SEMANTIC-AWARE DATA PROCESSING:
TOWARDS CROSS-MODAL MULTIMEDIA ANALYSIS AND CONTENT-BASED RETRIEVAL IN DISTRIBUTED AND MOBILE ENVIRONMENTS

A Thesis in

Computer Science and Engineering

by

Bo Yang

© 2007 Bo Yang

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

May 2007
The thesis of Bo Yang was reviewed and approved* by the following:

Ali R. Hurson  
Professor of Computer Science and Engineering  
Thesis Adviser  
Chair of Committee

Wang-Chien Lee  
Associate Professor of Computer Science and Engineering

John J. Hannan  
Associate Professor of Computer Science and Engineering

Kultegin Aydin  
Professor of Electrical Engineering

Raj Acharya  
Professor of Computer Science and Engineering  
Head of the Department of Computer Science and Engineering

*Signatures are on file in the Graduate School.
ABSTRACT

In the past decade, distributed multimedia data processing, i.e. the management of multimedia data objects from distributed data sources, has experienced an explosive development because of the technological advances in distributed computing, network infrastructures, and multimedia streaming. With the proliferation of the third generation wireless networks, it is expected that multimedia data transmission and manipulation will achieve even larger growth in the next decade. The improved bandwidth and mobility of computing devices enable the possibility of accessing multimedia data “anytime, and anywhere”, providing the foundation for mobile multimedia data management.

At the same time, due to the characteristics of multimedia data (e.g. large data volume and lack of accurate semantic content representation), the existing multimedia information retrieval systems cannot guarantee accuracy, efficiency, and robustness when performing distributed content-based retrieval. In addition, the mobile environment has put further requirements on the manipulation of multimedia data: 1) The mobile computing devices are often disconnected from the network for prolonged periods of time due to the battery power limitations; 2) The network infrastructure is organized in a peer-to-peer fashion, making the traditional centralized or flooding-based search schemes ineffective; 3) The mobile computing devices in the network have frequent relocations, making it important to consider location information in the query processing; 4) The limitation of system resources (e.g. bandwidth and storage) require efficient approaches for handling voluminous multimedia data; and 5) The information retrieval system needs to integrate the heterogeneous and autonomous mobile data sources to provide a global framework for content-based multimedia data access. Generally, the overall performance of multimedia information retrieval is greatly influenced by the emerging issues in mobile networks.

Despite the fact that a great deal of research has been done on multimedia data access, there has been little work done in integrating content-based multimedia retrieval in the mobile environment, especially in the wireless ad hoc networks. In addition, there is not much research work reported on the semantic analysis and content representation of mobile multimedia data. These research issues, however, are crucial for the successful and efficient multimedia communications in mobile networks. Therefore, it is highly necessary to investigate these challenges and devise a novel methodology for mobile multimedia data management.

This dissertation is intended to present and analyze a new semantic-aware multimedia representation and accessing model in distributed and mobile database environments. Semantic classification and categorization of the multimedia databases are based on the Summary Schemas Model (SSM). The ability of summarizing general
information provides a promising mechanism to represent and access multimedia data entities. In this dissertation, we propose a logic-based model that can be integrated in the SSM and used as the paradigm for multimedia data content representation. This dissertation also investigates the feasibility of the proposed model, compares and contrasts it against several models as advanced in the literature.

To provide an efficient platform for multimedia information retrieval in ad hoc networks, we propose to cluster ad hoc multimedia databases based on their semantic contents, and construct a virtual hierarchical indexing infrastructure overlaid on the mobile databases. This clustering scheme uses a semantic-aware framework as the theoretical foundation for multimedia data organization. Several novel techniques are presented to facilitate the representation and manipulation of multimedia data in ad hoc networks: 1) using concise distribution expressions to represent the semantic similarity of multimedia data, 2) constructing clusters based on the semantic relationships between multimedia entities, 3) reducing the cost of content-based multimedia retrieval through the restriction of semantic distances, and 4) employing a self-adaptive mechanism that dynamically adjusts to the content and topology changes of the ad hoc networks.

As an extension to the aforementioned multimedia content representation model, we also presented a semantic-aware image caching scheme to facilitate content-based multimedia information retrieval in ad hoc networks. The caching scheme can efficiently utilize the cache space and significantly reduce the cost of image retrieval. It is based on several innovative ideas: 1) multi-level partitioning of the semantic space, 2) association and Bayesian probability based content prediction, 3) constraint-based representation method showing the semantic similarity between images, and 4) adaptive QoS-aware cache consistency maintenance.

Overall, the focus of this dissertation is to provide a semantic-aware framework that is capable of representing, organizing, and searching multimedia data objects in the distributed and mobile environments. The proposed framework is scalable, fault-tolerant, and efficient in performing content-based multimedia retrieval as demonstrated in our combination of theoretical analysis and extensive experimental studies.
**TABLE OF CONTENTS**

LIST OF TABLES ..................................................................................................................... vii  
LIST OF FIGURES ................................................................................................................... viii  
LIST OF ABBREVIATIONS ....................................................................................................... ix  
ACKNOWLEDGEMENTS ........................................................................................................... x  
1  INTRODUCTION .................................................................................................................. 1  
   1.1 Multimedia Information Processing .............................................................................. 1  
   1.2 Mobile Multimedia: Promises and Challenges .......................................................... 4  
   1.3 Research Focuses ......................................................................................................... 6  
   1.4 Summary of Contributions ......................................................................................... 10  
   1.5 Dissertation Outline ....................................................................................................... 12  
2  BACKGROUND ................................................................................................................... 14  
   2.1 Distributed Multimedia Systems — General Overview ............................................. 14  
   2.2 Multimedia Content Representation .......................................................................... 17  
      2.2.1 Content representation approaches ................................................................... 18  
      2.2.2 The comparisons of representation approaches .................................................. 20  
      2.2.3 The weaknesses of existing representation approaches ...................................... 21  
   2.3 Cross-Modal Semantics Analysis ............................................................................... 22  
      2.3.1 Conversion from features to semantics ............................................................... 22  
      2.3.2 The impact of semantic analysis on multimedia data access ............................. 27  
   2.4 Distributed Multimedia Data Access in Mobile Environments .................................. 27  
      2.4.1 Mobile data management ...................................................................................... 28  
      2.4.2 Peer-to-peer information retrieval ....................................................................... 31  
      2.4.3 Wireless ad hoc networks .................................................................................... 32  
   2.5 Distributed Data Management ................................................................................... 33  
      2.5.1 Multidatabase systems ......................................................................................... 33  
      2.5.2 Summary schemas model .................................................................................... 35  
      2.5.3 SSM Prototypes .................................................................................................... 38  
      2.5.4 Extending SSM to support multimedia data manipulation .................................. 40  
      2.5.5 Integrating mobile ad hoc networks in SSM ......................................................... 41  
   2.6 Summary of Chapter Contents ..................................................................................... 42  
3  A LOGIC FOR REPRESENTING MULTIMEDIA SEMANTICS ............................................. 43  
   3.1 The Rationale of Logic-Based Representation ............................................................ 43  
      3.1.1 Logic-Based Multimedia Content Representation ............................................. 44  
      3.1.2 Automatic Integration of Semantic Contents ....................................................... 49  
   3.2 The Integration of Logic with the SSM (SumLog) ...................................................... 56  
      3.2.1 Integration of Logic Formula with the SSM .......................................................... 56  
      3.2.2 Query Processing in SumLog ............................................................................. 58  
   3.3 Performance Analysis ................................................................................................. 60  
      3.3.1 Theoretical Analysis ............................................................................................. 60  
      3.3.2 Experimental Evaluation .................................................................................... 63  
      3.3.3 Performance Analysis Conclusions ...................................................................... 75  
   3.4 Summary of Chapter Contents .................................................................................... 77  
4  MULTIMEDIA ACCESS IN AD HOC NETWORK ................................................................. 79  
   4.1 Introduction and Motivation ......................................................................................... 79  
   4.2 SSM-Based Clustering for Mobile Clients and Data Centers ...................................... 81  
      4.2.1 Problem Formulation ............................................................................................ 81  
      4.2.2 Content-Based Clustering ................................................................................... 82  
      4.2.3 Integrating Clusters into the SSM Hierarchy ......................................................... 83  
      4.2.4 Self-Adaptive Adjustment .................................................................................... 90  
   4.3 Performance Analysis of the Proposed Clustering Scheme ..................................... 91  
      4.3.1 Theoretical Analysis ............................................................................................ 91  
      4.3.2 Experimental Evaluation .................................................................................... 92  
   4.4 Summary of Chapter Contents .................................................................................... 101
5 SEMANTIC-AWARE MOBILE CACHING ................................................................................... 102
  5.1 Introduction and Motivation ................................................................................................. 102
  5.2 Design Goal ........................................................................................................................... 103
  5.3 Mathematical Basis of Semantic-Aware Caching ............................................................ 104
    5.3.1 Data Content Classification ........................................................................................... 105
    5.3.2 Profiling Semantic Contents ......................................................................................... 108
    5.3.3 Algorithms for Content Estimation .............................................................................. 112
    5.3.4 Profiling Data Contents of Mobile Nodes ...................................................................... 115
    5.3.5 Summary of the Mathematical Basis .............................................................................. 118
  5.4 Principles of Semantic-Aware Caching .............................................................................. 118
    5.4.1 Caching Rationale .......................................................................................................... 119
    5.4.2 Cache Model .................................................................................................................. 121
    5.4.3 QoS-Aware Organization ............................................................................................... 123
    5.4.4 Cache Management ........................................................................................................ 124
  5.5 Performance Study of Semantic-Aware Caching ............................................................. 126
    5.5.1 Simulation Setup ............................................................................................................. 126
    5.5.2 Simulation Result ............................................................................................................ 127
    5.5.3 Evaluation Conclusions .................................................................................................. 137
  5.6 Summary of Chapter Contents ............................................................................................ 137
  6 CONCLUDING REMARKS AND FUTURE WORK ............................................................. 139
    6.1 Contributions of this Thesis ............................................................................................... 139
    6.2 Future Research .................................................................................................................... 140
  7 BIBLIOGRAPHY ................................................................................................................. 142
LIST OF TABLES

TABLE 1: COMPARISONS OF REPRESENTATION APPROACHES .............................................. 21
TABLE 2: SIMULATION SETTING FOR DISTRIBUTED SEARCH ........................................... 64
TABLE 3: THE UNLIMITED 10-NN SEARCH TIME (50 LOCAL DATABASES) .......................... 66
TABLE 4: SIMULATION SETTING FOR DISTRIBUTED SEARCH ........................................... 83
TABLE 5: NOTATIONS RELATED TO THE T-PARTITION .................................................... 86
TABLE 6: NOTATIONS RELATED TO THE T-PARTITION .................................................... 87
TABLE 7: INPUT PARAMETERS TO THE EXPERIMENTAL SYSTEM ...................................... 94
TABLE 8: NOTATIONS RELATED TO ASSOCIATION RULE GENERATION .............................. 112
TABLE 9: NOTATIONS RELATED TO INTEREST NODE SET GENERATION ............................ 113
TABLE 10: NOTATIONS RELATED BAYESIAN CONTENT ESTIMATION .................................. 115
TABLE 11: THE SIMULATION PARAMETERS ........................................................................ 126
LIST OF FIGURES

FIGURE 1: IMAGE SEGMENTATION AND REGION SELECTION ................................................ 26
FIGURE 2: THE SUMMARY SCHEMAS MODEL ................................................................. 36
FIGURE 3: THE ARCHITECTURE OF INFORMATION BROKER SYSTEM ......................... 39
FIGURE 4: THE ARCHITECTURE OF MAMDAS ............................................................... 40
FIGURE 5: SEARCH SPHERE FOR NEAREST NEIGHBORS .......................................... 46
FIGURE 6: THE DATA CONTENT DESCRIPTION OF IMAGE OBJECTS ............................. 48
FIGURE 7: THE TRANSLATION OF FEATURE VALUES ................................................. 51
FIGURE 8: AN EXAMPLE IMAGE .................................................................................... 51
FIGURE 9: THE BOOLEAN VARIABLE COMBINATIONS IN THE COMBINATORIAL TABLE 52
FIGURE 10: DESCRIBING FIGURE 8'S CONTENT IN THE COMBINATORIAL TABLE ........ 53
FIGURE 11: THE CLUSTERING IN THE COMBINATORIAL TABLE ................................... 54
FIGURE 12: THE COMPARISON ON SEARCH COST ....................................................... 65
FIGURE 13: THE IMPACT OF SEARCH TIME ON ACCURACY (K=10) ......................... 66
FIGURE 14: THE IMPACT OF DATA DISTRIBUTION ON ACCURACY ............................. 68
FIGURE 15: THE IMPACT OF DATA SET SIZE ON DISK ACCESSES .............................. 70
FIGURE 16: THE EXECUTION TIME ON VARIED DATA SIZES ....................................... 70
FIGURE 17: THE COMPARISONS ON INDEX-NODE DISK ACCESSES ........................... 72
FIGURE 18: THE COMPARISONS ON LOCAL-NODE DISK ACCESSES ......................... 73
FIGURE 19: THE NETWORK TRAFFIC IN DIFFERENT DATA DENSITIES ...................... 74
FIGURE 20: THE NETWORK TRAFFIC WITH DIFFERENT QUERY RATES ....................... 75
FIGURE 21: DATA DISTRIBUTION IN AN AD HOC NETWORK .................................... 89
FIGURE 22: THE INDEXING HIERARCHY BASED ON CLUSTERS ................................... 89
FIGURE 23: THE RETRIEVAL RESULTS WITH VARIED HOP COUNT ............................ 95
FIGURE 24: IMPACT OF HOP COUNT (K = 10) ............................................................... 97
FIGURE 25: IMPACT OF DATA DENSITY (QUERY RATE = 10Q/S, K = 10) ................. 97
FIGURE 26: COMPARISON ON SEARCH PATH LENGTH (QUERY RATE = 10Q/S, K = 10) 98
FIGURE 27: COMPARISON ON SEARCH COST (QUERY RATE = 10Q/S, K = 10) .......... 98
FIGURE 28: IMPACT OF MOBILITY (QUERY RATE = 10Q/S, K = 10) ................................ 99
FIGURE 29: SEARCH COST IN LARGE-SCALE NETWORKS ........................................... 100
FIGURE 30: SEARCH COST AND MAINTENANCE COST ............................................. 100
FIGURE 31: MULTI-LEVEL SEMANTIC HIERARCHY ....................................................... 107
FIGURE 32: AN ILLUSTRATIVE EXAMPLE OF CONTENT ESTIMATION .................... 109
FIGURE 33: THE DISTRIBUTION OF DATA OBJECTS IN MOBILE NODES .................. 116
FIGURE 34: AN ILLUSTRATIVE EXAMPLE OF A NODE DESCRIPTION ........................ 118
FIGURE 35: QUERY PROCESSING AND CACHING RATIONALE ..................................... 119
FIGURE 36: QUERY FORWARDING USING GRID LOCATION SERVICE (GLS) ............ 120
FIGURE 37: THE CACHE STRUCTURE ............................................................................. 122
FIGURE 38: THE COMPARISONS OF CONTENT ESTIMATION IMPACTS ..................... 128
FIGURE 39: THE COMPARISONS OF CACHING STRATEGY IMPACTS ........................... 128
FIGURE 40: THE COMPARISONS OF CACHE SIZE IMPACTS .................................... 129
FIGURE 41: THE EFFECT OF DENSITY ON CACHE HIT RATIO .................................... 130
FIGURE 42: THE EFFECT OF MOBILITY ON CACHE HIT RATIO .................................. 131
FIGURE 43: THE EFFECT OF MOBILITY PATTERN ON CACHE HIT RATIO ................. 132
FIGURE 44: THE EFFECT OF DENSITY ON RESPONSE TIME ................................. 133
FIGURE 45: THE EFFECT OF CACHE SIZE ON RESPONSE TIME ............................... 133
FIGURE 46: THE EFFECT OF CONTENT ESTIMATION AND CACHING STRATEGY ........ 134
FIGURE 47: THE AVERAGE TRAFFIC ON EACH NODE ............................................. 135
FIGURE 48: THE EFFECT OF BANDWIDTH ON RESPONSE TIME ............................... 136
FIGURE 49: THE EFFECT OF ACCESS LOCALITY ON RESPONSE TIME .................... 136
LIST OF ABBREVIATIONS

PDA (Palm Digital Assistant).................................................................................................................4
NGMN (Next-Generation-Mobile-Network)...............................................................................................4
SSM (Summary Schemas Model)...............................................................................................................7
QoS (Quality of Service)............................................................................................................................8
QBIC (Query By Image Content)..............................................................................................................17
SVD (Singular Value Decomposition)........................................................................................................24
CCA (Canonical Correlation Analysis).......................................................................................................24
MBR (Minimum Bounding Rectangle)........................................................................................................26
AP (Access Point)......................................................................................................................................27
LAQ (Location Aware Query)......................................................................................................................30
LDQ (Location Dependent Query).............................................................................................................30
P2P (Peer to Peer).....................................................................................................................................31
MDBS (Multi-Database System)................................................................................................................34
FDBS (Federated Database System)...........................................................................................................34
kNN (k-Nearest-Neighbor).......................................................................................................................46
ACKNOWLEDGEMENTS

The Office of Naval Research under contract N00014-02-1-0282 and National Science Foundation under contract IIS-0324835 in part have supported this work.
1 INTRODUCTION

1.1 Multimedia Information Processing

In recent years, the exponential expansion of multimedia applications, partly due to the rapid growth of Web-based data sources, has proliferated over the daily life of computer users. The integration of wireless communication, pervasive computing, and ubiquitous data processing with multimedia database systems has enabled the connection and fusion of distributed multimedia data sources. In addition, the emerging applications such as smart classroom, digital library, habitat/environment surveillance, traffic monitoring, and battlefield sensing have provided increasing motivation for conducting research on multimedia content representation, data delivery and dissemination, data fusion and analysis, and content-based retrieval. Consequently, research on multimedia information processing is of increasing importance in computer society. In contrast with the traditional text-based systems, multimedia applications usually incorporate much more powerful descriptions of human thought — video, audio, and images [6]. Moreover, the large collections of multimedia data make it possible to resolve more complex data operations such as imprecise query processing or content-based information retrieval. However, the emergence of multimedia information processing has also brought challenges to the existing database systems. To understand more clearly, the following issues of existing multimedia applications are considered in the database systems:

- Semantic gap: There is a gap between the user perception of multimedia data contents and the physical presentation-and-access mechanism of multimedia data. Computer users often browse and desire to access multimedia data at the object level (“entities” such as human beings, animals, or buildings). However, the existing multimedia information retrieval systems tend to represent multimedia data based on their lower-level features (“characteristics” such as color patterns and textures), with little regard to combining these features into objects [7]. This representation gap often leads to unexpected information retrieval results. The representation of multimedia data according to human’s perspective is one of the focuses in recent research activities; however, few existing systems provide automated identification or classification of objects from general multimedia data collections [7].

- Heterogeneity: The collections of multimedia data are often diverse and poorly indexed [53]. In a distributed environment, due to the autonomy and
heterogeneity of data sources, multimedia data objects are often represented in heterogeneous formats [50]. The difference in data formats further leads to the difficulty of incorporating multimedia data objects under a unique indexing framework.

- Efficiency: Last but not the least, the present research on content-based multimedia information retrieval is mainly based on feature vectors [9]. These features are extracted from the audio/video streams or image pixels, with the empirical or heuristic selection, and then combined into vectors according to the application criteria. In a specific multimedia database system, the feature vector is often fixed sized to facilitate the computation and representation [6]. However, in many instances some features may be null. Although the null features do not contribute to the semantic contents of multimedia data objects, they still occupy space in the feature vectors — hence, lower storage utilization.

As more and more multimedia data repositories are connected within a network infrastructure, it is increasingly notable that the distributed data sources are merging into an integrated environment — the geographically distributed data objects of various modalities (text, image, video, audio) are integrated together; and at the same time the environment allows a great number of system users to access and manipulate information effectively and conveniently.

The current distributed system paradigms, based on either centralized data centers or separated non-cooperative clients, are not suitable for supporting the requirements of the emerging heterogeneous distributed multimedia information processing applications:

- Load balancing: The centralized paradigm has been used in existing multimedia information systems [5, 6, 8]. One common characteristic of these systems is the reliance on a centralized data server (or data center) that would handle the query messages from clients and return the query resolution results. This paradigm puts heavy workload on the data server, leading to its unsuitability for the drastically increasing volume of data and content-based queries over the widely distributed data sources.

- Search cost: Since all query messages are forwarded to the data center for query resolution, the centralized client-server architecture increases the number of messages in the network and therefore is not an effective solution. Moreover, it also causes single-point-of-failure problem and therefore reduce the scalability.
• Response time: Many multimedia information retrieval systems have special requirements on response time [6]. For the existing centralized paradigm, there is no guarantee for the response time in real-time multimedia applications, since the data organization requires far more computation cost and communication cost than the acceptable expectation.

• Failure resilience: The existing centralized information systems are susceptible to various failures such as network disconnection, incorrect resolution of data sources, and invalid data replications.

Based on the aforementioned observations, it is highly necessary to find a solution for the efficient and effective management of multimedia data in the distributed database environment. Generally, the successful storage and access of multimedia data requires careful analysis of the following issues:

1. Effective representation of multimedia data contents,
2. Appropriate indexing architecture of the distributed multimedia databases, and
3. Proper and efficient technique to browse and retrieve data objects in multimedia database systems.

Among the aforementioned three issues, content representation provides the foundation for indexing, classification, and query processing. The suitable representation of multimedia entities has significant impact on the efficiency of multimedia indexing and retrieval [53]. For instance, object-level representation of multimedia data usually provides a more convenient method in content-based indexing than pixel-level representation [7]. Similarly, queries are resolved within the representation domains of multimedia data, either at object-level or at pixel-level. The nearest-neighbor searching schemes are usually based on the careful analysis of multimedia representation — the knowledge of data contents and organization in multimedia systems. It is one of the objectives of this dissertation to propose and evaluate a multimedia content representation model suitable for heterogeneous distributed data sources.

Indexing infrastructure and search strategy have significant impact on the performance of multimedia information retrieval systems. This dissertation gives a content-aware clustering scheme that facilitates the content-based multimedia information retrieval in heterogeneous, distributed, and mobile environments.

For the simplicity and clarity of presentation, we focus on images in the descriptions of our multimedia information retrieval scheme. The definitions and examples in this
dissertation are also based on the assumption of image data management. However, our scheme can be easily extended to other multimedia modalities (e.g. video, audio, and tactile). From the perception of human beings, content-similar multimedia data objects of different modalities can be considered as similar data entities in the semantic space. Our model is oriented to the semantic-level manipulations of multimedia data objects.

1.2 Mobile Multimedia: Promises and Challenges

Advances and new standards in signal processing, pattern recognition, and computer vision have stimulated the explosive development of multimedia applications. The recent proliferation of portable computers, mobile devices, and personal digital assistants (PDAs) are becoming major sources of the networked environment. The pervasive computing of multimedia data has brought the following flexibilities and conveniences:

- Individualized computing: The mobile and pervasive computing paradigms are more individualized in comparison to the traditional wired networks due to the fact that the data contents and interests of the mobile users can be profiled and used for filtering the returned search results. Therefore, the information service in the mobile nodes is supported in a content-based and semantic-oriented fashion. The profiling of mobile users also provides the possibility of resolving user queries based on the current location, time, situation, connectivity, and user habit, which in combination can provide the more accurate query resolution.

- Accessing data anywhere and anytime: Through the wireless connections among the mobile computers, the users can obtain the accessibility to multimedia data repositories through direct or multi-hop network paths. The existing research and implementation in wireless communications have led to the widely used networking schemes that support the upper-layer multimedia data processing applications. The further research has also resulted in the trend of exploring the so called Next Generation Mobile Network (NGMN) that promises higher capability of supporting mobile multimedia data service.

However, the conveniences of multimedia applications come with challenges to the existing data management schemes:

- Resource limitations: Multimedia applications generally require more resources; however, the storage space and processing power are limited in many practical systems, e.g., mobile computing devices and wireless networks [41, 45, 47-53].
Due to the large data volume and complicated operations of multimedia applications, new methods are needed to facilitate efficient representation, access, and processing of multimedia data while considering the technical constraints.

• Data content distribution: When considering multimedia information retrieval in mobile environments, content distribution issue must be addressed to provide an efficient organization of mobile data. Many early proactive search protocols depend on stationary routing tables to maintain route information between data sources. However, most practical wireless networks have dynamic topology with considerable overhead in maintaining the frequently broken routes. To solve this problem, the reactive protocols employ on-demand discovery method to find the routes. However, the route may break as soon as they are discovered due to the mobility, wasting the bandwidth without getting any data. Moreover, these methods fail to integrate the information of data contents into the process of finding routes, which increases the complexity of content-based retrieval.

• Peer-to-peer connections: The mobile computing devices are often organized into an ad hoc network, in which all devices within the range of each other can discover and communicate in peer-to-peer fashion without involving centralized access points. The ad hoc networks are useful in environments where temporary connectivity and communication are needed, when stationary infrastructures are destroyed or too expensive to be built. The mobile nodes in ad hoc networks are capable of not only storing and processing data, but also performing complex operations through their communications, such as content-based retrieval of multimedia data objects. However, the flexible infrastructure-free characteristic of ad hoc networks also complicates the process of multimedia data access: The network topology is constantly changing due to node mobility. When a content-based query is issued, the data source nodes are unknown at the requesting node. As a result, traditionally data-search algorithms rely on flooding strategy to facilitate data access processing [84]. The flooding approach drastically consumes system resources — storage, bandwidth, and energy. Considering the sheer size of multimedia data, the performance degradation is more drastic. Consequently, ad hoc networks cannot utilize classical content-based retrieval methods that are based on flooding mechanisms.
Motivated by the aforementioned challenges, we analyze the characteristics of mobile multimedia and propose a solution which is both feasible and practical. This solution is based on a semantic-aware and self-stabilizing scheme that uses an overlaid infrastructure to organize multimedia data sources in an ad hoc network. The fundamental idea is to organize multimedia data based on concise and abstract description of their semantic contents, and cluster data sources with similar data contents.

1.3 Research Focuses

The development of a general-purpose multimedia content-processing platform that automatically analyzes and represents semantic contents is complex undertaking; hence any proposed platform is required to satisfy multiple requirements for various application domains: First, the proposed platform should be designed as a cross-modal umbrella that integrates features from various modalities — different data sources and heterogeneous data formats, allowing queries to be submitted in any data format at any local database, without consideration of autonomy and heterogeneity of data sources. Second, the representation of multimedia data objects should be performed automatically and efficiently. Some earlier systems have employed manual work in the multimedia data content representation. However, due to the proliferation of multimedia data repositories, there is a high need of automatic content extraction and representation methods. Third, the new platform should provide a scalable content-based multimedia retrieval scheme that can be easily applied to large collections of multimedia data for mobile environments satisfying the technological constraints. In an attempt to meet these requirements, the research scope in this document has proceeded as a three-stage process:

1) A literature review of some existing content-based multimedia representation / retrieval models has been done as the preparation step to provide the insight into the problem of multimedia content analysis. The knowledge about content processing was used as a guideline in constructing an appropriate scheme that achieves content-based multimedia information retrieval with the consideration of various performance metrics (chapter 2).

2) Based on the knowledge obtained from the literature review, a series of issues were investigated to find the solutions to distributed multimedia information processing (chapter 3 and chapter 4):
   - Content representation: The purpose was to simplify the representation of multimedia data contents from high-dimensional feature space to a low-dimensional
semantic space (i.e. from pixel-level to object-level). The dimension reduction was performed through eliminating the insignificant features that do not affect the semantic contents. The contents of multimedia data objects were further examined through semantic analysis. The merit of this content representation is reducing the computation complexity without loss of accuracy.

A logic-based content representation scheme is proposed as a cross-modal platform that accommodates heterogeneous multimedia data objects. This scheme employs first-order logic terms to describe the semantic contents. Based on carefully designed combinations of logic terms, multimedia data objects from heterogeneous data sources are represented and organized under a uniform logic umbrella regardless of their media types (image, video, audio, text, and etc.) — heterogeneous multimedia objects of different media types are treated as inter-convertible entities.

The most notable merit of the logic-based content representation scheme is the connection of two formerly separated realms in multimedia representation: low-level features and high-level semantics. A careful analysis of the logic-based content representation scheme indicates its conceptual consistency with the Summary Schemas Model (SSM). Consequently, we consider the logic-based content representation scheme as the conceptual extension of the SSM in the domain of multimedia applications. Since the SSM has been theoretically proven and practically verified as a highly efficient prototype for managing autonomous and heterogeneous data sources, we believe that logic-based content representation achieves similar merits in distributed and mobile heterogeneous environments.

- Data integration: The automated content-integration approaches were analyzed to validate the feasibility of the content representation scheme. One major goal of data integration was to optimize representation terms, thereby improving the utilization of database storage.

- Indexing: Content-based indexing needs analysis of data contents followed by keyword-based indexing. Content-based indexing implies the consideration of several research issues: First, the same multimedia data could mean different things in different context. Second, computer users typically have diverse information needs. Therefore, it is evident that features may not be sufficient to completely index a given set of data objects. More complicated indexing
infrastructures, e.g. linguistic-based models, are needed for effective indexing of multimedia data.

- **Query processing:** User queries are processed against the available indices. However, unlike the traditional textual databases, matches in multimedia queries are not exact matches. Given that various data objects can resemble the same input query, a single query might yield many results in response. Therefore, how to restrict the search scope and reduce the search cost has significant impact on the performance of multimedia information system.

  To examine the impact of the logic-based content representation scheme on multimedia retrieval, an enhanced content-based retrieval (ECBR) strategy was designed to search for multimedia data objects based on semantic contents. The semantic distances between multimedia data objects were defined as logic inclusion or intersection of their representation terms. Unlike other content-based retrieval methods proposed in the literature, the ECBR strategy performs nearest-neighbor retrieval considering not only feature similarities, but also semantic relevance.

- **Distributed data management:** Multimedia information systems are usually distributed in the sense that a single multimedia interaction often involves data obtained from distributed information repositories. In addition, issues like storage utilization and data generation may also force multimedia information system designers to place multimedia data in different physical locations. To support the information retrieval required in such distributed and cooperative environments, a distributed multimedia information system must address the general issues in distributed databases, such as distributed query processing and data location mapping. In addition, network issues such as limited bandwidth and network delays become important considerations, since they could have adverse effects on the QoS supported.

- **Peer-to-peer search:** Under the general SSM framework, an extension is to allow the local data source nodes to be a community of mobile computers, which form an ad hoc network using the peer-to-peer wireless connections. As a part of the SSM, the ad hoc network is mainly used for the content-based multimedia information retrieval in a restricted area. The focus is how to provide efficient peer-to-peer search schemes that accurately determine the data content distribution in the mobile nodes, and thereby facilitating the query processing.
Flooding is the most commonly used approach for information retrieval in ad hoc networks, since the requesting node does not have any information of the data contents of other nodes and has to employ the blind search. However, the flooding approach arouses drastic consumption of system resources — storage, bandwidth, and energy. Considering the sheer size of multimedia data, the performance degradation is more drastic. In addition, the flooding strategy may cause duplicated queries and retrieval results, which may further increase the cost of the query processing.

To overcome the shortcomings of blind search, this dissertation investigated two approaches — content-based clustering and cooperative semantic caching. The clustering approach groups the data source nodes according to their data contents and tries to find the shortest path for query forwarding between the requesting node and the data source node. The semantic caching approach analyzes the earlier query results and profiles the content distribution among mobile nodes. Later queries are forwarded to the nodes with most relevant data contents. With the help of the proposed approaches, the multimedia information retrieval in ad hoc networks becomes a clearly aimed searching process that offers reduced network traffic, energy consumption, and response time, regardless of the distribution, heterogeneity, and autonomy of the multimedia data sources.

3) To evaluate the performance of the proposed methods in different applications, a series of implementations have been carried out:

- A simulator has been developed in CSIM to compare and contrast the performance of the proposed logic-based content representation scheme against some other models that have advanced in the literature. The analysis is based on the performance metrics such as scalability, space utilization, accuracy, and search cost in performing nearest-neighbor retrieval.

- A small-scale simulation study on the mobile content description model has been conducted in NS2 environment. Performance analysis is carried out in a series of simulation runs that are designed for comparison on performance metrics (e.g. search cost and response time) with different system configurations such as node density, mobility, and data distribution.

- A prototype has been implemented to evaluate the proposed approaches in
handling real-world multimedia data objects, e.g. the Corel image dataset. The prototype is built based on the conceptual framework of the SSM. It accepts images as combinations of both color/texture features and visual objects. Given an example image as the query, the prototype system is able to return the top 15 most content-similar images as the query result.

1.4 **Summary of Contributions**

This dissertation makes four contributions in the aforementioned research topics: First, we propose an analytical model for automated summarization of data contents using a multi-level linguistic framework. The major contribution of this model is the capability of supporting imprecise queries over heterogeneous and autonomous multimedia data sources (especially image repositories). Second, we develop a predicate formula set that is capable of representing image semantics in the form of first-order logics. This predicate set is designed to provide a solution to the open problem of “semantic gap”— the difference in interpreting image contents from the human’s perception and the computer’s representation. Third, we propose an overlay infrastructure that automatically clusters the mobile nodes with semantically similar contents as a means to reduce the search domain of queries. Experimental results show that this infrastructure dynamically adjusts itself according to the topology or content changes in the network. Fourth, we investigate the shortcomings of traditional caching techniques in handling large-size content-rich multimedia data such as images in mobile environments. Based on the investigation, we propose a semantic-aware and QoS-aware caching scheme that allows mobile nodes to perform similarity retrieval with low average costs.

**Linguistic-Based Data Representation:** We introduce a scheme for semantic-based image content integration in distributed heterogeneous database environments. The extended summary-schemas model (ESSM) is used as the underlying platform. With the ability of summarizing the content information and guiding the data distribution, the ESSM provides a quality-guaranteed and time-efficient accessing strategy. The advantages of representing and indexing images on the platform of ESSM are as follows: 1) The ESSM has linguistic capability of representing semantic content precisely and concisely; 2) Queries are resolved through optimized comparisons against automatically extracted data contents; and 3) Imprecise text descriptions can be included in queries, which further enhance the searching capability of the databases.
**Data Integration and Query Optimization:** In the predicate set proposed in our research, an image data object is considered as a collection of logic terms, whose value represents its semantic content. The analysis of semantic contents is then converted to the evaluation of logic terms and their combinations. This content representation approach has the following properties:

- The logic terms provide a convenient way to describe semantic contents concisely and precisely. Easy and consistent representation of the elementary objects based on their semantic features simplifies the semantic content representation of complex objects using logic computations. As a result, the similarity between objects can be considered as the equivalence of their corresponding logic terms.

- This logic representation of data content is often more concise than feature vector. In addition, the logic representation can be optimized to improve storage utilization by eliminating the null features from logic terms.

- Compared with feature vectors, the logic terms provide an understanding of data contents that is closer to human perception.

- Optimization can be easily performed on logic terms using mathematical analysis. By replacing long terms with mathematically equivalent terms of shorter lengths, the data representation can be automatically and systematically optimized.

- Based on the equivalence of logic terms, the semantically similar data objects can be easily found and grouped into same clusters. This organization facilitates the nearest-neighbor retrieval, and at the same time reduces overlapping and redundancy, resulting in efficient search and storage utilization.

**Overlay Infrastructure for Mobile Ad Hoc Networks:** We propose a decentralized non-flooding retrieval scheme in multi-hop mobile ad hoc networks — Semantic Ad hoc Image Retrieval (SAIR). The proposed scheme makes use of the data content distribution in ad hoc networks to reduce the search cost without incurring high maintenance overhead. We have quantified the efficiency and effectiveness of our scheme with respect to various performance metrics — retrieval accuracy, search cost, and maintenance overhead. Through extensive theoretical and experimental analysis, we found that our search method has the following features:

- SAIR is a decentralized non-flooding search strategy performing content-based image retrieval in ad hoc networks. As shown in our simulation results, it can
achieve the comparable accuracy as centralized search schemes while visiting only a small portion of mobile nodes.

- We employed semantic-based clustering in the organization of image data. The content-related mobile nodes are grouped into clusters, which drastically reduce the search cost of content-based image retrieval.

- Our method has the capability of self-organizing according to the data content distribution and topology changes in an ad hoc network. This feature indicates the scalability and robustness of our method in large-scale networks.

**Semantic-Aware Caching:** We tackle the problem of content-based retrieval by using a semantic-aware caching method. The novelty of our method stems from several factors including: 1) Describing multi-dimensional multimedia data objects using constraint-based representation, 2) Forwarding similarity queries to the mobile nodes containing most closely related data, and 3) Reducing the cost of similarity search by restricting the search scope. Through extensive simulations, we show that relative to several recently proposed caching schemes such as CacheData and CachePath, our method can perform similarity search with less cost. Moreover, it is scalable to large network sizes and voluminous data.

### 1.5 Dissertation Outline

The rest of this dissertation is organized as follows: Chapter 2 provides an overview of research on multimedia content representation and distributed data management. We then present the shortcomings of the existing multimedia information systems and the motivation of our research work.

Chapter 3 presents the rationale of logic-based multimedia content representation approach and its integration with the SSM framework. We describe the conjunctive and disjunctive formula descriptions of data contents and the optimization algorithm. In addition, we analyze the query processing in the SSM using the logic expressions. The performance of logic-based representation is also evaluated using theoretical analysis and experimental study.

Chapter 4 discusses the content-based multimedia retrieval in ad hoc networks. We consider a network built on peer-to-peer connections and develop non-flooding search schemes to facilitate the efficient query processing. We validate our schemes through simulations and then use them to (1) study the effect of various system parameters on the search result, and (2) derive policies for determining optimized search strategy.
Chapter 5 describes the semantic-aware caching that profiles the content distribution of mobile nodes and facilitate content-based image retrieval on the mobile data sources. We propose to estimate the data contents using Bayesian and association based models, and to utilize the estimated content distribution information in the query resolution.

Finally, Chapter 6 summarizes our results and outlines directions of future research.
2 BACKGROUND

This chapter is intended to study and evaluate the existing research developments in three areas that are related to the research reported in this thesis: distributed information systems, multimedia content processing, and content-based information retrieval. We will go over the general methodologies in these areas, and discuss their integration with the semantic-based multimedia data manipulation in distributed and mobile environments.

2.1 Distributed Multimedia Systems — General Overview

In the past decade, distributed multimedia has attracted wide research attention and a considerable number of distributed multimedia information systems have been developed in various application domains [2, 4, 6]. The rapid development of such systems were also accelerated by the proliferation of the Internet — an increasing number of users can access the distributed data repositories (e.g. text, audio, video, and image data) conveniently and efficiently, thereby improving the accessibility of the networked data sources. Such repositories should be integrated to provide distributed users with a seamless environment for the authoring and presentation of multimedia data. To support multimedia applications, a distributed multimedia information system should have the capability of integrating the data contents of multimedia data objects from heterogeneous and autonomous data repositories, representing data contents across distributed locations, and managing multimedia data over potentially heterogeneous platforms.

A. Indexing and query processing

The organization models of data are crucial for the efficient storage and access of multimedia and for improving the overall performance. A multimedia data model captures the individual and interactive properties of the multimedia database contents [76]. The individual properties include the elementary component contents (e.g. image features) and the object characteristics. The interactive properties include the interactions among the data objects, such as video frame sequences.

Multimedia information systems have minimum requirements on performance metrics such as transfer rate in order to have acceptable quality of service (QoS). One of the methods to avoid the storage bottleneck is to decompose multimedia data objects into smaller elementary entities and stored on storage devices of different levels. For instance, a video clip can be considered as a collection of images. The data from different storage units
can then be integrated and synchronized according to the temporal/spatial constraints during the presentation [6]. The multimedia information systems are expected to have mechanisms to guarantee the QoS requirements.

Indexing is an important component for efficient search and retrieval of database contents. In a multimedia information system, the data objects are usually indexed using visual features, annotations, and their conjunctions using conditional statements. Manual keyword annotations on images is still the main stream since automatic generation of descriptive keywords is still beyond the capability of the existing computer vision and pattern recognition technologies [76]. This method is highly subjective and error-prone. Thus, indexing images on visual features such as colors, textures, and shape, is more desirable for supporting broader varieties of user queries. There are several approaches for indexing on visual features, which include K-D-B trees [33], Quad-trees [34], and R-trees [35, 36]. These methods are based on computing numeric values for features mapping into a point in the multi-dimensional space.

The query resolution in a multimedia information system may require approximate or similarity matching based on visual features. If queried by example, a content-based query is formulated by selecting one or more representative data objects that resemble the examples. This may lead to many responses of similar data objects, which may not be identical to the query object. With the support of user interactive functions, users can define queries, evaluate the returned results, and even refine the accurate through giving feedback. Other variations of queries that can be provided by the system include simple visual feature query, feature combination query, and localized feature query [3, 8, 33]. For such variations, the methodology applicable to content-based retrieval can also be easily extended and used in resolving these queries.

Many existing models for multimedia indexing and query processing are dependent on low-level features [6]. The feature-based methodology, although straight-forward in implementation, is far from providing satisfactory multimedia data service in real-world applications. First, multimedia data objects with similar features may not share common semantic contents, causing the so-called semantic gap between computer representation and human perception. Second, due to the reliance on feature representation, the query results might be semantically irrelevant with the query. Finally, the feature-based models were usually proposed for single-media centralized applications and cannot be easily extended to distributed environments.
B. **Heterogeneous multimedia data fusion**

One central problem in working with or creating distributed multimedia information systems is that of identifying and resolving the semantic heterogeneity that exists between the multimedia data sources. Semantic heterogeneity exists whenever databases are constructed independently, overtime, by different computer users. Semantic heterogeneity is represented in structured textual databases by differing attributes and structures used to model the same concepts in different systems. Semantic heterogeneity also exists between collections of unstructured data, such as between different collections of image or graphic data, as well as between audio and video data repositories.

Semantic heterogeneity exists even in systems that only consider a single media type (e.g. image). Two major reasons can cause the heterogeneity: 1) Data objects can be stored in various formats due to different requirements of compression and resolution. For example, there are many image data formats such as BMP, DIB, GIF, JFIF, JPEG, PNG, TIFF, and etc. Images in different formats, although they may show similar pictures, are considered as heterogeneous data objects in the traditional multimedia information systems. 2) For the purpose of efficient data management, some systems employ feature dimension optimization algorithms (e.g. PCA [6]) in the representation of multimedia data. Different algorithms may generate different optimizations to the same original data object.

The heterogeneity between distributed data repositories increases the difficulty of providing a uniform interface for the manipulation of multimedia data. A desirable system should be capable of handling heterogeneous data objects at the semantic level. Therefore, we will study existing multimedia content representation and try to find a semantic-aware method for the description of multimedia data contents.

C. **Autonomous data source integration**

In a distributed environment, the data sources are autonomous and independent in organizing multimedia data. A desirable multimedia information system, however, should make this autonomy transparent to the computer user. Users do not need to know where data sources are physically located and how the multimedia data objects are accessed. To the users, the data sources are considered to be organized under a global framework that can access data objects everywhere. Queries from users are first submitted to the global framework and translated into an access term that can be recognized in a physical data
source. However, existing multimedia information systems are generally based on centralized organizations and cannot support the integration of autonomous data sources.

D. Existing distributed multimedia information systems

Research in distributed multimedia information systems has mainly focused on content-based multimedia retrieval. Some efforts, though far from perfect, have resulted in some experimental multimedia database systems:

The IBM Almaden research center implemented an online Query-by-Image-Content system (QBIC) [3]. The system allows users to query an image data collection using color patterns, textures, and shapes. The underlying index on the image data collection is an R*-tree. Although QBIC is one of the earliest and most widely used multimedia systems, it still has some shortcomings such as lack of semantic content representation and long query response time [83].

The VisualSeek system is an online content-based multimedia retrieval system implemented by Columbia University [6]. The users can submit queries by indicating the color regions, and the system finds the images that contain the most similar arrangements of the color regions. The VisualSeek system employs the spatial relationships of the color regions to improve retrieval accuracy; however, it still cannot provide semantic-based query resolution on multimedia data repositories.

2.2 Multimedia Content Representation

The major challenges of traditional multimedia indexing and query processing are caused by the feature-based representation of multimedia data contents [83]. To analyze the impact of content representation, in this section, we will study the existing representation approaches and evaluate their effectiveness in describing multimedia contents.

Multimedia content representation is the process of mapping low-level perceptual features to high-level content information with high accuracy. It is the fundamental means for supporting effective multimedia information organization and retrieval. Several recent researches have proposed using either a generative statistical model (such as Markov Chain Monte Carlo model [60]) or a discriminative approach (such as Support Vector Machines model [59] and Bayesian Point Machines model [61]) to generate annotations for images. Some other studies employ clustering approach to classify multimedia objects into content-
similar groups [7]. The content-representation approaches are usually based on two assumptions:

- The pre-assumption that the set of content categories is already known. That is to say, the number of possible content categories is fixed, and the content of each category is determined a priori [76].

- A fixed set of domains is assigned for the low-level features.

The main goal of multimedia representation is to obtain a concise description of the contents during the analysis of multimedia objects. Representation approaches as advanced in the literature are classified into four groups: clustering-based approach, representative-region-based approach, decision-tree-based approach, and annotation-based approach.

### 2.2.1 Content representation approaches

#### 2.2.1.1 Clustering-based approach

The clustering-based approach recursively merges content-similar multimedia objects into clusters, with the help of either human intervention or automated classification algorithms, while obtaining the representation of these multimedia objects. There are two types of clustering schemes: supervised and unsupervised clustering [7]. The supervised clustering scheme utilizes the user’s knowledge as input to cluster multimedia objects, so it is not a general purpose clustering approach. As expected, the unsupervised clustering scheme does not need the interaction with user. Hence, it is an ideal way to cluster unknown multimedia data automatically. Because of the advantage of unsupervised clustering scheme, here we only discuss unsupervised clustering scheme.

In clustering-based approach, the content of a multimedia object is indicated as its cluster. The clusters are organized in a hierarchical fashion – a super cluster may be decomposed into several sub clusters and represented as the union of sub clusters. New characteristics are employed in the decomposition process to indicate the differences between sub clusters. Consequently, a sub cluster inherits the characteristics of its super cluster, while maintaining its individual contents.

#### 2.2.1.2 Representative-region-based approach

The representative-region-based approach selects several of the most representative regions from a multimedia object, and constructs a simple description of the object based on the selected regions [8]. The most representative regions are some small areas with the most notable characteristics of the whole object. In case of an image, the
representative regions can be areas that the color changes markedly, or areas that the texture varies greatly, etc.

The representative-region-based approach is performed as a sequence of three steps:

- Region selection: The original multimedia object consists of many small regions. Hence, the selection of representative regions is the process of analyzing the changes in those small regions. The difference with the neighboring regions is quantified as a numerical value to represent a region. Finally, based on such a quantitative value, the regions are ordered, and the most notable regions are selected.

- Function application: The foundation of the function application process is the Expectation Maximization (EM) algorithm. The EM algorithm is used to find the maximum likelihood function estimates when the multimedia object is represented by a small number of selected regions. The EM algorithm is divided into two steps: E-step and M-step. In the E-step, the features for the unselected regions are estimated. In the M-step, the system computes the maximum likelihood function estimates using the features obtained in the E-step. The two steps alternate until the functions are close enough to the original features in the unselected regions.

- Content representation: The content representation is the process that integrates the selected regions into a simple description that represents the content of the multimedia object. It should be noted that the simple description necessarily is not an exhaustive representation of the content. However, as reported in the literature, the overall accuracy of expressing the content of multimedia objects is acceptable [62].

2.2.1.3 Decision-tree-based approach

The decision-tree-based approach is the process of obtaining contents of multimedia objects through decision rules [10]. The decision rules are automatically generated standards that indicate the relationship between multimedia features and content information. In the process of comparing the multimedia objects with decision rules, some tree structures – decision trees – are constructed.

The decision-tree-based approach is mostly applicable in application domains where decision rules can be used as standard facts to classify the multimedia objects [26]. For
example, the satellite cloud images are categorized as rainy and cloudy according to the densities of clouds. In medical fields, the 2-D slices from magnetic resonance imaging (MRI) or computerized tomography (CT) are diagnosed as normal or abnormal according to the colors and shapes of body tissues. In these two examples, different cloud densities and different body tissue shapes are related to the different final conclusions. These relationships are the decision rules. And the final conclusions are the contents of the multimedia objects.

The decision-tree-based approach can improve its accuracy and precision as the number of analyzed multimedia objects increases [27]. Since the decision rules are obtained from statistical analysis of multimedia objects, more sample objects will result in improved accuracy.

2.2.1.4 Annotation-based approach

Annotation is the descriptive text attached to multimedia objects. Traditional multimedia database systems employ manual annotations to facilitate content-based retrieval [9]. Due to the explosive expansion of multimedia applications, it is both time consuming and impractical to obtain accurate manual annotations for every multimedia object. Hence, automated multimedia annotation is becoming a hot topic in recent research literature [6]. However, even though humans can easily recognize the contents of multimedia data through browsing, building an automated system that generates annotations is very challenging. In a heterogeneous distributed environment, the heterogeneity of local databases introduces additional complexity to the goal of obtaining accurate annotations [6].

Semantic analysis can be employed in annotation-based approach to obtain extended content description from multimedia annotations. For instance, an image containing “flowers” and “smiling faces” may be properly annotated as “happiness”. In addition, a more complex concept may be deduced from the combination of several simpler annotations. For example, the combination of “boys”, “playground”, and “soccer ball” may express the concept “soccer game”.

2.2.2 The comparisons of representation approaches

The different rationales of the aforementioned multimedia-representation approaches lead to their strengths and weaknesses in different application domains. In this sub-section, these approaches are compared under the consideration of various performance merits (Table 1).
Table 1: Comparisons of representation approaches.

<table>
<thead>
<tr>
<th></th>
<th>Clustering</th>
<th>Representative Region</th>
<th>Decision Tree</th>
<th>Annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Rationale</strong></td>
<td>Searching pixel-by-pixel, recognizing all details</td>
<td>Selecting representative regions</td>
<td>Treating annotations as multimedia contents</td>
<td>Using annotations as standard facts</td>
</tr>
<tr>
<td><strong>Reliability &amp; Accuracy</strong></td>
<td>Reliable and accurate</td>
<td>Lack of robustness</td>
<td>Depending on the accuracy of annotations</td>
<td>Robust and self-learning</td>
</tr>
<tr>
<td><strong>Time Complexity</strong></td>
<td>Exhaustive, very time consuming</td>
<td>Most time is spent on region selection</td>
<td>Fast text processing</td>
<td>Time is spent on decision rules and feedback</td>
</tr>
<tr>
<td><strong>Space Complexity</strong></td>
<td>Large space requirement</td>
<td>Relatively small space requirement</td>
<td>Very small storage needed</td>
<td>Only need storage for decision rules</td>
</tr>
<tr>
<td><strong>Application domain</strong></td>
<td>Suitable for all application domains</td>
<td>The objects that can be represented by regions</td>
<td>Need annotations as basis</td>
<td>Restricted to certain applications</td>
</tr>
<tr>
<td><strong>Implementation complexity</strong></td>
<td>Easy to classify objects into clusters</td>
<td>Difficult to choose proper regions</td>
<td>Easily obtaining content from annotations</td>
<td>Difficult to obtain proper decision rules</td>
</tr>
</tbody>
</table>

2.2.3 The weaknesses of existing representation approaches

Table 1 summarizes the important characteristics of the major multimedia content representation approaches surveyed in this thesis. It can be noticed that each approach relies on either features or keyword annotations to reveal the data contents. However, this reliance incurs several shortcomings in the content description:

First, the features are primarily used for the description of physical characteristics instead of conceptual contents of multimedia data. Although content analysis techniques (i.e. clustering, representative region selection, and decision rule generation) were used to obtain content information reflected from features, the feature-based content representation approaches suffer from inaccurate description caused by the semantic gap.

Second, the annotations of many existing multimedia information systems are manually added, which are often subjective and error-prone [6]. In addition, the annotations are usually based on a single media type (e.g. image), neglecting the cross-modal semantic
relationships among different media types. Therefore, multimedia systems relying only on annotations cannot guarantee accurate query resolution on multimedia data repositories.

Generally, there is a need for novel content representation models that integrates multimedia characteristics at both feature level and semantic level. The new models should also be easily scalable to distributed environments.

2.3 **Cross-Modal Semantics Analysis**

To bridge or narrow the semantic gap, one has to devise automatic semantic learning methods that map low-level features to high-level semantics [9, 36]. In this thesis we give a brief overview of semantic analysis, discuss its impact on multimedia data manipulation, and use the object-level information obtained from semantic analysis in our proposed logic-based representation model in chapter 3.

2.3.1 **Conversion from features to semantics**

As more and more multimedia repositories are built in different application areas, accurate content accessing strategies are needed for efficient retrieval of multimedia objects. Due to reliance of low-level features, the aforementioned content-based representation and retrieval approaches may not provide satisfactory performance when complex semantic concepts are considered. An ideal multimedia system should have the capability of understanding, not just similarity, of multimedia objects. Moreover, this understanding capability should provide an overview that includes similar semantic concepts in different modalities. For instance, when seeing a talking face we expect to see the words that he/she is expressing, the sound of car engine usually comes with an image of a running car. This cross-modal understanding of multimedia data needs content analysis at a higher semantic-concept level, instead of only involves content representation of low-level features.

Depending on the application domains, the cross-modal semantic-analysis methods can be grouped as two categories [46]:

(i) Knowledge-proactive methods, which utilize content correlation approaches such as Gaussian distribution or linear correlation [65], and

(ii) Knowledge-reactive methods, which require little knowledge before processing and are adaptable to most applications, e.g. neural-network-based models [66].

In practical applications, the knowledge-proactive methods usually require a smaller number of training samples and can achieve better performance [46]. Hence, we focus on the knowledge-proactive methods in this sub section.
2.3.1.1 Semantic analysis strategies

Latent semantic indexing (LSI)

Latent semantic indexing (LSI) was originally proposed as a statistical information retrieval approach to discover underlying semantic relationship between different textual units [46]. LSI mainly focuses on two types of semantic relationships:

- **Synonymy**, which refers to the fact that many words may indicate the same object. For instance, the word “picture” can be referred to either as an image or a photo.

- **Polysemy**, which refers to the fact that most words have more than one meaning in different contexts. For instance, “bus” may refer to a public passenger vehicle when it appears near the word “road”, while it may also mean electronic circuits in a paper of computer architecture.

As mentioned before, the semantic relationships also exist in multimedia applications. The feature-based multimedia systems usually employ large number of low-level features to increase the accuracy of content-based retrieval [56]; however, the large number of features also lead to both high computation complexity and incapability of manipulating semantic concepts. Most feature-vector-based multimedia systems decide the relationship between multimedia objects simply by comparing their common features. Hence, unrelated objects may be retrieved simply because similar feature values occur accidentally in them, and on the other hand related multimedia objects may be missed because no similar feature values occur in the query. The goal of LSI is to overcome these shortcomings by mapping the high-dimensional feature space to a lower dimensional “concept space”, thus reducing the computation complexity and providing the semantic relationships between multimedia objects.

A latent semantic index built on multimedia data usually employ a technique known as Singular Value Decomposition (SVD) to create the concept space. For instance, a multimedia system may try to set up the relationships between talking faces in video frames and their expressed words in the audio segments. A joint feature space with \( n \) video features and \( m \) audio features for \( t \) video frames may be expressed as follows [46]:

\[
X = [V_1, V_2, ..., V_n, A_1, A_2, ..., A_m]^T
\]

where

\[
V_i = (v_i(1), v_i(2), ..., v_i(t))
\]

and

\[
A_i = (a_i(1), a_i(2), ..., a_i(t))
\]
The singular value decomposition can be expressed as follows:
\[ X = K S D^T \]
where \( K \) and \( D \) are orthonormal matrices composing of left and right singular vectors, and \( S \) is a \( r \times r \) diagonal matrix of singular values sorted in descending order in which \( r = \min(n+m, t) \). It can be proved that such decomposition always exists [46].

By selecting the \( k \) largest singular values in the decomposition result, the original \( n+m \) dimensional feature space is mapped to a \( k \)-dimensional concept space. Hence, the decreased dimensionality reduces the computation complexity. Moreover, related multimedia features are clustered together by being assigned to the same concept. However, this performance improvement comes at the expense of the following drawbacks [46]:

- The concept space generated by LSI is not understandable by humans. The concepts and their relationships are all represented as numbers without semantic meaning. Hence, it is difficult to make modification to the LSI concepts.

- The SVD algorithm requires a complexity of \( O((n+m+t)^3 k^2) \). Typically, \( k \) can be a small value in practical applications, but the term \( n+m+t \) needs to be large enough to guarantee accuracy. This makes SVD algorithm unfeasible for large and dynamic multimedia data collections.

- Determination of the optimal number of dimensions in concept space is another difficult problem. The updates of multimedia data collections may request the changes of concepts, since some added multimedia objects may introduce new concepts, while some deleted multimedia objects may result in the obsolescence of old concepts. However, it is quite time consuming to perform a new run of SVD algorithm. Hence, LSI is not suitable for dynamically changing multimedia data collections.

**Canonical correlation analysis (CCA)**

In LSI model, the features from different modalities are treated equally. However, related features from different modalities may be assigned to different concepts. To overcome this weakness, related features from different modalities need to be coupled together. The canonical correlation analysis (CCA) is a method of measuring the linear relationship between two multi-dimensional variables. Since multimedia objects can be mapped to different multi-dimensional feature spaces (such as video and audio features), CCA is also useful for clustering features from different modalities.
Formally, CCA can be defined as the problem of finding two sets of base vectors for two matrices $X$ and $Y$, such that the correlations between the projections of the matrices onto the base vectors are mutually maximized. In another word, the goal is to find orthogonal transformation matrices $A$ and $B$ that can maximize the expression [65]:

$$
\| XA - YB \|_F^2
$$

where $A^T A = I$, and $BB^T = I$. $\| M \|_F$ is the Frobenius norm of matrix $M$ and can be expressed as:

$$
\| M \|_F = \left( \sum_i \sum_j |m_{ij}|^2 \right)^{1/2},
$$

where $m_{ij}$ indicates the element on the $i^{th}$ row and $j^{th}$ column of the matrix.

It has been shown that CCA outperforms LSI in matching video frames with their related audio sounds [46]. However, the linear relationships between multimedia objects are still represented as numbers that cannot be understood by humans. Moreover, neither LSI nor CCA is capable of describing the hypernym / hyponym relationship between multimedia objects or concepts.

### 2.3.1.2 Implementation of semantic analysis on image data

#### Image segmentation

In image retrieval systems, there are two types of features: granule-level features and object-level features. The granule-level features are those characteristics that directly or indirectly are derived from the original format of image storage — i.e., the pixels, such as hue, textures, and saturation. The object-level features, in contrast, are obtained from the recognition of the higher-level understanding of the images — the semantic topics of the image data [12].

The object-level features are obtained through detection and recognition of objects in images [16]. In this work, we analyze the object-level features of an image through a two-phase process:

- The image is first partitioned into several segments, which indicate the most significant visual components, and
- The semantics of the partitioned segments are then obtained through latent semantic indexing (LSI) [14, 15].
The image segmentation method is the binary-partition-tree approach [13]. Similar pixels are merged together as homogeneous regions, and these small regions are recursively merged into segments of the image. To distinguish these segments, we enclose them in minimum bounding rectangles (MBRs). For instance, the woman is Figure 1 (a) is segmented as the foreground segment of the image (Figure 1 (d)), and the remaining parts are considered as the background segment.

Suppose an image $A$ is partitioned as $n$ segments $P_1, P_2, \ldots, P_n$. The minimum-bounding rectangle of segment $P_i$ is defined as $MBR(P_i)$. For any two segments $P_i$ and $P_j$, if $MBR(P_i)$ is enclosed by $MBR(P_j)$, segment $P_i$ is then considered as a part of segment $P_j$. For instance, the MBR of the woman’s hat in Figure 1 (d) is enclosed by the MBR of the woman; hence the hat should be considered as a part of the woman.

**Capturing image segments**

The image segments are represented as low-level features such as color histogram or wavelet coefficients. These low-level features cannot represent the semantic contents, and therefore do not provide an ideal basis for semantic-based image retrieval. Hence, the semantic analysis process is to obtain the semantics of these image segments.

Singular Value Decomposition (SVD) is employed to uncover the hidden semantic relationships (such as synonyms) between data objects. The data set includes two set of entities: the objects whose semantics are already known (training samples), and image segments whose semantics remains unknown. For simplicity, we assume the two sets of entities share the same set of low-level features $f_1, f_2, \ldots, f_m$.

Suppose there are $N$ image segments $P_1, P_2, \ldots, P_N$ and $L$ training samples $T_1, T_2, \ldots, T_L$. For image segment $P_i$, the low-level feature values are defined as $f_1^i, f_2^i, \ldots, f_m^i$. For a
training sample $T_i$, the low-level feature values are defined as $t_1^i, t_2^i, \ldots, t_m^i$. Also suppose the training samples are classified as $K$ categories $C_1, C_2, \ldots, C_K$, each category includes at least one training sample. Thus for a category $C_j$, we define a feature vector

$$F_j = (f_1^j, f_2^j, \ldots, f_m^j, \ldots, f_1^N, f_2^N, \ldots, f_m^N, t_1^j, t_2^j, \ldots, t_m^j, \ldots, t_1^H, t_2^H, \ldots, t_m^H)$$

where $H$ is the number of training samples in category $C_j$.

Based on the aforementioned feature vector, a matrix $M$ can be built as follows:

$$M = (F_1, F_2, \ldots, F_K),$$

where each column $F_j$ in matrix $M$ indicates the feature vector for category $C_j$.

After normalization of matrix $M$, we perform the singular value decomposition on $M$ as follows: $M = KSD'$, where $K$ consists of the feature vectors of $MM'$ column-by-column, $D$ comprises the feature vectors of $M'M$, and $S$ is diagonal matrix. The image segments are classified into proper semantic categories after the singular value decomposition, and they are assigned with proper semantics.

### 2.3.2 The impact of semantic analysis on multimedia data access

Multimedia data contains an enormous amount of semantic information that is hidden from feature-based representation [76]. To facilitate content-based multimedia data manipulation, the multimedia information systems need to provide accessibility based on the semantic contents. Consequently, there is a need for algorithms that automatically convert features to semantics.

The research on semantic analysis, although still in progress, has provided the foundation for multimedia data manipulation at semantic level. Based on the semantic contents, complex data content representation methods can be developed to facilitate efficient indexing and access of multimedia data repositories.

### 2.4 Distributed Multimedia Data Access in Mobile Environments

As the wireless world opens up, more and more multimedia data are generated and stored on mobile computing devices (e.g. laptop, PDA, and cell phone). The indexing and query processing of multimedia data are more complicated than textual data due to the limitations of mobile data management — power, bandwidth, and storage. In addition, data retrieval and integration becomes more complicated due to the dynamic topology of wireless networks.
Considering the new characteristics of wireless networks, multimedia indexing and query processing approaches that were successful in the wired networks are not directly applicable to the mobile environments. This section will study the challenges to mobile multimedia data access and evaluate the existing research results.

### 2.4.1 Mobile data management

The realization of mobile data management is the combination of two existing new technologies: the proliferation of mobile computing devices, and the standards of wireless communications [108]. In the mobile and pervasive computing environment, the mobile clients can submit a large number of queries to the data servers through wireless channels. In the always connected situation, such a system is similar as traditional wired systems. However, mobile clients are often disconnected for power reasons; they also frequently relocate between different accessing points (APs) and connect to different data servers.

Although a wireless network with mobile nodes is essentially a distributed system, there are some characteristics that make the mobile data accessing systems different:

- **Un-balance in communications:** The data servers and mobile clients play different roles in the mobile data management. For instance, the downloading bandwidth (from server to client) is usually much larger than the uploading bandwidth [7]. In addition, the client-client ad hoc communication is not supported in some systems. Therefore, the communication is unbalanced.

- **Un-guaranteed connectivity:** The infrastructure of the wireless network is not as fixed as the wired networks. The mobile nodes may roam throughout the network, migrating from one wireless accessing point to another. Moreover, there exist two types of disconnections in wireless networks: voluntary ones that are caused by the mobile users switching on and off frequently, and involuntary ones that are caused by link failures and out-of-range reasons.

- **Un-reliable power:** The mobile computing devices usually have strict limitations on the power consumption since the batteries need to be recharged or replaced after a period of time (e.g. laptops and mobile sensors).

- **Display limitation:** For the sake of portability, the computing units (e.g. PDAs) usually have relatively small screens to show images or other information.
• Space limitation: The mobile computing devices, due to their compact sizes, usually have limited memory storage. Such limitation gives restrictions to the running of some memory-consuming algorithms such as matrix computation.

Each one of the aforementioned characteristics has an impact on efficient data retrieval in an infrastructure that supports mobility. The communication asymmetry along with the restriction in power, make data broadcasting an attractive solution in some application domains [90]. The roaming of clients through different accessing points motivates a new class of queries that are dependent on the current location of the mobile client, i.e., location dependent and location aware queries. Finally, screen and power limitations of mobile devices have impact on the performance and resolution of the query processing.

A. Broadcasting

As a data delivery method fitting very well mobile environments, broadcasting is the approach of disseminating information from a data source node to a large set of mobile clients, where mobile clients are usually unable to transmit data at a very high speed [21]. This is characterized by an inherent un-balanced communication pattern: The bandwidth in the downstream direction is much greater than in the upstream direction.

There are two types of broadcasting: periodic approach that requires the server to push messages to the mobile clients every period of time and aperiodic approach that relies on the refreshing requests from clients. Periodic push has the advantage of allowing clients to disconnect for a certain period of time and have the latest version of data copy since the messages are broadcasted periodically when the clients get connections [90, 91]. Aperiodic broadcasting, however, is more effective in utilizing available bandwidth [92, 93, 108].

For the purpose of improving broadcasting performance, indexing techniques are used in disseminating data using temporal signature or locations, meaning that the clients can determine the exact time for the publication of relevant data items. At the client side, some energy is consumed since the temporal signatures need to be matched through clock circuitry.

An alternative method to broadcasting is the multicast approach [94]. The server sends data to a group of clients using the same multicast address. Clients join multicast groups and filter the relevant data using hash function techniques.
Generally, the broadcasting method drastically consumes bandwidth and therefore is not efficient for large size data such as images and video. To support multimedia data processing in mobile environment, one needs to devise novel schemes for more efficient utilization of bandwidth.

B. Location dependence

The fact that clients in a mobile environment can change locations suggests the so-called location-aware queries (LAQs) and location-dependent queries (LDQs). The query whose result depends on certain locations is a location-aware query, while the query whose result depends on the mobile client’s current location is a location-dependent query [109]. For instance, “Find the Italian restaurants in New York city” is a LAQ, while “Find the nearest Italian restaurants within 5 miles” is a LDQ. This is especially important to mobile multimedia data management, since finding the data source node within the nearest spatial distance can significantly improve the quality of response time.

One location management method is to integrate Global Position System (GPS) into IP address of computers to enable the creation of location dependent services [95]. Examples of these services include multicasting messages selectively to specific geographical locations, providing services to clients within a certain distance from the server and providing information for mobile clients when the information depends on the user’s location (e.g. maps to drivers).

Mobile agents can also be used for the management of location information [92]. In the agent-based method, personal information (e.g. current location) is managed by user agents, while a partially decentralized location query service is used to facilitate location-based operations. There is a user agent for each user. The agent collects and controls all personal information regarding its user. Applications can only get personal information through a user’s agent. The sources of information collection for an agent include infrared-based active badges, GPS, motion sensors, and cameras.

C. Query processing

The query processing in a traditional distributed system has received a considerable amount of attention and been extensively studied in the literature [96, 97, 98]. As pointed out in [98], the objective in distributed query processing is to reduce the amount data transmissions between servers and clients. However, the traditional distributed query processing methods do not fully explore the characteristics of a mobile environment. Specially, the
energy consumption of mobile computers, one of the most important cost criteria, has much more significant impact on the mobile query processing than that in traditional distributed environments. In addition, the frequent disconnections also add more complexity to the query processing — it is important to give valid query results when mobile clients are disconnected from the network. Techniques such as caching are employed in the query resolution in such cases. There are other issues such as peer-to-peer routing and content distribution that need to be addressed in mobile query processing.

2.4.2 Peer-to-peer information retrieval

In recent years, peer-to-peer (P2P) networks are becoming popular in providing the ability of sharing data sources at a large scale. A P2P network is a collection of cooperative nodes that communicate with each other without the intervention of centralized indexing servers. These nodes are capable of not only storing data, but also performing complex operations through their communications, such as P2P lookup or multimedia data streaming.

Some earlier P2P networks, such as the Napster, are linked to centralized data source nodes (data centers) that host constantly updated directories of data contents. Queries issued from the client nodes are resolved at the data centers and the results are forwarded back to the requesting nodes through unicasting. Such centralized organization does not scale well and has the single points of failure. Moreover, the data center behaves as a hotspot and its data updates could increase the network traffic.

The more recently proposed P2P network frameworks are decentralized and have no data centers. The most commonly used frameworks are unstructured P2P networks, where the nodes form peer-to-peer connections among them and resolve queries through the co-operations with peers. Flooding is the most common approach for information retrieval in such P2P networks, since the requesting node does not have any information of the data contents of other nodes and has to employ the blind search. However, the flooding approach achieves good performance only when dealing with text information due to its drastic consumption of system resources — storage, bandwidth, and energy. Considering the sheer size of the multimedia data, the performance deterioration is more drastic. In addition, the flooding strategy may cause duplicated queries and retrieval results, which may further increase the cost of the query processing.

To overcome the shortcomings of the blind search, the structured P2P networks were proposed in the recent literature as an alternative framework. In such networks, the
data objects are placed not at random nodes but at specified locations that will make subsequent queries easier to satisfy. Moreover, the topology of such networks is strictly controlled and does not change drastically. Such designs improve the efficiency of information retrieval in some cases; nevertheless, at the cost of sacrificing the flexibility and scalability of the P2P networks. In practical applications, the network topology and the data contents of the nodes are constantly changing, which increase the difficulty of efficient data retrieval.

Due to the aforementioned reasons, P2P networks cannot utilize classical content-based retrieval methods that are based on centralized or flooding mechanisms. As an alternative solution, the P2P overlay network is explored by researchers [88]. Compared with the client/server based systems, the P2P paradigm holds many promises and alleviates the aforementioned problems. In a P2P system, nodes typically connect to a small set of random nodes (their neighbors) in order to fulfill a task, such as multimedia data streaming or location table lookup. Consequently, it can be scaled up under the computer user’s will. It alleviates the single-point-of-failure problem since it has no centralized server at all. In addition, P2P network increases system accessibility, by distributing the indexing and query processing tasks to multiple computing nodes.

Generally, the wired/wireless networks have already on the road of evolving from host-centric model to data-centric model, where P2P mechanism fits well. For multimedia data processing, the semantic-based information retrieval in the P2P networks is receiving more and more attentions [42]. In chapter 4, we will discuss P2P semantic-aware multimedia information retrieval in wireless ad hoc networks in details.

2.4.3 Wireless ad hoc networks

In practical applications, the multimedia data sources may sometimes form ad hoc networks for temporary wireless communications without location limitations and a predefined infrastructure. Hence, it is necessary to investigate the access of multimedia data in ad hoc networks.

A wireless ad hoc network is a small-scale local area network, in which the mobile nodes communicate with each in close distance proximity and cooperate in performing some complex data processing tasks such as on-demand routing and content-based retrieval.

Ad hoc networks have several advantages and disadvantages in dealing with multimedia data. In contrast with the lower-bandwidth wide-area wireless networks such as
cellular networks (100Kbps for GRPS and 384Kbps for W-CDMA), ad hoc networks comparatively offer higher bandwidth (11Mbps for IEEE 802.11b and up to 54Mbps for IEEE 802.11a and 802.11g) [70]. In addition, ad hoc networks do not rely on infrastructures in supporting node communications. However, this flexible infrastructure-free characteristic also complicates the process of multimedia data access: The network topology is constantly changing due to node mobility. When a content-based multimedia query is issued, the data source nodes are unknown at the requesting place. As a result, traditionally data-retrieval algorithms rely on flooding strategy to facilitate data access processing [38, 39]. The flooding approach drastically consumes system resources — i.e., bandwidth, and energy. Considering the sheer size of the multimedia data, the performance deterioration is more drastic than doing the same retrieval in the wired networks with infrastructures. Consequently, ad hoc networks cannot utilize classical content-based multimedia retrieval methods that are based on flooding mechanisms.

To overcome the shortcomings of flooding-based multimedia data retrieval in ad hoc networks, a variety of approaches have been presented in the literature. Researchers have proposed methods based on centralized client-server architecture. Some examples of such approaches are presented in [71, 72]. One common characteristic of these models is the reliance on a centralized storage (or head node) that would handle the queries from clients and forward back the results. This assumption violates the requirements of ad hoc networks where all nodes should be considered as equal peers and none of them should be given extra capability or responsibility. Moreover, the centralized models will cause single point of failure and therefore are not robust and scalable.

2.5 Distributed Data Management

In the earlier part of this chapter, within the scope of distributed multimedia information systems, we reviewed the challenges and potential solutions. The existing solutions, although effective in some specific application domains, still have their shortcomings due to the emerging new technology and its constraints. The weaknesses of the existing systems have led to our study of finding novel methods for multimedia data manipulation.

2.5.1 Multidatabase systems

The advent of large-scale data management began with the invention of database systems, which provide crucial services in business applications and academic research. In
many applications, existing geographically distributed, autonomous, and heterogeneous data sources need to be integrated to share information and perform cooperative search. Considering the large amount of time and capital required for redesigning and rebuilding a database system, it is impractical and time-consuming to construct a homogeneous system out of a collection of heterogeneous data sources.

As reported in the literature, the multidatabase (MDBSs) (heterogeneous distributed database system) paradigm was proposed to facilitate global information sharing process among heterogeneous data sources [1, 33]. Compared with the centralized database systems, the MDBSs hold many promises to allow the integration of the heterogeneous distributed data sources at reasonable cost. In a multidatabase system, a local node (i.e. a data source) physically or logically connects to a number of neighboring nodes to fulfill a task, such as file searching or service discovery.

According to the taxonomy introduced by Sheth and Larson [89], multidatabase systems can be classified as federated (FDBSs) and non-federated database systems. Due to the fact that non-federated database systems do not support local autonomy but federated database system do, the former is a more favorable choice in practice. As FDBS consists of component databases that are autonomous and yet sharing information with the federation. To overcome the local schema heterogeneity problem and support global data access, a FDBS normally adopts the layered schema architecture which evolves from heterogeneous local-level data models to a uniform global-level data model. This global-level data model is also called a canonical or common data model (CDM). Two problems are often associated with the layered schema architecture: 1) schema redundancy existing between different layers, and 2) the formidably high maintenance cost involved in updating the global-level schema as the sizes of FDBS increase.

Based on who creates, maintains, and controls the federation, the federated database systems can be loosely or tightly coupled. The local schemas of the preexisting data sources are defined by the users, while it is the FDBS administrator’s task to design a proper global framework to organize the local data sources into a globally accessible information system. The Summary Schemas Model (SSM), as reported in the literature, is a tightly coupled FDBS that can solve the two aforementioned problems associated with the layered schema architecture.
2.5.2 Summary schemas model

In this sub section, we give a general overview of the Summary Schemas Model (SSM), a linguistic-based content integration framework for multidatabase systems, by describing its structure, rationale, advantages, and extensions.

The SSM is used as an infrastructure for global information processing. The choice of SSM in this thesis stems from the following reasons:

- The effectiveness and efficiency of SSM in dealing with heterogeneous and distributed data sources.
- The semantic and ontology based data organization framework of SSM.
- Its capability of handling imprecise similarity-based queries.
- The availability of prototypes developed following the rationale of SSM.

A. The structure of SSM

As noted in section 2.5.1, a multidatabase is a distributed system that acts as a global layer overlaid on top of multiple preexisting, autonomous, and heterogeneous local databases. The summary schemas model was proposed as a solution to large-scale multidatabase systems that provides automatic support for semantic-based organization [1].

The SSM (Figure 2) consists of three major components: a thesaurus, the local nodes, and the summary schemas nodes. The thesaurus defines a set of globally recognizable terms that specifies the categories and relationships of semantic entities [1]. The online thesaurus allows automatic integration of local schemas and resolution of user’s queries based on linguistic knowledge. Two existing online taxonomies have been explored for use with the SSM [19]: The Roget’s Thesaurus, and the Wordnet. The thesaurus is used to derive the summary schemas hierarchy from local database schema access terms. A semantic-distance metric (SDM) was defined to provide quantitative measurement of “semantic similarity” between terms [1].

A local node is a physical database containing real data sources in various formats and modalities. The local node is organized autonomously, on condition that its semantic contents can be evaluated by the global access terms defined in the thesaurus.
A summary schemas node is a logical database that contains a metadata called summary schema, which represents the concise and abstract contents of its children’s schemas. Fewer terms are used to describe the information contents of a summary schema than the union of the terms in the input schemas while capturing the semantic contents of the input terms.

The hierarchical structure of SSM is derived from the linguistic knowledge represented in the SSM. Based on the general rationale of SSM, the semantic relationships among linguistic concepts are represented using synonyms, hypernyms, and hyponyms.

**B. The rationale of SSM**

The power of the SSM comes from its linguistic-based hierarchy that organizes and clusters data objects based on their semantic contents, regardless of their representation heterogeneity. The SSM metadata employs three types of links to indicate the semantic relationships:

- In the SSM, synonyms are semantically similar data objects in different formats at different physical locations. The SSM employs synonym links to connect and group the similar data objects together.

- A hypernym is the generalized description of the common characteristics of a group of data objects. For instance, the hypernym of dogs, monkeys, and horses
is mammal. To find the proper hypernyms of a collection of data objects, the SSM maintains an on-line thesaurus that provides the mapping from multimedia objects to hypernym terms. Based on the hypernyms of data objects, the SSM can generate the higher-level hypernyms that describe the more comprehensive concepts. Recursive application of hypernym relation generates the hierarchical metadata of the SSM. This in turn conceptually gives a concise semantic view of all the globally shared data objects.

- A hyponym is the counter concept of a hypernym in the SSM. It is the specialized description of the precise characteristics of data objects. It inherits the abstract description from its direct hypernym, and possesses its own particular features. The SSM uses hyponyms links to indicate the hyponyms of every hypernym. These links compose the routes from the most abstract descriptions to the specific data objects.

C. The advantages of SSM

The SSM proposes a novel approach to semantic-based schema integration — it possesses several particular advantages in the organization of data objects, which greatly improves the performance:

- SSM preserves the autonomy of local databases. Preserving the local autonomy is one of the important objectives of multidatabase systems, since such a violation forces frequent reorganization of data at the local level. In the SSM, the local DBA retains full control over the local data. The changes of local data will not incur restructuring of global schemas.

- SSM integrates heterogeneous data sources into a unified logical system. Within the scope of multidatabases, data heterogeneity greatly degrades the performance. The SSM organizes data objects regardless of their physical representation, uniformly, according to their semantic contents. In short, in organizing the metadata, the SSM avoids the complex conversion among data formats, and operates the data objects at a semantic level.

- SSM framework is easily scalable. A SSM system does not maintain a centralized global schema. As a result, adding and removing a local database from the SSM hierarchy does not change the logical structure or interfere with the operability of the system.
• SSM automatically organizes the data objects according to their semantic contents. The lower-level summary schema nodes are the abstract descriptions derived from analysis of the data contents representing semantically significant features of the objects.

• SSM has the capability of supporting nearest-neighbor retrieval. In the SSM, the nearest neighbors are considered as synonyms — objects connected through synonym links. As a result, the nearest-neighbor searching is simplified into a process of finding synonyms through links. In other indexing models, the nearest neighbor searching is a time-consuming process and needs to search a considerable section of multidatabases [6].

• In contrast to many multidatabases, SSM supports imprecise query processing. This is due to the fact that the SSM resolves queries at the semantic level.

• SSM metadata is dynamic and self-adjusting. As a result, it is possible to come up with a semi-balanced hierarchical structure in which the summary schema nodes and local nodes are distributed evenly. After its creation, the summary schemas hierarchy can be dynamically modified to ensure the minimum height. Moreover, the insertion and deletion operations do not require periodical halts and reorganization.

• SSM metadata by order of magnitudes is smaller than other global schema based solutions than proposed in the literature [19]. One problem in the content-based indexing models is the overlapping among branches [6]. This overlapping brings redundancy in indexing structures and hence results in performance degradation. In the creation of the summary schemas hierarchy, the SSM tries to partition the data objects into orthogonal semantic categories, and ensures the least overlapping among branches.

**2.5.3 SSM Prototypes**

The SSM was originally prototyped as part of an Information Broker for remote maintenance, diagnosis, and prognosis of electro-mechanical equipments based on the traditional client/server paradigm. The system consists of a Summary Schemas Network (SSN), a thesaurus server, a SSN administration server, a query server, and a retrieval server (Figure 3). Each server has a Graphic User Interface (GUI) to facilitate ease of
communication. Clients are mobile or stationary devices that can be connected to the SSN [44].

![Diagram of information broker system](image)

**Figure 3:** The architecture of information broker system.

The recent advances in wireless communication provide means for mobile users to access information sources anytime and anywhere; however, wireless communication and mobility also bring obstacles to data processing, i.e. restricted bandwidth, limited memory, higher transmission error ratio, and routing overhead. A mobile-agent-based prototype – MAMDAS – was developed to examine the application of SSM in providing global information retrieval in mobile environments [44].

MAMDAS comprises four major logical components: the hosts, the administrators, the thesaurus, and the users (Figure 4). A host can maintain any number and any type of nodes (local nodes and/or summary-schemas nodes) based on its resources. A host may consist of several node-manager agents that monitor and manipulate nodes. The administrators have complete control over the structure and data of the summary-schemas hierarchy. A thesaurus-master agent is defined as the interface between the thesaurus server and the other agents. User queries are resolved through the communication and cooperation
among the mobile agents. The simulation results [44] have indicated that MAMDAS outperforms Information Broker prototype for about 6 times. In addition, the mobile-agent based paradigm offers a robust and scalable computational infrastructure.

![Diagram](image)

**Figure 4:** The architecture of MAMDAS.

### 2.5.4 Extending SSM to support multimedia data manipulation

The SSM was originally proposed to deal with textual data contents. As more and more data processing applications are involved with multimedia, the SSM is also facing the problem of how to manage multimedia data using a linguistic-based methodology. Due to the differences between textual and multimedia data objects, two major issues need to be addressed:

First, the data volumes of multimedia data are normally much larger than those of textual data. Therefore, traditional methods to improve the performance in distributed textual databases (e.g. data replication and broadcasting) are not necessarily applicable for multimedia databases. To solve such a problem, the SSM needs to be extended to make full use of its synonym and hypernym/hyponym links during the query processing.
Second, the contents of multimedia data objects are difficult to be obtained and represented. Consequently, traditional data-value-based content representation, which is sufficient for textual data, does not provide a solid foundation for the manipulation of multimedia data (such as content-based retrieval). New methods are needed to facilitate multimedia data integration and representation.

In contrast to the existing distributed multimedia systems mentioned in section 2.1, the extended SSM has the following advantages:

- First, the extended SSM achieves high retrieval accuracy by exploiting the semantic contents of multimedia data objects. Unlike the existing systems, the SSM does not rely on only the low-level features for content representation, but employs high-level semantic descriptions to facilitate content-based retrieval.

- Second, the extended SSM allows users to submit queries using both keywords and example images. The integration of linguistic knowledge in the organization of multimedia data enables the search engine to resolve both textual and image queries. Therefore, the SSM can support all three modes of multimedia retrieval — query-by-example, query-by-keyword, and query-by-browsing.

- Finally, due to the awareness of the content distribution, the SSM is capable of using optimized time in finding the data source that is most relevant with the query. This merit of the SSM is especially important in mobile environments, where data source locations are usually unknown at the querying place.

### 2.5.5 Integrating mobile ad hoc networks in SSM

Due to the recent advances in wireless communication and mobile data access, the SSM is facing new challenges for providing more effective support to distributed data processing:

First, the SSM was originally based on the client/server paradigm to provide data management in a distributed heterogeneous environment. In addition, the client/server paradigm is implemented in a wired network, thereby connecting each node in the SSM hierarchy via wired links. In practical mobile applications, however, a local node can be a community of mobile computing devices that form an ad hoc network. The query processing in this community is performed in the peer-to-peer fashion. The communications between different computer communities are fulfilled through the summary schemas hierarchy.
Second, the SSM was used to provide the global information sharing among geographically dispersed data sources. Integration of ad hoc networks in the SSM requires analysis of mobile data sources in a proximity space, where connectivity among the nodes is dynamic and faces frequent disconnections. Therefore, the query processing algorithms for the original SSM cannot be directly used in the infrastructure-free mobile ad hoc environments.

2.6 Summary of Chapter Contents

In this chapter, we have given the background knowledge on distributed information systems and content-based multimedia retrieval. For multimedia applications, the distributed information systems should overcome several challenges — i.e., large data volumes and content representation. Therefore, new representation methods are needed for the efficient representation, integration, and manipulation of multimedia data contents.

As a promising solution to distributed data management, we reviewed the Summary Schemas Model (SSM) and evaluated its advantages. Considering the recent proliferation of mobile computing devices and wireless networks, we addressed the mobile data access systems and introduced a mobile agent-based implementation of SSM — MAMDAS.

In addition, the emerging P2P and ad hoc networks have given more flexibility and scalability to distributed information systems. However, search schemes in P2P and ad hoc networks are still far from being efficient due to the dynamic topology and resource limitations.

Finally, the representation of multimedia data contents is the fundamental issue in extending the SSM for distributed multimedia data management. In chapter, we will discuss about a logic-based multimedia content representation approach and its integration into the SSM framework.
3 A LOGIC FOR REPRESENTING MULTIMEDIA SEMANTICS

Devising a uniform paradigm for the representation of multimedia data contents in the presence of heterogeneity, distribution, and semantic gap is a difficult task [76]. When adding technological limitations to this mix, the problem becomes more complex. In the literature survey, we have reviewed an ontology-oriented system — the Summary-Schemas Model (SSM). The SSM is distinguished for its automatic semantic content integration; however, the original SSM was proposed for handling text information. The necessity of extending the SSM for the efficient manipulation of multimedia data motivated our work on finding distributed multimedia content-representation and searching methods.

In this chapter, the foundation of our work, we present a logic-based model for representing the semantics of complex multimedia data objects. The model employs first-order logic to describe the semantic contents of multimedia data, such as visual objects and color/texture features. The aim of this model is to provide general multimedia content representation that can be used in object-oriented information systems. Since the SSM is a framework for the global data management [1], it is highly desirable to integrate the logic-based representation into the SSM model for supporting multimedia data manipulation in large-scale distributed environments. Therefore, we will discuss the integration issue and the evaluation of overall performance of content-based retrieval in this chapter.

3.1 The Rationale of Logic-Based Representation

In this thesis, we address the representation of multimedia data contents at two levels: form and semantics [76]. Just as most of the prevailing multimedia information systems, our content representation model also needs to represent the media-dependent form features and media-independent semantic contents. The representation is through a format that is acceptable to both human and computers — logic.

Many of the multimedia information systems use automated mechanisms of content representation to facilitate retrieval. The analysis of multimedia representation approaches has shown that multimedia information systems may achieve the best performance through understanding the contents of multimedia data objects [54]. Motivated by this observation, we desire to find a paradigm that effectively reveals multimedia semantics.

One of the central issues in multimedia information systems is the content-based retrieval. There are three categories of retrieval formats according to the form and semantic representation of multimedia data contents: form-oriented retrieval, semantic-oriented retrieval, and the combination of both aforementioned categories. Form-oriented retrieval
methods rely on the retrieval process that employs the features of multimedia data objects (e.g., clustering and representative region schemes as noted in chapter 2). This type of retrieval is based on the multimedia features such as audio pitch frequency, image color histogram and wavelet pattern, video frame sequence, and frequent text pattern. Semantic-oriented retrieval approach is based on the symbolic representation of the aboutness of multimedia data objects, such as the scene, visual objects, and annotated information. The commonly used representation methods for such retrieval are annotation-based or decision-tree-based.

The rationale of the logic-based representation, which is proposed in this chapter, relies on the fact that the data organization of most of the existing information systems is based on ontology-based expressions, such as inductive and deductive models [10]. This validates the appropriateness of using logic representation for multimedia data contents. In addition, the components of the proposed logic-based representation approach, namely conjunctive and disjunctive predicates, are the fundamental building blocks for the logic theories and have been thoroughly analyzed [1, 19].

The proposed model supports all three aforementioned information retrieval methods in a formal, flexible, and extensible logic framework. The model allows us to represent both form features and semantic properties of multimedia data with the aim of developing an intelligent multimedia information retrieval system.

### 3.1.1 Logic-Based Multimedia Content Representation

Understanding and semantic representation of multimedia data have been identified as the important steps towards efficient manipulation and retrieval of multimedia data. Earlier research has extensively studied multimedia feature extraction and representation; however, the automatic integration of multimedia data content at a semantic level, remains a difficult task for such information systems.

#### 3.1.1.1 Multimedia data retrieval

In multimedia information systems, there are two types of features: granule-level features and object-level features. The granule-level features are those characteristics that are derived directly or indirectly from the original format of multimedia storage — i.e., the pixels, such as hue, textures, and saturation. These features collectively determine the form of a multimedia data object. The object-level features, in contrast, are obtained from the recognition of the higher-level understanding of the multimedia data — the semantics of the
multimedia data [58]. In this thesis, we analyze the contents of multimedia data objects by considering both granule-level features and object-level semantics.

**Definition 1: Object Content Distance**

Suppose \( I = \{I_j | 1 \leq j \leq n\} \) is the set of multimedia data objects, and \( \Phi = \{\varphi_i | 1 \leq i \leq m\} \) is the ordered set of features (including both granule-level and object-level features). The object content distance on feature \( \varphi_i \) is a function \( g^{\varphi_i}: I \times I \to R \), where \( R \) is the set of real numbers. The function \( g^{\varphi_i} \) compares the similarity distance between two multimedia data objects and satisfies the following characteristics (for any multimedia data objects \( x, y, \) and \( z \) in \( I \)):

1. \( g^{\varphi_i}(x, y) \geq 0 \),
2. \( g^{\varphi_i}(x, y) = 0 \) iff \( x = y \),
3. \( g^{\varphi_i}(x, y) = g^{\varphi_i}(y, x) \),
4. \( g^{\varphi_i}(x, y) + g^{\varphi_i}(y, z) \geq g^{\varphi_i}(x, z) \) (triangle rule).

Given the data objects \( x, y, \) and \( z \), their positions in the feature space form a triangle where the triangle rule holds. The triangle rule is defined based on the fact that the length of any edge in a triangle is smaller than the sum of the other two.

The object content distance metric provides a quantized measure of comparing the similarity between multimedia data objects. Based on this definition, we introduce the nearest-neighbor concept that is widely used in most existing multimedia information retrieval systems.

**Definition 2: The 1-Nearest-Neighbor**

With the aforementioned multimedia data set \( I \) and feature set \( \Phi \), let \( W = \{w_i | 1 \leq i \leq m\} \) be the set of feature weights, and \( X \) be the multimedia data object that is used as the query example. The nearest-neighbor searching process is a function \( Q \):

\[
Q(X, I, \Phi, W) = \{I_j | I_j = \min\{\sum_{k=1}^{m} (g^{\varphi_k}(X, I_j) \cdot w_k)\} \forall 1 \leq j \leq n \land |Q(X, I, \Phi, W)| = 1\}
\]

**Definition 3: The k-Nearest-Neighbor**

With the same multimedia data set and feature set, the \( k \)-nearest-neighbor searching process is a function \( Q^k \):
The 1-nearest-neighbor retrieval returns only one multimedia data object with the smallest semantic distance from the query example. The k-nearest-neighbor retrieval returns the top k similar multimedia data objects, with the decreasing order of their similarities to the query example object.

Based on the aforementioned definitions, the distance between multimedia data objects is quantified as the spatial distance between data points in the feature space. The nearest neighbors should have similar positions as the querying example object. In other words, the nearest neighbors resides within a sphere whose centre is the querying example object (Figure 5). In Figure 5, the distance between any nearest neighbor and the querying example object is less than the radius of the sphere.

Figure 5: Search sphere for nearest neighbors.

3.1.1.2 Logic formula descriptions

The representation and organization of multimedia data objects has great impact on the efficiency of nearest-neighbor searching; hence, much research has focused on proper content representation models [3]. Motivated by this observation, in this section we put forward a logic-based representation approach for multimedia data objects.
A complex multimedia data object, say, an image, can be considered as a collection of elementary entities, such as animals, vehicles, and buildings, where each elementary entity can be described using some logic expressions, which indicate the mapping of this elementary entity on different features in the feature space.

**Definition 4: The Elementary Entity**

The elementary entities are those data entities that semantically represent basic objects (objects that cannot be decomposed further). Formally, the content of an elementary entity \( E \) can be considered as a first-order logic expression.

Let \( E = f_1 \land f_2 \land \ldots \land f_n \), where \( f_i = p_{i1} \lor p_{i2} \lor \ldots \lor p_{im} \) is the disjunctive of some logic expressions \( p_{i1} \ldots p_{im} \), and \( p_{i1} \ldots p_{im} \) correspond to the possible values of the feature set \( F_i \),

\[
E = \bigwedge_{i=1}^{n} (\bigvee_{j=1}^{m} p_{ij}), \quad \text{for every } p_{ij} \in F_i. \tag{3}
\]

Note that in any term \( f_i = p_{i1} \lor p_{i2} \lor \ldots \lor p_{im} \), there is one and only one true expression \( p_{ij} \). For instance, if \( p_{i1}, p_{i2} \ldots p_{im} \) correspond to all possible color patterns, the semantic content of \( f_i \) at any time is a specific color pattern. Since \( f_i \) is disjunctive of \( p_{i1}, p_{i2} \ldots p_{im} \), the false predicates do not affect the final result. The content of an elementary entity is restricted by its conjunctive terms \( f_1, f_2 \ldots f_n \), which are the extracted features in application domains.

**Definition 5: The Multimedia Object**

A multimedia data object is the combination of a series of elementary entities. Given the definition of elementary entities \( E_1, E_2, \ldots, E_k \), the content of a multimedia data object can be defined as:

\[
S = \text{opt} \left( \bigcup_{i=1}^{k} E_i \right), \tag{4}
\]

where \( \text{opt} \) is a function that converts a logic expression into a semantically equivalent shorter form.

The logic-based representation provides a means to automatically define content description of objects. For instance, assume that the image objects are analyzed according to the following aspects: visual objects, colors, and textures (Figure 6). As can be noted
from Figure 6, the visual objects are dogs and cats. The color is grey, white, blue, or brown. The texture is texture1, texture2, or texture3. Consequently, we have:

\[ p_{1i} \in \{\text{object=dog, object=cat}\} \quad 1 \leq i \leq 2 \]
\[ p_{2i} \in \{\text{color=grey, color=white, color=blue, color=brown}\} \quad 1 \leq i \leq 4 \]
\[ p_{3i} \in \{\text{texture=texture1, texture=texture2, texture=texture3}\} \quad 1 \leq i \leq 3 \]

The data content of images can be represented by the union of a set of elementary entities that are combinations of logic expressions. For example, an image with a grey dog and a white dog can be described as the following logic term:

\[ [(\text{object} = \text{dog}) \land (\text{color} = \text{grey})] \cup [(\text{object} = \text{dog}) \land (\text{color} = \text{white})], \]

or in a simpler form:

\[ \{ (\text{object} = \text{dog}) \land [(\text{color} = \text{grey}) \lor (\text{color} = \text{white})] \}. \]

**Figure 6:** The data content description of image objects.

As noted in definition 4, a multimedia data object is considered as a combination of logic terms, whose value represents its data content. The analysis of data contents is then
converted to the evaluation process of logic terms and their combinations. This content representation approach has the following advantages:

(i) The logic terms provide automatic description of data contents. Easy and consistent representation of the elementary entities based on their data features simplifies the semantic content representation of complex multimedia data objects using logic computations. As a result, the similarity between objects can be considered as the equivalence of their corresponding logic terms.

(ii) This logic representation of multimedia data content is often more concise than feature vector. In a specific multimedia database system, the feature vector is often fixed sized to facilitate the computation and representation. However, some features may be null in many cases. Although these null features do not contribute to the data contents of multimedia data objects, they still occupy space in the feature vectors — hence, lower storage utilization. In contrast, the logic representation can improve storage utilization by eliminating the null features from logic terms.

(iii) Compared with feature vectors, the logic terms provide an understanding of multimedia data contents that is closer to human perception.

(iv) Optimization can be easily performed on logic terms using mathematical analysis. By replacing long terms with mathematically equivalent terms of shorter lengths, the multimedia content representation can be automatically and systematically optimized.

(v) Based on the equivalence of logic terms, the semantically similar objects can be easily found and grouped into the same clusters. This organization facilitates the nearest-neighbor retrieval, and at the same time reduces overlapping and redundancy, resulting in efficient search and storage utilization.

3.1.2 Automatic Integration of Semantic Contents

Based on the logic representation model, we provide a framework for expressing the data contents of the multimedia data objects. This framework has the advantage of formal machine-processable representation, and can be optimized to improve storage utilization and processing speed. In this section, we show an optimization algorithm that automatically integrates the content-representation terms of data objects into a shorter logic description.
3.1.2.1 Optimization algorithm

We have two major goals in the optimization process as follows:

- Optimize the content representation of the exported logic terms, and
- Integrate the multimedia data objects based on their contents.

These objectives allow higher QoS and performance, respectively. Inspired by the formation of Karnaugh Maps [37], we designed a combinatorial optimization table to shorten the complex combinations of features into condensed logic terms. The optimization table is a 2-dimensional table with \( k \) variables where \( 2^k \) is the cardinality of the feature sets representing the underlying multimedia data objects. As a result, each table entry represents an elementary entity. The content integration process is performed on this combinatorial optimization table in four steps:

Algorithm 1: Logic term generation

Step 1: The features are first translated into binary codes for convenience of processing — the features are translated into a series of Boolean variables. For instance, refer to the example presented in section 3.1.1.2, the feature “object” has two possible values \{dog, cat\}, so we use Boolean variable \( X \) to describe “cat”, and its complement \( X' \) to represent “dog”. Figure 7 illustrates the assignment to all three features.

<table>
<thead>
<tr>
<th>Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>( X' )</td>
</tr>
<tr>
<td>( X )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Y' ) ( Z' )</td>
</tr>
<tr>
<td>( Y ) ( Z' )</td>
</tr>
<tr>
<td>( Y' ) ( Z )</td>
</tr>
<tr>
<td>( Y ) ( Z )</td>
</tr>
<tr>
<td>Assignment</td>
</tr>
<tr>
<td>------------</td>
</tr>
<tr>
<td>$V'$ $W'$</td>
</tr>
<tr>
<td>$V'$ $W$</td>
</tr>
<tr>
<td>$V$ $W'$</td>
</tr>
<tr>
<td>$V$ $W$</td>
</tr>
</tbody>
</table>

**Figure 7:** The translation of feature values.

Based on the feature assignments and logic combinations, a logic term can be obtained to represent the data objects. For instance, the logic term

$$\{ (\text{cat} \land \text{brown} \land t1) \lor (\text{cat} \land \text{brown} \land t2) \lor (\text{cat} \land \text{white} \land t1) \lor (\text{cat} \land \text{white} \land t2) \lor (\text{cat} \land \text{white} \land t3) \lor (\text{cat} \land \text{blue} \land t1) \lor (\text{cat} \land \text{grey} \land t1) \lor (\text{dog} \land \text{grey} \land t2) \lor (\text{dog} \land \text{white} \land t2) \}$$

represents the data contents of the Figure 8.

**Figure 8:** An example image.

Based on the feature assignments, the logic term can be translated into the following binary code:
\[\{ (X \land YZ' \land V'W') \lor (X \land YZ' \land V'W) \lor (X \land Y'Z' \land V'W) \lor (X \land Y'Z' \land VW) \lor (X \land Y'Z' \land VW') \lor (X \land YZ \land V'W') \lor (X \land YZ \land VW') \lor (X' \land Y'Z \land V'W) \lor (X' \land Y'Z \land VW) \}\]

**Step 2:** A combinatorial table is constructed according to the logic terms and binary codes.

<table>
<thead>
<tr>
<th></th>
<th>XYZ</th>
<th>XY'Z</th>
<th>X'YZ</th>
<th>X'YZ'</th>
<th>XY'Z'</th>
<th>XYZ'</th>
</tr>
</thead>
<tbody>
<tr>
<td>( V ) ( W )</td>
<td>cat &amp; blue &amp; grey &amp; *</td>
<td>cat &amp; blue &amp; grey &amp; *</td>
<td>dog &amp; blue &amp; *</td>
<td>dog &amp; brown &amp; *</td>
<td>dog &amp; white &amp; *</td>
<td>cat &amp; brown &amp; *</td>
</tr>
<tr>
<td>( V ) ( W' )</td>
<td>cat &amp; blue &amp; t_3</td>
<td>cat &amp; grey &amp; t_3</td>
<td>dog &amp; blue &amp; t_3</td>
<td>dog &amp; brown &amp; t_3</td>
<td>dog &amp; white &amp; t_3</td>
<td>cat &amp; brown &amp; t_3</td>
</tr>
<tr>
<td>( V' ) ( W' )</td>
<td>cat &amp; blue &amp; t_1</td>
<td>cat &amp; grey &amp; t_1</td>
<td>dog &amp; blue &amp; t_1</td>
<td>dog &amp; brown &amp; t_1</td>
<td>dog &amp; white &amp; t_1</td>
<td>cat &amp; brown &amp; t_1</td>
</tr>
<tr>
<td>( V' ) ( W' )</td>
<td>cat &amp; blue &amp; t_2</td>
<td>cat &amp; grey &amp; t_2</td>
<td>dog &amp; blue &amp; t_2</td>
<td>dog &amp; brown &amp; t_2</td>
<td>dog &amp; white &amp; t_2</td>
<td>cat &amp; brown &amp; t_2</td>
</tr>
</tbody>
</table>

**Figure 9:** The Boolean variable combinations in the combinatorial table.

Assuming the feature sets \( F_1, F_2, \ldots, F_k \) have \( n_1, n_2, \ldots, n_k \) distinct values, respectively, we need \( \lceil \log_2 (n_1) \rceil + \lceil \log_2 (n_2) \rceil + \ldots + \lceil \log_2 (n_k) \rceil \) Boolean variables to construct a 2-dimensional table of \( 2^{\lceil \log_2 (n_1) \rceil} \times 2^{\lceil \log_2 (n_2) \rceil} \times \ldots \times 2^{\lceil \log_2 (n_k) \rceil} \) cells. Each cell, labeled with a combination of Boolean variables, either in the original form or in the complement form, represents an elementary entity in the information space. For instance, a cell labeled with \( X, Y', Z, V', \) and \( W \) means an elementary entity whose data content is \{\((\text{object} = \text{cat}) \land (\text{color} = \text{grey}) \land (\text{texture} = \text{texture}_2)\)\}. Figure 9 depicts the combinatorial table and corresponding data contents.
As the indication of data contents, the cells are filled with “1”s, “0”s, or “*”s. The “*”s indicate the non-applicable cases. If a multimedia elementary entity exists in the database, the corresponding cell is set to “1”; otherwise, it is set to “0”. Adjacent cells set to “1”s indicates the multimedia elementary entities sharing some common features. Hence, we can cluster the “neighboring” entities with the common features.

By using the aforementioned notations, the image shown in Figure 8 can be represented as the following table setting as in Figure 10.

<table>
<thead>
<tr>
<th></th>
<th>$XYZ$</th>
<th>$XY'Z$</th>
<th>$X'YZ$</th>
<th>$X'YZ'$</th>
<th>$XY'Z'$</th>
<th>$X'YZ'$</th>
<th>$XYZ'$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$VW$</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>$V'W'$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$V'W'$</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$V'W$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**Figure 10:** Describing Figure 8’s content in the combinatorial table.

**Step 3:** After initializing the cells in the combinatorial table to “1”, “0”, and “*”, optimization process will be initiated as the following:

- Initially, mark all cells as “unprocessed”.
- Cluster adjacent cells containing “1”s either in column or row fashion based on the following heuristic rules:
  1. Each cluster contains $2^k$ adjacent “1”s in a rectangular region in the combinatorial optimization table ($k$ is any non-negative integer).
  2. Cluster as many cells as possible.
(3) Avoid large overlapping among clusters except that the overlapping can result in the clustering of whole row/column.

**Step 4:** The clusters are translated back into logic terms. Each cluster represents a group of neighboring multimedia elementary entities that have common feature(s). Different clusters are integrated together by union operations.

Applying the aforementioned initialization and clustering steps, the values in the combinatorial optimization table can be clustered as follows:

![Combinatorial table](image)

**Figure 11:** The clustering in the combinatorial table.

By using the proposed data-content integration algorithm, optimized terms can be obtained to concisely represent the data contents of the underlying multimedia data objects. The data content of Figure 8 can be optimized into:

\[
\{\text{object=cat} \land [\text{texture=t_1} \lor \text{color=white} \lor (\text{color=brown} \land \text{texture=t_2})]\} \lor \{\text{object=dog} \land (\text{color=grey} \lor \text{color=white}) \land \text{texture=t_2}\}.
\]
3.1.2.2 Algorithm analysis

In section 3.1.2.1, the data contents of multimedia data objects were mapped to a multi-dimensional space of features, then expressed as the disjunctive of first-order logic terms, and finally converted to a content representation with the help of a combinatory optimization table.

The size of terms is measured by the number of logic expressions. Reducing the number of expressions can reduce the number of comparisons in multimedia data object matching and consequently, the communication cost and computation cost during the query processing.

We assume a multimedia data object (say, an image) having \( k \) elementary entities \( E_1, E_2, \ldots, E_k \). Each elementary entity is within the multidimensional feature space indicated by \( f_1, f_2, \ldots, f_n \), where \( f_i = p_{i1} \lor p_{i2} \lor \ldots \lor p_{i m} \) is the disjunctive of some logic expressions. As mentioned in section 3.1.1.2, the data content of the multimedia data object can be represented as the union of the elementary entities, which are expressed as the conjunctives of logic terms. Refer to Definitions 4 and 5, we have the following expression of data content:

\[
S = \bigcup_{i=1}^{k} E_i = \bigcup_{i=1}^{k} \left\{ \bigwedge_{j=1}^{n} \left( \bigvee_{h=1}^{m} p_{j h} \right) \right\}, \quad \text{for every } p_{j h} \in f_j. \tag{5}
\]

Suppose the data content of feature \( f_i \) is uniquely determined by the true term \( p_{ix} \) within \( p_{i1}, p_{i2}, \ldots, p_{im} \), we rewrite equation (5) into a simpler form:

\[
S = \bigcup_{i=1}^{k} \left( \bigwedge_{j=1}^{n} p_{j i}^{(i)} \right), \tag{6}
\]

where \( p_{j i}^{(i)} \) is the true term of the \( j^{th} \) feature of the \( i^{th} \) elementary entity (since only the true term determines the final value of a disjunctive expression).

Let \( S^* \) be the final result from the combinatory optimization table. Given the definition of combinatory optimization table, \( S^* \) by default expresses the same data content as \( S \). According to step 4, \( S^* \) is the union of a collection of clusters \( C_1, C_2, \ldots, C_q \), with each cluster indicating several elementary entities. Hence, \( S^* \) can be expressed as the following:
As mentioned in the algorithm, each cluster corresponds to a rectangular region in the combinatory table. Assume cluster \( C_i \) is horizontally indicated by labels \( L_1', L_2', \ldots, L_r' \), and vertically indicated by labels \( L_1'', L_2'', \ldots, L_s'' \). Here any label in \( L_1', L_2', \ldots, L_r' \) or \( L_1'', L_2'', \ldots, L_s'' \) can be the conjunctive of several predicates in equation (6). For instance, \( L_1' \) may be \((\text{object} = \text{cat}) \land (\text{color} = \text{grey})\). Then \( C_i \) can be expressed as \((L_1' \lor L_2' \lor \ldots \lor L_r') \land (L_1'' \lor L_2'' \lor \ldots \lor L_s'')\), or \( \bigvee_{i=1}^{r} \bigvee_{j=1}^{s} (L_i' \land L_j'') \).

### 3.2 The Integration of Logic with the SSM (SumLog)

#### 3.2.1 Integration of Logic Formula with the SSM

An interesting issue is that the logic representation of multimedia data contents can be seamlessly integrated with the SSM. Each concept in the logic representation can find its counterpart in the SSM, and hence can be represented within the domain of the SSM. For instance, the equivalence between logic terms can be considered as the synonym relationship between summary schemas. Hence, the operation of finding equivalent terms in logic domain can be mapped to searching for synonyms in summary schemas domain. Similarly, other relationships between logic terms can be conveniently represented in SSM. If term \( A \) is equal to a part of term \( B \), then this “inclusion” relationship between \( A \) and \( B \) can be described with hypernym and hyponym relationships in SSM. Considering the strengths of SSM in organizing data [1, 19], we incorporate the logic representation within the framework of the SSM, and call it Summary Logic (SumLog) in the following discussions.

#### 3.2.1.1 Integration

The SSM provides a framework for the organization of data objects based on contents. The description tool (i.e. the summary-schemas hierarchy) that composes the semantic domain is the set of semantic entities and their relationships. Similarly, in our logic-based representation model, the elementary entities are formed into a set \( \{E_1, E_2, \ldots, E_k\} \), and the data contents are defined as conjunctive and disjunctive expressions. As a result, we can consider logic terms and summary schemas as the same concept in the remaining part of this thesis.

Let \( O \) denote a complete ontology-based set of objects that exist in the real world. Let \( A \) be the set of terms used in the SSM. Since the SSM is the ontology-oriented model
for describing the real world entities, there should be a one-to-one mapping between the sets $O$ and $A$. In addition, let $F = \bigcup_{i=1}^{n} F_i$ denote the set of features in the logic-based representation model (here $F_i$ denote the $i^{th}$ feature for an elementary entity). For any image taken from a real world object, there exist a collection of features in set $F$ that corresponds to the object. Therefore, $O$ and $F$ can also be mapped to each other using a one-to-one mapping. Considering the transitive property of one-to-one mappings, there exist a one-to-one relationship between $A$ and $F$. In other words, the SSM and the logic-based representation model have the same description set for real world entities.

**Hypernym**

For two semantic terms $A_i$ and $A_j$ in set $A$, if $A_i$ describes a more generic concept that includes $A_j$, then $A_i$ is a hypernym of $A_j$, denoted as $A_i \succ A_j$. Suppose $T_u$ and $T_v$ are the corresponding logic representations of $A_i$ and $A_j$ in set $F$, respectively. The relationship between $T_u$ and $T_v$ in the logic domain can be defined as $T_u \succ T_v$. Furthermore, due to the one-to-one mapping between the SSM term set $A$ and the feature set $F$, $T_u$ describes a generic object-level concept in logic terms that include the concept described in $T_v$. For example, $T_v$ may only represent white cats, while $T_u$ may represent cats of all color patterns.

Generally, the hypernym relationship in the SSM domain can be represented in the logic domain by either using more generic object-level features or removing some specific granule-level features.

**Hyponym**

For two semantic terms $A_i$ and $A_j$ in set $A$, if $A_i$ describes a specific concept that is included in $A_j$, then $A_i$ is a hyponym of $A_j$, denoted as $A_i \prec A_j$. As the counter-concept of hypernym, a hyponym relationship can be also represented in the logic domain.

**Synonym**

For two semantic terms $A_i$ and $A_j$ in set $A$, it is defined that $A_i$ and $A_j$ are synonyms iff $A_i \succ A_j$ and $A_i \prec A_j$. Therefore, using the similar methods as describing hypernym/hyponym relationships, we can also represent synonym relationship in the logic domain.
3.2.2 Query Processing in SumLog

The nearest-neighbor retrieval returns a list of multimedia data objects that have minimal semantic distances to the query example object. The returned multimedia data objects are ordered according to semantic similarities. The query processor first finds the semantically most similar object to the query example, and searches for its synonyms (i.e. data objects with similar object-level features) in the summary-schemas hierarchy. If the number of synonyms is less than the requested number of nearest neighbors, then the query processor, repeatedly, increases the semantic distance threshold to find and collect more data objects. Algorithm 2 shows the process for nearest-neighbor retrieval.

Algorithm 2: Nearest neighbor retrieval

System Initialization:

1) Define $H$ as the SSM hierarchy that is built upon the distributed local data sources.
2) Each local database node in $H$ has a local schema based on logic representation that specifies local image data set. The local database nodes may have synonym links connecting content-similar data objects.
3) Each Summary Scheme node in $H$ maintains a summary schema that describes the contents of its children nodes. There are hypernym/hyponym links connecting the parent/children nodes.
4) Select a semantic distance metric ($SDM$) calculation method for multimedia data objects [116].
5) Let $X$ be the nearest-neighbor query example and $k$ be the total number of required query results. $X$ is described as a globally recognizable logic-based representation.
6) The query $X$ can be submitted at any node $N$ in the SSM hierarchy $H$.
7) Let $L$ be the search list of nodes in $H$ that perform nearest-neighbor search in parallel. $L$ is initialized as $\{N\}$.
8) Let $E$ be the list of nodes in $H$ that the nearest-neighbor search does not need to visit (i.e. the search can be done in all nodes except $E$). $E$ is initialized as $\emptyset$.
9) Let $R = \text{Search}(H,L,E,X,k)$ be the list of nearest-neighbor query results.
Query Processing Function \textit{Search}(H,L,E,X,k):

1. If \( L = \emptyset \), then return “Search Complete, Results in \( R \)”
2. If \( N \) is a local database node, then
   2.1 Perform local nearest-neighbor search based on \( SDM \)
   2.2 If the result of nearest-neighbor search has \( m \) \((m < k)\) data objects, then
      2.2.1 Append the found \( m \) query results to list \( R \)
      2.2.2 Find the parent node of \( N \) in SSM hierarchy \( H \), denoted as \( P(N) \)
      2.2.3 \( L = L - \{N\} \cup \{P(N)\} \)
      2.2.4 \( E = E \cup \{N\} \)
      2.2.5 Send a message \textit{Search}(H,L,E,X,k–m) to \( P(N) \)
      2.2.6 exit function
   2.3 \( L = L - \{N\} \)
   2.4 Append the found query results to list \( R \) and exit function
3. Compare query \( X \) with the summary schema at \( N \), denoted as \( S(N) \)
4. If query \( X \) is not matched at \( N \), denoted as \( X \cap S(N) = \emptyset \), then
   4.1 \( L = L - \{N\} \cup \{P(N)\} \)
   4.2 Send a message \textit{Search}(H,L,E,X,k) to \( P(N) \)
   4.3 exit function
5. If query \( X \) is matched at \( N \), denoted as \( X \subseteq S(N) \), then
   5.1 Calculate content summaries of children nodes, let the \( C(N) \) be the complete list of children nodes and \( M(N) \) be the children nodes whose contents match with query \( X \), we have \( M(N) \subseteq C(N) \) and \( X \cap S( \bigcup_{N \in M(N)} N_i) = X \cap S( \bigcup_{N \in C(N)} N_i) \)
   5.2 \( L = L - \{N\} \cup M(N) \)
   5.3 Calculate the content distribution of children nodes in \( M(N) \), and obtain the probability density function of summary schema node \( N \), denoted as \( PDF(N) = \sum_{N_i \in M(N)} PDF(N_i) \)
   5.4 For each child node in \( M(N) \), denoted as \( N_i \), send a message \textit{Search}(H,L,E,X, \left[ PDF(N_i)/ PDF(N) \right]) \)
5.5 exit function

The Algorithms 2 can make use of the SSM hierarchy to perform parallel searches in different nodes when there are multiple qualified nodes for the query. The nearest-neighbor retrieval maintains a query result set $R$ and a list of concurrently searching nodes search list $L$. When there are more than one qualified children nodes for the query, each child node will be put in the search list, and be forwarded a copy of the query. The data objects from different children nodes will then combined together as the top $k$ returned results in the set $R$.

3.3 Performance Analysis

3.3.1 Theoretical Analysis

Some content-based retrieval models evaluated search cost in terms of the number of comparisons, while others use the number of disk accesses as the metric [14]. We believe that both parameters should be accounted. In this section, the search cost of the summary-schemas hierarchy is calculated as the average number of accesses at the summary-schemas nodes (number of comparisons) and local nodes (number of disk accesses). We assume a set of $n$ multimedia data objects, $I_1, I_2, ..., I_n$ and the following notations in our analysis:

- $P(I_i)$: The probability of being queried for multimedia data objects $I_i$.
- $H(I_i)$: The minimum depth of any global indexing model.
- $W(I_i)$: The minimum search cost of $I_i$ in the local database in any indexing model.
- $N(I_i)$: The routing cost of $I_i$ in the global indexing infrastructure (i.e. the number of indexing nodes on the path from root to local database).
- $H^*(I_i)$: The depth of the SSM hierarchy.
- $W^*(I_i)$: The search cost for data object $I_i$ in the local database of the SSM.
- $N^*(I_i)$: The routing cost of $I_i$ in the SSM hierarchy (i.e. the number of summary schemas nodes on the path from requesting node to data source node).

Considering the definitions of the indexing models [11-18], the content-based retrieval can be performed as follows:

- Start from the root node of the indexing infrastructure.
- Traverse within the indexing infrastructure. Since most existing indexing models (R-tree family, VP-tree, SS-tree, SR-tree, and etc.) contain overlapping in their indexing infrastructures, the content-based traversal may consist of several upward and downward paths in the infrastructures — the routing cost may be much larger than the depth of indexing infrastructures.

- Find the data source nodes and perform disk access in the local database.

Given the above notations, the search cost for a query $I_i$ on the data set of $n$ random data objects \{I_1, I_2, ..., I_n\} is:

$$W = \sum_{i=1}^{n} [P(I_i) (W(I_i) + N(I_i))].$$

(8)

**Lemma 1:** The SSM hierarchy does not contain any overlapping between its indexing branches.

The elimination of overlapping between branches of the SSM hierarchy is due to the existence of synonym links. While the other indexing models (R-tree family, VP-tree, SS-tree, SR-tree etc.) are striving for the reduction of overlapping, the SSM hierarchy can completely remove the overlapping data by adding some synonym links to other branches.

**Proposition 1:** Given a fixed set of multimedia data objects, the SSM hierarchy has less than half depth of any indexing infrastructure.

**Proof:** We will show that any indexing infrastructure can be described using the SSM hierarchy with less than half depth. Given any arbitrary set of multimedia data objects $I = \{I_1, I_2, ..., I_n\}$ and any indexing model $M$, we can construct an equivalent SSM hierarchy in the following way:

Let $T$ be the indexing infrastructure generated based on model $M$ to organize the multimedia data set $I$. For any node $n_i$ in $T$, let $\text{content}(n_i)$ be the content description at node $n_i$, $\text{parent}(n_i)$ denote the parent node of $n_i$, and $\text{children}(n_i)$ be the set of child (children) node(s) of $n_i$.

The depth of $T$ can be reduced by replacing the two-level indexing relationship between $\text{parent}(n_i)$ and each node in $\text{children}(n_i)$ in $T$ into a one-level hypernym/hyponym relationship in the SSM: 1) Consider $\text{parent}(n_i)$ as a summary schemas node in the SSM hierarchy. The data content of $\text{parent}(n_i)$ can be obtained by the union (or integration) of
data contents of $n_i$ and its sibling nodes, i.e. $\bigcup_{n_i \in \text{children}(\text{parent}(n_i))} \text{content}(n_k)$. 2) Consider the children of $n_i$ and its sibling nodes as the direct children of $\text{parent}(n_i)$ in the SSM hierarchy. The content description of these children nodes remains the same as those in $T$. Merge the content-similar nodes based on the thesaurus. The merging operation reduces the number of nodes, however, the data contents of the merged parent/children nodes still follow the hypernym/hyponym relationships of the SSM. 3) The depth reduction of $T$ is performed by utilizing the semantic relationships between hypernyms and hyponyms. For example, $n_u$, $n_v$, and $n_w$ are nodes in the indexing tree $T$. Let $n_u$ be $\text{parent}(n_v)$, and $n_v$ be $\text{parent}(n_w)$. Since the data contents of nodes can be represented using the logic terms described in section 3.1.1.2, we can define the logic-based content descriptions of $n_u$, $n_v$, and $n_w$ as $S(n_u)$, $S(n_v)$, and $S(n_w)$ satisfying $S(n_u) \supset S(n_v) \supset S(n_w)$. Therefore, $S(n_u)$ can be treated as a hypernym of $S(n_w)$ and can be used for indexing from node $n_u$ to node $n_w$.

In the aforementioned process, the indexing infrastructure $T$ is represented using a SSM hierarchy with half depth. This depth reduction process can be used for obtaining a SSM hierarchy of smaller depth when larger-granularity hypernym/hyponym relationships are employed.

Generally, the Summary Logic can describe any content-based indexing infrastructure with less than half depth. In other words, for any multimedia data object $I_i$, we have $H^*(I_i) \leq H(I_i)/2$. ■

As mentioned in Algorithm 2, the query can be submitted at any arbitrary node in the SSM hierarchy. The routing process for data source node with the SSM hierarchy takes at most twice the depth of the hierarchy — upward from a local node to the root, and then downward from the root to another local node. Meanwhile, the local database can be organized using any efficient index, thereby finding the nearest neighbors with optimized search cost. Therefore,

$$\bar{W}^* = \sum_{i=1}^{n} \left[ P(I_i) \left( W^*(I_i) + N^*(I_i) \right) \right] \leq \sum_{i=1}^{n} \left[ P(I_i) \left( W^*(I_i) + 2H^*(I_i) \right) \right]$$

Considering $H^*(I_i) \leq H(I_i)/2$, we get
\[
\overline{W}^* \leq \sum_{i=1}^{n} [P(I_i) (W(I_i) + H(I_i))] \leq \sum_{i=1}^{n} [P(I_i) (W(I_i) + N(I_i))] = \overline{W}
\]

Hence, the Summary Logic achieves the optimal performance in terms of search cost.

### 3.3.2 Experimental Evaluation

A simulator was developed to compare and contrast the performance of the proposed scheme against some content-based indexing schemes that have advanced in the literature [11-16]. The analysis is based on the performance metrics such as search cost, retrieval accuracy, disk access frequency, and network traffic in the nearest-neighbor search.

For the fairness of comparisons, the simulator was developed using the same metrics as [11-16] (see Table 2). Since the traditional content-based indexing models were originally introduced as centralized solutions for multimedia information retrieval, while the SSM was proposed as a distributed paradigm, it is necessary to find the distributed implementation for the traditional schemes. The simulation follows the same method of implementing centralized indices in the distributed systems as proposed in [107]. We simulated the traditional indexing models as follows (here we use the R*-tree as an example): Suppose there exist \( m \) local databases. We implemented the distributed R*-tree using two levels — local level and global level. At the local level, we built an R*-tree for each local database. At the global level, we used the root node of each local R*-tree as the leaf node for a global R*-tree.

The simulator is able to generate different SumLog configurations based on parameters such as the number of levels, the number of local databases, and the range of fan-out nodes for each summary-schemas node. In addition, the simulator is capable of generating statistical multimedia databases based on parameters such as the size of each local database, the complexity of each data object, and the domain of each feature. Ability to generate statistical databases and different system configurations allow us to validate and compare SumLog model against other schemes under various considerations.

To make our database configuration comparable with the synthetic clustered datasets reported in [12], we used the following synthetic dataset obtained from variations
of 1000 basic images taken from standard Corel image dataset [84]. The images are partitioned into 64 semantic categories, each showing a class of objects (e.g. animals or plants). To map the semantic categories at object-level features, we set 64 object-level features for each image. The feature value is set either as 1 or 0 according to whether it belongs to this category. At the same time, 64 wavelet texture features and 128 color histogram features were extracted for each image. Consequently, each image data object has 64+64+128=256 features, which indicate both object-level and granule-level characteristics. For each basic image, we generated 29 “derived” image data objects whose features were obtained from the combinations of the basic image’s feature values and random numbers. The random number generator follows the normal distribution.

The simulation results reported in this thesis is based on the following SSM system configuration: the hierarchy is restricted to 10 levels; each summary-schemas node has the minimum fan-out of 2 and the maximum fan-out of 10. Finally, the experimental results shown in the following figures are average values taken from 1000 queries. Table 2 shows the simulation parameter setup.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation time</td>
<td>unlimited</td>
<td>1000 seconds to unlimited</td>
</tr>
<tr>
<td>Image dataset</td>
<td>1000</td>
<td>1000 - 30000</td>
</tr>
<tr>
<td>Image size</td>
<td>100 KB</td>
<td>10 – 1000 KB</td>
</tr>
<tr>
<td>Number of features</td>
<td>256</td>
<td>128 – 512</td>
</tr>
<tr>
<td>Network bandwidth</td>
<td>10 MB</td>
<td>1 – 1000 MB</td>
</tr>
<tr>
<td>Traffic type</td>
<td>constant rate</td>
<td></td>
</tr>
<tr>
<td>Number of local databases</td>
<td>50</td>
<td>10 - 1000</td>
</tr>
<tr>
<td>Local dataset size</td>
<td>20</td>
<td>1 - 30000</td>
</tr>
<tr>
<td>Number of nearest neighbors</td>
<td>1</td>
<td>1 - 30</td>
</tr>
<tr>
<td>Number of queries</td>
<td>1000</td>
<td>1 – 1000</td>
</tr>
<tr>
<td>Query rate</td>
<td>10 query / second</td>
<td>1 – 150 queries / second</td>
</tr>
<tr>
<td>Maximum SSM levels</td>
<td>10</td>
<td>5 – 10</td>
</tr>
<tr>
<td>Minimum fan-out</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Maximum fan-out</td>
<td>10</td>
<td>2 – 20</td>
</tr>
<tr>
<td>Index/local node access time ratio</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>
Search Cost

Figure 12 illustrates the number of images visited during the course of nearest-neighbor retrieval for different content-based indexing schemes — R*-tree, M-tree, SS-tree, SR-tree, VP-tree, and SumLog. The number of nearest neighbors varies from 5 to 25. As can be observed, SumLog offers better performance than most of the indexing models in term of search cost (i.e. the number of visited data objects). The superiority of SumLog over R*-tree, M-tree, SS-tree, and SR-tree is due to its content-based clustering capability. The other indexing trees (except VP-tree) enclose data objects with rectangles and sphere regions, respectively. As a result, in the nearest-neighbor searching operations, relative to the user request, many irrelevant data objects may be visited.

![Figure 12: The comparison on search cost.](image)

In addition, SumLog achieves comparable performance to VP-tree as the number of nearest neighbors increases from 5 to 25. VP-tree takes advantage of vantage-point-based partitioning to locate the data objects within a certain distance to the query example. This partitioning is highly efficient in removing the irrelevant data objects. SumLog, on the other hand, makes use of probability density function (PDF) to determine the number of possible query result candidates in each database and thereby only visiting the data objects that are most relevant to the query. Therefore, both SumLog and VP-tree requires much less number of visited data objects than the other indexing schemes.
Retrieval Accuracy

To evaluate the accuracy of retrieved query results, we compared and contrasted the indexing schemes against the query results obtained from centralized search. In the centralized search, we assumed all image data objects were stored in one database, and could be accessed using a centralized index for the resolution of nearest-neighbor queries. The results of distributed retrieval, either from SumLog or from other indexing schemes, were compared with the centralized scheme for the percentage of matches. Higher matching percentage indicates higher accuracy. We ran our simulator up to 1000 seconds, and the result is shown in Figure 13. At the same time, we also tested the unlimited search time for each scheme to achieve the 100% accuracy, which is shown in Table 3 (the values were taken from the average of 100 simulation runs).

![Figure 13: The impact of search time on accuracy (k=10).](image)

<table>
<thead>
<tr>
<th>Indexing Schemes</th>
<th>Unlimited Search Time (Sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>50 Databases</td>
</tr>
<tr>
<td><em><em>R</em>-tree</em>*</td>
<td>8377.56</td>
</tr>
<tr>
<td><strong>M-tree</strong></td>
<td>3427.76</td>
</tr>
<tr>
<td><strong>SS-tree</strong></td>
<td>2807.17</td>
</tr>
<tr>
<td><strong>SR-tree</strong></td>
<td>3644.72</td>
</tr>
<tr>
<td><strong>VP-tree</strong></td>
<td>1458.54</td>
</tr>
<tr>
<td><strong>SumLog</strong></td>
<td>1348.13</td>
</tr>
</tbody>
</table>
One important metric for content-based retrieval schemes is the retrieval accuracy. The performance of a scheme can be evaluated by its accuracy when searching within a limited time. In Figure 13, we compared the accuracy (in term of matching percentage) of R*-tree, M-tree, SS-tree, SR-tree, VP-tree, and SumLog. The search time ranges from 100 to 1000 seconds. The dataset consists of 5000 images randomly distributed among 10 local databases, thereby giving different number of data objects in different local databases.

As can be concluded from Figure 13, all of the search schemes achieve higher accuracy as search time increases. However, their accuracy at each time limit differentiates due to their different indexing rationales. R*-tree employs the minimum bounding rectangles (MBRs) to organize multimedia data objects. By recursively dividing larger rectangles into smaller ones, R*-tree provides the searching route to specific multimedia data objects. If the required number of nearest neighbors cannot be found in a small rectangle, R*-tree needs to return to a bigger rectangle to find more objects.

The accuracy of M-tree and SS-tree also shows a similar trend as the search time increases. M-tree and SS-tree are distance-based searching models, which organize data objects according to their distances computed from feature vectors. The data objects are divided into sphere regions of different centers and radiuses, and the M-tree and SS-tree perform content-based search by recursively reducing the searching sphere. The better performance of M-tree and SS-tree in comparison with R*-tree is due to the fact that their search involves less backtracks in the indexing tree.

The SR-tree, which employs MBRs for the indexing nodes and uses intersections of spheres and rectangles for the local nodes, incurs larger depth than the SS-tree. Therefore, it takes longer time for SR-tree than SS-tree to find the similar data objects. As an improvement to these models, the VP-tree outperforms the aforementioned schemes by using vantage points in its indexing infrastructure for distance-based partitioning of data objects.

In the nearest-neighbor retrieval, if given unlimited search time, each search scheme achieves 100% accuracy because it can traverse every local database for selecting the data objects with smallest distances. However, if the search time is limited, a search scheme can
only visit a limited number of local databases which may not contain the most similar data objects. Therefore, the search accuracy and search time are interrelated. From Figure 13, one can conclude that SumLog achieves higher accuracy, especially when the search time is small. This is due to the fact that the SSM organizes data objects based on their semantic distances and clusters content-similar data objects together. The content-based search can be performed in a set of data objects that are most relevant with the query. Therefore, within a limited time, SumLog is faster than other schemes when performing the nearest neighbors search.

In another simulation run, we evaluated the impact of data content distribution on the retrieval accuracy. Figure 14 shows the result. As expected, all search schemes achieve higher accuracy as the data distribution density increases (i.e., less number of local databases). This is reasonable since higher data density increases the probability of finding nearest neighbors within one local database, and thereby reducing the overall search time and network traffic. As can be seen from Figure 14, the data content distribution has less impact on SumLog than on other schemes. This is a justifiable observation, since SumLog has the capability of clustering multimedia data objects based on their semantic contents. Furthermore, SumLog can perform nearest-neighbor retrieval in parallel among several sibling nodes (as shown in Algorithm 2), and thus is capable of finding more similar data objects within a limited time.

![Figure 14: The impact of data distribution on accuracy.](k=10, 500 seconds)
In contrast with SumLog, the other schemes achieve relatively low accuracy in the nearest-neighbor retrieval, especially when the number of data objects per local database is small. The rectangle-based indexing schemes (R*-tree and SR-tree) rely on the MBRs to specify the data content relationships between local database nodes and indexing nodes, while the rectangles cannot provide the information about semantic distances between data objects. Therefore, R*-tree has the lowest accuracy and SR-tree achieves lower accuracy than SS-tree and VP-tree, although SR-tree has used several techniques to reduce the overlapping between indexing branches and improve the search accuracy [14]. At the same time, the distance-based indexing schemes (M-tree, SS-tree, VP-tree) construct the indexing trees based on the semantic distances, and use the distance information to facilitate the content-based retrieval. However, the SS-tree uses spheres for enclosing the data objects, which causes overlapping between indexing branches, thereby reducing its capability of finding the accurate local databases within limited time. The VP-tree, in contrast, uses the distances only to partition the data objects into separated data sets, and therefore avoids the overlapping between indexing branches and improves search accuracy.

**Disk Access Frequency**

Figures 15, 17, and 18 compare the number of disk accesses for different content-based indexing models. To compare the SSM hierarchy with the tree-shaped indexing models (R*-tree, M-tree, SR-tree, SS-tree, and VP-tree) presented in [7], we analyzed disk accesses at two different levels: 1) local nodes (physical databases) that contain multimedia data; and 2) index nodes (summary-schemas nodes) that form an indexing infrastructure. For this simulation run, the underlying summary-schemas nodes have the maximum fan-out of 20, and each local node has a maximum of 512 data objects. The actual number of data objects at each local node is randomly determined in the simulation run. The data set consists of randomly generated data objects that have 256 features. The query is intended to find the nearest 21 neighbors relative to a particular object in the data set. Experiments were run on different data set sizes, ranging from 4,000 to 32,000 image data objects.
To examine the impact of data set size on the search time, we also tested the execution time of each scheme, which is shown in Figure 16. As can be concluded from Figure 16, the increase in data size incurs drastic increase in execution time for R*-tree, M-tree, and SS-tree, while SumLog and VP-tree are less affected. This fact further illustrates that SumLog and VP-tree have the capability of searching similar data objects using relatively smaller time.
In the above discussion, Figure 15 shows the simulation result of different indexing schemes. The total number of disk access shown in Figure 15 is the sum of the number of disk accesses at indexing nodes (Figure 17) and the number of disk accesses at local nodes (Figure 18). As the data set size increases, both the average number of data objects per local node and the indexing tree depth increase, thereby causing more disk accesses for the content-based retrieval. From Figure 15, we can also observe that the region-based indexing trees (e.g. the rectangle-based SR-tree and the sphere-based SS-tree) require more disk accesses than the partition-based trees (e.g. VP-tree). This is due to the fact that VP-tree divides the data objects into separate data sets in each partition, which removes the overlapping between branches. In the search process, VP-tree can divide the data set into relevant and irrelevant sub sets, and thereby locating the similar data objects using less disk accesses.

As can be concluded (Figure 15), SumLog requires fewer number of disk accesses. In the SSM hierarchy, semantically similar data objects are organized within the same clusters. The intra cluster semantic distances are smaller than the inter cluster semantic distances. The difference between the clustering of SumLog and VP-tree is that SumLog is capable of describing the content of a cluster using logic expressions. Hence, when the number of image data objects within a cluster is greater than the required number of nearest neighbors, there is no need to traverse the SSM hierarchy in order to access other clusters. Otherwise, SumLog can forward the query to other relevant clusters through synonym and hypernym/hyponym links for the query resolution. In comparison, the other indexing models (R*-tree, M-tree, SS-tree, SR-tree, and VP-tree) are not capable of content description. As a result, the searching strategy leads to access “unrelated” data objects which results in a larger number of disk accesses.

Figure 17 depicts the number of disk accesses at the index nodes for different indexing models. In comparison with Figure 12, we noticed that although VP-tree visits fewer number of data objects than SS-tree and SR-tree, it requires comparable number of index node accesses as SS-tree and SR-tree. The reason is that VP-tree is a partition-based indexing model, and each partitioning operation divides a set of data objects into two
smaller data sets based on semantic distances. The partitioning causes two effects: 1) The
good effect is that VP-tree can locate the most relevant data set for the query; 2) However,
the depth of VP-tree is relatively larger than SS-tree and SR-tree, because each partitioning
operation can only generate two smaller data sets, making the VP-tree similar to a binary
indexing tree. Therefore, it takes longer route to travel from the root to the similar data
objects. As expected, content-based clustering capability of SumLog reduces the number
of disk accesses at the index nodes, since the nearest neighbors usually are grouped
together within one cluster.

![Figure 17](image)

**Figure 17:** The comparisons on index-node disk accesses.

Finally, Figure 18 shows the number of disk accesses at the local node level. Once
again, the superiority of SumLog in reduced number of disk accesses is due to its ability to
cluster semantically similar objects together. At the same time, the logic-based content
representation can also be used in local databases for the description of data contents.
Therefore, it also helps locating the similar data objects in the local databases and thereby
reducing the disk accesses.
Network Traffic

Network traffic is an important metric in evaluating the performance of content-based retrieval schemes in a distributed environment. Considering the large data volume of multimedia data objects, it is highly desirable to resolve queries in local databases, thereby reducing the costly communications between different local databases. In order to test the impact of search schemes on the network traffic, we tuned the simulator to examine the average number of messages per second between the nodes (including local nodes and index nodes).

In the query resolution, the local databases and indexing nodes communicate with each other for finding the databases containing the requested data objects. There are two types of messages during the query resolution: 1) the query messages that are forwarded to the local databases, and 2) the data messages that are returned to the query origin node as the query result. The depth of indexing tree has much impact on the number of query messages. Smaller depth implies less forwarding messages between the indexing nodes and hence the query can be forwarded to the local databases using fewer messages. Therefore, SS-tree and SR-tree have less number of query messages forwarding along the search paths in their indexing trees than R*-tree and M-tree. In addition, the retrieval
accuracy also has impact on the number of messages. VP-tree has a higher accuracy than SS-tree and SR-tree due to its capability of partitioning the data objects based on the semantic distances. To find the nearest neighbors, VP-tree needs to search less number of local databases and causes less mismatches. Therefore, VP-tree causes less data messages returned to the query origin node.

Figure 19: The network traffic in different data densities.
(query rate = 10 query/second)

Figure 19 also shows that SumLog incurs less message overhead than the other schemes. The reason is that SumLog resolves a query within the local nodes that contain the semantically most related data objects. As the number of local databases increases, the data density decreases, and the probability of resolving queries within one local database drops. The other search schemes need to search multiple local databases to resolve queries. In comparison, the SSM uses synonym links to find the shortcut paths between content-similar data objects, and uses probability density function to divide the $k$-NN query into sub queries that are easier to be resolved within a single local database. Therefore, SumLog incurs fewer query and data messages.
To study another factor that may affect the network traffic, we ran our simulator with different settings of query rate. The simulation result is shown in Figure 20. As the query rate increases from 1 to 150, all search schemes incur larger network traffic. However, SumLog has less traffic increase than the other schemes. The reason is that the other models incur network traffic proportional to the number of queries, since the query resolution always starts from the root node. In contrast, SumLog can start query resolution at any node in the SSM hierarchy and make use of the content-based clustering and the content distribution description (i.e. PDF) to reduce the overall network traffic.

![Figure 20: The network traffic with different query rates.](image)

### 3.3.3 Performance Analysis Conclusions

Section 3.3 includes the performance analysis of the integration of SSM and logic-based content representation from both theoretical and experimental aspects. Based on the theoretical study of the SSM’s indexing structure and content-based query resolution, one can discover two properties of SumLog:

- The logic-based content representation is integrated in the SSM hierarchy to facilitate the content-based organization. This property is obtained through two mechanisms: 1) The logic expressions help to describe the semantically similar multimedia data objects collectively as concise terms, making it easier to group content-similar data objects into clusters; 2) The combinatory table is used for
rewriting the data contents of multimedia data sources into simpler logic terms, thereby helping to remove the overlapping and redundancy.

- SumLog achieves the optimal performance in terms of search cost. As shown in section 3.3.1, the SSM hierarchy based on the hypernym/hyponym semantic relationships can effectively describe any indexing infrastructure and allow query resolution with less routing overhead. In addition, the SSM hierarchy accepts the query at any node, and in many occasion the query is resolved through a part of the indexing hierarchy. Generally, the SSM hierarchy utilizes the semantic locality of queries and fulfills the query resolution using relatively smaller search cost than other indexing schemes.

With the experimental evaluation of various performance metrics such as search cost, retrieval accuracy, disk access frequency, and network traffic, one can conclude that SumLog provides an efficient platform for content-based multimedia retrieval in the distributed environment based on the following observations:

- SumLog performs distributed content-based retrieval with reduced search cost. This is achieved through its capability of semantic-based clustering and parallel query processing.

- SumLog improves the retrieval accuracy in contrast to other indexing schemes by exploiting the logic-based content expressions to restrict the search scope to only the local databases that contain the most similar data objects.

- SumLog reduces the disk access frequency at both indexing nodes and local nodes by organizing the multimedia data objects based on their semantic similarity. The content-similar data objects are organized within one cluster — this increases the probability of resolving query in one local database and hence, reduced disk accesses.

- SumLog avoids large network traffic in the query resolution in a networked environment. By locating the most semantically relevant local databases in the
optimized short search path and returned the most similar data objects as query result, SumLog reduces the number of query messages and data messages in the query resolution process.

Summarizing the analysis of both theoretical and experimental aspects, the logic-based content representation, integrated within the framework of the SSM, demonstrates efficient content-based indexing and retrieval capability in the consideration of various performance metrics, making it suitable for distributed multimedia data access.

3.4 Summary of Chapter Contents

In this chapter, we introduced the general foundation of our work — a logic-based model for the representation of multimedia data contents. More specifically, we used first-order conjunctive and disjunctive expressions to describe the contents of data objects. This model has the capability of integrating both object-level and granule-level features of multimedia data objects and supporting all three types of content-based retrieval (form-based, pure-semantic-based, and the combination of both form and semantics).

To optimize the logic-based representation terms, we gave an algorithm that uses combinatory table to reduce the lengths of logic terms. We showed that the algorithm fulfills this task by using a collection of rules.

To provide logic-based content representation for distributed multimedia data repositories, we integrated the logic-based model in a global data access framework — the SSM. We discussed the mapping of hypernym/hyponym and synonym relationships with inclusion relationships in the logic domain, and examined the mapping method. In this thesis, we consider the logic model and the SSM as the same framework since they can be seamlessly integrated, and proposed the Summary Logic scheme (SumLog) based on this analysis.

Since the ultimate goal of content representation is to facilitate content-based multimedia information retrieval, we presented the query resolution method (i.e. nearest-neighbor retrieval) in the domain of SSM. We also conducted theoretical and experimental evaluations regarding the integration of the SSM and logic-based content representation for content-based multimedia retrieval.
The SSM is not limited to the usage of wired networks. It can also be used in wireless networks consisting of mobile data sources. In the next chapter, we will discuss the application of the SSM on multimedia data manipulation in ad hoc networks.
4 MULTIMEDIA ACCESS IN AD HOC NETWORK

The SSM was originally proposed based on client/server paradigm for semantic-based textual data management in distributed heterogeneous environments. An agent-based extension of the SSM, namely MAMDAS, has been reported in the literature as an attempt to eliminate the shortcomings of client/server paradigm. Furthermore, MAMDAS relies on IEEE 802.11b infrastructure wireless LAN as the target mobile environment. As shown in chapter 3, the SSM can be extended with a logic-based model for multimedia data content representation. These extensions were intended to improve the performance and flexibility of the SSM. However, these extensions assumed a single entity as a local node. The scope of the SSM can be improved further by allowing a collaborating community of interconnected wired/wireless nodes as a local node. Such a community of wired/wireless nodes can interact with each other via a structured/unstructured communication infrastructure.

In recent years, mobile ad hoc networks have been increasingly popular in building temporary network connections in special areas, such as battlefields or disaster spots, where infrastructures are destroyed or too expensive to be built. Consequently, content-based multimedia information retrieval in ad hoc networks is becoming an emerging research topic. In the SSM hierarchy, a local node can be a community of mobile computing devices that form an ad hoc network. In this community, communications and query processing is performed in the peer-to-peer fashion.

Multimedia data manipulation and access in an unstructured community of mobile data sources introduces new challenges that require initiative solutions and protocols to leverage the technological limitations, i.e., bandwidth, computing capability, and storage.

In this chapter, we propose novel solutions to facilitate multimedia data management in mobile ad hoc networks. Our goal is to provide a scheme that performs content-based multimedia information retrieval in mobile data sources with reduced search cost and improved accuracy. We elaborate the proposed solutions and evaluate their effectiveness using theoretical analysis and experimental study.

4.1 Introduction and Motivation

Due to the recent advances in visualization techniques, the communications in ad hoc networks are no longer limited to textual information, and other forms of information sources such as image and video are becoming desirable [84]. Within the framework of ad
hoc networks, multimedia information retrieval can enrich the communications between the mobile nodes, making their messages more expressive. It should also be noted that in contrast with the lower-bandwidth wide-area wireless networks such as cellular networks (100Kbps for GPRS and 384Kbps for W-CDMA), ad hoc networks comparatively offer higher bandwidth (11Mbps for IEEE 802.11b and up to 54Mbps for IEEE 802.11a and 802.11g) [70], technically allowing the communication of multimedia information.

The previous research did not provide efficient solutions to multimedia data retrieval in an ad hoc infrastructure. With very few exceptions [69], search techniques are based on centralized or flooding strategies, ignoring the content distribution among mobile nodes [68, 71, 75]. Centralization and flooding strategies make it possible to handle multimedia data retrieval; however, they may also result in either a single point of failure or high search cost. Generally, the challenges of ad hoc networks on multimedia information retrieval can be categorized as follows:

First, scalability may vary as the network configuration changes. A practical ad hoc network may consist of a few centralized data source nodes (data centers) and several client nodes that request data from the data centers. From practical point of view, the mobile clients and data servers have different data processing capability, memory space, and communication speed, and therefore they should be treated differently to offer a reasonable performance in resolving content-based multimedia queries. Also note that if one client node just accessed a data object from the data center, it is quite possible that nearby nodes may try to access the same data object in near future [69]. As a result, considering the constraints of mobile nodes, in an ad hoc infrastructure, methods should be developed to make full use of the capabilities of data centers while reducing the communication and computation costs of the query resolution.

Second, in an ad hoc network, the communications between mobile nodes may be carried out in a peer-to-peer fashion. However, the paths between the mobile nodes are constantly changing due to the mobility. As a result, the data source nodes are generally unknown at the requesting node and identification of data source in real-time applications is hard to achieve if traditional routing algorithms are employed [71]. One solution to the content-based queries is traversing the whole network [84]. However, this solution drastically consumes the resources — memory, network bandwidth, and power. To improve resource utilization, the accurate identification of data sources requires an organizational infrastructure according to data content distribution. Consequently, to facilitate efficient multimedia information retrieval while considering the constraints of ad hoc networks, one
needs to devise new methods that automatically and pervasively obtain information about
data content distribution.

Third, most of the existing retrieval schemes for ad hoc networks rely on textual
information and cannot be directly applied to multimedia data [42, 84].

Considering the above challenges, we propose an effective multimedia data access
method in the mobile environments. Two types of ad hoc networks will be considered:
Networks consisting of data centers and mobile clients, and networks composed of peer
nodes. In the first case, we cluster the mobile nodes based on the semantic relationships of
their data contents and use the clusters to facilitate query resolution. For the networks
composed of peer nodes, we exploit the content distribution over mobile nodes by using a
semantic-aware caching scheme. The semantic-aware caching scheme will be elaborated in
Chapter 5. We will not only describe the rationale of semantic-aware caching, but also give
the QoS-aware cache management and update methodology in Chapter 5. In the rest of this
chapter, we will discuss the proposed clustering technique and how to integrate it with the
SSM. In addition, we will simulate, evaluate, and analyze the technique.

4.2 SSM-Based Clustering for Mobile Clients and Data Centers

In the global information system built upon wired and wireless connections, the
local data source nodes can be a collection of mobile computing devices communicating as
ad hoc networks. The semantic relationships of the data contents in the mobile nodes make
it possible to partition the ad hoc networks into clusters of content-similar nodes. The
content-based clustering process requires effective representation of cluster contents, and a
cluster centroid node that integrates contents of the mobile data sources and facilitates
query resolution.

The logic-based representation model presented in Chapter 3 provides a paradigm to
represent multimedia data. Based on this paradigm, contents of the multimedia data objects
can be automatically identified, summarized, and expressed as logic expressions. Logic
representation of semantic contents of data sources would also allow simple and efficient
recognition of similar entities that assists classification and clustering process.

4.2.1 Problem Formulation

The content-based multimedia retrieval (similarity search) in an ad hoc network can
be described as follows: Given the set of multimedia data objects \( X \) and the ad hoc network
for a specific integer constant \( k \) and a given query object \( x_q \), return \( k \) data objects \( k\text{-NN}(x_q, X) = \{x_1, x_2, ..., x_k\} \subset X \) such that:

\[
\forall (x \in X \land x \notin k\text{-NN}(x_q, X)) \ dist(x_q, x) \geq dist(x_q, x_i) \ (1 \leq i \leq k)
\]

In the context of ad hoc networks, the resolution of content-based query may cause flooding and forcing pair-wise comparisons of multimedia data objects in each mobile node. The flooding approach is resource intensive and hence may not be applicable in real-time applications. Therefore, alternative approaches should be devised to perform the content-based multimedia retrieval with reduced search cost.

**4.2.2 Content-Based Clustering**

**Definition 6: Content-Related Nodes**

Suppose a mobile node \( n_i \) contains multimedia data objects \( \{x_1, x_2, ..., x_m\} \), which are collectively denoted as \( D(n_i) \). According to the logic-based representation of multimedia data contents defined in Definition 5, each mobile node \( n_i \) can obtain a content summary \( S(n_i) \) that abstracts the contents of its multimedia data objects. Given a pair of nodes \( n_0 \) and \( n_1 \), they are content-related iff:

\[
S(n_0) \oplus S(n_1) \neq S(n_0) \lor S(n_1)
\]

where \( S(n_0) \oplus S(n_1) \) is defined as \( (S(n_0) \land \neg S(n_1)) \lor (\neg S(n_0) \land S(n_1)) \). This definition means that the contents of \( S(n_0) \) and \( S(n_1) \) have some overlapping. In other words, if nodes \( n_0 \) and \( n_1 \) are content-related, they must have some common data entities, i.e. \( D(n_0) \cap D(n_1) \neq \emptyset \).

**Definition 7: The Content-Based Cluster**

Suppose an ad hoc network \( N \) comprises \( k \) mobile nodes \( n_1, n_2, ..., n_k \). Let \( n_i \approx n_j \) denote the content-related relationship between \( n_i \) and \( n_j \), and \( n_i \neq n_j \) denote that \( n_i \) and \( n_j \) are not content-related. Then a content-based cluster \( C \) is defined as follows:

\[
C_i = \{ n_i \mid \forall n_j \in C, n_j \approx n_i; \text{ and } \forall n_k \notin C, n_k \neq n_i \} \nonumber
\]

Based on the definitions of content-based clusters, we propose an algorithm that partitions an ad hoc network into clusters. Table 4 shows the notations used in the algorithm. Algorithm 3 describes the process of content-based clustering. The mobile nodes within a cluster are content-related, while mobile nodes of different clusters share very few common semantic contents.
Table 4: Simulation setting for distributed search.

<table>
<thead>
<tr>
<th>Items</th>
<th>Notations</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V(N)$</td>
<td>The set of mobile nodes { $n_1$, $n_2$, ..., $n_k$ }</td>
</tr>
<tr>
<td>$\Gamma$</td>
<td>The set of clusters</td>
</tr>
<tr>
<td>$F(x)$</td>
<td>The function of selecting an element from set $x$</td>
</tr>
<tr>
<td>$C(\Gamma, i)$</td>
<td>The $i^{th}$ cluster in $\Gamma$</td>
</tr>
<tr>
<td>$S(x)$</td>
<td>The function of generating content summaries of mobile node $x$</td>
</tr>
</tbody>
</table>

Algorithm 3: Content-based clustering

1. $\Gamma \leftarrow F(V(N))$
2. $K \leftarrow V(N) - \Gamma$
3. while $|K| > 0$ do
4. $x \leftarrow F(K)$
5. $K \leftarrow K - \{x\}$
6. for $i = 1, ..., |\Gamma|$ 
7. if $S(C(\Gamma, i)) \oplus S(x) \neq S(C(\Gamma, i)) \lor S(x)$
8. merge $x$ with $C(\Gamma, i)$
9. if $x$ is not merged with any cluster in $\Gamma$
10. create a new cluster in $\Gamma$

Algorithm 3 repeatedly creates content-based clusters as follows: Initially, the set of clusters $\Gamma$ only contains one cluster, which contains just one randomly selected node. In steps 3-5, a new node $x$ is selected from the remaining node set $K$ as the candidate. $x$ is then compared with the existing clusters in $\Gamma$ to locate the content-similar cluster. If no such a cluster is found, $x$ will then be used to create a new cluster in $\Gamma$. Finally, all nodes in $K$ will be selected and assigned to the proper content-based clusters.

4.2.3 Integrating Clusters into the SSM Hierarchy

As noted in section 4.2.2, an ad hoc network could be partitioned into clusters where each cluster contains mobile nodes with similar or overlapping data objects. The contents of nodes within a cluster are integrated to form a content summary -- a concise description of the semantic contents of the cluster. Cluster-level content summaries are then integrated and fused together (based on their semantic similarities) to form higher-level clustering,
whose contents are represented using more generalized semantic descriptions. This process is recursively performed until the whole ad hoc network is represented as one cluster. To accommodate the cluster-level content summaries, a centroid node is selected for each cluster.

Note that this level-by-level integration follows the same pattern of organizing the SSM hierarchy. When the whole ad hoc network is represented as one cluster, the cluster’s representation is then used as the local node in the SSM. Generally, the semantic-based clustering can be considered as the implementation and extension of the SSM in the domain of ad hoc networks.

**Definition 8: The Cluster-Level Content Summary**

Given a cluster \( C_i = \{ n_{i1}, n_{i2}, \ldots, n_{iq} \} \), the cluster-level content summary, denoted as \( S_c(C_i) \), is defined as follows:

\[
S_c(C_i) = \text{int}(\bigcup_{j=1}^q S(n_{ij})).
\]

(13)

where \( \text{int} \) is the data integration function that converts hyponym terms into hypernym terms.

**Definition 9: The Cluster Centroid**

Given cluster \( C_i = \{ n_{i1}, n_{i2}, \ldots, n_{iq} \} \), let \( V(n_{ij}) = \{ v_1(n_{ij}), v_2(n_{ij}), \ldots, v_k(n_{ij}) \} \) denote a vector of the hardware characteristics of node \( n_{ij} \), such as memory storage, computation capability, and communication speed. Then the centroid node of \( C_i \) can be defined as \( \mathbf{c}(C_i) \):

\[
\mathbf{c}(C_i) = n^*, \forall n' \in C_i, |n \cap S_c(C_i)| \leq |n^* \cap S_c(C_i)| \land c_s^*v_s(n^*) \leq c_s^*v_s(n') (1 \leq s \leq k).
\]

(14)

here \( c_1, c_2, \ldots, c_k \) are predefined coefficients and equation (14) indicates that the centroid node \( n^* \) possesses the largest portion of data contents in the cluster \( C_i \), and its hardware capabilities are superior than the other nodes, such as computationally most powerful, largest storage, highest communication speed, etc.

**Definition 10: The Content Similarity of Clusters**

Given a set of clusters \( \Gamma = \{ C_1, C_2, \ldots, C_r \} \), the content similarity of the clusters in \( \Gamma \) is denoted as \( S_s(\Gamma) \):

\[
S_s(\Gamma) = \text{int}(\bigcup_{C_i \cap C_j \neq \emptyset} (S_c(C_i) \cap S_c(C_j))).
\]

(15)

here \( \text{int} \) is the data integration function mentioned in Definition 8.
Definition 11: The $t$-Partition of Clusters

Given $r$ clusters $C_1, C_2, \ldots, C_r$ and an integer $t$ ($1 < t < r$), the $t$-partition of the clusters is the process of partitioning the $r$ clusters into $t$ groups $\Gamma_1, \Gamma_2, \ldots, \Gamma_t$ satisfying that any two different group do not share the same data contents or only share the possibly minimal overlapping. The $t$-partition is useful in the merging process of clusters and their data contents.

Consider an ad hoc network $N$ containing $k$ mobile nodes: $n_1, n_2, \ldots, n_k$. Let these nodes be partitioned into $r$ clusters $C_1, C_2, \ldots, C_r$ based on their data contents. For any cluster $C_i$, one can select the centroid node of the cluster as follows:

- Find a node $n^*$ that has highest computation capability, memory storage, and communication speed, and define it as the centroid of $C_i$. In case there is a tie among more than one centroid candidate in cluster $C_i$, we choose the node whose data content has maximum overlapping with the cluster-level content summary as the centroid.

- Build an indexing hierarchy on top of the centroid nodes as follows: 1) The mobile nodes of cluster $C_i$ are considered as children of its centroid; 2) Suppose the ad hoc network is decomposed into $r$ clusters $C_1, C_2, \ldots, C_r$, we partition these clusters into $t$ groups ($1 < t < r$) using $t$-partition as in definition 11, and represent the semantic contents of each group using more generalized descriptions (i.e. hypernyms). In the research reported in this thesis, we maintain an on-line taxonomy that can find the generalized descriptions for cluster-level content summaries of any given cluster [19]. Steps 1 and 2 are recursively applied until the whole ad hoc network is represented as one cluster. This clustering process constructs a hierarchy that can be used as an indexing infrastructure.

Note that the definitions of $t$-partition and indexing hierarchy are based on the assumption of reliable pair-wise message communications between mobile nodes in the order of their generation, using some existing hop-by-hop routing protocol, such as AODV or DSR. Due to the lack of static infrastructure in the ad hoc network, the parent/child links in the hierarchical index are virtual connections that correspond to multi-hop paths in the network. Based on these assumptions, we propose Algorithm 5 and Algorithm 6 for the $t$-partition and the construction of the hierarchical indexing, respectively.
In the algorithms 5 and 6, we address the content-aware organization of mobile data sources to facilitate multimedia retrieval in the ad hoc network. In comparison with the traditional centralized content-based indexing models, the proposed hierarchical indexing has the following properties:

- The fundamental idea of the indexing hierarchy is based on the content distribution of mobile nodes. The data content of each node is represented using first-order logic expressions. The ad hoc network is partitioned into clusters of semantically similar nodes to facilitate content-based search.
- The indexing hierarchy is overlaid on the mobile nodes, which corresponds to the infrastructure-free nature of ad hoc networks that supports scalability.
- A centroid node in the indexing hierarchy can “float” from one mobile node to another in accordance with bandwidth, topology changes, or query distribution to achieve better robustness, performance, and load balancing.
- The ad hoc network is partitioned into clusters based on semantic similarity. The mobile nodes in each cluster share the same or similar semantic contents. Due to the semantic locality of queries, in most cases the query processing is performed within one or a few clusters.

Table 5: Notations related to the t-Partition.

<table>
<thead>
<tr>
<th>Items</th>
<th>Notations</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Gamma )</td>
<td>The set of clusters ( \Gamma = { C_1, C_2, \ldots, C_r } )</td>
</tr>
<tr>
<td>( t )</td>
<td>The number of expected partitions</td>
</tr>
<tr>
<td>( \Theta )</td>
<td>The set of partitions</td>
</tr>
<tr>
<td>( F(\Theta, x) )</td>
<td>The function of selecting an element from set ( x ) satisfying the minimal overlapping with ( \Theta )</td>
</tr>
<tr>
<td>( P(\Theta, i) )</td>
<td>The ( i^{th} ) partition in ( \Theta )</td>
</tr>
<tr>
<td>( S_c(x) ) (in Definition 8)</td>
<td>The function of generating content summaries of cluster ( x )</td>
</tr>
<tr>
<td>( S_s(x) ) (in Definition 10)</td>
<td>The function of generating content similarity (or summary) of clusters in partition ( x )</td>
</tr>
</tbody>
</table>

86
Algorithm 5: t-Partition

1. \( \Theta \leftarrow F(\Theta, \Gamma) \)
2. \( \Gamma \leftarrow \Gamma - \Theta \)
3. while \(|\Gamma| > 0\) do
4. \( K \leftarrow F(\Theta, \Gamma) \)
5. if \(|\Theta| < t\)
6. \( \Theta \leftarrow \Theta \cup K \)
7. \( \Gamma \leftarrow \Gamma - K \)
8. else
9. \( L \leftarrow S_s(K) \cap S_s(P(\Theta, 1)) \)
10. for \( i = 2 \) to \( t \)
11. if \(|S_s(K) \cap S_s(P(\Theta, i))| > |L|\)
12. \( L \leftarrow S_s(K) \cap S_s(P(\Theta, i)) \)
13. \( J \leftarrow i \)
14. \( P(\Theta, J) \leftarrow P(\Theta, J) \cup K \)

In the initialization of Algorithm 5, the partition set \( \Theta \) contains only one randomly selected cluster from \( \Gamma \). Then more clusters are selected and inserted in \( \Theta \) until \(|\Theta| = t\). If the cluster set \( \Gamma \) still contains additional clusters, these clusters will be repeatedly selected and compared with the existing partitions in \( \Theta \). Each cluster will then choose the most relevant partition and merge in it. Finally, the clusters are divided into \( t \) partitions with close inner-partition content relationships.

Table 6: Notations related to the t-Partition.

<table>
<thead>
<tr>
<th>Items</th>
<th>Notations</th>
</tr>
</thead>
<tbody>
<tr>
<td>( h )</td>
<td>The expected height of the hierarchy</td>
</tr>
<tr>
<td>( t(i) )</td>
<td>The expected partitions at level ( i ) of hierarchy</td>
</tr>
<tr>
<td>( P_t(x, t) )</td>
<td>The function of ( t )-partition of elements in ( x )</td>
</tr>
<tr>
<td>( \omega(x) )</td>
<td>The function of describing the semantic content of ( x ) using a word in the on-line thesaurus ( \psi )</td>
</tr>
<tr>
<td>( P_u(x) )</td>
<td>The function of finding the parent node for ( x )</td>
</tr>
<tr>
<td>( H(x, y) )</td>
<td>The function of computing the hypernym of the words ( x ) and ( y )</td>
</tr>
</tbody>
</table>
Algorithm 6: Constructing hierarchical indexing

1. $\Theta \leftarrow P_\ell(\Gamma, t(1))$
2. \textbf{for} $i = 1$ to $t(1)$
3. \hspace{1em} compute $\omega(P(\Theta, i))$
4. \textbf{for level} = 2 to $h$
5. $\Omega \leftarrow P_\ell(\Theta, t(\text{level}))$
6. \textbf{for} $j = 1$ to $t(\text{level})$
7. \hspace{1em} $\omega(P(\Omega, j)) \leftarrow \emptyset$
8. \textbf{for} $k = 1$ to $t(\text{level} - 1)$
9. \hspace{1em} $\omega(P_d(P(\Theta, k))) \leftarrow H(\omega(P_d(P(\Theta, k))), \omega(P(\Theta, k)))$
10. $\Theta \leftarrow \Omega$

The purpose of Algorithm 6 is to build the SSM hierarchy based on $t$-partition and hypernym/hyponym relationships. Starting from the bottom level, each node calculates its content description as the hypernym summary of its children nodes’ contents. This bottom-up summarization process continues until the hierarchy is completely built.

With the help of this indexing hierarchy, content-based multimedia retrieval is performed as a clearly aimed searching process, navigated by the semantic content descriptions on the centroid nodes. The content-based search process is described as follows:

- When a query (i.e. an example object) is submitted to a mobile node, first try to resolve the query locally.
- If the query cannot be resolved locally, then forward the query to the cluster centroid node, and try to resolve it at the cluster-level.
- If the query is resolved at the cluster level, then forward the result to the mobile node containing the corresponding multimedia data; otherwise, keep forwarding the query to higher-level clusters until it is resolved.

With this search strategy, a query may be resolved within a single cluster (intra-cluster queries) or by cooperation of several clusters (inter-cluster queries). Note that, in general, the queries generated by a mobile node are likely to be semantically related with the data contents of corresponding cluster; hence, this “semantic locality” helps to reduce the average search cost by restricting the search scope to a cluster.
As an example of the content-based clustering, Figure 21 represents a collection of multimedia data objects stored randomly on the mobile nodes in an ad hoc network. As can be observed, the semantically similar data contents (say, cars) are scattered on nodes not spatially in close vicinity. However, these nodes can be treated collectively as a cluster based on their semantic contents. Figure 22 illustrates the hierarchical indexing infrastructure obtained from content-based clustering and recursive abstraction of semantic contents.

Figure 21: Data distribution in an ad hoc network.

Figure 22: The indexing hierarchy based on clusters.
4.2.4 Self-Adaptive Adjustment

The proposed indexing hierarchy has the self-organizing capability as the network conditions changes. The cluster centroid nodes are selected according to the hardware characteristics of mobile nodes which are likely to remain static and independent of the network topology. After a period of time, some mobile nodes may break down, while some other nodes may try to join this network. The dynamic nature of the ad hoc network and the data contents of mobile nodes require dynamic maintenance of the proposed indexing scheme to guarantee the optimized and accurate data retrieval.

Because each cluster chooses the mobile node with the most powerful hardware and the largest data content as its centroid, the centroid nodes at top level of the indexing hierarchy should always be the most powerful one throughout the cluster. Suppose \( n' \) is the current centroid node. If as a result of content changes or hardware configuration changes node \( n'' \) becomes the most powerful one, then \( n'' \) should be the new centroid node.

Based on the Definition 9, an adaptive scheme that dynamically adjusts the centroid node of a cluster can be proposed. The key point in this scheme is that the current centroid node of the cluster needs to dynamically keep updated information about the hardware and data content changes of the other nodes in its cluster. The updated \( S(n_i) \) can be obtained by allowing query packets to piggyback it when forwarded to the centroid node. Since the piggybacking can take advantage of the wireless links established by the queries and be attached as the “tail” to the query messages, the updated \( S(n_i) \) can be delivered with very low overhead [115] — much less than the cost of resolving a query. The detailed update strategy is as follows:

- **Dynamic maintenance**: Each mobile node \( n_i \) can process a sequence of queries \( d_{q_1}, \ldots, d_{q_m} \). For any content-based query \( d_{q_i} \) submitted to node \( n_i \), if \( d_{q_i} \) cannot be resolved locally, it is forwarded to the centroid node of \( n_i \)’s cluster in an attempt to be resolved against the cluster-level content distribution information. In case of any changes to the \( S(n_i) \), after the last query forwarded to the centroid node, the \( S(n_i) \) will be attached with the query packet and forwarded to the centroid node. The overhead of piggybacking can be analyzed as follows: In an ad hoc network, more than 2/3 of the query resolution overhead is spent on the routing process [38]. The updated \( S(n_i) \) information, which has a smaller size than a query message (since it does not contain the message head and control bits), is attached as a “tail” to the query message \( d_{q_i} \) that is delivered to the centroid node of \( n_i \)’s cluster. Therefore, the amortized cost for piggybacking
updated \( S(n_i) \) for \( n_i \) is less than \((1/3) \times (1/m) = 1/3m\) of a message resolution cost. The new capacity entropy of \( n_i \) will be computed at the centroid node, and corresponding adjustments will be performed if a new largest entropy is found.

- **Node joining**: For a new node joining the ad hoc network, first an attempt is made to locate its content-related cluster and then its content is added to the indexing hierarchy.

- **Node leaving**: If a mobile node breaks down or leaves the network, its content summary will be removed from cluster centroid and ultimately from the indexing hierarchy.

### 4.3 Performance Analysis of the Proposed Clustering Scheme

#### 4.3.1 Theoretical Analysis

The performance of the proposed scheme is impacted by several factors as such; the initialization overhead of building the indexing hierarchy (including clustering and centroid selection), the cost of performing content-based multimedia queries, the overhead of maintaining the indexing hierarchy when network status changes. To facilitate content-based retrieval, these factors need to be analyzed in details.

Our analysis is based on the following notations:

- \( r \): the number of nodes in the ad hoc network.
- \( m \): the minimum number of children for an indexing node (minimum fan-out).
- \( P_J \): the probability of a node joining the network.
- \( P_R \): the probability of a node leaving the network.
- \( P_M \): the probability of a modify operation.
- \( Q \): the query rate.

#### 4.3.1.1 Initialization overhead

In the initialization step, a node with highest computational power and largest communication capability, say \( n_h \), is chosen as the coordinator to construct the indexing hierarchy. The selection of the coordinator node takes \( \theta(r) \) hops. The coordinator needs to send each node a message to notify the coordinator’s location and to collect data content descriptions, which takes \( O(r)\).\( r \) hops. In addition, the construction of the hierarchical infrastructure takes \( O(r) \) hops because the message cost is proportional to the number of edges (or links): the hierarchy has less depth than the binary search tree due to its minimum
m, thereby possessing edges less than $r+r/2+r/4+\ldots = 2r-1$. Hence, the initialization overhead amortized on each node is $\frac{\theta(r) + O(r)r + O(r)}{r} = O(r)$ hops, where $r$ is the number of nodes in the ad hoc network.

### 4.3.1.2 Search cost

Resolution of a query requires at most $2\log_m(r)$ logical steps to traverse the indexing hierarchy. Each forwarding operation takes at most $r$ hops. Hence the average searching cost for a query is $O(r \log_m(r))$ hops. In contrast, the flooding strategy requires $\Omega(r^2)$ hops to resolve a query, since a network of $k$ mobile nodes can have $\theta(r^2)$ connections, and a query may be transmitted on each connection at least once.

### 4.3.1.3 Maintenance overhead

As noted before, the proposed indexing hierarchy does not change as long as the data source contents are intact. As a result, the indexing hierarchy changes if a data source is inserted/deleted in/from the network (including the removal of the centroid node) or a modification is made to the contents of an existing node. As noted in section 4.2.4, content changes on each node is piggybacked with unresolved query and forwarded to the cluster centroid to potentially trigger maintenance overhead due to the reconstruction of the indexing hierarchy. Since the whole network is connected through the indexing hierarchy, the deletion or removal of any centroid node can be detected when a query cannot be resolved locally and forwarded to the centroid node. Therefore, the query resolution also provides a mechanism for checking the connectivity.

A new node $n_{r+1}$ joining the network requires at most $r \log_m(r)$ hops to be included in its related cluster: joining any cluster from the bottom level of the hierarchy takes $\log_m(r)$ hops, therefore comparing all possible clusters takes $r \log_m(r)$ hops. Hence, the average cost for processing new nodes is $(P_J) Q r \log_m(r)$ hops: with query rate $Q$, the node joining event happens $(P_J) Q$ time, each time takes $r \log_m(r)$ hops. Similarly, the processing of leaving nodes requires $(P_R) Q r \log_m(r)$ hops. The modification operation can be viewed as a deletion of a node followed by an insertion of a new node. Consequently, it requires $((P_M) Q r \log_m(r) + (P_M) Q r \log_m(r))$ hops. Finally, the average modification cost is $\theta ((2P_M+P_J+P_R) Q r \log_m(r))$.

### 4.3.2 Experimental Evaluation

In this section, we present the experimental analysis of the proposed content-based clustering model (called Extended SSM or **ESSM** in the following discussion) against the
flooding-based search scheme [37, 67, 71]. As noted earlier, in an ad hoc network with distributed multimedia data sources, flooding-based blind search strategy may cause extra computational and communication overheads due to the forwarding of the queries to every node. In contrast with the flooding-based schemes, the ESSM organizes nodes with similar contents into clusters, and forwards query packets only to the content-related nodes.

The evaluation consists of a series of experiments conducted using both real data set and synthetic data set. Our comparative analysis is based on various performance metrics such as accuracy, search cost, scalability, and physical characteristics of the mobile nodes.

4.3.2.1 Experimental setup

The experiments were run on the basis of ad hoc network prototype with CMU extension to the ns-2 version 2.26 [43]. Since ns-2 does not support content-based information retrieval, a semantic-representation module was developed and added to facilitate multimedia data organization. In addition, the routing and data transmission processes were implemented under the framework of extended summary-schemas hierarchy.

The experiment was initialized by assuming a default number of pre-existing nodes in the network with random connectivity among the nodes. A mixture of operations, including querying, node joining, and node leaving, were randomly generated and submitted to the network. The query generation time follows the exponential distribution, which is similar to the previous work [110]. The access pattern in the queries follows Zipf-like distribution, which is widely used to model non-uniformly distributed queries [111]. The input parameters are summarized in Table 7. Most of these parameters are self-explanatory. More details for some parameters are given below.

Node Movement Parameters: Each node in the experimental environment randomly selects waypoints within a 670m * 670m flat area. The node density can be adjusted by changing the number of mobile nodes in the range of 1 to 16,384 in the flat area. The node movement pattern follows the random waypoint movement model [112]: Initially, the nodes are placed randomly in the area. Each node selects a random destination and moves toward the destination with a speed selected randomly from \([0, v_{\text{max}}]\). After reaching its destination, the node pauses for a period of time and repeats this movement pattern.

Dataset Parameters: To examine both the accuracy and the scalability, we used two sets of experimental datasets as the test beds — the synthetic dataset and the real dataset as follows:
• The synthetic dataset employed in the experiments is similar to the one used in
[113], which includes up to 65,536 data points in a 256-dimension feature space
whose feature values are assigned by a random number generator abiding
normal distribution in the interval [0, 1] on each dimension.

• The real dataset we used consists of 2,560 images of 32 semantic categories
from the Corel dataset [84] (see Table 6). Each image in the dataset is
represented as a vector of 256 features (color histograms and texture wavelet
coefficients) and 4 annotation keywords. It is a large and heterogeneous image
dataset. 2,048 images in the test dataset were used to train a semantic subspace
learning module (i.e. LPP) integrated in the ESSM system that deduces the
relationships among the keywords and the semantic categories. The extra 512
images were used as the candidate pool for query examples.

**Table 7: Input parameters to the experimental system**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default value</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of nodes</td>
<td>128</td>
<td>1 to 16,384</td>
</tr>
<tr>
<td>Image dataset size</td>
<td>2,560</td>
<td>1 to 6,5536</td>
</tr>
<tr>
<td>Texture feature type-1</td>
<td>Daubechies wavelet coefficients</td>
<td></td>
</tr>
<tr>
<td>Texture feature type-2</td>
<td>Tamura coarseness histogram</td>
<td></td>
</tr>
<tr>
<td>Texture feature type-3</td>
<td>Pyramid wavelet coefficients</td>
<td></td>
</tr>
<tr>
<td>Color feature type-1</td>
<td>Color histogram in HSV space</td>
<td></td>
</tr>
<tr>
<td>Color feature type-2</td>
<td>Color coherence vector in LUV space</td>
<td></td>
</tr>
<tr>
<td>Color feature type-3</td>
<td>First and second moment in Lab space</td>
<td></td>
</tr>
<tr>
<td>Environment size</td>
<td>670m * 670m</td>
<td>100m² – 10,000m²</td>
</tr>
<tr>
<td>Transmitter range</td>
<td>100m</td>
<td>100m to 1,000m</td>
</tr>
<tr>
<td>Node mobility (v_{max})</td>
<td>1 m/s</td>
<td>1 to 50 m/s</td>
</tr>
<tr>
<td>Pause time</td>
<td>1s</td>
<td>1ms – 1s</td>
</tr>
<tr>
<td>Bandwidth</td>
<td>1M bps</td>
<td>1 – 10M bps</td>
</tr>
<tr>
<td>Query packet size</td>
<td>512 bytes</td>
<td>512 – 1,024 bytes</td>
</tr>
<tr>
<td>Query rate (Q_{rate})</td>
<td>10 query/s</td>
<td>1 – 100 query/s</td>
</tr>
</tbody>
</table>

**4.3.2.2 Retrieval process**

The content-based multimedia retrieval under the framework of ESSM can be
considered as a process of checking content distribution of the integrated information: the
query is compared with a series of content summaries navigated in the ESSM infrastructure
to find a cluster that contains the semantically most similar data objects.
In addition, the ESSM shows its adaptation to the ad hoc network environment in the experiments. As mentioned in section 4.1, the centralized search strategy achieves good performance; however, it is not suitable for ad hoc networks. Therefore, the ESSM can be shown efficient if it obtains comparable results as the centralized search strategy. Consequently, we used the result from a centralized search system as the standard for content-based image retrieval. The performance of ESSM can be evaluated by matching the percentage of its search result against the centralized search scheme. Similar as the metric used in [42], the matching percentage against centralized search reflects the accuracy of distributed content-based search performed by the ESSM. Figure 23 shows the 5-nearest-neighbor retrieval results of ESSM with different hop-count constraints. The environment comprises 2048 images distributed among 128 mobile multimedia databases. The query is an image of a horse (the leftmost image in the bottom row of Figure 23). As can be seen from Figure 23, ESSM finds the images that are very similar to the centralized search. In addition, limiting the number of hops (10-30 hops) reduces the accuracy of the search results — i.e., results are not identical to the centralized search, however, a close observation.
shows that the returned results are somehow semantically related to the query image — i.e., they are mainly mammals. This shows that unlike the conventional feature-based information retrieval systems ESSM retrieves multimedia data objects based on the gradually increasing granularities of semantic categories.

4.3.2.3 Retrieval performance evaluation

As noted earlier, the simulator is intended to evaluate the performance improvement of the proposed clustering scheme based on metrics such as search cost and accuracy:

- **Accuracy** is the percentage of the results generated by the search schemes (either the proposed clustering model or flooding) matching the results of centralized strategy. The higher matching percentage implies higher retrieval accuracy.

- **Search path length** is the average number of hops spent on locating the data source node that contains the semantically nearest neighbors of the query image.

- **Search cost** is the average number of messages incurred during the process of query resolution. Flooding-based schemes may have short search path length, but their search cost is high due to the duplicate query replies.

- **Maintenance cost** is the number of messages spent on adjusting the indexing hierarchy according to the topology and content changes in the network.

**Retrieval Accuracy**

The retrieval accuracy is evaluated for different simulation settings, varying the number of mobile nodes, the speed of node movement, the rate of query submissions, and the number of hops, during the query resolution process. In these experimental runs, we assumed 2,048 real images randomly scattered on mobile nodes. The maximum fan-out of the indexing hierarchy was set to 10, and the number of nearest neighbors in the content-based search was set to 10.

Figure 24 illustrates the impact of the number of hops on the retrieval accuracy for a network of 128 mobile nodes moving at maximum speed of 1m/s. Given a fixed setting of network scale (128 mobile nodes), the number of hops was varied from 10 to 80. As can be observed the retrieval results generated by the ESSM are more semantically related with the query image. Also note that both schemes (ESSM and flooding) achieves better accuracy as queries are generated at lower rate. This can be explained by the following fact: as query rate increases, more query packets are added into the network traffic and thereby causing
more packet loss, and more hops are required for resending and forwarding the queries. Therefore, the query results obtained from large hop counts are removed from the result list due to the limitation of hops in Figure 24. The superior performance of the ESSM in contrast with flooding stems from its content-based clustering capability, since the search domain is restricted to a few clusters that are semantically most relevant to the query.

![Figure 24: Impact of hop count (k = 10).](image1)

In a similar simulation run, we examined the impact of data density on the search accuracy. Using the same real image dataset, we varied the number of mobile nodes from 256 down to 32, increasing the average number of images per node from 8 to 64. With the fixed mobility and query rate (moving speed is limited to 1m/s and an average of 10 queries per second), the number of hops was limited to 64. Figure 25 depicts the experimental results. Since the probability of finding semantically related results increases as the data density increases, both schemes offer better accuracy as data density increases. However,
the accuracy of flooding-based schemes is still less than that of the ESSM. Also note that the ESSM achieves over 90% search accuracy at a relatively low data density (16 images per node). This implies that in a relatively large-scale network (2048/16 = 128 nodes), the ESSM almost achieves its peak performance with a small search cost (less than 64 hops). In the scope of wireless ad hoc networks, this is a significant observation since the mobile nodes usually have small storage and cannot support large data density.

**Search Cost**

In an ad hoc network, the search path length is usually calculated as the number of hops needed to deliver the query to proper mobile nodes that contain requested data. The real image dataset was used in this simulation run. Several factors, such as mobility, node density, and query rate, have direct impacts on the search cost; hence, we ran our simulator with different combinations of these parameters.

![Figure 26: Comparison on search path length (query rate = 10q/s, k = 10).](image)

![Figure 27: Comparison on search cost (query rate = 10q/s, k = 10).](image)
Figure 26 illustrates the number of hops needed during the query resolution for various network densities. As can be seen from Figure 26, the flooding scheme finds the data source node using less number of hops. This is due to the fact that flooding broadcasts the query to all its neighboring nodes and reaches the data source node using the shortest path. However, due to the broadcasting and duplicate replies, the search cost of flooding may be formidably high (see Figure 27).

Figure 27 shows the search cost of both schemes in the same environment. As anticipated the ESSM resolves a query much faster than flooding-based scheme due to its content-based clustering characteristic. In addition, from Figure 27 it can also be concluded that as node density increase, the ESSM offers even better performance than the traditional flooding schemes — steady performance improvement as the network scales up. The better performance of ESSM in comparison to flooding can be explained by the fact that larger node density will incur heavier flooding overhead, while ESSM can utilize the denser node distribution to form larger size clusters, which increases the probability of resolving a query within one cluster.

![Graph showing impact of mobility](image)

**Figure 28:** Impact of mobility (query rate = 10q/s, k = 10).

We also evaluated the impact of node mobility on search cost. For an ad hoc network of 64 nodes the maximum speed of mobile nodes varied from 10 to 50 m/s. Figure 28 shows the simulation results. As can be concluded, the search cost of both schemes increases as the node mobility increases, since increased mobility can cause more frequent breaks of the connectivity among mobile nodes, leading to higher cost in creating and maintaining the routes; however, the ESSM resolves queries at comparatively much less cost than flooding. This is due to the very nature of the flooding scheme — higher network traffic and higher workload at mobile nodes. In contrast, the ESSM resolves the queries
within the scope of content-related nodes, and avoids forwarding queries to irrelevant nodes.

**Scalability**

In a separate simulation run, the search cost of both schemes was evaluated as the network scales up. In this simulation run, we varied the number of nodes from 128 to 512 and randomly distributed 65,536 synthetic data points on the mobile nodes. Figure 29 illustrates the result. Similar to our earlier observation (Figure 27) one can conclude that the ESSM is scalable to large network sizes and large number of data objects.

![Figure 29: Search cost in large-scale networks.](query rate = 10q/s, k = 10)

![Figure 30: Search cost and maintenance cost](65,536 data points, mobility = 1m/s, query rate = 10q/s, k = 10).
Maintenance Overhead

As mentioned in section 4.3.1.3, self-adjustment capability of the proposed model incurs maintenance overhead that needs to be evaluated. In contrast to the ESSM, the maintenance cost of flooding scheme is very limited to the messages needed to update the neighborhood relationships between mobile nodes. Figure 30 shows the average search cost and the maintenance cost of both strategies as the network becomes denser. As one can conclude, even taking the maintenance overhead into account, the ESSM still offers a better overall performance than flooding.

4.4 Summary of Chapter Contents

The proposed SSM-based clustering scheme (ESSM) makes use of the data content relationship in ad hoc networks to reduce the search cost without incurring high maintenance overhead. We have quantified the efficiency and effectiveness of our scheme with respect to various performance metrics — retrieval accuracy, search cost, and maintenance overhead. Through extensive theoretical and experimental analysis, we found that the clustering scheme has the following features:

- The ESSM is a decentralized non-flooding search strategy performing content-based multimedia retrieval in ad hoc networks. As shown in our simulation results, it can achieve high accuracy while visiting only a small portion of mobile nodes.

- We employed semantic-based clustering to organize multimedia data — the content-related mobile nodes are grouped into clusters. As witnessed by simulation results, this approach reduces the search cost drastically relative to the traditional flooding strategy.

- The ESSM scheme is dynamic and capable of self-organization as the network status changes. This further offers scalability and robustness in large-scale networks.

In ad hoc networks, the performance of content-based retrieval is often affected by the utilization of semantic locality of data objects. Caching is an effective approach for utilizing semantic locality. In Chapter 5, we will introduce the semantic-aware image caching (SAIC) scheme that employs the cooperative caching among mobile nodes to reduce search cost.
5 SEMANTIC-AWARE MOBILE CACHING

Caching is a technique to profile content distribution and improve system performance in the mobile environments. However, the traditional caching technique proposed for text data cannot be directly applied to multimedia information retrieval: First, traditional caching methodology relies on exact match and therefore is not suitable for approximate and similarity-based queries. Second, the description of cached data is defined based on the query context instead of data content, which leads to inefficient use of cache storage. Third, the description of cached data does not reflect the popularity of the data, making it inefficient in providing QoS-related services.

In this chapter, a mobile image caching scheme will be proposed and analyzed. The caching scheme, namely Semantic-Aware Image Caching (SAIC), is inspired by the SSM and explores the semantics of mobile image data sources to facilitate content-based retrieval. We introduce the major principles of the caching scheme and evaluate it using extensive simulations.

5.1 Introduction and Motivation

In the domain of ad hoc networks, most of the previous study of caching focused on the efficient exploration of routing information [38] with only a few caching schemes (data caching and path caching) to address the data retrieval issue [69].

The data caching in ad hoc networks is a natural extension of the caching schemes in wired networks — to keep a copy of the data items that have recently been accessed. Traditional schemes let a mobile node cache either the results of its recent queries or the data that have been forwarded though it to other nodes [69]. The data caching scheme proposed in [79] allows the caching of queries at the semantic descriptions of the cached data. Such a caching scheme is efficient only for small-size data items, and cannot effectively deal with large-size data such as images in mobile databases.

Path caching is another application of caching in ad hoc networks — to record a path to the data source. The scope of path caching was further extended to the domain of data replica allocation in [80]. The CachePath scheme proposed in [69] dynamically caches the path information of passing-by data.

These schemes, in general, consider the data items as independent entities and fail to utilize the semantic locality among them. As a result, they do not explore content distribution in the ad hoc network.
Motivated by the challenges of caching in ad hoc networks and the shortcomings of the existing schemes, we applied summary schemas methodology into the caching strategy. Our proposed SSM application in semantic caching is based on the following observations: First, a mobile node may have specific interest and query some multimedia data objects more frequently. Second, the data contents of a node are often semantically similar to the queries it has issued. Therefore, by analyzing the earlier queries delivered among the nodes, the system may forward later queries with similar semantics to a small collection of nodes which requested or resolved the earlier queries. Third, some nodes in the network may share similar interest and generate the same queries. These nodes can be grouped into common-interest clusters, in which any query can be cached and analyzed for the purpose of facilitating later queries issued by other nodes. Therefore, we propose to exploit the query history of mobile nodes to find the relationships among data objects, use this knowledge to determine the data contents of mobile nodes, and ultimately improve the performance of content-based multimedia information retrieval.

5.2 Design Goal

Our goal is to design a content-based retrieval scheme that employ data content distribution knowledge to optimize the search cost, response time, and system overhead. Although these metrics are sometimes contradicting with each other, e.g., the relatively convenient content/location mapping of dynamic hash table at the price of formidable maintenance overhead, we propose a novel solution that leverage these metrics. Such a solution should satisfy the following:

1) **Accuracy**: In wireless communication environments, it is difficult to perform content-based retrieval through traversing the whole network, since some nodes may be out of reach due to mobility, disconnection, or power limitation. Therefore, one major consideration is how to guarantee comparable accuracy without traversing the whole network.

2) **Efficiency**: Since we are targeting at the resource-constrained mobile ad hoc networks, the solution should be efficient in terms of communication, computation, and storage utilization.

3) **On-demand**: The system should be able to allow queries on demand anywhere and anytime, as long as the network connectivity and data availability are satisfied. The mobile nodes in an ad hoc network do not need to know the query types and contents beforehand, nor should they periodically exchange their meta data.
4) **Overlay-free**: Due to the mobility and dynamic topology, it is theoretically complex and computationally inefficient to maintain an indexing hierarchy overlaid on the ad hoc network. In addition, all nodes should be treated as equal as possible, avoiding giving some nodes too much responsibility and privilege. The proposed scheme should be able to provide low-weight retrieval mechanisms without the help of indices or hash tables.

5) **Non-flooding**: As one of the most importance communication patterns in ad hoc network, flooding can obtain query results using relatively small response time, at the price of drastic consumption of system resources. The scheme should use content distribution knowledge to find the requested image data, avoiding the resource-consuming flooding mechanism.

6) **Self-adaptive**: The proposed scheme should be self-organized and adaptable as the network condition changes.

### 5.3 Mathematical Basis of Semantic-Aware Caching

In order to reach the expected goals, we developed the mathematical basis of the semantic-aware caching scheme. In this section, we first outline the general overview of content distribution representation, then describe the approaches for content estimation and profiling of mobile nodes. The aim of theoretical analysis is to provide a foundation for efficient cache utilization and effective query resolution. The proposed caching scheme depends on semantic-aware data preprocessing in three issues: 1) the classification of data contents in the semantic space; 2) the estimation of mobile node data contents based on query history; and 3) the profiling of mobile nodes based on content estimation.

The data content classification, as the name suggests, is the process of categorizing the image data objects according to quantitative information on feature characteristics inherent in the objects and setting up the relationships among the categories based on the data semantics. The purpose of classification is to provide a method of describing images not as individual entities but as groups of semantically similar data objects. With the help of classification, we can not only define the similarity between images but also describe a collection of images using concise terms.

The data content classification is a process to categorize the image data objects and set up the relationships among the categories based on semantics. The purpose of classification is to provide a method of describing images not as individual entities but as groups of semantically similar data objects. With the help of classification, we can not only
define the similarity between images but also describe a collection of images using concise terms.

In an ad hoc network, the data content distribution may not be known to each node. Although the semantic-based representation can be obtained through classification, it is difficult to profile the data content of a mobile node based on query results because the observed query results may only show part of the node’s content. To solve this problem, two estimation approaches are introduced in this section: The association-based estimation approach makes use of semantic locality of data objects to estimate the content distribution. The Bayesian-based content estimation employs Bayes’ rule to calculate the conditional probability of data objects.

Based on the estimated content distribution knowledge, the profiling of mobile nodes can be obtained to facilitate semantic-aware caching. To achieve more accurate profiling, we propose to use the integration vicinity constraints and semantic category information in the representation of data objects. This representation method, when used in the semantic-aware caching scheme, gives a concise and effective description of cache contents which facilitates the resolution of queries.

### 5.3.1 Data Content Classification

In the context of distributed data sources in an ad hoc network, the cost of k-NN retrieval is formidably high due to the necessity of traversing the whole network. Note that semantically similar data objects are densely located clusters in the semantic space (i.e. semantic categories). As a result, the cost of k-NN retrieval could be reduced through restricting the search region within a semantic category.

**Definition 12: The Semantic Category**

An \( n \)-dimensional semantic space \( \mathbb{R}^n \) can be partitioned into a collection of orthogonal regions, which are referred to as semantic categories \( \mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_r \).

As mentioned in section 2.3.1, the semantic categories can be deducted from a training sample, formally denoted as \( \tau = \{ \overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n \} \), which satisfies \( \tau = \bigcup_{i=1}^{r} \mathcal{E}_i \). Let \( \delta(\mathcal{E}_i, \tau) \) represent the sample data points in category \( \mathcal{E}_i \), then the corresponding region of \( \mathcal{E}_i \) in the semantic space \( \mathbb{R}^n \), denoted as \( \zeta(\mathcal{E}_i) \), can be viewed as the locus of points whose semantic distance to the data points in \( \delta(\mathcal{E}_i, \tau) \) is smaller than to those in any other semantic category.
Given a multimedia data object $x_i$ and a semantic category $\mathcal{C}_j$, if $x_i$ belongs to $\mathcal{C}_j$, their relationship is denoted as $x_i \in \mathcal{C}(\mathcal{C}_i)$.

**Definition 13: Inner-Category Similarity Search**

Unlike the global $k$-NN in Definition 3, the inner-category $k$-NN, denoted as $k$-NN$_c$, is the top $k$ nearest neighbors within the given semantic category. Given a data object $x_i$ and a semantic category $\mathcal{C}_j$, the inner-category $k$-NN of $x_i$ within category $\mathcal{C}_j$ is a set:

$$k\text{-NN}_c(x_i, \mathcal{C}_j) = \{x_k \mid \forall y \notin k\text{-NN}_c(x_i, \mathcal{C}_j), \text{dist}(y, x_i) \geq \text{dist}(x_k, x_i) \land x_k \in \mathcal{C}(\mathcal{C}_i)\}$$ (16)

Generally, due to the limitation of the semantic category, $k$-NN$_c(x_i, \mathcal{C}_j)$ may only return less than $k$ similar data objects that belong to the semantic category $\mathcal{C}_j$. In contrast to the normal similarity search (Definition 3), we have the following claim:

**Claim 1:** If $x_q \in \mathcal{C}_j$ and $|k$-NN$_c(x_q, \mathcal{C}_j)| = k$, then $k$-NN$_c(x_q, \mathcal{C}_j)$ contains exactly the same data points as $k$-NN($x_q, \mathcal{I}$).

**Proof:** The data points in $k$-NN$_c(x_q, \mathcal{C}_j)$ are semantically closest to $x_q$, therefore $k$-NN$_c(x_q, \mathcal{C}_j) \subseteq k$-NN($x_q, \mathcal{I}$). (a)

Moreover, since in the assumption of Claim 1 we know that $|k$-NN$_c(x_q, \mathcal{C}_j)| = |k$-NN($x_q, \mathcal{I}$)| = $k$, and for any other category $\mathcal{C}_k$, the distance between $x_q$ and data points in $\delta(\mathcal{C}_j, \mathcal{I})$ is smaller than those in $\delta(\mathcal{C}_k, \mathcal{I})$, therefore $k$-NN($x_q, \mathcal{I}) \subseteq k$-NN$_c(x_q, \mathcal{C}_j)$. (b)

from (a) and (b) we conclude $k$-NN$_c(x_q, \mathcal{C}_j) = k$-NN($x_q, \mathcal{I}$). ■

Based on Claim 1, we may perform content-based retrieval (CBIR) in an alternative approach: Given an image $x_q$, first find its semantically most related category $\mathcal{C}_j$, and perform inner-category $k$-NN in $\mathcal{C}_j$. This alternative approach returns the same results as the classical CBIR methods but with possibly much less search cost, because the search process is restricted within the region of a semantic category.

A hierarchical representation model is used to reduce the search space to a subset of categories when $|k$-NN$_c(x_q, \mathcal{C}_j)| \neq k$. The main idea of the hierarchical representation model is based on the observation that $k$-NN retrieval may involve images from several basic semantic categories, which form a more generic semantic category with some common semantic characteristics. The interrelationship between semantic categories is defined as follows.
Figure 31: Multi-level semantic hierarchy.

Given a set of orthogonal semantic categories $\Omega = \{\epsilon_1, \epsilon_2, \ldots, \epsilon_t\}$ and an on-line thesaurus $\psi$, based on the hypernym/hyponym relationships given in section 2.5.2, we can consider the semantic hierarchy (i.e. the SSM hierarchy) as a Hasse Diagram $H_S(\Omega, \psi)$. As shown in Figure 31, a semantic category may have super category and sub category in the hierarchy, and their relationships are defined as hypernym/hyponyms.

**Claim 2:** Given the set of semantic categories $\Omega = \{\epsilon_1, \epsilon_2, \ldots, \epsilon_t\}$, a multimedia data object set $I = \{x_1, x_2, \ldots, x_m\}$, and a query object $x_q$, the content-based retrieval $k$-NN($x_q, I$) can be viewed as follows:

- $\exists$ A $\Omega^* = \{\epsilon_1^*, \epsilon_2^*, \ldots, \epsilon_s^*\} \subset \Omega$ such that $\bigcup_{i=1}^s k$-NN$_c(x_q, \epsilon_i) = k$-NN($x_q, I$). (i.e. the global $k$-NN retrieval in semantic space can be decomposed into a set of inner-category retrievals in semantic categories)

- $\exists$ A semantic category $\epsilon'$ in $H_S(\Omega, \psi)$ that satisfies $|\epsilon'| \leq \bigcup_{i=1}^s |\epsilon_i|$ and $k$-NN$_c(x_q, \epsilon') = k$-NN($x_q, I$). (i.e. there exist a “super category” $\epsilon'$ that encloses all the similar data objects returned by the global $k$-NN)

**Proof:**

1. Based on Claim 1, $k$-NN($x_q, I$) can be performed as a series of $k$-NN$_c(x_q, \epsilon_i)$ in several semantic categories, until the sum of returned data objects is $k$. 

107
2. The semantic hierarchy $H_S(\Omega, \psi)$ represents $\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_t$ using more generic concepts, hence there exists a semantic category $\mathcal{E}'$ in $H_S(\Omega, \psi)$ satisfying $\bigcup_{i=1}^{s} \mathcal{E}_i \subseteq \mathcal{E}'$.

The semantic hierarchy provides a means of partitioning the semantic space $\mathbb{R}^n$ into sub regions. The space $\mathbb{R}^n$ is first partitioned into regions $r_1, r_2, \ldots, r_t$, corresponding to the basic semantic categories $\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_t$. The regions are merged together based on the hypernym/hyponym relationships. Finally, a multi-level partitioning of $\mathbb{R}^n$ is constructed, where each partition corresponds to a category in the semantic hierarchy.

### 5.3.2 Profiling Semantic Contents

The aforementioned multi-level semantic content representation is based on the following assumption: the content distribution over mobile nodes is already known to the information retrieval system. However, in a wireless ad hoc network, this assumption cannot be guaranteed because of the mobility and frequent disconnections. Therefore, we propose to profile the semantic contents of mobile nodes through statistical analysis of cached queries. In this chapter, we will discuss two content analysis approaches — association-based and Bayes-based estimations, which provide the foundation for describing the mobile content distribution.

The rationale of content analysis is based on the semantic locality of image data objects. As mentioned in Definition 12, multimedia data objects can be described as semantic categories, each category containing a set of data objects with similar semantic contents. In a content-based retrieval on node $n_j$, for a given query $x_q$ and its $k$-NN query results $\{x^*_1, \ldots, x^*_k\}$, it is likely that $n_j$ also contains other data objects related with $x^*_1, \ldots, x^*_k$. Therefore when $x^*_1, \ldots, x^*_k$ are used in later queries, node $n_j$ is also the proper data source that can resolve such queries.

The content analysis can be formalized as follows: For a given collection of queries $Q = \{x_{q1}, x_{q2}, \ldots, x_{qw}\}$ issued by a node $n_i$, suppose the queries are resolved at node $n_j$ and the set of query result data objects is $\chi(Q, n_j) = \{x_{j1}, x_{j2}, \ldots, x_{ju}\}$. One can estimate another set of data objects $\mu(\chi(Q, n_j), \tau) = \{x^*_{j1}, x^*_{j2}, \ldots, x^*_{ju}\}$ with confidence more than a predefined threshold $\tau$, and consider the data content of node $n_j$ as $\chi(Q, n_j) \cup \mu(\chi(Q, n_j), \tau)$.

Figure 32 gives an illustrative example about the rationale of content estimation. Given a query $x_q$, the information retrieval system performs 3-NN search in the semantic
space and finds the three most similar data objects $x_1$, $x_2$, and $x_3$ as the query result. However, the mobile node $n_j$ returning $x_1$, $x_2$, and $x_3$ may also contain other content-similar data objects that are not shown in the query result of $x_q$. For instance, if one uses $x_1$ and $x_3$ for content-based retrieval, the system may also find data objects $x_4$ and $x_5$ as related data objects. Therefore, to profile the content of node $n_j$, $\{x_1, x_2, x_3\} \cup \{x_4, x_5\}$ is a better estimation than $\{x_1, x_2, x_3\}$.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{content_estimation.png}
\caption{An illustrative example of content estimation.}
\end{figure}

In the association-based content estimation, the estimation is obtained through association rules. Let the query history $L = \{\ldots, x_{qi}, \ldots\}$ be a complete list of earlier queries where each query $x_{qi}$ is a data object in $I$. We make this assumption because in content-based query processing, each query is a sample data object. We also assume query result history $T = \{\ldots, k$-NN$(x_{qi}, I), \ldots\}$ be a complete list of earlier query results (i.e. nearest neighbors). Based on these assumptions, let $x_i$ and $x_j$ be two data objects satisfying: (1) $x_i \in L$, (2) $x_j \in k$-NN$(x_i, I)$, (3) $k$-NN$(x_i, I) \in T$, and (4) $x_i \neq x_j$. Then an association rule is a repeatedly happened pattern $r$: $x_i \rightarrow x_j$, where $x_i$ and $x_j$ are called the antecedent and consequence of the association rule $r$, respectively.

**Definition 14: The Query Popularity**

Given the query history $L$ and a query $x_q$, the query popularity is the appearance frequency of $x_q$ in $L$:

$$\psi(x_q, L) = \text{appearance}(x_q)/|L|, \text{ for } \forall x_q \in L. \quad (17)$$
where $appearance(x_q)$ is the number of $x_q$’s appearances in $L$, and $|L|$ is the number of queries in $L$.

In content-based multimedia information retrieval, each query is also a data object in $I$. Therefore, $L$ is a list of repeated appearances of data objects from $I$. The higher appearance of a data object indicates its higher popularity. If an object is frequently used as queries, and some other data objects are often returned as query results, then there is a semantic locality between these data objects and can be used for content estimation.

Due to the data replications in mobile data source nodes, the query result of $k$-NN($x_q$) may contain one or more $x_q$’s replicas. However, in the generation of association rules, these replicas should not be considered as consequences of $x_q$. Therefore, the association rules are defined as $r: x_q \rightarrow x_i$, where $x_i \in k$-NN($x_q$) – {$x_q$}.

**Definition 15: The Association Rule Set**

Given a query history $L$ and a popularity threshold $\tau$, the association rule set $R(L, \tau)$ is the set of association rules having query popularity no smaller than $\tau$:

$$R(L, \tau) = \{ r: x_q \rightarrow x_i \mid \psi(x_q, L) \geq \tau \}$$ (18)

**Definition 16: The Estimated Node Content**

Given a mobile node $n_j$ and a set of queries $Q = \{x_{q1}, x_{q2}, \ldots, x_{qw}\}$ that are resolved by data objects from $n_j$, the estimated node content of $n_j$ can be defined as follows:

$$\xi(n_j) = \{ x_k \mid r: x_p \rightarrow x_k, \text{ for } \forall x_{qi} \in Q \land x_p \in k$-NN($x_{qi}$) – {$x_{qi}$} \land r \in R(L, \tau) \}$$ (19)

In Definition 16, the associate rule set is denoted as $R(L, \tau)$. We use $R(L, \tau)$ instead of $R(Q, \tau)$ because the associate rules are not only generated from the queries resolved at node $n_j$. For any query $x_{qi}$ in $Q$, we can obtain $x_p$ as one of the nearest neighbors of it. Therefore, we are confident that data object $x_p$ belongs to the content of node $n_j$. At the same time, by using the associate rule $r: x_p \rightarrow x_k$, we can also estimate that data object $x_k$ is within the data content of node $n_j$. Thus the data content of node $n_j$ can be estimated with confidence $\tau$.

**Definition 17: The Estimated Node Interest**

Given a mobile node $n_j$ and a set of queries $Q^* = \{x_{q1}^*, x_{q2}^*, \ldots, x_{qs}^*\}$ that are issued by $n_j$, the estimated node interest of $n_j$ can be defined as follows:

$$\lambda(n_j) = \{ x_i \mid r: x_q^* \rightarrow x_i, \text{ for } \forall x_q^* \in Q^* \land r \in R(L, \tau) \}$$ (20)
Similar to the Definition 16, we can use the associate rules and the earlier queries issued by node $n_j$ to estimate its interest. Since we already know that the queries in $Q^*$ are objects of interest to $n_j$, we can estimate that the related objects deducted from associate rules are also of interest to $n_j$.

**Definition 18: The Interest-Content Overlap**

Given a mobile node $n_i$ and the query history $L$, the node set that contains data contents related with the interest of node $n_i$ is considered as the interest node set:

$$\theta(n_i) = \{ n_k | \xi(n_k) \cap \lambda(n_i) \neq \phi \}$$  \hspace{1cm} (21)

The estimation of node contents and interests can be used for determining the relevant data source nodes when resolving queries. For a new query $x_q$, the ad hoc network can be partitioned into two sets of data source nodes — the “relevant” nodes whose estimated data contents or interests include $x_q$, and the “irrelevant” nodes that do not have such an inclusion. To reduce the cost of content-based image retrieval, the system needs to avoid forwarding the query $x_q$ to irrelevant nodes.

The content estimation can also be performed with a Bayesian probability based approach. Given a mobile node $n_j$, its relevance to multimedia data objects can be iteratively modified according to the Bayesian probabilities. Initially, all data objects in $I = \{x_1, x_2, \ldots, x_m\}$ have the same relevance probability. The query resolutions change the probabilities of data objects, and give a general profile of $n_j$’s data content: the objects with high probabilities are more relevant with node $n_j$.

Formally, assume that for each data object $x_i$ there exists some underlying probability distribution $P(.|x_i)$, which is referred as the fundamental relevance probability. Such initial probability model can be constructed using existing semantic analysis systems, e.g. Latent Semantic Analysis (LSA) [12]. The data objects will then be retrieved with a collection of queries $Q = \{x_{q1}, x_{q2}, \ldots, x_{qw}\}$, which can be considered as a series of query resolution iterations — each iteration containing the query results and their relevance to node $n_j$. In a specific iteration $t_k$, the relevance probabilities of data objects can be calculated based on the conditional probability $P(x = x_i) \forall x_i \in k-\text{NN}(x_{qk})$.

**Definition 19: The Bayes’ Rule**

Based on above descriptions of content estimation, the Bayes’ rule can be defined as:
\[ P(x \in k-\text{NN}(x_{qk})\mid t_1 \ldots t_k) \]
\[ = \frac{P(t_1 \ldots t_k \mid x \in k - \text{NN}(x_{qk})) P(x \in k - \text{NN}(x_{qk}))}{P(t_1 \ldots t_k)} \]
\[ = \frac{P(t_1 \ldots t_k \mid x \in k - \text{NN}(x_{qk})) P(x \in k - \text{NN}(x_{qk}))}{\sum_{i=1}^{m} P(t_1 \ldots t_k \mid x = x_i) P(x = x_i)} \]
\[ = \sum_{j=1}^{m} \sum_{v=0}^{k-1} P(t_v \mid t_{v+1}) P(t_k \mid x \in k - \text{NN}(x_{qk})) P(x \in k - \text{NN}(x_{qk})) \]
\[ \sum_{j=1}^{m} \prod_{v=0}^{k-1} P(t_v \mid t_{v+1}) P(t_k \mid x = x_j) \]  

where \( P(t_0) = 1. \)

### 5.3.3 Algorithms for Content Estimation

- **Frequent Pattern Based Estimation**

In the following discussion, we present the algorithms for computing the association rule set and the interest node sets based on the analysis of the query history and node data contents. The fundamental idea is to first allow the mobile nodes to record the queries and their results passing by it as the query history, and then gradually increase the size of association set based on the appearance ratio of image data objects in the queries (i.e. antecedents) and results (i.e. consequences). Table 8 shows the notations used in the association rule set generation algorithm. Algorithm 7 shows the details of the rule generation process.

**Table 8:** Notations related to association rule generation.

<table>
<thead>
<tr>
<th>Items</th>
<th>Notations</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{qry}(L) )</td>
<td>The function of selecting a query from ( L )</td>
<td>( \text{qry}(L) )</td>
</tr>
<tr>
<td>( \text{rslt}(x_q) )</td>
<td>The function of obtaining the result of query ( x_q )</td>
<td>( \text{rslt}(x_q) )</td>
</tr>
<tr>
<td>( \text{popu}(x_q) )</td>
<td>The function of computing the popularity of ( x_q )</td>
<td>( \text{popu}(x_q) )</td>
</tr>
</tbody>
</table>
**Algorithm 7: Constructing association rule set**

*Input:* Query history $L$ and popularity threshold $\tau$

*Output:* Association rule set $R(L, \tau)$

1) $R(L, \tau) \leftarrow \emptyset$

2) for $i \leftarrow 1$ to $|L|$

3) $x_q \leftarrow qry(L)$

4) $cons \leftarrow popu(rslt(x_q) - \{x_q\})$

5) $ante \leftarrow popu(x_q)$

6) if ($cons/ante > \tau$)

7) $R(L, \tau) \leftarrow R(L, \tau) \cup \{ r: x_q \rightarrow rslt(x_q) - \{x_q\} \}$

8) return $R(L, \tau)$

The algorithm accepts the query history and a minimum popularity threshold as the input. It computes the popularity values of antecedents and consequences, and compares their ratio with the threshold $\tau$. By this statistical analysis, some repeatedly happened associations between data objects are discovered and kept as association rules.

We use Algorithm 8 to obtain the node set whose data contents are of interest of node $n_i$. The notations and algorithm for computing interest node set are listed in Table 9.

**Table 9:** Notations related to interest node set generation.

<table>
<thead>
<tr>
<th>Items</th>
<th>Notations</th>
</tr>
</thead>
<tbody>
<tr>
<td>$inst(n_i)$</td>
<td>The interest node set of $n_i$</td>
</tr>
<tr>
<td>$Hq(n_i)$</td>
<td>The history of queries issued by node $n_i$</td>
</tr>
<tr>
<td>$\rho(n_i)$</td>
<td>The function of obtaining a query issued by $n_i$</td>
</tr>
<tr>
<td>$node(x_q)$</td>
<td>The function of finding the nodes resolving $x_q$</td>
</tr>
<tr>
<td>$est(n_k)$</td>
<td>The function of estimating the content of node $n_k$</td>
</tr>
<tr>
<td>$\xi inst(n_i)$</td>
<td>The estimated interest of node $n_i$</td>
</tr>
</tbody>
</table>
Algorithm 8: Generation of interest node set

**Input:**
- Mobile node $n_i$
- Association rule set $R(L, \tau)$

**Output:**
- Interest node set of $n_i$

1) $\text{inst}(n_i) \leftarrow \emptyset$
2) while ($Hq(n_i) \neq \emptyset$)
3) $x_q \leftarrow \rho(n_i)$
4) $Hq(n_i) \leftarrow Hq(n_i) - \{x_q\}$
5) for $i \leftarrow 1$ to $|\text{node}(x_q)|$
6) select a node $n_k$ from $\text{node}(x_q)$
7) if ($\text{est}(n_k) \cap \xi_{\text{ist}}(n_i) \neq \emptyset$)
8) $\text{inst}(n_i) \leftarrow \text{inst}(n_i) \cup \{n_k\}$
9) return $\text{inst}(n_i)$

The proposed algorithm generates the interest node set for a mobile node $n_i$ based on its query history and the association rules. For the simplicity of computation, the algorithm employs the estimated interest of the input node to filter out the nodes that do not contain overlapping image data contents. Therefore, a node $n_i$ only shares its interest with a small collection of other nodes. When a new query $x_q$ is issued, $n_i$ will first forward it to the interest related node set, with the purpose of resolving the query with a small portion of the network. Due to the semantic locality of multimedia data, most queries will be processed in the small set of nodes and the search cost is minimized.

- **Probability Based Profiling**

The Bayesian-based content estimation is a process of estimating the probability of the user’s next query being satisfied by the data content of a mobile node according to the observation of earlier query resolutions. In Bayesian-based estimation, the query history is first divided into a series of iterations $t_1, \ldots, t_k$, each iteration containing a collection of queries and the related results. To determine the content of a specific node $n_i$, we assign each data object $x_j$ a value indicating the probability of $x_j$ existing in the database of node $n_i$. Initially, all data objects have the same probability. As the queries in iterations $t_1, \ldots, t_k$
be resolved, the probabilities change after each iteration according to Bayes’ rule. The detailed process is described in Algorithm 9.

**Table 10:** Notations related Bayesian content estimation.

<table>
<thead>
<tr>
<th>Items</th>
<th>Notations</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P(x_j \in n_i)$</td>
<td>The probability of data object $x_j$ existing in $n_i$</td>
</tr>
<tr>
<td>$P(I</td>
<td>n_i, t_k)$</td>
</tr>
<tr>
<td>query($t_k$)</td>
<td>The function of taking a query from iteration $t_k$</td>
</tr>
<tr>
<td>satisfy($x_q, n_i$)</td>
<td>Whether the query $x_q$ is satisfied at node $n_i$</td>
</tr>
</tbody>
</table>

**Algorithm 9: Bayesian content estimation**

*Input:* Query iterations $t_k$

- Data content mapping $P(I|n_i, t_{k-1})$ before $t_k$

*Output:* Data content mapping $P(I|n_i, t_k)$ after $t_k$

1) for $i \leftarrow 1$ to $|t_k|$
2) $x_q \leftarrow$ query($t_k$)
3) if (satisfy($x_q, n_i$))
4) $P(I|n_i, t_k) \leftarrow P(I|n_i, t_{k-1})P(x_p \in n_i, \forall x_p \in k-NN(x_q))$
5) else
6) $P(I|n_i, t_k) \leftarrow P(I|n_i, t_{k-1})(1 - P(x_p \in n_i, \forall x_p \in k-NN(x_q)))$
7) return $P(I|n_i, t_k)$

The algorithm accepts the query history before iteration $t_k$ and uses the query history for the computation of a new data content mapping $P(I|n_i, t_k)$. Using the Bayesian rule, $P(I|n_i, t_k)$ can be obtained as the conditional probability of the appearance of data objects in query($t_k$).

### 5.3.4 Profiling Data Contents of Mobile Nodes

As mentioned in section 4.2.1, the data objects $I = \{x_1, x_2, \ldots, x_m\}$ are disseminated among a collection of mobile nodes $n_1, n_2, \ldots, n_r$. Let $\xi(n_i)$ denote the estimated data
contents in node $n_i$, as a result, $\bigcup_{i=1}^{r} \xi_i(n_i) = I$. In practical systems, a mobile node may possess data objects of different semantic categories, and multiple copies of a data object may exist in multiple nodes (i.e. replications). Therefore, the distribution pattern of data objects over the mobile nodes can be considered as a many-to-many relationship, which is illustrated in Figure 33.

**Figure 33:** The distribution of data objects in mobile nodes.

Figure 33 illustrates an ad hoc network and the data objects distributed among the mobile nodes. The data objects, represented as small grey dots, are considered as data points in the semantic space. Each mobile node $n_i$ in the network may contain a set of data objects, denoted as $D(n_i) = \{x_{i1}, x_{i2}, \ldots, x_{ir}\}$, from the semantic space. As can be noted from Figure 33, in this semantic space, mobile nodes may contain overlapping data (i.e. replications). For example, both node $n_1$ and node $n_2$ contain the data object $x_2$. Generally, if an object $x_j$ is distributed over a collection of mobile nodes, we use $\beta(x_j)$ to denote the set of mobile nodes containing the replicas of $x_j$.

Because of the replication and semantic similarity of node contents, a semantic category $\mathcal{E}_i$ may cover a collection of mobile nodes. Let $\mathcal{N}^i(\mathcal{E}_i)$ denote the nodes covered by semantic category $\mathcal{E}_i$, one can make the following observations:
• \( \bigcup_{i=1}^{t} N_c(\epsilon_i) = N \)

• \( \bigcup_{i=1}^{t} \zeta(N_c(\epsilon_i)) = I \)

• Given two categories \( \epsilon_i \) and \( \epsilon_j \), we get \( k-NN_c(x_q, \epsilon_i \cap \epsilon_j) = k-NN(x_q, \zeta(N_c(\epsilon_i) \cap N_c(\epsilon_j))) \)

Based on these observations and Definition 12 of semantic categories, we propose to represent the data objects in \( \zeta(n_k) \) using the combination of semantic categories and the minimum bounding region of the objects. Here we define the concept of vicinity constraint.

**Definition 20: The Vicinity Constraint**

Given a set of data objects \( I^* = \{x_{i1}, x_{i2}, \ldots, x_{ih}\} \subset I \), each object \( x_{ij} \) can be represented as a vector of semantic attributes \( \phi_{x_{ij}} = (\omega_1^{x_{ij}}, \omega_2^{x_{ij}}, \ldots, \omega_n^{x_{ij}}) \). The vicinity constraint \( C^*(I^*) \) is the constraint showing the \( n \)-dimensional minimum bounding region of \( x_{i1}, x_{i2}, \ldots, x_{ih} \):

\[
C^*(I^*) = ([\min\{\omega_1^{x_{i1}}, \ldots, \omega_1^{x_{ih}}\}, \max\{\omega_1^{x_{i1}}, \ldots, \omega_1^{x_{ih}}\}], \ldots, \ [\min\{\omega_n^{x_{i1}}, \ldots, \omega_n^{x_{ih}}\}, \max\{\omega_n^{x_{i1}}, \ldots, \omega_n^{x_{ih}}\}])
\] (23)

Similar to minimum bounding rectangle that shows the region of 2-dimensional data points, the vicinity constraint is used to show the region of data objects in the \( n \)-dimensional space. The integration of vicinity constraint and semantic categories can give a more accurate description of a collection of data objects in the semantic space.

**Definition 21: The Node Content Descriptor**

Given a node \( n_k \) and a semantic category \( \epsilon_i \), if every estimated object \( x_j \) in \( \zeta(n_k) \) satisfies \( x_j \in \zeta(\epsilon_i) \), then the node content descriptor \( \eta(n_k) \) is denoted as:

\[
\eta(n_k) = \zeta(\epsilon_i) \cap C^*(\zeta(n_k))
\] (24)

The node content descriptor as defined can be used to represent a collection of estimated multimedia data contents as follows. Given a set of data objects \( I^* = \{x_{i1}, x_{i2}, \ldots, x_{ih}\} \), first find their semantic categories as described in definition 12. In each category, use the intersection of category region and vicinity constraint to describe a tightly bounding
region that encloses the data objects. Figure 34 shows an illustrative example of the node content description.

![Figure 34: An illustrative example of a node description.](image)

### 5.3.5 Summary of the Mathematical Basis

In section 5.3, we introduced the mathematical basis of semantic caching — data content classification, association and Bayesian based content estimation, and mobile node content profiling. The mathematical basis provides the capability of describing a collection of data objects (i.e. the image queries and results cached in the mobile nodes) effectively, which is further exploited in the following discussion of semantic-aware caching scheme.

### 5.4 Principles of Semantic-Aware Caching

In this section, a semantic-aware information caching scheme (SAIC) is presented and then we investigate how to process CBIR in such an organization. We also examine how to effectively utilize cache storage with respect to the QoS requirements.
5.4.1 Caching Rationale

The basic idea of the caching scheme proposed in this section, called Semantic-Aware Information Caching (SAIC), is as follows: Each node in an ad hoc network is allowed to gradually record semantic descriptions of the query results passing by it. The semantic descriptions, in the form of semantic categories and vicinity constraints (section 5.3.4), characterize the content distribution in the network. Figure 35 illustrates the basic idea of SAIC. Suppose node $n_r$ issues a similarity query $x_q$. Further assume that $x_q$ is resolved in node $n_s$ through flooding. The query result from $n_s$ will be relayed by a series of nodes to $n_r$. At this point, the chain of nodes from $n_s$ to $n_r$ could partition the network into two parts. Any future semantically similar query that crosses this chain will be resolved immediately without further communication. Suppose another node $n_r^*$ issues a query $x_q^*$ that is semantically similar to $x_q$. The $x_q^*$ is flooded in the network and encounters one of the relaying node. As a result, the $x_q^*$ is resolved at this relaying node and consequently, the query result from source node (i.e. $n_s$ or $n_r$) will be sent to $n_r^*$. Note that in this process the broadcasting of $x_q^*$ could be restricted within a section of the network. In addition, subsequent queries would partition the network into smaller groups. Thus the content-distribution information is propagated in the process of query resolution, and flooding is avoided for later queries.
**Query forwarding**: Initially, the local caches are empty and every query, if not resolved locally, is forwarded to other nodes for resolution. To minimize the number of messages spent on query forwarding, we use *Grid Location Service* (GLS) that tracks the location information of mobile nodes [114]. Figure 36 gives an illustrative example of query forwarding. The geographic space is divided into a collection of predetermined squares, each one containing a set of nodes. A query $x_q$, starting from the requesting node $n_r$, is forwarded between the grids following the spiral pattern. The spiral curve keeps growing until the data source node $n_s$ is found and the query result is forwarded back thru the shortest path between $n_s$ and $n_r$. Here we have an observation of the query forwarding process: Let $d$ denote the distance between $n_r$ and $n_s$, then all the nodes taking part in the query forwarding are within the sphere $\sigma(n_r, d)$ centered at $n_r$ with a radius $d$. In addition, each node only forwards the query along the spiral curve instead of re-broadcasting in all directions. Based on this observation, we can claim that the message complexity is restricted to linear order of the number of nodes, which is much smaller than that of flooding.

![Query forwarding using Grid Location Service (GLS).](image)

In choosing the grid cell size for the mobile data source nodes, we need to consider its dual impact on the resolution of content-based queries:

If the cells are set to small size, some grid cells may not contain any mobile node, therefore the query processing algorithm has to traverse multiple neighboring cells around the query forwarding curve. However, this traversal process consumes more system resources.
If the cell size is much larger than the normal distances between mobile nodes, the query forwarding algorithm needs to forward the query to many mobile nodes in a cell. Therefore, the query resolution may involve more than necessary mobile nodes.

Based on the above considerations, in this work we set the cell size to the mean value of node distances, which can be obtained from the history of query forwarding. With this configuration of cell size, most queries will be forwarded to a few nodes within one cell and the network traffic is drastically reduced.

**Cache updating:** When query \(x_q\) is forwarded between the nodes, its semantic content (i.e., the vicinity constraints Definition 20) is cached by the relaying nodes for future query processing purposes. The vicinity constraints are capable of representing one or multiple data objects: If only query \(x_q\) is cached, then the vicinity constraints are identical with the feature values of \(x_q\); if there are multiple data objects besides \(x_q\), the vicinity constraints represents the maximum and minimum values of all data objects in each feature dimension. When \(x_q\) is resolved at the data source \(n_s\), node \(n_r\) will send an updating message along the same route as the query forwarding. Thus each relaying node on the route is notified of the data source information. Later, suppose a node \(n_r^*\) issues a query \(x_q^*\), semantically similar to \(x_q\). The \(x_q^*\) is forwarded among the nodes following the same spiral pattern. When \(x_q^*\) meets with a node \(n_i\) which is the relaying node of \(x_q\), the query \(x_q^*\) is resolved at \(n_i\) and the result from source node (i.e. \(n_s\) or \(n_r\)) will be sent to \(n_r^*\).

### 5.4.2 Cache Model

Definitions 20 and 21 allow one to describe a set of multimedia data objects based on their semantic category regions and vicinity constraints. Similarly, the cache content of a mobile node can also be represented in the same way. The difference is that the mobile nodes only cache the semantic description of data objects, while the raw multimedia data are kept in the source nodes.

Two issues distinguish the SAIC caching scheme from the model proposed in [38]: First, the SAIC assumes an ad hoc network as its infrastructure, where each node behaves as both server and client. Therefore, each cached data item is labeled with a node id, which is used in the cache entry for routing of data sources in query resolution. Second, instead of caching raw multimedia data, the SAIC keeps a description of the semantics of a collection of multimedia data objects, and hence as will be shown later, it will offer a higher cache utilization.
In the SAIC scheme, the cache content is intended to characterize the data distribution of remote nodes through analysis of earlier queries. If an inner-category $k$-NN query $k$-NN$_c(x_q, \mathcal{E}_j)$ finds any semantically similar data object at a remote node $n_i$, the node id $n_i$ along with the semantic description of query result will be cached for future use. If $k$-NN$_c(x_q, \mathcal{E}_j)$ is not resolved at $n_i$, then node $n_i$ contains no semantically similar data objects in category $\mathcal{E}_j$ (i.e. $\mathcal{E}_j$ is vacant for query $x_q$ in $n_i$), thus $\mathcal{E}_j$ will be marked as a vacant region for $x_q$ and its semantically similar queries. As more queries are submitted, the categories $\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_t$ are potentially partitioned into two groups according to the data content of $n_i$: the categories that contain similar data and the ones that do not have similar data.

Logically, the local cache of a mobile node $n_i$ is a set of cache entries — each entry indicates one or multiple remote nodes in the network. A cache entry is a triplet $T_i = (\text{node list}, \text{matching region}, \text{vacant region})$. The matching region is the content description of resolved queries as defined in Definition 21, which can be considered as $n$-dimensional subspaces covering the data points of earlier query results. The vacant region shows the unresolved queries, which can be represented as a collection of subspaces where no query results are found. The node list shows the mobile nodes whose data contents can be
characterized by the matching region and the vacant region. Figure 37 illustrates an example of a cache entry.

5.4.3 QoS-Aware Organization

The scope of the proposed semantic caching scheme was extended to guarantee lower access latency for frequently requested data items. In this work, we define the QoS requirement for CBIR as the acceptable access latency. Normally, the frequently accessed multimedia data have higher QoS requirement.

The QoS-aware caching utilizes the multi-level semantic hierarchy, discussed in section 3.2, and the access frequency in order to define the level of abstraction in the cache — i.e., more popular data are cached at lower level of abstraction while less popular data are cached at higher level of abstraction. In other words, the frequently accessed data are represented as basic semantic categories (i.e., lower level in the semantic hierarchy), this facilitates faster resolution of queries directed to the popular data objects. For instance, given two objects \(x_i\) and \(x_j\), let \(\mathcal{E}_i\) and \(\mathcal{E}_j\) be the basic semantic categories that include \(x_i\) and \(x_j\). If \(x_i\) and \(x_j\) are popular data items, and there is enough space in the mobile node cache, we can use \(\mathcal{E}_i (\mathcal{E}_j)\) and \(\mathcal{N}^c(\mathcal{E}_i) (\mathcal{N}^c(\mathcal{E}_j))\) as the matching region and the node list, respectively, in the cache entry of \(x_i (x_j)\). If the objects \(x_i\) and \(x_j\) are accessed rarely or the cache space is not enough, we may use \(\mathcal{E}_i \cup \mathcal{E}_j\) and \(\mathcal{N}^c(\mathcal{E}_i \cup \mathcal{E}_j)\) as the cache entry for \(x_i\) and \(x_j\) to save cache storage. This scheme dynamically adjusts the representation granularity according to the access frequency, which improves cache utilization without violating the QoS requirement.

The performance improvement of the QoS-aware caching can be analyzed as follows: Consider the data set \(I = \{x_1, x_2, \ldots, x_m\}\) disseminated among mobile nodes \(N = \{n_1, n_2, \ldots, n_r\}\), let \(f(x_i)\) denote the access frequency of data object \(x_i\), then the probability of accessing \(x_i\) is:

\[
p(x_i) = f(x_i) / \sum_{x_j \in I} f(x_j) \tag{25}
\]

For the semantic category \(\mathcal{E}_i\) that includes \(x_i\), the probability of searching category \(\mathcal{E}_i\) is:

\[
p(\mathcal{E}_i) = \sum_{x_i \in \mathcal{E}_i} f(x_i) / \sum_{x_j \in I} f(x_j) \tag{26}
\]

And the conditional probability of choosing \(x_i\) from \(\mathcal{E}_i\) as the query result is:

\[
p(x_i|\mathcal{E}_i) = p(x_i)p(\mathcal{E}_i)
\]
Thus the probability of resolving the query $x_q$ in the semantic category $\mathcal{E}_i$ can be denoted as:

$$p(x_q|\mathcal{E}_i) = \sum_{x_j \in \mathcal{E}_i} p(x_j|\mathcal{E}_i)$$  \quad (28)$$

Considering a mobile node $n_s$ whose local cache contains $w$ entries, each entry stores the description of one or multiple semantic categories, partitioning $\mathcal{E}_1$, $\mathcal{E}_2$, ..., $\mathcal{E}_t$ into $w$ groups $G_1,...,G_w$. Let $\chi_i$ denote the number of semantic categories in the $i^{th}$ group, whose value varies from 1 to $t-w+1$. In the QoS-aware caching scheme, an entry representing several categories incurs higher resolving cost — to multicast the query among the nodes covered by these categories. Let $\overline{\chi}$ denote the mean cardinality of $\delta(\mathcal{E}_1,I),...,\delta(\mathcal{E}_t,I)$, thus the average query resolution cost of the QoS-aware caching in term of the number of hops can be denoted as:

$$\text{Cost}_{QoS} = \sum_{i=1}^{w} p(G_i) \sum_{\mathcal{E}_i \in G_i} p(x_q|\mathcal{E}_i)O(|\delta(\mathcal{E}_i,I)|^2)$$

$$\leq \sum_{i=1}^{w} p(G_i)O((t-w+1)^2\overline{\chi}^2)$$

$$= O((t-w+1)^2\overline{\chi}^2) \quad (29)$$

Without the QoS-aware scheme, the local cache can only accommodate $w$ semantic categories, using flooding to resolve queries in the remaining $t-w$ categories, resulting in an average cost of $O((t-w)\overline{\chi}^2m^2)$. Therefore, the performance improvement in term of average search cost is $O\left(\frac{m^2}{(t-w+1)\overline{\chi}}\right)$, where $\overline{\chi}<<m$ and $t-w+1<m$.

### 5.4.4 Cache Management

As discussed before, multimedia data are cached according to their semantic contents and access frequencies. In the SAIC, we use a two-phase step to facilitate effective management of caches, while at the same time avoiding unnecessary network traffic: 1) Initially, the local caches are empty and every query is flooded in the network. The result of a query is forwarded back thru a collection of relaying nodes, where the content description (i.e. the vicinity constraints) is cached for future query processing purposes. 2) The cached description is used to obtain a content distribution overview of the network. When a query is issued to a node $n_i$, it will be searched against its cache contents with the matching regions and vacant regions recorded in $n_i$, and then forwarded to the relevant nodes (i.e. the vicinity constraints) for further processing.
whose matching regions overlap with the query and whose vacant regions do not cover the query); if no relevant node is found, the query will be broadcasted to neighboring nodes and searched against their caches.

Another important issue in cache management is the consistency of local caches. Due to the bandwidth and power constraints of ad hoc networks, it is unrealistic to maintain strong cache consistency—broadcasting or pulling updates [69]. In this section, we propose a dynamic cache consistency approach that explores the semantic locality to avoid unnecessary broadcasting.

Our cache consistency approach deals with two cases: data insertion and data deletion. The update operation can be viewed as a deletion followed by an insertion. The dynamic cache consistency maintenance approach is performed as follows:

**Insertion:**

Suppose an object $x_i$ is inserted in node $n_j$. If $x_i$ is enclosed in the matching region of cache entries of $n_j$, then it is unnecessary to broadcast the insertion of $x_i$, since any future query semantically similar to $x_i$ will be forwarded to $n_j$. If $x_i$ is outside the matching region of $n_j$, then cache $x_i$ and notify the insertion to the nodes whose vacant regions overlap with $x_i$ through broadcasting, since any future query relevant to $x_i$ will be resolved at these nodes, relaying the query to $n_j$.

**Deletion:**

Upon the deletion of an object $x_i$ from node $n_j$, if $x_i$ is outside the matching region of $n_j$, then it is unnecessary to broadcast the deletion of $x_i$, since it will not affect the cache contents of other nodes. Only when $x_i$ is enclosed in the matching region of $n_j$ and there are no other objects in $n_j$ that belong to the same semantic category as $x_i$, it is necessary to broadcast the deletion of $x_i$ to the nodes whose matching regions overlap with $x_i$.

The main idea of the cache consistency maintenance approach is to flood the query when a node does not have proactive knowledge about content distribution, and forward the query only to relevant nodes when enough knowledge is obtained from the cache. Due to the limitation of cache size, the local cache may not have enough space for new query results. Instead of simply dropping the less frequently accessed data, we replace them with a coarser semantic category description, which represents a larger space that is composed of several smaller subspaces. When data updates occur, we only notify the nodes whose cache validity is affected. Due to the semantic locality, in most cases the insertion/deletion occurs
in a small region and the cache validity of other nodes is not affected [38], hence cache consistency maintenance only adds a trivial load to the network traffic.

5.5 Performance Study of Semantic-Aware Caching

To evaluate the overall performance of the proposed SAIC caching scheme, we implemented a simulator in ns-2 environment (version 2.26) [43]. It should be noted that our scheme does not rely on any specific network routing protocol; however, the simulation results are based on AODV routing protocol [38].

5.5.1 Simulation Setup

The simulation was initialized by assuming a default number of pre-existing nodes in the network and randomly setting up the connections between the nodes. In addition, to mimic the dynamic structure of the ad hoc networks, during the course of the simulation, a mixture of operations, including querying, updating, node joining, and node leaving, are randomly submitted to the network. The simulator relies on a set of input parameters that are summarized in Table 11.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default value</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation time</td>
<td>5000 seconds</td>
<td>100 – 20000</td>
</tr>
<tr>
<td>Environment size</td>
<td>1500m*320m</td>
<td>10^4 m^2 to 10^8 m^2</td>
</tr>
<tr>
<td>Transmitter range</td>
<td>100m</td>
<td>100m to 1,000m</td>
</tr>
<tr>
<td>Bandwidth</td>
<td>1M bps</td>
<td>0.1 – 10M bps</td>
</tr>
<tr>
<td>Number of nodes</td>
<td>100</td>
<td>50 to 100</td>
</tr>
<tr>
<td>Node mobility (v_max)</td>
<td>2 m/s</td>
<td>1 to 20 m/s</td>
</tr>
<tr>
<td>Local cache size</td>
<td>8 MB</td>
<td>20 KB to 8 MB</td>
</tr>
<tr>
<td>Query rate (Q_rate)</td>
<td>0.1 query/s</td>
<td>0.001 to 10 query/s</td>
</tr>
<tr>
<td>Control message size</td>
<td>2 KB</td>
<td></td>
</tr>
<tr>
<td>Data message size</td>
<td>20 KB</td>
<td>10 KB to 1 MB</td>
</tr>
<tr>
<td>Image dataset size</td>
<td>2000</td>
<td></td>
</tr>
<tr>
<td>Semantic category</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>Hot data probability(P_h)</td>
<td>0.7</td>
<td>0.1 – 1</td>
</tr>
<tr>
<td>Simulation time</td>
<td>5000 seconds</td>
<td>100 – 20000</td>
</tr>
<tr>
<td>Nearest neighbors</td>
<td>10</td>
<td>1 to 20</td>
</tr>
</tbody>
</table>
Node movement pattern: Two mobility models — random way point (RWP) [38] and Manhattan [39] — are used in the simulator. Each node randomly selects its movement (i.e. direction and velocity) within a 1500m×320m flat area. The node density can be adjusted by changing the number of nodes from 50 to 200.

Data and query distribution: The test bed comprises up to 2000 images of 100 semantic categories from the Corel dataset, which is similar to the dataset used in [78]. 1500 images in the test bed are used to train a LPP subspace learning module that partitions the semantic space into 100 orthogonal regions, and the remaining 500 images are used as test images for CBIR queries. To examine the effect of semantic locality, we randomly select 100 images as a “hot” dataset, and set $P_h$ as the probability of choosing these “hot” data from the whole data set. The query generation time follows the exponential distribution, which is similar to the previous work in [69]. The access pattern in the queries follows Zipf-like distribution, which is widely used to model non-uniformly distributed queries [83].

5.5.2 Simulation Result

Experiments were run using different workloads and system settings. The proposed model has been compared and contrasted against those presented in [69] based on performance metrics such as cache hit ratio and response time for different workloads, i.e., mean query rate and nearest-neighbor number, and different system parameters, i.e. cache size, network density, and node mobility.

- Query Resolution Accuracy

One of important metrics for content-based information retrieval systems is the retrieval accuracy. The performance of a search scheme can be evaluated by its accuracy when accessing a restricted portion of mobile nodes. In the simulation we restricted the search schemes to visit less than 10% of the nodes. The accuracy is evaluated as the percentage of the query results that belong to the same semantic categories as queries. In the simulation we used the 500 testing images as queries. Figure 38 compares the accuracy of query results returned by semantic caching schemes with no content estimation, association rule based estimation, and Bayesian based estimation as the number of queries increases. Generally speaking, more query history improves the accuracy of both association-based estimation and Bayesian-based estimation. Note that in Figure 38, the accuracy of “no estimation” also improves as the number of queries increases. This improvement is due to the warm up period of cache — i.e. the cache achieves its peak performance after a certain amount of queries. One can notice that the “no estimation”
scheme achieves its peak performance at 600 queries, and keeps comparable performance at 800 and 1000 queries. The association-based scheme performs worse than Bayesian at the beginning due to the poor stability of association rules generated with small sample set of queries. As the query history becomes longer, the association rules precisely describe the relationships among data objects and therefore help improving the retrieval accuracy.

![Figure 38: The comparisons of content estimation impacts.](image)

![Figure 39: The comparisons of caching strategy impacts.](image)

In another simulation run, we tested the impacts of different caching strategies. The total time of query history ranges from 5 hours to 55 hours, with the query rate of 0.1 query/second. Since the queries are taken from the data set of 2000 image data objects, for simplicity, we limited the query history to 20,000 queries, giving an average of 10 queries
per image. We used Bayesian based estimation in this simulation run for all caching schemes. Because Bayesian based estimation does not need an initial period of time to accumulate query history as association based estimation does, and the cache update of SAIC takes roughly similar time as CacheData and CachePath, there is no initial overhead of SAIC in comparison to CacheData and CachePath. As can be seen from Figure 39, the SAIC outperforms Cache-Path and achieves comparable accuracy as CacheData. The slightly better performance of CacheData is due to its caching of raw images and the relatively large cache size (8MB). However, CacheData relies on large cache storage to achieve the accuracy. From Figure 40 one can conclude that when the cache size is smaller than 3MB, CacheData is less accurate than SAIC. In a mobile environment, such as ad hoc networks, it is unlikely that mobile units have large cache storage. Therefore, SAIC is more practical for real mobile applications. The SAIC returns more accurate query results than CachePath due to its exploitation of image content distribution, which gives the heuristic information for well-aimed search.

![Figure 40: The comparisons of cache size impacts.]

- **Cache Hit Ratio**

  Traditional performance metrics of a caching scheme, i.e., query response time, search cost, and throughput, are highly dependent on cache hit ratio. In a series of simulations, we examined how cache hit ratios are affected by different caching strategies. Figure 41 shows the cache hit ratio as a function of the node density and cache size. As can be seen, our scheme regardless of the cache size, offers a higher hit ratio than the CachePath and CacheData models. This implies that relative to the CacheData and
CachPath models, the SAIC offers a reasonable hit ratio with smaller cache size. Within the scope of mobile paradigm, this is an interesting observation since mobile nodes always strive for limited memory.

Figure 41: The effect of density on cache hit ratio.
(query history limited to 20,000 queries)

In addition, the cache hit ratio of SAIC increases as the number of nodes increases, while CacheData and CachePath have decreased hit ratio with increased node density — For a fixed data set and a fixed set of queries, the semantic locality of queries decreases as the number of data objects per node decreases and hence cache effectiveness drops. However, in SAIC, increase in the node density also implies an increase in semantic replica of the query results. This increases the probability of cache hit for future semantically similar queries. Also note in this simulation run, the cache hit ratio of the SAIC achieves its peak at the cache size 1MB, while the hit ratio of the other two schemes keeps growing as the cache size increases. This implies that the SAIC does not rely on large-size cache to achieve the highest hit ratio, and can give satisfactory performance with reasonable cache size.

From Figure 42, as one can expect, the cache hit ratio of SAIC and CachePath drops as the node mobility increases — increased in mobility incurs more changes in network topology, making it more difficult for CachePath and SAIC to locate remote data source nodes. The cache hit ratio of CacheData, in comparison, is not drastically affected by the mobility due to its independence from path information. The cache hit ratio of CachePath drops at a faster rate than the SAIC. This is due to the cache replacement policies adapted by these two caching schemes: CachePath simply removes the less frequently accessed data items to save cache space. However, the SAIC attempts to increase the semantic contents of
the cache by using coarser semantic descriptions for the less frequently accessed cache entries. Increase in the semantic contents of the cache improves the cache hit ratio and hence, implies better cache utilization and more robust model adaptable to dynamic network topology.

![Figure 42](image)

**Figure 42:** The effect of mobility on cache hit ratio.

Figure 42 also depicts the correlation between the cache hit ratio and the query generation rate. In general, the hit ratio improves as queries are generated at slower rate. Although network topology could change between consecutive queries as the query rate decreases, it is the responsibility of routing protocols (e.g. AODV) to find the new routes. Since the existing routing protocols work steadily when the nodes keep a fixed mobility speed, the cache hit ratio in such a case in mainly related with query rate. However, in contrast with CacheData and CachePath, after a threshold point, the hit ratio for the SAIC remains almost constant.

Figure 43 shows the cache hit ratios of the caching schemes using different mobility patterns. All schemes improve their performance in Manhattan pattern. This is due to the spatial and temporal dependence in Manhattan pattern that leads to reduced probability of route breaks and topology changes. In addition, the performance variation of the SAIC is less drastic than CachePath. This is due to the cache replacement policies adapted by these two caching schemes: CachePath simply removes the less frequently accessed data items to save cache space. However, the SAIC attempts to increase the semantic contents of the cache by using coarser semantic descriptions for the less frequently accessed cache entries. Increase in the semantic contents of the cache improves the cache hit ratio and hence,
implies better cache utilization and more robust model adaptable to dynamic network topology.

**Figure 43:** The effect of mobility pattern on cache hit ratio.

- **Query Response Time**

  In another simulation run, we measured the query response time as a function of node density, cache size, and average query generation time. Figure 44 shows the impact of node density on average query response time. As examined before, the increased node density: 1) Changes the network topology more frequently, resulting in longer response time. 2) Increases the number of queries, leading to a higher network traffic and hence longer response time. Also note that the increased node density causes lower hit ratio of CacheData and CachePath (Figure 41), and the hit ratio of SAIC increases as the node density increases. However, the combined effect of longer query result delivery time, higher traffic, and frequent disconnections counteract with the improved hit ratio and thereby making the response time longer. Compared with CacheData and CachePath the average response time of SAIC increases at a much slower rate. This is justifiable since more queries allow more semantic contents to be captured at each node, which results in higher probability of hits for semantically equivalent queries.
Figure 44: The effect of density on response time.

Figure 45 shows the combined effect of cache size and mobility on query response time. As can be seen from Figure 45, CacheData is more sensitive to the cache size than CachePath and SAIC. For a fixed set of queries, CacheData requires much more cache space because it stores the raw image data. CachePath and SAIC, in comparison, do not directly cache the raw image data, making better utilization of cache storage.

Figure 45: The effect of cache size on response time.

From Figure 45 one can also conclude that the average response time of SAIC is less than that of CachePath. However, the gap increases as cache size increases. This can be
explained by their difference in caching rationale: The cached data items are considered as independent entities in CachePath, and cache hits only imply exact match. Cache misses will cause flooding in the whole network. In comparison, SAIC exploits the semantic locality of data items to increase hit ratio. The SAIC also makes use of semantic replicas to partition the network into small regions, and avoids large-scale flooding caused by cache misses. As more queries are resolved, more semantic replicas are disseminated densely and evenly in the network, which reduce the average latency of resolving queries.

In a separate simulation run, we also evaluated the combined effect of content estimation methods, node density, and caching strategies. The CacheData and CachePath schemes can also take advantage of Bayesian and association based content estimation by pre-fetching the relevant data objects that have high probability of being accessed after the query resolution. As can be seen from Figure 46, the content estimation helps reducing the response time in all caching strategies. Also note that as the number of nodes increases, the response time increases due to the combined effect of higher hit ratio, longer query result delivery path, and larger traffic. Generally, as one can conclude from Figure 46, SAIC achieves smallest response time due to its highest hit ratio and efficient representation of semantic locality.

![Figure 46: The effect of content estimation and caching strategy.](image)

- **Network Traffic**

  In order to evaluate the impact of the caching strategies on the network traffic, we tuned the simulator to examine the message overhead on mobile nodes. Figure 47 shows that SAIC incurs much less message overhead than CachePath and CacheData. The reason
is that SAIC resolves queries using semantic replicas in nearby nodes instead of faraway data source nodes, if possible. Therefore, the data requests and replies need to travel less number of hops and mobile nodes need to process less number of messages. As cache size increases, the hit ratio increases and message overhead decreases. However, SAIC is not sensitive to cache size (Figure 41), implying its usefulness in applications with small storage.

![Figure 47: The average traffic on each node.](image)

In CacheData and CachePath the cache misses could incur flooding in the whole network, and each node may reply the query with the most similar images in its local database. The multiple data replies to the query further increases the network traffic, and thus implies higher requirement for bandwidth. SAIC solves this issue by performing inner-category $k$-NN on a small portion of the network — the semantically most relevant nodes. Figure 48 shows that the bandwidth has much less impact on SAIC than CacheData and CachePath. Note that CacheData achieves comparable response time to SAIC as bandwidth increases. The reason is that the increased bandwidth remedies the difference of access latency between local cache and remote nodes, reducing the effect of flooding and duplicated query results.
The Effect of QoS-Aware Organization

As mentioned in section 5.4.3, SAIC employs a QoS-aware cache organization policy to improve cache space utilization. The frequently accessed data are represented using finer semantic descriptions in the cache, which implies faster resolution of queries. This QoS-aware organization utilizes the cache space according to the popularity of data items. CacheData and CachePath, in comparison, do not provide favorable cache content to frequently accessed data, failing to exploit the data access locality.
In our simulation, the test image dataset is divided into two sections: the hot dataset (20% of the total data) and the cold dataset (80% of the total data). Queries are generated by selecting images from the hot dataset and the cold dataset, abiding a pre-defined probability $P_h$. Figure 49 shows how the access locality affects the response time. As hot data access probability increases, the response time decreases for all three caching schemes, but the response time of SAIC drops at a much faster rate than CacheData and CachePath. This implies the better utilization of data access locality due to the QoS-aware organization of SAIC.

5.5.3 Evaluation Conclusions

We proposed a semantic-aware caching scheme (i.e. SAIC) to facilitate content-based multimedia retrieval in ad hoc networks. This scheme is based on analysis of cached query results to represent the data contents in each node. It has several novel characteristics such as content distribution estimation and QoS-aware cache management.

The proposed SAIC scheme makes use of the data content distribution in ad hoc networks to resolve $k$-NN multimedia queries without incurring flooding in the network. Through extensive experimental study, we found that the semantic-aware caching scheme has the following features:

- multi-level partitioning of the semantic space based on hypernym/hyponyms,
- association and Bayesian probability based content estimation,
- constraint-based representation method showing the semantic similarity between multimedia data objects,
- non-flooding query processing, and
- adaptive QoS-aware cache consistency maintenance.

5.6 Summary of Chapter Contents

In this chapter, we have given a semantic-aware caching scheme to facilitate the content-based image retrieval in mobile ad hoc networks where the local databases are composed of a community of mobile computing devices. The semantic-aware caching scheme inherits the multi-granularity content description method from the SSM and uses it in the QoS-aware cache management.

To describe the content distribution of mobile data sources with consideration of their mobility, we gave frequent pattern based and Bayesian based approaches for the
profiling of the data contents of mobile nodes. We also gave algorithms to generate the association rules or update Bayesian probabilities for the content profiling and estimation.

To facilitate the content-based retrieval on the mobile nodes, we introduced the semantic-aware caching that keeps track of the earlier query results and helps to locate the relevant nodes for new queries. We also gave a dynamic cache updating approach that adjusts the data granularity according to the cache storage, thereby providing a QoS-aware mechanism for the cache management.

The effectiveness of the proposed extension in performing content-based retrieval has been validated by extensive experimental study. Although SAIC is proposed for ad hoc networks, it can also be applied to other wireless networks such as cellular networks and WLANs.
6 CONCLUDING REMARKS AND FUTURE WORK

In this chapter, we thread the pieces of this research together and present the overall picture by summarizing the major contributions of this thesis and pointing out several future research directions.

6.1 Contributions of this Thesis

Our research aimed at the following challenges:

- How to represent the semantic contents of multimedia data objects and their relationships?
- How to integrate a collection of existing heterogeneous multimedia data repositories?
- How to describe the multimedia data content distribution in ad hoc networks?
- How to achieve low-cost content-based multimedia information retrieval in the dynamically changing environment such as mobile ad hoc networks?

Our study has contributed to several research topics: multimedia data representation, semantic-based integration of data sources in mobile ad hoc networks, and semantic-aware and QoS-aware information retrieval. The following summarizes our work in these topics:

Logic-Based Multimedia Representation: We introduced a logic-based framework for multimedia content representation in Chapter 3. The Summary-Schemas Model (SSM) framework was used as the underlying platform. More specifically, we used first-order conjunctive and disjunctive expressions to describe the contents of data objects. This model has the capability of integrating both object-level and granule-level features of multimedia data objects and supporting all three types of content-based retrieval (form-based, pure-semantic-based, and the combination of both form and semantics). In addition, we presented the query resolution method in the domain of SSM.

We conducted theoretical and experimental evaluations regarding the integration of the SSM and logic-based content representation for content-based multimedia retrieval. We proved that the integration of logic-based representation and the SSM requires the smallest search cost in performing content-based multimedia retrieval. Experimental results show that the performance of the logic-based model is always an envelope of other content-based
indexing models (e.g. R*-tree, M-tree, SS-tree, SR-tree, and VP-tree), with various metrics including search cost, retrieval accuracy, disk access frequency, and network traffic.

Semantic-Based Clustering in Ad Hoc Networks: Content-based image retrieval is a challenging problem in mobile ad hoc networks due to the multiple limitations such as network bandwidth, infrastructure-free nature, and node mobility. The traditional systems employ either centralized or flooding strategies, which may result in low fault tolerance or high search cost. In Chapter 4, we proposed a decentralized non-flooding semantic-based clustering scheme — Extended Summary Schemas Model (ESSM). The ESSM scheme makes use of the data content distribution in ad hoc networks to reduce the search cost without incurring high maintenance overhead. Experimental results, presented in section 4.3.2, have shown that ESSM drastically reduces search cost relative to traditional flooding approach, achieves high accuracy while visiting only a small portion of mobile nodes, and offers scalability and robustness in large-scale networks.

Similarity Retrieval in Ad Hoc Networks Using Semantic-Aware Caching: In the domain of ad hoc networks, most of the previous studies focused on the efficient exploration of routing information with less attention on caching of data contents. In Chapter 5, a semantic-aware caching scheme (SAIC) was introduced to facilitate content-based image retrieval. SAIC has the capability of keeping track of the earlier query results and helping to locate the relevant nodes for new queries. It also possesses a dynamic cache maintenance mechanism that adjusts the data granularity according to the cache storage and data popularity, thereby providing a QoS-aware service. Through extensive theoretical and experimental study, we found that the semantic-aware caching scheme drastically reduces the search cost, effectively utilizes the cache storage, and significantly improves retrieval accuracy and response time.

In summary, we believe that the semantic-aware representation and retrieval models based on the SSM have provided efficient and effective platforms for multimedia data access in various environments. They can be applied to a variety of applications such as digital library, search engine, pervasive computing, etc.

6.2 Future Research

Our research can be further extended in the following directions:

- The logic-based content representation can be used to describe other forms of content-rich data, such as spatial, XML, and biomedical data. Since these data can be described as some kinds of feature vectors, we can utilize the
logic-based optimization method proposed in section 3.1.2 to obtain a more effective representation.

- The semantic-based clustering approach can be extended for the efficient description of content distribution in peer-to-peer networks and content-addressable networks (CANs) [42]. The dynamic organization of clusters can provide more flexible content-based indexing on such networks.

- The fast growth of sensor networks has attracted considerable research attention. The sensor networks have several prominent challenges such as limited power and life time, and non-fixed topology. The semantic-aware and QoS-aware caching scheme can be utilized for the content delivery and retrieval in sensor networks, considering its capability of adjusting to dynamic topologies and reducing the average search cost.

Generally, we believe that the proposed methodologies in the thesis can alleviate the challenges in the aforementioned research directions and provide desirable solutions.
7 BIBLIOGRAPHY


VITA

Bo Yang is a Ph.D. candidate of the Computer Science and Engineering Department at the Pennsylvania State University. He received the B.S. degree and M.Engr. degree from Shandong University, China, in 1997 and 2000, respectively, both in Computer Science. His main research interests include multimedia content analysis and representation, semantic-based multimedia information retrieval, mobile data management, and ad hoc networks.