

8-2009

# Specifying and Detecting Topological Changes to an Areal Object

Jixiang Jiang

Follow this and additional works at: <http://digitalcommons.library.umaine.edu/etd>



Part of the [Geographic Information Sciences Commons](#)

---

## Recommended Citation

Jiang, Jixiang, "Specifying and Detecting Topological Changes to an Areal Object" (2009). *Electronic Theses and Dissertations*. 554.  
<http://digitalcommons.library.umaine.edu/etd/554>

This Open-Access Dissertation is brought to you for free and open access by DigitalCommons@UMaine. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of DigitalCommons@UMaine.

**SPECIFYING AND DETECTING TOPOLOGICAL CHANGES TO  
AN AREAL OBJECT**

By

Jixiang Jiang

B.Eng. Southeast University, China, 2003

M.Eng. Southeast University, China, 2005

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

August, 2009

Advisory Committee:

Michael F. Worboys, Professor in Spatial Information Science and Engineering

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

Max J. Egenhofer, Professor in Spatial Information Science and Engineering

Robert Franzosa, Professor in Mathematics and Statistics

Silvia Nittel, Associate Professor in Spatial Information Science and Engineering

**THESIS/DISSERTATION/PROJECT**  
**ACCEPTANCE STATEMENT**

On behalf of the Graduate Committee for Jixiang Jiang, I affirm that this manuscript is the final and accepted thesis/dissertation/project. Signatures of all committee members are on file with the Graduate School at the University of Maine, 42 Stodder Hall, Orono, Maine.

---

Michael F. Worboys,

Professor in Department of Spatial Information Science and Engineering

August, 2009

## LIBRARY RIGHTS STATEMENT

In presenting this thesis in partial fulfillment of the requirements for an advanced degree at The University of Maine, I agree that the Library shall 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.

Signature:

Date:

# SPECIFYING AND DETECTING TOPOLOGICAL CHANGES TO AN AREAL OBJECT

By Jixiang Jiang

Thesis Advisor: Dr. Michael F. Worboys

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)  
August, 2009

Topological changes to regions, such as merging, splitting, hole formation and elimination, are significant events in the evolution of regions. Wireless sensor network technology, which provides real-time information about the environment, can play an important role in detecting and reporting such topological changes.

This thesis provides theoretical foundations and algorithmic solutions to topological change detection using sensor networks. Two models, the *morphism-based model* and the *local tree model*, are developed, providing formal semantics of topological changes. The morphism-based model represents dynamic topological properties of continuously evolving areal objects, in which basic and complex topological changes are represented and classified using trees and structure-preserving mappings between them. Based on this model, this work constructs a normal form and proves that it is the simplest form that could represent all the changes under consideration. The local tree model represents discrete and incremental changes of the areal objects based on selected components and relations between them. It allows us to specify different kinds of topological changes using information within the locality of the change. Based on the local tree model, we

develop two decentralized and energy-efficient approaches, the *transient group-based (TG-based)* and the *adaptive group-based (AG-based)* approaches, to topological change detection using sensor networks. The TG-based approach employs the boundary group framework, which reduces the communication cost by reporting only the group level data instead of data from each individual node. The AG-based approach further reduces the communication cost by reusing the time-invariant information.

Experimental results show that when the configurations of sensor networks satisfy certain density and communication constraints, the proposed approaches are able to generate correct reports on the topological changes, and at the same time reduce the communication cost to a level much lower than that of a basic boundary-based data collection approach.

## ACKNOWLEDGMENTS

I would like to thank my advisor Dr. Mike Worboys for his invaluable guidance and constant support during my Ph.D. study, without which I can not have completed this thesis. My deepest thanks also go to other members of my thesis advisory committee, Dr. Kate Beard, Dr. Max Egenhofer, Dr. Robert Franzosa, and Dr. Silvia Nittel for their helpful advices and suggestions on my research. I want to thank Dr. John Stell for his insightful comments and feedbacks on my work.

My appreciation goes to Dr. Matt Duckham and other members in Spatial Information Science Group at the University of Melbourne for valuable inputs during our online meetings. My thanks also goes to all fellow graduates in the Department of Spatial Information Science and Engineering, especially to Christopher Farah, Cheng Zhong, Lisa Walton, Danqing Xiao, Guang Jin, and Qinghan Liang for the useful discussions on my research.

I would like to thank Ms. Ellen Huff, Dr. Edward Huff, and Ms. Jane Morse for their help in improving my English. I also thank my former advisor Dr. Baowen Xu, as well as colleagues Dr. Jianjiang Lu and Dr. Lei Xu, who introduced me to research and supported me to study abroad. Finally, this thesis could not have been undertaken without the support and patience of my wife Hongping Zhao and other family members and friends.

This work was partially supported by the National Science Foundation under Grant numbers IIS-0429644 and IIS-0534429, as well as the University of Maine Graduate Summer Research Awards.

## TABLE OF CONTENTS

ACKNOWLEDGMENTS . . . . .	iii
LIST OF TABLES . . . . .	viii
LIST OF FIGURES . . . . .	ix
Chapter	
1 INTRODUCTION . . . . .	1
1.1 Problem statement . . . . .	1
1.2 Goal and hypothesis . . . . .	3
1.3 Scope of the thesis . . . . .	4
1.4 Approach . . . . .	5
1.5 Major results . . . . .	8
1.6 Intended audience . . . . .	9
1.7 Organization of the remaining chapters . . . . .	9
2 RELATED WORK . . . . .	11
2.1 Modeling dynamic phenomena . . . . .	11
2.1.1 Definition of events . . . . .	11
2.1.2 Events and objects . . . . .	13
2.1.3 Classification of events . . . . .	14
2.2 Representation of topological properties . . . . .	16
2.2.1 Point-set approach . . . . .	16
2.2.2 Graph based representation . . . . .	19
2.2.3 Topological representation of areal objects . . . . .	22
2.2.4 Binary topological relations . . . . .	25



2.3	Sensor networks . . . . .	27
2.3.1	Characteristics of sensor networks . . . . .	27
2.3.2	Environmental data collecting . . . . .	28
2.3.3	Topology monitoring in sensor networks . . . . .	31
2.4	Summary . . . . .	32
3	TOPOLOGICAL CHANGES OF AN AREAL OBJECT . . . . .	33
3.1	Dynamic fields and areal objects . . . . .	33
3.2	Areal objects . . . . .	34
3.3	Topological relations between components of an areal object . . . . .	36
3.4	Basic definitions . . . . .	42
3.5	Tree representation of areal object . . . . .	44
3.5.1	Basic topological changes . . . . .	45
3.5.2	Complex topological changes . . . . .	47
3.6	The normal form for representing complex changes . . . . .	50
3.7	Properties of the normal form . . . . .	57
3.7.1	Need for all types of NTTCs in the normal form . . . . .	57
3.7.2	Need for the sequence <i>ISMD</i> in the simplest normal form . . . . .	59
3.8	Summary . . . . .	61
4	PRELIMINARIES FOR TOPOLOGICAL CHANGE	
	DETECTION IN SENSOR NETWORKS . . . . .	62
4.1	Local tree model . . . . .	63
4.2	Completeness of local tree model . . . . .	66
4.3	Sensor network configuration . . . . .	75
4.4	Density and communication issues . . . . .	79
4.5	Summary . . . . .	81

5	DISTRIBUTED APPROACHES FOR TOPOLOGICAL CHANGE	
	DETECTION USING SENSOR NETWORKS . . . . .	82
5.1	Basic approaches . . . . .	82
5.2	Decentralized topological change detection using transient groups . . . . .	84
5.2.1	Boundary group based representation . . . . .	86
5.2.2	Algorithm . . . . .	89
5.3	Decentralized topological change detection using adaptive groups . . . . .	96
5.3.1	Initialization . . . . .	97
5.3.2	Group formation . . . . .	97
5.3.3	Group information aggregation and reporting . . . . .	98
5.3.4	Group update . . . . .	99
5.3.5	Update aggregation and reporting . . . . .	100
5.3.6	Data reporting and analysis . . . . .	101
5.4	Summary . . . . .	102
6	EXPERIMENTS AND EVALUATIONS . . . . .	103
6.1	Experiments . . . . .	103
6.2	Evaluation of the transient group-based method . . . . .	106
6.3	Evaluation of the adaptive group-based method . . . . .	109
6.4	Effect of group size for TG-based and AG-based methods . . . . .	111
6.5	Summary . . . . .	112
7	CONCLUSIONS . . . . .	113
7.1	Summary of thesis . . . . .	113
7.2	Major findings and limitations . . . . .	115
7.2.1	The morphism-based model . . . . .	115
7.2.2	The normal form . . . . .	115
7.2.3	The local tree model . . . . .	116

7.2.4	Distributed algorithms of topological change detection . . . . .	116
7.3	Testing the validity of the hypothesis . . . . .	117
7.4	Future work . . . . .	118
7.4.1	Dynamic topological models of weakly connected areal objects . . . . .	118
7.4.2	Formal specification of continuous changes . . . . .	120
7.4.3	Unique representation of topological changes . . . . .	121
7.4.4	Topological change detection during non-incremental evolution of areal objects . . . . .	122
7.4.5	Topological change detection in three dimensional space . . . . .	124
	APPENDIX: PROOFS OF LEMMAS RELATED TO MORPHISMS . . . . .	126
	BIBLIOGRAPHY . . . . .	143
	BIOGRAPHY OF THE AUTHOR . . . . .	156

## LIST OF TABLES

Table 3.1	Topological relations between components . . . . .	41
Table 4.1	Approximated components\relations in sensor networks . . . . .	78
Table 6.1	Experimental data descriptions . . . . .	106

## LIST OF FIGURES

Figure 1.1	Consecutive snapshots of sea surface height . . . . .	2
Figure 1.2	Examples of different topological changes . . . . .	6
Figure 1.3	Examples of topological changes of the same type . . . . .	6
Figure 2.1	Examples of subsets of the topological space . . . . .	18
Figure 2.2	An example of a region adjacency graph . . . . .	19
Figure 2.3	Example of region adjacency tree . . . . .	20
Figure 2.4	Example of incidence graph . . . . .	21
Figure 2.5	An example of DAM model . . . . .	22
Figure 2.6	A scalar field and its contour tree . . . . .	23
Figure 2.7	An example of areal object with ambiguity . . . . .	23
Figure 3.1	An example of “pathological” cases . . . . .	35
Figure 3.2	Examples of weakly connected areal objects . . . . .	36
Figure 3.3	An example of tree representation of components . . . . .	37
Figure 3.4	Topological relations between $R$ and $R_h$ . . . . .	38
Figure 3.5	Topological relations between a pair of $R_h$ s . . . . .	39
Figure 3.6	Topological relations between $R$ and $\tilde{R}_h$ . . . . .	39
Figure 3.7	Topological relations between a pair of $\tilde{R}_h$ s . . . . .	40
Figure 3.8	The representation tree of an areal object . . . . .	45
Figure 3.9	Representation of different topological changes . . . . .	46
Figure 3.10	Two different topological changes . . . . .	47
Figure 3.11	Complex changes of the areal objects in Figure 1.1 . . . . .	48
Figure 3.12	Two equivalent complex changes . . . . .	49

Figure 3.13	A simplification of complex change in Figure 3.11(b)	51
Figure 3.14	The simplification of two MD-changes	54
Figure 3.15	The simplification of two SMD-changes	55
Figure 3.16	A complex change	58
Figure 3.17	Counter example 1	59
Figure 3.18	Counter example 2	60
Figure 3.19	Counter example 3	60
Figure 4.1	A basic transition	64
Figure 4.2	Tree representations for different configurations	66
Figure 4.3	Classification of basic transitions	66
Figure 4.4	Examples of different types of topological changes	67
Figure 4.5	An example of uncommon topological change	67
Figure 4.6	A sequence of $(f(x_i), f(y_i))$	68
Figure 4.7	Example of partially-surrounded-by relation	69
Figure 4.8	An illustration of Lemma 4.2	69
Figure 4.9	An illustration of Lemma 4.3	71
Figure 4.10	An illustration of Lemma 4.5	72
Figure 4.11	Sensor network configuration	76
Figure 4.12	An example of homogeneous components	79
Figure 4.13	Configurations that lead to errors in reports	80
Figure 5.1	Boundary nodes	83
Figure 5.2	The C-nodes and T-nodes	85
Figure 5.3	Groups among C-nodes	86
Figure 5.4	Information received at the representative node	87
Figure 5.5	Examples of C-nodes identification	91
Figure 5.6	Boundary groups and the routing tree	93

Figure 5.7	Details of group 0 . . . . .	94
Figure 5.8	An example of boundary group formation . . . . .	98
Figure 5.9	An example of group update . . . . .	100
Figure 5.10	An example of group update aggregation . . . . .	101
Figure 6.1	The hop distance of the sensor nodes in the experiment . . . . .	104
Figure 6.2	Snapshots of experimental data . . . . .	105
Figure 6.3	Communication cost of NBC and TG-based method . . . . .	106
Figure 6.4	The C-nodes and T-nodes at sensing round $t_{11}$ . . . . .	107
Figure 6.5	The boundary nodes at sensing round $t_{11}$ . . . . .	108
Figure 6.6	The C-nodes and T-nodes at sensing round $t_{20}$ . . . . .	108
Figure 6.7	Communication cost of the group-based methods . . . . .	109
Figure 6.8	The boundary groups after sensing rounds $t_{19}$ and $t_{20}$ . . . . .	109
Figure 6.9	The nodes involving in group update at sensing rounds $t_{20}$ . . . . .	110
Figure 6.10	Communication cost in different number of groups . . . . .	111
Figure 7.1	Two different splits (from [Gal97]) . . . . .	119
Figure 7.2	Decomposition of a transition region . . . . .	122
Figure 7.3	Examples of non-incremental changes . . . . .	123
Figure 7.4	Decomposition of a transition region . . . . .	124
Figure A.1	An illustration of Lemma 1 . . . . .	130
Figure A.2	An illustration of Lemma 2 . . . . .	136
Figure A.3	An illustration of Lemma 3 . . . . .	141

# Chapter 1

## INTRODUCTION

Advances in sensor networks make it possible to produce real-time and nearly continuous data sets describing geographic phenomena. The data capture not only static but also dynamic characteristics of the phenomena. When analyzing the sensed data, users are often interested in discovering salient events, so that they can make appropriate responses. Such events also provide areas where more detailed data analysis can be useful. In order for the sensor networks to provide more meaningful information, a treatment of the dynamic aspects of the phenomena in terms of dynamic happenings (including events, changes, and processes) is necessary.

### 1.1 Problem statement

Consider the following examples of topological changes from the domain of meteorology. Figure 1.1 shows eight consecutive snapshots of ocean areas with sea surface height (SSH) below a threshold (-15 cm) at the beginning of the onset of El Niño [Shi97]. It is easy to identify several changes in topological structure during this period, including:

1. Within the area indicated by an ellipse, between  $t_0$  and  $t_1$ , two regions *appear*. Between  $t_1$  and  $t_2$ , a region *merges* with itself and forms a *hole*, and between  $t_2$  and  $t_7$ , the *hole* is *merged* back to the exterior.



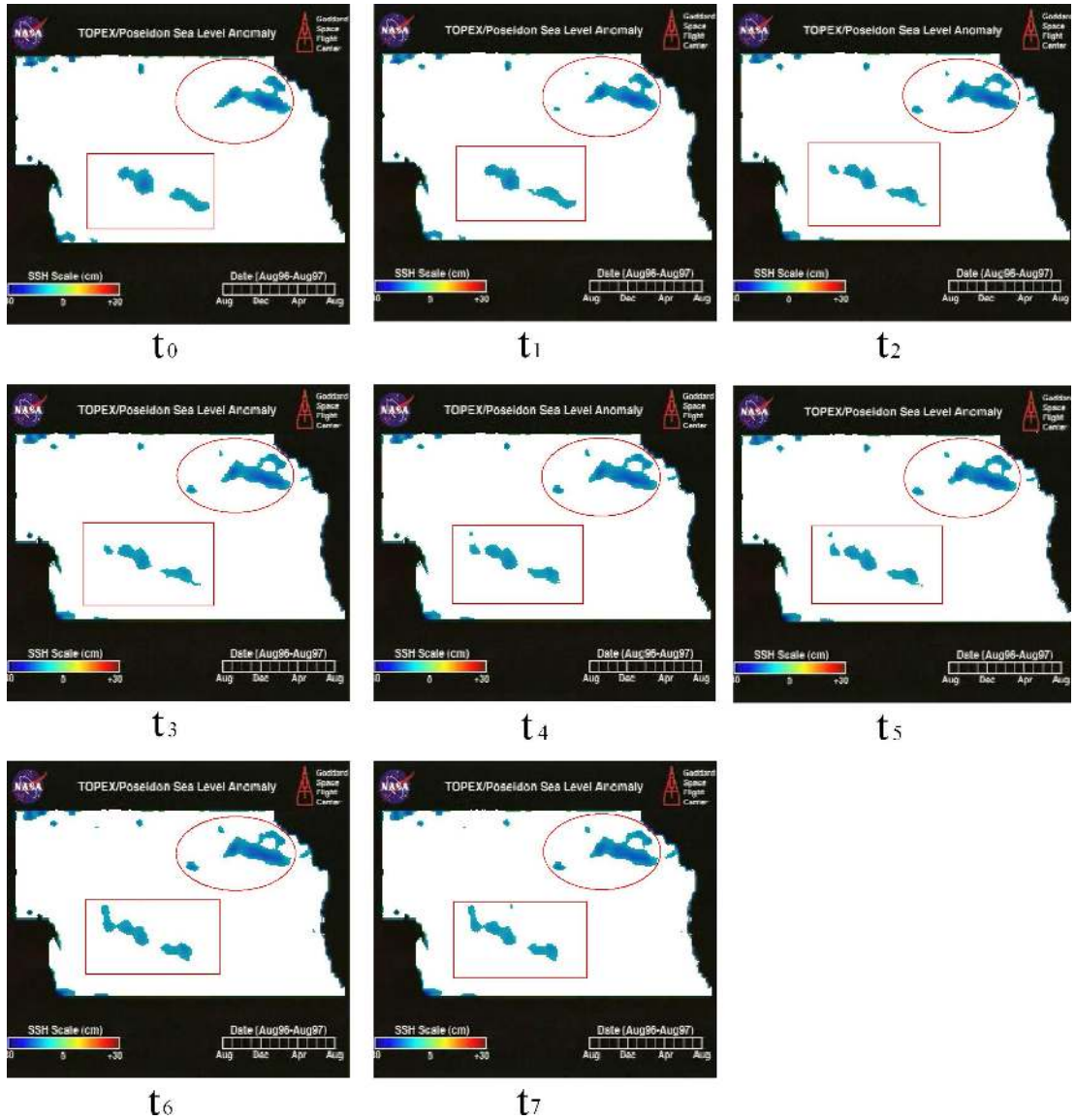


Figure 1.1 Consecutive snapshots of sea surface height

2. Within the area indicated by a rectangle, between  $t_0$  and  $t_2$ , a new region *appears*. Between  $t_2$  and  $t_4$ , another new region *appears*. Between  $t_4$  and  $t_6$ , three regions *merge* together. Finally, between  $t_6$  and  $t_7$ , a third region *appears*.

In addition to the SSH data, topological changes are common in many other spatial-temporal data provided by data acquisition technology, such as sensor networks. Users can be interested in these topological changes. For example, in the case of wildfire, fire fighters might be interested if the fire zone regions split and become disconnected, so that they can reorganize the team accordingly. They might also be interested in merging

fires, as it sometimes slows down the burn when the fires are burning over each other. In the case of a flood, the emergency services may be interested in the appearance of an island in the flood, because this indicates the locations of possible safe areas. Instead of emerging from the flood directly, it is also possible that an island is formed by the flood engulfing a piece of land. In this scenario, people on the island become separated and may have difficulty escaping. Therefore, rescue from such a newly formed island might have higher priority in the overall hazard management strategy.

To enable an event-based approach, it is important to identify a collection of primitive types of changes that are commonly found in geographic phenomena. These primitive changes can form the basic elements for querying and reasoning in terms of events. Examples of previous work that fit into this area include [HE00, Cla95, Ren00]. Such research captures properties of events and changes based on transitions of locations or identifications of objects, but does not consider topological characteristics, and in particular does not deal with changes in different kinds of connectivity, such as region appearance or merge. However, in some applications, especially when objects are derived from discretization of continuous fields, the topological structure of the objects can be complex, and changes in the topological structures are pervasive.

*This thesis concentrates on the analysis of topological changes during the evolution of areal objects. In particular, it yields a category of primitive types of topological changes together with methods for reporting these changes in sensor networks.*

## 1.2 Goal and hypothesis

The main goal of this thesis is to provide a computational foundation for specifying, classifying, and detecting salient events according to changes in the topological structure of areal objects. We formally define an areal object later, but for now an areal object can be thought of as a collection of region components, possibly with holes and islands.

Formal models are developed to represent the properties of these topological changes and to serve as a tool for us to determine the type of each change. In addition, the formalism provides the basis for algorithms to compute and process different types of topological changes in a selection of applications. We have the following key goals:

- (1) Choice and representation of salient static topological properties of areal objects.
- (2) Construction of a theory of dynamic topological properties of areal objects.
- (3) Development of algorithms for capturing the dynamic topological properties of areal objects by sensor networks.

Addressing the first goal leads to the identification of the necessary topological features of areal objects used to distinguish topological changes. Work on the second goal leads to a development of mathematical models with the following properties: (1) these models are rich enough to represent the static topological characteristics of areal objects necessary for the classification of changes. (2) The dynamic topological characteristics resulting from the evolution of areal objects can be explicitly represented using these models. The third goal concerns distributed approaches for detecting topological changes in sensor networks. These goals and problems allow us to make the following hypothesis:

*Models that represent dynamic topological properties of areal objects provide the capability of formally specifying and analyzing topological changes, which go beyond the models that only represent static topological properties. These models also allow the construction of distributed algorithms of topological change detection that are more energy-efficient than current approaches, such as the basic boundary-based approach.*

### **1.3 Scope of the thesis**

The research in this thesis focuses on the development of models that are useful for classifying topological changes during geographic events and processes. First, the model

takes the geographic world as a 2-dimensional space, and the areal objects considered are 2-dimensional entities with finite connected components. We neither consider 0-dimensional point-like or 1-dimensional line-like entities, nor do we consider objects with unbounded components. Second, changes are classified based on dynamic topological properties. Therefore when constructing the models for evolving areal objects, only topological properties of the dynamic areal objects are explicitly represented. Other properties, such as size, shape, or locations of areal objects, are not represented by the models. Third, most geographic phenomena evolve continuously, and as a consequence, in this thesis we assume areal objects also evolve continuously or incrementally. Discontinuous changes, like shifts of county boundaries, are beyond the scope of this thesis. Fourth, we focus on the analysis of properties of independent objects, not relations between or among objects. Finally, the models focus on patterns of changes from a topological perspective, and we do not go into the analysis of internal or external causes of these changes.

## 1.4 Approach

In this research, we analyze changes systematically based on their topological characteristics. Applying the concepts of the event-oriented approaches [Wor05], changes can be partitioned into classes, where each class characterizes common properties of its members.

Topology is an important component in the analysis of areal objects. The topological properties of dynamic areal objects can change as they evolve, which causes the topological changes. Topological changes can be differentiated not based on the static topological properties of the areal objects, but on their dynamic topological properties, or the way in which the topological properties change. For example, consider the changes shown in Figure 1.2. Both changes start with the same state and end with the

same state, but they are considered to be different changes, as the first change can be interpreted as a hole evolving its shape to engulf a part of the region that surrounds it, while the second change can be interpreted as the appearance of an island inside the hole. In the example shown in Figure 1.3, the areal objects in both cases have different static topological properties, but both changes are similar and can be classified into the same type of change, because both of them can be interpreted as the appearances of an island inside a hole.

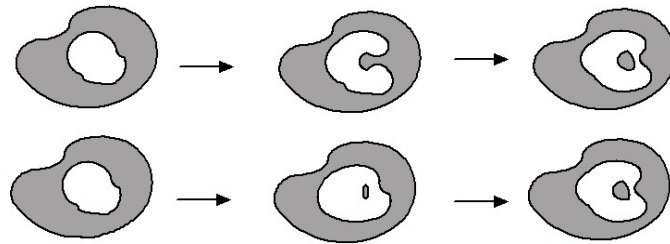


Figure 1.2 Examples of different topological changes

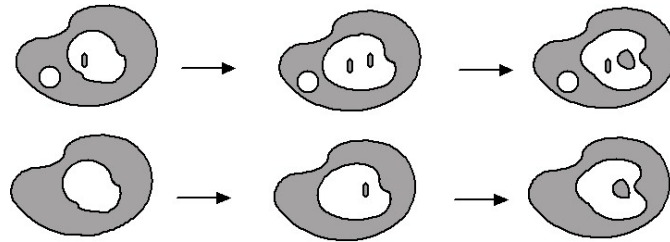


Figure 1.3 Examples of topological changes of the same type

In order to differentiate topological changes, models are developed to represent the dynamic topological properties of these changes. Most existing topological models represent the static topology of areal objects, and it is difficult to extend them for the representation of dynamic topology. To represent dynamic topology, we need new models.

In the first part of this research, we assume that the state of an areal object under consideration is available at any point in space and time. In this case, the static topological properties of the areal object can be represented by a rooted tree. A tree morphism can be built between a pair of consecutive representation trees according to

their genidentity relation [SG04], a transtemporal relation that relates an continuant entity to what it comes from. To be more specific, according to [SG04], “the formal-ontological relation of genidentity, or in other words the being-such-as-to-have-come-forth-from relation, comes in a number of different varieties. If you cut a chunk of matter in two, the sum of the remaining pieces is physically completely genidentical to the chunk before cutting. A new-born baby is biologically partially genidentical to its mother. Genidentity is thus a transtemporal - which for us means transontological - generalization of the relations of identity and part-whole among continuant entities.”

Different types of topological changes can be specified by tree morphisms. Based on the tree and morphism representation, some important properties of different types of topological changes can be revealed and proved.

In real-world monitoring, it is impossible to make perfect observations and sensing reports are usually generated at discrete locations and discrete times; therefore, continuous changes are often approximated by a series of discrete transitions. In each transition, a transition region topologically equivalent to a disk is added to or removed from the areal object. It turns out that the topological changes resulting from a transition can be classified according to the topological structures of the regions and holes that are connected to the transition region. These topological structures can be represented by a single rooted tree, and all kinds of specific types of topological changes can be specified based on the structure of the representation.

Because the model of basic transitions classifies the changes based on information only from the area where the change is observed, it allows us to design a distributed algorithm for sensor networks to detect such topological changes with low communication cost. Therefore, along with the theoretical research, algorithmic approaches to detecting the topological changes in sensor networks are also presented and tested.

## 1.5 Major results

The major achievement of this research is the construction of mathematical Models representing the dynamic topology of areal objects. These models explicitly represent the dynamic topology during the evolution of the areal objects, and capture the semantics of topological changes. The major results are as follows:

1. Models that specify primitive topological changes in which evolving areal objects can participate.
2. Proof of an important property of topological changes, namely that any topological change to the structure of regions can be formed as an ordered composition of primitive topological changes.
3. Identification of the important features necessary for classifying topological changes.
4. Design of distributed algorithms for capturing and classifying topological changes using real-time detection.

These results can be applied to the implementation of a sensor network monitoring system which is able to track dynamic geospatial phenomena and form qualitative reports describing their evolution in terms of topological changes. In addition, this research provides a foundation for further study in spatial and temporal queries in terms of topological changes. With the formal model, primitive types of changes can be incorporated into query languages, and this makes it possible to form queries concerning the dynamic topological properties of phenomena. Examples of such queries include, (1) Retrieve the data describing the events in the forest when a wild fire *appears*, (2) Retrieve the data describing the typhoons that occurred last year and *split* during their evolution.

## 1.6 Intended audience

The intended audience of this thesis primarily consists of researchers and developers who are interested in qualitative approaches to spatio-temporal changes, especially the modeling of changes from a topological perspective. The audience also includes experts from the fields of computer science whose research focuses on spatial or spatio-temporal information processing.

## 1.7 Organization of the remaining chapters

The remainder of the thesis is organized as follows.

Chapter 2 reviews research areas related to modeling and detecting topological changes. Research on dynamic phenomena modeling and classifications of events and changes is presented. The importance of topological property of spatial objects and different topological models are discussed. Related work in the literature of sensor networks is also introduced.

Chapter 3 presents the first model of topological change we proposed. A tree structure is employed to represent static topological properties of an areal object. Tree morphisms are used to represent dynamic topological changes of an areal object. Basic and complex topological changes are specified and classified based on the morphism representation. Finally, a normal form is constructed for the simplification of complex topological changes.

Chapter 4 studies theoretical foundations for topological detection in sensor networks. It focuses on discrete and incremental transitions of areal objects. A local tree model is proposed in  $\mathfrak{R}^2$  which represent the topological properties of a basic transition. Based on the local tree model different types of topological changes are specified. It also



presents a sensor network framework that captures the necessary information required by the tree model.

Chapter 5 creates two algorithms for topological change detection that incorporate the analysis results of Chapter 4. By focusing on capturing the fundamental information required by the local tree model, both proposed approaches are able to provide sensing reports on topological changes during the discrete and incremental changes of areal objects. By employing the group based framework and reusing the time-invariant group level data, energy-efficiency is achieved.

Chapter 6 describes the experiments that were conducted in order to test the proposed topological change detection approaches. The experimental results are also evaluated.

Chapter 7 concludes this thesis. It presents a summary, the major results and limitations of this thesis. It also provides discussions of future research areas.

## Chapter 2

### RELATED WORK

This research is related to the following two areas: (1) the development of models for representing changes to the topological properties of evolving regions, and (2) the construction of distributed algorithms of environmental monitoring using sensor networks.

This chapter presents the previous work in both areas.

#### 2.1 Modeling dynamic phenomena

Entities in the world are divided into two categories: *continuant* (e.g., a human being), which have continuous existence and a capacity to endure through time, and *occurrents* (e.g., a wild fire), which occur and unfold themselves through a period of time [GS04]. The *occurrents* include events and changes, and differ from *continuants* in that they have temporal parts [Sim87]. This section reviews previous work relating to *occurrents*.

##### 2.1.1 Definition of events

The definition of events has been the focus of considerable debate among philosophers. There are multiple definitions of events in the literature [PV00]. The extent of existing definitions of events may also contain other *occurrents*, such as processes or states, therefore in this subsection we do not distinguish events from *occurrents*.

Montague [Mon69] defines events to be properties of moments or intervals of time. The sun's rising is an example of an event defined as a property of a moment of time; that is, the property of being the moment at which the sun rises. The American presidential campaign belongs to the category of events identified with properties of intervals, rather than moments, of time.

Chisholm [Chi70] identifies both events and propositions to be species of states of affairs. As defined in [Chi70], a state of affairs here refers to "anything capable of serving as the object of belief, or of hope, or of wonderment, or of any of those other intentional attitudes that take things other than attributes or individuals as their objects." In his definition, a proposition is defined to be a special type of state of affairs which satisfies that either it or its negation always holds, and an event is defined to be any contingent state of affairs which is not a proposition and which implies change. As indicated in [Chi70], the state of affairs that "John walking at 3 P.M. on Feb 5, 1970" is a proposition, whereas, "John walking" is an event. "John sitting" is a state of affairs that is neither a proposition nor an event, for it does not imply any changes.

Quine [Qui64] gives events the same status as objects in four-dimensional space-time. In his definition, both events and objects are considered to comprise simply the content of some portion of space-time. By this definition, events are not distinguished from objects. Moreover, different events occupying the same space-time are not distinguished from each other. For example, if the Earth is rotating and simultaneously heating up, then by Quine's definition, the rotation and the heating up of the Earth are one and the same event, even though the property of rotating and the property of heating up are distinct.

Kim [Kim73] defines an event to be a concrete object (or an  $n$ -tuple of objects) exemplifying a property (or a  $n$ -adic relation) at a time. The event defined in this way can be described as "object (or a  $n$ -tuple object)  $x$  has property  $P$  at time  $t$ ." Identical

events must be exemplifications of the same property (or relations) by the same object (or  $n$ -tuples) at the same time. Therefore, the rotation and the heating up of the Earth are different by this definition, as they involve exemplifications of distinct properties.

Davidson [Dav80] identifies events by their positions in a causal network. He defines events to be the same if and only if they have exactly the same causes and effects. This definition gives each event a unique position in the framework of causal relations between events, which is in somewhat the same way as objects having a unique position in the spatial framework of objects.

None of the above definitions are perfect, and a common definition of events agreed upon by everyone has not yet been reached. However, each of the definitions contributes to our understanding.

### **2.1.2 Events and objects**

It is generally agreed that events and objects are ontologically different. However, from a modeling perspective, events and objects can be treated in many ways as structurally similar [WH04]. Several facts support this argument. First, similar to classes and instances in the object world, there are generic event-types and individual event-tokens for events [Gal00]. Second, like objects, events might have instances (occurrences), attributes, belong to a subsumption hierarchy, and have relations to other events [Wor05]. Finally, facts coming from the field of psychology also emphasize the analogy of events and objects. Zacks and Tversky [ZT01] have shown that event perception could be regarded as the temporally extended analog of object perception. As stated in the paper [ZT01], “events are objects in the manifold of the three dimensions of space plus the one dimension of time.” Hence, modeling events in the same way as objects are modeled also meets the nature of human perception and conceptualization.

The similarities between events and objects allow applying object-oriented approaches to event modeling. Worboys and Hornsby [WH04] propose a unified modeling approach for both events and objects. This model extends the geospatial object model to encompass both geospatial objects and events with their geosettings. Zhang [Zha05] applies the abstraction processes in the object-oriented paradigm, generalization and aggregation, to event modeling. The results of the abstraction are different event hierarchies that define mappings from specialized events (low-level events) to more general events (higher-level events). The generalization describes the *is – a* relation between events, which leads to the event taxonomy. The aggregation explains the compositional structure of complex events and establishes the *part – of* relations between events, which leads to the event partonomy.

As Galton points out [Gal04], there is no clean separation between (spatial) objects and (temporal) events, since in the real world we can find many phenomena that are not easy to fit into a simple object/event dichotomy, and can be viewed in multiple aspects. [Gal04] gives various examples of multi-aspect phenomena, including floods, wild fires, storms, weather fronts, epidemics, pollution incidents, invasions, processions, protest marches, traffic jams, bees swarming, and a plague of locusts.

### 2.1.3 Classification of events

The analogy between events and objects also sheds light on event classification. Objects with similar properties can form a class and be described by a concept (e.g., lakes or mountains). These concepts form the basis of a domain ontology [Gua98]. Similarly, events can be classified according to certain criteria to form event classes. In the literature, some research has been done to classify events and to form event classes.

Based on three basic components of geo-objects [Arm88], eight change scenarios are proposed [RK95], including change in geometry, change in topology, change in at-

tribute, change in topology and attribute, change in attribute and geometry, change in geometry and topology, change in geometry, topology and attribute, and no change.

Hornsby and Egenhofer [HE00] present research on specifying identification based change. They systematically analyze changes with respect to states of existence and nonexistence for identifiable objects. In their work, an object is modeled by its identification. The status of an object can be one of: existence, non-existence without history, and non-existence with history. The changes of the object are tracked by the changes in the status of its identification. These identification changes can be classified according to the status of the object before and after the changes. According to the model, the identification changes of a single object can be classified into types of continue non-existence without history, create, recall, destroy, continue existence, eliminate, forget, reincarnate, and continue non-existence with history. More types of identification changes are specified by considering interactions between the objects. Based on the identification based changes to simple objects, the effects of temporal zooming on the transitions between the identity states of the objects are discussed in [HE99], and the identification-based change operations to composite objects are analyzed in [HE98].

Claramunt [Cla95] analyzes possible changes that could take place in objects and categorizes them into three different classes: evolution concerning a single entity, evolution in the functional relations between several entities, and evolution in spatial structure involving several entities.

Renolen [Ren00] identifies, by case study, six types of changes according to the transitions of objects. They are creation (an object is created), alteration (an object is changed or modified), cessation (an object is destroyed or removed), reincarnation (an object that previously has been destroyed or removed is reintroduced), split/deduction (an object is subdivided in two or more new objects, or one or more objects is deducted

from an existing object), and merge/annexation (two or more objects are joined together to form a new object, or one or more objects are “swallowed” into another object).

Galton and Worboys [GW05] discuss the ontological categories of events and states in dynamic geo-networks and possible kinds of causal relations. Three classes of events in networks are distinguished, namely, changes to the structure of the network itself (e.g., introduction of a new link), changes which do not affect the structure of the network itself but which may affect the flows in the networks (e.g., removal of an obstruction), and changes that occur in the flows themselves (e.g., creation of a new flow on a link).

Wilmsen [Wil06] analyzes identities and topological states of objects in snapshots and derives different types of changes, including continuous changes (such as growing, shrinking, and moving), as well as discrete changes (such as splitting and merging).

Coan and Egenhofer [CE96] studies the possible types of changes incurred by erosion and accretion of small islands Nantucket Sound off the Cape Cod shoreline, and develops a concise formal model that summarizes these changes.

## **2.2 Representation of topological properties**

Many fields of science and engineering dealing with spatial data benefit from formal models that are capable of defining and representing essential topological properties of spatial objects. The following subsections present various approaches to represent the topology of space and objects.

### **2.2.1 Point-set approach**

In this approach, each object that occupies space is modeled as a set of points. The open sets defined on the space provide a framework to specify the topological properties of the objects.

**Definition 2.1.** Let  $X$  be a set of points. A topology  $\tau$  on  $X$  is a collection of subsets of  $X$ , such that

1.  $\emptyset \in \tau$ , and  $X \in \tau$ .
2. The union of any collection of sets in  $\tau$  is in  $\tau$ .
3. The intersection of any finite number of sets in  $\tau$  is in  $\tau$ .

The elements in  $\tau$  are called *open sets*, and their complements in  $X$  are *closed sets*. Usually  $X$  is considered to be the 2-dimensional Euclidean space  $\mathfrak{R}^2$ , and the topology defined on it is the *usual topology*, which is a collection containing all the unions of open discs  $\{(x, y) : (x - a)^2 + (y - b)^2 < r^2, a \in \mathfrak{R}, b \in \mathfrak{R}, r \in \mathfrak{R}\}$ .

From the definition of an open set, the most important concepts in spatial data modeling, including continuity, interior, boundary, exterior, and different kinds of connectivity can be derived. These definitions are given here for the sake of completeness. A further account, which motivates and explains the definitions, can be found in [Lip65].

**Definition 2.2.** Let  $X_1$  and  $X_2$  be topological spaces with topology  $\tau_1$  and  $\tau_2$ , respectively.

A function  $f$  from  $X_1$  to  $X_2$  is continuous iff  $\forall H \in \tau_2, f^{-1}[H] \in \tau_1$ .

**Definition 2.3.** Let  $A$  be a subset of  $\mathfrak{R}^2$ .

1. The union of all open sets of  $X$  contained in  $A$  is called the interior of  $A$ . The intersection of all the closed sets of  $X$  containing  $A$  is called the closure of  $A$ . The difference between the closure and interior of  $A$  is the boundary of  $A$ .
2.  $A$  is defined to be disconnected if there exist open subsets  $G$  and  $H$  of  $X$  such that  $A \cap G$  and  $A \cap H$  are disjoint non-empty sets whose union is  $A$ .  $A$  is connected if it is not disconnected.



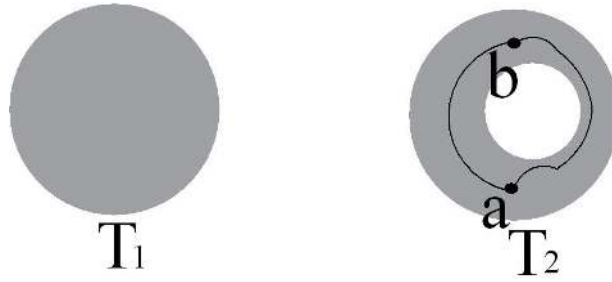


Figure 2.1 Examples of subsets of the topological space

3.  $A$  is defined to be path connected, if for any two points  $\mathbf{a}, \mathbf{b} \in A$ , there exists a path in  $A$  which connects  $\mathbf{a}$  and  $\mathbf{b}$ ; that is, there is a continuous function  $f : [0, 1] \rightarrow A$  with  $f(0) = \mathbf{a}$  and  $f(1) = \mathbf{b}$ .
4.  $A$  is defined to be simply connected if 1) it is path connected, and 2) every closed path can be contracted to a point; that is, given any continuous function  $f$  from the unit circle to  $A$ , there exists a continuous function  $F$  from a closed unit disk to  $A$  such that the restriction of  $F$  to the unit circle is  $f$ .

As an example, Figure 2.1 shows two subsets  $T_1$  and  $T_2$  in space.  $T_1$  is simply connected, whereas  $T_2$  is not, because two different paths connecting points  $a$  and  $b$  in  $T_2$  form a closed path that is impossible to be contracted to a point such that it would be in  $T_2$ .

An important property of the elements in the 2-dimensional Euclidean space is the *Jordan Curve Theorem*, stated as follows. Let  $c$  be a simply-closed curve (a Jordan curve) in  $\mathbb{R}^2$ . Then the complement of the image of  $c$  consists of two distinct connected components. One of these components is bounded and the other is unbounded. Also,  $c$  is the boundary of both components.

The usual topology defined on the 2-dimensional Euclidean space forms the basis of formalizing topological concepts of spatial objects. However, as the number of open sets in the usual topology are infinite, and computers are only capable of dealing with

finite-precision information, additional data structures on the points in space are needed in order to compute their topological properties.

### 2.2.2 Graph based representation

An elementary structure that can be handled easily on computers is a graph. Graphs, including trees, are widely used in modeling topology.

In the field of image processing, the region adjacency graph (RAG) [Pav81] is a commonly-used representation of topological structures in an image. A RAG is an undirected graph  $G(N, A)$  that stores region adjacency information. Each vertex  $n \in N$  of the graph represents a connected component in the image. Two vertices are connected by an arc  $a \in A$  in the graph if, and only if, the components they represent are adjacent. As an example, Figures 2.2(a) and (b) present an image and its corresponding RAG. A RAG provides a way to transfer the image comparison into graph matching. The comparisons are independent of geometry and are tolerant to deformation as long as the topology of the image is preserved. Therefore, a RAG with its extensions has a wide application in region-based image processing, including image retrieval [FTGL04], image registration [AHF05], and building image pyramids [Kro95].

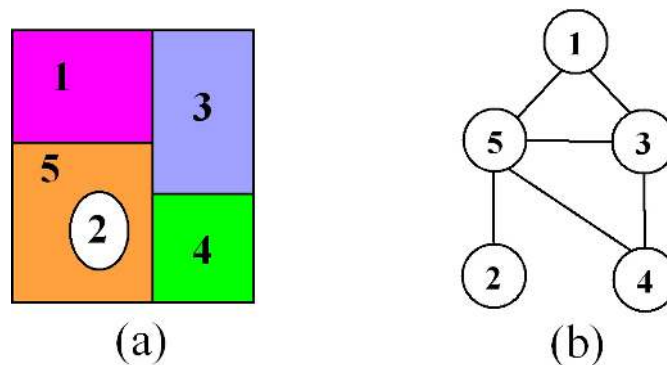


Figure 2.2 An example of a region adjacency graph  
(a) regions, and (b) RAG

As a special case of a RAG, it has been proved by Rosenfeld [Ros74] that the region adjacency graph of a binary image is a tree. Each node of the tree represents a

connected component of the image, and each edge of the tree represents, in addition to the adjacent relation, the surrounding or inside relation between the components. As the surrounding and inside relations are relations with strict partial order, the tree is in fact a rooted tree with the node that represents the background component being the root. Image processing based on the region adjacency tree is faster than that based on the RAG, as efficient algorithms of tree comparison exist. As an example, Costanza and Robinson [CR03] apply the region adjacency tree to encode the topology of fiducials (distinctive markers) for real-time recognition. Figures 2.3(a) and (b) show an example of a binary image (a fiducial) and its representation tree.

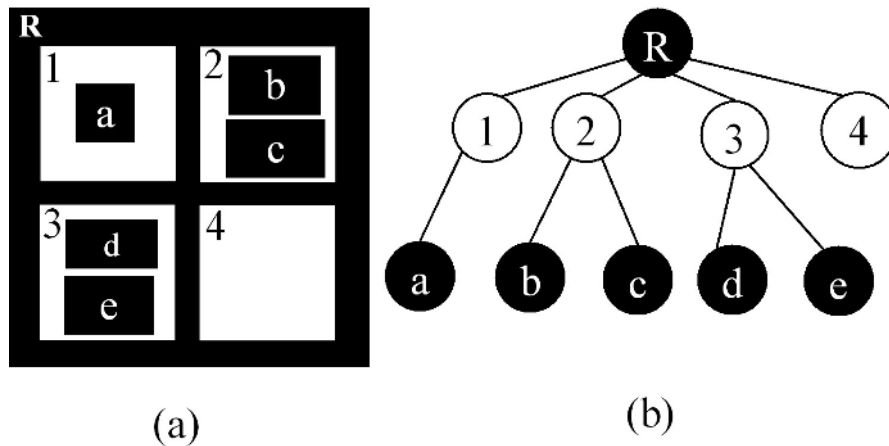


Figure 2.3 Example of region adjacency tree  
(a) a fiducial, and (b) an adjacency tree

From another perspective, objects can also be represented explicitly by different cells in a subdivision of the space [Ape04]. Incidence relations between cells are important topological relations that can be represented by *incidence graphs*. As the incidence relation is asymmetric, an incidence graph is a directed acyclic graph. The nodes of the graph correspond to the cells of the object, and an edge of a graph is defined to be from an  $(i + 1)$ -dimensional cell  $C_1$  to an  $i$ -dimensional cell  $C_2$  that is incident to  $C_1$ . Figure 2.4(a) shows an object, and its corresponding incidence graph is shown in Figure 2.4(b).

Trees are employed in the CAD field to represent the hierarchy formed by the disconnected regions and holes. One example is the disassociative area model (DAM)

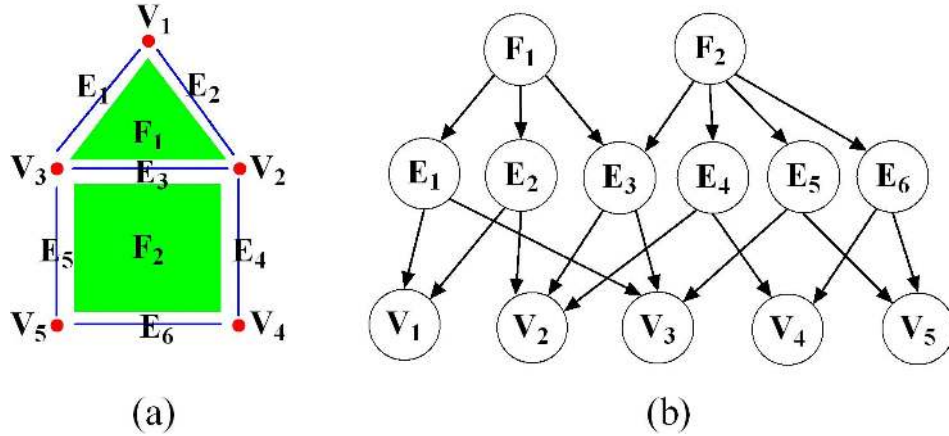


Figure 2.4 Example of incidence graph  
(a) an object, and (b) an incidence graph

[KVV89]. In this model, an area map is first dissected into a set of non-overlapping primitive regions, whose boundaries are simple closed loops. The representation of the topology hinges on the boundary of these primitive regions. The outer boundary of a primitive region is known as an enclosing boundary, and any inner boundary is known as a hole. The containment relations between a complete set of boundaries form a hierarchy, which can be represented by a rooted tree. An example of such a representation is shown in Figures 2.5(a)-(c).

Besides representing the topology of an object, a tree is capable of representing the topology of contours in a scalar field, which is usually referred to as a *contour tree*. The contour tree is obtained by continuous contraction of each contour to a single point. Adjacent contours are contracted to adjacent points. Distinct contours are contracted to distinct points [CSA00]. The resulting tree satisfies that:

1. Each leaf vertex represents the creation or deletion of a component at a local extreme of the parameter.
2. Each interior vertex represents the joining and/or splitting of two or more components at a critical point.

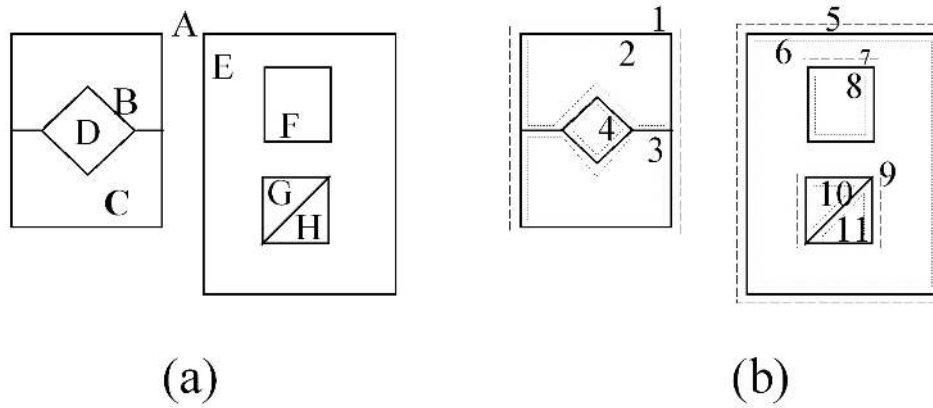


Figure 2.5 An example of DAM model

(a) a set of primitive regions, (b) boundary of regions, and (c) representation tree

- Each edge represents a component in the level sets for all values of the parameter between the values of the data points at each end of the edge.

Figure 2.6 shows an example of a scalar field together with its corresponding contour tree.

### 2.2.3 Topological representation of areal objects

Topological features are important in spatial data modeling for characterizing relationship between spatial objects, such as adjacency, connectivity, and inclusion. Points, lines, and polygons are provided by most spatial data models. Usually, the polygons supported by data models are topologically equivalent to disks. There is some research on extending these spatial data types with more complex semantics that support the representation of richer topological properties.

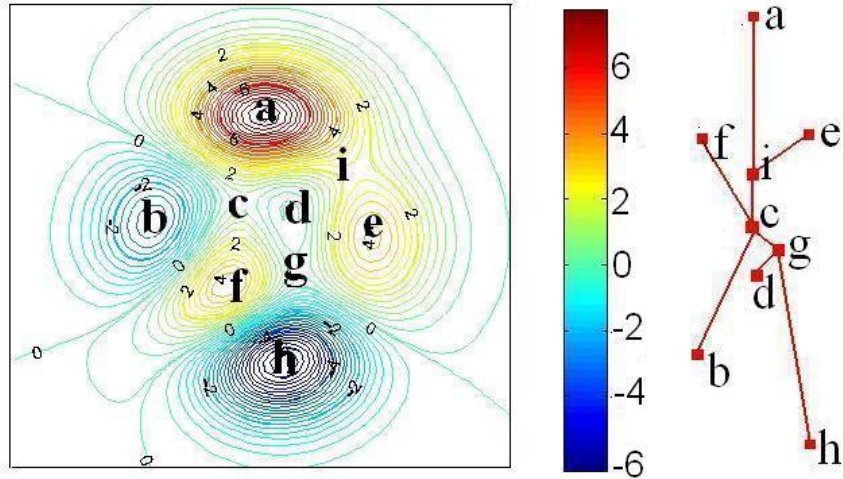


Figure 2.6 A scalar field and its contour tree

The first type of extension aims at supporting representations of regions that are not simply connected. Examples include [ECF94, SV92], in which a region with holes is represented by a set of simple polygons. One of the polygons represents the region, and the others represent the holes contained in the region. Another modeling approach, commonly used in VLSI design or image analysis, represents a spatial region with holes by a single polygon. It connects the boundaries of holes to the outer boundary of the region by introducing an additional line segment for each hole [Kil86, Kro95]. The polygon resulting from the insertion of “bridges” is no longer simple, but many graphics and geometric algorithms handle this type of polygon successfully.

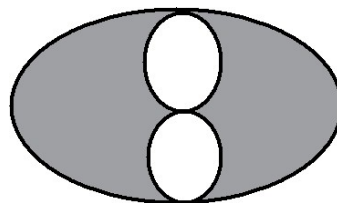


Figure 2.7 An example of areal object with ambiguity

More complex topological properties have also been considered, where areal objects can be disconnected. An areal object can be composed of one or more components, and each component is allowed to have holes. This requires the models to represent a hierarchy of regions, holes, regions with holes, which contain islands with holes to any

finite level. The representation becomes more complex if weakly connected components are considered; that is, the boundaries of an areal object are allowed to touch at a single or a finite number of points. This brings in an ambiguity. For example, the areal object shown in Figure 2.7 has two interpretations. It can either be interpreted as a region with two holes that are weakly connected, or be interpreted as two regions being weakly connected at three points. Examples of models of areal objects representing richer topological properties include [SB06, WB93].

In [SB06], the regions with richer topological properties are called *complex spatial regions*. Complex spatial regions are built from simple regions, which are regularly closed regions homeomorphic to closed disks. In the model any complex spatial region is composed of a set of simple regions, a subset of which represents holes. Several conditions are set up to avoid ambiguity, which include:

1. the boundary between any pair of single regions can touch in at most one point,
2. open hole chains are prevented, and
3. close hole chains are prevented.

Here, an *open hole chain* is a sequence of holes in which any two subsequent holes meet, and both the first and the last hole touch the boundary of the outside region. A *close hole chain* is a sequence of holes in which two subsequent holes meet, and both the first and the last holes in the sequence meet. The three conditions ensure a unique interpretation of the complex regions. Under these conditions, the example in Figure 2.7 can only have the second interpretation since the first interpretation leads to an open hole chain, which is prevented by condition 2.

In [WB93], both graphs and trees are used to model areal objects. The areal objects are built up from basic objects called *atoms*, which are defined as those subsets of  $\mathbb{R}^2$  topologically equivalent to the closed disk. This model is extended by aggre-

gating atoms to regularly closed areal objects, termed *base areas*, whose structure is described by skeleton graphs. Based on the atoms and base areas, the *generic areas* are constructed, in which the recursive construction of areal objects with holes and islands within holes are represented by trees.

In [Pai98], graphs are used to represent the topological invariants of a spatial scene (a set of spatial objects and their embeddings in space). A node of a graph represents an object, and the arcs represent the topological relations between the objects in the scene. A spatial scene with various topological relations, such as meet, disjoint and coveredBy, is described by one or more graphs in a hierarchy. Using the graph based representation, the equivalence between a pair of spatial scenes can be evaluated.

#### 2.2.4 Binary topological relations

An active research area related to topological representation is the study of binary topological relations. Topological relations are spatial relations invariant under topological transformations, such as translation, rotation, and scaling.

Based on the 4-intersection framework [EF91], in which the intersections of the interior and boundary of two simple regions are considered, eight topological relations between simple regions can be specified, namely, disjoint, meet, equal, inside, contains, covers, coveredBy and overlap. The 9-intersection model [EH90] extends the 4-intersection model [EF91] by considering the intersections with respect to the object's exterior. The intersection models have also been applied to study the topological relations between regions with holes [ECF94, EV07], between a line and a region [EM95], between directed line segments [KE06], and between two regions on the sphere [Ege05].

Another extension of the 4-intersection is described in [EF95], which refines the model of empty/non-empty 4-intersections with further topological invariants to account for more details about topological relations. The result of the extension is a model that



is capable of classifying all the topological relations between 2-dimensional regions with non-empty boundary-boundary interactions.

Binary topological relations can be used to derive some types of changes if objects are restricted to be simply connected [Wil06]. For example, let  $A_0$  and  $A_1$  be the regions before and after the change respectively. A *growing* occurs if  $A_1$  contains or covers  $A_0$ . A *shrinking* occurs if  $A_1$  is inside or is covered by  $A_0$ . A *moving* occurs if  $A_1$  overlaps, meets, or is disjoint with  $A_0$ . Finally, *no change* occurs, if  $A_1$  equals  $A_0$ . In addition, if there are two objects in the scenario, the types of changes can be identified according to the sequence of topological relations that hold between the objects over time.

Existing research also focuses on the analysis of transition between binary topological relations [EAT92]. Research in this area aims at building conceptual neighborhood graphs in order to describe possible direct transitions between certain kinds of binary topological relations during the continuous change of two spatial objects. This research has similarities to our work; however, the study of transitions between topological relations is different from the study of changes in topological structure. In the former, the emphasis is on exploring the constraints that continuity imposes on transitions between binary topological relations. In our work, the topological structure of spatial objects is more complex, for example allowing objects to have holes, and our analysis focuses on changes in the structure of such complex objects.

In the field of qualitative spatial reasoning, Cohn and Bennett *et al.* [CBGG97] developed the Region Connection Calculus (RCC). RCC is a topological approach to spatial representation and reasoning where spatial regions are non-empty regular subsets of some topological space. In this work, topological relations between regions are formalized based on a single primitive relation - the “connected” relation  $C$ . For any spatial regions  $S_1$  and  $S_2$ ,  $C(S_1, S_2)$  holds if and only if the closures of  $S_1$  and  $S_2$  are connected, or share a common point. Based on the connected relation, eight basic topo-

logical relations can be formally defined, including  $P(x, y)$  ( $x$  is a part of  $y$ ),  $PP(x, y)$  ( $x$  is a proper part of  $y$ ),  $EQ(x, y)$  ( $x$  is equal to  $y$ ),  $O(x, y)$  ( $x$  overlaps  $y$ ),  $PO(x, y)$  ( $x$  partially overlaps  $y$ ),  $DR(x, y)$  ( $x$  is discrete from  $y$ ),  $EC(x, y)$  ( $x$  is externally connected with  $y$ ),  $TPP(x, y)$  ( $x$  is a tangential proper part of  $y$ ),  $NTPP(x, y)$  ( $x$  is a non-tangential proper part of  $y$ ). The formal semantics associated with the relations form a foundation of reasoning about the spatial relations [CR07, Haz05].

The work of RCC is extended to the “Egg-Yolk” theory [CG96], in which a region with undetermined boundaries is represented by a pair of concentric regions, each with a determinate boundary. Based on an adaptation of the RCC, 46 possible relations between egg-yolk pairs are generated.

## 2.3 Sensor networks

In recent years, sensor networks have attracted a lot of attentions from researchers in both academia and industry [EGHK99, CK03]. A sensor node is a small embedded device able to acquire, process, and transmit data. There are different types of sensor nodes capable of monitoring a wide variety of ambient conditions, such as temperature, light, humidity, pressure, and noise level. By communicating with each other, sensor nodes form networks and work together to cover a larger sensing area.

Many research topics are related to sensor networks. In this section, we first present the characteristics of sensor networks agreed upon by the research community, and then we briefly review literature relevant to environmental data collecting and topology processing in sensor networks.

### 2.3.1 Characteristics of sensor networks

Sensor networks have the following characteristics. First of all, sensor nodes have limitations in energy, memory and computational capacities. As sensor nodes are deployed

with non-rechargeable batteries in many applications, the constraint in energy is the most critical concern among all these limitations [EGHK99]. In order to ensure a long lifetime, most research in sensor networks is directed to the energy-efficient approaches, and low-power consumption becomes the principle of sensor networks design from network architectures to protocols and algorithms [JSAC01, CTLW05].

Second, in a sensor network application, usually communication consumes much more energy than computation. As indicated in [SBA04], “transmitting one bit over radio is at least three orders of magnitude more expensive in terms of energy consumption than executing a single instruction.” Hence, an effective way to save energy is to perform computations before the data is transmitted, so as to reduce communication to the minimum.

Third, it is common for a huge number of sensor nodes to be deployed in an unattended manner without any post-deployment configuration. Also, the high cost in communication prevents sensor nodes from knowing the global state of the whole network. Therefore, distributed and localized algorithms are often preferred in sensor network applications [MSKP01].

Finally, sensor nodes are unreliable. They easily fail to work due to energy exhaustion or hardware failure. Therefore, sensor networks are usually considered to be dynamic, and self-organization is performed during their operation.

### **2.3.2 Environmental data collecting**

Although the research in sensor networks was initially driven by military purposes, sensor networks are currently used in various applications [Xu02]. As the ambient conditions to be monitored are usually distributed over a large area in space, a very straightforward application of sensor networks is to the field of environmental monitoring [BMPW00]. Usually, in these applications, sensor nodes are deployed over a large geographic area

to collect continuous data about physical processes. These sensed data are transmitted back to the base stations for further processing [Xu02]. Example applications include the CORIE [COR08] system, which is a monitoring system for the Columbia river; as well as the ALERT system [ALE08], which is deployed across most of the western United States and is used for flood alert in California and Arizona.

With the energy constraint, energy-efficient approaches are usually preferred in the environmental data collecting research literature. The basic data aggregations operations, such as MAX, MIN, AVERAGE, are the first type of functions that can be performed by sensor networks with low communication cost. These functions are supported by sensor database systems, like TinyDB [MFHH05], as standard operations. The aggregations are usually coupled with tree-based routing, where each routing tree is rooted at the sink. During the aggregation, data are transmitted from leafs to the root. Aggregation can take place at each node in the tree before it resends the data toward the root, and therefore the communication cost can be significantly reduced. In addition to the basic aggregations, more generalized aggregation results can be obtained via wavelets or distributed regressions to construct summaries of the entire sensed data [NGSA08, HW04].

As the environmental data are often expected to be spatially and temporally correlated, the communication cost can be reduced by minimizing the correlation redundancy. To minimize the temporal correlation redundancy, a node can transmit data only if a significant change in the sensed data is observed since the last reporting round. Sharaf *et al.* [SBLC03] use this mechanism to support continuous aggregate queries with bounded error. To minimize the spatial correlation redundancy, a node does not transmit data if its measurement is not significantly different from its neighbors. Usually both compression mechanisms are combined together for better performance. Examples of applications using both mechanisms include [SBY06, MLNL04], in which sensor net-

works transfer a large portion or even the entire sensed data back to the base station in order to form reports, such as contour maps, of the whole phenomena.

The spatial correlation of the sensed data also allows cluster-based approaches to have a good compression ratio during data transmission. In such approaches, each sensor first sends the data to a local cluster head, which is able to detect the spatial correlation and perform compression before resending the data back to the sink. An example approach in this category is [CDHH06], in which dynamic probabilistic models are exploited to predict sensor values in a cluster based on both the spatial and temporal neighboring information. The cluster heads report to the sink only if a significant difference between the predicted and actual sensing value is detected. The impact of different spatial correlations on optimal clustering schemes has been studied by Pattem *et al.* [PKG04], who discover a clustering scheme that has near-optimal performance for a wide range of spatial correlations.

As the representation of the boundary provides a more concise description of large-scale phenomena than an enumeration of all the nodes, some approaches focus on the detection of boundaries of the phenomena. By only having the nodes located near the boundary to report, energy efficiency is achieved [SO05], and the communication can be further reduced by performing aggregation during the transmission of the boundary information [LL07, ZW07]. In this research area, approaches to localized boundary detection are important. Chintalapudi and Govindan [CG03] propose three algorithms for sensor nodes to construct the boundary according to local information. The proposed algorithms include the statistical approach, the image processing approach, and the classifier-based approach. Wang *et al.* [WGM06] propose a distributed algorithm that correctly detects nodes on the boundary based only on the network topology.

### 2.3.3 Topology monitoring in sensor networks

A straightforward application of sensor networks is the monitoring of geographic phenomena [NSC<sup>+</sup>04]. Previous research either focuses on proposing energy-efficient approaches to transmitting the entire sensed data back to base stations [SBLC03, SBY06], or focuses on providing important spatial properties of the phenomena. For example, the snake model used by Jin and Nittel [JN08] is able to derive the area and centroid of a deformable 2D object over time.

Recently, there is an increasing interest in considering topological information when processing sensed data. Gandhi, Hershberger and Suri [GHS07] emphasize the topology of the isolines in a scalar field and propose an approach that approximates a family of isolines by a collection of topology-preserving polygons. Sarkar and Zhu *et al.* [SZG<sup>+</sup>08] present a distributed algorithm for the construction of a contour tree to represent the topological structure of contours in a scalar field, based on which isoline queries can be enabled.

In addition to the research of topology detection in static data, detection of topology in time-varying data with sensor networks is also considered. Worboys and Duckham [WD06] provide a computational model for sensor networks to detect global high-level topological changes based on low-level “snapshot” of spatiotemporal data. Zhu *et al.* [ZSGM08] propose a distributed algorithm for sensor networks to maintain contours (or boundaries) of a binary object incrementally as they deform, while guaranteeing that the maintained contours capture the global topological features of the object boundary. Current research has also made initial attempts to topological change detection using wireless sensor networks. Farah *et al.* [FZWN08] provide initial attempts to detect topological changes in responsive sensor networks by an event-driven approach. Sadeq [Sad07] proposed the idea of detecting topological changes by maintaining the boundary

state of areal objects. In [JW08], we present a topological change detection approach based on local aggregation.

Another area of research of sensor networks related to the detection of areal object topology is the retrieval of network topology itself, especially the connectivity among sensor nodes in their communication graphs [LY06]. Deb, Bhatnagar and Nath [DBN04] in their STREAM algorithm enable the retrieval of the entire network topology with predefined resolution, which allows users to make a trade-off between topology details and resources expended. As holes in the sensor network are of primary interest in network management, some existing work detects sensor network holes using methods based on topological information [Fun05], location information [FGG06], or statistical information [FKP<sup>+</sup>04]. Although the detection of some topological properties of areal objects can be achieved based on the analysis of the network topology among sensor nodes that observe the areal object [FGG06], the cost for network topology detection is very high. Therefore, more efficient areal object topology detection methods are needed.

## 2.4 Summary

This chapter reviewed related work on the modeling and detection of topological events. Various definitions of events and event classifications were discussed. Different models of static topological properties representation were also reviewed. Finally, existing work of topology detection in sensor networks was presented.

## Chapter 3

### TOPOLOGICAL CHANGES OF AN AREAL OBJECT

Areal objects arising from dynamic fields have complex topological structures. In this chapter, we introduce the basic definitions of dynamic fields, areal objects, and topological changes of areal objects. A model is presented for the classification of topological changes. Based on the model, important properties of these topological changes are proved.

#### 3.1 Dynamic fields and areal objects

From a set-theoretic point of view, a dynamic field is a function [Gal04]

$$f : S \times T \rightarrow V$$

in which  $S$ ,  $T$ ,  $V$  are the spatial domain  $\mathfrak{R}^2$ , the temporal domain, and a set of possible scalar values, respectively. Taking a snapshot view, the dynamic field can be reorganized into the form

$$f : T \rightarrow (S \rightarrow V)$$

In this thesis, we assume that  $f$  is a binary field, and the set  $V$  of scalar values is  $\{0, 1\}$ . Therefore, the function  $f$  has an equivalent specification:



$$\widehat{f}: T \rightarrow 2^S$$

For  $\forall t \in T$ ,  $\widehat{f}(t)$  gives the set of points that are mapped to 1 at time  $t$ ; that is,  $\widehat{f}(t) = \{s : s \in S, f(t, s) = 1\}$ . In this way, the dynamic field is converted into  $\widehat{f}$ , which is a function from the temporal domain to a collection of subsets of  $\mathfrak{R}^2$ . Each subset of the  $\mathfrak{R}^2$  is an areal object, and the dynamic field can be considered as a description of the evolution of an areal object. As the areal object evolves through time, its topological properties may change, and topological changes can be identified.

### 3.2 Areal objects

An areal object is a subset of  $\mathfrak{R}^2$ . As we are using areal objects to represent geographic phenomena at a particular time, an areal object is usually assumed to be bounded and to be composed of a finite number of connected components. We call these components *regions*. Each region has zero or a finite number of holes. Besides that, any regions or holes of an areal object are assumed to be purely two-dimensional; that is, they cannot be single points or line segments [SB06]. Definitions 3.1 and 3.2 formalize the concept of bounded subset, and definition 3.3 defines areal objects.

**Definition 3.1.** *Let  $C$  be a simple closed curve (i.e., a Jordan curve) in  $\mathfrak{R}^2$ . According to the Jordan Curve Theorem, the complement of  $C$  consists of two distinct connected components. One of these components is bounded, and the other is unbounded. We define the bounded component to be the interior of  $C$ , and the unbounded component to be the exterior of  $C$ .*

**Definition 3.2.** *Let  $R$  be a subset of  $\mathfrak{R}^2$ .  $R$  is defined to be bounded if there is a Jordan curve in  $\mathfrak{R}^2$ , whose interior contains  $R$ .*

An areal object in  $\mathfrak{R}^2$  is defined as follows:

**Definition 3.3.** An areal object is a bounded set  $R$  in  $\mathbb{R}^2$ , such that:

1.  $R$  is regularly closed; that is, the closure of the interior of  $R$  is  $R$ .
2. Both interiors of  $R$  and  $\mathbb{R}^2 \setminus R$  have a finite number of connected components.

The interior of an areal object, defined in definition 3.3, is locally Euclidean; that is, every point in the interior of  $R$  has a neighborhood that resembles the 2-dimensional Euclidean space. Hence definition 3.3 allows us to define areal objects to be a union of a finite number of 2-manifolds together with their boundaries. This definition excludes some other cases: the first condition excludes the cases that contain 0-dimensional points or 1-dimensional lines. The second condition excludes the “pathological” cases, such as the infinite number of components shown in Figure 3.1.

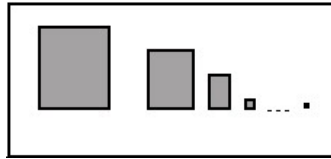


Figure 3.1 An example of “pathological” cases

The topological properties of areal objects include different types of connectivity of the regions and holes and the topological structure of the areal object. For example, a region can be simply connected (consisting of one piece and no holes), and a hole can be simply connected (consisting of one piece and does not contain any regions as its islands.) In addition, the areal object can be weakly connected, which is defined as follows.

**Definition 3.4.** An areal object  $R$  is defined to be weakly connected if the number of connected components of  $R$  or of  $S \setminus R$  can be changed by removing a finite number of points from  $R$ . Otherwise, it is defined to be strongly connected.

The weak/strong connectivity of an areal object is independent of whether it is connected or disconnected. A weakly connected areal object may be either connected

or disconnected, as may be a strongly connected areal object. Figure 3.2 shows disconnected and connected areal objects  $R_1$  and  $R_2$ , both being weakly connected. The number of connected components of  $R_1$  can be increased by removing the point  $P_1$  from  $R_1$ . The number of connected components of  $S \setminus R_2$  can be decreased by removing the point  $P_2$  from  $R_2$ .

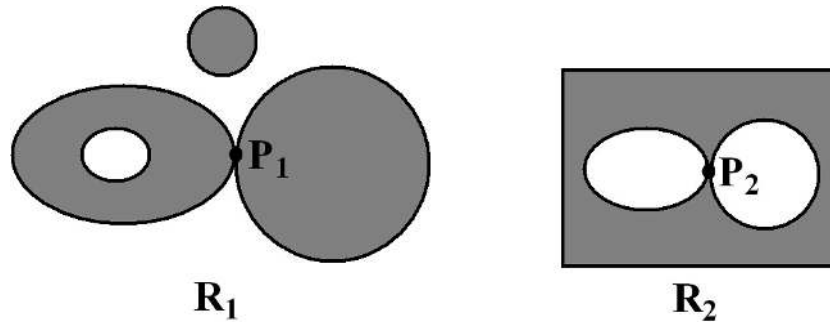


Figure 3.2 Examples of weakly connected areal objects

Topological structures of areal objects may change as they evolve over time. In the following sections, we discuss the possible topological relations between the components and provide a model based on tree morphisms to represent and specify different types of topological changes.

### 3.3 Topological relations between components of an areal object

Given an areal object  $R$ , connected components of  $R$  are said to be its *positive components* and connected components of the complement of  $R$  are said to be its *negative components*. Both positive and negative components are referred to as *components*. We note that components of an areal object form a partition of the whole space  $\mathfrak{R}^2$ . An areal object must have one and may have more than one positive component as well as more than one negative component, and both positive and negative components may have holes. As an example, Figure 3.3(a) shows an areal object, whose positive and

negative components are represented by the shaded areas in Figures 3.3(b) and 3.3(c), respectively. In the figure, the outer rectangles represent the extent of the  $\mathbb{R}^2$ .

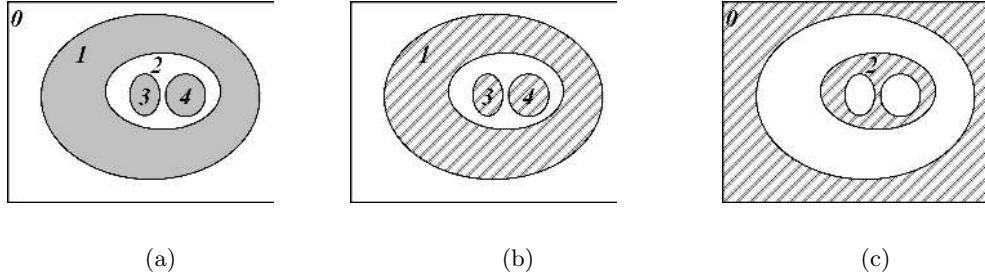


Figure 3.3 An example of tree representation of components  
(a) an areal object, (b) positive components, and (c) negative components

The topological relations between the components characterize the topological structure of the whole areal object. The closure of each component is a region, possibly with holes. By the definition of components, the interiors of different components never intersect. Therefore, the topological relations between a pair of components in this thesis are a subset of those specified based on the intersection models. Let  $A$  and  $B$  be two components. We analyze possible topological relations between  $A$  and  $B$  by the following five steps.

**Step 1**, consider the case in which both  $A$  and  $B$  are simple regions without holes. As  $A^\circ \cap B^\circ = \emptyset$ , by the 9-intersection model [EH90], the topological relations between  $A$  and  $B$  can only be either *meet* or *disjoint*.

**Step 2**, consider the case in which  $A$  is a simple region without holes and  $B$  is a region with a hole. From [EV07], we use the generalized region  $B^*$  as the union of the holed region and the hole and  $B_H$  as the hole of  $B$ .

As  $A^\circ \cap B^\circ = \emptyset$ , it follows that  $A^\circ \subseteq (B_H)^\circ$  whenever  $A^\circ \cap (B^*)^\circ \neq \emptyset$ ; that is, whenever  $A^\circ$  intersects  $(B^*)^\circ$ ,  $A^\circ$  must be completely contained in the interior of  $B_H$ , and therefore intersects neither the boundary nor the exterior of  $B_H$ . In terms of the 9-intersection model, if the relation between  $A$  and  $B^*$  is contained in the set of

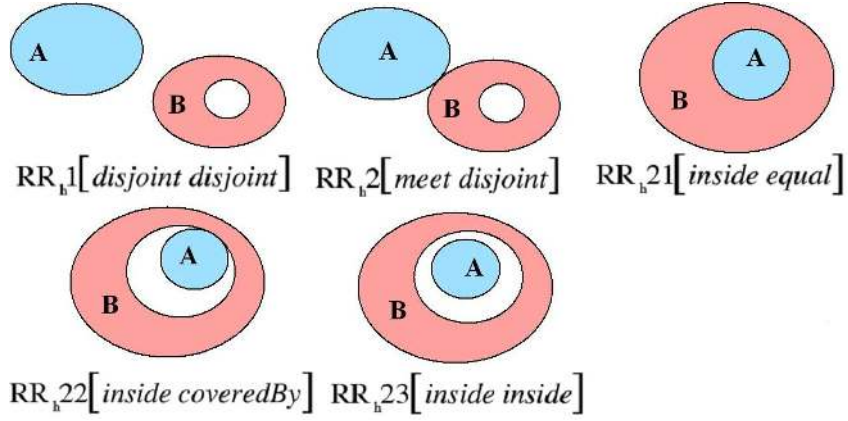


Figure 3.4 Topological relations between  $R$  and  $R_h$

$\{contains, covers, equal, overlap, inside, coveredBy\}$ , the relation between  $A$  and  $B_H$  must be contained in the set of  $\{inside, coveredBy, equal\}$ . With this constraint, among all the 23 topological relations between a region and a region with a hole [EV07], the topological relation between  $A$  and  $B$  can only be one of the five, including  $RR_h1$ ,  $RR_h2$ ,  $RR_h21$ ,  $RR_h22$ ,  $RR_h23$ . Figure 3.4 shows example configurations for each of the topological relations between  $A$  and  $B$ .

**Step 3**, consider the case in which both  $A$  and  $B$  are single-holed regions. From [VE08], we use  $A^*$  and  $B^*$  to represent the generalized regions of  $A$  and  $B$ , respectively, and use  $A_H$  and  $B_H$  to represent the holes of  $A$  and  $B$ , respectively.

As  $A^\circ \cap B^\circ = \emptyset$ , it follows that whenever  $(A^*)^\circ \cap (B^*)^\circ \neq \emptyset$ , either  $(A^*)^\circ \subseteq (A_H)^\circ$  or  $(B^*)^\circ \subseteq (A_H)^\circ$ . In terms of the 9-intersection model, if the relation between  $A^*$  and  $B^*$  is in the set of  $\{contains, covers, equal, overlap, inside, coveredBy\}$ , either the relation between  $A^*$  and  $B_H$  is in the set of  $\{inside, coveredBy, equal\}$ , or the relation between  $A_H$  and  $B^*$  is in the set of  $\{contains, covers, equal\}$ . With this constraint, among all the 152 topological relations between two single-holed regions [VE08], the topological relations between  $A$  and  $B$  can only be one of the eight, including  $t_1$ ,  $t_2$ ,  $t_{99}$ ,  $t_{100}$ ,  $t_{109}$ ,  $t_{141}$ ,  $t_{143}$ , and  $t_{152}$ . Figure 3.5 shows example configurations for each of the topological relations between  $A$  and  $B$ .

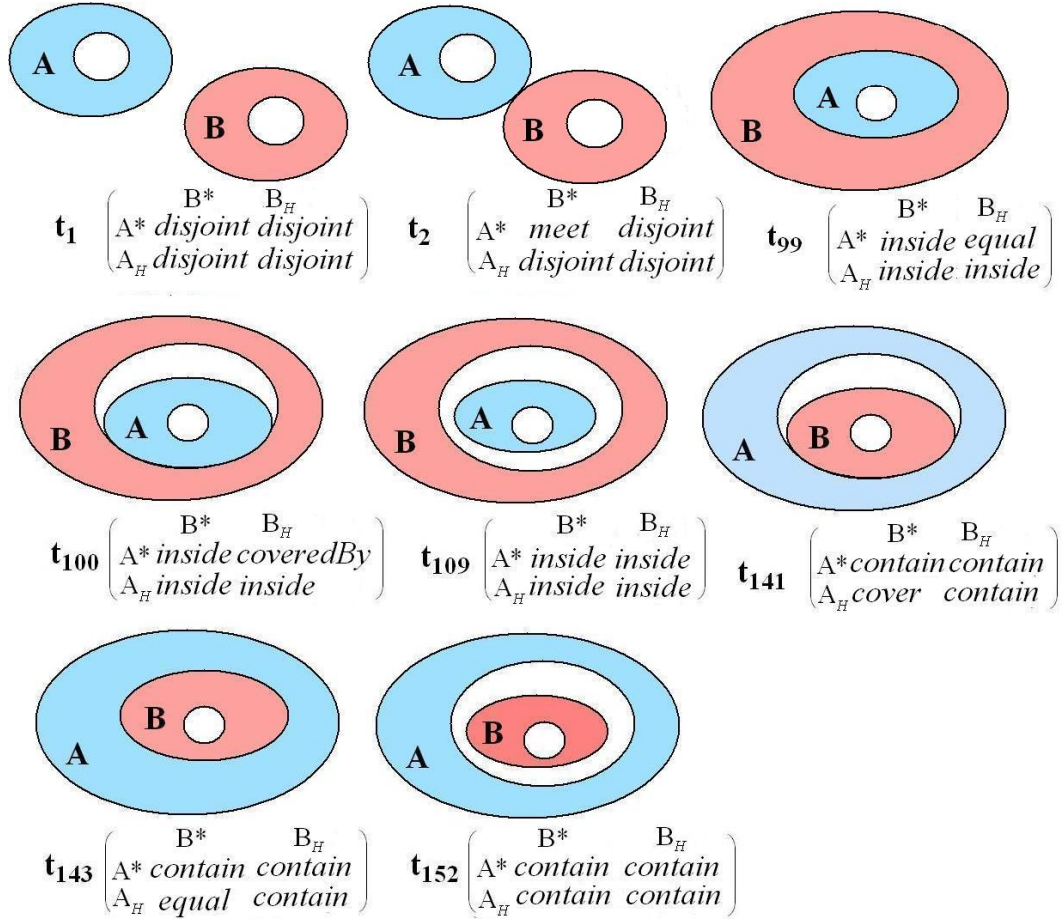


Figure 3.5 Topological relations between a pair of  $R_h$ s

**Step 4**, consider the case in which  $A$  is a simple region without a hole and  $B$  is a region with multiple holes. We use  $B^*$  to represent the union of  $B$  and all its holes.

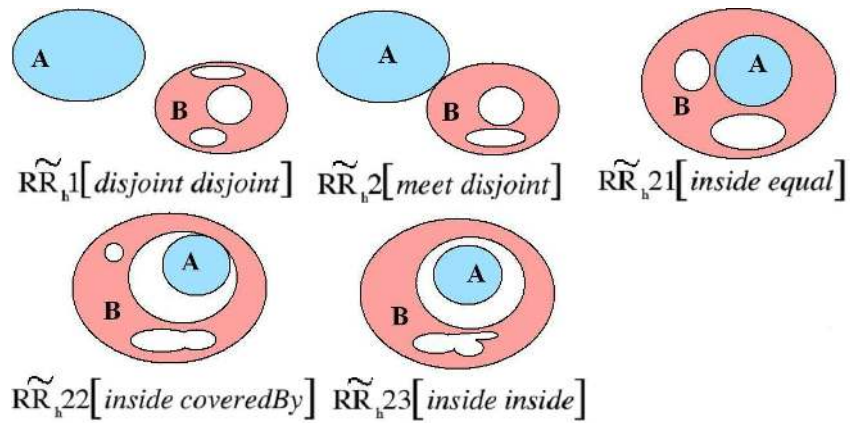


Figure 3.6 Topological relations between  $R$  and  $\widetilde{R}_h$

As  $A^\circ \cap B^\circ = \emptyset$ , either of the following conditions must be satisfied: (1)  $A^\circ$  is completely contained in the interior of a hole of  $B$ , or (2)  $A^\circ \cap (B^*)^\circ = \emptyset$ . If condition

(1) is satisfied, we use  $B_H$  to represent the hole that contains  $A^\circ$ . Otherwise, we use  $B_H$  to represent one of the holes of  $B$ . The possible topological configurations between  $A$ ,  $B^*$  and  $B_H$  are the same as the results derived in step 2. Therefore, the topological relations between components  $A$  and  $B$  are an extension of the results in step 2, in which  $B$  is allowed to have multiple holes. We name these relations  $R\tilde{R}_h1$ ,  $R\tilde{R}_h2$ ,  $R\tilde{R}_h21$ ,  $R\tilde{R}_h22$ , and  $R\tilde{R}_h23$ , respectively. Figure 3.6 shows example configurations for each of the topological relations between  $A$  and  $B$ .

**Finally**, consider the case in which both  $A$  and  $B$  are regions with multiple holes. We use  $A^*$  to represent the union of  $A$  and all its holes, and  $B^*$  for component  $B$ .

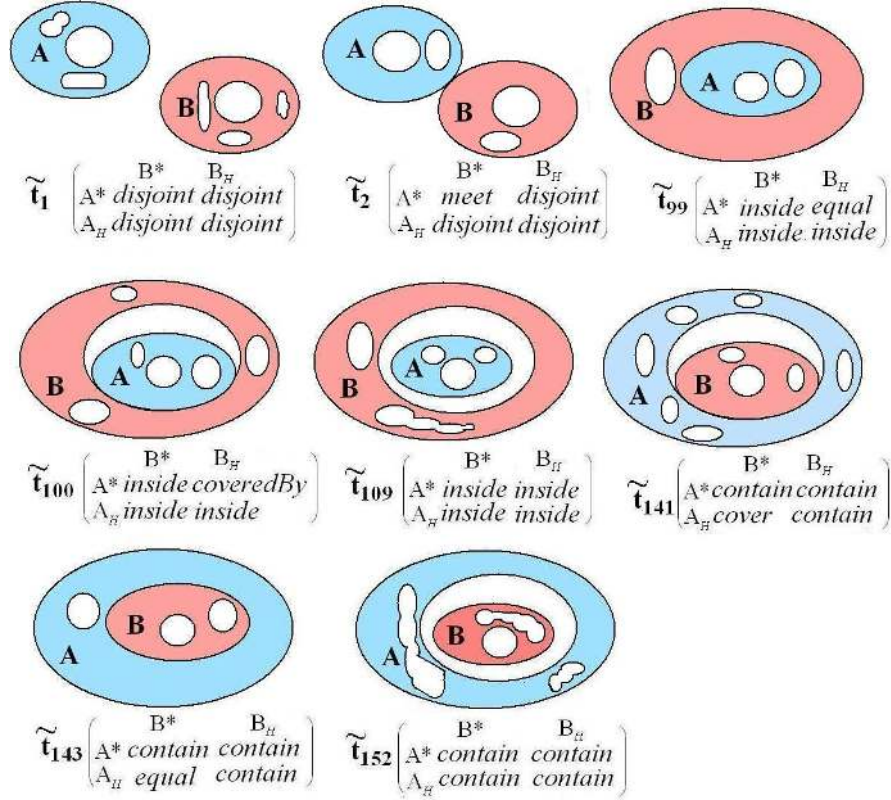


Figure 3.7 Topological relations between a pair of  $\tilde{R}_h$ s

As  $A^\circ \cap B^\circ = \emptyset$ , one of the following three conditions must be satisfied: (1)  $(A^*)^\circ$  is completely contained in the interior of a hole of  $B$ , (2)  $(B^*)^\circ$  is completely contained in the interior of a hole of  $A$ , or (3)  $(A^*)^\circ \cap (B^*)^\circ = \emptyset$ . If condition (1) is satisfied, we use  $B_H$  to represent the hole of  $B$  that contains  $(A^*)^\circ$ , and we use  $A_H$  to represent a hole of

Table 3.1 Topological relations between components

Topological relation $R(A, B)$	Type of A	Type of B	adjacent-to relation	surrounded-by relation
<i>disjoint</i>	Simple	Simple	No	No
<i>meet</i>	Simple	Simple	Yes	No
$RR_h$ 1	Simple	Single-holed	No	No
$RR_h$ 2	Simple	Single-holed	Yes	No
$RR_h$ 21	Simple	Single-holed	Yes	B surrounds A
$RR_h$ 22	Simple	Single-holed	Yes	B surrounds A
$RR_h$ 23	Simple	Single-holed	No	B surrounds A
$t_1$	Single-holed	Single-holed	No	No
$t_2$	Single-holed	Single-holed	Yes	No
$t_{99}$	Single-holed	Single-holed	Yes	B surrounds A
$t_{100}$	Single-holed	Single-holed	Yes	B surrounds A
$t_{109}$	Single-holed	Single-holed	No	B surrounds A
$t_{141}$	Single-holed	Single-holed	No	A surrounds B
$t_{143}$	Single-holed	Single-holed	Yes	A surrounds B
$t_{152}$	Single-holed	Single-holed	No	A surrounds B
$\widetilde{RR}_h$ 1	Simple	Multi-holed	No	No
$\widetilde{RR}_h$ 2	Simple	Multi-holed	Yes	No
$\widetilde{RR}_h$ 21	Simple	Multi-holed	Yes	B surrounds A
$\widetilde{RR}_h$ 22	Simple	Multi-holed	Yes	B surrounds A
$\widetilde{RR}_h$ 23	Simple	Multi-holed	No	B surrounds A
$\widetilde{t}_1$	Multi-holed	Multi-holed	No	No
$\widetilde{t}_2$	Multi-holed	Multi-holed	Yes	No
$\widetilde{t}_{99}$	Multi-holed	Multi-holed	Yes	B surrounds A
$\widetilde{t}_{100}$	Multi-holed	Multi-holed	Yes	B surrounds A
$\widetilde{t}_{109}$	Multi-holed	Multi-holed	No	B surrounds A
$\widetilde{t}_{141}$	Multi-holed	Multi-holed	No	A surrounds B
$\widetilde{t}_{143}$	Multi-holed	Multi-holed	Yes	A surrounds B
$\widetilde{t}_{152}$	Multi-holed	Multi-holed	No	A surrounds B

A. If condition (2) is satisfied, we use  $A_H$  to represent the hole of  $A$  that contains  $(B^*)^\circ$ , and we use  $B_H$  to represent a hole of  $B$ . Otherwise, we use  $A_H$  and  $B_H$  to represent two holes of  $A$  and  $B$ , respectively. The possible topological configurations between  $A^*$ ,  $A_H$ ,  $B^*$  and  $B_H$  are the same as the results of step 3. Therefore, the topological relations between components  $A$  and  $B$  are an extension of the results in step 3, in which both  $A$  and  $B$  are allowed to have multiple holes. We name these relations  $\widetilde{t}_1$ ,  $\widetilde{t}_2$ ,  $\widetilde{t}_{99}$ ,  $\widetilde{t}_{100}$ ,  $\widetilde{t}_{109}$ ,  $\widetilde{t}_{141}$ ,  $\widetilde{t}_{143}$ , and  $\widetilde{t}_{152}$ , respectively. Figure 3.7 shows example configurations for each of the topological relations between  $A$  and  $B$ .



In all, based on the intersection models, we have explored the topological relations between two components. In this thesis, we are most interested in the adjacency relations and the surrounded-by relations between the components. They are defined as follows:

**Definition 3.5.** *Let  $X_1$  and  $X_2$  be a pair of different components in a partition of  $\mathbb{R}^2$ .*

1.  $X_1$  is said to be adjacent to  $X_2$  if the boundary of  $X_1$  intersects the boundary of  $X_2$ .
2.  $X_1$  is said to be surrounded by  $X_2$  if any path that connects a point in the closure of  $X_1$  to a point at infinity intersects the closure of  $X_2 \setminus X_1$ .  $X_1$  is said to surround  $X_2$ , if  $X_2$  is surrounded by  $X_1$ .

Table 3.1 shows the adjacent-to and surrounded-by relations between the all the possible configurations of components  $A$  and  $B$  whose topological relations are specified by the intersection models. As shown in the table, the adjacency and surrounded-by relations together with the types of the components are able to differentiate most of the topological relations specified by the intersection models. Therefore, when specifying and detecting topological changes, we focus on the adjacency and surrounded-by relations.

### 3.4 Basic definitions

This section gives the formal definitions that relate to trees and tree morphisms, which form the basis for modeling the topological changes. (Note: in this and the following sections, trees always refer to rooted trees.)

**Definition 3.6.** *A graph  $G$  is a pair  $\langle V, E \rangle$ , where  $V$  is the set of vertices in the graph, and  $E$  is a set of subsets of  $V$  representing the edges of the graph. Each element in  $E$  has the form  $\{v_1, v_2\}$ , where  $v_1$  and  $v_2$  are different vertices in  $V$ .*

**Definition 3.7.** In a graph  $G = \langle V, E \rangle$ , a path is a sequence of vertices of the form:

$$[v_1, v_2, \dots, v_{k-1}, v_k]$$

where  $v_i \in V$  for  $i = 1, 2, \dots, k$ , and  $\{v_i, v_{i+1}\} \in E$  for  $i = 1, 2, \dots, k - 1$ . A path is defined to be simple, if  $\forall i, j \in \{1, 2, \dots, k\}$ ,  $i \neq j$  implies  $v_i \neq v_j$ . A cycle is defined to be a path  $[v_1, v_2, \dots, v_{k-1}, v_k]$  such that  $v_1 = v_k$ . A cycle is defined to be simple, if  $\forall i, j \in \{1, 2, \dots, k - 1\}$ ,  $i \neq j$  implies  $v_i \neq v_j$ .

We will use  $V(G)$ ,  $E(G)$  to represent the set of vertices and edges of the graph  $G$  respectively. We say vertices  $v_1$  and  $v_2$  are *adjacent* in  $G$ , if  $\{v_1, v_2\} \in E(G)$ .

**Definition 3.8.** A graph morphism from a graph  $G_1$  to a graph  $G_2$  is a function  $\varphi : V(G_1) \rightarrow V(G_2)$  such that for all  $v_i, v_j \in V(G_1)$ ,  $\varphi(v_i)$  and  $\varphi(v_j)$  are adjacent in  $G_2$  if and only if  $v_i$  and  $v_j$  are adjacent in  $G_1$ .

**Definition 3.9.** A tree  $T$  is defined to be a graph with the properties that:

1. For all different vertices  $v_1, v_2$ , there is one and only one simple path in  $T$  that connects  $v_1$  and  $v_2$ .
2. There is a distinguished vertex  $r$  called the root of the tree.

To emphasize the root of the tree, we will use a triple  $\langle V, E, r \rangle$  to represent a tree.

**Definition 3.10.** Given two trees  $T_1 = \langle V_1, E_1, r_1 \rangle$  and  $T_2 = \langle V_2, E_2, r_2 \rangle$ , a tree morphism  $\varphi$  from  $T_1$  to  $T_2$  is defined to be the graph morphism  $\varphi$  from  $T_1$  to  $T_2$ , with an additional requirement that  $\varphi(r_1) = r_2$ .

A tree morphism is *injective* if and only if  $\varphi$  is an injective function; that is, distinct vertices of  $T_1$  are mapped to distinct vertices of  $T_2$  through  $\varphi$ . A tree morphism

is *surjective* if and only if  $\varphi$  is a surjective function; that is, every vertex of  $T_2$  is mapped onto by at least one vertex of  $T_1$  through  $\varphi$ . A tree morphism is an *isomorphism*, if and only if it is both an injective and a surjective tree morphism.

### 3.5 Tree representation of areal object

This section considers the representation of components and topological relations of an areal object. The surrounded-by relations of the components in a snapshot satisfy the following properties [BD07]:

1. Transitivity: for any components  $C_1$ ,  $C_2$ , and  $C_3$ , whenever  $C_1$  is surrounded by  $C_2$ , and  $C_2$  is surrounded by  $C_3$ ,  $C_1$  is surrounded by  $C_3$ .
2. Asymmetry: for any components  $C_1$  and  $C_2$ , if  $C_1$  is surrounded by  $C_2$ , then  $C_2$  cannot be surrounded by  $C_1$ .
3. The root property: there is one and only one component  $C$  such that all the other components are surrounded by  $C$ .
4. No-partial-overlap: for any two distinct components  $C_1$  and  $C_2$  surrounding the same component, it holds that either  $C_1$  is surrounded by  $C_2$ , or  $C_2$  is surrounded by  $C_1$ .

According to these properties, rooted tree structures are employed to represent the surrounded by relation between all the components. A rooted tree is a special type of directed tree, in which edges have a natural orientation (being away from the root). In the rooted tree, each vertex represents a component, and the direct descendants of each vertex represent the components that are both adjacent to and surrounded by the component it represents. A component  $C_1$  is surrounded by a component  $C_2$  if and only if there is a directed path from the vertex representing  $C_2$  to the vertex representing  $C_1$ .

Figure 3.8 shows such a tree representation of an areal object. The root of the tree is doubly circled and directions of edges are indicated by arrows.

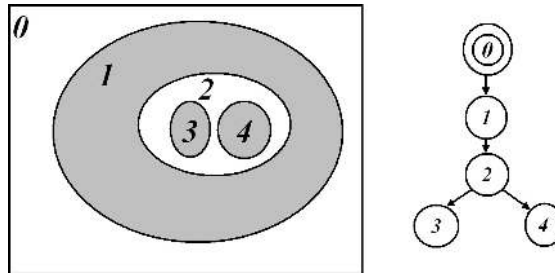


Figure 3.8 The representation tree of an areal object

### 3.5.1 Basic topological changes

Topological changes occur when components appear, disappear, merge, split, etc. All of these topological changes have corresponding changes in the tree structure. We first define some basic changes of the tree structure, from which all the changes of interest can be constructed.

Suppose  $T_1$  and  $T_2$  are trees representing the topological structures of an areal object at time  $t_1$  and  $t_2$  ( $t_1 < t_2$ ). A basic change is denoted by the expression “ $T_1 \xrightarrow{\gamma} T_2$ ”, where  $\gamma$  is the change, and can be represented by a single morphism  $\varphi$ , either from  $T_1$  to  $T_2$  or from  $T_2$  to  $T_1$ . Five types of basic changes can be specified according to the properties of the tree morphisms between  $T_1$  and  $T_2$ .

**Definition 3.11.** A basic change  $T_1 \xrightarrow{\gamma} T_2$  is one of the following types:

- (1)  $T_1 \xrightarrow{\gamma} T_2$  is of type **insert**, if the effect of the change can be represented by an injective but not surjective tree morphism  $\varphi$  from  $T_1$  to  $T_2$ .
- (2)  $T_1 \xrightarrow{\gamma} T_2$  is of type **merge**, if the effect of the change can be represented by a surjective but not injective tree morphism  $\varphi$  from  $T_1$  to  $T_2$ .
- (3)  $T_1 \xrightarrow{\gamma} T_2$  is of type **delete**, if the effect of the change can be represented by an injective but not surjective tree morphism  $\varphi$  from  $T_2$  to  $T_1$ .

(4)  $T_1 \xrightarrow{\gamma} T_2$  is of type **split**, if the effect of the change can be represented by a surjective but not injective tree morphism  $\varphi$  from  $T_2$  to  $T_1$ .

(5)  $T_1 \xrightarrow{\gamma} T_2$  is of type **no change**, if the effect of the change can be represented by a tree isomorphism  $\varphi$  either from  $T_1$  to  $T_2$ , or from  $T_2$  to  $T_1$ .

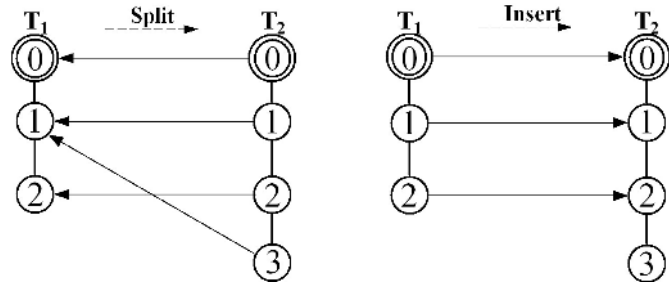


Figure 3.9 Representation of different topological changes

As an example, Figure 3.9 shows representations of a basic split and a basic insert between times  $t_1$  and  $t_2$ . The dashed arrows show the direction of changes, and solid arrows between vertices show the morphisms. Both representations indicate different ways that the topological structure of an areal object changes. A change modeled by the basic split is shown in Figure 3.10(a), in which component 2 evolves its shape to engulf a new component 3. A change modeled by the basic insert is shown in Figure 3.10(b), in which component 3 arises differently, this time emerging and growing inside component 2.

A basic change of type insert, split, merge, and delete causes the essential changes in the topological structure of an areal object, and it is called a *non-trivial topological change*, NTTC for short. A basic change of type no-change does not cause any changes in the topological structure of an areal object, and it is called a *trivial topological change*. In representation, we can omit trivial topological changes for simplicity.

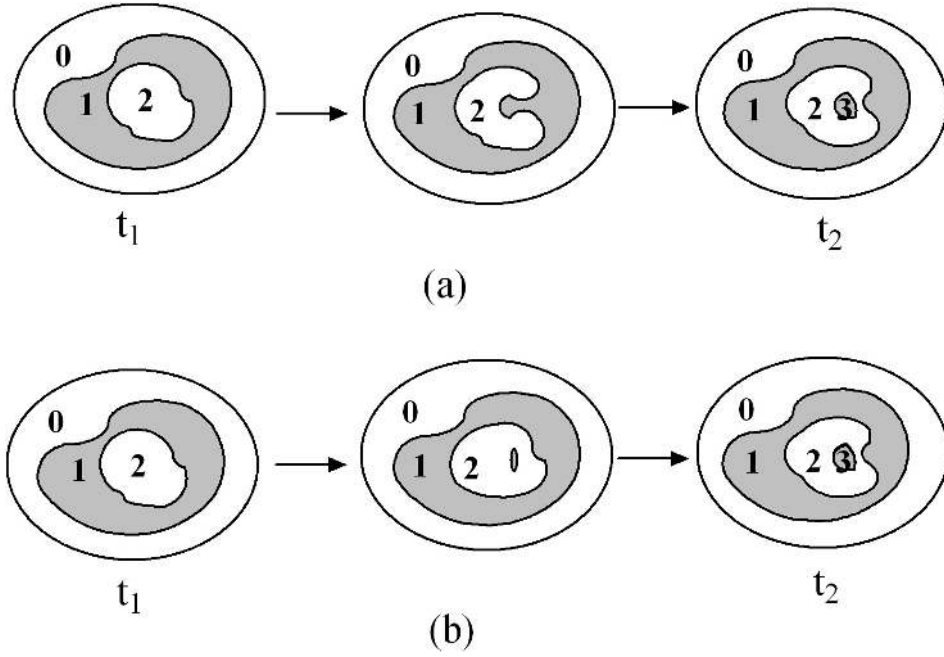


Figure 3.10 Two different topological changes

### 3.5.2 Complex topological changes

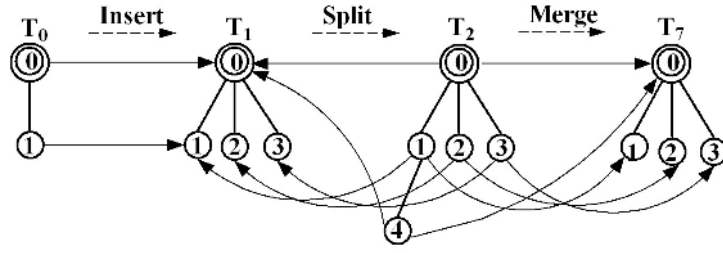
As an areal object evolves through time, a sequence of basic changes is established. We define a sequence of basic changes to be a complex change. The definition is as follows:

**Definition 3.12.** A complex change from  $T_0$  to  $T_n$  is of the form

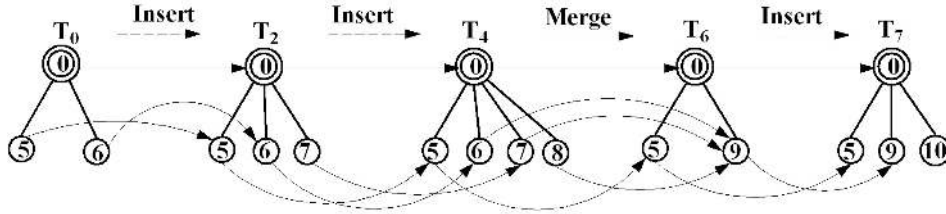
$$T_0 \xrightarrow{\gamma_0} T_1 \xrightarrow{\gamma_1} T_2 \xrightarrow{\gamma_2} \dots \xrightarrow{\gamma_{i-1}} T_i \xrightarrow{\gamma_i} \dots \xrightarrow{\gamma_{n-2}} T_{n-1} \xrightarrow{\gamma_{n-1}} T_n$$

in which each  $T_i \xrightarrow{\gamma_i} T_{i+1}$  ( $0 \leq i \leq n-1$ ) represents a NTTC from  $T_i$  to  $T_{i+1}$ .

For example, consider the case shown in Figure 1.1 (on page 3). We define the selected locations in the ellipse area and the rectangle area to be areal objects  $R_1$  and  $R_2$  respectively. The evolution of  $R_1$  and  $R_2$  can be represented by complex changes  $C_1$  and  $C_2$  as shown in Figures 3.11(a) and 3.11(b), respectively. In complex change  $C_1$  we omit the *no change* events starting from  $T_2$ ,  $T_3$ ,  $T_4$ , and  $T_5$ . In complex change  $C_2$  we omit the *no change* events starting from  $T_0$ ,  $T_2$ , and  $T_4$ .



(a)



(b)

Figure 3.11 Complex changes of the areal objects in Figure 1.1  
 (a) topological changes as  $R_1$  evolves, and (b) topological changes as  $R_2$  evolves

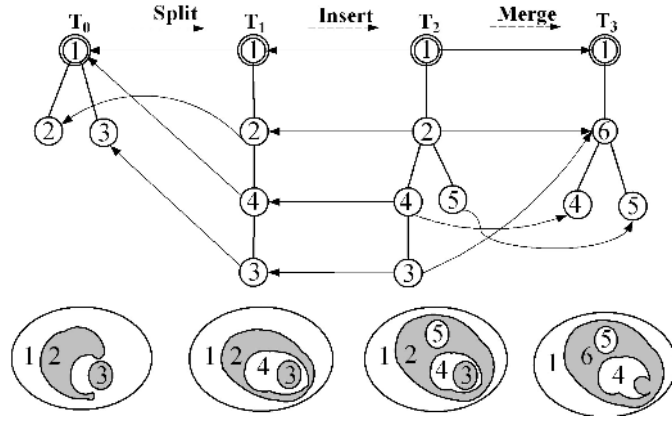
A basic issue is to define the notion of equivalence of complex changes. For example, Figure 3.12 shows two equivalent complex changes. Both changes result in a new component denoted by vertex 5. Vertices 1 and 4 in the final state originate from vertex 1 in the original state, and vertex 6 in the final state originates from vertices 2 and 3 in the original state.

The following definitions formalize our intuitions about equivalent complex changes in terms of trees and tree morphisms.

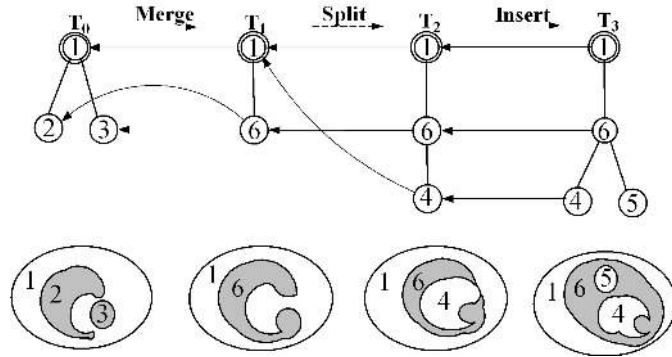
**Definition 3.13.** Let  $C$  be a basic change  $T_1 \xrightarrow{\gamma} T_2$  specified by a morphism  $\varphi$ .  $C$  induces a transform-to relation  $R \subset V(T_1) \times V(T_2)$  such that  $\forall v_1 \in V(T_1), v_2 \in V(T_2)$ ,  $(v_1, v_2) \in R$  if and only if either  $\varphi(v_1) = v_2$  or  $\varphi(v_2) = v_1$ .

**Definition 3.14.** Let  $C$  be a complex change from  $T_0$  to  $T_n$ :

$$T_0 \xrightarrow{\gamma_0} T_1 \xrightarrow{\gamma_1} T_2 \xrightarrow{\gamma_2} \dots T_i \xrightarrow{\gamma_i} \dots \xrightarrow{\gamma_{n-2}} T_{n-1} \xrightarrow{\gamma_{n-1}} T_n,$$



(a)



(b)

Figure 3.12 Two equivalent complex changes  
 (a) complex change A, and (b) complex change B

and let  $R_k (0 \leq k \leq n - 1)$  denote the transform-to relation induced by the basic change from  $T_k$  to  $T_{k+1}$ .

1. For any  $0 \leq i \leq n - 1$ , the future of  $v \in V(T_i)$  from stage  $i$  is defined to be  $F(v, i) = \{w \in V(T_n) | (v, w) \in R_i \circ R_{i+1} \circ \dots \circ R_{n-1}\}$ . The future of  $v \in V(T_n)$  from stage  $n$  is defined to be  $F(v, n) = \{v\}$ .
2. For any  $1 \leq i \leq n$ , the past of  $v \in V(T_i)$  from stage  $i$  is defined to be  $P(v, i) = \{w \in V(T_0) | (w, v) \in R_0 \circ R_1 \circ \dots \circ R_{i-1}\}$ . The past of  $v \in V(T_0)$  from stage  $0$  is defined to be  $P(v, 0) = \{v\}$ .



3. The set of essential insertions of  $C$  is defined to be

$$I(C) = \{(v, i) | i \in \{1, 2, \dots, n\} \wedge v \in V(T_i) \wedge F(v, i) \neq \emptyset \wedge P(v, i) = \emptyset\}.$$

In definition 3.14, the future of a vertex  $v$  is the set of vertices in the final state to which  $v$  transforms. The past of  $v$  is the set of vertices in the initial state which transform to  $v$ . An essential insertion  $(v, i)$  refers to a vertex  $v$  at stage  $i$  that is introduced by a basic insert, and which transforms to some vertices in the final state.

Two changes are defined to be equivalent if and only if both changes start from the same tree  $T_0$ , end at the same tree  $T_n$ , have the same set of essential insertions  $I$ , and have the same transform-to relation, characterized by the future functions, from the vertices of  $T_0$  and  $I$  to the vertices of  $T_n$ . The formal definition is given as follows:

**Definition 3.15.** Let  $C$  and  $C'$  be two changes from  $T_0$  to  $T_n$ , and from  $T'_0$  to  $T'_n$ , respectively. Let  $F$  and  $F'$  be the future functions of  $C$  and  $C'$ . Let  $I(C)$  and  $I(C')$  be the essential insertion sets of  $C$  and  $C'$ .

$C$  is defined to be equivalent to  $C'$  if and only if there is a tree isomorphism  $\varphi_0$  from  $V(T_0)$  to  $V(T'_0)$ , a tree isomorphism  $\varphi_n$  from  $V(T_n)$  to  $V(T'_n)$ , and a bijective function  $f$  from  $I(C)$  to  $I(C')$  such that,

1.  $\forall v \in V(T_0), \varphi_n(F(v, 0)) = F'(\varphi_0(v), 0)$ , and
2.  $\forall (v, i) \in I(C), \varphi_n(F(v, i)) = F'(f(v, i))$ .

### 3.6 The normal form for representing complex changes

Complex change can be composed of a sequence of NTTCs. Similar to graph generalization [SW99], it can be useful to represent a complex change by an equivalent change in a unified form. For example, the complex change shown in Figure 3.11(b) can be sim-

plified by combining the first two basic inserts into a single basic insert. The resulting equivalent change is shown in Figure 3.13. Can we further simplify the representation of the resulting change? In this section, we provide a normal form and prove that any complex change can be expressed in this form.

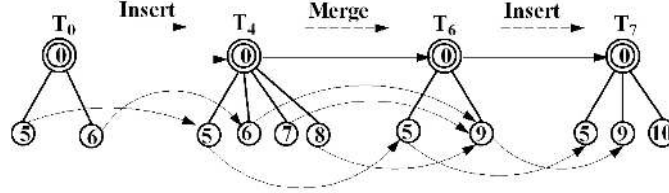


Figure 3.13 A simplification of complex change in Figure 3.11(b)

**Definition 3.16.** Let  $C$  be a complex change from  $T_0$  to  $T_n$ :

$$T_0 \xrightarrow{\gamma_0} T_1 \xrightarrow{\gamma_1} T_2 \xrightarrow{\gamma_2} \dots T_i \xrightarrow{\gamma_i} \dots \xrightarrow{\gamma_{n-2}} T_{n-1} \xrightarrow{\gamma_{n-1}} T_n.$$

The signature of  $C$  is defined to be a string  $s_0s_1\dots s_{n-1}$ , in which for  $\forall i(0 \leq i \leq n-1)$ ,  $s_i$  is a letter in the set  $\{I, S, M, D\}$ , such that

1.  $s_i = I$ , if  $T_i \xrightarrow{\gamma_i} T_{i+1}$  is a basic change of type insert.
2.  $s_i = S$ , if  $T_i \xrightarrow{\gamma_i} T_{i+1}$  is a basic change of type split.
3.  $s_i = M$ , if  $T_i \xrightarrow{\gamma_i} T_{i+1}$  is a basic change of type merge.
4.  $s_i = D$ , if  $T_i \xrightarrow{\gamma_i} T_{i+1}$  is a basic change of type delete.

**Definition 3.17.** A form is defined to be a string  $F = s_0s_1\dots s_n$  such that for  $\forall i(0 \leq i \leq n)$ ,  $s_i$  is a letter in the set  $\{I, S, M, D\}$ . A change  $C$  is defined to be represented by the form  $F$  if  $C$  is equivalent to a change  $C'$  whose signature is  $F$  or a subsequence of  $F$ .

In this section, we prove that any complex change is represented by the form of  $ISMD$ . We begin by presenting some technical results on trees and morphisms used to

prove our main results. The proofs of Lemmas 3.1, 3.2, and 3.3 are shown as lemmas 1, 2 and 3 in appendix of this thesis. Note: we use  $\varphi$  to represent a general function,  $\iota$  to represent an injective function, and  $\sigma$  to represent a surjective function.

**Lemma 3.1.** *Let  $T_1$  and  $T_2$  be trees, and  $\varphi$  be a tree morphism from  $T_1$  to  $T_2$ . Then, it is possible to find another tree  $T'$ , an injective tree morphism  $\iota$  from  $T_1$  to  $T'$ , and a surjective tree morphism  $\sigma$  from  $T'$  to  $T_2$ , satisfying:*

- (1) *Given any vertex  $v$  of  $T_1$ ,  $(\iota \circ \sigma)(v) = \varphi(v)$ .*
- (2) *Let  $S_1 = V(T') \setminus \text{img}(\iota)$  and  $S_2 = V(T_2) \setminus \text{img}(\varphi)$ . Then  $\sigma$  defines a bijection between  $S_1$  and  $S_2$ , by restricting the domain of  $\sigma$  to  $S_1$ .*

$$\begin{array}{ccc}
 T_1 & \xrightarrow{\varphi} & T_2 \\
 & \searrow \iota & \nearrow \sigma \\
 & & T'
 \end{array}$$

**Lemma 3.2.** *Let  $T_1$ ,  $T_2$  and  $T_3$  be trees,  $\iota$  be an injective tree morphism from  $T_2$  to  $T_1$ , and  $\varphi$  be a tree morphism from  $T_2$  to  $T_3$ . Then, it is possible to find a tree  $T'$ , an injective tree morphism  $\iota'$  from  $T_3$  to  $T'$ , and a tree morphism  $\varphi'$  from  $T_1$  to  $T'$ , satisfying:*

- (1) *Let  $v_1$  and  $v_3$  be vertices of  $T_1$  and  $T_3$  respectively.  $\varphi'(v_1) = \iota'(v_3)$  if and only if  $\exists v_2 \in V(T_2)$ , such that  $\iota(v_2) = v_1$  and  $\varphi(v_2) = v_3$ .*
- (2)  *$\varphi'$  is surjective whenever  $\varphi$  is surjective, and  $\varphi'$  is injective whenever  $\varphi$  is injective.*
- (3) *Let  $S_1 = V(T_1) \setminus \text{img}(\iota)$  and  $S_2 = V(T') \setminus \text{img}(\iota')$ . Then  $\varphi'$  defines a bijection between  $S_1$  and  $S_2$ , by restricting the domain of  $\varphi'$  to  $S_1$ .*
- (4) *Let  $S_3 = V(T_3) \setminus \text{img}(\varphi)$  and  $S_4 = V(T') \setminus \text{img}(\varphi')$ . Then  $\iota'$  defines a bijection between  $S_3$  and  $S_4$ , by restricting the domain of  $\iota'$  to  $S_3$ .*

$$\begin{array}{ccc}
T_1 & \xleftarrow{\iota} & T_2 \\
\varphi' \downarrow & & \downarrow \varphi \\
T' & \xleftarrow{\iota'} & T_3
\end{array}$$

**Lemma 3.3.** *Let  $T_1$ ,  $T_2$  and  $T_3$  be trees,  $\sigma_1$  be a surjective morphism from  $T_1$  to  $T_2$ , and  $\sigma_2$  be a surjective morphism from  $T_3$  to  $T_2$ . Then, it is possible to find two trees  $T'_4$  and  $T'_5$ , a surjective morphism  $\sigma'_1$  from  $T'_4$  to  $T_1$ , a surjective morphism  $\sigma'_2$  from  $T'_4$  to  $T'_5$ , and an injective morphism  $\iota'$  from  $T_3$  to  $T'_5$ , satisfying:*

*Given any vertex  $v_1$  of  $T_1$  and any vertex  $v_3$  of  $T_3$ ,  $\sigma_1(v_1) = \sigma_2(v_3)$  if and only if  $\exists v_4 \in V(T'_4), v_5 \in V(T'_5)$ , such that  $\sigma'_1(v_4) = v_1$ ,  $\sigma'_2(v_4) = v_5$ , and  $\iota'(v_3) = v_5$ .*

$$\begin{array}{ccccc}
T_1 & \xrightarrow{\sigma_1} & T_2 & \xleftarrow{\sigma_2} & T_3 \\
& & & & \downarrow \iota' \\
& & & & T'_5 \\
& \swarrow \sigma'_1 & & \searrow \sigma'_2 & \\
& & T'_4 & \xrightarrow{\sigma'_2} & T'_5
\end{array}$$

We next discuss some special complex changes, after which the discussion will be extended to arbitrary complex changes.

**Definition 3.18.** *An MD-change is defined to be a complex change with a signature MD. An SMD-change is defined to be a complex change with a signature SMD.*

**Theorem 3.1.** *Any complex change that is arbitrarily composed of basic merges and basic deletes can be represented by the form of MD.*

*Proof.* It is straightforward to prove this theorem if the complex change  $C$  is composed of only basic merges, or  $C$  is composed of only basic deletes. So, consider the case in which  $C$  is composed of both basic merges and basic deletes in any order. By composing adjacent basic merges into one basic merge and adjacent basic deletes into one basic delete, we are able to obtain a complex change  $C'$  which is equivalent to  $C$ , and is composed of a sequence of MD-changes.

Consider two adjacent MD-changes of the form:

$$\begin{array}{ccccccc}
 T_1 & \xrightarrow{\sigma_1} & T_2 & \xleftarrow{\iota_1} & T_3 & \xrightarrow{\sigma_2} & T_4 & \xleftarrow{\iota_2} & T_5 \\
 | & \cdots & \xrightarrow{\text{MD-change}} & \cdots & | & \cdots & \xrightarrow{\text{MD-change}} & \cdots & |
 \end{array}$$

in which,  $\sigma_1, \sigma_2$  are surjective morphisms specifying basic merges, and  $\iota_1$  and  $\iota_2$  are injective tree morphisms specifying basic deletes.

As shown in Figure 3.14, by Lemma 3.2 we are able to construct a tree  $T'_6$ , a surjective tree morphism  $\sigma'_3$  from  $T_2$  to  $T'_6$ , and an injective tree morphism  $\iota'_3$  from  $T_4$  to  $T'_6$ , satisfying the condition that the complex change from  $T_2$  to  $T_4$  specified by the sequence  $\sigma'_3$  and  $\iota'_3$  is equivalent to the complex change from  $T_2$  to  $T_4$  specified by the sequence  $\iota_1$  and  $\sigma_2$ .

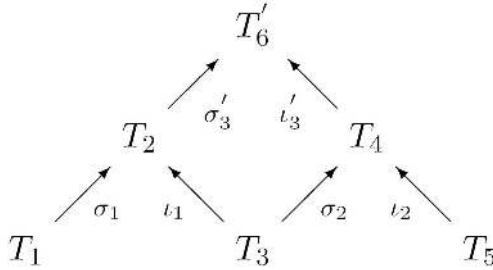


Figure 3.14 The simplification of two MD-changes

Hence, the composition of the two adjacent MD-changes is equivalent to one MD-change, which is composed of a basic merge from  $T_1$  to  $T'_6$  specified by  $\sigma_1 \circ \sigma'_3$ , and a basic delete from  $T'_6$  to  $T_5$  specified by  $\iota_2 \circ \iota'_3$ .

Repeating this procedure, we finally get one MD-change, which has the signature of  $MD$  and is equivalent to the complex change  $C$ . Therefore,  $C$  is represented by the form of  $MD$ .

□

**Theorem 3.2.** *Any complex change, which is arbitrarily composed of basic splits, basic merges, and basic deletes, is represented by the form of  $SMD$ .*

*Proof.* It is straightforward to prove this theorem if complex change  $C$  is composed of only basic splits, or is composed of basic merges and basic deletes. So, consider the case in which complex change  $C$  contains basic splits, as well as some basic merges and basic deletes in any order. By composing adjacent basic merges and basic deletes to form one MD-change and adjacent basic splits to form one basic split, we obtain a complex change  $C'$  which is equivalent to  $C$ , and is composed of a sequence of SMD-changes.

Consider two adjacent SMD-changes of the form:

$$\begin{array}{ccccccccccc}
 T_1 & \xleftarrow{\sigma_1} & T_2 & \xrightarrow{\sigma_2} & T_3 & \xleftarrow{\iota_1} & T_4 & \xleftarrow{\sigma_3} & T_5 & \xrightarrow{\sigma_4} & T_6 & \xleftarrow{\iota_2} & T_7 \\
 | & & \dots & \xrightarrow{\text{SMD-change}} & \dots & & \dots & \xrightarrow{\text{SMD-change}} & \dots & & \dots & & |
 \end{array}$$

in which,  $\sigma_1$  and  $\sigma_3$  are surjective morphisms specifying basic splits,  $\sigma_2$  and  $\sigma_4$  are surjective morphisms specifying basic merges, and  $\iota_1$  and  $\iota_2$  are injective morphisms specifying basic deletes.

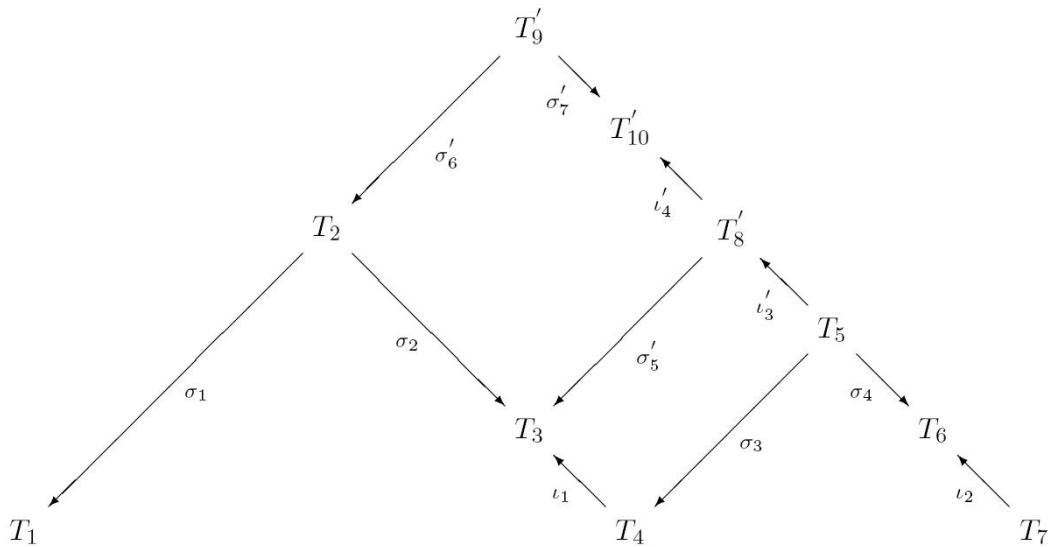


Figure 3.15 The simplification of two SMD-changes

The simplification of two SMD-changes is shown in Figure 3.15. Since there is a tree morphism  $\sigma_3 \circ \iota_1$  from  $T_5$  to  $T_3$ , using Lemma 3.1, we are able to find a tree  $T'_8$ , an injective tree morphism  $\iota'_3$  from  $T_5$  to  $T'_8$ , and a surjective morphism  $\sigma'_5$  from  $T'_8$  to  $T_3$ ,

satisfying the condition that the complex change from  $T_3$  to  $T_5$  specified by the sequence  $\iota_1, \sigma_3$  is equivalent to the complex change from  $T_3$  to  $T_5$  specified by the sequence  $\sigma'_5, \iota'_3$ . Since there is a surjective tree morphism from  $T_2$  to  $T_3$  and a surjective morphism from  $T'_8$  to  $T_3$ . Then, using Lemma 3.3 we are able to find trees  $T'_9$  and  $T'_{10}$ , together with tree morphisms  $\sigma'_6, \sigma'_7$  and  $\iota'_4$ . Here,  $\sigma'_6$  and  $\sigma'_7$  are surjective morphisms from  $T'_9$  to  $T_2$  and from  $T'_9$  to  $T'_{10}$  respectively.  $\iota'_4$  is an injective morphism from  $T'_8$  to  $T'_{10}$ . Therefore, the complex change from  $T_2$  to  $T'_8$  specified by the sequence  $\sigma'_6, \sigma'_7$  and  $\iota'_4$  is equivalent to the complex change from  $T_2$  to  $T'_8$  specified by the sequence  $\sigma_2$  and  $\iota'_5$ .

It follows that the composition of the two adjacent SMD-changes is equivalent to one SMD-change, which is composed of a basic split specified by  $\sigma'_6 \circ \sigma_1$ , and a MD-change composed of basic merges and basic deletes specified by the sequence  $\sigma'_7, \iota'_4, \iota'_3, \sigma_4$  and  $\iota_2$ , respectively.

Repeating this procedure, we finally get one SMD-change, which has the signature of  $SMD$  and is equivalent to the complex change  $C$ . Therefore,  $C$  is represented by the form  $SMD$ . □

We are now ready to prove the main result of this section.

**Theorem 3.3.** *Any complex change can be represented by the form of ISMD.*

*Proof.* It is straightforward to prove this theorem if complex change  $C$  is composed of only basic inserts, or is composed of basic changes of any type except basic inserts.

So, consider the case in which  $C$  is composed of basic inserts, as well as other types of basic changes in any order. By composing adjacent basic splits, basic merges and basic deletes together to make one SMD-change, and composing adjacent basic inserts together to one basic insert, we are able to obtain a complex change  $C'$ , which is composed of a set of changes in normal form.

Using Lemmas 3.1 and 3.2, we can prove that for any complex change composed of a SMD-change followed by a basic insert, we are able to find an equivalent complex change composed of a basic insert followed by a SMD-change change. Hence, any two adjacent complex changes in normal form can be composed together to form one complex change in normal form.

Repeating this procedure, we finally get one complex change that has the signature of  $ISMD$  and is equivalent to the complex change  $C$ . Therefore,  $C$  is represented by the form of  $ISMD$ .

□

As the form  $ISMD$  is able to represent all the complex changes, it is a *normal form*.

### 3.7 Properties of the normal form

In the previous section, we introduced a normal form  $ISMD$ , and proved that every change can be represented by the normal form. In this section we show that the  $ISMD$  is the simplest normal form. We first show that all four types of NTTCs are required as constituents in the form for representing every possible change. Therefore, the forms composed of less than four letters (such as  $I$ ,  $M$ , and  $IMD$ ) are not normal forms. Next, we show that no other form composed of four basic changes with a different sequence from  $ISMD$  can represent all the complex changes.

#### 3.7.1 Need for all types of NTTCs in the normal form

Let  $C$  be a complex change from  $T_1$  to  $T_2$ . We first note the following observations:

1. If  $C$  is composed of basic changes of any type excluding basic insert, for any vertex  $v$  of  $T_2$ , there is at least one vertex of  $T_1$  that transforms to  $v$  through  $C$ .



2. If  $C$  is composed of basic changes of any type excluding basic split, any vertex  $v$  of  $T_1$  transforms to at most one vertex of  $T_2$  through  $C$ .
3. If  $C$  is composed of basic changes of any type excluding basic merge, for any vertex  $v$  of  $T_2$ , there is at most one vertex of  $T_1$  that transforms to  $v$  through  $C$ .
4. If  $C$  is composed of basic changes of any type excluding basic delete, any vertex  $v$  in  $T_1$  transforms to at least one vertex of  $T_2$  through  $C$ .

With these observations, we show that any form that represent all the complex changes must contain at least one letter for each type of NTTCs. Consider the example shown in Figure 3.16. Let  $C$  be the complex change from  $T_0$  to  $T_4$ . No vertex of  $T_0$  transforms to vertex 5 of  $T_4$ . By observation 1, any complex change that is equivalent to  $C$  must contain a basic insert. Vertex 2 of  $T_0$  transforms to two vertices 6, 7 of  $T_4$ . By observation 2, any change that is equivalent to  $C$  must contain a basic split. Vertices 2, 3 of  $T_0$  transform to vertex 7 of  $T_4$ . By observation 3, any change that is equivalent to  $C$  must contain a basic merge. Vertex 4 of  $T_0$  does not transform to any vertex of  $T_4$ . By observation 4, any change that is equivalent to  $C$  must contain a basic delete. In all, the representation of the complex change  $C$  requires all four types of NTTCs. Thus, any form that does not allow all four types of NTTCs as constituents cannot represent the complex change  $C$ , and therefore cannot represent all possible complex changes.

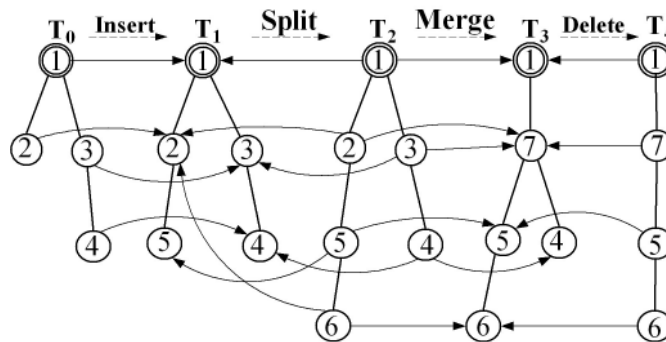


Figure 3.16 A complex change

### 3.7.2 Need for the sequence *ISMD* in the simplest normal form

The normal form must include all types of NTTCs, therefore the simplest normal form must be a composition of four letters, including *I*, *S*, *M*, and *D*. We now show that the four letters in a normal form must be structured in a particular order as *ISMD*. We prove this by showing examples that cannot be represented by forms that are different from *ISMD* (In the discussion, we use  $X_i$  to represent the component represented by the vertex  $i$ .)

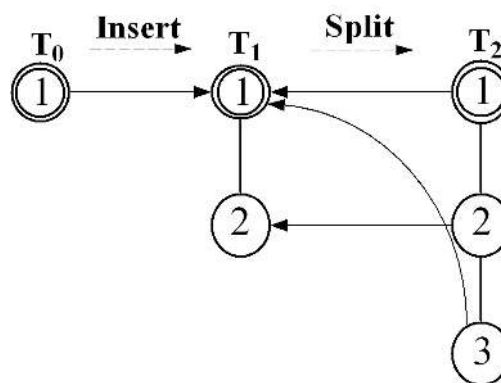


Figure 3.17 Counter example 1

Figure 3.17 shows a complex change  $C_1$  that is composed of a basic insert and a basic split.  $C_1$  starts from the state of a single negative component  $X_1$  (the whole spatial domain). During the change another negative component  $X_3$  is split from  $X_1$ . The split would never occur before a positive component exists to separate  $X_1$  and  $X_3$ . The positive component must be introduced by an insert. Thus,  $C_1$  can never be equivalent to a complex change, in which there is no basic split after a basic insert. Hence in order to represent all the changes, including  $C_1$ , a letter *S* must be after a letter *I* in the normal form.

Figure 3.18 shows a complex change  $C_2$  that is composed of a basic split and a basic merge. At the beginning of the change there are positive components  $X_2$  and  $X_3$ . During the change,  $X_3$  splits. Part of  $X_3$  merges with  $X_2$  and transforms to component  $X_5$ . The remaining part of  $X_3$  transforms to  $X_4$ .  $X_3$  can never merge with  $X_2$  before

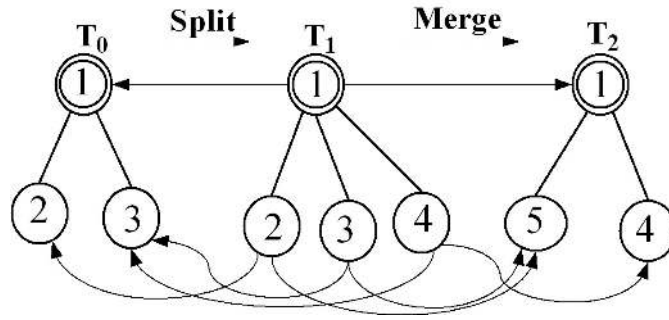


Figure 3.18 Counter example 2

$X_3$  splits, otherwise it is unable to get  $X_4$  at the end of the change. Thus, this change  $C_2$  could never be equivalent to a complex change, in which there is no basic merge after a basic split. Hence in order to represent all the complex changes, including  $C_2$ , a letter  $M$  must be after a letter  $S$  in the normal form.

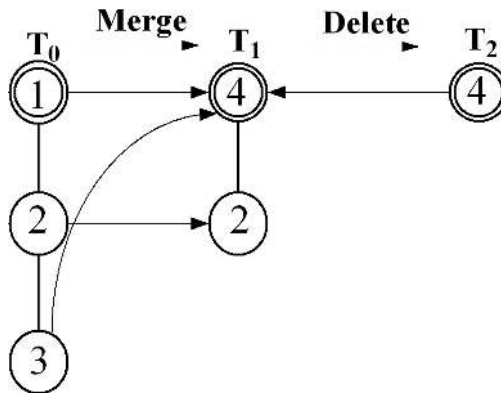


Figure 3.19 Counter example 3

Figure 3.19 shows a complex change  $C_3$  that is composed of a basic merge and a basic delete. In this change, in order to separate components  $X_3$  and  $X_1$ , component  $X_2$  can never be deleted before the  $X_3$  is merged with  $X_1$ . Thus,  $C_3$  can never be equivalent to a complex change, in which there is no basic delete after a basic merge. Hence in order to represent all the complex changes, including  $C_3$ , a letter  $D$  must be after a letter  $M$  in the normal form.

The normal form is able to represent all the topological changes including the three examples. Therefore any form that is composed of four types of NTTCs and is able to represent all complex changes must be *ISMD*.

### 3.8 Summary

In this chapter, we have specified basic and complex changes of areal objects as tree morphisms. We also provide a normal form that allows us to formally describe and compare spatial events according to changes in their topological structure.

## Chapter 4

# PRELIMINARIES FOR TOPOLOGICAL CHANGE DETECTION IN SENSOR NETWORKS

Wireless sensor network technology provides real-time information about the environment, which can play an important role in the monitoring of geographic phenomena. The application of sensor networks to the detection of topological changes is also an important consideration in this research. Chapters 4, 5 and 6 focus on this topic.

The model introduced in Chapter 3 specifies topological changes based on global topological structures of areal objects. In order to determine the types of topological changes using this model, it is necessary to have a finite sequence of snapshots describing the evolution of areal objects continuously both in time and in space. This is because we need to consider the identity of objects from one moment to the next. Sensor nodes take readings at discrete moments in time and discrete points in space, and cannot provide continuous sensing data. Therefore, it is difficult for sensor networks to determine the type of topological changes directly based on the model proposed in Chapter 3. Also, the theorem developed in Chapter 3 is not immediately suitable for distributed algorithms required by sensor networks.

In this chapter, we lay out the theoretical foundations of distributed approaches to topological change detection in sensor networks. A new model called the *local tree model* is developed. It represents and classifies topological changes based on local and

temporally discrete data. In addition, the completeness of the local tree model is proved. Also, the approximations of elements required by the local tree model, including components and their relations, are represented in sensor networks based on sensor readings and node connectivity. This allows us to capture the necessary information for topological change detection using sensor networks. Finally, possible inconsistencies between the basic elements in  $\mathfrak{R}^2$  and their approximations in sensor networks are discussed.

## 4.1 Local tree model

By definition 3.3, an areal object at a particular time  $t$  can be considered as a set  $R$  of points in the spatial domain  $\mathfrak{R}^2$ . An areal object evolves through time, and its evolution can be observed at a sequence of snapshots of the areal object. Each pair of consecutive snapshots describes a change, called a *basic transition*. Topological changes can be derived during a basic transition. In the following discussion, we present the elements to distinguish different types of topological changes incurred by a basic transition.

Let  $R_1$  and  $R_2$  be a pair of areal objects derived from the start and end snapshots, respectively, which define a basic transition. Any point  $p$  in  $\mathfrak{R}^2$  must have one of the following four states: (1)  $p \notin R_1$  and  $p \in R_2$ , (2)  $p \in R_1$  and  $p \notin R_2$ , (3)  $p \in R_1$  and  $p \in R_2$ , or (4)  $p \notin R_1$  and  $p \notin R_2$ . Based on the states of each point, the entire spatial domain can be partitioned into several components. Each component  $X$  is a subset of the spatial domain satisfying the following three conditions: (1)  $X$  is topologically connected. (2) The points in  $X$  have the same state. (3)  $X$  is maximal; that is, for  $\forall p \in \mathfrak{R}^2 \setminus X$ , either  $X \cup \{p\}$  is not connected, or  $p$  is in a different state from the points in  $X$ .

As an example, Figure 4.1(a) shows a basic transition of an areal object, and the spatial domain  $\mathfrak{R}^2$  is indicated by the dashed-line box. Based on the transition, the spatial domain is partitioned into components  $a-g$  (shown in Figure 4.1(b)), among

which, component  $b$  consists of points in state 2. Components  $a$ ,  $d$ , and  $f$  consist of points in state 3. Components  $c$ ,  $e$ , and  $g$  consist of points in state 4. To specify topological changes, we are most interested in the states of points in these components, as well as the adjacency and the surrounded-by relations between the components. The definition of the adjacency and surrounded by relations between the components are the same as definition 3.5.

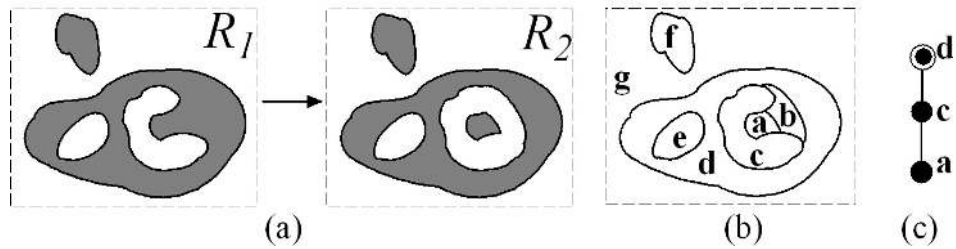


Figure 4.1 A basic transition

A component that consists of points in state 1 or state 2 is a piece that is either added to or removed from the areal object during the basic transition. We call such component a *transition region*. In the example shown in Figure 4.1, component  $b$  is a transition region. We assume each basic transition has only one transition region that is topologically equivalent to a disk; that is, the transition region is a single piece without any holes. Both assumptions simplify our discussion on basic transitions. More complex transitions, in which more than one transition region exists and each transition region is allowed to have holes, are not considered directly in this thesis. It should be pointed out that any complex transition can be decomposed into a sequence of basic transitions by first partitioning its transition regions into several simple transition regions, and then forcing the simple transition regions to switch one after another.

The components that consist of points in state 3 or state 4 do not change during the basic transition. The structure of such components is important to determine the type of the basic transition. However, not all of the components are necessary in the determination of the type of topological change. Only the components that are adjacent

to the transition region are key to determining the type. These components are referred to as *C-components*. In the example shown in Figure 4.1, only the components  $a$ ,  $c$  and  $d$  are C-components, as they are adjacent to the transition region  $b$ .

The structure of the C-components in a basic transition have the properties stated as follows, and Section 4.2 provides detailed proofs of these properties.

1. There is exactly one C-component  $X$  which surrounds all the other C-components.  $X$  is referred to as the *background C-component* of the basic transition.
2. The topological structure of the C-components in a basic transition can be represented by a rooted tree. A *vertex* of the tree represents a C-component, and an *edge* of the tree connects a pair of vertices representing adjacent C-components. *The root* of the tree represents the background C-component.

Among the three C-components  $a$ ,  $c$ , and  $d$  in the example shown in Figure 4.1,  $d$  is the background C-component as it surrounds both  $a$  and  $c$ . In addition, C-components  $a$  and  $c$  are adjacent, as well as the C-components  $c$  and  $d$ . The structure of the C-components can be represented by the rooted tree in Figure 4.1(c), in which the root is indicated by a double-circled vertex.

As different rooted trees can be explored in a systematic way, we are able to generate the possible topological structures between the C-components of a basic transition. Figure 4.2 lists all the rooted trees with less than 4 vertices, and examples of structures represented by the rooted trees are also provided. In the figure, the transition region is indicated by shaded area, and the vertices of the representation tree are placed inside the C-components they represent.

The classification of a basic transition is based on the following three factors: (1) the topological structure of its C-components, (2) the state of the points in its transition region, and (3) the state of the points in its background C-component.



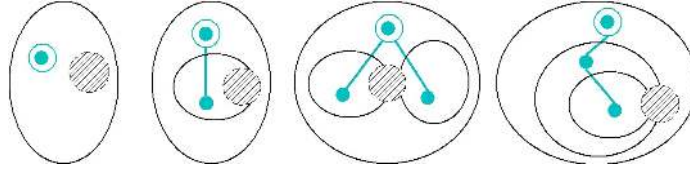


Figure 4.2 Tree representations for different configurations

The classification yields different types of topological changes incurred by a basic transition. Figure 4.3 shows the classification results, in which the 13 types of topological changes are distinguished, and Figure 4.4 provides an example of a basic transition for each type of topological change. These types of topological changes will be used in the sensor report to describe the observed basic transition.

Local topological structure	State of transition region (T)		State of background C-component (B)	
	T = 1 B = 4	T = 2 B = 4	T = 1 B = 3	T = 2 B = 3
	Region appear (A)	Region disappear (B)	Hole disappear (I)	Hole appear (J)
	Topology-preserving changes (C, D, K, L)			
	Regions merge (E)	Region split (F)	Hole split (M)	Hole merge (N)
	Region self-merge (G)	Region self-split (H)	Hole self-split (O)	Hole self-merge (P)

Figure 4.3 Classification of basic transitions

## 4.2 Completeness of local tree model

In addition to the specific types of topological changes listed in Figure 4.3, there are other basic transitions whose representations of C-components require trees with four or more vertices. Figure 4.5 gives an example of such a change and its representation tree. Such changes may not have a commonly accepted name associated with them, but they can also be predicted and represented by the local tree model. In this section, we prove that the structure of C-components in any basic transition can be represented by

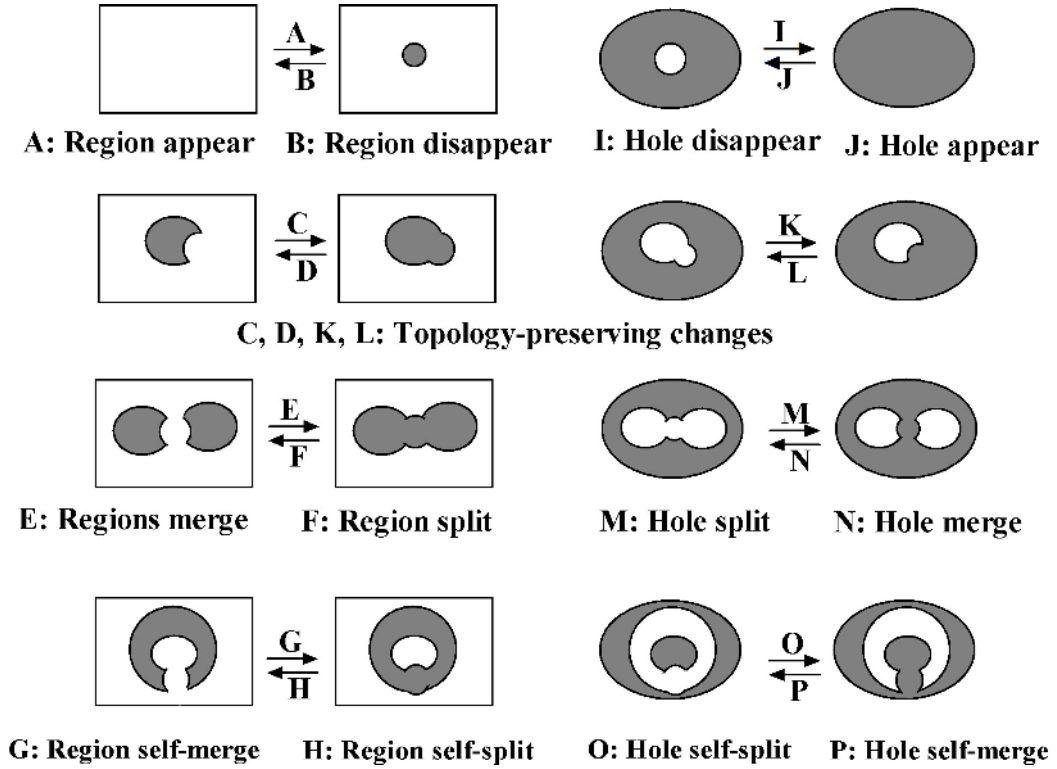


Figure 4.4 Examples of different types of topological changes

a rooted tree. Therefore, any basic transition is mapped onto a type specified by the local tree model, and the local tree model provide a complete coverage to all the basic transitions.

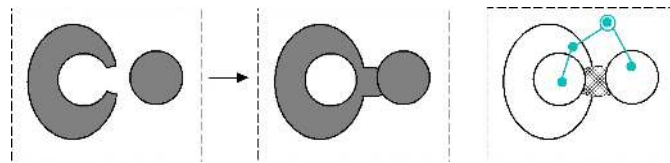


Figure 4.5 An example of uncommon topological change

We first present several lemmas that show the properties of the C-components in a basic transition.

**Lemma 4.1.** *The adjacency graph of the C-components in a basic transition is connected.*

*Proof.* We prove this lemma by contradiction. Assuming that the adjacency graph is not connected, let  $X_1, X_2, \dots, X_k$  be the C-components that are represented by the vertices of a maximal connected component in the adjacency graph. Let  $U = cl(\bigcup_{i=1}^k X_i)$ , where

$cl(X)$  represents the closure of  $X$ . By the definition of adjacency relations and the definition of C-components,  $U$  is a connected subset of  $\mathfrak{R}^2$  that intersects the boundary of the transition region.

We show that  $U$  is adjacent to a C-component that is not contained in  $U$ . As the transition region  $T$  is assumed to be topologically equivalent to a disk, its boundary is a Jordan curve  $C$ ; that is, there is a continuous mapping  $f : [0, 1] \rightarrow C$ , such that  $\forall x, y \in [0, 1), f(x) \neq f(y)$ , and  $f(0) = f(1)$ . As  $U$  intersects the boundary of the transition region  $T$ , we are able to find  $x_0 \in [0, 1]$  such that  $f(x_0)$  belongs to  $U$ . As  $U$  does not contain all the C-components, we are able to find  $y_0 \in [0, 1]$  such that  $f(y_0)$  does not belong to  $U$ . Consider the sequence of pairs:  $(x_0, y_0), (x_1, y_1), (x_2, y_2), \dots, (x_i, y_i), \dots$ , in which for  $\forall i > 0, (x_i, y_i) = ((x_{i-1} + y_{i-1})/2, y_{i-1})$  if  $f((x_{i-1} + y_{i-1})/2)$  intersects  $U$ , and  $(x_i, y_i) = (x_{i-1}, (x_{i-1} + y_{i-1})/2)$  if otherwise.

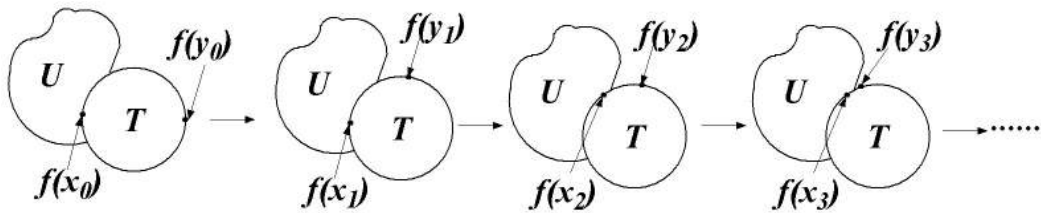


Figure 4.6 A sequence of  $(f(x_i), f(y_i))$

As illustrated in Figure 4.6, it can be proved that the sequence  $\{(x_i, y_i)\}$  converges to  $(x, y)$  in which  $x = y$ . Consider the point of  $f(x)$ , any open set in  $\mathfrak{R}^2$  that contains  $f(x)$  must contain points in  $U$ , points in  $T$ , and points in a component  $Y$  that is not contained in  $U \cup T$ . Therefore,  $f(x)$  is the intersection of  $U, T$  and  $Y$ . It follows that  $Y$  is a C-component, and  $U$  is adjacent to a C-component that is not contained in  $U$ , which contradicts the assumption that the vertices representing C-components contained in  $U$  form a maximal connected component in the adjacency graph. Hence, the adjacency graph of the C-components in a basic transition must be connected.  $\square$

The following lemmas and theorems use the notion of “partially surrounded-by”. Let  $X_1$  and  $X_2$  be a pair of distinctive C-components in a basic transition, and let  $T$  be the transition region.  $X_1$  is defined to be *partially surrounded by*  $X_2$  (or  $X_2$  *partially surrounds*  $X_1$ ) if  $X_1$  is surrounded by  $T \cup X_2$ . Figure 4.7 shows an example in which C-component  $X_1$  is partially surrounded by C-component  $X_2$ .

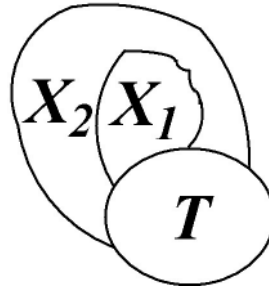


Figure 4.7 Example of partially-surrounded-by relation

**Lemma 4.2.** *Let  $X_1$  and  $X_2$  be a pair of adjacent C-components in a basic transition. One of the following statements must be true: either  $X_1$  partially surrounds  $X_2$ , or  $X_2$  partially surrounds  $X_1$ .*

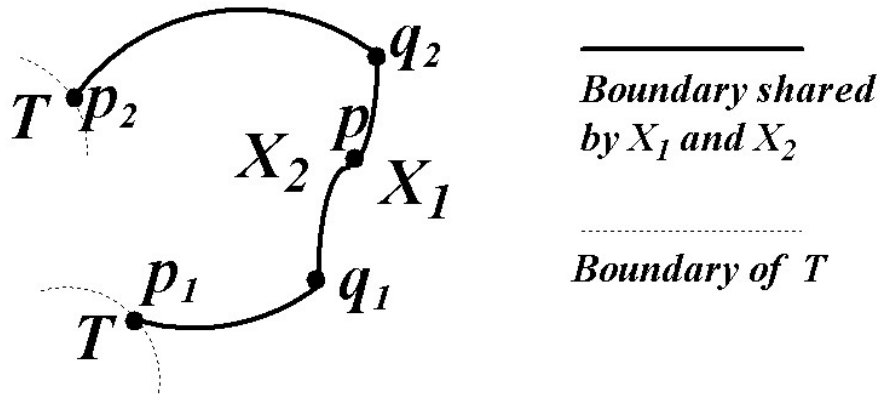


Figure 4.8 An illustration of Lemma 4.2

*Proof.*  $X_1$  and  $X_2$  are adjacent, so the intersection of their boundaries is not an empty set. Let  $p$  be a point in  $bd(X_1) \cap bd(X_2)$  that is located on a segment  $\overline{q_1 q_2}$  of the boundary formed by  $X_1$  and  $X_2$ .

Move along the common boundary shared by  $X_1$  and  $X_2$  in the direction from  $p$  to  $q_1$  until reaching a point  $p_1$ , where the boundary segment diverges, as shown in Figure 4.8. Because the areal objects are assumed to be strongly connected, there must be three sectors around  $p_1$ , among which one is contained in  $X_1$ , one is contained in  $X_2$ , and the third one is contained in the transition region. Therefore,  $p_1$  is a point located on the boundary of the transition region. Similarly, if we stroll along the boundary between  $X_1$  and  $X_2$  in the direction from  $p$  to  $q_2$ , we reach another point  $p_2$  located on the boundary of the transition region.

As the boundary of the transition region is a Jordan curve, it must contain line segment  $l$  which connects  $p_1$  and  $p_2$  and which intersects  $\overline{pp_1}$  and  $\overline{pp_2}$  only at the endpoints. Hence, the line segments  $\overline{pp_1}$ ,  $\overline{pp_2}$  and  $l$  form a Jordan curve  $C$ . As  $X_1$  and  $X_2$  are separated by  $C$ ,  $C$  must enclose exactly one of  $X_1$  and  $X_2$ . Suppose  $X_2$  is enclosed inside the Jordan curve  $C$ . By the Jordan Curve Theorem, any path that originates from a point in the closure of  $X_2$  to a point at infinity must contain some points in  $C$ , which is contained in the closure of  $X_1 \cup T$ . Therefore,  $X_2$  is partially surrounded by  $X_1$ .

Similarly,  $X_1$  must be partially surrounded by  $X_2$ , if  $X_1$  is enclosed inside the Jordan curve  $C$ . □

**Lemma 4.3.** *Let  $X_1$ ,  $X_2$ , and  $Y$  be distinct components such that  $X_1 \cup Y$  surrounds  $X_2$  and  $X_2 \cup Y$  surrounds  $X_1$ .  $Y$  must surround both  $X_1$  and  $X_2$ .*

*Proof.* We prove  $Y$  surrounds  $X_1$  by contradiction. Otherwise, we are able to find a path  $P$  such that  $P$  originates at a point  $p_0$  in  $X_1$  and goes to a point at infinity, and  $P$  does not intersect the closure of  $Y$ . Consider the travel from  $p_0$  to the point at infinity along  $P$ . As  $X_2 \cup Y$  surrounds  $X_1$  and  $P$  does not intersect the closure of  $Y$ , the travel along  $P$  must encounter a point  $p_1$  that is contained the closure of  $X_2$ . Similarly, as

$X_1 \cup Y$  surrounds  $X_2$  and  $P$  does not intersect the closure of  $Y$ , after passing by  $p_1$ , the travel must encounter another point  $p_2$  that is contained the closure of  $X_1$ . As shown in Figure 4.9, this continues recursively, and the travel enters the closure of  $X_1$  and  $X_2$  alternatively. As both  $X_1$  and  $X_2$  are bounded,  $P$  can never reach the point at infinity, contradicting to the assumption that  $P$  can reach that point. Therefore,  $Y$  surrounds  $X_1$ .

Similarly, we can prove that  $Y$  surrounds  $X_2$ .

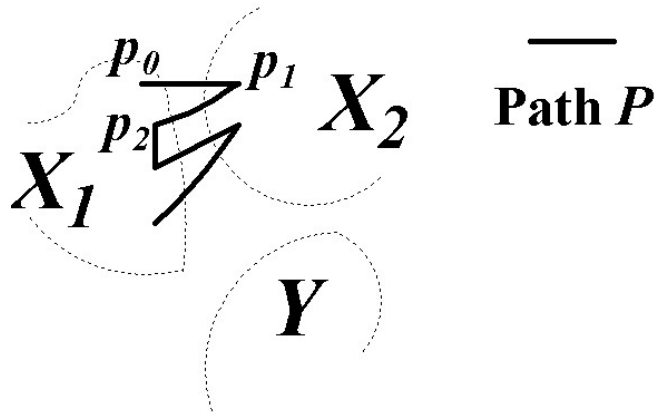


Figure 4.9 An illustration of Lemma 4.3

□

**Lemma 4.4.** *Let  $X_1$  and  $X_2$  be a pair of  $C$ -components. The following statements cannot both be true:*

1.  $X_1$  is partially surrounded by  $X_2$ .
2.  $X_2$  is partially surrounded by  $X_1$ .

*Proof.* Let  $T$  be the transition region. If both statements are true, we have  $X_1 \cup T$  surrounds  $X_2$  and  $X_2 \cup T$  surrounds  $X_1$ . By Lemma 4.3,  $T$  must surround both  $X_1$  and  $X_2$ . However,  $T$  is assumed to be simply connected, and therefore cannot surround any other components. Hence, the two statements cannot both be true. □

**Lemma 4.5.** *Let  $X_1$ ,  $X_2$  and  $Y$  be distinct  $C$ -components. The following statements cannot both be true:*

1.  $X_1$  is adjacent to  $Y$  and partially surrounds  $Y$ .
2.  $X_2$  is adjacent to  $Y$  and partially surrounds  $Y$ .

*Proof.* We prove this lemma by contradiction. Suppose both statements are true. We show that  $X_1$  partially surrounds  $X_2$ , and vice versa.

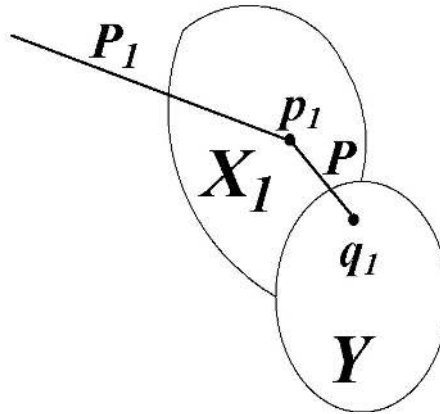


Figure 4.10 An illustration of Lemma 4.5

As  $X_1$  is adjacent to  $Y$ , for any point  $p_1$  in the closure of  $X_1$ ,  $p_1$  can be connected with a point  $q_1$  in  $Y$  by a path  $P$  that is completely contained in the interior of  $X_1$ , the interior of  $Y$ , and the boundary between them. As shown in Figure 4.10, for any path  $P_1$  that originates at  $p_1$  and goes to a point at infinity, the union of  $P$  and  $P_1$  forms a path  $P'$  which originates at  $q_1$  and goes to a point at infinity. Let  $T$  be the transition region. As  $Y$  is partially surrounded by  $X_2$ ,  $P'$  contains points in the closure of  $X_2 \cup T$ . In addition, as no points in  $P$  are contained in the closure of  $X_2 \cup T$ , there must be some points in  $P_1$  that are contained in the closure of  $X_2 \cup T$ . It follows that  $X_2$  partially surrounds  $X_1$ . Similarly, it can be proved that  $X_1$  partially surrounds  $X_2$ . This contradicts the conclusion of Lemma 4.4. Therefore, the two statements cannot both be true.

□

The following theorems present the major results of this section. Theorem 4.1 proves that the C-components in a basic transition can always be represented by a tree, and theorem 4.2 proves that a unique background C-component exists in any basic transition and can be represented by the root of the tree.

**Theorem 4.1.** *The adjacency graph of the C-components in a basic transition is a tree.*

*Proof.* This can be proved by showing that the adjacency graph of the C-components in a basic transition is both connected and cycle-free.

Lemma 4.1 proves that the adjacency graph is connected. We prove it is cycle-free by contradiction. Suppose we are able to find a cycle in the adjacency graph, which must contain more than 2 vertices. By Lemma 4.2, there must be three consecutive vertices  $v_{i-1}$ ,  $v_i$ , and  $v_{i+1}$  in the cycle representing C-components  $X_{i-1}$ ,  $X_i$ , and  $X_{i+1}$ , respectively, such that both  $X_{i-1}$  and  $X_{i+1}$  are adjacent to  $X_i$  and surround  $X_i$ . This contradicts the conclusion of Lemma 4.5.

In all, the adjacency graph of the C-components is both connected and cycle-free, and therefore must be a tree. □

**Theorem 4.2.** *Among all the C-components of a basic transition, there must be exactly one C-component that surrounds all the other C-components.*

*Proof.* It is straightforward to prove this theorem, if there is one or two C-components in the basic transition. Consider the basic transitions which have more than two C-components.

Let  $X_1, X_2, \dots, X_k (k > 2)$  be the C-components in a basic transition. We show that there must be exactly one C-component  $B$  that partially surrounds all the other C-components.

First, we show that there must be at least one C-component  $B$  that partially surrounds all the other C-components. If not, we are able to find two C-components  $X_n$



and  $X_m$  such that neither of them is partially surrounded by any other C-components. By theorem 4.1, the adjacency graph of these C-components is a tree. Let  $v_n$  and  $v_m$  be the vertices representing C-components  $X_n$  and  $X_m$  in the adjacency tree. Let  $[v_0 = v_n, v_1, v_2, \dots, v_l = v_m]$  be a path in the tree that connects  $v_n$  and  $v_m$ . As neither  $v_n$  nor  $v_m$  is surrounded by any other C-components, by Lemma 4.2 there must be three consecutive vertices  $v_{i-1}$ ,  $v_i$ , and  $v_{i+1}$  of the path that represent C-components  $X_{i-1}$ ,  $X_i$ , and  $X_{i+1}$ , such that both  $X_{i-1}$  and  $X_{i+1}$  are adjacent to  $X_i$  and partially surround  $X_i$ . This contradicts the result of Lemma 4.5. Therefore, there must be at least one C-component  $B$  that partially surrounds all the other C-components.

In addition, by Lemma 4.4, there is at most one C-component that partially surrounds all the other C-components. In all, there is exactly one C-component  $B$  that partially surrounds all the other C-components.

Finally, we show that  $B$  surrounds all the other C-components. Let  $T$  be the transition region, and  $U$  be the union of all the C-components except for  $B$ . As  $B$  partially surrounds all the other C-components, it follows that  $B \cup T$  surrounds  $U$ . In addition, by the definition of C-components, the union of all the C-components surrounds  $T$ ; that is,  $B \cup U$  surrounds  $T$ . By Lemma 4.3, we have  $B$  surrounds  $U$ , and therefore  $B$  surrounds all the other C-components.

□

Theorems 4.1 and 4.2 prove that the structure of C-components in any basic transition can be represented by a rooted tree, in which the root represents the unique background C-component. Therefore the rooted tree model is complete for representing basic transitions.

### 4.3 Sensor network configuration

We are using sensor networks to track and report topological changes. This section provides the basic assumptions we have made about the sensor networks in this thesis, as well as the definitions of basic elements based on sensor network configuration, which capture the properties of a basic transition required by the local tree model.

We assume that a large number of sensor nodes are deployed in the sensing area. Each sensor node is initialized with a unique identifier and records the values of measurements. Each sensor is able to communicate with the nodes nearby, and a pair of sensors that is able to communicate directly is define to be *direct neighbors*. A node located near the boundary of the sensing area is selected to be the *reference node*, which is assumed to be located outside the scope of the observing phenomena. The sensor nodes in the sensing area induce a Voronoi diagram, and each sensor node  $n$  is associated with a Voronoi cell consisting of all the points that are closer to  $n$  than to any other sensor node. We stipulate that the sensor node deployment satisfies the following constraints:

1. **Density constraint**, sensor nodes are deployed densely enough so that a sensor node measurement reflects, with sufficient accuracy, all points in its Voronoi cell.
2. **Communication constraint**, each sensor node communicates exactly with the nodes in its adjacent Voronoi cells.

Figure 4.11 shows an example deployment of sensor nodes and their associated Voronoi cells. It also shows a possible location of the reference node  $r$ .

The sensor nodes take measurements at a sequence of sampling rounds  $t_0, t_1, \dots, t_n$ . We assume that the reading of a sensor node at any of the sampling rounds is either 0 or 1. Our interpretation is that the reading is 1 if the sensor node is in an area of high intensity (reading above a threshold), otherwise it is 0. Take the monitoring of a

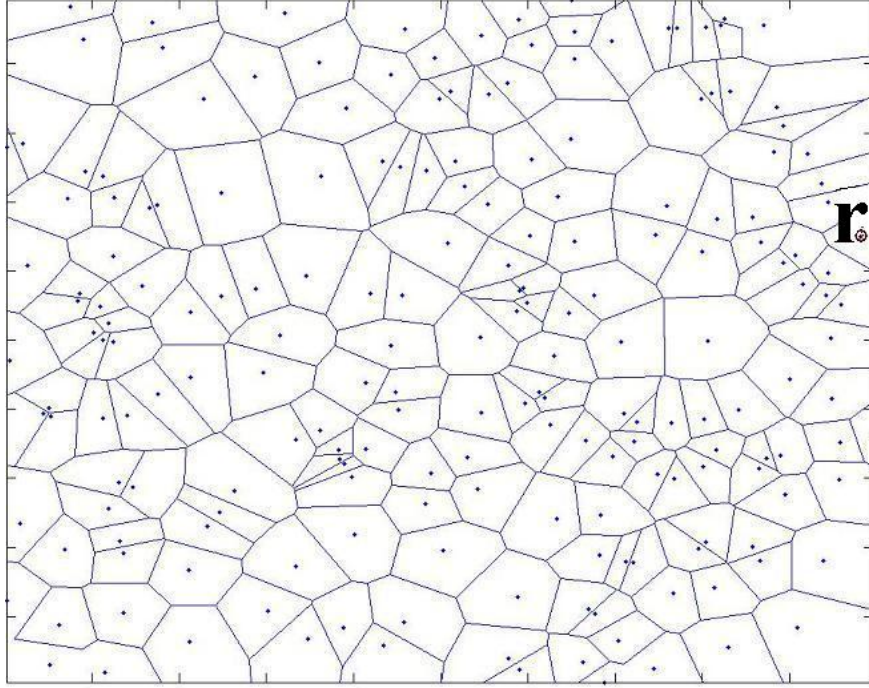


Figure 4.11 Sensor network configuration

wildfire as an example, sensors with temperature reading being 1 indicate the location of the fire. The 0/1 approach enables the derivation of regions of interest directly from the sensing reports, and at the same time the 0/1 approach reduces energy consumption in communication due to the small and discrete domain of values [DNW05].

A change is captured by sensor readings at a pair of consecutive sampling rounds, and the type of a transition is determined by comparing the readings. The comparison first defines four states of nodes at sampling round  $t_i$ .

**Definition 4.1.** Let  $r(n, t) \in \{0, 1\}$  denote the reading of a node  $n$  at a time  $t$ . The state of  $n$  at a sampling round  $t_i (1 \leq i \leq k)$  is defined to be a pair  $h = (r(n, t_{i-1}), r(n, t_i))$ , such that  $h \in \{(0, 1), (0, 0), (1, 0), (1, 1)\}$ .

The state of a node varies with time. In the following discussion, everything is assumed to be in the same snapshot at sampling round  $t_i (1 \leq i \leq n)$  unless time is explicitly specified. The states of the sensor nodes together with the sensor connectivity yield the following concepts that are foundations for topological change detection.

**Definition 4.2.** Let  $N$  be a set of sensor nodes,  $N$  is said to be a homogeneous sensor network component if the nodes in  $N$  are in the same state and induce a connected component in the communication graph. Moreover,  $N$  is defined to be a maximal homogeneous sensor network component, if it is impossible to find a node  $n$  in the sensing area such that (1)  $n \notin N$ , and (2)  $N \cup \{n\}$  is a homogeneous sensor network component.

**Definition 4.3.** Let  $N_1$  and  $N_2$  ( $N_1 \cap N_2 = \emptyset$ ) be a pair of homogeneous sensor network components.

1.  $N_1$  is said to be adjacent to  $N_2$  if there are nodes  $n_1 \in N_1$  and  $n_2 \in N_2$  such that  $n_1$  and  $n_2$  are direct neighbors in the communication graph. Otherwise,  $N_1$  and  $N_2$  are said to be separated.
2.  $N_1$  is said to be surrounded by  $N_2$ , if any path in the communication graph that starts from the reference node and contains a node of  $N_1$  must contain a node of  $N_2$ .  $N_1$  is said to surround  $N_2$ , if  $N_2$  is surrounded by  $N_1$ .

**Definition 4.4.** Let  $N$  be a maximal homogeneous sensor network component.

1.  $N$  is defined to be a transition sensor network component, if  $N$  consists of only nodes either in state  $(0,1)$  or in state  $(1,0)$ .
2.  $N$  is defined to be a sensor network C-component, if both of the following conditions are satisfied: (1)  $N$  consists of nodes either in state  $(0,0)$  or in state  $(1,1)$ , and (2)  $N$  is adjacent to transition sensor network component.
3.  $N$  is defined to be a background sensor network C-component, if it is a sensor network C-component and it surrounds all the other sensor network C-components.

We use the elements defined in sensor networks to represent the basic features that are necessary for topological change detection. Table 4.1 shows the correspondences between the elements we defined in spatial domain  $\mathbb{R}^2$  and their representations in sensor

Table 4.1 Approximated components\relations in sensor networks

In spatial domain $\mathfrak{R}^2$	In sensor networks
A C-component in state 3	A sensor network C-component in state (1, 1)
A C-component in state 4	A sensor network C-component in state (0, 0)
The transition region	The transition sensor network component
The background C-component	The background sensor network C-component
Adjacency relations between C-components	Adjacency relations between sensor network C-components
Surrounded-by relations between C-components	Surrounded-by relations between sensor network C-components

networks. Based on the correspondences, all the concepts defined in the spatial domain  $\mathfrak{R}^2$  can be represented and computed in terms of states of sensor nodes and connectivity between them.

As an example, Figure 4.12(a) and 4.12(b) show the readings of nodes at consecutive sampling rounds  $t_1$  and  $t_2$ , when the basic transition shown in Figure 4.1 is observed. The black points denote nodes with reading 1, and the white points denote nodes with reading 0. The only difference between the two snapshots is that the nodes located in the area enclosed by a polygon change their readings. These nodes form a maximal homogeneous sensor network component in state (1,0), which is a transition component at  $t_2$  by definition 4.5. The remaining nodes are in states (0,0) and (1,1) at  $t_2$ , and they form the six other maximal homogeneous sensor network components. The seven maximal homogeneous sensor network components represent the components in the spatial domain  $\mathfrak{R}^2$  as shown in Figure 4.12(c). In the figure, component  $b$  is represented by the transition sensor network component. Components  $a$ ,  $d$ , and  $f$  are represented by maximal homogeneous sensor network components that consist of nodes in state (1,1). Components  $c$ ,  $e$ , and  $g$  are represented by the maximal homogeneous sensor network components that consist of nodes in state (0,0). In addition, the maximal homogeneous sensor network components that represent the components  $a$ ,  $c$ , and  $d$  are shown in Figures 4.12(d), 4.12(e), and 4.12(f). They are adjacent to the transition sensor network component, and therefore both are sensor C-components at  $t_2$ .

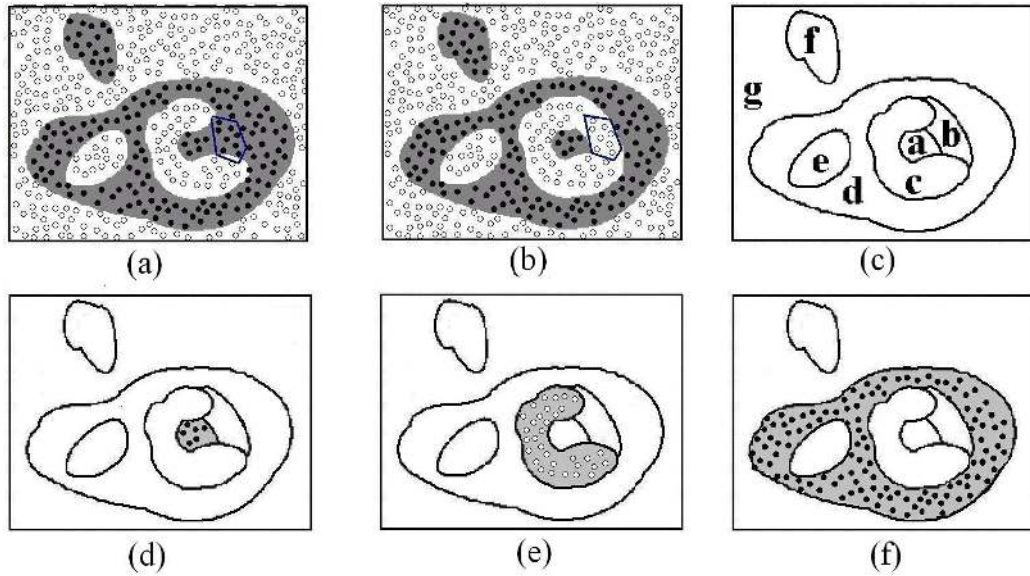


Figure 4.12 An example of homogeneous components

#### 4.4 Density and communication issues

Ideally, the elements defined in sensor networks represent the properties of the areal objects in the spatial domain  $\mathbb{R}^2$ . The nodes located in a component of the spatial domain form exactly one maximal homogeneous sensor network component. Components in the spatial domain are adjacent if and only if they are represented by adjacent maximal homogeneous sensor network components. Components in the spatial domain surround each other if and only if they are represented by maximal sensor network components that surround each other. These properties are expressed by the density and communication constraints in Section 4.3.

However, such a perfect matching may not always exist. Inconsistency may be caused by low node density and improper setting of communication ranges. Here are some of the examples.

First, if the density of the nodes is low, a component in the spatial domain that is small enough may not contain any sensor node, and therefore is not represented by any sensor network component, as shown in Figure 4.13(a).

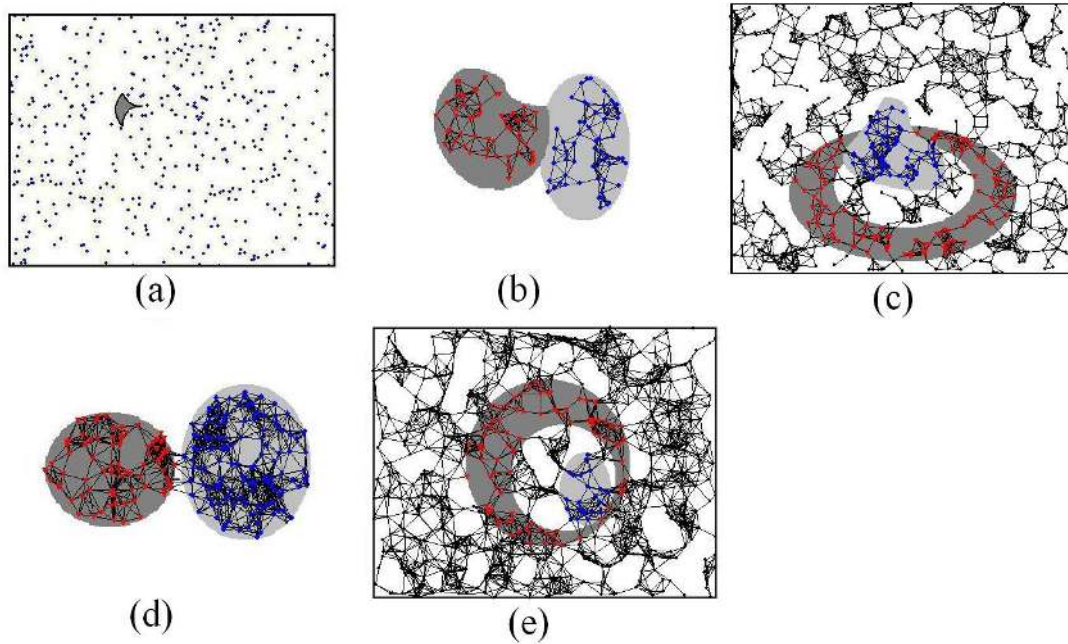


Figure 4.13 Configurations that lead to errors in reports

Second, if the communication range of the sensor nodes is not large enough, two types of inconsistencies may occur:

1. A pair of adjacent components in the spatial domain are represented by a pair of separated maximal homogeneous sensor network components, as shown in Figure 4.13(b).
2. A pair of components in the spatial domain that are not surrounded by each other are represented by a pair of adjacent maximal homogeneous sensor network components such that one is surrounded by the other, as shown in Figure 4.13(c).

Finally, if the communication range of the sensor nodes is not small enough, two types of inconsistencies may occur.

1. A pair of components that are not adjacent in the spatial domain is represented by a pair of adjacent maximal homogeneous sensor network components, as shown in Figure 4.13(d).

2. A pair of components in the spatial domain, such that one surrounded by the other, is represented by a pair of maximal homogeneous sensor network components which does not surround each other, as shown in Figure 4.13(e).

In order to avoid inconsistency, the density of the sensor nodes should be high enough to distinguish the smallest variation in the spatial domain, and the sensing range of the nodes should be set properly. This is why we stipulate that the sensor network satisfies the density and communication constraints.

## 4.5 Summary

This chapter provides the computational foundations for topological change detection in sensor networks. Based on the local tree model, basic transitions are classified into a complete set of classes, and each class specifies a type of topological change. We also analyze the corresponding elements required by the local tree model in the sensor network configuration, so that the necessary information required by the model can be fully captured using sensor networks.



## Chapter 5

# DISTRIBUTED APPROACHES FOR TOPOLOGICAL CHANGE DETECTION USING SENSOR NETWORKS

The discussion in Chapter 4 shows that to detect topological changes, we need to identify both the transition sensor network component and the sensor network C-components, together with their states, and more importantly, to determine the topological relations between the sensor network C-components. This chapter provides distributed algorithms for topological change detection using sensor networks based on the foundation laid in Chapter 4. After present some basic approaches in Section 5.1, two energy-efficient topological change detection approaches are proposed in Sections 5.2 and 5.3, respectively.

### 5.1 Basic approaches

Using a basic approach, we require the entire collection of nodes located in a sensing area to report their readings and geographic locations back to the base station after each sensing round. With all the received data, the base station is able to determine the types of topological changes using centralized computation. However, it is too energy-consuming to gather all of the raw sensor node information to the base station.

To improve on the basic approach, we postulate that only the nodes located near the boundary of the areal object report after each sensing round. A boundary node is defined as a node that has a direct neighbor with a different reading. As an example, Figures 5.1(a) and 5.1(b) show the boundary nodes that are required to report during a basic transition. Because the boundary nodes carry the necessary information for topological change detection, the base station is able to detect the topological changes based on the data received from the boundary nodes. The polygon in Figure 5.1(c) encloses the boundary nodes that are identified to be part of the transition sensor network component. Figures 5.1(d), 5.1(e), and 5.1(f) show the boundary nodes contained in different sensor network C-components. By analyzing the properties of the boundary nodes, the structure of the representation tree can also be identified, as shown in Figure 5.1(g).

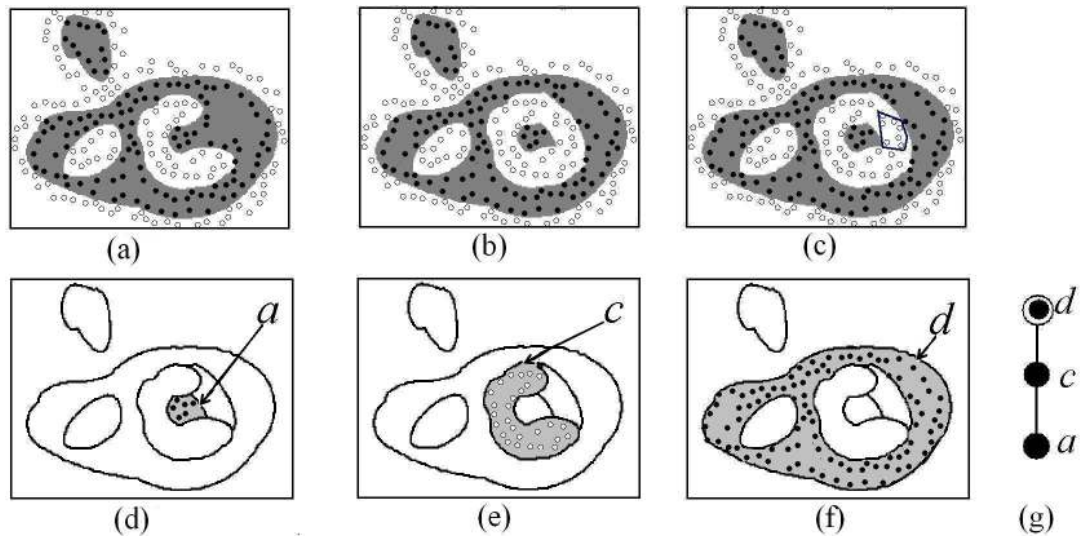


Figure 5.1 Boundary nodes

## 5.2 Decentralized topological change detection using transient groups

The basic boundary-based approach reduces the number of reporting nodes. However, as we have assumed a dense deployment of sensor nodes, the number of boundary nodes may still be large. One option for further reducing the communication cost in boundary reporting is to suppress the amount of data being transferred based on in-network compression or boundary simplification [ZW07, LL07]. However, as indicated in [GHS07], most of these approaches cannot guarantee the topology preservation in their compression results. Therefore, in some cases it is inaccurate to derive topological changes based on the approximate boundaries constructed by those compression approaches. In order to retrieve all the necessary information for topological change detection with low communication cost, we developed the approach of decentralized topological change detection using transient group approach (TG-based approach).

The first improvement of the TG-based approach over the basic boundary-based approach is that instead of requiring all boundary nodes to report, the nodes required to report in the TG-based approach only include a small portion of boundary nodes. These nodes are basically located close to both the boundary formed by the C-components and the boundary of the transition region, and are referred to as C-nodes and T-nodes, defined as follows.

1. A *C-node* is a node located in a sensor network C-component and has a direct neighbor located in a different sensor network C-component.
2. A *T-node* is a node located in the sensor transition component and has a direct neighbor located in a sensor network C-component.

As an example, Figure 5.2(b) shows the C-nodes and T-nodes that are involved in the reporting procedure after the basic transition described by Figures 5.2(a) and

5.2(b) is observed. A method of identifying the C-nodes and T-nodes is detailed in the diffusion phase of Section 5.2.2.

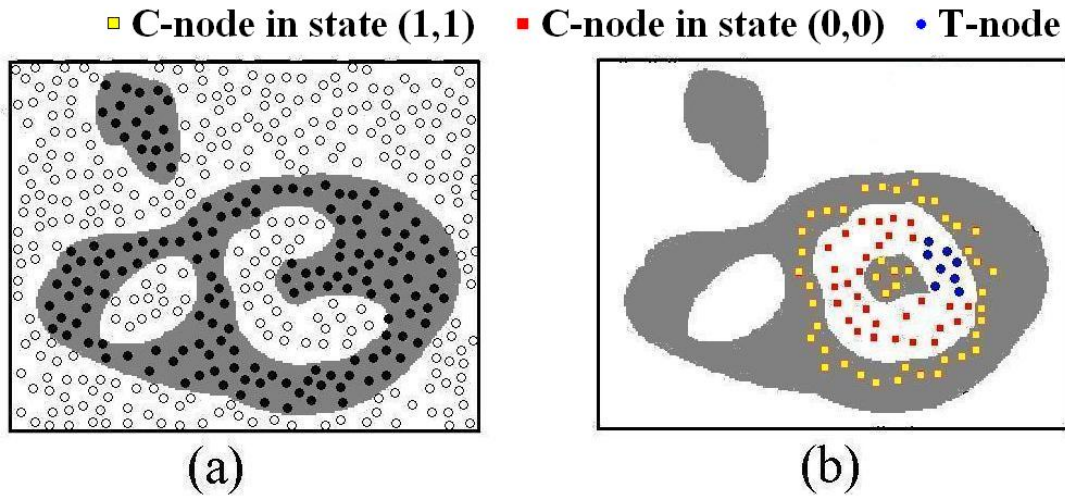


Figure 5.2 The C-nodes and T-nodes

The second improvement of the TG-based approach over the basic boundary-based approach is that instead of reporting the data directly to the base station, a *representative node* in the TG-based approach is selected among the T-nodes at each sensing round. This representative node is responsible for collecting the data sent from the C-nodes and forming the final report on the observed topological change. In this way, the computation can be performed locally by the C-nodes and T-nodes.

The third improvement of the TG-based approach over the basic boundary-based approach is that in the TG-based approach, the C-nodes form groups, and only a node, the group leader, in each group is responsible to report to the representative node. The data sent from the group leader represents information about the group as a whole, instead of information about each single node in the group. With the group-level data available, the representative node is able to find the necessary information required by the local tree model, and finally determine the type of topological change. As the amount of group-level data is much smaller than the node-level data, the communication cost in the TG-based approach is further reduced compared to the boundary-based approach.

Section 5.2.1 discusses the necessary group level data that are needed in order to determine the type of a topological change, and Section 5.2.2 presents the algorithm of the TG-based approach in detail.

### 5.2.1 Boundary group based representation

In this section, we show that the necessary features for topological change detection can be completely represented by properties of groups formed among the C-nodes; that is, groups of nodes along the boundary of C-components.

In the TG-based approach, C-nodes at each sensing round form groups. The nodes in a group are assigned a unique integer label, and the group is indicated by the label. The nodes in the same group form a homogeneous sensor network component; that is, they have the same state and form a connected component in the communication graph. A single node in each group is selected to be the group leader, which is responsible for sending data about the group back to the representative node.

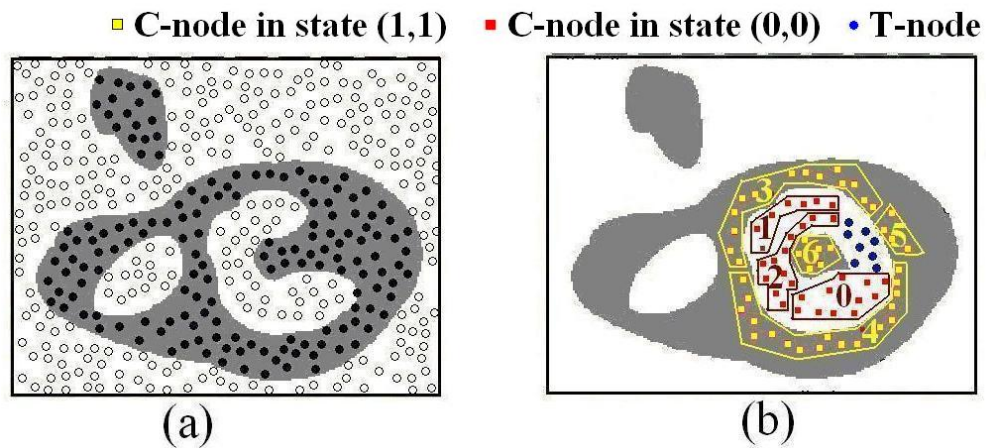


Figure 5.3 Groups among C-nodes

To determine the structure of the sensor network C-components in a basic transition, the group level data sent from the leader of a group  $G$  include the integer label of  $G$ , the common state of the nodes in  $G$ , and the *neighboring label set* of  $G$ . The neighboring label set of  $G$  is defined to be the set of labels of groups that are adjacent

to  $G$ . In order to reduce the redundant communication cost, we also require that the labels in the neighboring label set of  $G$  must be greater than the label of  $G$ . In this way, the adjacency relation between two groups is captured and reported by exactly one of the group leaders.

As an example, Figure 5.3(b) shows a possible formation of groups among the C-nodes identified after the basic transition described by Figures 5.3(a) and 5.3(b) is observed. Seven groups labeled from 0 to 6 exist. Taking group 0 as an example, the data sent from this group to the representative node include group label 0, the state  $(0,0)$  of the group, and the neighboring label set  $\{2, 4, 6\}$ . The data received by the base station from all the groups can be represented by the graph shown in Figure 5.4(a). In this graph, a vertex represents a group of that label, and the shading of a vertex represents the state of the group. Edges of the graph represent the adjacency relations between the groups, which are derived of the neighboring label sets of the groups.

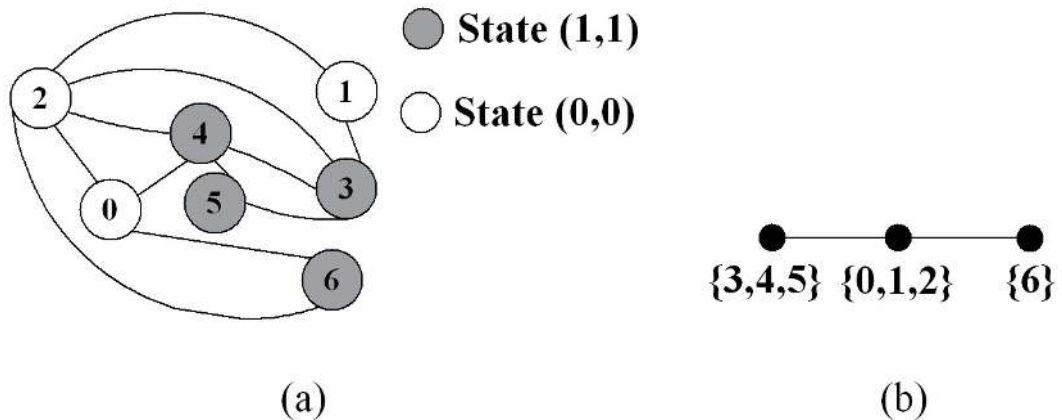


Figure 5.4 Information received at the representative node

The representative node is able to identify the C-components solely based on the group level data. In the example shown in Figure 5.4(a), as all the groups are adjacent to the transition sensor network component, they are contained in various sensor network C-components. First, consider group 0. As group 2 is adjacent to group 0 and both have the same state  $(0,0)$ , they are contained in the same sensor network C-component

by definition 4.3. Similarly, as group 1 is adjacent to group 2 and both have the same state, group 1 must be contained in the same sensor network C-component as group 2 and 0. Besides that, no other groups in state (0,0) are adjacent to groups 0, 1 and 2. So there must be a C-component in state (0,0) that contains groups 0, 1 and 2.

Next, consider groups 3, 4 and 5. As all of them are adjacent and have the same state (1,1), they are contained in the same C-component. In addition, as no other group in state (1,1) is adjacent to them, we identify a sensor network C-component in state (1,1), which contains the groups 3, 4 and 5. Finally, consider group 6: as no other group in state (1, 1) is adjacent to group 6, another sensor network C-component in state (1,1) is identified that contains group 6. In all, we have identified three sensor network C-components of this basic transition.

The adjacency relations between the sensor network C-components can be identified based on the adjacency relations between the groups. A pair of sensor network C-components are adjacent if one contains a group that is adjacent to a group in the other C-component. In this example, as group 0 is adjacent to group 4, the sensor C-component containing groups 0, 1 and 2 must be adjacent to the sensor network C-component containing groups 3, 4 and 5. Similarly, as group 0 is adjacent to group 6, the sensor network C-component containing groups 0, 1 and 2 must be adjacent to the sensor C-component containing group 6. Figure 5.4(b) shows the representation tree of the sensor network C-components and adjacency relations identified based on the group level data.

Finally, to identify the background sensor network C-component, we require each group to send its group hop distance to the base station, in addition to the other data. The *group hop distance* is the number of hops of the shortest path between any node in the group and the reference node. As the background sensor network C-component surrounds any of the other sensor network C-components and the reference node is

located outside all the sensor network C-components, it follows that any path that connects a node in a sensor network C-component to the reference node must cross a group contained in the background sensor network C-component. Therefore, among all the groups identified to be contained in sensor C-components, the one with the minimal group hop distance must belong to the background sensor network C-component.

In all, the information required to determine the type of a topological change can be represented by the group level data.

### 5.2.2 Algorithm

This section provides the detailed algorithm of the TG-based approach. An initialization is performed at the beginning of the sensing task. Next, the detection procedure is repeated after each sensing round, which consists of the following four phases:

1. **Diffusion phase**, in which the C-nodes and T-nodes are identified, and the representative node is selected among the T-nodes.
2. **Group formation phase**, in which the C-nodes form groups, and nodes in each group hold a unique label. At the same time a routing tree is built, which connects all the C-nodes to the representative node.
3. **Aggregation phase**, in which the group level data necessary for detecting the topological changes from the C-nodes are transmitted back to the representative node along the routing tree.
4. **Analysis phase**, in which the representative node analyzes the data it has received and forms the report to be sent back to the base station.

Although the four phases are presented and will be described in a sequential order, in implementation the nodes are not required to be globally synchronized. Each



node can start the next phase as long as it is confirmed to have finished the tasks in the previous phase. The following subsections describe each phase in detail.

### **Initialization**

During initialization, each sensor node computes its hop distance, and the base station sends out a query to all the nodes in the whole sensing area.

Similar to the approach described in [FK06], hop distances of nodes can be computed by flooding originating from the reference node. The reference node broadcasts a *HELLO* message maintaining a distance counter that is incremented at every hop. The minimum counter value over all messages received by a node  $n$  is the hop distance of  $n$ .

Query propagation is done by flooding originating from the base station, which broadcasts a query  $TPQ(t_s, t_f, t_\Delta)$  to all the nodes located in the sensing area. In the query,  $t_s$  and  $t_f$  state the time to start and to finish the monitoring, and  $t_\Delta$  specifies the time period between a pair of consecutive sensing rounds.

### **Diffusion phase**

In this phase, C-nodes and T-nodes are identified, and the representative node is selected. After each sensing round a node can determine its state directly based on its local readings. The identification of C-nodes and T-nodes needs the communication between the nodes. The nodes in state  $(0, 1)$  or state  $(1, 0)$  have different behavior from the nodes in state  $(1, 1)$  or state  $(0, 0)$ , and we describe them separately.

In this phase, the nodes in state  $(0, 1)$  or state  $(1, 0)$  communicate with their neighbors to identify the T-nodes. A representative node election procedure is performed among the identified T-nodes, after which, one of the T-nodes is elected to be the representative node. A by-product of the election is a routing tree, which connects all the T-nodes back to the representative node.

Assume a node  $n$  discovers itself to be in state  $(0, 1)$  or state  $(1, 0)$  after a sensing takes place. Node  $n$  first broadcasts a message *IS-CHANGED* to its neighbors and waits. If  $n$  discovers one of its neighbors to be in state  $(1, 1)$  or state  $(0, 0)$ ,  $n$  is upgraded to a T-node, and enters the representative node election procedure.

The general idea of representative node election is that each T-node  $n$  generates a random key value, and propagates it to its neighbors. The node with the smallest key value is selected to be the representative node. At the same time, each node selects the node from which the smallest key is first received as its parent, so that a routing tree is built that connects all the T-nodes to the representative node.

In this phase, the task of the nodes in state  $(1, 1)$  or state  $(0, 0)$  is to identify the C-nodes. By definition 4.5, a node  $n$  in state  $(1, 1)$  or state  $(0, 0)$  can be upgraded to a C-node if either of the following conditions is satisfied:

1.  $n$  discovers a direct neighbor node in state  $(0, 1)$  or state  $(1, 0)$ .
2.  $n$  discovers a pair of direct neighbors that are C-nodes and are in state of  $(1, 1)$  and  $(0, 0)$ , respectively.

As an example, after the transition described by Figure 5.1 is observed, the C-nodes shown in Figure 5.5(a) are identified as satisfying condition 1, and the C-node  $n$  shown in Figure 5.5(b) is identified as satisfying condition 2.

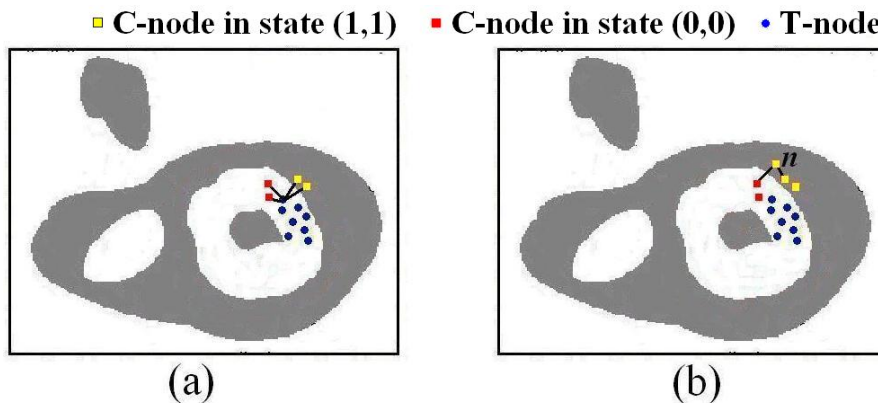


Figure 5.5 Examples of C-nodes identification

To confirm itself to be a C-node, a node in state  $(1, 1)$  or state  $(0, 0)$  waits for messages from its neighbors after each sensing round. It is upgraded to a C-node if either condition is satisfied. If a node  $n$  in state  $(1, 1)$  or state  $(0, 0)$  receives a message indicating the existence of a direct neighboring node in state  $(1, 0)$  or state  $(0, 1)$  (such as an *IS-CHANGED* message from a direct neighbor), using condition 1, node  $n$  is upgraded to a C-node directly, and after that broadcasts a message informing its neighbors about its upgrade. If a node  $n$  in state  $(1, 1)$  or state  $(0, 0)$  discovers a pair of direct neighboring C-nodes upgraded one from a node in state  $(1, 1)$  and one from a node in state  $(0, 0)$ , using condition 2, node  $n$  is upgraded to a C-node. Node  $n$  then broadcasts an upgrade message in order to continue identifying the other C-nodes.

### **Group formation phase**

After the C-nodes are identified, groups are formed, and labels and routing tree structures are maintained amongst the C-nodes. During group formation, a C-node  $n$  first waits for a random amount of time  $t_w$  in the range  $[1, T_w]$ , in which  $T_w$  is the maximum waiting time. When the wait finishes, if no existing group is found for  $n$  to join,  $n$  becomes a group leader. Each group leader propagates a group call message toward the other C-nodes of the same reading. A C-node joins with the same group as the sender of the group call message it first hears. During the group formation, all the C-nodes in the same group are assigned the same label, which is the unique identifier of its group leader. Also, a routing tree is set up within each group that is rooted at the group leader and connects all the boundary nodes in the group. In addition, each group leader is connected in turn to one of its neighboring T-nodes. As a routing tree is also formed amongst the T-nodes during the diffusion phase, all the C-nodes and T-nodes are connected in a single routing tree.

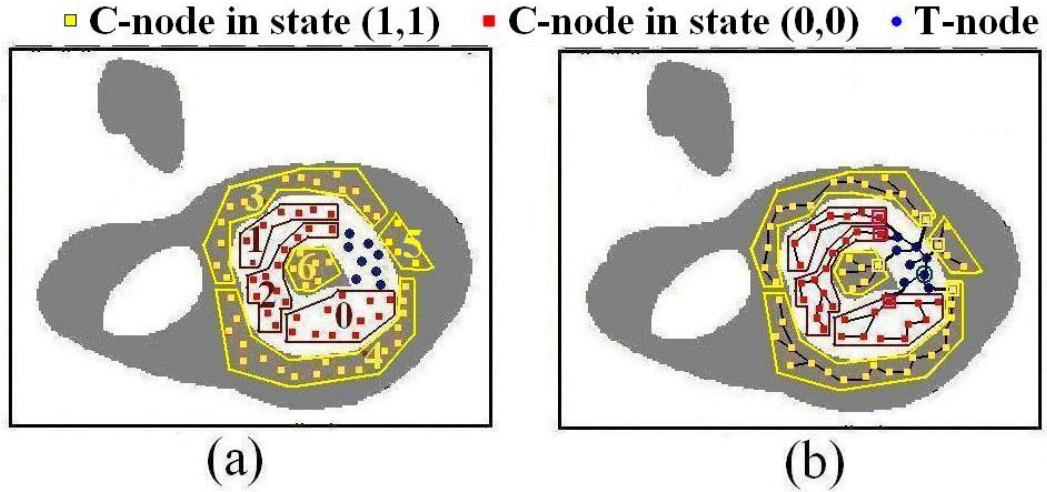


Figure 5.6 Boundary groups and the routing tree

As an example, Figure 5.6(a) shows a possible distribution of groups formed among the C-nodes that are detected immediately after the basic transition in Figure 5.1 is observed. Figure 5.6(b) shows the routing tree that is built in the C-nodes and T-nodes, in which group leaders are indicated by double rectangles, and the representative node is indicated by a double circle.

### Aggregation phase

In this phase, the necessary data, including neighboring label set and group hop distance of each group, are collected and transmitted back to the representative node.

By communication, a C-node  $n$  in a group observes the labels of its direct neighbors. The observed labels that are greater than the label of  $n$  form a *local label set* of  $n$ , denoted by  $L(n)$ . In addition, we define the *neighboring label set* of  $n$ , denoted by  $N(n)$ , to be the union of all the local label sets of nodes contained in the subtree rooted at  $n$ ; that is,  $N(n) = L(n) \cup (\bigcup_{m \in D(n)} L(m))$ , in which  $D(n)$  is the set of descents of  $n$  in the routing tree.

By definition, the neighboring label set of the group  $G$  is  $N(r)$ , where  $r$  is the root of the routing tree in  $G$ . The neighboring label sets can be computed by aggregation. During the aggregation, each node  $n$  computes  $N(n)$  based on the data received from

its direct children, and sends the result to its parent. If node  $n$  is a leaf node in the tree,  $N(n) = L(n)$ . Otherwise, let  $c_1, c_2, \dots, c_k$  be the direct children of  $n$  in the routing tree,  $N(n) = N(c_1) \cup N(c_2) \dots \cup N(c_k) \cup L(n)$ .

As an example, Figure 5.7 shows the detailed structure of group 0. The nodes of the part are named  $a$  to  $g$ , and the node  $a$  is the root. Each node knows its local label set, in which  $L(a) = L(k) = \{6\}$ ,  $L(b) = \emptyset$ ,  $L(c) = L(d) = L(e) = L(f) = L(g) = L(h) = \{4\}$ ,  $L(i) = \{2, 4\}$ , and  $L(j) = \{2, 6\}$ . After the aggregation, each node knows its neighboring label set, in which  $N(b) = N(c) = N(d) = N(e) = N(f) = N(g) = \{4\}$ ,  $N(i) = N(h) = \{2, 4\}$ ,  $N(j) = \{2, 6\}$ , and  $N(a) = N(k) = \{2, 4, 6\}$ .

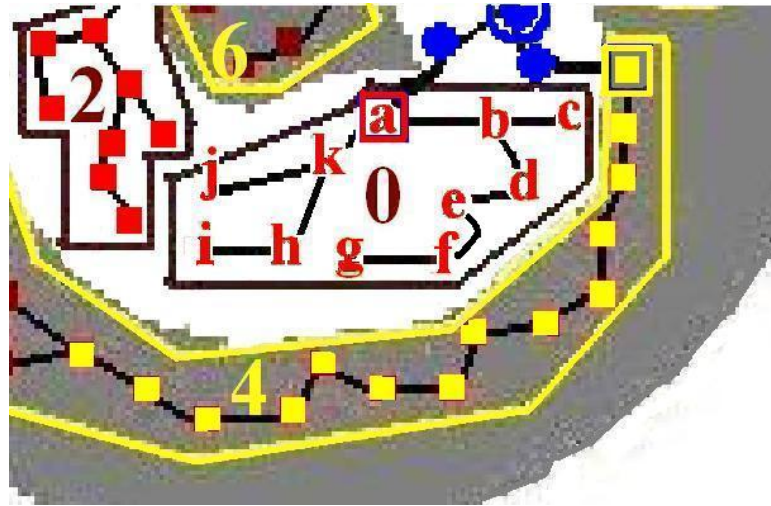


Figure 5.7 Details of group 0

Similarly, with the routing tree, the group hop distance, which is the minimal hop distance of the nodes in the group, can be found at the group leader by a standard aggregation method. Therefore, at the end of the aggregation phase, the group leader is able to send the necessary group information back to the representative node, including the state of the group, its integer label, its neighboring label set, and its group hop distance.

## Analysis phase

After the aggregation phase, the data from all the groups of C-nodes are transmitted to the representative node. In this phase, the representative node forms the report on the observed topological change based on the group level data.

In the analysis, groups form different sets, each set having these properties:

1. All groups in the same set have the same state.
2. Groups in the same set are connected with respect to the adjacency relation.
3. Each set is maximal with respect to properties 1 and 2.

A set satisfying these properties is formed by groups that are located in the same sensor network C-component, and can be used to represent that sensor network C-component. By generating such sets of groups, all the sensor network C-components can be found. Based on the analysis of Section 5.2.1, a pair of sensor network C-components are adjacent if a group in one of the sensor network C-components is adjacent to a group in the other sensor network C-component. The background sensor network C-component is the one that contains the group with the minimal group hop distance. Finally, the state of the transition sensor network component can be determined based on the local readings of the representative node, and the state of the background sensor network C-component can be determined by the state of the group contained in it. Therefore all the necessary information is collected at the representative node, which is able to form the final report about the type of the observed transition.

### 5.3 Decentralized topological change detection using adaptive groups

The TG-based approach reduces the communication cost by requiring the C-nodes to report using the group based technique. However, as the group formed in a sensing round cannot be reused over time in the TG-based approach, the groups may be formed repeatedly among the same set of C-nodes. In some cases, especially when the boundary of the areal object does not change significantly, the communication cost by TG-based method can still be high. This section presents the adaptive group-based (AG-based) approach, which is an improvement on the TG-based approach by reusing the time-invariant information.

By the AG-based approach, some nodes in the sensing area form groups, and each group is assigned a unique integer label, and is allowed to endure over time after it is formed. The structure of groups is dynamically modified after each sensing round, so that the nodes in the same group are always ensured to form a homogeneous component. After each sensing round, only groups that are located near the boundary of the areal object and that have changed their properties are required to report. Based on the information received from the updated groups, the base station maintains the structure of all the boundary groups, and the types of topological changes can be detected at the base station.

As a smaller portion of the groups needs to be updated after each sensing round, the AG-based approach has a significant improvement in reducing the communication cost compared to the TG-based approach.

Algorithm 1 sets out the sketch of the proposed approach, and the following subsections present each step in detail.

---

**Algorithm 1** Topological change detection

---

**1 Initialization**

- 1.1 Each node in the sensor network computes its hop distance to the reference node.
- 1.2 The base station sends out a query to all of the nodes in the whole sensing area.

**2 Boundary group initialization**

The following steps are performed immediately after the first sensing round. Sensor nodes take measurements at specified sensing rounds, and the following steps are performed at each sensing round in a distributed manner:

**2.1 Group formation:** After the first sensing round, groups are formed among the boundary nodes.

**2.2 Group aggregation and reporting:** The group information is aggregated and sent back to the base station.

**3 Monitoring**

Sensor nodes take measurements in specified sensing rounds, and the following steps are performed in each sensing round in a distributed manner:

**3.1 Group update:** Existing groups are modified to ensure that (1) every node located near the boundary of C-components is included in a group, and (2) each group is a homogeneous component.

**3.2 Update aggregation:** The modified groups perform an update, and update messages are sent back to the base station.

**3.3 Data reporting and analysis:** Data are sent back to the base station from group leaders. The base station analyzes data it receives to determine the type of changes.

---

**5.3.1 Initialization**

During initialization, each sensor node computes its hop distance, and a query request is propagated from the base station to the nodes in the sensing area. The initialization procedure in this algorithm is the same as the initialization step that is described in Section 5.2.2.

**5.3.2 Group formation**

After the first sensing round, the initial boundary groups are formed. Each node first communicates with its direct neighbors to identify the boundary nodes. A node is a boundary node if it has a direct neighbor with a different reading. Groups are formed among those boundary nodes.



The group formation of the AG-based approach is the same as the group formation phase in TG-based approach described in Section 5.2.2. The only difference is that in the AG-based approach the groups are formed among all the boundary nodes, instead of only the C-nodes. During the group formation, the boundary nodes in the same group are assigned the same label, which is the unique identifier of its group leader. Also, a routing tree is set up within each group that is rooted at the group leader and connects all the boundary nodes in the group.

As an example, Figure 5.8(a) shows the boundary groups formed among the boundary nodes detected immediately after the first sensing round of the basic transition in Figure 5.1. Figure 5.8(a) shows the details of the boundary group 0, in which the group leader is indicated by a double circle, and the routing tree built among the nodes in the group is indicated by the edges between the nodes.

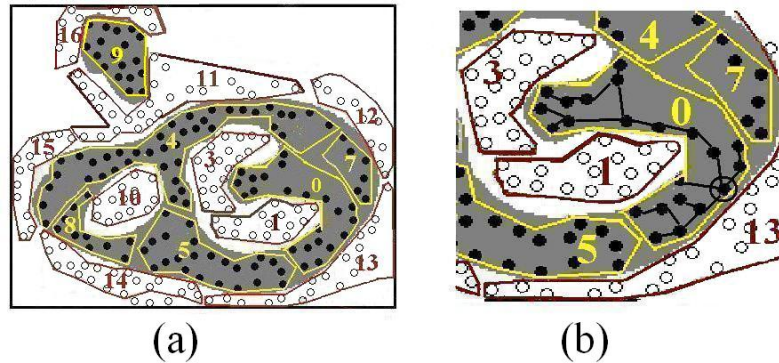


Figure 5.8 An example of boundary group formation

### 5.3.3 Group information aggregation and reporting

After the boundary groups are formed, data necessary for topological change detection are aggregated within each group. The aggregation result of a group includes its neighboring label set, as well as its group hop distance. The aggregation and reporting procedure is the same as the aggregation phase in TG-based approach described in Sec-

tion 5.2.2. The only different is that in AG-based approach, the data are sent from the group leaders to the base station, instead of the representative node.

### 5.3.4 Group update

In the AG-based approach, a group endures after it is formed. However, the readings of nodes in a sensing round  $t_i$  can be different from that in  $t_{i-1}$ . Therefore, the nodes in the same group formed in  $t_{i-1}$  may have different states in  $t_i$ . In order to ensure that the nodes in the same group form a homogeneous component, some groups are modified at  $t_i$ . The modifications include group partial deletions and creations.

The nodes in a group may change its reading after a sensing round. Suppose the node  $n$  in a group  $G$  changes its reading. Then group  $G$  is no longer a homogeneous component, and needs to be updated. A partial deletion is performed, in which the nodes in the subtree originating at  $n$  is deleted from the group  $G$ . To perform the partial deletion, after the sensing round  $t_i$ , any labeled node in state (0, 1) or state (1, 0) is set to be unlabeled, leaves the group, and propagates a *DESTROY* message to its decedents in the routing tree. A labeled node that receives a *DESTROY* message is set to be unlabeled and leaves the group.

The partial deletion of existing groups, as well as the change of the areal object boundary, results in unlabeled boundary nodes that are not included in any groups. To report the properties of these nodes, groups are formed among the unlabeled boundary nodes. The formation procedure is the same as the group formation phase of the TG-based approach described in Section 5.2.2. For an illustration, consider the basic transition described by Figures 5.9(a) and 5.9(a). During the transition some nodes in group 0 change their readings from 1 to 0, which incur the partial deletions. These nodes together with their descendants leave group 0 and become unlabeled boundary nodes, as shown in Figure 5.9(c). In addition, the transition also results in some new

boundary nodes depicted as squares in Figure 5.9(c). These boundary nodes perform a group formation procedure, and a possible result is shown in Figure 5.9(d), in which new groups 2 and 6 are created, and group 0 is expanded to include two boundary nodes.

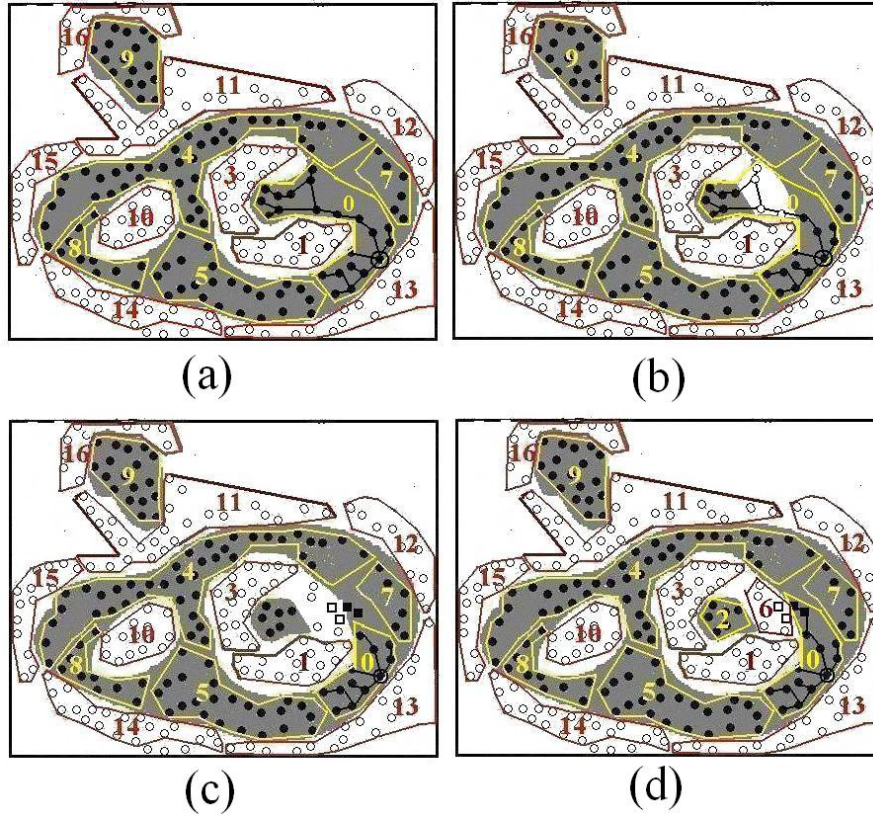


Figure 5.9 An example of group update

### 5.3.5 Update aggregation and reporting

After the group update, information of some groups needs to be computed. The same procedure in the aggregation phase as described in Section 5.2.2 is performed in the new groups to get group level data. In addition, the aggregation is also performed in the groups that are created in previous sensing rounds. The aggregation in these groups only takes place where nodes have different data. As an example, the nodes  $m$ ,  $n$ ,  $o$ ,  $p$ , and  $q$  shown in Figure 5.10 are the only nodes in group 0 whose neighboring label sets and group hop distances may need to be updated. As the data in the rest of the nodes in group 0 do not change, an aggregation among them is unnecessary.

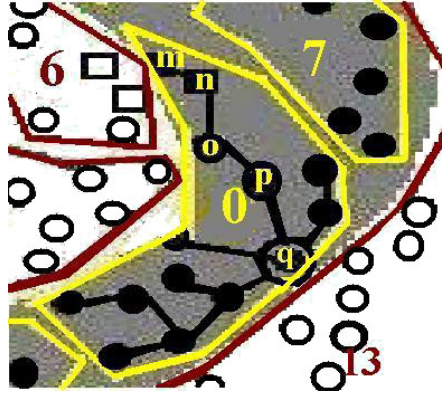


Figure 5.10 An example of group update aggregation

After the update, the messages are sent from the leaders of some groups to the base station. A **creation message** is sent from the leader of each new group  $G$ . This message includes the label of  $G$ , the state of  $G$ , neighboring label set of  $G$ , and the group hop distance of  $G$ . An **update message** is sent back from the leader of a group  $G$  whose group data changed. The update message includes the labels that are added to, or removed from, the neighboring label set of  $G$ , and the group hop distance of  $G$ , if it changes.

### 5.3.6 Data reporting and analysis

Based on the data received after each sensing round, the base station knows the label, the state, and the group hop distance of all the existing groups, as well as the adjacency relations between them.

The following procedure can be performed in the base station to determine the type of a change. First, the base station identifies sets of groups, each set having the following properties:

1. At least one of the groups in the set is adjacent to a group in state  $(0, 1)$  or state  $(1, 0)$ .
2. Groups in the same set have the same state.

3. Groups in the same set are connected with respect to the adjacency relation.
4. Each set is maximal with respect to properties 2 and 3.

Each set satisfying these properties is formed by groups that are located in the same C-component, and can be used to represent that C-component. By generating such sets of groups, all the sensor network C-components can be found. The other necessary information can be obtained in the same way as the analysis phase of the TG-based approach as described in Section 5.2.2. Finally, the type of the topological change can be determined at the base station.

## 5.4 Summary

This chapter presents a basic approach and two improved approaches to topological change detection in sensor networks. Both approaches maintain and report properties of groups after each sensing round in different ways. The collected information allows the detection of topological changes based on the framework presented in Chapter 4.

## Chapter 6

### EXPERIMENTS AND EVALUATIONS

This chapter presents a description of the experiments that were conducted using simulation techniques in order to test the performance of the proposed approaches to topological change detection. We also evaluate the experimental results.

#### 6.1 Experiments

We used Prowler [Pro08], a MATLAB based network simulator, as our simulation environment. In the experiments, the size of the sensing area was set to be 420 units $\times$ 600 units. 2500 nodes were randomly deployed in the sensing area, and the communication graph between the sensor nodes formed a Delaunay triangulation of the whole space. The base station, which collected data and reported the types of observed topological changes, was placed in the top-left corner of the sensing area. In addition, the base station was assigned to be the reference node. Figure 6.1 shows the deployed sensor nodes and their communication links. In the figure, the base station is indicated by a rectangle and the hop distances of the sensor nodes are indicated by different colors.

Both the transient groups-based (TG-based) method and the adaptive group-based (AG-based) method were tested along with the baseline method of basic boundary construction (NBC) in such a configuration. In each experiment, the sensor nodes



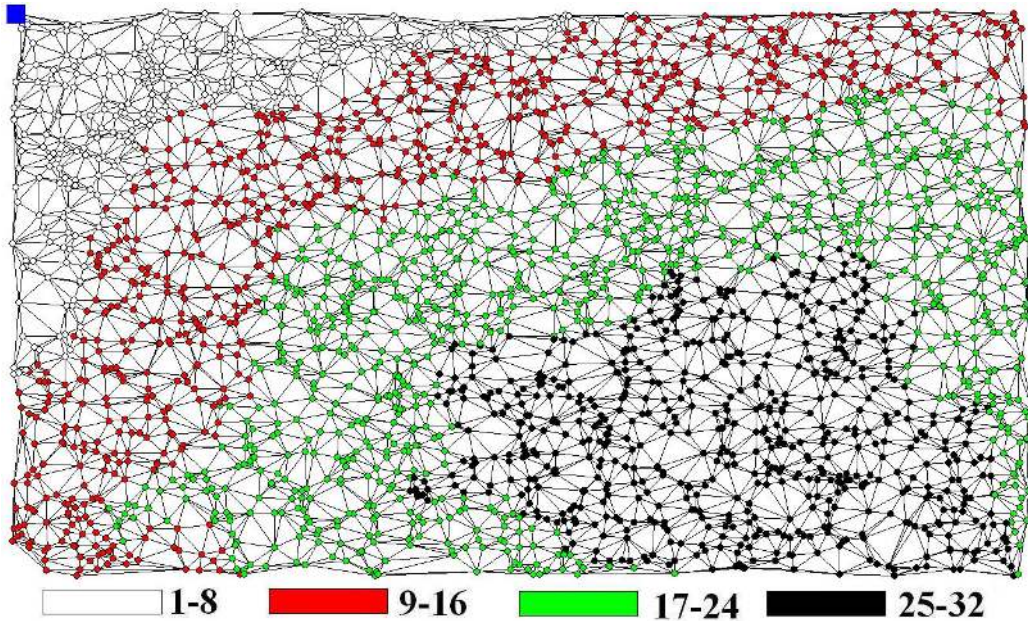


Figure 6.1 The hop distance of the sensor nodes in the experiment

performed topological change detection in sensing rounds  $t_1, t_2, \dots$ , and  $t_{20}$ , in which the difference between  $t_{i+1}$  and  $t_i$  was 60 seconds.

A sequence of 20 snapshots, as shown in Figure 6.2, was generated to provide the sensing data from  $t_1$  to  $t_{20}$ . These snapshots described a scenario, in which an areal object evolved in the form of basic transitions, from a small areal object to a large one. Table 6.1 presents the ratio of the size of the areal object to that of the whole sensing area in each snapshot, as well as the type of topological change that occurs in each sensing round.

In the first and second experiments, we performed the topological change detection by the TG-based and AG-based method, respectively. The maximum waiting time  $T_w$  of a sensor node in both experiments was set to be 25 seconds. In the experimental results, all the reported types of topological changes were the same as expected.

For comparison, in the third experiment we performed the boundary detection using the NBC method. By the NBC method, sensor nodes first communicated with their neighbors at each sensing round to identify the boundary nodes (the nodes that

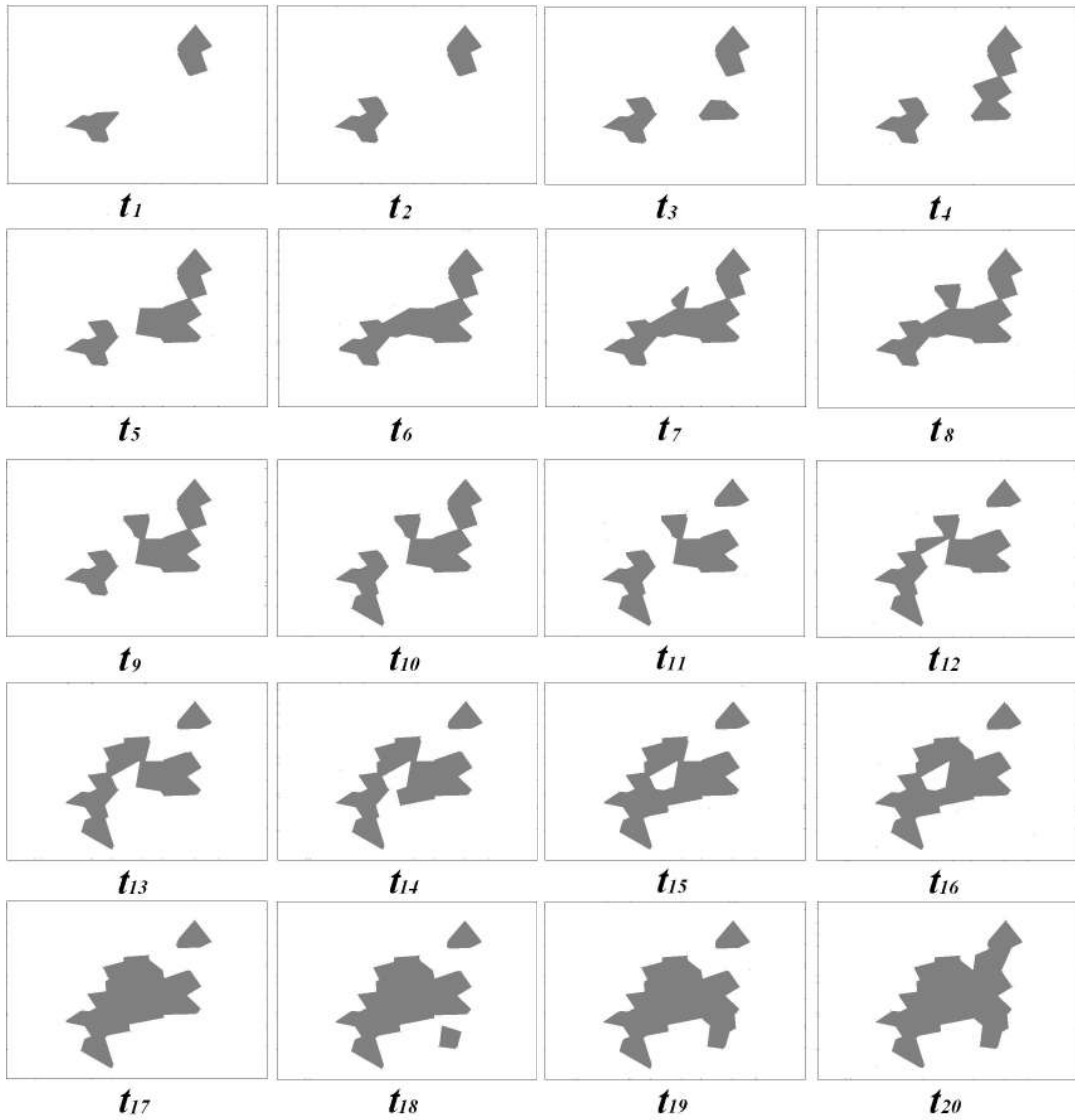


Figure 6.2 Snapshots of experimental data

had a direct neighbor with a different reading). After that, the boundary nodes located in the areal object reported their location information back to the base station via the shortest routes. With the location information received, the base station was able to generate the boundary of the areal object after each sensing round. Because the boundary characterized the topological properties of an areal object, the topological changes could be identified based on the consecutive snapshots of the boundary.



Table 6.1 Experimental data descriptions

<b>Time</b>	$t_1$	$t_2$	$t_3$	$t_4$
<b>Size ratio</b>	7.77%	8.45%	9.71%	11.05%
<b>Type</b>		Region Enlarge	Region Appear	Region Merge
<b>Time</b>	$t_5$	$t_6$	$t_7$	$t_8$
<b>Size ratio</b>	12.85%	13.94%	14.4%	14.88%
<b>Type</b>	Region Enlarge	Region Merge	Region Enlarge	Region Enlarge
<b>Time</b>	$t_9$	$t_{10}$	$t_{11}$	$t_{12}$
<b>Size ratio</b>	13.79%	15.06%	14.03%	14.72%
<b>Type</b>	Region Split	Region Enlarge	Region Split	Region Merge
<b>Time</b>	$t_{13}$	$t_{14}$	$t_{15}$	$t_{16}$
<b>Size ratio</b>	15.49%	16.51%	17.51%	18.03%
<b>Type</b>	Region Enlarge	Region Enlarge	Region self-merge	Region Enlarge
<b>Time</b>	$t_{17}$	$t_{18}$	$t_{19}$	$t_{20}$
<b>Size ratio</b>	19.12%	19.9%	20.81%	21.8%
<b>Type</b>	Hole disappear	Region appear	Region Merge	Region Merge

## 6.2 Evaluation of the transient group-based method

Figure 6.3 shows the communication cost of both the NBC and the TG-based method. In each sensing round, the TG-based method has a lower communication cost than that of the NBC method. The total communication cost of the NBC method in detecting the 19 topological changes is 247743 bytes, and the total communication cost of the TG-based method is 126183 bytes, which is 50.9% of that of the NBC method.

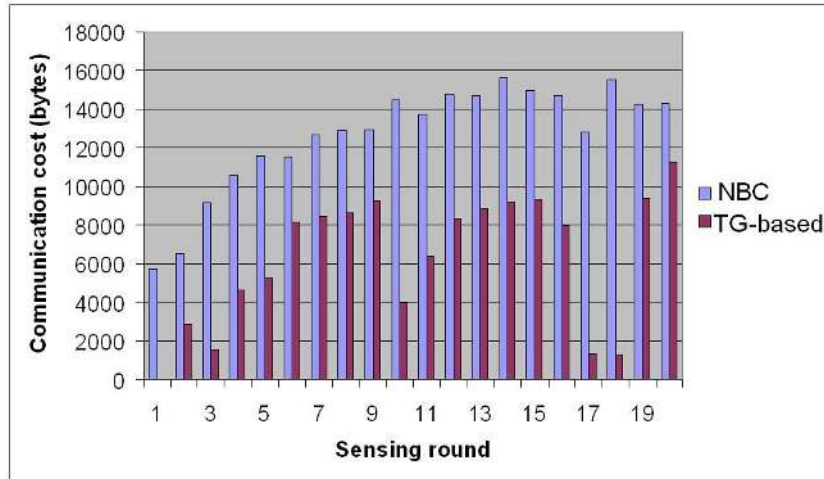


Figure 6.3 Communication cost of NBC and TG-based method

The ratio of the communication cost of the TG-based method to that of the NBC method is not constant in all sensing rounds. We first consider a sensing round

$t_{11}$ , in which the communication cost of the TG-based method is 46% of that of the NBC method. Figure 6.4 shows the C-nodes and T-nodes of the TG-based method at  $t_{11}$ . The report on topological change is first formed by communication amongst the C-nodes and T-nodes, and a unique representative node sends the final report back to the base station. Because the number of the C-nodes and T-nodes is small, only 6350 bytes transmission are needed for reporting. By the NBC method, each boundary node located within the areal object is required to send its location information back to the base station. Figure 6.5 shows the reporting nodes at time  $t_{11}$  and an example of a route for data transmission. As shown in the figure, the number of boundary nodes is large and the cost of sending data back to the base station is high, which results in a higher communication cost of 13734 bytes transmission in reporting by the NBC method.

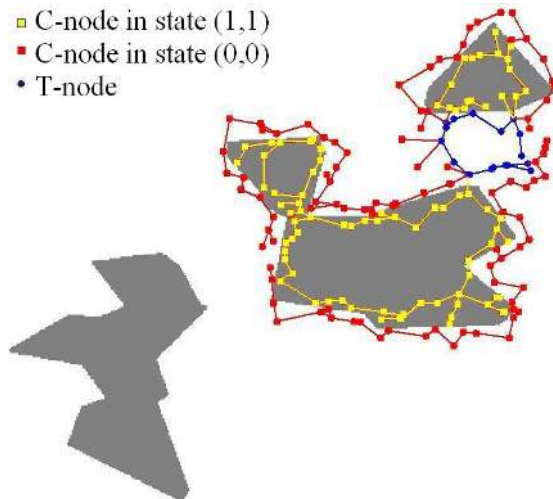


Figure 6.4 The C-nodes and T-nodes at sensing round  $t_{11}$

However, there are some cases in which the communication cost of the TG-based method is close to that of the NBC method. Consider the sensing round  $t_{20}$ , in which the communication cost of the TG-based method is 78% of that of the NBC method. Figure 6.6 shows the C-nodes and T-nodes of the TG-based method at  $t_{20}$ . The number of the C-nodes nodes in sensing round  $t_{20}$  is much larger than that in the sensing round  $t_{11}$ , which increases the communication cost in reporting by the TG-based method.

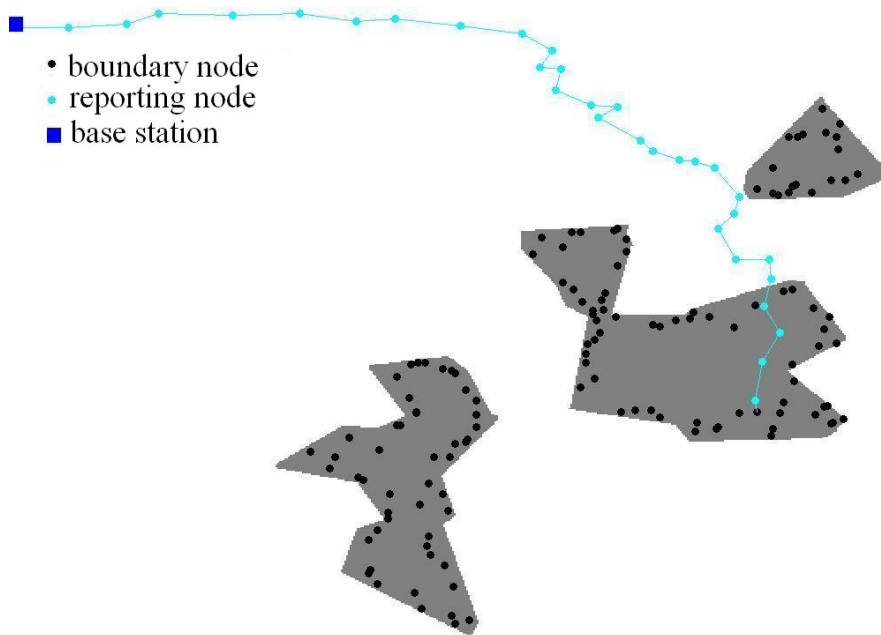


Figure 6.5 The boundary nodes at sensing round  $t_{11}$

However, the number of boundary nodes in  $t_{20}$  is similar to that in  $t_{11}$ . Therefore, the communication cost of the NBC method is similar in both sensing rounds.

Even though there are special cases where TG-based and NBC are similar, in general the TG-based method outperforms the NBC method in topological change detection. This is especially the case when the topological structure of the areal object is complex and the components of the areal object that change their topological properties are small.

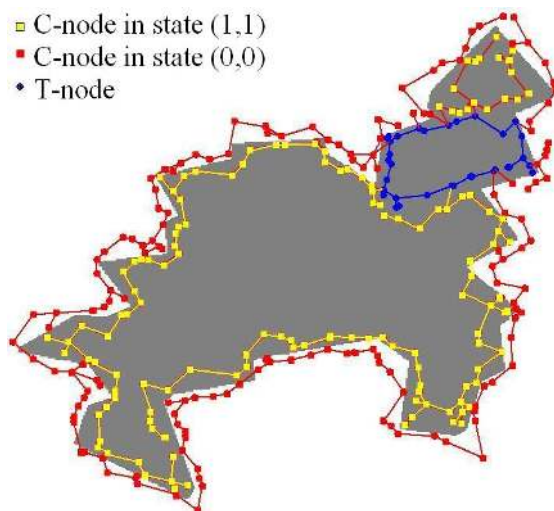


Figure 6.6 The C-nodes and T-nodes at sensing round  $t_{20}$

### 6.3 Evaluation of the adaptive group-based method

Figure 6.7 shows the communication cost of the TG-based and the AG-based method. The AG-based method requires higher communication cost in the first sensing round  $t_1$  than that in the other sensing rounds, because it needs to perform group formation among all the boundary nodes at  $t_1$ . After that, the communication cost is reduced. The total communication cost by the AG-based method is 46159 bytes, which is 36.6% of that by the TG-based method.

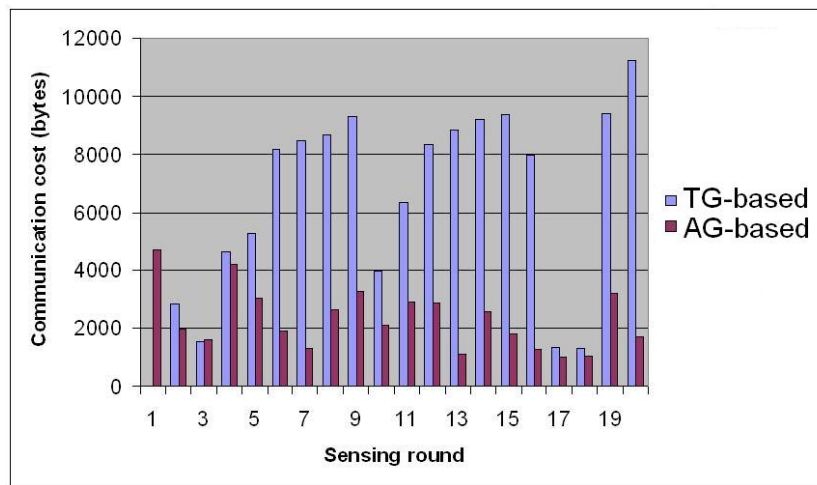


Figure 6.7 Communication cost of the group-based methods

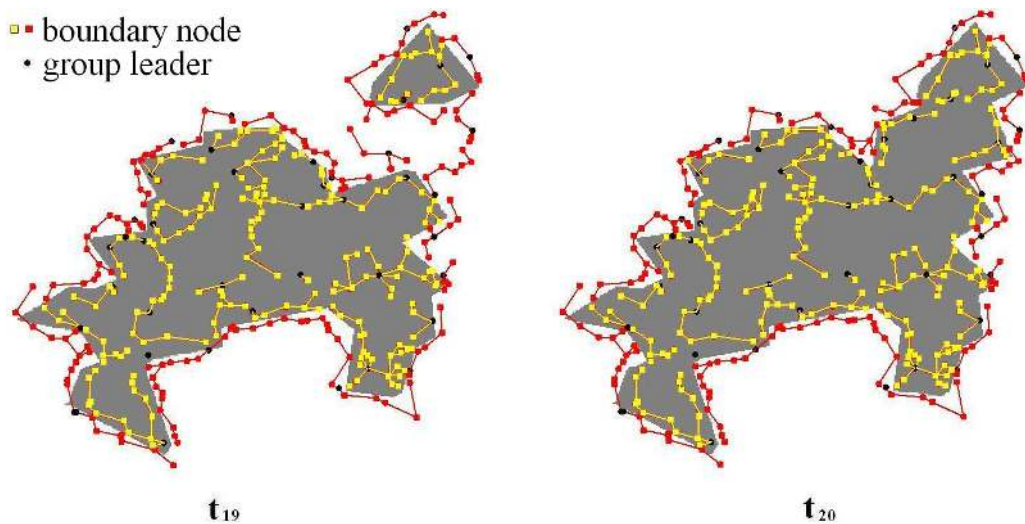


Figure 6.8 The boundary groups after sensing rounds  $t_{19}$  and  $t_{20}$

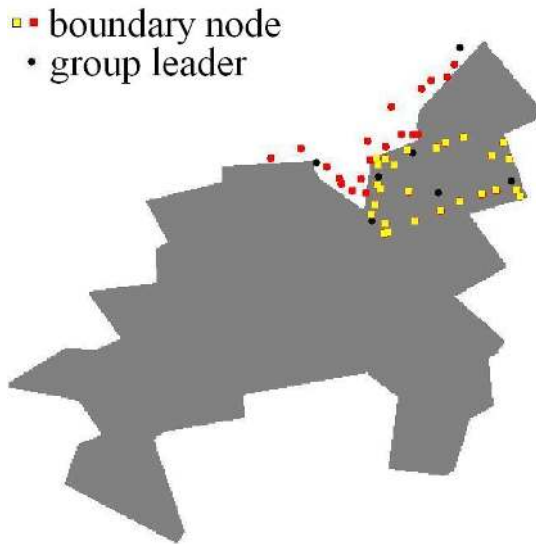


Figure 6.9 The nodes involving in group update at sensing rounds  $t_{20}$

The communication cost of the AG-based method is reduced compared to that of the TG-based method, because the aggregation in the AG-based method takes place only within the locality of the change, and only the updated information is transferred to the base station. Consider the typical sensing round  $t_{20}$ , in which the communication cost of the TG-based method is relatively high. By the AG-based method, not all the boundary groups are formed at  $t_{20}$ . Instead, the boundary groups at  $t_{20}$  are obtained by updating a small number of groups at sensing round  $t_{19}$ . Figure 6.8 shows the boundary groups formed at  $t_{19}$  and  $t_{20}$  in an experimental run. Nodes in the same group are connected by edges, and group leaders are indicated by black circles. Many of the boundary groups at  $t_{19}$  do not change during the transition and can be reused at  $t_{20}$ . Only the group structures among the nodes shown in Figure 6.9 change, and messages describing the changes are sent to the base station from the group leaders indicated by black circles.

This result implies that by reusing the time-invariant information, the AG-based method can reach a lower communication cost than that of the TG-based method, especially when the number of boundary groups that change their structure in each sensing round is small.

## 6.4 Effect of group size for TG-based and AG-based methods

The maximum waiting time  $T_w$  in both the TG-based method and the AG-based method affects the number of boundary groups formed during the detection. With the increase of  $T_w$ , the number of groups formed decreases.

The number of boundary groups may affect the total communication cost. In order to study this effect, we set up different  $T_w$ s within the range  $[5s, 30s]$  in different runs for experiments 1 and 2. The number of groups formed in each run varied, so did the communication cost. Figure 6.10 shows the correspondences between the communication cost and the total number of groups for both methods. The communication cost of the TG-based method is nearly constant for different number of groups.

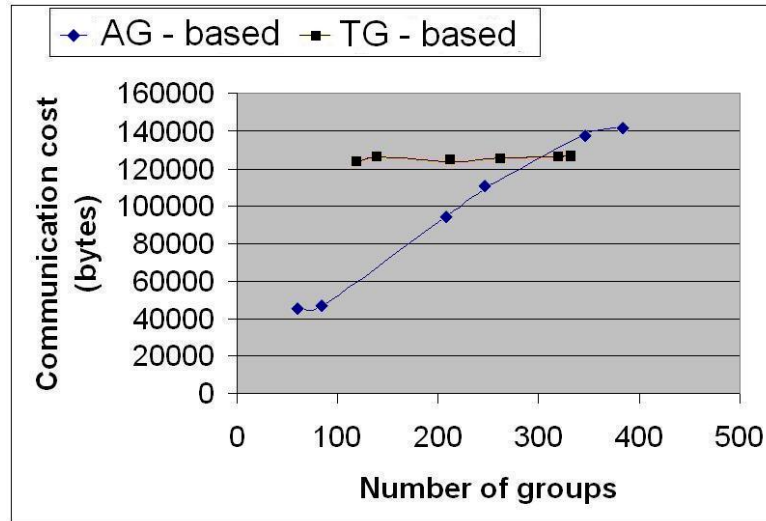


Figure 6.10 Communication cost in different number of groups

However, the communication cost of the AG-based method increases significantly as the total number of groups increases. In some extreme cases, the total communication cost of the AG-based method is even higher than that of the TG-based method. The reason is as follows. With the number of groups increasing, more data need to be transferred back either to the representative node or to the base station. The representative

node in the TG-based method is ensured to be located close to the group leaders, so the communication cost remains nearly constant when the amount of data being transferred increases. Whereas the base station in the AG-based method is located at the boundary of the whole sensing area and is usually far away from the group leaders. Hence, the number of transmissions increases significantly when more data need to be transferred back to the base station.

The communication cost of the AG-based method increases with the increase of the number of groups formed, which is caused by the decrease of the maximum waiting time  $T_w$ . Therefore, the AG-based method may not have a better performance than the TG-based method in some applications that require a higher temporal resolution.

## 6.5 Summary

This chapter presents the experimental results of the topological detection methods we have developed in Chapter 5. Both methods generate correct sensing reports using a lower communication cost than that of the basic boundary construction method. By employing the boundary group based framework, the transient group-based method reduces the communication cost to 50% of that of the basic boundary construction method in topological change detection. By reusing the time-invariant information, the adaptive group-based method further reduces the communication cost to 36.6% of that of the transient group-based method.

## Chapter 7

### CONCLUSIONS

Chapter 7 concludes this thesis. It presents a summary and lists the main contributions of our work. We also provide a critical analysis of the research undertaken and presented in this thesis. Finally, we provide suggestions for future areas of research.

#### 7.1 Summary of thesis

Topological changes to regions, such as merging, splitting, hole formation and elimination, are significant events in the evolution of regions. Information about such salient changes is useful in many applications. Sensor network technology has the potential to play an important role in detecting topological changes. However, due to the limited computational ability and the energy constraints of sensor networks, applications of topological change detection using sensor networks require both formalization of topological changes and development of energy-efficient detection algorithms. The solutions to these problems will contribute to the design and deployment of large scale sensor-network applications in the future.

The thesis provides theoretical foundations and algorithmic solutions to topological change detection. Two models, the *morphism-based model* and the *local tree model*, are developed, providing formal semantics of topological changes.



The morphism-based model represents dynamic topological properties of continuously evolving areal objects. Assuming that the areal objects under consideration are strongly connected, tree structures are employed to represent topological relations between regions and holes of areal objects. Basic and complex changes are specified using structure-preserving mappings between trees. The morphism-based model allows us to perform a detailed analysis of topological changes. Based on the model this work constructs a normal form and proves that it is the “simplest” form that could represent all the changes under consideration.

The local tree model represents the discrete and incremental changes of the areal objects based on selected components and relations between them. It allows us to formally specify different kinds of topological change, among which some specific types of changes, such as appear, disappear, split, and merge, are differentiated.

Based on the local tree model, we develop two distributed and energy-efficient approaches, the *transient group-based (TG-based) approach* and the *adaptive group-based (AG-based) approach*, to topological change detection. Both approaches capture the fundamental information required by the local tree model. Compared to the basic data collection approach, the TG-based approach employs the boundary group framework, which reduces the communication cost by reporting only the group level data instead of data from each individual node. The AG-based approach further reduces the communication cost by reusing the time-invariant group level data.

The proposed approaches are evaluated by experiments using simulation techniques. The experimental results show that when the configurations of sensor networks satisfy certain density and communication constraints, the proposed approaches are able to generate correct reports on the topological changes, and at the same time reduce the communication cost to a level much lower than that of the basic boundary-based approach.

## 7.2 Major findings and limitations

### 7.2.1 The morphism-based model

This work discovers the correspondence between tree morphisms and dynamic topology of evolving areal objects. A tree can be employed to represent the static topological structure of an areal object in a single snapshot. A tree morphism, which is a structure-preserving mapping between a pair of trees, provides a representation of an evolving areal object within a specific time period. Based on the properties of the morphism, four basic types of basic topological changes, insert, delete, split and merge, are formally specified.

In terms of shortcomings, the morphism-based model does not make the distinction between strong and weak connectivity, and therefore is unable to differentiate certain types of topological changes. See Section 7.4.1 for detailed discussions and suggestions of future work of this area. In addition, the completeness of the morphism-based model is not proved, the proof of the completeness requires a formalization of continuous spatial change, which is discussed in detail in Section 7.4.2.

### 7.2.2 The normal form

Any composition of a finite sequence of basic topological changes form a complex topological change. This work proposes a normal form, which is the structured composition of four basic types of topological changes, for the simplification of any complex topological change. This work proves that any complex topological change can be represented by a change in normal form. In addition, it also proves that the normal form is the “simplest” form for the representation of all the complex topological changes.

On the other hand, the normal form does not ensure that any complex topological change has a unique representation. Please refer to Section 7.4.3 for detailed discussions and suggestions for future work of this area.

### **7.2.3 The local tree model**

This thesis analyzes a simple pattern of changes, called basic transitions, in which areal objects evolve discretely and incrementally. The analysis identifies that the properties and relations between the components located near the transition region are the fundamental features to determine the type of a topological change. The local tree model is proposed to represent these fundamental properties and relations, based on which different kinds of topological changes are specified. This work also proves that these fundamental properties and relations in any basic transition can be represented by a rooted tree. Therefore, the types of topological changes specified based on the local tree model provide a complete coverage over all the basic transitions.

The local tree model does have some limitations. Similar to the morphism-based model, the local tree model does not make the distinction between weak and strong connectivity of areal objects. In addition, the local tree model does not handle continuous or non-incremental changes. Please refer to Section 7.4.4 for detailed discussions and suggestions for future work of this area.

### **7.2.4 Distributed algorithms of topological change detection**

Based on the local tree model, distributed algorithms of topological change detection in sensor networks are proposed. These algorithms capture exactly the fundamental properties and relations required by the local tree model. Therefore, they are able to provide sensing reports for any basic transition of the areal object when the sensor network is properly configured. In addition, the thesis proposes the boundary group based

framework to organize the sensor nodes during reporting, which significantly reduces communication cost in topological change detection compared to the basic boundary-based approach.

On the other hand, both proposed algorithms are based on the local tree model. Thus, they may not be able to generate correct reports when monitoring continuous or non-incremental changes. In addition, both algorithms have limitations in detecting topological changes in 3-dimensional space. Please refer to Section 7.4.5 for a detailed discussion. Finally, both approaches may have low reporting accuracy if the network configuration fails to conform to the density and communication constraints, which may be caused by energy exhaustion or hardware failure.

### 7.3 Testing the validity of the hypothesis

The morphism-based model and the local tree model presented in this thesis represent fundamental dynamic topological properties of areal objects. Both models allow us to specify basic types of topological changes. Based on the morphism model, we identified a normal form of *ISMD* for the simplification of complex topological changes, and proved that this form is the simplest form to represent all the complex changes. The results support the first part of the hypothesis; that is, “models that represent dynamic topological properties of areal objects provide the capability of formally specifying and analyzing dynamic topological changes, which go beyond previous models that represent only static topological properties and relations.”

In addition, the local tree model allows us to develop distributed algorithms for topological change detection, which include both the transient group-based and the adaptive group-based approaches. The experiments show that, under the density and communication constraints, both the transient group-based and the adaptive group-based approaches are able to generate correct sensing reports on topological changes,

and their communication costs are 50.9% and 18.6% of that of the basic boundary-based data collection approach. This result supports the second part of the hypothesis; that is, “these models also allow the construction of distributed algorithms of topological change detection that are more energy-efficient than current approaches, such as the basic boundary-based approach.”

However, it is important to mention that the improvement of both topological change detection approaches are achieved based on two assumptions. First, the areal object that is being monitored evolves in the pattern of basic transitions. Second, the sensor nodes are densely deployed and the communication graph between the sensor nodes form a Delaunay triangulation of the whole space. Both assumptions limit the applications of the proposed approaches. The extension of both approaches to process more complex transitions under a simpler network configuration is an important part of future work.

## **7.4 Future work**

The future work in the area of dynamic areal object modeling includes dynamic topological models of weakly connected areal objects, the formal specification of continuous changes, as well as the unique representation of topological changes. The future work in the area of topological change detection includes topological change detection during non-incremental evolution of areal objects and topological change detection in three dimensional space.

### **7.4.1 Dynamic topological models of weakly connected areal objects**

A basic assumption of this thesis is that the areal objects under consideration are strongly connected; that is, the boundary of an areal object is a set of disjoint Jor-

dan curves. However, the problem becomes more complicated if weakly connected areal objects are considered. An areal object  $R$  is defined to be *weakly connected* if the number of connected components of  $R$  or of  $S \setminus R$  can be changed by removing a finite number of points from  $R$ . Distinguishing weak connectivity from strong connectivity enables us to differentiate some salient changes, for example, the two different splits in Figure 7.1, as presented in [Gal97].

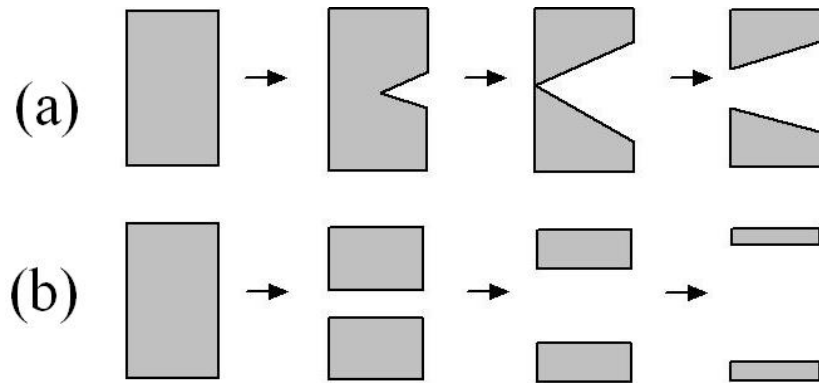


Figure 7.1 Two different splits (from [Gal97])

The specification of the topological changes of weakly connected areal objects requires a faithful model for the representation of static weakly connected areal objects. The models developed in this thesis are component-based and they do not make the distinctions between weakly and strongly connected areal objects. Therefore, it is difficult to represent weakly connected areal objects based on these models.

One of the possible models of weakly connected areal objects is the boundary-based representation, as there is a correspondence between the boundary of an areal object and a planar Eulerian map with a distinguished face. We assume that the boundary of areal object  $R$  is planar map  $M$  embedded in the plane, and the unbounded face of  $M$  is defined to be the distinguished face.  $M$  must be a planar Eulerian map. The proof of the Eulerian property is straightforward. It can be verified that each vertex of the planar graph  $M$  is of an even degree, because each vertex must be surrounded by sectors that belong to  $R$  and to the complement of  $R$  alternatively. Conversely, the dual

graph of a planar Eulerian graph is a planar bipartite graph. It follows that for any planar Eulerian map  $M$  with a distinguished face there must be an areal object whose boundary can be represented by  $M$ .

In addition, paper [BFG04] provides an one-to-one representation of a planar Eulerian map with a distinguished face. This representation is based on a special kind of labeled tree, defined as follows:

**Definition 7.1.** *A planar tree is defined to be a well-labeled tree, if it has the following properties:*

1. *Its vertices are of two types, unlabeled ones and labeled ones carrying strict positive integer labels.*
2. *Each edge connects a labeled vertex to an unlabeled vertex.*
3. *For each unlabeled vertex  $v$ , the labels  $n$  and  $m$  of two labeled vertices adjacent to  $v$  and consecutive in clockwise direction satisfy  $m \geq n - 1$ .*

*In addition, a well-labeled tree is defined to be a minimal well-labeled tree, if at least one of its vertices has label 1.*

It is proved that there is a one-to-one mapping between the minimal well-labeled trees and the Eulerian planar maps with distinguished faces. Therefore, the well-labeled tree provides a canonical representation of an areal object whose boundary is connected. Extensions of the well-labeled trees are expected to provide faithful models of any dynamic weakly connected areal object.

#### **7.4.2 Formal specification of continuous changes**

The tree morphism model in this thesis is presented by assuming that an areal object changes continuously. Studying the properties of the morphism model relies on a formal

definition of continuous changes. Basically, an areal object can be defined as a subset of  $\mathfrak{R}^2$ . Let  $\Delta$  be the set of all the possible areal objects in  $\mathfrak{R}^2$ , and let  $T$  be the temporal domain. Therefore, a change  $C$  can be represented by a function  $f : T \rightarrow \Delta$ , which indicates the state of the areal object at any time during the change. The specification of a continuous change  $C$  depends on the specification of the continuity of  $f$ .

An initial attempt to define the continuity of  $f$  is done by Galton [Gal97]. This work defines metrics on the regions in  $\Delta$ , and defines the continuity of  $f$  in the form of the standard  $\epsilon - \delta$  definition for continuous functions on the real numbers. The definition provides a classification of continuous changes, but it has some limitations, as some changes that are intuitively non-continuous are defined to be continuous changes by this definition. The limitations prevent us from getting comprehensive analysis results of the tree morphism model.

The general topology offers an extended way for the definition of continuous changes. Let  $(X, \tau)$  and  $(Y, \tau^*)$  be topological spaces. A function  $f$  from  $X$  to  $Y$  is continuous relative to  $\tau$  and  $\tau^*$ , iff  $H \in \tau^*$  implies  $f^{-1}[H] \in \tau$ . This definition of continuity gives a framework of definition of continuous changes. However, to make the definition applicable to the function  $f : T \rightarrow \Delta$ , it is necessary to specify a topology on  $\Delta$ . The specification of topology on the set of all areal objects requires the studies of both the properties of areal objects and the properties of continuous changes in human intuition.

### 7.4.3 Unique representation of topological changes

The normal form proposed in this thesis for the simplification of topological changes does not ensure a unique representation. A complex change can have more than one representation in normal form. For example, the complex change shown in Figure 7.2(a)



is equivalent to the complex changes shown in Figure 7.2(a) and 7.2(c), both of which are in normal form.

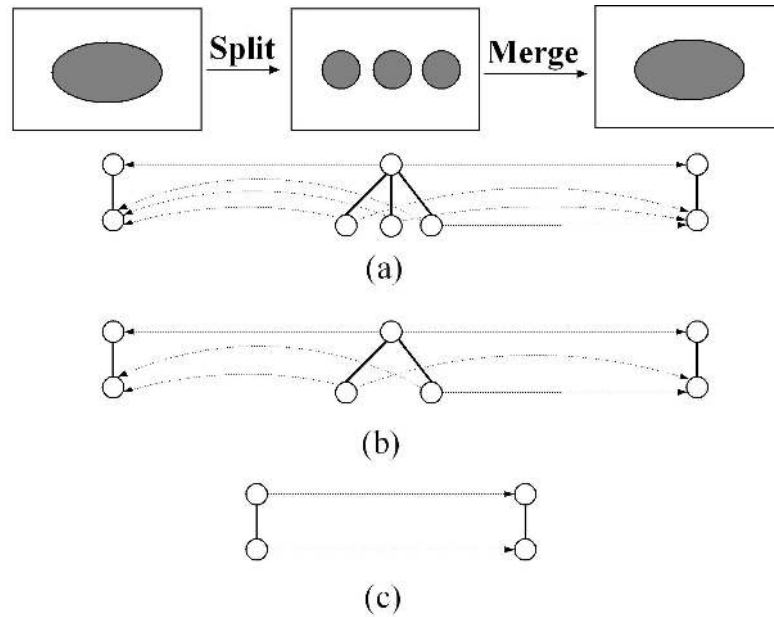


Figure 7.2 Decomposition of a transition region

In order to make sure that each complex change has a unique representation, additional constraints need to be added to the normal form. For example, we can require that among all the equivalent changes in normal form, the representation change must have the smallest number of vertices in the trees. With this requirement, the complex change in Figure 7.2(a) is not considered to be a representation for the change in Figure 7.2(a). In the future work, all the possible factors leading to non-unique representation need to be found. After that, a refined normal form with constraints can be proposed, which ensures a simple and unique representation for all complex changes.

#### 7.4.4 Topological change detection during non-incremental evolution of areal objects

The local tree model and topological change detection algorithms proposed in this thesis successfully specify and detect topological changes under the assumption that the areal objects evolve discretely and incrementally. During each incremental transition, a piece

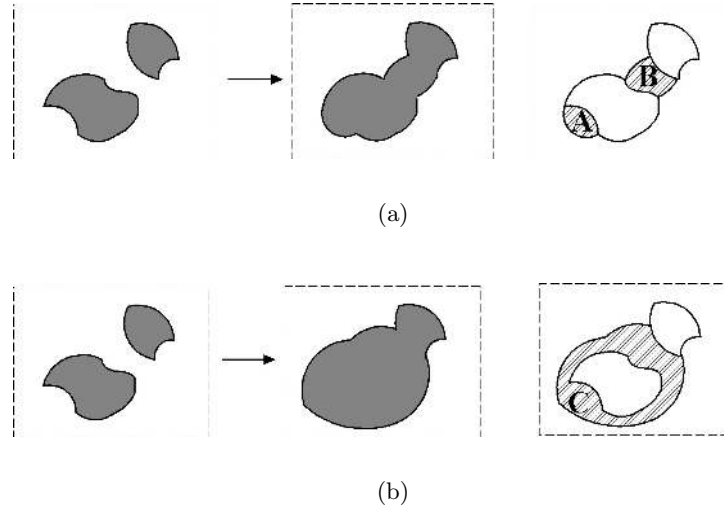


Figure 7.3 Examples of non-incremental changes  
 (a) two regions  $A$  and  $B$  are added to the areal object, and (b) region  $C$  is added to the areal object

of the areal object topologically equivalent to a disk is either added to or removed from the areal object.

However, the incremental transition assumption is strict and has some limitations. Extensions of current research are required in order to handle non-incremental transitions, which include: (1) changes with more than one transition region (as shown in Figure 7.3(a)), and (2) changes whose transition regions have holes (as shown in 7.2(b)).

One of the possible approaches to deal with these complex transitions is to decompose their transition region into several simple transition regions, and topological changes incurred by each simple transition region are processed separately using the proposed models and algorithms developed in this thesis. As an example, Figure 7.4 shows a possible decomposition of the complex change in Figure 7.3(b) into two changes with transition regions being  $M$  and  $N$ , respectively. Detailed analysis is required to show that (1) the new model is complete after these extensions; that is, after the extensions, the approach can be applied to detect any topological changes during continuous evolu-

tion of an areal object. (2) The extended algorithms are able to capture the necessary information required by the new model.

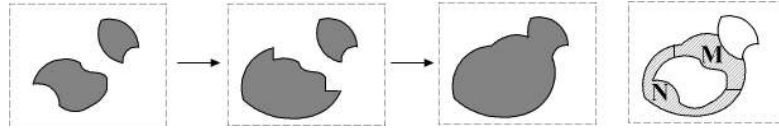


Figure 7.4 Decomposition of a transition region

### 7.4.5 Topological change detection in three dimensional space

This work focuses on processing data in two dimensional space. Many environmental phenomena, such as the temperature distribution over a whole lake, require the handling of three dimensional spatial objects and fields. Topological change detection in three dimensional space is more challenging than that in two dimensional space. First, models of three dimensional objects are more complex; for example, they need to deal with different types of holes. In addition, as presented in [PPKS06], the deployment of sensor networks in three dimensional space presents the following challenges:

1. In a random deployment, the number of communicating neighbors within a sensing range of a single node in three dimensional space is on average twice the corresponding number for two dimensions. This increases the communication cost for the distributed algorithms that are based on communication between neighboring nodes.
2. In three dimensional space there is no natural ordering of neighbors based on angles as in two dimensional space. This makes an approach based on the orderings of nodes more difficult to develop in three dimensional applications.
3. To check the coverage in three dimensional space, each node takes  $O(d^3)$  time, where  $d$  is the average number of neighbors in the sensing area. In two dimensional space, the cost is  $O(d^2)$ .

4. Regular arrangement of nearest neighbors around a node becomes a simple problem in two dimensions and is possible for any number of neighbors. In three dimensions the only regular arrangements of nearest neighbors correspond only to the five regular convex polyhedral.

These challenges require the development of new techniques of topological change representation and detection in three-dimensional space.

## APPENDIX: PROOFS OF LEMMAS RELATED TO MORPHISMS

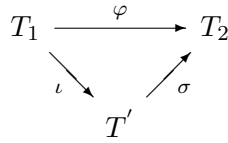
The proofs of lemmas 3.1-3.3 are provided in this appendix as lemmas 1-3. After presenting each lemma, we also state the applications of the lemma to the equivalence complex changes by corollaries.

Note: In the proofs, we use  $\varphi$  to represent a general function, and  $img(\varphi)$  to represent the image of  $\varphi$ . We use  $\sigma$  and  $\iota$  to represent surjective and injective functions respectively. Given a graph  $G$ , we use  $V(G)$ ,  $E(G)$  to represent the sets of vertices and edges of the graph  $G$  respectively. We use  $r(G)$  to represent the root of  $G$ , if  $G$  is a tree.

**Lemma 1.** *Let  $T_1$  and  $T_2$  be trees, and  $\varphi$  be a tree morphism from  $T_1$  to  $T_2$ . Then, it is possible to find another tree  $T'$ , an injective tree morphism  $\iota$  from  $T_1$  to  $T'$ , and a surjective tree morphism  $\sigma$  from  $T'$  to  $T_2$ , satisfying:*

(1)  $(\iota \circ \sigma) = \varphi$ .

(2) *Let  $S_1 = V(T') \setminus img(\iota)$  and  $S_2 = V(T_2) \setminus img(\varphi)$ . Then  $\sigma$  defines a bijection between  $S_1$  and  $S_2$ , by restricting the domain of  $\sigma$  to  $S_1$ .*



*Proof.* We prove this lemma by providing an approach to construct  $T'$ ,  $\sigma$  and  $\iota$ .

**First**, we define vertices of  $T'$ , and functions  $\iota$  and  $\sigma$  as follows:

1.  $V(T') = V(T_1) \cup (V(T_2) \setminus img(\varphi))$ .
2.  $\iota$  is a function with domain  $V(T_1)$  and codomain  $V(T')$ .  $\forall x \in V(T_1), \iota(x) = x$ .
3.  $\sigma$  is a function with domain  $V(T')$  and codomain  $V(T_2)$ .  $\forall x \in V(T')$ ,  $\sigma(x) = \varphi(x)$  if  $x \in V(T_1)$ , and  $\sigma(x) = x$  if  $x \in V(T_2) \setminus img(\varphi)$ .

It can be verified, as follows, that  $\sigma$  and  $\iota$  satisfy conditions (1) and (2) described in the lemma.

- (1) According to the definitions of  $\iota$  and  $\sigma$ ,  $\forall v \in V(T_1)$ ,  $(\iota \circ \sigma)(v) = \sigma(v) = \varphi(v)$ .
- (2) According to the definition of  $\iota$ ,  $S_1 = V(T') \setminus \text{img}(\iota) = V(T') \setminus V(T_1) = V(T_2) \setminus \text{img}(\varphi) = S_2$ . Moreover, according to the definition of  $\sigma$ , by restricting the domain of  $\sigma$  to  $S_1$ ,  $\sigma$  defines an inclusion map from  $S_1$  to  $S_2$ , which is a bijection between  $S_1$  and  $S_2$ .

**Second**, we define root and edges of  $T'$ . The root of  $T'$  is defined to be  $r(T_1) \in V(T')$ , and the edges of  $T'$  are defined as follows:

Let  $v_1$  and  $v_2$  be vertices of  $V(T')$ .

1. In the case that  $v_1$  and  $v_2$  are contained in  $V(T_1)$ ,  $v_1$  and  $v_2$  are adjacent in  $T'$  if and only if  $v_1$  and  $v_2$  are adjacent in  $T_1$ .
2. In the case that  $v_1$  and  $v_2$  are contained in  $V(T_2) \setminus \text{img}(\varphi)$ .  $v_1$  and  $v_2$  are adjacent in  $T'$  if and only if  $v_1$  and  $v_2$  are adjacent in  $T_2$ .
3. In the case that  $v_1 \in V(T_1)$  and  $v_2 \in V(T_2) \setminus \text{img}(\varphi)$ , let

$$U(v_1) = \varphi^{-1}(\varphi(v_1)) = \{v : v \in V(T_1), \varphi(v) = \varphi(v_1)\}.$$

$v_2$  is adjacent to one of the vertex in  $U(v_1)$  if and only if  $v_2$  is adjacent to  $\varphi(v_1)$  in  $T_2$ .  $v_2$  is adjacent to none of the vertex in  $U(v_1)$  if and only if  $v_2$  is not adjacent to  $\varphi(v_1)$  in  $T_2$ .

Based on the definition of edges of  $T'$ , it can be proved, as follows, that (1)  $T'$  is a tree, (2)  $\iota$  is a tree morphism, and (3)  $\sigma$  is a tree morphism.

- (1) We prove  $T'$  is a tree by showing that  $T'$  is both connected and cycle-free.

We first prove that any vertex of  $T'$  is connected to the same vertex in  $T'$ , and hence  $T'$  is connected.

Let  $v$  be a vertex of  $T'$ . There are two cases to consider.

a) If  $v \in V(T_1)$ , we are able to find a path  $P$  in  $T_1$  connecting  $v$  and  $r(T_1)$ . It is trivial to infer that  $P$  is also a path in  $T'$  connecting  $v$  and  $r(T_1)$ . Therefore, any vertex of  $T'$  contained in  $V(T_1)$  is connected to  $r(T_1)$  in  $T'$ .

b) Otherwise, consider  $v \in V(T_2) \setminus \text{img}(\varphi)$ . In this case, let

$$[v = v_0, v_1, \dots, v_n = r(T_2)]$$

be a path in  $T_2$  connecting  $v$  and  $r(T_2)$ . Since  $v_0 = v \in V(T_2) \setminus \text{img}(\varphi)$  and  $v_n = r(T_2) = \varphi(r(T_1)) \in \text{img}(\varphi)$ , there must be an integer  $m$  ( $0 < m \leq n$ ), such that  $v_m \in \text{img}(\varphi)$  and  $\forall i$  ( $0 \leq i < m$ ),  $v_i \in V(T_2) \setminus \text{img}(\varphi)$ . Then according to the definition of edges of  $T'$ ,  $[v = v_0, v_1, \dots, v_{m-1}]$  is a path in  $T'$ , and  $v_{m-1}$  is adjacent to a vertex  $v' \in \varphi^{-1}(v_m) \subset V(T')$ . As shown in a),  $v'$  is connected to  $r(T_1) \subset V(T')$ , it follows that  $v$  is connected to  $r(T_1)$  in  $T'$ .

Any vertex of  $T'$  is connected to  $r(T_1)$  in  $T'$ , therefore  $T'$  is connected.

We prove that  $T'$  is cycle-free by contradiction. Assume to the contrary that there is a simple cycle  $C = [v_0, v_1, \dots, v_n = v_0]$ , ( $n > 1$ ) in  $T'$ .

If  $\forall i$  ( $0 \leq i \leq n$ ),  $v_i \in V(T_1)$ , according to the definition of edges of  $T'$ ,  $C$  is a simple cycle of  $T_1$ , contradicting to the fact that  $T_1$  is a tree.

If  $\forall i$  ( $0 \leq i \leq n$ ),  $v_i \in V(T_2) \setminus \text{img}(\varphi)$ , according to the definition of edges of  $T'$ ,  $C$  is a simple cycle of  $T_2$ , contradicting to the fact that  $T_2$  is a tree.

Otherwise, consider the case in which  $\exists i, j (0 \leq i, j \leq n)$ , such that  $v_i \in V(T_2) \setminus \text{img}(\varphi)$  and  $v_j \in V(T_1)$ . In this case, we are able to find a connected part of the cycle  $C$ , either a path on  $C$  or the whole cycle of  $C$ , with the form  $[v_{i1}, v_{i2}, \dots, v_{ik}]$  such that  $k > 2$ ,  $v_{i1}, v_{ik} \in V(T_1)$ , and  $\forall m (2 \leq m \leq k - 1), v_{im} \in V(T_2) \setminus \text{img}(\varphi)$ . According to the definition of edges of  $T'$ , it follows that  $[\varphi(v_{i1}), v_{i2}, v_{i3}, \dots, v_{i(k-1)}, \varphi(v_{ik})]$  is a simple path in  $T_2$  connecting  $\varphi(v_{i1})$  and  $\varphi(v_{ik})$  with vertices in  $V(T_2) \setminus \text{img}(\varphi)$ . As either  $\varphi(v_{i1})$  and  $\varphi(v_{ik})$  is also connected in  $T_2$  by path with vertices in  $\text{img}(\varphi)$  or  $\varphi(v_{i1}) = \varphi(v_{ik})$ , there must be a simple cycle in  $T_2$ , contradicting to the fact that  $T_2$  is a tree.

All the possible cases lead to contradictions, therefore  $T'$  is cycle free, and  $T'$  is a tree.

(2) We prove  $\iota$  is a tree morphism.

$\iota$  is a function with domain  $V(T_1)$  and codomain  $V(T')$ . For any pair of adjacent vertices  $v_1$  and  $v_2$  of  $T_1$ ,  $\iota(v_1) = v_1$  and  $\iota(v_2) = v_2$ . According to the definition of the edges of  $T'$ , we have  $\iota(v_1)$  and  $\iota(v_2)$  are adjacent in  $T'$  whenever  $v_1$  and  $v_2$  are adjacent in  $T_1$ . In addition,  $\iota(r(T_1)) = r(T_1) = r(T')$ . Therefore  $\iota$  defines a tree morphism from  $T_1$  to  $T'$ .

(3) We prove  $\sigma$  is a tree morphism.

$\sigma$  is a function with domain  $V(T')$  and codomain  $V(T_2)$ . Let  $v_1$  and  $v_2$  be adjacent vertices of  $T'$ .

If both  $v_1$  and  $v_2$  are in  $V(T_1)$ , since  $\varphi$  defines a tree morphism from  $T_1$  to  $T_2$ , both  $\sigma(v_1) = \varphi(v_1)$  and  $\sigma(v_2) = \varphi(v_2)$  are adjacent in  $T_2$ .

If both  $v_1$  and  $v_2$  are in  $V(T_2) \setminus \text{img}(\varphi)$ ,  $\sigma(v_1) = v_1$  and  $\sigma(v_2) = v_2$  are adjacent in  $T_2$ , according to the definition of edge of  $T'$ .



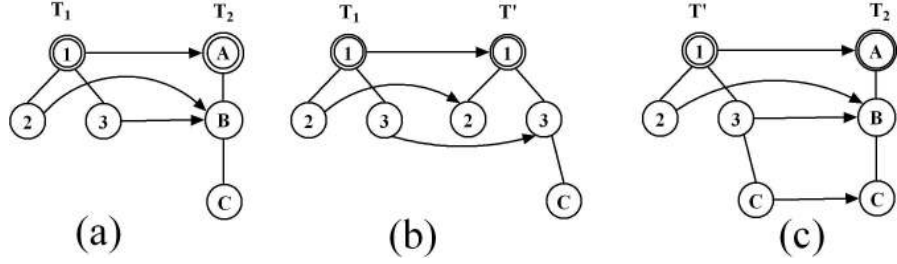


Figure A.1 An illustration of Lemma 1

Otherwise, suppose  $v_1 \in V(T_1)$  and  $v_2 \in V(T_2) \setminus \text{img}(\varphi)$ . According to the definition of edge of  $T'$ ,  $\sigma(v_1) = \varphi(v_1)$  and  $\sigma(v_2) = v_2$  are adjacent in  $T_2$ .

In all,  $\sigma(v_1)$  and  $\sigma(v_2)$  are adjacent in  $T_2$  whenever  $v_1$  and  $v_2$  are adjacent in  $T'$ . In addition,  $\sigma(r(T')) = \sigma(r(T_1)) = \varphi(r(T_1)) = r(T_2)$ . Therefore,  $\sigma$  defines a tree morphism from  $T'$  to  $T_2$ .

**Finally**, we have  $T'$ ,  $\sigma$  and  $\iota$  that are the tree and morphisms satisfying the conditions.

□

As an example, Figure A.1(a) shows a morphism between a pair of trees. The constructed morphisms  $\iota : T_1 \rightarrow T'$  and  $\sigma : T' \rightarrow T_2$  are shown in Figures A.1(a) and A.1(c) respectively.

Using lemma 1, the following corollaries are proved:

**Corollary 1.** *Any complex change  $C$  composed of a basic delete from  $T_0$  to  $T_1$  followed by a basic split from  $T_1$  to  $T_2$  is equivalent to a complex change  $C'$  composed of a basic split from  $T_0$  to  $T'_1$  followed by a basic delete from  $T'_1$  to  $T_2$ .*

*Proof.* Let  $C$  be specified by an injective morphism  $\iota_1$  from  $T_1$  to  $T_0$  and a surjective morphism  $\sigma_1$  from  $T_2$  to  $T_1$ . Let  $\varphi = \sigma_1 \circ \iota_1$ .

Using Lemma 1, we are able to find another tree  $T'_1$ , a surjective tree morphism  $\sigma$  from  $T'_1$  to  $T_0$ , and an injective tree morphism  $\iota$  from  $T_2$  to  $T'_1$ , satisfying properties as

stated in the Lemma. The  $\sigma$  and  $\iota$  specify a complex change  $C'$  from  $T_0$  to  $T_2$  composed of a basic split followed by a basic delete.

The property (1) ensures that the future of any vertex of  $T_0$  is the same in  $T_2$  either through  $C$  or through  $C'$ . In addition, the essential insertion sets of both  $C$  and  $C'$  are empty. Therefore,  $C$  is an equivalent change of  $C'$ .  $\square$

**Corollary 2.** *Any complex change  $C$  composed of a basic merge from  $T_0$  to  $T_1$  followed by a basic insert from  $T_1$  to  $T_2$  is equivalent to a complex change  $C'$  composed of a basic insert from  $T_0$  to  $T'_1$  followed by a basic merge from  $T'_1$  to  $T_2$ .*

*Proof.* Let  $C$  be specified by a surjective morphism  $\sigma_1$  from  $T_0$  to  $T_1$  and an injective morphism  $\iota_1$  from  $T_1$  to  $T_2$ . Let  $\varphi = \sigma_1 \circ \iota_1$ .

Using Lemma 1, we are able to find another tree  $T'_1$ , an injective tree morphism  $\iota$  from  $T_0$  to  $T'_1$ , and a surjective morphism  $\sigma$  from  $T'_1$  to  $T_2$ , satisfying properties as stated in the Lemma. The  $\iota$  and  $\sigma$  specify a complex change  $C'$  from  $T_0$  to  $T_2$  composed of a basic insert followed by a basic merge.

The property (1) ensures that the future of any vertex of  $T_0$  is the same in  $T_2$  either through  $C$  or through  $C'$ .

By definition, the essential insertion set of  $C$  is  $I(C) = \{(v, 2) | v \in (V(T_2) \setminus \text{img}(\varphi))\}$ , and the essential insertion set of  $C'$  is  $I(C') = \{(v, 1) | v \in (V(T'_1) \setminus \text{img}(\iota))\}$ . For  $\forall (v, 1) \in I(C')$ , we define  $f(v, 1) = (\sigma(v), 2)$ . According to the property (2),  $f$  is a bijection between  $I(C)$  and  $I(C')$ , such that the corresponding essential insertions have the same future.

Therefore,  $C$  is an equivalent change of  $C'$ .

$\square$

**Lemma 2.** Let  $T_1, T_2$  and  $T_3$  be trees,  $\iota$  be an injective tree morphism from  $T_2$  to  $T_1$ , and  $\varphi$  be a tree morphism from  $T_2$  to  $T_3$ . Then, it is possible to find a tree  $T'$ , an injective tree morphism  $\iota'$  from  $T_3$  to  $T'$ , and a tree morphism  $\varphi'$  from  $T_1$  to  $T'$ , satisfying:

$$(1) (\iota^{-1} \circ \varphi) = (\varphi' \circ \iota'^{-1}).$$

(2)  $\varphi'$  is surjective whenever  $\varphi$  is surjective, and  $\varphi'$  is injective whenever  $\varphi$  is injective.

(3) Let  $S_1 = V(T_1) \setminus \text{img}(\iota)$  and  $S_2 = V(T') \setminus \text{img}(\iota')$ . Then  $\varphi'$  defines a bijection between  $S_1$  and  $S_2$ , by restricting the domain of  $\varphi'$  to  $S_1$ .

(4) Let  $S_3 = V(T_3) \setminus \text{img}(\varphi)$  and  $S_4 = V(T') \setminus \text{img}(\varphi')$ . Then  $\iota'$  defines a bijection between  $S_3$  and  $S_4$ , by restricting the domain of  $\iota'$  to  $S_3$ .

$$\begin{array}{ccc} T_1 & \xleftarrow{\iota} & T_2 \\ \varphi' \downarrow & & \downarrow \varphi \\ T' & \xleftarrow{\iota'} & T_3 \end{array}$$

*Proof.* We prove this lemma by providing an approach to construct  $T'$ ,  $\varphi'$  and  $\iota'$ .

**First**, we define vertices of  $T'$ , and functions  $\iota'$  and  $\varphi'$  as follows:

1.  $V(T')$  is defined to be  $V(T_3) \cup (V(T_1) \setminus \text{img}(\iota))$ .
2.  $\varphi'$  is a function with domain  $V(T_1)$  and codomain  $V(T')$ .  $\forall x \in V(T_1)$ ,  $\varphi'(x) = \varphi(\iota^{-1}(x))$  if  $x \in \text{img}(\iota)$ , and  $\varphi'(x) = x$  if  $x \in V(T_1) \setminus \text{img}(\iota)$ .
3.  $\iota'$  is a function with domain  $V(T_3)$  and codomain  $V(T')$ ,  $\forall x \in V(T_3)$ ,  $\iota'(x) = x$ .

It can be verified, as follows, that  $V(T')$ ,  $\varphi'$ , and  $\iota'$  satisfy conditions (1),(2),(3) and (4) described in the lemma.

- (1) Let  $v_1$  and  $v_3$  be vertices of  $T_1$  and  $T_3$  respectively. If  $\varphi'(v_1) = \iota'(v_3)$ , it follows that  $\varphi'(v_1) = v_3 \in V(T_3)$ , hence  $v_1$  must belong to  $\text{img}(\iota)$ . Let  $v_2 = \iota^{-1}(v_1)$ ,  $v_2$  satisfies that  $\iota(v_2) = v_1$  and  $\varphi(v_2) = \varphi(\iota^{-1}(v_1)) = \varphi'(v_1) = v_3$ .

Conversely, if  $\exists v_2 \in V(T_2)$ , such that  $\iota(v_2) = v_1$  and  $\varphi(v_2) = v_3$ , according to the definitions of  $\iota'$  and  $\varphi'$ ,  $\varphi'(v_1) = \varphi(\iota^{-1}(v_1)) = \varphi(v_2) = v_3$ .

In all,  $\varphi'(v_1) = \iota'(v_3)$  if and only if  $\exists v_2 \in V(T_2)$ , such that  $\iota(v_2) = v_1$  and  $\varphi(v_2) = v_3$ .

- (2) Let  $\varphi$  be a surjective function from  $T_2$  to  $T_3$ . Consider any vertex  $v_3$  of  $T_3$ . Since  $\varphi$  is surjective, there must be a vertex  $v_2$  of  $T_2$  such that  $\varphi(v_2) = v_3$ . Let  $v_1 = \iota(v_2)$ , then  $v_1$  is a vertex of  $T_1$ , and according to the definition of  $\varphi'$ ,  $\varphi'(v_1) = \varphi(\iota^{-1}(v_1)) = \varphi(v_2) = v_3$ . Moreover,  $\forall v \in V(T_1) \setminus \text{img}(\iota)$ ,  $\varphi'(v) = v$ . It follows that for  $\forall v \in V(T') = V(T_3) \cup (V(T_1) \setminus \text{img}(\iota))$ ,  $\exists v_1 \in V(T_1)$  such that  $\varphi'(v_1) = v$ . Hence  $\varphi'$  is a surjective function.

Let  $\varphi$  be an injective function from  $T_2$  to  $T_3$ , then  $\iota^{-1} \circ \varphi$  is an injective function with domain  $\text{img}(\iota)$ . In addition, by restricting the domain of  $\varphi'$  to  $V(T_1) \setminus \text{img}(\iota)$ ,  $\varphi'$  is a inclusion mapping with disjoint range to  $\iota^{-1} \circ \varphi$ . According to the definition of  $\varphi'$ , it follows that  $\varphi'$  is injective.

- (3)  $S_2 = V(T') \setminus \text{img}(\iota') = V(T') \setminus V(T_3) = V(T_1) \setminus \text{img}(\iota) = S_1$ . According to the definition of  $\varphi'$ , by restricting the domain of  $\varphi'$  to  $S_1$ ,  $\varphi'$  is an inclusion mapping that defines a bijection between  $S_1$  and  $S_2$ .

- (4) According to the definition of  $\varphi'$ ,  $\text{img}(\varphi') = \text{img}(\varphi) \cup (V(T_1) \setminus \text{img}(\iota))$ . Therefore,  $S_4 = V(T') \setminus \text{img}(\varphi') = V(T_3) \setminus \text{img}(\varphi) = S_3$ . In addition,  $\iota'$  is an inclusion mapping, it follows that  $\iota'$  defines a bijection between  $S_3$  and  $S_4$ , by restricting the domain of  $\iota'$  to  $S_3$ .

**Second**, we define root and edges of  $T'$ . The root of  $T'$  is defined to be  $r(T_3)$ .

Let  $v_1$  and  $v_2$  be a pair of vertices of  $T'$ ,  $v_1$  and  $v_2$  are adjacent in  $T'$  if and only if one of the following conditions are satisfied:

1. both  $v_1$  and  $v_2$  belong to  $V(T_3)$ , and  $v_1$  and  $v_2$  are adjacent in  $T_3$ .
2. both  $v_1$  and  $v_2$  belong to  $V(T_1) \setminus \text{img}(\iota)$ , and  $v_1$  and  $v_2$  are adjacent in  $T_1$ .
3.  $v_1 \in (V(T_1) \setminus \text{img}(\iota))$  and  $v_2 \in V(T_3)$ , and  $\exists v \in V(T_2)$  such that  $\iota(v)$  and  $v_1$  are adjacent in  $T_1$ , and  $\varphi(v) = v_2$ .

Based on the definition of edges of  $T'$ , it can be proved that (1)  $T'$  is a tree, and (2)  $\varphi'$  and  $\iota'$  are tree morphisms.

- (1) We prove  $T'$  is a tree by showing that  $T'$  is both connected and cycle-free.

We first prove that  $T'$  is connected. Let  $v$  be a vertex of  $T'$ .

Consider  $v \in V(T_3)$ . Let  $P$  be a path connecting  $v$  with  $r(T_3)$  in  $T_3$ . According to the definition of  $T'$ ,  $P$  is also a path connecting  $v$  with  $r(T_3)$  in  $T'$ .

Consider  $v \in (V(T_1) \setminus \text{img}(\iota))$ , let  $[v, v_1, \dots, v_{n-1}, r(T_1)]$  be a path connecting  $v$  with  $r(T_1)$  in  $T_1$ . According to the definition of  $\varphi'$  and of  $T'$ ,  $[v, \varphi'(v_1), \dots, \varphi'(v_{n-1}), r(T_3)]$  is a path in  $T'$  connecting  $v$  and  $r(T_3)$ .

Any vertex in  $V(T') = V(T_3) \cup (V(T_1) \setminus \text{img}(\iota))$  is connected to  $r(T_3)$  of  $T'$ . Therefore  $T'$  is connected.

We prove  $T'$  is cycle-free by contradiction. Assume to the contrary that  $T'$  contains a simple cycle  $C$ . If all the vertices in  $C$  belong to  $V(T_3)$ ,  $C$  is a simple cycle of  $T_3$ , contradicting to the fact that  $T_3$  is a tree.

If all the vertices in  $C$  belong to  $V(T_1) \setminus \text{img}(\iota)$ ,  $C$  is a simple cycle of  $T_1$ , contradicting to the fact that  $T_1$  is a tree.

Otherwise,  $C$  contains vertices from both  $V(T_3)$  and  $V(T_1) \setminus \text{img}(\iota)$ . Then there must be a simple path or simple cycle in  $C$  with the form

$$[s_1, v_1, v_2, \dots, v_n, s_2].$$

$C$  satisfies that  $n > 0$ ,  $s_1$  and  $s_2$  belong to  $V(T_3)$ ,  $s_1 = s_2$  if it is a cycle, and  $\forall i, 1 \leq i \leq n, v_i \in (V(T_1) \setminus \text{img}(\iota))$ . Hence according to the definition of edges of  $T'$ , we are able to find  $u_1 \in \varphi^{-1}(s_1)$ , and  $u_2 \in \varphi^{-1}(s_2)$ , such that  $[\iota(u_1), v_1, v_2, \dots, v_n, \iota(u_2)]$  is a simple path or a simple cycle in  $T_1$  containing vertices in  $V(T_1) \setminus \text{img}(\iota)$ . In case it is a simple path, as  $\iota(u_1)$  and  $\iota(u_2)$  are also connected by a simple path in  $T_1$  that is completely contained in  $\text{img}(\iota)$ , which infers a simple cycle of  $T_1$ , both contradict to the fact that  $T_1$  is a tree.

All the possible cases lead to contradiction, therefore  $T'$  is cycle-free, and  $T'$  is a tree.

- (2)  $\varphi'$  and  $\iota'$  are tree morphisms is a direct consequence of the definition of the edges of  $T'$ .

**Finally**, we have  $T'$ ,  $\varphi'$  and  $\iota'$  that are the tree and morphisms satisfying the conditions.

□

As an example, Figure A.2(a) shows three trees with an injective and a surjective morphisms between them. The constructed tree  $T'$  and morphisms  $\varphi' : T_1 \rightarrow T'$  and  $\iota' : T_3 \rightarrow T'$  are shown in Figures A.2(a) and A.2(c) respectively.

Using lemma 2, the following corollaries are proved:

**Corollary 3.** *Any complex change  $C$  composed of a basic delete from  $T_0$  to  $T_1$  followed by a basic merge from  $T_1$  to  $T_2$  is equivalent to a complex change  $C'$  composed of a basic merge from  $T_0$  to  $T'_1$  followed by a basic delete from  $T'_1$  to  $T_2$ .*

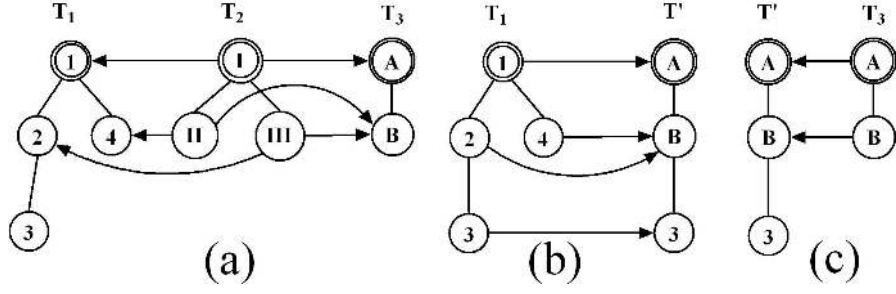


Figure A.2 An illustration of Lemma 2

*Proof.* Let  $C$  be specified by an injective morphism  $\iota$  from  $T_1$  to  $T_0$  and a surjective morphism  $\sigma$  from  $T_1$  to  $T_2$ .

Using Lemma 2, we are able to find another tree  $T'_1$ , a surjective tree morphism  $\varphi'$  from  $T_0$  to  $T'_1$ , and an injective tree morphism  $\iota'$  from  $T_2$  to  $T'_1$ , satisfying the properties as stated in the Lemma. The  $\varphi'$  and  $\iota'$  specify a complex change  $C'$  from  $T_0$  to  $T_2$  composed of a basic merge followed by a basic delete.

The property (1) ensures that the future of any vertex of  $T_0$  is the same in  $T_2$  either through  $C$  or through  $C'$ . In addition, the essential insertion sets of both  $C$  and  $C'$  are empty. Therefore,  $C$  is an equivalent change of  $C'$ .  $\square$

**Corollary 4.** *Any complex change  $C$  composed of a basic split from  $T_0$  to  $T_1$  followed by a basic insert from  $T_1$  to  $T_2$  is equivalent to a complex change  $C'$  composed of a basic insert from  $T_0$  to  $T'_1$  followed by a basic split from  $T'_1$  to  $T_2$ .*

*Proof.* Let  $C$  be specified by a surjective morphism  $\sigma$  from  $T_1$  to  $T_0$  and an injective morphism  $\iota$  from  $T_1$  to  $T_2$ .

Using Lemma 2, we are able to find another tree  $T'_1$ , an injective tree morphism  $\iota'$  from  $T_0$  to  $T'_1$ , and a surjective tree morphism  $\varphi'$  from  $T_2$  to  $T'_1$ , satisfying the properties as stated in the Lemma. The  $\iota'$  and  $\varphi'$  specify a complex change  $C'$  from  $T_0$  to  $T_2$  composed of a basic insert followed by a basic split.

The property (1) ensures that the future of any vertex of  $T_0$  is the same in  $T_2$  either through  $C$  or through  $C'$ .

By definition, the essential insertion set of  $C$  is  $I(C) = \{(v, 2) | v \in (V(T_2) \setminus \text{img}(\iota))\}$ , and the essential insertion set of  $C'$  is  $I(C') = \{(v, 1) | v \in (V(T'_1) \setminus \text{img}(\iota'))\}$ . For  $\forall (v, 2) \in I(C)$ , we define  $f(v, 2) = (\sigma'(v), 1)$ . According to the property (3),  $f$  is a bijection between  $I(C)$  and  $I(C')$ , such that the corresponding insertions have the same future.

Therefore,  $C$  is an equivalent change of  $C'$ . □

**Corollary 5.** *Any complex change  $C$  composed of a basic delete from  $T_0$  to  $T_1$  followed by a basic insert from  $T_1$  to  $T_2$  is equivalent to a complex change  $C'$  composed of a basic insert from  $T_0$  to  $T'_1$  followed by a basic delete from  $T'_1$  to  $T_2$ .*

*Proof.* The proof is similar to corollary 4 except that  $\varphi'$  is an injective morphism, and we need property (4) stated in the Lemma in order to prove the equivalence relations. □

**Lemma 3.** *Let  $T_1, T_2$  and  $T_3$  be trees,  $\sigma_1$  be a surjective morphism from  $T_1$  to  $T_2$ , and  $\sigma_2$  be a surjective morphism from  $T_3$  to  $T_2$ . Then, it is possible to find two trees  $T'_4$  and  $T'_5$ , a surjective morphism  $\sigma'_1$  from  $T'_4$  to  $T_1$ , a surjective morphism  $\sigma'_2$  from  $T'_4$  to  $T'_5$ , and an injective morphism  $\iota'$  from  $T_3$  to  $T'_5$ , satisfying:  $(\sigma_1 \circ \sigma_2^{-1}) = (\sigma'_1{}^{-1} \circ \sigma'_2 \circ \iota'^{-1})$*

$$\begin{array}{ccccc}
 T_1 & \xrightarrow{\sigma_1} & T_2 & \xleftarrow{\sigma_2} & T_3 \\
 & \swarrow \sigma'_1 & & & \downarrow \iota' \\
 & & T'_4 & \xrightarrow{\sigma'_2} & T'_5
 \end{array}$$

*Proof.* We prove this lemma by providing an approach to construct  $T'$ ,  $\varphi'$  and  $\iota'$ .

**First**, we define  $T'_4$  and  $\sigma'_1$  based on a graph  $G$ . The graph  $G$  is defined as follows:

1.  $V(G) = \{(u, v) : u \in V(T_1), v \in V(T_3), \sigma_1(u) = \sigma_2(v)\}$ .
2. For  $\forall (u_1, v_1), (u_2, v_2) \in V(G)$ ,  $(u_1, v_1)$  and  $(u_2, v_2)$  are adjacent in  $G$  if and only if  $u_1$  and  $u_2$  are adjacent in  $T_1$ , and  $v_1$  and  $v_2$  are adjacent in  $T_3$ .



$G$  is not always connected, we need to add additional vertices and edges to  $G$  that “glue” its disconnected components together. These additional vertices and edges are in the form of gluing path defined as follows:

**Definition A.2.** A gluing path has the form

$$[(u_1, v_1), u_2, u_3, \dots, u_{k-1}, (u_k, v_k)],$$

in which  $(u_1, v_1)$  and  $(u_k, v_k)$  are adjacent vertices of  $G$ , and  $\forall i, 1 \leq i \leq k - 1$ , both  $u_i$  and  $u_{i+1}$  are adjacent vertices of  $T_1$ .

In the gluing path,  $(u_1, v_1)$  and  $(u_k, v_k)$  are said to be its  $G$ -vertices, and  $\forall i, 2 \leq i \leq k - 1$ ,  $u_i$  is said to be its  $T_1$ -vertex.

It is trivial to prove that if  $G$  is an disconnected graph with  $n$  components, it is possible to connect the  $n$  components by adding  $n - 1$  gluing paths  $P_0, P_1, \dots, P_{n-1}$ . The union of  $G$  and the  $n$  gluing path is a connected graph  $G'$ . A vertex  $v$  of  $T_1$  may be  $T_1$ -vertices of different gluing paths. We use superscript to differentiate the same vertex  $u$  in different gluing path.  $u^i$  represents a  $T_1$ -vertex  $u$  in gluing path  $P_i$  being the same as vertex  $u$  of  $T_1$ .

$T'_4$  and  $\sigma'_1$  are defined as follows:

1.  $T'_4$  is defined to be a spanning tree of  $G'$  with root  $(r(T_1), r(T_3))$ ; that is  $V(T'_4) = V(G') = V(G) \cup X$ , in which  $X$  is a set containing the  $T_1$ -vertices of the  $n$  gluing paths in  $G'$ .  $E(T')$  is a subset of  $E(G')$  which ensure that  $T'_4$  is a tree.
2.  $\sigma'_1$  is a function with domain  $V(T'_4)$  and codomain  $V(T_1)$ .  $\forall (u_1, v_1) \in V(G) \subset (T')$ ,  $\sigma'_1((u_1, v_1)) = u_1$ , and  $\forall u^i \in X \subset (T')$ ,  $\sigma'_1(u^i) = u$ . It is trivial to infer that  $\sigma'_1$  defines a surjective morphism from  $T'_4$  to  $T_1$ .

**Second** We define tree  $T'_5$  and functions  $\sigma'_2$  and  $\iota$ .

$T'_5$  is an extension of  $T_3$ . It contains  $T_3$  as a subtree, and it also contains a set of vertices  $Y$  as well as additional edges.  $T'_5$  is defined as follows:

1.  $V(T'_5) = V(T_3) \cup Y$ .  $Y = \{\sigma_1(u)^i : u \in X\} \setminus \{\sigma_1(u)^i : \exists v, (u, v) \in V(P_i)\}$ .
2. The edges of  $T'_5$  are defined as follows: let  $v_1$  and  $v_2$  be vertices of  $T'_5$ .  $v_1$  and  $v_2$  are adjacent if and only if one of the following conditions are satisfied:
  - (a)  $v_1, v_2 \in V(T_3)$ , and  $v_1$  and  $v_2$  are adjacent in  $T_3$ .
  - (b)  $v_1, v_2 \in Y$ , and there is a gluing path  $P_i$  with  $T_1$ -vertices  $u_1^i$  and  $u_2^i$  such that  $\sigma_1(u_1)$  and  $\sigma_1(u_2)$  are adjacent in  $T_2$ ,  $\sigma_1(u_1)^i = v_1$ , and  $\sigma_2(u_2)^i = v_2$ .
  - (c)  $v_1 \in V(T_3)$ ,  $v_2 \in Y \subset T'_5$ , and there is a path  $P_i$  with a  $G$ -vertex  $(u_1, v_1)$  and a  $T_1$ -vertices  $u^i$  such that  $\sigma_1(u_1)$  and  $\sigma_1(u)$  are adjacent in  $T_2$ , and  $\sigma_1(u)^i = v_2$ .

We define  $r(T_3)$  to be the root to  $T'_5$ . It can be proved that  $T'_5$  is connected and cycle-free, and hence is a tree.

We first prove that each vertex of  $T'_5$  is connected to  $r(T_3)$ .

$T'_5$  contains  $T_3$  as a subtree, hence for any vertex  $v \in V(T_3) \subset V(T'_5)$ ,  $v$  is connected to  $r(T_3)$  in  $T'_5$ . For any vertex  $v^i \in Y \subset V(T'_5)$ , let  $u^i$  be a vertex on the gluing path  $P_i$  such that  $\sigma_1(u)^i = v^i$ . Then we are able to get a connected part of  $P_i$  with the form  $[u_0^i = u^i, u_1^i, \dots, u_{k-1}^i, (u_k, v_k)]$  connecting  $u^i$  to one of the  $G$ -vertex of path  $P_i$ . According to the definition of edges of  $T'_5$ , there must be a path in  $T'_5$  connecting  $\sigma_1(u_0)^i = v'$  to  $v_k \in V(T_3)$ , which is already proved to be connected to  $r(T_3)$  in  $V'_5$ .

Each vertex of  $T'_5$  is connected to the  $r(T_3)$ , therefore  $T'_5$  is connected.

We prove that  $T'_5$  is cycle-free by contradiction. Assume to the contrary that  $T'$  contains a simple cycle  $C$ .

If all vertices of  $C$  are in  $V(T_3) \subset T'_5$ , it follows that there is a cycle in  $T_3$ , contradicting to the fact that  $T_3$  is a tree.

If all vertices of  $C$  are in  $Y \subset T'_5$ , according to the definition of the edges of  $T'_5$ , any vertex in  $Y$  can only be adjacent to the vertex in  $Y$  of the same superscript or to a vertex in  $V(T_3)$ . It follows that all vertices in  $C$  must be the images of  $T_1$ -vertices of the same gluing path  $P_i$  through  $\sigma_1$ . Therefore  $C$  have the form  $[u_0^i, u_1^i, \dots, u_k^i = u_0^i]$ . According to the definition of edges of  $T'_5$ ,  $[u_0, u_1, \dots, u_k = u_0]$  is a simple cycle of  $T_2$ , contradicting to the fact that  $T_2$  is a tree.

Otherwise,  $C$  contains vertices from both  $Y \subset T'_5$  and  $V(T_3) \subset T'_5$ . In this case we are able to find a simple path on  $C$  with the form

$$[u_1, v_2^i, v_3^i, \dots, v_k^i, u_{k+1}]$$

such that  $k > 1$ ,  $u_1, u_2 \in V(T_3)$ , and  $\forall m, 1 < m \leq k, v_m^i \in Y$ . Then according to the definitions of gluing path and edges of  $T'_5$ ,  $[\sigma_2(u_1), v_2, v_2, \dots, v_k, \sigma_2(u_{k+1})]$  is either a simple path of  $T_2$  connecting adjacent vertices  $\sigma_2(u_1)$  and  $\sigma_2(u_{k+1})$ , or a simple cycle including  $\sigma_2(u_1) = \sigma_2(u_{k+1})$ . Either lead to the contradiction that  $T_2$  is a tree.

All the possible cases lead to contradiction, therefore  $T'$  is cycle-free.

**Third**, we define function  $\iota$  and  $\sigma'_2$  as follows:

1.  $\iota$  is a function with domain  $V(T_3)$  and codomain  $V(T'_5)$ . For any vertex  $v \in V(T_3)$ ,  $\iota(v) = v$ .
2.  $\sigma'_2$  is a function with domain  $V(T'_4)$  and codomain  $V(T'_5)$ . For any vertex  $(u, v) \in V(G) \subset V(T'_4)$ ,  $\sigma'_2((u, v)) = v$ . For any vertex  $u^i \in X \subset V(T'_4)$ , let  $P_i = [(u_1, v_1), u_2^i, u_3^i, \dots, u_k^i, (u_{k+1}, v_{k+1})]$  containing  $u_i$ .  $\sigma'_2(u^i) = v_1$  if  $\sigma_1(u^i) = \sigma_1(u_1)$ ;  $\sigma'_2(u^i) = v_2$  if  $\sigma_1(u^i) = \sigma_1(u_2)$ ;  $\sigma'_2(u^i) = \sigma_1(u)^i$ , if otherwise.

It can be verified that  $\iota$  defines an injective tree morphism from  $T_3$  to  $T'_5$ , and  $\sigma'_2$  defines a surjective tree morphism from  $T'_4$  to  $T'_5$ .

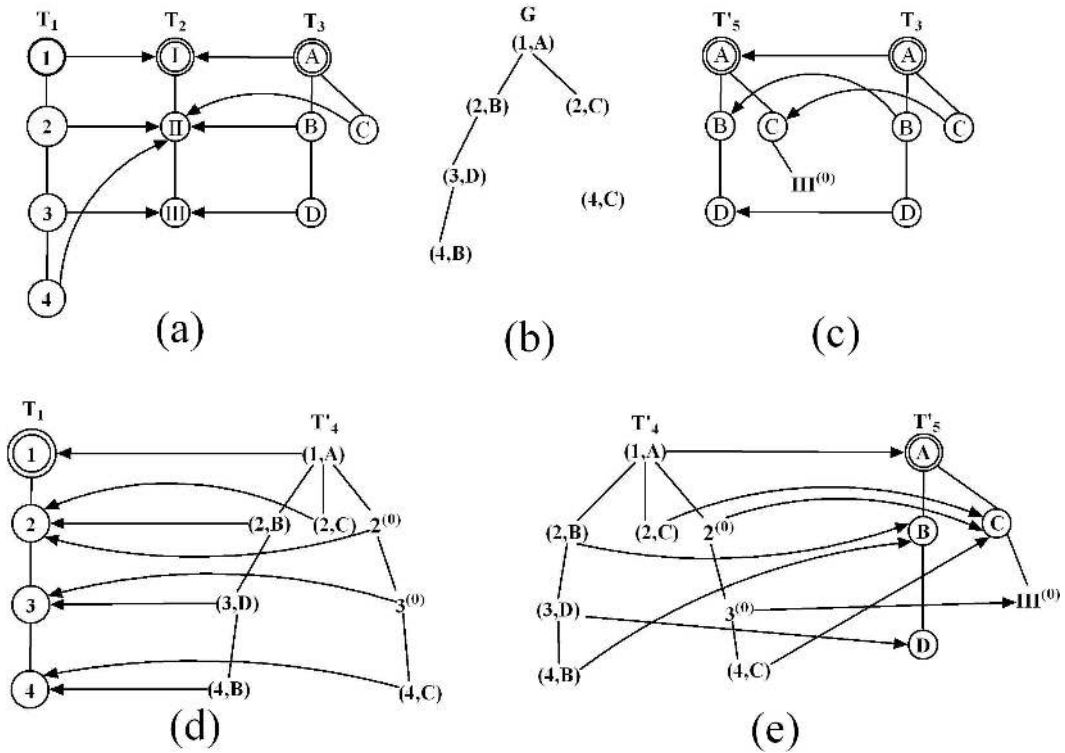


Figure A.3 An illustration of Lemma 3

Finally, we got  $T'_4$ ,  $T'_5$ ,  $\sigma'_1$ ,  $\sigma'_2$  and  $\iota$  that are the tree and morphisms satisfying the conditions.

□

We give an example of these trees and morphisms. Figure A.3(a) shows an example of trees  $T_1$ ,  $T_2$  and  $T_3$ , such that there are surjective morphisms from  $T_3$  and  $T_1$  to  $T_2$  respectively. According to the correspondence between  $T_1$  and  $T_3$ , we are able to construct a graph  $G$  shown in Figure A.3(a). By extending  $G$ , we get the tree  $T'_4$ . By extending  $T_3$ , we get the tree  $T'_5$ . The morphisms  $\iota' : T_3 \rightarrow T'_5$ ,  $\sigma'_1 : T'_4 \rightarrow T_1$ , and  $\sigma'_2 : T'_4 \rightarrow T'_5$  are shown in Figures A.3(c), A.3(d) and A.3(e) respectively.

A direct consequence of Lemma 3 is the corollary stated as follows:

**Corollary 6.** *Any complex change  $C$  composed of a basic merge from  $T_0$  to  $T_1$  followed by a basic split from  $T_1$  to  $T_2$  is equivalent to a complex change  $C'$  composed of a basic*

*split from  $T_0$  to  $T_1'$ , followed by a basic merge from  $T_1'$  to  $T_2'$ , and followed by a basic delete from  $T_2'$  to  $T_2$ .*

## BIBLIOGRAPHY

- [AHF05] M. Al-Hasan and M. Fisher. Image registration using a weighted region adjacency graph. In M.J. Flynn, editor, *Medical Imaging 2005: Proceedings of the SPIE Physics of Medical Imaging*, volume 5745, pages 1038–1045, 2005.
- [ALE08] ALERT. Automated local evaluation in real time. <http://www.alertsystems.org/>, Last accessed July, 2008.
- [Ape04] M. Apel. *A 3D geoscience information system framework*. PhD thesis, Institut National Polytechnique de Lorraine, Nancy, France, 2004.
- [Arm88] M.P. Armstrong. Temporality in spatial databases. In *Proceedings of GIS/LIS 88: Accessing the World*, pages 880–889, San Antonio, TX, USA, 1988.
- [BD07] T. Bittner and M. Donnelly. Logical properties of foundational relations in bio-ontologies. *Artificial Intelligence in Medicine*, pages 197–216, 2007.
- [BFG04] J. Bouttier, P. Di Francesco, and E. Guitter. Planar maps as labeled mobiles. *The Electronic Journal of Combinatorics*, 11:1–27, 2004.
- [BMPW00] A. Baptista, D. McNamee, C. Pu, and J. Walpole. Research challenges in environmental observation and forecasting systems. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, pages 292–299, Boston, MA, USA, 2000.

- [CBGG97] A.G. Cohn, B. Bennett, J. Gooday, and N.M. Gotts. Qualitative spatial representation and reasoning with the region connection calculus. *GeoInformatica*, 1(2):275–316, 1997.
- [CDHH06] D. Chu, A. Deshpande, J. M. Hellerstein, and W. Hong. Approximate data collection in sensor networks using probabilistic models. In *Proceedings of the 22nd International Conference on Data Engineering (ICDE'06)*, page 48, 2006.
- [CE96] M. Coan and M. Egenhofer. The ontology of land boundaries under natural change: erosion and accretion of sandy shoals in nantucket sound. In *Proceedings of the ACSM/ASPRS Conference*, Baltimore, MD, 1996.
- [CG96] A.G. Cohn and N.M. Gotts. The ‘egg-yolk’ representation of regions with indeterminate boundaries. In *Proceedings of GISDATA Specialist Meeting on Geographical Objects with Undetermined Boundaries*, pages 171–187, 1996.
- [CG03] K. Chintalapudi and R. Govindan. Localized edge detection in sensor fields. *Ad Hoc Networks*, 1(2-3):273–291, 2003.
- [Chi70] R. Chisholm. Events and propositions. *Noûs*, 4(1):15–24, 1970.
- [CK03] C.Y. Chong and S.P. Kumar. Sensor networks: Evolution, opportunities and challenges. *Proceedings of the IEEE*, 91(8):1247–1256, 2003.
- [Cla95] C. Claramunt. Managing time in GIS: An event-oriented approach. In *Proceedings of the International Workshop on Temporal Databases: Recent Advances in Temporal Databases*, pages 23–42, Zurich, Switzerland, 1995.
- [COR08] CORIE. Environmental observation and forecasting system for the columbia river. <http://www.ccalmr.ogi.edu/CORIE/>, Last accessed July, 2008.

- [CR03] E. Costanza and J. Robinson. A region adjacency tree approach to the detection and design of fiducials. In *Proceedings of Vision, Video and Graphics*, pages 63–69, 2003.
- [CR07] A.G. Cohn and J. Renz. Qualitative spatial reasoning. In F. Harmelen, V. Lifschitz, and B. Porter, editors, *Handbook of Knowledge Representation*. Elsevier, 2007.
- [CSA00] H. Carr, J. Snoeyink, and U. Axen. Computing contour trees in all dimensions. In *Symposium on Discrete Algorithms*, pages 918–926, 2000.
- [CTLW05] M. Cardei, M.T. Thai, Y. Li, and W. Wu. Energy-efficient target coverage in wireless sensor networks. In *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'05)*, pages 1976–1984, 2005.
- [Dav80] D. Davidson. The individuation of events. In *Essays on Actions and Events*, pages 163–180. Oxford University Press, 1980.
- [DBN04] B. Deb, S. Bhatnagar, and B. Nath. STREAM: Sensor topology retrieval at multiple resolutions. *Telecommunication Systems*, 26(2-4):285–320, 2004.
- [DNW05] M. Duckham, S. Nittel, and M. Worboys. Monitoring dynamic spatial fields using responsive geosensor networks. In *ACM-GIS 2005*, pages 51–60, Bremen, Germany, 2005.
- [EAT92] M. Egenhofer and K. Al-Taha. Reasoning about gradual changes of topological relationships. In *Proceedings of the International Conference GIS - From Space to Territory: Theories and Methods of Spatio Temporal Reasoning, LNCS 639*, pages 196–219, 1992.



- [ECF94] M. Egenhofer, E. Clementini, and P.D. Felice. Topological relations between regions with holes. *International Journal of Geographical Information Systems*, 8(2):129–144, 1994.
- [EF91] M. Egenhofer and R. Franzosa. Point-set topological spatial relations. *International Journal of Geographical Information Systems*, 5(2):161–174, 1991.
- [EF95] M. Egenhofer and R. Franzosa. On the equivalence of topological relations. *International Journal of Geographical Information Systems*, 9(2):133–152, 1995.
- [Ege05] M. Egenhofer. Spherical topological relations. *Journal on Data Semantics III*, pages 25–49, 2005.
- [EGHK99] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. Next century challenges: Scalable coordination in sensor networks. In *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM'99)*, pages 263–270, Seattle, Washington, USA, 1999.
- [EH90] M. Egenhofer and J. Herring. Categorizing binary topological relations between regions, lines, and points in geographic databases. Technical report, Department of Surveying Engineering, The University of Maine, 1990.
- [EM95] M. Egenhofer and D. Mark. Modeling conceptual neighborhoods of topological line-region relations. *International Journal of Geographical Information Systems*, 9(5):555–565, 1995.
- [EV07] M. Egenhofer and M. Vasardani. Spatial reasoning with a hole. In S. Winter, M. Duckham, L. Kulik, and B. Kuipers, editors, *Proceedings of COSIT '07, LNCS 4736*, pages 303–320, 2007.

- [FGG06] Q. Fang, J. Gao, and L. Guibas. Locating and bypassing routing holes in sensor networks. *Mobile Networks and Applications*, 11(2):187–200, 2006.
- [FK06] S. Funke and C. Klein. Hole detection or: ‘how much geometry hides in connectivity?’. In *Proceedings of the Twenty-second Annual Symposium on Computational Geometry*, pages 377 – 385, 2006.
- [FKP<sup>+</sup>04] S.P. Fekete, A. Kröller, D. Pfisterer, S. Fischer, and C. Buschmann. Neighborhood-based topology recognition in sensor networks. In *Algorithmic Aspects of Wireless Sensor Networks, LNCS 3121*, pages 123–136, 2004.
- [FTGL04] B. Fischer, C.J. Thies, M.O. Guld, and T.M. Lehmann. Content-based image retrieval by matching hierarchical attributed region adjacency graphs. *Medical Imaging*, 5370:598–606, 2004.
- [Fun05] S. Funke. Topological hole detection in wireless sensor networks and its applications. In *Proceedings of the 2005 Joint Workshop on Foundations of Mobile Computing*, pages 44–53, Cologne, Germany, 2005.
- [FZWN08] C. Farah, C. Zhong, M. Worboys, and S. Nittel. Detecting topological change using wireless sensor networks. In *5th International Conference of Geographic Information Science (GIScience 2008), LNCS 5266*, pages 55–69, 2008.
- [Gal97] A. Galton. Continuous change in spatial regions. In *Proceedings of Conference on Spatial Information Theory (COSIT’97), LNCS 1329*, pages 1–13, Laurel Highlands, PA, USA, 1997.
- [Gal00] A. Galton. *Qualitative spatial change*. Oxford University Press, 2000.
- [Gal04] A. Galton. Fields and objects in space, time, and space-time. *Spatial Cognition and Computation*, 4(1):39–68, 2004.

- [GHS07] S. Gandhi, J. Hershberger, and S. Suri. Approximate isocontours and spatial summaries for sensor networks. In *Proceedings of the 6th International Symposium on Information Processing in Sensor Networks (IPSN'07)*, pages 400–409, Cambridge, MA, USA, 2007.
- [GS04] P. Grenon and B. Smith. SNAP and SPAN: Towards dynamic spatial ontology. *Spatial Cognition and Computation*, 4(1):69–104, 2004.
- [Gua98] N. Guarino. Formal ontology and information systems. In *Proceedings of the 1st International Conference on Formal Ontologies in Information Systems (FOIS'98)*, pages 3–15, 1998.
- [GW05] A. Galton and M.F. Worboys. Processes and events in dynamic geospatial networks. In *Proceedings of 1st International Conference on Geospatial Semantics, LNCS 3799*, pages 45–59, Mexico City, Mexico, 2005.
- [Haz05] S. M. Hazarika. *Qualitative spatial change: Space-time histories and continuity*. PhD thesis, The University of Leeds, 2005.
- [HE98] K. Hornsby and M. Egenhofer. Identity-based change operations for composite objects. In *Proceedings of the Eighth International Symposium on Spatial Data Handling*, Vancouver, Canada, 1998.
- [HE99] K. Hornsby and M. Egenhofer. Shifts in detail through temporal zooming. In *Proceedings of DEXA Workshop*, pages 487–491, Florence, Italy, 1999.
- [HE00] K. Hornsby and M. Egenhofer. Identity-based change: A foundation for spatio-temporal knowledge representation. *International Journal of Geographical Information Science*, 14(3):207–224, 2000.

- [HW04] J.M. Hellerstein and W. Wang. Optimization of in-network data reduction. In *Proceedings of the 1st International Workshop on Data Management for Sensor Networks*, pages 40–47, 2004.
- [JN08] G. Jin and S. Nittel. Tracking deformable 2D objects in wireless sensor networks. In *16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 491–494, 2008.
- [JSAC01] C.E. Jones, K.M. Sivalingam, P. Agrawal, and J.C. Chen. A survey of energy efficient network protocol. *Wireless Networks*, 7:343–358, 2001.
- [JW08] J. Jiang and M. Worboys. Detecting basic topological changes in sensor networks by local aggregation. In *16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 13–22, 2008.
- [KE06] Y. Kurata and M. Egenhofer. The head-body-tail intersection for spatial relations between directed line segments. In *Proceedings of the 4th International Conference on Geographic Information Science (GIScience 2006)*, LNCS 4197, pages 269–286, Münster, Germany, 2006.
- [Kil86] A.C. Kilgour. Techniques for modeling and displaying 3D scenes. *Advances in Computer Graphics II*, pages 55–113, 1986.
- [Kim73] J. Kim. Causation, nomic subsumption, and the concept of event. *The Journal of Philosophy*, 70(8):217–236, 1973.
- [Kro95] W.G. Kropatsch. Building irregulars pyramids by dual graph contraction. *IEE Proceedings on Vision, Image and Signal Processing*, 142(6):366–374, 1995.

- [KVVW89] G.H. Kirby, M. Visvalingam, and P. Wade. The recognition and representation of polygons with holes. *Computer Journal*, 32(6):554 – 562, 1989.
- [Lip65] S. Lipschutz. *Theory and problems of general topology (Schaum's Outline Series)*. McGraw-Hill Publishing, 1965.
- [LL07] Y. Liu and M. Li. Iso-map: Energy-efficient contour mapping in wireless sensor networks. In *Proceedings of the 27th International Conference on Distributed Computing Systems (ICDCS'07)*, page 36, 2007.
- [LY06] M. Li and B. Yang. A survey on topology issues in wireless sensor network. In *Proceedings of the 2006 International Conference on Wireless Networks*, page 503, Las Vegas, Nevada, USA, 2006.
- [MFHH05] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. TinyDB: An acquisitional query processing system for sensor networks. *ACM Transactions on Database Systems*, 30(1):122–173, 2005.
- [MLNL04] X. Meng, L. Li, T. Nandagopal, and S. Lu. Event contour: An efficient and robust mechanism for tasks in sensor networks. Technical report, TR-040018, Computer Science Department, UCLA, 2004.
- [Mon69] R. Montague. On the nature of certain philosophical entities. *The Monist*, 53(1):159–194, 1969.
- [MSKP01] S. Meguerdichian, S. Slijepcevic, V. Karayan, and M. Potkonjak. Localized algorithms in wireless ad-hoc networks: Location discovery and sensor exposure. In *Proceedings of MobiHOC 2001*, pages 106–116, Long Beach, CA, USA, 2001.

- [NGSA08] S. Nath, P.B. Gibbons, S. Seshan, and Z.R. Anderson. Synopsis diffusion for robust aggregation in sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 4(2), 2008.
- [NSC<sup>+</sup>04] S. Nittel, A. Stefanidis, I. Cruz, M. Egenhofer, D. Goldin, A. Howard, A. Labrinidis, S. Madden, A. Voisard, and M. Worboys. Report from the First Workshop on Geo Sensor Networks. *ACM SIGMOD Record*, 33(1):141–144, 2004.
- [Pai98] J. Paiva. *Topological Equivalence and Similarity in Multi-Representation Geographic Databases*. PhD thesis, The University of Maine, 1998.
- [Pav81] T. Pavlidis. *Algorithms for graphics and image processing*. Computer Science Press, 1981.
- [PKG04] S. Patterm, B. Krishnamachari, and R. Govindan. The impact of spatial correlation on routing with compression in wireless sensor networks. In *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN'04)*, pages 28–35, 2004.
- [PPKS06] S. Poduri, S. Patterm, B. Krishnamachari, and G.S. Sukhatme. Sensor network configuration and the curse of dimensionality. In *3rd IEEE Workshop on Embedded Networked Sensors*, 2006.
- [Pro08] Prowler. Prowler-probabilistic wireless network simulator. <http://www.isis.vanderbilt.edu/Projects/nest/prowler/>, 2008. Last accessed June 4, 2008.
- [PV00] F. Pianesi and A. Varzi. Events and event talk: An introduction. In F. Pianesi and A. Varzi, editors, *Speaking Events*, pages 3–47. Oxford University Press, 2000.

- [Qui64] W.V. Quine. *Word and object*. The MIT Press, 1964.
- [Ren00] A. Renolen. Modeling the real world: Conceptual modeling in spatiotemporal information system design. *Transactions in GIS*, 4(1):23–42, 2000.
- [RK95] A.A. Roshannejad and W. Kainz. Handling identities in spatio-temporal databases. In *Technical papers of the 1995 ACSM/ASPRS Annual Convention and Exposition*, volume 4, pages 119–126, 1995.
- [Ros74] A. Rosenfeld. Adjacency in digital pictures. *Information and Control*, 26:24–33, 1974.
- [Sad07] M.J. Sadeq. Distributed detection of spatiotemporal change of regions in wireless sensor networks using boundary state. In *ISSNIP/ISPRS joint international workshop on distributed geoinformatics and sensing, ubiquity, and mobility, DGSUM'07*, 2007.
- [SB06] M. Schneider and T. Behr. Topological relationships between complex spatial objects. *ACM Transactions on Databases Systems*, 31(1):39–81, 2006.
- [SBA04] N. Shrivastava, C. Buragohain, and D. Agrawal. Medians and beyond: New aggregation techniques for sensor networks. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, pages 239–249, Baltimore, MD, USA, 2004.
- [SBLC03] M.A. Sharaf, J. Beaver, A. Labrinidis, and P.K. Chrysanthis. TiNA: A scheme for temporal coherency-aware in-network aggregation. In *Proceedings of the 5th International ACM Workshop on Data Engineering for Wireless and Mobile Access*, pages 69–76, 2003.
- [SBY06] A. Silberstein, R. Braynard, and J. Yang. Constraint chaining: On energy-efficient continuous monitoring in sensor networks. In *Proceedings of the*

*2006 ACM SIGMOD International Conference on Management of Data (SIGMOD '06)*, pages 157–168, 2006.

- [SG04] B. Smith and P.B. Grenon. The cornucopia of formal-ontological relations. *Dialectica*, 58(3):279–296, 2004.
- [Shi97] G. Shirah. Sea surface height anomalies showing the onslaught of El Niño: August 1, 1996 to august 3, 1997. <http://svs.gsfc.nasa.gov/vis/a000000/a000100/a000124/>, 1997. Last accessed Oct. 29, 2007.
- [Sim87] P. Simons. *Parts: A study in Ontology*. Oxford University Press, 1987.
- [SO05] I. Solis and K. Obraczka. Isolines: Energy-efficient mapping in sensor networks. In *Proceedings of the 10th IEEE Symposium on Computers and Communications (ISCC'05)*, pages 379–385, 2005.
- [SV92] M. Scholl and A. Voisard. Object-oriented database systems for geographic applications: An experiment with  $O_2$ . In F. Bancilhon, C. Delobel, and P. C. Kanellakis, editors, *The  $O_2$  Book*, pages 585–618. Morgan Kaufmann, 1992.
- [SW99] J. G. Stell and M. F. Worboys. Generalizing graphs using amalgamation and selection. In *Proceedings of Advances in Spatial Databases 1999, LNCS 1651*, pages 19–32, 1999.
- [SZG<sup>+</sup>08] R. Sarkar, X. Zhu, J. Gao, L.J. Guibas, and J.S.B. Mitchell. Iso-contour queries and gradient descent with guaranteed delivery in sensor networks. In *Proceedings of the 27th Annual IEEE Conference on Computer Communications (INFOCOM'08)*, Phoenix, AZ, USA, 2008.



- [VE08] M. Vasardani and M. Egenhofer. Single-holed regions: their relations and inferences. In *Proceedings of GIScience 2008, LNCS 5266*, pages 337–353, Park City, UT, 2008.
- [WB93] M.F. Worboys and P. Bofakos. A canonical model for a class of areal spatial data. In *Proceedings of the 3rd International Symposium, SSD'93, LNCS 692*, pages 36–52, 1993.
- [WD06] M. Worboys and M. Duckham. Monitoring qualitative spatiotemporal change for geosensor networks. *International Journal of Geographical Information Science*, 20(10):1087–1108, 2006.
- [WGM06] Y. Wang, J. Gao, and J. S. B. Mitchell. Boundary recognition in sensor networks by topological methods. In *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking (MOBICOM'06)*, pages 122–133, 2006.
- [WH04] M.F. Worboys and K. Hornsby. From objects to events: GEM, the geospatial event model. In C. Freksa M. Egenhofer and H. Miller, editors, *Proceedings of the 3rd International Conference on GIScience, LNCS 3234*, pages 327–344, 2004.
- [Wil06] D. Wilmsen. Derivation of change from sequences of snapshots. Master's thesis, The University of Maine, 2006.
- [Wor05] M.F. Worboys. Event-oriented approaches to geographic phenomena. *International Journal of Geographical Information Science*, 19(1):1–28, 2005.
- [Xu02] N. Xu. A survey of sensor network applications. *IEEE Communications Magazine*, 40(8):102–114, 2002.

- [Zha05] R. Zhang. Hierarchies for event-based modeling of geographic phenomena. Master's thesis, The University of Maine, 2005.
- [ZSGM08] X. Zhu, R. Sarkar, J. Gao, and J.S.B. Mitchell. Light-weight contour tracking in wireless sensor networks. In *Proceedings of the 27th Annual IEEE Conference on Computer Communications (INFOCOM'08)*, Phoenix, AZ, USA, 2008.
- [ZT01] J.M. Zacks and B. Tversky. Event structure in perception and conception. *Psychological Bulletin*, 127:3–21, 2001.
- [ZW07] C. Zhong and M. Worboys. Continuous contour mapping in sensor networks. In *Proceedings of the 5th IEEE Consumer Communications and Networking Conference (CCNC 2008)*, pages 152–156, 2007.

## BIOGRAPHY OF THE AUTHOR

Jixiang Jiang was born in Shengze, Jiangsu, China on December 18, 1980. He attended Suzhou middle school and graduated in 1999.

Jixiang Jiang received his B.E. and M.E. degrees in 2003 and 2005 respectively, both in computer engineering from Southeast University, Nanjing, China. From September 2003 to June 2005, he was a research assistant of web information processing group at Southeast University. His research focused on developing new approaches to web information processing. His master thesis was titled “Application of non-negative matrix factorization to web information processing”. He also participated in and made a significant contribution to a research project for Jiangsu information center, in which he implemented a Chinese news search engine with his colleagues.

In August 2005, Jixiang was enrolled for Ph.D. study in the Department of Spatial Information Science and Engineering, University of Maine. Since then, he has been a graduate research assistant with the National Center for Geographic Information and Analysis and received the graduate research assistant award from College of Engineering, University of Maine in 2008.

Jixiang Jiang is a candidate for the Doctor of Philosophy degree in Spatial Information Science and Engineering from the University of Maine in August, 2009.