Journal of Mobile Multimedia, Vol. 1, No.3 (2005) 235-254

HIERARCHICAL SEMANTIC-BASED INDEX FOR AD HOC IMAGE RETRIEVAL

BO YANG

The Pennsylvania State University, University Park, U.S.A.
byang@cse.psu.edu

ALI R. HURSON

The Pennsylvania State University, University Park, U.S.A.
hurson@cse.psu.edu

Received May 12, 2005
Revised Aug 11, 2005

Ad hoc networks have received considerable research attention by provisions of wireless communications without location limitations and pre-built fixed infrastructure. Because of the absence of any static support infrastructure, ad hoc networks are prone to several limitations such as bandwidth, connectivity, and power. The traditional content-based image retrieval approaches employed in ad hoc networks may result in either high search cost or low fault tolerance. In this paper, we propose and analyze a decentralized non-flooding image retrieval scheme in multi-hop mobile ad hoc networks ─ Semantic Ad hoc Image Retrieval (SAIR). The novelty of SAIR stems from several factors including: (1) representation of image contents using first-order logic expressions; (2) clustering mobile nodes based on their data contents; (3) organizing image data with a hierarchical semantic-based indexing infrastructure; (4) performing content-based image retrieval within a reduced scope of mobile nodes. Through extensive simulations, we show that relative to the flooding strategy, SAIR can retrieve the semantically most similar image objects by accessing only a small portion of the mobile nodes with much lower search cost. Moreover, it is scalable to large network sizes and large number of data objects.

Key words: Wireless ad hoc network, image retrieval, semantic-based organization

1 Introduction

The rapid development of wireless communication standards such as Bluetooth and WiFi has made image applications more and more popular in mobile environments. Efficient access to image data requires large storage space, more computation capability, and higher network bandwidth. Moreover, indexing on image data is rather difficult, which has been witnessed by the wealth of research literature [1,9,14,16]. In a mobile network, the autonomy and heterogeneity of mobile databases introduce additional complexity to the representation and manipulation of image data, making the task of image data retrieval more challenging.

Ad hoc networks, as a special class of mobile networks, are gaining popularity due to their infrastructure-free and self-organizing characteristics. Due to the lack of infrastructures, mobile nodes may not maintain direct point-to-point connections to other nodes in the system. Hence, most research assumes ad hoc network communications in a hop-by-hop fashion, where nodes behave as routers, forwarding messages for other nodes [2].

Due to the recent advances in visualization techniques, the communications in ad hoc networks are no longer limited to textual information, and other form of information sources such as images are becoming desirable [16]. Within the framework of ad hoc networks, the ability of image retrieval is desirable and using images can enrich the communications between the mobile nodes, making their messages more expressive. For example, one of the most common ad hoc network applications is building temporary network among a rescue team in a disaster area. Each rescuer needs to make correct judgements and take actions swiftly based on his/her communications with other ones in the team. However, some instances, such as the damages, may not be easily described using only textual information. Consequently, there is a great need for novel methods that facilitate efficient management of image data. It should also be noted that 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) [32], technically allowing the communication of image information.

The previous research in ad hoc networks does not provide efficient solutions to image data accessing. With a very few exceptions [17,19], search techniques are based on centralized or flooding strategies, ignoring the content distribution among mobile nodes [3,4,5]. Centralization and flooding strategies make it possible to handle image data retrieval; however, they may also result in either low fault tolerance or high search cost. Generally, the challenges of ad hoc networks on image retrieval can be categorized as follows:

First, in an ad hoc network, the mobile nodes communicate with each other in a hop-by-hop fashion; however, the paths between these nodes are constantly changing due to the mobility. As a result, the data source nodes are unknown at the requesting node and identification of data source in real-time applications is hard to achieve if traditional routing algorithms are employed [5]. One solution to the content-based queries is traversing the whole network [21]. However, this solution drastically consumes the system resources — memory, network bandwidth, and battery power. To save the system resources, the accurate identification of data sources requires an organizational infrastructure according to data content distribution. Consequently, to facilitate efficient image 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.

Second, scalability and robustness may vary as the network configuration changes. A practical ad hoc network may consist of a centralized data source node (data center) and several client nodes that request data from the data center. However, this infrastructure is not robust, scalable, or fault tolerant ─ the data center behaves as a hotspot and its movement within the area could increase the network traffic. Note that if one node just access a data object from the data center, it is quite possible that nearby nodes try to access the same data object in near future [7]. As a result, means should be developed to reduce the communication and computation of the semantically similar queries. This brings out issues such as data replication, replica control, and coherency among replicated data.

Third, an ad hoc network may consist of autonomous and heterogeneous data sources. Heterogeneity and autonomy require means to pervasively facilitate data integration.

Finally, most of the existing search schemes proposed for ad hoc networks are based on text information and cannot be directly applied to image data [5,15].

This work is intended to address the aforementioned issues by proposing a semantic-based search scheme — Semantic-based Ad hoc Image Retrieval (SAIR). The proposed scheme employs a hierarchical structure that organizes image data based on their semantic contents. In addition, SAIR is based on three innovative ideas: (1) representation of image contents using optimized first-order logic expressions; (2) clustering mobile nodes based on data contents; (3) performing content-based image retrieval within a reduced scope of mobile nodes. Experimental results have shown that in comparison with flooding, SAIR can reduce network traffic and search cost, regardless of the distribution, heterogeneity, and autonomy of data sources.

The remainder of this paper is organized as follows: Section 2 overviews the background materials. Section 3 addresses the semantic clustering and dynamic organization of mobile nodes to facilitate content-based retrieval. Section 4 analyzes the performance of the proposed scheme. Finally, section 5 draws the paper to conclusions.

2 Preliminaries

2.1 Content-Based Retrieval

The importance of image retrieval has motivated considerable research on content-based search schemes. Many of the present approaches employ the feature vectors to facilitate content-based retrieval [1,9]. The features are extracted from image pixels, heuristically or empirically, and combined into vectors according to the application criteria. These features, in various data formats, represent a multi-dimensional feature space in which images can be considered as data points. The approaches proposed on the feature-based representation can be categorized as: The partition-based approaches (e.g. quad-tree [8], k-d-tree [9], and vp-tree [9]) that recursively divide the image data objects (or k-dimensional feature spaces) into disjoint partitions, with clustering or classifying algorithms, while generating a hierarchical indexing structure on these partitions. The region-based approaches (r-tree family [10,11]) employing small regions (either in the form of Minimum Bounding Rectangles or Spheres) to cover the data objects. These approaches work well in retrieving image data in a centralized configuration; however, they rely on an assumption that the whole network is homogeneous and can be organized under a centralized indexing structure.

In an ad hoc network with heterogeneous distributed data sources, traditional information retrieval methods, i.e., centralized or flooding strategies, may not work efficiently mainly because of the changes in network topology.

- Due to the heterogeneity and node mobility, the centralized search strategy is not an efficient choice in performing content-based image retrieval. Moreover, this strategy may also cause single point of failure, which results in low robustness.
- The flooding search strategy achieves good performance only when dealing with text information [5]. Due to the sheer size of the image data, the flooding strategy may drastically consume system resources.

To improve the efficiency of content-based retrieval, we propose a novel content-aware clustering scheme for the organization of image data. The basic concept is as follows:

- In response to the heterogeneity of data sources, the semantic contents of images on each mobile node are extracted and represented using a logic-based content representation that can summarize the image contents. We will discuss this in detail in section 3.
- The mobile nodes are partitioned into clusters based on the semantic similarity of their data contents. These clusters are recursively merged into bigger clusters, and at the same time their content representations are fused together in a hierarchical fashion using gradually more general

expressions at each level of the hierarchy. Hence a hierarchical infrastructure can be constructed based on the data contents of the mobile nodes. This hierarchical organization restricts the query processing within a subset of all mobile nodes in an ad hoc network.

It should be noted that a conceptually similar data organization prototype — *Summary-Schemas Model* (SSM) — has been proposed for multi-database systems [6,7]. However, there are several significant differences between the SSM and our model:

- The SSM was proposed to deal with information in a distributed heterogeneous environment. Its indexing nodes (also called summary-schemas nodes) are connected via wired connections. While in our model, the parent / children relationships between nodes can be constructed using wireless connections. Hence our model does not require physical infrastructures to build the whole system.
- The SSM can be used to provide the global information sharing among data sources of large geographic distances. Our model focuses on providing an efficient search scheme that facilitates content-based image retrieval within a restricted area (i.e. an ad hoc network).
- The SSM was originally proposed to handle text-based information. In contrast, our model is intended to accommodate more complex data objects, such as images.

There are several crucial issues that need to be addressed in the design of our model: 1) the extraction and representation of image semantic contents; 2) the construction of the indexing hierarchy based on the semantic contents of images; and 3) the self-adaptive mechanism that can adjust to the changes of network topologies and data contents. The first issue is closely related to image content extraction and will be addressed in this section. The remaining issues will be discussed in section 3.

2.2 Image Semantics Extraction

In image database 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. Most traditional feature-based image retrieval systems employ three types of features in image representation: color, shape, and texture. Color features are widely used in image retrieval systems for its simplicity and effectiveness. Typical color features include color histogram [27] and color moment [28]. Shape features are employed in distinguishing images when the contour lines evidently profile the visual objects [29]. Texture features are provided as an important tool for image retrieval. A variety of texture analysis methods have been studied in the past years, such as Daubechies wavelet [30]. However, the performance of feature-based systems is far from satisfactory due to the fact that images with similar features may not share common semantic contents, which is known as the *semantic gap* [1].

The object-level features, in contrast, are obtained from the higher-level understanding of the images — the semantic contents of the image data. From a cognitive point of view, images are a communication medium that reflects the natural objects in the real world. Thus the content of an image can be described within a linguistic system capable of finding the objects in the image and deducing the semantics from the relationships of these objects.

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 analysis (LSA) [12, 16].

The segmentation method employed in this work is similar to the binary-partition-tree approach proposed in [13]. Similar pixels are merged together as homogeneous segments, which are enclosed in minimum bounding rectangles (MBR).

The low-level feature representation of image segments cannot represent the semantic contents, and therefore do not provide an ideal basis for semantic-based retrieval. Hence, the LSA is employed to obtain the semantics of these image segments. The LSA uses *singular value decomposition* (SVD) to uncover the hidden semantic relationships between image objects. We use two set of entities to perform latent semantic analysis: the objects whose semantics are already known (i.e. training samples), and the 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 semantic categories $\zeta_1, \zeta_2, \ldots, \zeta_k$, each category includes at least one training sample. Thus for a category ζ_j, we define a feature vector

$$F_j = (f_1^1, f_2^1, \ldots, f_m^1, \ldots, f_1^n, f_2^n, \ldots, f_m^n, t_1^1, t_2^1, \ldots, t_m^1, \ldots, t_1^H, t_2^H, \ldots, t_m^H)'$$

where H is the number of training samples in semantic category ζ_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 ζ_j.

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

2.3 Representing Image Semantics

We proposed a logic-based platform that represents and organizes the image data into layers according to their semantics [16]. As a result, an image can be considered as the combination of a series of elementary entities, each entity representing an object of the basic semantic categories as mentioned in section 2.2.

Definition 1: The Elementary Entity

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

Let $E = g_1 \wedge g_2 \wedge \ldots \wedge g_n$, where $g_i = p_{i1} \vee p_{i2} \vee \ldots \vee p_{im}$ is the disjunctive form of some logic predicates (true/false values) and $p_{i1} \ldots p_{im}$ form a logic predicate set G_i, — In the feature-based image data sets, g_i indicates the i^{th} feature of the elementary entity. The content of an elementary entity can then be defined as:

$$\boldsymbol{E} = \bigwedge_{i=1}^{n} \left(\bigvee_{j=1}^{m} \boldsymbol{p_{ij}} \right), \qquad \forall \boldsymbol{p_{ij}} \in \boldsymbol{G_i}. \tag{1}$$

Note that in any term $g_i = p_{i1} \vee p_{i2} \vee \ldots \vee p_{im}$, there is one and only one true predicate p_{ij}. For instance, if p_{i1}, p_{i2} … p_{im} correspond to all possible color patterns, the semantic content of g_i at any time is a specific color pattern. Since g_i is disjunction of p_{i1}, p_{i2} … p_{im}, the false predicates do not affect the final result. The content of an elementary entity is restricted by its conjunctive terms g_1, g_2 … g_n, which are the extracted features in application domains.

Definition 2: The Image Object

An image object is the union of a series of elementary entities. Given the above definition of elementary entities E_1, E_2, …, E_k, the content of an image object can be defined as:

$$S = opt\left(\bigcup_{i=1}^{k} E_j\right), \tag{2}$$

where *opt*(.) is a function that rewrites a first-order logic expression into a semantically equivalent and shorter form.

The logic-based representation provides a means of automatically obtaining the semantic content of images: For a given image, a series of visual components can be obtained from it through the aforementioned latent semantic analysis. These visual components are then described using elementary entity descriptors, as noted in definition 1. Finally, the image semantic content is represented as the optimized union of the elementary entity descriptors as shown in figure 1. Since the logic expressions are abstract descriptions of image semantic contents, we use a term "*content summary*" to refer to these expressions.

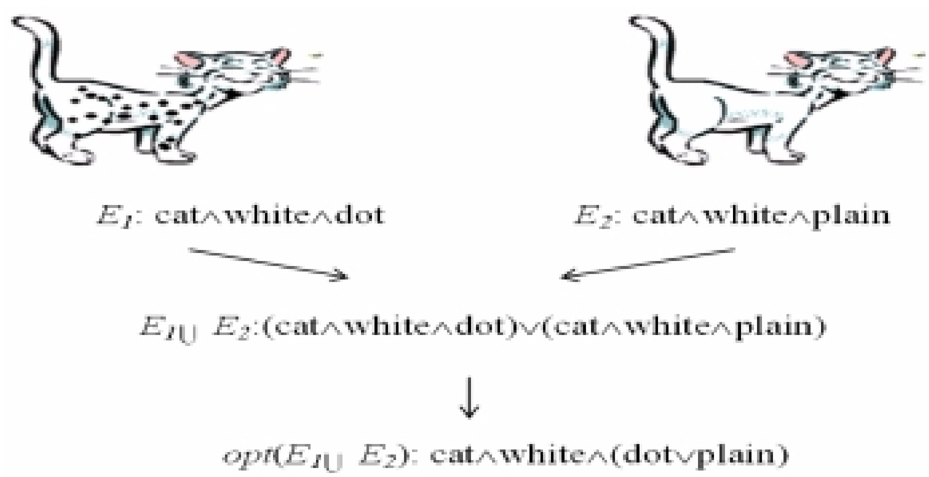

Figure 1 An illustrative example of content summary.

3. Semantic-Based Image Retrieval

3.1 Content-Based Clustering

The semantic-based representation presented in section 2 provides a paradigm to describe image data. Based on this paradigm, contents of the images on mobile nodes can be automatically identified, summarized, and expressed as logic expressions (content summaries). Logic representation of semantic contents of data sources would also allow simple and efficient recognition of similar entities that assists classification and clustering process.

Definition 3: Content-Related Nodes

Suppose a mobile node $\boldsymbol{n_i}$ contains image objects $\boldsymbol{d_1, d_2, \ldots, d_m}$, which are collectively denoted as $\boldsymbol{D}(\boldsymbol{n_i})$. Based on the aforementioned representation of image objects, each mobile node $\boldsymbol{n_i}$ can obtain a content summary $\boldsymbol{S}(\boldsymbol{n_i})$ that abstracts the contents of images in $\boldsymbol{D}(\boldsymbol{n_i})$. Given a pair of nodes $\boldsymbol{n_0}$ and $\boldsymbol{n_1}$, they are content-related *iff*:

$$\boldsymbol{S}(\boldsymbol{n_0}) \oplus \boldsymbol{S}(\boldsymbol{n_1}) \neq \boldsymbol{S}(\boldsymbol{n_0}) \vee \boldsymbol{S}(\boldsymbol{n_1}), \tag{3}$$

where $\boldsymbol{S}(\boldsymbol{n_0}) \oplus \boldsymbol{S}(\boldsymbol{n_1})$ is defined as $(\boldsymbol{S}(\boldsymbol{n_0}) \wedge \sim \boldsymbol{S}(\boldsymbol{n_1})) \vee (\sim \boldsymbol{S}(\boldsymbol{n_0}) \wedge \boldsymbol{S}(\boldsymbol{n_1}))$.

In other words, if nodes $\boldsymbol{n_0}$ and $\boldsymbol{n_1}$ are content-related, they must have some common data entities, i.e. $\boldsymbol{D}(\boldsymbol{n_0}) \cap \boldsymbol{D}(\boldsymbol{n_1}) \neq \emptyset$.

Definition 4: The Content-Based Cluster

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

$$\boldsymbol{C_i} = \{\, \boldsymbol{n_i} \mid \forall\, \boldsymbol{n_j} \in \boldsymbol{C}, \boldsymbol{n_j} \approx \boldsymbol{n_i};\ \forall\, \boldsymbol{n_k} \notin \boldsymbol{C_i}, \boldsymbol{n_k} \neq \boldsymbol{n_i} \} \tag{4}$$

Based on the definitions of content-based clusters, we developed an algorithm that partitions an ad hoc network into clusters. Table1 shows the notations used in the algorithm. Algorithm1 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 1 Notations related to the clustering.

Items	Notations
$V(N)$	The set of mobile nodes $\{n_1, n_2, \ldots, n_k\}$
Γ	The set of clusters
$F(x)$	The function of selecting an element from set x
$C(\Gamma, i)$	The i^{th} cluster in Γ
$S(x)$	The function of generating content summaries of mobile node x

Algorithm1: Content-based clustering

```
Γ ← F(V(N))
K ← V(N) − Γ
while |K| > 0 do
   x ← F(K)
   K ← K − {x}
   for i = 1, …, |Γ|
      if S(C(Γ, i)) ⊕ S (x) ≠ S(C(Γ, i))∨S(x)
         merge x with C(Γ, i)
   if x is not merged with any cluster in Γ
      create a new cluster in Γ
```

3.2 Organizing Clusters into a Hierarchy

As noted in section 3.1, 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 this 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.

Definition 5: 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:

$$S_c(C_i) = \bigcup_{j=1}^{q} S(n_{ij}). \tag{5}$$

Definition 6: 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 of C_i can be defined as $\epsilon(C_i)$:

$$\epsilon(C_i) = \{n^* \mid \forall n' \in C_i, \sum_{s=1}^{k} c_s * v_s(n') + |n' \cap S_c(C_i)| \le \sum_{s=1}^{k} c_s * v_s(n^*) + |n^* \cap S_c(C_i)|\}. \tag{6}$$

where $c_1, c_2, \ldots, c_k$ are predefined coefficients.

Definition 7: 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 Γ is denoted as $S_s(\Gamma)$:

$$S_s(\Gamma) = \bigcup_{\forall C_i, C_j \in \Gamma} (S_c(C_i) \cap S_c(C_j)). \tag{7}$$

Definition 8: 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 $\left| \bigcup_{j=1}^{t} S_s(\Gamma_j) \right|$ is maximal.

Definition 9: The Hypernym/Hyponym Relationship

The semantics of a given cluster C_i is represented as a word, denoted as $\omega(C_i)$. An on-line thesaurus ψ (e.g. Roget's thesaurus or Wordnet) can be used to define the semantic inter-relationships between clusters:

For two given clusters $C_i, C_j \in \Gamma$,

(1) If $\omega(C_i)$ describes a generic concept that includes $\omega(C_j)$, then $\omega(C_i)$ is a hypernym of $\omega(C_j)$, denoted as $\omega(C_i) \succ_{\mathrm{H}} \omega(C_j)$.

(2) If $\omega(C_i)$ describes a specialized concept that is included in $\omega(C_j)$, then $\omega(C_i)$ is a hyponym of $\omega(C_j)$, denoted as $\omega(C_i) \prec_{\mathrm{H}} \omega(C_j)$.

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 8, and represent the semantic contents of each group using more generalized descriptions (i.e. hypernyms). In this work, we maintain an on-line taxonomy that can find the generalized descriptions for cluster-level content summaries of any given group [16]. Steps 1 and 2 are recursively applied until the whole ad hoc network is represented as one group. This grouping 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 Algorithm2 and Algorithm3 for the t-partition and the construction of the hierarchical indexing, respectively.

In the algorithms2 and 3, we address the content-aware organization of mobile databases to facilitate image retrieval in the ad hoc network. In comparison with the traditional centralized content-based indexing models, the proposed hierarchical indexing has the following properties:

1) 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.
2) The indexing hierarchy is overlaid on the mobile nodes, which corresponds to the infrastructure-free nature of ad hoc networks and provides high scalability.
3) 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 and load balancing.
4) 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 2 Notations related to the t-Partition.

Items	Notations
Γ	The set of clusters $\Gamma = \{C_1, C_2, \ldots, C_r\}$
t	The number of expected partitions
Θ	The set of partitions
$F(\Theta, x)$	The function of selecting an element from set x satisfying the minimal overlapping with Θ
$P(\Theta, i)$	The i^{th} partition in Θ
$S_c(x)$	The function of generating content summaries of cluster x
$S_s(x)$	The function of generating content similarity of clusters in partition x

Algorithm2: t-Partition

$\Theta \leftarrow F(\Theta, \Gamma)$
$\Gamma \leftarrow \Gamma - \Theta$
while $|\Gamma| > 0$ *do*
    $K \leftarrow F(\Theta, \Gamma)$
    if $|\Theta| < t$
        $\Theta \leftarrow K$
        $\Gamma \leftarrow \Gamma - K$
    else
        $L \leftarrow S_c(K) \cap S_s(P(\Theta, 1))$
    for $i = 2$ to t
        if $|S_c(K) \cap S_s(P(\Theta, i))| > |L|$
            $L \leftarrow S_c(K) \cap S_s(P(\Theta, i))$
            $J \leftarrow i$
    $P(\Theta, J) \leftarrow P(\Theta, J) \cup K$

Table 3 Notations related to the hierarchical indexing.

Items	Notations
h	The expected height of the hierarchy
$t(i)$	The expected partitions at level i of hierarchy
$P_t(x, t)$	The function of t-partition of elements in x
$\omega(x)$	The function of describing the semantic content of x using a word in the on-line thesaurus ψ
$P_u(x)$	The function of finding the parent node for x
$H(x,y)$	The function of computing the hypernym of the words x and y

Algorithm3: Constructing hierarchical indexing

$\Theta \leftarrow P_t(\Gamma, t(1))$
for $i = 1$ to $t(1)$
    compute $\omega(P(\Theta, i))$
for $level = 2$ to h
    $\Omega \leftarrow P_t(\Theta, t(level))$
    for $j = 1$ to $t(level)$
        $\omega(P(\Omega, j)) \leftarrow \varnothing$
    for $k = 1$ to $t(level - 1)$
        $\omega(P_u(P(\Theta, k))) \leftarrow H(\omega(P_u(P(\Theta, k))), \omega(P(\Theta, k)))$
$\Theta \leftarrow \Omega$

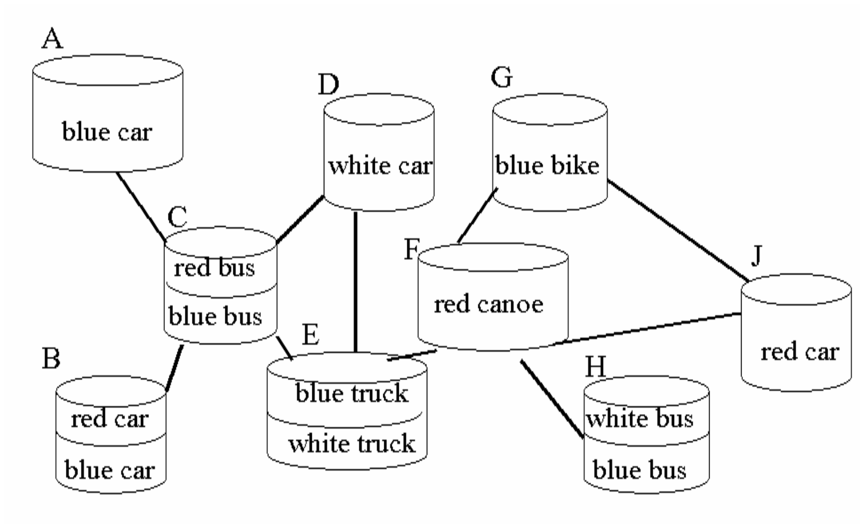

Figure 2 Data distribution in an ad hoc network.

Figure 2 represents a collection of image data objects located irregularly on the mobile nodes in an ad hoc network. As can be observed, the semantically similar contents (say, cars) are scattered on nodes not spatially located together. However, these nodes can be treated collectively as a cluster based on their semantic similarity. Figure 3 illustrates the hierarchical indexing infrastructure obtained from content-based clustering and recursive abstraction of semantic contents.

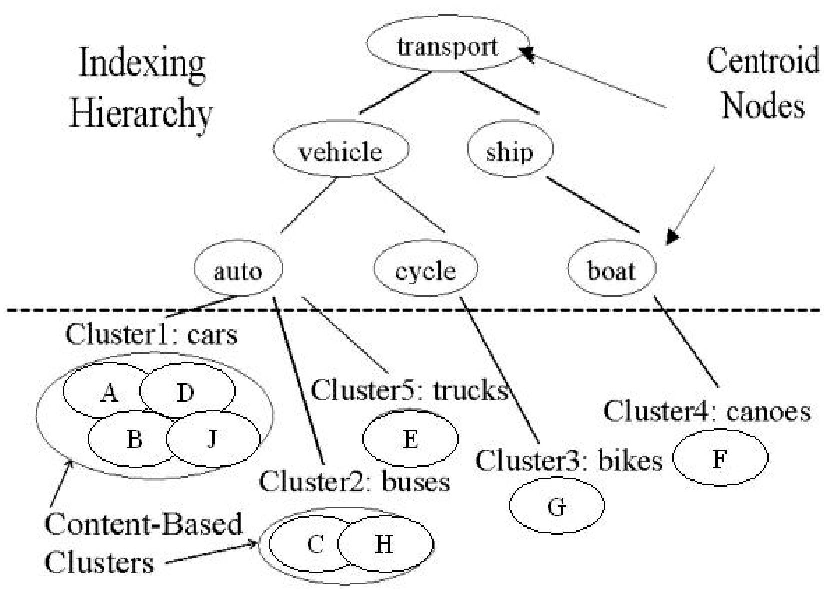

Figure 3 The indexing hierarchy based on clusters.

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

- When a query (i.e. an example image) 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 query to the mobile node containing the corresponding image data; otherwise, keep forwarding the query to higher-level clusters until it is matched.

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 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.

It should be noted that this indexing hierarchy has the self-organizing capability according to the network status. After a period of time, some mobile nodes may break down, while some other nodes may try to join this network. Hence, this indexing hierarchy should make dynamic adjustments to these changes to guarantee the optimal organization and accurate data retrieval. Although some maintenance overhead needs to be paid, this overhead occurs only occasionally, and does not have much impact on the overall performance. Our simulation results show that with the consideration of maintenance overhead, our model still offers better performance than flooding-based strategy.

4. Performance Evaluation

A simulator was implemented in *ns*-2 environment (version 2.26) to evaluate the proposed semantic-based search scheme. Performance of the proposed scheme is compared against the flooding strategy based on various metrics, such as accuracy, search cost, and scalability.

4.1 Simulation Setup

To examine both the accuracy and the scalability, we used two sets of experimental datasets as the testbed: the real dataset and the synthetic dataset as follows:

- The real dataset contains up to 2,048 real images from Corel dataset, which is similar as the dataset used in [31]. In the simulator, 256 color histogram features are extracted from each image for representation purpose. The semantic contents of images are obtained through the latent semantic analysis as discussed in section 2.
- The synthetic dataset comprises 65,536 data points whose feature values are assigned by a random number generator abiding normal distribution in the interval [0, 1].

As mentioned earlier, the data contents of mobile nodes (or clusters) are represented as content summaries, and then integrated in the indexing hierarchy. To allow more flexibility and comprehensive analysis, the simulator relies on a set of input parameters and conditions: The mobile nodes are scattered randomly in an area of 670*m* * 670*m*, moving at speeds randomly selected from [0, v_{max}]. The test data are disseminated randomly on the mobile nodes. The contents of these images are concisely described as content summaries, which are kept at the centroid nodes. The simulator is also capable of building the indexing hierarchy with pre-defined heights and complexities. Table 4 summarizes some of these parameters.

Table 4 Parameters used in the simulations.

Parameter	***Default***	***Range***
Environment size	670m * 670m	$100m^2$ to 10^6m^2
Transmitter range	100m	100m to 1,000m
Bandwidth	1M bps	
Traffic type	constant rate	
Pause time	1s	
Number of nodes	128	1 to 16,384
Node mobility (v_{max})	1 m/s	1 to 50 m/s
Query rate (Q_{rate})	10 query/s	
Query packet size	2 KB	
Data packet size	512 KB	0 – 1 MB
Node cache size	8 MB	0 to 8 MB
Image dataset size	2048	1 to 6,5536
Objects per image	8	1 to 8

4.2 Evaluation Metrics

While the main focus of this work is to improve the search performance of content-based image retrieval in ad hoc networks (e.g. reduced search cost and increased accuracy), we also tried to study the other aspects of the proposed scheme (e.g. scalability and maintenance cost). Our simulations are based on the following metrics:

- **Search cost** is the average number of messages spent on transmitting the request to the data source node.
- **Response time** is the average delay from the query submission to the determination of data source location.
- **Maintenance cost** is the number of messages incurred in the process of adjusting the indexing hierarchy according to network topology and data content changes.
- **Accuracy** is the percentage of the results generated by the decentralized search strategy (i.e., SAIR or flooding) matching with the results generated by the centralized search strategy [15].

4.3 Simulation Results

- **Accuracy**

In terms of accuracy, we ran our simulator on 2,048 real images randomly scattered on 128 mobile nodes. We set the number of nearest-neighbor search to 10. For the two search schemes (SAIR and flooding), we varied the number of visited nodes from 2 to 96 to evaluate the matching percentage of returned content-similar images in comparison with the centralized search results based on latent semantic analysis. Figure 4 illustrates the trends of matching percentage with different numbers of visited nodes. For instance, over 80% of the top 10 images returned by SAIR match with the centralized search results when 48 nodes are visited, under the similar condition the flooding strategy only retrieves 20% of the top 10 images. The superior performance of SAIR in contrast with flooding stems from its content-based clustering capability. The search scope is restricted to a few clusters that are semantically most related with the query, hence SAIR can achieve higher accuracy by accessing only a smaller number of nodes.

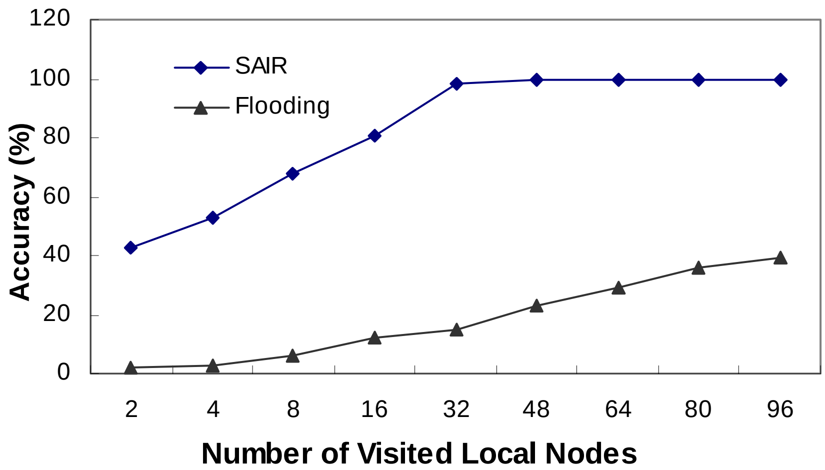

Figure 4 Impact of the number of visited nodes (2,048 images, 128 nodes).

To examine the effect of different indexing hierarchy configurations, we use a parameter ─ m*aximum fan-out* (MF) ─ in the simulation. In general, in the indexing hierarchy with a fixed number of the mobile nodes, the hierarchy's depth decreases as the nodes' fan-out increases, hence reduces the length of search path (in term of the number of visited nodes) to the data sources. In the simulation run depicted in figure 4, the maximum fan-out was set to 10.

In another simulation run, we examined the impact of data density to the search accuracy. Using the same real image dataset, we varied the number of mobile nodes from 512 down to 16 ─ increasing the average number of images per node from 4 to 128, while forcing the simulator to visit only 8 nodes. Figure 5 shows the results. Both strategies achieve higher accuracy when local nodes contain more images. This is due to the fact that more images on local nodes increases the chance of finding nearest neighbors locally. As expected, SAIR outperforms the flooding strategy because of its capability to reduce the search scope to the semantically related local nodes.

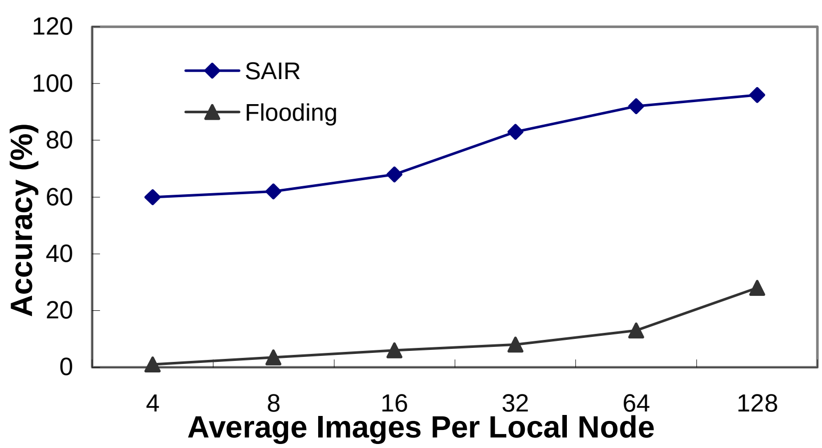

Figure 5 Impact of data density (2,048 images, 8 nodes visited).

- **Response Time and Message Complexity**

In the simulation, we also compared the performance of SAIR and flooding in terms of average query response time and message complexity. A commonly used message complexity metric is the amortized number of messages pushed to the network during the query resolution process [18].

Figure 6 shows the average query response time as a function of the number of nodes in the network. As can be concluded, flooding achieves shorter response time than SAIR. In the flooding scheme, each query message is first broadcast to the neighboring nodes of the requesting node, then copied and re-broadcast throughout the network, therefore a copy of the query message can arrive at the data source node thru the shortest path. In contrast, SAIR first finds the location of data source node with the help of indexing hierarchy, then unicasts the query message to the data source.

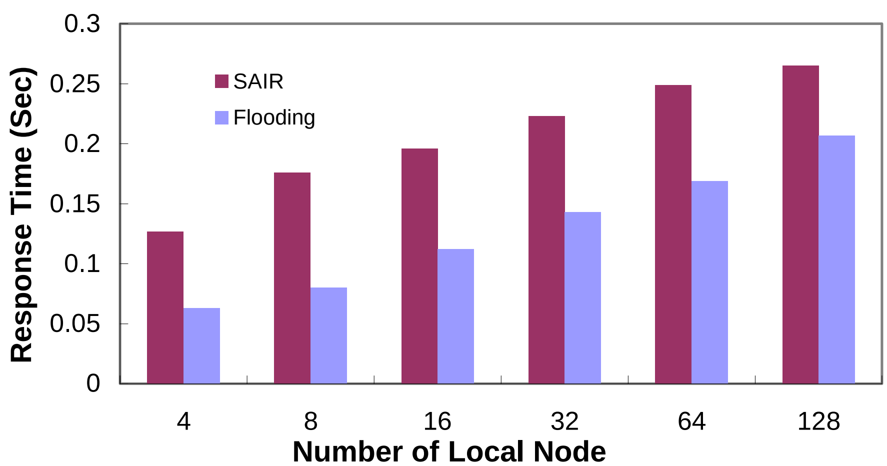

Figure 6 Average response time.

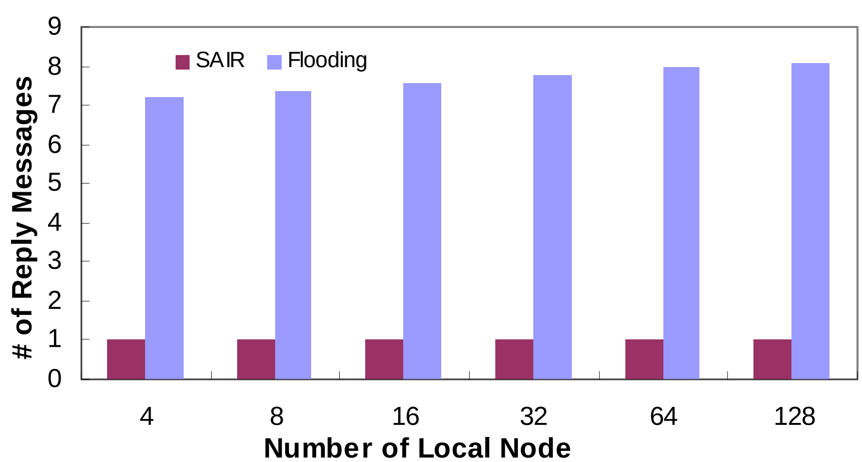

Figure 7 The number of replies per query.

SAIR does not perform as well as flooding in terms of response time; however, it incurs much less message complexity than flooding. As illustrated in figure 7, flooding strategy results in multiple query resolutions. Considering the sheer size of images, the multiple resolutions consume the network bandwidth and mobile node power. In contrast, SAIR offers more efficient processing in terms of system resources. As a final note, the indexing hierarchy also offers better load balancing through dynamic reconfiguration of the network according to the query distribution.

- **Search Cost**

The search cost can be evaluated from two perspectives: 1) the number of local nodes visited to achieve an expect accuracy (in term of matching percentage of results); 2) the average number of hops traversed by the search message to the destination.

Figure 8 illustrates the study of search cost based on the number of local nodes visited. As expected, the flooding strategy needs to visit more nodes to achieve an expected accuracy. Notice that the flooding strategy visits almost all mobile nodes (120 out of 128) to achieve the accuracy over 50%, while SAIR only need to visit less than 10 nodes for the same accuracy.

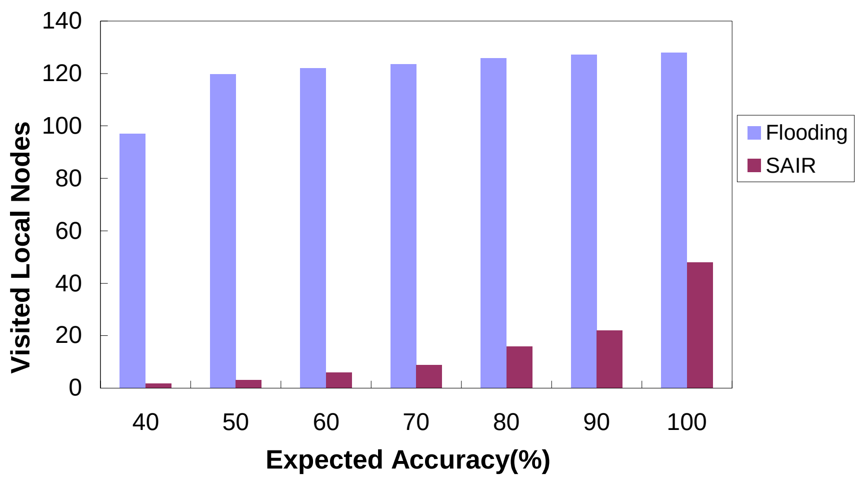

Figure 8 Visited nodes for expected accuracy.

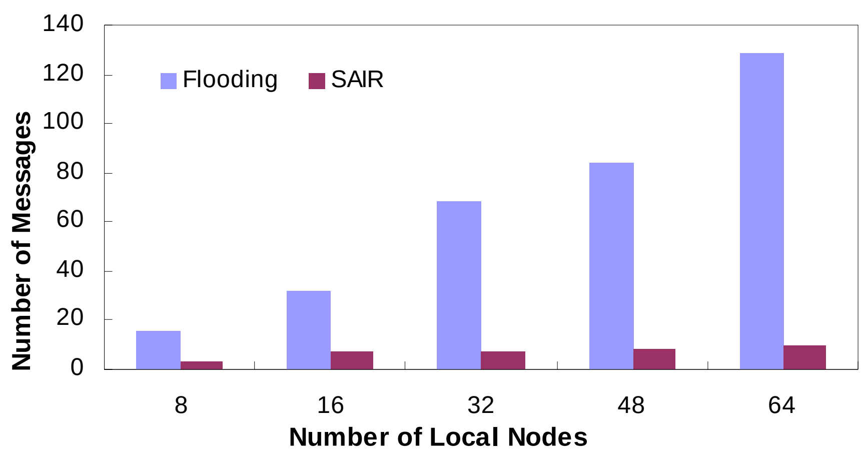

Figure 9 Average number of messages per query.

Figure 9 shows the average number of messages spent on resolving a query. This set of simulation run confirms the capability of SAIR in reducing the search cost. Moreover, the search cost of SAIR only has slight increase as the number of local nodes increases. Hence, SAIR is a low-cost search scheme with steady quality.

- **Scalability**

In terms of scalability to network size, we varied the number of nodes from 2^7 to 2^{12} to evaluate the search complexity and maintenance cost. In this simulation run, we used the synthetic dataset of 65,536 data points.

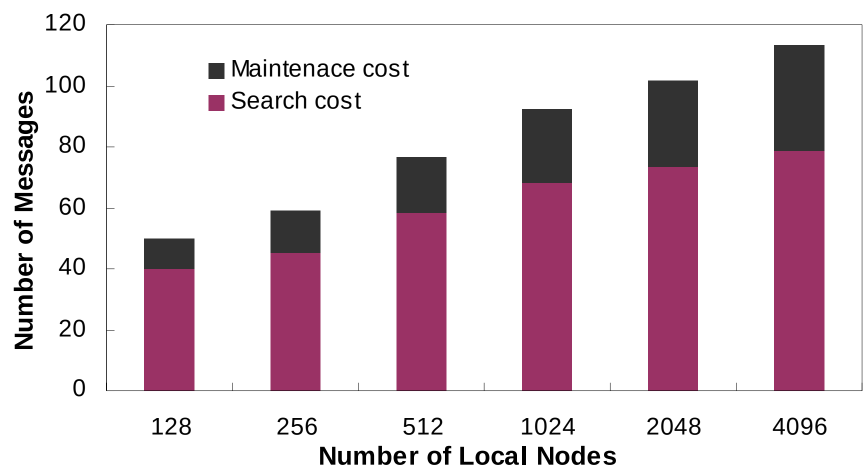

Figure 10 Search and maintenance cost of SAIR (65,536 data points).

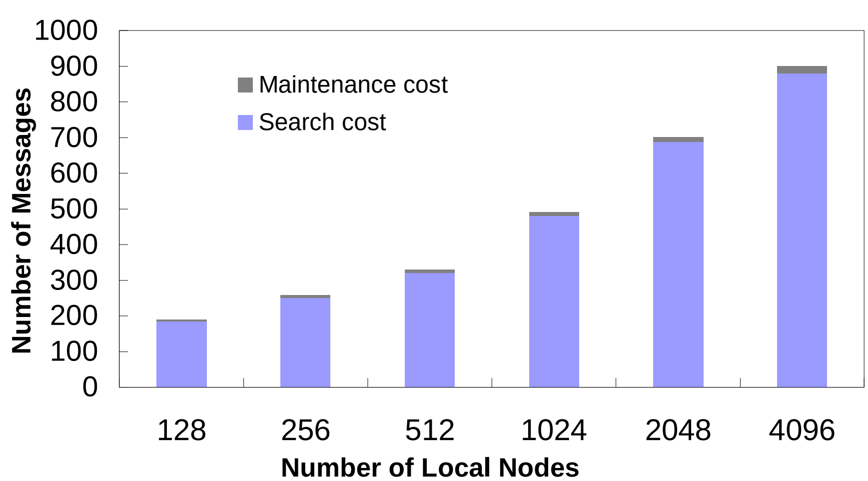

Figure 11 Search and maintenance cost of flooding (65,536 data points).

An important issue of SAIR is the maintenance overhead of its indexing hierarchy due to the changes in network configuration. Whether the benefit obtained from content-based indexing can compensate this overhead should be considered in the large-scale networks, since the data contents change more frequently. In contrast to SAIR, the maintenance overhead is not significant in the flooding strategy. Its maintenance cost only includes messages to update the neighborhood relationship between mobile nodes. The queries are broadcast to every node in the network; hence it is not necessary to keep the proactive indexing information in the flooding scheme. Figure 10 shows the average search cost and the maintenance cost of SAIR as the number of mobile nodes increases. Figure 11 depicts the average search cost and the maintenance cost of the flooding strategy under the same condition. As one can conclude, with the consideration of maintenance overhead, SAIR still provides better performance than flooding.

5. Conclusions

In this paper, we proposed a self-adaptive semantic-based indexing scheme, SAIR, to facilitate content-based image retrieval in ad hoc networks. This scheme 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, scalability, 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. It offers similar accuracy as centralized search scheme while visiting a small portion of mobile nodes.
- We employed semantic-based clustering to organize 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.

We are intended to expand the scope of this work to include multimedia data sources (e.g. audio, video, and text). In addition, we are interested in integrating the proposed method with other mobile environments such as WLAN and cellular networks to allow global sharing among mobile nodes in larger distributed environment.

Acknowledgements

The authors wish to acknowledge the collaborative funding support from the Office of Naval Research under contract N00014-02-1-0282 and National Science Foundation under contract IIS-0324835.

References

1. Auffret, G., Foote, J., Li, C., Shahraray, B., Syeda-Mahmood, T., and Zhang, H., Multimedia access and retrieval: The state of the art and future directions, in Proceedings of ACM International Conference on Multimedia 1999, (1999), 443-445.
2. Broch, J., Maltz, D., Johnson, D., Hu, Y., and Jetcheva, J., A performance comparison of multi-hop wireless ad hoc network routing protocols. in Proceedings of ACM International Conference on Mobile Computing (MOBICOM 1998), (1998), 85-97.

3. Oikonomou, K., Vaios, A., Simoens, S., Pellati, P., and Stavrakakis, I., A centralized ad hoc network architecture (CANA) based on enhanced HiperLAN/2. in Proceedings of IEEE International Conference on Personal, Indoor and Mobile Radio Communications 2003, (2003), 1336-1340.
4. Castro, M., Costa, M., and Rowstron, A., Should we build Gnutella on a structured overlay? ACM SIGCOMM Computer Communication Review, 2004(1), 131-136.
5. Dheap, V., Munawar, M., and Ward, S., Parameterized neighborhood based flooding for ad hoc wireless networks. in Proceedings of IEEE International Conference on Militory Communications 2003, (2003), 1048-1053.
6. Bright, M., W., Hurson, A., R., and Pakzad, S., A taxonomy and current issues in multidatabase systems. IEEE Computer, 1992, 25(3), 50–60.
7. Yang, B., and Hurson, A., R., Ad hoc image retrieval using hierarchical semantic-based index, in Proceedings of IEEE International Conference on Advances in Network Applications 2005, (2005), 629-634.
8. Samet, H., The Quadtree and related hierarchical data structures. ACM computing surveys, 1984, 16(2), 187-260.
9. Fu, A., W., Chan, P., M., Cheung, Y., and Moon, Y., S., Dynamic VP-tree indexing for n-nearest neighbor search given pair-wise distances. VLDB Journal, 2000, 9(2), 154–173.
10. Beckmann, N., Kriegel, H., Schneider, R., and Seeger, B., The R*-tree: An efficient and robust access method for points and rectangles, in Proceedings of ACM International Conference on Management of Data 1990, (1990), 322-331.
11. Katayama, N., and Satoh, S., The SR-tree: an index structure for high-dimensional nearest neighbor queries, in Proceedings of ACM International Conference on Management of Data 1997, (1997), 369-380.
12. Li, D., Dimitrova, N., Li, M., and Sethi, I., K., Multimedia content processing through cross-modal association. in Proceedings of ACM International Conference on Multimedia 2003, (2003), 604-611.
13. Salembier, P., and Garrido, L., Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval. IEEE Transaction on Image Processing, 2000, 9(4), 561-576.
14. Liu, C., Gabor-based kernel PCA with fractional power polynomial models for face recognition. IEEE Transaction on Pattern Analysis and Machine Intelligence, 2004(26), 572-581.
15. Tang, C., Xu, Z., and Dwarkadas, S., Peer-to-peer information retrieval using self-organizing semantic overlay networks. in Proceedings of ACM Conference of the Special Interest Group on Data Communication 2003, (2003), 175-186.
16. Yang, B., and Hurson, A., R., An extendible semantic-based content representation method for distributed image databases, in Proceedings of IEEE Symposium on Multimedia Software Engineering 2004, (2004), 222-226.
17. Woodrow, E., and Heinzelman, W., SPIN-IT: a data centric routing protocol for image retrieval in wireless networks. in Proceedings of IEEE International Conference on Image Processing 2002, (2002), 913-916.
18. Lau, W., Kumar, M., and Venkatesh, S., A coopera-tive cache architecture in supporting caching multimedia objects in MANETs, in Proceedings of International Workshop on Wireless Mobile Multimedia 2002, (2002), 237-243.
19. Cao, G., A Scalable low-latency cache invalidation strategy for mobile environments. IEEE Transaction on Knowledge and Data Engineering, 2003, 15(5), 1251-1265.
20. Beyer, K., Goldstein, J., Ramakrishnan, R., and Shaft, U., When is "nearest neighbor" meaningful? in Proceeding of the 20th International Conference on Very Large Data Bases 1994, (1994), 487-499.
21. Bhattacharya, I., Parthasarathy, S., and Kashyap, S., R., Similarity searching in peer-to-peer databases. in Proceedings of IEEE International Conference on Distributed Computing Systems 2005, (2005), 329-338.

22. Gionis, A., Gunopulos, D., and Koudas, N., Efficient and tunable similar set retrieval. in Proceedings of ACM Conference on Management of Data 2001, (2001), 132-143.
23. Sarwar, B., Karypis, G., Konstan, J., and Riedl, J., Analysis of recommendation algorithms for e-commerce. in Proceedings of ACM International Conference on Electronic Commerce 2000, (2000), 158–167.
24. Aggarwal, C., Wolf, J., and Yu, P., S., A new method for similarity indexing of market basket data. in Proceedings of ACM International Conference on Management of Data 1999, (1999), 407-418.
25. Tousidou, E., Nanopoulos, A., and Manolopou-los, Y., Improved methods for signature tree construction. Journal of Computing, 2000, 301-314.
26. Nanopoulos, A., Katsaros, D., and Manolopou-los, Y., A data mining algorithm for generalized web prefetching. IEEE Transaction of Knowledge and Data Engineering, 2003, 15(5), 1155-1169.
27. Swain, M., and Ballard, D., Color indexing. International Journal of Computer Vision, 1991, 7(1), 11-32.
28. He, J., Li, M., Zhang, H., Tong, H., Zhang, C., Manifold-ranking vased image retrieval, in Proceedings of ACM International Conference on Multimedia 2004, (2004), 9-16.
29. Schmid, C., Selection of Scale-Invariant Parts for Object Class Recognition. in Proceedings of IEEE International Conference on Computer Vision 2003, (2003), 634-640.
30. Wang, J., Z., Wiederhold, G., Firschein, O., and Sha, X., W., Content-based image indexing and searching using Daubechies's wavelet. International Journal of Digital Libraries, 1998, 1(4), 311-328.
31. He, X., King, O., Ma, W., Y., Li, M., J., and Zhang, H., J., Learning a locality preserving subspace for visual recognition. in Proceedings of IEEE International Conference on Computer Vision 2003, (2003), 385-393.
32. Luo, H., Ramjee, R., Sinha, P., Li, L., and Lu, S., UCAN: A unified cellular and ad hoc network architecture. in Proceedings of ACM International Conference on Mobile Computing 2003, (2003), 353-367.