UBIQUITOUS MULTIMEDIA INFORMATION RETRIEVAL IN AD HOC NETWORKS USING SEMANTIC CLUSTERING

Bo Yang, Manohar Mareboyana
Bowie State University, MD, USA
byang@bowiestate.edu

ABSTRACT
Ubiquitous multimedia data processing, i.e. the management of multimedia data objects from ubiquitous data sources, has experienced an explosive development because of the technological advances in mobile and ubiquitous computing, network infrastructures, and multimedia streaming. 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. To provide an efficient platform for multimedia information retrieval in ad hoc networks, we propose to cluster mobile multimedia databases based on their semantic contents, and construct a virtual hierarchical indexing infrastructure overlaid on the mobile databases. 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.

Keywords: semantics, multimedia information retrieval, content-based indexing, content distribution.

1 INTRODUCTION
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. An ad hoc network is a collection of cooperative mobile nodes that communicate with each other without the intervention of accessing points. These mobile nodes are capable of not only storing and processing data, but also performing complex operations through their communications, such as on-demand routing [4,5] or multimedia data streaming [2]. Within the scope of ad hoc networks, most of the previous research focuses on the routing protocols that adapt to the dynamic network topology [6]; however, information retrieval is becoming an important issue in a variety of recent applications [3]. From the viewpoint of information retrieval, the organization of ad hoc networks can be classified into three categories [32].

Some earlier ad hoc network prototypes are linked to centralized data source nodes (data centers) that host constantly updated directories of data contents [4]. 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 point of failure. Moreover, the data center behaves as a hotspot and its movement within the area could increase the network traffic.

The recently proposed ad hoc network frameworks are decentralized and have no data centers — the network is unstructured and mobile nodes form peer-to-peer connections to resolve queries through the cooperation with each other [7]. Flooding is the most common approach for information retrieval in such 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. The flooding approach consumes system resources, i.e., storage, bandwidth, and energy, drastically and hence offers good performance only when dealing with text information [21]. Considering the sheer size of the multimedia data, the performance deterioration is more drastic. In addition, the flooding strategy may cause duplicate queries and retrieval results, which may further increase the cost of the query processing [30].
To overcome the shortcomings of the blind search approach, alternatively, the literature has proposed the structured ad hoc networks in which the network topology does not change drastically [22, 23]. In such networks, the data objects are not allocated to the network nodes randomly, but at specified locations that will make subsequent queries easier to satisfy [3]. Such design improves the efficiency of information retrieval in some cases at the expense of the flexibility and scalability of the ad hoc networks. In practical applications, the network topology and the data contents of the mobile nodes are constantly changing, which increase the difficulty of efficient multimedia data retrieval.

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 ad hoc networks have 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 mobile computing devices in the network have frequent relocations, making it important to consider location information in the query processing; 3) The limitation of system resources (e.g. bandwidth and storage) require efficient approaches for handling voluminous multimedia data; and 4) 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 ad hoc networks.

Motivated by the aforementioned challenges, we introduced a semantic-based, 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. With the help of the proposed scheme, the multimedia data 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.

The remaining part of this paper is organized into seven sections: Section 2 introduces the background information and related work. Section 3 outlines the preliminary concepts of the system model. Section 4 introduces the clustering of mobile nodes according to their data contents and presents the dynamic adjustment of the clusters in accordance with content and topology changes in an ad hoc network. Section 5 analyzes the performance of the proposed scheme. Section 6 evaluates the proposed scheme with experimental results. Finally, section 7 draws the paper to conclusions.

2 BACKGROUND AND RELATED WORK

2.1 Multimedia Content Extraction

Due to the sheer size of multimedia data, considerable research work has been done on extracting semantic contents that concisely describe multimedia objects [8-10]. In this work, the semantic contents can be obtained through the recognition of objects [11], which is performed in two phases:

- The multimedia object is first partitioned into several segments, which indicate the most significant visual/audio components [12], and
- The semantics of the segments are obtained through Latent Semantic Analysis (LSA) [8, 10].

The segmentation method employed in this work is similar to the binary-partition-tree approach [9]. Similar pixels are merged together as homogeneous regions, and these small regions are recursively combined into segments of the multimedia data object.

The low-level feature representation obtained from segmentation cannot represent the semantic contents, and therefore does not provide an ideal basis for semantic-based retrieval. Hence, the LSA is employed to obtain the semantics of these segments. In this work, the data set of the LSA includes two sets of entities: the objects whose semantics are already known (training samples), and multimedia segments whose semantics remain unknown [8]. The LSA uses Singular Value Decomposition (SVD) to uncover the hidden semantic relationships (e.g. synonym and polysemy) between data objects: the multimedia segments are classified into proper semantic categories after the singular value decomposition, and are assigned with proper semantics [13].

2.2 Content-Based Retrieval

Based on the method in which multimedia data contents is extracted, the literature has recognized several content-based retrieval techniques [14-17]. The traditional centralized content-based retrieval approaches are based on feature representation of multimedia data that can be categorized as two classes:

- The partition-based retrieval approaches recursively divide the multimedia objects (or k-dimensional feature spaces) into disjoint partitions, with clustering or classifying algorithms, while generating a hierarchical indexing structure on these partitions. The
earlier models in this class include Quad-tree [14], K-D-tree [15], and VP-tree [15]. The recent research has focused on models based on clusters [16]. The partition-based approaches normally employ very complex computations on features [15], which make these models inefficient for real-time multimedia applications. Moreover, the partition-based models do not consider feature spaces of different dimensions — i.e., multimedia objects from heterogeneous data sources [2].

The region-based retrieval approaches employ small regions (either in the form of Minimum Bounding Rectangles or Spheres) to cover all the multimedia objects. Based on these bounding regions, a balanced tree is constructed as the indexing structure. This class of indexing models includes the R-tree and its variations (e.g. R*-tree, R+*-tree, and SR-tree) [17, 18]. Relative to the partition-based approach, the region-based methods improve storage utilization by avoiding forced splits. However, due to the shapes of the bounding regions, the region-based approaches have the born weakness of overlapping [18], which deteriorates the searching performance. Furthermore, the data objects grouped in the same region may not share common semantic contents.

The disadvantages of centralized strategy have motivated the research on various decentralized information retrieval models for ad hoc networks [19-21]. Some recently proposed models in the literature try to make use of data distribution to facilitate information retrieval [22-24]. Content-Addressable Network (CAN) model was proposed as a decentralized infrastructure that provides hash-table-like functionality on large-scale distributed networks [22]. A Distributed Hash Table (DHT) is employed to map the data objects into a k-dimensional logical Cartesian space, which allows the storage and retrieval of (key, object) pairs. Further improvements of CAN have led to the logical overlay networks (e.g. pSearch [23] and SSW [24]) that use dimension-reduction techniques to reduce search cost. These content-navigated retrieval models are based on the feature vector representation of object contents [23], which is inefficient in describing multimedia data: In practical multimedia database systems, the feature vectors are often of fixed sizes to facilitate the computation and representation. However, in many instances, some features may be null. The null features do not contribute to the semantic contents of multimedia objects; however, they still occupy space in the feature vectors — hence, lower storage utilization.

In an ad hoc network with heterogeneous distributed data sources, traditional centralized retrieval methods may not work efficiently. Due to the node mobility and dynamic topology, the centralized strategy may require extra network traffic to locate the data sources [5]. In many cases, the extra traffic load is formidable high to ad hoc networks, considering their limited bandwidths and computation resources. Moreover, the centralized data server behaves as the single point of failure, which deteriorates the robustness and scalability of the system.

2.3 Object-Based Indexing Infrastructure

The SSM was originally proposed as an object-oriented search engine that allows imprecise access to text information in a multi-database environment [25]. It is a global layer located on top of multiple pre-existing autonomous and heterogeneous local databases. The SSM comprises three major components: a thesaurus, the local nodes, and the summary-schemas nodes (figure 1). The thesaurus defines a set of standard access terms, the semantic categories they belong to, and their semantic relationships. It may utilize any of the off-the-shelf thesauruses (e.g. Roget’s thesaurus) as its basis. A Semantic Distance Metric (SDM) is defined to provide quantitative measurement of “semantic similarity” between user query and the database contents [25, 26]. A local node is a physical database containing physical data. A summary-schemas node is a logical data entity, a meta-data (summary schema), representing 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 SSM organizes data objects in a hierarchical fashion based on synonyms, hypernyms and hyponyms relationships. Synonyms are semantically similar objects in different data formats or same-format objects in different physical locations. The SSM employs synonym links to connect and cluster the “similar-content” data objects together. A hypernym is the generalized description of the common characteristics of a set of data objects. To find the proper hypernyms of data objects, the SSM checks the on-line thesaurus to obtain the mapping from data objects to hypernym terms. The higher-level hypernyms that describe the more comprehensive concepts can be obtained from abstracting the hypernym terms of data objects. Recursive application of hypernym relation generates the hierarchical meta-data of the SSM. This in turn conceptually gives a concise semantic view of all the globally shared data objects. As the counter concept of hypernym, a hyponym is the specialized description of the characteristics of data objects. In the SSM prototype, the hyponym links compose the routes from the most abstract descriptions to the
specific data objects. More details about the SSM can be found in [25-27].

3 PRELIMINARIES

3.1 Extended Summary Schemas Model

As mentioned before, the SSM was originally proposed for efficient management of textual and traditional databases in a wired multi-database environment. The original SSM prototype cannot be directly applied in accessing multimedia data in ad hoc networks due to several challenges: 1) Unlike textual and conventional databases, the multimedia data usually contain more complex semantic attributes (e.g., color, texture, and annotations) that cannot be represented by conventional data models; 2) The underlying fixed and predefined network topology of the SSM does not accommodate the infrastructure-free and dynamic nature of ad hoc networks.

Considering these limitations, in this paper, we strengthened the representation capability of the SSM, and proposed a clustering strategy for organizing multimedia data in ad hoc networks. The proposed scheme is called Extended Summary-Schemas Model (ESSM). The major properties of the ESSM can be summarized as follows:

- The semantic contents of multimedia data objects on each mobile node are extracted and represented using concise data-content-distribution expressions. The distribution-based representation approach we proposed in this work is capable of accommodating heterogeneity of the data sources. The details of this representation approach will be discussed in section 3.2.

- The mobile nodes are partitioned into clusters based on the semantic similarity of their data contents. The mobile nodes with similar data contents are called Content-Related Nodes. And the clusters are denoted as Content-Based Clusters. These clusters are recursively aggregated together to form larger clusters. More detailed description will appear in section 4.1.

3.2 Multimedia Content Representation

To represent the contents of multimedia objects in ad hoc networks, we present a novel semantic-based scheme, which can be briefly described as follows: Given a set of multimedia data objects distributed among a collection of mobile nodes, our goal is to find the data content distribution over the mobile nodes and set up the content-based relationships among these nodes.

The ad hoc network can be considered as an undirected connected graph \( G = (N, M) \), where \( N = \{n_1, n_2, \ldots, n_r\} \) is the set of mobile nodes and \( M = \{m_1, m_2, \ldots, m_s\} \) is the set of wireless connections between mobile nodes. \( X = \{x_1, x_2, \ldots, x_m\} \) is the set of multimedia data objects. Each data object \( x_i \) is represented as an \( n \)-dimensional semantic vector \( \phi_{x_i} = (\omega_1^{x_i}, \omega_2^{x_i}, \ldots, \omega_n^{x_i}) \). The elements in the semantic vector could be: manually added keywords, automatically extracted features, or relevance feedback generated by users. The distribution pattern of the data objects over the mobile nodes can be considered as a many-to-many relationship, i.e., each data object may be distributed among multiple mobile nodes, and each mobile node may contain a collection of data objects.

Figure 2 illustrates an ad hoc network and the data objects distributed among the mobile nodes. The multimedia data objects, represented as small circles, 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 \( \chi(n_i) = \{x_1, x_2, \ldots, x_t\} \), from the semantic space. The information contents of a node collectively represent an area in this semantic space (in Figure 2, this is represented as a rectangle). As can be noted from Figure 2, in this information space, mobile nodes may contain overlapping data (duplication) or semantically overlapping information. For example, both node \( A \) and node \( B \) contain the data objects in the semantic...
region of $\chi(A) \cap \chi(B)$. Generally, if an object $x_i$ is distributed over a collection of mobile nodes, we use $\psi(x_i)$ to denote the set of mobile nodes containing the replicas of $x_i$.

**Definition 1: The Semantic Similarity**

The semantic similarity between two data objects is defined based on the Euclidean distance between their corresponding semantic vectors. Formally, the distance between two data objects $x_i$ and $x_j$ is defined as a function $\text{dist}(x_i, x_j) = \cos \frac{\|\psi_{ai} \cdot \psi_{aj}\|}{\|\psi_{ai}\| \cdot \|\psi_{aj}\|}$, where $\|\psi_{ai}\|$ denotes the Euclidean vector norm of semantic vector $\psi_{ai}$ indicating the length of $\psi_{ai}$ in the Euclidean geometry ($\|\psi_{ai}\| = \left(\sum_{k=1}^{n} (\omega_{ai}^{k} \cdot \omega_{aj}^{k})^{2}\right)^{1/2}$) and $(\psi_{ai} \cdot \phi_{aj})$ is the scalar product of two semantic vectors $\psi_{ai}$ and $\phi_{aj}$ ($\psi_{ai} \cdot \phi_{aj} = \sum_{k=1}^{n} (\omega_{ai}^{k} \cdot \omega_{aj}^{k})$).

### 3.3 Retrieval Formalization

The content-based multimedia retrieval (nearest-neighbor query) in an ad hoc network can be described as follows: Given the set of multimedia data objects $X \subset \mathbb{R}^n$ and the ad hoc network $G$, for a specific integer constant $k$ and a given query object $x_q$, return $k$ data objects $X' = \{x_1, x_2, \ldots, x_k\} \subset X$ such that for $\forall \ (x \in X \land x \not\in X')$, $\text{dist}(x_q, x) \geq \text{dist}(x_q, x_i)$ (for $i \in [1, k]$) is always satisfied. Formally, the nearest-neighbor retrieval is defined as:

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

Given an object set $X = \{x_1, x_2, \ldots, x_m\}$ and a query object $x_q$, the nearest-neighbor retrieval of $x_q$ within $X$, denoted as $k$-NN$_k(x_q, X)$, is the following set:

$$k$-NN$_k(x_q, X) = \{x_i \mid \forall x \in \text{NN}(x_q, X), \text{dist}(x_q, x) \geq \text{dist}(x_q, x_i) \land \text{NN}(x_q, X) = k\}$

In the context of ad hoc networks, the resolution of nearest-neighbor query may cause flooding and forcing pair-wise comparisons of 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 optimized search cost.

### 3.4 Data Content Classification

In the context of distributed data sources in an ad hoc network, the cost of k-NN retrieval is formidable high due to the necessity of traversing the whole network. Note that semantically similar data objects are densely located 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 3: 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 $\mathcal{E}_i$, $\mathcal{E}_2$, ..., $\mathcal{E}_t$.

The semantic categories can be deducted from a training sample, formally denoted as $T = (x_1, x_2, \ldots, x_t)$, which satisfies $T = \bigcup_{i=1}^{t} \mathcal{E}_i$. Let $\delta(\mathcal{E}_i, T)$ 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 $\chi(\mathcal{E}_i)$, can be viewed as the locus of points whose semantic distance to the data points in $\delta(\mathcal{E}_i, T)$ is smaller than to those in any other semantic category.

Given a multimedia data object $x_i$ and a semantic category $\mathcal{E}_j$, if $x_i$ belongs to $\mathcal{E}_j$, their relationship is denoted as $x_i \in \mathcal{E}_j$.

**Definition 4: Inner-Category Similarity Search**

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

$$k$-NN$_k(x_i, \mathcal{E}_j) = \{x_i \mid \forall y \in k$-NN$_k(x_i, \mathcal{E}_j), \text{dist}(x_i, x_i) \geq \text{dist}(x_i, x_j) \land x_j \in \chi(\mathcal{E}_j)\}$

Generally, due to the limitation of the semantic category, $k$-NN$_k(x_i, \mathcal{E}_j)$ may only return less than $k$ similar data objects that belong to the semantic category $\mathcal{E}_j$. In contrast to the normal similarity search (Definition 2), we have the following claim:

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

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

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

From (a) and (b) we conclude that $k$-NN$_k(x_q, \mathcal{E}_j) = k$-NN$_k(x_q, X)$.

Based on Claim 1, we may perform content-based retrieval in an alternative approach: Given an image $x_q$, first find its semantically most related category $\mathcal{E}_p$, and perform inner-category k-NN in $\mathcal{E}_p$. This alternative approach returns the same results as the classical retrieval 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$_k(x_q, \mathcal{E}_j)| \neq k$. 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 is defined as follows.

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

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

\[
\exists \Omega' = \{ \epsilon'_1, \epsilon'_2, \ldots, \epsilon'_j \} \subseteq \Omega \text{ such that } \bigcup_{i=1}^j k-\text{NN}(x_q, \epsilon_i) = k-\text{NN}(x_q, X). \text{ (i.e. the global k-NN retrieval in semantic space can be decomposed into a set of inner-category retrievals in semantic categories)}
\]

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

Proof:
1) Based on Claim 1, k-NN(\( x_q, X \)) can be performed as a series of k-NN(\( x_q, \epsilon_i \)) in several semantic categories, until the sum of returned data objects is \( k \).
2) The semantic hierarchy \( H_S(\Omega, \psi) \) represents \( \epsilon_1, \epsilon_2, \ldots, \epsilon_l \) using more generic concepts, hence there exists a semantic category \( \epsilon' \) in \( H_S(\Omega, \psi) \) satisfying \( \bigcup_{i=1}^j \epsilon_i \subseteq \epsilon' \).

The semantic hierarchy provides a means of partitioning the semantic space \( R^o \) into sub regions. The space \( R^o \) is first partitioned into regions \( r_1, r_2, \ldots, r_r \) corresponding to the basic semantic categories \( \epsilon_1, \epsilon_2, \ldots, \epsilon_l \). The regions are merged together based on the hypernym/hyponym relationships. Finally, a multi-level partitioning of \( R^o \) is constructed, where each partition corresponds to a category in the semantic hierarchy.

3.5 Semantic Content Profiling

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. We 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. 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 \( Q = \{ x_{q1}, x_{q2}, \ldots, x_{qn} \} \) issued by a node \( n_i \), suppose the queries are resolved at node \( n_j \) and the set of query result data objects is \( x(Q, n_j) = \{ x_{j1}, x_{j2}, \ldots, x_{jn} \} \). One can estimate another set of data objects \( x'(Q, n_j, \tau) = \{ x'_{j1}, x'_{j2}, \ldots, x'_{jn} \} \) with confidence more than a predefined threshold \( \tau \), and consider the data content of node \( n_j \) as \( x(Q, n_j) \cup x'(Q, n_j, \tau) \).

In the association-based content estimation, the estimation is obtained through association rules. Let the query history \( I = \{ x^*_1, x^*_2, \ldots, x^*_k \} \) be a complete list of earlier queries where each query \( x_q \) 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 \( I = \{ x^*_1, x^*_2, \ldots, x^*_k \} \) be a complete list of earlier query results (i.e. nearest neighbors). Based on these assumptions, let \( x_{i1} \) and \( x_{i2} \) be two data objects satisfying: (1) \( x_{i1} \in L \), (2) \( x_{i2} \in k-\text{NN}(x_{i1}, I) \), (3) \( k-\text{NN}(x_{i2}, I) \in T \), and (4) \( x_{i1} \neq x_{i2} \). Then an association rule is a repeatedly happened pattern \( r: x_{i1} \rightarrow x_{i2} \), where \( x_{i1} \) and \( x_{i2} \) are called the antecedent and consequence of the association rule \( r \), respectively.

Definition 5: Query Popularity

Given the query history \( L \) and a query \( x_q \), the query popularity is the appearance frequency of \( x_q \) in \( L \): \( \psi(x_q, L) = \text{appearance}(x_q)/|L| \), for \( \forall x_q \in L \). (3) where \( \text{appearance}(x_q) \) 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 \( X \). Therefore, \( L \) is a list of repeated appearances of data objects from \( X \). 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(\( x_{q_k} \), \( X \)) may contain one or more \( x_{q_k}' \)s replicas. However, in the generation of association rules, these replicas should not be considered as consequences of \( x_{q_k} \). Therefore, the association rules are defined as \( r: x_q \mapsto x_i \) where \( x_i \in k\)-NN(\( x_{q_k} \), \( X \)) - \{ \( x_{q_k} \) \}.

**Definition 6:** 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 \( \tau \):

\[
R(L, \tau) = \{ \, r: x_q \mapsto x_i \mid \mu(x_q, L) \geq \tau \, \} \tag{4}
\]

**Definition 7:** Estimated Node Content

Given a mobile node \( n_j \) and a set of queries \( Q = \{ x_{q_1}, x_{q_2}, \ldots, x_{q_n} \} \) that are resolved by data objects from \( n_j \), the estimated node content of \( n_j \) can be defined as follows:

\[
x(\tau) = \{ x_q \mid r: x_q \mapsto x_h, \text{ for } \forall x_h \in Q \land x_h \in k\)-NN(\( x_{q_k} \), \( X \)) \land r \in R(L, \tau) \} \tag{5}
\]

In Definition 7 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 \( x_{q_k} \in Q \), we can obtain \( x_{q_k} \) as one of the nearest neighbors of it. Therefore, we are confident that data object \( x_{q_k} \) belongs to the content of node \( n_j \). At the same time, by using the associate rule \( r: x_{q_k} \mapsto x_h \), we can also estimate that data object \( x_h \) is within the data content of node \( n_j \). Thus the data content of node \( n_j \) can be estimated with confidence \( \tau \).

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 \( X = \{ x_1, x_2, \ldots, x_m \} \) 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 \( P(\cdot | x_i) \), which is referred as the fundamental relevance probability. The data objects will then be retrieved with a collection of queries \( Q = \{ x_{q_1}, x_{q_2}, \ldots, x_{q_n} \} \), 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 probability \( P(x = x_i) \forall x_i \in k\)-NN(\( x_{q_k} \), \( X \)).

**Definition 8:** The Bayes’ Rule

Based on above descriptions of content estimation, the Bayes’ rule can be defined as:

\[
P(x \in k\)-NN(\( x_{q_k} \)) | t_1 \ldots t_k = \frac{P(t_1 \ldots t_k | x \in k\)-NN(\( x_{q_k} \))P(x \in k\)-NN(\( x_{q_k} \))}{P(t_1 \ldots t_k)} = \frac{\sum_{i=1}^{k} P(t_i | x = x_i)P(x = x_i)}{\sum_{i=1}^{k} P(t_i | x \neq x_i)P(x = x_i)} \tag{6}
\]

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

**4 CONTENT-BASED CLUSTERING**

**4.1 Clustering Methodology**

The main purpose of our semantic-based multimedia representation is to provide a paradigm that concisely describes the semantic contents of multimedia data objects and cluster the semantically similar ones. Based on this paradigm, the multimedia data contents of each mobile node can be automatically recognized and represented as semantic vectors with strong mathematical backbone. This representation would also allow simple and efficient analysis of similar data objects which assists classification, clustering, and integration of multimedia data. A clustering method that utilizes the aforementioned semantic-based paradigm is given in this section.

**Definition 9:** The Content-Based Cluster

Suppose an ad hoc network \( G \) comprises \( r \) mobile nodes \( n_1, n_2, \ldots, n_r \). 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 \( C_i \) is defined as follows:

\[
C_i = \{ n_j \mid \forall n_k \in C_i, n_k \neq n_j; \forall n_l \in C_i, n_l \neq n_j \} \tag{6}
\]

Based on the content distribution of each mobile node, the ad hoc network is divided into clusters with similar or overlapping multimedia data objects. A new cluster-level content distribution function is generated by describing the contents of mobile nodes within a cluster. A query, with high probability, will be directed and resolved within a specific cluster. This characteristic reduces the search space for a query, which should result in an improved query processing.

The cluster content distribution information is further integrated and fused together to form higher-level clustering, whose contents are represented using distribution functions in larger hyper-rectangle regions. This process is recursively performed until the whole ad hoc network is represented as one cluster. To accommodate the cluster-level content distribution information, a centroid node is selected
for each cluster. The centroid is a node \( n^* \) who has most powerful capabilities, such as computation speed, memory space, communication bandwidth, and etc. In case there is a tie among more than one centroid candidate in a cluster, we choose the node whose data content has maximum overlapping with the cluster-level hyper-rectangle region of this cluster as the centroid node.

**Definition 10: The Cluster Centroid**

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

\[
\Xi(C_i) = \{n' \mid n' \in C_i, \sum_{x=1}^{k} c_x v_x(n') + |\chi(n') \cap \chi(C_i)| \leq \sum_{x=1}^{k} c_x v_x(n') + |\chi(n') \cap \chi(C_i)| \}
\]

where \( c_1, c_2, \ldots, c_k \) are predefined coefficient parameters.

To reduce the searching cost, an indexing hierarchy is built as a wrapper on top of the content-based clusters. The construction of the indexing hierarchy is a two-step process:

1) Within a cluster, the mobile nodes are grouped and one node is selected as the centroid of the cluster. This centroid node acts as the lowest indexing node in the indexing hierarchy — it contains the semantic data contents of the cluster. Connections between centroid node and other nodes provide a virtual infrastructure within the cluster.

2) Based on the virtual infrastructures of clusters, the high-level indexing nodes are created by abstracting the data contents of several clusters. Similarly, each high-level indexing node is mapped to an ad hoc node that has most powerful capabilities among its children nodes.

### 4.2 Scalability and Self-Adaptive Capability

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. However, dynamic nature of the ad hoc network and the dynamic nature of the data contents of the mobile nodes require dynamic maintenance of the proposed indexing scheme to guarantee the optimized and accurate data retrieval.

**Definition 11: The Capacity Entropy**

Given a cluster \( C_i = \{n_{i1}, n_{i2}, \ldots, n_{ir}\} \), let \( V(n_i) = \{v_1(n_i), v_2(n_i), \ldots, v_k(n_i)\} \) denote a vector of the hardware characteristics of node \( n_i \), and \( P = \{c_1, c_2, \ldots, c_k\} \) be a vector of predefined coefficients. The capacity entropy of node \( n_i \), denoted as \( E_c(n_i, C_i) \), can be described as follows:

\[
E_c(n_i, C_i) = P \cdot V(n_i) + |\chi(n_i) \cap \chi(C_i)|
\]

where \( P \cdot V(n_i) \) denotes \( c_1 v_1(n_i) + c_2 v_2(n_i) + \ldots + c_k v_k(n_i) \).

Because each cluster chooses the node with the largest capacity entropy as its centroid, the centroid nodes at top level of the indexing hierarchy should have the largest capacity entropy throughout the cluster. Suppose \( n \) is the current centroid node. If another node \( n \) achieves the largest capacity entropy as a result of content changes, then \( n \) should be the new centroid node.

**Definition 12: The t-Partition**

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 groups 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 \( C_{11}, C_{21}, \ldots, C_{1r} \) based on their data contents. For any cluster \( C_i \), one can select the centroid node of the cluster as follows:

Step 1: Find a node \( n' \) with 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. Step 2: Build 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 12, 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. 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.

In the algorithms 1 and 2, 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 1:** 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 $\Gamma = {C_1, C_2, ..., C_n}$</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>$S,(x)$</td>
<td>The function of generating content summaries of cluster $x$</td>
</tr>
<tr>
<td>$\hat{S}(x)$</td>
<td>The function of generating content similarity (or summary) of clusters in partition $x$</td>
</tr>
</tbody>
</table>

**Algorithm 1: 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 $|\Theta| < t$ then
6. $\Theta \leftarrow \Theta \cup K$
7. $\Gamma \leftarrow \Gamma - K$
8. else
9. $L \leftarrow S,(K) \cap S,(P(\Theta, 1))$
10. for $i = 2$ to $t$
11. if $|S,(K) \cap S,(P(\Theta, i))| > |L|$ then
12. $L \leftarrow S,(K) \cap S,(P(\Theta, i))$
13. $j \leftarrow i$
14. $P(\Theta, j) \leftarrow P(\Theta, j) \cup K$

**Table 2:** Notations related to clustering.

<table>
<thead>
<tr>
<th>Items</th>
<th>Notations</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h$</td>
<td>The height of the hierarchy</td>
</tr>
<tr>
<td>$t(1)$</td>
<td>The expected partitions at level $1$ of the hierarchy</td>
</tr>
<tr>
<td>$P(x, t)$</td>
<td>The function of $t$-partition of elements in set $x$</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 $\nu$</td>
</tr>
<tr>
<td>$P,(x)$</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 2: Constructing hierarchical indexing**

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

The purpose of algorithms 1 and 2 is to build the ESSM indexing 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.

Based on the definitions of indexing hierarchy and cluster capacity entropy, 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 capacity entropies of the other nodes in its cluster. Since the hardware features of a mobile node $n_i$ will keep unchanged for a long period, the only information that needs to be updated is the content summary of node $n_i$, which can be concisely described as a logic expression $S(n_i)$. The updated $S(n_i)$ can be obtained by allowing query packets to piggyback it when forwarded to the centroid node. The detailed update strategy is as follows:

- **Dynamic maintenance:** Each mobile node $n_i$ can process a sequence of queries $Q_1^n$, ..., $Q_m^n$. For any content-based query $Q_i^n$ submitted to node $n_i$, if $Q_i^n$ 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 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.

5. PERFORMANCE 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, …etc. To facilitate content-based retrieval, these factors need to be analyzed in details.

Our analysis is based on the following notations:

- $k$: the number of nodes in the ad hoc network.
- $m$: the minimum number of children for an indexing node (minimum fan-out).
- $P_f$: the probability of a database node joining the network.
- $P_j$: the probability of a node removed from the network.
- $P_m$: the probability of a modify operation.
- $Q$: the query rate.

5.1 Initialization Overhead

In the initialization step, a randomly selected node, say $n_i$, is chosen as the coordinator to construct the indexing hierarchy. The selection of the coordinator node takes $\Theta(k)$ hops. The coordinator needs to send each node a message to notify the coordinator’s location and to collect data content descriptions, which takes $O(k)\cdot k$ hops. Hence, the initialization overhead amortized on each nodes is $\frac{\Theta(k) + O(k)\cdot k}{k} = O(k)$ hops.

5.2 Searching Cost

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

5.3 Maintenance Overhead

As noted before, the proposed indexing hierarchy does not change as long as the data source contents are in tact. As a result, the indexing hierarchy changes if a new data source is inserted/deleted in/from the network or a modification is made to the contents of an existing node. Moreover, 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.

A new node $n_{new}$ joining the network requires at most $k \log_w(k)$ hops to be included in its related cluster. Hence, the average cost for processing new nodes is $2(P_f)Q \cdot k \log_w(k)$ hops. Similarly, the processing of leaving nodes requires $(P_j)Q \cdot k \log_w(k)$ hops. The modification operation can be viewed as a deletion of a node followed by an insertion of a new node. Consequently it requires $((P_m)Q \cdot k \log_w(k) + 2(P_j)Q \cdot k \log_w(k))$ hops. Finally, the average modification cost is $\Theta((3P_m + 2P_j + P_f)Q \cdot k \log_w(k))$.

6 EXPERIMENTAL EVALUATION

In this section, we present the experimental analysis of the proposed content-aware clustering scheme against the flooding-based search scheme [4,5,30]. As noted earlier, in an ad hoc network with distributed data sources, flooding-based blind search strategy may cause extra computational and communication overheads due to the forwarding of the queries to some “irrelevant” nodes. 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.

6.1 Experimental Setup

The experiments were run on the basis of ad hoc network prototype with CMU extension to the ns-2 version 2.26 [29]. However, since ns-2 does not support content-based 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 and randomly setting up the connections between the
nodes. A mixture of operations, including querying, node joining, and node leaving, will then be randomly submitted to the network. The query generation time follows the exponential distribution, which is similar to the previous work [16]. The access pattern in the queries follows Zipf-like distribution, which is widely used to model non-uniformly distributed queries [17]. The input parameters are summarized in Table 3. Most of these parameters are self-explanatory. More details for some parameters are given below.

Table 3: 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 6,536</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>100m^2 – 10,000m^2</td>
</tr>
<tr>
<td>Transmitter range</td>
<td>100m</td>
<td>100m to 1,000m</td>
</tr>
<tr>
<td>Node mobility</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_{max})</td>
<td>10 query/s</td>
<td>1 – 100 query/s</td>
</tr>
</tbody>
</table>

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 moving pattern follows the random waypoint movement model [5]: 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_{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 as the one used in [11], 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 [12] (see table 1). Each image in the dataset is represented as a vector of 435 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 image were used as the candidate pool for query examples.

6.2 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 semantic content of the query (i.e. a semantic vector) is compared with a series of distribution density functions navigated in the ESSM infrastructure to find a cluster that contains the semantically most similar images. In the experiments reported in this section, the randomly generated query issued to a mobile node is resolved as follows:

1) The query object is represented as a semantic vector, which is forwarded to the ESSM infrastructure and compared against the contents of the clusters (hyper-rectangle regions and distribution density functions).

2) By comparing the overlapping of the query search sphere and the cluster-level contents, the ESSM finds a cluster that is semantically most relevant to the query. The centroid node of the selected cluster will then examine the semantic content of the query, and forward it to the mobile multimedia databases that may contain the semantically most similar objects.

3) For a specific mobile multimedia database, the content-based retrieval will employ a comparison metric including both semantic concepts and low-level features to find the nearest neighbors to the query object.

The ESSM shows its adaptation to the ad hoc environment in the experiments. As mentioned in section 2.2, the centralized search strategy achieves good performance; however, it is not suitable for ad hoc networks. Therefore, the ESSM can be proved 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. Figure 3 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
shortcut of a horse (the leftmost image in the bottom row of figure 3). As can be seen from figure 3, 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 CBR systems ESSM retrieves multimedia objects based on the gradually increasing granularities of semantic categories.

- Query Resolution Accuracy

One important metric for content-based 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 3 compares the accuracy of query results returned by search 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 3, the accuracy of “no estimation” also improves as the number of queries increases. 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.

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 4 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 4. 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.

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.
speed is limited to 1m/s and an average of 10 queries per second), the number of hops was limited to 64. Figure 5 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 6: Comparison on search path length (query rate = 10q/s, k = 10).

Figure 7: Comparison on search cost (query rate = 10q/s, k = 10).

Figure 6 illustrates the number of hops needed during the query resolution for various network densities. As can be seen from Figure 6, 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 7).

Figure 7 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 7 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.

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 8 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.

Figure 8: Impact of mobility (query rate = 10q/s, k = 10).

- 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 9 illustrates the result. Similar to our earlier observation (Figure 7) one can conclude that the ESSM is scalable to large network sizes and large number of data objects.

Figure 9: Search cost in large-scale networks. (query rate = 10q/s, k = 10)

- Maintenance Overhead

As mentioned before, 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 10 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.

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

7 CONCLUSIONS

We proposed a novel dynamic content-aware clustering scheme that facilitates content-based multimedia retrieval in ad hoc networks. This scheme is based on a concise descriptive framework – Extended Summary-Schemas Model (ESSM) – to represent the semantic contents of multimedia objects. It has several innovative characteristics such as logic-based content representation and semantic-based clustering of mobile nodes.

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 content-based indexing methodology has the following properties: 1) 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. 2) 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 dramatically relative to the traditional flooding strategy. 3) Our model is dynamic and capable of self-organizing itself as the network status changes. This further offers scalability and robustness in large-scale networks.

8 REFERENCES


