

**EFFICIENT INFORMATION INTEGRATION SYSTEM FOR TEMPORAL AND  
SPATIAL DATA**

by

**Pei-Ju (Julian) Lee**

B.A. Electrical Engineering, National Taiwan University of Science and Technology, 2005

M.S. Electrical Engineering, National Chung-Hsing University, 2007

M.S. Industrial Engineering, University of Pittsburgh, 2009

Submitted to the Graduate Faculty of  
The School of Information Sciences in partial fulfillment  
of the requirements for the degree of  
Doctor of Philosophy

University of Pittsburgh

2015

UNIVERSITY OF PITTSBURGH  
SCHOOL OF INFORMATION SCIENCES

This dissertation was presented

by

Pei-Ju (Julian) Lee

It was defended on

March 17, 2015

and approved by

Vladimir Zadorozhny, Associate Professor, School of Information

Stephen C. Hirtle, Professor, School of Information Sciences

Marek Druzdzel, Associate Professor, School of Information Sciences

Paul Munro, Associate Professor, School of Information Sciences

John Grant, Adjunct Professor, Department of Computer Science, University of Maryland

Dissertation Advisor: Vladimir Zadorozhny, Associate Professor, School of Information

Sciences

Copyright © by Pei-Ju (Julian) Lee

2015

# **EFFICIENT INFORMATION INTEGRATION SYSTEM FOR TEMPORAL AND SPATIAL DATA**

Pei-Ju (Julian) Lee, PhD

University of Pittsburgh, 2015

In this dissertation, I develop a novel inconsistency detection and data fusion method for data integration systems. Inconsistent data may lead to incorrect query results and induce unexplainable outcomes. I propose an inconsistency detection method to find out which data items (e.g., temporal or spatial report) have the higher potential to cause data conflicts as well as to estimate a reasonable consistent reported value. My approach is based on representing overlapping data reports as a characteristic linear system. The characteristic linear system can be used to estimate consistent reported values within overlapping time and space intervals. I explore applicability of the proposed approach in different domains. In particular, I perform temporal data fusion with time-overlapping reports using a historical database. I also experiment with spatial data fusion involving space-overlapping reports using simulation of sensor data sets of robots performing search and rescue task. Finally, I apply the proposed approach to combine temporal and spatial fusion and demonstrate that such multidimensional fusion improves inconsistency detection and target value estimation.

## TABLE OF CONTENTS

<b>1.0</b>	<b>INTRODUCTION.....</b>	<b>1</b>
<b>1.1</b>	<b>MOTIVATION AND PROBLEM STATEMENTS.....</b>	<b>1</b>
<b>1.2</b>	<b>OBJECTIVE AND FOCUS OF THIS STUDY .....</b>	<b>3</b>
<b>1.3</b>	<b>STRUCTURE AND OVERVIEW OF THIS DISSERTATION.....</b>	<b>5</b>
<b>2.0</b>	<b>BACKGROUND AND RELATED WORKS .....</b>	<b>7</b>
<b>2.1</b>	<b>EFFICIENT DATA FUSION FOR HETEROGENEOUS DATA SOURCES</b>	
	<b>7</b>	
<b>2.2</b>	<b>INFORMATION FUSION .....</b>	<b>10</b>
<b>2.2.1</b>	<b>Information fusion for data integration .....</b>	<b>11</b>
<b>2.2.2</b>	<b>Multisensory information fusion.....</b>	<b>14</b>
<b>3.0</b>	<b>SYSTEM DESIGN – TEMPORAL INFORMATION FUSION.....</b>	<b>19</b>
<b>3.1</b>	<b>HISTORICAL DATA SOURCES.....</b>	<b>19</b>
<b>3.2</b>	<b>PROPOSED APPROACH.....</b>	<b>22</b>
<b>3.2.1</b>	<b>Characteristic linear system .....</b>	<b>23</b>
<b>3.2.1.1</b>	<b>Underdetermined linear system.....</b>	<b>27</b>
<b>3.2.1.2</b>	<b>Linear programming .....</b>	<b>29</b>
<b>3.2.1.3</b>	<b>Nonnegative least squares method.....</b>	<b>34</b>
<b>3.2.2</b>	<b>Reverse substitution method: inconsistency detection.....</b>	<b>37</b>

3.2.3	Reverse substitution method: data fusion .....	45
3.3	STUDY 1 – INCONSISTENCY DETECTION IN REAL DATA.....	48
3.4	STUDY 2 – INCONSISTENCY DETECTION IN SIMULATED DATA ...	51
4.0	IMPLEMENTATION AND EVALUATION OF TEMPORAL FUSION.....	57
4.1	BACKGROUND OF THE CONFLICT DEGREE METHOD.....	57
4.2	EXPERIMENT SETUP AND CONSISTENCY CONDITIONS .....	60
4.3	EFFECT OF INCONSISTENCY.....	64
4.4	ACCURACY AND CONSISTENCY.....	69
5.0	SYSTEM DESIGN - TEMPORAL SPATIAL INFORMATION FUSION .....	72
5.1	GENERAL ARCHITECTURE OVERVIEW .....	72
5.2	INFORMATION FUSION TAXONOMY .....	78
5.3	SPATIAL INFORMATION FUSION SIMULATION.....	80
5.4	PILOT STUDY OF TEMPORAL-SPATIAL FUSION SYSTEM .....	99
5.5	MULTIDIMENSIONAL TEMPORAL SPATIAL INFORMATION FUSION	104
6.0	CONCLUSIONS .....	117
6.1	DISCUSSIONS AND APPLICATIONS.....	117
6.2	FUTURE WORK.....	123
	BIBLIOGRAPHY .....	125

## LIST OF TABLES

Table 1. Example of historical report with time overlapping .....	3
Table 2. Example of an integrated historical tuple .....	21
Table 3. Example of report overlapping .....	25
Table 4. Example of inconsistent report values .....	39
Table 5. Subsets with nonzero delta .....	42
Table 6. Find conflict report ID .....	43
Table 7. Example of pyramids with 5 reports.....	53
Table 8. Corresponding RO, RC, and CD for different report structure .....	59
Table 9. Configuration of inconsistency simulation .....	62
Table 10. Inaccuracy and subsumption condition.....	71
Table 11. Static target, static robot .....	75
Table 12. Static target, moving robot.....	75
Table 13. Moving target, static robot.....	76
Table 14. Moving target, moving robot .....	77
Table 15. Configurations of space report coverage .....	88
Table 16. Probability distribution of basic method.....	95
Table 17. Example of temporal spatial fusion .....	101

Table 18. Ground truth of space reports at $T_1$ .....	105
Table 19. Configurations of two-dimensional fusion .....	109

## LIST OF FIGURES

Figure 1. Data integration architecture in historical data center based on collective intelligence..	8
Figure 2. Information integration architecture.....	9
Figure 3. Example of two measles reports in overlapped time interval.....	22
Figure 4. Proposed data fusion model overview.....	23
Figure 5. Example of four reports with overlapped time intervals .....	25
Figure 6. Iterative procedure.....	32
Figure 7. Example of number-of-conflict-report .....	44
Figure 8. Example of nonzero number-of-conflict-report subsets.....	45
Figure 9. Example of data fusion and values for each time interval.....	46
Figure 10. Example of inconsistent reports and interval values .....	47
Figure 11. Simulation result of inconsistency detection.....	49
Figure 12. Tycho disease reference table.....	50
Figure 13. Tycho data reference report table .....	51
Figure 14. Example of 5 reports .....	52
Figure 15. Simulation results of 5 reports.....	53
Figure 16. Relation between number of conflicting reports and delta value .....	54
Figure 17. Magnitude difference and delta with two reports.....	56

Figure 18. Scenarios of CD.....	58
Figure 19. Relative distance of CD and RS for 1000 time units.....	62
Figure 20. Relative distance of CD and RS for 150 time units.....	63
Figure 21. Comparison with 25% and 95% probability of swap.....	66
Figure 22. Comparison with 50% and 95% probability of swap.....	66
Figure 23. Total delta for 25%, 50%, and 100% probability of swap .....	67
Figure 24. Comparison of CD and RS in each probability of swap conditions.....	68
Figure 25. OCD distribution for all inconsistent condition .....	69
Figure 26. Target(victim) detection categories of temporal and spatial fusion .....	79
Figure 27. Fusion point.....	81
Figure 28. The RD of the RS method and the CD method for 300 time units .....	83
Figure 29. Avg. RD for fusion point at 10TU (left) and 100TU (right) .....	84
Figure 30. Run Time difference of TU300.....	85
Figure 31. The RD of SU100(left) and SU25(right).....	86
Figure 32. The Avg. RD and RT diff. of SU100 (left) and SU25 (right) .....	87
Figure 33. RC comparisons of ED10.....	89
Figure 34. RC comparisons of ED100.....	90
Figure 35. RD diff. between CD and RS of ED10 (left) and ED100 (right) .....	91
Figure 36. Sum RD of the RS method.....	92
Figure 37. RD of the RS method of ED10.....	92
Figure 38. RD of the RS method of ED100.....	93
Figure 39. RD vs. RT diff. ....	94
Figure 40. JSD of different sparsity.....	97

Figure 41. AUC of different sparsity .....	98
Figure 42. RD of the RS method of different sparsity .....	99
Figure 43. Spatial layout in grids .....	100
Figure 44. Number of target in each cell and time unit .....	100
Figure 45. Accuracy of different number of report.....	102
Figure 46. Example of spatial fusion .....	103
Figure 47. Two-dimensional reports generation.....	104
Figure 48. Two-dimensional data of dynamic target .....	108
Figure 49. RD across time units after temporal fusion .....	111
Figure 50. Percentile plot across time units after temporal fusion .....	111
Figure 51. RD across space units after temporal spatial fusion .....	112
Figure 52. Avg. RD of TF and TFSF fusions .....	113
Figure 53. Percentile RD of TF and TFSF fusions .....	114
Figure 54. Avg. RD in Low/High TRN/TRD/SRN .....	116

## **1.0 INTRODUCTION**

### **1.1 MOTICATION AND PROBLEM STATEMENTS**

With the emergence of new data sources on the Internet, integrating data from heterogeneous data repositories has become critical. For example, RFID (radio frequency identification) tags circulation increased from 1.3 billion to 30 billion from 2005 to 2011; the data generated from a single engine of an airplane in a half hour is about 10 terabyte; and generally Facebook can have 2.5 billion likes and 300 million photo uploads per day (Zikopoulos et al., 2012). The general ground truths we can acquire from the data above include the amount and location of products obtained by RFID, airplane and flight circumstance records provided by log data, and the relationship of a photo and specific users on social media. In addition to the basic information above, aggregating data from heterogeneous data repositories can provide us with other aspects of data analysis such as logistic optimization for saving storage and transportation costs using RFID, risk management, and maintenance of aircraft and analysis of social networks in cyber space. “This data comes from everywhere: sensors used to gather climate information, posts to social media sites, digital pictures and videos, purchase transaction records, and cell phone GPS signals to name a few. This data is big data...and big data spans four dimensions: volume, velocity, variety, and veracity.... Big Data is all about better analytics on a broader spectrum of data, and therefore represents an opportunity to create even more differentiation among industry

peers” (Zikopoulos et al., 2012). The amount of data is increasing on a large scale and at a fast pace due to the improvement of storage devices and processing ability. Therefore, researchers have started to explore methods and techniques to process and analyze big data, and to solve related problems that did not exist or have not been valued before. The size of available data, the connectivity between data sources, and abilities of analytic technologies make the information integration for big data prominent.

Using multiple data repositories and sensors provide users with comprehensive and complementary information. However, multiple sources of data introduce problems such as redundancy, conflicts, or missing data reports. The two major categories of challenges for large scale data integration systems are (1) heterogeneous data and (2) conflicting data (Zadorozhny & Hsu, 2011). Heterogeneous data refer to data stored in different schemas or in different representations, and conflicting data refer to data stored in multiple databases with inconsistent attributes (i.e. time/location/name). The solutions for heterogeneous data have been researched for many years, but the challenges of conflicting data are not well explored yet. My approach aims at improving the quality of information integration via data inconsistency detection and information fusion. Of interest to me in the case studies are historical data sources which include numerous events with a wide range of time duration, as well as the simulated sensor data sets of robots performing search and rescue task with overlapping temporal and spatial reports. These historical or sensor data may overlap due to redundancy of records, or inaccuracy of original data. Inaccurate results and poor decision making may occur during the integration process if the data is redundant and inconsistent. Users should spend a large amount of time and effort to analyze and extract the correct information from the distributed data sources, which involve conflict detection and processing of conflicting reports. Related reliability assessment protocols

based on analysis of data inconsistencies is critical to form a consistent repository of integrated data.

## 1.2 OBJECTIVE AND FOCUS OF THIS STUDY

The unified repository of a large number of data sets provides researchers an easier way to access multiple resources in a single set with a homogeneous schema. However, some implicit problems for this mass of data will obstruct the analysis of these historical data. For example, Table 1 shows the mockup data of two historical reports with overlapping time intervals:

**Table 1.** Example of historical report with time overlapping

<b>ID</b>	<b>Value</b>	<b>Name</b>	<b>Location</b>	<b>Start time</b>	<b>End time</b>
<b>1</b>	100	Measles	Pittsburgh, PA	1/1/2001	12/31/2013
<b>2</b>	500	Measles	Pittsburgh, PA	1/1/2001	12/31/2005

This time overlapping condition is very common in historical data sets. For example, when researchers interested in the total Measles cases in the greater Pittsburgh area, they collect data sets from different resources with varying time coverage. The above example shows an erroneous number of incompatible total cases of Measles – we cannot calculate the total cases of Measles by simply summing up the reports values. The reports’ times are overlapping and for larger time interval the reported value is smaller, which indicates inconsistency. We cannot differentiate what caused this error because it may result from human error of recording tuple name, time, or location. However, we should be able to detect this inconsistency.

I propose to represent the overlapping reports using an underdetermined linear system called *characteristic linear system* in my dissertation. For reports be modeled in the characteristic linear system, I can detect data inconsistencies before performing data fusion to estimate most likely consistent value. This reduces the consumption of time and effort for the fusion, and also reduces potential incorrect query results. The underdetermined linear system and its solution set can be used to detect the occurrence of inconsistent reports, ascertain the ID of conflict reports, decrease the inconsistency by suggesting possible real report values or eliminate these conflict reports, and improve data accuracy and reliability.

In this dissertation, I test my algorithm for temporal data fusion using the historical data source of an integrated epidemiological data warehouse that records sequential diseases information from heterogeneous data sources. This data warehouse contains about 50,000 reports for more than 100 years of United States epidemiology data. The data I use in this dissertation is integrated from these 50,000 reports across different data sources that are represented as heterogeneous data formats. I perform inconsistency detection and data fusion for aggregated epidemiological records. After conflict detection, I perform temporal data fusion for this data set to provide reasonable estimated value for each time interval. In addition, I test my proposed algorithm of spatial data fusion through the simulation of the task of robots conducting urban search and rescue mission. Robots mounted with lasers and cameras can explore the environment and produce video streams and laser logs for the user. Robots detect immobilized targets when they explore different areas, but the laser logs may have multiple scans with overlapping areas from different robots. The overlapping spatial logs may result in double counted targets. In order to further involve both temporal and spatial dimensions in the process of data fusion, I extend the scenario of the search and rescue task of target detection at specific locations and time intervals

with dynamic targets. My major research questions and corresponding hypothesis for this dissertation are:

- **Research question 1:** How to detect inconsistency in temporal and spatial data?

**Hypothesis 1:** My proposed characteristic linear system and reverse substitution method can be used to indicate which report(s) have the higher degree of inconsistency, or to indicate which report(s) cause the inconsistency. Thus, the user can spend less time to find the targeted problem reports.

- **Research question 2:** How can inconsistent temporal and spatial data be processed?

**Hypothesis 2:** I can detect inconsistency for different configurations of temporal and spatial reports (i.e. overlap, subsumption, number of report, etc) through the degree of inconsistency and perform data fusion through the estimated values generated by the characteristic linear system.

- **Research question 3:** How can the inconsistency detection and analysis be used for scalable data fusion?

**Hypothesis 3:** The reverse substitution method can provide a good estimate of aggregate value for reports with inconsistency in any single data dimension as well as in multidimensional data such as the temporal and spatial dimensions in this dissertation.

### **1.3 STRUCTURE AND OVERVIEW OF THIS DISSERTATION**

The remainder of this dissertation is structured as follows: Section 2.0 describes the background knowledge of data inconsistency detection, data fusion (Section 2.1) and other related works of information integration (Section 2.2.1) and multisensory information fusion (Section 2.2.2). My

proposed Reverse Substitution (RS) method of characteristic linear system for temporal data fusion will be introduced in Section 3.0 . I present the background knowledge of historical data sources in Section 3.1. The proposed RS method (Section 3.2 ) includes the generation of the characteristic linear system (Section 3.2.1) and the nonnegative least squares method to generate the solution set (Section 3.2.2). The experiment of inconsistency detection using real data set and simulation-based study is presented in Section 3.3 and Section 3.4 correspondingly. The evaluations and comparisons of my proposed approach and the related conflict degree method are shown in Section 4.1, 4.2, and 4.3. Section 4.4 discusses performance of the proposed RS method. Section 5.1 and 5.2 outline the work of target observation for the task of temporal and spatial data fusion, which includes target identification and target movement trajectory estimation at specific locations and time intervals. Section 5.3 addresses the spatial fusion, and Section 5.4 and 5.5 address the multidimensional temporal-spatial fusion. Section 6.0 concludes discussing applications of the proposed approach (Section 6.1) as well as the future work for its possible improvements (Section 6.2).

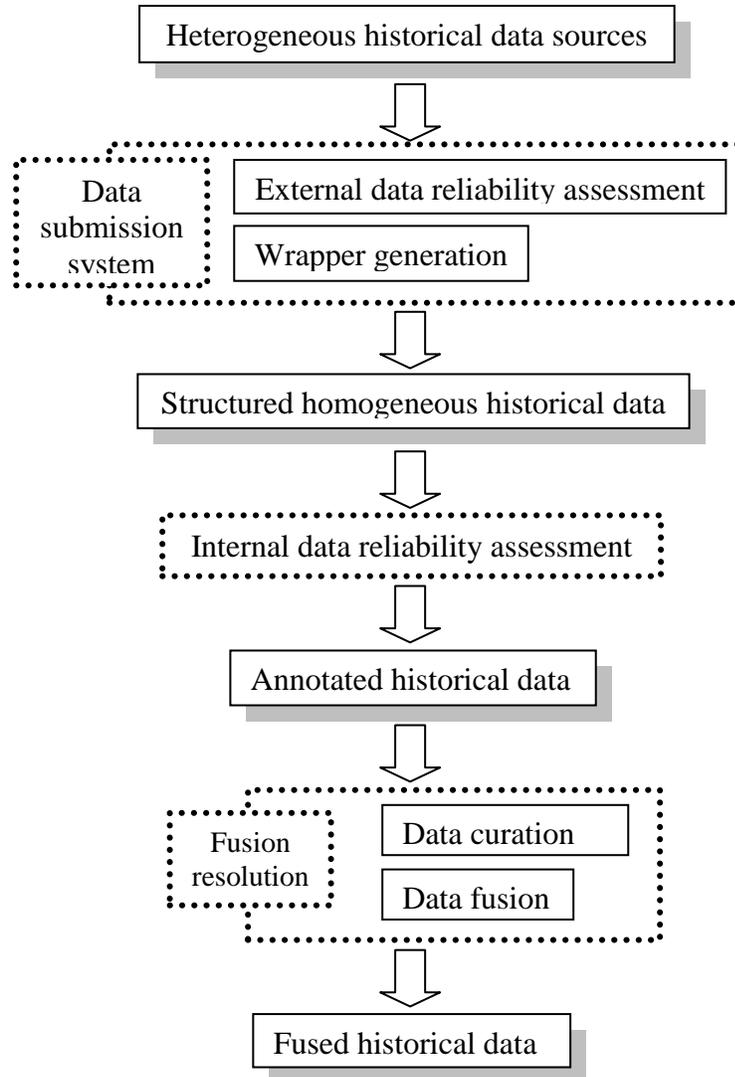
## **2.0 BACKGROUND AND RELATED WORKS**

### **2.1 EFFICIENT DATA FUSION FOR HETEROGENEOUS DATA SOURCES**

Data integration from heterogeneous data sources requires a tremendous amount of work. The possible problems that users may encounter during data integration processes are inaccurate data, inconsistent data and redundant data. These problems are either caused by heterogeneous data sets or conflicting data sets. Heterogeneous data is defined as data stored in different schemas or in different representations. Redundant data is defined as data stored in multiple databases with overlapping time, location, or name. These redundant data may result in inconsistency if the overlapping parts are inconsistent (i.e. temporal/spatial/naming inconsistency) (Zadorozhny & Hsu, 2011).

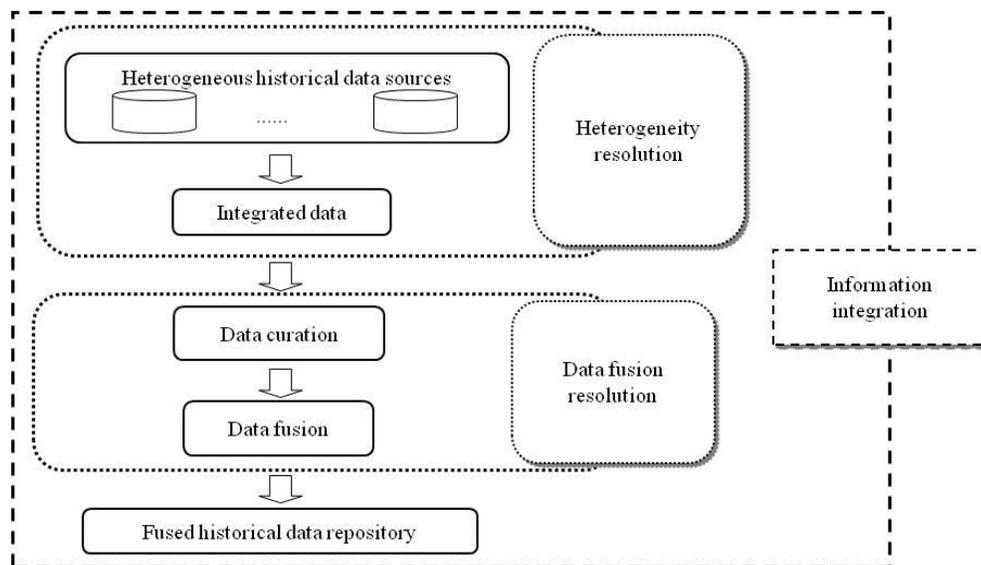
From the database point of view, data integration may be performed when there is heterogeneity at the schema level, tuple level, or value level. Information resulted from data integration process at different levels may have different representations, information types, and functionality, etc. Thus, when a designer starts to create a data integration system, the factors that needs to be considered includes the type of data, the algorithm of data merge and the level where the data integration process happens. A common approach to perform data integration involves the following steps: (1) identify the corresponding attributes in the sources, (2) differentiate objects that originate in different sources and if these data describe the same attributes, and (3) merge these sources into a single representation. Figure 1 describes an architecture of

information integration proposed by (Zadorozhny, Manning, Bain, & Mostern, 2013) to integrate historical data from heterogeneous data sources using collective intelligence. Information providers in each repository submit their data sources through wrapper generation into a structured historical data in a homogeneous global schema. The wrapper transforms different kinds of data source, such as CSV, into a target schema. The data submission system contains wrapper generation, wrapper registration, and external data reliability assessment.



**Figure 1.** Data integration architecture in historical data center based on collective intelligence

In this dissertation, the information integration is defined as two major processes: the *heterogeneity resolution* and the *data fusion* as shown in Figure 2. The heterogeneity resolution refers to unify the heterogeneous data sources from various schemas, types, and representations into a global format. The data fusion refers to process of data inconsistency conflicts resolution in the integrated data repository. In this dissertation I focus mostly on the data fusion, since this area becomes more significant and is not well explored. The duplicate detection, also known as record linkage, object identification, or reference reconciliation are relevant for data fusion (Bleiholder & Naumann, 2008). It can be accomplished by comparing each object using similarity measurement. A drawback to this method is that sometimes it is neither effective nor efficient especially when the amount of data is very large. Moreover, conflicts may still occur after heterogeneity is resolved. Therefore, the problems are how to detect data redundancy or inconsistency before performing similarity measurements to improve scalability as well as how to correct these inconsistent data during data fusion.



**Figure 2.** Information integration architecture

## 2.2 INFORMATION FUSION

The term "information fusion", or "data fusion" has been used in different contexts. According to (Bleiholder & Naumann, 2008): "There are two other fields in computer science that also use the term (data) fusion. In information retrieval it means the combination of search results of different search engines into one single ranking, therefore it is also called rank merging. In networking it means the combination of data from a network of sensors to infer high-level knowledge, therefore also called sensor fusion. Beyond computer science, in market research, the term data fusion is used when referring to the process of combining two datasets on different, similar, but not identical objects that overlap in their descriptions".

The most important problem or premise that both multisensory data fusion and information integration data fusion face is the huge volume of heterogeneous data. The emergence of the Internet makes it easier to access different data resource systems worldwide in order to obtain information. Types of sensors are usually classified by their physical nature such as electromagnetic spectrum, vision (e.g. video camera), sound waves (e.g. sonar), touch (e.g. tactile sensor), odor, or the absolute position of the system (e.g. range finder) (Esteban, Starr, Willetts, Hannah, & Bryanston-Cross, 2005). And systems usually use a large number of sensors for their tasks. Integrating data from those large-scale data resources, therefore, becomes critical. Here I focus on two major applications of information integration data fusion and multisensory data fusion.

### 2.2.1 Information fusion for data integration

For databases information integration, data fusion can happen at schema level, tuple level, or value level. A variety of techniques for data fusion at each level are described below: for Schema level, (Rahm & Bernstein, 2001) surveyed data integration (or schema matching) approaches following varying criteria such as instance vs. schema, element vs. structure matching, language vs. constraint, matching cardinality, or auxiliary information, etc.; for Tuple level, (Han et al., 1997) proposed merging identical tuples when each attribute in the relevant set of data is generalized to a minimally generalized concept; and for Value level, (Naumann, Bilke, Bleiholder, & Weis, 2006) specified that data fusion occurs at the value level to resolve value inconsistency. This is the last step of their data fusion process which is described as Step 1: Schema matching (i.e. resolve inconsistencies at schema level), Step 2: Duplicate detection (i.e. resolve inconsistencies at tuple level), and Step 3: Data fusion (i.e. resolve inconsistencies at value level).

There are many advantages to using data integration systems such as: 1. Completeness (i.e. no object will be ignored or missed by adding more data sources (i.e. more objects, attributes describing objects) to the system); 2. Robustness (i.e. increase the reliability of datasets); 3. Conciseness (i.e. to access data in different categories or to capture data that happened in different time periods after performing the data fusion process) (Bleiholder & Naumann, 2008).

However, problems or conflicts may occur when accessing data stored in multiple heterogeneous databases. The heterogeneous databases either do not use the same schema or do not represent the same entities in the same way (Hernandez & Stolfo, 1998). Two types of the later incompatibility of heterogeneous databases also addressed in (Chatterjee & Arie, 1991;

Elmagarmid, Ipeirotis, & Verykios, 2007) in two categories: Structural incompatibility: including type mismatch, formats, units, and granularity; and Semantic incompatibility (i.e. lexical heterogeneity in (Elmagarmid et al., 2007): including synonyms, homonyms, codes, incomplete information, recording errors, surrogates, and asynchronous updates.

Some techniques, in addition to query optimization, to resolve these structural or semantic incompatibility issues are listed below: (a) Schema matching approaches in (Rahm & Bernstein, 2001) presented in the similarity session above; (b) Data exchange: uses a set of potential answers instead of a single universal solution to the target schema (Fagin, Kolaitis, & Popa, 2005); (c) Conflict handling strategies such as 1. Conflict ignoring: consider all possibilities (i.e. ignore conflicts and pass all possibly combinations of values to the user, the user needs to choose and decide from these data) and pass it on (i.e. pass all conflicts to users); 2. Conflict avoiding: trust your friends (i.e. avoid conflicts by leaving values only from a specific resource through a decision rule), no gossiping (i.e. return consistent tuples only), and take the information; 3. Conflict resolving: cry with the wolves (i.e. resolve conflicts by leaving the values that are most often used), roll the dice (i.e. take the random values), meet in the middle (i.e. resolve conflicts by creating a new value which is a compromise among all possible values, for example averaging over all numerical values), and keep up to date (i.e. takes the most recent value) (Bleiholder & Naumann, 2008; Naumann et al., 2006); (d) Relational operators: basic operators include union (union-based techniques) and join (join-based techniques). Other techniques extending the relational models for example by considering all possibilities or considering only consistent possibilities (Bleiholder & Naumann, 2008). Another technique of entity operators (i.e. Entity Join) for entities named differently inter-databases is addressed in (Chatterjee & Arie, 1991). The authors also surveyed other strategies such as standardizing the

names, rule based approach, information theoretic approach, and imprecise query specification for heterogeneous databases; (e) Wrapper construction: (Chawathe et al., 1994) focus on translator and mediator generators and proposes the OEM (object exchange model) to provide resource access and information integration; (f) Mass collaboration approach: adjusts system parameters for semantic mapping by users feedback (Doan & McCann, 2003); (g) Virtual attribute (Ex. maybe tuple): expands the notion of dynamic attributes to map conflicting attributes to a common domain that then can use extended relational algebra operations (DeMichiel, 1989); (h) Combining with data clustering: modifies the sorted-neighborhood method by cut the data cleaning process to multiple small windows of passes (Hernandez & Stolfo, 1998); (i) Semantic correspondence: uses degree and cardinality measurements to represent closeness of links between data and mappings between domains (Mahoui, Kulkarni, Li, Ben-Miled, & Borner, 2005); (j) Self-configuration system: based on the probabilistic mediated schema from sources to the mediated schema (Sarma, Dong, & Halevy, 2008); (k) Graph-based data integration framework: combining three basic methods such as accession based mapping (i.e. use an accession coding system to link concepts with same reference between different databases), synonym mapping (i.e. link two concepts if they have same preferred concept names) and StructAlign mapping (i.e. use the graph neighborhood of two concepts to disambiguate their meaning) (Taubert et al., 2009); (l) Google Fusion Table: allows users uploading tabular data files to a big cloud storage and management service which supports SQL queries (Gonzalez et al., 2010); or (m) Similarity metrics: uses metrics such as character-based similarity metrics (including edit distance, affine gap distance, Smith-Waterman distance, Jaro distance metric, and Q-gram distance), token-based similarity metrics (including atomic string, WHIRL system, and Q-grams with tfidf), phonetic similarity metrics (including soundex, NYSIIS, ONCA, and

metaphone), and numeric similarity metrics to detect similar field entries (Elmagarmid et al., 2007).

In the data fusion, conflict handling strategies can be seen as a subarea or as a synonym of data fusion. There is a huge volume of techniques to resolve conflicts for information integration data fusion and multisensory data fusion, some are listed in this dissertation. The benefits of data fusion have motivated a variety research in areas such as maintenance engineering, robotics, pattern recognition and radar tracking, mine detection and other military applications, remote sensing, traffic control, aerospace systems, law enforcement, medicine, finance, metrology, and geo-science (Esteban et al., 2005). Other application areas are such as integrate data from earth's ecosystem (i.e., climate data, geospatial data, etc.), biomedical data, web service, and business or any other areas have mass data collection.

### **2.2.2 Multisensory information fusion**

Data fusion is most used in multisensory environment and the advantages of using multiple sensors over a single sensor including higher signal-to-noise ratio, robustness and reliability in the evident of sensor failure, parameter coverage, dimensionality of the measurement, confidence and resolution, hypothesis discrimination with the aid of more complete information arriving from multiple sensors, obtaining information regarding independent features in the system, and lower uncertainty, measurement time, as well as possibly costs (Esteban et al., 2005). Typically, more sensors can accomplish more tasks than a single sensor or can accomplish these tasks with better performance. The advantages of using multiple sensors are: (a) Redundancy (i.e. each sensor provides part of information in the environment, through data integration or fusion of data the accuracy can be increased and uncertainty will be decreased); (b) Complementarity (i.e.

different type of sensors can enforce the information perceived when sensors are independent. For example, using both atmospheric sensors and a Webcam for detecting an operators' absence will be more accurate compared with only using one type of sensor); (c) Timeliness (i.e. providing information within one integration process by processing multiple sensors parallel); and (d) Cost of the information provided by the system (i.e. less costly information from a multiple sensor system compare with potentially more costly information obtained from a single sensor) (Luo & Kay, 1989).

Some research distinguishes between data fusion and data integration in multisensory environment. In (Luo & Kay, 1989), multisensory integration “refers to the synergistic use of the information provided by multiple sensory devices to assist in the accomplishment of a task by a system” and multisensory fusion “refers to any stage in the integration process where is an actual combination (or fusion) of different sources of sensory information into one representational format.” Although many researchers use these terminologies, this differentiation is not standard and other researchers may treat these terms as applying to the same process. Early previous approaches to data fusion and data integration were considered in (US Navel Observatory & Almanac, 1960). (Hall & McMullen, 2004) separated data fusion model of functional model (i.e. model contains primary functions, relevant database, and interconnectivity to perform data fusion), architectural model (i.e. focus on the hardware/software, the data flow and external operator interfaces), and mathematical model (i.e. describes the algorithm performing data fusion and logical process).

The data fusion process can happen in a hierarchical or sequential manner or hybrid of these two. Where the fusion process takes place needs to be considered when constructing the data fusion system in a hierarchical framework. Multisensory fusion can happen at different

levels depending upon the requirements of the users and the characteristics of the system. Here, some data integration models and their process levels are described (Esteban et al., 2005) such as Thomopoulos architecture that divided into signal level fusion (i.e. data correlated through learning), evidence level fusion (i.e. data correlated through statistical model or decision making), and dynamics level fusion (i.e. data correlated through mathematical models) (Thomopoulos, 1990); Luo and Kay's framework that divided into signal, pixel, feature, and symbol levels of fusion as the level of representation increases from signal to symbol, the level of information provided to users also increases (Luo & Kay, 1989); or the Waterfall model that divided into signal (i.e. preprocessing the raw data), feature (i.e. feature extraction and pattern processing), and interrogation (i.e. situation assessment and decision making) level (Harris, Bailey, & Dodd, 1998).

Some researchers have classified multisensory data fusion as one subarea of data fusion for information integration. A variety of solutions have been proposed for the problems faced by both of these data fusion tasks. Apart of some common algorithms such as averaging, weighted averaging, or data mining techniques, here are some techniques for these two data fusion areas. First area focuses on data fusion process such as (a) The JDL framework (Hall & Llinas, 1997), (b) Waterfall model (Harris et al., 1998), (c) Omnibus data fusion model: focuses on functional objectives at different data fusion steps (Bedworth & O'Brien, 2000), (d) System-based data fusion architecture: address the requirements for engineering guidelines; there are three steps, identification, estimation, and validation, in this framework (Esteban et al., 2005); (e) Thomopoulos's architecture (Thomopoulos, 1990); and (f) Luo & Kay's framework (Luo & Kay, 1989). The second area focuses on data fusion strategy which will be explained in the following section

(Hackett & Shah, 1990) also put data fusion and data integration into two different categories in which the sensor fusion uses fusion strategy to put multiple sensors into equivalent form to perform fusion and consent of all sensors must be reached; the sensor integration uses sensors sequentially to achieve a particular task, consensus for all sensors is not needed and data of prior sensor can be used to help next sensor performing its task. The sensor fusion also can be put into two categories that direct fusion method using raw data from sensors without any manipulation and indirect fusion method using transformed sensor measurements. Bayesian theory was being introduced in their work to check consistency of sensor data before any direct/indirect fusion is performed. The type of sensors and the level at which data fusion will occur are all needed to be taken into account. For same type of sensors the data screening and data fusion are required, however, for different types of sensors are used then the collected data from heterogeneous sources need to be transformed into the same schema/form and perform data fusion according to the occurrence time, etc.

The most simple algorithm to perform data fusion is using averaging (or weighted averaging in extension) under the environment of same type of sensors (Hackett & Shah, 1990). The complexity increases while there is a large number of sensors or sensors interaction are complex, this condition can be modeled using a probability distribution and a more sophisticated method is needed. The fusion strategies for multisensory are such as (a) Distributed blackboard data fusion model: assigns confidence level to each sensor by supervisors (Schoess & Castore, 1988); (b) Six basic methods of Segmentation, Representation, 3-D shape, Sensor modeling, Autonomous robots, and Recognition are addressed in (Hackett & Shah, 1990); (c) Basic arithmetic methods such as deciding, guiding, averaging, weighting, Bayesian, statistics, integration, and maximum-likelihood are also mentioned in (Hackett & Shah, 1990; Marano,

Matta, & Willett, 2008; Zubko, Leptoukh, & Gopalan, 2010); (d) Locally optimum estimator (LOE) consider a quantize design in the case of an unknown quantity of sensors in a wireless network of a group of independent sensors (Marano et al., 2008); (e) Fuzzy inference method with heterogeneous sensors such as Webcam and atmospheric sensor (Lecce & Amato, 2009), the authors introduced a data fusion approach combining different sensor types in the task of user presence monitoring; and (f) (Chang, Costagliola, Jungert, & Orciuolo, 2004) introduced a spatial/temporal query language  $\Sigma$ QL to perform retrieval and fusion of multimedia sensor data fusion.

The frameworks used to perform data fusion of multisensors are as follows: Joint Directors of Laboratories Data Fusion Framework, Thomopoulos architecture framework, multi-sensor integration fusion model, behavioural knowledge based data fusion model, waterfall model, distributed blackboard data fusion architecture, and omnibus data fusion model. Therefore, some difficulties may encounter for multisensory data fusion are diversity and registration of sensor and data representation, calibration of the sensors when errors in the system operation occur, sensors operability limitations, and deficiencies in the statistical model of the sensors and limitations in the algorithm development (Esteban et al., 2005). In general, the multisensory data fusion strategies include more arithmetic methods because the data's unity and the simplicity of the sensors. The goals of multiple sensors also has a wide variety for object recognition using different types of sensor (Hackett & Shah, 1990).

### **3.0 SYSTEM DESIGN – TEMPORAL INFORMATION FUSION**

In this dissertation, I consider merging data of reports from integrated heterogeneous data sources with temporal or spatial overlapping of events. In this section I focus on inconsistency detection and information fusion for time-overlapping historical data. First I introduce the historical data in Section 3.1 and then consider my proposed model approach in Section 3.2.

#### **3.1 HISTORICAL DATA SOURCES**

The historical data reports record events of users' interest within a time range. The characteristics of historical data reports of dynamic changing and data continuity require a comprehensive consolidation of data sets. These continuous data reports can be found in different areas such as environmental data (ex. climate change), health data (ex. disease contagion), biological data (ex. species migration), or financial data (ex. stock rating), and the data analysis is based on events within some time intervals or location intervals. For example, users may be interested in getting to know the climate change in the Arctic Circle within the past decade or the migration track of zebras across South Africa last year. The type of data introduced in Chapter 2.0 such as the RFID data log, the airplane data log, or the web usage log also have the same features. These data can be stored in disparate data warehouses at distinct locations in which each warehouse contains a portion of the whole data source. The consolidated data

composed of heterogeneous data sources and various time intervals have great potential of data overlap in time, name, or location. Conflicts may still exist after the heterogeneous data have been transformed into a structured historical data with a homogeneous global schema. This dissertation focuses on solving temporal and spatial inconsistency as a pre-procedure for data fusion on integrated data reports. My proposed algorithm formalizes the historical data as a mathematical model of an underdetermined linear system and performs consistency checking of data sets in a global repository. Therefore, we can merge data into a large-scale data integration repository with consistent data to provide completeness and robustness. When the consistency detection fails, my model can perform consistency adjustment with two possible approaches: (1) eliminate inconsistent report data; (2) adjust the data value by suggesting possible real report values. The proposed algorithm has to consider either temporal fusion or spatial fusion separately. Therefore, in the following descriptions I will use temporal fusion as an example for explanation.

For historical data, I assume that reported events reflect aggregated historical statistics (e.g. the total number of cases of specific disease in a duration of time). The historical information can be represented in the following generic schema:

| *Data Source Reference* | *Data Reference* | *Time Duration* | *Data Value* |

The schema contains four components: Data Source Reference, Data Reference, and Time Duration, each of which are comprised of several components. The Data Source Reference is comprised by Source Identifier (SID), Source Publication Date (SPD) and Data Recording Date (DRD). SID is a unique identifier for data resource. SPD indicates the date when the data item is published from the data source. DRD refers to the date when data item is referred to in a historical document such as in a recorded history of the data source. The Data Reference is

comprised by Data Name (DName), Location (Loc), and Aggregation Type (AggrType). DName refers to the name of data item, Loc indicates where it exists (i.e. city, state, or continent), and AggrType represents its statistic function (i.e. total number of case). Time Duration contains a pair of From and To time points. In addition, the Time Duration can be days, years, or any time granularities and may not represent the smallest granularity of time (i.e. time unit). Data Value (DValue) is the report value generated according to the function of AggrType. For example, the DValue 700 represents the total number of cases if I have the AggrType as Total\_cases. I use the epidemiology data set Tycho to test my system and the descriptions of Tycho.

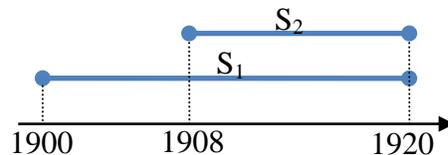
Aggregation of data from different resources has been exploited. These data may describe the same type of event but occurring at a different time. Table 2 shows an example of two sources for the same data reference with different data source reference, time duration, and data value. Assuming we consider measles cases from 1900 to 1920 from multiple sources  $S_1$  and  $S_2$ , Table 2 shows these integrated historical tuples reporting the total number of measles cases in LA from 10/10/1900 to 10/10/1920 and from 1/1/1908 to 10/10/1920 respectively.

**Table 2.** Example of an integrated historical tuple

Data Source Reference			Data Reference			Time Duration		Data Value
SI	SPD	DRD	DName	Loc	AggrType	From	To	DValue
$S_1$	9/9/1930	11/10/1920	Measles	LA	Total_cases	10/10/1900	10/10/1920	700
$S_2$	12/1/1930	11/10/1920	Measles	LA	Total_cases	1/1/1908	10/10/1920	1000

Figure 3 shows these two sources on the time series. Therefore, there should be one consistent data value for each time interval, and the value is identical across reports since these sources describe the same data reference even though they were collected from disparate resources. If the data value in each interval is inconsistent across reports, for example source  $S_1$

and  $S_2$  have an overlapping time duration but records contradict data values where the Dvalue for  $S_1$  is 700 but the Dvalue for  $S_2$  is 1000, then this inconsistency cannot be caught by traditional algorithms.

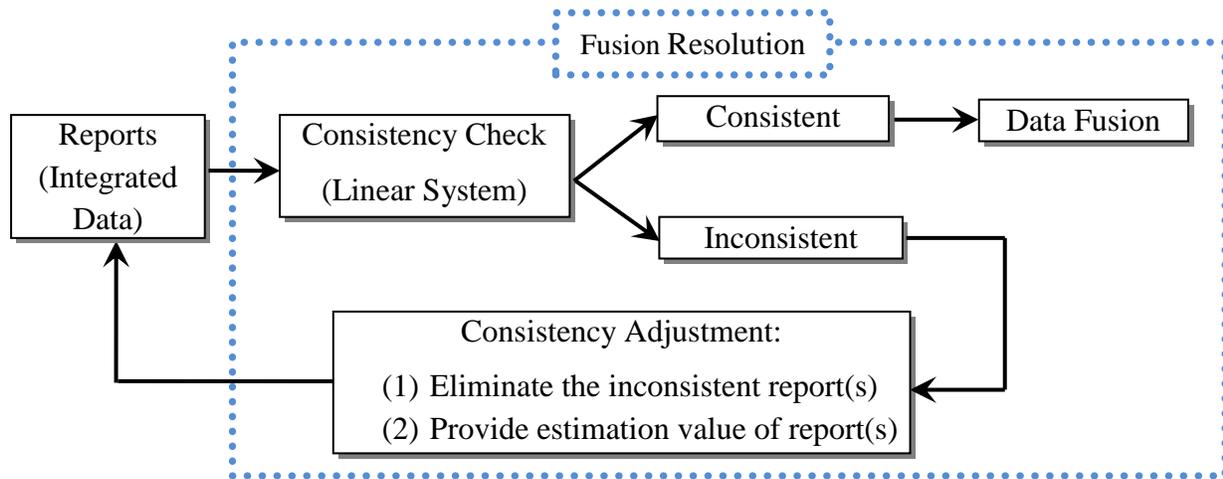


**Figure 3.** Example of two measles reports in overlapped time interval

### 3.2 PROPOSED APPROACH

Figure 4 shows a high-level overview of my proposed model. As an input the user expects an integrated data, or reports, having uniform homogeneous schema. If reports refer to the same data reference, any overlap in time or location between reports may cause an inconsistency, e.g., the total number of specific diseases in a specific location within a time duration is incompatible, or that number in specific time around a region is incompatible. Therefore, first the user has to perform a consistency check of the integrated data. The user can further perform data fusion if he/she cannot find any inconsistency; otherwise he/she needs to correct the reported values to make them consistent. The consistency adjustment aims to consolidate as many reports as possible under the presumption of consistent data. Therefore, the users can (1) eliminate the inconsistent report(s), or (2) adjust report values (using the solution

set generated from the nonnegative least squares method for characteristic linear system, as explained in the next section) and to modify the report values as little as possible.



**Figure 4.** Proposed data fusion model overview

### 3.2.1 Characteristic linear system

To provide inconsistency detection my model generates an underdetermined linear system corresponding to overlapping reports. The linear system is called *characteristic linear system* in this dissertation. After the system is built, the solution set for this it is generated by the nonnegative least squares method. The solution sets can be used to determine if these reports are inconsistent or to approximate reported interval values. The approach of solution set generation and inconsistency determination is called *reverse substitution (RS) method*. The goals of this method are to detect inconsistency occurrences and to provide proper values for each reported interval to mitigate diminish inconsistent data skewing the result. In this section I consider the

details of the characteristic linear system generating and solving along with related background theory. The reverse substitution method is introduced in Section 3.2.1 and 3.2.2.

When data sources are integrated, reports can be grouped in several linear systems depending on their overlapping conditions. The unknown variable vector  $\mathbf{X}$  represents unknown event density for each time interval as shown below,

$$\mathbf{X} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}.$$

The size of vector  $\mathbf{X}$  depends on the overlap condition of these reports; in other words,  $n$  is different for every linear system. The coefficient matrix  $\mathbf{A}$  denotes the existence of reports at corresponding time interval of  $\mathbf{X}$  is

$$\mathbf{A} = \begin{bmatrix} A_{11} & \cdots & A_{1n} \\ \vdots & \ddots & \vdots \\ A_{m1} & \cdots & A_{mn} \end{bmatrix},$$

where  $i \in (0, 1), i = 1 \dots m$  and  $j \in (0, 1), j = 1 \dots n$  for  $A_{ij}$ . The aggregated statistic value of reports as a constraint value vector  $\mathbf{b}$  is

$$\mathbf{b} = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}.$$

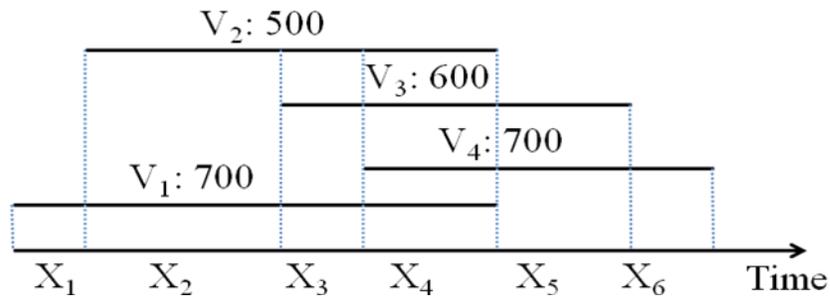
And the linear system represents as  $\mathbf{AX} = \mathbf{b}$ .

I am going to illustrate this approach with a simple example. Consider four reports from heterogeneous data sources of events with temporal overlapping (Table 3). The four reports represent the cases of pneumonia in Pennsylvania from epidemiological records in the 19th century.  $R_i$  represents report ID and the corresponding number of cases is denoted as  $V_i$  ( $i$ : report

ID). When we position these reports sequentially on the timeline by their occurrence time the timeline will be divided into smaller units of time intervals. The number of intervals is varied by overlapping condition of reports and range from 1 to  $2n-1$  ( $n$ : the number of report). There are six intervals in this example denoted as  $X_j$  ( $j$ : interval ID) in Figure 5. Ideally, redundant reports from heterogeneous data sources are consistent. Thus, each interval must have a non-negative value shared by all reports, and the sum of corresponding intervals will be equal to the sum of the reported values.

**Table 3.** Example of report overlapping

Report ID ( $R_i$ )	Disease	Location	From	To	Duration (year)	Report Value(v)
$R_1$	pneumonia	Pennsylvania	1900	1970	70	700
$R_2$	pneumonia	Pennsylvania	1920	1970	50	500
$R_3$	pneumonia	Pennsylvania	1940	1980	40	600
$R_4$	pneumonia	Pennsylvania	1950	1990	40	700



**Figure 5.** Example of four reports with overlapped time intervals

In this example, four reports divide the timeline into six intervals. The number of intervals depends on the number of reports and how they overlap. The above report configuration can be represented as the following underdetermined linear system:

$$\text{Max. } x_1 + x_2 + x_3 + x_4 + x_5 + x_6$$

$$\text{Subject to } x_1 + x_2 + x_3 + x_4 = 700$$

$$x_2 + x_3 + x_4 = 500$$

$$x_3 + x_4 + x_5 = 600$$

$$x_4 + x_5 + x_6 = 700$$

$$x_1, x_2, x_3, x_4, x_5, x_6 \geq 0.$$

The equations provide consistency constraints for the reported values. The unknown vector of interval values can be computed using nonnegative least squares method for this underdetermined system. In case of inconsistent reported values we will not be able to find nonnegative solutions of this characteristic linear system.

The above underdetermined linear system in matrix form  $\mathbf{AX} = \mathbf{b}$  is as follows:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} 700 \\ 500 \\ 600 \\ 700 \end{bmatrix}.$$

The matrix  $\mathbf{A}$  represents interval coefficients of equations where 1 represents the existence (coverage) of a given report for a specific time intervals and 0 indicates that corresponding report

does not cover that interval. Take the first row as an example, it shows this report covers from time interval  $x_1$  through  $x_4$  with report value 700 as recorded in Table 3. The unknown variable vector  $X$  represents values of a given report at each time interval, and the constraint value vectors  $b$  represents reported values. My goal is to generate a reasonable solution set for unknown vector  $X$ .

### 3.2.1.1 Underdetermined linear system

Considering the number of unknown variables ( $n$ ) and equations ( $m$ ), the system of linear equations will be described as an underdetermined linear system if there are more unknown variables than equations ( $n > m$ ). The linear system is overdetermined if there are less or unknown variables than equations ( $n < m$ ), or an exact if the number of unknown variables is equal to the number of equations ( $n = m$ ). Underdetermined linear systems naturally appear my data fusion problem with overlapping reports.

Another subarea of underdetermined linear system application uses sparse matrix to represent original data to reduce costs of transmission and storage space, capacity of information transmission channel, and complexity of computation. The  $k$ -term approximation selects  $k$  element to approximate the original data matrix ( $k < m$ ). “Coding with (this model) assumes the packing of  $k$ -sparse  $n$ -dimensional vectors in  $m$ -dimensional space...compressed sensing approach is an opportunity to reduce dimension(compression) of the data with a linear method even without solid knowledge about the data or the type of the basis providing sparse representation (Kozlov & Petukhov, 2010).” The goal of  $k$ -term approximation is to select which vectors should be kept/purged and minimize the approximation error with estimation and signal basis. A considerable amount of research is related to this task (Cevher & Guerra, 2008).

An example of efficient utilization of linear system is compressive sensing. The compressive sensing (i.e. compressed sensing) is the approach used in many areas such as image compression, data transmission, and signal processing to generate an underdetermined linear system of sampling recorded using much less data, and to reconstruct the original signal. There are two methods used to reconstruct the original signal in (Candès & Wakin, 2008)'s paper: the  $L_1$ -minimization (i.e.  $L_1$  norm) and the greedy algorithm. The  $L_1$ -minimization under the linear constraints can be considered as a linear programming problem.

In the real world, the case of underdetermined linear system is more popular compared with the overdetermined system. Other algorithms to solve linear system such as the Gauss-Jordan elimination is widely used, but cannot compute nonzero solution set only, and the Cramer's rule solves for square matrix only. For algorithms to solve linear system iteratively, they can be categorized in two areas of stationary iterative methods. The Jacobi method, the Gauss-Seidel method, the successive over-relaxation method, and the Krylov subspace method contain the conjugate gradient method, the generalized minimal residual method, and the biconjugate gradient method (Wikipedia). In addition, I would like to use less time points (i.e. in my case, only the start and end time of the report) to detect conflict because using this causes most cases in my system to be an underdetermined system. Therefore, I focus on solution set generation for underdetermined linear system in this dissertation. Some algorithms known as finding sparse solutions such as greedy algorithm, linear programming, or least squares algorithm are used to find solutions of underdetermined linear system. It is an NP-hard problem to find the sparsest solution for an underdetermined linear system (Natarajan, 1995). Given the wide selection of solution algorithms for the underdetermined linear system, I have investigated which method is more suitable for my needs. I am going to explain more of the nonnegative least

squares algorithm and the comparison with other methods such as linear programming in the following Section 3.2.1.2 and Section 3.2.1.3.

### 3.2.1.2 Linear programming

Linear programming is a method used to determine the optimal solutions to maximize the profit or minimize the cost for a model that includes linear equations representing a list of restriction or requirements. “The word linear suggests that feasible plans are restricted by linear constraints (inequalities), and also that the quality of the plan (e.g., costs or duration) is also measured by a linear function of the considered qualities.” (Matoušek & Gärtner, 2007) The linear programming model can be applied in many areas such as investment planning in economic analysis, resource allocation for engineering problems, the salesman traveling problem in logistical algorithm, genome analysis in biological problems, and most popular, profit/cost estimation in industry problems. In my model, I want to minimize the difference between solution sets and the real report values for each interval, which can be referred to as restrictions for these linear equations. The linear programming (with  $m$  constraints and  $n$  variables) is shown below, and each row is linearly independent from each other:

$$\begin{array}{ll}
 \text{Max (or Min)} & C_1x_1 + C_2x_2 + \dots + C_nx_n \\
 \text{Subject to} & a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \leq b_1 \\
 & a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \leq b_2 \\
 & \vdots \\
 & a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \leq b_m \\
 & x_1, x_2, \dots, x_n \geq 0, \quad b_1, b_2, \dots, b_m \geq 0,
 \end{array}$$

The linear programming shown in canonical form is:

$$\mathbf{C} = [C_1 \ C_2 \ \dots \ C_n], \mathbf{A} = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \dots & a_{mn} \end{bmatrix}, \mathbf{X} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \text{ and } \mathbf{b} = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}.$$

Here  $\mathbf{A} \in \mathbb{R}^{m \times n}$  is a  $m \times n$  matrix,  $\mathbf{X} \in \mathbb{R}^n$  is the vector of unknown variables,  $\mathbf{b} \in \mathbb{R}^m$  and  $\mathbf{C} \in \mathbb{R}^n$  are given vectors. There are three conditions that must be met: (1) all constraints must be stated as equalities of the form  $\mathbf{AX} = \mathbf{b}$  where  $\mathbf{AX}$  is a linear function of  $\mathbf{X}$ , (2) the right hand side for each constraint must be nonnegative, i.e.  $\mathbf{b} \geq 0$ , (3) all variables must be nonnegative, i.e.  $\mathbf{X} \geq 0$ .

Then the linear programming can be described as

$$\text{Max (or Min) } \mathbf{C}^T \mathbf{X},$$

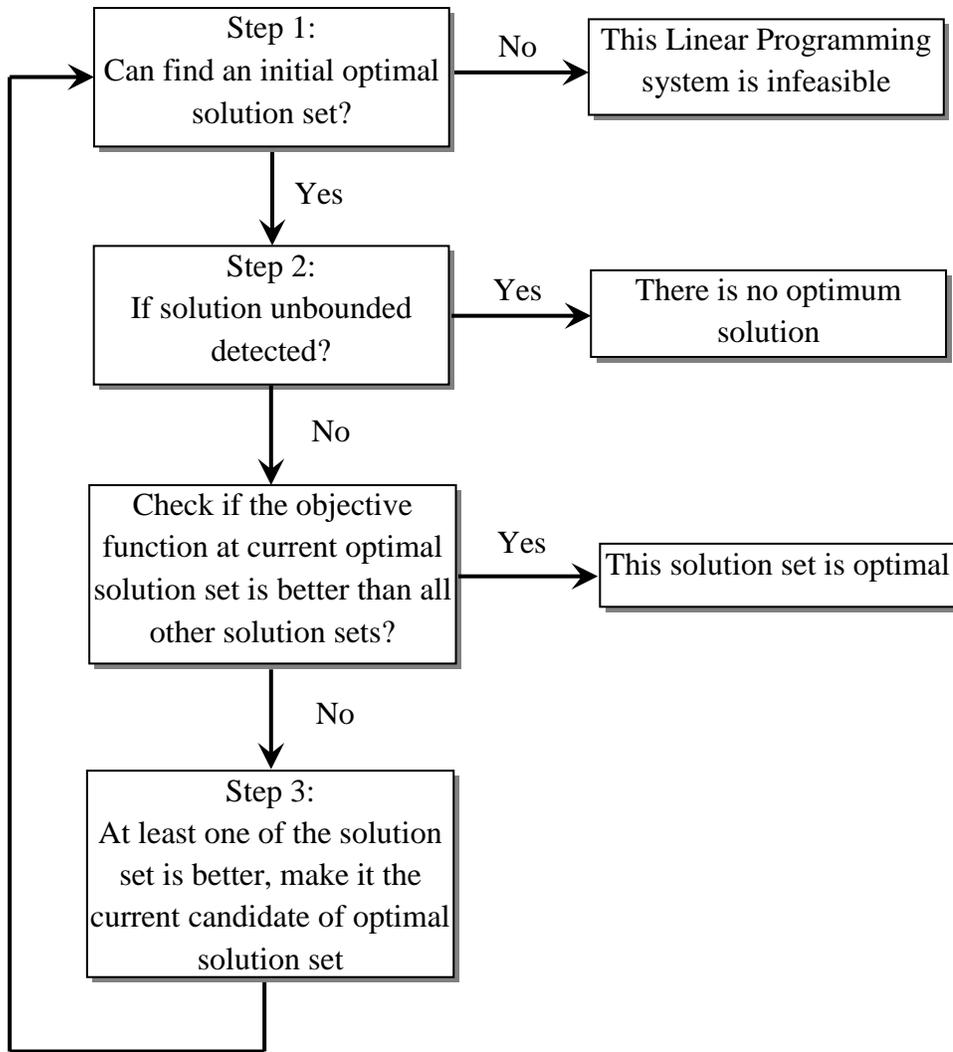
$$\text{S. t. } \mathbf{AX} \leq \mathbf{b}, \mathbf{X} \geq 0,$$

where  $\mathbf{b} \geq 0$ ,  $\mathbf{0}$  is the zero vector.

Any  $\mathbf{X} \in \mathbb{R}^n$  to the augmented system that satisfies these linear constraints the non-negativity are *feasible solutions* and when the vector  $\mathbf{X}$  reaches the maximum or minimum value of the given objective function (i.e.  $\text{Max (or Min) } \mathbf{C}^T \mathbf{X}$ ) it is the *optimal solution*. Note that the augmented system above does not have to include the nonnegative conditions. Also note that this system has these possible states: (1) feasible with a unique optimum solution, (2) feasible with infinitely many optimal solutions, (3) feasible with acceptable solutions because the objective function is unbounded, (4) infeasible and no optimum solution. If the solution set for the objective function is as well as or better than other solution sets, then this solution set is optimal for the linear programming. I use an iterative procedure listed below, to find the optimal solution and a detailed flow chart is shown in Figure 6.

Step one is to find an initial optimal solution set and make it the current candidate. If one cannot be found, the Linear Programming is infeasible. Step two checks if the solution unboundedness is detected. If yes, then there is no optimum solution. If no, check if the objective function at the current optimal solution set is at least as good as or better than all of its other solution sets. If yes, then this solution set is optimal and stop iteration; otherwise, go to Step three. Step three: if at least one of the solution sets is better, make it the current candidate and go to Step one.

Each linear programming system falls into one of three possible statuses: (1) no solution, (2) exactly one solution, or (3) infinitely many solutions, and hence a single optimal solution or none. Every feasible solution satisfies constraint of the objective function and all constraint equalities. In addition, it provides a bound of optimal solution until the single optimal solution is generated by the iterative procedure. However, this single optimal solution may reach the objective function  $Max (or Min) C^T X$  but the difference under all constraint equalities may also be large. Another restriction I have for data fusion is to find an optimal solution set of integer values. This constraint is common in scenarios such as case of death, hiring worker number, or purchase equipment amount. However, solving an integer programming system is more difficult than normal non-integer-restriction programming systems in computation and generating optimal solutions. For the cost of computation, I do not include integer constraint in this algorithm.



**Figure 6.** Iterative procedure

The linear programming system can be presented in an inequality form or equation form within its requirements. The equation form of linear programming shown below is also known as the *nonnegativity constraints* (Matoušek & Gärtner, 2007):

$$\begin{aligned} & \text{Max (or Min) } \mathbf{C}^T \mathbf{X}, \\ & \text{S. t. } \mathbf{AX} = \mathbf{b}, \mathbf{X} \geq 0, \text{ where } \mathbf{b} \geq 0. \end{aligned}$$

There are many algorithms for finding optimal solutions for the linear programming system such as the basis exchange method Simplex algorithm (George Dantzig,1947) and Criss-cross algorithm; the interior point method Ellipsoid Algorithm, Projective algorithm, and Path-following Algorithm; and the branch and cut method. The Simplex algorithm finds feasible solution sets by determining vertices of edges of feasible region plants. Similar to the Simplex algorithm, the Criss-cross algorithm is a basis-exchange algorithm, but have loose restriction of feasible solution sets. Interior point methods such as Ellipsoid algorithm, Projective algorithm, and Path-following algorithm were developed to finds feasible solutions for minimizing convex functions for worst-case polynomial-time solutions (Wikipedia).

Here is an example of the Simplex algorithm which is one of the most widely used algorithms that uses iteration procedure from one solution set of the feasible polyhedron to another set in order to find the unique feasible optimal solution set:

$$\text{Max } x_1+x_2+x_3+x_4$$

$$\text{S.t. } x_1+x_2 = 16$$

$$x_2+x_3 = 25$$

$$x_3 + x_4 = 17$$

$$x_1, x_2, x_3, x_4 \geq 0.$$

In the form of a matrix,  $\mathbf{C} = [1 \ 1 \ 1 \ 1]$ ,  $\mathbf{A} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$ ,  $\mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$  and  $\mathbf{b} = \begin{bmatrix} 16 \\ 25 \\ 17 \end{bmatrix}$ . Letting

the variable equal 0, i.e.  $x_1 = 0$ , the first equation results in  $x_2 = 16$ . Substituting  $x_1, x_2$  to other equations then it generates  $x_3 = 9$  and  $x_4 = 8$ . This solution set is feasible and optimal compared with other candidate solution sets. Given the solution generating methods for the underdetermined system provided above, the restriction of simplex algorithm in linear

programming is that it generates solutions from zero and stops when it finds a feasible solution. I assume the intervals for each report has a consistent value. In addition, I would like to make the generated solution set closer to the consistent report values. Therefore, I adopt the nonnegative least squares method, which will be described in following Section 3.2.1.3.

### 3.2.1.3 Nonnegative least squares method

The least squares method is an algorithm for solving linear equations and widely used in statistics, signal processing, or optimization. In general, the least squares method is used to find a solution that minimize errors in an overdetermined system, the system that having more equations than unknown variable and has no solution. It also includes the following algorithms: the nonnegative least squares algorithm, the least-square primal-dual algorithm, the least-square network flow algorithm, and combined-objective least-square algorithm, etc (Kong, 2007). In addition, we can also use the least squares method to find a solution or find solutions in underdetermined linear system that has more unknown variables than equations. We can also find infinite solutions if we pick the solution that has smallest errors (Horn, n.d.). To calculate two vectors' distance, similarity, or fitness, the least squares method is complemented by the  $L_1$ -norm and the  $L_2$ -norm. "The  $L_1$ -norm  $\|\mathbf{X}\|_1$  is that if  $\mathbf{X}$  is a vector with complex components  $x_1, x_2, \dots, x_n$ , then  $\|\mathbf{X}\|_1 = \sum_{r=1}^n |x_r|$ ; The  $L_2$ -norm  $\|\mathbf{X}\|_2$  or the Euclidean norm means if  $\mathbf{X}$  is a vector with complex components  $x_1, x_2, \dots, x_n$ , then  $\|\mathbf{X}\|_2 = (\sum_{r=1}^n |x_r|^2)^{1/2}$ " (Jeffrey & Zwillinger, 1971).

The nonnegative least squares method is one of the least squares methods with a specific constraint of nonnegative solutions. It solves linear programming systems by QR-factorization that adds/deletes a column iteratively and updates the R matrix (Kong, 2007). The model can be written as

$$\text{Min } \|b - AX\|_2^2$$

$$\text{s. t. } AX = b$$

$$X \geq 0.$$

Or it can be written similar to the Simplex algorithm (Phase I) that minimizes the  $L_2$  norm of the residual  $\rho$

$$\text{Min } \sum |\delta_i|$$

$$\text{s. t. } AX + \delta = b$$

$$X \geq 0.$$

“The nonnegative least squares algorithm was introduced by (Lawson & Hanson, 1974) and was used to solve the Phase I problem in linear programming in (Davis & Dantzig, 1992)”(Kong, 2007).  $X$  is the solution set of the linear programming system. The paper of (Horn, n.d.) shows how to solve underdetermined linear system and is briefly described here.

To find the minimal solution set  $X$  of system  $AX = b$ , the Lagrange multipliers is used to add a term to the equations to minimize

$$\|X\|^2 + \lambda^T (b - AX).$$

Differentiating with respect to  $\mathbf{X}$  and setting the result to zero we get

$$2\mathbf{X} - \mathbf{A}^T \boldsymbol{\lambda} = 0.$$

Multiply by  $\mathbf{A}$  therefore

$$2\mathbf{AX} - \mathbf{AA}^T \boldsymbol{\lambda} = 0.$$

Replace  $\mathbf{AX}$  by  $\mathbf{b}$ ,

$$2\mathbf{b} - \mathbf{AA}^T \boldsymbol{\lambda} = 0,$$

$$2\mathbf{b} = \mathbf{AA}^T \boldsymbol{\lambda}, \text{ and}$$

$$\boldsymbol{\lambda} = 2(\mathbf{AA}^T)^{-1} \mathbf{b}.$$

Therefore the solution set is

$$\mathbf{X} = \mathbf{A}^T (\mathbf{AA}^T)^{-1} \mathbf{b}$$

and  $\mathbf{A}^T (\mathbf{AA}^T)^{-1}$  called as a pseudo-inverse.

The nonnegative least squares method iterates to reach an acceptable approximation or an optimal solution. Each iteration of the nonnegative least squares tries to *minimize*  $\|\mathbf{b} - \mathbf{AX}\|_2^2$ .

To prove that  $\mathbf{X} = \mathbf{A}^T (\mathbf{AA}^T)^{-1} \mathbf{b}$  is the optimal solution that is the minimum  $\|\mathbf{b} - \mathbf{AX}\|_2^2$ , we

assume the optimal solution as  $\mathbf{X}'$  where  $\mathbf{X}' = \mathbf{A}^T (\mathbf{AA}^T)^{-1} \mathbf{b}$ . If we can find the optimal solution

makes

$$\mathbf{A}(\mathbf{X} - \mathbf{X}') = 0 \text{ and}$$

$$(\mathbf{X} - \mathbf{X}')^T \mathbf{X}' = (\mathbf{X} - \mathbf{X}')^T \mathbf{A}^T (\mathbf{A} \mathbf{A}^T)^{-1} \mathbf{b} = (\mathbf{A}(\mathbf{X} - \mathbf{X}'))^T (\mathbf{A} \mathbf{A}^T)^{-1} \mathbf{b} = \mathbf{0}$$

Therefore

$$\mathbf{X} - \mathbf{X}' \perp \mathbf{X}'$$

and

$$\|\mathbf{X}\|^2 = \|\mathbf{X}' + \mathbf{X} - \mathbf{X}'\|^2 = \|\mathbf{X}'\|^2 + \|\mathbf{X} - \mathbf{X}'\|^2 \geq \|\mathbf{X}'\|^2.$$

Therefore,  $\mathbf{X}'$  is the optimal solution that minimize  $\|\mathbf{b} - \mathbf{A}\mathbf{X}\|_2^2$ .

The nonnegative least squares method generates the solution set that minimizes the difference between the actual values and estimated values by iteration. To empirically prove the viability of the nonnegative least squares method, I am going to implement it in simulations in the following sections.

### 3.2.2 Reverse substitution method: inconsistency detection

There are many solution generation methods for underdetermined linear system. In this dissertation, I use the nonnegative least squares method to solve the linear system  $\mathbf{A}\mathbf{X} = \mathbf{b}$  and

the solution sets of unknown variables  $\mathbf{X}$  can be computed by  $\mathbf{X} = \mathbf{A}^T (\mathbf{A} \mathbf{A}^T)^{-1} \mathbf{b}$ . The

underdetermined condition is more common in reports modeling for the characteristic of linear system generation when there is a high degree of overlapping. The theory of the underdetermined linear system and how to solve it was explained in detail in the previous

sections. I use  $\mathbf{X}'$  to indicate the solution set value computed by the nonnegative least squares method. Below I illustrate computing  $\mathbf{X}'$  for the example in Figure 5 of Section 3.2.1;

$$\mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{b} = \mathbf{X}' = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} 200 \\ 0 \\ 0 \\ 500 \\ 100 \\ 100 \end{bmatrix}.$$

Then, I substitute  $\mathbf{X}'$  in the original equation to obtain consistent values  $\mathbf{b}'$ . The matrix  $\mathbf{b}'$  generated by the solution set  $\mathbf{X}'$  is

$$\mathbf{A}\mathbf{X}' = \mathbf{b}' = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} = \begin{bmatrix} 700 \\ 500 \\ 600 \\ 700 \end{bmatrix}.$$

I compare this value of  $\mathbf{b}'$  with the original value  $\mathbf{b}$ . This process of checking the difference between estimated value and actual values is called the *reverse substitution method (RS)* in my dissertation. Consider the solution set for my example from Table 3: Ideally the report values are consistent if  $\mathbf{b}' = \mathbf{b}$  and thus delta ( $\delta$ ) equals zero in  $\mathbf{A}\mathbf{X} + \delta = \mathbf{b}$ ,

$$\delta = \begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \end{bmatrix} = \begin{bmatrix} b'_1 - b_1 \\ b'_2 - b_2 \\ b'_3 - b_3 \\ b'_4 - b_4 \end{bmatrix} = \begin{bmatrix} 700 - 700 \\ 500 - 500 \\ 600 - 600 \\ 700 - 700 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

These  $\delta$  values can be zero or nonzero; zero  $\delta$  indicates that the reports are consistent, but nonzero  $\delta$  values give us a warning of inconsistent reports. After generating  $\delta$  values of the underdetermined linear system, the temporal data fusion system performs further analysis using

these  $\delta$  values and the number-of-conflict-report( $C$ ) values to point out which reports have a higher potential to cause inconsistency (i.e. the nonzero  $\delta$  value).

I use the following example to show how to use the  $\delta$  values and the  $C$  values to point out which reports have the higher potential to cause inconsistency. If I manipulate the report values of  $R_2$  from 500 to 900 to introduce some inconsistency as shown in Table 4, Report  $R_2$  has shorter time duration but higher value of report compare with  $R_1$ . This condition will be valid if one report partly overlies on the other report, but it will be conflicting if one report is subsumes the other.

**Table 4.** Example of inconsistent report values

Report ID ( $R_i$ )	Disease	Location	From	To	Duration (year)	Report Value ( $V_i$ )
$R_1$	pneumonia	Pennsylvania	1900	1970	70	700
$R_2$	pneumonia	Pennsylvania	1920	1970	50	<b>900</b>
$R_3$	pneumonia	Pennsylvania	1940	1980	40	600
$R_4$	pneumonia	Pennsylvania	1950	1990	40	700

The new linear system is  $\mathbf{AX} = \mathbf{b}$

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} 700 \\ 900 \\ 600 \\ 700 \end{bmatrix}.$$

And the value for each interval generated by nonnegative least squares method is

$$\mathbf{X}' = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} 0 \\ 200 \\ 0 \\ 600 \\ 0 \\ 100 \end{bmatrix}.$$

The corresponding matrix  $\mathbf{b}'$  is

$$\mathbf{AX}' = \mathbf{b}' = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} = \begin{bmatrix} 800 \\ 800 \\ 600 \\ 700 \end{bmatrix}.$$

$\mathbf{AX}' = \mathbf{b}' \neq \mathbf{b}$  since the nonnegative least squares method cannot find a feasible solution for reported values  $\mathbf{b}$ . Therefore the difference  $\delta$  is

$$\delta = \begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \end{bmatrix} = \begin{bmatrix} 700 - 800 \\ 900 - 800 \\ 600 - 600 \\ 700 - 700 \end{bmatrix} = \begin{bmatrix} -100 \\ 100 \\ 0 \\ 0 \end{bmatrix}.$$

If we only consider the largest subset which contains all reports, some consistent reports may be punished by a nonzero  $\delta$ . The reason is that the nonnegative least squares method can generate interval values, which satisfy most equalities, but cannot find a perfect solution set for all inconsistent equalities. Thus, I search through all subsets of reports from the largest subset (includes all reports) to the smallest subset (includes only one report). Some reports have nonzero  $\delta$  values inside of larger subsets, but not inside of smaller subsets. By comparing the list of conflict report to non-conflict list of reports, I can find the exact list of reports that are in conflict with one specific report. For example, the large subset contains reports  $\{R_1, R_2, R_3\}$  which has nonzero  $\delta$  value. The smaller subsets contain reports  $\{R_1, R_2\}$  and  $\{R_2, R_3\}$  all with

zero  $\delta$  value, but the other subset contains reports  $\{R_1, R_3\}$  with nonzero  $\delta$  value. Therefore, I can deduce that inconsistency only exists in reports  $\{R_1, R_3\}$ .

Using the  $\delta$  vector I define the largest subset listed above as

$$\delta^1 = \begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \end{bmatrix} = \begin{bmatrix} -100 \\ 100 \\ 0 \\ 0 \end{bmatrix}.$$

By summing up the absolute value of  $\delta^s$  ( $s$ : subset ID) across all subsets I can get

$$\delta^{total} = \begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \end{bmatrix} = \begin{bmatrix} 1600 \\ 2000 \\ 600 \\ 0 \end{bmatrix}.$$

$\delta^s$  can be used to indicate the existence of inconsistency between reports because when the merged data is inconsistent, I will not be able to find identical values for each time interval. If the non-negative least squares method cannot find identical values for each time interval between reports, or there is no feasible solution, the merged data contain report values that conflict with others. The matrix  $\mathbf{b}'$  generated by solution set  $\mathbf{X}'$  cannot satisfy all linear equations in this underdetermined linear system, and this is reflected in the nonzero  $\delta$  when  $\mathbf{b}' \neq \mathbf{b}$ .

In order to identify the existence of conflict between reports and the IDs of conflict reports, I consider both  $\delta$  and the  $C$  values for each report of all subsets. In this example, I have  $C_4^1 + C_4^2 + C_4^3 + C_4^4 = 15$  combinations of report subset for four reports. Each report has a delta

value  $\delta_i^s$  and a number-of-conflict-report ( $c_i^s$ ) in each subset. After calculation, there are only four subsets with nonzero  $\delta$  shown in Table 5.

**Table 5.** Subsets with nonzero delta

Vector	Subset			
	Subset ID: 1 R <sub>1</sub> , R <sub>2</sub> , R <sub>3</sub> , R <sub>4</sub>	Subset ID: 2 R <sub>1</sub> , R <sub>2</sub> , R <sub>3</sub>	Subset ID: 3 R <sub>1</sub> , R <sub>2</sub> , R <sub>4</sub>	Subset ID: 6 R <sub>1</sub> , R <sub>2</sub>
$\mathbf{X}^{(s)}$	$[0 \ 200 \ 0 \ 600 \ 0 \ 100]^T$	$[0 \ 0 \ 0 \ 0 \ 0 \ 0]^T$	$[0 \ 100 \ 0 \ 700 \ 0 \ 0]^T$	$[0 \ 0 \ 0 \ 0 \ 0 \ 0]^T$
$\mathbf{b}^{(s)}$	$[800 \ 800 \ 600 \ 700]^T$	$[0 \ 0 \ 0]^T$	$[800 \ 800 \ 700]^T$	$[0 \ 0]^T$
$\delta^s$	$[-100 \ 100 \ 0 \ 0]^T$	$[700 \ 900 \ 600]^T$	$[-100 \ 100 \ 0]^T$	$[700 \ 900]^T$

The vector  $\mathbf{X}^{(s)}$  represents the computed interval value for the subset  $s$ ; the vector  $\mathbf{b}^{(s)}$  is the report value for the subset  $s$ , and the difference vector  $\delta^s = \mathbf{A}\mathbf{X}^{(s)} - \mathbf{b}^{(s)}$ . Any nonzero  $\delta^s$  indicates that the report values are inconsistent in the subset. If the report has one subset containing inconsistent values, then I set the  $C$  value as one for this report such as in subset 1, the  $\delta^1 = [\delta_1^1 \ \delta_2^1 \ \delta_3^1 \ \delta_4^1]^T = [-100 \ 100 \ 0 \ 0]^T$  and the  $c^1 = [c_1^1 \ c_2^1 \ c_3^1 \ c_4^1]^T = [1 \ 1 \ 1 \ 1]^T$ . By going through all subsets with zero  $\delta$  and nonzero  $\delta$ , I list the conflicting condition for each report:

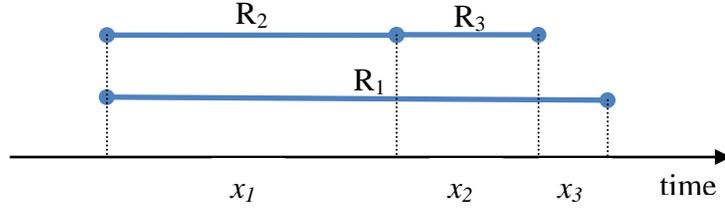
**Table 6.** Find conflict report ID

Report ID	potential conflict report list	Non-Conflict report list	Difference
<b>R<sub>1</sub></b>	R <sub>2</sub> , R <sub>3</sub> , R <sub>4</sub>	R <sub>3</sub> , R <sub>4</sub>	R <sub>2</sub>
<b>R<sub>2</sub></b>	R <sub>1</sub> , R <sub>3</sub> , R <sub>4</sub>	R <sub>3</sub> , R <sub>4</sub>	R <sub>1</sub>
<b>R<sub>3</sub></b>	R <sub>1</sub> , R <sub>2</sub> , R <sub>4</sub>	R <sub>1</sub> , R <sub>2</sub> , R <sub>4</sub>	∅
<b>R<sub>4</sub></b>	R <sub>1</sub> , R <sub>2</sub> , R <sub>3</sub>	R <sub>1</sub> , R <sub>2</sub> , R <sub>3</sub>	∅

Take  $R_1$  as an example, there is a non-zero  $\delta$  value in subset 1, therefore the potential conflict report list for  $R_1$  is  $R_2$ ,  $R_3$  and  $R_4$ . However, other smaller subsets (i.e. Subset 4:  $R_1$ ,  $R_3$ ,  $R_4$ ; subset 7:  $R_1$ ,  $R_3$ ,... etc.) have zero  $\delta$  values for  $R_1$ . Therefore,  $R_1$  is not in conflict with these reports. After comparing the potential conflict report list with the non-conflict report list, I found that  $R_1$  is only in conflict with  $R_2$ , but not in conflict with reports  $R_3$  and  $R_4$ . The  $C$  value and the  $\delta$  value in each subset have a notable impact on indication of conflict report ID. The  $C^{total} = [4\ 4\ 2\ 2]^T$  represents the summation of the  $C$  value across all subsets. These  $\delta$  values,  $\delta^1$ ,  $\delta^2$ ,  $\delta^3$ , and  $\delta^6$  for corresponding subsets are the only nonzero  $\delta$  values for all combination of subsets. In addition, I observe that the  $C$  value and nonzero  $\delta$  value only occur in subsets that contain  $R_1$  and  $R_2$ . The  $C^{exact} = [1\ 1\ 0\ 0]^T$  represents the  $C$  value for each report after excluding the conflict subsets caused by indirect conflict (i.e. in subset 2,  $R_3$  has non-zero  $\delta$  value because it is inside the subset with real conflict reports  $R_1$  and  $R_2$ ). Thus, comparing all subsets can help users find the exact inconsistent reports. The values of  $C$  and  $\delta$  provide information of the report consistency for each subset. The higher the value of  $C$  of a report

indicates the higher the conflict is between this report with other reports. In this example,  $R_1$  and  $R_2$  have the highest potential for inconsistency, and they are actually contradicted in report values.

Here I present another example to illustrate the  $C$  value.



**Figure 7.** Example of number-of-conflict-report

The linear system for Figure 7 is

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = \begin{bmatrix} 5 \\ 20 \\ 10 \end{bmatrix}.$$

The nonnegative least squares method generated solution set, the corresponding report value, the  $\delta$  value, and the  $C$  value are given as

$$\mathbf{X}' = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 11.667 \\ 1.667 \\ 0 \end{bmatrix}, \mathbf{b}' = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 13.333 \\ 11.667 \\ 1.667 \end{bmatrix}, \boldsymbol{\delta}^{total} = \begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \end{bmatrix} = \begin{bmatrix} 18 \\ 16 \\ 11 \end{bmatrix}, \text{ and } \mathbf{C}^{total} = \begin{bmatrix} C_1 \\ C_2 \\ C_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \\ 2 \end{bmatrix}.$$

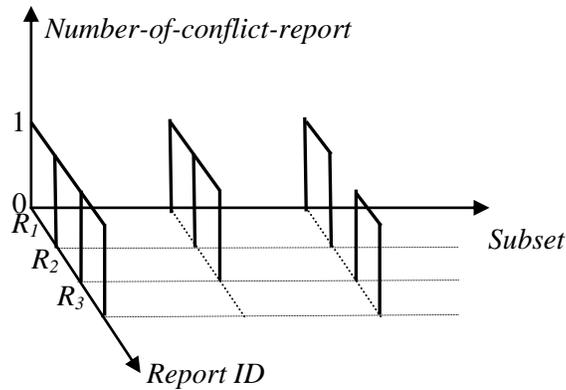
In this case, both  $R_2$  and  $R_3$  have inconsistent report values with  $R_1$ , which cause  $R_1$  to have the highest  $C$  value. Therefore, I expect the  $\delta$  value is higher for  $R_1$  because it has a higher  $C$  value (i.e. conflict with many other reports). The total number of conflict  $\mathbf{C}^{total}$  indicates

which report causes problems inside the system. Therefore the nonzero  $C$  value can be used as an indicator of the occurrence of conflict, and its value represents the critical level of conflict.

Figure 8 shows the  $C$  value accumulated across nonzero  $C$  value subsets. Subset 1 contains report  $R_1$ ,  $R_2$  and  $R_3$  with nonzero  $C$  value, Subset 2 contains nonzero  $C$  value with  $R_1$  and  $R_2$ , and Subset 3 contains nonzero  $C$  value with  $R_1$  and  $R_3$ . Therefore the summation of these  $C$

values across subsets is  $C^{total} = \begin{bmatrix} C_1 \\ C_2 \\ C_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \\ 2 \end{bmatrix}$ . The number of subsets with nonzero  $C$  value

depends on the linear system. My evaluations of the  $C$  values inside the  $C^{total}$  vector indicate that it can accurately represent the reliability of these reports.



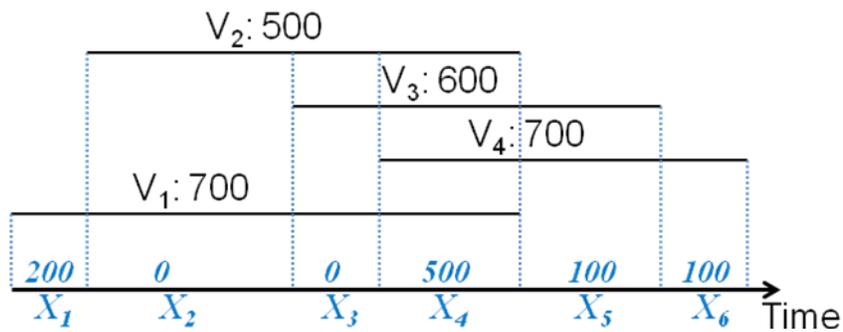
**Figure 8.** Example of nonzero number-of-conflict-report subsets

### 3.2.3 Reverse substitution method: data fusion

In the previous section I showed the proposed RS method that can be used for inconsistency detection. In this section I will illustrate how the RS method can be used for efficient data fusion.

Specifically, I will use generated solutions for characteristic linear system to estimate values from historical reports. The nonnegative least squares method iterates to reach an acceptable approximation or an optimal solution. Each iteration of the nonnegative least squares tries to minimize  $\|b - AX\|_2^2$ . The generated optimal solution set provides us the reference for data fusion. I use the same examples from previous sections to explain how to use the RS method for data fusion. Under consistent conditions, the example in Section 3.2.1 that report values are 700, 500, 600, and 700 for  $R_1$ ,  $R_2$ ,  $R_3$ , and  $R_4$ . My proposed RS method first shows there is no inconsistency with these reports since the  $\delta$  values of the corresponding characteristic linear system are zero. Second, the RS method suggests potential case values for each time intervals are 200, 0, 0, 500, 100, and 100 by iterating the equation for optimize solution set where

$$A^T(AA^T)^{-1}b = X' = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} 200 \\ 0 \\ 0 \\ 500 \\ 100 \\ 100 \end{bmatrix}$$



**Figure 9.** Example of data fusion and values for each time interval

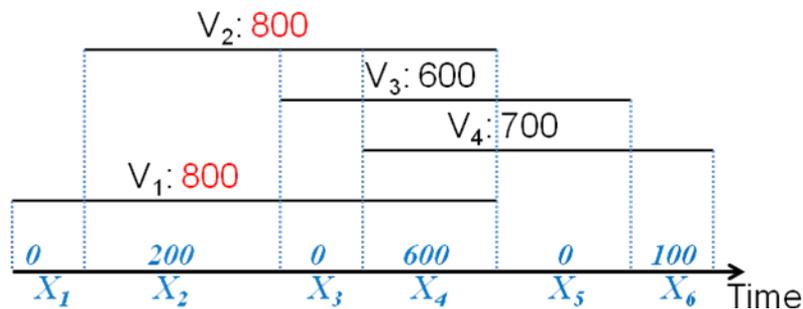
The generated solution set provides users a general idea of possible value for each time interval. The estimated interval values sometimes are too arbitrary since some interval values are zero; however, the accuracy of the estimated values can be improved by increasing the number of reports or overlapping of the report structure. Under inconsistent conditions, the example in Section 3.2.2 that report values are 700, 900, 600, and 700, but the actual values are 700, 500, 600, and 700 respectively. And the value for each interval generated by nonnegative least squares method is

$$\mathbf{X}' = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} 0 \\ 200 \\ 0 \\ 600 \\ 0 \\ 100 \end{bmatrix}.$$

The corresponding matrix  $\mathbf{b}'$  is

$$\mathbf{AX}' = \mathbf{b}' = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} = \begin{bmatrix} 800 \\ 800 \\ 600 \\ 700 \end{bmatrix}.$$

Figure 10 shows the new values of reports and time intervals.



**Figure 10.** Example of inconsistent reports and interval values

$AX' = b' \neq b$  since the nonnegative least squares method cannot find a feasible solution for report value  $b$ . Therefore the difference  $\delta$  is

$$\delta = \begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \end{bmatrix} = \begin{bmatrix} 700 - 800 \\ 900 - 800 \\ 600 - 600 \\ 700 - 700 \end{bmatrix} = \begin{bmatrix} -100 \\ 100 \\ 0 \\ 0 \end{bmatrix}.$$

In this example, the  $R_1$  and  $R_2$  have equal probability to cause inconsistency according to the  $\delta$  matrix. I randomly select  $R_1$  as the report that has erroneous and adjust its report value to 700. Therefore, the report set will be adjusted as:

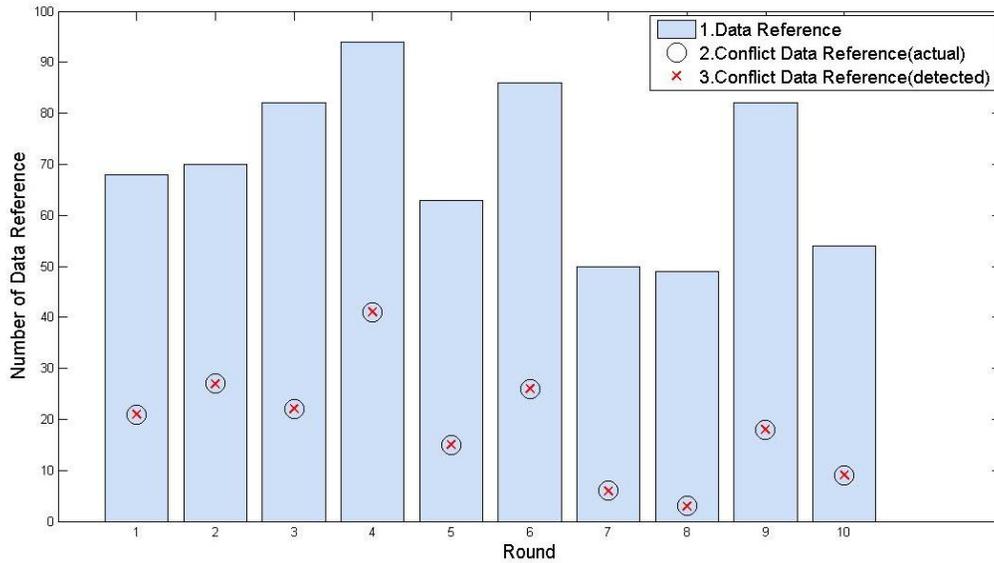
$$\begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} = \begin{bmatrix} 700 \\ 700 \\ 600 \\ 700 \end{bmatrix}.$$

By decreasing the degree of freedom, or in other words, increasing the number of reports, I can find report sets with increased accuracy.

### 3.3 STUDY 1 – INCONSISTENCY DETECTION IN REAL DATA

I implement the proposed approach and apply it to both simulated and real data sets in study 1. In the simulation, I generate the actual inconsistent data references and the number of reports randomly. In each simulation run, the number of actual inconsistent data references is randomly chosen within the range of the total data references. The number of data references and the number of reports for each data reference are also randomly generated between 0 and 100. The result of this simulation in Figure 11 shows that the number of inconsistent reports detected by

the proposed RS method (indicated with x) and the number of actual inconsistent reports (indicated with circles) matches under different configurations of conflicts/reports/data references. We observe that, my proposed RS method was able to detect accurately the occurrences of data inconsistency.



**Figure 11.** Simulation result of inconsistency detection

For real data set, I have tested my algorithm on the Tycho database. The integrated Tycho repository, an integrated epidemiological data warehouse that records diseases information from heterogeneous data sources, has 1,826,583 reports. The Tycho database describes the epidemiology reports for more than 100 years from 01-Jan-1895 to 03-Nov-2001. This data is collected in the School of Public Health at the University of Pittsburgh. Each disease was described by multiple reports of different time durations (i.e. weeks). Therefore I have about 9,416 data references in which each data reference contains information of a given disease in a given location reported at different times. I perform the simulation with Matlab environment version 7.12.0.635 (R2011a) 32-bit.

The Tycho database I worked with contains nineteen diseases with different outcomes (a case or a death). Each row in the disease-reference-table consists of disease ID, disease name, empty column, city, state, start number of the data reference report, end number of the data reference report, and number of data reference report as shown in Figure 12. Take row 1 as an example; it shows data reference ID#1, which is a case of brucellosis in New York City, NY. The reports about this disease start from row 1 to row 14 with 14 reports in data-number-table.

	1	2	3	4	5	6	7	8
1	1	'brucellosis'	[]	'NEW YORK'	'NY'	1	14	14
2	2	'chickenpox'	[]	'ABERDEEN'	'SD'	15	413	399
3	3	'chickenpox'	[]	'ALBUQUER...	'NM'	414	840	427
4	4	'chickenpox'	[]	'ATLANTA'	'GA'	841	1290	450
5	5	'chickenpox'	[]	'BALTIMORE'	'MD'	1291	1746	456

**Figure 12.** Tycho disease reference table

The data-number-table contains reports of each disease at various times in a specific location. It contains disease ID, start date of the report, end date of the report, number of cases, date when the report was published and sequence number as shown in Figure 13. For example, row 1 shows the first report of data reference ID#1 starts from `datestr(718097)=30-Jan-1966` to `datestr(718103)=05-Feb-1966` with one case and the date of publishing `datestr(718109)=11-Feb-1966`.

data_num <1826583x6 double>						
	1	2	3	4	5	6
1	1	718097	718103	1	718109	1
2	1	718125	718131	1	718137	2
3	1	718209	718215	1	718221	3
4	1	718433	718439	2	718445	4
5	1	719504	719510	1	719516	5
6	1	719833	719839	953	719845	6
7	1	720400	720406	1	720419	7
8	1	720575	720581	1	720587	8
9	1	721240	721246	1	721259	9
10	1	722157	722163	1	722162	10
11	1	723130	723136	1	723142	11
12	1	723144	723150	1	723156	12
13	1	723165	723171	1	723184	13
14	1	723816	723822	1	723828	14
15	2	702809	702815	1	702835	15

Figure 13. Tycho data reference report table

Although this repository has about 1.8 million reports after initial data integrating and about 9,000 data references, I only consider 4836 data references, whose time intervals overlap or subsume each other. After implementing the proposed RS approach I detect fifty-seven conflicts, and all of them are confirmed with inconsistent report values.

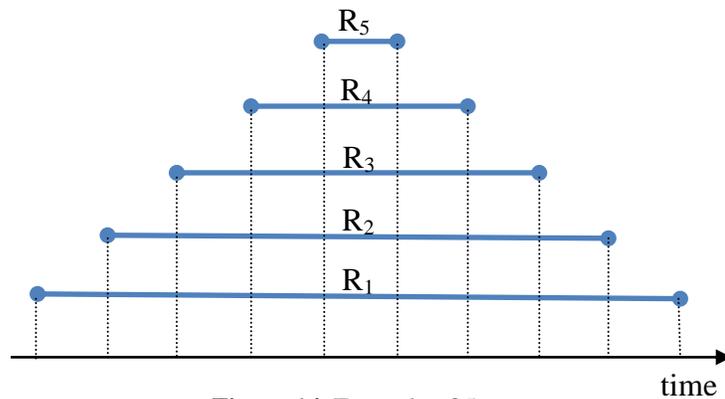
### 3.4 STUDY 2 – INCONSISTENCY DETECTION IN SIMULATED DATA

In this dissertation, I consider several simulations in the study 2:

- **Simulation 1:** The effect of the number of conflicting reports on the degree of inconsistency

In order to investigate the relationships between the  $C$  values, the  $\delta$  values, and reported values, I use a simple example to illustrate inconsistency in a controlled environment. Gaussian distribution is widely used for simulations of error distribution; however, it cannot give us

enough error to simulate inconsistency for the experiment (i.e. most runs have no inconsistency or have partially overlapping reports). Therefore, I manipulate reported values from the bottom of a triangle subsumption report hierarchy (Figure 14). I swap reports in order to create inconsistency, or to increase the degree of inconsistency. The degree of inconsistency can be defined by the number of conflicting reports, the overlap of reported time intervals, or the differences between reported values. In this simulation I only use the number of conflicting reports as a measure of inconsistency degree. The inconsistency is due to swapping report values making a report of a longer length have a lower reported value compared to a value of a subsumed report. The triangle structure forms a total subsumption hierarchy, in which shorter reports are subsumed by longer reports. Figure 14 shows five reports in the subsumption hierarchy.



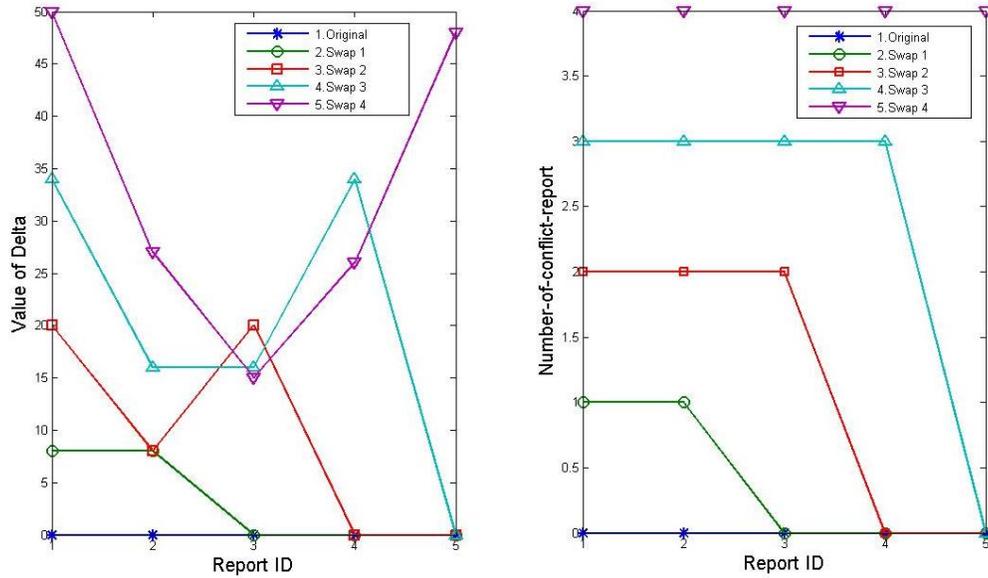
**Figure 14.** Example of 5 reports

At the beginning, we do not have any conflicts in the subsumption hierarchy. At each swap, I exchange the report value from bottom-up to inject inconsistency in this data reference since reports that have longer length should have greater report value compared with shorter reports under total subsuming condition. Take swap 1 as an example. The  $V_1$  changes to 4 from 5

and the  $V_2$  changes to 5 from 4 at the same time. Obviously these reported values are going to cause conflict since the length of  $R_1$  is larger than  $R_2$ . Table 7 shows these report lengths, and how these report values are going to exchange in each swap.

**Table 7.** Example of pyramids with 5 reports

Report ID	Report Length (1: shortest, 5: longest)	Report Value (1: smallest, 5: largest)				
		Original	Swap 1	Swap 2	Swap 3	Swap 4
$R_1$	5	5	4	3	2	1
$R_2$	4	4	5	4	3	2
$R_3$	3	3	3	5	4	3
$R_4$	2	2	2	2	5	4
$R_5$	1	1	1	1	1	5

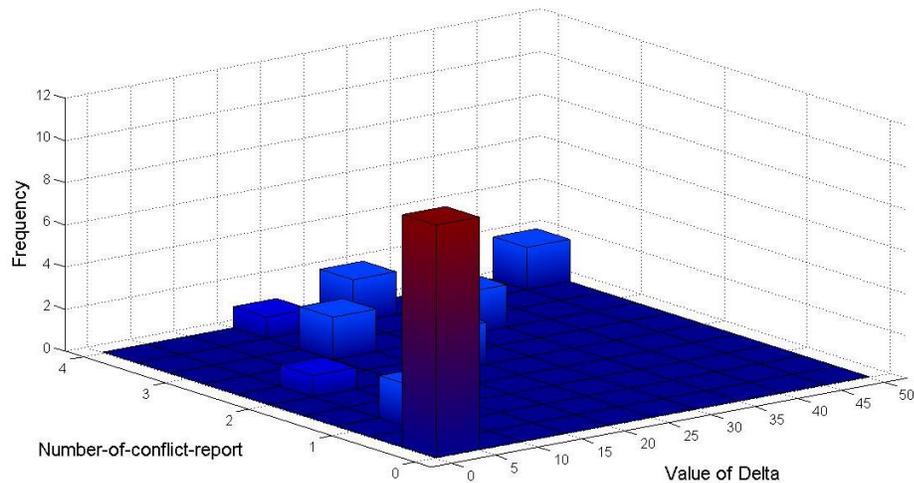


**Figure 15.** Simulation results of 5 reports

Figure 15 shows the  $\delta$  value and the  $C$  value of every report in each swap. In the default configuration, the  $\delta$  value and the  $C$  value for each report are all zero since reports are consistent.

At the first swap, I have nonzero  $\delta$  value and nonzero  $C$  value for  $R_1$  and  $R_2$ , which indicate that there is inconsistency between the reported values since I exchange the reported values of  $R_1$  and  $R_2$ . For the second swap, the  $\delta$  value for  $R_1$ ,  $R_2$ , and  $R_3$  are nonzero, and so is the  $C$  value for these three reports. Continuing to swap the rest of the reports will generate more conflicting reports. Figure 15 shows that the  $C$  value increases when I exchange more reported values. The  $\delta$  value for each report increases as the  $C$  value increases at each swap, but the  $\delta$  value does not always increase proportionally with higher degree of inconsistency. Therefore, the  $\delta$  value indicates the existence of inconsistency, but cannot represent the degree of the conflict.

In Figure 16, the three-dimensional figure of frequency of the  $C$  value and the  $\delta$  value shows an implicit trend that the  $\delta$  value increases proportionally when the  $C$  value increases. The  $\delta$  value is zero when the  $C$  value is zero. Therefore, the existence of inconsistency can be expected with nonzero  $C$  and nonzero  $\delta$  value.



**Figure 16.** Relation between number of conflicting reports and delta value

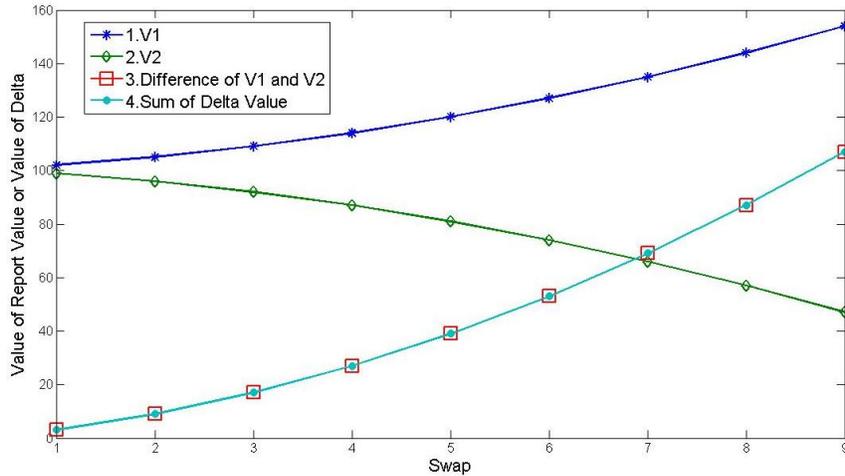
- **Simulation 2:** The effect of the magnitude of value difference between conflicting reports on the degree of inconsistency.

Here is another simulation to explain the effect of the  $\delta$  value and the magnitude of value difference between conflicting reports. In other words, this simulation uses the magnitude of value difference between conflicting reports as a measure of conflict degree. For two conflicting reports, I denote the reported values as  $V_i$  ( $i$ : report ID) and the difference between reported values as  $V_{i_1 i_2}$  ( $i_1 \neq i_2$ ). The characteristic linear system of these two reports is

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} 100 \\ 100 \end{bmatrix}.$$

I increase  $V_1$  by 5 units and also decrease  $V_2$  by 5 units at each turn to increase the magnitude of difference between reports  $R_1$  and  $R_2$  under conflict conditions in order to investigate whether the magnitude of reported value difference impacts the  $\delta$  value. Figure 17 plots the magnitude of difference of report value versus summation of  $\delta$  values across reports in each turn. Figure 17 shows the summation of the  $\delta$  values is the same with the magnitude difference. The  $\delta$  value increases when the magnitude of difference increases. Under conflict conditions, meaning there are no feasible solutions of linear equations, the solution set generated by the nonnegative least squares method aims to satisfy as many equations as possible. For example, when there are two inconsistent equations in the characteristic linear system, the solution set generated by nonnegative least squares method can satisfy one of the linear equations. Thus, the  $\delta$  value increases when the difference between values of these two reports increases. If there is any conflict, then the  $\delta$  value will not be zero and its value is proportional to the magnitude of difference between reports. Therefore, the nonzero  $\delta$  value can be used as an indicator of the occurrence of inconsistency and the degree of inconsistency in this simulation.

Summarizing the simulations that I have described in this section, the better way to show an inconsistency degree is combining the  $\delta$  value and the  $C$  value. The nonzero  $\delta$  value and  $C$  value show that reports are inconsistent and identify these inconsistent report IDs. In addition, the  $\delta$  value increases when the number of conflicting reports increases, or the magnitude of difference between reported values increases. As a result, this proposed approach can work as an inconsistency detector and as an indicator to assist users with early awareness of data inconsistency before performing data fusion.



**Figure 17.** Magnitude difference and delta with two reports

## 4.0 IMPLEMENTATION AND EVALUATION OF TEMPORAL FUSION

### 4.1 BACKGROUND OF THE CONFLICT DEGREE METHOD

The inconsistency detection can assess the reliability of data, and the data fusion procedure can further improve the quality and utility of data. I am going to elaborate and compare my proposed data fusion strategy with a well-known optimal conflict degree method in the following sections.

In my simulations, I compare the difference between the actual value and the estimated value generated by the *conflict degree (CD)* method and my *reverse substitution (RS)* method from reports from heterogeneous resources. The concept of the CD method is proposed by (Zadorozhny & Hsu, 2011), where the authors use the CD method to estimate aggregate values from redundant (overlapping) reports. Each report is represented as a tuple/triple (From, To, Value) or abbreviated as (F, T, V) which stands for report start time (From), report end time (To), and the number of events reported within that time interval (Value). The value of CD between two historical tuples  $r_1$  and  $r_2$ , where  $r_1 = (F_1, T_1, V_1)$  and  $r_2 = (F_2, T_2, V_2)$ , is computed by the equation below

$$CD(r_1, r_2) = RO(r_1, r_2) \times e^{k(1-RC(r_1, r_2))(RO(r_1, r_2)-1)}.$$

The *relative contribution (RC)* in the equation is defined as

$$RC(r1, r2) = 1 - \frac{|v1-v2|}{(v1+v2)}.$$

And the *relative overlap (RO)* is

$$RO(r1, r2) = \max\left(\frac{|t1 \cap t2|}{|t1+t2|}, 0\right), \text{ where } t1 = [F1, T1] \text{ and } t2 = [F2, T2].$$

The  $t1 \cap t2 = [\max(F1, F2), \min(T1, T2)]$ , which represents the intersection of time intervals

of  $r1$  and  $r2$ . In a similar way, the  $t1 + t2 = [\min(F1, F2), \max(T1, T2)]$ , which represents the

total time range of these two tuples. The length of time interval  $= [F, T]$ , or the number of time

unit covered by  $t$ , can be computed as  $|t| = T - F + 1$ . The value of CD varies between 0 and 1

where 0 means no overlapping and 1 means total overlapping with the same report values. The

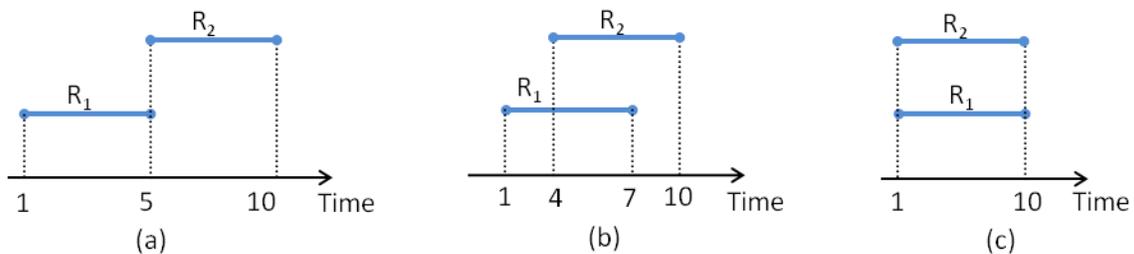
higher the CD, the more similar the report values are or the higher time overlap is. A

disadvantage of the CD method is that we cannot differentiate whether the high CD value is a

result of the high relative contribution or high relative overlap. In addition, we do not know

whether these reported values are trustworthy. The example of CD values for the different

configurations are shown in Figure 18 and Table 8.



**Figure 18.** Scenarios of CD

**Table 8.** Corresponding RO, RC, and CD for different report structure

<b>CD Scenarios in Figure 18.</b>	<b>(a)</b>	<b>(b)</b>	<b>(c)</b>
$/t_1 \circ t_2/$	1	4	10
$/t_1 + t_2/$	10	10	10
<i>RO</i>	0.1	0.4	1
$R_1=100, R_2=100$	<i>RC</i>	1	1
	<i>CD</i>	0.1	0.4
$R_1=100, R_2=10$	<i>RC</i>	0.18	0.18
	<i>CD</i>	0.0478	0.24456

Scenario (a) shows the reports  $R_1$  and  $R_2$  with no overlap, scenario (b) shows partial overlap, and scenario (c) shows total subsumption. As we observe from Table 8, both high overlap and small value difference between reports will cause high CD value. However, this method cannot ascertain that this high CD value indicates the existence of inconsistency, or that the smaller value difference really has higher probability to cause conflict.

There is an optimal CD threshold for each configuration that minimizes the estimation error. The authors observed that there is an optimal CD threshold associated with each scenario. The optimal CD threshold for each group of conflicting reports that would minimize the estimation error cannot be defined without the knowledge of actual time unit numbers. This pre-generated optimal CD threshold differs under various event densities, report numbers, and report densities. Therefore, this CD algorithm is sensitive to prior knowledge of actual time unit numbers.

I compare the performance of the CD method and the RS method of data fusion under consistent scenario and also explored the effect of inconsistency. The inconsistency condition

was reflected in three error probability settings: 25%, 50%, and 95% with 100 runs of simulation. These conditions were simulated by swapping have reported values to create inconsistency. In each swap, I exchange the report values between shortest and longest of overlapping reports. Therefore, each swap is able to create inconsistency. The closer the report time stamps, the higher the probability to have overlapping or subsumption between reports, which increases the likelihood of inconsistency (i.e. reports for the same events at different time intervals with contradicting values). The maximum number of subsumptions for each report structure is  $n(n-1)/2$  ( $n$ : report number). Finding a way to introduce more inconsistency and to increase the degree of inconsistency will be discussed in the future study section.

The swap probability distribution also represents the degree of inaccuracy of the real data. Inaccurate report values are difficult to detect, and make it difficult to recover the original interval values. This causes problems in my proposed inconsistency detection system since it may be unaware of the inaccurate reports. For example, report  $R_1$  is subsumed by  $R_2$ , and values are 500 and 1000 respectively. The system cannot detect the occurrence of inaccuracy if the report values of  $R_1$  and  $R_2$  have been accidentally recorded as 50 and 100. The modified report values will not cause any inconsistency since the shorter report still has smaller value even through these report values are inaccurate.

## **4.2 EXPERIMENT SETUP AND CONSISTENCY CONDITIONS**

I have two different conditions for the performance comparison: the simulations in Section 4.2 are under consistency condition and the simulations in Section 4.3 are under inconsistency condition. In my simulation, I varied the event density, report number, report duration of 20 and

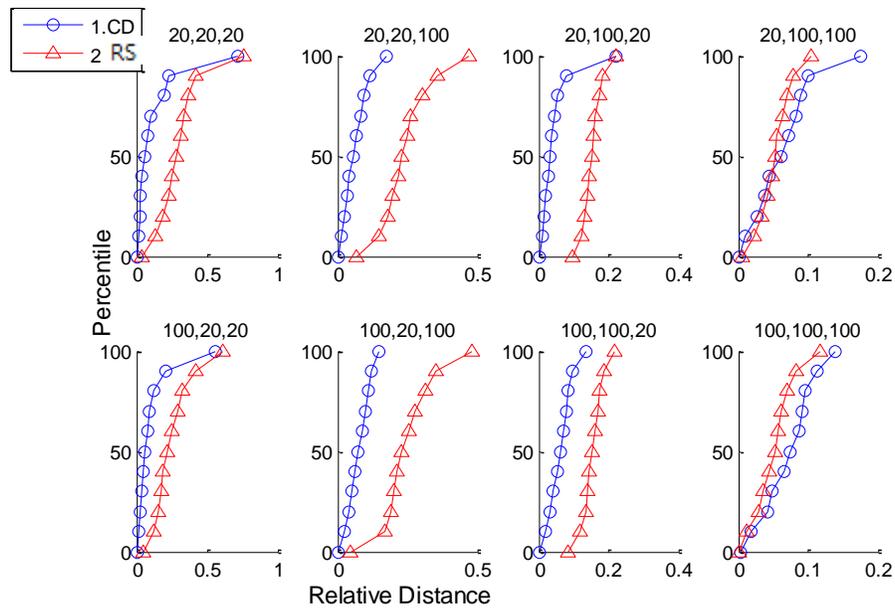
100, and the total number of time units of 150 and 1000. In reality, we have no information about data distribution, optimal CD threshold, and the actual event density for each interval in advance. The only available information in the historical data center is the start time, end time, and a value of each report. Therefore, the goal is to find a method for each configuration to minimize the misestimating error with little or no knowledge of actual number of events. The performance measurement compares the estimation error, which is the difference between the summation of the actual value and the summation of the estimated value of the event values across each interval. I use the relative distance for performance measurement, and it is defined as

$$\text{Relative Distance (RD)} = \frac{|\text{estimate value} - \text{actual value}|}{\max(\text{estimate value}, \text{actual value})}$$

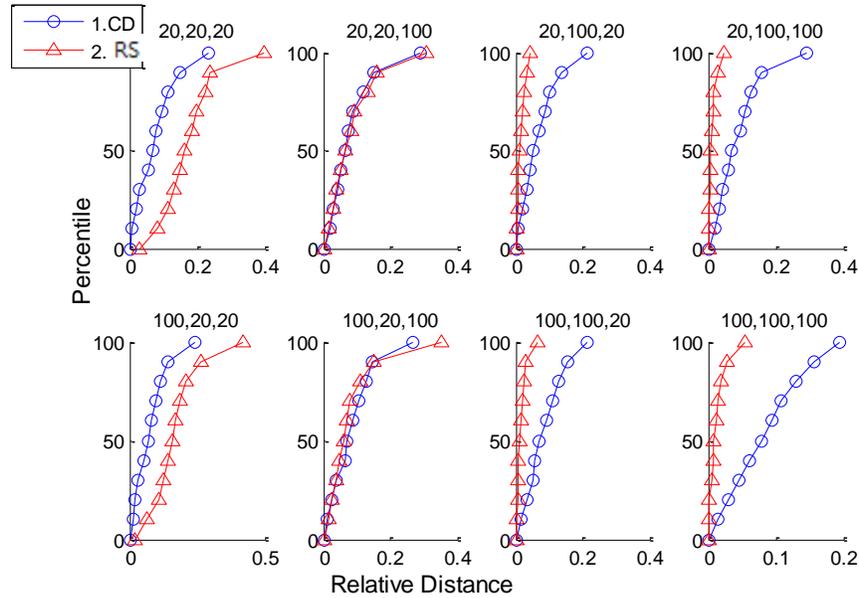
The configurations of the experiment are described in Table 9. I use normal distribution to configure the experiment. The numbers in the table for the event density, report number, and report duration are expected values of corresponding normal distribution. In each case we set the deviation of 5. Take the first row as an example, the expected number of reports is 20, and the expected duration for each report is 20 time units. Each time unit contains a number of events. The expected density of events in each time unit is 20 units, and there are a total number of time units 1000. The reports aggregated from events will be allocated sparsely on the time line since we expect about 20 short reports over 1000 time units. Figure 19 and Figure 20 show simulation results when the total number of time units is 1000 and 150 respectively. The smaller the RD, the better the performance is because the difference between estimated and actual values is smaller. For the case of consistent reports, the measurement of performance is mainly focused on RD because the user does not need to worry about inconsistency.

**Table 9.** Configuration of inconsistency simulation

Event density	Report number	Report duration	Total number of time units	Description
20	20	20	1000	Low event density, few short report, and sparse overlap
20	20	100	1000	Low event density, few long report, and sparse overlap
20	100	20	1000	Low event density, many short report, and sparse overlap
20	100	100	1000	Low event density, many long report, and sparse overlap
100	20	20	150	High event density, few short report, and dense overlap
100	20	100	150	High event density, few long report, and dense overlap
100	100	20	150	High event density, many short report, and dense overlap
100	100	100	150	High event density, many long report, and dense overlap



**Figure 19.** Relative distance of CD and RS for 1000 time units



**Figure 20.** Relative distance of CD and RS for 150 time units

Both figures show the performance evaluation of RD in percentile. Figure 19 shows the performance of the CD and the RS with a large total number of time units of 1000. Generally in all cases in this figure, the CD method has lower RD except for the cases [20, 100, 100] and [100, 100, 100]. Figure 20 shows the same configurations with the total number of time units of 150. The performances of the CD and the RS are slightly different in this situation. For all cases of report number 20, the CD performs equally or better than the RS. Only for the cases [20, 20, 20] and [100, 20, 20] are the values of RD in Figure 20 significantly lower than those in Figure 19. For all cases with report number 100, the RD of the RS is generally lower in Figure 20.

There are fewer intervals with larger number of time units, while other configuration settings remain the same. In other words, the sparse report distribution will reduce the report overlapping. Therefore, I assume the performance for both methods will be better in this case, especially for the RS since higher overlapping may increase the number of unknown variables in

the underdetermined linear system. However, we observe that the RD is lower for both the CD and the RS method for 150 time units when compared with the case of 1000 time units. Comparing the CD and RS for different number time units gives us the same observation. For my proposed RS approach, I assume that decreasing the number of time units will increase the overlapping of events, which provides more information to determine the values from the characteristic linear system.

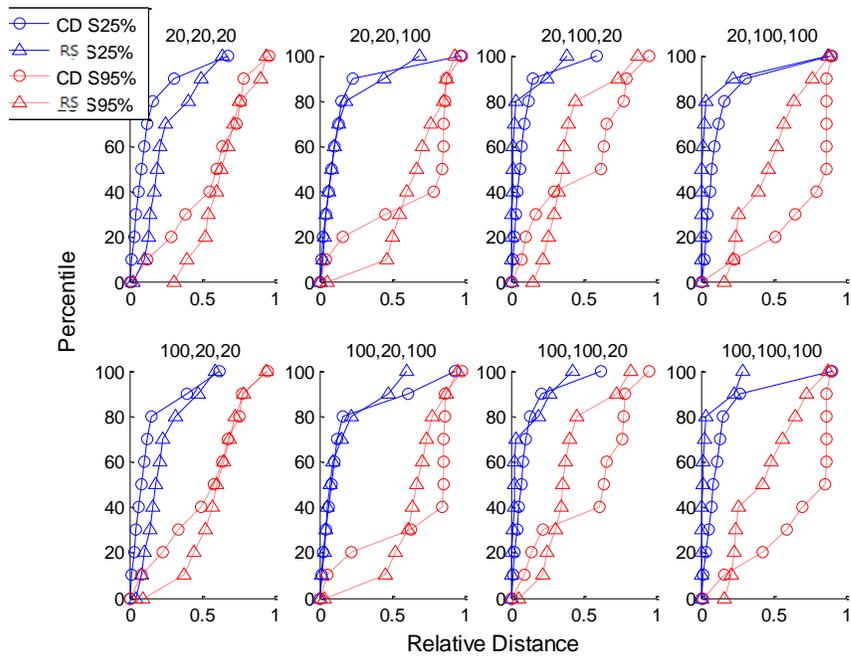
Moreover, for all cases with expected report number of 100, the RS outperforms the CD. The values of RD of the CD under different conditions with the same report number are similar, but the values of RD for the RS method is significantly lower for expected total number of time units of 150 and expected report number of 100. In this simulation, I observe that increasing the number of reports leads to performance improvements of RS. The effect of report number may be explained by the fact that more reports can provide more information for the underdetermined linear system, and the degree of freedom of the system is decreased. Therefore the generated solution set tend to decrease from finitely many solutions to one unique solution. In summary, for both methods, as the number of time units decreases, the RD decreases. Increasing the report number will cause the RS method to outperform the CD method. Therefore, the RS is a better option for data fusion for large number of reports with more overlapping and more subsumptions.

### **4.3 EFFECT OF INCONSISTENCY**

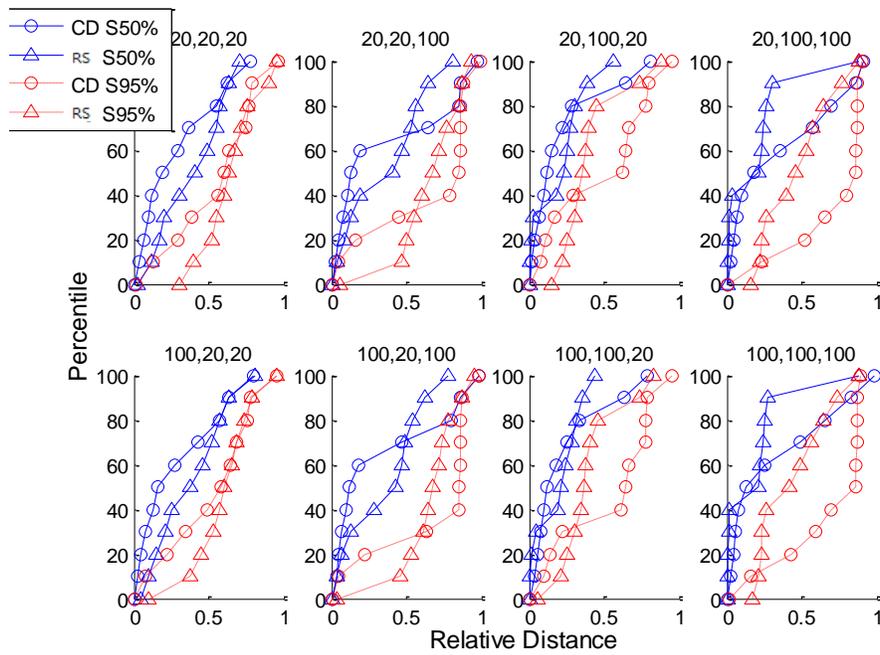
I generate inconsistency between reports by swapping values of reports. If the reports overlap considerably, then there will be a higher chance of inconsistency produced by this method. For

consistent reports, the  $\delta$  values and the  $C$  values will be zero across all configurations. For inconsistent reports, my proposed RS method will produce nonzero  $\delta$  value, which is the indicator of inconsistency. The concept of the degree of inconsistency was demonstrated with the process of swapping in the triangle subsumption hierarchy as explained in Study 2. At each swap, I exchange the value of shortest reports with the values of longest reports to generate inconsistency. The inconsistency increases as the number-of-swap increases. In this simulation, I keep the same report structure (i.e. the number, duration, and allocation of reports) in each run, but with different probability of swapping overlapping reports. I use Normal distribution to configure the probability of swapping. As previously, the inconsistency condition is reflected in three error probability settings: 25%, 50%, and 95% with 100 runs of simulation. For example, the scenario of 25% probability of inconsistency means that 25% of the 100 runs will have their reports swapped. The percentiles of RD for both CD and RS with 150 time units comparing configurations of 25%, 50%, and 95% probability of swap are shown in Figure 21 and Figure 22.

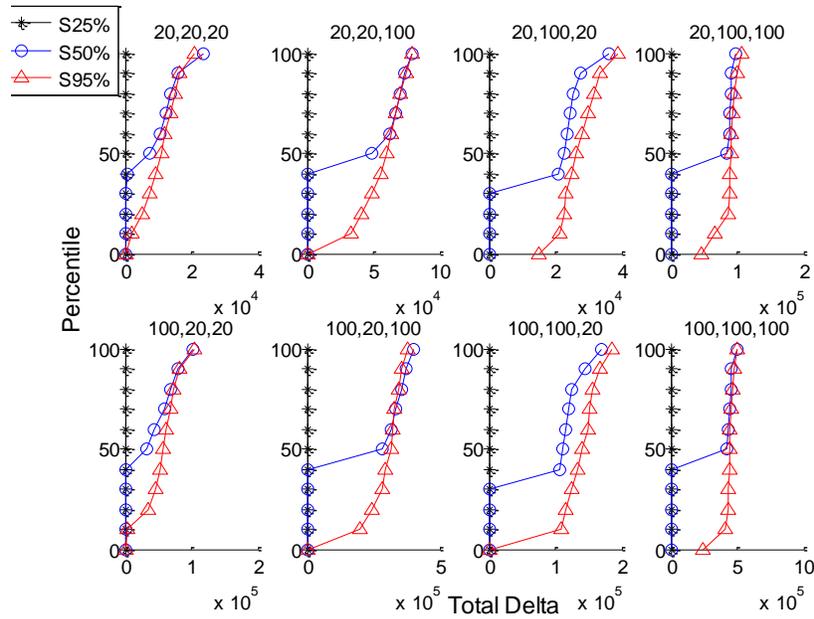
These figures show similar aspects relating to the probability of swap. The values of RD are higher in these figures than in case of the consistency conditions. The RD values of RS are higher than the RD values of CD when the expected report number is 20 across the swap probabilities. The RS performs better than the CD for expected report number of 100 for swap probability 25% and 95%. At swap probability of 50%, the performance is only slightly better. The difference between the RS and the CD increases as the swap probability increases. In case of expected report number of 100, the RS performs better than the CD for about 50% of the simulation runs at swap probability 95%. The RS method and the CD method perform similarly at different swap probabilities for 150 time units.



**Figure 21.** Comparison with 25% and 95% probability of swap



**Figure 22.** Comparison with 50% and 95% probability of swap

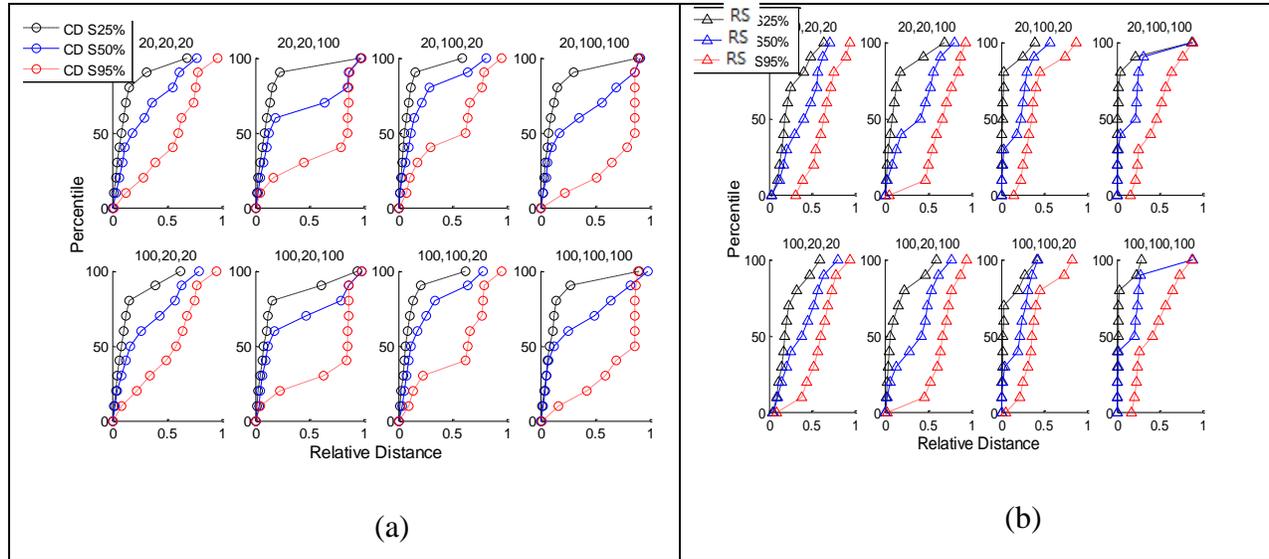


**Figure 23.** Total delta for 25%, 50%, and 100% probability of swap

To verify if the performance difference is caused by different swap probabilities, I compare the total  $\delta$  value (i.e. the summation of the  $\delta$  value across the reports for each run) for each number-of-swap condition. As seen in Figure 23 I found that the total  $\delta$  value is proportional to the swap probability and its percentile. The nonzero  $\delta$  value can be used as an indicator for the existence of inconsistency. Its value is also related to many other variables such as the magnitude of difference between reported values and the-number-of-inconsistent-reports ( $C$ ). However, the total  $\delta$  value still represents the degree of inconsistency (the total  $\delta$  value is nonzero if the report values are inconsistent within a run); and the higher percentile of nonzero total  $\delta$  value corresponds to the higher the number of runs that are inconsistent. According to Figure 23, the total  $\delta$  value has about 100% zero values at swap probability 25%, about 60% nonzero values at swap probability 50%, and about 90% nonzero values at swap probability 95%. The RS method also has a significant improvement compared with the CD method as the

swap probability increases. Therefore, the degree of inconsistency is represented as a function of the probability of number-of-swap.

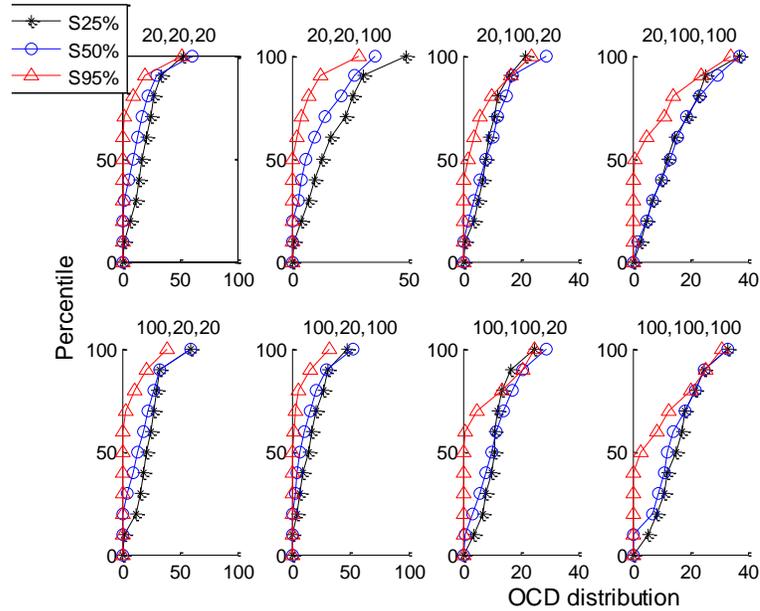
In Figure 24 I compare the performance of the CD method and the RS method across number-of-swap conditions. We observe, that the RD between these two methods increases when the probability of number-of-swap increases.



**Figure 24.** Comparison of CD and RS in each probability of swap conditions

In each run, I change the CD from 0 to 1 with steps of 0.01. The optimal conflict degree (*OCD*) is the CD threshold that has the minimum estimation error. The *OCD* is generated based on the pre-generated estimated event distribution at each run. The CD value between reports in each run changes along with the probability of the number of swaps. This is because RC of the CD is a function of the probability of the number of swap. Therefore, CD, which is a function of RC will be affected by this probability. Thus, the *OCD* is not a stable value at each run even though the report structure is the same. The change of *OCD* across the probability of the number-of-swap is shown in Figure 25.

In this simulation, I found that when the probability of swap increases, the total  $\delta$  value also increases. The RS method can identify this inconsistency and perform well in data fusion. As the probability of the swap increases, the performance of the RS method improves, and the performance difference between the RS method and the CD method also increases in favor of RS.



**Figure 25.** OCD distribution for all inconsistent condition

#### 4.4 ACCURACY AND CONSISTENCY

Assuming every report refers to the same data reference, my proposed algorithm formalizes the historical data as a mathematical model of a characteristic linear system and performs consistency checking and data fusion of data sets in an integrated repository. The redundant reports may produce issues of inaccuracy and inconsistency in an integrated database. An *accurate* report reflects correct reported value, and a *consistent* report does not conflict with

overlapping reports. Here are some possible cases of redundant reports for inconsistency detection:

- (1) Accurate and consistent:  $\delta = 0$  of the linear system and these reports will be described as consistent.
- (2) Inaccurate and consistent:  $\delta = 0$ , and reports will be described as consistent.
- (3) Inaccurate and inconsistent:  $\delta \neq 0$ , and reports will be described as inconsistent.
- (4) Accurate and inconsistent:  $\delta = 0$ , this not possible under my model.

For case (2), the integrated database includes inaccurate report values. As a result, it is hard to detect this inaccuracy when there are recording errors and these report values do not contradict each other (For example, the miss-recorded report values for  $R_1$  and  $R_2$  are 100 and 50 respectively even though the actual values are 1000 and 500. This will not cause any inconsistency even when  $R_2$  is subsumed by  $R_1$ ).

In case (3), my proposed method can detect the occurrence of inconsistency and perform data fusion with the estimated interval values close to the ground truth (i.e., decreasing the level of inaccuracy). Other data fusion methods used in sensor network such as averaging, Bayesian, or Dempster-Shafer are focused on the consensus of sensor data to achieve the advantages of multiple sensors for reducing data uncertainty and unreliability. However, these methods cannot easily handle the case (3), or even to find out which report causes the inconsistency. A single sensor or many sensors that only include one type of sensor may be insufficient or ambiguous in many applications such as user appearance detection, map merging, and surveillance monitoring. As a result, the data from multiple sensors, or the information combining different types of sensors becomes more important since they can be integrated and provide more concrete and comprehensive information. For example, combining the thermal, acoustic, and oxygen sensors

to detect a living object’s existence will be more accurate and reliable as opposed to using only a camera.

The taxonomy of inconsistency detection for different subsumption and different inaccuracy conditions is shown in Table 10.

**Table 10.** Inaccuracy and subsumption condition

		<b>Inaccuracy condition</b>		
		<b>No inaccuracy</b>	<b>Low inaccuracy</b>	<b>High inaccuracy</b>
<b>Subsumption condition</b>	<b>Random</b>	No inconsistency	Low inconsistency	Low inconsistency
	<b>Low subsumption</b>	No inconsistency	Low inconsistency	Low inconsistency
	<b>High subsumption</b>	No inconsistency	Low inconsistency	High inconsistency

Under no inaccuracy conditions, I assume that the RS method will outperform the CD method if the number of reports in the linear system is large enough to generate a correct solution set. For low inaccuracy, if the report structure is sparse, then the probability of being diagnosed as inconsistency using the proposed RS method may be very low since the inaccuracy is hard to identify when the overlapping is scarce. Therefore, the low inaccuracy is hard to detect, especially under random or low subsumption conditions. For the case of high inaccuracy, it would be problematic to detect which report is correct. Therefore, when the report structure has low subsumption, only some inaccurate overlapping reports could be found and diagnosed as low inconsistency. The high subsumption condition with highly inaccurate reports will lead to larger  $\delta$  values. I assume that the proposed RS method will perform better under no inaccuracy conditions as well as high inaccuracy conditions when there are numerous overlapping reports.

## **5.0 SYSTEM DESIGN – TEMPORAL SPATIAL INFORMATION FUSION**

### **5.1 GENERAL ARCHITECTURE OVERVIEW**

In order to further involve spatial dimension in the process of data fusion, I adopt a scenario of an urban search and rescue task using mobile robots. I extend my information fusion strategy for the task of target detection at specific locations and time intervals. The search targets can be either static or dynamic within the environment. The issue of moving target detection in the robot search and rescue task is a major focus of the environmental knowledge. In this dissertation, I address the RS method that includes temporal and spatial fusion for inconsistent report detection and target detection. I implement this approach with the simulated data sets of sensors on ground-moving robots. For the inconsistent report detection, the overlapping routes with a large group of robots will mislead the result since targets may be double counted. For the target projection, knowing the accurate number of targets at each location and the trajectories of moving targets can help systems to make decisions with prior knowledge of the environment and using related data mining techniques. My proposed approach RS can be interpreted in terms of multisensory integration and data fusion.

The problems of targets observation are focused on targets' local information that includes the location, number, appearance time, and trajectory of targets. One application of the moving target observation is utilizing robots to perform the search and rescue task to find targets

(victims) in an extremely harsh environment that is dangerous for humans. Robots and targets may be static or dynamic; the moving target will increase the task difficulty significantly. The techniques used to detect local information of targets can be categorized into three major methods. The first type of method uses sensors mounted on robots such as camera, laser, or GPS, and users can only identify robots' locations by the sensor feedback. The second type uses multiple static sensors spread throughout the environment and these sensors are located at specific locations, such as the entrances of corridor. The last type uses robots moving around following the targets in the environment. The report type of local information from multiple sensors or multiple types of sensors can be homogeneous or heterogeneous. The issue of data duplication across reports is very common since designers often use redundant sensors with location overlap or time overlap to achieve the benefits of data reliability, accuracy, complementary, timeliness, and cost reduction of the information (Luo & Kay, 1989).

The problem of two-dimensional fusion becomes more complex if we consider issues of reports overlapping and dynamic target moving together. In Section 3.0 of my previous temporal fusion studies, the proposed RS method models report structure using the characteristic linear system, and generates the solution set using the nonnegative least squares method. The RS method can be used for temporal data conflict detection and data fusion. The strategy of the RS method is to map data from heterogeneous sources into a linear system, and to find potential inconsistent reports based on the fact that the number of data references in each time interval should be identical across all data sources. To detect potential data inconsistency for data sets along more than one-dimension, my approach works for the multidimensional inconsistency detection when the user tackles one-dimension of inconsistency at a time, and progressively extends to all the other dimensions.

I use the simulated robot laser log data to test the RS method of target observations and temporal and spatial fusion. The purposes are to provide greater scalability from one-dimension (temporal fusion) to two-dimensions (temporal fusion and spatial fusion), to provide better accuracy in inconsistency detection, and to provide target observation in each location for a specific time. The processes for the two-dimensional fusion studies are to use the temporal fusion to estimate the number of targets per time interval for a given location (according to the laser data of robots, location can be defined by a group of points) and to use the spatial fusion to estimate the number of targets per location (group of points) for a given time. Finding the potential temporal and spatial conflicts requires at least two linear system models – one focused on temporal fusion that generates the solution set (i.e. estimated value) for time intervals, and another focused on spatial fusion that generates the solution set for spatial intervals. The sequence of this two-dimensional fusion has two steps. First, it generates estimated values for each *time interval* of the characteristic linear system. These values can be used in another characteristic linear system. Second, it generates estimated values for each *spatial interval*. The tasks of conflict detection and target number identification at a specific time and location can be accomplished after these two steps. Given the estimated values generated by multiple linear systems for each time and location, I can acquire the local information of targets, and describe the target moving trajectories.

For different datasets, I observe four basic patterns that provide us with more information to determine the functional dependency of data. From the data references (reports) recorded in each table, I can differentiate between these four patterns.

- (1) Static target, static robot

**Table 11.** Static target, static robot

<b>Location</b>	<b>Time_From</b>	<b>Time_To</b>	<b>Target_Num</b>
<b>L1</b>	T1	T3	V1
<b>L1</b>	T2	T4	V1

Table 11 shows a log example of one robot. The first row indicates that there is V1 number of targets at location L1 from time T1 to time T3. Reports from each robot show activities at the fixed locations during different times and record the same numbers of targets. These log data of the same locations refer to the static robots, and the unchanged number of targets in a specific location refers to static targets. Therefore, I can use locations or robot IDs to determine the number of targets. The functional dependency for this condition is

$$f(\text{location or robot ID}) \rightarrow \text{number of target of given locations.}$$

The minimum number of total targets denoted as  $x$

$$x = \arg \min (\sum_{j=1}^n NT_{ij}),$$

where the time  $i$  ranges between 1 to  $m$  and space  $j$  ranges from 1 to  $n$ . The number of targets at a given time  $i$  and location  $j$  is denoted as  $NT_{ij}$ .

(2) Static target, moving robot

**Table 12.** Static target, moving robot

<b>Location</b>	<b>Time_From</b>	<b>Time_To</b>	<b>Target_Num</b>
<b>L1</b>	T1	T3	V1
<b>L2</b>	T1	T3	V2
<b>L1</b>	T2	T4	V1
<b>L2</b>	T2	T4	V2

If I release the constraint of robots' moving ability, reports from each robot will contain information about different locations, but the number of targets will remain unchanged for each location at different times. The functional dependency is

$$f(\text{location}) \rightarrow \text{number of target per location.}$$

The equation for condition (2) is

$$x = \arg \min_{i \in [1, m]} (\sum_{j=1}^n \Delta NT_{ij}),$$

where  $\Delta$  represents the estimated value of the RS method.

(2) Moving target, static robot

**Table 13.** Moving target, static robot

<b>Location</b>	<b>Time_From</b>	<b>Time_To</b>	<b>Target_Num</b>
<b>L1</b>	T1	T3	V1
<b>L1</b>	T2	T4	V1'

Reports contain tuples of the same location, but different number of targets at different times. In this condition, I rely on collaborative data from different robots to achieve general information of the whole environment. Each robot contains data in a specific location, but with different number of targets at different times. The functional dependency is

$$f(\text{time and data from robots}) \rightarrow \text{number of target of given locations of given time.}$$

The equation for condition (3) is

$$x = \arg \min_{j \in [1, n]} (\sum_{i=1}^m \Delta NT_{ij}).$$

(3) Moving target, moving robot

**Table 14.** Moving target, moving robot

<b>Location</b>	<b>Time_From</b>	<b>Time_To</b>	<b>Target_Num</b>
<b>L1</b>	T1	T3	V1
<b>L2</b>	T1	T3	V2
<b>L1</b>	T2	T4	V1'
<b>L2</b>	T2	T4	V2'

There are different numbers of targets for the same location at different times. I assume that targets and robots are moving at a dynamic speed and are more coordinated. The functional dependency is

*f(time and location data of all robots)*

*→ number of target of given locations of given time.*

The equation for condition (4) is

$$x = \arg \min \sum_{i=1}^m \Delta (\sum_{j=1}^n \Delta NT_{ij}).$$

From these characteristic patterns of data in these aggregated tables, researchers can (1) verify the moving accessibility of the targets and the robots (static or dynamic); and (2) determine the number of targets at a given time and location with these four dependency functions. Calculation of the number of targets from these equations will be introduced in detail in the following section.

## 5.2 INFORMATION FUSION TAXONOMY

In the previous section I identified the four patterns of the functional dependency of data sets for the search and rescue task. In this section I will elaborate on which fusion method we should use under different circumstances. Definitions of fusion strategies can be specified for time and space as follows:

(A) Temporal fusion: to find the number of targets per time interval per location;

(B) Spatial fusion: to find the number of targets per location per time interval.

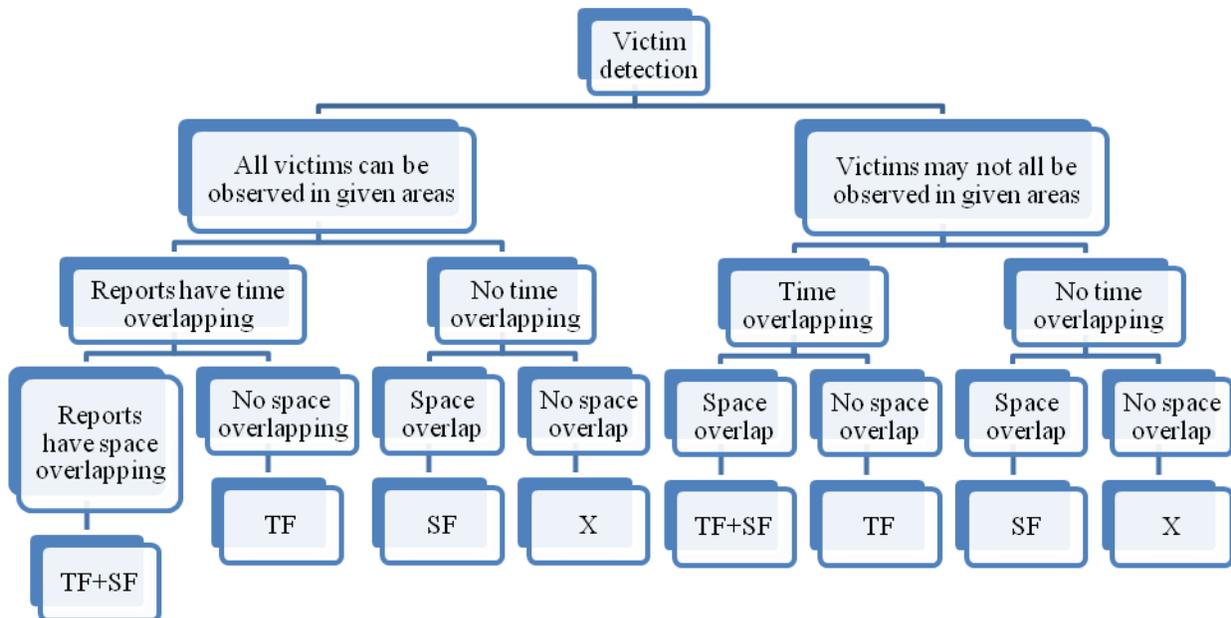
The sequences of creating the underdetermined linear system for temporal or spatial fusion and the analysis for these four patterns may be different. From the four cases above, the functional dependencies are specified as follows:

(1) Static targets:  $f(\text{location}) \rightarrow \text{number of target}$ . The locations can be used to determine target numbers since the targets are static. The number of targets at each location should be unchanged at various times.

(2) Static robots:  $f[f(\text{time}) \rightarrow \text{location}] \rightarrow \text{number of targets}$ . The time can be used to determine target locations and to determine target numbers. Targets are moving randomly inside the environment, but the recorded logs provide only partial information about these targets since the number of robots or sensors are not enough to cover the entire environment. For example, the information about minimum number of targets at a given time or location is incomplete and the logs are insufficient for target number determination in every location at a specific time. However, the time factor is usually related to the properties of space, i.e. the number of targets at a specific time and location is unique. Therefore, knowing the minimum number of total targets across all locations at a given time provides the researchers with a general

overview of the data properties (e.g. the minimum target number), accessibility of robot and target (e.g. static or dynamic), and the sequence of performing multidimensional data fusion (e.g. perform temporal or spatial fusion first).

Integrated information from multiple reports of different robots provides users the data of interest with better coverage compared with single data source. Figure 26 shows which of the data fusion should be used under various conditions. For example, if we consider the case that (1) there are enough sensors or robots so all targets can be observed across all the areas, and (2) data about recorded locations and times may be redundant (time overlapping), the temporal fusion and spatial fusion should be performed sequentially. The type of data fusion for various conditions depends on the factors in Figure 26 (where TF denotes temporal fusion, SF denotes spatial fusion, and X denotes that neither TF nor SF will improve target number estimation since there is no overlapping reports).



**Figure 26.** Target(victim) detection categories of temporal and spatial fusion

Researchers can also consider some basic constraints listed below to decrease the degree of freedom on the linear system in order to have better accuracy of the estimated number:

(a) All targets can be observed: the summation of total number of targets across all locations at any time unit is a fixed value. This condition provides the characteristic linear system with more information (i.e. decrease the degree of freedom by adding one extra equation restricting the total number of targets), and then computes the solution set with better accuracy. However, this constraint may be not satisfied in real life.

(b) Total number of target for each location (cell) at time  $T_{x+1} \leq$  summation of number of target of each cell's neighbor cells at  $T_x$ : e.g. Total number of targets for cell  $C_1$  at  $T_2 \leq$  summation of total target of all cells  $C_2, C_4, C_5$  around  $C_1$  at  $T_1$ . This constraint ensures the number of targets in each report is a reasonable value.

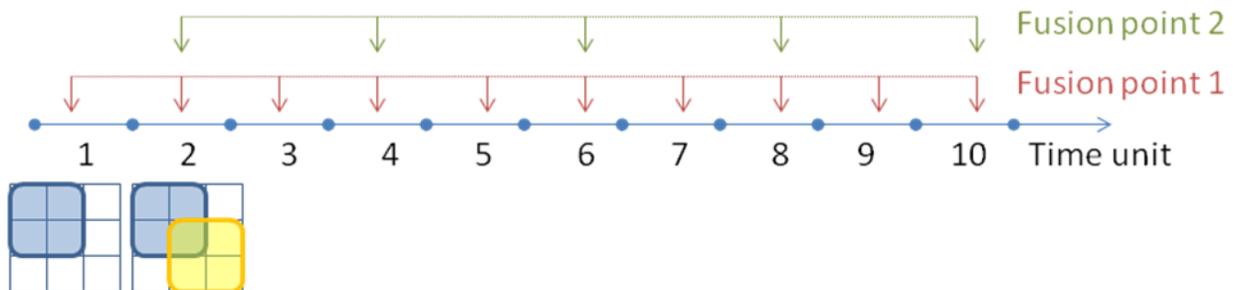
### **5.3 SPATIAL INFORMATION FUSION SIMULATION**

A key application of my proposed RS method is in two different situations: using the temporal data fusion to track the number of dynamic targets changing their positions across time durations, as well as using the spatial data fusion to monitor the number and allocation of static targets in a specific area. In order to further investigate spatial dimension in the process of data fusion, I use the scenario of search and rescue task using mobile robots. I extend my information fusion strategy to the problem of target detection at specific locations and time intervals. The occupancy status of each space unit of the environment is represented as an occupancy grid (Elfes, 1989; Konolige, 1997). The targets can be either static or dynamic in the environment. I then describe my current effort in applying the proposed approach through the simulation of the data sets of

sensors on ground-moving robots searching for static targets to the problem of spatial fusion. Robots exploration is problematic with overlapping routes with a large group of robots. Some targets may be double counted and mislead the result. The following simulations illustrate the strategies underlying my approach:

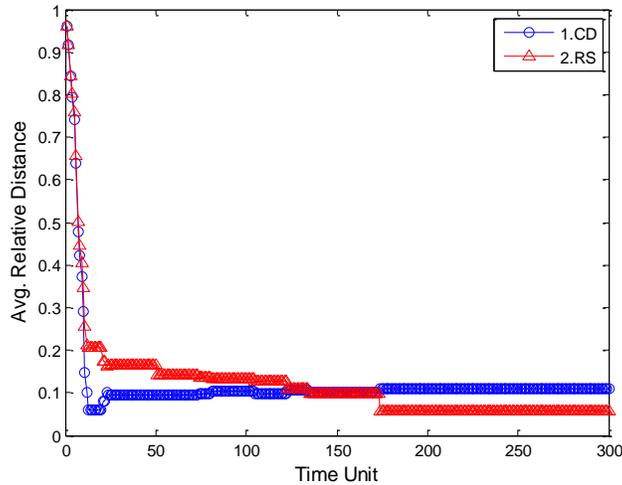
- **Simulation 1:** The timing of spatial fusion

The time to performing data fusion is one parameter of the spatial fusion simulation because it may important for the accuracy and computation time. In Figure 27 I take a time series with 10 time units as an example; the number of space reports is accumulating as the number of time unit increases. As a result, there has been only one space report at  $T_1$ , but two reports at  $T_2$  since more areas are explored by the moving robots. I assume the accuracy will increase as the time unit increases because the number of space reports is also increasing; however, the delay may increase as well. In that context, I have two fusion timings in the figure below referred to as fusion *points*. Under fusion point 1, I perform spatial fusion at each time unit, under fusion point 2 I perform spatial fusion at each 2 time units.



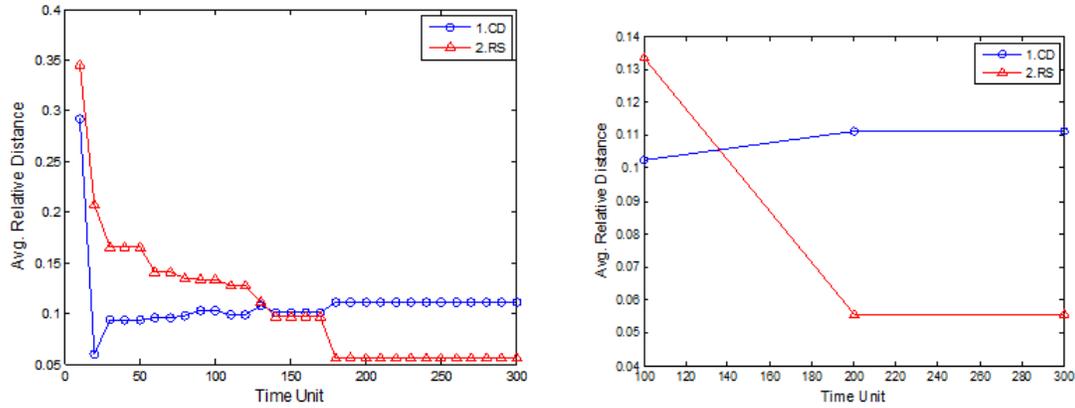
**Figure 27.** Fusion point

The functional dependency of the scenario of static target and dynamic robot are formalized as  $f(\text{location}) \rightarrow \text{number of target}$ . The locations can be used to determine target numbers. Targets in each cell are immobilized at their locations. Robots move around the environment and generate the space reports. The space reports contain the log of robot reports including the robot location, the target location and the number of targets. Thus, I can use the spatial fusion to determine the number of targets in each location. In the following simulation of spatial fusion, the number of time units is 300, and the number of space reports is increasing one as at each time unit. The number of space reports varies from only one report at  $T_1$  to three hundred reports at  $T_{300}$  to simulate the robots continuous exploring of the environment. The fusion point configurations are: at each time unit, at each 10 time unit, and at each 100 time unit, so there will be 300, 30, and 3 data fusion points correspondingly. The ground truth of the total number of targets in the environment at each time unit (i.e. number of events) aggregates the values from detected space units. The size of the ground truth table is 100 (space unit) \* 300 (time unit).



**Figure 28.** The RD of the RS method and the CD method for 300 time units

Figure 28 shows the average RD across simulation iterations at each time unit. The RD decreases when the time increases for both the CD method and the RS method. The RDs of the CD method and the RS method drop significantly at the beginning before  $T_{10}$  and then reach a saturation point. The RD of the RS method has minor slope change beyond  $T_{10}$  and stays invariant after  $T_{160}$ . The saturation point of  $T_{10}$  shows a more efficient way to reach a high degree of accuracy with minimum number of reports equal to the total number of the space units. The performance graph with fusion point at every 10 time units and at every 100 time units is shown in Figure 29.

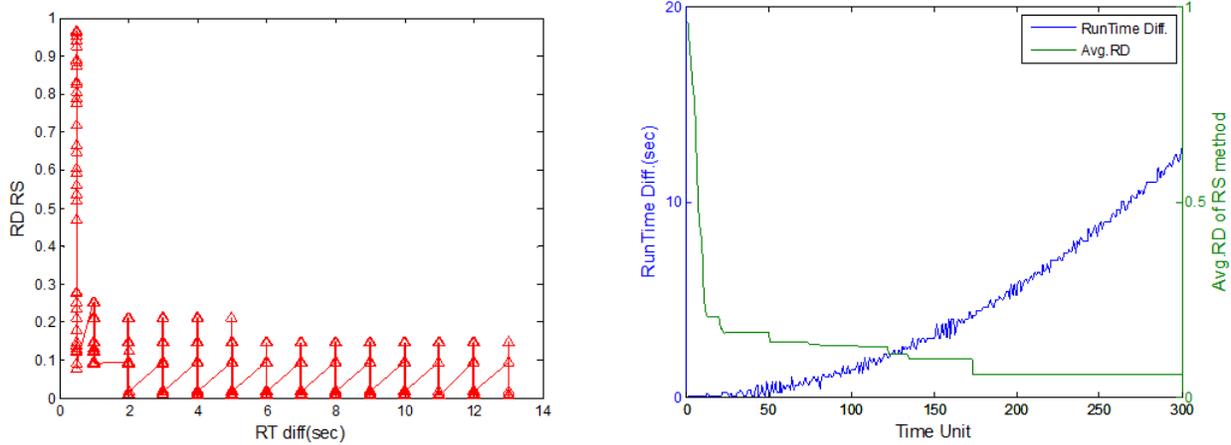


**Figure 29.** Avg. RD for fusion point at 10TU (left) and 100TU (right)

The fusion point at every 10 time units has similar performance with fusion point at every time unit; moreover, the fusion point at every 100 time units has sharper slope and lower RD. The average RDs in these fusion points are lower than the RD at each time unit since there are more space reports in the characteristic linear system; therefore, the system can achieve higher accuracy. Performing data fusion at an early point may generate the estimated result in a shorter period of time, but it requires more reports in the characteristic linear system in order to have good performance. If there aren't enough reports, chances are the RD will be high. Meanwhile, the computation time may increase when the number of reports in the system increases.

To explore the tradeoff between accuracy and efficiency, I evaluate the run time difference for each time unit. Run time difference (*RT diff.*) is the time difference in seconds between the time that the model of the characteristic linear system is generated and the time that the solution set is generated. From the left part of Figure 30, the time difference figure shows that the RD decreases as the RT diff. increases. The more reports are in the characteristic linear system, the more time the system requires to compute the solution set; however, the accuracy will be higher since information of target location is increasing. The RT diff. values dominate the

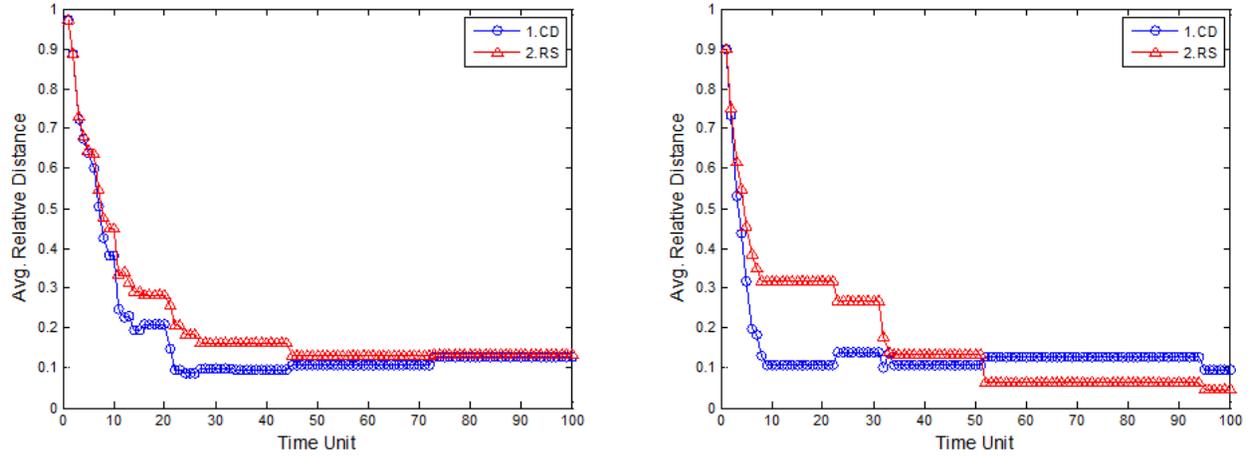
RT diff. distribution in the left part of Figure 30. I observe that the characteristic linear system generates solutions very quickly with a few reports most of the time. However, the accuracy varies significantly. The right part of Figure 30 shows the RT diff. is increasing but the average RD is decreasing when time unit increases. The average time difference is minor, and from 0 to 13 seconds; however, as the number of time units increases, the number of reports also increases, and the accuracy versus the computation time will be the trade off of this system.



**Figure 30.** Run Time difference of TU300

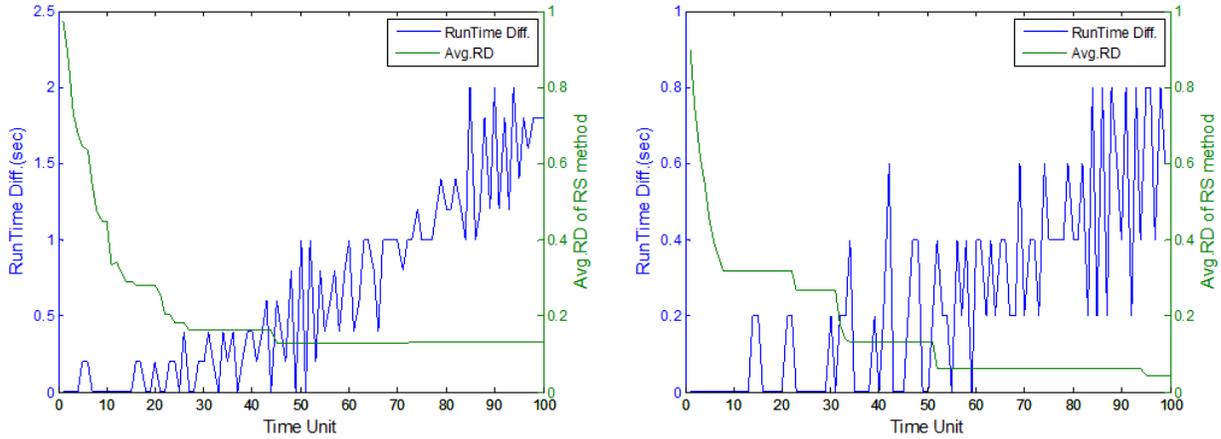
- **Simulation 2:** The size of space unit

The following simulation compares RD and RT diff. at different size of space unit. I hypothesize that the RT diff. will be similar if I keep the same size of time unit, but the RD may be different because of changing granularity of space units.



**Figure 31.** The RD of SU100(left) and SU25(right)

The left part of Figure 31 shows the comparison of the CD method and the RS method of 100 time units (100TU), and 100 space units (100 SU) and the right part of Figure 31 shows 100 time units (100TU) and 25 space units (25SU). The average RD drops to a steady level at around  $T_{25}$ . The average RD keeps decreasing as time unit increases and becomes lower than CD around  $T_{50}$ . With lower space unit or lower granularity of occupancy grid of space on the right part of Figure 31, the RD of the RS method is lower compared with the RD with larger space unit on the left part of Figure 31. Decreasing the total grid number of space units may indicate considerable performance advantage, which supports my hypothesis that more overlapping reports can be utilized to compute more accurate solution sets. In addition, my approach supports performing data fusion over different granularity of space corresponding to users' needs.



**Figure 32.** The Avg. RD and RT diff. of SU100 (left) and SU25 (right)

Figure 32 shows the comparison between RT diff. and average RD; the left part shows performance of TU100 and SU100, and the right part is for TU100 and SU25. At the beginning of simulation 2, I observed that lower space granularity could derive lower RD at the end. Considering the computation cost to achieve better performance, the RT diff. in Figure 32 did not show critical difference. The maximum RT diff. of SU100 is around 2 seconds, while the maximum of RT diff. of SU25 is around 0.8 second. The size of the space grid of SU100 is 4 times bigger than the size of SU25, but the RT diff. increases 2.5 times. The computation time is not much different between different space unit sizes since the number of reports increases at the same rate.

- **Simulation 3:** Event density and coverage of space report

In my temporal fusion simulation I have considered three major parameters, which are event density, report number and report duration. The report number is critical when the conflict degree of the report is large. As a result of the pilot multidimensional fusion simulation, we observe that the coverage of the space report affects the performance. For example, the space report covering the whole area provides more information than the space report covering one

specific area. Therefore, I consider the space report coverage as the main factor in the following simulation.

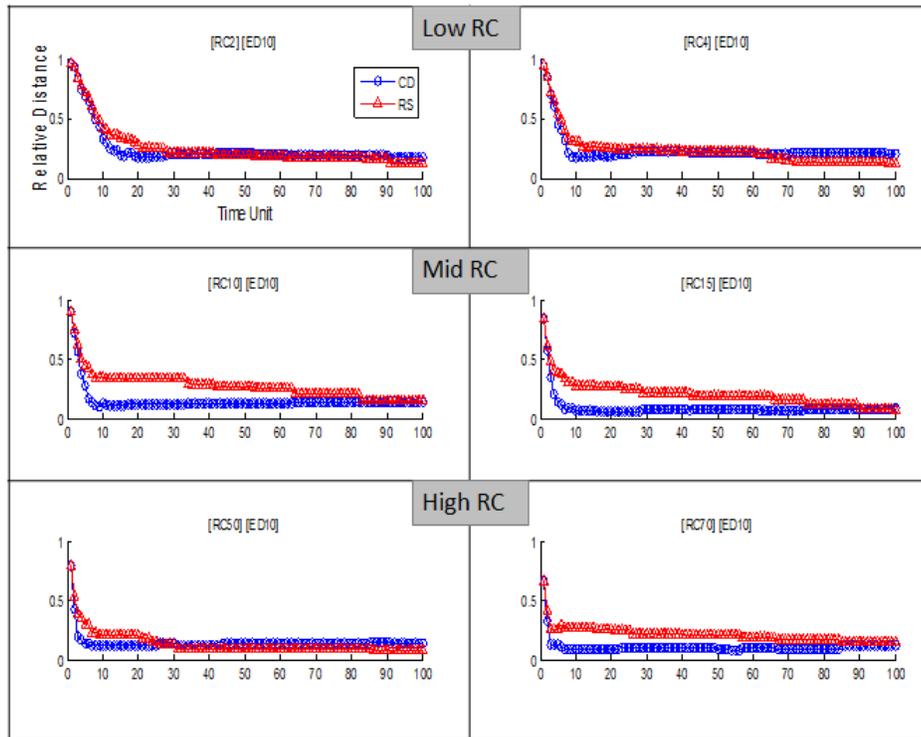
The event density, i.e. ED, and space report coverage (the same as report duration), i.e. RC, will be considered in the following experiment. I vary the event density to be 10 or 100 for each space unit, and vary the report coverage to be 2, 4, 10, 15, 50 and 70 space units for each report. The configurations are in the Table 15. The event density indicates how many targets are in each cell, which can also be referred to as target density. The report coverage specifies how many space units are included in the space report. The more the robots explores, the wider report coverage will correspond to its space reports. I consider a maximal time unit of 100 and maximal space unit of 100.

**Table 15.** Configurations of space report coverage

		Space Report coverage (RC)					
		Low RC		Mid RC		High RC	
		2	4	10	15	50	70
Event density (ED)	[10, 5]	Low overlap reports, low target density		Medium overlap reports, low target density		High overlapping reports, low target density	
	[100, 5]	Less overlap reports, high target density		Medium overlap reports, high target density		High overlapping reports, high target density	

For every configuration of simulation scenario, I performed multiple simulation iterations. Figure 33 shows RD for the CD method, and the RS methods for different RC of ED10. There are two different RCs in each group; RC2 (left) and RC4 (right) belong to the low RC group, RC10 (left) and RC15 (right) belong to the medium RC group, and RC50 (left) and RC70 (right) belong to the high RC group. I observe that the RDs of the RS method and the CD method share the same trend; the RDs are close to 1 at the beginning of time unit and decrease as

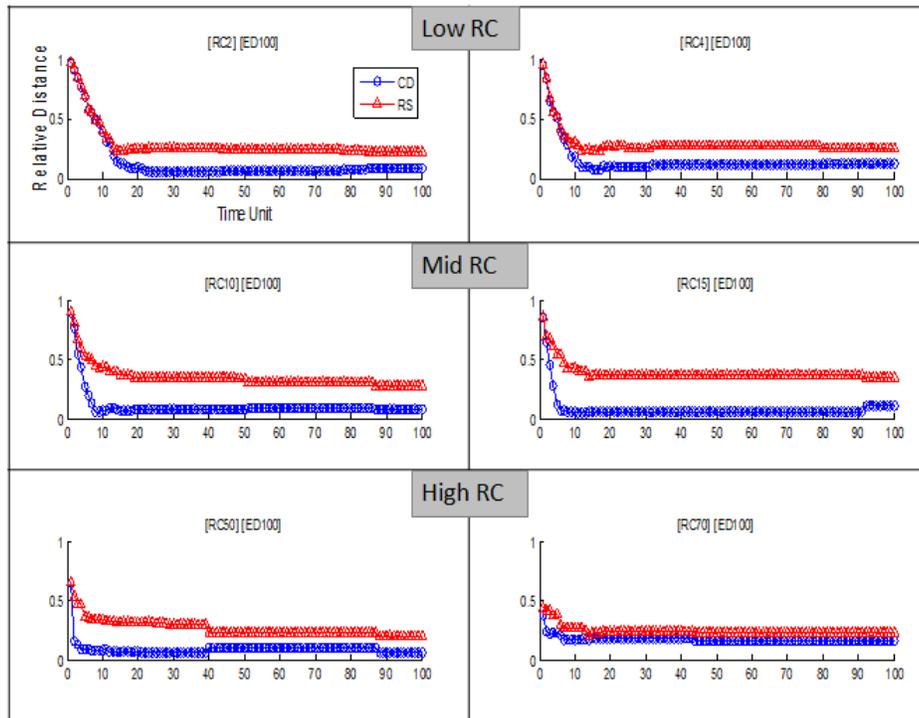
the time unit increases. In RC2, RC4 and RC50 the RS method has lower RD than the CD method at high time unit; however, in medium RC group and the RC70, the RS method has higher RD. In general, the RS method has similar performance with the CD method of ED10. However, the RD is lower in the high RC group for both methods.



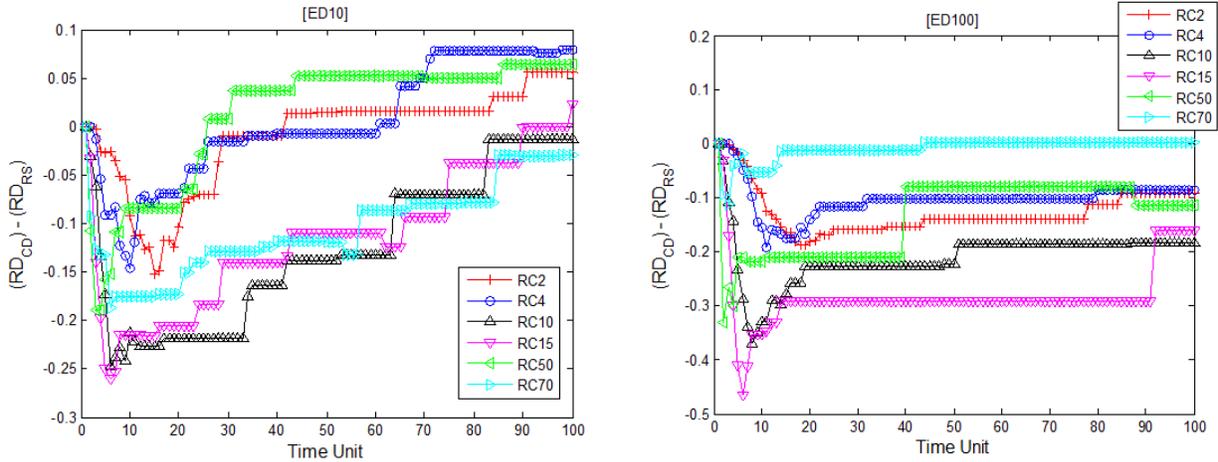
**Figure 33.** RC comparisons of ED10

The simulation result for configurations in ED100 is shown in Figure 34. In all cases of ED100, the RS method has a notable impact on the RD compared to the CD method. The high RC group, RC50 and RC70, has lower RD at the beginning time unit. The lowest RD of the high RC group in ED10 is about 0.6, and in ED100 is about 0.4. In the high RC group, although the RS method cannot outperform the CD method, it has lower RD in ED100 at the beginning of

time unit, and converges in a shorter time. I plot the RD difference between the CD method and the RS method of all scenarios in Figure 35; the difference will be negative if the CD method has the lower RD. In the scenario of ED10, the RC2, RC4 and RC50 become positive at around  $T_{25}$ ; in the scenario of ED100, the CD method is outperform in all configurations so the lines are all negative. The medium RC group, RC50 and RC70, has the largest RD difference, and the differences in other RC groups are less than 0.2. Overall, the RS method can outperform the CD method at the scenario of high RC at low ED.



**Figure 34.** RC comparisons of ED100



**Figure 35.** RD diff. between CD and RS of ED10 (left) and ED100 (right)

Next I compare the total RD values of the RS methods at ED10 and ED100 together. Figure 36 shows the comparison of RC size to RD performance for different ED size of the RS method. The Sum RD value returns the summation of the RD value across time units. The lower Sum RD corresponds to higher accuracy. I observe that the Sum RD for ED100 is higher than for ED10 in most groups of different RCs except the RC70; the groups of RC10, RC15 and RC50 have major differences of Sum RD value. From this result I could suggest that users choose either low RC of the report that provides more location information, or high RC that has more overlapping reports. Both can help the characteristic linear system to compute more accurate solution sets.

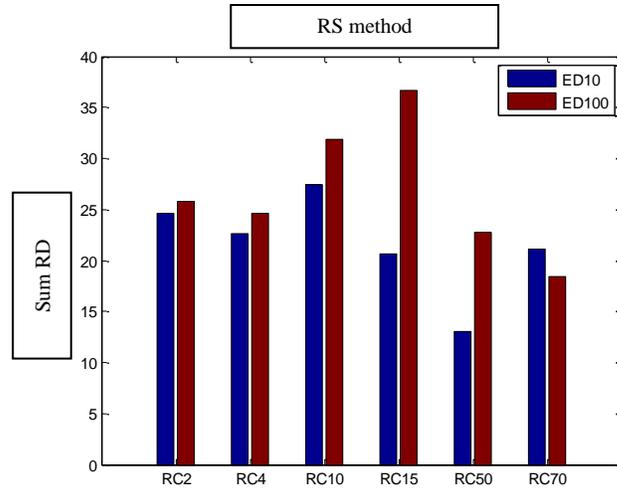


Figure 36. Sum RD of the RS method

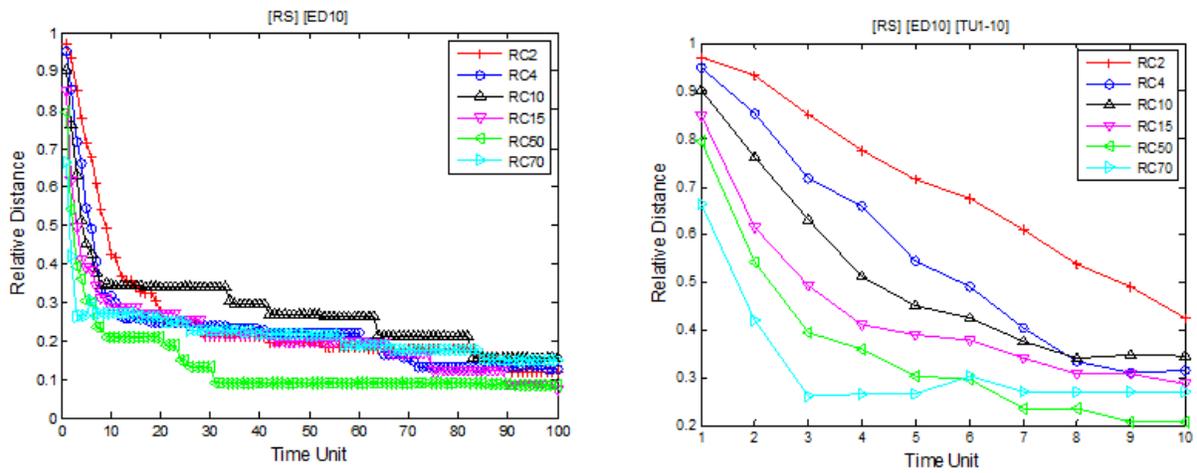
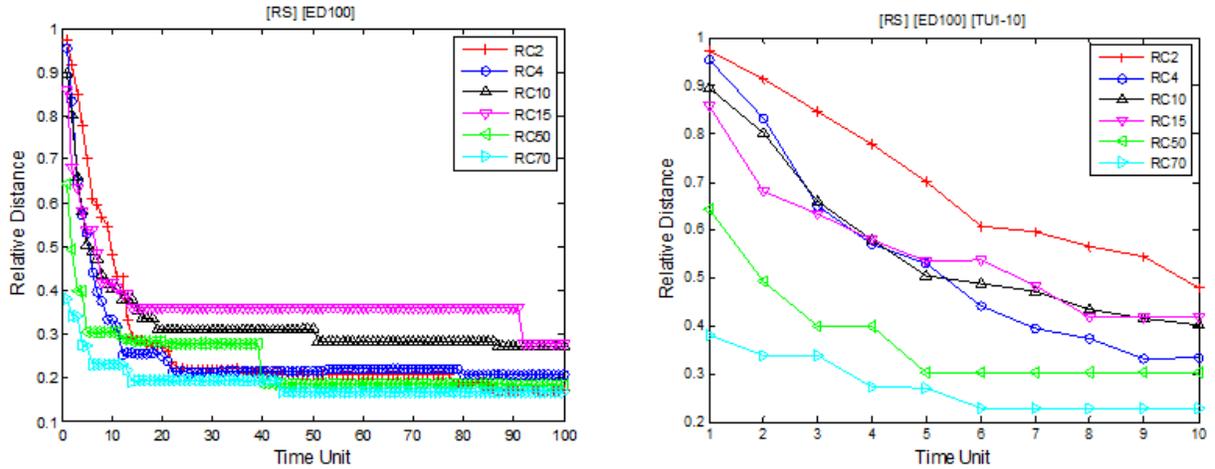


Figure 37. RD of the RS method of ED10

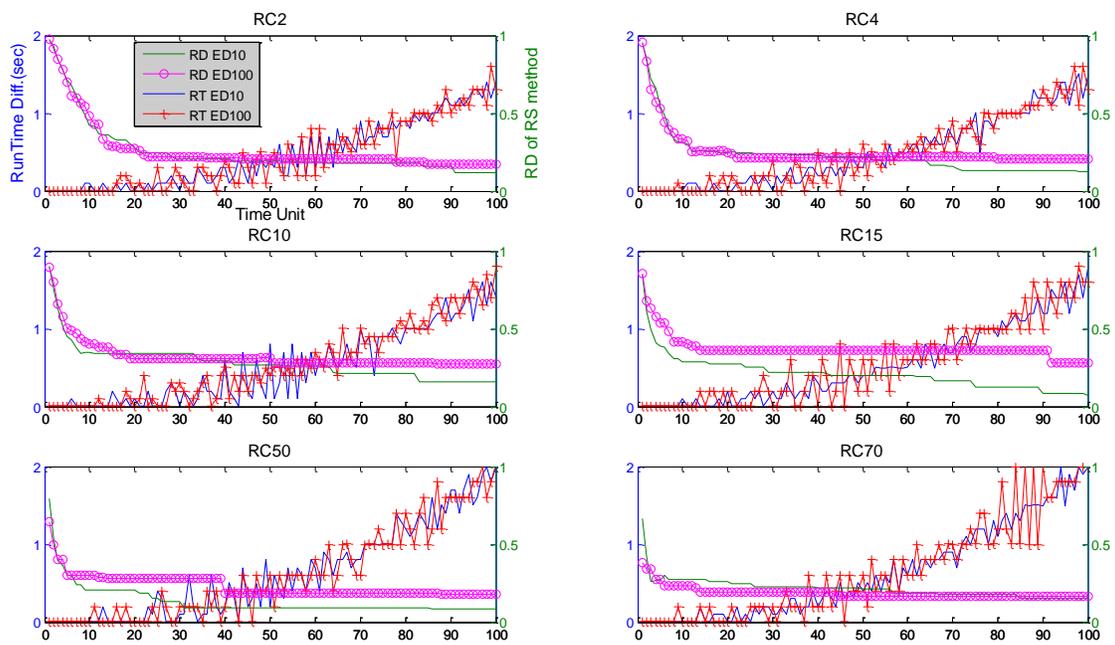


**Figure 38.** RD of the RS method of ED100

Figure 37 and Figure 38 report the RD value across different RC of the RS method in different ED respectively. A major observation here is a decrease of RD of variability in RC increases from  $TU_1$  to  $TU_{10}$ . For ED10 the RD varies between 0.66 and 0.96, while for ED100 the RD varies between 0.38 and 0.97. Both figures appear to have RD saturated beyond  $TU_{10}$  with variety of rates. Simulations reveal the RD is inverse relational to the RC size in both ED size; the RD decreases when RC size increases. Therefore, I would like to compare the RD and the RT diff. in order to choose the most efficient configuration.

In Figure 39, I compare the RT diff. with RD value for each RC size. In Figure 39, RT diff. in ED10 and ED100 share the same curve and often overlap. The differences of the Run time for each RC size are not significant. The range of run time is between 0 and 2 seconds. There is a significant effect for ED size in the group of RC10, RC15, RC50 and RC70,  $p < 0.01$ . In RC70, the RD of ED100 is slightly less in ED10; the RD of ED100 is higher than in ED10 in the rest of the groups. Figure 39 shows that the change of RT diff. is minor even though the ED is different. I believe this is because the run time is related to the size of the characteristic linear

system, or the number of report; therefore, the RT diff. is very close across RCs at each time unit. I observe that the lower RD value results from lower ED size with similar value and slope of the RT diff. The system can achieve better performance of target number estimation with lower target density in the search area. In other words, the dense target distribution will penalize the characteristic linear system by generating a high RD from high report value.



**Figure 39.** RD vs. RT diff.

Overall, this simulation introduces the following strategies of space fusion: to perform data fusion with a longer sampling time; to have lower space granularity; and to choose either low RC, which provides more location information, or to choose high RC, which has more overlapping reports. I conduct a series of comparisons to explore the tradeoff between accuracy and computational cost.

- **Simulation 4:** Probability distribution over the occupancy grid

I utilize two different methods, which are the basic method and the naïve Bayesian method, to compare their performance with the RS method in this simulation. The basic method considers any space unit overlapping with robot scan lines as a potential target location. The probability of a potential target in a time unit can be calculated as a ratio of number of target scans overlapping with the space unit to the total number of space units. For example, there are three cells  $C_1, C_2, C_3$  covered in one scan, and there are two cells  $C_3, C_4$  covered in another scan in Table 16. The probabilities for each cell in Scan 1 are  $1/3, 1/3$  and  $1/3$ ; and the probabilities in Scan 2 are  $2/4$  and  $1/4$ . The probability of a target in  $C_3$  is increased from  $1/3$  to  $1/2$  with 2 scans. It is expected that, as the number of scans grows, the estimated probability distribution converge to the actual distribution of targets over the occupancy grid.

**Table 16.** Probability distribution of basic method

<b>Scan ID</b>	<b>Covered Cell</b>	<b>Probability distribution</b>
<b>Scan 1</b>	$C_1, C_2, C_3$	$1/3, 1/3, 1/3$
<b>Scan 2</b>	$C_3, C_4$	$2/4, 1/4$

In another comparison I use Bayes' rule to estimate the probability distribution as a conditional probability  $P(V|VS)$ , where  $V$  is a property reflecting target presence in a cell,  $VS$  is a condition that the cell overlaps with a target scan (Zadorozhny & Lewis, 2013). This probability can be estimated as follows:

$$P(V | VS) = \frac{P(VS | V)P(V)}{P(VS | V)P(V) + P(VS | noV)P(noV)}$$

In the simulation, I set up three different levels of sparsity. Sparsity refers to the number of spatial cells that have zero targets, with up to 20 robots exploring the spatial environment of 36 space units of an occupancy grid within 20 time units. I use Jensen-Shannon Divergence (*JSD*) to measure similarity between two probability distributions of estimated and actual target distribution across spatial units. The lower *JSD* is better since the two probability distributions have less difference. Another measurement I use is area-under-the-curve (*AUC*) that reflects sensitivity about the results' true positive and false positive rate. The higher *AUC* means better performance since the true positive rate is higher.

The Figure 40 shows *JSD* for low sparsity (around 78 cells out of 360 have zero target), medium sparsity (around 147 cells), and high sparsity (around 195 cells) scenarios. I observe that both the basic method and the RS method are very close to each other when the sparsity is low. The performance of the basic method is more invariant under different sparsities, but the RS method has higher variability at medium and high sparsity. Overall, at the early search stage of time unit and at low sparsity the RS method can overperform the basic method.

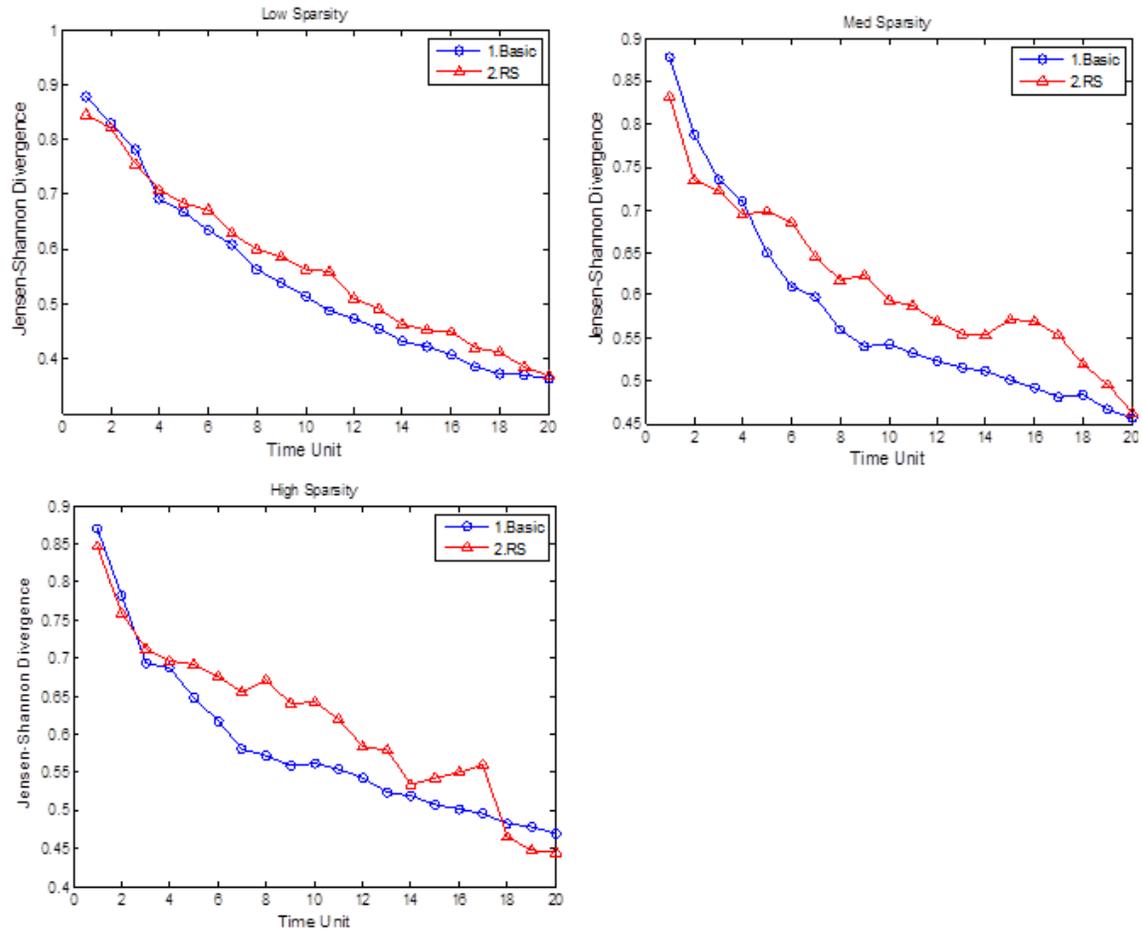
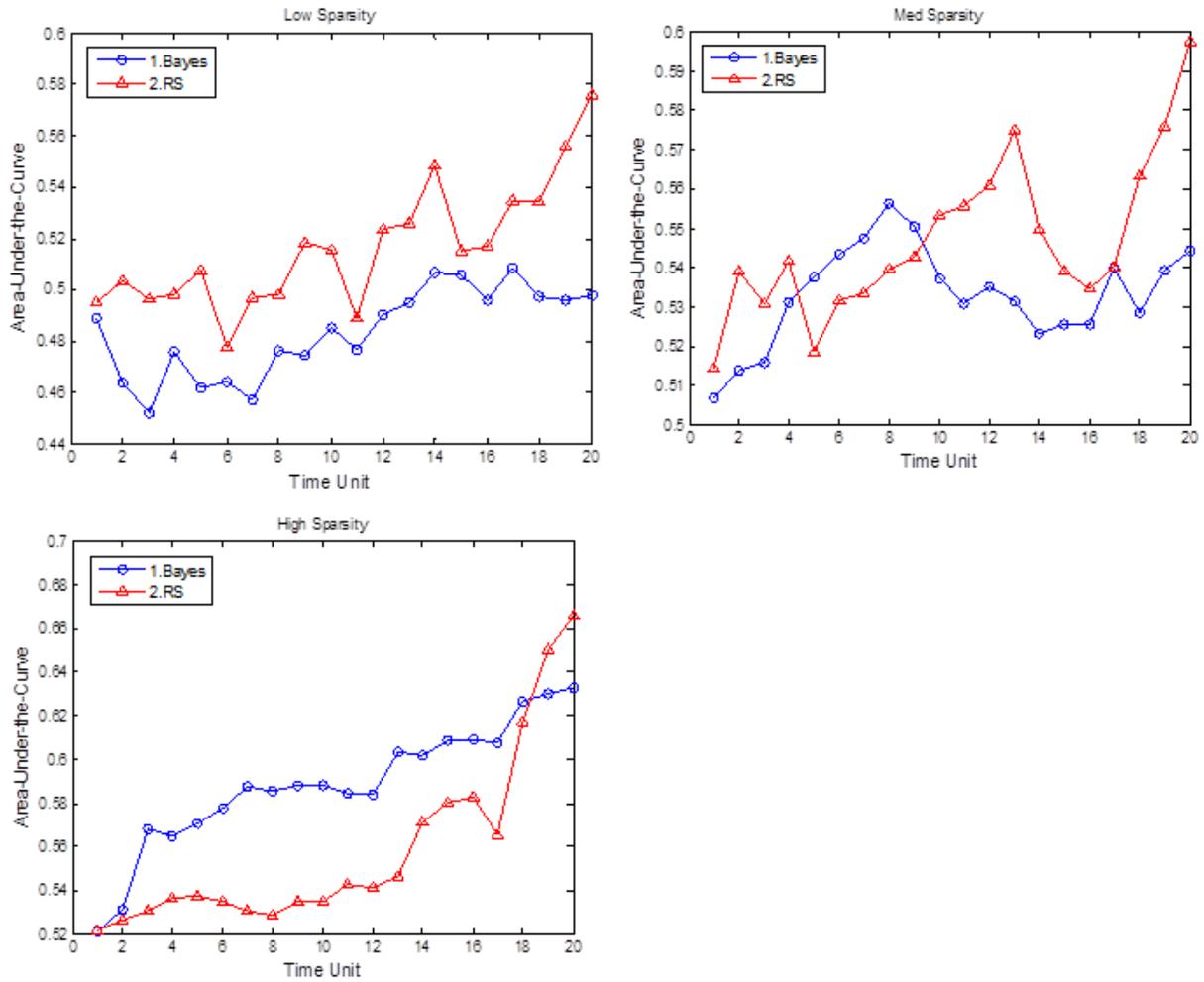


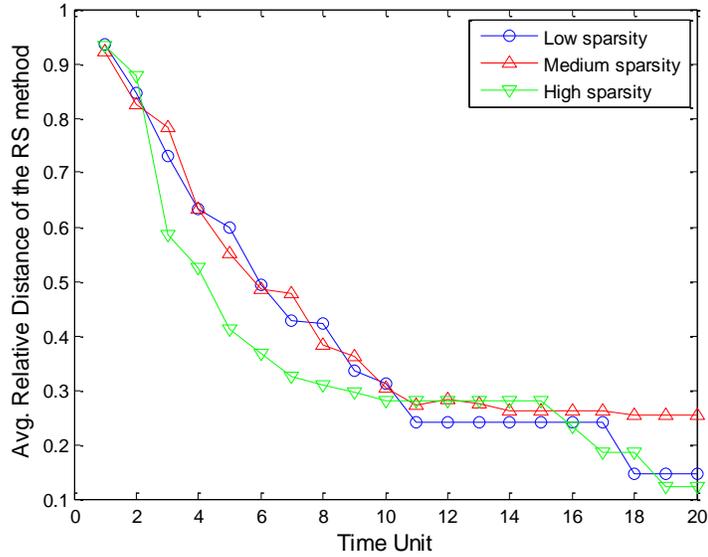
Figure 40. JSD of different sparsity

Figure 41 shows target detection sensitivity. I observe that the RS method outperforms the Bayes method under low sparsity. For medium sparsity, the performance of both methods varies. The RS method has better performance at the beginning as well as from the middle to the end of the time unit. For high sparsity, the Bayes method improves its performance with time. The RS method has lower AUC compared with the Bayes method except at the very end part of time unit. In general, both methods improve their performance with time. However, the performance of these two methods shows different trends; the Bayes method degrades as sparsity decreases, while the RS method performs better.



**Figure 41.** AUC of different sparsity

From these results I would recommend to use the RS method in the scenario of low sparsity environment. I compare the RD of the RS method under different sparsities. Figure 42 shows that the RDs in these three scenarios are very close, but the medium sparsity corresponds to the highest RD followed by low and high sparsity.



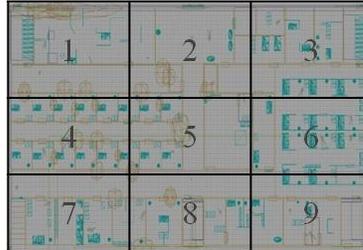
**Figure 42.** RD of the RS method of different sparsity

#### 5.4 PILOT STUDY OF TEMPORAL SPATIAL FUSION SYSTEM

In this section I will use a pilot example to illustrate how the proposed RS method performs temporal and spatial fusion for the target observation task. The space layout in Figure 43 shows an office like environment that has been divided into occupancy grids of small spatial units, which are also called space cells. The size and the numbering of each cell depend on designers' preference or area of interest. The number of targets in each cell is recorded continuously throughout the time interval. There are five time units and nine spatial cells in the example shown in Figure 44; this Time-Space matrix shows the actual number of targets and their locations. The targets are moving across cells during time units.

There are two constraints that I have introduced in Section 5.2: (1) all targets are being observed and (2) total number of target for each cell at  $T_{x+1} \leq$  summation of the number of targets for each cell's neighbor cells at  $T_x$ . I hypothesize that more constraints will help to detect

inconsistency and help to compute optimal solution sets. The total number of targets moving around the environment is forty-five.



**Figure 43.** Spatial layout in grids

		Time unit				
		T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	T <sub>4</sub>	T <sub>5</sub>
Spatial cell	C <sub>1</sub>	1	7	4	4	8
	C <sub>2</sub>	2	3	2	1	6
	C <sub>3</sub>	8	1	3	3	1
	C <sub>4</sub>	6	4	1	7	7
	C <sub>5</sub>	7	5	5	9	9
	C <sub>6</sub>	5	2	6	8	4
	C <sub>7</sub>	3	9	9	2	2
	C <sub>8</sub>	4	6	8	6	3
	C <sub>9</sub>	9	8	7	5	5
Sum		45	45	45	45	45

**Figure 44.** Number of target in each cell and time unit

I use two comparisons to evaluate the performance of the multidimensional fusion approach. For the first comparison, I compute the RD value of the RS method considering either the one-dimensional temporal or spatial fusion. In the second comparison, I compare the RD values given the combination of temporal and spatial fusion together. I expect two-dimensional

fusion will provide better accuracy of target observation and will have lower RD value. The following are the results of my comparisons.

(1) Report number and accuracy of temporal fusion

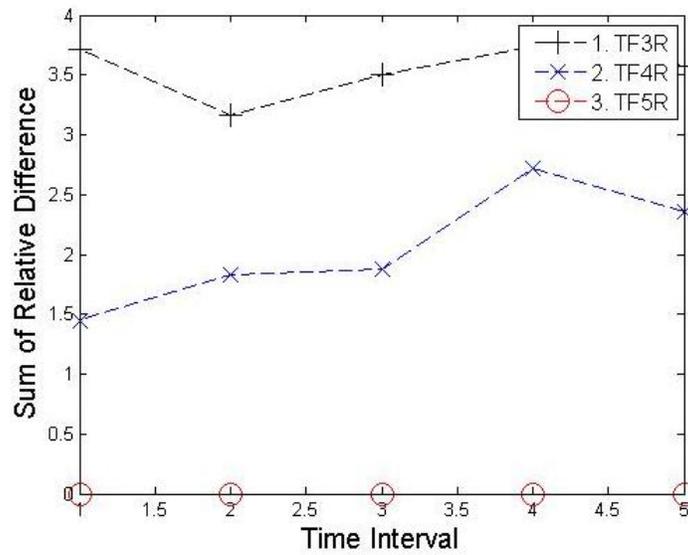
I perform temporal fusion across cells, and the RD value is calculated based on the actual cell value. I try to use the least amount of information from reports as possible to estimate the number of targets in each spatial interval. This can save the cost of time and computation. Therefore, I only consider the start time, end time, and total number of targets detected in the overall report duration.

**Table 17.** Example of temporal spatial fusion

<b>Report ID</b>	<b>Report Value</b>	<b>Time_from</b>	<b>Time_to</b>
<b>R<sub>1</sub></b>	V <sub>1</sub>	T <sub>1</sub>	T <sub>3</sub>
<b>R<sub>2</sub></b>	V <sub>2</sub>	T <sub>2</sub>	T <sub>5</sub>
<b>R<sub>3</sub></b>	V <sub>3</sub>	T <sub>3</sub>	T <sub>4</sub>
<b>R<sub>4</sub></b>	V <sub>4</sub>	T <sub>3</sub>	T <sub>5</sub>
<b>R<sub>5</sub></b>	V <sub>5</sub>	T <sub>4</sub>	T <sub>5</sub>

Reports record target number at given locations that can be cells or space intervals depending on the granularity of users' interest. For example, R<sub>1</sub> describes the number of targets as V<sub>1</sub> at different locations from time T<sub>1</sub> to T<sub>3</sub>. In Figure 45, I compare the accuracy between different numbers of temporal reports of the underdetermined linear system for predicting target numbers in each cell. The notation TF3R on the figure indicates that there are three temporal reports available from sensors or robots in time fusion. Similarly, TF4R and TF5R mean that there are four and five reports available respectively. The RD value is summarized across cells C<sub>1</sub> to C<sub>9</sub> at each time interval.

Figure 45 shows that the total RD decreases as the number of report available increases. The total RD decreases about 50% when the number of report increases from three to four. In addition, the total RD is zero when I have five reports for five time intervals. Therefore, I hypothesize that the number of reports that a linear system needs to generate the optimal report value estimation is the same as the number of interval. This also confirms my results (in Section 4.3) that show that the more reports there are, the better performance of the RS method. Having more reports provides the characteristic linear system with more equations.

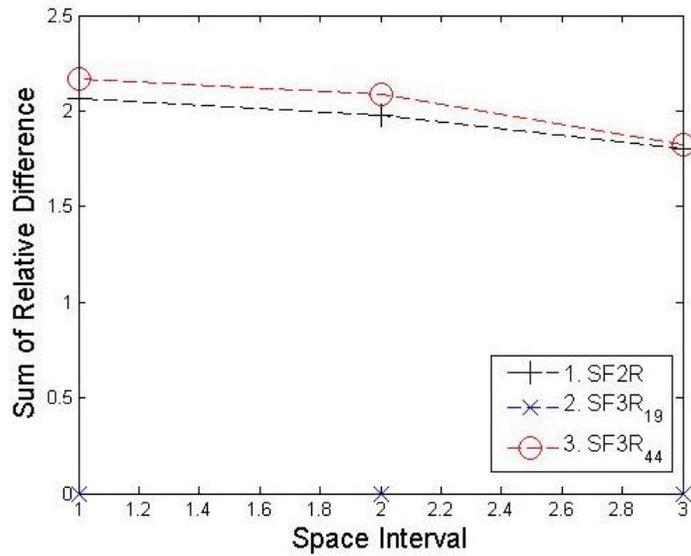


**Figure 45.** Accuracy of different number of report

(2) Report type and accuracy of spatial fusion

In this example I compare RD of spatial fusion for three conditions that are (a) two reports (SF2R), (b) three reports, two of which from condition (a) and the other report covers all space units from  $C_1$  to  $C_9$ , SF3R<sub>19</sub>, and (c) three reports, two of which from condition (a) and the other report covers a given cell  $C_4$ , SF3R<sub>44</sub>). In Figure 46, the SF3R<sub>19</sub> performs better than SF3R<sub>44</sub> even though these two conditions consider the same number of reports. In addition, the total RD

of three reports SD3R<sub>44</sub> is even higher than the two reports condition SD2R. I hypothesize that the increase of RD is proportional to the increase of intervals. Therefore, providing more information to the linear system in order to have better accuracy by increasing report number is reasonable strategy, but I also need to take the report structure into account.

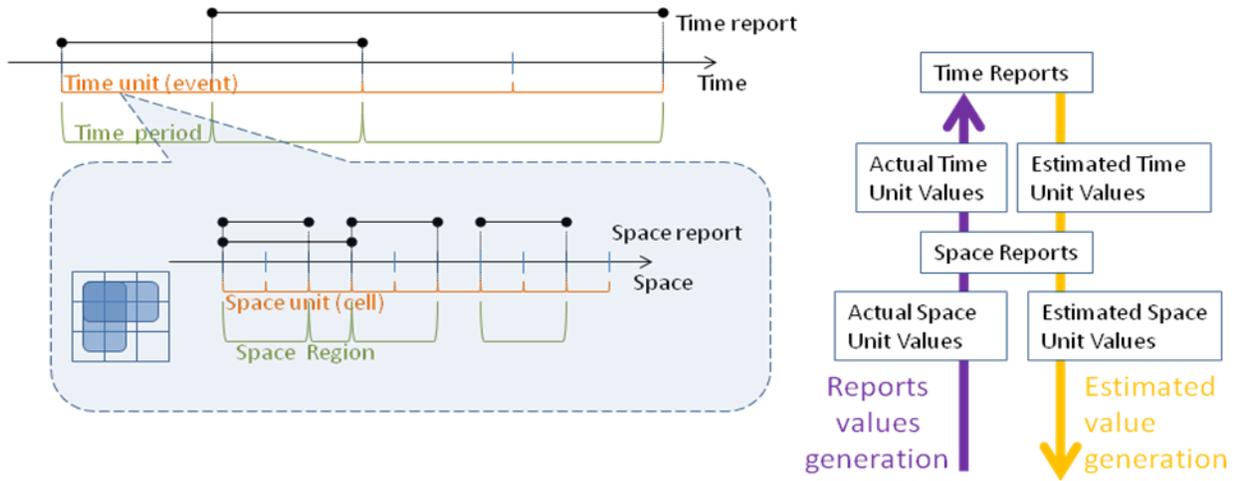


**Figure 46.** Example of spatial fusion

### (3) Combining with temporal fusion and spatial fusion

To perform the two-dimensional temporal and spatial fusion, users have to decide which fusion strategy is to be performed first. The strategy of how to determine the sequence of performing different types of data fusion depends on the report number and interval size. For example, if we perform temporal fusion with five reports first, the estimated value for each time interval will be close to the actual values. This improves the result accuracy for the following spatial fusion because the result of temporal fusion provided to it is quite accurate.

## 5.5 MULTIDIMENSIONAL TEMPORAL SPATIAL INFORMATION FUSION



**Figure 47.** Two-dimensional reports generation

The report generation and data fusion process of the multidimensional temporal and spatial fusion are shown in Figure 47. The purple arrow indicates the generation process of temporal and spatial reports and ground truth tables; the yellow arrow shows the computation sequence of unit value estimation. In reports values generation step, the values of space reports are aggregated from the actual space unit values of the given space region. The time unit values are the aggregated value from observed areas at each time unit. And in the same way, the value of time reports are from the given time period. In the estimated values generation step, we first generate the solution set for the estimated time unit value using the RS method from time reports, then compute the solution set for the estimated space unit value by space reports. The two-dimensional fusion processes can be broken down as follows: use the characteristic linear system from time reports to generate estimated time unit value, and then use the characteristic linear system from space reports as well as the fusion result of the estimated time unit value to compute

the solution set of the second characteristic linear system. We use the temporal fusion result, which is the estimated time unit value, to approximate the space unit values; therefore, the accuracy and complexity of the system may be affected.

Here I use a small characteristic linear system for multiple regions at  $T_1$  to illustrate the two-dimensional data fusion process. Take the ground truth value of space unit at  $T_1$  in the two-dimensional pilot study as an example; the space reports only cover partial areas and I am going to find out the values in each space unit at  $T_1$ . The ground truth table of the four space reports at  $T_1$  is in Table 18. There are 4 space reports covering partial space units and the table shows the number of target in each space unit in  $T_1$ . Because I am considering the condition of dynamic targets, the ground truth in each space unit will be different at other time units.

**Table 18.** Ground truth of space reports at  $T_1$

		Time Unit $T_1$				
		Ground truth	Space Report			
			S1	S2	S3	S4
Space Unit	$C_1$	1	1	1	0	0
	$C_2$	2	1	1	0	0
	$C_3$	8	0	1	0	0
	$C_4$	6	0	0	1	0
	$C_5$	7	0	0	1	0
	$C_6$	5	0	0	0	0
	$C_7$	3	0	0	0	1
	$C_8$	4	0	0	0	1
	$C_9$	9	0	0	0	0
Sum		45	3	11	13	7

The characteristic linear system from the time reports is

$$\begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 155 \\ 97 \\ 124 \\ 162 \end{bmatrix}$$

and the corresponding solution set for each time unit is

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 31 \\ 66 \\ 58 \\ 38 \end{bmatrix}.$$

The estimated value for time unit  $T_1$  is 31, which is going to be added in the second characteristic system. Therefore, the characteristic linear system of the four space reports and the estimated  $T_1$  number is

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \end{bmatrix} = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \end{bmatrix} = \begin{bmatrix} 3 \\ 11 \\ 13 \\ 7 \\ 31 \end{bmatrix}.$$

The solution set for every space unit at  $T_1$  is

$$[x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7 \ x_8 \ x_9]' = [3 \ 0 \ 8 \ 13 \ 0 \ 0 \ 7 \ 0 \ 0]'$$

The values for each time unit are the aggregated value from all space units at each time unit. The estimated time unit value of  $T_1$  will be the summation of estimated values of all space units, which is 31. The actual value and the estimated value for each time unit are

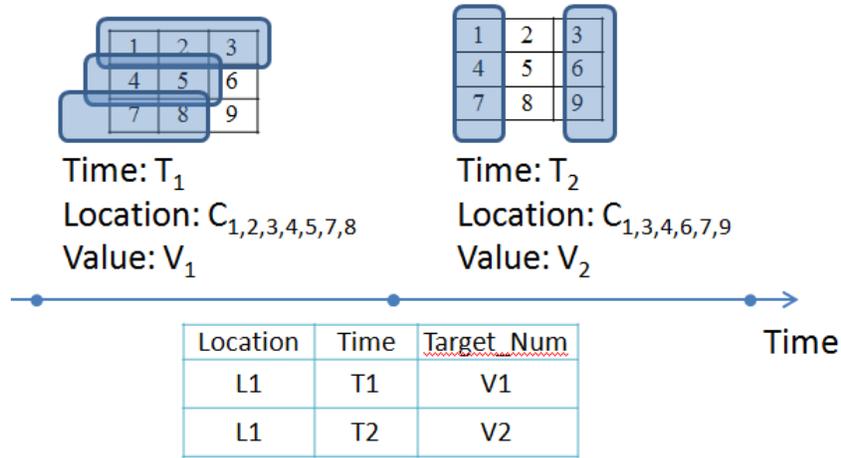
$$\begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \end{bmatrix} = \begin{bmatrix} 45 \\ 45 \\ 45 \\ 45 \\ 45 \end{bmatrix} \text{ and } \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 31 \\ 36 \\ 45 \\ 41 \\ 44 \end{bmatrix} \text{ respectively.}$$

In addition, the actual and the estimated value for each space unit at time unit  $T_1$  are

$$\begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6 \\ C_7 \\ C_8 \\ C_9 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 8 \\ 6 \\ 7 \\ 5 \\ 3 \\ 4 \\ 9 \end{bmatrix} \text{ and } \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \\ 8 \\ 13 \\ 0 \\ 0 \\ 7 \\ 0 \\ 0 \end{bmatrix}.$$

Therefore, the RD of temporal fusion of each time unit is [0.3111, 0.2, 0, 0.0889, 0.0222], and the summation of all RDs is 0.6222. On the contrary, the RD of spatial fusion in  $T_1$  is [0.667, 1, 0, 0.5385, 1, 1, 0.5714, 1, 1], and the summation of all RDs is 6.7766.

This example shows the characteristics of the data set contains moving robots and targets for which time reports of the given locations have different numbers of targets at different times. In this condition, I rely on collaborative data from different robots to achieve general information of the whole environment. Each robot's log may contain data of the same location, but with different number of targets at different times. Figure 48 illustrates the target number changes across time and locations. At time unit  $T_1$ , there are three space reports that record the space unit 1, 2, 3; space unit 4, 5; and space unit 7, 8. The aggregated space report value  $V_1$  is the time report value for  $T_1$ . Similarly, there are two space reports covered space unit 1, 4, 7 and space unit 3, 6, 9 at time unit  $T_2$ . The aggregated value from space reports  $V_2$  is also the time report value for  $T_2$ . In summary, since the location  $L_1$  covered nine space units and the targets are moving around the closed space, I will have different target numbers at different times for the same location.



**Figure 48.** Two-dimensional data of dynamic target

The configurations of following simulations are the accessibility of targets and robots (i.e., dynamic or static), the number of reports, the number of intervals, and the value of reports. I vary the report number and report duration of 5 and 20 in the simulation in the same way I conducted the one-dimension temporal information fusion with 10 targets randomly distributed in each cell at each time unit. Each time report value comes from aggregating the values of reported time unit during the given time interval. The reported value of each time unit represents the statistical summation by aggregating the reported values in each space unit. I use Normal distribution of the value in each configuration. The descriptions and configurations of the experiment design are described in Table 19. Take the first row as an example; there are 5 time reports and the duration for each report is up to 5 time units. Each time unit contains up to 5 spatial reports.

**Table 19.** Configurations of two-dimensional fusion

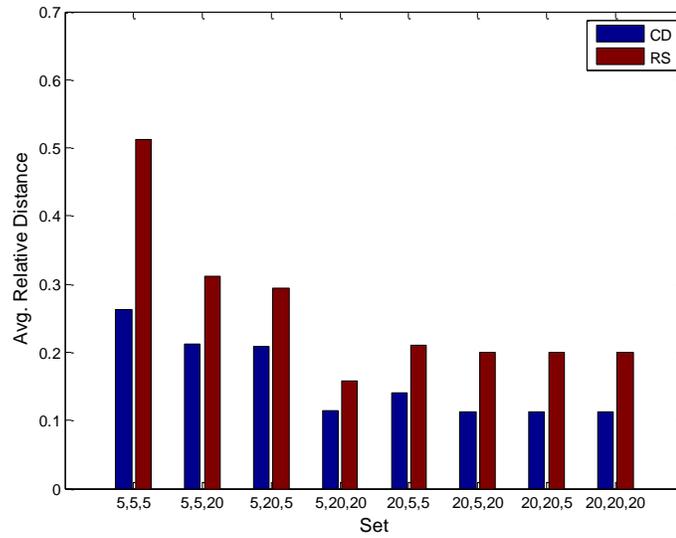
<b>Time Report Number</b>	<b>Time Report Duration</b>	<b>Space Report Number</b>	<b>Description</b>	<b>Scenario</b>
5	5	5	Few short time reports and sparse spatial overlap	Few intervals with low report values
5	5	20	Few short time reports and dense spatial overlap	Few intervals with low report values
5	20	5	Few long time reports and sparse spatial overlap	Few intervals with high report values
5	20	20	Few long time reports and dense spatial overlap	Few intervals with high report values
20	5	5	Many short time reports and sparse spatial overlap	Many intervals with low report values
20	5	20	Many short time reports and dense spatial overlap	Many intervals with low report values
20	20	5	Many long time reports and sparse spatial overlap	Many intervals with high report values
20	20	20	Many long time reports and dense spatial overlap	Many intervals with high report values

The simulation considers targets moving randomly in a closed space. The space is evenly divided into 9 space units (i.e. cells) and the numbers as well as the coverage of space reports are randomly generated. The time series is evenly divided into 24 time units. The ground truth of the number of total targets across the space at each time unit (i.e. event) is the aggregated number of space reports. Then the number of targets in this closed space for specific time duration can be computed. Every entity value in each time unit is aggregating from all detected space unit values. The ground truth table will be a 9 (space unit) \* 24 (time unit) matrix. Figure 47 shows an example of the hierarchical structure of the multidimensional data. In most cases, I can use reports which have numbers less than the number of units to recover the values for all units. However, the estimated value of units may not be accurate if the number of report is not enough, or the reports did not cover all units well. In my preliminary study of temporal fusion, I focus on figuring out the number of cases in each time unit; however, in the two-dimensional temporal

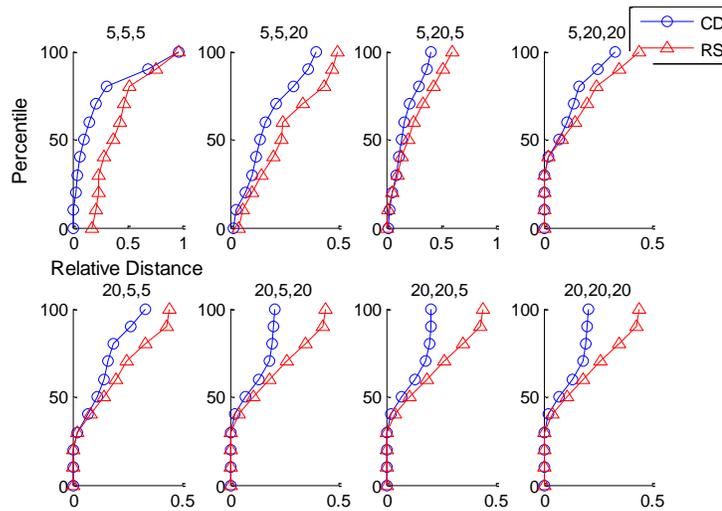
spatial fusion I am able to compute the case number and where these cases are located at each time unit.

- **Simulation 1:** Compare the performances of the CD method and the RS method

To compare the performance of one-dimensional temporal fusion and two-dimensional temporal spatial fusion of the same data set, I perform a simple simulation. These configurations are illustrated in Table 19. First, I compare the RDs of time units from the RS method and the CD method. Figure 49 shows the performance of time fusion (i.e. the first fusion result of the two-dimensional fusion). The RS method leads to larger average RD compared with the CD method. The comparison is based on the average of value difference between the actual values and the estimated value of each time unit and the optimal CD threshold is selected for each configuration. In my earlier simulation of time fusion with the number of report 20 and 100, the RS method required larger number of report to have better performance. The percentile plot of these two methods is shown in Figure 50. I observe that the configuration [5, 5, 5] shows a significant difference between the CD, and that the RS method and the RDs values are closer in other configurations.



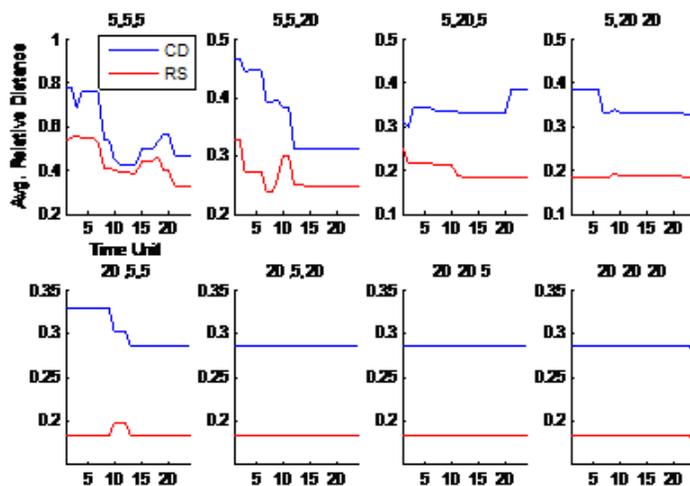
**Figure 49.** RD across time units after temporal fusion



**Figure 50.** Percentile plot across time units after temporal fusion

In the second step, I analyze the performance of space unit at each time unit at the micro level. The targets are randomly distributed across 9 space units and keep changing location across 24 time units. The comparison of these two methods with 8 configurations is shown in

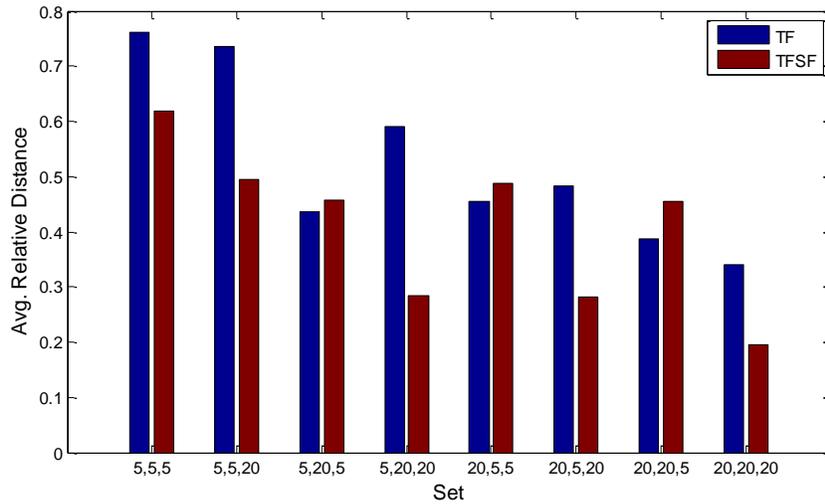
Figure 51. The value of average RD is the mean value of all space units in each time unit. The performances of the RS method with large number of time reports are better than small number of time reports since the estimated values of the time unit are more accurate. The CD method has weaker performance than the RS method of the two-dimensional fusion in each configuration. The estimated space unit value of the RS method is computed using the second characteristic system from spatial reports, and the estimated value of the given time unit. The estimated space unit value of the CD method is the estimated time unit value evenly divided by the number of covered cells because we do not have any prior information of the space unit distribution. For example, if the estimated target value of  $T_1$  is 18, then the estimated target number for each space unit at  $T_1$  will be 2 (i.e.  $18$  (case number in  $T_1$ ) /  $9$  (number of total cell) =  $2$ ). In one-dimensional data fusion, I assume that the estimated space unit values are Uniform distributed. In two-dimensional data fusion, the space report values in the characteristic linear system are based on the estimated value of time unit and space reports to compute the estimated value of space units. Therefore, the RS method could have better accuracy than the CD method.



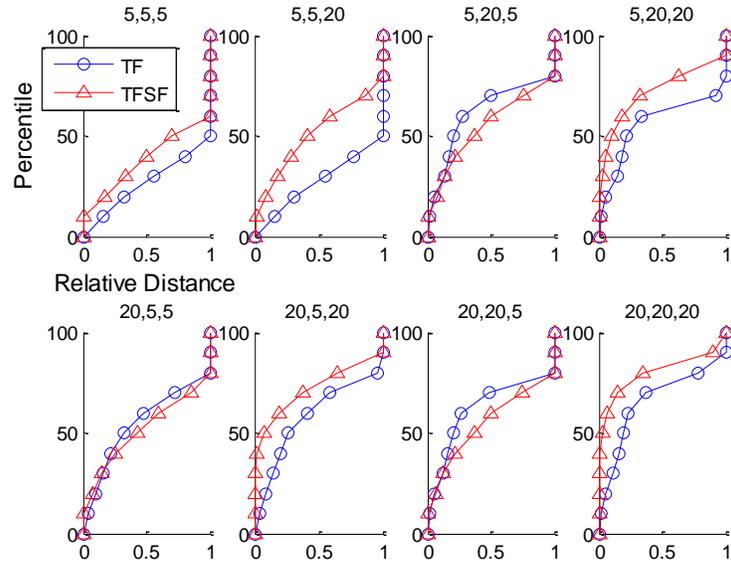
**Figure 51.** RD across space units after temporal spatial fusion

- **Simulation 2:** Compare the performance of the RS method in one-dimensional fusion and two-dimensional fusion

This simulation considers the similar configurations: 24 time units and 9 space units with 5 dynamic targets in each space unit. The parameters are the number of time report, the length of time report, and the number of space report varying between 5 and 20. In my hypothesis, the performance of the RS method in time unit level and in space unit level should be the same if there are enough temporal and spatial reports with good coverage. The performance comparison between one-dimensional and two-dimensional fusion uses the RS method for the same data set shown in Figure 52, and the percentile figure shown in Figure 53.



**Figure 52.** Avg. RD of TF and TFSF fusions

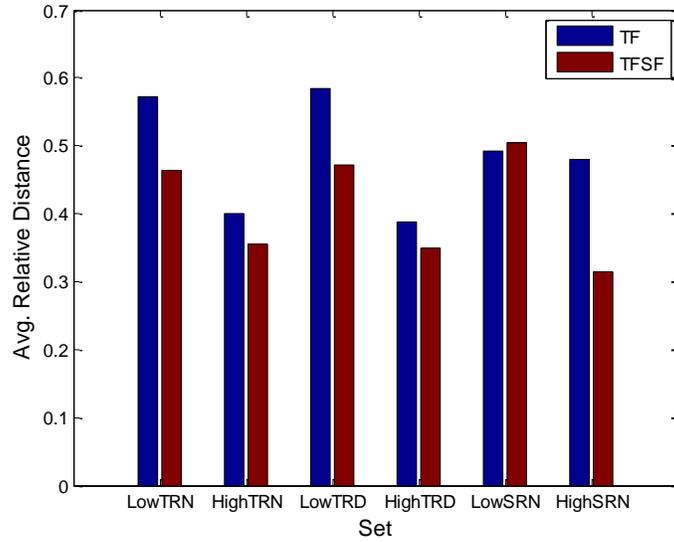


**Figure 53.** Percentile RD of TF and TFSF fusions

The configuration [5, 5, 5] has the fewest time reports, shortest duration of time reports, and fewest space reports; therefore, it has the highest RD in the figure. Comparing the configurations [5, 5, 5] and [5, 5, 20] together, these two figures have similar RDs for the TF fusion, but the RD of [5, 5, 20] is lower for TFSF fusion because it has higher number of space reports. Comparing the configurations [5, 5, 5] and [5, 20, 5], the configuration [5, 20, 5] has lower RD in both TF and TFSF fusions since it has more report overlapping. Comparing the configurations [5, 20, 5] and [5, 20, 20], the RD of TF fusion in these two configurations are close, but the RD is significantly lower in [5, 20, 20]. This shows that increasing the space report number will have lower RD in TFSF fusion. For the configurations [5, 5, 5] and [20, 5, 5], they both have short time reports and fewer numbers of space reports, but the scenario [20, 5, 5] has lower RD in both TF and TFSF fusion. I hypothesize that this is because the increasing of time report provides better accuracy of solution set; therefore, the RD is lower in the TFSF fusion. Comparing groups of configuration with the same length of time report and same number of space report such as [5,

5, 20] and [20, 5, 20], [5, 20, 5] and [20, 20, 5], as well as [5, 20, 20] and [20, 20, 20], I found that the higher number of time report usually results in lower RD in both TF and TFSF fusions. However, the RD of TFSF fusion in [20, 20, 5] is slightly higher than the RD in [5, 20, 5]. For the configurations [20, 5, 5] and [20, 5, 20], the RDs of TF fusion are similar, but the RD of [20, 5, 20] in TFSF fusion is better than [20, 5, 5]. There is the same performance tendency for the configurations [20, 20, 5] and [20, 20, 20]. There are some scenarios which have higher RD in TFSF fusion than in TF fusion: [5, 20, 5], [20, 5, 5], and [20, 20, 5]. Additionally, these all have fewer numbers of space reports.

In order to get a better understanding of the effects of the configuration, I compare the RD of both TF fusion and TFSF fusion in the group of Low Time Report Number (LowTRN), High Time Report Number (HighTRN), Low Time Report Duration (LowTRD), High Time Report Duration (HighTRD), Low Space Report Number (LowSRN), and High Space Report Number (HighSRN). Figure 54 shows that the RD of TFSF fusion in High groups is lower than in the Low groups. The RD of TF fusion shows a similar trend except in the SRN group. Therefore, I observe that the number of space reports will not affect the performance of TF fusion much, but will affect the performance of TFSF fusion.



**Figure 54.** Avg. RD in Low/High TRN/TRD/SRN

In summary, the factors of time report number, time report duration, and space report number all affect the fusion performance. To have higher numbers of these factors improves the performance of the characteristic linear system. In addition, the effectiveness of the time report number and duration is higher than the space report number in the CD method.

## 6.0 CONCLUSIONS

### 6.1 DISCUSSIONS AND APPLICATIONS

This dissertation covers three major topics: temporal fusion, spatial fusion, and multidimensional temporal spatial fusion. For the one-dimensional temporal fusion, I have implemented my system with two studies of inconsistency detection and data fusion. The inconsistency detection study contains inconsistency occurrence detection and inconsistent reports identification. I have observed that the  $C$  value and the  $\delta$  value could be used as an indicator of the existence of inconsistent reports. The data fusion section considers report value estimation and accuracy improvement. In addition, an efficient approach of using the underdetermined linear system to detect inconsistency and to perform data fusion of large amounts of data is required. I have found that the RD decreases as the number of events increases, and the RS method outperforms the CD method when the report number increases. Therefore, the RS method is a better option for data fusion of reports with more overlapping, more subsumptions, and a large report number.

For the one-dimensional spatial fusion, I have implemented my approach for simulated multi-robot search and rescue task. In simulation 1, I observed lower RD at high fusion points since there are more space reports in the characteristic linear system. As a result, the system can achieve better accuracy. Meanwhile, the computation time may increase when the number of reports in the system increases. In my simulation with 300 time units and 9 space units, the time

difference was less than 13 seconds. The time difference is minor in this configuration; however, having a smaller characteristic linear system in order to have better performance and efficient computation time is suggested. In simulation 2, my results showed that the RD of the RS method is lower with smaller occupancy grid of space compared to the RD with larger space grid. Decreasing the number of grid cells (number of space units) indicates considerable performance advantage, which supports the hypothesis that more overlapping reports can be utilized to compute more accurate solution sets. Results in simulation 3 revealed that the RD is inverse relational to the RC size at different event densities (ED10 and ED100). The system can achieve better estimation of target number with lower target density in the search area. These results suggest strategies that include performing data fusion over a longer period of sampling time, using lower space granularity, and choosing either low RC, which provides more location information, or choosing high RC, which has more overlapping reports. In simulation 4, both the basic method and the RS method are very close to each other when the sparsity is low, and the RS method outperforms the basic method at the early search stage (i.e., at lower time units). In addition, the RS method also outperforms the Bayesian method under low sparsity; however, the performance of RS method degrades as sparsity increases. My approach implements major functionalities of space fusion and supports data fusion over different granularity of space units corresponding to users' needs. I introduced an automatic information fusion method for multi-robot search and rescue representing overlapping reports from robots as an underdetermined linear system (characteristic linear system). The solution sets from the characteristic linear system efficiently approximates number of targets in particular locations. My simulation-based study demonstrated high performance of the proposed approach.

I also implemented an approach for two-dimensional temporal and spatial fusion to test my system with different types of data sets. From the simulation results in pilot studies we were able to understand the effect of the event density, report number, report duration, and total number of time unit on system performance. More reports and higher overlap of report structure (i.e. the smaller number of time unit) elicit better performance of my characteristic linear system. In addition, I would like to explore other algorithms that can overcome the deficiency of report numbers so I can apply appropriate data fusion strategies depending on the number of reports. My goal is to find an approach that can estimate interval values accurately, satisfy most constraints of linear equations, timely detect the inconsistency occurrence, and adjust the difference between estimated and actual values.

This study was conducted to explore the effectiveness of my proposed RS method of inconsistency detection and data fusion in multidimensional data. I would like to make the following observations related to my major research questions hypothesis:

- **Research question 1:** How to detect inconsistency in temporal and spatial data?

**Hypothesis:** My method can be used to indicate which report(s) has the higher degree of inconsistency, or to indicate which report(s) causes the inconsistency. Therefore, the user can spend less time finding the targeted problem reports.

**Observation:** The obtained results showed that the number of inconsistent reports detected by the characteristic linear system using the RS method and the number of actual inconsistent data reference matches well under any configuration of conflict/report/data reference density in the temporal simulation. In addition, after implementing the proposed approach, I detected the occurrence of fifty-seven conflicts all of which were confirmed with inconsistent report values in Tycho database. The proposed approach can be used to indicate the degree of

inconsistency, or the conflict with the nonzero  $C$  values and  $\delta$  values. In the simulation of the effect of the number of conflicting reports on the degree of inconsistency, the  $\delta$  value and the  $C$  value both increase when inconsistency increases; however, the  $\delta$  value does not always increase proportionally with higher degree of inconsistency. Therefore, the  $\delta$  value indicates the existence of inconsistency, but cannot represent the degree of conflict.

- **Research question 2:** How can inconsistent temporal and spatial data be processed?

**Hypothesis:** I can detect inconsistency for different configurations (i.e. overlap, subsumption, number of reports, etc) of temporal and spatial reports by the estimated value generated from the characteristic linear system.

**Observation:** I have implemented my system for inconsistency detection and data fusion. The nonzero  $C$  values and  $\delta$  values represent the existence of inconsistency and the solution sets generated by the characteristic linear system provide approximate interval values. In temporal fusion simulation, I used the RD for performance measurement to compare the estimation error, which is the difference between the summation of the actual values and the estimated total value of the event values across each interval. In the configuration of different event size, the RD is lower for both the CD and the RS method for 1000 total number of time units when compared with 150 total number of time units. Moreover, for all conditions of report number 100 (many reports), the RS outperforms the CD. In spatial data fusion, simulations revealed that the RD of the CD method and the RS method are very close with respect to the following strategies of space fusion: to perform data fusion over a longer period of sampling time; to have lower space granularity; and to choose either low RC, which provides more location information, or to choose high RC, which has more overlapping reports. In two-dimensional spatial temporal fusion, the factors of time report number, time

report duration, and space report number all affect the fusion performance. Having a higher number of these factors improves the performance of the characteristic linear system. In addition, the effectiveness of the time report number and duration is higher than the space report number.

- **Research question 3:** How can the inconsistency detection and analysis be used for scalable data fusion?

**Hypothesis:** The RS method can provide a good estimation of aggregate value for reports with inconsistency in any single dimension data as well as in multidimensional data, such as temporal and spatial dimensions in this dissertation.

**Observation:** The simulations of one-dimensional temporal and spatial fusion demonstrated low RD value at high fusion point, small occupancy grid (low number of cells), low target density, and either low or high report coverage. The number of reports has a major effect on the RD, but the value of RD becomes stable after a certain number of reports are considered. In addition, the computation time does not change considerably for different space unit sizes since the number of reports increases at the same rate. I extend the scenario of the search and rescue task of target detection at specific locations and time intervals with dynamic targets to test the two-dimensional fusion. In temporal-spatial fusion, the RS method has better accuracy than the CD method. Furthermore, the factors of time report number, time report duration, and space report number all affect the fusion performance.

To summarize, my proposed approach can provide an estimation of aggregate value for reports with inconsistency in any single dimension data or in multidimensional (temporal and spatial) data. The estimated value generated by the RS method has higher accuracy when there are a large number of reports. In spatial data fusion, simulations reveal that the RD of the CD

method and the RS method are close with the following strategies of space fusion: to perform data fusion in a longer period of sampling time; to have lower space granularity; and to choose either low RC, which provides more location information or to choose high RC, which has more overlapping reports. In temporal-spatial fusion, the RS method has better accuracy with higher number of time reports, duration of time reports, and number of space reports.

The major goals of this dissertation are to provide a systemic approach of inconsistency detection and data fusion in different domains in an efficient way. The importance of inconsistency detection for data fusion is increasing because of amount of data is thriving from distributed heterogeneous databases. There are many areas that require data reliability assessment and data fusion such as multisensory systems, image processing, interactive online systems, and data mining. My methods can be applied in each of those areas. Data centers can take advantage of increasing robustness and reliability of data by using multiple sensors data or multiple data sources. However, reaching consensus between all data reports is a considerable problem. One application of the temporal and spatial fusion is the target observation in sensor networks. The tasks focus on checking the origin of the information from sensor registrations, checking the consistency of sensor data, and tracking target movements. It will be more efficient when the system can provide these benefits automatically rather than requiring a feedback from humans, especially when there is a large number of sensors/robots.

Another application of my method is related to the usage of web data. Using data sources from Internet often applies concept of crowdsourcing or collective intelligence. In order to benefits from this data, companies should have (1) multiple data sets from inter-company or data sources, (2) prediction and optimization models to help them analyze data and make decisions more robust, and (3) organizational transformation that allow them to manipulate and extract

information from these data to be more concise (Barton & Court, 2012). My approach contributes to proper utilization of this data in terms of agility, scalability, and lower cost.

## 6.2 FUTURE WORK

My proposed characteristic linear system approach can be used to detect the inconsistency between reports, reveal the ID of inconsistent reports, and decrease the inconsistency by eliminating inconsistent reports or substituting more accurate estimated report values in order to improve data accuracy and reliability. The estimated report values, which are generated by the RS method, provide users with the more accurate information at each interval. There are several ways to adjust a group of inconsistent reports that may help to improve data reliability. The first method is to eliminate the inconsistent reports entirely, the second method is to adjust reported values to make it consistent, and the third method is to modify report values by the  $\delta$  value and the  $C$  value. The first method is simple and straightforward, but will affect the accuracy of data fusion dramatically if the report has large reported values and a small degree of overlap. The second method uses the generated solution set by the RS approach. This method relies on the generated solution set; the accuracy can be improved if there are many overlapping reports. The third method uses additional information about reports; the nonzero  $C$  value (i.e. number-of-conflict-report) and the nonzero  $\delta$  value (i.e. difference with the original report value) of each report indicate how exactly these reports contradict each other. Thus, I can eliminate or modify reports using their  $C$  value or the  $\delta$  value separately. I performed a prior test of using the  $C$  value and the  $\delta$  value separately for report value modification, and I found that using the descendent ranking of the  $\delta$  value as the order to eliminate reports results in reaching consistency faster (i.e.

it converges faster) compared to using the descendent ranking of  $C$  value. This is because the reports often have the same  $C$  value, which slows down finding a consistent system. From these three methods, the second method adjusts reported values without eliminating any one of them. The solution set generated by nonnegative least squares method provides estimated interval values and corresponding reported values of a consistent system. My simulations show that the estimated values of the RS approach are close to the actual interval values at various configurations of reports and measured events. One or more reported values should be modified to make the linear system consistent if the researchers do not want to eliminate reports with nonzero  $C$  value or nonzero  $\delta$ . The nonnegative least squares method I use in this dissertation will generate an optimal solution set via iterative computation. Through this approach, I can find a solution set that satisfies all equations and adjusts reported values minimally.

Finding methods to optimize the solution set of the underdetermined linear system with the presence of inconsistent reports is an area for further research. The optimal solution would improve inconsistency detection, temporal, and spatial data fusion and estimation accuracy. This may require developing a pre-screening algorithm to group reports with overlap into several smaller linear systems, as well as to apply parallel computing to speed up the computation.

## BIBLIOGRAPHY

- Barton, D., & Court, D. (2012). Making Advanced Analytics Work for You. *Harvard Business Review*.
- Bedworth, M., & O'Brien, J. (2000). The Omnibus model: a new model of data fusion? *IEEE Aerospace and Electronic Systems Magazine*, 15(4), 30–36.
- Bleiholder, J., & Naumann, F. (2008). Data fusion. *ACM Computing Surveys (CSUR)*, 41(1), 1–41. doi:10.1145/1456650.1456651
- Candès, E. J., & Wakin, M. B. (2008). An introduction to compressive sampling. *Signal Processing Magazine, IEEE*, (March 2008), 21–30.
- Cevher, V., & Guerra, R. E. (2008). *Sparse Approximation note*. *Pattern Recognition* (pp. 1–3).
- Chang, S.-K., Costagliola, G., Jungert, E., & Orciuolo, F. (2004). Querying distributed multimedia databases and data sources for sensor data fusion. *Multimedia, IEEE*, 06(5), 687–702.
- Chatterjee, A., & Arie, S. (1991). Data manipulation in heterogeneous databases. *ACM SIGMOD international conference on Management of data*, 20(4), 64–68.
- Chawathe, S., Garcia-Molina, H., Hammer, J., Ireland, K., Papakonstantinou, Y., Ullman, J., & Widom, J. (1994). The TSIMMIS project: Integration of heterogeneous information sources. *Proceedings of the 10th meeting of the Information Processing Society* (pp. 7–18).
- Davis, S., & Dantzig, G. (1992). A Strictly Improving Phase I Algorithm Using Least-Squares Subproblems.
- DeMichiel, L. G. (1989). Resolving database incompatibility: an approach to performing relational operations over mismatched domains. *IEEE Transactions on Knowledge and Data Engineering*, 1(4), 485–493. doi:10.1109/69.43423
- Doan, A., & McCann, R. (2003). Building data integration systems: A mass collaboration approach. *International Workshop on Web and Databases*.

- Elfes, A. (1989). Using occupancy grids for mobile robot perception and navigation.pdf. *IEEE Computer Society*, 22(6), 46–57. doi:10.1109/2.30720
- Elmagarmid, A. K., Ipeirotis, P. G., & Verykios, V. S. (2007). Duplicate Record Detection : A Survey. *Knowledge Creation Diffusion Utilization*, 19(1), 1–16.
- Esteban, J., Starr, A., Willetts, R., Hannah, P., & Bryanston-Cross, P. (2005). A review of data fusion models and architectures: towards engineering guidelines. *Neural Computing &*, 1–27.
- Fagin, R., Kolaitis, P. G., & Popa, L. (2005). Data exchange: getting to the core. *ACM Transactions on Database Systems*, 30(1), 174–210. doi:10.1145/1061318.1061323
- Gonzalez, H., Halevy, A., Jensen, C. S., Langen, A., Madhavan, J., Shapley, R., & Shen, W. (2010). Google Fusion Tables : Data Management , Integration and Collaboration in the Cloud. *Processing of the 1st ACM symposium on Cloud computing(SoCC)* (pp. 175–180). doi:10.1145/1807128.1807158
- Hackett, J. K., & Shah, M. (1990). Multi-sensor fusion: a perspective. *Robotics and Automation, IEEE* (pp. 1324–1330).
- Hall, D. L., & Llinas, J. (1997). An Introduction to Multisensor Data Fusion. *Proceeding of the IEEE*, 85(1).
- Hall, D. L., & McMullen, S. A. H. (2004). *Mathematical Techniques in Multisensor Data Fusion. Library* (p. 466).
- Han, J., Chiang, J. Y., Chee, S., Chen, J., Chen, Q., Cheng, S., Gong, W., et al. (1997). DBMiner: A system for data mining in relational databases and data warehouses. *Proceedings of the conference of the Centre for Advanced Studies on Collaborative research (CASCON)*, 1–12.
- Harris, C. J., Bailey, A., & Dodd, T. J. (1998). Multi-sensor data fusion in defence and aerospace. *The Aeronautical Journal*, 102(1015), 229–244.
- Hernandez, M., & Stolfo, S. J. (1998). Real-world Data is Dirty : Data Cleansing and The Merge / Purge Problem. *Data Mining and Knowledge Discovery*, 2, 9–37.
- Horn, B. K. P. (Massachusetts. I. of T. (n.d.). Solving over- and under-determined sets of equations, 2, 2.
- Jeffrey, & Zwillinger. (1971). Tables of integrals, series and products. (A. Jeffrey, Ed.), 1114–1125.
- Kong, S. (2007). *Linear Programming Algorithms using Least-Squares Method*.

- Konolige, K. (1997). Improved Occupancy Grids for Map Building. *Autonomous Robots*, 4(4), 351–367. doi:10.1023/A:1008806422571
- Kozlov, I., & Petukhov, A. (2010). *Sparse Solutions of Underdetermined Linear Systems* (pp. 1243–1259).
- Lawson, C. L., & Hanson, R. J. (1974). *Solving Least Squares Problems*. Philadelphia, PA: Prentice-Hall series in automatic computation. doi:10.1137/1.9781611971217
- Lecce, V. Di, & Amato, A. (2009). Data fusion for user presence identification. *Computational Intelligence for*, 1–5.
- Luo, R. C., & Kay, M. G. (1989). Multisensor integration and fusion in intelligent systems. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(5), 901–931. doi:10.1109/21.44007
- Mahoui, M., Kulkarni, H., Li, N., Ben-Miled, Z., & Borner, K. (2005). Semantic Correspondence in Federated Life Science Data Integration Systems. *Data Integration in the Life Sciences*, 3615, 137–144. doi:10.1007/11530084\_12
- Marano, S., Matta, V., & Willett, P. (2008). Distributed estimation in large wireless sensor networks via a locally optimum approach. *Signal Processing, IEEE*, 56(2), 748–756.
- Matoušek, J., & Gärtner, B. (2007). *Understanding and using linear programming*. Chemistry & Springer-Verlag.
- Natarajan, B. (1995). Sparse approximate solutions to linear systems. *SIAM J. Comput.*, 24(2), 227–234.
- Naumann, F., Bilke, A., Bleiholder, J., & Weis, M. (2006). Data Fusion in Three Steps : Resolving Inconsistencies at Schema-, Tuple-, and Value-level. *IEEE Data Eng. Bull*, 29(2), 21–31.
- Rahm, E., & Bernstein, P. a. (2001). A survey of approaches to automatic schema matching. *The VLDB Journal*, 10(4), 334–350. doi:10.1007/s007780100057
- Sarma, A. Das, Dong, X., & Halevy, A. (2008). Bootstrapping pay-as-you-go data integration systems. *conference on Management of data* (pp. 861–874). doi:10.1145/1376616.1376702
- Schoess, J., & Castore, G. (1988). A distributed sensor architecture for advanced aerospace systems. *SPIE* (pp. 74–86).
- Taubert, J., Hindle, M., Lysenko, A., Weile, J., Köhler, J., & Rawlings, C. J. (2009). Linking Life Sciences Data Using Graph-Based Mapping. *Data Integration in the Life Sciences, Lecture Notes in Computer Science*, 5647, 16–30. doi:10.1007/978-3-642-02879-3\_3

- Thomopoulos, S. C. A. (1990). Sensor integration and data fusion. *Robotic Systems*, 7(3), 337–372.
- US Navel Observatory, & Almanac, N. (1960). Explanatory Supplement to the American Ephemeris and Nautical Almanac.
- Zadorozhny, V., & Hsu, Y.-F. (2011). Conflict-aware historical data fusion. *Scalable Uncertainty Management(SUM)* (pp. 331–345). doi:10.1007/978-3-642-23963-2\_26
- Zadorozhny, V., & Lewis, M. (2013). Information fusion for USAR operations based on crowdsourcing. *IEEE Information Fusion (FUSION), 2013 16th International Conference* (pp. 1450–1457).
- Zadorozhny, V., Manning, P., Bain, D. J., & Mostern, R. (2013). Collaborative for Historical Information and Analysis: Vision and Work Plan. *Journal of World-Historical Information*, 1(1), 1–14. doi:10.5195/jwhi.2013.2
- Zikopoulos, P., Deroos, D., Parasuraman, K., Deutsch, T., Corrigan, D., & Giles, J. (2012). *Harness the Power of Big Data: The IBM Big Data Platform*.
- Zubko, V., Leptoukh, G. G., & Gopalan, A. (2010). Study of Data-Merging and Interpolation Methods for Use in an Interactive Online Analysis System: MODIS Terra and Aqua Daily Aerosol Case. *IEEE Transactions on Geoscience and Remote Sensing*, 48(12), 4219–4235. doi:10.1109/TGRS.2010.2050893