarity between scenes, this chapter develops a method to assess similarity between cardinal directions. The similarity assessment uses the detailed direction-relation matrix, which for simplicity is referred as *direction-relation matrix* in this chapter.

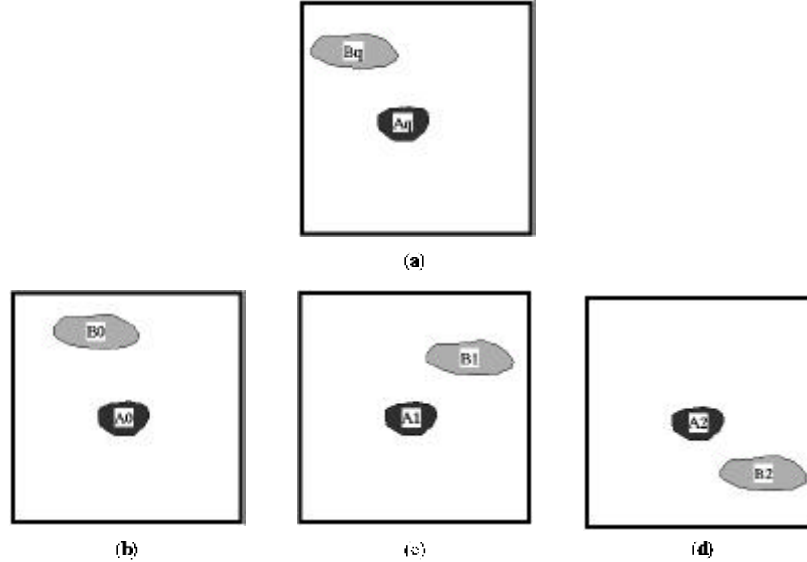


Figure 5.1: (a) The query scene and (b)-(d) Scenes 0-2 in a database.

We determine the similarity between two directions by actually assessing the dissimilarity between them, which depends on the distance between these directions. Section 5.1 discusses the distance computation, and converts it into a problem of transforming a direction-relation matrix into another direction-relation matrix. Section 5.2 describes a method to compute the minimum cost for this transformation using the transportation algorithm. Section 5.3 describes the method to convert the minimum cost into a similarity value. Section 5.4 summarizes the results of this chapter.

5.1 The Distance between Two Cardinal Directions

As a quantitative measure for direction similarity, we introduce a distance measure between two cardinal directions, such that (1) a zero value implies that both directions are identical and (2) $distance(D_0, D_1) > distance(D_0, D_2)$ means D_0 is more similar to D_2 than D_0 to D_1 . The symbol D_0 denotes the direction of object B with respect to object A in Scene 0.

A direction-relation matrix must have at least one element with a non-zero value. If a direction-relation matrix has exactly one non-zero element, we call it a *single-element direction-relation matrix*. A direction that corresponds to a single-element direction-relation matrix is called a *single-element direction*. There are nine single-element

directions corresponding to nine cardinal directions, which are also called *mutually exclusive directions*. A direction-relation matrix with more than one non-zero element is called a *multi-element direction-relation matrix*. A direction that corresponds to a multi-element direction-relation matrix is referred as a *multi-element direction*. The distance measures are needed for all direction-relation matrices. We develop first a method to compute distance between single-element directions (Section 5.1.1), and extend it subsequently to multi-element directions (Section 5.1.2).

5.1.1 Distance Between Two Single-Element Direction-Relation Matrices

A conceptual neighborhood graph, for a set of mutually exclusive spatial relations, serves as the basis for computing distances between the relations in this set. Conceptual neighborhood graphs have been used for deriving distance measures between 1-D interval relations (Freksa 1992a), topological relations (Egenhofer and Al-Taha 1992; Egenhofer and Mark 1995a), and minimum bounding rectangle relations (Papadias and Dellis 1997). A continuously changing relation follows a path along the conceptual neighborhood graph. For example, if a target object *B* moves eastward from the northwest tile (Figure 5.2a) it cannot move directly to the northeast tile (Figure 5.2c). It must go through the direction tile(s) that connect the northwest and northeast tiles. The shortest path would lead through the north tile (Figure 5.2b), although other connected paths are possible as well, e.g., through west, same, and east tiles.

In order to compute the distance between cardinal directions, we construct a conceptual neighborhood graph for the nine cardinal directions using the 4-neighborhood of the nine tiles. This graph has a vertex for each cardinal direction and an edge for each pair of cardinal directions that are horizontally or vertically adjacent (Figure 5.3). The distance between two cardinal directions is the length of the shortest path between two directions in the conceptual neighborhood graph (Figure 5.4). The distance between two identical directions is zero, which is the shortest of all distances. The distance between the cardinal directions northwest and southeast is four, which is the maximum. The only

other pair with the maximum distance is northeast and southwest. The distance function abides by the axioms of a distance, that is, positivity, symmetry, and triangle inequality.

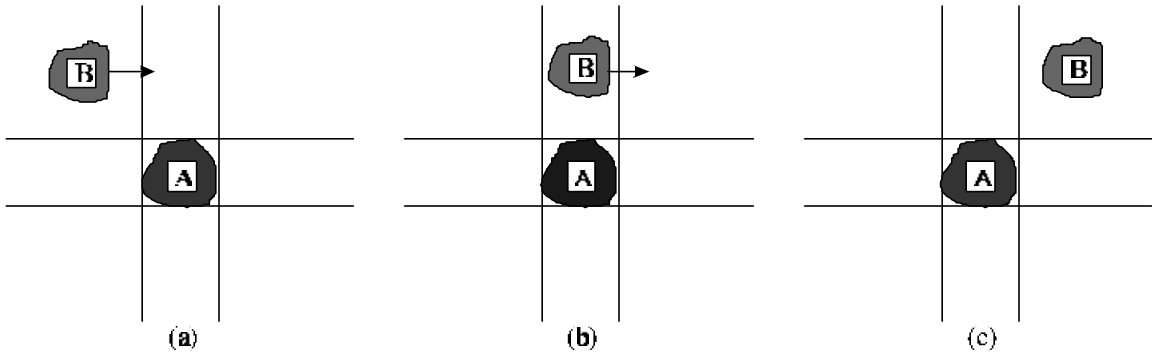


Figure 5.2: The shortest path to move the target object B from the northwest tile to the northeast tile is through the north tile, while considering only single-element directions.

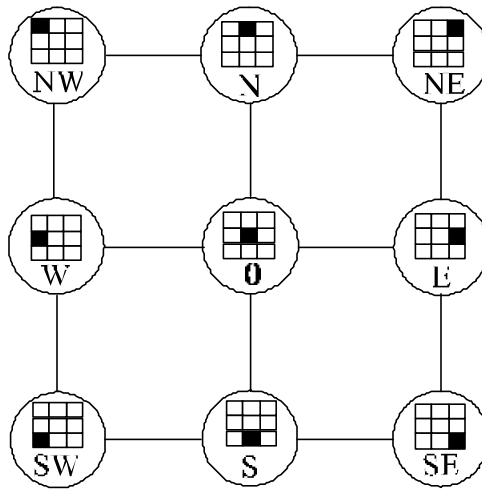


Figure 5.3: The conceptual neighborhood graph for nine cardinal directions based on the 4-neighborhood between tiles.

The distance between the directions of B with respect to A along the conceptual neighborhood graph in Figures 5.2a and 5.2b is 1, and the distance between the directions in Figures 5.2a and 5.2c is 2. Based on these distances, we infer that the direction of B with respect to A in Figure 5.2a is more similar to the direction in Figure 5.2b than the direction in Figure 5.2a to the direction in Figure 5.2c. Qualitative changes of direction relations for an ordered pair of regions follow a 4-conceptual neighborhood graph. Such

changes can occur due to increasing or decreasing sizes of objects, movement of objects, or rotation of objects. The distance between single element direction-relation matrices serves as the basis for the distance between multi-element direction-relation matrices.



















									
	N	NE	E	SE	S	SW	W	NW	0
 N	0	1	2	3	2	3	2	1	1
 NE	1	0	1	2	3	4	3	2	2
 E	2	1	0	1	2	3	2	3	1
 SE	3	2	1	0	1	2	3	4	2
 S	2	3	2	1	0	1	2	3	1
 SW	3	4	3	2	1	0	1	2	2
 W	2	3	2	3	2	1	0	1	1
 NW	1	2	3	4	3	2	1	0	2
 0	1	2	1	2	1	2	1	2	0

Figure 5.4: Four-neighbor distances between cardinal directions for regions.

5.1.2 Distance Between Two Multi-Element Direction-Relation Matrices

In this section, we extend the method of computing the distance between cardinal directions from single-element direction-relation matrices to multi-element direction-relation matrices. If the target object B moves eastward from the northwest tile (Figure 5.5a) to the northeast tile (Figure 5.5e), it will have the following direction relations with respect to A on its trajectory: (1) northwest and north (Figure 5.5b), (2) north (Figure

5.5c), and (3) north and northeast (Figure 5.5d). The directions in Figures 5.5b and 5.5d require multi-element direction-relation matrices for their representation.

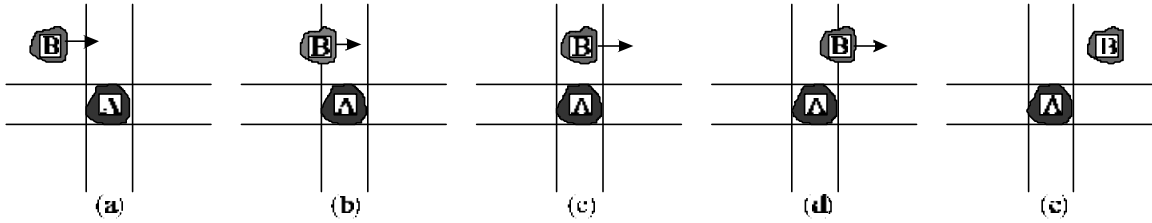


Figure 5.5: The target object moves across single as well as multi-element cardinal directions from (a) northwest through (b) northwest and north, (c) north, (d) north and northeast, to (e) northeast.

There are 218 coarse direction-relation matrices (Figure 3.6), out of which only nine directions are represented by single-element direction-relation matrices. The remaining 209 directions are represented by multi-element direction-relation matrices. A conceptual neighborhood graph for all 218 directions can be constructed, assuming a uniform distribution of the target object in non-zero direction tiles. For example, a target object that intersects with the north and northeast tiles can be assumed to have 50% of its area in each of these tiles; however, such an assumption will mostly give values of direction-relation matrices that are different from the actual values. Instead of making such assumptions, we develop a computational method for determining the distance between two arbitrary multi-element direction-relation matrices. This method is based on the conceptual neighborhood graph for nine cardinal directions (Figure 5.3).

Definition 5.1: The distance between two direction-relation matrices, D^0 and D^1 , is the minimum cost for transforming matrix D^0 into D^1 by moving the non-zero elements of D^0 from their locations to the locations of the non-zero elements of D^1 along the conceptual neighborhood graph.

The cost of this transformation is the weighted sum of the distances along the neighborhood graph between the *source* and *destination* direction tiles, where a source refers to a cardinal direction from where a non-zero element is moved and a destination

refers to a cardinal direction where the element is moved to. The weighting of a distance between a source and a destination is done by the element values moved between them. For example, transforming matrix D^0 (Equation 5.1a) into D^1 (Equation 5.1b) requires the movement of the value 0.4 from northwest to northeast, and the value 0.6 from north to northeast. The cost of this transformation is $0.4 \times \text{distance}(\text{NW}, \text{NE}) + 0.6 \times \text{distance}(\text{N}, \text{NE})$, which is $0.4 \times 2 + 0.6 \times 1 = 1.4$.

$$D^0 = \begin{bmatrix} 0.4 & 0.6 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (5.1a)$$

$$D^1 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (5.1b)$$

The remainder of this section introduces consistency constraints and properties of intermediate matrices, which are needed to develop the method for distance computation between arbitrary direction-relation matrices.

Definition 5.2: The sum of a matrix P is defined as the sum of the values of its elements (Equation 5.2).

$$\text{sum}(P) := \sum_{\forall i} \sum_{\forall j} P_{i,j} \quad (5.2)$$

Definition 5.3: The commonality C^{01} between two direction-relation matrices, D^0 and D^1 , is a 3x3 matrix defined as the minimum of each pair of corresponding element values (Equation 5.3).

$$\forall i, j: C_{ij}^{01} := \min(D_{ij}^0, D_{ij}^1) \quad (5.3)$$

The values of elements in C^{01} lie in the interval $[0, 1]$. The value of $sum(C^{01})$ also lies in the interval $[0, 1]$. It is 0 if all the corresponding pairs of elements have at least one 0. It would be 1 if the distribution of the target objects in both D^0 and D^1 is identical. Since the minimum of a set of numbers is unique and does not depend on their order, the calculation of the commonality is commutative (Equation 5.4).

$$C^{01} = C^{10} \quad (5.4)$$

For example, the commonality matrices C^{01} and C^{10} for directions of B with respect to A in Scene 0 and Scene 1 (Figure 5.6) have the same values (Equation 5.5.). This scenario is used as a running example throughout this chapter.

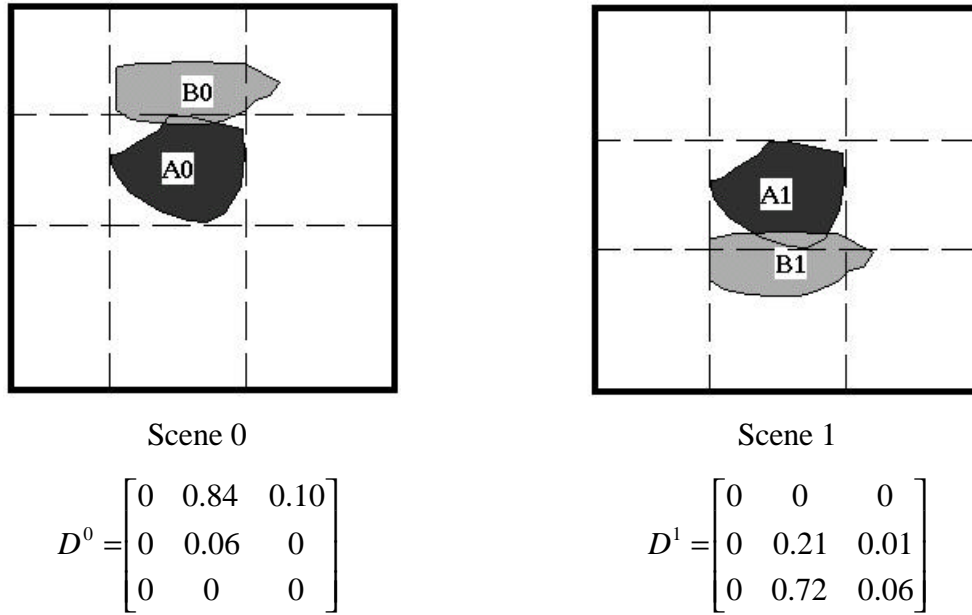


Figure 5.6: A direction comparison example Scene 0 and Scene 1 with identical objects, but different directions D^0 and D^1 , respectively.

$$C^{01} = C^{10} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0.06 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (5.5)$$

Definition 5.4: The *asymmetric difference* R^{01} between two direction-relation matrices, D^0 and D^1 , is defined as the difference of the direction-relation matrix D^0 and the commonality (Equation 5.6a), and has a corresponding value for R^{10} (Equation 5.6b).

$$R^{01} := D^0 - C^{01} \quad (5.6a)$$

$$R^{10} := D^1 - C^{10} \quad (5.6b)$$

We use the term *non-zero part of a matrix* for a non-zero element or a fraction of a non-zero element. The asymmetric difference R^{01} has the distinct *non-zero parts* of D^0 that are not in D^1 . Conversely R^{10} (Equation 5.6b) has the distinct non-zero parts of D^1 that are not in D^0 . For example, the asymmetric difference matrix R^{01} (Equation 5.7a) for scenes in Figure 5.6 has no non-zero parts that are in D^1 , while R^{10} (Equation 5.7b) has no non-zero parts that are in D^0 .

$$R^{01} = \begin{bmatrix} 0 & 0.84 & 0.10 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (5.7a)$$

$$R^{10} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0.15 & 0.01 \\ 0 & 0.72 & 0.06 \end{bmatrix} \quad (5.7b)$$

The values of elements in R^{01} and R^{10} lie in the closed interval $[0, 1]$. The values of $sum(R^{01})$ and $sum(R^{10})$ also lie in the interval $[0, 1]$. The value 0 for $sum(R^{10})$ means there is no difference between matrices D_0 and D_1 , whereas the value 1 means there is no commonality between matrices D_0 and D_1 .

Definition 5.5: The *direction-difference* (Δ^{01}) between two direction-relation matrices, D^0 and D^1 , is defined as the difference of the two relations' asymmetric differences (Equation 5.8).

$$\Delta^{01} := R^{01} - R^{10} \quad (5.8)$$

The values of the elements in Δ^{01} lies in the interval [-1, 1]. In order to express the direction-difference in terms of direction-relation matrices, we substitute the value of R^{01} and R^{10} (Equations 5.6a and 5.6b) in Equation 5.8, and obtain Equation 5.9, as commonality matrices C^{01} and C^{10} cancel each other. For example, Equation 5.10 gives the direction-difference matrices Δ^{01} for the scenes in Figure 5.6.

$$\Delta^{01} = D^0 - D^1 \quad (5.9)$$

$$\Delta^{01} = \begin{bmatrix} 0 & 0.84 & 0.10 \\ 0 & -0.15 & -0.01 \\ 0 & -0.72 & -0.06 \end{bmatrix} \quad (5.10)$$

Theorem 5.1.1:

The sum of the elements in R^{01} equals the sum of the elements in R^{10} .

Proof:

The sum of the elements of a detailed direction-relation matrix is 1 (Equation 5.11).

$$\text{sum}(D^0) = \text{sum}(D^1) = 1 \quad (5.11)$$

The addition operation (+) follows the associative law; therefore, we can express the asymmetric difference matrices (Equations 5.6a-b) in the sum form (Equations 5.12a-b).

$$\text{sum}(R^{01}) = \text{sum}(D^0) - \text{sum}(C^{01}) \quad (5.12a)$$

$$\text{sum}(R^{10}) = \text{sum}(D^1) - \text{sum}(C^{10}) \quad (5.12b)$$

Let us assume the value of $\text{sum}(C^{01})$ is x , which equals $\text{sum}(C^{10})$; substituting x for these sums in Equation 5.12a-b and combining them with Equation 5.11, we obtain expressions for the sum of asymmetric difference matrices in terms of x (Equations 5.13a-b).

$$\text{sum}(R^{01}) = 1 - x \quad (5.13a)$$

$$\text{sum}(R^{10}) = 1 - x \quad (5.13b)$$

The right hand sides of both asymmetric difference matrices' sums are identical (Equations 5.13a-b), which proves the theorem (Equation 5.14).

$$\text{sum}(R^{01}) = \text{sum}(R^{10}) \quad (5.14)$$

□

For example, the sum of elements in R^{01} (Equation 5.7a) is 0.94, which equals the sum of elements in R^{10} (Equation 5.7b).

Corollary 5.1.2:

The sum of the elements in a direction-difference matrix is 0.

Proof:

We can express the direction-difference matrix (Equation 5.8) in the sum form (Equation 5.16). The values of the sums of asymmetric difference matrices' elements is identical, (Equation 5.14), which proves this corollary (Equation 5.16).

$$\text{sum}(\Delta^{01}) := \text{sum}(R^{01}) - \text{sum}(R^{10}) \quad (5.15)$$

$$\text{sum}(\Delta^{01}) = 0 \quad (5.16)$$

□

For example, the sum of elements of Δ^{01} in Equation 5.10 is zero. Non-zero elements of the commonality matrix C^{01} capture the common non-zero parts of the matrices D^0 and D^1 ; therefore, the non-zero parts that correspond to non-zero parts of C^{01} must not be moved while transforming D^0 into D^1 (Definition 5.1). Moving them would increase the cost of this transformation, such that the computed cost for this transformation would not be the minimum value. Only those non-zero parts of D^0 should be moved that are zero in D^1 , which means only non-zero elements of R^{01} must be moved to obtain R^{10} .

Definition 6: The distance between two matrices D^0 and D^1 is the minimum cost incurred in transforming R^{01} into R^{10} .

Theorem 5.2.1:

The maximum 4-neighbor distance ($distance_{\max}^4$) between two direction-relation matrices is 4.

Proof:

The maximum cost is incurred when the maximum possible value of $sum(R^{01})$ is moved by the maximum possible distance. The maximum value of $sum(R^{01})$ is 1 and the maximum 4-neighbor distance between two cardinal directions is 4 (Figure 5.4); therefore, the value of $distance_{\max}^4$ is $4 \times 1 = 4$. □

The maximum distance between two direction-relation matrices can occur only between single-element direction-relation matrices that have non-zero values for the farthest direction tiles. For example, the value of 4-neighbor distance between two single-element direction-relation matrices with non-zero values in the southwest and northeast tiles is 4.

In the direction-difference Δ^{01} , non-zero elements that correspond to non-zero elements of R^{01} are of positive polarity, while non-zero elements that correspond to non-zero elements of R^{10} are of negative polarity (Equation 5.8). The sum of the elements of the matrix Δ^{01} is zero (Corollary 5.1.2). The matrix Δ^{01} has all the necessary information to compute the minimum cost of transforming D^0 into D^1 , which is the same as the minimum cost of transforming R^{01} into R^{10} . Section 5.2 uses Δ^{01} for computing the distance between two multi-element direction-relation matrices.

5.2 The Minimum Cost Solution for the Transformation Problem

The problem of determining the minimum cost for transforming matrix R^{01} into matrix R^{10} can be formulated as a balanced transportation problem (Murty 1976; Strayer

1989) or a minimum-cost maximum-flow problem across a flow-network (Ford and Fulkerson 1962). Both the transportation problem and the network flow problem are special cases of the linear programming problem (Dantzig 1963; Dantzig and Thapa 1997). This section formulates the problem of transforming R^{01} into R^{10} as a balanced transportation problem and describes a method to solve the transportation problem.

A transportation problem is graphically represented by a *transportation tableau* (Figure 5.7), which records the supplies of all the *warehouses*, the demands of all the *markets*, and the *unit costs* for all the pairs of the warehouses and the markets. Let us assume there are n negative and p positive elements in the direction difference matrix Δ^{01} . Each positive element in Δ^{01} corresponds to a warehouse, and each negative element corresponds to a market in the transportation tableau for Δ^{01} . The supply of the i th warehouse W_i is s_i , which equals the magnitude of the corresponding element in Δ^{01} . Similarly, the demand of the j th market M_j is d_j , which also equals the magnitude of the corresponding element in Δ^{01} . We identify the markets and warehouses in the transportation tableau by the names of corresponding direction tiles. The cost c_{ij} for moving a unit supply from W_i to M_j is $distance(W_i, M_j)$.

	M_1	M_2	...	M_n	
W_1	c_{11}	c_{12}	...	c_{1n}	s_1
W_2	c_{21}	c_{22}	...	c_{2n}	s_2
\vdots	\vdots	\vdots		\vdots	\vdots
W_p	c_{p1}	c_{p2}	...	c_{pn}	s_p
	d_1	d_2	...	d_n	
					$\sum_{i=1}^p s_i - \sum_{j=1}^n d_j$

Figure 5.7: The balanced transportation tableau.

The $sum(R^{01})$ equals the $sum(R^{10})$ (Theorem 5.1.1); therefore, the sum of the supplies of all warehouses equals the sum of the demands of all markets. Due to this equality, this transportation problem is a balanced transportation problem. Let us assume the number of units to be shipped from the warehouse W_i to the market M_j is x_{ij} in the final solution. The transportation problem is to determine the values of x such that the total cost (z) is minimum (Equation 5.17) and the warehouse and market constraints are satisfied (Equations 5.18a-b).

$$z = \sum_{i=1}^p \sum_{j=1}^n c_{ij} x_{ij} \quad (5.17)$$

$$\forall i \sum_{j=1}^n x_{ij} = s_i \quad (5.18a)$$

$$\forall j \sum_{i=1}^p x_{ij} = d_j \quad (5.18b)$$

For example, the problem of transforming D^0 into D^1 in Figure 5.6 is formulated as a transportation problem (Figure 5.8) from Equation 5.10. The distance between the markets and the warehouses is obtained from the table of 4-neighbor distances (Figure 5.4). The transportation problem is to determine a set of x values that gives the minimum value of z (Equation 5.19), such that it satisfies warehouse and market constraints (Equation 5.20a-f).

$$z = 3x_{11} + 2x_{12} + 2x_{13} + x_{14} + 2x_{21} + 3x_{22} + x_{23} + 2x_{24} \quad (5.19)$$

$$x_{11} + x_{12} + x_{13} + x_{14} = 0.84 \quad (5.20a)$$

$$x_{21} + x_{22} + x_{23} + x_{24} = 0.10 \quad (5.20b)$$

$$x_{11} + x_{21} = 0.06 \quad (5.20c)$$

$$x_{12} + x_{22} = 0.72 \quad (5.20d)$$

$$x_{13} + x_{23} = 0.01 \quad (5.20e)$$

$$x_{14} + x_{24} = 0.15 \quad (5.20f)$$

	SE	S	E	Same	
N	x_{11} 3	x_{12} 2	x_{13} 2	x_{14} 1	0.84
NE	x_{21} 2	x_{22} 3	x_{23} 1	x_{24} 2	0.10
	0.06	0.72	0.01	0.15	0.94

Figure 5.8: The transportation tableau for the direction difference matrix of the example.

The transportation problem is solved in two phases: (1) finding a basic feasible solution and (2) improving the basic feasible solution iteratively, until an optimal solution is obtained. A basic feasible solution is a set of x values that satisfy the market and warehouse constraints, but it may not give the minimum value of z . There can be more than one set of x values that yield the minimum value of z . Section 5.2.1 describes a method to obtain a basic feasible solution and Section 5.2.2 describes an algorithm for optimizing a basic feasible solution.

5.2.1 A Basic Feasible Solution

A basic feasible solution can be obtained with the northwest corner method (Strayer 1989). The term northwest corner corresponds to the top-left cost value, that is, first row and first column, in the transportation tableau. Alternatives to the northwest corner method are the minimum entry method and Vogel's advanced start method (Strayer 1989). We use here the northwest corner method.

The Northwest-Corner Method (Strayer 1989, p. 180)

- (0) Given an initial balanced transportation tableau.
- (1) Use the northwest-most cost in the tableau to empty a warehouse or completely fill a market demand. The northwest-most cost is the cost in the top-left position of the tableau. Circle the cost used and write above the circle the amount of goods shipped by that route. Reduce the supply and demand in the row and column containing the cost used.

- (2) Delete the row or column corresponding to the emptied warehouse or fully supplied market; if both happens simultaneously, delete the row unless that row is the only row remaining in which case delete the column.
- (3) If all tableau entries are deleted, STOP; otherwise go to (1).

For example, when we apply the northwest corner method to the transportation problem in Figure 5.8, we obtain a basic feasible solution (Figure 5.9). A circle on a cost implies that the value of x for this path is non-zero, that is, this path is used for transferring the element values. These non-zero values of x are written at the top of the circles. The value of z for the solution in Figure 5.9 is 1.89 (Equations 5.21a-c).

	SE	S	E	Same	
N	3 (3) 0.06	2 (2) 0.72	2 (2) 0.01	1 (1) 0.05	0.84
NE	2	3	1	2 (2) 0.10	0.10
	0.06	0.72	0.01	0.15	$z=1.89$

Figure 5.9: A basic feasible solution of the transportation problem in the example using the northwest corner method.

$$z=3*0.06+2*0.72+2*0.01+1*0.05+2*0.10 \quad (5.21a)$$

$$z=0.18+1.44+0.02+0.05+0.20 \quad (5.21b)$$

$$z=1.89 \quad (5.21c)$$

5.2.2 Optimizing a Basic Feasible Solution

A basic feasible solution may not be an optimum solution; therefore, it must be tested for the optimality and optimized, if necessary. This section describes the transportation algorithm (Strayer 1989), which is used for optimizing a basic feasible solution. The transportation algorithm uses the term cycle (C), which is a subset of cells of the transportation tableau (T) such that each row and each column of T contains exactly zero or two cells of C (Strayer 1989, p. 151).

The Transportation Algorithm (Strayer 1989, p. 153)

- (0) Given an initial balanced transportation tableau.
- (1) Obtain a basic feasible solution and a corresponding basis using a method such as northwest corner method.
- (2) Let $b_1 = 0$. Determine $a_1, a_2, \dots, a_p, b_1, b_2, \dots, b_n$ uniquely such that $a_i + b_j = c_{ij}$ for all basis cells c_{ij} .
- (3) Replace c_{ij} by $c_{ij} - a_i - b_j$; these are the new cells c_{ij} .
- (4) If $c_{ij} \geq 0$ for all i and j , STOP; replace all cells with their original costs from (0); the basic feasible solution given by the current basis cell is optimal. Otherwise, continue.
- (5) Choose $c_{ij} < 0$. Usually, the most appropriate choice is the most negative c_{ij} . Label this cell as a “getter” cell (+). By convention this cell is distinguished by squaring it instead of circling it. Find the unique cycle C in the tableau determined by this (squared) cell and basis cells. Label the cells in C alternately as “giver” cells (-) and “getter” cells (+). Choose the “giver” cell associated with the smallest amount of goods. If there is a tie among certain “giver” cells for the smallest amount of goods, choose any such cell.

- (6) Add the squared cell of (5) to the basis, that is, circle it in a new tableau. Remove the chosen “giver” cell of (5) from the basis, that is, do not encircle it in a new tableau. Add the amount of goods given up by this “giver” cell to all amounts of “getter” cells in C ; subtract the amount of goods given by this “giver” cell from all amounts of goods of “giver” cells in C . Go to (2).

The following example demonstrates the optimization process (Figure 5.10) using the transportation algorithm on the basic feasible solution in Figure 5.9.

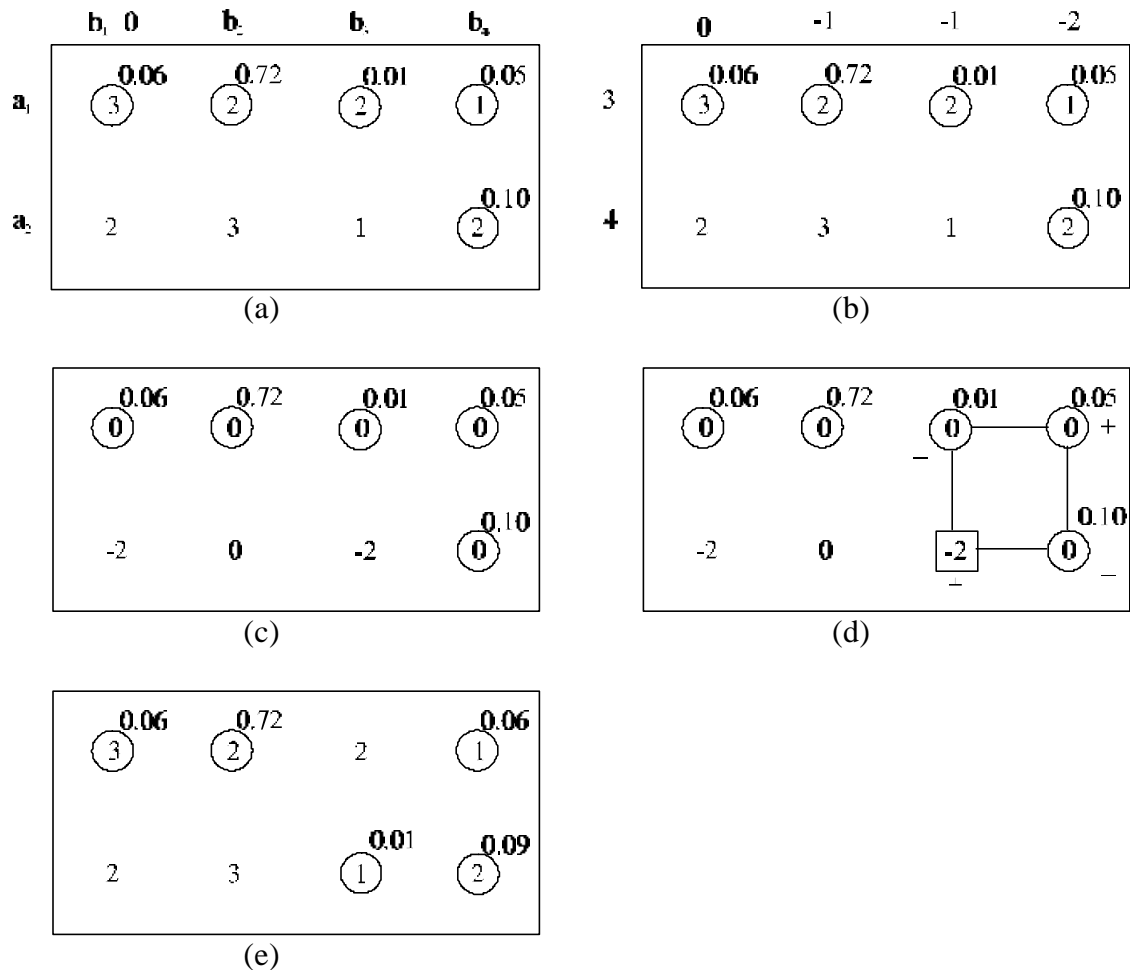


Figure 5.10: An iteration of the transportation algorithm for the example.

According to step 2 in the transportation algorithm, we put the value of b_1 as 0 (Figure 5.10a). The values of a_1 , a_2 , b_2 , b_3 , and b_4 are computed according to this step

(Figure 5.10b). We compute new values of c_{ij} according to step 3, and replace the old values of c_{ij} by the new values of c_{ij} in the tableau (Figure 5.10c). There are two cells with negative values of c_{ij} in the first and third columns of the second row (Figure 5.10c). The values of the new c_{ij} in these cells are the same; therefore, we arbitrarily choose the cell in the third column and put a square on it (Figure 5.10d). This cell becomes a getter cell, and we form a cycle by marking the getter cells as “+” and giver cells as “-” alternately. The chosen giver cell is the cell in the first row and third column, as it has the least value of x_{ij} (0.01) among the giver cells. The values of the giver cells (x_{ij}) are reduced by 0.01 and the values of the getter cells (x_{ij}) are increased by 0.01. The chosen giver cell leaves the basis, the squared cell joins the basis, and the original value of c_{ij} are replaced back in the tableau (Figure 5.10e). The second and last iteration of the optimization process for this example is performed similarly (Figure 5.11).

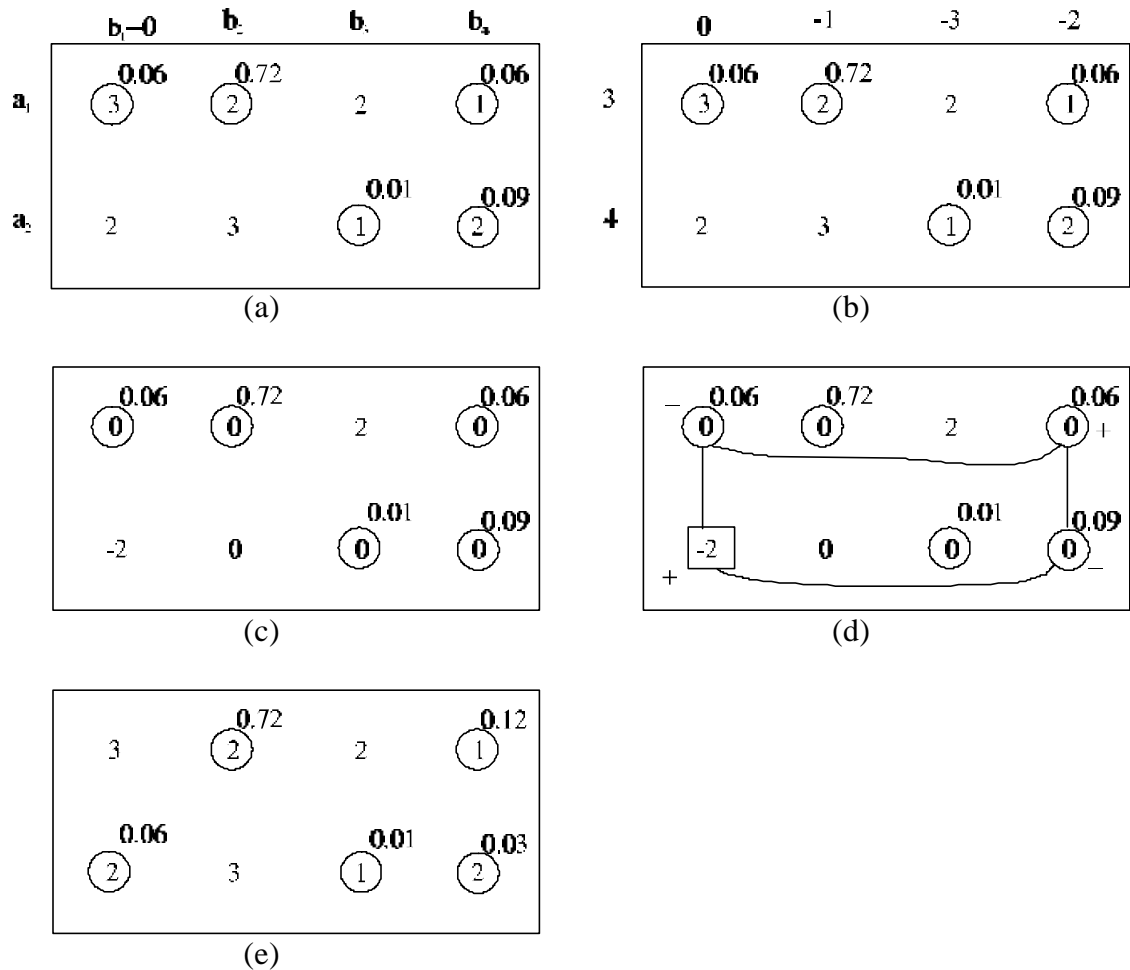


Figure 5.11: The second and last iteration of the transportation algorithm for the example.

The minimum cost of the transformation from the obtained optimum solution (Figure 5.11e) is 1.75 (Equations 5.22a-c). This value of z is the 4-neighbor *distance* between the directions D^0 and D^1 in Figure 5.6.

$$z=2*0.72+1*0.12+2*0.06+1*0.01+2*0.03 \quad (5.22a)$$

$$z=1.44+0.12+0.12+0.01+0.06 \quad (5.22b)$$

$$z=1.75 \quad (5.22c)$$

5.3 The Similarity Value

Tversky (1977) describes a set-theoretical approach using common and distinct features of two objects to assess the similarity between them. If we compare an object and a direction-relation matrix, an element value in a direction-relation matrix is analogous to an object feature. The commonality C^{01} corresponds to common features, and asymmetric differences R^{01} and R^{10} correspond to distinct features. The commonality contributes to similarity (s) and distinct features contribute to dissimilarity (\mathbf{d}). The result of adding $sum(C^{01})$ and $sum(R^{01})$ is always a constant (i.e., 1); therefore, the result of adding a similarity value and a corresponding dissimilarity value should also be a constant, we use 1.0 for this constant (Equation 5.23).

$$s(D^0, D^1) + \mathbf{d}(D^0, D^1) = 1.0 \quad (5.23)$$

The distance between two directions lies in the interval $[0, dist_{max}]$. We normalize this distance into a dissimilarity value in the closed interval $[0, 1.0]$ (Equation 5.24). The similarity value is complimentary to the dissimilarity value (Equation 5.25).

$$\mathbf{d}(D^0, D^1) = \frac{distance(D^0, D^1)}{distance_{max}} \quad (5.24)$$

$$s(D^0, D^1) = 1.0 - \frac{distance(D^0, D^1)}{distance_{max}} \quad (5.25)$$

A value 1.0 for the similarity between the direction-relation matrices D^0 and D^l implies that the matrices are identical, whereas a similarity value of 0 implies that matrices D^0 and D^l are single-element direction-relation matrices and their non-zero elements are located in the farthest direction tiles. Using the computed distance (Equation 5.22c), the value of similarity for the directions of object B with respect to object A in Scene 0 and Scene 1 in Figure 5.6 is $1.0 - 1.75/4 = 0.5625$.

Applying the similarity assessment method to the query scene in Figure 5.1a and Scenes 0-2 in the database (Figures 5.1b-d), the similarities values are $s(D^q, D^0) = 0.92$, $s(D^q, D^1) = 0.56$, and $s(D^q, D^2) = 0.04$. From these values, we can infer that the direction in the query scene is most similar to the direction in Scene 0, followed by Scene 1, and Scene 2, as the similarity is decreasing in this order.

5.4 Summary

This chapter presented a method to assess similarity between cardinal directions. The scheme to assess similarity needs a distance measure between cardinal directions. We presented a method to compute distance between cardinal directions based on their conceptual neighborhood graph. The distance between two direction-relation matrices is the minimum cost to transform a matrix into another matrix, which can be computed using the transportation algorithm. We use this distance to compute the similarity between two direction-relation matrices. The similarity value between two directions lies in the range 0 to 1.0. The method presented in this chapter for similarity assessment between cardinal directions is useful for the assessment of spatial similarity between scenes. The method developed in this chapter is systematically evaluated in the next chapter.

Chapter 6

Evaluation of the Similarity Assessment Method

The three levels of the direction-relation matrix—coarse, detailed, and deep form a flexible framework for modeling direction relations. So far, however, the assessment of the direction-relation matrix and its compliance with an expected behavior have been purely theoretical. This chapter provides an empirical evaluation of the direction-relation matrix by testing a set of scenarios with the help of a software prototype.

The prototype called *Disha*, a Sanskrit word for direction uses the coarse direction-relation matrix to check consistency of directions, and the detailed direction-relation matrix to assess similarity between cardinal directions. It has a graphical user interface, which allows a user to draw the outlines of the objects for which directions are compared. After a user has drawn the outlines of two ordered pairs of objects, *Disha* computes and displays the similarity value between the directions in each ordered pair. *Disha* allows a user to choose a conceptual neighborhood graph, either the 4-neighbor graph (Figure 5.3) or the 8-neighbor graph (Figure 6.22). It has been developed using an object-oriented development methodology (Sommerville 1996), and has been implemented in Visual C++ 6.0 (Horton 1998; Prosis 1999) under Windows 98 running on a Pentium PC.

Section 6.1 describes the architecture of *Disha*. Section 6.2 discusses implementation issues, such as handling of the rounding errors. Section 6.3 describes user interface components of the system. In Section 6.4, we evaluate the method of similarity assessment. Section 6.5 introduces the 8-neighbor graph an alternated to the 4-neighborhood graph. To test the hypothesis of this thesis, Section 6.6 compares the mappings provided by the 4-neighbor and 8-neighbor graphs. Section 6.7 summarizes the results of this chapter.

6.1 Disha's Architecture

Disha is designed for the maximum reuse of the code; therefore, the project is divided into two parts: platform independent reusable classes in C++ and platform dependent classes for interfacing with MS Windows (Figure 6.1).

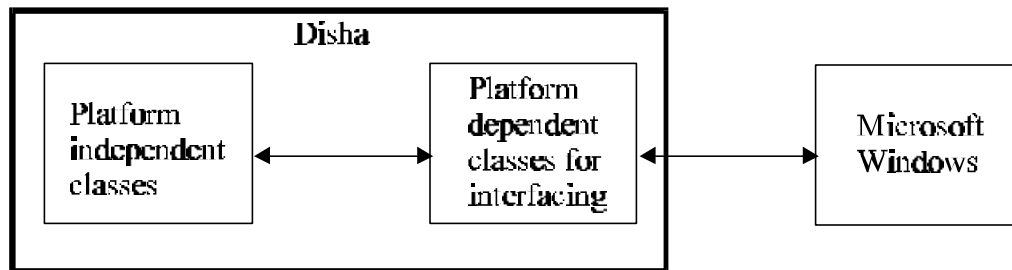


Figure 6.1: Architecture of the prototype Disha.

The prototype is divided into different types of objects, each implemented in an object class having data members and methods. In this chapter, we specify the complete name of a method by appending the class name by the scope resolution operator (`::`) and the method name. For example, the *Draw* method of the *CPolygon* class is written as *CPolygon::Draw*. The prototype has two major types of classes: the classes without components of the user interface are the platform-independent classes (Section 6.1.1) and the classes with user interface components are platform-dependent classes (Section 6.1.2). This separation enables portability such that the code can be transferred to other platforms with minimal modifications.

6.1.1 Platform-Independent Classes

The computational parts of the prototype are implemented in platform-independent classes in C++. These classes are for polygons, direction-relation matrices, and the transportation tableau. The class *HDPolygon* (Paiva 1998) is used for the polygon object. The method *HDPolygon::DirectionMatrix* computes the detailed direction-relation matrix for the direction of the polygon passed as an argument with respect to the calling polygon. The class *CDrm* (Figure 6.2a) implements the direction-relation matrix, which

is computed using the method *HDPolygon::DirectionMatrix*. The class *CDrm* has methods for checking the consistency of the detailed direction-relation matrix.

The class *CDrmGraph* (Figure 6.3) constructs a graph from a direction-relation matrix to check the 4-connectedness and 8-connectedness of the non-zero elements. The class *CDrmGraph* has the class *CVertex* as its data members to represent information about vertices. Each non-empty element in a direction-relation matrix has a vertex in the graph. The method *CountDisjointSets* computes the number of disjoint sets in this graph. If there is only one disjoint set, the graph is connected (Cormen *et al.* 1990). If there is more than one disjoint set, the graph is disconnected. The class *CDeltaMatrix* (Figure 6.2b) implements the direction difference Δ^{01} (Equation 5.10), which is constructed using two direction-relation matrices.

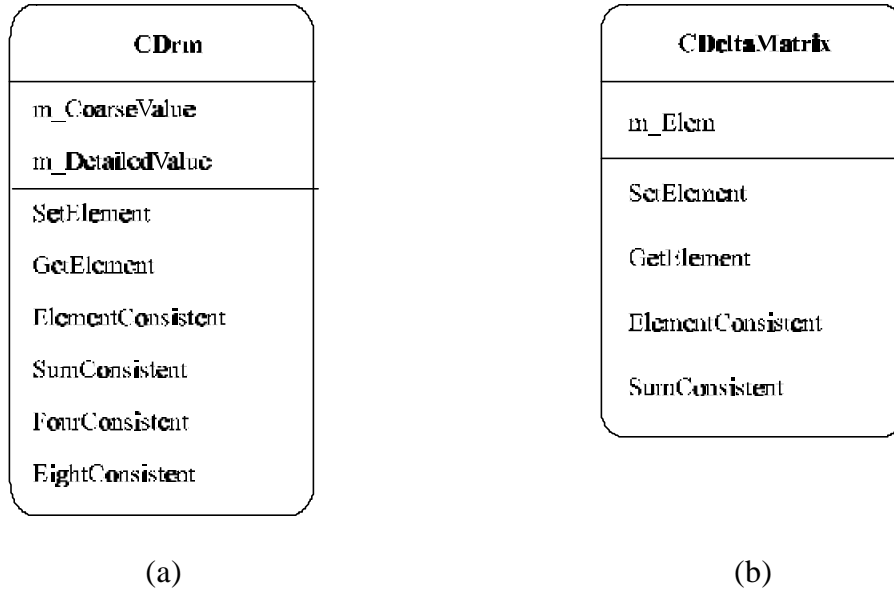


Figure 6.2: The object classes for (a) the direction-relation matrix and (b) the direction-difference matrix.

The transportation tableau for the distance computation problem is implemented in the class *CBTTableau* (Figure 6.4). This class is constructed from the data obtained from the *CDeltaMatrix* class. It has a method to compute the cost of transforming the matrix D^0 into D^1 . The transportation problem is a special case of the standard linear programming problem (Dantzig and Thapa 1997). Due to the wide application of linear

programming problems in engineering and management problems, libraries for solving linear programming problems are commercially available (LINDO 1999). A linear programming problem is solved in two phases, similar to the phases for solving a transportation problem (Section 5.2).

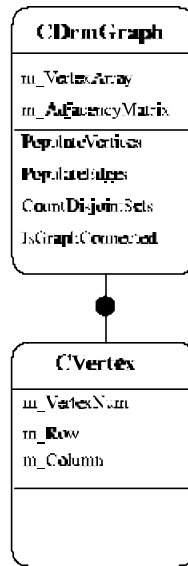


Figure 6.3: The aggregation hierarchy for the classes *CDrmGraph* and *CVertex*.

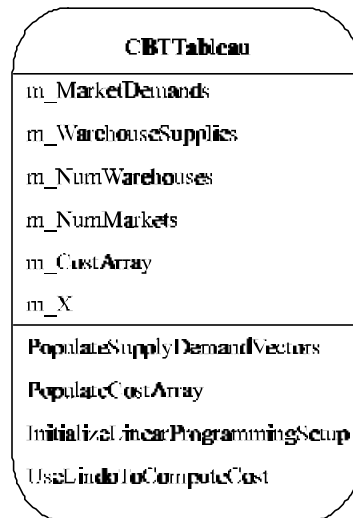


Figure 6.4: The class *CBTableau* for the balanced transportation tableau.

Disha uses the transportation tableau to visualize the cost computation problem and the LINDO library to compute the solution of the corresponding linear programming problem. The method *CBTableau::UseLindoToComputeCost* uses LINDO functions to compute the minimum cost.

6.1.2 Platform-Dependent Classes for the User Interface

This part of the prototype is tailored to Microsoft Windows, using the Microsoft Foundation Class (MFC) library. All classes in the MFC library are derived from the *CObject* class. The *CObject* class has a virtual function *Draw* for drawing on the screen. We implemented the *CElement* class for geometric elements, which is derived from the *CObject* class (Figure 6.5). The *CElement* class inherits the *Draw* function from the *CObject* class. The *CLine*, *CCurve*, and *CPolygon* classes are derived from the *CElement* class. These classes inherit data members and methods of the *CElement* class and define additional data members according to their requirements. The virtual function *Draw* in these classes is used for drawing these geometric elements on the screen. Similarly, the virtual function *Serialize* is used to read and write elements in a file on disk.

The class *CDishaApp* is derived from the class *CWinApp*, which creates the Disha application. We derive the class *CDishaView* from the MFC class *CView*. The class *CDishaView* captures messages from the mouse while drawing the outlines of the extended spatial objects. We also derive the class *CDishaDoc* from the MFC class *CDocument*, which stores all the objects and matrices. The elements in a Disha document are stored in the *m_ElementList*, which is a list of *CElement* type. When a user creates a geometry, the geometry is created in the current type, for instance, a user can create a *CPolygon* type object. After creating this geometric element, it is cast as a *CElement* type object and inserted into the *m_ElementList*. Whenever the document is to be drawn, the *m_ElementList* is traversed and all the elements are drawn using the virtual function *Draw*. To draw a direction-relation matrix on the screen, we define the class *CObDrm*, which is derived from the classes *CDrm* and *CObject* (Figure 6.6).

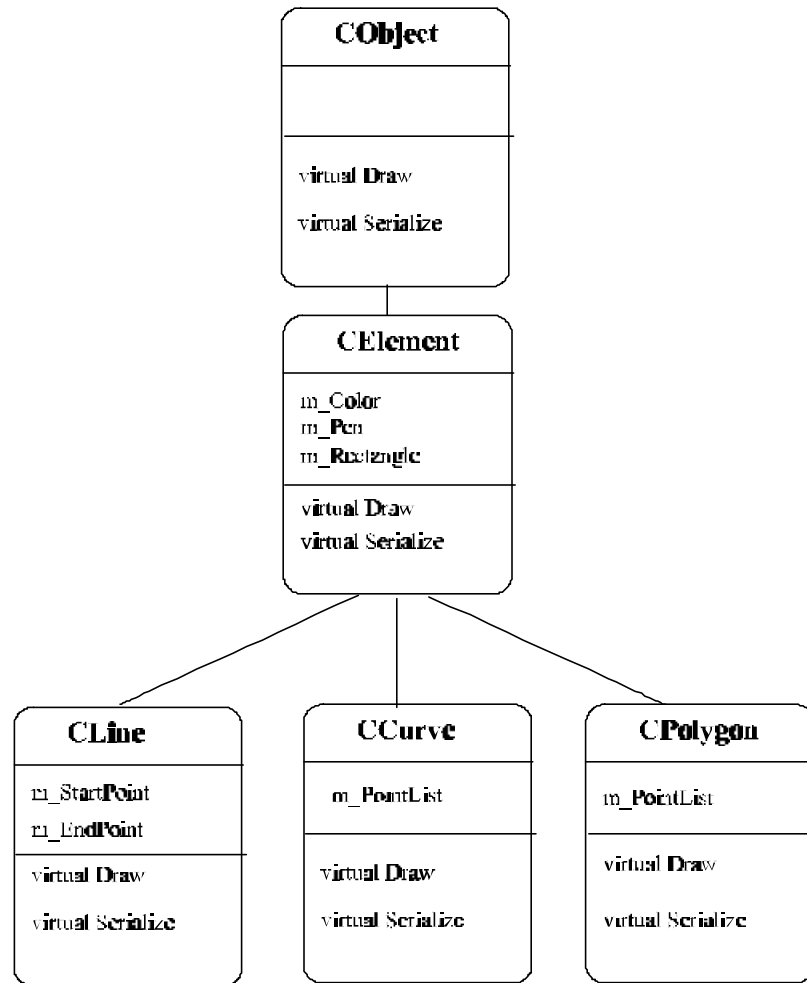


Figure 6.5: Inheritance hierarchy for the classes corresponding to geometric elements.

The class *CObDrm* inherits all data members and functions of the class *CDrm* and implements the virtual function *Draw*. It has an additional data member *m_Name* to store the name of a matrix and methods *SetName* and *GetName* for setting and getting the name of a *CObDrm* object. Similarly, we derive the class *CObDeltaMatrix* from the classes *CDeltaMatrix* and *CObject* (Figure 6.7a), and the class *CObTableau* from the classes *CBTableau* and *CObject* (Figure 6.7b).

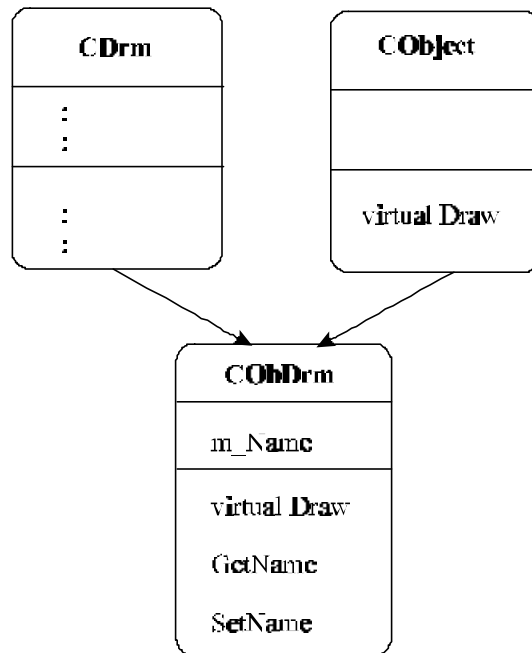


Figure 6.6: The class *CObDrm* inherits from the classes *CDrm* and *CObject*.

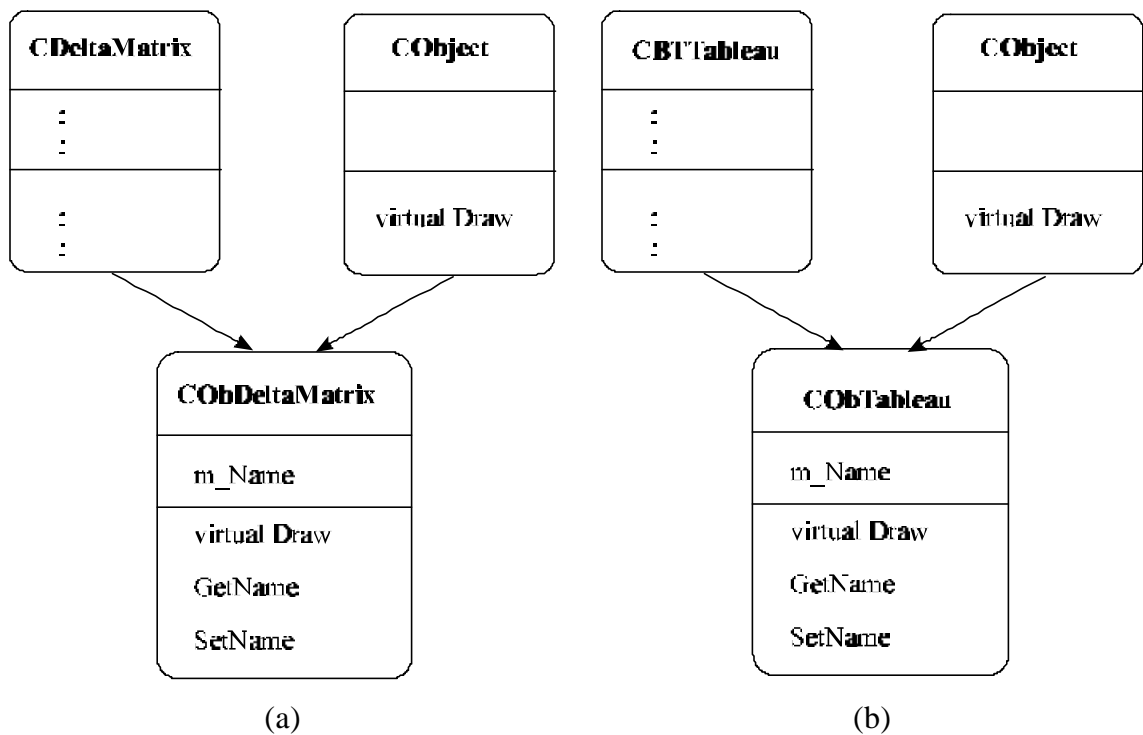


Figure 6.7: Inheritance hierarchy for the classes (a) *CObDeltaMatrix* and (b) *CObTableau*.

We use the methods *CObDrm::Draw* and *CObDeltaMatrix::Draw* to draw the matrices at the screen. The methods *DrawTransportationSetup*, *DrawSimplexSetupAndSolution*, and *DrawTransportationSolution* of the *CObTableau* class are used to draw the problem and solution of the transportation problem and the linear programming problem. The methods *DrawMinimumCostLine* and *DrawSimilarityLine* of the *CObTableau* class display the results on the screen.

6.2 Implementation Issues

The detailed direction-relation matrix records the portion of the target region in a direction tile. The value (x) of an element must satisfy $0 \leq x \leq 1$. The sum of all elements in a detailed direction-relation matrix must be 1. In Disha, there are two methods for creating a direction-relation matrix: (1) drawing the polygonal outline of the objects and computing it using the method *HDPolygon::DirectionMatrix* and (2) entering the values of elements directly. In the case of graphical sketching, the sum of elements may not be exactly 1 due to the rounding of the values of the elements. In the case of directly entering values, the user may enter element values beyond the range or may not do his or her math right. In both cases, Disha checks the consistency of each element and the consistency of the sum of elements. Disha allows a minor tolerance so that the sum may deviate from 1.0. To make a matrix sum-consistent, the difference between 1 and the sum of the elements is distributed proportionately over the non-zero elements. For example, if the tolerance value is 4%, the sum in the range of 0.96 to 1.04 is accepted. A 4% tolerance value is very high for most practical situations; this high value is used here for illustration purpose only. If there are only two non-zero elements with values 0.72 and 0.24 (Figure 6.8), the difference $1 - (0.72 + 0.24) = 0.04$ is distributed over the non-zero elements to yield the values 0.75 and 0.25, respectively. This distribution of difference over non-zero elements has the following features: (1) it leaves zero elements as zero, therefore, does not alter qualitative information and (2) it distributes the difference proportionately among non-zero elements. A sum-consistent detailed direction-relation matrix is an essential prerequisite for the similarity computation.

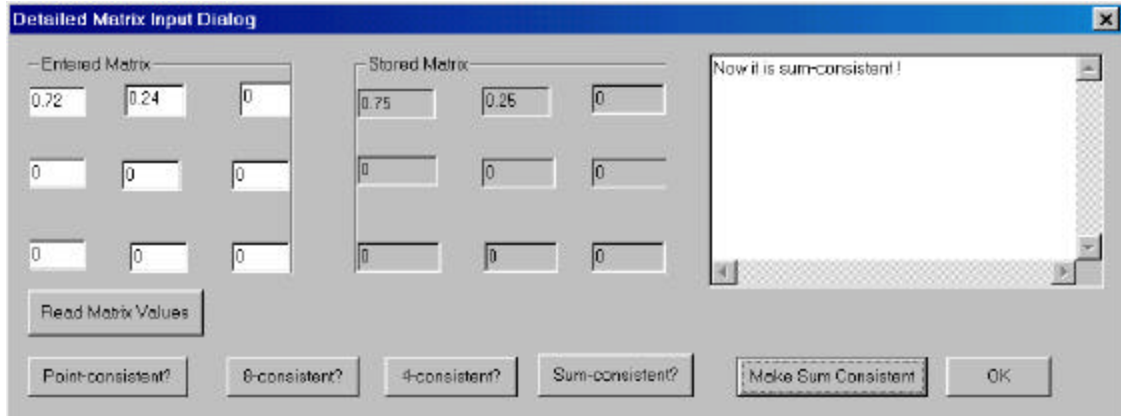


Figure 6.8: Making a detailed direction-relation matrix sum-consistent.

To have a uniform resolution of the values of a detailed direction-relation matrix between 0 and 1, the values of detailed direction-relation matrices are internally represented as integer values. We multiply the values of the detailed elements obtained from the method *HDPolygon::DirectionMatrix* by 100 and round off to yield integer values and apply sum-consistency over the matrix. This in turn gives integer values for warehouse supplies and market demands; however, these values are displayed on the screen as decimal values.

6.3 Disha's Features

Disha has a number of user interface components that use the classes described in Section 6.1. Some of these components are discussed in Sections 6.3.1-6.3.4.

6.3.1 Checking Consistency for Direction-Relation Matrices

Disha checks the consistency of matrices before using them in similarity computations. To illustrate the consistency checking of matrices, we have implemented the *coarse matrix input dialog* (Figure 6.9) and the *detailed matrix input dialog* (Figure 6.8) for coarse and detailed direction-relation matrices, respectively. The coarse input dialog allows a user to enter a coarse direction-relation matrix by clicking the checkboxes corresponding to each element. A checked box selects a non-empty element and an unchecked box corresponds to an empty element.

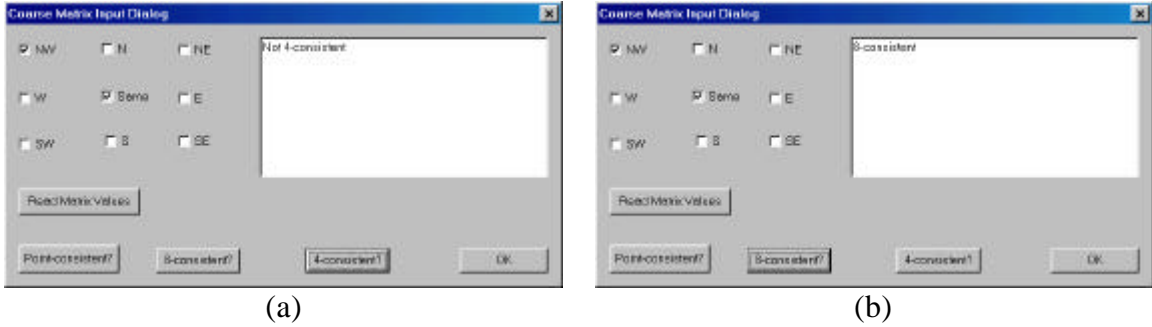


Figure 6.9: The coarse matrix input dialog for checking the consistency of a coarse direction-relation matrix.

Once a matrix is entered, it can be checked for 4-consistency, 8-consistency, and point-consistency, which apply to region, line, and point target objects, respectively. A matrix is point-consistent if it has only one non-empty element. In order to check the point-consistency, we count the number of non-empty elements. Four-consistency and eight-consistency checks are based on the respective type of connectedness of non-empty elements. For example, a matrix having only two diagonal non-empty elements is not 4-consistent (Figure 6.9a) but it is 8-consistent (Figure 6.9b).

In the case of detailed direction-relation matrices, there are additional constraints: (1) the value of an element must be between 0 and 1, and (2) the sum of all elements must be 1. The detailed matrix input dialog (Figure 6.8) allows a user to enter a detailed direction-relation matrix and offers buttons to check the following consistencies: point-consistency, 4-consistency, 8-consistency, and sum-consistency. The sum-consistency also checks the element consistency for all elements. Additionally, it has a button with which a user can make a matrix sum consistent.

6.3.2 Determining the Number of Consistent Configurations

The prototype implementation was also used to confirm the number of consistent configurations (Section 3.2.2). For a 3x3 direction-relation matrix, there are 512 possible configurations, but not all of these configurations are consistent. To count the number of consistent configurations for a 3x3 matrix, Disha loops through all 512 matrices and checks whether this configuration is consistent or not for the considered types of

consistency (Figure 6.10). Out of these 512, there are only 218 four-consistent matrices, 388 eight-consistent matrices, and 9 point-consistent matrices.

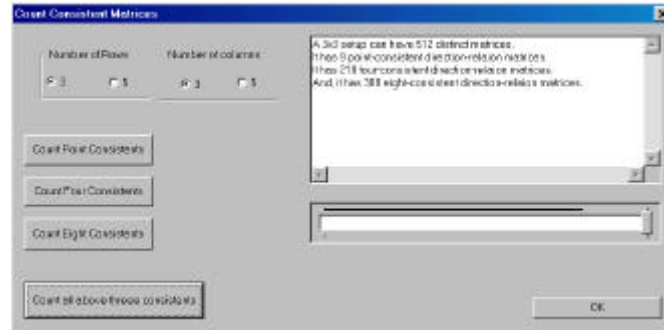


Figure 6.10: Counting consistent configurations.

6.3.3 Direction-Similarity from a Direction-Difference Matrix

To compute the distance between two direction-relation matrices from a direction-difference, Disha has a *delta matrix input dialog* (Figure 6.11). It allows a user to enter the values of elements in a direction-difference matrix and checks the consistency of the difference matrix. For a direction-difference, there are two consistency constraints: (1) each element's value (x) must satisfy $-1 \leq x \leq 1$ and (2) the sum of all elements must be 0. A user can select either the 4-neighborhood or the 8-neighborhood graph by pressing the respective radio button in the dialog box. Disha computes the similarity using the method described in Chapter 5 and displays the results in the textbox of the dialog. Disha displays the values of dissimilarity and similarity in the range of $[0, 100]$.

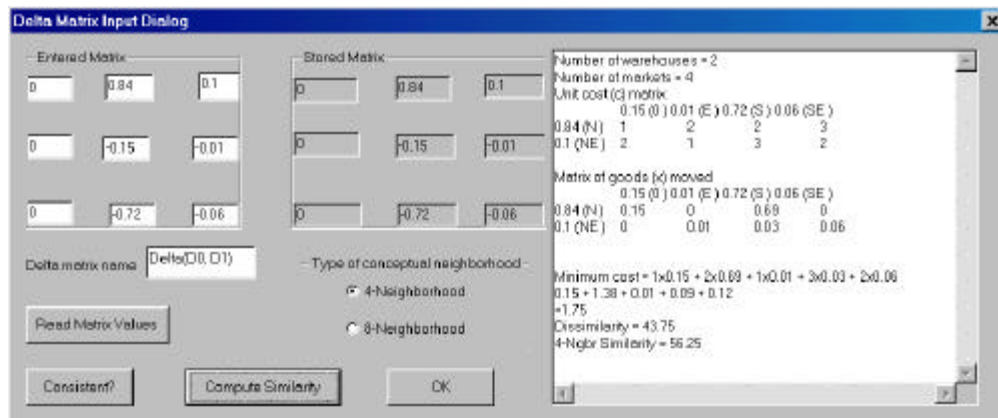


Figure 6.11: Computing the distance from a direction-difference matrix.

6.3.4 Direction-Similarity between Two Ordered Pairs of Extended Objects

Disha has a graphical user interface, which allows a user to draw the outlines of two pairs of extended objects (Figure 6.12). The user can translate, scale, and rotate these objects using the tools provided in Disha. To compute the similarity value between directions, it computes detailed direction-relation matrices and the direction-difference (Figure 6.13a). The transportation tableau (Figure 6.13b) is constructed from this direction-difference. Since we are using the LINDO libraries to determine the minimum cost of transforming a matrix D^0 into D^1 , we generate a linear programming problem from this transportation problem (Figure 6.14). The LINDO libraries compute an optimum solution for the problem (Figure 6.15), which is the solution of the transportation problem (Figure 6.16). From this solution, Disha computes the minimum cost and the values for dissimilarity and similarity (Figure 6.17).

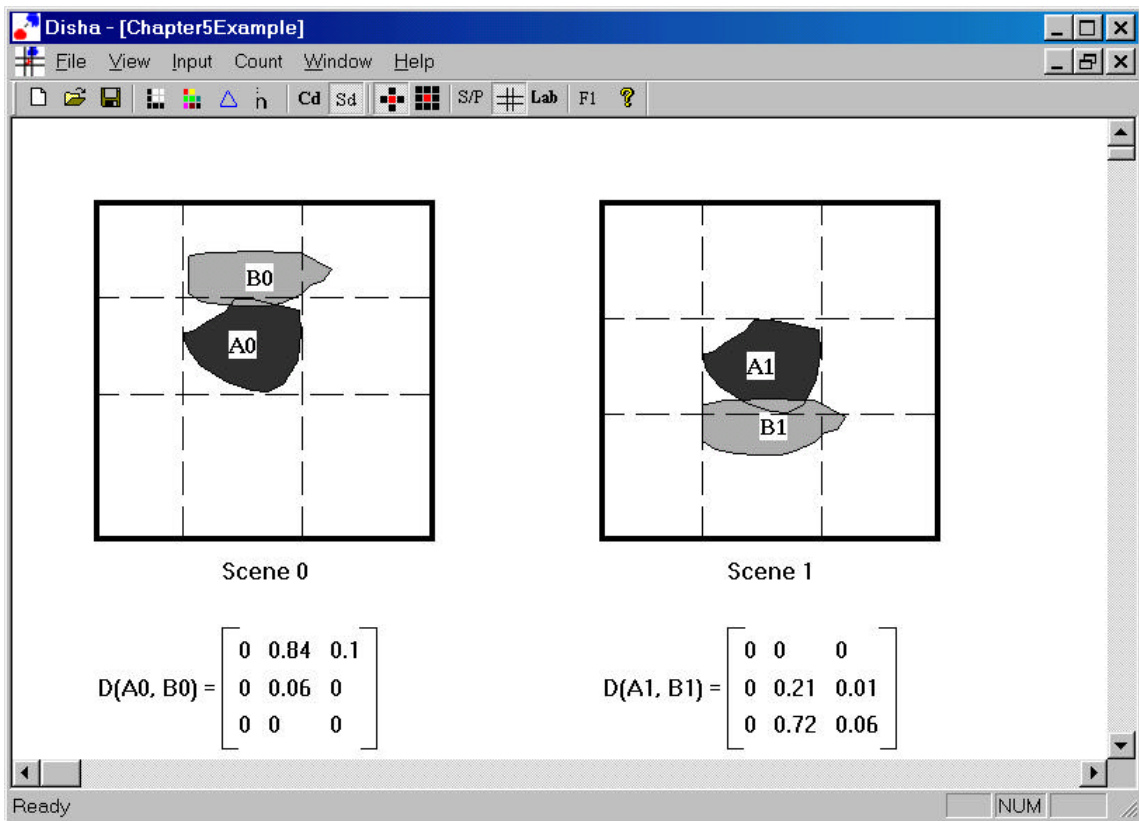


Figure 6.12: A snapshot of Disha with two pairs of objects and their direction-relation matrices.

$$\text{Delta}(D(A0, B0), D(A1, B1)) = \begin{bmatrix} 0 & 0.84 & 0.1 \\ 0 & -0.15 & -0.01 \\ 0 & -0.72 & -0.06 \end{bmatrix}$$

(a)

	0	E	S	SE	
N	1	2	2	3	0.84
NE	2	1	3	2	0.1
	0.15	0.01	0.72	0.06	0.94

(b)

Figure 6.13: (a) The direction difference matrix and (b) the transportation tableau using the 4-neighbor graph.

Number of constraints = 6

Number of variables = 8

$$X1 + 2X2 + 2X3 + 3X4 + 2X5 + X6 + 3X7 + 2X8 = z \text{ (min)}$$

such that

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X1 \\ X2 \\ X3 \\ X4 \\ X5 \\ X6 \\ X7 \\ X8 \end{bmatrix} = \begin{bmatrix} 0.84 \\ 0.1 \\ 0.15 \\ 0.01 \\ 0.72 \\ 0.06 \end{bmatrix}$$

Figure 6.14: The linear programming problem for the transportation problem in Figure 6.13b.

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0.15 \\ 0 \\ 0.69 \\ 0 \\ 0 \\ 0.01 \\ 0.03 \\ 0.06 \end{bmatrix} = \begin{bmatrix} 0.84 \\ 0.1 \\ 0.15 \\ 0.01 \\ 0.72 \\ 0.06 \end{bmatrix}$$

Figure 6.15: A solution for the linear programming problem in Figure 6.14 using LINDO (1999).

	0	E	S	SE	
N	0.15 ①	2	0.69 ②	3	0.84
NE	2	0.01 ①	0.03 ③	0.06 ②	0.1
	0.15	0.01	0.72	0.06	0.94

Figure 6.16: A solution of the transportation problem in Figure 6.13b.

$$\begin{aligned}
 \text{Minimum cost} &= 1 \times 0.15 + 2 \times 0.69 + 1 \times 0.01 + 3 \times 0.03 + 2 \times 0.06 \\
 &= 0.15 + 1.38 + 0.01 + 0.09 + 0.12 \\
 \text{Minimum cost} &= 1.75 \\
 \text{Dissimilarity} &= 43.75 \\
 \text{4-Neighbor Similarity} &= 56.25
 \end{aligned}$$

Figure: 6.17: Computing the similarity value from the transportation problem solution in Figure 6.16.

6.4 Systematic Evaluation of the Method

Similarity assessment is the estimation of deviation from equivalence (Bruns and Egenhofer 1996). The dissimilarity value between two direction relations records the magnitude of deviation from equivalence, and the similarity value is the complement of the dissimilarity value (Chapter 5).

To evaluate systematically the method for similarity assessment, we start with a query scene containing a pair of reference and target objects, and generate database scenes by gradually changing the target object. The variations in the target object change the direction of the target object with respect to the reference object. We study the changes in similarity between the direction in the query scene and the directions in database scenes. We consider two types of changes in the target object: (1) movement

(Section 6.4.1) and (2) rotation (Section 6.4.2) of target object with respect to reference object. These types of changes were selected because they establish a clear baseline for the expected similarity value. A cognitively plausible similarity assessment method is expected to give the highest similarity value (100) of similarity to the scene without any change and decreasing values of similarity with increasing changes.

6.4.1 Moving the Target Object Over the Reference Object

The target object *B* in the query scene (Figure 6.18) is to the northeast of the reference object *A*. Scene 0 is identical to the query scene, and Scenes 1-7 are generated by moving object *B* diagonally from the northeast to the southwest tile. The number of a scene is subscripted to the objects' labels in the scene, and the values of 4-neighbor similarity between the direction in the query scene and the direction in a scene is written under the respective database scene.

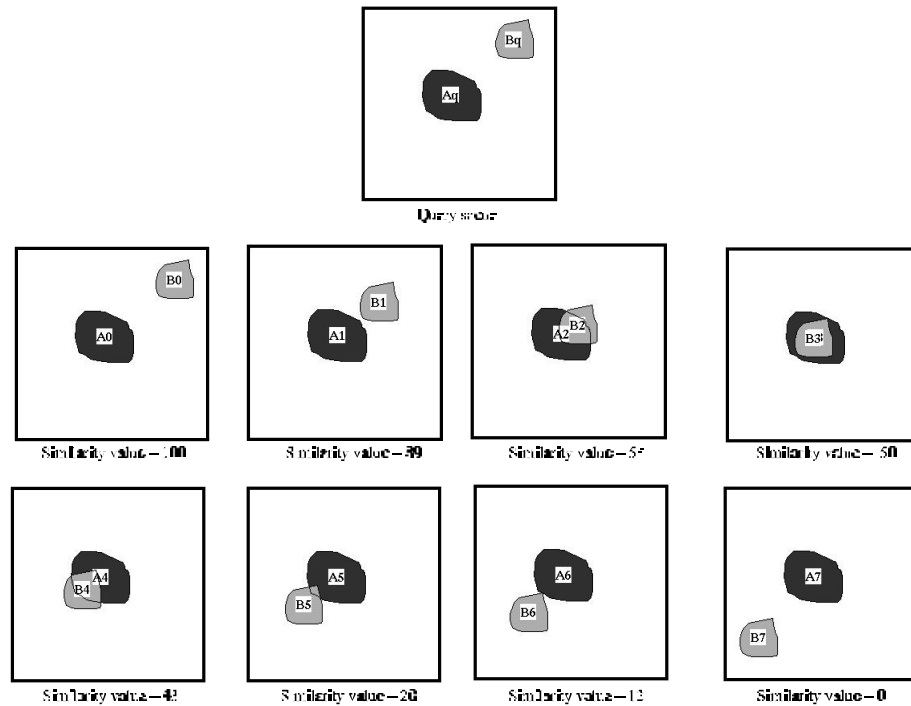


Figure 6.18: Scenes generated by moving the target object over the reference object.

The similarity between the direction in the query scene and directions in Scenes 0-7 decreases as the degree of change gets larger, which is as expected (Figure 6.19). If object *B* in Scene 7 were moved further southwest, no further decrease in direction-similarity could be detected. To distinguish such scenarios, metric properties would need to be employed, much like the enhancement of topological properties with metric (Egenhofer and Shariff 1998).

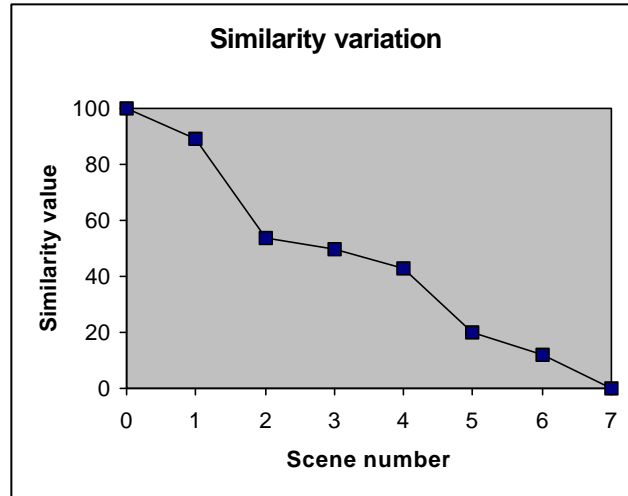


Figure 6.19: The pattern of similarity values for Scenes 0-7 in Figure 6.18.

6.4.2 Rotating the Target Object Around the Reference Object

In another case of similarity assessment we compute the similarity between the direction in the query scene and directions in Scenes 0-16 that are generated by rotating the target object around the reference object (Figure 6.20). This clockwise rotation starts in the northwest tile, goes for a full circle so that it ends again in the northwest tile. The similarity between the direction in the query scene and the database scenes decreases from Scene 0 to Scene 8 as the rotation takes object *B* to the tile that is farthest from the northwest tile (Figure 6.21). The similarity value is 0 in Scene 8, as the southeast tile is the farthest tile from the northwest tile. As object *B* moves further clockwise, the similarity value increases, because object *B* gets closer to the northwest tile. The similarity value becomes 100 when object *B* is back in the northwest tile (Scene 16). To reach the value 100 for similarity, object *B* in Scene 16 need not be exactly at the same location as object *B* in Scene 0, but in the same direction tile. This pattern of similarity

values is cognitively plausible, because it matches the expected behavior of a full circle rotation.



Figure 6.20: Scenes generated by moving the target object around the reference object.

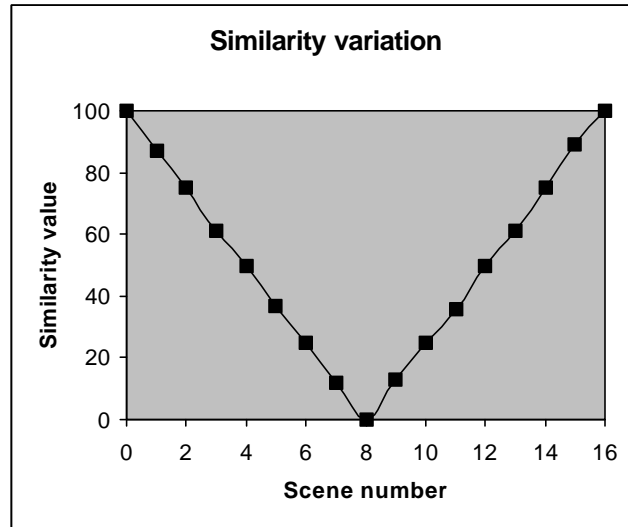


Figure 6.21: The pattern of similarity values for Scenes 0-16 in Figure 6.20.

6.5 Eight-Nighbor Conceptual Graph

In Chapter 5, we described the 4-neighbor conceptual neighborhood graph (Figure 5.3). Another alternate is the 8-neighbor conceptual neighborhood graph, which has additional edges for diagonally adjacent pairs of tiles (Figure 6.22).

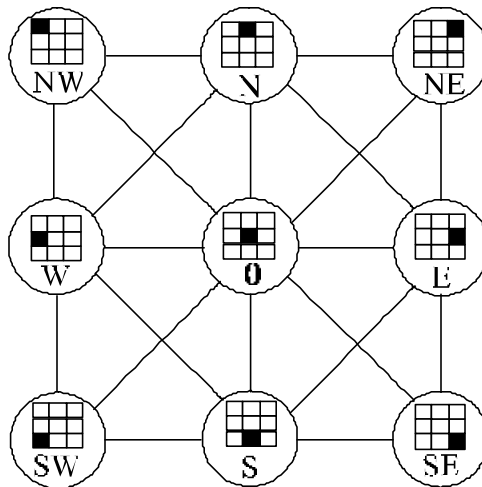


Figure 6.22: The conceptual neighborhood graph for nine cardinal directions based on the 8-neighborhood between tiles.

The maximum 8-neighbor distance between two directions is 2 (Figure 6.23); therefore, the range of the distance using 8-neighbor graph is 0-2. The next section compares the soundness of the similarity assessment provided by both the graphs.



















									
	N	NE	E	SE	S	SW	W	NW	0
 N	0	1	1	2	2	2	1	1	1
 NE	1	0	1	2	2	2	2	2	1
 E	1	1	0	1	1	2	2	2	1
 SE	2	2	1	0	1	2	2	2	1
 S	2	2	1	1	0	1	1	2	1
 SW	2	2	2	2	1	0	1	2	1
 W	1	2	2	2	1	1	0	1	1
 NW	1	2	2	2	2	2	1	0	1
 0	1	1	1	1	1	1	1	1	0

Figure 6.23: Eight-neighbor distances between cardinal directions for regions.

6.6 Comparison of the Soundness of the Mappings Provided by Four-Neighbor and Eight-Neighbor Graphs

There are two types of neighborhood graphs that can be used to compute distances between cardinal directions: the 4-neighborhood and 8-neighborhood graphs. A *sound mapping* would give decreasing similarity values for increasing larger direction changes. This section compares the soundness of the mappings from direction changes onto the similarity values provided by these graphs. The hypothesis (Section 1.4.2) of this thesis is about the soundness of mappings provided by these graphs. The hypothesis is:

“The four-neighborhood and eight-neighborhood graphs provide equally sound mappings of direction changes onto similarity values.”

To test the hypothesis, we study the similarity profiles and compare the rankings provided by both the 4-neighborhood and 8-neighborhood graphs. We consider nine different scenarios of gradual changes in the target object and compute the values of 4-neighbor and 8-neighbor similarities between the query scene and the scenes generated by the gradual changes (Sections 6.6.1-9). The neighborhood graph that has higher number of monotonic similarity profiles provides a sounder mapping. Section 6.6.10 discusses the results of this comparison and summarizes the test of hypothesis.

6.6.1 Curved Movement of a Disjoint Target Object

In this scenario, the target object is disjoint with respect to the reference object, and we generate database scenes by rotating the target object around the reference object (Figure 6.24). This clockwise rotation starts in the northwest tile, and ends in the southeast tile.

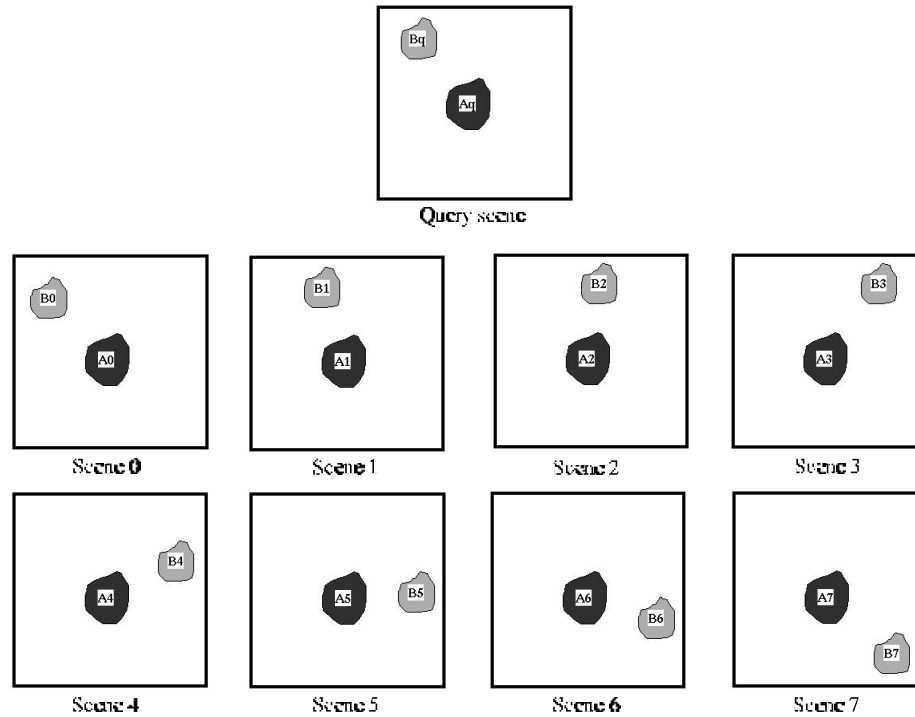


Figure 6.24: The scenes generated by curved movement of the disjoint target object.

The 8-neighbor similarity is identical for Scenes 3-7 (Table 6.1; Figure 6.25). We can rank these scenes for their similarity with the query scene such that the most similar scene is at rank 1. The 8-neighbor graph will give the Scenes 0-2 ranks 1-3, and Scenes 3-7 will be placed at rank 4; however, the 4-neighbor graph will give Scenes 0-7 ranks 1-8, respectively. The similarity values in this scenario reveal the following characteristics of the graphs: (1) the 4-neighbor graph and the 8-neighbor graph can detect small changes, (2) the 4-neighbor graph continues to distinguish between large changes, and (3) the 8-neighbor graph fails to distinguish between large changes. Therefore, in this scenario, the 4-neighborhood provides a somewhat sounder mapping than the 8-neighborhood graph.

Scene	4-neighbor		8-neighbor	
	similarity	rank	similarity	rank
0	100	1	100	1
1	86	2	72	2
2	70	3	40	3
3	50	4	0	4
4	44	5	0	4
5	25	6	0	4
6	15	7	0	4
7	0	8	0	4

Table 6.1: Similarities and ranks of Scenes 0-7 in Figure 6.24.

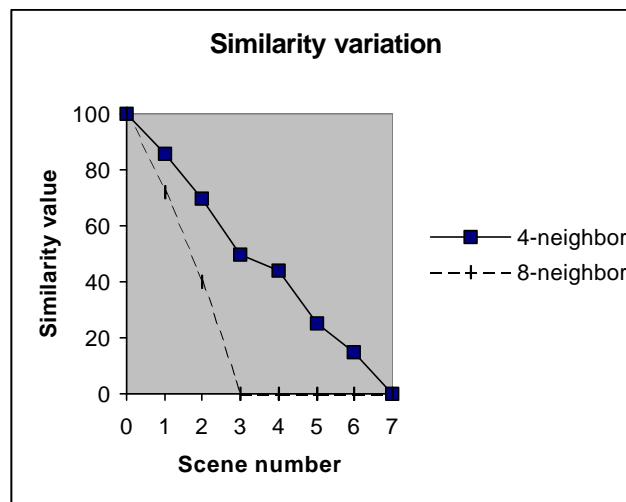


Figure 6.25: The pattern of similarity values for Scenes 0-7 in Figure 6.24.

6.6.2 Curved Movement of an Overlapping Target Object

In this scenario, the target object overlaps with the reference object, and we generate database scenes by rotating the target object around the reference object (Figure 6.26). The rotation is clockwise and starts from the northwest tile and ends at the southeast tile.

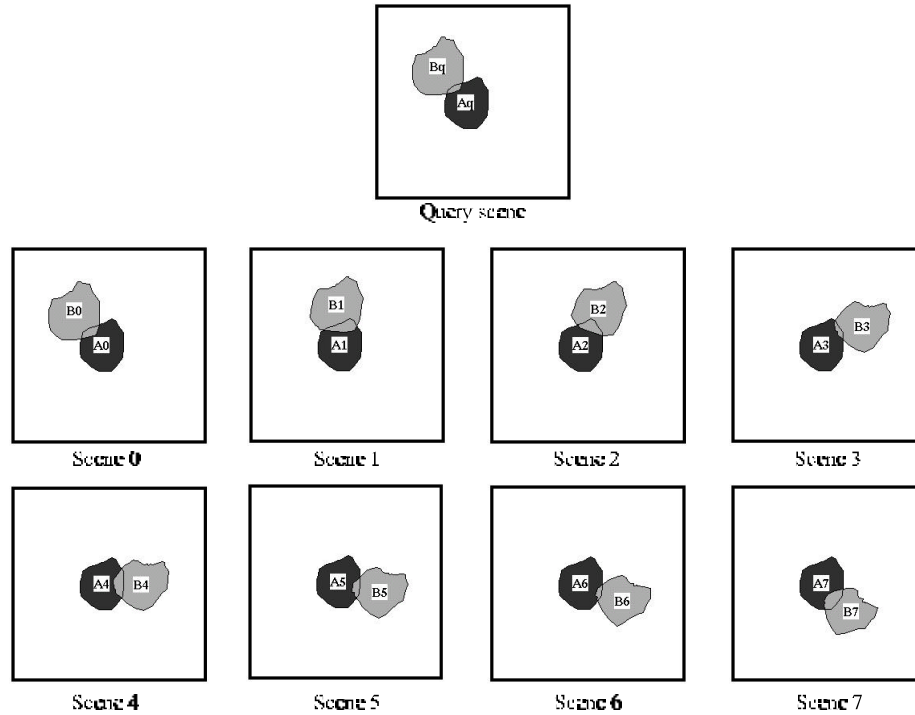


Figure 6.26: The scenes generated by curved movement of the overlapping target object.

In this scenario, the value for the 4-neighbor similarity of Scene 0-7 decreases monotonically (Table 6.2; Figure 6.27), which is expected from a cognitively plausible graph.

Scene	4-neighbor		8-neighbor	
	similarity	rank	similarity	rank
0	98	1	98	1
1	83	2	72	2
2	71	3	46	3
3	53	4	22	6
4	45	5	24	5
5	42	6	21	7
6	39	7	21	7
7	38	8	37	4

Table 6.2: Similarities and ranks of Scenes 0-7 in Figure 6.26.

The values of 8-neighbor similarity, however, decrease from Scene 0 to Scene 3; then Scene 4 has a similarity value that is higher than the similarity value for Scene 3; Scenes 5 and 6 have the same similarity value; and the similarity value for Scene 7 is more than the similarity value for Scene 6. The pattern of the 8-neighbor similarity values for Scenes 4-7 does not match with human intuition. Additionally, the 8-neighbor similarity for Scene 7 is higher than the similarity for Scenes 3-6, which is counter intuitive.

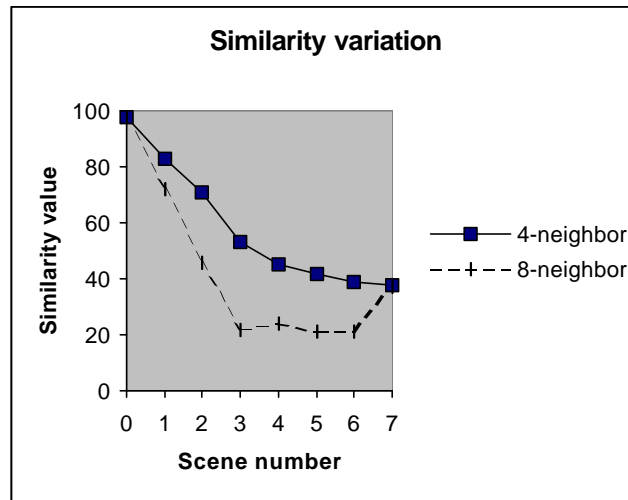


Figure 6.27: The pattern of similarity values for Scenes 0-7 in Figure 6.26.

6.6.3 Diagonal Movement of a Larger Target Object

In this scenario, the target object B is larger than the reference object A , and the database scenes are generated by moving B diagonally from the northwest tile to the southeast tile (Figure 6.28). Both the 4-neighbor graph and 8-neighbor graph rank Scenes 0-4 in the same order (Table 6.3; Figure 6.29); therefore, these mappings are equally sound.

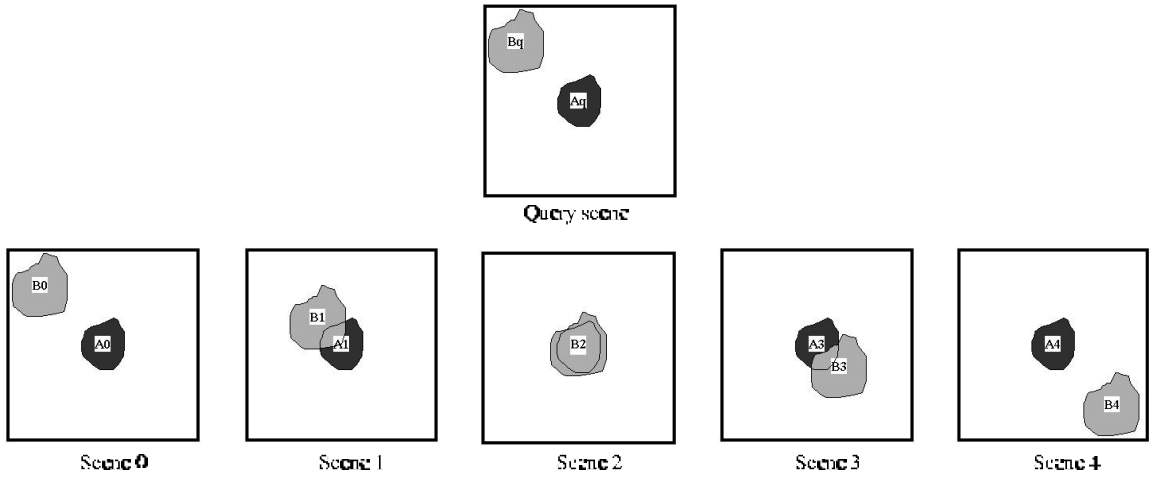


Figure 6.28: The scenes generated by diagonal movement of the larger target object.

Scene	4-neighbor		8-neighbor	
	similarity	rank	similarity	rank
0	100	1	100	1
1	74	2	60	2
2	51	3	46	3
3	24	4	10	4
4	0	5	0	5

Table 6.3: Similarities and ranks of Scenes 0-4 in Figure 6.28.

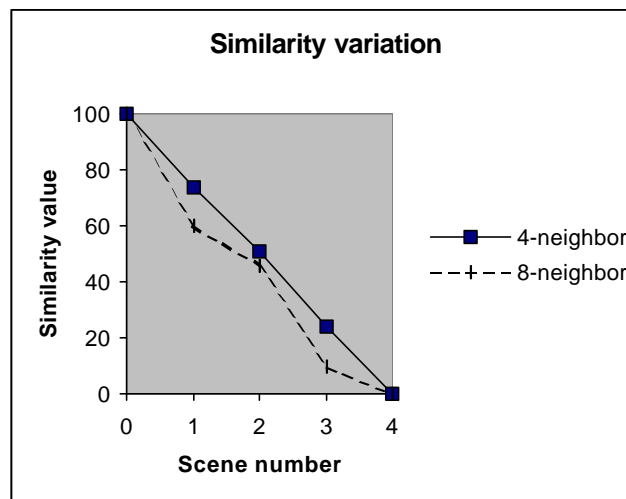


Figure 6.29: The pattern of similarity values for Scenes 0-4 in Figure 6.28.

6.6.4 Diagonal Movement of a Smaller Target Object

In this scenario, the target object B is smaller than the reference object A (Figure 6.30). We generate database scenes by moving object B diagonally from the northwest tile to the southeast tile. In this scenario also both the 4-neighborhood and 8-neighborhood graphs rank Scenes 0-4 in the same order (Table 6.4; Figure 6.31).

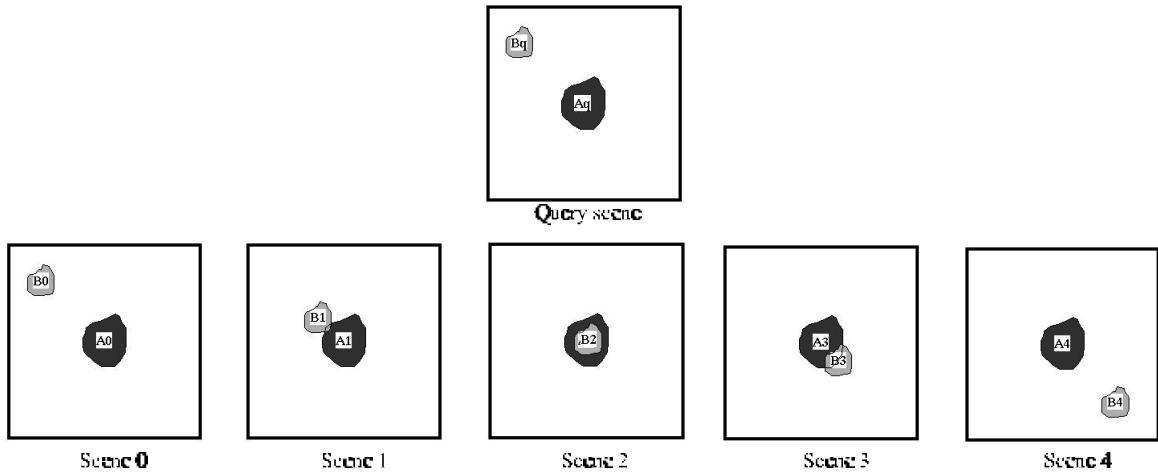


Figure 6.30: The scenes generated by diagonal movement of the smaller target object.

Scene	4-neighbor		8-neighbor	
	similarity	rank	similarity	rank
0	100	1	100	1
1	74	2	59	2
2	50	3	50	3
3	36	4	25	4
4	0	5	0	5

Table 6.4: Similarities and ranks of Scenes 0-4 in Figure 6.30.

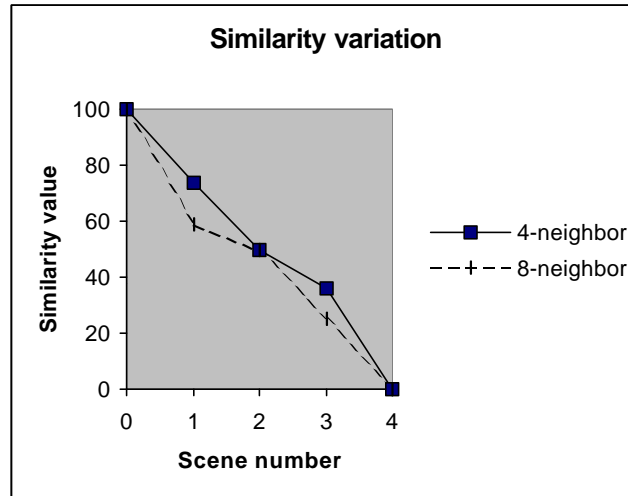


Figure 6.31: The pattern of similarity values for Scenes 0-4 in Figure 6.30.

6.6.5 Vertical Movement of a Larger Target Object

In this scenario, the target object *B* is larger than the reference object *A*, and the database scenes are generated by moving object *B* vertically from the north tile to the south tile (Figure 6.32).

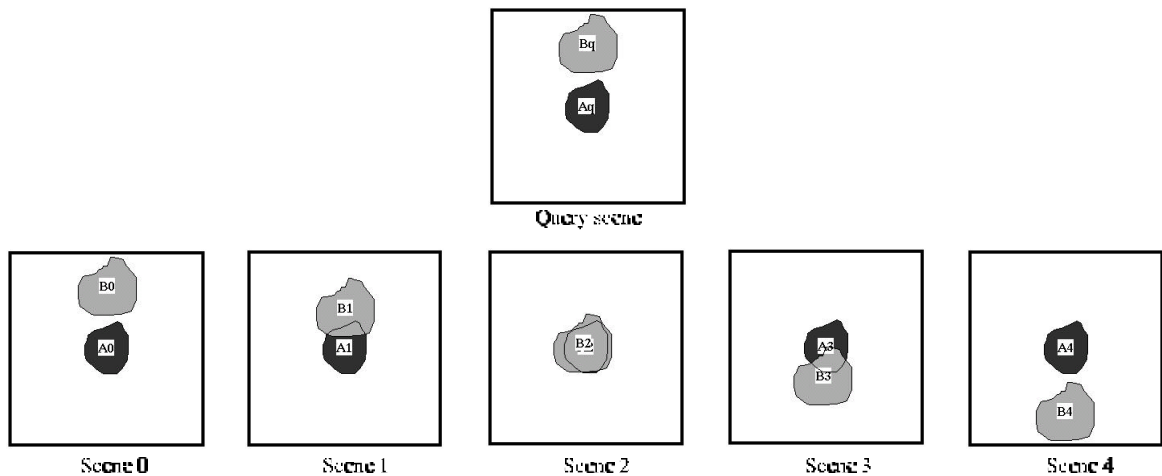


Figure 6.32: The scenes generated by vertical movement of the larger target object.

In this case, both graphs rank Scenes 0-4 in the same order (Table 6.5; Figure 6.33). For Scene 4, however, the value of the 8-neighbor similarity is 0 while the value of 4-neighbor similarity is 48. The value 48 in the case of 4-neighbor graph implies that there

is more room for distinguishing additional changes, which is not the case for 8-neighbor graph.

Scene	4-neighbor		8-neighbor	
	similarity	rank	similarity	rank
0	100	1	100	1
1	93	2	87	2
2	73	3	52	3
3	55	4	15	4
4	48	5	0	5

Table 6.5: Similarities and ranks of Scenes 0-4 in Figure 6.32

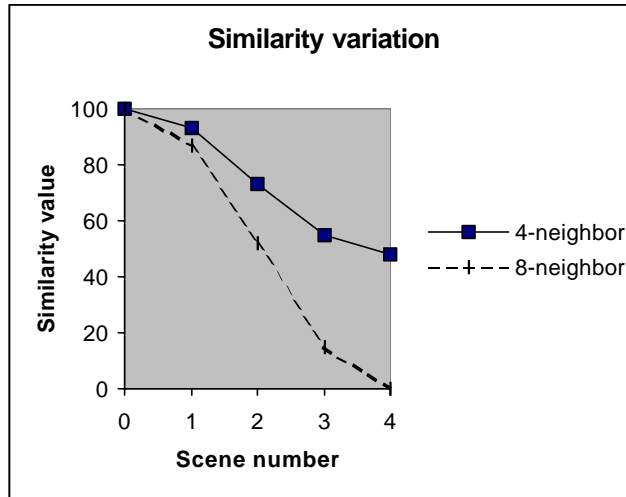


Figure 6.33: The pattern of similarity values for Scenes 0-4 in Figure 6.32.

6.6.6 Vertical Movement of a Smaller Target Object

In this scenario, the target object B is smaller than the reference object A , and we generate the database scenes by moving object B vertically from the north tile to the south tile (Figure 6.34). In this case, the ranking of the scenes using both the 4-neighbor and 8-neighbor graphs is identical (Table 6.6; Figure 6.35).

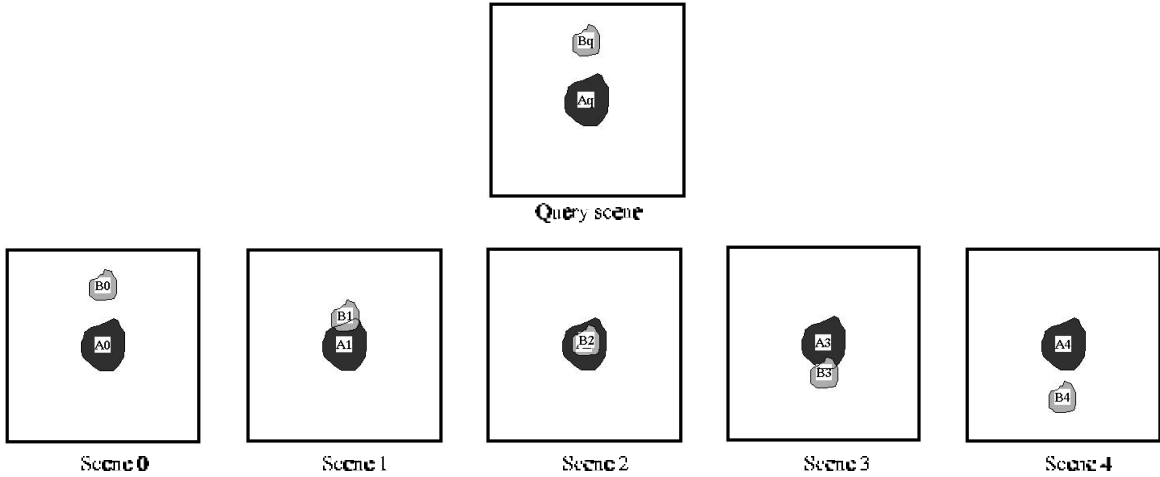


Figure 6.34: The scenes generated by vertical movement of the smaller target object.

Scene	4-neighbor		8-neighbor	
	similarity	rank	similarity	rank
0	100	1	100	1
1	88	2	76	2
2	75	3	50	3
3	56	4	12	4
4	50	5	0	5

Table 6.6: Similarities and ranks of Scenes 0-4 in Figure 6.34.

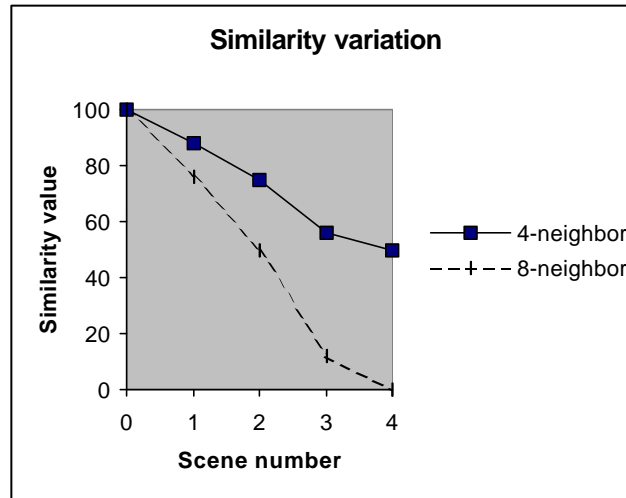


Figure 6.35: The pattern of similarity values for Scenes 0-4 in Figure 6.34.

6.6.7 Scaling up the Target Object in the Northwest Tile

In this scenario, the target object B is in the northwest tile and we enlarge it such that it expands up to the southeast tile (Figure 6.36). In this case, the ranking of the scenes is identical (Table 6.7; Figure 6.37) for both the 4-neighbor and the 8-neighbor graphs.

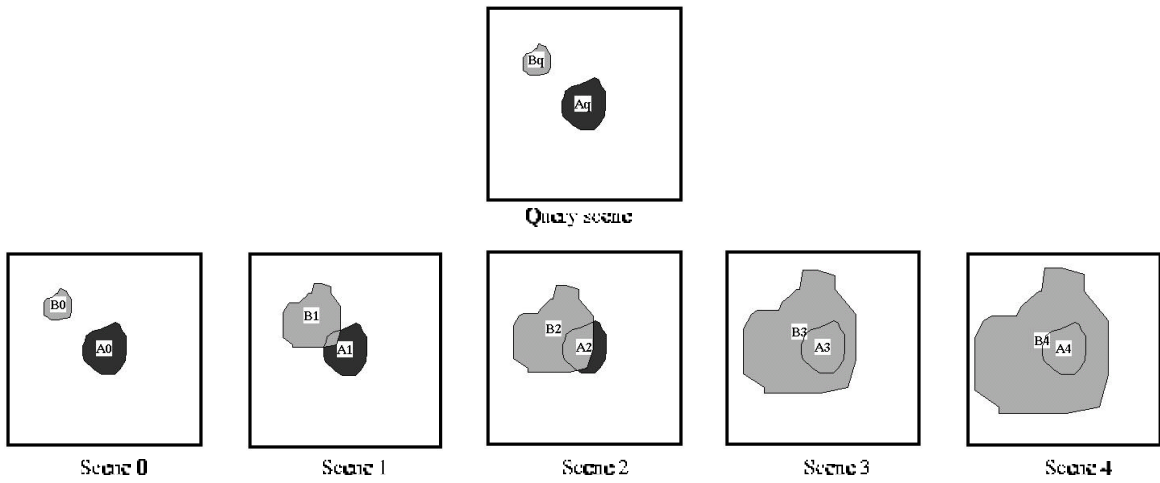


Figure 6.36: The scenes generated by enlarging the target object in the northwest tile.

Scene	4-neighbor		8-neighbor	
	similarity	rank	similarity	rank
0	100	1	100	1
1	83	2	71	2
2	72	3	56	3
3	64	4	43	4
4	56	5	33	5

Table 6.7: Similarities and ranks of Scenes 0-4 in Figure 6.36.

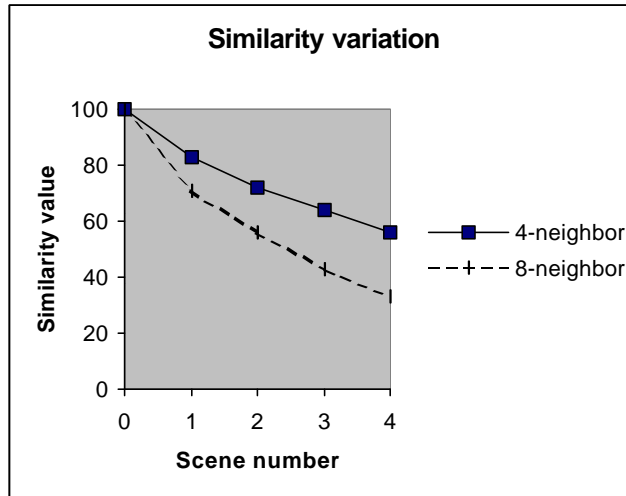


Figure 6.37: The pattern of similarity values for Scenes 0-4 in Figure 6.36.

6.6.8 Scaling up the Target Object in the North Tile

In this scenario, the target object B is in the north tile, and we enlarge it such that it expands up to the south tile (Figure 6.38). In this case, the rankings of the scenes using 4-neighbor similarities and 8-neighbor similarities are identical (Table 6.8; Figure 6.39).

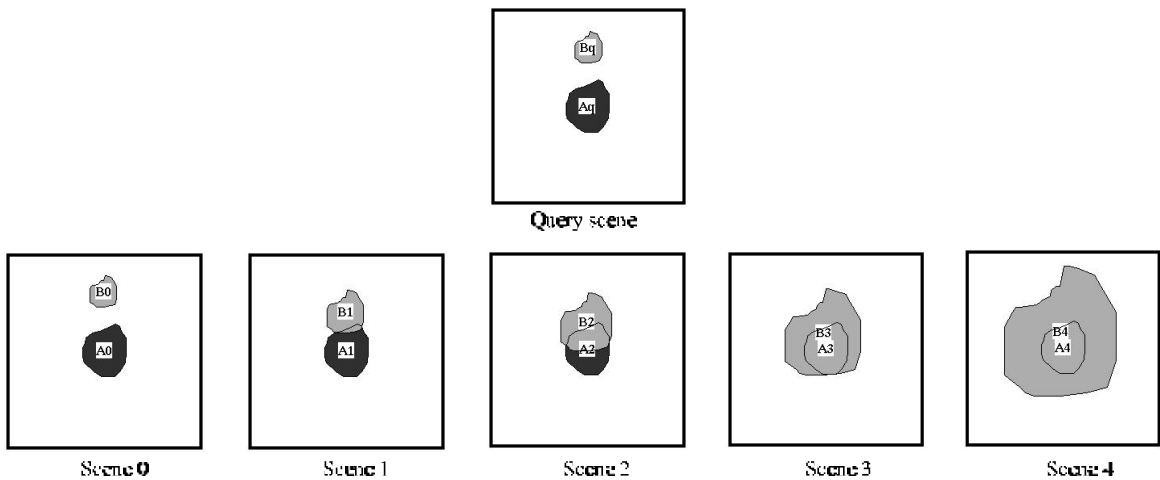


Figure 6.38: The scenes generated by enlarging the target object in the north tile.

Scene	4-neighbor		8-neighbor	
	similarity	rank	similarity	rank
0	100	1	100	1
1	95	2	89	2
2	84	3	70	3
3	73	4	60	4
4	65	5	50	5

Table 6.8: Similarities and ranks of Scenes 0-4 in Figure 6.38.

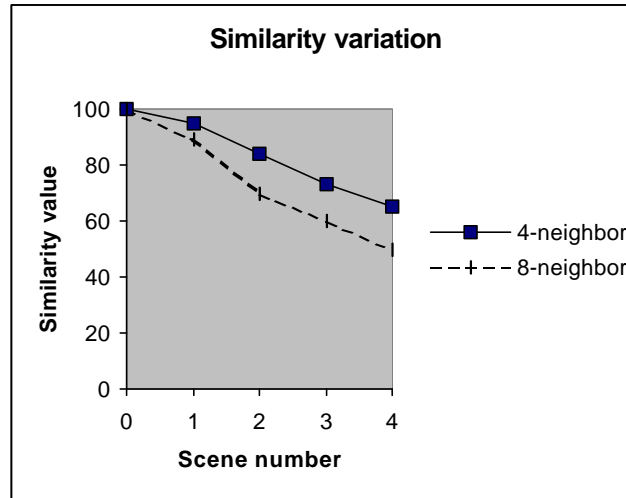


Figure 6.39: The pattern of similarity values for Scenes 0-4 in Figure 6.38.

6.6.9 Rotation of the Target Object

In this scenario, the target object B is in the northwest tile, and it is rotated clockwise with respect to its southeast extreme (Figure 6.40). The similarity value in the case of 4-neighbor graph decrease monotonically from Scene 0 to Scene 4 and increase monotonically from Scene 4 to Scene 7 (Table 6.9; Figure 6.41); however, in the case of 8-neighbor graph it decrease from Scene 0 to Scene 3, increase from Scene 3 to Scene 4, decreases from Scene 4 to Scene 5, and increases from Scene 5 to Scene 7 again. The similarity profile in the case of 4-neighbor graph reflects a sounder mapping of direction changes onto similarity values than the 8-neighbor graph.

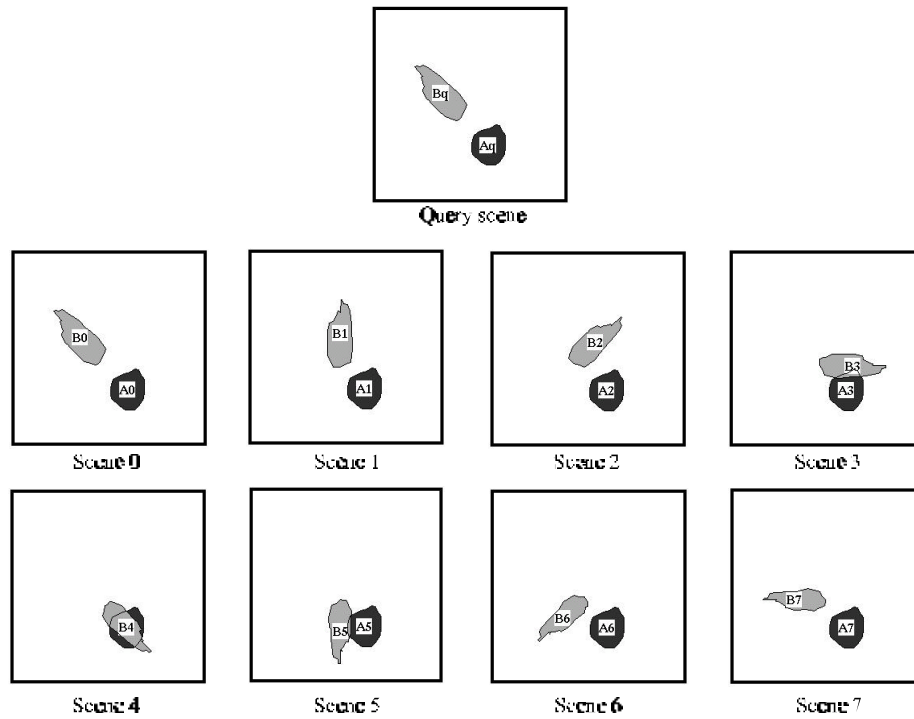


Figure 6.40: The scenes generated by rotating the target object.

Scene	4-neighbor		8-neighbor	
	similarity	rank	similarity	rank
0	100	1	100	1
1	96	3	92	3
2	85	4	70	4
3	66	7	46	8
4	52	8	49	6
5	72	6	48	7
6	81	5	63	5
7	98	2	95	2

Table 6.9: Similarities and ranks of Scenes 0-7 in Figure 6.40.

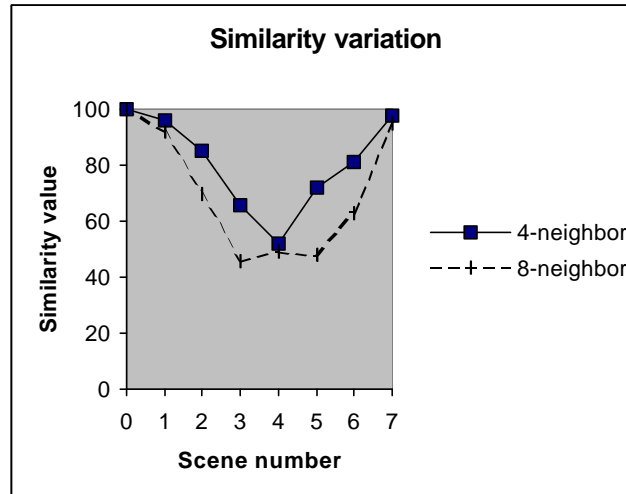


Figure 6.41: The pattern of similarity values for Scenes 0-7 in Figure 6.40.

6.6.10 Discussion

Out of nine scenarios considered, the 4-neighbor graph gives more sound rankings in three scenarios (Sections 6.6.1, 6.6.2, and 6.6.9), while in the remaining six scenarios both graphs give identical rankings. The 8-neighbor graph is able to distinguish between small changes, but cannot distinguish between large changes. The 4-neighbor graph, however, is capable of distinguishing not only between small changes, but also between large changes. In two of these scenarios (Sections 6.6.2 and 6.6.9), the 8-neighborhood graph gives similarity values that are non-monotonic, that is, it provides less sound mappings than the 4-neighborhood graph.

This insight refutes the hypothesis of this thesis, such that the four-neighborhood and eight-neighborhood graphs do not provide equally sound mappings of direction changes onto similarity values. The efforts in computing 4-neighbor and 8-neighbor similarities are the same; therefore, the 4-neighbor graph is preferred over the 8-neighbor graph for similarity assessments of cardinal directions using the model developed in this thesis.

6.7 Summary

This chapter described the implementation of Disha, a direction comparison system, and used it to computationally test the hypothesis. The study showed that the similarity values

obtained using the method developed in the previous chapter are cognitively plausible. We compared the mappings provided by the 4-neighbor and 8-neighbor graphs from direction changes to similarity values, and found that the former provides a sounder mapping than the latter.

Chapter 7

Conclusions

Qualitative spatial relations are essential components of questions that people would like to ask a Geographic Information System (GIS). To give plausible answers to the questions that involve spatial relations, GISs need to employ effective models of spatial relations. This thesis focused on cardinal directions, a type of spatial relation that captures the information about relative placement of objects in an extrinsic reference frame. Current models of cardinal directions approximate objects by points or minimum bounding rectangles (MBRs). To record exact directions, including the influence of the shape of objects on cardinal directions, we developed a model of cardinal directions based on direction-relation matrices. This chapter, summarizes the thesis (Section 7.1), describes results and major findings (Section 7.2), and highlights future work made possible by this research (Section 7.3).

7.1 Summary of the Thesis

Cardinal directions are used in spatial databases for answering queries about the directions between objects and inferring unknown directions between the objects from the known directions. They are also employed for retrieving spatially similar datasets from a digital library of geographic datasets. Keeping these applications in mind, we established five requirements for a direction model: formal, inferential, shape-sensitive, dimension-neutral, and comparable. This thesis evaluated existing models of cardinal directions to test whether or not these models possess the desired properties. No model was found that has all the properties, because existing models approximate the geometry of the extended objects either by points or by minimum bounding rectangles.

This thesis developed a model of cardinal direction using the coarse direction-relation matrix, which divides the space into nine tiles, each corresponding to a cardinal direction with respect to the reference object, and records with which tiles the target object intersects. An element in a coarse direction-relation matrix is empty if the target object does not intersect with the corresponding tile, and non-empty if it intersects. The thesis studied the effects of the objects' shapes and sizes and distance between them on the recorded direction-relation matrix. It compared the direction-relation matrix with the model based on minimum bounding rectangles, and developed detailed direction-relation matrix to record more details about directions, such as the ratio of target object and number of target object separations in a tile.

While the coarse direction model is powerful enough to capture directions between regions, it does not immediately apply to other types of geographic representation, such as lines and points; therefore, this thesis extended it to a deep direction model, which is capable of recording directions between arbitrary pairs of points, lines, and regions. The deep direction model uses deep direction-relation matrices, which additionally record the neighbor codes, if necessary. The neighbor code is information about the intersections of the target object with the boundaries of direction partitions. This thesis analyzed the patterns of neighbor code for various types of reference and target objects and showed that the directions recorded with the deep direction-relation matrices at smaller scales are compatible with the directions recorded at larger scales. It demonstrated that the deep direction-relation matrix records equal values of directions for cognitively equivalent directions.

Based on the conceptual neighborhood of cardinal directions, this thesis developed a method for assessing similarity between cardinal directions. The similarity between two directions is complimentary to the dissimilarity, which is proportional to the distance between two directions. To compute the distance between two directions, we calculate the cost of transforming a direction-relation matrix into another by moving non-zero elements along the conceptual neighborhood graph. Direction-relation matrices that have more than one non-zero elements can have multiple possibilities of transforming a matrix

into another. To compute the minimum cost, this thesis employed the transportation algorithm in linear programming.

Since different conceptual neighborhood graphs may be designed, possibly yielding different similarity values for a given pair of directions, we compared the soundness of mappings provided by the four- and eight-neighborhood graphs. For this purpose, we developed a software prototype of a direction-comparison system that uses the model developed in this thesis. Through a systematic evaluation of the similarity assessment method, we experimentally established that the four-neighborhood conceptual graph provides more sound mapping than the eight-neighborhood graph.

7.2 Results and Major Findings

The major results of this thesis are:

- *The direction-relation matrix is an exact model, which does not approximate the objects' shapes.*

The shape of objects influences the cardinal directions between them. Earlier models of directions approximated the objects' shapes; therefore, they are not able to capture the effect of shape on cardinal directions, and often record misleading directions. The computational model based on direction-relation matrices records exact directions without approximating shapes; therefore, it applies alike to objects that are concave or convex; regularly or irregularly shaped; and intertwined or clearly separate. This characteristic of the model can significantly improve the results of database queries and results of spatial reasoning over directions.

- *Direction-relation matrices provide a knowledge structure that can record multiple directions.*

A target object can be in multiple directions with respect to a reference object. Most models of cardinal directions were designed to determine from the objects' geometries whether or not a target object exists in a particular direction with respect to a reference object. The direction-relation matrix provides a knowledge structure to encode multiple

directions, such that all questions regarding directions can be answered from a direction-relation matrix that is computed once and stored as a *spatial data-object*. The similarity assessment and spatial reasoning operations make the spatial data-object useful for geographic databases and geographic datasets.

- *Direction-relation matrices allow recording of multiple levels of detail in the same framework.*

A particular feature of the model based on direction-relation matrices is its ability to describe direction relations at multiple levels of detail. At a coarse level, the direction-relation matrix records into which tiles around the reference object the region target object falls. At a finer level, it captures how much of the target object falls into each tile, and an even more detailed view is given if the direction-relation matrix records properties of the components in each tile. This multi-resolution model has significant implications for spatial query processing when a user, for instance, sketches the objects of interest. The coarse direction-relation may then act as a filter to quickly retrieve candidates, whereas the more detailed direction-relation matrices can be used to prioritize the candidates for a similarity assessment.

- *The model based on direction-relation matrices applies to points, lines, and regions alike.*

The deep direction model records directions in deep direction-relation matrices, which are based on coarse direction-relation matrices and additionally record the intersection of the target objects with the boundaries of the partitions, if necessary. This model yields equal values for cognitively equivalent directions regardless of the dimensions of the objects, and frees the user from pondering about the dimension of the objects while making queries in a GIS.

- *The directions recorded at smaller scales using the deep direction model are compatible with the directions recorded at larger scales.*

This thesis demonstrated that directions recorded with the deep direction model at smaller scales are compatible with the directions recorded at larger scales. The compatibility

makes this model useful for querying spatial databases at multiple scales, as it ensures the consistency in the results of the direction queries.

- *The model based on direction-relation matrices provides a solid foundation for spatial reasoning.*

The correct and multiple directions recorded with direction-relation matrices make this model very useful for exact spatial reasoning. Deep direction-relation matrices can be used for spatial reasoning for directions between arbitrary pairs of points, lines, and regions.

- *The model is useful for retrieving spatially similar scenes.*

Retrieval of similar scenes is a common task in most domains including geographic data. The method of similarity assessment makes this model useful for retrieving spatially similar scenes in image databases, video databases, multimedia databases, and web-databases.

- *The model based on direction-relation matrices can be used to detect and predict changes.*

Direction-relation matrices and similarity assessment method capable of detecting and quantifying changes between snapshots of varying scene, such as changes in the landscape of the city over a period or a video clip of a fast change. The changes in objects are primarily movement, rotation, and enlargement. Using the conceptual neighborhood graph and the trend of change, one can predict the state of objects in future.

- *The rejection of the hypothesis reveals that the 4-neighborhood graph provides a sounder mapping than the 8-neighborhood graph.*

We investigated the 4-neighborhood and 8-neighborhood graphs as the basis for the direction-similarity assessment. A comparative study shows that the 4-neighborhood graph provides a sounder mapping than the 8-neighborhood graph.

- *The direction-relation matrix meets all five requirements for a cardinal-direction model.*

This thesis analyzed the requirements of a direction model before developing it. The desired properties are *formal*, *inferential*, *shape-sensitive*, *dimension-neutral*, and *comparable*. The model applies to all objects alike; therefore, it is *formal*. It forms a strong foundation for spatial reasoning; therefore, it is *inferential*. It does not approximate the shapes of objects; therefore, it is *shape-sensitive*. It applies to objects of different dimensions (i.e., points, lines, and regions) and records compatible directions; therefore, it is *dimension-neutral*. The method for similarity assessment makes the directions *comparable*.

7.3 Future Work

This thesis presented a cardinal direction model that is computationally sound and has the properties established before developing the model. Although this thesis has presented significant results of the research pursued while designing this direction model, this research also uncovered avenues for further research, which are closely related to this thesis.

7.3.1 Direction Reasoning using Direction-Relation Matrix

The deep direction-relation matrices (Chapter 4) are capable of representing directions between objects of different dimensions. A method for direction reasoning using this model can be developed by performing reasoning on the x - and y -axes and combining results by performing a cross product. There are four types of pairs of projections of reference and target objects on an axis: (1) a pair of points (PP), (2) a pair of a point and an interval (PI), (3) a pair of an interval and a point (IP), and (4) a pair of intervals (II).

In order to obtain the composition of relations along an axis, we will need composition tables for eight possible pairs on an axis (Table 7.1). The composition tables for the case PP-PP is trivial. The composition table for the case II-II is available for interval relations (Allen 1983). For other six cases, composition table must be generated.

Case	Type of projection for A	Type of projection for B	Type of projection for C	Pair code for $dir(A, C)$
PP-PP	point	point	point	PP
PP-PI	point	point	interval	PI
PP-IP	point	interval	point	PP
PP-II	point	interval	interval	PI
IP-PP	interval	point	point	IP
IP-PI	interval	point	interval	II
II-IP	interval	interval	point	IP
II-II	interval	interval	interval	II

Table 7.1: The types of projections of objects A , B , and C along an axis and the composition types. The codes P and I stand for point and interval types of projections, respectively.

For a converse operations there are four types of pairs: PP, PI, IP, and II. Computing the converse for the case PP is trivial and the method for computing the converse for the case II is available (Allen 1983). For the cases IP and PI, methods for computing converse relations must be derived, for instance, from Figures 4.15a-b.

The deep direction-relation matrices, however, do not record the information about their types of projections, they just record the intersection with partitions and neighbor codes, if necessary. A method will be needed that can extract the projection information from the deep direction-relation matrices and use it for spatial reasoning operations.

7.3.2 Deep Detailed Direction-Relation Matrix

The detailed direction-relation matrix is used for assessing similarity between cardinal directions for regions. Similarly, a deep detailed direction-relation matrix for arbitrary pairs of points, lines, and regions will be useful for similarity assessment between directions across different dimensions of objects. Detailed directions can be recorded for line and region targets only, as a point has no extent; therefore, there is no detail to record for a point target. In the case of a line target object, the length ratio in each partition can be recorded. The primary challenge lies in recording details about the intersection of a region with lines and points and the intersection of a line with lines and points. For example, if the reference object is a point and the target object is a region, north, south, east, and west directions are lines, and the same direction is a point.

7.3.3 Cognitive Evaluations

The direction model and the similarity assessment techniques were targeted for spatial querying in databases; therefore, they are formal and computationally elegant. In order to use this model for natural language queries, however, the model must be tested for its cognitive plausibility. The following areas would need such testing:

7.3.3.1 Cognitive Plausibility of the Direction-Relation Matrix

The direction-relation matrix partitions the space in rectangular partitions corresponding to cardinal directions. Cognitive studies about the directions in an intrinsic reference frame show that people parse space into unequal and overlapping cones (Franklin *et al.* 1995). Similar cognitive studies to learn how people perceive cardinal directions between extended objects can be conducted. Using the results of the studies, one can evaluate how well the direction-relation matrix adheres to people's intuition.

7.3.3.2 Distances between Single-Element Direction-Relation Matrices

The distance between two single-element matrices is the distance between their non-zero elements along a conceptual neighborhood graph. We have computed distances between single-element direction-relation matrices based on the 4-neighborhood (Figures 5.4) and 8-neighborhood (Figures 6.22) conceptual neighborhood graphs. The 4-neighborhood distances give cognitively more plausible rankings of the scenes than the 8-neighborhood distances, however, the values of distances that conform to people's perception should be determined by performing human subject testing. The method for similarity assessment is flexible enough to accommodate these distances.

7.3.3.3 Similarity Assessment

The systematic evaluation of the similarity assessment method showed that the values of similarity adhered to our expectations. A study involving human subject testing for the similarity values can reveal how well these values conform to people's perception of similarity.

7.3.4 Mapping of Natural Language Direction Terms onto Direction-Relation Matrices

In order to make spatial queries such as “Find all towns that are north of Bangor in the state of Maine,” the natural-language direction terms should be mapped onto a corresponding direction-relation matrix. For example, some people may consider the area of acceptance for north as the north tile, while some others may consider it as the union of the northwest, north, and northeast tiles. Shariff (1996) presented the correspondence between topological relations and their metric parameters for relations between a region and a line. A similar effort for direction terms will be useful for natural-language queries using directions. We feel that all the necessary intersections are recorded in deep direction-relation matrices, therefore, the direction terms can be defined in terms of the values of these matrices.

7.3.5 Detecting and Quantifying Change

The change in the directions can be detected from an ordered sequence of direction-relation matrices. For instance, if one monitors objects by extracting the direction-relation matrices from digital imagery, one obtains a higher-level description of the spatial configuration. Of particular interest for us is inferring automatically what qualitative changes have occurred, and what changes are expected. The conceptual neighborhood graphs can be used to infer new relations over the period. The similarity measure can be used to quantify the degree of change.

7.3.6 Inferring the Type of Change from the Profile of Similarity Values

The graphs of similarity values for various changes (Chapter 6) exhibit a degree of correlation between the profile of similarity values and the type of change. An investigation can be conducted to determine whether or not there is sufficient correlation in the profiles of similarity values and the types of change: curved movement, diagonal movement, horizontal movement, vertical movement, enlargement in a direction, enlargement in all directions, and rotation with respect to its own extreme. If there is

sufficient correlation, the profile could be used to infer the type of change and, therefore, abstract change to a high-level concept.

7.3.7 Extension to 3-Dimensional Space

The model of directions in 2-D using direction-relation matrices can be extended to 3-D. The minimum enclosing box for the 3-D reference object will be central to the reference grid and will constitute the *same* volume. The reference grid with 125 parts of the space will have 27 volumes, $6*9=54$ areas, $12*3 = 36$ lines, and 8 points. If the point, lines, and areas are considered to have no extent for directions between volumes, a 3-D matrix with 27 element that is analogous to coarse direction-relation matrix in 2-D is adequate. Similarly, detailed and deep direction-relation matrices can also be developed for directions in 3-D space, forming the foundation for direction reasoning in 3-D.

7.3.8 Similarity between Raster Templates Using the Conceptual Neighborhood Graph

In pattern recognition (Schalkoff 1992, p. 330-331), the metrics m_1 and m_2 are used for matching templates f and g (Equations 7.1a-b), where R is the extent over which a match occurs. Other methods for comparing image templates are Fourier descriptors (Persoon and Fu 1977), least square matching (Agouris and Gruen 1994), and modified least square matching (Carswell 2000).

$$m_1 = \sum_R |f - g| \quad (7.1a)$$

$$m_2 = \sum_R (f - g)^2 \quad (7.1b)$$

The coarse and detailed direction-relation matrix can alternately be visualized as 3×3 binary and gray raster templates. The method developed for similarity assessment (Chapter 5) compares 3×3 templates, where the sum of the gray (detailed) values of each template is 1. This method of similarity assessment can be extended to assess similarity between $n \times n$ raster templates. Such an extension has the potential of giving better

measures than m_1 and m_2 . This extension may be applicable for comparison of well-defined objects such as brain tumors, where the exact shape of the artifacts are of great importance, and scale and orientation of the objects are fixed.

Bibliography

- A. Abdelmoty (1995). *Modelling and Reasoning in Spatial Databases: A Deductive Object-Oriented Approach*, Ph. D. Thesis, Department of Computing and Electrical Engineering, Heriot-Watt University, Edinburgh, Scotland.
- R. Abler (1987). The National Science Foundation National Center for Geographic Information and Analysis. *International Journal of Geographical Information Systems* 1(4): 303-326.
- P. Agouris and A. Gruen (1994). Linear Feature Extraction by Least Squares Template Matching Constrained by Internal Shape Forces. *International Archives of Photogrammetry and Remote Sensing* XXX(Part 3/1): 316-323.
- W. Al-Khatib, Y. Day, A. Ghafoor and P. Berra (1999). Semantic Modeling and Knowledge Representation in Multimedia Databases. *IEEE Transactions on Knowledge and Data Engineering* 11(1): 64-80.
- J. Allen (1983). Maintaining Knowledge about Temporal Intervals. *Communications of the ACM* 26(11): 832-843.
- Y. Aslandogan and C. Yu (1999). Techniques and Systems for Image and Video Retrieval. *IEEE Transactions on Knowledge and Data Engineering* 11(1): 56-63.
- M. Bertolotto (1998). *Geometric Modeling of Spatial Entities at Multiple Levels of Resolution*, Ph. D. Thesis, Dipartimento di Informatica e Scienze dell'Informazione, Univesita di Genova, Genova, Italy.

- A. Bimbo and P. Pala (1997). Visual Image Retrieval by Elastic Matching of User Sketches. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19(2): 121-132.
- A. Bimbo, E. Vicario and D. Zingoni (1995). Symbolic Description and Visual Querying of Image Sequences Using Spatio-Temporal Logic. *IEEE Transactions on Knowledge and Data Engineering* 7(4): 609-622.
- T. Bittner (1997). A Qualitative Coordinate Language of Location of Figures within the Ground. *Spatial Information Theory—A Theoretical Basis for GIS, International Conference COSIT '97*, Laurel Highlands, PA, S. Hirtle and A. Frank, Eds., Lecture Notes in Computer Science, 1329: 223-240.
- T. Bittner (1999). On Ontology and Epistemology of Rough Location. *Spatial Information Theory—Cognitive and Computational Foundations of Geographic Information Science COSIT' 99*, Stade, Germany, C. Freksa and D. Mark, Eds., Lecture Notes in Computer Science, 1661: 433-448.
- T. Bruns and M. Egenhofer (1996). Similarity of Spatial Scenes. *Seventh International Symposium on Spatial Data Handling*, Delft, The Netherlands, M.-J. Kraak and M. Molenaar, Eds., 173-184.
- B. Buttenfield (1989). *Multiple Representations: Initiative 3 Specialist Meeting Report*. Technical Report: 89-3, National Center for Geographic Information and Analysis, Santa Barbara, CA.
- R. Byrne and P. Johnson-Laird (1989). Spatial Reasoning. *Journal of Memory and Language* 28: 564-575.

- J. Carswell (2000). *Using Raster Sketches for Digital Image Retrieval*, Ph. D. Thesis, Department of Spatial Information Science and Engineering, University of Maine, Orono, ME.
- C.-C. Chang and T.-C. Wu (1992). Retrieving the Most Similar Symbolic Pictures from Pictorial Databases. *Information Processing and Management* 28(5): 581-588.
- S.-K. Chang and E. Jungert (1996). *Symbolic Projection for Image Information Retrieval and Spatial Reasoning*. Academic Press, London.
- S.-K. Chang, Q.-S. Shi and C.-W. Yan (1987). Iconic Indexing by 2-D Strings. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 9(6): 413-428.
- W. Chu, C. Hsu, A. Cardenas and R. Taira (1998). Knowledge-based Image Retrieval with Spatial and Temporal Constructs. *IEEE Transactions on Knowledge and Data Engineering* 10(6): 872-888.
- W. Chu, I. Leong and R. Taira (1994). A Semantic Modeling Approach for Image Retrieval by Content. *VLDB Journal* 3(4): 445-477.
- A. Cohn (1996). Calculi for Qualitative Spatial Reasoning. *Artificial Intelligence and Symbolic Mathematical Computation*, J. Calmet, J. Campbell and J. Pfalzgraf, Eds., Lecture Notes in Computer Science, 1138: 124-143 Springer-Verlag, NY.
- T. Cormen, C. Leiserson and R. Rivest (1990). *Introductions to Algorithms*. The MIT Electrical Engineering and Computer Science The MIT Press, Cambridge, MA.
- G. Dantzig (1963). *Linear Programming And Extensions*. Princeton University Press, Princeton, NJ.
- G. Dantzig and M. Thapa (1997). *Linear Programming*. Springer-Verlag, New-York.

- M. Egenhofer (1996). Multi-Modal Spatial Querying. *Seventh International Symposium on Spatial Data Handling*, Delft, The Netherlands, M.-J. Kraak and M. Molenaar, Eds., 785-799.
- M. Egenhofer and K. Al-Taha (1992). Reasoning About Gradual Changes of Topological Relationships. *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space*, Pisa, Italy, A. Frank, I. Campari and U. Formentini, Eds., Lecture Notes in Computer Science, 639: 196-219.
- M. Egenhofer and R. Franzosa (1991). Point-Set Topological Spatial Relations. *International Journal of Geographical Information Systems* 5(2): 161-174.
- M. Egenhofer and R. Golledge, Eds. (1998). *Spatial and Temporal Reasoning in Geographic Information Systems*. Spatial Information Systems. Oxford University Press, New York.
- M. Egenhofer and D. Mark (1995a). Modeling Conceptual Neighbourhoods of Topological Line-Region Relations. *International Journal of Geographical Information Systems* 9(5): 555-565.
- M. Egenhofer and D. Mark (1995b). Naive Geography. *Spatial Information Theory—A Theoretical Basis for GIS, International Conference COSIT '95*, Semmering, Austria, A. Frank and W. Kuhn, Eds., Lecture Notes in Computer Science, 988: 1-15.
- M. Egenhofer and R. Shariff (1998). Metric Details for Natural-Language Spatial Relations. *ACM Transactions on Information Systems* 16(4): 295-321.
- M. Flickner, H. Sawhney, W. Niblack, *et al.* (1995). Query by Image and Video Content: The QBIC System. *IEEE Computer* 28(9): 23-32.

- L. Ford and D. Fulkerson (1962). *Flow in Networks*. Princeton University Press, Princeton, NJ.
- A. Frank (1996). Qualitative Spatial Reasoning: Cardinal Directions as an Example. *International Journal of Geographic Information Systems* 10(3): 269-290.
- A. Frank and S. Timpf (1994). Multiple Representations for a Cartographic Objects in a Multiscale tree-an Intelligent Graphical Zoom. *Computers & Graphics* 18(6): 823-829.
- N. Franklin, L. A. Henkel and T. Zangas (1995). Parsing Surrounding Space into Regions. *Memory and Cognition* 23(4): 397-407.
- J. Freeman (1975). The Modelling of Spatial Relations. *Computer Graphics and Image Processing* 4: 156-171.
- C. Freksa (1992a). Temporal Reasoning Based on Semi-Intervals. *Artificial Intelligence* 54: 199-227.
- C. Freksa (1992b). Using Orientation Information for Qualitative Spatial Reasoning. *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space*, Pisa, Italy, A. Frank, I. Campari and U. Formentini, Eds., Lecture Notes in Computer Science, 639: 162-178.
- C. Freksa and K. Zimmermann (1992). On the Utilization of Spatial Structures for Cognitively Plausible and Efficient Reasoning. *IEEE International Conference on Systems, Man, and Cybernetics*, Chicago, IL, 261-266.
- T. Fuhr, G. Socher, C. Scheering and G. Sagerer (1998). A Three-Dimensional Spatial Model for the Interpretation of Image Data. *Representation and Processing of*

Spatial Expressions, P. Olivier and K. Gapp, Eds., 103-118 Lawrence Erlbaum Associates, Publishers, Mahwah, NJ.

- S. Futch, D. Chin, M. McGranaghan and J.-G. Lay (1992). Spatial-Linguistic Reasoning in LEI (Locality and Elevation Interpreter). *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space*, Pisa, Italy, A. Frank, I. Campari and U. Formentini, Eds., Lecture Notes in Computer Science, 639: 318-327.
- J. Glasgow and D. Papadias (1992). Computational Imagery. *Cognitive Science* 16(3): 355-394.
- R. Gonzalez and R. Woods (1992). *Digital Image Processing*. Addison-Wesley Publishing Company, Reading, MA.
- M. Goodchild and J. Proctor (1997). Scale in a Digital Geographic World. *Geographical & Environmental Modelling* 1(1): 5-23.
- V. Gudivada and V. Raghavan (1995). Design and Evaluation of Algorithms for Image Retrieval by Spatial Similarity. *ACM Transactions on Information Systems* 13(2): 115-144.
- H. Guesgen (1989). *Spatial Reasoning Based on Allen's Temporal Logic*. Technical Report: TR-89-049, International Computer Science Institute, Berkley, CA.
- O. Günther and A. Buchmann (1990). Research Issues in Spatial Databases. *SIGMOD RECORD* 19(4): 61-68.
- R. Güting (1994). An Introduction to Spatial Database Systems. *VLDB Journal*: 357-399.
- A. Guttman (1984). R-Trees: A Dynamic Index Structure for Spatial Searching. *Annual Meeting ACM SIGMOD*, Boston, MA, B. Yormark, Ed. 47-57.

- R. Haar (1976). *Computational Models of Spatial Relations*. Technical Report: TR-478, MSC-72-03610, Computer Science, University of Maryland, College Park, MD.
- P. Hayes (1978). The Naive Physics Manifesto. *Expert Systems in the Microelectronic Age*, D. Michie, Ed. 242-270 Edinburgh University Press, Edinburgh, Scotland.
- P. Hayes (1985). The Second Naive Physics Manifesto. *Formal Theories of the Commonsense World*, J. Hobbs and B. Moore, Eds., 1-36 Ablex Publishing Corporation, Norwood, NJ.
- D. Hernández (1993). Maintaining Qualitative Spatial Knowledge. *Spatial Information Theory, European Conference COSIT '93*, Marciana Marina, Elba Island, Italy, A. Frank and I. Campari, Eds., Lecture Notes in Computer Science, 716: 36-53.
- J. Hong (1994). *Qualitative Distance and Direction Reasoning in Geographic Space*, Ph.D. Thesis, Department of Surveying Engineering, University of Maine, Orono, ME.
- I. Horton (1998). *Beginning Visual C++ 6*. Wrox Press Ltd., Birmingham, UK.
- H. Jiang and A. Elmagarmid (1998). Spatial and Temporal Content-based Access to Hypervideo Databases. *VLDB Journal* 7: 226-238.
- L. Kulik and A. Klippel (1999). Reasoning about Cardinal Directions Using Grid as Qualitative Geographic Coordinates. *Spatial Information Theory—Cognitive and Computational Foundations of Geographic Information Science COSIT' 99*, Stade, Germany, C. Freksa and D. Mark, Eds., Lecture Notes in Computer Science, 1661: 205-220.
- S.-Y. Lee and F.-J. Hsu (1990). 2D C-String: A New Spatial Knowledge Representation for Image Database Systems. *Pattern Recognition* 23(10): 1077-1087.

- S.-Y. Lee, M.-K. Shan and W.-P. Yang (1989). Similarity Retrieval of Iconic Image Database. *Pattern Recognition* 22(6): 675-682.
- S.-Y. Lee, M.-C. Yang and J.-W. Chen (1992). Signature File as a Spatial Filter for Iconic Image Database. *Journal of Visual Languages and Computing* 3(4): 373-397.
- LINDO (1999). LINDO Systems Home Page: Optimization Modeling Tools for Linear, Nonlinear and Integer Programming, *LINDO Systems Inc.*, <http://www.lindo.com/>.
- D. Mark (1992). Counter-Intuitive Geographic 'Facts:' Clues for Spatial Reasoning at Geographic Scales. *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space*, Pisa, Italy, A. Frank, I. Campari and U. Formentini, Eds., Lecture Notes in Computer Science, 639: 305-317.
- M. McGranaghan (1989). Context-Free Recursive-Descent Parsing of Location-Descriptive Text. *AUTO-CARTO 9, Ninth International Symposium on Computer-Assisted Cartography*, 580-587 Baltimore, MD.
- M. McGranaghan and L. Wester (1988). Prototyping an Herbarium Collection Mapping System. *ACSM/ASPRS Annual Convention*, 5: 232-238 St. Louis, MO.
- R. McMaster and S. Shea (1992). *Generalization in Digital Cartography*. American Association of Geographers, Washington, DC.
- G. Miller (1956). The Magical Number Seven Plus or Minus Two: Some Limits on our Capacity for Processing Information. *Psychological Review* 63: 81-97.
- A. Mukerjee (1990). *A Representation for Modelling Functional Knowledge in Geometric Structures*. Technical Report: TAMU 90-004, Computer Science Department, Texas A & M University, College Station, TX.

- A. Mukerjee and G. Joe (1990). A Qualitative Model for Space. *Eighth National Conference on Artificial Intelligence*, Boston, MA, 2: 721-727.
- K. Murty (1976). *Linear and Combinatorial Programming*. John Wiley & Sons, Inc., New York.
- M. Nabil, J. Shepherd and A. Ngu (1995). 2D Projection Interval Relationships: A Symbolic Representation of Spatial Relationships. *Advances in Spatial Databases—4th International Symposium, SSD '95, Portland, ME*, M. Egenhofer and J. Herring, Eds., Lecture Notes in Computer Science, 951: 292-309 Springer-Verlag, Berlin.
- NCGIA (1995). Advancing Geographic Information Science, *National Center for Geographic Information and Analysis*, <http://www.ncgia.ucsb.edu/secure/secC.html>.
- R. Newell (1992). Practical Experiences of Using Object-Orientation to Implement a GIS. *GIS/LIS '92*, San Jose, CA, 624-629.
- P. Olivier and K. Gapp, Eds. (1998). *Representation and Processing of Spatial Expressions*. Lawrence Erlbaum Associates, Publishers, Mahwah, NJ.
- J. Paiva (1998). *Topological Equivalence and Similarity in Multiple Representation Geographic Databases*, Ph. D. Thesis, Department of Spatial Information Science and Engineering, University of Maine, Orono, ME.
- D. Papadias and V. Dellis (1997). Relation-Based Similarity. *Fifth ACM Workshop on Advances in Geographic Information Systems*, Las Vegas, NV, 1-4.

- D. Papadias, T. Sellis, Y. Theodoridis and M. Egenhofer (1995). Topological Relations in the World of Minimum Bounding Rectangles: A Study with R-trees. *Proceedings of the 1995 ACM SIGMOD International Conference on Management of data*, 92-103.
- D. Papadias, Y. Theodoridis and T. Sellis (1994). The Retrieval of Direction Relations using R-Trees. *Database and Expert Systems Applications—5th International Conference, DEXA '94, Athens, Greece*, D. Karagiannis, Ed. Lecture Notes in Computer Science, 856: 173-182 Springer-Verlag, New York.
- E. Persoon and K.-S. Fu (1977). Shape Discrimination using Fourier Descriptors. *IEEE Transactions on Systems, Man, and Cybernetics* 7(2): 170-179.
- D. Peuquet and C.-X. Zhan (1987). An Algorithm to Determine the Directional Relationship Between Arbitrarily-Shaped Polygons in the Plane. *Pattern Recognition* 20(1): 65-74.
- N. Pissinou, I. Radev, K. Makki and W. Campbell (1998). A Topological-Directional Model for the Spatio-Temporal Composition of the Video Objects. *Eighth International Workshop on Research Issues on Data Engineering, Continuous-Media Databases and Applications*, Orlando, FL, A. Silberschatz, A. Zhang and S. Mehrotra, Eds., 17-24.
- F. Preparata and M. Shamos (1985). *Computational Geometry*. Springer-Verlag, New York.
- J. Prosis (1999). *Programming with Windows MFC*. Microsoft Press, Redmond, WA.
- E. Puppo and G. Dettori (1995). Towards a Formal Model for Multi-Resolution Spatial Maps. *Advances in Spatial Databases—Fourth International Symposium on Large Spatial Databases, SSD '95, Portland, ME*, M. Egenhofer and J. Herring, Eds., Lecture Notes in Computer Science, 951: 152-169 Springer-Verlag, New York.

- G. Retz-Schmidt (1988). Various Views on Spatial Prepositions. *AI Magazine* 9(2): 95-105.
- C. Riesbeck (1980). "You Can't Miss It": Judging the Clarity of Directions. *Cognitive Science* 4: 285-303.
- H. Samet (1989a). *Applications of Spatial Data Structures: Computer Graphics, Image Processing, and GIS*. Addison-Wesley Publishing Company, Reading, MA.
- H. Samet (1989b). *The Design and Analysis of Spatial Data Structures*. Addison-Wesley Publishing Company, Reading, MA.
- R. Schalkoff (1992). *Pattern Recognition: Statistical, Structural and Neural Approaches*. John Wiley & Sons Inc., New York.
- R. Shariff (1996). *Natural Language Spatial Relations: Metric Refinements of Topological Properties*, Ph. D. Thesis, Department of Spatial Information Science and Engineering, University of Maine, Orono, ME.
- J. Sharma (1996). *Integrated Spatial Reasoning in Geographic Information Systems: Combining Topology and Direction*, Ph. D. Thesis, Department of Spatial Information Science and Engineering, University of Maine, Orono, ME.
- S. Shekhar and X. Liu (1998). Direction as a Spatial Object: A Summary of Results. *The sixth International Symposium on Advances in Geographic Information Systems*, Washington, 69-75.
- I. Sommerville (1996). *Software Engineering*. Addison-Wesley, Harlow, England.
- J. Strayer (1989). *Linear Programming and Its Applications*. Springer-Verlag, New York.

- L. Talmy (1983). How Language Structures Space. *Spatial Orientation: Theory, Research, and Application*, H. Pick and L. Acredolo, Eds., 225-282 Plenum Press, New York.
- N. Tryfona and M. Egenhofer (1996). Multi-Resolution Spatial Databases: Consistency Among Networks. *Integrity in Databases, Sixth International Workshop on Foundations of Models and Languages for Data and Objects*, S. Conrad, H.-J. Klein and K.-D. Schewe, Eds., 119-132 Schloss Dagstuhl, Germany.
- A. Tversky (1977). Features of Similarity. *Psychological Review* 84(4): 327-352.
- UCGIS (1996). Research Priorities for Geographic Information Science, *University Consortium for Geographic Information Science*, <http://www.ncgia.ucsb.edu/other/ucgis/CAGIS.html>.
- A. Yoshitaka and T. Ichikawa (1999). A Survey on Content-based Retrieval for Multimedia Databases. *IEEE Transactions on Knowledge and Data Engineering* 11(1): 81-93.
- K. Zimmermann (1993). Enhancing Qualitative Spatial Reasoning—Combining Orientation and Distance. *Spatial Information Theory, European Conference COSIT '93*, Marciana Marina, Elba Island, Italy, A. Frank and I. Campari, Eds., Lecture Notes in Computer Science, 716: 69-76.
- K. Zimmermann and C. Freksa (1996). Qualitative Spatial Reasoning Using Orientation, Distance, and Path Knowledge. *Applied Intelligence* 6(1): 49-58.

Biography of the Author

Roop Goyal was born in Mathura, Uttar Pradesh, India in 1967. He received his Bachelor of Engineering (Electronics and Communication Engineering) degree from the University of Roorkee, India in 1988. He then joined CMC Limited (A Government of India Enterprise), New Delhi where he worked as an information technology engineer until 1992. His responsibilities included technical support on computer systems for commercial and institutional clients.

In 1992 he enrolled in the Master of Engineering (research) program in Electrical and Electronic Engineering (EEE), Nanyang Technological University (NTU), Singapore, and joined the school of EEE, NTU as a graduate assistant. He received his Master of Engineering degree in 1995. In 1994 he joined the Department of Electronics and Communication Engineering, Singapore Polytechnic, where he worked as a faculty member until 1996. His responsibilities included teaching and supervising projects in the area of computer programming, digital image processing, digital signal processing, and electrical engineering. He is a member of the Institute of Electrical and Electronic Engineers (IEEE), IEEE Computer Society, the Association for Computing Machinery (ACM), and ACM Special Interest Group on Modeling of Data (SIGMOD).

In the summer of 1996 he enrolled for graduate studies in Spatial Information Science and Engineering at the University of Maine, and joined the National Center for Geographic Information and Analysis (NCGIA) as a research assistant. He is a candidate for the Doctor of Philosophy degree in Spatial Information Science and Engineering from the University of Maine in May 2000.