On the Content Predictability of Cooperative Image Caching in Ad Hoc Networks

Bo Yang, Ali R. Hurson, and Yu Jiao
Department of Computer Science and Engineering
The Pennsylvania State University, University Park, PA 16802-6106
{byang,hurson,yjiao}@cse.psu.edu

Abstract

Emerging applications of mobile data management, such as content-based image retrieval in ad hoc networks, require the awareness of content distribution to enact and optimize the communication and search tasks. Caching is a widely used approach in mobile environments to improve system performance and keep track of replications. Within the scope of mobile and ubiquitous computing infrastructure, traditional caching techniques are not effective for large-size data such as images due to the limitations of bandwidth, storage, and power. The functionality of caching techniques relies on exact match, making them unsuitable for imprecise and similarity-based queries. In addition, the description of cached contents is defined based on the query context instead of data content, which fails to exploit the semantic locality of cached data and makes the traditional caching techniques inefficient in utilizing cache storage. In this paper, we present a semantic-aware caching scheme (SAIC) for image databases in ad hoc networks. The proposed scheme is designed based on several novel ideas: 1) multi-level partitioning of semantic space, 2) constraint-based representation of image semantics, 3) non-flooding query resolution, and 4) adaptive cache consistency maintenance. Our combination of theoretical analysis and simulation show the efficiency of the proposed scheme, and evaluate its performance in comparison against two ad hoc caching schemes as advanced in the literature.

1. Introduction

With the advent of mobile networks and ubiquitous computing, ad hoc networks are becoming popular in environments where the infrastructures are either destroyed or too expensive to be built. Most of the previous research focuses on routing protocols adaptive to the dynamic network topologies [1], and relatively few works have been reported on data processing [2]. The study of routing is important for successful communications; nevertheless, data processing is an equally significant issue in the applications of ad hoc networks, since the ultimate task of a network is to support data sharing and to allow timely and reliable access to the information. Consequently, there is a great need for methods that facilitate efficient accessing of voluminous data in ad hoc networks.

One important data processing application for ad hoc networks is content-based image retrieval (CBIR). Due to the advances in visualization techniques, more and more information is represented as images instead of plain texts. Using images can enrich the communications between the nodes in an ad hoc network, making their messages more expressive. We use an example to show the necessity of accessing image data in ad hoc networks.

Example 1 – Consider the information sharing of a rescue team in a flooded area. Messages transmitted between team members may include descriptions of wounded victims, damaged buildings, and dangerous circumstances. Correct judgments and actions need to be taken swiftly based on the situations described in the messages. However, some situations, e.g. the wound of a victim, cannot be easily described using only textual expressions. Moreover, the victims in the flooded area may have similar wound, hence similar queries may be issued continuously. In such a case, the capability of handling image data is of great importance.

Ad hoc networks have both advantages and disadvantages in dealing with image data. In contrast with the lower-bandwidth wide-area wireless networks such as cellular networks (100Kbps for GRPS and 384Kbps for W-CDMA), ad hoc networks comparatively offer higher bandwidth (11Mbps for IEEE 802.11b and up to 54Mbps for IEEE 802.11a and 802.11g) [1]. In addition, ad hoc networks do not rely on infrastructures to support node communications. However, this flexible infrastructure-free characteristic also complicates the process of image data access: The network topology is constantly changing due to node mobility. When a content-based query (e.g. nearest-neighbor image query) is issued, the data source nodes are unknown at the requesting node. As a result, traditionally data-retrieval algorithms rely on flooding strategy to facilitate data access processing [3]. The flooding approach drastically consumes system resources — storage, bandwidth, and energy. Considering the sheer size of the images, the performance deterioration is more drastic. Consequently, ad hoc networks cannot utilize classical content-based image retrieval methods that are based on centralized or flooding mechanisms. To overcome this difficulty, this paper describes a semantic-aware image caching scheme (SAIC). We address the fundamental problem of supporting nearest-neighbor retrieval within the network, in which each node caches a semantic content of earlier queries. By analyzing the cache content, an overview of data distribution in the network is obtained, and the later queries are resolved with optimized search cost.

The rest of this paper is organized into five sections: Section 2 introduces the background knowledge and related work. Section 3 outlines the preliminary concepts. Section 4 introduces the caching rationale. Section 5 evaluates the proposed scheme using experimental analysis. Section 6 draws the paper into conclusions.

2. Background

2.1. Image Representation

Image representation provides the foundation for CBIR. Traditional feature-based image retrieval systems employ three types of features in image representation: color, shape, and texture [4]. 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* [5].

To bridge or narrow the semantic gap, one approach is to devise automatic semantic learning functions that map low-level feature space to high-level semantic space [4]. Based on the principles of semantic learning, the methods can be categorized as inductive and transductive ones [4]: 1) The goal of inductive methods is to create a classifier that identifies some training images based on semantic contents (e.g. annotations) and generalizes well on images without annotations. A widely used inductive model is support vector machine (SVM) [6]. 2) Transductive methods aim at accurately predicting the semantic relevance of the non-annotated images which are attainable during the training process. Methods belonging to this category include latent semantic analysis (LSA) [7], principal component analysis (PCA) [4], and locality preserving projection (LPP) [7].

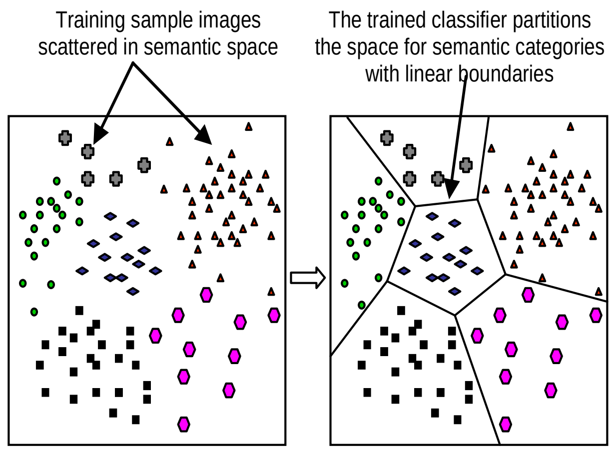

Figure 1: An example of semantic categories.

The inductive and transductive methods provide techniques for constructing and training classifiers that are capable of dividing the image data set (or semantic space) into regions with linear boundaries, where each region corresponds to a category of semantically similar images. Figure 1 illustrates an example of semantic categories in the semantic space.

The partitioning of semantic space provides a means of representing and organizing images based on their semantic contents. Given a collection of semantically similar images, one can collectively represent them using the description of semantic categories. Based on this observation, a semantic caching scheme will be proposed in this paper to facilitate content based image retrieval in ad hoc networks.

2.2. Ad Hoc Caching

Caching has been widely used in mobile environment to reduce network traffic and deal with disconnections. Most of the previous study of caching for ad hoc networks focused on the efficient exploration of routing information [1] with only a few caching schemes to address the data retrieval issue [2].

The data caching in ad hoc networks is a natural extension of the caching schemes in wired networks ─ to keep a copy of the data items that have recently been accessed. Traditional schemes let a mobile node cache either the results of its recent queries or the data that have been forwarded though it to other nodes [2]. The data caching scheme proposed in [8] allows the caching of queries as the semantic descriptions of the cached data. Such a caching scheme is efficient only for small-size data items, and cannot effectively deal with large-size data such as images in mobile databases. In general, the classical semantic caching approaches have three major drawbacks in dealing with image retrieval: First, the traditional techniques rely on exact match of data items; however, image queries are usually imprecise and similarity-based. This characteristic of image retrieval makes it difficult for the traditional techniques to exploit the cached data. Second, the semantic description of the cached contents is obtained in the context of queries, and any change of queries will cause reorganization of the cache, which leads to the inefficient utilization of the cache space. Third, the semantic description does not reflect the popularity of data, making it inefficient in providing QoS-related services.

Path caching is another application of caching in ad hoc networks ─ to record a path to the data source. The scope of path caching was further extended to the domain of data replica allocation in [9]. The CachePath scheme proposed in [2] dynamically caches the path information of passing-by data. These schemes, in general, consider the data items as independent entities and do not utilize the semantic locality among them. As a result, they do not explore content distribution in the ad hoc network.

The work presented in this paper differs from the previous efforts since it is intended to devise a caching scheme that facilitates the content-based image retrieval in a dynamic distributed environment such as an ad hoc network.

3. Problem Formulation

3.1. Overview

In this section, we introduce the formalized problem description of content-based image retrieval in ad hoc networks and the optimization goals in the context of the problem description. We assume 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.

The problem of content-based image retrieval in ad hoc networks can be viewed as follows: Given a set of images $\boldsymbol{X} = \{\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_m}\}$ disseminated among a collection of mobile nodes $\boldsymbol{N} = \{\boldsymbol{n_1}, \boldsymbol{n_2}, \ldots, \boldsymbol{n_r}\}$, a query image $\boldsymbol{x_q}$, and an integer $\boldsymbol{k}$, find the minimum subset $\boldsymbol{N}^*$ = $\{\boldsymbol{n_1}^*, \boldsymbol{n_2}^*, \ldots, \boldsymbol{n_s}^*\} \subset \boldsymbol{N}$ containing $\boldsymbol{k}$ images with smallest semantic distances to $\boldsymbol{x_q}$.

Without loss of generality, we assume the images $\boldsymbol{x_1}$, $\boldsymbol{x_2}$, …, $\boldsymbol{x_m}$ are represented as data points in a n-dimensional semantic space $\boldsymbol{R^n}$. The semantic similarity between two images is defined based on the Euclidean distance between their corresponding data points in $\boldsymbol{R^n}$.

Definition 1: Nearest-neighbor retrieval (k-NN)
Given an image set $\boldsymbol{X} = \{\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_m}\}$ and a query image $\boldsymbol{x_q}$, the nearest-neighbor retrieval of $\boldsymbol{x_q}$ within $\boldsymbol{X}$, denoted as k-NN($\boldsymbol{x_q}$, $\boldsymbol{X}$), is the following set:

$$k\text{-NN}(\boldsymbol{x_q}, \boldsymbol{X}) = \{\boldsymbol{x_i} \mid \forall \boldsymbol{y} \notin k\text{-NN}(\boldsymbol{x_q}, \boldsymbol{X}),\ dist(\boldsymbol{y}, \boldsymbol{x_q}) \geq dist(\boldsymbol{x_i}, \boldsymbol{x_q}) \wedge |\, k\text{-NN}(\boldsymbol{x_q}, \boldsymbol{X})| = k\} \quad (1)$$

where $dist(.)$ denotes the semantic distance between images.

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

Definition 2: Semantic category
An n-dimensional semantic space $\boldsymbol{R^n}$ can be partitioned into a collection of orthogonal regions, which are referred to as semantic categories of images $\boldsymbol{\epsilon_1}, \boldsymbol{\epsilon_2}, \ldots, \boldsymbol{\epsilon_t}$.

As mentioned in section 2, the semantic categories are deducted from a training sample $\overline{X}$ = $\{\overline{x_1}, \overline{x_2}, \ldots, \overline{x_n}\}$ that satisfies $\overline{X} = \bigcup_{i=1}^{t} \boldsymbol{\epsilon_i}$. Let $\boldsymbol{\delta}(\boldsymbol{\epsilon_i}, \overline{X})$ represent the sample data points in category $\boldsymbol{\epsilon_i}$, then the corresponding region of $\boldsymbol{\epsilon_i}$ in the semantic space $\boldsymbol{R^n}$, denoted as $\boldsymbol{\zeta}(\boldsymbol{\epsilon_i})$, can be viewed as the locus of points whose semantic distance to the data points in $\boldsymbol{\delta}(\boldsymbol{\epsilon_i}, \overline{X})$ is smaller than to those in any other semantic category.

Given an image $\boldsymbol{x_i}$ and a semantic category $\boldsymbol{\epsilon_j}$, if $\boldsymbol{x_i}$ belongs to $\boldsymbol{\epsilon_j}$, their relationship is denoted as $\boldsymbol{x_i} \in \boldsymbol{\zeta}(\boldsymbol{\epsilon_i})$.

Definition 3: Inner-category k-NN
Given an image $\boldsymbol{x_q}$ and a semantic category $\boldsymbol{\epsilon_j}$, the inner-category k-NN of $\boldsymbol{x_q}$ within category $\boldsymbol{\epsilon_j}$ is a set:

$$k\text{-NN}_c(\boldsymbol{x_q}, \boldsymbol{\epsilon_j}) = \{\boldsymbol{x_i} \mid \forall \boldsymbol{y} \notin k\text{-NN}_c(\boldsymbol{x_q}, \boldsymbol{\epsilon_j}),\ dist(\boldsymbol{y}, \boldsymbol{x_q}) \geq dist(\boldsymbol{x_i}, \boldsymbol{x_q}) \wedge \boldsymbol{x_i} \in \boldsymbol{\zeta}(\boldsymbol{\epsilon_j})\ \} \quad (2)$$

3.2. Multi-Level Semantic Description

A hierarchical representation model is used to reduce the search space to a subset of categories when $|k\text{-NN}_c(\boldsymbol{x_q}, \boldsymbol{\epsilon_j})| \neq k$. 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 scope with common semantic characteristics. The interrelationship between semantic categories is defined as follows.

Definition 4: Hypernym/Hyponym relationship

The semantics of a given category $\boldsymbol{\epsilon_i}$ could be annotated as $\boldsymbol{\omega}(\boldsymbol{\epsilon_i})$. Then an on-line thesaurus $\boldsymbol{\psi}$ (e.g. Roget's thesaurus or Wordnet) can be used to define the inter-relationship between semantic categories:

For two given semantic categories $\boldsymbol{\epsilon_i}$ and $\boldsymbol{\epsilon_j}$,

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

(2) If $\boldsymbol{\omega}(\boldsymbol{\epsilon_i})$ describes a specific concept that is included in $\boldsymbol{\omega}(\boldsymbol{\epsilon_j})$, then $\boldsymbol{\omega}(\boldsymbol{\epsilon_i})$ is a hyponym of $\boldsymbol{\omega}(\boldsymbol{\epsilon_j})$, denoted as $\boldsymbol{\omega}(\boldsymbol{\epsilon_i}) \prec_{\mathrm{H}} \boldsymbol{\omega}(\boldsymbol{\epsilon_i})$.

It can be proven that the hypernym/hyponym relationship is partial order, which shows the "inclusion" relationship between semantic categories. Based on this definition, we can construct a hierarchical semantic organization for semantic categories.

Definition 5: Semantic hierarchy

Given a set of orthogonal semantic categories $\boldsymbol{\Omega} = \{\boldsymbol{\epsilon_1}, \boldsymbol{\epsilon_2}, \ldots, \boldsymbol{\epsilon_t}\}$ and an on-line thesaurus $\boldsymbol{\psi}$, the semantic hierarchy $\boldsymbol{H_S}(\boldsymbol{\Omega}, \boldsymbol{\psi})$ is defined as the Hasse Diagram of $(\boldsymbol{\Omega}, \prec_{\mathrm{H}})$.

3.3. Data Contents of Mobile Nodes

As mentioned before, the images $\boldsymbol{X} = \{\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_m}\}$ are disseminated among a collection of mobile nodes $\boldsymbol{n_1}, \boldsymbol{n_2}, \ldots, \boldsymbol{n_r}$. Let $\boldsymbol{D^c}(\boldsymbol{n_i})$ denote the images in node $\boldsymbol{n_i}$, as a result, $\bigcup_{i=1}^{r} \boldsymbol{D^c}(\boldsymbol{n_i}) = \boldsymbol{X}$. The distribution pattern of the images over the nodes can be considered as a many-to-many relationship, i.e., each image may be distributed among multiple nodes, and each node may contain a collection of images. Based on this observation, we propose to represent the images in $\boldsymbol{D^c}(\boldsymbol{n_i})$ using the combination of semantic categories and the minimum bounding region of the images. Here we define the concept of vicinity constraint.

Definition 6: Vicinity constraint

Given a set of images $\boldsymbol{X^*} = \{\boldsymbol{x_1^*}, \boldsymbol{x_2^*}, \ldots, \boldsymbol{x_h^*}\}$, each image $\boldsymbol{x_i}$ is represented as a vector of semantic attributes $\boldsymbol{v_i} = (\boldsymbol{a^i_1}, \ldots, \boldsymbol{a^i_n})$. The vicinity constraint $\boldsymbol{C^v}(\boldsymbol{X^*})$ is a collection of constraints showing the n-dimensional minimum bounding region of $\boldsymbol{x_1^*}, \boldsymbol{x_2^*}, \ldots, \boldsymbol{x_h^*}$:

$$\boldsymbol{C^v}(\boldsymbol{X^*}) = ([min\{a^1_1, \ldots, a^h_1\}, max\{a^1_1, \ldots, a^h_1\}], \ldots, [min\{a^1_n, \ldots, a^h_n\}, max\{a^1_n, \ldots, a^h_n\}]) \quad (3)$$

Definition 7: Node content descriptor

Given a node $\boldsymbol{n_k}$ and a semantic category $\boldsymbol{\epsilon_i}$, if every image $\boldsymbol{x_j}$ in $\boldsymbol{D^c}(\boldsymbol{n_k})$ satisfies $\boldsymbol{x_j} \in \zeta(\boldsymbol{\epsilon_i})$, then the node content descriptor $\xi(\boldsymbol{n_k})$ is denoted as:

$$\xi(\boldsymbol{n_k}) = \zeta(\boldsymbol{\epsilon_i}) \cap \boldsymbol{C^v}(\boldsymbol{D^c}(\boldsymbol{n_k})) \quad (4)$$

The node content descriptor as defined can be used to represent a collection of images as follows. Given a set of images $\boldsymbol{X^*} = \{\boldsymbol{x_1^*}, \boldsymbol{x_2^*}, \ldots, \boldsymbol{x_h^*}\}$, first find their semantic categories as described in definition 2. In each category, use the intersection of category region and vicinity constraint to describe a tightly bounding region that encloses the images. Figure 2 shows an illustrative example of the node content description.

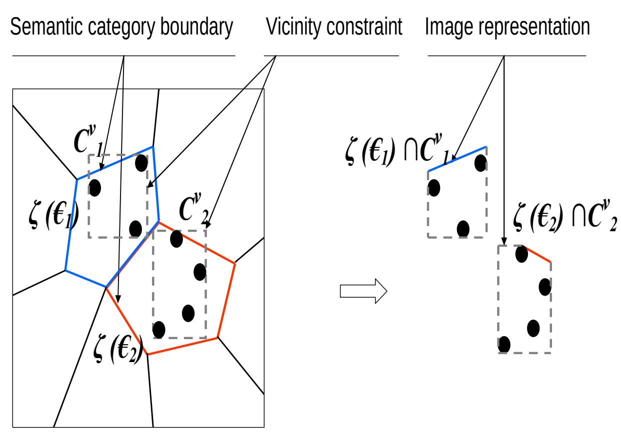

Figure 2: An illustrative example of a node content.

4. Semantic-Aware Image Caching

A semantic-aware image caching scheme (SAIC) is first presented in this section and then we investigate how to process CBIR in such an organization. We also examine how to effectively utilize cache storage with respect to the QoS requirements.

4.1. Caching Rationale

The basic idea of the caching scheme proposed in this paper, called Semantic-Aware Image Caching (SAIC), is to allow each node in an ad hoc network to gradually record semantic descriptions of the image query results passing by it. The query resolution is performed through the cooperation of two phases — query forwarding and cache updating:

Query forwarding: Initially, the local caches are empty and every query, if not resolved locally, is forwarded to other nodes for appropriate data result. To minimize the number of messages spent on query forwarding, we use *Grid Location Service* (GLS) that tracks the location information of mobile nodes [5]. Figure 3 gives an illustrative example of query forwarding. The geographic space is divided into a collection of predetermined squares, each one containing a set of nodes. A query $\boldsymbol{Q}$, starting from the requesting node $\boldsymbol{n_r}$, is forwarded between the grids following the spiral pattern. The spiral curve keeps growing until the data source node $\boldsymbol{n_s}$ is found and the query result is forwarded back thru the shortest path between $\boldsymbol{n_s}$ and $\boldsymbol{n_r}$. Here we have an observation of the query forwarding process: Let $\boldsymbol{d}$ denote the distance between $\boldsymbol{n_r}$ and $\boldsymbol{n_s}$, then all the nodes taking part in the query forwarding are within the sphere $\sigma(\boldsymbol{n_r}, \boldsymbol{d})$ centered at $\boldsymbol{n_r}$ with a radius $\boldsymbol{d}$. In addition, each node only forwards the query along the spiral curve instead of rebroadcasting to all directions. Based on this observation, we can claim that the message complexity is restricted to linear order of the number of nodes, which is much smaller than that of flooding.

Cache updating: When query $\boldsymbol{Q}$ is forwarded between the nodes, its semantic content (i.e. the vicinity constraints) is cached by the relaying nodes for future query processing purposes. When $\boldsymbol{Q}$ is resolved at the data source $\boldsymbol{n_s}$, node $\boldsymbol{n_r}$ will send an updating message along the same route as the query forwarding. The nodes on the route will choose the nearer one from $\boldsymbol{n_r}$ and $\boldsymbol{n_s}$ to cache along with query $\boldsymbol{Q}$. Thus the network is divided into two parts: the nodes within and outside the sphere $\sigma(\boldsymbol{n_r}, \boldsymbol{d})$. Later, suppose a node $\boldsymbol{n_r}^*$ issues a query $\boldsymbol{Q}^*$, semantically similar to $\boldsymbol{Q}$. The $\boldsymbol{Q}^*$ is forwarded among the nodes following the same spiral pattern. When $\boldsymbol{Q}^*$ meets with a node $\boldsymbol{n_i}$ within the sphere $\sigma(\boldsymbol{n_r}, \boldsymbol{d})$, the query is resolved at $\boldsymbol{n_i}$ and the result from source node (i.e. $\boldsymbol{n_s}$ or $\boldsymbol{n_r}$) will be sent to $\boldsymbol{n_r}^*$. Note that the forwarding of $\boldsymbol{Q}^*$ could be restricted within a section of the network. Therefore the data-distribution information is propagated in the process of query resolution, and flooding is avoided.

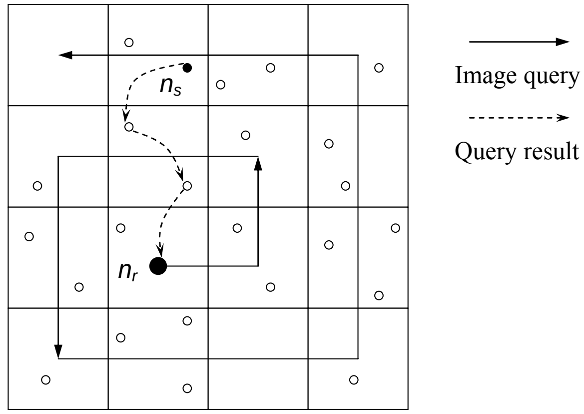

Figure 3: Query processing and caching rationale.

4.2. Cache Model

Definitions 6 and 7 allow one to describe a set of images based on their semantic categories and vicinity constraints. Similarly, the cache content of a mobile node can also be represented in the same way. The difference is that the mobile nodes only cache the semantic description of images, while the raw image data are kept in the source nodes.

In the SAIC scheme, the cache content is intended to characterize the data distribution of remote nodes through analysis of earlier queries. If an inner-category k-NN query k-NN$_c$($\boldsymbol{x_q}$, **€**$_j$) is resolved successfully at a remote node $\boldsymbol{n_i}$, the node id $\boldsymbol{n_i}$ along with the semantic description of query result will be cached for future use. If k-NN$_c$($\boldsymbol{x_q}$, **€**$_j$) is not resolved at $\boldsymbol{n_i}$, then node $\boldsymbol{n_i}$ contains no semantically similar images in category **€**$_j$ (i.e. **€**$_j$ is vacant for query $\boldsymbol{x_q}$ in $\boldsymbol{n_i}$), thus **€**$_j$ will be marked as a vacant region for $\boldsymbol{x_q}$ and its semantically similar queries. As more queries are submitted, the categories **€**$_1$, **€**$_2$, …, **€**$_t$ are potentially partitioned into two groups according to the data content of $\boldsymbol{n_i}$: the categories that contain similar image data and the ones that do not have similar data.

The local cache of a mobile node $\boldsymbol{n_i}$ is a set of cache entries — each entry indicates one or multiple remote nodes in the network. An entry is a triplet $\boldsymbol{T_i}$ = (***matching region***, ***vacant region***, ***node list***). The matching region is the description of resolved queries as defined in definition 7, which can be considered as subspaces covering the data points of earlier query results. The vacant region shows the unresolved queries, which can be represented as a collection of subspaces where no query results are found. The node list shows the mobile nodes whose data contents can be characterized by the matching region and the vacant region.

5. Performance Study

To evaluate the performance of the proposed caching scheme, we implemented a simulator in *ns-2* environment (version 2.26) [10]. To facilitate image retrieval, a semantic-representation module was also developed and added to the simulator.

5.1. Simulation Setup

The simulation was initialized by assuming a default number of pre-existing nodes in the network and randomly setting up the connections between the nodes. In addition, to mimic the dynamic structure of the ad hoc networks, during the course of the simulation, a mixture of operations, including querying, updating, node joining, and node leaving, are randomly submitted to the network. The simulator relies on a set of input parameters that are summarized in table 1.

Node mobility patterns: Two mobility models — *random way point* (RWP) [3] and *Manhattan* [10] — are used in the simulator. Each node randomly selects its movement (i.e. direction and velocity) within an 1500×320m^2 area. The node density can be adjusted by changing the number of nodes from 50 to 100.

Data and query distribution: The test bed comprises up to 3000 images (435 features) of 100 semantic categories from the Corel dataset, which is similar as the dataset used in [7]. 2000 images in the test bed are used to train a LPP subspace learning module that partitions the semantic space into 100 orthogonal regions, and the remaining 1000 images are used as test images for CBIR queries. The query generation pattern follows *Zipf-like* distribution, which is widely used to model non-uniformly distributed queries [11].

Table 1: The simulation parameters.

Parameter	*Default*	*Range*
Simulation time	5000 sec	100 – 20000
Environment size	1500*320m^2	10^4 m^2 to 10^8m^2
Transmitter range	100m	100m to 1,000m
Bandwidth	1M bps	0.1 – 10M bps
Number of nodes	100	50 to 100
Node mobility (v_{max})	2 m/s	1 to 20 m/s
Local cache size	800 KB	20 KB to 2 MB
Query rate (Q_{rate})	0.1 query/s	0.01 to 10 query/s
Control message	2 KB	
Data message size	20 KB	10 KB to 1 MB
Image dataset size	3000	
Semantic category	100	
Nearest neighbors	10	1 to 20

5.2. Simulation Result

The experiments were run using different workloads and system settings. The proposed model has been compared and contrasted against those presented in [2] based on performance metrics such as cache hit ratio and response time for different workloads, i.e., mean query rate and nearest-neighbor number, and different system parameters, i.e, cache size, network density, and node mobility.

- **Cache Hit Ratio**

Traditional performance metrics of a caching scheme, i.e., query response time, throughput, and search cost, are highly dependent on cache hit ratio.

Figure 4 shows the cache hit ratio as a function of the node density and cache size. As can be seen, the SAIC scheme regardless of the cache size, offers a higher hit ratio than the CachePath and CacheData models. This implies that relative to the CacheData and CachPath models, the SAIC offers a reasonable hit ratio with smaller cache size. Within the scope of mobile paradigm, this is an interesting observation since mobile nodes always strive for limited memory.

In addition, the cache hit ratio of SAIC increases as the number of nodes increases, while CacheData and CachePath have decreased hit ratio with increased node density ─ For a fixed data set and a fixed set of queries, the semantic locality of queries decreases and hence cache effectiveness drops. However, in SAIC, increase in the node density also implies an increase in semantic replica of the query results. This increases the probability of cache hit for future semantically similar queries.

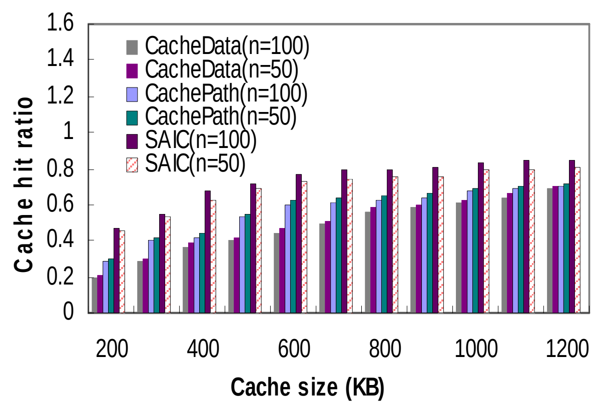

Figure 4: The effect of density on cache hit ratio.

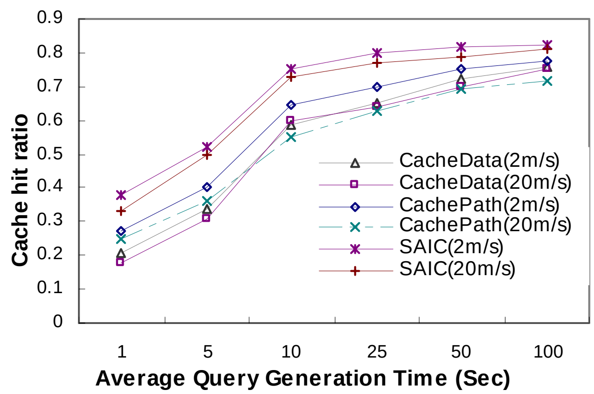

Figure 5: The effect of speed on cache hit ratio.

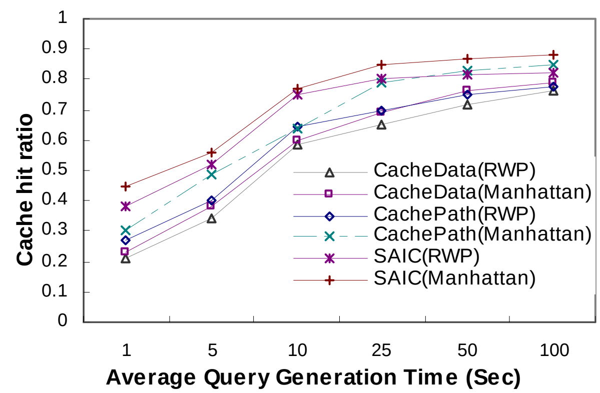

Figure 6: The effect of mobility pattern on hit ratio.

From figure 5, as one can expect, the cache hit ratio of SAIC and CachePath drops as the node mobility increases — increase in mobility incurs more changes in network topology, making it more difficult for CachePath and SAIC to locate remote data source nodes. The cache hit ratio of CacheData, in comparison, is not drastically affected by the mobility due to its independence from path information.

Figure 6 shows the cache hit ratios of the caching schemes using different mobility patterns. All schemes improve their performance in Manhattan pattern. This is due to the spatial and temporal dependence in Manhattan pattern that leads to reduced probability of route breaks and topology changes. In addition, the performance variation of the SAIC is less drastic than CachePath. This is due to the cache replacement policies adapted by these two caching schemes: CachePath simply removes the less frequently accessed data items to save cache space. However, the SAIC attempts to increase the semantic contents of the cache by using coarser semantic descriptions for the less frequently accessed cache entries. Increase in the semantic contents of the cache improves the cache hit ratio and hence, implies better cache utilization and more robust model adaptable to dynamic network topology.

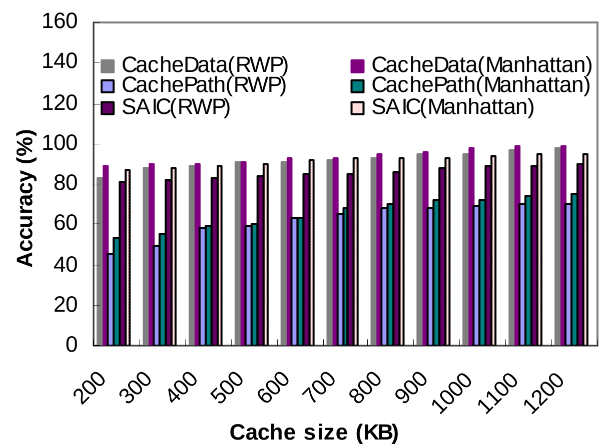

Figure 7: The accuracy of the caching schemes.

An important issue for caching schemes is the accuracy of query results when cache hits are obtained. The accuracy is evaluated through the comparisons of query results returned from caches against the ones from flooding-based retrieval. As can be seen from figure 7, the SAIC outperforms Cache-Path and achieves comparable accuracy as CacheData. The better performance of CacheData is due to its caching of raw images. The SAIC returns more accurate query results than CachePath due to its exploitation of content distribution, which gives the heuristic information for well-aimed search.

- **Query Response Time**

In another simulation run, we measured the query response time as a function of node density, cache size, and average query generation time. Figure 8 shows the combined effect of cache size and mobility on query response time. As can be seen from figure 8, Cache-Data is more sensitive to the cache size than CachePath and SAIC. For a fixed set of queries, CacheData requires much more cache space because it stores the raw images. CachePath and SAIC, in comparison, only cache the data content description and the ids of data source nodes, making better utilization of storage.

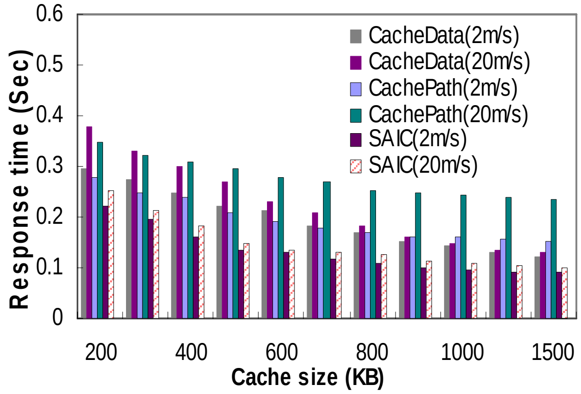

Figure 8: The effect of cache size on response time.

- **The Network Traffic**

In order to evaluate the impact of the caching strategies on the network traffic, we tuned the simulator to examine the message overhead on mobile nodes. Figure 9 shows that SAIC incurs much less message overhead than CachePath and CacheData. The reason is that SAIC resolves queries using semantic replicas in nearby nodes instead of faraway data source nodes if possible. Therefore, the data requests and replies need to travel less number of hops and mobile nodes need to process less number of messages.

In CacheData and CachePath the cache misses could incur flooding in the whole network, and each node may reply the query with the most similar images in its local database. The multiple replies to the query further increases the network traffic, and thus implies higher requirement for bandwidth. SAIC solves this issue by performing inner-category *k*-NN on a small portion of the network — the semantically most relevant nodes. Figure 10 shows the impact of bandwidth on response time. Note that CacheData achieves comparable perfor-mance to SAIC as bandwidth increases. The reason is that the large bandwidth remedies the difference of access latency between local cache and remote nodes, reducing the effect of flooding and duplicated query results.

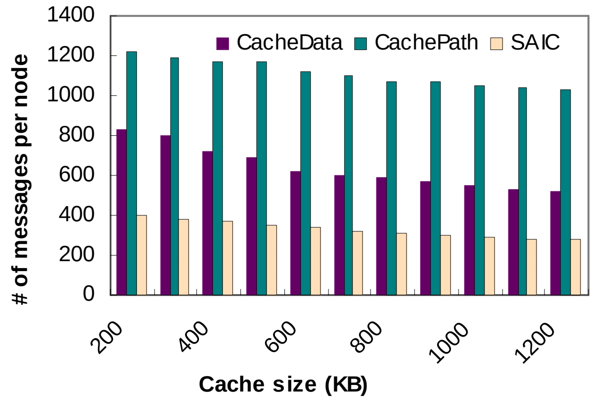

Figure 9: The average traffic on each node.

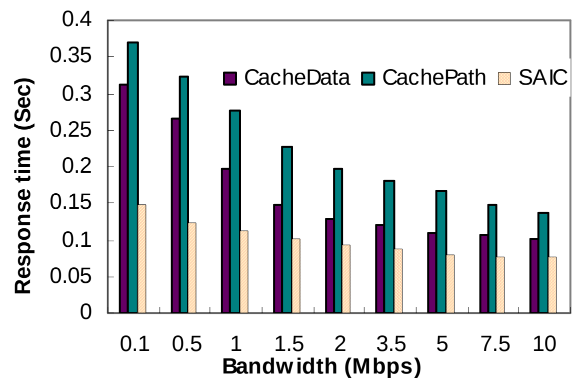

Figure 10: The effect of bandwidth.

6. Conclusions

Content-based image retrieval is a challenging task in ad hoc networks due to the node mobility, the network bandwidth, and the lack of infrastructure. In this paper we proposed a semantic-aware caching scheme to facilitate the efficient image retrieval in ad hoc networks. It employs vicinity constraints to represent image contents, offering higher cache hit ratio, higher space utilization, reduced network traffic, and lower average image retrieval time.

Extensive simulation results also showed that the performance of SAIC does not change drastically with various network settings (e.g. node density and mobility), which shows the robustness and scalability of SAIC.

We are tuning the performance of SAIC further and exploiting its application in other mobile environments such as sensor networks and WLANs. In addition, the scope of SAIC can be extended to accommodate other multimedia data (e.g. audio and video data).

7. References

[1] H. Luo, R. Ramjee, P.Sinha. UCAN: A unified cellular and ad hoc network architecture. ACM Mobicom, 2003: 353-367.

[2] L. Yin and G. Cao. Supporting cooperative caching in ad hoc networks. IEEE INFOCOM, 2004.

[3] V. Dheap, M. Munawar, and S. Ward, Parameterized neighborhood based flooding for ad hoc wireless networks. IEEE MILCOM, 2003: 1048-1053.

[4] J. He, M. Li, H. Zhang, H. Tong, C. Zhang. Manifold-ranking vased image retrieval, ACM Multimedia, 2004:9-16.

[5] G. Auffret. Multimedia access and retrieval: The state of the art and future directions, ACM Multimedia, 1999: 443-445.

[6] P. Hong, Q. Tian, and T. Huang. Incorporate support vector machine to content-based image retrieval with relevance feedback. IEEE ICIP, 2000:750-753.

[7] X. He, O. King, W.-Y. Ma, M.-J. Li, H.-J. Zhang. Learning a locality preserving subspace for visual recognition. Proc. IEEE Conf. on Computer Vision, 2003.

[8] Q. Ren and M.H. Dunham. Using semantic caching to manage location dependent data in mobile computing. ACM Mobicom, 2000: 211-221.

[9] T. Hara. Efficient replica allocation in ad hoc networks for improving data accessibility. IEEE INFOCOM, 2001.

[10] NS notes, http://www.isi.edu/nsnam/ns/. 2004.

[11] L. Breslau, P. Cao, L. fan, G. Phillips, and S. Shenker. Web caching and Zipf-like distribution: evidence and implications. IEEE INFOCOM, 1999.