# PROCEEDINGS

**The HKBU Second Computer Science Postgraduate Research Symposium**

**July 4, 2005**

# PG Day 2005

**Department of Computer Science**
**Hong Kong Baptist University**

# TABLE OF CONTENTS

## Session 1: Networking

Monitoring Top-k Query in Wireless Sensor Networks…………………………….…………………………….……….1
*Jianliang Xu, MinjiWu, Xueyan Tang, Wang-Chien Lee*

On-Demand Broadcast Algorithms with Caching on improving Response Time for Real-Time Information Dispatch Systems……………………………………………………………………………………………………..………..13
*Chui Ying Hui, Joseph Kee-Yin Ng*

An Improved Ellipse Propagation Model for Location Estimation in facilitating Ubiquitous Computing………………24
*Junyang Zhou*

## Session 2: Intelligent Informatics

A FragileWatermarking Scheme for 3D Mesh Verification…………………………………………………………...32
*Hao-tian Wu*

Learning Regularized Optimal Ensembles of Classifiers……………….……………………………………………..40
*Zhili Wu, Chunhung Li*

Solving Large-scale POMDP Problems Via Belief State Analysis……………………..………………………………..45
*Li Xin, William K. Cheung, Jiming Liu*

Visualizing Global Models from Distributed Sources……………………………………………………………………53
*Xiaofeng Zhang, William Kwok-Wai Cheung*

## Session 3: Pattern Recognition

A Contourlet-based Method for Chinese Handwriting-based Writer Identification …………..…………………………61
*Zhenyu He*

Protecting PCA Or LDA based Face Recognition System With Reed-Solomon Code…..………………………………67
*Feng Yicheng*

Enhancements of Generalized Hough Transform for Improving Object Recognition…………………………………...75
*Ho Fai Wong*

# Monitoring Top-$k$ Query in Wireless Sensor Networks

Jianliang Xu Minji Wu
Hong Kong Baptist Univ.
Kowloon Tong, Hong Kong
xujl@comp.hkbu.edu.hk

Xueyan Tang
Nanyang Technological Univ.
Singapore
asxytang@ntu.edu.sg

Wang-Chien Lee
Penn State Univ.
University Park, PA
wlee@cse.psu.edu

## Abstract

*Monitoring top-$k$ query is important to many wireless sensor applications. This paper exploits the semantics of top-$k$ query and proposes a novel energy-efficient monitoring approach, called* FILA. *The basic idea is to install a filter at each sensor node to suppress unnecessary sensor updates. The correctness of the top-$k$ result is ensured if all sensor nodes perform updates according to their filters. We propose detailed algorithms for filter setting and query reevaluation with the objective of reducing network traffic and prolonging network lifetime. We also extend the algorithms to two variants of top-$k$ query, i.e., order-insensitive and approximate top-$k$ monitoring. The performance of the proposed FILA approach is extensively evaluated using both synthetic and real traces. The results show that FILA outperforms the existing TAG-based approach by an order of magnitude under various network configurations.*

## 1 Introduction

Owing to the rapid advances in sensing and wireless communication technologies, wireless sensor networks have been available for use in a wide range of *in-situ* sensing applications, such as habitat monitoring, wild-fire prevention, and environmental monitoring [23]. A wireless sensor network typically consists of a base station and a group of sensor nodes (see Figure 1). The base station serves as a gateway for the sensor network to exchange data with external users. The sensor nodes, on the other hand, are responsible for sensing and collecting data from their local environments. They are also capable of processing sensed data and communicating with their neighbors and the base station.

Monitoring of aggregate functions is important to many sensor applications and has drawn a lot of research attention [6, 7, 16, 24, 25]. Among those aggregates, a top-$k$ query continuously retrieves the set of $k$ sensor nodes with the highest (or lowest) readings. For example:

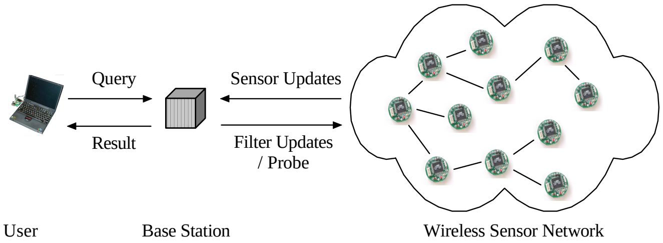


**Figure 1. The System Architecture**

**[Environmental Monitoring]** *Consider an environment-monitoring sensor network. A top-$k$ query is issued to find out the nodes and their corresponding areas with the highest pollution indexes for the purpose of pollution control or research study.*

**[Network Management]** *Power supply is critical for the operation of a wireless sensor network. Thus, a top-$k$ query may be issued to continuously monitor the sensor nodes with the lowest residual energy so that these sensor nodes can be instructed to adapt themselves (e.g., reducing sampling rates) to extend network lifetime.*

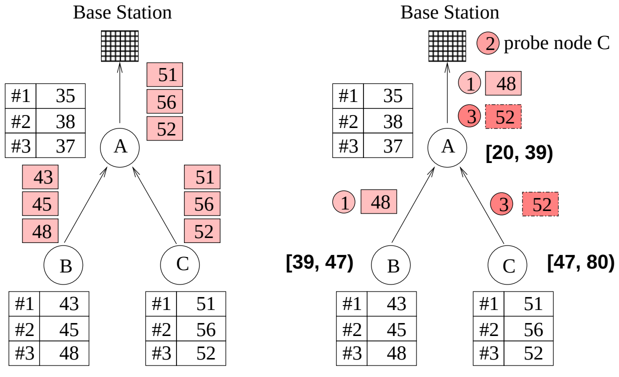
How to energy-efficiently answer top-$k$ queries is a great challenge to wireless sensor networks. The sensor nodes usually operate in an unattended manner and are battery powered; replacing the batteries is not only costly but also impossible in many situations (e.g., in a hard-to-reach area). If a certain portion of the nodes run out of their power and lose their coverage, the whole network will be down. Thus, in addition to reducing network traffic, a distinguished requirement for wireless sensor networks is to balance the energy consumption at the sensor nodes to prolong network lifetime [14, 30].

A basic implementation of monitoring top-$k$ query would be to use a centralized approach where all sensor readings are collected by the base station, which then computes the top-$k$ result set. In order to reduce network traffic for data collection, an *in-network data aggregation* technique, known as *TAG*, has been proposed [16]. Specifically, a routing tree rooted at the base station is first established and the data is then aggregated and collected along the way to the base station through the routing tree. Consider a simple example shown in Figure 2a, where sensor nodes $A$, $B$, and $C$ form a routing tree. The readings of these sensor nodes at three successive sampling instances are shown in the tables of Figure 2a. Suppose we are monitoring a top-1 query. Employing TAG, at each sampling instance, nodes $B$ and $C$ send their current readings to the parent (i.e., node $A$), which aggregates the data received with its own reading and sends the highest (i.e., the readings from node $C$ in this example) to the base station. The top-1 result is always node $C$, but nine update messages (three at each sampling instance) are used. As such, this approach incurs unnecessary updates in the network and, hence, is not energy efficient.

In this paper, we exploit the semantics of top-$k$ query and propose a novel filter based monitoring approach called *FILA*. The basic idea is to install a filter at each sensor node to suppress unnecessary sensor updates. The base station also keeps a copy of the filter setting to maintain a *view* of each node's reading. A sensor node reports the reading update to the base station only when it passes the filter. The correctness of the top-$k$ result is ensured if all sensor nodes perform updates according to their filters. Figure 2b shows an example, where the base station has collected the initial sensor readings and installed three filters [20, 39), [39, 47), and [47, 80) at sensor nodes $A$, $B$, and $C$, respectively. At sampling instances 1 and 2, no updates are reported since all updates are filtered out by the nodes' respective filters. At instance 3, the updated reading of node $B$ (i.e., 48) passes its filter [39, 47). Hence, node $B$ sends the reading 48 to the base station via node $A$ (step ①). Since 48 lies in the filtering window of node $C$ (i.e., [47, 80)), the top-1 result becomes undecided as either node $B$ or $C$ can have the highest reading. In this case, we probe node $C$ for its current reading to resolve the ambiguity (steps ② and ③). Thus, a total of four update messages and one probe message is incurred in this approach.[1] Compared with the aforementioned TAG-based aggregation approach, five update messages are saved at the cost of one probe message. Obviously, this approach achieves a better performance than the TAG approach.

Yet, in order to make FILA to work efficiently, two fundamental issues arising at the base station server have to be addressed:

- How to set the filter for each sensor node in a coordinated manner such that the top-$k$ result set is correctly returned if all nodes perform updates according to their filters? The filter setting is critical to the performance of FILA. In the above example, if nodes $B$ and $C$ have the filters set to [39, 50) and [50, 80), respectively, no updates need to be reported for all three samplings.



(a) TAG (b) FILA

**Figure 2. An Example of Top-$k$ Monitoring**

- Upon receiving an update from a sensor node, how to reevaluate the top-$k$ result and how to update the affected filters?

We answer in this paper the above two questions with the objective of reducing network traffic and prolonging network lifetime. In particular, this paper makes the following contributions:

- To the best of our knowledge, this is the first effort dedicated to investigating the problem of monitoring top-$k$ query in wireless sensor networks. Different from monitoring top-$k$ query in traditional distributed networks, our main objective is to extend network lifetime.

- We propose a novel approach called FILA for monitoring top-$k$ query (and its variants) in wireless sensor networks. We examine in detail the critical issues of filter setting and query reevaluation under this approach. Two filter setting schemes (i.e., *uniform* and *skewed*) and two filter updating strategies (i.e., *eager* and *lazy*) are proposed.

- Extensive experiments are conducted to evaluate the performance of the proposed FILA approach using both synthetic and real traces. The results provide a number of insightful observations and show that FILA outperforms TAG by an order of magnitude under various network configurations.

The remainder of this paper proceeds as follows. Section 2 reviews the related work on processing top-$k$ queries in distributed environments. Section 3 presents our proposed approach, FILA, and discusses how to set the filter for each sensor node and how to reevaluate the top-$k$ query result when updates occur. We extend FILA to handle approximate and order-insensitive top-$k$ queries in Section 4. The performance of the FILA approach is evaluated in Section 5. Finally, we conclude this paper and present some future research plans in Section 6.

[1] For simplicity, the overhead for initial data collection and filter setting is not shown here, but counted in our experiments.

## 2 Related Work

Evaluating top-$k$ queries in distributed networks has been extensively studied in the literature (e.g., [4, 5, 9, 11, 27, 31]). A typical assumption is that the ranking score of an object should be aggregated from a number of attribute values which are stored at distributed data sources (formally called *vertically partitioned datasets*). The best known algorithm is the threshold algorithm (TA) [9, 11, 19]. While the TA algorithm requires data sources to support sorted access, Bruno *et al.* [4] proposed the Upper algorithm for sources that support random access only. Cao and Wang [5] developed a three-phase uniform threshold (TPUT) algorithm that significantly reduces remote access in large networks. In [27], Theobald *et al.* further extended the TA algorithm by introducing a family of approximate variables based on probabilistic arguments to reduce runtime costs. Michel *et al.* [18] proposed a flexible framework for distributed top-$k$ algorithms that allows for trading-off efficiency versus result quality and bandwidth savings versus number of communication phases. More recently, top-$k$ processing algorithms have been developed for peer-to-peer networks [2] and private databases [28]. However, all these studies have focused on *one-shot* top-$k$ queries, whereas we are interested in monitoring continuous top-$k$ queries in this paper. As pointed out in [1], while continuous monitoring could be simulated by repeatedly executing a one-shot query, many queries would be executed in vain if the answer remains unchanged, hence being cost inefficient. Moreover, it is difficult to determine the optimal frequency of repeated query execution.

Babcock and Olston [1] performed a pioneering research on monitoring continuous top-$k$ queries over distributed data sources, which is the most similar work to this paper. Their idea is to add an adjustment factor to each source to ensure that the local top-$k$ list aligns to the global top-$k$ list maintained at the coordinator. However, as their work targets on vertically partitioned datasets, their proposed algorithm is not effective to our scenario where data objects are not partitioned. More specifically, their algorithm maintains an invariant that the adjustment factors allocated to different sources for each data object sum to zero. This means each object is allocated with an adjustment factor of zero when generalizing it to non-partitioned data, which is indeed similar to the basic approach discussed the Introduction. Furthermore, their work is limited to order-insensitive top-$k$ monitoring; the more challenging order-sensitive top-$k$ monitoring was not studied.

Monitoring of aggregation functions (such as average, sum, count, min, and max) in sensor networks has been investigated in the past few years. However, the main focus has been on how to establish the routing architecture for continuous data collection [7, 12, 13, 16, 24] such that in-network aggregation techniques [6, 25] can be applied to reduce network traffic. Taking a different angle, this paper exploits the semantics of top-$k$ query and proposes a new method to reduce network traffic and prolong network lifetime. Data storage and query processing for one-shot queries in sensor networks have also been studied in the literature (e.g., [3, 8, 10, 15, 17, 29]), which focused on applications different from this paper.

## 3 Top-$k$ Monitoring

We first describe the system model and give a formal problem definition in Section 3.1. Then, Section 3.2 provides an overview of the proposed FILA monitoring approach. Finally, the query reevaluation and filter setting issues are discussed in Sections 3.3 and 3.4, respectively.

### 3.1 System Model and Problem Definition

We consider a wireless sensor network as depicted in Figure 1. It is assumed that the base station has continuous power supply and its radio strength is strong enough to cover all sensor nodes. In other words, a probe message broadcast by the base station can reach all sensor nodes in a single hop. In contrast, the sensor nodes are powered by battery. Their radio coverage is constrained to a local area. When the base station is beyond a sensor node's radio coverage, an underlying routing infrastructure (e.g., a TAG tree [16]) is used to route data to the base station.

Each sensor node $i$ measures the local physical phenomenon $v_i$ (e.g., pollution index, temperature, or residual energy, etc.) at a fixed sampling rate. Without loss of generality, we consider top-$k$ monitoring query that continuously retrieves the set of sensor nodes $\mathcal{R}$ with the highest readings, i.e.,

$$\mathcal{R} = \{n_1, n_2, \cdots, n_k \mid \forall n_i < n_j, v_{n_i} \geq v_{n_j}; \forall l \notin \mathcal{R}, v_l \leq v_{n_k}\}$$

The monitoring result is maintained by the base station and disseminated to the user. To produce continuous query results, the proposed monitoring approach controls when and how to collect sensor reading updates to the base station.

For simplicity, we assume the sensor updates arrive at the base station sequentially. That is, no sensor updates take place during the processing of another sensor update. Although this is not a prerequisite for the proposed FILA approach, this assumption simplifies our discussion.

### 3.2 FILA Overview

Initially, the base station collects the readings from all sensors. It then sorts the sensor readings and obtains the initial top-$k$ result set. Next, the base station computes a filter (represented by a window of [$l_i$, $u_i$)) for each sensor node $i$ and sends it to the node for installation. At the next sensor sampling instance, if the new reading of sensor node $i$ is within [$l_i$, $u_i$), no update to the base station is needed. Otherwise, if the new reading goes beyond the filtering window and passes the filter, meaning the top-$k$ order might be violated, an update is sent to the base station. The base station will then reevaluate the top-$k$ result and adjust the filter setting(s) for some sensor node(s) if necessary. The query reevaluation algorithm will be discussed in detail in Section 3.3.

As can be seen, the purpose of using filters is to filter out some local sensor updates and hence suppressing the traffic in the network. The correctness of the top-$k$ result must be guaranteed provided that all sensor nodes perform updates according to their filters. Thus, the filter settings have to be carefully planned in a coordinated manner. Denote the current reading of node $i$ by $v_i$. Without loss of generality, we number the sensor nodes in decreasing order of their sensor readings, i.e., $v_1 > v_2 > \cdots > v_N$, where $N$ is the number of sensor nodes under monitoring. Intuitively, to maintain the monitoring correctness, the filters assigned to the nodes in the top-$k$ result set should cover their current readings but not overlap with each other. On the other hand, the nodes in the non-top-$k$ set could share the same filter setting. Thus, we consider the filter settings only for the top-$k$+1 nodes. A *feasible* filter setting scheme, represented as $\{[l_i, u_i) \mid i = 1, \cdots, k+1\}$, must satisfy the following conditions:

$$\begin{cases} u_1 > v_1; & \\ v_{i+1} < u_{i+1} \leq l_i \leq v_i, & (1 \leq i \leq k); \\ l_{k+1} \leq v_N. & \end{cases} \tag{1}$$

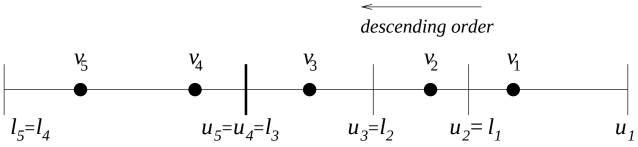


**Figure 3. Filter Settings for Top-3 Monitoring**

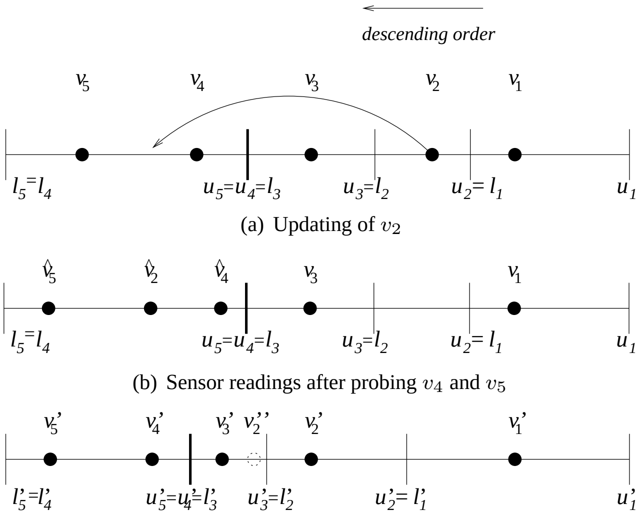
Figure 3 shows a feasible filter setting for top-3 monitoring, where nodes 4 and 5 share a filter setting and $u_{i+1}$ is set equal to $l_i$ for $1 \leq i \leq 3$ in order to maximize the filtering capability. Intuitively, *a filter setting is a (constrained) partitioning of the data space*. A formal discussion on filter setting is to be presented in Section 3.4.

We here make a remark before we go further into the details of the FILA approach. The FILA approach makes use of filters to keep track of the ordered list of top-$k$ sensor nodes. A side product is that FILA also returns for each node an approximate reading bounded by a filtering window. While dynamically maintaining the exact sensor readings is too costly to be desirable, such approximate information is useful yet sufficient to many applications. We will measure the level of approximation using trace-driven simulation in Section 5.

## 3.3 Query Reevaluation

We now discuss the query reevaluation algorithm. Under the proposed FILA monitoring approach, a sensor node sends an update to the base station only when its reading passes the filter. In this case, if the new reading overlaps with the filtering window of any other sensor node, the top-$k$ result becomes undecided. Hence, the base station will have to probe the corresponding sensor(s) to reevaluate the top-$k$ result. Let's call the lower bound of the top-$k$th node's filter the *critical bound*, e.g., $l_3$ in Figure 3. We discuss the query reevaluation algorithm for three scenarios: 1) the update is originated from a top-$k$ node and jumps over the critical bound (see Figure 4a); 2) the update is originated from a non-top-$k$ node and jumps over the critical bound (see Figure 5a); 3) the update is from a top-$k$ node but does not jump over the critical bound (see Figure 6a). For the first two scenarios, the node may leave or join the top-$k$ set respectively; for the third scenario, two top-$k$ nodes may need to swap their positions in the top-$k$ set.

### 3.3.1 Scenario 1



(a) Updating of $v_2$

(b) Sensor readings after probing $v_4$ and $v_5$

(c) Updated order and new filter settings

**Figure 4. A Top-$k$ Node Jumps over Critical Bound**

As shown in Figure 4a, the new reading of the original top-2nd node falls in the filtering window of non-top-3 nodes (i.e., $[l_4, u_4)$). To determine which among nodes 2, 4, and 5 is the new top-3rd node, we have to probe the non-top-3 nodes 4 and 5 to update their current readings. Assuming their new readings, $\hat{v}_4$ and $\hat{v}_5$, are as illustrated in Figure 4b; the node 4 is the new top-3rd node. For clarity of presentation, we re-number them as $v'_1$ through $v'_5$ as shown in Figure 4c.

Note that the probing cost would be high for a small value of $k$ given a large number of sensor nodes. By taking advantage of the architecture of wireless sensor network (i.e., hierarchical routing and message broadcast), two enhancements are proposed to reduce the probing cost. First, when a sensor's new reading (denoted by $u'$) passes the filter, it propagates up the routing tree towards the base station. If an intermediate node is a non-top-$k$ node and its current reading is higher than $u'$, the reading of this intermediate node will be propagated to the base station instead since only this node can possibly be the top-$k$th node. Second, instead of probing all non-top-$k$ nodes, we include the newly updated sensor reading (denoted by $u''$) in the probe message and, thus, only the sensor nodes with current readings higher than $u''$ have a chance to be the top-$k$th node and will respond the probe. In the above example, only node 4 will report its current reading to the base station in response to the probe.

In order to adjust the filter settings, we propose to recompute the settings for all nodes based on their readings stored at the base station (see Section 3.4 for how to compute filter settings). Consider a node $i$. We discuss two approaches for filter updating:

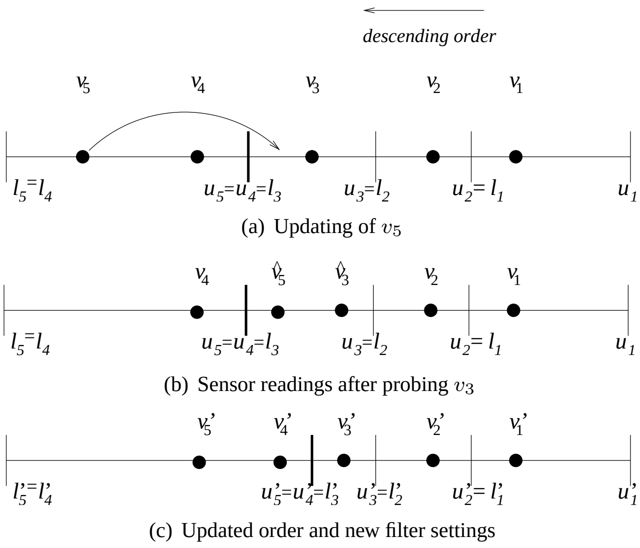
- **Eager Filter Updating:** With this eager approach, if a new filtering window $[l'_i, u'_i)$ is different from the current one $[l_i, u_i)$, the new filter $[l'_i, u'_i)$ is immediately sent to the node to replace $[l_i, u_i)$. In the example shown in Figure 4, all nodes will be updated with their new filters.

- **Lazy Filter Updating:** With the lazy approach, if a new filtering window $[l'_i, u'_i)$ is wider than the current one $[l_i, u_i)$, i.e., $[l_i, u_i) \subset [l'_i, u'_i)$ (e.g., the filters for $v_1$ and $v'_1$ in Figure 4), we delay the filter updating until when the filter is violated. During this period, the sensor node will continue to use the current filter $[l_i, u_i)$ to filter out sensor updates. It is easy to verify that the validity of the top-$k$ order is still guaranteed with such a *conservative* filter setting. The advantage of this approach is that we can avoid unnecessary filter updates. On the other hand, the narrower filtering window may make the filter violation (i.e., updating with the base station) to occur earlier. At the next update, the new reading $\hat{v}_i$ must be out of $[l_i, u_i)$ since it passes the filter. If $\hat{v}_i$ is within $[l'_i, u'_i)$, this filtering window will then be passed to the sensor node, which is all the base station needs to do. Otherwise, $\hat{v}_i$ is out of $[l'_i, u'_i)$, it will be treated as a normal update, which follows the query reevaluation algorithm discussed in this section to recompute the top-$k$ result and update the filter settings.

Whether the eager or lazy approach would perform better depends on the reading changing pattern. Consider two extreme cases. If the next update jumps out of the new filter $[l'_i, u'_i)$, the lazy approach can help save a filter updating message. Otherwise if the next update is within $[l'_i, u'_i)$, it does not help save any filter updating message but incurs an additional data update message. We will investigate their performance using trace-driven simulation (see Section 5.2).

There is a subtle point to note here. For a sensor node which has not updated with the base station, its new filter overlaps with the current one (e.g., the filters for $v'_2$ and $v_3$ in Figure 4). It is possible that the actual sensor reading is beyond the new filter (e.g., at point $v''_2$). Nevertheless, in this case, the sensor node does not need to immediately report an update after the new filter is received. This is because the order of $v'_2$ and $v'_3$ is still preserved as the current update is originated from $v_2$ and hence $v'_2$ (originally $v_3$) must lie in $[l_3, u_3)$ and $v'_3$ (originally $v_4$) lie in $[l_4, u_4)$. However, if the next sampled reading $\hat{v}'_2$ is beyond the new filter, an update will be reported. This point is valid to all three scenarios.

### 3.3.2 Scenario 2



(a) Updating of $v_5$

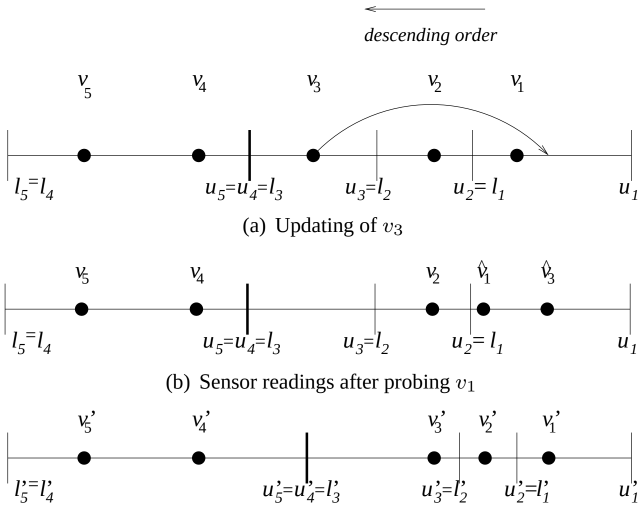
(b) Sensor readings after probing $v_3$

(c) Updated order and new filter settings

**Figure 5. A Non-top-$k$ Node Jumps over Critical Bound**

Now suppose that the update is originated from a non-top-$k$ node and jumps over the critical bound (see Figure 5a). In this case, the updating node will have a chance to join the top-$k$ set. Similar to Scenario 1, probe is needed to resolve the ambiguity. However, we only need to probe the node whose filtering window covers the updated reading (i.e., node 3 in Figure 5a). In addition, if the node to be probed happens to be on the routing path from the updating sensor node to the base station, its reading will be piggybacked during the update propagation. Thus, the probe is not needed. In any case, as a next step, the base station determines the new top-$k$ order and re-computes the filtering window for each affected node (i.e., nodes 3, 4, and 5), as illustrated in Figure 5c. The filter updates will be performed according to the approach employed. If the eager approach is used, the new filters for nodes 3 through 5 will be propagated right away; if the lazy approach is employed, the propagation of filters for nodes 4 and 5 will be delayed as discussed in the last subsection.

### 3.3.3 Scenario 3



(a) Updating of $v_3$

(b) Sensor readings after probing $v_1$

(c) Updated order and new filter settings

**Figure 6. Update within Top-$k$ Set**

Finally, we consider the case in which a top-$k$ node updates its reading but the new reading does not go beyond the critical bound (see Figure 6a). The handling of this case is similar to that for Scenario 2 except that here only the top-$k$ list might be re-ordered while in Scenario 2 the membership of top-$k$ set might be changed as well. Figure 6b shows an example where $\hat{v}_3$ takes the top-1st position after updating. Algorithm 1 outlines the algorithm that the base station uses to handle sensor-initiated updates and reevaluate top-$k$ query.

**Algorithm 1** Query Reevaluation Algorithm (Performed at Base Station)

```
while receiving a sensor-initiated update from a sensor node do
    if the update is originated from a top-$k$ node and jumps over the critical bound then
        probe all non-top-$k$ nodes that have a reading higher than the updated value
    else
        probe the node whose filtering window covers the updated value
    end if
    re-compute the top-$k$ set
    adjust the filter settings if necessary
end while
```

## 3.4 Filter Setting

As mentioned, another crucial issue in the FILA approach is the filter setting for each sensor node. In this section, we first discuss the settings for all nodes except the upper bound of the top-1st's filter (i.e., $u_1$) and the lower bound of non-top-$k$ node's filter (i.e., $l_{k+1}$). The intuitive way is to set the filter bound at the midpoint of two sensor readings, i.e.:

$$u_{i+1} = l_i = \frac{v_i + v_{i+1}}{2}, \quad (1 \leq i \leq k). \tag{2}$$

Obviously, this is a feasible filter setting satisfying (1), and we call it *uniform* filter setting. It is simple and favorable in the case where the sensor readings from all sensor nodes follow a similar changing pattern. On the other hand, the uniform setting fails to consider the changing patterns of sensor readings. If the reading changing patterns differ dramatically among the sensor nodes, the uniform setting might result in unbalanced energy consumption. Next, we develop a *skewed* filter setting algorithm by taking into account the reading changing pattern. Our objective is to balance the energy consumption between the "neighboring" nodes (with close sensor readings).

Suppose the average time for the reading of node $i$ to go beyond $\delta$ is known as $f_i(\delta)$. Thus, the node update rate (with the base station) is given by $\frac{1}{f_i(\delta)}$. In order to balance the energy consumption of nodes $i + 1$ and $i$, we should choose proper $u_{i+1}$ and $l_i$ such that their update rates are equal:

$$\frac{1}{f_{i+1}(u_{i+1} - v_{i+1})} = \frac{1}{f_i(v_i - l_i)}. \tag{3}$$

In practice, it is usually difficult to know in advance how the sensor readings evolve dynamically and to estimate $f_i(\delta)$. One approach is to use the historical sensor readings to predict $f_i(\delta)$. However, this approach is costly as the base station has to collect all changes of sensor readings (with base station performing prediction) or periodically refresh the detailed functions from the sensor nodes (with sensor node performing prediction). We propose a practical low-cost approach by assuming the readings change follow a well-known random walk model [12, 22, 29]. Under the random walk model, the value changes in steps. At each step, the value increases or decreases by an amount of $d$. Denote the inter-step interval by $l$. The average time for the value to go beyond $\delta$ can be expressed as follows [29]:

$$f(\delta) = (\frac{\delta}{d})^2 \cdot l. \tag{4}$$

We let every node measure the average delta change, $d_i$, of their sensor readings at a fixed rate. When the sensor node reports an update to the base station, it will piggyback the measured value of $d_i$. Let $L$ be the time interval chosen to measure the average delta change. Then, the $f_i(\delta)$ function can be approximated by:

$$f_i(\delta) = (\frac{\delta}{d_i})^2 \cdot L. \tag{5}$$

Substituting (5) into (3), we obtain

$$(\frac{d_{i+1}}{u_{i+1} - v_{i+1}})^2 / L = (\frac{d_i}{v_i - l_i})^2 / L.$$

Solving this equation, we get:

$$\frac{u_{i+1} - v_{i+1}}{v_i - l_i} = \frac{d_{i+1}}{d_i}.$$

Letting $u_{i+1} = l_i$, we have

$$u_{i+1} = l_i = v_{i+1} + \frac{d_{i+1}}{d_i + d_{i+1}} \cdot (v_i - v_{i+1}), \qquad (1 \leq i \leq k). \tag{6}$$

We now discuss the settings for the upper bound of the top-1st's filter (i.e., $u_1$) and the lower bound of non-top-$k$ node's filter (i.e., $l_{k+1}$). Theoretically, $u_1$ can be set at $+\infty$. However, in this case, the node will not trigger an update even if its reading has gone up remarkably. To adjust the filter setting for such cases (hence giving other nodes a chance to increase their filters' upper bounds and reducing update rates), we set $u_1$ to $2 \cdot v_1$, i.e., twice of its current reading $v_1$. Similarly, $l_{k+1}$ is set to $\frac{v_N}{2}$, i.e., half of the lowest sensor reading.

# 4 Extensions

So far we have focused on order-sensitive exact top-$k$ monitoring. However, the ordering information may not be needed in the top-$k$ set for all applications. Moreover, a certain degree of data approximation may be tolerable by some applications to trade for energy efficiency. Motivated by these observations, in this section, we extend FILA to handle order-insensitive top-$k$ monitoring and approximate top-$k$ monitoring.

## 4.1 Order-Insensitive Top-$k$ Monitoring

The order-sensitive algorithm discussed in Section 3 is also applicable to order-insensitive top-$k$ monitoring. Nevertheless, since we now do not care the exact order of sensor readings in the top-$k$ set, the updates within the top-$k$ readings do not have to be reported. Therefore, we only need to set a critical bound between the top-$k$ nodes and the non-top-$k$ nodes, as shown in Figure 7.

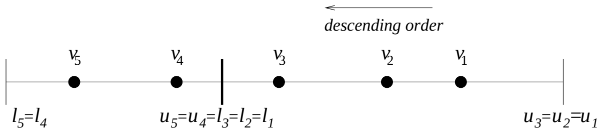


**Figure 7. Filter Settings for Order-Insensitive Top-$k$ Monitoring**

Only the updates jumping over the critical bound need to be reported. When this happens, we will have to probe all the nodes in the top-$k$ list or those in non-top-$k$ list, depending on where the update is originated. The query reevaluation algorithm is similar to what was discussed in Section 3.3, and the two enhancements (Section 3.3.1) can be used to reduce the probing cost.

## 4.2 Approximate Top-$k$ Monitoring

We now consider approximate top-$k$ monitoring assuming a certain *degree of approximation* is acceptable. An approximate top-1 query retrieves the sensor node with the highest reading $v_i$ such that $\forall v_j (j \neq i), v_j < v_i + \epsilon$, where $\epsilon$ is the approximation degree. Intuitively, if two sensor readings are within a difference of $\epsilon$, either one can be taken as the top-1 result. It is straightforward to extend this definition to a top-$k$ query.

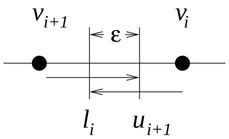


**Figure 8. Filter Settings for Approximate Top-$k$ Monitoring**

A *feasible* filter setting scheme for approximate top-$k$ monitoring, represented as $\{[l_i, u_i) \mid i = 1, \cdots, k+1\}$, should satisfy the following conditions:

$$\begin{cases} u_1 > v_1; \\ v_{i+1} < u_{i+1},\ u_{i+1} \leq l_i + \epsilon,\ l_i \leq v_i; \quad (1 \leq i \leq k); \\ l_{k+1} \leq v_N. \end{cases}$$

This means that it allows an overlap of $\epsilon$ between two neighboring filters (see Figure 8 for an illustration). Setting $\epsilon = 0$ degenerates approximate top-k monitoring to exact top-k monitoring.

The filter settings should be revised accordingly. Under the *uniform* filter setting, for each $1 \leq i \leq k$,

$$\begin{cases} u_{i+1} = \frac{v_i+v_{i+1}+\epsilon}{2}; \\ l_i = \frac{v_i+v_{i+1}-\epsilon}{2}. \end{cases}$$

Under the *skewed* filter setting, for each $1 \leq i \leq k$,

$$\begin{cases} u_{i+1} = v_{i+1} + \frac{d_{i+1}}{d_i+d_{i+1}} \cdot (v_i - v_{i+1} + \epsilon); \\ l_i = v_i - \frac{d_i}{d_i+d_{i+1}} \cdot (v_i - v_{i+1} + \epsilon). \end{cases}$$

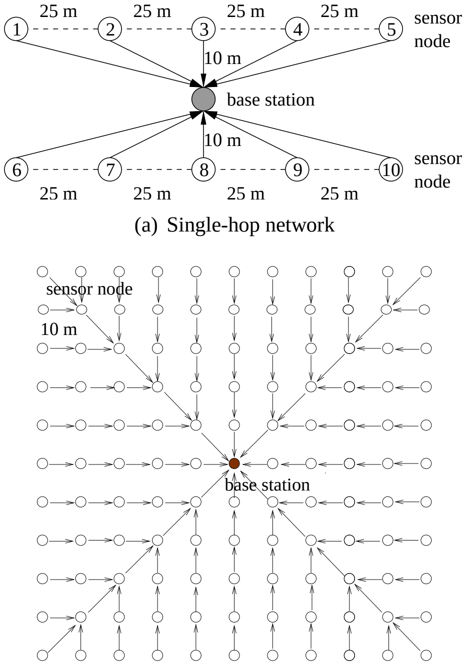
## 5 Performance Evaluation

### 5.1 Simulation Setup

We have developed a simulator based on ns-2 (version 2.26) [20] and NRL's sensor network extension [21] to evaluate the proposed FILA approach. The simulator includes the detailed models of the MAC and physical layers for wireless sensor networks. The sensor nodes can operate in one of three modes: sending message, receiving message, and sleeping. These modes differ in energy consumption. The energy consumption for sending a message is determined by a cost function: $s \cdot (\alpha + \beta \cdot d^q)$, where $s$ is the message size, $\alpha$ is a distance-independent term, $\beta$ is the coefficient for a distance-dependent term, $q$ is the component for the distance-dependent term, and $d$ is the distance of message transmission. We set $\alpha$=50 nJ/b, $\beta$=100 pJ/b/m$^2$, and $q$=2 in the simulation. The energy consumption for receiving a message is given by $s \cdot \gamma$, where $\gamma$ is set at 50 nJ/b. The power consumption in sleeping mode is set at 0.016 mW. For simplicity, the energy overhead of mode switching is ignored. We set the size of a data update message and the size of a filter update message both at 8 bytes, and the size of a probe message at 4 bytes. The initial energy budget at each sensor node was set at 0.01 Joule.

We simulated a single-hop network of 10 sensor nodes and a multi-hop network of 120 sensor nodes. Their layouts are shown in Figures 9a and 9b. The sensor readings are simulated using both synthetic and real traces.

- **Synthetic traces (RAN)**: The readings of each sensor node change following a one-dimensional random walk model [12, 22, 29]. Specifically, the reading updates in regular steps with an inter-step duration $t$. At each step, the reading changes by an amount (called *step size*) which is randomly assigned from a uniform distribution over $[-\delta, \delta]$, where $\delta$ is the maximum step size. We set $t$ at 10 time units and $\delta$ at 0.5-1.0 in the simulation.



(a) Single-hop network

(b) Multi-hop network

**Figure 9. Network Layouts**

- **Real traces (SEA / SUB)**: The real traces are provided by the Tropical Atmosphere Ocean (TAO) project [26], in which real-time oceanographic and meteorological data are collected from a wide range of monitoring sites for improved detection, understanding, and prediction of El Nino and La Nina. We selected the traces during 1 Jan. 1999−31 Dec. 2000 from a subset of the sites and mapped them to the sensor nodes in our networks at random. We used the sea surface temperature (SEA) and sea subsurface temperature (SUB) data in our experiments. The data of different sites are similar in magnitude. Figure 10 shows some representative segments of the SEA and SUB data traces. In general, the SEA data fluctuate more widely than the SUB data. We modified the sensor sampling interval to simulate two different workloads. In the homogeneous (HM) setting, the sampling interval for all sensors is set at 1 time unit; in the heterogeneous (HT) setting, the sampling interval for half sensors is set at 1 time unit and that for the other half is set at 5 time units.

We used the real traces, SEA and SUB, for the single-hop network configuration (Figure 9a) and the synthetic trace, RAN, for the multi-hop network configuration (Figure 9b). The default values of $k$ are set at 3 and 10 for these two configurations, respectively. In the following, we first compare the two filter updating strategies (i.e., eager and lazy) with the proposed FILA approach. We then evaluate FILA (with two different filter setting schemes, i.e.,

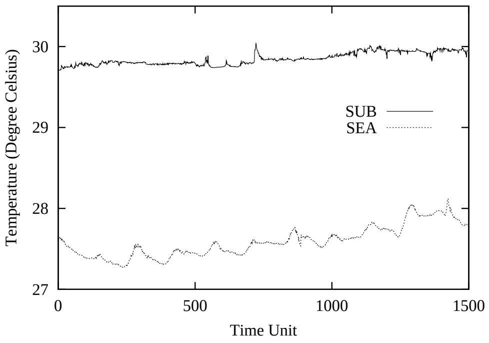
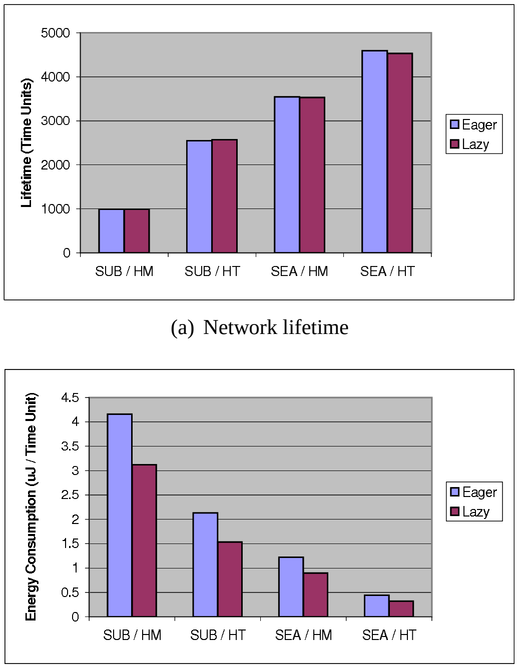


**Figure 10. Sample Real Data Traces**

uniform and skewed) against the TAG-based periodic aggregation approach (or TAG for short, which was illustrated in the Introduction). The following metrics are used in the comparison:

- **Network Lifetime**: As in the previous work [14, 30], the network lifetime is defined as the time duration before the first sensor node runs out of power. It serves as the primary metric in the performance evaluation.
- **Average Energy Consumption**: It is defined as the total amount of energy consumed in the network averaged for all sensor nodes over time.

### 5.2 Eager vs. Lazy Filter Updating



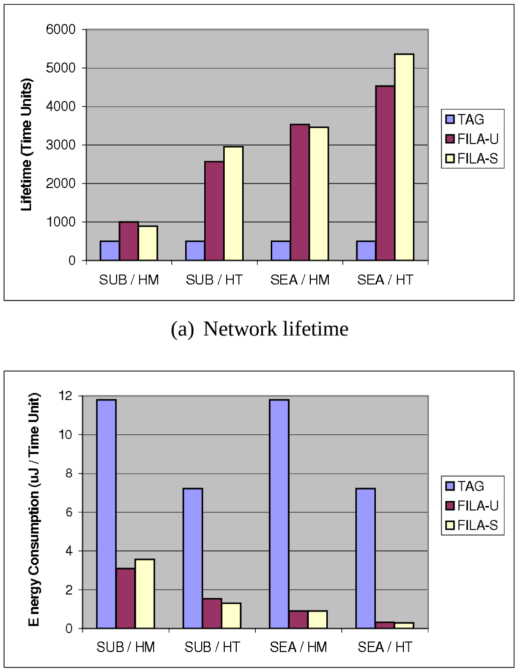
(a) Network lifetime

(b) Average energy consumption

**Figure 11. Eager vs. Lazy Updating ($k$=3)**

This set of experiments compares the eager and lazy filter updating strategies (discussed in Section 3.3). The uniform filter setting is employed. As shown in Figure 11a, these two approaches achieve a very similar network lifetime for the SEA and SUB traces. Yet the lazy approach performs much better in terms of average energy consumption, as plotted in Figure 11b. The energy saving is about 25%-28%. Based on the discussions in Section 3.3.1, this implies that most sensor reading changes have a magnitude wider than the new filtering windows and the lazy approach helps save the filter update messages, which contributes to a significant portion of the overall traffic (more than 40% for the eager approach as observed in the experiments). Therefore, the lazy approach is considered having a better overall performance than the eager approach. Similar performance trends are obtained for the RAN data trace; the result is not shown here due to space limitations. In the following experiments, we employ the lazy as the default filter updating strategy working with FILA.

### 5.3 Performance Comparison against TAG



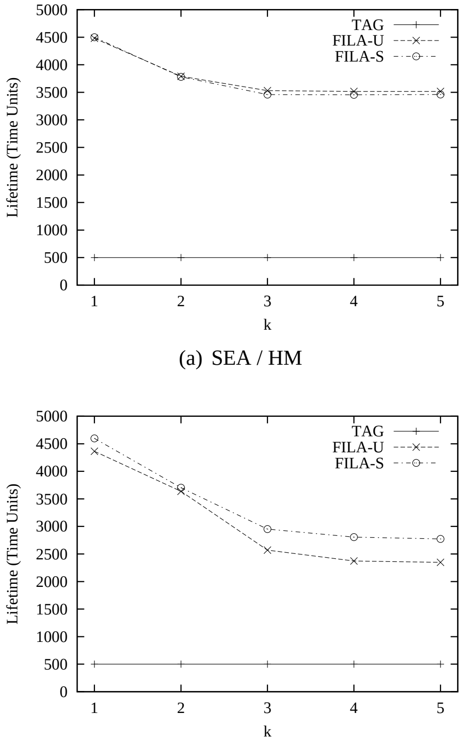
(a) Network lifetime

(b) Average energy consumption

**Figure 12. Performance Comparison with TAG (Single-Hop, $k$=3)**

In this section, we evaluate the performance of FILA against TAG. We denote the FILA approach with uniform filter setting as *FILA-U* and the FILA approach with skewed filter setting as *FILA-S*.

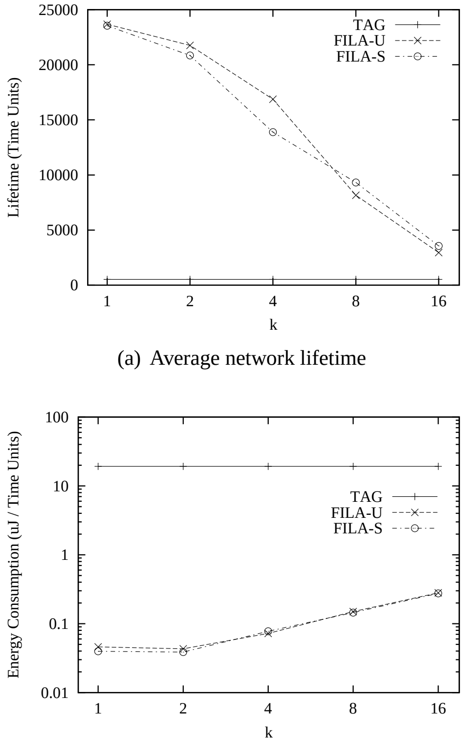
Figure 12 shows the results for the SUB and SEA traces under the single-hop network configuration, where $k$ is set



(a) SEA / HM

(b) SUB / HT

**Figure 13. Lifetime as a Function of $k$ (Single-Hop)**



(a) Average network lifetime

(b) Average energy consumption

**Figure 14. Performance Comparison with TAG (RAN, Multi-Hop)**

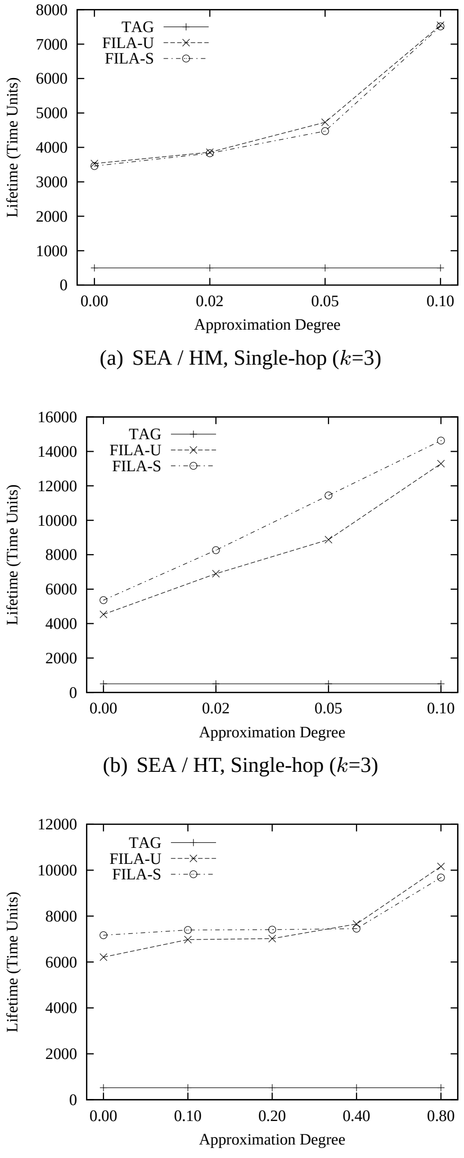
at 3. Several observations are obtained. First, both FILA-U and FILA-S improve the network lifetime over TAG by an order of magnitude while achieving a much lower average energy consumption. This result is consistent across all data traces examined, indicating the great performance advantage of our proposed approach. Second, when comparing FILA-U and FILA-S, FILA-U slightly outperforms FILA-S for the homogeneous (HM) sampling scenario but FILA-S gets a longer lifetime for the heterogeneous (HT) sampling scenario. This can be explained as follows. In the HM scenario, all sensors have a similar reading changing pattern such that the uniform filter setting performs good enough. However, as FILA-S incurs overhead in collecting changing patterns, its overall performance is slightly worse than that of FILA-U. On the other hand, in the HT scenario, FILA-S takes into consideration the changing patterns and thus obtains a better performance. Third, we observed in the experiments that the average size of top-$k$ nodes' filtering windows (except those for the top-1st node) is 0.26 degree Celsius. This confirms our argument in Section 3.2 that FILA not only returns the top-$k$ result set, but also provides a tightly bounded approximation for each of the top-$k$ sensor readings.

Figure 13 shows the network lifetime as a function of $k$ for SEA / HM and SUB / HT; similar performance trends are obtained for SUB / HM and SEA / HT. Again, FILA-U and FILA-S significantly outperform TAG for all cases examined. For SEA / HM (Figure 13a), for a similar reason explained in the last paragraph, FILA-U achieves a similar or slightly longer lifetime than FILA-S. For SUB / HT (Figure 13b), it is observed that FILA-S improves the network lifetime over FILA-U by 5%-18%, and in general, the improvement increases with increasing $k$.

We next examine the performance under the multi-hop network configuration. To do so, ten sets of random traces following the random walk model were generated for ten simulation runs. Figure 14 plots the average results over the ten runs. Similar to the single-hop configuration, FILA-U and FILA-S gain a much better performance than TAG in terms of both network lifetime and average energy consumption. It was observed that neither of them dominates the other for all traces tested. On average, FILA-U performs slightly better than FILA-S for a small value of $k$, whereas FILA-S is better for a large value of $k$.

## 5.4 Approximate Top-$k$ Monitoring

This section investigates the performance of approximate top-$k$ monitoring. We vary the approximation degree $\epsilon$ from 0.0 to 0.1 for SEA / HM and SEA / HT under the single network configuration and from 0.0 to 0.8 for RAN



(a) SEA / HM, Single-hop ($k$=3)

(b) SEA / HT, Single-hop ($k$=3)

(c) RAN, Multi-hop ($k$=10)

**Figure 15. Approximate Top-$k$ Monitoring**

under the multi-hop network configuration. The values of $k$ are set at 3 and 10 for these two configurations, respectively. As can be seen in Figure 15, FILA-U and FILA-S consistently outperforms TAG by an order of magnitude. Similar to the observations made for exact top-$k$ monitoring, FILA-S gets a similar or slightly worse performance than FILA-U for the homogeneous (HM) sampling scenario but performs better (with 10%-29% of improvement) for the heterogeneous (HT) scenario under the single-hop configuration.

It is interesting to observe that for all traces, the network lifetime can be noticeably extended with a small degree of approximation allowed. For example, with an approximation degree of 0.1 in Figures 15a and 15b, using FILA-U or FILA-S, the network lifetime is prolonged by more than 2 times. Even under the multi-hop network configuration (Figure 15c), the network lifetime is improved by at least 65% with the FILA approaches as the approximation degree is increased from 0.0 to 0.8.

## 5.5 Order-Insensitive Top-$k$ Monitoring

Finally, in this section, we evaluate the performance of order-insensitive top-$k$ monitoring. Figure 16 shows the results under the default system settings (i.e., $k$ set at 3 and 10 for SEA / SUB and RAN, respectively). For order-insensitive top-$k$ monitoring, only the critical bound is maintained as the filter by all sensors; hence, the filter setting does not have a very high impact on the performance. As a result, FILA-U and FILA-S perform similarly in most cases. The network lifetime for SEA / HT is dramatically improved from order-sensitive monitoring. As observed in the experiments, this is because the order-sensitive monitoring extends the network lifetime and the trace after an extended time point becomes more stable than before. Hence, FILA makes use of the filters and runs a longer lifetime.

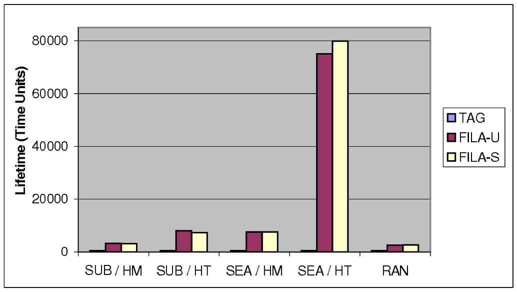


**Figure 16. Order-Insensitive Monitoring**

# 6 Conclusions

This paper has performed a comprehensive study on monitoring top-$k$ query in wireless sensor networks. Different from existing work focusing on in-network data aggregation techniques, we exploited the semantics of top-$k$ query and proposed a novel energy-efficient monitoring approach, called FILA. We presented detailed algorithms to address two critical issues under the FILA approach, i.e., filter setting and query reevaluation. Two filter setting algorithms (i.e., uniform and skewed) and two filter updating strategies (i.e., eager and lazy) were proposed. We have also extended the algorithms to two variants of top-$k$ query, i.e., order-insensitive and approximate top-$k$ monitoring.

A series of simulation experiments has been conducted to evaluate the performance of the proposed FILA approach based on both synthetic and real traces. The following results were obtained: 1) FILA consistently outperforms the existing TAG-based approach by an order of magnitude under various network configurations; 2) FILA can also provide a tightly bounded approximation for each of the top-$k$ sensor readings, in addition to returning the top-$k$ result set; 3) the lazy filter updating approach obtains a better overall performance than the eager approach for the traces examined; 4) the uniform filter setting performs slightly better than the skewed filter setting for the homogeneous sampling scenario, whereas the skewed filter setting is better for the heterogeneous sampling scenarios; 5) their relative performance under the multi-hop network configuration varies with the application scenarios, depending on the factors such as value of $k$, approximation degree, and order insensitiveness; 6) using FILA, a small degree of approximation in the top-$k$ order improves the network lifetime substantially.

As for future work, we plan to extend the proposed monitoring approach to other aggregate functions such as kNN, average, and sum. We are going to build a prototype based on Motes and measure the performance in real environments. We are also interested in monitoring spatial queries in object-tracking sensor networks.

## References

[1] B. Babcock and C. Olston. Distributed top-k monitoring. In *Proc. ACM SIGMOD*, pages 28–39, June 2003.

[2] W.-T. Balke, W. Nejdl, W. Siberski, and U. Thaden. Progressive distributed top-k retrieval in peer-to-peer networks. In *Proc. IEEE ICDE*, April 2005.

[3] P. Bonnet, J. E. Gerhke, and P. Seshadri. Towards sensor database systems. In *Proc. Int. Conf. on Mobile Data Management (MDM)*, January 2001.

[4] N. Bruno, L. Gravano, and A. Marian. Evaluating top-k queries over web-accessible databases. In *Proc. IEEE ICDE*, Feburary 2002.

[5] P. Cao and Z. Wang. Efficient top-k query calculation in distributed networks. In *Proc. PODC*, July 2004.

[6] J. Considine, F. Li, G. Kollios, and J. Byers. Approximate aggregation techniques for sensor databases. In *Proc. IEEE ICDE*, March 2004.

[7] A. Deligiannakis, Y. Kotidis, and N. Roussopoulos. Hierarchical in-network data aggregation with quality guarantees. In *Proc. EDBT*, March 2004.

[8] A. Deshpande, C. Guestrin, W. Hong, and S. Madden. Exploiting correlated attributes in acquisitional query processing. In *Proc. IEEE ICDE*, April 2005.

[9] R. Fagin, A. Lotem, and M. Naor. Optimal aggregation algorithms for middleware. In *Proc. PODS*, August 2001.

[10] R. Gummadi, X. Li, R. Govindan, C. Shahabi, W. Hong. Energy-efficient data organization and query processing in sensor networks. In *Proc. IEEE ICDE*, April 2005.

[11] U. G$\ddot{u}$ntzer, W.-T. Balke, and W. Kie$\beta$ling. Optimizing multi-feature queries for image databases. In *Proc. VLDB*, 2000.

[12] Q. Han, S. Mehrotra, and N. Venkatasubramanian. Energy efficient data collection in distributed sensor environments. In *Proc. IEEE ICDCS*, March 2004.

[13] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Proc. ACM MobiCom*, August 2000.

[14] I. Kang and R. Poovendran. Maximizing network lifetime of broadcast over wireless stationary ad hoc networks. To appear in *ACM/Kluwer J. Mobile Networks and Applications (MONET)*, 11(2), April 2006.

[15] Y. Kotidis. Snapshot queries: Towards data-centric sensor networks. In *Proc. IEEE ICDE*, April 2005.

[16] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. TAG: A tiny aggregation service for ad-hoc sensor networks. In *Proc. USENIX OSDI*, pages 131–146, December 2002.

[17] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. The design of an acquisitional query processor for sensor networks. In *Proc. ACM SIGMOD*, June 2003.

[18] S. Michel, P. Triantafillou, and G. Weikum. KLEE: A framework for distributed top-k query algorithms. In *Proc. VLDB*, 2005.

[19] S. Nepal and M. V. Ramakrishna. Query processing issues in image (multimedia) databases. In *Proc. IEEE ICDE*, 1999.

[20] The network simulator - ns-2. http://www.isi.edu/nsnam/ns/.

[21] NRL's sensor network extension to ns-2. http://nrlsensorsim.pf.itd.nrl.navy.mil/.

[22] C. Olston, J. Jiang, and J. Widom. Adaptive filters for continuous queries over distributed data streams. In *Proc. ACM SIGMOD*, pages 563–574, June 2003.

[23] R. Szewczyk, E. Osterweil, J. Polastre, M. Hamilton, A. Mainwaring, and D. Estrin. Habitat monitoring with sensor networks. *Communications of the ACM*, 47(6):34–40, June 2004.

[24] M. A. Sharaf, J. Beaver, A. Labrinidis, and P. K. Chrysanthis. Balancing energy efficiency and quality of aggregate data in sensor networks. *VLDB Journal*, 13(4):374–403, Dec. 2004.

[25] N. Shrivastava, C. Buragohain, D. Agrawal and S. Suri. Medians and beyond: New aggregation techniques for sensor networks. In *Proc. ACM Sensys*, 2004.

[26] Tropical Atmosphere Ocean (TAO) Project. http://www.pmel.noaa.gov/tao/data_deliv.

[27] M. Theobald, G. Weikum, and R. Schenkel. Top-k query evaluation with probabilistic guarantees. In *Proc. VLDB*, August 2004.

[28] L. Xiong, S. Chitti, L. Liu. Top$k$ queries across multiple private databases. In *Proc. IEEE ICDCS*, June 2005.

[29] J. Xu, X. Tang, and W.-C. Lee. EASE: An energy-efficient in-network storage scheme for object tracking in sensor networks. To appear in *Proc. IEEE Conf. on Sensor and Ad Hoc Communications and Networks (SECON'05)*, September 2005. (Available at http://www.comp.hkbu.edu.hk/$\sim$xujl/ease.pdf.)

[30] O. Younis and S. Fahmy. Distributed clustering for ad-hoc sensor networks: A hybrid, energy-efficient approach. In *Proc. IEEE INFOCOM*, March 2004.

[31] C. T. Yu, G. Philip, and W. Meng. Distributed top-n query processing with possibly uncooperative local systems. In *Proc. VLDB*, 2003.

# On-Demand Broadcast Algorithms with Caching on improving Response Time for Real-Time Information Dispatch Systems [*]

Chui Ying Hui, Joseph Kee-Yin Ng

Department of Computer Science, Hong Kong Baptist University,
Kowloon Tong, Hong Kong
Tel:(852)-3411-7864 Fax:(852)-3411-7892
Email:{cyhui, jng}@comp.hkbu.edu.hk

## Abstract

*This paper presents a performance study on various broadcast algorithms and caching strategies for on-time delivery of data in a Real-Time Information Dispatch System. The objective of the study is not just aiming at on-time delivery, but to improve the response time on the data requests. We propose and perform a series of simulation experiments, using real traffic data from the access log of the official web site for FIFA 2002 World Cup. Simulation results show that our proposed broadcast algorithm not only succeeds in providing good on-time delivery of data but at the same time provides 2 to 3 times of improvement in response time over traditional scheduling algorithms like First-In-First-Out (FIFO) and Earliest-Deadline-First (EDF). The simulation results also show that our proposed caching strategy provides further improvement in percentage of requests finished in time over traditional caching strategy like Least Recently Used (LRU).*

## 1 Introduction

With the recent advances in wireless communications, satellite and cellular phone networks [19] have been developed and deployed to provide broadband access to the Internet for mobile users. The support of *broadcast* is a distinguished feature of these new delivery technologies over traditional wired networks. In contrast to *unicast*, where a data item must be individually transmitted to each client that requests it, broadcast allows many users requesting the same data to be satisfied by a single transmission of the data through a common channel.

There are basically two broadcast mechanisms: *pushed broadcast* determines the broadcast program based on collected statistics without user intervention; *on-demand broadcast* schedules items to be broadcast based on the current client requests that are submitted through an uplink channel. While pushed broadcast is useful for certain situations (e.g., small database, stable access pattern), on-demand broadcast is considered a more promising technique for dynamic and large-scale data dissemination. Hence, this paper focuses on on-demand broadcast.

The scheduling algorithm employed to select items among outstanding requests to broadcast is a key design of an on-demand broadcast system. Extensive studies have been carried out to develop on-demand scheduling algorithms in the literature (e.g., [1, 2, 11, 20]). However, all of these existing studies ignored the existence of timing constraints associated with data requests. Indeed, in many situations, a user request is associated with a *deadline*. A typical example is considering a traffic information server and a driver who, at some point ahead in the road, needs to take one of two possible routes in order to get to his/her destination [10, 13]. Clearly, it is necessary for the server to provide the driver with the desired traffic information (for example, one of the routes is congested) *before* the decision point is reached; otherwise the information is of no value to the driver.

Scheduling algorithms have also been investigated for real-time systems and pushed broadcast systems. Unfortunately, they are inapplicable or ineffective to the on-demand broadcast scenario. In [23], there is a scheduling algorithm SIN-$\alpha$ for time-critical on-demand broadcast. However, [23] considers fixed-size data objects only. This paper based on our previous investigation for on-demand

[*] The work is supported in part by the RGC Earmarked Research Grant from the HKSAR Government under the grant number: RGC HKBU 2174/03E

broadcasting[24] and develops an improved scheduling algorithm for variable-size data objects which aims at not just to deliver the data objects on time but to improve the performance by reducing the response time and the introduction of cache on the mobile client side.

The rest of the paper is organized as follows. In Section 2, we present the related work and in Section 3, we discuss about the system model. We describe a number of scheduling algorithms and propose our algorithm with its improved variations in Section 4. We discuss the traditional caching strategy and our proposed caching strategy in Section 5. In Section 6, we discuss the simulation setup, the experiments and present the results of the performance evaluation. Finally, in Section 7, we summarize our research findings and discuss some possible future work.

## 2 Related Work

Scheduling algorithms, as a critical design issue in on-demand broadcast, have been extensively studied in the literature [20]. R×W, LTSF, MAX are among a number of recently proposed on-demand scheduling algorithms [1, 2]. However, none of the existing algorithms considers timing constraints in making the scheduling decision.

While there are a few studies on developing periodic broadcast programs with timing constraints [6, 12], the only two studies on on-demand broadcast with timing constraints are [22] and [10], which differ from the former in that the scheduling relies on current queue status instead of precompiled access pattern. In [22], Xuan et al. evaluated several alternative mechanisms for serving data requests with deadlines through broadcast channels, including: pushed broadcast, unicast with EDF scheduling, on-demand broadcast with EDF scheduling, and hybrid pushed and on-demand broadcast. Their evaluation results showed that the on-demand broadcast with the EDF policy achieves good performance. In [10], Fernandez and Ramamritham studied an adaptive hybrid broadcast system that takes into account dynamic user access patterns and deadline constraints. Unfortunately, no attempts in the literature have been made for scheduling algorithms to exploit the properties of on-demand broadcast systems, where a transmission of a single data item may satisfy a number of pending requests.

A closely related area is task scheduling in real-time systems and databases. Many basic algorithms and theoretical results have been developed [5, 7, 15, 16, 17]. The objective of these scheduling algorithms is often to minimize the number of deadlines missed, or to maximize the effective processor utilization when the service times are variable during execution.

One of the most classical scheduling algorithms is the Earliest Deadline First (EDF) algorithm [14], which offers the optimal performance under various conditions in the sense that it meets all deadlines whenever it is feasible to do so [9]. In overload conditions, not all tasks can be completed by their deadlines. If the task service times are not available, EDF performs very poorly because it gives the highest priority to tasks that are close to missing their deadlines. As a consequence, the scheduled tasks are more likely to miss their deadlines during execution and further cause all subsequent tasks to miss their deadlines, thereby resulting in the “domino effect.” Two categories of techniques have been proposed to deal with this situation [7]: *guaranteed* algorithms, characterized by an admission control policy, and *robust* algorithms, characterized by a more sophisticated rejection strategy and a reclaiming mechanism. Unfortunately, all the existing scheduling algorithms are designed for a *unicast* environment where a newly arrived task cannot join any existing tasks. In contrast, this paper focuses on a broadcast environment where a new request can join an existing outstanding request when they request the same item and later they are served by a single process. This fundamental difference in system models motivates us to develop new scheduling algorithms.

Another related work is probably value-deadline task scheduling in real-time systems, where each task is characterized an importance value and the scheduling aims to maximize the cumulative value gained on a task set, i.e., the sum of the values of those tasks that completed by their deadlines [7]. At a first glance, our problem resembles the value-deadline scheduling if the number of requests posed on an item is thought of as the value of the item. This is however, not true as the number of requests on an item may change over time due to new arrivals and deadline expiration, but the value in value-deadline scheduling is always a constant. Due to this difference, the best policy, *value-density scheduling* (analogous to the MRF algorithm in our case), for value-deadline scheduling as observed in [7], shows a poor performance in on-demand scheduling, as we will see in Section 6.3.

Other related work concerning on-demand broadcast includes data staging [3], energy-efficient retrieval [8], client cache management [21], and fault-tolerant broadcast [4]. These studies complement to our work in different aspects.

## 3 System Model

Figure 1 shows the overall architecture for a typical on-demand broadcast system for real-time data dissemination. In this architecture, a large group of clients retrieve data items (e.g., web data objects) maintained by a database (e.g., a web server). If a client cannot find the requested data object in the cache or the data object in the cache is expired, the client sends a request to the server through an uplink channel. Each request is characterized by a 3-tuple: $< data\_id, atime, dline >$ where $data\_id$ is the unique

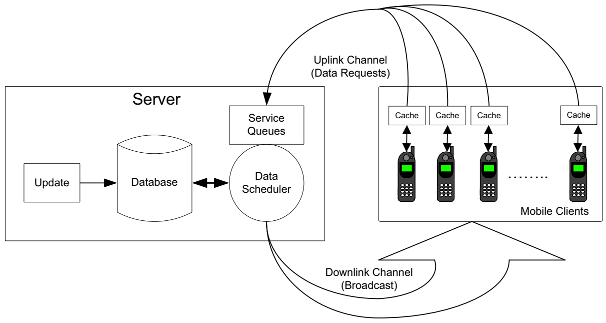


**Figure 1. Overall System Architecture**

identifier of the requested data object, $atime$ is the time of request, and $dline$ being the relative deadline for the request. Hence, the absolute deadline of the request is given by $atime + dline$, beyond which the reception of the requested data object will be of no use to the client. The client, on the other hand, monitors a downlink broadcast channel for the requested data objects until the lifetime of the request expires. We made an assumption that the uplink and downlink channels are independent.

Upon receiving a request, the server inserts it into the service queue to wait for broadcast. An outstanding request is said to be $active$ at time $t$ if its lifetime has not expired. An active request is called $feasible$ at time $t$ if it is still possible to be transmitted before its deadline. Otherwise, the active request is called $degenerated$, that is, it cannot meet its timing constraints. Degenerated requests are removed from the service queues when they are expired. Active requests remain in the service queues until they are served (transmitted) or degenerated, whichever takes place earlier.

All data objects are maintained by a database in the server. All data objects will be updated periodically. The server broadcasts data objects chosen from feasible requests based on the data scheduler. These selected data objects are broadcasted to the clients through the downlink channel with the corresponding requests removed from the service queues.

We assumed a non-preemptive broadcast system, i.e., when a request is being served, it is allowed to complete without interruption by request arrivals.

## 4 Scheduling Algorithms & Their Complexity

We have explored a number of broadcast/scheduling algorithms, ranging from the fairness algorithm, i.e., the First-In-First-Out (FIFO) algorithm, the most commonly adopted Earliest-Deadline-First (EDF) for real-time computing, the Most-Request-First (MRF) algorithm which is a greedy algorithm that serves the data object with the most requests to the Most-Request-Served (MRS) [24] algorithm which serves the data object with percentage of requests satisfied on time and at the same time produces the lowest average response time. With these algorithms as references, we proposed two new algorithm - Most-Request-Served with less object size(MRS-LOS) and Most-Request-Served with 2 level (MRS-2Level). MRS-LOS shows that it is indeed an all around algorithm that achieves a good performance in terms of percentage of requests satisfied on time and at the same time produces the lowest average response time.

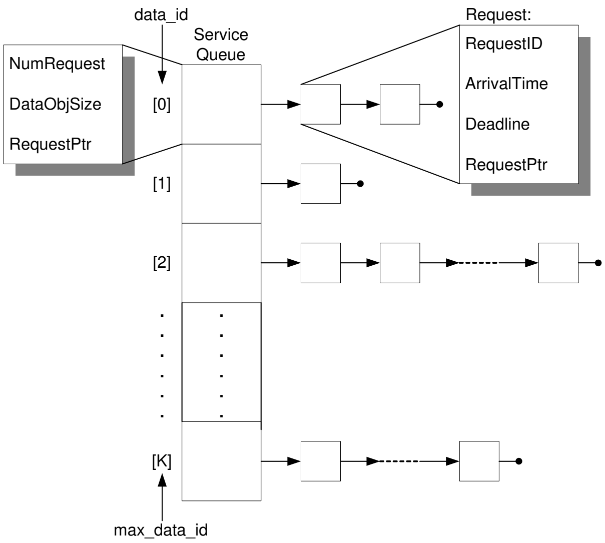


**Figure 2. Common Data Structure used among our Algorithms**

In the following sections, we will present each of the algorithms we have explored and discuss their scheduling complexity using the data structure we have constructed in our simulation experiments.

Figure 2 shows the common data structure we have adopted in our performance study. We used the same data structure among our scheduling algorithms. This data structure is basically an array of $K$ $ServiceQueues$ where $K$ is the maximum number of data objects to be served in the system. Each service queue is in turn serving a data object identified by a $data\_id$. As indicated in Figure 2, each array element of the Service Queues contains three attributes keeping track of the number of requests ($NumRequest$), the size of the data object ($DataObjSize$) , and a pointer pointing to a list of $Request$ ($RequestPtr$). This list of requests refers to each distinct request for the data object under the same $data\_id$ and each node contains the following information about the unique $RequestID$, its arrival time ($ArrivalTime$), and its $Deadline$.

### 4.1 First-In-First-Out & Earliest-Deadline-First

The most natural or the fairness algorithm for the broadcast strategy is First-In-First-Out (FIFO). Every request arrived at the system will be served according to its arrival time, or in our case, since we give every request an unique $RequestID$, the smallest $RequestID$ will be served first.

On the other hand, the Earliest-Deadline-First (EDF) strategy is most common for real-time system. With EDF, the request that has the earliest deadline will be served next. Since we assumed a non-preemptive broadcast system, that it, when a request is being served, it is allowed to finish without any interruption even though another request which has an earlier deadline had arrived. With the same relative deadline for all requests, the EDF algorithm is the same as the FIFO algorithm.

As for scheduling complexity, for the FIFO algorithm, we just search for the smallest $RequestID$. Hence, we have $O(N)$ for building the data structure as indicated in Figure 2, where $N$ is the total number of requests currently in the system. We then need $O(N)$ for the search of $\forall_{i=1}^{N} \min(RequestID_i)$, and $O(N)$ for removing all the requests with the same $data\_id$ from the data structure. Thus, the complexity of FIFO is $O(N)$.

On the other hand, for EDF, we need to search for the smallest $Deadline$. Hence, again we have $O(N)$ for building the data structure, $O(K)$ for the search of $\forall_{i=1}^{K} \min(Deadline_i)$, where $K$ is the number of distinct data objects within the system (i.e., $max\_data\_id$). We then still need $O(N)$ for removing all the requests with the same $data\_id$ from the data structure. Thus, the complexity of EDF is also $O(N)$ since $N > K$.

### 4.2 Most-Request-First

The approach of the Most-Request-First (MRF) algorithm is simple. It is a greedy algorithm that serves the data object with the most requests. Although not taking the size of the data object and the deadline into consideration, this scheduling algorithm is simple to implement and in fact produced very reasonable results in terms of response time.

Since we just need to search for the maximum number of requests, that is, $\forall_{i=1}^{K} \max(NumRequest_i)$, the complexity of such a search is $O(K)$. Together with the $O(N)$ for building the data structure and the $O(N)$ for removing all the requests with the same $data\_id$ from the data structure. Thus, the complexity of MRF is also $O(N)$ since $N > K$.

### 4.3 Most-Request-Served & Its Variations

Observing that a simple greedy algorithm like MRF already yields reasonably good results, and that the size of the data object and its corresponding deadlines should have some effects on the performance, we propose a new algorithm for the data scheduler that incorporates these attributes into our design.

Intuitively, the amount of slack time reflects the urgency when a data object has to be transmitted. $SlackTime = Deadline - (CurrentTime + \frac{DataObjSize}{ServiceRate})$. Now, instead of making use of slack time, we determine the latest time a request has to be served as $LatestStartTime$, which is defined as $Deadline - \frac{DataObjSize}{ServiceRate}$.

Knowing the $Deadline$, the $LatestStartTime$, and the $NumRequest$, i.e., the number of request, for each data object requested, we can calculate a score which reflects how good it is if this data object is broadcasted at this moment. In other words, the score predicts how much one gains and how much one loses if data object $i$ is broadcasted at this moment. This can be translated into the followings: At the current time, $t$, for each active request of data object $i$, if it is broadcasted, then at time $t'_i = t + \frac{DataObjSize_i}{ServiceRate}$, $NumRequest_i$ clients will be served. But at the same time, for all active data object $j$ such that $j \neq i$, and that if $t'_i > LatestStartTime_j$, one or more of the clients that request the data object $j$ will miss their deadlines. Adding how many requests will be served and how many requests will be missed, we come up with a score for each active data object requested in the database. And we chose to broadcast the data object that produces the highest score.

We can still make use of the common data structure to implement this algorithm, namely the Most-Request-Served (MRS) algorithm. For the scheduling complexity, we again need $O(N)$ to build the data structure. With the search of the data object $i$ that produces the highest score as indicated below:

$$\forall_{i=1}^{K} \max(NumRequest_i + \sum_{j=1}^{K} f(i,j))$$

with

$$f(i,j) = \begin{cases} 0 & \text{if } i = j \\ -NumRequest_j^{t'_i} & \text{Otherwise} \end{cases}$$

where $NumRequest_j^{t'_i}$ is the number of $Request_j$ that will miss its deadline by time $t'_i$.

We then need an $O(K^2)$ algorithm to do this search, where $K$ is the number of data objects in the database. Then we still need $O(N)$ for removing all the requests with the same $data\_id$ from the data structure. Thus, the complexity of MRS is $O(K^2 + N)$.

### Most-Request-Served with Less Object Size (MRS-LOS)

In the course of calculating the score for each data object requested, there is a chance to have more than one data objects that produce the same highest score. With that in mind, we propose a second attribute to be used as a tie-breaker when the scores are the same. With the same score for both data objects $i\&j$, we can break the tie by giving preference to whoever that has a smaller object size (MRS-LOS). We will conduct experiments to see if the second attributes will have any effect on the performance of MRS. As for the computation complexity, since we can use a variable to store this extra attribute to be the tie-breaker, it does not impose any extra computation complexity on the algorithm, hence, the scheduling complexity remains $O(K^2 + N)$.

### Most-Request-Served with 2 Levels

MRS calculates a score which reflects how much one gains and how much one loses if data object $i$ is broadcasted at the scheduling instance. By looking one step further, we calculate a score which reflects how good it is if this data object is broadcasted along with another data object. This can be translated into the followings: At the current time, $t$, for each active request of data object $i$, if it is broadcasted along with data object $j$ such that $j \neq i$, then at time $t'_{i+j} = t + \frac{DataObjSize_i}{ServiceRate} + \frac{DataObjSize_j}{ServiceRate}$, $(NumRequest_i$ + $(NumRequest_j$ - $NumRequest_j^{t'_i}))$ clients will be served where $NumRequest_j^{t'_i}$ is the number of $Request_j$ that will miss its deadline by time $t'_i$. But at the same time, for all active data object $k$ such that $k \neq i$ and $k \neq j$, and that if $t'_{i+j} > LatestStartTime_k$, $NumRequest_k^{t'_{i+j}}$ will miss their deadlines. Adding how many requests will be served and how many requests will miss their deadlines, we come up with a score for each group of active data object requested in the database. And we chose to broadcast the first data object of the grouped data objects that produces the highest score.

We can still make use of the common data structure to implement this algorithm, namely the Most-Request-Served with 2 Level(MRS-2Level) algorithm. For the scheduling complexity, we again need $O(N)$ to build the data structure. With the search of the data object $i$ that produces the highest score as indicated below:

$$\forall_{i=1}^{K}\forall_{j=1}^{K}|j \neq i \max \left\{ \begin{array}{l} NumRequest_i \\ +(NumRequest_j - NumRequest_j^{t'_i}) \\ +\sum_{k=1}^{K} g(i,j,k) \end{array} \right\}$$

with

$$g(i,j,k) = \begin{cases} 0 & \text{if } k = i \\ 0 & \text{if } k = j \\ -NumRequest_k^{t'_{i+j}} & \text{Otherwise} \end{cases}$$

where $NumRequest_k^{t'_{i+j}}$ is the number of $Request_k$ that will miss its deadline by time $t'_{i+j}$.

We then need an $O(K^3)$ algorithm to do this search, where $K$ is the number of data objects in the database. Then we still need $O(N)$ for removing all the requests with the same $data\_id$ from the data structure. Thus, the complexity of MRS-2Level is $O(K^3 + N)$. Since we used dynamic programming for the implementation of the MRS-2Level algorithm the runtime is still reasonable for practical use.

### Most-Request-Served with 3 Levels and beyond

Formally speaking, the broadcast schedule problem can be defined as:

At each scheduling instance of time $T$, there exists an optimal schedule $S$ for the set of $N$ requests for $K$ data objects with each Request $R_i$ having a deadline $D_i$ associated with it. Our problem is to find the first element of $S$, i.e., the first data object $O$ in the optimal schedule $S$ and broadcast it through the broadcast channel. After the broadcast, time is advanced to $T + \frac{DataObjSize(O)}{ServiceRate}$, with a set of $N'$ requests for $K'$ data objects with each Request $R_j$ having a deadline $D_j$ associated with it. And at this scheduling instance of time, i.e., $T + \frac{DataObjSize(O)}{ServiceRate}$, we have to find the optimal schedule $S'$.

As we define optimality is on minimal number of requests missing their deadlines, finding the optimal schedule $S$ for the given set of requests is NP-complete. It will be too computation intensive to be useful if we find the optimal solution, hence we should look for heuristics.

The search for the optimal schedule $S$ is analogous to the search for the best move in a chess game. We can always do an exhaustive search and find the best move. But for practical purpose, people will cut the search short by searching the first few levels and then give each chess board a score and do a min-max decision. In search of the optimal broadcast strategy, as we perform the search for more levels, the run-time increases exponentially. Thus, it is almost too long to be practical for searching at Level 3 and beyond. Hence, we just perform the search up to Level 2.

## 5 Caching Strategies

In this section, we examine the Least-Recently-Used (LRU) caching strategy at the client side. Using this algorithm as reference, we proposed two new strategies - Least-Recently-Used-Minus-Expired (LRU-expire) and Least-Slack-First (LSF). In the following sections, we will present each of the algorithms we have studied on caching strategies.

Figure 3 shows the common cache structure we have adopted in our performance study. We used the same cache structure among our caching strategies. This caching structure is a pointer array of $CacheQueues$. As indicated in

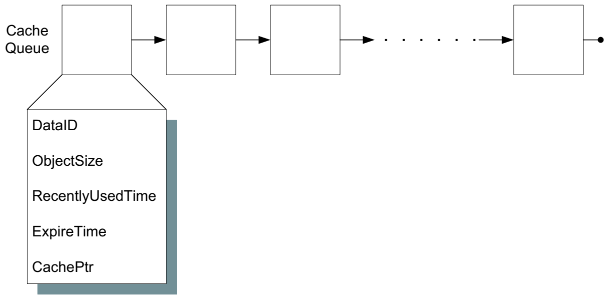


**Figure 3. Common Cache Structure used among our Strategies**

Figure 3, each array element of the Cache Queues contains five attributes keeping track of the Data ID ($DataID$), the size of the data object($ObjectSize$), the time the data object was last referenced($RecentlyUsedTime$), the time that the data object no longer up-to-date ($ExpireTime$) and a pointer pointing to the next element($CachePtr$).

### 5.1 Least-Recently-Used

The most common caching strategy is Least-Recently-Used (LRU). It is a scheme to select which cache entry to be flushed. When a new cache entry is brought into the cache and there is not enough space for the new cache entry, one or more of the existing cache entry must be replaced. This scheme is based on temporal locality - the observation is that, in general, the cache entry which has not been accessed for the longest time is least likely to be accessed in the near future. Each cache entry is associated with an expiry time, If the client requests the data object in cache is expired, the cache will throw away that entry and the client will send a request to the server asking for the most up-to-date copy of the data object.

### 5.2 Least-Recently-Used-Minus-Expired

Since data objects are updated periodically at the database, data objects in the client cache will expire and their values become invalid. As LRU did not handle the expired data objects, LRU will then have expired data objects occupying a certain amount of space in the cache and replace the least recently used data object. Base on this observation, we propose a new scheme namely, the Least-Recently-Used-Minus-Expired (LRU-expire). Before we insert a new cache entry, we remove all the expired data objects in the cache. If there is no expired data objects and not enough space in the cache for the new cache entry, the data object which has not been accessed for longest is replaced.

### 5.3 Least-Slack-First

From our previous observation, the expiry time and the size of the data object should have some effect on the performance. Since every data object can be of different size, we may as well take the transmission time of the data object into consideration. Hence, we examine the slack time for each request and define $SlackTime$ as follows:

$$SlackTime = ExpireTime - (CurrentTime + \frac{DataObjSize}{ServiceRate})$$

With $SlackTime$ defined, we propose yet another scheme - Least-Slack-First (LSF) as the caching strategy for our system. The amount of slack time reflects the freshness of a cache entry. When a new cache entry is brought into the cache and there is not enough space for the new cache entry, the cache entry with the least $SlackTime$ will be replaced.

## 6 Performance Evaluation

We have constructed and conducted a series of simulation experiments to look into the performance of each scheduling algorithm as well as the caching strategies on our broadcasting system. In the following sections, we will discuss about the simulation setup, the workload being used, performance metrics, and finally present our research findings in these experiments.

### 6.1 Simulation Experiments

#### 6.1.1 Setup

We have written a simulator using C++ to simulate the mobile transactions and the data dissemination and all the simulation experiments were executed on the Windows XP platform with an Intel 2.4GHz CPU. As indicated by Figure 1, we simulate a server holding up all the data objects in a database, and keep receiving requests from a number of simulated clients through the uplink channel. The data request traffic is actually straight from the data access log for the FIFA 2002 World Cup web site which will be described in detail below. When a request arrives at the server, it will be put to the corresponding Service Queue according to their $data\_id$, thus updating the $NumRequest$ for each service queue. Before picking which request to be served next, the system will check if any of the individual request is degenerated (that is, missed its deadline). The system will remove all those requests who had missed their deadlines and pass them to the statistic collector. With feasible requests in the service queues, the adopted scheduling algorithm will have to choose a data object to be broadcasted through the downlink channel. We assume here that the broadcast is non- preemptive such that once the data object is chosen to be broadcasted, it will not be interrupted by any

| Items | Values |
|---|---|
| Total Number of Requests $(N)$ | 7,149,766 |
| Total Number of Data Objects $(K)$ | 23,904 |
| Maximum Data Object Size | 2,891,887 Byte |
| Average Data Object Size | 5725.84 Byte |
| Average Inter-arrival Time | 0.012084 second |
| Average Bandwidth | 3.615 Mbps |

**Table 1. A Summary on the Workload used in our study**

recent request arrivals. After the broadcast, the corresponding service queue will be emptied and served requests will be forward to the statistic collector for analysis.

### 6.1.2 Workload being used

In order to test our algorithm rigorously, we use real traffic data from the log file of the FIFA 2002 World Cup web site. The log we used was recorded on Day 38, which is the date for the World Cup final. The log file is about 150 Mbyte in size and it basically contains three kinds of information that is needed for our simulation. It includes the arrival time, the data object identifier, and the size of the data object. Table 1 shows a summary on the data we used to test our algorithms throughout the simulation experiments.

## 6.2 Performance Metrics

In this study, we concern about the real-time performance on broadcast strategies. Since each request has its own deadline to meet, the number of requests that can finish on time naturally becomes one of the performance indicator. Beside that, if we look from the users' point of view, response time is the critical performance indices. Hence, in our simulation experiments, performance are measured by the followings:

**Percentage Finished in Time** Percentage Finished in Time indicates how many requests can be served before their deadlines expired. Hence,

$$\text{Percentage Finished in Time} = \frac{\text{n}}{\text{N}}$$

$$\text{where} \left\{ \begin{array}{l} \text{n = Number of Requests Finished in Time} \\ \text{N = Total Number of Requests in the System} \end{array} \right.$$

**Response Time** We define response time as the time between the time of request and the time of receiving the data object requested.

**Expected Response Time** Since the system will remove the degenerated request from the service queues, average response time alone will then be misleading. One possible scenario is that a scheduling algorithm may provide a small average response time but in fact it has discarded a high percentage of requests for the trade off. Hence, we defined a way to calculate the Expected Response Time by the following equation:

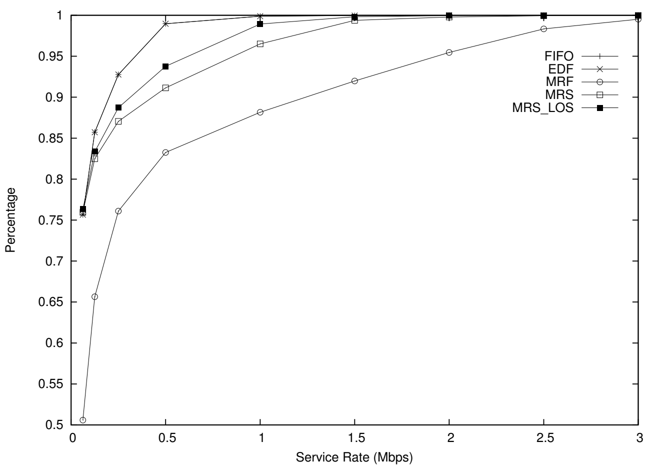


**Figure 4. Service Rate vs. Percentage Finished in Time**

$$\text{Expected Response Time} = \frac{\sum_{i=1}^{M} h(i)}{M},$$

where $M$ is the total number of requests in the system and

$$h(i) = \left\{ \begin{array}{ll} Deadline_i & \text{if it misses its deadline} \\ ResponseTime_i & \text{otherwise} \end{array} \right.$$

## 6.3 Simulation Results

### 6.3.1 Effect of Different Service Rates

We will first look at the effect of different service rates. The service rate is determined by the bandwidth of the broadcast channel. If there is enough bandwidth, then it does not matter which algorithm is adopted, and the data requests can be served before their deadlines. Hence, it is when the bandwidth is not enough, then the choice of algorithm will make a difference.

In Figure 4, we varied the service rate from 64kbps to 3Mbps, and the percentage of data requests finished in time for different algorithms is shown. We fixed the deadline for each request at 30 seconds, cache size at 100 Kbyte with the LRU caching strategy and 10 seconds data freshness, that is, a data object is updated every 10 seconds, and we assumed that all scheduling algorithms are non-preemptive. All algorithms show the trend that when service rate is low, the percentage of requests finished in time is low, and when

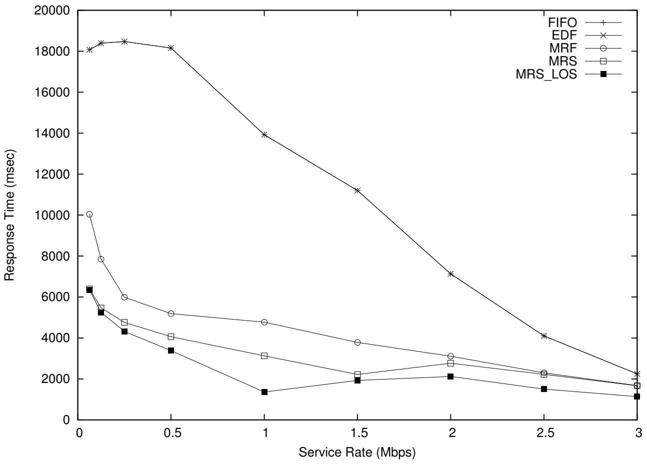


Figure 5. Service Rate vs. Response Time

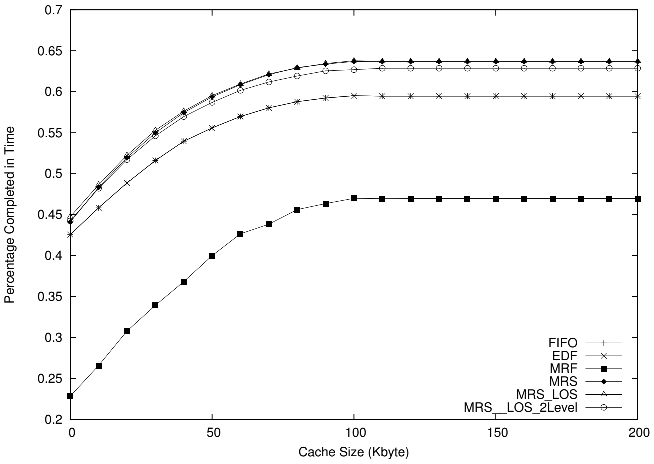


Figure 6. Percentage Finished in Time

we increase the service rate, all algorithms show an increase in percentage of requests finished in time. Recall that we fixed a constant deadline of 30 seconds for each request and the broadcast algorithms are non-preemptive. Under these assumption, the schedule by FIFO and EDF are exactly the same. And in fact, EDF and FIFO perform the best in terms of percentage of requests finished in time.

The MRF algorithm does not perform as well especially when the service rate is really low (i.e., at 64kbps). In the meantime, our proposed MRS-LOS perform reasonably close to the EDF and FIFO algorithm at both ends and is better than MRS.

However, the percentage of requests finished in time is just one of the performance measures of our concern. When we look at the average response time, a clearer picture can be drawn. Figure 5 shows the average response time at various service rates with different scheduling algorithms. From Figure 5, we can observe that as we increase the service rate from 64kbps to 3Mbps, the average response time decreases accordingly.

For FIFO and EDF, although they perform the best in term of percentage finished in time, they are the ones of the worst algorithms in terms of average response time. While MRF is known to provide good response time at the expense of some requests missing their deadlines, our proposed MRS-LOS performs the best in terms of average response time and at the same time shows only a little sacrifice in terms of requests missing their deadlines. Thus, MRS-LOS is the best in terms of overall performance.

### 6.3.2 Effect of Different Cache Size

In the previous section, we varied the service rates and looked into the average response time and percentage finished in time with a fixed deadline of 30 seconds. In this section, we look into the effect of different cache size at a fixed service rate, deadline and data freshness.

From our previous observation, the algorithms are mostly affected by a low service rate. Hence, we fixed the service rate at 64kbps, which is a comparable to the actual transmission rate of GPRS and a reasonable rate for the downlink channel, the deadline at 3 seconds and mean data freshness at 10 seconds with a standard deviation of 2.0.

Figure 6 shows the percentage of requests finished in time. We varied the cache size from no cache to 200 Kbyte with a step increment of 10 Kbyte. At 64kbps, and with the deadline set at 3 seconds and data freshness at 10 seconds, all algorithms show a low percentage of requests finished in time. As we increase the cache size gradually from 10 Kbyte to 200 Kbyte, all algorithms show an increase in percentage of request finished in time. We should notice that even when we set the cache size to 200 Kbyte, MRF is still performing badly. On the other hand, MRS's and MRS-LOS's performance are better than that of FIFO and EDF. We also exam the percentage of requests finished in time with deadline set at 15 seconds and 30 seconds. But it does not show any significant indications. Referring to Figure 6, we observe that there is not much performance gain between MRS-LOS-2Level and MRS-LOS. We also believe that the performance will not be improved significantly even if we seek for higher level search. Hence, we think that MRS-LOS is a good approximate algorithm for the optimal solution.

Since performance should be a combination of response time and percentage of requests finished in time, we try to merge these two factors together and come out with a good comparison among the scheduling algorithms. Thus, we employ a performance measure called *Expected Response Time*. Since we set the deadline for each request, if the request misses its deadline, it will be given a response time

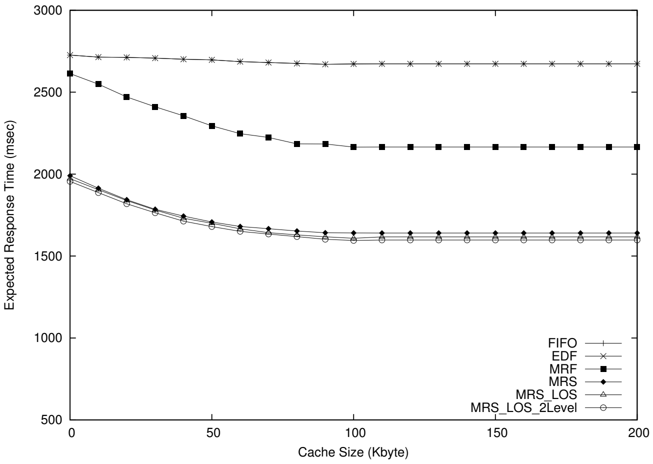


**Figure 7. Expected Response Time**

of its deadline. This is like when the deadline is reached, the system will ”respond” to the user that his/her request is removed from the system.

Figure 7 shows the expected response time for each algorithm with different cache size at a service rate of 64kbps, deadline at 3 seconds and mean data freshness at 10 seconds with a standard deviation of 2.0 . It shows that our proposed algorithm - MRS-LOS, out performs all other algorithms. In particular, with a cache size of 200 Kbyte, the expected response times in seconds for MRF, FIFO, EDF, MRS and MRS-LOS are 21, 26, 26, 16, and 15, respectively. MRS-LOS is 42% better than FIFO & EDF. Figure 7 shows that our proposed algorithm - MRS-LOS-2Level performs a little bit better than MRS-LOS.

### 6.3.3 Effect of Different Cache Strategy

In this section, we look into the effect of different cache strategies at a fixed service rate, deadline and mean data freshness.

From our previous observation, the algorithms are mostly affected by a low service rate. Hence, we fixed the service rate at 64kbps, the deadline at 3 seconds and mean data freshness at 10 seconds with a standard deviation of 2.0.

Figure 8 shows the percentage of requests finished in time. We varied the cache size from no cache to 200 Kbyte with a step increment of 10 Kbyte. At 64kbps, with the deadline set at 3 seconds and mean data freshness at 10 seconds with a standard deviation of 2.0, all strategies show a low percentage of requests finished in time. As we increase the cache size gradually from 10 Kbyte to 100 Kbyte, all strategies show an increase in percentage of request finished in time. We should notice that when the cache size gradually increase from 100K to 200K, all strategies becomes flat. It is due to with larger cache size, most of the cache entry in the cache is expired. Even the data object is in the cache, it will also be counted as a cache miss. When cache size is between 40 Kbyte and 100 Kbyte , our proposed strategy LRU-expire performs better than LRU. Since LRU only flush out the one with least reference, the one with least reference may still be fresh and the expired cache entry may then be kept in the cache. LEF performs worst all the time excepts when cache size is between 70 Kbyte and 100 Kbyte. We also try LEF-LRU which flushes the cache entry by LEF first, if there is no expired data objects in the cache, it flushes the cache entry according to LRU. However, in this case, the LEF-LRU’s performance make no different with LEF.

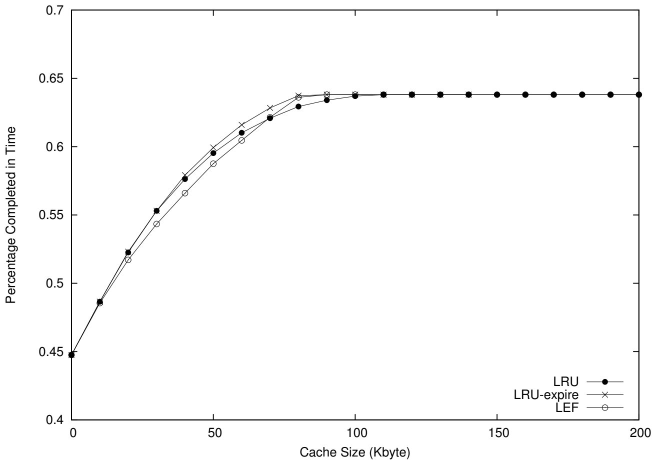


**Figure 8. Percentage Finished in Time**

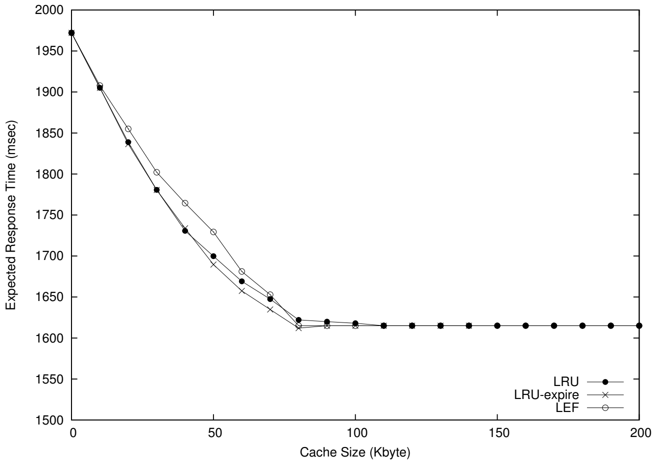


**Figure 9. Expected Response Time**

Figure 9 shows the expected response time for each strategies with different cache size at a service rate of 64kbps, deadline at 3 seconds and mean data freshness at

| Algorithm | No Cache | Cache |
|---|---|---|
| EDF | 2726.695 | 2672.701 |
| MRF | 2613.739 | 2165.137 |
| MRS | 2005.561 | 1660.970 |
| MRS-LOS | 1972.171 | 1608.7618 |

**Table 2. A Summary on the Expected Response Time (in msec) with deadline at 3 Seconds**

| Algorithm | No Cache | Cache |
|---|---|---|
| EDF | 0.4257 | 0.5952 |
| MRF | 0.2286 | 0.4700 |
| MRS | 0.4449 | 0.6397 |
| MRS-LOS | 0.4475 | 0.6379 |

**Table 3. A Summary on the Percentage Finished in Time**

10 seconds with a standard deviation of 2. It shows that our proposed strategies - LRU-expire performs better when the cache size is between 50 Kbyte and 90 Kbyte. When the cache size is gradually increased from 90 Kbyte to 200 Kbyte, all strategies becomes stable. As we mentioned before, it is due to most of the cache entry in the cache is expired.

## 7 Summary & Future Work

In this paper, we presented a performance study on various broadcast algorithms and caching strategies for on-time delivery of data in a Real-Time Information Dispatch System. The objective of the study is not just aiming at on-time delivery, but to improve the response time on the data requests and percentage of requests completed in time. We have performed a series of simulation experiments, using real traffic data from the access log of the official web site for FIFA 2002 World Cup. Table 2 & 3 gave a summary of the results for our simulation experiments. Simulation results show that our proposed broadcast algorithm not only succeeds in providing good on-time delivery of data but at the same time provides 2 to 3 times of improvement in response time over traditional scheduling algorithms like First-In-First-Out (FIFO), and Earliest-Deadline-First (EDF). We also found out that with limited service rate and when the deadline is tight, our proposed MRS-LOS performs the best among the algorithms we have investigated. Simulation results also show that our proposed caching strategy LRU-expire improves the percentage of requests completed in time and also provides improvement in expected response time over Least-Recently-Used (LRU).

As for future work, we will take a closer look on data freshness, transaction deadlines, and cache size on the client side. We will also look at sets of related data objects involved in a transaction rather than independent data objects. We are also interested in a mix of real-time data and non real-time data in the database, and the use of multiple versions of the data to trade off between accuracy and transactions meeting their deadlines. All these could be affected by different broadcast algorithms.

## References

[1] S. Acharya and S. Muthukrishnan. Scheduling on-demand broadcasts: New metrics and algorithms. In *Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'98)*, pages 43–54, Dallas, TX, USA, October 1998.

[2] D. Aksoy and M. Franklin. R x W: A scheduling approach for large-scale on-demand data broadcast. *IEEE/ACM Transactions on Networking*, 7(6):846–860, December 1999.

[3] D. Aksoy, M. J. Franklin, and S. Zdonik. Data staging for on-demand broadcast. In *Proceedings of the 27th International Conference on Very Large Data Bases (VLDB'01)*, pages 571–580, Roma, Italy, September 2001.

[4] S. K. Baruah and A. Bestavros. Pinwheel scheduling for fault-tolerant broadcast disks in real-time database systems. In *Proceedings of the 13th International Conference on Data Engineering (ICDE'97)*, pages 543–551, Birmingham, UK, April 1997.

[5] S. K. Baruah and J. R. Haritsa. Scheduling for overload in real-time systems. *IEEE Transactions on Computers*, 46(9):1034–1039, September 1997.

[6] A. Bestavros. AIDA-based real-time fault-tolerant broadcast disks. In *Proceedings of the 3rd IEEE Real-Time Technology and Applications Symposium (RTAS'96)*, pages 49–58, Boston, MA, June 1996.

[7] G. Buttazzo, M. Spuri, and F. Sensini. Value vs. deadline scheduling in overload conditions. In *Proceedings of the 16th IEEE Real-Time Systems Symposium (RTSS'95)*, pages 571–580, Pisa, Italy, December 1995.

[8] A. Datta, D. E. VanderMeer, A. Celik, and V. Kumar. Broadcast protocols to support efficient retrieval from databases by mobile users. *ACM Transactions on Database Systems (TODS)*, 24(1):1–79, March 1999.

[9] M. L. Dertouzos. Control robotics: The procedural control of physical processes. *Information Processing*, 1974.

[10] J. Fernandez and K. Ramamritham. Adaptive dissemination of data in time-critical asymmetric communication environments. In *Proceedings of Euromicro Real-Time Systems Symposium*, pages 195–203, 1999.

[11] S. Hameed and N. H. Vaidya. Efficient algorithms for scheduling data broadcast. *ACM/Baltzer Journal of Wireless Networks (WINET)*, 5(3):183–193, 1999.

[12] S. Jiang and N. H. Vaidya. Scheduling data broadcast to impatient users. In *Proceedings of International Workshop on*

*Data Engineering for Wireless and Mobile Access*, Seattle, WA, August 1999.

[13] D. L. Lee, J. Xu, B. Zheng, and W.-C. Lee. Data management in location-dependent information services. *IEEE Pervasive Computing*, 1(3):65–72, July-September 2002.

[14] C. L. Liu and J. W. Layland. Scheduling algorithms for multiprogramming in a hard real-time environments. *Journal of ACM*, 20(1):46–61, 1973.

[15] P. Mejia-Alvarez, R. Melhem, and D. Mosse. An incremental approach to scheduling during overloads in real-time systems. In *Proceedings of the 21th IEEE Real-Time Systems Symposium (RTSS'00)*, pages 283–293, Orlando, FL, November 2000.

[16] G. Ozsoyoglu and R. T. Snodgrass. Temporal and real-time databases: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 7(4):513–532, August 1995.

[17] J. A. Stankovic, M. Spuri, M. D. Natale, and G. C. Buttazzo. Implications of classical scheduling results for real-time systems. *IEEE Computer*, 28(6):16–25, 1995.

[18] C. J. Su, L. Tassiulas, and V. J. Tsotras. Broadcast scheduling for information distribution. *ACM/Baltzer Journal of Wireless Networks (WINET)*, 5(2):137–147, 1999.

[19] Hughes Network Systems. DIRECWAY homepage. Website at http://www.direcway.com/.

[20] J. W. Wong. Broadcast delivery. *Proceedings of the IEEE*, 76(12):1566–1577, December 1988.

[21] J. Xu, Q. L. Hu, W.-C. Lee, and D. L. Lee. Performance evaluation of an optimal cache replacement policy for wireless data dissemination. *IEEE Trans. on Knowledge and Data Engineering (TKDE)*, 2004.

[22] P. Xuan, S. Sen, O. Gonzalez, J. Fernandez, and K. Ramamritham. Broadcast on demand: Efficient and timely dissemination of data in mobile environments. In *Proceedings of the 3rd IEEE Real-Time Technology and Applications Symposium (RTAS '97)*, pages 38–48, Montreal, Canada, June 1997.

[23] J. Xu, X. Tang, and W. Lee. Time-Critical On-Demand Broadcast: Algorithms, Analysis, and Performance Evaluation. In *Technical Report*, COMP-03-015, Department of Computer Science, Hong Kong Baptist University.

[24] J. Ng, J. Xu. On-Demand Broadcast Algorithms on improving Response Time for Real-Time Information Dispatch Systems. In *Proceedings of the 10th International Conference on Real-Time and Embedded Computing Systems and Applications (RTCSA 2004).*

# An Improved Ellipse Propagation Model for Location Estimation in facilitating Ubiquitous Computing

Junyang Zhou
Department of Computer Science, HKBU
Kowloon Tong, Hong Kong
Email: jyzhou@comp.hkbu.edu.hk

## Abstract

*Mobile location estimation is a crucial technology for ubiquitous computing. A directional propagation model - the Ellipse Propagation Model (EPM) is proposed by our research group for locating a mobile station (MS) within a radio cellular network with an accuracy that can enable a number of location based services to realize ubiquitous computing. EPM assumes the contour line of signal strength resembles an ellipse with the base station situates at one of the focus. By using a Geometric Algorithm, the location of the mobile station can be estimated. However, since one parameter in our Geometric Algorithm is fixed, errors may be induced as the surrounding environment changes. In view of this, we would like to propose a new algorithm - the Iterative Algorithm to estimate the location of mobile station based on EPM. With the technical support of two local mobile phone operators, we have conducted a series of experiments using real data and experiment results showed that the proposed Iterative Algorithm outperforms the Geometric Algorithm by a good margin of 18% in terms of average error.*

## 1 Introduction

Recently, mobile location estimation is receiving considerable attention in the field of wireless communications. Many positioning technologies have been developed. The most famous location system is The Global Positioning System (GPS) is one of the location systems that is mature enough and commercially available [1, 2]. Other proposed location estimation methods include Time-Of-Arrival (TOA), Time Difference Of Arrival (TDOA), Enhanced Observed Time Difference (E-OTD) and Angle-Of-Arrival (AOA) [3]. These positioning technologies are based on timing information or angular information. Time based methods, such as, TOA, TDOA and E-OTD, calculate the distance between the Mobile Station (MS) and the Base Station (BS) by measuring the propagation time of the signal and multiply it by the speed of light. By using trilateration, the position of the MS can be estimated. On the other hand, angular approaches, like AOA, measure the angle between the MS and the BS and then estimated the location of the MS by using triangulation. Although these positioning technologies are simple, these approaches are only applicable to CDMA system since it can provide the timing or angular information. However, quite a number of countries have adopted the GSM network instead of the CDMA network. And the GSM network can only provide the loss of signal strength due to signal attenuation [4].

Since the loss of signal strength is the common attribute of all radio cellular network, thus location estimation algorithms proposed here are applicable to all radio cellular network for ubiquitous computing.

Our group has also proposed several location estimation approaches based on the received signal strength (RSS) [5, 6, 7]. However, these methods have not included the directional properties of the antenna. From our observations, we found that the BSs have directional properties. That is, BSs always transmit signal in a direction. In view of that, our research group has proposed an Ellipse Propagation Model (EPM) in [8]. EPM is derived from the original propagation model [3]. We observed that the antenna transmits the signal in a direction. Thus, the contour line of signal strength should not be a circle. We therefore modify the original propagation model by considering the contour line of signal strength as an ellipse with the BS sitting at one of the focuses. Since the RSS is the only attribute we have, therefore EPM focuses on the relationship between the MS-BS distance and the RSS. We also proposed a Geometric Algorithm to estimate the location of the MS based on EPM. Experiment results have proven that EPM using the Geometric Algorithm is very encouraging. However, one of the parameters in the Geometric Algorithm is fixed. This may cause some defects in the accuracy since the parameters may vary in different environment. Moreover, the Geometric Algorithm does not have self-modification property, namely, the estimation of the Geometric Algorithm is unstable. In view of this, we would like to propose a new approach to estimate the location of the MS with EPM - the Iterative Algorithm. The Iterative Algorithm has self-modification property. It chooses a convergence value as the estimation. And our experiment results have proven that the Iterative Algorithm is superior to the Geometric Algorithm with a trade off between accuracy and computational cost.

This paper is divided into five sections. In the following section, we will depict some location estimation algorithms proposed by our group. In section 3, the Iterative Algorithm will be presented. Afterwards, we will discuss the simulation results of the Iterative Algorithm using real data. And at last, the conclusion and future work for our research will be presented.

# 2 Related Works

Previously, our group has proposed two algorithms for location estimation, namely, the Center of Gravity (CG) algorithm and the Circular Trilateraion (CT) algorithm [5, 7]. Both CG and CT are making use of the RSS for estimating the position of the MS. They assumed the relationship between the MS-BS distance ($d$) and the RSS ($s$) is $s \propto d^{-2}$ based on the inverse square law [9]. However, due to the interference and distortion by buildings, the relationship is remodelled into $s \propto d^{-\alpha}$, where $\alpha$ is a variable being related to the environment.

We have also proposed an Ellipse Propagation Model (EPM) and a Geometric Algorithm to provide the location of the MS in [8]. The Ellipse Propagation Model (EPM) considers the antenna directional transmission property and assumes the contour line of the signal strength for one antenna as an ellipse which the base station is on one of the focuses. While the Geometric Algorithm is a simple and useful method to provide the location of the MS; it is derived from the CT algorithm and to reduce the defects of the CT algorithm.

## 2.1 Center of Gravity (CG)

The CG approach defines the location estimation formula as,

$$\begin{cases} x = \frac{x_1 s_1^{-\alpha} + x_2 s_2^{-\alpha} + x_3 s_3^{-\alpha} + \ldots + x_n s_n^{-\alpha}}{s_1^{-\alpha} + s_2^{-\alpha} + s_3^{-\alpha} + \ldots + s_n^{-\alpha}} \\ y = \frac{y_1 s_1^{-\alpha} + y_2 s_2^{-\alpha} + y_3 s_3^{-\alpha} + \ldots + y_n s_n^{-\alpha}}{s_1^{-\alpha} + s_2^{-\alpha} + s_3^{-\alpha} + \ldots + s_n^{-\alpha}} \end{cases} \tag{2.1}$$

where $(x, y)$ is the estimated location of the MS. $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$ are the locations of $n$ receiving BSs. $s_1, s_2, ..., s_n$ are the corresponding RSS from each BS [5].

Although the CG approach has proven its outstanding performance in metropolitan area, it can only estimate a mobile device inside the convex hull of the BSs involved.

## 2.2 Circular Trilateration (CT)

The basic idea of the CT approach is to construct 3 circles with the RSS for 3 different BSs. By knowing the location of the three BSs and the mapping between RSS and MS-BS distance, the intersection of these 3 circles is the estimated location of the MS. Similar to CG, CT models the relationship between the MS-BS distance ($d$) and the RSS ($s$) as $d \propto (N + s)^{-\alpha}$, where $N$ is the normalization constant. By making use of this relationship, we can constructed 3 circles as follows,

$$\begin{cases} (x - x_1)^2 + (y - y_1)^2 = (\frac{k}{s_1{}^{\alpha}})^2 \\ (x - x_2)^2 + (y - y_2)^2 = (\frac{k}{s_2{}^{\alpha}})^2 \\ (x - x_3)^2 + (y - y_3)^2 = (\frac{k}{s_3{}^{\alpha}})^2 \end{cases} \tag{2.2}$$

where, $s_1$, $s_2$ and $s_3$ are the RSSs from 3 receiving BSs with their geographic locations as $(x_1, y_1)$, $(x_2, y_2)$ and $(x_3, y_3)$ respectively and $k$ be a common scaling factor. The location of the MS is then estimated as the intersection point of these 3 circles [7].

Although CT does not have the convex hull problem, it does not always provide an estimation. This is because intersection may not always appear due to signal fading. Moreover, the CG and the CT approaches do not take the transmission direction of the BS into account, which is not realistic. Thus, we have designed a new approach, the EPM which is an improvement of the CT algorithm.

## 2.3 The Ellipse Propagation Model

From our observations, we found that the BSs have directional transmission property. The antenna transmits the signal in some directions. That is, the antenna transmits the largest power in one direction,

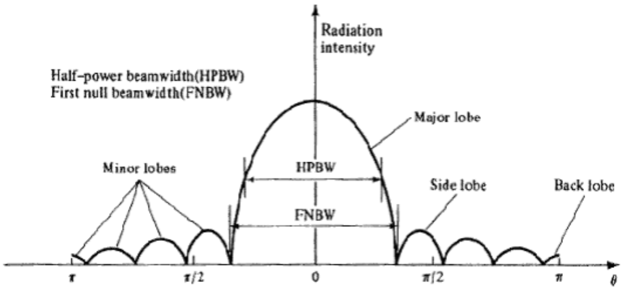


Figure 1. The directional antenna

while transmits small or none power in other directions. We can plot the contour line of signal strength around an antenna as **Figure 1**. As shown in the Figure 1, the contour line of the signal strength for the directional antenna is not a circle, which violates the assumption of the original propagation model in free space.

We assume the contour line of the signal strength as an ellipse instead of a circle based on our studies. The EPM adopts a different relationship between the MS-BS distance and the RSS. We can use the following mathematical formula to depict the EPM.

$$d = k(s_0/s)^{1/\alpha}(1-e)/(1-e\cos(\theta)) \tag{2.3}$$

where

$d$ is the distance between MS and BS;
$k$ is the proportion constant;
$s_0$ is the transmitting power of the BS;
$s$ is the signal power received;
$e$ is the eccentricity value of the ellipse, it describes the figure of the signal strength contour line;
$\theta$ is the deviation between the ellipse principal axis and the line of MS and BS;
$\alpha$ is called the path loss exponent.

We call this relationship as the Ellipse Propagation Model (EPM), where the contour line of the signal strength is an ellipse [8]. The Ellipse Propagation Model (EPM) can be illustrated in **Figure 2**.

The EPM has four parameters: $k$, $\alpha$ , $e$ and $\theta$. We consider the parameters of $k$ and $\alpha$ as the region parameters, and the parameter $e$ is used to describe the figure of the ellipse, while the deviation $\theta$ is parameter for each MS. We can use the field test data to find out the values of these parameters, then translate the signal strength into two points distance. That is the main idea of the EPM.

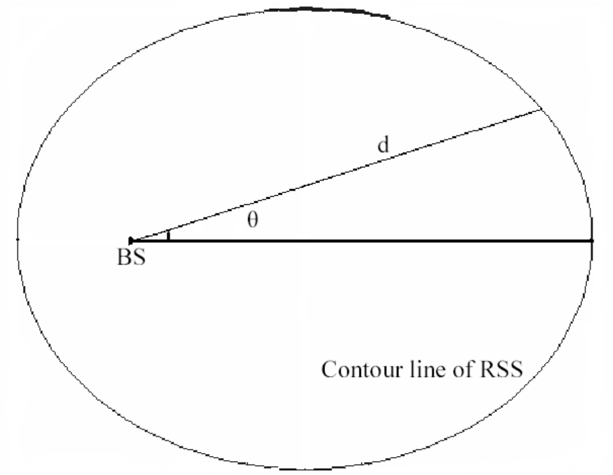


Figure 2. The Ellipse Propagation Model(EPM)

## 2.4 The Geometric Algorithm under EPM

The Geometric Algorithm is used to provide the estimation of the location of the MS. We choose the value of deviation between the ellipse principal axis and the line of the BS and the center (or the weighted center) of the BSs locations as the estimation of $\theta$. So the EPM exactly has three parameters: $k$, $\alpha$ and $e$.

Suppose a MS receives RSS, $s_1$, $s_2$, $s_3$ from three BSs with locations, $(x_1, y_1)$, $(x_2, y_2)$, $(x_3, y_3)$ respectively. In addition, the distances between the MS and the BSs are denoted by $d(s_1)$, $d(s_2)$, $d(s_3)$, sometimes, and they are simply denoted by $d_1$,$d_2$ and $d_3$. Thus, by the formula of the two points distance in the 2-D Euclidian space, we can form three circles formulas which are shown as follows,

$$\left\{ \begin{array}{l} (x-x_1)^2 + (y-y_1)^2 = d_1^2 \\ (x-x_2)^2 + (y-y_2)^2 = d_2^2 \\ (x-x_3)^2 + (y-y_3)^2 = d_3^2 \end{array} \right. \tag{2.4}$$

Basically, the geometric interpretation of this equation group means three circles in 2-D space, and the solution is the intersection point of these circles. However, if an intersection point does not exist, we may not be able to provide an estimation for the MS. In order to solve this problem, the Geometric Algorithm can not solve this equation group directly. Instead, the Geometric Algorithm derives another three equation group from the equation group (2.4). As each new equation group can provide a solution, thus, we will have three solutions. The estimation of the MS will then be the center of these three solutions [8].

The estimation of the Geometric Algorithm is:

$$\begin{aligned} x &= (2m(y_3 - y_1) - 2n(y_2 - y_1))/|A| \\ y &= (-2m(x_3 - x_1) + 2n(x_2 - x_1))/|A| \end{aligned} \tag{2.5}$$

where

$|A| = 4[(x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1)]$;
$m = [d_1^2 - (x_1^2 + y_1^2)] - [d_2^2 - (x_2^2 + y_2^2)]$;
$n = [d_1^2 - (x_1^2 + y_1^2)] - [d_3^2 - (x_3^2 + y_3^2)]$;
$(x_1, y_1)$,$(x_2, y_2)$ ,$(x_3, y_3)$ are the BSs locations.

# 3 The Iterative Algorithm under EPM

In our previous research, we have proposed a EPM and a Geometric Algorithm to provide the location of the MS [8]. We based on these results to further develop an Iterative Algorithm under EPM. The basic idea of the Iterative Algorithm is to improve the Geometric Algorithm to provide a more accurate and more stable estimation.

The Geometric Algorithm based on EPM can always provide an estimation whenever the MS receives three or more antenna signals. However, the deviation between the major transmitting direction and the line of MS-BS, $\theta$, in the Geometric Algorithm is fixed and is dependent on the location of the MS. Hence, we present a self-modification method illumined by the Iterative Algorithm to provide the location of the MS and the value of deviation $\theta$. We name this method as the Iterative Algorithm. Since the $\theta$ in Geometric Algorithm is an approximate value only, some error may be induced in the estimation. Thus, the Iterative Algorithm should have better accuracy than the Geometric Algorithm by eliminating these errors.

The EPM with the Iterative Algorithm has four parameters: $k$, $\alpha$, $e$ and $\theta$. $k$ and $\alpha$ are the region parameters, and $e$ is the parameter to describe the shape of the contour lines. They all can be provided by training with the field test data. Thus, parameter $\theta$ and the location of the MS can be calculated with the Iterative Algorithm. In the Iterative Algorithm, we do not use a constant value to estimate $\theta$, but update the estimation of $\theta$ for each iteration. Since the value of $\theta$ depends on the location of the MS, the estimation of the location of the MS can be used to calculate the value of $\theta$.

We choose an initial value to calculate the value of $\theta$. Then, the distance between MS and BS is derived, and a solution is calculated by the Geometric Algorithm. This solution is a new location of the MS. For each iterative computing, we update the values of $\theta$ and the location of the MS. If the series of the locations of the MS have a convergence value, we choose the stabilized convergence value as the estimation of the location of the MS.

## 3.1 Structure of the Iterative Algorithm

Let $(x, y)$ be the coordinates of the location of the MS, and $x, y$ are two independent random variables. We consider the estimation of the location of the MS as a conditional expectation of the RSS and the locations of BSs, denoted by,

$$(x', y')^T = E((x, y)^T | x_0, y_0; s; l) \tag{3.1}$$

where $x_0$ and $y_0$ are the location of the MS we want to find; $s$ is the information of RSS, and $l$ is the location information of BSs; $x'$ and $y'$ are the estimation of the location of the MS, and they are random variables too.

By rewriting the exact location of the MS with $x$ and $y$ as two parts, we have,

$$\begin{cases} x = f(x_0, y_0; s; l) + \epsilon \\ y = g(x_0, y_0; s; l) + \eta \end{cases} \tag{3.2}$$

where $f(x_0, y_0; s, l)$ and $g(x_0, y_0; s, l)$ are the certain terms; $x_0$ and $y_0$ are the estimations of $x$ and $y$; $\epsilon$ and $\eta$ are random variables; $f$ and $g$ are some functions structures, which have first order and second order derivatives. Furthermore, we assume that $E(\epsilon) = 0$, $E(\eta) = 0$, $var(\epsilon) = \sigma_\epsilon^2$, $var(\eta) = \sigma_\eta^2$, $cov(\epsilon, \eta) = 0$ , $x_0$ and $y_0$ are unbiased estimations of $x$ and $y$. Thus,$E(x - x_0) = 0$ ,$E(y - y_0) = 0$.

We choose the certain terms as the estimation of the location of the MS.

$$\begin{cases} x_0 = f(x_0, y_0; s; l) \\ y_0 = g(x_0, y_0; s; l) \end{cases} \tag{3.3}$$

Since $x_0$ and $y_0$ appear in both sides, the iterative method can be considered to provide the solution. We call the equation group (3.3) as the structure of the Iterative Algorithm.

Therefore, by picking an initial value, for example, the estimation from the CG algorithm $(x_{CG}, y_{CG})$, we can provide the estimation of the MS if the iterative formulas are convergent.

## 3.2 Using Geometric Algorithm with the Iterative Algorithm

Suppose the MS, with location $(x, y)$, received RSS, $s_1$, $s_2$, $s_3$ from three BSs at locations, $l_1(\alpha_1, \beta_1)$, $l_2(\alpha_2, \beta_2)$, $l_3(\alpha_3, \beta_3)$ and the output powers, $o_1, o_2, o_3$ respectively. In addition, the distances between the MS and the BSs are denoted by $d_1$, $d_2$, $d_3$.

We depict the Iterative Algorithm by using the Geometric Algorithm structure based on EPM. We can get the following formulas in [8]

$$\begin{cases} x = (2m(\beta_3 - \beta_1) - 2n(\beta_2 - \beta_1))/|A| \\ y = (-2m(\alpha_3 - \alpha_1) + 2n(\alpha_2 - \alpha_1))/|A| \end{cases} \tag{3.4}$$

Then we define the deviation $\theta_l$ as

$$\theta_l = \begin{cases} \frac{5\pi}{2} - bear_l - \arccos(\frac{x-\alpha_l}{\sqrt{(x-\alpha_l)^2+(y-\beta_l)^2}}) & \text{if } y > \beta_l \\ \frac{\pi}{2} - bear_l + \arccos(\frac{x-\alpha_l}{\sqrt{(x-\alpha_l)^2+(y-\beta_l)^2}}) & \text{if } y \leq \beta_l \end{cases}$$

where $\theta_l$ is the deviation which contains the bearing information; $bear_l$ is the bearing information; $(x, y)$ is the MS location; $(\alpha_l, \beta_l)$ is the BS location.

So the iterative formulas become,

$$\begin{array}{rcl} x_{n+1} & = & 2[(\beta_3 - \beta_2)(d_1^2(n) - (\alpha_1^2 + \beta_1^2)) \\ & & +(\beta_1 - \beta_3)((d_2^2(n)) - (\alpha_2^2 + \beta_2^2)) \\ & & +\beta_2 - \beta_1)(d_3^2(n) - (\alpha_3^2 + \beta_3^2))]/|A| \\ y_{n+1} & = & 2[(\alpha_2 - \alpha_3)(d_1^2(n) - (\alpha_1^2 + \beta_1^2)) \\ & & +(\alpha_3 - \alpha_1)((d_2^2(n)) - (\alpha_2^2 + \beta_2^2)) \\ & & +(\alpha_1 - \alpha_2)(d_3^2(n) - (\alpha_3^2 + \beta_3^2))]/|A| \end{array}$$

Thus,

$$\begin{cases} x_{n+1} = f(x_n, y_n; s_1, s_2, s_3; l_1, l_2, l_3) \\ y_{n+1} = g(x_n, y_n; s_1, s_2, s_3; l_1, l_2, l_3) \end{cases} \quad (3.5)$$

Eq.(3.5) is called the iterative formulas. Given an initial value, we can derive a series of $(x_n, y_n)$ from the iterative formulas. If $(x_n, y_n)$ converges to one point $(\hat{x}, \hat{y})$, then $(\hat{x}, \hat{y})$ is considered to be the location estimation of the MS.

### 3.3 Convergence of the Iterative algorithm

We define two distance functions for our discussion: one is a distance based on EPM, the other is a common distance in a 2-D Euclidean space, denoted by,

$d(x, y, \alpha_l, \beta_l, p_{epm}, s_0, s) = \frac{k(s_0/s)^{1/\alpha}(1-e)}{1-e\cdot\cos(\theta(x,y,\alpha_l,\beta_l,bear_l))}$

$de(x, y, \alpha_l, \beta_l) = \sqrt{(x - \alpha_l)^2 + (y - \beta_l)^2}$

where $d(x, y, \alpha_l, \beta_l, p_{epm}, s_0, s)$ and $de(x, y, \alpha_l, \beta_l)$ are two distance functions, and $p_{epm} = (k, \alpha, e, \theta)$.

To simplify our discussion, we fix one variable when the other changes, for example, we fix $y$ for $x$, then we denote them as $d(x)$ and $de(x)$. We assume the estimated distance between the MS and the $i^{th}$ BS using EPM is $d_i(x, y)$. If these three circles intersect at one point, then $d(x, y) = de(x, y)$. Otherwise, we assume $d(x, y) \leq de(x, y)$.

**Theorem 3.1** *If $(\alpha_1, \beta_1)$, $(\alpha_2, \beta_2)$, $(\alpha_3, \beta_3)$ are not in the same line, and suppose that $|\frac{d_i^2(x,y)}{de_i^2(x,y)} \cdot \frac{(y-\beta_i)}{de_i(x,y)} \cdot \frac{e_i \sin(\theta_i)}{1-e_i \cos(\theta_i)}| \leq 1$, $|\alpha_i - \alpha_j| \geq 4$, if $|\alpha_i - \alpha_j| \neq 0$; $|\beta_i - \beta_j| \geq 4$, if $|\beta_i - \beta_j| \neq 0$; where $i \neq j$ and $i, j = 1, 2, 3$. Then $\{x_n\}$ and $\{y_n\}$ converge.*

**Proof:** For our convenience discussion. We assume that $\alpha_1 \leq \alpha_3 \leq \alpha_2$ and $\beta_2 \leq \beta_1 \leq \beta_3$. If $\alpha_1 = \alpha_3$, we add the condition, $\frac{\beta_1-\beta_2}{\beta_3-\beta_2} \leq \frac{1}{2}$.

Set

$$A = \begin{pmatrix} 2(\alpha_2 - \alpha_1), & 2(\beta_2 - \beta_1) \\ 2(\alpha_3 - \alpha_1), & 2(\beta_3 - \beta_1) \end{pmatrix}$$

$$b = \begin{pmatrix} (d_1^2 - (\alpha_1^2 + \beta_1^2)) - (d_2^2 - (\alpha_2^2 + \beta_2^2)) \\ (d_1^2 - (\alpha_1^2 + \beta_1^2)) - (d_3^2 - (\alpha_3^2 + \beta_3^2)) \end{pmatrix}$$

We rewrite the equation as the matrix formula, $AX = b$, where $X = (x, y)^T$.

Since $(\alpha_1, \beta_1)$,$(\alpha_2, \beta_2)$ ,$(\alpha_3, \beta_3)$ are not in the same straight line, so $det(A) \neq 0$. And the solution is $X = A^{-1}b$. Then we rewrite it as an iterative formula, $X_{n+1} = A^{-1}b(X_n)$.

Based on the definition of EPM, $k(o_i/s_i)^{1/\alpha}\frac{1-e_i}{1+e_i} \leq d_i(n) \leq k(o_i/s_i)^{1/\alpha}$, $i = 1, 2, 3$.

So both $\{x_n\}$ and $\{y_n\}$ have **bounds**.

We fix $y$ when $x_n$ changes, and fix $x$ when $y_n$ changes for our convenience discussion. In this paper, the convergence of $\{x_n\}$ will be discussed in detail while $\{y_n\}$ is a similar case.

Define $f(x) = (\beta_3-\beta_2)d_1^2(x)+(\beta_1-\beta_3)d_2^2(x)+(\beta_2-\beta_1)d_3^2(x)$ and $f(x)$ is a continuous function, then we obtain:

$(x_{n+1} - x_n)det(A)/2 = f(x_n) - f(x_{n-1})$.

By **the Lagrange Theory** [10], $f(x)$ has first derivative, we have: $f(x_n) - f(x_{n-1}) = f^{'}(x)(x_n - x_{n-1})$, where $x$ is between $x_n$ and $x_{n-1}$.

$$\begin{array}{rcl} \frac{f^{'}(x)}{2} & = & [\frac{(\beta_3-\beta_2)\cdot d_1^2(x)\cdot(y-\beta_1)\cdot e_1 \sin(\theta_1)}{de_1^3(x)\cdot(1-e_1 \cos(\theta_1))} \\ & & +\frac{(\beta_1-\beta_3)\cdot d_2^2(x)\cdot(y-\beta_2)\cdot e_2 \sin(\theta_2)}{de_2^3(x)\cdot(1-e_2 \cos(\theta_2))} \\ & & +\frac{(\beta_2-\beta_1)\cdot d_3^2(x)\cdot(y-\beta_3)\cdot e_3 \sin(\theta_3)}{de_3^3(x)\cdot(1-e_3 \cos(\theta_3))}] \end{array}$$

Set $K = 2f^{'}(x)/det(A)$, and

$det(A) = (\alpha_2-\alpha_1)(\beta_3-\beta_1)-(\alpha_3-\alpha_1)(\beta_2-\beta_1)$.

If $\alpha_3 - \alpha_1 \neq 0$ then $\alpha_3 - \alpha_1 \geq 4$.

By the theory condition,

$|K| < \frac{2}{(\alpha_2-\alpha_1)} + \frac{2}{\alpha_3-\alpha_1} \leq \frac{4}{\alpha_3-\alpha_1} \leq 1$

If $\alpha_3 - \alpha_1 = 0$,

$|K| < \frac{2}{(\alpha_2-\alpha_1)-(\alpha_2-\alpha_1)\frac{\beta_1-\beta_2}{\beta_3-\beta_2}} \leq \frac{4}{\alpha_2-\alpha_1} \leq 1$

That is , $|K| < 1$, and

$(x_{n+1} - x_n) = K(x_n - x_{n-1})$.

So $\{x_n\}$ converges. Similarly, $\{y_n\}$ converges too.

Q.E.D.

# 4 Simulation Results

With the technical support of two mobile operators in Hong Kong, we have conducted an intensive field test in many regions in Hong Kong in order to validate our model. We have divided the data into two parts: 30% of the data for training the models, and 70% of the data for estimating the location of the MS. We choose the distance between the exact location and the estimation location as the error criteria to describe the estimation accuracy.

With divided our experiment in two phases: the first phase is to train the model using 30% of the field test data, and the second phase is to estimate the location of the MS with the rest of the 70% of the field test data. With the effect of signal attenuation, our method sometimes provides an unreasonable solution. Hence, we use an anchor point to select the estimation of the location of the MS. If our estimation is not within a radius of 1000 meter with respect to the anchor point, we discard our solution and consider the our method fail to provide the estimation of the location of the MS. We choose the exact location of the MS as the anchor point in our model training phase. And we choose a weighted center of the locations of the BSs which we received signal from as the anchor point for estimating the location of the MS. So the missing ratio of our method includes two cases: One case is the number of BSs we received signal from is less than three, which does not meet our model assumption; the other case is the solution provided by our method is not within the predefined area in radius.

The data we collected from the field test are first used to provide the EPM parameters. These parameters are then put into a ***Lookup Table*** which will be used during the testing process. In the testing phase, we apply the Geometric Algorithm and the Iterative Algorithm to compute the field test data, then calculate their errors for the MS estimation. Lastly, we compared the results of the Iterative Algorithm with the results of CG, CT and Geometric Algorithm.

## 4.1 Estimating the EPM parameters

We choose 30% of the field test data to find out the parameters of EPM. For saving the computation cost, we divide all types of BSs into three groups for each region, which are denoted by Macro, Micro and others. Since the environment condition is similar within the same region, we assume the value of path loss exponent, $\alpha$, and the proportion value $k$ are the same throughout a region. Thus, there will be five parameters need to be trained in each region, $e_1$, $e_2$, $e_3$, $k$ and $\alpha$, where $e_1$, $e_2$ and $e_3$ are the EPM parameters for Macro, Micro and others respectively within one region. We set the step of the eccentricity, $e_1$, $e_2$ and $e_3$ as 0.1, and the step of path loss exponent, $\alpha$, is 0.1 , while the step of the proportion value $k$ is 0.05 for saving the computational cost. As $\alpha$ is expected to be within a range as suggested by [3], therefore, we choose $\alpha$ to vary within a range of 3 and 10 in order to meet the situation of Hong Kong and the proportion value is between 0.5 and 1.5. While the eccentricity of an ellipse has its natural limitation, it can vary within a range of 0 and 1 only. We provide the values of these parameters for each region. The results are shown in the ***Lookup Table***: **Table 1**.

## 4.2 Results of the Iterative Algorithm

After obtaining the ***Lookup Table***, Geometric Algorithm and Iterative Algorithm are used to estimate the location of the MS.

The Iterative Algorithm is used to estimate the location of the MS. Results are shown in **Table 2**. From **Table 2**, the estimation results using the Iterative Algorithm have good performance in many regions, such as Central, CheungShaWan, KwunTong, Mongkok, PE-MK, PrinceEdward, ShamShuiPo and SheungWan, where the averages of the estimating errors are below 125 meters. And in some regions, the 67% point values are below 125 meters, such as Mongkok and ShamShuiPo. In general, the improvement of the Iterative Algorithm over the Geometric Algorithm is inspiring.

There exist some regions where both the Geometric Algorithm and the Iterative Algorithm are not performing well, such as HungHum and LaiKing. We believe that these regions have special terrains and network layouts which may not fit the EPM directly. For example, in HungHum, we found that the MS always received signals from other regions. Moreover, some areas, like LaiKing, have very serious signal fading problem on the grounds that these regions are in front of the hills. In addition, we noticed that the RRS in these regions did not follow the rule of path loss well. Therefore, Some modifications may be required to readjust the EPM in order to handle these special regions.

## 4.3 Compare among the CG, CT and Geometric Algorithm

In this experiment, we compare our results with the results of the CG , CT and Geometric Algorithm. We present the mean, the standard deviation of the estimating errors and the success ratio of computing to describe the quality of the estimations. And the results are shown in **Table 3**.

The CG algorithm has the best success ratio for providing the estimation of the location of the MS, but its estimation has the worst performance within these algorithms. Since the CG estimation is just a weighted

| Region | $e_1$ | $e_2$ | $e_3$ | $k$ | $\alpha$ | Region | $e_1$ | $e_2$ | $e_3$ | $k$ | $\alpha$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Aberdeen | 0.4 | 0 | 0 | 1.5 | 7.3 | ShamShuiPo | 0.4 | 0.6 | 0 | 0.55 | 8.4 |
| CauseWayBay | 0.4 | 0.9 | 0 | 1.45 | 4.5 | ShaTin | 0.7 | 0.7 | 0 | 1.45 | 6.1 |
| Central | 0 | 0.2 | 0 | 1.45 | 7.2 | ShekKipMeiPark | 0.6 | 0 | 0 | 1.5 | 6.6 |
| CheungShaWan | 0.1 | 0 | 0 | 1.25 | 8.4 | SheungShui | 0 | 0 | 0 | 1.5 | 8.1 |
| FoTan | 0.9 | 0 | 0 | 1.5 | 6.4 | SheungWan | 0.4 | 0.9 | 0 | 1.45 | 8.4 |
| HappyValley | 0.6 | 0 | 0 | 1.5 | 8.1 | TaiKooShing | 0.9 | 0 | 0 | 0.55 | 4.9 |
| HungHom | 0 | 0 | 0.1 | 1.45 | 7.5 | TaiWai | 0.3 | 0.4 | 0 | 0.55 | 5.6 |
| KowloonBay | 0.4 | 0.2 | 0.9 | 0.55 | 6.4 | TaiWoHau | 0.2 | 0 | 0 | 0.65 | 5.3 |
| KowloonCity | 0.6 | 0.3 | 0.5 | 1.5 | 6.7 | TinShuiWai | 0.7 | 0 | 0 | 0.55 | 5.6 |
| KowloonTong | 0 | 0 | 0 | 1.5 | 7.8 | TsingYi | 0.1 | 0.1 | 0 | 1.5 | 7.7 |
| KwaiFong | 0.9 | 0 | 0 | 0.55 | 6.1 | TsuenWan | 0.9 | 0.5 | 0 | 0.55 | 8.4 |
| KwunTong | 0.9 | 0 | 0 | 1.4 | 7.5 | TszWanShan | 0.4 | 0 | 0 | 1.45 | 7.6 |
| LaiChiKok | 0 | 0.9 | 0 | 0.55 | 6.9 | TuenMun | 0.3 | 0 | 0 | 1.5 | 7.3 |
| LaiKing | 0.7 | 0.9 | 0.7 | 0.55 | 5.7 | WanChai | 0.2 | 0 | 0 | 1.5 | 7.4 |
| MaOnShan | 0.8 | 0 | 0 | 0.55 | 8.4 | WongTaiSin | 0.9 | 0 | 0 | 1.5 | 7.7 |
| Mongkok | 0.6 | 0 | 0 | 1.5 | 6.7 | YauTong | 0 | 0 | 0 | 1.5 | 7.1 |
| PE-MK | 0.6 | 0.9 | 0 | 1.5 | 7.8 | YauYatChuen | 0.3 | 0 | 0 | 1.5 | 7.5 |
| PrinceEdward | 0.9 | 0.9 | 0.3 | 0.55 | 8.4 | YuenLong | 0.9 | 0 | 0.9 | 1.5 | 5.9 |

**Table 1. The Lookup Table**

| Region | Ave. | Imp. | Std. | 67% | 90% | Region | Ave. | Imp. | Std. | 67% | 90% |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Aberdeen | 228.85 | 5.26% | 135.40 | 226.30 | 483.61 | ShamShuiPo | 103.36 | 6.79% | 55.17 | 119.61 | 193.02 |
| CauseWayBay | 258.62 | 28.42% | 189.56 | 323.19 | 619.47 | ShaTin | 351.58 | 5.72% | 172.12 | 401.61 | 577.97 |
| Central | 109.64 | 4.96% | 64.37 | 143.96 | 201.92 | ShekKipMeiPark | 314.36 | 3.80% | 170.07 | 379.47 | 570.76 |
| CheungShaWan | 120.47 | 12.45% | 66.34 | 158.94 | 211.89 | SheungShui | 507.99 | -9.37% | 333.80 | 798.05 | 1007.03 |
| FoTan | 280.15 | 8.16% | 134.60 | 356.50 | 460.99 | SheungWan | 116.10 | 7.25% | 58.98 | 144.56 | 190.77 |
| HappyValley | 343.89 | 9.86% | 222.31 | 430.83 | 714.78 | TaiKooShing | 250.84 | 36.26% | 153.91 | 295.97 | 440.13 |
| HungHom | 934.48 | -0.27% | 528.76 | 1414.11 | 1838.55 | TaiWai | 226.03 | 3.20% | 90.86 | 269.04 | 319.49 |
| KowloonBay | 199.82 | 14.3% | 108.01 | 227.17 | 343.94 | TaiWoHau | 277.40 | -6.97% | 144.96 | 356.22 | 508.62 |
| KowloonCity | 223.78 | 10.06% | 129.99 | 264.94 | 550.48 | TinShuiWai | 384.60 | -0.29% | 197.60 | 513.49 | 779.81 |
| KowloonTong | 255.83 | 13.45% | 153.84 | 209.41 | 499.45 | TsingYi | 554.74 | -13.17% | 262.30 | 674.04 | 931.13 |
| KwaiFong | 182.95 | 3.21% | 94.83 | 224.52 | 318.17 | TsuenWan | 143.42 | 8.78% | 60.28 | 138.80 | 236.72 |
| KwunTong | 115.05 | 8.78% | 58.84 | 126.32 | 202.43 | TszWanShan | 232.29 | 11.86% | 135.04 | 274.57 | 448.45 |
| LaiChiKok | 325.22 | 2.34% | 175.28 | 214.62 | 617.10 | TuenMun | 319.73 | 5.05% | 153.89 | 421.09 | 516.47 |
| LaiKing | 682.62 | 14.58% | 249.34 | 996.45 | 1181.29 | WanChai | 166.92 | 16.64% | 112.18 | 182.60 | 334.99 |
| MaOnShan | 353.11 | 7.27% | 116.73 | 410.16 | 500.64 | WongTaiSin | 306.57 | -20.35% | 147.16 | 377.40 | 515.72 |
| Mongkok | 88.36 | 3.49% | 50.04 | 111.08 | 150.08 | YauTong | 387.53 | 5.18% | 321.40 | 484.65 | 1030.33 |
| PE-MK | 110.08 | 4.68% | 62.25 | 134.99 | 202.64 | YauYatChuen | 227.77 | 1.8% | 113.69 | 278.87 | 389.61 |
| PrinceEdward | 118.29 | 14.51% | 69.91 | 138.67 | 214.91 | YuenLong | 267.59 | 18.29% | 285.83 | 229.26 | 976.63 |

**Table 2. Result with Iterative Algorithm and improvement over the Geometric Algorithm (Unit: meter)**

| Model | Average Error | Improvement % | Std. | sample number | success ratio % |
|---|---|---|---|---|---|
| CG | 495.76 | 27.08% | 787.05 | 116354 | 96.93% |
| CT | 470.99 | 23.24% | 986.06 | 116354 | 76.28% |
| Geometric Algorithm | 441.09 | 18.04% | 721.12 | 116354 | 84.08 % |
| Iterative Algorithm | 361.52 | 0% | 541.87 | 116354 | 79.13% |

**Table 3. Compare among the CG, CT, Geometric and Iterative Algorithm and improvement(Unit: meter)**

mean of the locations of the BSs which we received the signal from, its recommendation is always within the convex hull formed by these BSs locations, regardless whether the exact location of the MS is inside or outside the convex hull!

The CT algorithm has been proposed to solve the convex hull problem by trilateration. The CT estimation has better performance than the CG estimation does, but the CT algorithm discards some snapshots information for the cost. The CT algorithm has the least success ratio within these algorithms. Since the CT algorithm uses three base stations information to provide the location of the MS, it can not always provide a solution.

The Geometric Algorithm has derived from the CT algorithm. The Geometric Algorithm yet again improves the CT algorithm, since it can always provide a solution for the estimation of the location to the MS.

The results of the Geometric Algorithm has better performance than the CT and CG algorithms based on the average and the standard deviation of the estimation of errors. The success ratio of the Geometric Algorithm is better than that of the CT algorithm.

The Iterative Algorithm has improved the Geometric Algorithm. Since the Geometric Algorithm fixes a parameter of EPM to provide the location of the MS, it reduces the accuracy of estimation of the Geometric Algorithm. Since the Iterative Algorithm needs more extra conditions to guarantee to provide a convergence solution for the estimation, the success ratio of the Iterative Algorithm is some what lower than that of the Geometric Algorithm. But the Iterative Algorithm has the best performance in terms of the average and standard deviation of the estimating errors among these algorithms. Hence, the Iterative Algorithm is the best among these algorithms. However, the Iterative Algorithm needs more computational cost to provide a better estimation of the MS, and this is the trade between computation cost and accuracy. As a whole, the Iterative Algorithm provides an 18% ($\frac{441.09-361.52}{441.09} \times 100\%$) of improvement in terms of average error over the Geometric Algorithm on locating the MS within a radio cellular network.

## 5 Conclusions and Future work

In this paper, we present an iterative method to estimate the location of a MS under EPM. The Iterative Algorithm is an improvement over the Geometric Algorithm. From our simulation results, we have shown that the Iterative Algorithm has better accuracy than the Geometric Algorithm. Furthermore, we have proven that the Iterative Algorithm is superior to the existing algorithms. However, the Iterative Algorithm requires more computational cost for the implementation.

During this research, we found that signals do fluctuate at the same place. Signal attenuation can be affected by conditions, such as weather and car movement. The fluctuating signals will induce more errors in our estimation. As for our future work, we follow two threads to go forward our research. We will try to find out a filtering method to reduce the effect of signal fluctuation. Moreover, we will extend EPM to provide a more accuracy estimation. EPM is just a simple relationship between the RSS and the MS-BS distance. But it is a crude relationship for the RSS and the distance between MS and BS. And we use the same path loss exponent, $\alpha$, and proportion value $k$ in one region for saving computational cost, which may increase more error of the estimation. On the other hand, since EPM is a 2-D model only, we should extend it into a 3-D model to meet the real situation for locating the MS and to facilitate location based services. So to provide a filter method to reduce the effect of the signal fluctuation and to extend our EPM into a 3-D model for providing a 3-D location estimation are our future research work.

## References

[1] Peter H. Dana, *Global Positioning System Overview*, The University of Texas, http://www.colorado.Edu/geography/gcraft/notes/gps/gps.html.

[2] Richard Walter Klukas, Gerard Lachapelle, and Michel Fattouche, *Cellular Telephone Positioning Using GPS Time Synchronization*, The University of Calgary, http://www.geomatics.ucalgary.ca/Papers/Thesis/GL/97.20114.R

[3] Kaveh Pahlavan, Prashant Krishnamurthy, *Principles of Wireless Networks a Unified Approach*. Pearson Education, Inc., 2002.

[4] Svein Yngvar Willassen, Steinar Andresen, *A Method of implementing Mobile Station Location in GSM*, Norwegian University of Science and Technology, http://www.willassen.no/msl/bakgrunn.html.

[5] Joseph K. Ng, Stephan K. Chan, And Kenny K. Kan, "Location Estimation Algorithms for Providing Location Services within a Metropolitan area based on a Mobile Phone Network," in *Proceedings of The 5th International Workshop on Mobility Databases and Distributed Systems(MDDS 2002)*, Aix-en-Provence, France, September 2002, pp. 710–715.

[6] Joseph Kee-Yin Ng, Stephen Ka Chun Chan, and Shibin Song, "A Study on the Sensitivity of the Center of Gravity Algorithm for Location Estimation," Hong Kong Baptist University, Tech. Rep., May 2003, http://www.comp.hkbu.edu.hk/tech-report/tr03014f.pdf.

[7] Kenny K.H. Kan, Stephen K,C. Chan, and Joseph K. Ng, "A Dual-Channel Location Estimation System for providing Location Services based on the GPS and GSM Networks," in *Proceedings of The 17th International Conference on Advanced Information Networking and Applications(AINA 2003)*, Xi'an, China, March 2003, pp. 7–12.

[8] Junyang Zhou, Kenneth Man-Kin Chu, Joseph Kee-Yin Ng, "Providing Location Services within a Radio Cellular Network using Ellipse Propagation Model," in *Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA 2005)*, Taipei, Taiwan, March 28-30 2005, pp. 559–564.

[9] "Inverse square law," http://hyperphysics.phy-astr.gsu.edu/hbase/forces/isq.html.

[10] Chen Chuanzhang, Jin fulin, Zhu Xueyan, OYang Guangzhong, Ed., *Mathematics Analysis Lecture (The first Part) (The Second Edition)*. High Education Publishing Company, 1993, ch. 5, pp. 174–175.

# A Fragile Watermarking Scheme for 3D Mesh Verification

Hao-Tian Wu
Department of Computer Science
Hong Kong Baptist University

## Abstract

*In this paper, a fragile watermarking scheme is presented to detect the illegal tampering of 3D triangle mesh blindly. We have proposed a fragile watermarking algorithm for 3D mesh before, which embeds a sequence of data bits into the mesh for authentication purpose. The proposed watermarking algorithm can be generalized to a scheme that enables the tampering detection of 3D mesh by verifying its integrity. To address the verification of the mesh integrity, the problem of mesh traversal, as well as the vulnerability, robustness and reversibility of the embedded watermark is discussed. The optimal mesh traversal strategy is proposed to maximize the capacity of the mesh and thus most suitable for the tampering detection. The implementation of the presented scheme has demonstrated that unauthorized modifications on mesh models can be efficiently detected.*

## 1 Introduction

With the wide use of 3D models, the polygonal meshes in particular, it has become a real need to verify the integrity of 3D models, especially in the web environment. Traditional encryption algorithms can be used to restrict the access to the encrypted data, however, it cannot provide further protection if the encrypted data is decrypted. In other words, traditional encryption algorithms are independent from the content of the protected data. As an effective measurement, digital watermarking has been proposed for multimedia works (e.g. digital images, 3D models, audio and video streams) to give rise to some desired properties.

In general, digital watermarking can be classified into robust watermarking and fragile one. In fragile watermarking, the embedded watermark is vulnerable to a variety of operations imposed on the watermarked object so that it will change or disappear if the watermarked object is processed. By comparing the extracted watermark with the original one, fragile watermarking can be used for tampering detection and integrity verification of the watermarked object.

Only a few watermarking algorithms have been proposed to embed the fragile watermarks into 3D polygonal meshes for authentication purpose. In [2], the first fragile watermarking of 3D objects is addressed by Yeo and Yeung, as a 3D version of the approach proposed for 2D image watermarking. However, the distortion introduced by the encoding process of their proposed algorithm has not been numerically discussed in their paper. It seems that the watermarking strength in their proposed algorithm is not easy to adjust. The adjustable watermarking strength is preferred to satisfy different precisions of the 3D data, since relatively large error would make watermarking process meaningless. The mesh traversal in their method is according to the sequence of the vertices due to the causality problem.

In our proposed method, a new geometrical configuration is constructed to generate a host signal invariant to translation, rotation and uniformly scaling and embed the watermark by modulating the generated signal so that the embedded watermark is invariant to these transforms but sensitive to other geometrical or topological processing [5]-[6]. Our method is similar to one of the algorithms called Vertex Flood Algorithm proposed in [3], which can also be used for model authentication with certain tolerances, e.g. truncation of mantissas of vertex coordinates. Basically, their algorithm modifies the vertices so that their distances to the center of mass of a designated start triangle encode the watermark bits. In contrast, our method modulates the distances from the mesh faces to the mesh centroid using dither modulation technique [1]. Compared to the Vertex Flood Algorithm, the embedded watermark using our method is more sensitive to modifications on the watermarked model, while the integrity of the watermarked mesh can be verified.

It should be noted that a fragile watermarking scheme for 3D triangle meshes is presented by Cayre et al. in [4] to embed a watermark with robustness against translation, rotation and scaling transforms. Their main goal to present a new steganographic system designed for 3D triangle meshes and its performance in term of capacity, complexity, visibility and security is discussed. In case that the originality of the mesh model need to be verified, translation, rotation and scaling transforms should also be detected. To meet the requirement in this scenario, one extension of our proposed scheme can detect and distinguish those transforms from other processing with a reference position. Another extension of our proposed scheme is to make the encoding process reversible, i.e. the original mesh can be recovered from the watermarked mesh with some priori knowledge.

The rest of the paper is organized as follows. In Section 2, a new fragile watermarking scheme for 3D mesh is proposed in detail, including the encoding and decoding process, the problem of mesh traversal, as well as the vulnerability and robustness of the embedded watermark. Then, two extensions of the proposed scheme will be addressed respectively in Section 3. The implementation of the proposed scheme and the experimental results are given and discussed in Section 4. Finally, the conclusion is summarized and some future work is pointed out in Section 5.

## 2 A Fragile Watermarking Scheme for 3D Mesh

The watermarking process is performed on the polygonal mesh, which is considered as the “lowest common denominator” of surface representations. It is easy to convert other representations of 3D models to meshes. For convenience, we only deal with the manifold triangle mesh, in which every edge belongs exactly to two adjacent triangles. Let ($C$, $V$) presents the mesh geometry, where $C$ specifies the connectivity of the mesh (i.e. the adjacency of the vertices, edges, and faces), and $V = \{v_1, \cdots, v_m\}$ is the set of vertex positions defining the shape of the mesh in $R^3$.

The general procedure of our proposed scheme includes the encoding process and the decoding process, as shown in Fig. 1. In the encoding process, the watermark is embedded into the mesh model and the watermarked mesh is generated by restoring the position of the mesh centroid. Using some priori knowledge, the watermark can be extracted from the watermarked mesh and compared with the original one to obtain the verification result in the decoding process. The security of the watermarking scheme is guaranteed by the secret key $K$, which is used as the seed to scramble the face indices of the mesh in both encoding and decoding process.

### 2.1 The Encoding Process

In the watermark embedding, a special case of quantization index modulation (QIM) called dither modulation [1] is used to embed a sequence of data bits into the mesh model. We extend the dither modulation technique to the mesh model by constructing a geometrical configuration and modulating the host signal according to the bit value of the watermark, as detailed in the following. The mesh centroid restoration is performed to compensate the error introduced by the watermark embedding on the mesh centroid position so that the geometrical configuration can be recovered from the watermarked mesh to retrieve the embedded watermark in the decoding process.

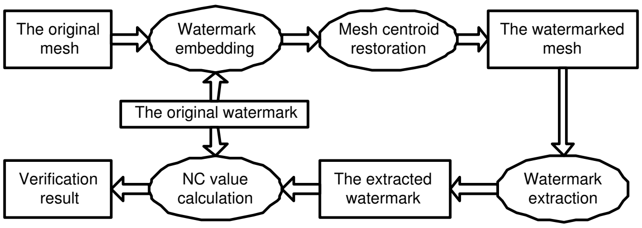


**Figure 1. The proposed procedure of verifying 3D mesh**

#### 2.1.1 The Watermark Embedding

To extend dither modulation to the mesh, two things need to be specified. One is the host signal of the mesh model that will be modulated to embed the watermark, and the other is the quantization step of the modulation. Since we aim to embed a fragile watermark which is sensitive to the modifications on the watermarked mesh, the host signal that is vulnerable to geometrical and topological modifications needs to be generated. Suppose $V = \{v_1, \cdot\cdot\cdot, v_m\}$ is the set of vertex positions in $R^3$, the position $v_c$ of the mesh centroid is defined as

$$v_c = \frac{1}{m}\sum_{i=1}^{m} v_i. \tag{1}$$

The distance from a given triangle to the mesh centroid is defined as the Euclidean distance from the triangle centroid to the mesh centroid. The ratios between the defined distances can be easily changed by those geometrical processing other than translation, rotation and uniformly scaling. The connectivity of vertices represented by the face indices will be modified by any topological processing meanwhile the ratios between the distances from the triangles to the mesh centroid will be scrambled. So the defined distance can be chosen as the host signal to embed the watermark. Furthermore, the centroid position $v_{ic}$ of a given triangle $f_i$ is obtained by

$$v_{ic} = \frac{1}{3}\sum_{j=1}^{3} v_{ij}, \tag{2}$$

where $v_{ij},\ j \in \{1, 2, 3\}$ is the vertex position in $f_i$. The distance $d_{fi}$ from $f_i$ to $v_c$ can be calculated by

$$d_{fi} = \sqrt{(v_{icx} - v_{cx})^2 + (v_{icy} - v_{cy})^2 + (v_{icz} - v_{cz})^2}, \tag{3}$$

where $\{v_{icx}, v_{icy}, v_{icz}\}$ and $\{v_{cx}, v_{cy}, v_{cz}\}$ are the coordinates of the face centroid and the mesh centroid in $R^3$, respectively.

The distance $d_i$ from a vertex with the position $v_i$ to the mesh centroid is given by

$$d_i = \sqrt{(v_{ix} - v_{cx})^2 + (v_{iy} - v_{cy})^2 + (v_{iz} - v_{cz})^2}, \quad (4)$$

where $\{v_{ix}, v_{iy}, v_{iz}\}$ is the vertex coordinate in $R^3$. The quantization step of the modulation $S$ is chosen as

$$S = D/N, \quad (5)$$

where $D$ is the distance from the first vertex in the starting triangle to the mesh centroid and the value of the parameter $N$ can be specified with respect to the precision of the mesh data.

With the specified $S$, the integer quotient $Q_i$ and the remainder $R_i$ are calculated with the distance $d_{fi}$ by

$$Q_i = \lfloor d_{fi}/S \rfloor, \quad (6)$$

$$R_i = d_{fi} \% S. \quad (7)$$

To embed one watermark bit $w(i)$, we modify the centroid position of $f_i$ so that the modulated integer quotient $Q'_i$ is an even value as $w(i) = 0$, or an odd value as $w(i) = 1$. The distance $d_{fi}$ is modulated by

$$d'_{fi} = \begin{cases} Q_i \times S + \frac{S}{2} & \text{if } Q_i\%2 = w(i) \\ Q_i \times S - \frac{S}{2} & \text{if } Q_i\%2 = \overline{w(i)} \ \& \ R_i < \frac{S}{2} \\ Q_i \times S + \frac{3S}{2} & \text{if } Q_i\%2 = \overline{w(i)} \ \& \ R_i \geq \frac{S}{2} \end{cases} + e_i, \quad (8)$$

where $\overline{w(i)} = 1 - w(i)$ and $d'_{fi}$ is the modulated distance. The component $e_i$ is added in the formula to make the embedded watermark statistically undetectable and $e_i$ is distributed within the interval $(-\frac{S}{2}, \frac{S}{2})$. It can be concluded from (8) that the modulated integer quotient

$$Q'_i = \lfloor \frac{d'_{fi}}{S} \rfloor = \begin{cases} Q_i & \text{if } Q_i\%2 = w(i) \\ Q_i - 1 & \text{if } Q_i\%2 = \overline{w(i)} \ \& \ R_i < \frac{S}{2} \\ Q_i + 1 & \text{if } Q_i\%2 = \overline{w(i)} \ \& \ R_i \geq \frac{S}{2} \end{cases} \quad (9)$$

so that $Q'_i\%2 = w(i)$. Consequently, the resulting $d'_{fi}$ is used to adjust the position of the triangle centroid. Only one vertex in the triangle is selected to move the face centroid to the desired position. Suppose $v_{is}$ of the position of the selected vertex in the triangle $f_i$, whose centroid position is $v_{ic}$, the adjusted vertex position would be

$$v'_{is} = [v_c + (v_{ic} - v_c) \times \frac{d'_{fi}}{d_{fi}}] \times 3 - \sum_{j=1, j \neq s}^{3} v_{ij}, \quad (10)$$

where $v_{ij}$ is the vertex position in the face $f_i$. To prevent the embedded bit value from being changed by the subsequent encoding operations, all vertex positions in $f_i$ should not be modified any more after the triangle centroid position is adjusted.

In the watermark embedding, a secret key $K$ is used as the seed of the random number generator to scramble the face indices $I$ to generate the scrambled indices $I'$. The embedding process is conducted following $I'$ as the starting triangle is the first triangle indexed by $I'$. To ensure that the modification made to each vertex position can be detected by modulating the distances from the triangles to the mesh centroid, the encoding process should traverse all the vertices in the mesh and the optimal mesh traversal which maximizes the capacity of the mesh needs to be found out and performed.

### 2.1.2 The Mesh Centroid Restoration

The watermark embedding inevitably introduces the distortion to the mesh geometry as some vertex coordinates are modified. The distortion of the mesh geometry also changes the position of the mesh centroid, although adjusting the vertex position may counteract each other. So in the encoding process, at least one vertex is needed to restore the position of mesh centroid so that the same configuration can be constructed to retrieve the embedded watermark in the decoding process. We refer to this operation as mesh centroid restoration, which modifies the position of the last vertex in the mesh traversal to compensate the error on the mesh centroid.

The restoration begins with the calculation of the introduced error by

$$E = \sum_{j=1}^{m} v'_j - \sum_{j=1}^{m} v_j, \quad (11)$$

where $m$ is the vertex number of the mesh model, $v_j$ and $v'_j$ are the vertex positions before and after the embedding process, respectively. The last vertex in the mesh traversal is moved by

$$v'_{last} = v_{last} - E, \quad (12)$$

where $v_{last}$ and $v'_{last}$ are the positions of the last vertex before and after the restoration. We further calculate the distance $D'$ from the last vertex after the restoration to the mesh centroid using (4) and obtain the value of the parameter $N'$ by

$$N' = D'/S. \quad (13)$$

The value of $N'$ will be used to calculate the modulation step in the decoding process. The encoding process ends as the position of the mesh centroid is restored.

## 2.2 The Decoding Process

In the decoding process, only the original watermark $W$, the value of parameter $N'$ and the secret key $K$ are required to verify the integrity of the watermarked mesh.

### 2.2.1 The Watermark Extraction

Similar to the encoding process, the centroid position $v'_c$ of the watermarked mesh model can be obtained by (1). With the secret key $K$, the face indices $I$ are scrambled to generate the scrambled indices $I'$, which is followed to trace the last vertex in the mesh traversal and its distance $D'$ to the mesh centroid is calculated. The modulation step $S'$ is calculated with the provided parameter $N'$ by

$$S' = D'/N'. \tag{14}$$

The embedded watermark bit is extracted from a triangle position in the following way. The centroid position $v'_{ic}$ of the triangle $f'_i$ is calculated by (2). Then, the modulated distance $d'_{fi}$ from $f'_i$ to $v'_{ic}$ is calculated by

$$d'_{fi} = \sqrt{(v'_{icx} - v'_{cx})^2 + (v'_{icy} - v'_{cy})^2 + (v'_{icz} - v'_{cz})^2}, \tag{15}$$

and the modulated integer quotient $Q'_i$ is obtained by

$$Q'_i = \lfloor d'_{fi}/S' \rfloor. \tag{16}$$

The watermark bit $w'(i)$ is extracted by

$$w'(i) = Q'_i \% 2. \tag{17}$$

The same mesh traversal as in the encoding process will be performed until all the watermark bits are extracted.

### 2.2.2 The NC Value Calculation

After the extraction process, the extracted watermark $W'$ is compared with the original watermark $W$ using the following cross-correlation function to measure the their correlation. Supposing their lengths are both identical to $K$, the normalized cross-correlation value $NC$ between the original and the extracted watermarks is calculated by

$$NC = \frac{1}{K} \sum_{i=1}^{K} I(w'(i), w(i)), \tag{18}$$

where

$$I(w'(i), w(i)) = \begin{cases} 1 & \text{if } w'(i) = w(i) \\ -1 & \text{otherwise} \end{cases}. \tag{19}$$

If the watermarked mesh geometry is intact, the $NC$ will be 1. Otherwise, if the watermarked mesh has gone through any geometrical or topological processing other than translation, rotation and uniformly scaling, the resulting $NC$ will be less than 1. We claim the mesh geometry as being tampered if the obtained $NC$ from (19) is less than 1.

## 2.3 The Mesh Traversal

To ensure that the modification made to each vertex position can be detected, those triangles whose centroid positions have been adjusted to embed the watermark bits should consist all the vertices in the mesh except that last one, which is used for the mesh centroid restoration in the encoding process and the calculation of the modulation step in the decoding process. So in the encoding process, all the vertices in the mesh should be traversed and all of them except the last one should be contained in the adjusted triangles.

The more triangles whose centroid positions hide the watermark bits a vertex belongs to, the more sensitive the embedded watermark is to the vertex position. We wish to embed as many as possible watermark bits into the mesh model so that the average number of the adjusted triangles each vertex belongs to is maximized. There exists the problem of finding out the optimal mesh traversal. The triangle position can be adjusted by moving only one vertex in the triangle, however, once one vertex in the triangle is selected to modulate the distance from the triangle to the mesh centroid, its adjusted position is determined by (8) and (10). So adjusting one vertex position at most embeds one watermark bit. Because there are two vertices whose positions are not modified in the starting triangle and the last visited vertex is used for mesh centroid restoration, the maximum number of the adjusted triangles in the watermark embedding is $m - 3$, supposing $m$ is the vertex number of the mesh. In the following, we shall propose the optimal mesh traversal strategy to achieve the maximum capacity.

As discussed above, the mesh traversal is intentionally performed following the scrambled face indices $I'$ to enhance the security of the watermarking scheme. At first, the starting triangle which is firstly indexed by $I'$ is picked and its position is adjusted by modifying one vertex position. After that, we always consider those triangles named as candidate triangles with two visited vertices and one unvisited vertex. A simple way to find those candidate triangles is to define a frontier [7], which is an imaginary polygon enclosing all the visited vertices without any unvisited ones. We define the visited edges as those edges contained in the triangles with three visited vertices. And the frontier is formed by those visited edges that are not enclosed by other visited edges.

At first, the frontier includes three edges of the starting triangle after its centroid position is adjusted. Based on the current frontier, those triangles consist of exactly one edge in the frontier and one unvisited vertex are found out and the one firstly indexed by $I'$ is chosen to embed the watermark bit. Each time a new triangle is traversed, the frontier needs to be updated as some new visited edges are added (the number may be more than two). At last, the watermark embedding stops until the last vertex is left unvisited.

Shall the mesh traversal visit all the vertices without blocking? Now we prove that the mesh traversal strategy can traverse all the vertices of a manifold triangle mesh. Since every edge in a manifold triangle mesh belongs exactly to two triangles, each edge in the frontier must belong to two triangles, one inside the frontier with three visited vertices as in the definition of the visited edge, and the other with an unvisited vertex outside the frontier (Otherwise, if the third vertex has also been visited, the edge will be enclosed by the other four visited edges in the two triangles, which contradicts with the definition of the frontier because all edges in the frontier are not enclosed by other visited edges). Therefore, new triangles with two visited vertices and one unvisited vertex can be found out based on the frontier until all the vertices are traversed.

The complexity of choosing one candidate triangle to embed one watermark bit at each time is as follows. Firstly, all the edges in the frontier are enumerated to find all the candidate triangles and among them the one firstly indexed by $I'$ is chosen. To update the frontier after each watermark bit is embedded, all the triangles that the latest traversed vertex belongs to need to be examined and if all three vertices have been visited, all three edges in the triangle will become or remain visited edges. Among all visited edge, those belong to two visited triangles are enclosed by other visited edges, and those belong to exactly one visited triangle and one unvisited triangle will form the new frontier. The candidate triangle whose position will be adjusted to embed the next watermark bit is the one which is firstly indexed by $I'$ among all the unvisited triangles that the edges in the frontier belong to.

### 2.4 The Properties of The Embedded Watermark

The watermark embedded by (8) and (10) is invariant to translation, rotation and uniformly scaling because the ratio between the distance from the triangle to the mesh centroid and the quantization step $S$, which is proportional to the distance from the first vertex in the starting triangle to the mesh centroid, remains the same after the model is translated, rotated or uniformly scaled. Otherwise, if the mesh model is processed by other operations that change the ratios, the formula $Q_i'\%2 = w(i)$ will not stand and the embedded watermark will be changed. Since we need to detect a trivial modification on the mesh model, the integer value $Q_i'$ should be sensitive to the distance from the triangle to the mesh centroid. It can be achieved by assigning $N$ a proper value to obtain a small value of $S$ with respect to the precision of 3D data.

In [2], the mesh topology is denoted by the set of adjacent vertices whose indices are less than that of a given vertex plus the vertex itself. In contrast, our method uses the face indices to represent the connectivity of the vertices. Furthermore, the key $K$ is used as the seed of pseudo-random number generator to generate the scrambled indices $I'$ so that the security of the watermarking scheme is guaranteed. Since the mesh traversal is dependent on $I'$, the extracted watermark $W'$ will be dramatically different from the original watermark $W$ without the correct $I'$. If there is any change made to the mesh topology, such as mesh decimation, resampling or refinement, the face indices $I$ will be modified so that the scrambled indices $I'$ will not be correctly generated. In other words, any modification on the mesh topology will lead $I'$ as well as $W'$ to change. Therefore the unauthorized modification on mesh topology can be detected.

## 3 Two Extensions of The Proposed Scheme

The proposed scheme can be extended to various algorithms by assigning different meaningful values to $e_i$ in (8) so that the desired properties can be achieved. To make the embedded watermark statistically undetectable, $e_i$ should be uniformly distributed within $(-\frac{S}{2}, \frac{S}{2})$. In practise, to reduce the false alarm probability, the distribution range of $e_i$ can be a little smaller, $(-\frac{3S}{8}, \frac{3S}{8})$ for example.

### 3.1 Detecting Translation, Rotation and Uniformly Scaling

In our proposed scheme, the embedded watermark is invariant to translation, rotation and uniformly scaling, but sensitive to other processing. By using a reference position $p_r$, the extended algorithm can detect those transforms.

In the encoding process, besides one unvisited vertex in the chosen triangle $f_i$ is adjusted to modulate the distance from $f_i$ to the mesh centroid, another vertex in $f_i$ is chosen as the reference vertex. The distance from the reference vertex to the reference position $p_r$ is calculated by

$$|v_{ir}-p_r| = \sqrt{(v_{irx}-p_{rx})^2 + (v_{iry}-p_{ry})^2 + (v_{irz}-p_{rz})^2}, \tag{20}$$

where $v_{ir}$ is the reference vertex position. Then $e_i$ in (8) is assigned by

$$e_i = \frac{3(|v_{ir}-p_r|\%S - S/2)}{4}. \tag{21}$$

Suppose $|v_{ir}-p_r|\%S$ is uniformly distributed within $[0, S)$, $e_i$ is within the interval $[-\frac{3S}{8}, \frac{3S}{8})$. Therefore, the adding of $e_i$ in (8) will not interfere the embedded watermark and it can be derived that $e_i = d'_{fi} - Q_i' \times S - S/2$.

In the decoding process, the reference position $p_r$ need to be provided. Firstly, the modulated distance $d'_{fi}$ is calculated using (15). Then the integer quotient $Q_i'$ can be obtained by (16) and the bit value $w'(i)$ is retrieved by (17).

The distance from the reference vertex in $f_i$ to the reference position $p_r$ is calculated by (20) and $e_i$ is obtained by (21). After that, the difference between $e_i$ and $d'_{fi}-Q'_i\times S-S/2$ is calculated. For each triangle that carries the watermark bit, if $|d'_{fi}-Q'_i\times S-S/2-e_i|<\epsilon$ and the extracted bit $w'(i)$ equals to the original watermark bit $w(i)$, the mesh model is considered as not having been modified.

If the watermarked mesh model is processed by translation, rotation and uniformly scaling operations, the vertex positions will be modified and the distance from the reference position to the reference vertex in $f_i$ will change, as well as the ratio between $e_i$ and the quantization step $S$. However, the ratio between the distance $d'_{fi}$ and $S$, as well as the ratio between $d'_{fi}-Q'_i\times S-S/2$ and $S$, remains the same. If a relatively small value, $S/8$ for example, is assigned to $\epsilon$, then $|d'_{fi}-Q'_i\times S-S/2-e_i|>\epsilon$. Translation, rotation and uniformly scaling operations will be identified if $e_i$ does not match $d'_{fi}-Q'_i\times S-S/2$ while the extracted watermark is the same as the original one. Although the reference position $p_r$ can be arbitrarily chosen, it should be chosen nearby the mesh centroid to make the ratio between $e_i$ and $S$ more sensitive to those transforms.

In case that the watermarked mesh model is processed by other geometrical modification, there exists the triangle $f_i$ from which the distance $d'_{fi}$ to the mesh centroid is altered. So the ratio between $d'_{fi}-Q'_i\times S-S/2$ and $S$ is changed. However, the ratio between $e_i$ and $S$ is not certain to vary, depending on whether the reference vertex in $f_i$ is moved. Even if the ratio between $e_i$ and $S$ is also changed, $|d'_{fi}-Q'_i\times S-S/2-e_i|$ will probably exceed $\epsilon$. Therefore, besides the embedded watermark, those processing other than translation, rotation and uniformly scaling can be detected by comparing $e_i$ with $d'_{fi}-Q'_i\times S-S/2$.

Suppose there are $K$ faces used to hide the watermark information, we need to compare $e_i$ with $d'_{fi}-Q'_i\times S-S/2$ for $K$ times to verity the originality of the watermarked mesh model. To estimate the strength of tampering, the normalized matching number $NM$ between $e_i$ and $d'_{fi}-Q'_i\times S-S/2$ is obtained by

$$NM=\frac{1}{K}\sum_{i=1}^{K}m(i) \tag{22}$$

where

$$m(i)=\left\{\begin{array}{ll}1 & \text{if } |d'_{fi}-Q'_i\times S-S/2-e_i|<\epsilon\\ 0 & \text{otherwise}\end{array}\right. . \tag{23}$$

The mesh model will be considered as being tampered either $NC$ in (19) or $NM$ in (22) is less than 1.

## 3.2 The Reversible Algorithm of The Proposed Scheme

The proposed scheme can be used to detect the illegal tampering of the watermarked mesh, however, the original mesh is slighted changed once it is watermarked since nearly all vertex positions are modified. In case that the original mesh needs to be recovered, the distortion introduced by the encoding process requires to be compensated. Since the mesh topology is not changed during the encoding process, each vertex needs to be moved back to its original position. By keeping the modulation information in the watermarked mesh, the reverse process of the encoding process can be performed to recover the original mesh.

### 3.2.1 Reversibly Embedding

Suppose there are $K$ faces used to embed the watermark, to keep the modulation information in the watermarked mesh, $e_i$ is defined as follows: for $i=3,\cdots,K$, $e_i=\frac{d'_{f(i-1)}-d_{f(i-1)}}{4}$, while $e_1=\frac{d'_{fK}-d_{fK}}{4}$ and $e_2=\frac{Q'_1\times S+S/2-d_{f1}}{4}$ with $d_{fi}$, $d'_{fi}$ and $Q'_1$ provided in (3), (8) and (9), respectively.

The detailed procedure to reversibly embed the watermark is as follows:

Step 1: Calculate $d_{f1}$ and $Q'_1$ by (3) and (9), respectively, then obtain $e_2=\frac{Q'_1\times S+S/2-d_{f1}}{4}$ and $e_2\in(-\frac{S}{4},\frac{S}{4})$;

Step 2: Keep $e_2$ in $d'_{f2}$ using (8) and obtain $e_3=\frac{d'_{f2}-d_{f2}}{4}$ and $e_3\in(-\frac{5S}{16},\frac{5S}{16})$;

Step 3: Keep $e_3$ in $d'_{f3}$ using (8), then obtain $e_4=\frac{d'_{f3}-d_{f3}}{4}$ and $e_4\in(-\frac{21S}{64},\frac{21S}{64})$, and so on until $e_K$ is kept in $d'_{fK}$ and $e_1=\frac{d'_{fK}-d_{fK}}{4}$ is obtained;

Step 4: Keep $e_1$ in $d'_{f1}$ using (8).

It can be concluded that $e_i$ is distributed within $(-\frac{2S}{5},\frac{2S}{5})$ from its definition together with (8), which implies that the adding of $e_i$ does not interfere the embedded watermark.

The mesh centroid restoration is performed as the same as in Section 2.

### 3.2.2 Recovering the Original Mesh

To recover the original mesh, the modulation information, $e_i$, needs to be retrieved from the modulated distance $d'_{fi}$ obtained from (15) and the modulated integer quotient $Q'_i$ obtained from (16). For $i=1,2,\cdots,K$,

$$e_i=d'_{fi}-(Q'_i\times S'+S'/2). \tag{24}$$

According to the definition of $e_i$, for $i=2,\cdots,K-1$, the original distance $d_{fi}$ is obtained by

$$d_{fi}=d'_{fi}-e_{i+1}\times 4, \tag{25}$$

while

$$d_{fK} = d'_{fK} - e_1 \times 4, \tag{26}$$

and

$$d_{f1} = Q'_1 \times S' + S'/2 - e_2 \times 4. \tag{27}$$

With $d_{fi}$ provided by (24), (25) and (26), all the vertices whose positions have been adjusted in the watermark embedding can be shifted back by

$$v_{is} = (v'_c + (v'_{ic} - v'_c) \times \frac{d_{fi}}{d'_{fi}}) \times u - \sum_{j=1, j \neq s}^{u} v'_{ij}, \tag{28}$$

where $v'_{ij}$ is the vertex position in the triangle $f'_i$ with $v'_{ic}$ as the adjusted position of the face centroid, $v_{is}$ is the restored vertex position in $f'_i$ and $v'_c$ is the mesh centroid position.

After we shift back the vertices used in the watermark embedding, the next step is to move the last vertex used in the mesh centroid restoration to its former position. Firstly, the vector $E'$ of the above shifting operations, which should be the opposite of the error vector $E$ introduced by the watermark embedding, is calculated by

$$E' = \sum_{j=1}^{m} v_j - \sum_{j=1}^{m} v'_j, \tag{29}$$

where $v'_j$ and $v_j$ are the vertex positions before and after the reverse of the watermark embedding process, respectively. Subsequently, the vertex used in mesh centroid restoration can be moved back by

$$v_{last} = v'_{last} - E', \tag{30}$$

where $v_{last}$ is the recovered position of $v'_{last}$. By this means, the original mesh is recovered from the watermarked mesh.

# 4 Implementation and Discussion

## 4.1 Practical Implementation

Our proposed scheme is conducted in spatial domain and applicable to the manifold triangle mesh to verify its integrity without any special restriction. It can be extended to other polygonal meshes with the corresponding mesh traversal strategy. In the implementation, the modulation step $S$ should be carefully selected with respect to the precision of 3D data, providing a trade-off between imperceptibility of the embedded watermark and false alarm probability. We have investigated two proposed algorithms on several mesh models listed in Table 1. In case that the originality of the mesh model needs to be verified, the first algorithm is preferred since it can detect translation, rotation and uniformly scaling; while the original mesh can be recovered using the reversible algorithm. A 2D binary image is chosen as the watermark, which can also be a hashed value.

**Table 1.** THE MESH MODELS USED IN THE EXPERIMENTS

| Models | Vertices | Faces | Capacity(bits) |
|---|---|---|---|
| dog | 7616 | 13176 | 7519 |
| wolf | 8176 | 13992 | 7995 |
| horse | 10316 | 18359 | 10253 |
| raptor | 10853 | 14547 | 10218 |
| cat | 11074 | 19093 | 10859 |
| lion | 20315 | 32094 | 19688 |

## 4.2 Capacity

The capacity of the manifold triangle mesh to carry the watermark is the same using the two algorithms, since the difference only takes place in the extension of $e_i$ in (8). The maximum number of the embedded bits that can be carried by each manifold triangle mesh is the vertex number minus 3. The bit number listed in Table 1 of each model is below the maximum number is due to each model is a combination of many manifold triangle meshes. To verify the integrity of the mesh, the optimal mesh traversal maximizing the capacity of the mesh is best suitable since it also maximizes the average number of the adjusted triangles that each vertex belongs to so that the embedded watermark is sensitive to the tampering of every vertex position.

## 4.3 Imperceptibility

To evaluate the imperceptibility of the embedded watermark, the Hausdorff Distance between the original mesh model and the watermarked mesh model normalized by the largest dimension of the mesh, which is defined as the longest distance from the vertex to the mesh centroid, as in [4], upon the fact that the mesh topology is unchanged. The normalized Hausdorff Distance between the original and recovered mesh models was also calculated to test the reversible algorithm. Fig. 2 shows the amount of the distortion subject to the modulation step $S$ using the first algorithm (the upper curve) and the reversible algorithm (the below curve), respectively, given that only one vertex was used in the mesh centroid restoration. From the experimental results, it can be seen that the normalized Hausdorff Distance between the original mesh and the watermarked mesh increases as $S$ increases. The reversibility mechanism significantly reduces the introduced distortion, since the difference between the original and recovered meshes is much smaller than that between the original and water-

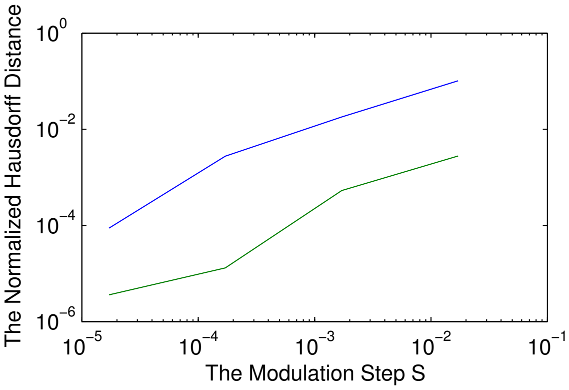


**Figure 2. The normalized Hausdorff distance subject to the step S**

marked meshes given the same $S$.

### 4.4 Tampering Detection

Using the first algorithm, the watermarked mesh models went through translation, rotation and uniformly scaling transforms, changing the positions of two vertices oppositely (respectively by adding the vectors $\{2S, 2S, 2S\}$ and $\{-2S, -2S, -2S\}$), modifying one vertex position by adding the vector $\{\frac{mD}{2N}, \frac{mD}{2N}, \frac{mD}{2N}\}$, reducing one face from the mesh, and adding Gaussian noise signal $\{n_x, n_y, n_z\}$ with zero mean and standard deviation $S$ to all the vertex positions, respectively. The watermarks are extracted from the processed mesh models with and without the key $K$ and the normalized cross-correlation value $NC$ between the extracted and the original watermarks are calculated using (19). The normalized matching number $NM$ between $e_i$ and $d'_{fi}\%S' - S'/2$ is also calculated by (22). The obtained results of $NC$ and $NM$ shew that the embedded watermark is invariant to translation, rotation and uniformly scaling but sensitive to other processing, while the normalized matching number is sensitive to all those processing.

## 5 Concluding Remarks and Future Work

In this paper, we have proposed a new fragile watermarking scheme for 3D manifold triangle mesh, which can embed the position-rotation-size invariant watermark. The watermarking strength is adjustable by properly choosing the modulation step with respect to the precision of 3D data, providing a good trade-off between the imperceptibility of the watermark and false alarm probability. With some priori knowledge, the integrity of the mesh can be verified by extracting and comparing the embedded watermark with the original one. To make the embedded watermark sensitive to the modification made to each vertex position, the optimal mesh traversal strategy is proposed to maximize the capacity of the mesh as well as the average number of the adjusted triangles that each vertex belongs to.

Depending on the end applications, some desired properties can be achieved by assigning meaning values to the plastic component in the proposed scheme. In this paper, two detailed algorithms are proposed, one enables the detection of translation, rotation and uniformly scaling and the other recovers the original mesh from its watermarked version. However, the constructed work is far from perfect. Future work is needed to make it useful in practice for 3D mesh verification. A general model of false alarm probability needs to be constructed to analyze the computational error due to the limited precision and the distribution of plastic component.

## References

[1] B. Chen and G.W.Wornell, "Digital watermarking and information embedding using dither modulation," *IEEE Second Workshop on Multimedia Signal Processing*, pp. 273-278, 1998.

[2] M. M. Yeung and B.-L. Yeo, "Fragile watermarking of three dimensional objects," *Proc. 1998 Int'l Conf. Image Processing, ICIP98*, volume 2, pp. 442-446. IEEE Computer Society, 1998.

[3] O. Benedens and C. Busch, "Toward blind detection of robust watermarks in polygonal models," *Proc. EUROGRAPHICS Comput. Graph. Forum*, vol. 19, pp. C199-C208, 2000.

[4] F. Cayre and B. Macq, "Data hiding on 3-D triangle meshes," *IEEE Trans. Signal. Processing*, vol. 51, pp. 939-949 (4), 2003.

[5] Hao-Tian Wu and Yiu-Ming Cheung, "A New Fragile Mesh Watermarking Algorithm for Authentication," *IFIP 20th International Information Security Conference*, Chiba, Japan, pp.509-523, 2005.

[6] Hao-Tian Wu and Yiu-Ming Cheung, "A Fragile Watermarking Scheme for 3D Meshes," *to be appeared in ACM Multimedia and Security Workshop(ACM2005)*, New York, USA, August, 2005.

[7] Tulika Mitra and Tzi-cker Chiueh, "A Breadth-First Approach to Efficient Mesh Traversal," *13th ACM SIGGRAPH/Eurographics Graphics Hardware Workshop*, August 1998.

[8] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald and W. Stuetzle, "Mesh optimization," *Computer Graphics (SIGGRAPH '93 Proceedings)*, pp.19-26, 1993.

# Learning Regularized Optimal Ensembles of Classifiers

Zhili Wu, Chunhung Li
Department of Computer Science
Hong Kong Baptist University
Kowloon Tong, Hong Kong
Email:{vincent,chli}@comp.hkbu.edu.hk

## Abstract

*Combining classifiers has shown proven performance benefits in many reported work. However, methods for selecting and combining multiple classifiers' output are often heuristic in nature and does not have a well-defined objective function. We propose the use of a linear prediction approach on the multiple classifiers' output and optimize the classification task with a hinge loss regularized objective function. By observing the relationships of the regularized functions with the $l$-2 SVM, the regularized objective function is then solved via standard SVM approaches. We demonstrate how the linear prediction approach can be adopted to give different optimized combination strategies to guide the selection and combination of base classifiers. Results on standard datasets shown improved performances in all combination strategies.*

## 1 Introduction

Ensemble is a useful technique where the outputs of a set of base learners are combined to form a unified prediction [19], e.g. neural network ensemble [10]. For classification tasks dealt with in this paper, typical ensemble learning is to construct a collection of individual classifiers, then obtain the class prediction by voting the outputs of the individual classifiers in the ensemble. Many researchers have demonstrated that ensembles generally outperform the best single classifier in the ensemble. Typical applications of ensembles includes face recognition [9], hand written word recognition [8], medical diagnosis [28],etc.

Bagging ("bootstrap aggregation") [2] and, boosting (e.g. Adaboost [6, 7]) are two major techniques for constructing ensembles [16, 4]. Both techniques are thought to generate different classifiers by training on different training sets [4]. In bagging, each training set is constructed by drawing a certain number of examples uniformly (with replacement) from the original training set. In boosting, the typical Adaboost algorithm starts from a set of weights over the original training set, and constructs new training sets through ıboosting by sampling or ıboosting by weighting. In ıboosting by sampling, it constructs a new training set by drawing examples (with replacement) from the original training set with probability proportional to their weights. In ıboosting by weighting, it inputs the entire original training set and the weights into base algorithms which can accept a weighted training set directly. The weights in Adaboost are then adjusted after training a base classifier, by increasing the weights of misclassified examples, and decreasing the ones of correctly classified examples.

It has been argued that diversity and accuracy of base classifiers play an important role in obtaining a good ensemble [15]. For bagging, it relies on unstable base classifiers sensitive to sampled training sets to achieve diverse classifiers. For boosting, it doesn't require the unstable condition for base classifiers because it can realize the diversity of base classifiers through re-weighting training sets. Some other approaches many aiming at further diversifying base classifiers are available, by inducing randomness into the learning algorithm [4], or manipulating the input attributes [11], and the model outputs [3].

Ensemble learning typically adopts weighted or unweighted voting for prediction combination [1, 22]. In the unweighted majority(Plurality) voting, the class with the most votes from all base classifiers is regarded as the prediction by the ensemble. In weighted voting, base classifiers may have different weights associated, their outputs are then weighted and linearly combined. Adaboost comes up with the set of weights for combination at the end of training all base classifiers, but other ensemble techniques like bagging do not produce such set of weights.

Stacked generalization [26] is another scheme for combining outputs of base classifiers. It learns an upper-level meta-classifier based on the predictions of all base classifiers. A recent study [18] shows stacking enabled by multi-response model trees performs better than selecting the best single classifier in the ensemble. The selective ensemble [27] has some common points with the stacking approaches, it also reported better performance than typical bagging and boosting approaches with majority voting. The stacked generalization is even drawn relationship with meta-learning, which is about ”learning to learn” [24]. This ensemble schema also allows easy combination of heterogenous base learners, which usually means base learners are trained by several different learning algorithms, rather than a single learning algorithm typically used in bagging and boosting.

Other than the issue of (learning) how to combine base learners, choosing which set of base learners for the combination is also important. Each base learner in an ensemble usually is not tuned to be the optimal single learner. On the other hand, some of base learners might be redundant, due to the large overlapping of training sets, or the stable performance of the base learning algorithms, etc. It is thus helpful to consider the selective combination of base learners. Along this direction of thought, some approaches can be found, which deal with dynamically adding base learners [13], learning the optimal combination of neural networks [23] or doing selective ensemble [27].

This paper focuses on the issue of how to select and combine base learners. It will present the regularized objective form for base learner combination, together with possible variants. And then the practical steps are briefly stated, including the discussion of selecting base learners. After that it is the experimental section, which provides the comparisons between our approaches and the genetic algorithm-based ensemble (GASEN), and include a brief introduction to two projects which our ensemble approaches to be applied to. At the end, there will be discussion and open problem discussion.

## 2 Formulation and Solution

### 2.1 The Regularized Combination of Base Learners

Given a set of $n$ examples $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$ to be classified into the positive class or negative one, that is, a target output vector $\mathbf{Y} = \{y_i\}_{i=1}^n$, an ensemble of $N$ base learners can give $N$ predictions to each example. This actually results in a prediction matrix $\mathbf{M}$ with the size of $n \times N$.

From the perspective of voting for or linearly combining the outputs of base classifiers, liking bagging or boosting, the following form is implied or approximated,

$$f : \mathbf{M}\beta \rightarrow \mathbf{Y},$$

where $\beta \in \mathbf{R}^N$ is a weighting vector. Majority vote alike combination is to let all $\beta_i = \frac{1}{N}$ and then take the *sign* function, while in Adaboost the weights $\beta$ to base learners are obtained through the updating of the weights to training examples, that is, the weights $\beta$ are indirectly learned from the process of updating the weights to training examples. Note this two approaches do not have a clear objective function to quantify the goodness of the $\beta$, a way thus is to formulate the learning of the $\beta$ into optimizing an objective function, and solve the objective function in a systematic manner.

Assuming the $\mathbf{Y}$ is known for the purpose of learning the $\beta$. It is clear the $l_2$ SVM can be used here as the objective function, that is, it is to learn a SVM model to comply with the dataset $\mathbf{M}$ and their labels $\mathbf{Y}$. Assuming $f_i = \mathbf{M}_{i.}\beta + \beta_0$ where $\mathbf{M}_{i.}$ as a row vector is the $i-$th row of the prediction matrix $\mathbf{M}$. The $\beta_0$ is an offset constant, which can also be forced to zero [12] in standard SVM training. The objective function adopted by $l_2-$SVM is

$$\min C \sum_{i=1}^{n} \max\{1 - y_i f_i, 0\} + \frac{1}{2}||\beta||^2.$$

This objective form can be efficiently solved by QP routines, or simply calling standard SVM packages.

The term $\max\{1 - y_i f_i, 0\}$ is also called hinge loss. From this perspective of defining such a loss term, the above objective function has the following variant [ref],

$$\min\{\sum_{i=1}^{n} \max\{1 - y_i f_i, 0\} + \frac{\lambda}{2}||\beta||^2\},$$

where $\lambda = 1/C$. This form can be regarded as minimizing both the hinge loss term and the penalty as measured by $\frac{\lambda}{2}||\beta||^2$. The rewritten form does not change the nature of the objective, but has been regarded as one member of the family of regularization [5]. It actually hints many variants of the objective functions can be used to deal with the problem of combining base classifiers.

To change the penalty term $||\beta||^2$ to $||\beta||^0$ or $||\beta||^1$, zero or one norm SVMs [25, 29] are formulated. They have the property of forcing some entries of $\beta$ to be zero, which implies that by solving them, the way of combining learners and the selection of learners can be simultaneously fulfilled with. This paper only presents the approaches of combining learners through the way of two norm SVMs, but keeps the study of selecting learners by zero or two norm svms ongoing.

There are some further adapted objective formulations. By taking the square loss $||\mathbf{Y} - \mathbf{M}\beta||^2$, and forcing a small bound to the $l_1$ of $\beta$, the Least Absolute Shrinkage and Selection Operator (LASSO) [21] is used, which also leads to some zeros entries for the resulting $\beta$.

$$\min_{\beta} ||\mathbf{Y} - \mathbf{M}\beta||^2, s.t.||\beta||^1 < s$$

where $s$ is a small constant. Furthermore, by using the penalization term $||\beta||^2$, the Least Square SVM [20] is obtained.

### 2.2 The Implementation

The current implementation of base classifier combination is very simple, mainly benefited from the standard SVM implementation. The overall steps can be summarized as follows:

- Train multiple base classifiers (e.g. Neural networks, SVMs or others, even their mixture) based on different data split, feature subsets and randomness injection.

- For a separate set with labels known, take all the predictions from all base classifiers, and train a $l_2$ linear SVM upon the prediction matrix and the known labels (Possibly tuning the model parameter like $C$ to ensure a good generalization ability. The weights to each base learners can be calculated from the built SVM model.

- For small $\beta_i$ less than a preset threshold $\theta$, their associated base classifiers are deleted from the ensemble, the weighting set $\beta$ is accordingly reduced and renormalized.

## 3 Experiments

To test the performance of our approach, a set of experiments on real data are conducted. This section gives an introduction to datasets, experimental setup and results, together with a brief and high-level introduction to two ongoing projects which has been adopting this approach, one is siRNA efficacy prediction, the other is the web query categorization task in KDD-CUP 2005.

### 3.1 Datasets

The first dataset is the numerical version of the Credit (German) data from the UCI machine learning repository [ref]. It is a unbalance binary classification task. It has 700 positive data points and 300 negative ones, each has 24 numerical features.

The second dataset is the chess data, also from the UCI machine learning repository. It has 36 nominal features, 1669 positive examples and 1527 negative ones. It becomes a numerical data matrix with the dimension of 38 after conversion.

The third dataset is the waveform data. It has 4000 examples and three classes. In our experiments we only use two classes of data. The extension of SVMs to multiclass feature/learner selection remains to be an unsolved issue. But the ECOC [14] would be helpful in dealing with the combination of base learners under the regularized framework.

### 3.2 Experimental Setup and Results

We mainly follow the experimental settings for the GASEN in the selective ensemble paper [27]. The comparison is also performed to be with the GASEN approach only. Note the GASEN has been shown outperforming other ensemble approaches including bagging and boosting, hereby the comparison results between our approaches and the popular voting bagging and boosting approaches are not listed.

For each dataset, we use half of the randomly drawn examples to form the original training data set, on which ten ensembles of neural networks are trained by bootstrap sampling. The remaining half of examples are then evenly divided into a separate validation set and a testing set. Taking the Credit(German) data of 1000 points as an example, 500 points with respect to the ratio of positive and negative examples are randomly drawn out to form the original training set. Based on the 500 points, ten neural networks are trained. Each neural network takes a subset (roughly 63%) points bootstrapped from the 500 points for training, validated by the remaining set of points (about 37%) [ref out-of-bag estimation].

For the remaining 500 points, 250 are taken as a validation set and 250 as a testing set. After training the ten base neural networks, their prediction over the validation set are then used to learn the selection and combination strategy. The $l2-$SVM is selected for learning how to select and combine the ten base neural networks. As a comparison, the genetic algorithm for learning the weighted combination of base neural networks (GASEN) is compared, as well as some variants of these two approaches. Finally, the base neural networks are combined to give a unified prediction to the 250 testing points, and the classification error is reported. The whole process is repeated ten times to get an average error reporting.

Several variants of GASEN and our approach are implemented. They are GASEN-w, GASEN-wa, SVM-w, SVM-wa. GASEN-w uses the evolved weights by genetic algorithms to select the base neural networks, and combines the predictions of the selected networks with the normalized version of their evolved weights. SVM-w similarly selects base neural networks by the weights learned during forming the SVM model, and combines the predictions of selected networks with the normalized SVM weights associated with these selected networks. GASEN-wa and SVM-wa do not select the base neural networks, just do a weighted combination of the outputs of all base networks.

From the Table 1, it can be shown when selecting base learners and combining them through majority voting, the SVM-guided learner selection is better than GASEN, with the equal number of base classifiers selected. When the weighted combination of selected base networks are conducted, both the GASEN-w and SVM-w have reduced er-

**Table 1. Selecting and Combining Base Classifiers by SVMs and Genetic Algorithms**

| | Credit(German) | Chess | Waveform |
|---|---|---|---|
| GASEN | 0.2813 | 0.0497 | 0.0961 |
| SVM | 0.2717 | 0.0465 | 0.0885 |
| GASEN-w | 0.2470 | 0.0247 | 0.0825 |
| SVM-w | 0.2313 | 0.0192 | 0.0813 |
| GASEN-wa | 0.2393 | 0.0224 | 0.0823 |
| SVM-wa | 0.2263 | 0.0179 | 0.0823 |

ror rates compared with the GASEN and SVM. It can be noted that SVM-w still outperforms the GASEN in this case. Finally, for the GASEN-wa and SVM-wa, they are the weighted combination of all learners without any learner selection, and have the lowest error rates, compared with the approach of the majority voting of all learners without learner selection (Data not shown), and the GASEN or SVM, as well as the GASEN-w or SVM-w. But it has been pointed out [27] that the selection approach has the advantage of significantly reducing the number of base learners in the final ensemble. In our experiments, the number is reduced from $10$ to the average of $4$ $5$ for different datasets.

### 3.3 Applications

We have two ongoing collaboration projects, on which the SVM-guided ensemble approaches will be tested. The first project is siRNA efficacy prediction [17]. It has been conjectured the 19 nucleotides of each siRNA determine its efficacy in achieving gene silencing, but others argued that sequence features of siRNA may not enough for knowing the efficacy. Nevertheless, the current performance of predictors are not supportive enough for reliable siRNA design. However, current approaches are often guided by experts' domain knowledge and involves heuristic or manually constructed rule-based predictors. These classifiers have been demonstrating not too bad performance on small siRNA data sets while demonstrating diverse even conflicting predictions when being compared. We are conducting the study of combining these diverse predictors, based on a moderately large siRNA data set collected and screened by our collaborators.

The second applications ensemble learning can work on is web query categorization. Given a huge set of user queries in a less labelled manner, the task is to predict the queries' categories automatically. Provided that many search engines can return query results, it is thus meaningful to study the combination of the responses from multiple search engines or related sources. Provided that single source may always have its hard-to-improve baseline accuracy, the improvement to the combination should play an important role in increasing the overall accuracy. We study this task setup by KDDCUP2005, due to the collaboration nature of this project and its large-scale, the way to combine the results from hybrid approaches needs some careful consideration.

## 4. Conclusion

This short paper presents the regularization framework for base learner selection and combination. This formulation can result in better weights for selecting and combining base learners. Results show the approach performs better than the GASEN approach, which in turn has been verified to be better than popular voting-based bagging and boosting approaches.

In our current implementation, the standard $l-2$ SVM is used for the selection and combination. But other variants can be adapted to the learning of selecting and combining base classifiers. But standard SVMs do not solve multiclass tasks very well, which will be a further study issue for learning regularized classifier combination.

It is intuitive that techniques for feature selection specific for data analysis are most likely usable for the base learner selection. It is essential to investigate whether they are exactly the same.

In ensemble study, the properties of base learners are intensively studied, including their noise tolerance, complexity, bias and variance error decomposition, etc. Can these studies lead to better ways of selecting and combining base learners, or has they already defined the limitation?

## References

[1] E. Bauer and R. Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, 36(1-2):105–139, 1999.

[2] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[3] T. G. Dietterich. Ensemble methods in machine learning. *Lecture Notes in Computer Science*, 1857:1–15, 2000.

[4] T. G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, 40(2):139–157, 2000.

[5] T. Evgeniou, M. Pontil, and T. Poggio. Regularization networks and support vector machines, 2000.

[6] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *European Conference on Computational Learning Theory*, pages 23–37, 1995.

[7] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *International Conference on Machine Learning*, pages 148–156, 1996.

[8] S. Gunter and H. Bunke. An evaluation of ensemble methods in handwritten word recognition based on feature selection. In *ICPR '04: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 1*, pages 388–392, Washington, DC, USA, 2004. IEEE Computer Society.

[9] S. Gutta, J. Huang, B. Takacs, and H. Wechsler. Face recognition using ensembles of networks, 1996.

[10] L. K. Hansen and P. Salamon. Neural network ensembles. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(10):993–1001, 1990.

[11] Y. Jiang and Z.-H. Zhou. Editing training data for knn classifiers with neural network ensemble, 2004.

[12] T. Joachims. Making large-scale support vector machine learning practical. In A. S. B. Schölkopf, C. Burges, editor, *Advances in Kernel Methods: Support Vector Machines*. MIT Press, Cambridge, MA, 1998.

[13] J. Kolter and M. Maloof. Dynamic weighted majority: A new ensemble method for tracking concept drift. Technical Report CSTR-20030610-3, Department of Computer Science, Georgetown University, Washington, DC, June 2003.

[14] E. B. Kong and T. G. Dietterich. Error-correcting output coding corrects bias and variance. In *International Conference on Machine Learning*, pages 313–321, 1995.

[15] A. Krogh and J. Vedelsby. Neural network ensembles, cross validation, and active learning. In G. Tesauro, D. Touretzky, and T. Leen, editors, *Advances in Neural Information Processing Systems*, volume 7, pages 231–238. The MIT Press, 1995.

[16] D. Opitz and R. Maclin. Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, 11:169–198, 1999.

[17] S. O. J. Saetrom P. A comparison of sirna efficacy predictors. *Biochem Biophys Res Commun.*, 13(321):247–253, 2004.

[18] B. Z. Saso Dzeroski. Is combining classifiers with stacking better than selecting the best one? *Machine Learning*, 54(3):255–273, 2004.

[19] P. Sollich and A. Krogh. Learning with ensembles: How overfitting can be useful. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems*, volume 8, pages 190–196. The MIT Press, 1996.

[20] J. A. K. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural Processing Letters*, 9(3):293–300, 1999.

[21] R. Tibshirani. Regression shrinkage and selection via the lasso, 1994.

[22] G. Tsoumakas, L. Katakis, and I. Vlahavas. Effective Voting of Heterogeneous Classifiers. In *The 15th European Conference on Machine Learning (ECML) and the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)* , Pisa, Italy, September 2004.

[23] N. Ueda. Optimal linear combination of neural networks for improving classification performance. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(2):207–215, 2000.

[24] R. Vilalta and Y. Drissi. A perspective view and survey of metalearning, 2002.

[25] J. Weston, A. Elisseeff, B. Scholkopf, and M. Tipping. The use of zero-norm with linear models and kernel methods, 2002.

[26] D. H. Wolpert. Stacked generalization. *Neural Networks*, 5:241–259, 1992.

[27] W. T. Z.-H. Zhou, J. Wu and Z.-Q. Chen. Selectively ensembling neural classifiers, 2002.

[28] Z.-H. Zhou and Y. Jiang. Medical diagnosis with c4.5 rule preceded by artificial neural network ensemble, 2003.

[29] J. Zhu, S. Rosset, T. Hastie, and R. Tibshirani. norm support vector machines, 2003.

# Solving Large-scale POMDP Problems Via Belief State Analysis

Li Xin, William K. Cheung, Jiming Liu
Department of Computer Science
Hong Kong Baptist University
Kowloon Tong, Hong Kong
Email:{lixin, william, jiming}@comp.hkbu.edu.hk

## Abstract

*Partially observable Markov decision process (POMDP) is commonly used to model a stochastic environment for supporting optimal decision making. Computing the optimal policy for a large-scale POMDP is known to be intractable. Belief compression, being an approximate solution, reduces the belief state to be of low dimension and has recently been shown to be both efficient and effective in improving the problem tractability. In this paper, with the conjecture that temporally close belief states could be characterized by a lower intrinsic dimension, this paper proposes to cluster belief states based on their spatial and temporal similarities, resulting in belief state clusters as sub-POMDPs of much lower intrinsic dimension and to be distributed to a set of agents for collaborative problem solving. The proposed method has been tested using a synthesized navigation problem (Hallway2) and empirically shown to be able to result in policies of superior long-term rewards when compared with those based on only belief compression. Some future research directions for extending this belief state analysis approach are also included.*

## 1 Introduction

Markov decision process (MDP) is commonly used to model a stochastic environment for supporting optimal decision making. An MDP model consists of a finite set of states, a set of corresponding state transition probabilities and a reward function. Solving an MDP problem means finding an optimal policy which maps each state to an action so as to achieve the best long-term reward. One of the most important assumptions in MDP is that the state of the environment is fully observable. This, however, is unfit to a lot of real-world problems. Partially observable Markov decision process (POMDP) generalizes MDP in which the decision process is based on incomplete information about the state. A POMDP model is essentially equivalent to that of MDP with the addition of a finite set of observations and a set of corresponding observation probabilities. The policy of a POMDP is now a mapping from histories of observations to actions.

For POMDPs, belief state is taken as a probability distribution over the unobservable real states as an effective summary of the observation history and it is updated based on the last action, the current observation, and the previous belief state using the Bayes rule. The policy of a POMDP is thus a mapping from a belief state to an action. The complexity of computing the optimal policy for a POMDP is much higher than that of an MDP with finite states due to the continuous belief space. The best bound for obtaining the exact solution is doubly exponential in the horizon (the time steps which the problem will iterated over) [3]. For large-scale POMDP problems, it is computationally infeasible even though it is known that the value function can be proven piecewise linear and convex (PWLC) over the internal state space [1].

In the literature, there exist a number of different methods proposed to solve large-scale POMDP problems efficiently via different elegant approximation used, including the witness algorithm [1], VDC algorithm [10], BFSC algorithm [11], etc. Another orthogonal direction is to take the divide-and-conquer approach so as to result in some scalable solutions. In addition, the problem decomposition approach can further facilitate the problem solving to be conducted in a multi-agent setting. While there have been some previously work on automatic decomposition of POMDP, efficient and effective paradigms to support POMDP decomposition and distribution are still lacking.

In this paper, we are inspired by the recently proposed belief compression approach for the fact that analyzing a sample of belief states computed based observations could provide us a lot of hints for reducing the problem complexity in a problem-specific manner. For example, based on the observation that belief states of POMDP can typically be characterized by a much lower dimensional state space, the belief compression approach uses dimension reduction techniques, like PCA and exponential PCA, to reduce the problem complexity. With the conjecture that temporally close belief states could be characterized by a set of clusters, each with a further reduced intrinsic dimension, this paper proposes to cluster belief states based on their spatial (in the belief space) and temporal similarities, resulting in belief state clusters as sub-POMDPs of much lower intrinsic dimension and to be distributed to a set of agents for collaborative problem solving. We have tested the proposed method using a synthesized navigation problem and showed that the belief state clustering approach can result in policies of superior long-term rewards when compared with those based on standard belief compression.

The remaining of this paper is organized as follows. Section 2 provides the background on belief compression. Section 3 describes the proposed belief state clustering technique. Section 4 provides the details for computing the sub-POMDP policies and how they are used for solving the entire POMDP as a whole. Experimental results are reported in Section 5 with possible extensions included in Section 6. Section 7 concludes the paper.

## 2 Belief Compression

Belief compression is a recently proposed paradigm [7], which reduces the sparse high-dimensional belief space to a low-dimensional one via projection. The principle behind is to explore the redundancy in computing the optimal policy for the entire belief space which is typically sparse. Using a sample of belief states computed based on observations of a specific problem, data analysis techniques like exponential principal component analysis (EPCA) can be adopted for characterizing the originally high-dimensional belief state space using a compact set of belief state bases. This paradigm has been found to be effective in making POMDP problems much more tractable.

Let $S$ denote the set of true states, $\mathbb{B}$ denote the belief state space of dimension $|S|$, $b \in \mathbb{B}$ denote the belief state where its $j^{th}$ element $b_i(j) \geq 0$ and $\sum_{j=0}^{|S|} b_i(j) = 1$, $B$ denote a $|S| \times n$ matrix defined as $[b_1|b_2|...|b_n]$ where $n$ is the number of belief states in the training sample.

According to [7], one can apply EPCA and obtain a $|S| \times l$ transformation matrix $U$ which factors $B$ into the matrices $U$ and $\widetilde{B}$ such that

$$B \approx e^{U\widetilde{B}} \tag{1}$$

where each column of $B$ equals $b \approx b^r = e^{U\widetilde{b}}$ and the dimension of $\widetilde{B}$ is $l \times n$. As the main objective of $U$ is for dimension reduction, it is typical that $l << |S|$.

To compare with some standard dimension reduction techniques like PCA, EPCA is found to be more effective in dimension reduction. Also, EPCA can guarantee all the elements of a belief state to be positive, which is important as each belief state is a probability distribution by itself. However, the transformation is a non-linear one, making the value function of the projected belief states no longer piecewise linear. The consequence is that many existing algorithms taking the advantage of the piecewise-linear value function become not applicable together with belief compression. As suggested in [7], those sampled belief states in the projected space can be used as the states of a correspondingly formed MDP. One can then compute the optimal policy for that associated MDP.

## 3 Clustering Belief States for POMDP Decomposition

### 3.1 General Ideas

Rather than being yet another technique to address the POMDP's scalability issue, we perceive that the belief compression approach in fact opens up a new dimension for tackling POMDP problems. That is the possibility to apply data analysis techniques to the belief state space, leading to the possibilities of having more elegant problem solving tricks.

### 3.2 Dimension Reduction Oriented Clustering

As mentioned in [7], the efficiency of belief compression is owing to its strategy for tackling the high-dimensional belief state which is one of the main causes for the exponential complexity. To further exploit the dimension reduction paradigm, we propose to decompose POMDP by analyzing the manifold of a set of sampled belief states for clustering. We anticipate that in those of the cases, there should exist some clusterings which could result in more substantial dimension reduction per cluster when compared with that of the overall belief states. In other words, the clustering criterion that we are looking for is one that is formulated to maximize the with-in cluster problem regularity to account for the further reduction. To contrast, most of the conventional data clustering techniques try to identify data clusters for maximizing the overall inter-cluster variance/distance while at the same time minimizing the overall intra-cluster variance/distance.

This idea can be intuitively interpreted as exploitation of the structural modularization from the belief state perspective. Thus, the proposed belief state clustering has some analogy with POMDP decomposition. However, in the literature, most of the proposed POMDP decomposition techniques focus on analyzing the original states of the POMDP, instead of based on the statistical properties of belief state occurrence as what being proposed in this paper.

### 3.3 A Spatio-Temporal Criterion Function for Clustering

In this paper, we propose to cluster the belief states based on both their euclidean distance as well as their temporal difference, with the conjecture that regularities should be easier to identify for temporally close belief states. Among all the clustering algorithms, the $k$-means algorithm [6] is here chosen just for the simplicity reason. It bases on a function defined for measuring the distance between the cluster means and each data item. Data found to be closest to one of the cluster means will contribute to the update of that mean in the next iteration. The whole process will repeat until it converges. For clustering belief states with the dimension-reduction objective, we define a spatio-temporal distance function between two belief states, given as

$$\begin{aligned} dist(b_i, b_j) &= dist_{spatial}(b_i, b_j) + \lambda dist_{temporal}(b_i, b_j) \\ &= \sqrt{\|b_i - b_j\|^2 + \lambda \|\frac{i-j}{n|S|}\|^2} \end{aligned} \tag{2}$$

where $\lambda$ is a trade-off parameter for controlling the relative contribution of the first (spatial) term and the second (temporal) term. $n|S|$ is introduced to normalize the second term to be within $[0, \frac{1}{n}]$. If $\lambda$ is too large, it will dominate the first term and the $k$-means clustering results will essentially be cutting the belief state sample into some consecutive parts according to the belief state appearance sequence in the sample. Also, the value of $k$, i.e., the number of clusters, is another parameter that one can tune for optimal belief state dimension reduction. To determine the values of $\lambda$ and $k$, we only used an empirical procedure in this paper to be explained in the subsequent section. It is in fact possible to replace the $k$-means clustering with methods like mixture of Gaussians so that the data partition becomes soft instead of hard and the analytical derivation of optimal $\lambda$ could be possible. This part will further be pursued due to the promising empirical results we obtained in this paper.

As just mentioned, the optimality of $k$ and $\lambda$ should be defined based on some criterion function which measures the difference between the original belief states and the reconstructed belief states after belief compression is applied. As each belief state is a probability distribution, Kullback-Leibler (KL) divergence could be used for evaluating the discrepancy between the original belief states and the reconstructed belief states, as given in Eq.(3).

$$\overline{KL}(B) = \frac{\sum_{i=1}^{n} KL(b_i \| b_i^r)}{n} \tag{3}$$

$$KL(b_i \| b_i^r) = \sum_{j=1}^{|S|} b_i(s_j) \ln \left( \frac{b_i(s_j)}{b_i^r(s_j)} \right). \tag{4}$$

For the original belief compression, the compression is based on primarily one transformation matrix $U$ as described in Section 2. Now, as the belief states are clustered, there will be several transformation matrices, each corresponding to a particular cluster. Let the belief state sample be partitioned into $P$ clusters $\{C_1, C_2, ..., C_P\}$ and the transformation matrix of the $p^{th}$ cluster $C_p$ to be $U_p$. The reconstructed belief states associated to $C_p$ can then be approximated as $b^{r,C_p} = e^{U_p \widetilde{b}}$. To measure the dimension reduction effectiveness via the clustering, the KL divergence per cluster is to be computed, given as

$$\overline{KL}(C_p) = \frac{\sum_{b_j \in C_p} KL(b_j \| b_j^{r,C_p})}{|C_p|}. \tag{5}$$

Before proceeding to the next section for computing the policy, we would like to highlight the fact that clustering the belief states can result not only in reducing the overall complexity for solving the original POMDP problem, but also that for performing the EPCA for belief compression and that for computing the transition probabilities of the projected belief states. This computational gain is achieved at the expense of the clustering overhead as well as the optimality of the resulting policy that we may sacrifice after the problem decomposition. Fortunately, the clustering overhead is found to be not significant when compared with the overall complexity. For the resulting policy's optimality, the results we obtained so far are very positive.

## 4 Computing POMDP Policy

As mentioned in Section 2, those existing efficient exact algorithms (e.g. Witness algorithm [1]) no longer fit to solve the POMDP problem with reduced dimension due to the non-linear projection due to EPCA. As in [7], we use the MDP value iteration method on the low-dimensional sampled belief states to get an approximate policy, which has been proven to be a bounded-error approximation in [4]. While we do not have major contribution in this part, related formulations are still repeated here for completeness.

Let $\widetilde{B}$ denote the set of belief state clusters, each being associated with a different transformation matrix $U_p$. Thus, we have

$$\widetilde{B} = \{\widetilde{B}^{C_1}, \widetilde{B}^{C_2}, ..., \widetilde{B}^{C_P}\} \tag{6}$$

where

$$\widetilde{B}^{C_i} = \{\widetilde{b}_j | b_j^r \in C_i\}. \tag{7}$$

The approximate value iteration algorithm uses the following rule to compute a $t$-step lookahead value function $V^t$ from a $(t-1)$-step lookahead value function $V^{t-1}$, given as

$$
\begin{aligned}
V^t(\widetilde{b}_i^{C_k}) &= max_a(\widetilde{R}^{C_k}(\widetilde{b}_i^{C_k}, a) \\
&\quad +\gamma \sum_{\widetilde{b}_j^{C_k}} \widetilde{T}^{C_k}(\widetilde{b}_i^{C_k}, a, \widetilde{b}_j^{C_k}) \cdot V^{t-1}(\widetilde{b}_j^{C_k}))
\end{aligned} \tag{8}
$$

where $\widetilde{R}^{C_k}$ and $\widetilde{T}^{C_k}$ are the approximate reward and transition functions in the corresponding partitioned low-dimensional space.

### 4.1 Computing the Reward Function

The reward function $R(s_i, a)$ denotes an immediate reward if taking an action $a$ at state $s_i$. Naturally, an immediate reward after taking an action $a$ at belief state $b_j$ or $\widetilde{b_j}$ should be the expected value over the all true states. See following equation:

$$
\widetilde{R}^{C_i}(\widetilde{b_j}, a) = \sum_{i=1}^{|S|} R(s_i, a) b_j(s_i) \tag{9}
$$

Note that in some problems, there is another form of reward function $R(s_i, a, s_j)$ which means the immediate reward is also relative to the state to be reached. Also, we can get the expected $R(s_i, a)$ from $R(s_i, a, s_j)$,

$$
\widetilde{R}(s_i, a) = \sum_{j=1}^{|S|} R(s_i, a, s_j) T(s_i, a, s_j) \tag{10}
$$

### 4.2 Computing the Transition Function

Computing the transition function of the projected belief states is a bit more complicated. One should first recur to the transition trajectory of the high-dimensional space based on the Bayes rules. It is a process in and out of the high-dimensional and low-dimensional space to accomplish the beliefs' evolvement, projection, reconstruction and matching, as described in [7]. For our proposed method, we only consider pairs of low-dimensional beliefs in the same cluster, regardless of the possible transitions between clusters. Thus, we get the transition function $\widetilde{T}^{C_k}(\widetilde{b}_i^{C_k}, a, \widetilde{b}_j^{C_k})$. as the sum of $p(z, j|i, a)$ over all observations $z$, *i.e.*,

$$
p(z, j|i, a) = \omega(\widetilde{b}_j^{C_k}, \widetilde{b'}^{C_k}) \sum_{l=1}^{|S|} p(z|s_l) b_a^{C_k}(s_l) \tag{11}
$$

where $\widetilde{b'}^{C_k}$ is the low-dimensional belief projected from a high-dimensional belief $b^{C_k}$ of a cluster $C_k$, which is updated after executing an action and receiving an observation from the high-dimensional reconstruction of $\widetilde{b}_i^{C_k}$ using $b^{r,C_k} = e^{U\widetilde{b}^{C_k}}$. $\omega(\widetilde{b}_j^{C_k}, \widetilde{b'}^{C_k}) = \frac{1}{k}$ as we use $k$-nearest-neighbor for approximate discretization on the low-dimensional belief space. Also, $p(z|s_l)$ in Eq.(11) can be computed as

$$
p(z|s_l) = \sum_{i=1}^{|S|} p(a_i|s_l) p(z|a_i, s_l) \tag{12}
$$

with $p(a_i|s_l) = \frac{1}{|Actions|}$ and $p(z|a_i, s_l)$ is the given observation probability. For $b_a^{C_k}(s_l)$ in Eq.(11), it denotes the expected belief and can be computed as

$$
b_a^{C_k}(s_l) = \sum_{j=1}^{|S|} T(s_j, a, s_l) b_i^{C_k}(s_j) \tag{13}
$$

. It is updated only by executing an action instead of using both action and observation.

Generally speaking, constraining the transitions within clusters will bring some reward information loss and weaken the policy quality accordingly. However, the spatio-temporal clustering we adopted is essentially geared to reduce the loss to a certain extent since it is based on the conjecture that belief state visited within a short period will try to be clustered as far as possible based on the spatio-temporal notion. In other words, good clustering results should benefit not only dimension reduction, but also the accuracy of the subsequently computed policy.

### 4.3 Value Function Computation and Policy Application

The final step is to compute the value function for each cluster to get the policy tables corresponding to the clusters using the reward and transition functions computed according to the previous two subsections. Based on Eq.( 8), the conventional MDP value iteration algorithm can be used, which will stop when the value at time step $t + 1$ is mathematically close to the value at time step $t$.

To apply the policy in a multi-agent setting, the computed policy tables will be distributed to different agents and a coordinating agent is needed with the role of selecting which agents to forward a new observation to based on comparing the corresponding high-dimensional belief state with the sampled beliefs. Our implementation selects the nearest one which is indexed with the corresponding agent for taking the next step action based on the agent's policy table.

## 5 Experimental Results

### 5.1 The Hallway2 Problem

The Hallway2 Problem which is defined with a specific maze is commonly used to test the scalability of algorithms for solving POMDP problems (see also [1]). The problem is to find the goals in the maze with 92 states (4 being the goal states), and contains 5 actions and 17 types of observations. Reaching one of the goal states will yield a $+1$ reward and then the next state will be set to a random non-goal state. In addition, it is assumed that all the non-goal states of the problem are equally likely to be the initial state location and thus the starting belief state is $b_1 = (\frac{1}{88}, ..., \frac{1}{88})^T$. Also, the discount factor used is 0.95. In this paper, all the experimental results reported are based on this problem setting.

### 5.2 Belief State Sampling

The process of belief compression is operated on a belief state sample generated via simulation. During the simulation for sample generation, two levels of random numbers are used to select an action, and the Bayes rules are used to evolve the belief states. When one random number is found to be less than the threshold defined as 0.5, another random number will be generated to decide the next action. Otherwise, it will sum up all the beliefs generated so far and take the state with the maximal sum of probabilities as the current state. Then, an MDP solver will be called to get the corresponding policy table to choose the next action for its 'current state'.

The belief states in consecutive time steps often have similar shape with the same number of modes. These "structural" similar belief states could have them represented at a much lower dimension. That's why the belief state space is often considered to be sparse.

### 5.3 Performance of Belief State Clustering

The first experiment focuses on evaluating the effectiveness of the proposed spatio-temporal clustering scheme for overall dimension reduction. We enumerated a set of different values for the trade-off parameter $\lambda$ and evaluated the corresponding dimension reduction performance. For performance measurement, we contrasted the values of the KL-divergence between the set of original belief states and the ones reconstructed based on the conventional belief state compression (i.e., without clustering) and the one we proposed in this paper with belief state clustering. Figure 1 shows a cube with three axes being reduced dimension, the number of clusters and KL-divergence. We only plot the dots where the averaged KL divergence of the clusters is less than the averaged KL divergence using the conventional belief compression. According to Figure 1, it is noted that a number of settings can result in better overall dimension reduction. Among those settings, we set a filter and highlight those with high reduction. The filtering is based on a ratio $R$, defined as

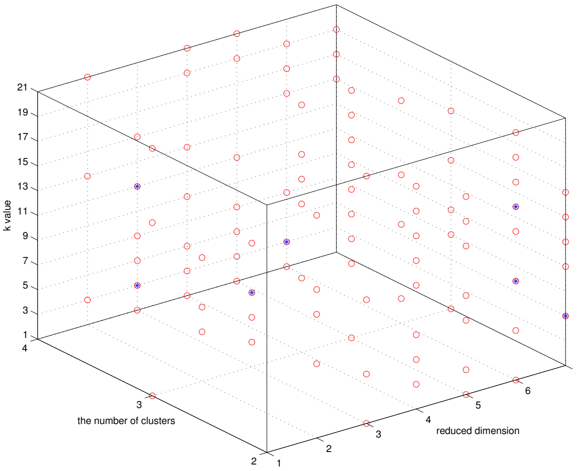


**Figure 1. The best parameter settings with $R > 0.95$.**

$$R(\lambda, l, P) = 1/P * \sum_{p=1}^{P} \frac{(KL_U(\lambda, l, C_p) - KL_{U_p}(\lambda, l, C_p))}{KL_U(\lambda, l, C_p)} \tag{14}$$

where $KL_U(\lambda, l, C_p)$ stands for the KL-divergence between the original belief states in the $p^{th}$ cluster and the corresponding reconstructed belief states based on only $U$ (original EPCA), and $KL_{\{U_p\}}(\lambda, l, C_p)$ stands for the KL-divergence between the original belief states in the $p^{th}$ cluster and the corresponding reconstructed belief states based on $U_p$ (resulted from applying EPCA to the cluster).

In Figure 1, those solid points show the parameter settings which result in having $R > 0.95$. Among them, the operation point $R(3, 3, 4)$ was chosen. This point is equivalent to the situation that the belief state sample is partitioned into 4 clusters, its dimension is reduced to 3, and the trade-off parameter $\lambda$ is 3. Figure 2 shows a particular belief state and two reconstruction using the conventional EPCA (the upper diagram) and the proposed clustered EPCA (the lower diagram). The latter one's reconstruction can almost completely overlap the original belief state, which is a more accurate reconstruction than the former one. Also, Figure 3 shows the temporal sequence of the belief states in each cluster. It is noted that in some clusters (e.g., Cluster 1, 3), the belief states under them are only partially ordered, which is consistent to the spatio-temporal criterion function used.

Table 1 tabulates the performance measures for comparing the KL divergence given the belief state dimension is

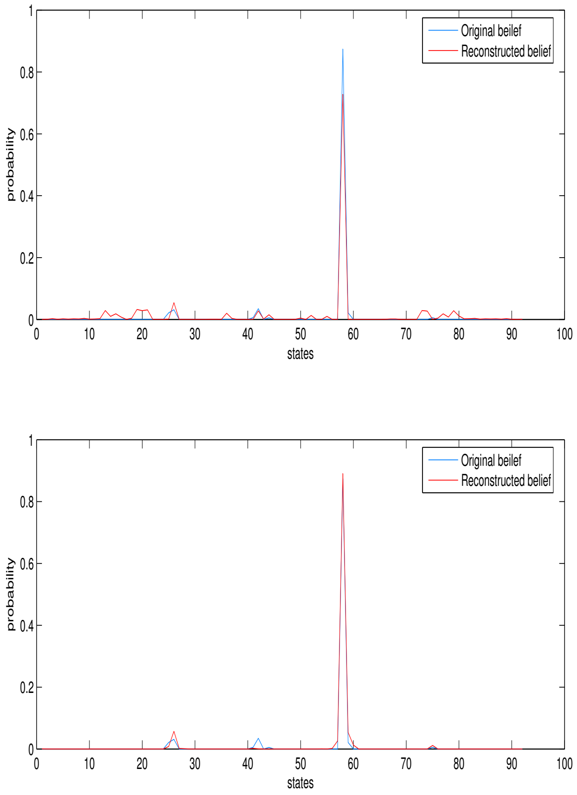


**Figure 2. Reconstructed belief states using the conventional EPCA and the proposed method.**

reduced to three. Obviously, the values of the KL divergence obtained using the proposed spatio-temporal clustering were much lower than the case using only EPCA. Figure 4 shows the changes of the average KL-divergence at different reduced dimensions using EPCA. Note that the KL-divergence values in the figure are much bigger than those reported in Table 1. In addition, as reported in the last column of Table 1, our proposed method took 71.94 seconds while the conventional EPCA took 153.08 seconds.

## 5.4 Policy Quality with Spatio-Temporal Clustering Introduced

In terms of those operation points $(\lambda, l, P)$ (could be interpreted as model configurations) with significant intrinsic KL-Divergence reduction, we compute the policy for each cluster and test the policy. The comparison of policy per-

| | # Items | Original EPCA | Proposed Method | Comp. Cost (sec.) |
|---|---|---|---|---|
| $Cluster1$ | 96 | 1.3997 | 0.0024 | 1.84 |
| $Cluster2$ | 16 | 0.4998 | 0.0004 | 0.34 |
| $Cluster3$ | 36 | 0.1893 | 0.0003 | 0.49 |
| $Cluster4$ | 352 | 4.2596 | 0.4938 | 69.27 |

**Table 1. Performance comparison between the conventional EPCA for belief compression and the proposed method, where the number of clusters is 4, the reduced dimension is 3 and $\lambda = 3$.**

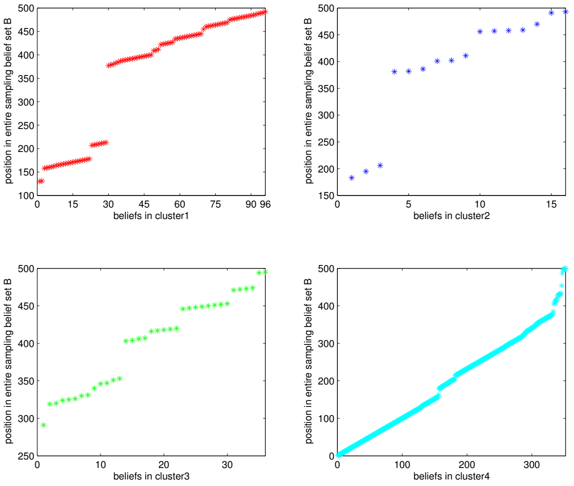


**Figure 3. Temporal sequence of the belief states in the four clusters.**

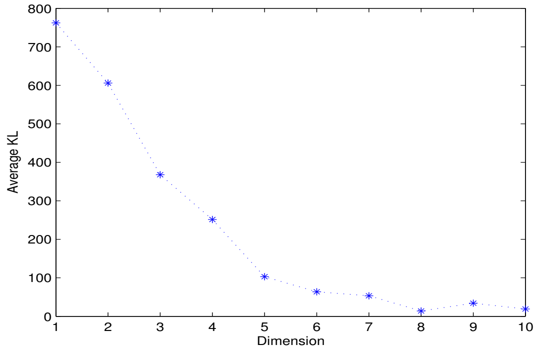


**Figure 4. Average KL Divergence for conventional EPCA.**

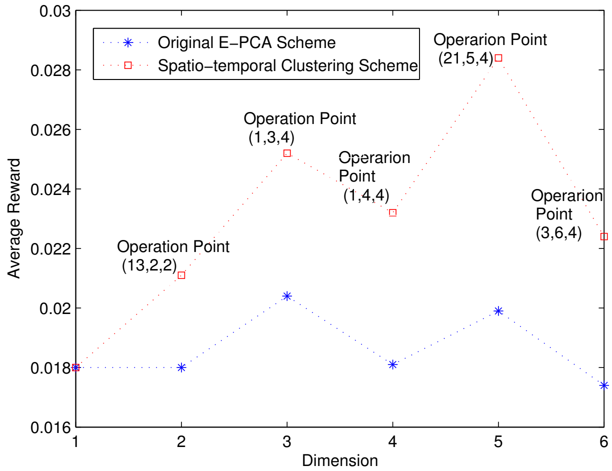


**Figure 5. A comparison of policy performance using different schemes, different number of clusters $P$ and different reduced dimensions $l$ for average reward over 1000 trials. The operation point is labelled as $(\lambda, l, P)$.**

formance occurs between the computing policy via dimension reduction directly and computing policies via spatio-temporal clustering using different numbers of bases. For each operation point, we execute 1000 trails. Each trail is a trajectory with maximal 251 steps before any one of the objective states is reached. The trajectory is evolved by executing the computed policies. Our experimental results show that nearly half of these operation points result in the policy quality enhancement, and some of them help increase the average reward greatly.

According to Figure 5, we can see obvious performance enhancement over the conventional belief compression. For the Hallway2 problem with 500 sampling beliefs, it also shows that using four clusters is a generally better strategy.

Table 2 lists out the detailed parameter settings of some operation points selected for performance comparsion. Generally speaking, those settings with relatively higher $R(\lambda, l, P)$ ratio induces a better average reward, which is consistent to our conjecture that a clustering with a better dimension reduction power should also result in a policy of higher quality. It is also noted that it is hard to set a threshold of getting a good set of $l$ and $P$ as the value of the ratio $R(\lambda, l, P)$ for resulting in better policy varies quite a lot given different set of $l$ and $P$.

As being discussed before, the performance enhancement is induced by the much more accurate low-dimensional representation, though some rewards among clusters are lost inevitably. Our experimental results are consistent with what we have discussed in the previous section and show that the reward loss do not affect much the overall performance given good spatio-temporal clustering results.

| avg. reward without clustering: | avg. reward with clustering: | | | | |
|---|---|---|---|---|---|
| at $l$ | at O-Point | $\lambda$ | $l$ | $P$ | $R(\lambda, l, P)$ |
| 0.0180 | 0.0191 | 11 | 2 | 2 | 0.1781 |
| | 0.0211 | 13 | 2 | 2 | 0.3646 |
| | 0.0182 | 19 | 2 | 2 | 0.2167 |
| 0.0180 | 0.0180 | 3 | 2 | 4 | 0.7857 |
| | 0.0161 | 13 | 2 | 4 | 0.5229 |
| | 0.0149 | 21 | 2 | 4 | 0.4954 |
| 0.0199 | 0.0258 | 19 | 5 | 4 | 0.6387 |
| | 0.0284 | 21 | 5 | 4 | 0.7044 |
| 0.0174 | 0.0224 | 3 | 6 | 4 | 0.7857 |
| | 0.0214 | 17 | 6 | 4 | 0.5229 |
| | 0.0185 | 19 | 6 | 4 | 0.4954 |

**Table 2. Performance comparision for different parameter settings.**

## 6 Discussion and Future Works

This paper mainly demonstrates the possibility of clustering the belief states in a spatio-temporal manner for achieving further belief state compression and good policy performance. We are currently working on several extensions of this work as depicted as follows.

### 6.1 Towards Optimal Spatio-Temporal Clustering

While the criterion function used in this paper has shown to be effective empirically, it is by no means an optimal choice. In addition, we still lack automatic mechanisms (other than exhaustive search) for setting the parameters to govern the clustering. We believe that this is an immediate and important research direction to be pursued in the future.

### 6.2 Hierarchical POMDP Decomposition

Hierarchical POMDP (HPOMDP) Decomposition has recently been proposed for decomposing a POMDP problem into a hierarchy of related POMDP models of reduced size. PolCA [9] is a representative example where the decomposition is resulting from the decomposability of the action domain. The limitation of HPOMDP is that the decomposition is not fully automatic, where the underlying hierarchy of actions requires knowledge of domain experts. In other words, this is domain-specific. Also, the decomposition is not based on the belief states. It would be interested to see if the notion of hierarchical decomposition can be incorporated in the proposed spatio-temporal clustering framework with further performance gain.

### 6.3 The Multi-Agent Consideration

In this paper, we distribute each sub-POMDP to a problem solving agent. The agents are basically independent to each other, except to be coordinated by the brokering agent. As the decomposition based on the proposed belief state clustering may not result in a set of sub-POMDP problems which are equivalent to the original POMDP problems, interaction between those agents for achieving the overall optimal policy is an important research issue. Nash Equilibrium is an important concept commonly used in multi-agent learning [5] for solving decentralized MDP [2] and POMDP problems [8]. Our research agenda also includes how to apply this paradigm to the our decomposition scheme for further performance boosting. The basic idea is that every agent would conjecture other agents' behaviors and give the best response to other agents from its local view. A Nash equilibrium usually would not deduce the optimal policy. However, it should be able to guarantee a not-too-bad sub-optimal one.

What being described so far assumes that the whole model of the decision process is known. That is, we have the perfect knowledge about the reward function, transition function and observation function. Solving the corresponding POMDP problems is an off-line process. It is also interested to see how the multi-agent approach can be extend to support online learning (e.g., Q-learning [12]) for POMDP under partial observation scenarios.

## 7 Conclusion

This paper extends the recently proposed belief compression by introducing a spatio-temporal belief state clustering for addressing large-scale POMDP problems. It was found that the proposed spatio-temporal method can further compress the belief states in each cluster to a much lower dimension while maintaining similar belief state reconstruction accuracy and thus a better policy. Also, each cluster of belief states can naturely been distributed to different agents for collaborative problem solving. Future research directions include at least further enhancement in automatic determination of clustering parameters, hierarchical clustering of the belief states and the integration of the proposed belief state clustering and the multi-agent paradigm as a unified solution for solving large-scale POMDP problems.

## References

[1] A.Cassandra. *Exact and approximate algorithms for partially observable Markov decision processes*. U.Brown, 1998.

[2] R. Becker, S. Zilberstein, V. Lesser, and C. V. Goldman. Transition-Independent Decentralized Markov Decision Processes. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multi Agent Systems*, pages 41–48, Melbourne, Australia, July 2003. ACM Press.

[3] D. Burago, M. de Rougemont, and A. Slissenko. On the complexity of partially observed Markov decision processes. *Theoretical Computer Science*, 157(2):161–183, 1996.

[4] G. J. Gordon. Stable function approximation in dynamic programming. In A. Prieditis and S. Russell, editors, *Proceedings of the Twelfth International Conference on Machine Learning*, pages 261–268, San Francisco, CA, 1995. Morgan Kaufmann.

[5] M. P. W. Junling Hu. Nash q-learning for general-sum stochastic games. *Journal of Machine Learning Research*, 4:1039–1069, 2003.

[6] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *5th Berkley Symposium on Mathematics and Probability*, pages 281–297, 1967.

[7] N. Roy, G. Gordon and S. Thrun. Finding approximate POMDP solutions through belief compressions. *Journal of Artificial Intelligence Research*, 23:1–40, 2005.

[8] R. Nair, M. Tambe, M. Yokoo, D. Pynadath, and S. Marsella. Taming decentralized POMDPS: Towards efficient policy computation for multiagent settings. 2003.

[9] J. Pineau and S. Thrun. An integrated approach to hierarchy and abstraction for POMDPS. Technical Report CMU-RI-TR-02-21, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, August 2002.

[10] P. Poupart and C. Boutilier. Value-directed compression of POMDPS. In S. T. S. Becker and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 1547–1554. MIT Press, Cambridge, MA, 2003.

[11] P. Poupart and C. Boutilier. Bounded finite state controllers. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

[12] C. Watkins. *Learning from Delayed Rewards*. PhD thesis, Cambridge Univ., Cambridge England, 1989.

# Visualizing Global Manifold Based on Distributed Local Data Abstraction

Xiaofeng Zhang
Department of Computer Science
Hong Kong Baptist University
Kowloon Tong, Hong Kong
xfzhang@comp.hkbu.edu.hk

William K. Cheung
Department of Computer Science
Hong Kong Baptist University
Kowloon Tong, Hong Kong
william@comp.hkbu.edu.hk

## Abstract

*Mining distributed data for global knowledge is getting more attention recently. The problem is especially challenging when data sharing is prohibited due to local constraints like limited bandwidth and data privacy. In this paper, we investigate, in particular, how to derive the intrinsic manifold (as a 2-D map) for a set of horizontally partitioned data which cannot be shared directly. The proposed methodology is a model-based one. It first computes hierarchical local data abstractions, then aggregates the abstractions, and finally learns a global generative model – generative topographic mapping (GTM) based on the aggregated data abstraction. We applied the proposed method to both synthetic and real datasets. The experimental results show that the derived manifold is found to be comparable to that of the original GTM without the local abstraction introduced.*

## 1. Introduction

Recent progress in automatic data collection, data storage and networking technologies has resulted in high accessability of distributed and massive data for application domains like e-Science and e-Commerce. This in turn triggered the need of performing data mining in a distributed environment. The distributed data mining problem is challenging as data sharing is in many cases prohibited due to local constraints like limited bandwidth and data privacy. The former limitation is faced when the distributed data are of high volume. The latter limitation happens when the local data owners indicate high privacy concern, even though they still prefer to have some degree of personalized e-services. Distributed data mining [13] has found applications in financial data analysis, personal transaction records analysis. medical data analysis, intrusion detection, etc.

Distributed data mining typically involves two steps – first performing local data analysis and followed by combining the local results to form a global one. For example, in [16], a meta-learning process was proposed for combining a set of locally learned classifiers (decision trees in particular) to achieve high classification accuracy. A related implementation has been realized on a Grid platform known as the *Knowledge Grid* [5]. In [13], Kargupta *et al.* proposed collective data mining where the distributed data sources possess different sets of features (also known as vertical data partition [19]). They considered each source as an orthogonal basis, and then combine them to form the overall result. This method has been applied to learning Bayesian Networks for Web log analysis [6]. In addition, distributed association rules mining algorithms with privacy preservation capability has been proposed in [1, 2].

An inevitable limitation of the aforementioned methodology is that aggregating local analysis results could result in the loss of information which is essential for the subsequent global pattern discovery. One possible remedy is to allow partial information exchange among the data sources during the local data analysis step [21]. An alternative approach for minimizing the chance of losing important local information is to adopt a flexible statistical model for abstracting the local data. The model flexibility allows the local data granularity, and thus the extent of information loss, to be controlled. A model with high complexity usually can retain more details when compared with one of low complexity The use of the model-based approach for distributed data mining was first proposed in [15, 14, 20]. In [20], Zhang *et al.* demonstrated how a global cluster model can effectively be learned based on local data abstractions.

In this paper, the distributed model-based approach was extended to derive the intrinsic manifold of a set of distributed data. Generative topographic mapping (GTM), which is an effective nonlinear mapping tool for visualizing high dimensional data sets, was chosen to be the global model due to its generative nature. Gaussian mixture model (GMM) [12] was chosen for the local data abstraction as it is generally considered to be a universal approximator (thus flexible) for arbitrary data distributions [3]. From the perspective of data privacy control, a data set represented by a GMM with fewer components provides coarser information (i.e., higher level of privacy) than one with more components. In the extreme case, a GMM with only one single component provides the coarsest information about the data set but at the highest privacy level. The model accuracy increases (and thus the privacy decreases) as the number of the GMM's components increases. This kind of representation flexibility of GMM makes it an excellent candidate to meet the diverse data privacy requirement of each disjointed local source. From the perspective of bandwidth requirement for data sharing, a GMM with a single component requires the lowest while a GMM with its number of components equal to that of the data[1] is the highest.

To learn the global model (in our case GTM), the conventional approach of regenerating *virtual* data by applying, say, Monte Carlo Markov Chain (MCMC) sampling [10] to the aggregated local data abstraction can be adopted [15]. While it has the advantage that most of the existing data mining and machine learning techniques can be used directly for the global analysis, the resampling step could be computationally expensive. Also, a sufficiently large set of virtual data has to be generated in order to result in an accurate global model, which in turn will lead to long global model training time. In this paper, we propose a modified EM-like algorithm for learning a global GTM directly from the aggregated local data abstraction. We applied the proposed method to both synthetic datasets (S-curve[2], oil data [4]) as well as real dataset (WebKB[3]) for intrinsic data manifold visualization. The experimental results showed that the proposed distributed learning approach can achieve comparably good visualization results and at the same time satisfy the limited bandwidth and data privacy requirements of the local sources in a controlled manner.

It may be worth mentioning that the distributed data mining problem concerned here is different from some related fields, *e.g.*, distributed query processing and parallel clustering. Distributed query processing [8] mainly concerns query optimization in a distributed environment instead of data analysis. Parallel clustering [18] mostly assumes that the data partitioning can be under the user's control for facilitating the global analysis/processing objective. Instead, we consider that the local data are by default distributed and cannot be partitioned purposely.

The rest of the paper is organized as follow. Section 2 describes in detail the problem formulation and a novel EM-like algorithm for learning a global GTM from a set local GMMs aggregated as the local data abstraction. Details about the experiment design as well as the corresponding results demonstrating the feasibility and effectiveness of the proposed method can be found in Section 3. Section 4 concludes the paper.

## 2 Problem Formulation

### 2.1 Local Data Abstraction

Local data abstraction is here defined as the process of representing a given set of data by its statistics forming an abstraction. Via the abstractions, the statistical information of the data can be shared, instead of the data themselves. As discussed in Section 1, we need the abstraction which can provide a handle for sharing local data as well as controlling dynamically the degree of data privacy and the bandwidth required in a distributed environment like the Internet. We formulate this abstraction process as parametric density estimation and GMMs with different numbers of components are adopted to support sharing local data details at different granularity levels.

Assume that there are totally $L$ distributed data sources. Let $t_i \in \Re^d$ denote the $i^{th}$ observed data item of dimension $d$, $\theta_l$ denote the set of parameters of the local model (GMM) as the abstraction of the $l^{th}$ source, $\theta_{lj}$ denote the $j^{th}$ component's parameters of the $l^{th}$ local model (including the component's mean $\mu_{lj}$ and covariance matrix $\Sigma_{lj}$), $\alpha_{jl}$ denote the mixing proportion of the $j^{th}$ component in the $l^{th}$ local model. The probability density function (pdf) of the $l^{th}$ local model $p_{local}(t_i|\theta_l)$ with $K_l$ components is given as,

$$\begin{aligned} p_{local}(t_i|\theta_l) &= \sum_{j=1}^{K_l} \alpha_{jl} p_j(t_i|\theta_{lj}) \\ \sum_{j=1}^{K_l} \alpha_{jl} &= 1 \end{aligned}$$

$$p_j(t_i|\theta_{lj}) = (2\pi)^{-\frac{d}{2}} |\Sigma_{lj}|^{-\frac{1}{2}} \exp\{-\frac{1}{2}(t_i - \mu_{lj})^T \Sigma_{lj}^{-1} (t_i - \mu_{lj})\}.$$

The local GMM parameters (abstractions) extracted from each of the sources, *i.e.*, $\{\theta_1, \theta_2, ..., \theta_L\}$, can then be sent to a global server to learn a global data model. Figure 1 is an illustration of the problem we are addressing. There are three distributed local data sources (Figure 1 (a)-(c)) and the objective is to identify data clusters in the global sense. As shown in Figure 1 (d)-(f), the local GMMs' parameters $\{\theta_1, \theta_2, \theta_3\}$ are first derived as the abstractions of the distributed local data by applying some model-based clustering algorithm to each of the local data sets. By aggregating the local GMMs' parameters to form an aggregated local

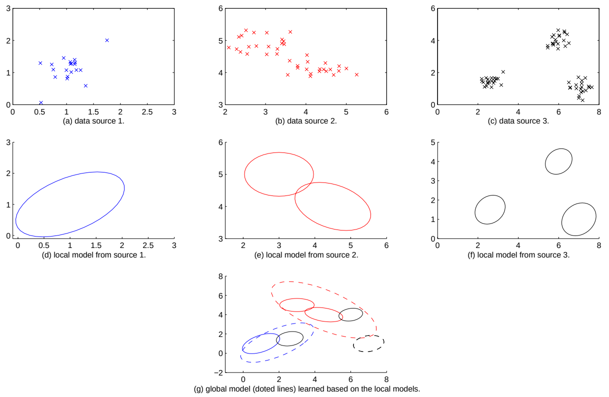
[1] In that case, one GMM component is supposed to represent one data item.

[2] The S-curve is downloadable at ”http://www.cs.toronto.edu/ roweis/lle/code.html”

[3] The WebKB dataset can be downloaded at ”http://www-2.cs.cmu.edu/afs/cs.cmu.edu/project/theo-11/www/wwkb/”

model, our goal is to acquire the global model, *i.e.*, to estimate the global model parameters directly from the aggregated local model, as shown in Figure 1(g). In principle, the global model can be any generative model. In this paper, we show the details about how a global generative topographic mapping can be learned.

To compute the local data abstraction, a fast algorithm with the ability to derive a family of GMMs with different number of components for representing the local data at different granularity levels is generally needed. Instead of using the conventional Expectation and Maximization (EM) algorithm [7] to derive the local GMMs parameters, we used in this experiment first the agglomerative hierarchical algorithm (AGH) which is one of the most commonly used bottom-up method for hierarchical data clustering. Each cluster at the bottom level of the dendrogram derived by AGH is an original data item, and a big cluster including all the data items lies on the top of AGH hierarchical tree. Levels in-between reflect different levels of data clustering details. Given a particular level of clustering (say chosen based on an individual's privacy concern), each cluster can form a local Gaussian component by computing the mean and covariance matrix of the data within the cluster. Figure 2 illustrates how the hierarchy of local data abstraction is built up.



(a) data source 1.

(b) data source 2.

(c) data source 3.

(d) local model from source 1.

(e) local model from source 2.

(f) local model from source 3.

(g) global model (doted lines) learned based on the local models.

**Figure 1. Originated from three local data sources with 1, 2 and 3 data clusters respectively (i.e., altogether 6 aggregated local components), the proposed methodology learns the global model by aggregating the abstractions from the local sources.**

## 2.2 Learning A Global GTM

Generative topographic mapping (GTM) [4] is a probabilistic non-linear latent variable model which can be used to explore the intrinsic manifold of a set of high-dimensional data. GTM assumes that the data are generated due to a set of latent variables in a low-dimensional (usually 2D) latent space via a non-linear mapping that maps a lattice in the *latent* space to the observed data in the *data* space, with the original data topology preserved in the latent space, as shown in Figure 3. Visualizing the latent space with the original high-dimensional data projected back to it can result in a map (for the 2D case) as an "unfolded" version of the intrinsic data manifold. The unfolded manifold, in many cases, can help understanding the structure and organization of the data. In the literature, there also exist other nonlinear manifold learning methods, for instance, locally linear embedding (LLE) [17] and isometric feature fapping (ISOMAP) [11]. They can also find a 2D embedding of image expression for the purpose of data visualization. Both of them work well on capturing the intrinsic nonlinear structure presented in the high dimensional data space by preserving local linear combination among data items or finding the geometric shortest distance between data items. Similar to GTM, these methods also assume the data to be fully observed, which is sometimes not the case. GTM is chosen in this paper instead of the others mainly because of its generative nature. It will be interesting to see if models related to ISOMAP and LLE can have some generative interpretations so that the proposed method can also be applied.

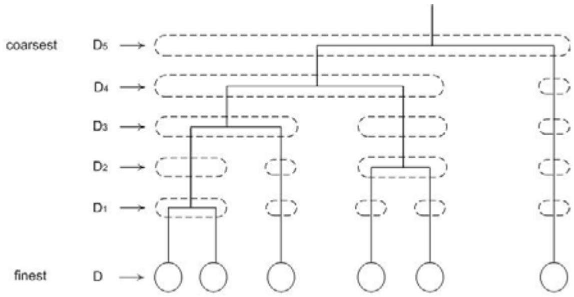


**Figure 2. A hierarchy of data abstractions, $D_1, ..., D_5$, where a higher level of abstraction is acquired by merging two nearest data subgroups at the next layer and of finer data details. For instance, $D_2$ contains four sets of means and covariance matrices to be shared for global analysis.**

### 2.2.1 GTM Formulation

Let $t_i \in \Re^d$ denote the $i^{th}$ observed data item in data space, $z_k \in \Re^H$ denote the $k^{th}$ lattice point (altogether $M$) defined in the latent space. $y(z; W) := W\Psi(z)$ maps in an

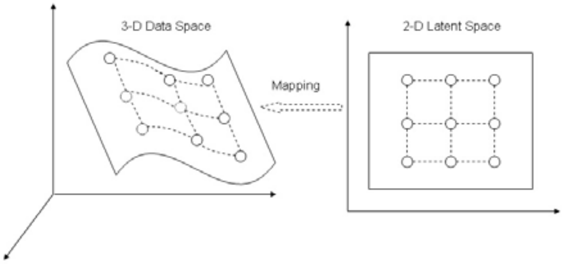


**Figure 3. An illustration of GTM.**

non-linear fashion a point $z$ in the latent space onto a corresponding point $y$ in the data space, with the mapping governed by a generalized linear regression model $\Psi$ weighted by $W$. Thus, a lattice pre-defined in the latent space would be mapped onto an $L$ dimensional non-Euclidean manifold in the data space. A multivariate Gaussian distribution in the data space is assumed in GTM for $t_i$ given $z_k$, given as

$$p(t_i|z_k, W, \beta) = (2\pi)^{-\frac{d}{2}}\beta^{\frac{d}{2}} \exp\{-\frac{\beta}{2}||(t_i - y(z_k; W)||^2\} \tag{1}$$

where $\beta$ is the reciprocal of the data variance.

The overall log likelihood function for GTM is given as

$$\sum_{i=1}^{N} \ln \frac{1}{M} \sum_{k=1}^{M} p(t_i|z_k, W, \beta) \tag{2}$$

Where $N$ is the total number of data items.

The EM algorithm is typically used for estimating the parameters $W$ and $\beta$. The E-step for the original GTM is given as

$$R_{ik}(W_{old}, \beta_{old}) = P(z_k|t_i, W, \beta) = \frac{p(t_i|z_k, W_{old}, \beta_{old})}{\sum_{j=1}^{M} p(t_i|z_j, W_{old}, \beta_{old})}$$

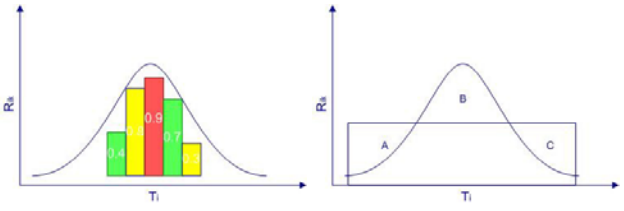
where $R_{ik}$ is the estimated indicator for the $k^{th}$ latent lattice point of the global model generating the $i^{th}$ data item and $W_{old}$ and $\beta_{old}$ are the current estimates of the GTM's parameters. The M-step is then given as

$$\sum_{i=1}^{N} \sum_{k=1}^{M} R_{ik}(W_{old}, \beta_{old})\{W_{new}\Psi(z_k) - t_i\}\Psi(z_k)^T = 0 \tag{3}$$

$$\frac{1}{\beta_{new}} = \frac{1}{Nd} \sum_{i=1}^{N} \sum_{k=1}^{M} R_{ik}(W_{old}, \beta_{old})||W_{new}\Psi(z_k) - t_i||^2 \tag{4}$$

### 2.2.2 Learning from Local Data Abstraction

In order to learn the global GTM model parameters directly from local GMM parameters, we first approximate the original estimated indicators $R_{ik}$, as shown in Figure 4 (a), by a uniform distribution, as shown in Figure 4 (b), over the data items corresponding to a particular GMM component which is now to be shared instead of the data.



(a) $R_{ik}$.

(b) $R_{lk}$.

**Figure 4. Illustration of approximating $R_{ik}$ with $R_{lk}$. Regions A, B and C correspond to the inaccuracy resulted due to the approximation.**

Assume that $R_{lk}$ is now an indicator for the $l^{th}$ local component[4] with its underlying data to be generated by the $k^{th}$ global component. That is, the likelihood of the subset of the data generated by the $k^{th}$ component of the global model is assumed to be approximated by an overall estimate of the corresponding $l^{th}$ local component being generated by the same component of the global model. By approximating $R_{ik}$ as

$$R_{ik} \approx \frac{\sum_{i \in l^{th} source} R_{ik}}{N_l} \tag{5}$$

and defining

$$R_{lk} = \sum_{i \in l^{th} source} R_{ik} \tag{6}$$

where $N_l$ denotes the number of data from the $l^{th}$ source.

To estimate $R_{lk}$ which is now an indicator of a local Gaussian component being generated by a global Gaussian component, the Kullback Leibler (KL) divergence between global and local components is used with the formulation given as

$$R_{lk} = \frac{\exp\{-D(p_{local}(t|\theta_l)||p_{gtm}(t|z_k, W, \beta))\}}{\sum_{j=1}^{M} \exp\{-D(p_{local}(t|\theta_l)||p_{gtm}(t|z_j, W, \beta))\}} \tag{7}$$

where the $KL$-divergence $D(p_{local}||p_{gtm})$ can be derived as

$$\ln \frac{\beta^{-\frac{d}{2}}}{|\Sigma_l|^{\frac{1}{2}}} + \frac{\beta}{2} tr(\Sigma_l) + \frac{1}{2}(\beta(y(z_k; W) - \mu_l)^T(y(z_k; W) - \mu_l) - d). \tag{8}$$

[4] Note that by aggregating a set of local GMMs, an equivalent overall GMM can readily be formed. In the subsequent derivation, we abuse the index "$l$" to refer to one of the local components of the aggregated local model.

When the local GMM is reduced back to a data item, the first two terms of Eq.(8) will become constant with respect to the local data and thus only the third term will be in effect. Eq.(8) is then degenerated back to the original GTM's E-step.

Accordingly, the new M-step can be derived as

$$\sum_{l=1}^{L}\sum_{k=1}^{M} R_{lk}(W_{old},\beta_{old})\{W_{new}\Psi(z_k)-\mu_l\}\Psi(z_k)^T = 0 \quad (9)$$

$$\begin{aligned}\frac{1}{\beta_{new}} &= \frac{1}{Nd}\sum_{k=1}^{M}(\sum_{l=1}^{L} R_{lk}(W_{old},\beta_{old})(\Sigma_l+\mu_l\mu_l^T)) \\ &- \frac{1}{Nd}\sum_{k=1}^{M}((W_{new}\Psi(z_k))^2\sum_{l=1}^{L} R_{lk}). \qquad (10)\end{aligned}$$

### 2.2.3 GTM Initialization Based on Local Abstraction

Given the aggregated local model, the initialization of the global GTM can be obtained as equivalent to that of the original GTM. Original GTM uses principle component analysis (PCA) for initializing $\beta$ and $W$. For the proposed method, original data are lacking for computing the global data covariance matrix, and thus the PCA. Fortunately, one can easily show that the global covariance matrix can analytically be derived based on the covariance matrices of the local data, given as

$$\begin{aligned}\mu_{global} &= \frac{\sum_{l=1}^{L} N_l\mu_l}{N} \\ \Sigma_{global} &= \frac{\sum_{l=1}^{L} N_l(\Sigma_l+\mu_l\mu_l^T)}{N} - \mu_{global}\mu_{global}^T.\end{aligned}$$

## 3 Experiments on Distributed Data Visualization

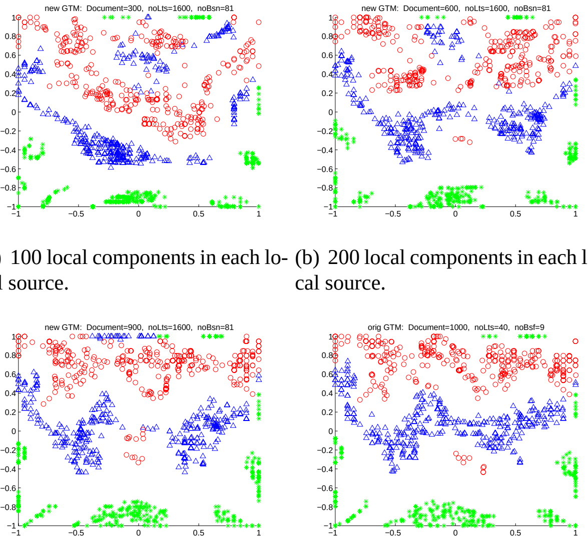
To evaluate the effectiveness of the proposed approach of visualizing distributed data using GTM, experiments were performed based on two synthetic datasets (oil flow data and S-curve data) and one real dataset (WebKB). In each experiment, the dataset was first horizontally partitioned in a random manner into three equal parts as local data sources. Then, global GTMs were to be learned in each experiment for comparison. Both the original GTM learned directly from the original dataset and the new GTM learned from the aggregated local model were tested.

For the experiments on the oil flow and S-curve datasets, 1600 latent lattice points were chosen as the global GTM parameters. For the experiments on the WebKB dataset, 400 latent lattice points were selected instead as the dataset consists of less data items in comparison with the two synthetic ones.

### 3.1 Visualizing Oil Flow Data

The oil flow dataset was originally used in [4] for mimicking the measurements of oil flows mixed with gas and water along multi-phase pipelines. The 12-dimensional data set consists of 1000 instances evenly distributed among three different geometrical configurations – stratified, annular and homogeneous.

In this experiments, 100, 200 and 300 local components were tested for the abstraction of each local data source. The global GTM learned from the local model parameters were compared with the GTM learned directly from the oil flow data. We expected that if each local data item is to be represented by one local Gaussian component (the extreme case), the performance of the proposed approach will be equivalent to that of the original GTM. If less local components are assumed, the visualization results may start to degrade.



(a) 100 local components in each local source.

(b) 200 local components in each local source.

(c) 300 local components in each local source.

(d) The visualization of the original GTM.

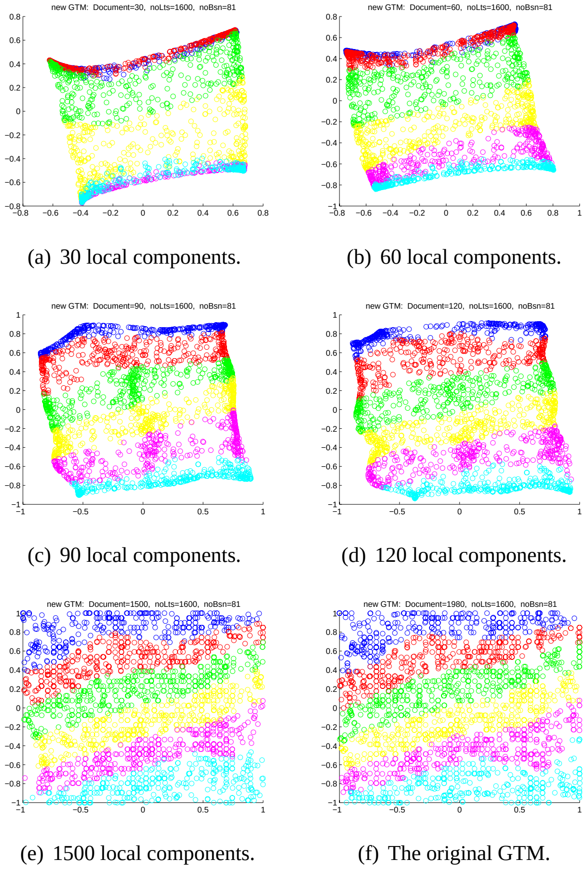
**Figure 5. The visualization of the oil flow data in the latent space using GTMs with 1600 latent variables. The posterior means of the projected data of the three different configurations, namely homogeneous, annular and stratified, are labelled as red circles, blue triangles and green asterisks, respectively. Their posterior modes are all shown as crosses. Three equally weighted distributed local sources are assumed.**

The visualization results obtained are shown in Figure 5. Figure 5(d) reveals the oil flow data manifold obtained via the GTM learned from the original dataset. Figure 5(a-c) show the visualization results obtained using the proposed method but with different numbers of local components, and thus different data granularity levels. It was observed that the visualization map obtained using 300 local components for each source was comparable to that of the original GTM, as shown in Figure 5(d). The visualization result degraded gracefully when the number of local components was dropped to 200 and then to 100. This is consistent to what being anticipated.

## 3.2 Visualizing S-curve Data

S-curve is another commonly used dataset adopted by many nonlinear manifold learning algorithms for testing the algorithms' capability to unfold its intrinsic 2-D plane hidden in a 3-D data space of which the shape likes the alphabet 'S'. For the purpose of visualization, the 2000 data items in the dataset were labelled. We first divided the dataset into six continuous parts along the data instrinsic 2D manifold. They were then labelled as blue, red, green, yellow, magenta and cyan circles respectively. 30, 60, 90, 120 and 1500 local components were chosen for the abstractions of each local source. And the visualization results were shown in Figure 6.

Figure 6(f) reveals the intrinsic manifold unfolded by the original GTM learned directly from the data. Those obtained using the proposed GTM based on different numbers of local components are shown in Figure 6(a-e). The manifold in Figure 6(a) was obtained according to the situation with only 30 local components per source and found to be the worst when compared with the other maps using more local components. In the top region of the map, it can been seen that the blue circles tangled up with the red ones which means that it fails to unfold the top part of the original S-curve data well. A similar situation was observed for the bottom part. In Figure 6(b-d), the aforementioned two unfolded areas, *i.e.* the top and bottom parts, started to be folded up and were finally completely unfolded as the number of local components per source was increased from 30 to 60, 90 and 120. When the number of local components was close to the number of data items, as shown in Figure 6(e), the visualization results was found to be almost equivalent to that of the original one shown in Figure 6(f).



(a) 30 local components.

(b) 60 local components.

(c) 90 local components.

(d) 120 local components.

(e) 1500 local components.

(f) The original GTM.

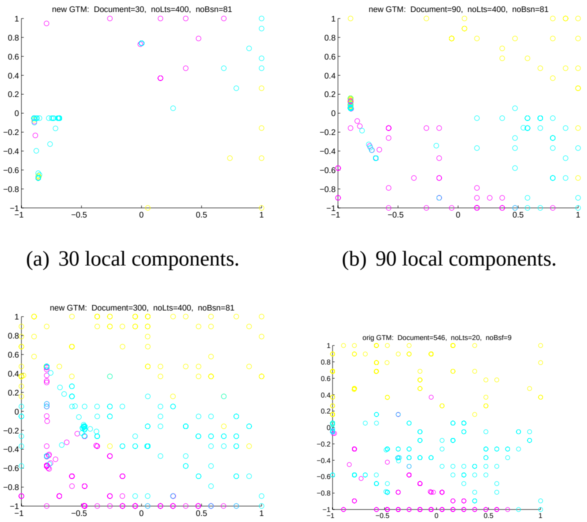
**Figure 6. The visualization of the S-Curve data which was partitioned into three equally weighted distributed local sources for this experiment.**

### 3.2.1 Visualizing WebKB Dataset

The original WebKB dataset contains 8275 university Web pages of 7 pre-defined categories, including course, department, project, faculty, etc. To evaluate the effectiveness of the proposed approach for unfolding the manifold of this real dataset, we prepared a subset from the WebKB with 182 Web pages from 3 categories of WebKB: course, department and faculty, and labelled the subset in yellow, magenta and cyan circles respectively. Some pre-processing steps including the removal of stop words, stemming and merging were performed. Finally, each Web page was represented as a feature vector with the conventional *tf-idf*[5] of a set of globally indexed terms computed as its elements. The dimension of the feature vector used in this experiment was 551, each corresponds to one distinct term.

In this experiment, 30, 90 and 300 local components were chosen as the abstraction of each local source. Then, the manifolds of the WebKB unfolded by the original GTM and the modified GTM were shown in Figure 7. It was observed that when less local components were used, as shown in Figure 7(a), the projected feature vectors were sparsely spread over the map and it was hard to tell how the three categories of data are distributed based on the visualization. When the number of local components increases, Fig-

[5] "*tf*" stands for term frequency which counts the term's occurrence in a document. "*idf*" stands for inverse document frequency which counts the reciprocal of the term's occurrence in the whole document population. *tf-idf* is a product of the two.



(a) 30 local components.

(b) 90 local components.

(c) 300 local components.

(d) Original result.

**Figure 7. The visualization of a subset of the WebKB dataset which was partitioned into three equally weighted distributed local sources for this experiment.**

ure 7(b) shows three separated clusters more clearly, except that some red and blue circles are overlapping withe each other on the left side of the map. Figure 7(c) shows the setting with 300 local components per sources and the result was comparable to that of the original GTM (Figure 7(d)).

## 4 Conclusions and Future Work

In this paper, we propose the use of the model-based approach for visualizing distributed data with the constraint that the distributed local data cannot be shared directly. Gaussian mixture models (GMM) are adopted for local data abstraction and generative topological mapping (GTM) is chosen as the global model for high-dimensional data visualization. A novel EM-like algorithm is proposed for learning the global GTM solely based on the aggregated local GMM. The effectiveness of the proposed method was rigorously evaluated using a number of datasets with promising results. Gracefully degrading global visualization results were obtained as the granularity level of the local data became finer. We believe that the positive results we obtained and the formal steps we used in this paper hints the potential of the proposed method to form a principled way to tackle the mining of highly distributed high-dimensional data in a networked environment with limited bandwidth or high data privacy concern.

As mentioned in the paper, the flexible local abstraction provides a mechanism to control the level of local data granularity. The control can then be based on some quantitative measures indicating individual's privacy concern or bandwidth requirement. This will be related to data privacy management and this work should lay a technical foundation for enabling the related management systems to be developed. One important research issue is to find a formal framework, say based on information theoretic, to quantify data privacy level. In addition, it will also be interesting to see how the global model inaccuracy can be related to the local data uncertainty (say caused by local data privacy concern). Based on the relation, the global and the local servers can "negotiate" in a self-organized manner to achieve the highest global model accuracy with less local data privacy further compromised. Such an active and collaborative data mining problem can easily be seen to be a natural extension of this work, which we will further pursue in the future.

## 5 Acknowledgement

This work has been partially supported by RGC Central Allocation Group Research Grant (HKBU 2/03/C).

## References

[1] D. Agrawal and C. C. Aggarwal. The Design and Quantification of Privacy Preserving Data Mining Algorithms. In *The Twentieth ACM-SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, Santa Barbara, CA, May 2001.

[2] A. S. B. Gilburd and R. Wolff. k-TTP: A New Privacy Model for Large-Scale Distributed Environments. In *The Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Seattle, August 2004.

[3] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[4] C. M. Bishop, M. Svensén, and C. K. I. Williams. GTM: The generative topographic mapping. *Neural Computation*, 10(1):215–235, 1998.

[5] M. Cannataro and D. Talia. The Knowledge Grid. *Communications of the ACM*, 46(1):89–93, January 2003.

[6] R. Chen and S. Krishnamoorthy. A New Algorithm for Learning Parameters of a Bayesian Network from Distributed Data. In *Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM 2002)*, pages 585–588, Maebashi City, Japan, December 2002. IEEE Computer Society.

[7] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[8] H. Dhraief and A. Kemper. Distributed Queries and Query Optimization in Schema-Based P2P-Systems. In *Proceedings of 13th World Wide Web Conference (WWW 2004)*, New York, May 2004.

[9] FangKai-Tai and Z. Yao-Ting. *Generalized Multivariate Analysis*. Springer-Verlag, Berlin, 1990.

[10] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter. *Markov Chain Monte Carlo in Practice*. Chapman and Hall, London, 1996.

[11] V. d. S. J. Tenenbaum and J. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, pages 2319–2323, 2000.

[12] G. J.McLachlan and K. E. Basford. *Mixture Models - Inference and Applications to Clustering*. Marcel Dekker, New York, 1988.

[13] H. Kargupta, B. Park, D. Hershberger, and E. Johnson. Collective Data Mining: A New Perspective Towards Distributed Data Mining. In H. Kargupta and P. Chan, editors, *Advances in Distributed and Parallel Knowledge Discovery*, pages 133–184. MIT/AAAI Press, 2000.

[14] M. Klusch, S. Lodi, and G. L. Moro. Distributed Clustering Based on Sampling Local Density Estimates. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI 2003)*, pages 485–490, Mexico, August 2003.

[15] S. Merugu and J. Ghosh. Privacy-preserving Distributed Clustering using Generative Models. In *The Third IEEE International Conference on Data Mining (ICDM'03)*, Melbourne, FL, November 2003.

[16] A. Prodromidis and P. Chan. Meta-learning in Distributed Data Mining Systems: Issues and Approaches. In H. Kargupta and P. Chan, editors, *Advances of Distributed Data Mining*. MIT/AAAI Press, 2000.

[17] S. Roweis and L. Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, pages 2323–2326, 2000.

[18] D. Tasoulis and M. Vrahatis. Unsupervised Distributed Clustering. In *Proceedings of International Joint Conference on Parallel and Distributed Computing and Networks*, pages 347–351, 2004.

[19] J. Vaidya and C. Clifton. Privacy-Preserving K-Means Clustering over Vertically Partitioned Data. In *The Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Washington, DC, August 2003.

[20] X. Zhang and W. K. Cheung. Learning Global Models Based on Distributed Data Abstractions. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI 2005)*, Edinburgh, August 2005.

[21] X. Zhang, C. Lam, and W. K. Cheung. Mining local data sources for learning global cluster models via local model exchange. *The IEEE Intelligent Informatics Bulletin*, 4(2), 2004.

## A Detailed derivation of the Proposed EM Steps

Suppose there are $L$ local components in the $d$-dimensional data space and $K$ lattice points in the latent space. By approximating $R_{ik}$ with $R_{lk}$, the E-step for GTM is redefined based on $KL$-divergence between the local and global Gaussian components. Using $P_l$ for $P_{local}$ and $P_g$ for $P_{gtm}$, $R_{lk}$ is given as

$$R_{lk} = \frac{\exp\{-D(p_l(t|\theta_l)||p_g(t|z_k, W, \beta))\}}{\sum_{j=1}^{M} \exp\{-D(p_l(t|\theta_l)||p_g(t|z_j, W, \beta))\}}$$

According to the definition of $KL$-divergence, $D(p_l||p_g)$ can be computed as

$$D(p_l(t|\theta_l)||p_g(t|z_k, W, \beta)) = \int_t p_l(t|\theta_l)\ln p_l(t|\theta_l)dt$$

$$-\int_t p_l(t|\theta_l)\ln p_g(t|z_k, W, \beta)dt \qquad (11)$$

By substituting $p_g$ with its definition, the second term of Eq.(11) in vector form can be derived as

$$-\int_t p_l[\ln(\frac{\beta}{2\pi})^{\frac{d}{2}} - \frac{\beta}{2}(t - y(z_k;W))^T(t - y(z_k;W))]dt \quad (12)$$

It is known that the integral of $p_l$ is 1. Thus we only need to compute the second term in Eq.(12). According to [9], the result of this second term can be analytical computed and the second term of Eq.(11) can consequently be given as

$$\ln(\frac{2\pi}{\beta})^{\frac{d}{2}} + \frac{\beta}{2}(tr(\Sigma_l) + (y(z_k;W) - \mu_l)^T(y(z_k;W) - \mu_l))$$

Similarly, the first term of Eq.(11) can be obtained as

$$-\ln((2\pi)^{\frac{d}{2}}|\Sigma_l|^{\frac{1}{2}}) - \frac{d}{2}$$

Finally, substituting the two terms back, we get Eq.(11) as

$$\begin{aligned} &\ln\frac{\beta^{-\frac{d}{2}}}{|\Sigma_l|^{\frac{1}{2}}} + \frac{\beta}{2}tr(\Sigma_l) \\ &+\frac{1}{2}(\beta(y(z_k;W) - \mu_l)^T(y(z_k;W) - \mu_l) - d). \end{aligned}$$

For the M-step, we can obtain

$$\begin{aligned} & \sum_{i=1}^{N}\sum_{k=1}^{M} R_{ik}(W,\beta)\{W\Psi(z_k) - t_i\}\Psi(z_k)^T \\ = & \sum_{k=1}^{M}(\sum_{i=1}^{N} R_{ik}(W,\beta)W\Psi(z_k)\Psi(z_k)^T - \sum_{i=1}^{N} R_{ik}(W,\beta)t_i\Psi(z_k)^T) \\ \approx & \sum_{k=1}^{M}((\sum_{l=1}^{L} R_{lk})W\Psi(z_k)\Psi(z_k)^T - \sum_{l=1}^{L} R_{lk}\mu_l\Psi(z_k)^T) \\ = & \sum_{k=1}^{M}(\sum_{l=1}^{L} R_{lk}W\psi(z_k) - \sum_{l=1}^{L} R_{lk}\mu_l)\psi(z_k)^T \end{aligned}$$

and $\beta$ can be acquired in a similar way.

# A Contourlet-based Method for Writer Identification

Z.Y.HE
Department of Computer Science
Hong Kong Baptist University
Kowloon Tong, Hong Kong
zyhe@comp.hkbu.edu.hk

## Abstract

*Handwriting-based writer identification is a hot research top in the field of pattern recognition. Nowadays, on-line handwriting-based writer identification is steadily growing toward its maturity. On the contrary, off-line handwriting-based writer identification still remains as a challenging problem because writing features only can be extracted from the handwriting image in this situation. As a result, plenty of dynamic writing information, which is very valuable for writer identification, is lost. In this paper, we focus on the writer identification based on off-line Chinese handwriting and present a new contourlet-based GGD (Generalized Gaussian Density) method. Shown in our experiments, this novel method achieves good experiment results.*

## 1 Introduction

Even in such a highly developed society, handwriting has still continued to persist as a main means of communication and recording information in daily life because it is the most nature and easiest way for communication and recording. Given its ubiquity in human transactions, automated writer identification of handwriting has practical significance in document authentication, cheque verification, access control, and etc.

We can classify handwriting-based writer identification in several ways. However, the most straightforward one is to distinguish between on-line and off-line writer identification by input method [1] [3]. The former assumes that a transducer device is connected to the computer, which can convert writing movement into a sequence of signals and then send the information to the computer. Off-line handwriting-based writer identification usually deals with handwriting materials scanned into a computer in two-dimensional image representation. Since information on the time order and dynamics of the writing process which is captured by the transducer device contains many useful writing features of the writer, on-line handwriting-based writer identification, compared with off-line handwriting-based writer identification, is easier to deal with and achieve a higher accuracy. But unfortunately, on-line system is inapplicable in many cases, thus developing techniques on off-line writer identification is an urgent task.

Further, the off-line writer identification can also be divided into two parts: text-dependent and text-independent [1] [3]. Text-dependent methods refer to the study of one or a limited group of characters, so that they require the writers to write the same text. While text-independent approaches look at a feature set whose components describe global statistical features extracted from the entire image of a text [3]. Generally, text-dependent methods have better performances on writer identification, however they are inapplicable in many practical applications because of their strict requirement on same writing content. In this paper, we focuses on the off-line, text-independent writer identification based on handwriting.

## 2 Relative work

Writer identification is a process of confirming a writers identity by comparing some specific attributes of his handwriting with those of all the writers enrolled in a reference database. Commonly, writer identification is regarded as a typical problem of pattern recognition and contains basically 3 steps: pre-processing, feature extraction, feature matching.

Nowadays, writer identification is an active research field, and more and more researchers have touched on this field and some attempts have been presented [2]. For text-independent writer identification, Duverony has reported that the most important variation of the

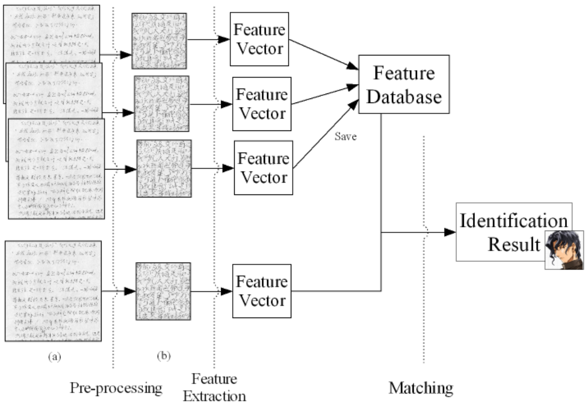


**Figure 1. Flow chart of automatic writer identification, (a)original handwriting image, (b)image after pre-processing**

writers transfer is reflected in the low-frequency band of Fourier spectrum of the handwriting images. And he aslo has designed a hybrid optical-digital image processing system to extract features from Fourier spectra of handwritten text [3]. Similarly, Kuckuck has used Fourier transform techniques to process handwritten text as texture. The feature sets extracted in this study were either composed of a sequence of spectrum mean values per bandwidth, or polynomial fitting coefficients or linear transform of these coefficients [3]. Inspired by the idea of multichannel spatial filtering technique, Said, Tan and Baker propose a texture analysis approach based on multichannel filters [1]. In this method, they regard the handwriting as an image containing some special textures and apply a well-established 2-D Gabor filtering technique to extract feature of such textures. Besides the methods based on frequency-domain analysis, other type approaches are also presented on the text-independent writer identification. In 2000, Schrihari and Cha extract twelve shape features from the handwriting text lines to represent personal handwriting style. The features mainly contain visible characteristics of the handwriting, such as width, slant and height of the main writing zones [4]. Some other papers also adopte multiple features integration to writer identification [4] [5].

# 3 Pre-processing

The origin handwriting image contains characters of different sizes, spaces between text lines and even noises. So before feature extraction, origin image should be processed to facilitate the feature extraction step followed . In our application, we design a pre-processing method which produces texture image for text-independent writer identification and fixed character images for text-dependent writer identification both from original handwriting image. Since some papers have discussed pre-processing [1][2], and this problem is not our focus in this paper, we do not introduce our methods on pre-processing in details.

# 4 Text-independent method

In reference [1], a well-designed 2-D Gabor filters is proposed for text-independent writer identification. Following this paper, reference [2] also applies the same technique on Chinese text-independent writer identification. Both of the two papers show good results are achieved in their experiments. And the academia also widely acknowledges that Gabor method is an effective method on text-independent writer identification. In this paper, to display the advantage of our new algorithm, we will contrast our method with the 2-D Gabor method. While at first, we will introduce the Gabor method briefly.

## 4.1 Gabor algorithm

The Gabor function is the name given to a Gaussian weighted sinusoid. The function is named after Dennis Gabor who used this function in the 1940s. Later, Daugman proposed the function to describe the spatial response of cells in visual stimuli experiments [9].The preprocessing of images by Gabor function is chosen for its biological relevance and technical properties. The Gabor function is of similar shape as the receptive fields of simple cells in the primary visual cortex. It is localized in both space and frequency domains and has the shape of plane waves restricted by a Gaussian function.

The spatial frequency responses of 2-D Gabor functions used in [1] [2] are

$$H_e(u,v) = \frac{[H_1(u,v) + H_2(u,v)]}{2} \quad (1)$$

$$H_o(u,v) = \frac{[H_1(u,v) - H_2(u,v)]}{2j} \quad (2)$$

where $H_e$ and $H_o$ denote the so-called even- and odd- symetric Gabor filter, $j = \sqrt{-1}$ and

$$H_1(u,v) = \exp\{-2\pi^2\sigma^2[(u - f\cos\theta)^2 + (v - f\sin\theta)^2]\}$$

$$H_2(u,v) = \exp\{-2\pi^2\sigma^2[(u + f\cos\theta)^2 + (v + f\cos\theta)^2]\}$$

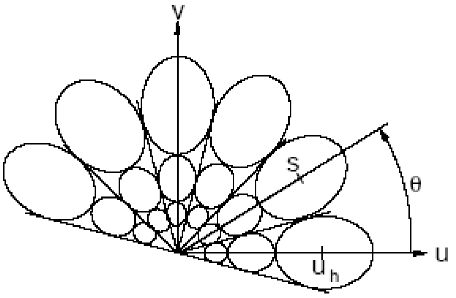


**Figure 2. Tiling of the frequency plane by 2-D Gabor**

Here, $f, \theta, \sigma$ are the spatial frequency, orientation, and space constant of the Gabor envelope, separately. For a given input image, $h_e(x, y)$ and $h_o(x, y)$ will combine to provide different channel outputs of the input image with different $f, \theta$ and $\sigma$.

The mean values(M) and the standard deviation(S) of the channel outputs are used to represent writer global feature for writer identification. If J orientations and L frequencies for each orientation are selected for Gabor filter, a total of J×L features will be obtained from a given handwriting image, as form a feature vector with J×L elements.

After extracting the writing features, Weighted Eucliden Distance(WED) is applied for feature matching.

$$WED(k) = \sum_{i=1}^{N} \frac{(M_i - M_i^k)^2}{\delta_i^k} \tag{3}$$

where $M_i$ denotes the $i$th mean feature of the handwriting image whose writer is unknown, $M_i^k$ and $\delta_i^k$ denote the $i$th mean feature and its standard deviation of the handwriting written by writer K separately, and N denotes the total number of mean values.

### 4.2 Contourlet-based GGD algorithm

Though references [1] [2] both show 2-D Gabor filters is an effective method in handwriting-based writer identification, this method still suffers from some disadvantages as greatly limit its practicability. One of the most serious disadvantages is its intensively computational cost, because the 2-D Gabor filters have to convolute the whole image at each orientation and each frequency.

Contrast to the Gabor filters, 2-D wavelet can decompose the image into subbands with different frequency and orientation. So, we only need to deal with the specified wavelet subbands according the selected values at frequency and orientation. In [6], we have present a new wavelet-based method for writer identification, which improves the identification accuracy and greatly reduces the computational cost as well.

However, wavelet is still not a ideal representation of 2-D image because of its limited ability in capturing directional information, which is very valuable in image analysis and pattern recognition. To address this problem, some multiscale and directional representations have been presented to efficiently capture the image's geometrical structures such as edges or contours. These representation methods include steerable pyramid , brushlets, complex wavelets, and the curvelet transform [7]. Particularly, the curvelet transform, firstly proposed by Candes and Donoho, was shown to achieve essentially optimal approximation in a certain sense for functions in the continuous domain with curved singularities. Inspired by curvelets, Do and Vetterli developed the contourlet transform based on an efficient two-dimensional multiscale and directional filter bank that can deal effectively with images having smooth contours [8]. Contourlets not only possess the main features of wavelets (namely, multiscale and timefrequency localization), but also offer a high degree of directionality and anisotropy. The main difference between contourlets and other multiscale directional systems is that the contourlet transform allows for different and flexible number of directions at each scale, while achieving nearly critical sampling. In addition, the contourlet transform uses iterated filter banks, which makes it computationally efficient [7].

The contourlet transform is implemented via a double filter bank named pyramidal directional filter bank (PDFB), where the Laplacian pyramid is first used to decompose images into multiscale, then followed by a directional filter bank to decompose multiscale image into directional subbands. Fig 3 shows the PDFB as a cascade of a Laplacian pyramid and a directional filter bank at each scale. The directional filter bank is a critically sampled filter bank that can decompose images into any power of twos number of directions. Due to this cascade structure, multiscale and directional decomposition stages in the contourlet transform are independent of each other. One can decompose each scale into any arbitrary power of twos number of directions, and different scales can be decomposed into different numbers of directions [7].

Here, we also assume the contourlet coefficients satisfy the General Gaussian Density (GGD) model.

The Generalized Gaussian Density(GGD) model is

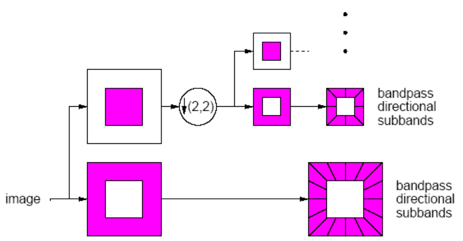


**Figure 3. Pyramidal directional filter bank that implements a discrete contourlet transform, this figure is quoted from [8]**

given as

$$p(x;\alpha,\beta) = \frac{\beta}{2\alpha\Gamma(1/\beta)} \exp^{-1(|x|/\alpha)^{\beta}} \qquad (4)$$

where $\Gamma(\cdot)$ is the Gamma function, i.e.,

$$\Gamma(\cdot) = \int_0^{\infty} \exp^{-t} t^{Z-1} dt, Z > 0.$$

The parameter $\alpha > 0$, called scale parameter, describes the standard deviation and $\beta > 0$, called shape parameter, is inversely proportional to the decreasing rate of the peak. The basic idea of GGD model is to use the GGD model to approximate the statistical distribution of the contourlet coefficients in one contourlet subband and then take the parameter couple $\{\alpha, \beta\}$ of GGD model as the features to represent contourlet subband. There are varied methods to estimate $\alpha$, $\beta$, here we adopt the maximum-likelihood estimator(MLE). The following is how to use MLE for GGD.

The likelihood function of the data vector $x = (x_1, ..., x_L)$ (here we should convert the sub-band image $s$ into a multi-dimensional vector $x$) having independent component can be defined as

$$L(x;\alpha,\beta) = \log \prod_{i=1}^{L} p(x_i;\alpha,\beta) \qquad (5)$$

And using MLE, $\alpha$, $\beta$ can be deduced as the roots of following likelyhood equations [10]:

$$\frac{\partial L(x;\alpha,\beta)}{\partial \alpha} = -\frac{L}{\alpha} + \sum_{1}^{L} \frac{\beta |x_i|^{\beta} a^{-\beta}}{\alpha} \qquad (6)$$

$$\frac{\partial L(x;\alpha,\beta)}{\partial \beta} = -\frac{L}{\beta} + \frac{L\Psi(1/\beta)}{\beta^2} - \sum_{i=1}^{L} (\frac{|x_i|}{\alpha}) \log(\frac{|x_i|}{\alpha}) \qquad (7)$$

where $\Psi(.)$ is the digamma function, i.e. $\Psi(z) = \frac{\Gamma'z}{\Gamma(z)}$. We ignore the deduction process to solve the equations above. For more details, please refer to reference [10].

To replace the typical norm-based distance (e.g. Euclidean distance), we use Kullback-Leibler Distance (KLD) for feature matching. The Kullback-Leibler Distance (KLD) between two sub-bands is as

$$D(p(\cdot;\alpha_1,\beta_1) \| p(\cdot;\alpha_2,\beta_2)) = \log(\frac{\beta_1 \alpha \Gamma(1/\beta_2)}{\beta_2 \alpha_1 \Gamma(1/\beta_1)}) + (\frac{\alpha_1}{\alpha_2})^{\beta_2} \frac{\Gamma((\beta_2+1)/\beta_1)}{\Gamma(1/\beta_1)} - \frac{1}{\beta_1} \qquad (8)$$

and the KLD between two handwriting image is the sum of all the distances across all selected wavelet subbands.

## 5 Experiment

In our experiments, all handwriting are scanned into computer with a resolution of 300 dpi. Then via the pre-processing procedure mentioned in section 3, we produce the handwriting texture images from the original scanned images, as shown in fig 4. Experiments show the size of handwriting texture image should be suitable, since large size image leads to high computational cost and small size image reduces the identification accuracy. In our experiment, we select size as 512 pixels.

20 Chinese handwritings written by 10 persons have been carried out in this experiment, with one training handwriting and one testing handwriting for each person. We produce one handwriting texture image from each handwriting, and thus a total of 20 handwriting texture image are obtained. The training and testing texture image consisting of 64 Chinese characters with size 64 × 64 pixels, as is shown in fig 4. In Gabor method, 4 spatial frequencies are used: 32, 54, 128, 256, and for each spatial frequency, we select 0, 45, 90 and 135 degree as orientations, because both reference [1] and [2] say that the highest accuracy is obtained in this case. In wavelet-based method, we firstly decompose the handwriting image via db4 wavelet transform at 3 levels, and then apply GGD model on the subbands produced in wavelet decomposition expecting for the HH subband at the finest scale. In contourlet-based method, we first decompose the handwriting image into 4 scales using 9-7 filter bank, then each of the 3 fine scales are analyzed into 4 directional subbands, at last we model each subband using GGD model.

A testing handwriting texture image is matched with all training handwriting texture images. Then we sort the matching results in an ascending order to produce a list. And the position of writer of the testing handwriting in the list is regarded as the experiment result to evaluate algorithm accuracy. (For example, if the matching result between the training handwriting and testing handwriting, both of which are written by the same writer, is minimum in the list and consequently occupies the position 1, we say the position of real writer is top 1; in other words, the topper the position of one writer is, the more possibility of being the real writer of the testing handwriting the writer has).The experiment result is in the table 1. The table shows that contourlet-based GGD method is better than wavelet-based GGD method, which is superior to Gabor method.

**Table 1. COMPARISON OF EXPERIMENT RESULTS**

| Method name | Top 1 | Top 2 |
|---|---|---|
| Gabor | 70% | 30% |
| Wavelet-based GGD | 80% | 20% |
| Contourlet-based GGD | 90% | 10% |

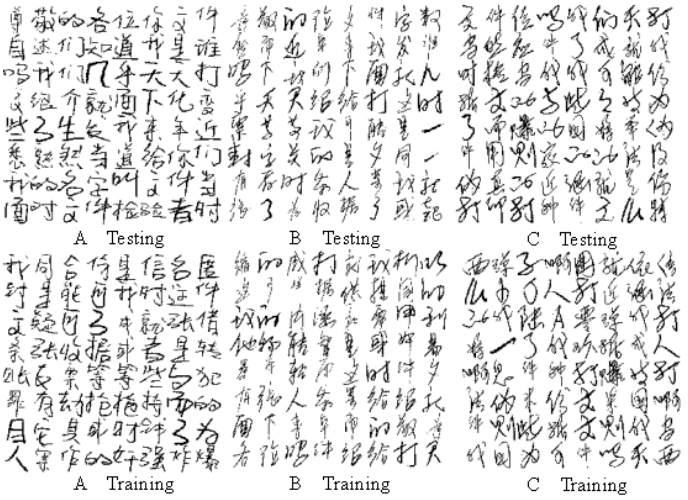


**Figure 4. Some samples of texture image used in our experiments. "A Training" refers to training sample of writer A, and "A Testing" refers to testing sample of writer A.**

## 6 Conclusion

In this paper, we presented a new contourlet-based GGD method on off-line text-independent handwriting identification. Compared with methods via 2-D Gabor filter and wavelet, contourlet-based method achieves a higher accuracy because contourlet transform has capacity to capture comparatively richer directional information, which is important feature to represent the writing style of a handwriting. Because text-independent methods do not care about the writing content, the text-independent methods discussed in this paper are also available for English, Korean, Japanese and Latin Language, etc.

## References

[1] H.E.S.Said, T.N.Tan, K.D.Baker, *Personal identification based on handwriting* Pattern Recognition, vol.33, no.1, pp.149-160, 2000.

[2] Y.Zhu, Y.Wang, T.Tan, *"Biometric personal identification based on handwriting"* 15th International Conference on Pattern Recognition (ICPR), Vol.2, pp.801-804, Barcelona, Spain, Sep 2000.

[3] Rejean Plamondon, Guy Lorette *Automatic signature verification and writer identification-the state of the art* Pattern Recognition, Vol.22, No.2, pp.107-131, 1989.

[4] S.Cha, S.N.Srihari, *Multiple feature integration for writer verification* the Precddings of 7th IWFHR2000, amstredam, Netherland, pp.333-342 Sep 2000.

[5] E.N.Zois, V.anastassopousls, *Fusion of correlated decisions for writer verification* Pattern Recognition, vol.33, no.10, pp.1821-1829, 1999.

[6] Zhenyu He, Bin Fang, Yuan Yan Tang et al,*A Novel Method for Off-line Handwriting-based Writer Identification* Accepted by The 8th International Conference on Documnet Analysis and Recognition(ICDAR2005), Seoul, Korean, Aug 2005.

[7] D. D.-Y. Po, M. N. Do, *Directional multiscale modeling of images using the contourlet transform* IEEE Transactions on Image Processing, to appear.

[8] M. N. Do, M. Vetterli, *The contourlet transform: an efficient directional multiresolution image representation* IEEE Transactions Image on Processing, to appear.

[9] J.G.Daugman, *Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters* J.Opt.Soc.Am. A2, pp.1160-1169, 1985.

[10] M. N. Do and M. Vetterli, *Wavelet-based texture retrieval using generalized Gaussian density and Kullback-Leibler distance* IEEE Transactions on Image Processing, vol. 11, pp. 146-158, Feb. 2002.

# Protecting PCA Or LDA based Face Recognition System With Reed-Solomon Code

Feng Yicheng
Department of Computer Science
Hong Kong Baptist University
Kowloon Tong, Hong Kong
Email:ycfeng@comp.hkbu.edu.hk

## Abstract

Biometric security has been largely regarded and researched within the latest 20 years. However, researchers focus in face recognition have not paid enough attention to the security of face biometric data. We propose a cryptography algorithm to protect face recognition process against attack, which uses the Reed-Solomon code. When the feature vectors are extracted with PCA or LDA algorithms in registration, they are encoded by Reed-Solomon code with a key and stored. The user presents his face image to authentication system to decode the stored data, then gets a new key. A decision is gotten by comparing this two key.

## 1 Introduction

Reliable user authentication has become more and more important. Various methods have been implemented to enhance user authentication security, including passwords, PINs, and biometrics. Comparing with passwords and PINs, biometric is much more convenience and secure. The information contained in biometric can range from several hundred bytes to over a million bytes, quite larger than the information contained in passwords or PINs. And biometrics are physical characters which is highly linked with people. So biometrics-based authentication system enhance higher security than passwords or PINs. On the other hand, biometric is very convenience because it just exits on human's body. Various biometrics-based authentication systems have been widely applied in different domains, including fingerprint, iris, face and so on.

It's very important to protect the security of biometric systems. From last 20 years this issue is largely and adequately considered by different researchers[5, **?**, 2]. However, most of the researchers only concern themselves with security of biometric systems using fingerprint or iris, especially fingerprint. The security of face recognition system receives very little attention. In this paper we propose a method to protect face recognition systems using PCA or LDA methods, with a "fuzzy vault" proposed by A. Juels and M. Sudan[4].

The ”fuzzy vault” scheme proposed by A. Juels and M. Sudan[4] gives a method to embed message into a set to protect it. Suppose Alice wants to protect message *m*. She hide the message *m* in the coefficients of a polynomial *f*, and offers a set *A*. She computes the evaluations of *f* on each elements in set *A*, gets some points whose abscissas are the elements and Y-coordinates are the evaluations. these points constitutes a set *U*. Some chaff points are generated semi-randomly and inserted to the set *U* to build the fuzzy vault *V*. As a result, *V* contains information about the original message *m* in the useful points, with useless chaff points included. If person Bob wants to extract the message from *V*, he has to offer a new set *B* which overlaps enough with *A* to decode the fuzzy vault *V*, using the Reed-Solomon decoding[3]. If *B* overlaps enough with *A*, Bob compares *B* with *V* to choose the useful points out, with a few error points. With these points the polynomial *f* is reconstructed by Reed-Solomon decoding[3].

Then Bob can get the message from coefficients in *f*.

Clancy *et al.*[6] applied this method to fingerprint authentication system and proposed a fingerprint vault scheme. In the scheme a secret key *k* is hide in a polynomial *f*, and the set *A* is constructed with minutiae in the user's fingerprint. The fingerprint of user is scanned for *N* times and *N* images of the fingerprint are gotten. Each feature of each image is proceeded. If the distance between the feature and any feature in set *A* is larger than threshold *T*, it's stored in a set *A*. If the distance between the feature and some feature in set *A* is no larger than *T*, the feature is considered to be overlapped with that one in *A*. Compute the overlapping times of each feature in *A*, and discard the ones whose overlapping times are smaller than threshold *a*. After these operations the set *A* is treated as the locking set, which is used to hide the secret key. The set is encoded with a polynomial *f* whose coefficients hide the secret key, and corresponding points are computed out. Some chaff points are inserted between these points and they all constitute a locked set *B*, with each chaff point at least distance *d* between any real points. The locked set and secret key is stored in a smartcard. In authentication, the user presents his smartcard and scans his fingerprint. The fingerprint is used to decode the locked set and a new secret is computed with Reed-Solomon decoding algorithm. Compare the two secrets and a decision is derived.

In our scheme we apply Clancy's method to face recognition which uses PCA or LDA. When the face image is transformed to a vector using PCA or LDA algorithm, the vector is considered as a set consisting of the dimensions and encoded with the Reed-Solomon algorithm. In authentication the encoded data is transmitted to the system with a secret key. The user also presents his face image and the system uses it to decode the data, as thus gets a new secret key. Compare this two keys and the decision is derived.

# 2 REVIEW

## 2.1 Reed-Solomon codes

The coding algorithm used in our method is a generalized Reed-Solomon code[3]. It's used to encode a polynomial with a set or vector and we simplified it in our scheme. Assume the secret we want to encode is polynomial $f$ and we use a vector $v = (v_1, v_2, v_3, \ldots, v_n)$, the codeword of original polynomial would be $(f(v_1), f(v_2), f(v_3), \ldots, f(v_n))$. If someone want to decode the codeword and resume the polynomial, he should offer another vector $v' = (v'_1, v'_2, v'_3, \ldots, v'_n)$. If $v'$ is nearly the same with $v$, then he could use the Reed-Solomon decoding algorithm to resume the original data $f$. Suppose the degree of polynomial $f$ is $k$, and the vector length is $n$, the vector $v'$ can have at most $(n-k-1)/2$ dimensions different from the corresponding dimensions in $v$.

Because in our scheme some chaff points is inserted, the method should be modified a little. The codeword should not be $(f(v_1), f(v_2), f(v_3), \ldots, f(v_n))$ but a set of points $\{(v_1, f(v_1)), (v_2, f(v_2)), (v_3, f(v_3)), \ldots, (v_n, f(v_n))\}$. Some chaff points such as $(x_1, y_1), (x_2, y_2), \ldots, (x_{ram}, y_{ram})$ are inserted in the set and change it to

$$\{(v_1, f(v_1)), \ldots, (v_n, f(v_n)), (x_1, y_1), \ldots, (x_{ram}, y_{ram})\}$$

which is called encoded set. If someone want to decode the codeword and resume the polynomial, he should presents a vector $v'$ which is nearly the same as $v$, and compare each dimension of $v'$ with x-coordinates of the points in the encoded set to choose out the real points in the set. When the real points have been chosen out with a few fault ones (because $v'$ may be not the same with $v$), the y-coordinates of these points are used for Reed-Solomon decoding, and a new secret is extracted.

## 2.2 System structure design

A most common structure of authentication system is smartcard system. Smartcard structure removes the need of database, so the attacks which aim at the database or the transmission from database to matcher will not work. The card structure design is as bellows:

1. The user's name, which is denoted as $NAME$.
2. The user's privilege and other attributes, denoted as $ATTR$.
3. The encrypted biometric data, which is denoted as $En(data)$.

4. The Signature of the authorization officer, denoted as $SIGN(Hash(NAME, ATTR, k))$, in which $k$ is the secret.

When the user presents his smartcard to the system, the sensor scans his biometric information and extract the feature data. This feature data is used to decode the encrypted biometric data $En(data)$ and a new secret $k'$ is gotten. Compare $SIGN(Hash(NAME, ATTR, k))$ with $SIGN(Hash(NAME, ATTR, k'))$, a decision could be derived. Because the biometric data is encrypted and the secret $k$ is hashed and signed, it certainly enhances the security of system.

### 2.3 Cryptography algorithm

While users' biometric templates are stored in identifier cards, it would not be secure because the storage cards are always insecure. It is required that some protections are added to the stored templates. This would be necessary when the card is stolen or lost. The most common method is cryptography. Davida, *et al.*[2] proposed an error-correcting code utilizing scheme. In enrollment, it transforms the original *K* bits biometric data to *N* bits codewords, whose first $K$ bits remains the same as the original data and the last $N - K$ bits is a check vector, which is derived by multiplying a *K* by *N-K* matrix to the biometric data. The codeword is hashed and stored in the smartcard with the check vector. In authentication, the system use the check vector and the biometric data presented by applicant to construct a new codeword. Hash the new codeword and compare it with the stored one, then a decision will come out. The computation of this scheme is small and it can certainly enhance some security. However, the error correcting ability of this scheme is weak, and it results in some data leakage, because the check vector stored in the smartcard is undefended.

Juels and Wattenberg[**?**] improved Davida *et al.*'s method[2]. They use a error correcting code. In enrollment, they compute the distance between the original data *T* and the codeword *C*. The hash of the codeword *C* is stored with the distance *T-C*. In authentication, the user presents his biometric data *T'*. Compute *T'* minus *T-C* and the result is *C+(T'-T)*. A Bounded distance decoding algorithm is used to transform the result to *C'*. If the distance between *T'* and *T* is smaller than a threshold *d*, then *C'* will equal to *C*. Compare the hash of *C'* and *C* and a decision will be derived. This scheme has some disadvantages. The effect of data various is not clear and the author could not explain how the system will deal with unordered feature representation.

In our scheme we apply and modify Clancy *et al.*'s method[6]. After the system transforms the input face image into a feature vector using PCA or LDA, the vector is encoded with Reed-Solomon codes.

## 3 OUR PROPOSED SCHEME BASED ON CLANCY'S SCHEME

Before encoding, there are some problems which need to be overcome. First, the coding algorithm which we apply just encodes integer vectors, as the feature vector can be any real number. Second, the computation of decoding process highly depends on the length of vector, and the length of feature vector is too large. Third, the variation of face feature vectors are much larger than fingerprint features. In Barral *et al.*'s fingerprint biometric system[1] , system only chooses some points of the fingerprint image and uses the positions and angles of these points, while in PCA or LDA algorithms such as Turk *et al.*'s eigenface algorithm[7] the face feature vector is derived from the whole image. Light or face angle will severely affect the extracted features by sensor. As these reasons, The variation of face feature vector extracted by system is larger than fingerprint feature. In order to overcome these problems, some modifications of Clancy *et al.*'s method[6] should be made.

### 3.1 Variation problem

To overcome the variation problem, we apply the bounded distance decoding algorithm to transform the original face feature vector into a new one, which would be closer to each other. Suppose there are *n* users in the system, each user has *m* face images in database, and the length of each feature vector extracted from each face image is $l$. For user $A$, there are $m$ face feature vectors representing him, which is shown as follows:

$$\begin{array}{cccccc} x_{11} & x_{21} & x_{31} & x_{41} & \dots & x_{l1} \\ x_{12} & x_{22} & x_{32} & x_{42} & \dots & x_{l2} \\ \vdots & & & & & \\ x_{1m} & x_{2m} & x_{3m} & x_{4m} & \dots & x_{lm} \end{array}$$

A composed vector is computed from these $m$ vectors to represent person $A$. In dimension $i$ of the 10 vectors, there are 10 elements $x_{i1}, x_{i2}, x_{i3}, x_{i4}, ...x_{im}$. A minimal range $[a_i, b_i]$ is found which contains exactly $t$ subjects of the 10 elements. It means that in elements $x_{i1}, x_{i2}, x_{i3}, x_{i4}, ...x_{in}$ there are exactly $t$ subjects (represented as $x$) which satisfy the condition $b_i \geq x \geq a_i$. Then, the dimension $i$ of the composed vector would be $a_i/(b_i - a_i)$. this process would be done for each dimension, and a composed vector with length $l$ is computed. This composed vector would be stored in the database as a representation of person $A$. When a new applicant who claims he is person A, an eigenface-vector is computed from his face image, represented as $u_1, u_2, u_3, u_4,$ $u_5, ...u_n$. The dimension $i$ of this eigenface-vector $u_i$ is divided by $b_i - a_i$, in which $[a_i, b_i]$ is the corresponding range for dimension i of person A. After this process a new vector is derived and compared with the stored composed vector.

If $u_i$ belongs to range $[a_i, b_i]$, then $0 \leq [u_i/(b_i - a_i)] - [a_i/(b_i - a_i)] \leq 1$. Using this bounded distance decoding algorithm can help decrease the distance of different features extracted from the same person. Because range $[a_i, b_i]$ contains $t$ subjects of the 10 elements in dimension i, so the probability of $u_i$ belongs to range $[ai, bi]$ would be high when $t$ nears 10. Then, most dimensions of the transformed new vector would near the corresponding dimensions of stored vector, except for a few ones. For each user there are $l$ range numbers:

$$b_1 - a_1, b_2 - a_2, b_3 - a_3, \dots, b_l - a_l;$$

### 3.2 Length problem

In our experiment, we can see the length of the vector should not be larger than 70. To overcome the length problem, we divide each vector into some parts, encode each part separately and then combine all the parts together. Suppose the length of each part is $d$. Because the length of vector may not be multiple of $d$, we have

$$l = qd + r\,(r < d)$$

in which $l$ is the length of the composed vector for person $A$. Suppose the composed vector is

$$c_1, c_2, c_3, \dots, c_d, c_{d+1}, \dots, c_{2d}, \dots, c_{3d}, \dots c_{qd}, \dots, c_{qd+r}$$

A residue vector is used and combined to the composed vector. The dimensions of the residue vector is the head part of the composed vector, and its length is $d - r$, which is shown as follows:

$$c_1, c_2, c_3, \dots, c_{d-r}$$

Combine these two vectors together, and a new complement vector is derived:

$$c_1, c_2, c_3, \dots, c_{qd+r}, c_1, c_2, \dots, c_{d-r}$$

The length of derived vector is $(q + 1)d$, which is just $(q + 1)$ times of $d$. So it could be divided spang into $q + 1$ parts, each part has a length of $d$. Each part of the vector would be encoded with Reed-Solomon algorithm and stored together in the database.

### 3.3 Integer vector problem

To overcome this problem, we transform the complement vector $c$ into two vectors: an integer vector $ci$ and an remained vector $cr$. The integer vector $ci$ has the same length of complement vector and is randomly generated. The remained vector $cr$ is then $c - ci$. In encoding process, the integer part of the complement vector $ci$ is encoded with Reed-Solomon algorithm, and the remained part $cr$ is remained. The remained vector is stored in the database with the encoded integer part. Of cause, the integer vector should be divided into $q + 1$ parts first.

## 4 DESIGN OF OUR SYSTEM

### 4.1 Basic design of our secure system

Taking above measures the three problems are solved, and the finally system design is shown in figure 1.

In enrollment, the sensor extracts each user's face image for $m$ times and $m$ face images are gotten. After face

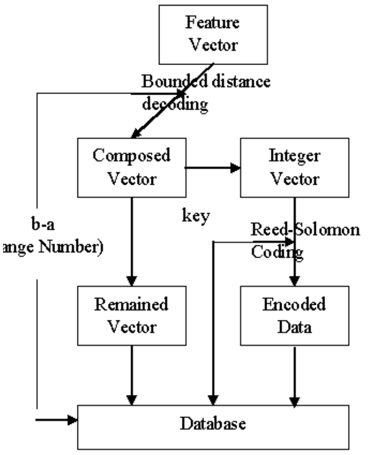


Figure 1: Flow chart for enrollment process.

feature extracting process such as some PCA or LDA algorithm, $m$ feature vectors are gotten. A bounded distance decoding algorithm is used to transform these vectors into one composed vector, and derives $l$ ranges. The composed vector is combined with a residue vector and a complement vector $c$ is derived. Divide $c$ into two parts: the integer vector $ci$ and the remained vector $cr$, and encode $ci$ with the Reed-Solomon algorithm using a secret key. Store the hashed secret key, the $l$ range numbers derived by bounded distance decoding algorithm, encoded integer vector $En(ci)$ and the remained vector $cr$ in smartcard as the representation of the corresponding user, as shown in figure 1.

In authentication process, the applicant presents his smartcard to the system and the sensor extracts the applicant's face image. After face feature extracting process the image is transformed into a feature vector $(v_1, v_2, \ldots, v_l)$. Take out the range numbers stored in the smartcard and use it to transform the feature vector into another one by divide each dimension with the corresponding range number:

$$(v_1/(b_1-a_1), v_2/(b_2-a_2), v_3/(b_3-a_3), \ldots, v_l/(b_l-a_l));$$

After this process, the vector is combined with a residue vector to get the complement vector $v'$, such as the process in enrollment. The remained vector $cr$ is taken up from the smartcard. We compute $v' - cr$. We can see the distance between $v' - cr$ and $ci$ is

$$\begin{aligned} v' - cr - ci = \\ (v_1/(b_1-a_1), v_2/(b_2-a_2), \ldots, v_l/(b_l-a_l)) \\ -(a_1/(b_1-a_1), a_2/(b_2-a_2), \ldots, a_l/(b_l-a_l)) \\ = (v_1 - a_1/(b_1-a_1), \ldots, (v_l-a_l)/(b_l-a_l)) \end{aligned}$$

If $v_i$ belongs to range $[a_i, b_i)$, then $0 \leq (v_i - a_i/(b_i - a_i) < 1$. It means that the distance between the $i$-th dimension of $v' - cr$ and $ci$ is smaller than 1. If all the dimensions satisfy this condition, we can transform $v' - cr$ to $ci$ by a floor function. However, there may be some $v_i$ which do not satisfy the condition. So the vector after floor function would not be completely $ci$, but different in a few dimensions. Assume the vector after floor function on $v' - cr$ is $ci'$, then $ci'$ would be close to $ci$ but a little different. $ci'$ is used to decode the encoded $ci$ taken out from the card using Reed-Solomon decoding algorithm, and a new secret key is derived. If the differences between $ci'$ and $ci$ is enough few, the derived key would be the same as the stored one. Compare hash of the new key and the stored hash of the original one and then a decision is gotten.

## 4.2 A modified design of our secure system

In our method to solve the variation problem, we use a range number to divide the corresponding dimension of original feature vector. If the corresponding dimension belongs to the range, we treat this dimension of the vector as the legitimate one. If most of the dimensions in the feature vector belong to the corresponding ranges, we treat the vector as the legitimate one representing a user. However, because the ranges are computed from just 10 images of one user, it would not be very exact. There is another method to solve the variation problem and we use it to improve my original work. We use eigenface[7] method for example.

While the system uses eigenface algorithm to compute out the feature vectors of each face image, a series of eigenvalues are computed out too, represented as $\lambda_1, \lambda_2, \ldots, \lambda_l$. In original eigenface algorithm, the authentication process uses the nearest neighbor classifier and the threshold of this algorithm is $th$. If vector $c = (c_1, c_2, \ldots, c_l)$ and vector $v_1, v_2, \ldots, v_l$ represent the same person, then we have:

$$(c_1 - v_1)^2 + (c_2 - v_2)^2 + \cdots + (c_l - v_l)^2 \le th^2; \quad (1)$$

Because the weights of each dimension in the feature vector are their eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_l$, then we have:

$$\frac{c_1 - v_1}{\lambda_1} = \frac{c_2 - v_2}{\lambda_2} = \cdots = \frac{c_l - v_l}{\lambda_l} = \alpha \qquad (2)$$

From (1) and (2) we have:

$$(\lambda_1^2 + \lambda_2^2 + \cdots + \lambda_l^2)\alpha^2 \le th^2 \qquad (3)$$

$$|\alpha| \le th/\sqrt{\lambda_1^2 + \lambda_2^2 + \cdots + \lambda_l^2} = \beta \qquad (4)$$

$$|c_i - v_i| = \lambda_i |\alpha| \le \lambda_i \beta (i = 1, 2, \ldots, l) \qquad (5)$$

Then we have rudely estimated the variation of each dimension of the feature vector, which are $\lambda_1\beta, \lambda_2\beta, \ldots, \lambda_n\beta$.

Use these numbers as the range numbers of corresponding dimensions instead of the original ranges, do bounded distance decoding algorithm. The other process of the system remains the same.

# 5 EXPERIMENT RESULTS

## 5.1 Database

The face image database we use is the orl database, which has 40 people and each person 10 images. The size of each image is 92*112.

## 5.2 Parameter choosing

In our experiment we choose the eigenface method proposed by Turk *et al.*[7] as the feature extracting algorithm, and choose the first 100 dimensions of the computed vectors as the feature vectors. The parameter of our algorithm should be chosen carefully. Suppose the degree of the polynomial which hides the secret is $k$, and each divided part of the vector has a length $d$, then the Reed-Solomon code can correct $(d - k - 1)/2$ errors. As description in paragraph 3.1, in our bounded distance decoding algorithm the system computes the range which contains just $t$ dimensions of all 10. It means t/10 elements in the 10 vectors would near the corresponding dimensions after transformation, so in average each vector would has a rate of t/10 dimensions near the corresponding one, and each vector would have a rate of (10-t)/10 errors in average.

So, when the transformed vector is divided into $s + 1$ parts, each part will have $(10 - t)d/10$ errors. Then $k$ should be less than $d - (10 - t)d/5$, which is equal to $(t/5 - 1)d$. Then we have:

$$k < (t/5 - 1)d \qquad (6)$$

$$t > 5 \qquad (7)$$

In our experiment it shows that the divided part length should not be larger than 70, or the computation of decoding process would take much time (much more than one second each time). In our experiment, we choose 2 lengths: 20 and 50. The finite field $F_q$ which is used for the Reed-Solomon code is chosen as $q = 251$. And the parameter $t$ is chosen as 9. The number of chaff points which are used to insert into the coded sets are chosen 40 for each part when the length of divided part is 20 and 100 when the length of divided part is 50.

In experiment for algorithm 2, $d$ is chosen 20 and there is another parameter $s$. This parameter is used for bounded distance decoding. When the original feature vector is transformed into composed vector, it is divided by scale $s$. This operation is in order to adjust the error in transformation. Actually, the eduction from (1) to (5) is approximate because equation (2) is not exactly tenable. We use a scale $s$ to modify the error in the range estimation.

## 5.3 Results and figures

In our experiment for algorithm 1, each face image will play the role of applicant and claim to be any person

| $d$ | $k$ | FAR | FRR | Computation |
|---|---|---|---|---|
| 20 | 11 | 34.96% | 1.00% | - |
| 20 | 12 | 13.01% | 4.50% | - |
| 20 | 13 | 12.96% | 3.25% | - |
| 20 | 14 | 29.66% | 2.75% | - |
| 50 | 36 | 2.88% | 2.75% | 29532.731000 |
| 50 | 36 | 2.97% | 2.50% | 29417.234000 |
| 50 | 37 | 1.08% | 4.50% | 30400.031000 |
| 50 | 37 | 1.21% | 6.50% | 30013.536000 |
| 50 | 38 | 1.08% | 5.00% | 29485.093000 |
| 50 | 38 | 1.09% | 5.75% | 30035.797000 |
| 50 | 38 | 1.15% | 6.00% | 29913.318000 |
| 50 | 40 | 0.33% | 12.00% | 29920.766000 |
| 50 | 40 | 0.35% | 12.00% | - |
| 50 | 42 | 0.05% | 23.00% | - |

Figure 2: The experiment result of algorithm 1.

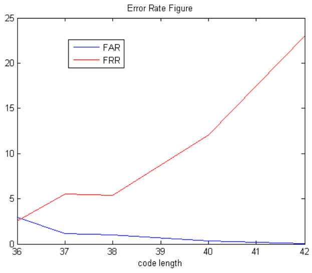


Figure 3: Figure for algorithm1.

| $k$ | $s$ | FAR | FRR | Computation |
|---|---|---|---|---|
| 14 | 1 | 28.15% | 29% | 9651.094000 |
| 15 | 1.5 | 37.99% | 24.75% | 9124.047000 |
| 15 | 2 | 70.51% | 8.00% | 8999.360000 |
| 15 | 2.5 | 89.38% | 2.25% | 8931.453000 |
| 17 | 1.5 | 5.33% | 62.5% | 9065.375000 |
| 17 | 2 | 24.54% | 33.5% | 8993.859000 |
| 17 | 2.5 | 51.87% | 13.25% | 8932.110000 |
| 19 | 2 | 0.84% | 78.5% | 9104.938000 |

Figure 4: The experiment result of algorithm 2.

recorded in database. So the system will be tested for 400*40=16000 times. The result is shown in figure 2,3.

The experiment result in figure 2 and 3 shows that the algorithm 1 works nicely. It can achieve a low error rate of about $2.8\%$ in both FAR and FRR when the parameters of the algorithm is carefully chosen. The computation time of this system is nearly 2 seconds each time. The performance of authentication is not weakened much.

Figure 4 shows that the improved algorithm 2 doesn't work well. The FAR and FRR of this algorithm reach a high rate of $29\%$ when they get trade-off.

### 5.4 Security analysis

The security of this algorithm depends on how hard the attacker can find the right points in the stored set. In our algorithm, there are $q+1$ divided parts to encode and each part has a length $d$. Suppose the number of chaff points inserted to each part is $p$. Because the Reed-Solomon decoding algorithm can correct at most $(d-k-1)/2$ errors, the attacker needs only to pick out $(d-(d-k-1)/2) = (d+k+1)/2$ real points. It means that for each part, when the attacker picks $d$ points out from the whole set which contains $d+p$ points, if there are at least $(d+k+1)/2$ points in them, the attacker can get the information he wants. The probability that the attacker gets the information is

$$pr = \left(\frac{\sum_{x=z}^{d} \binom{d}{x}\binom{p}{d-x}}{\binom{d+p}{d}}\right)^q$$

in which $z$ equals to the minimal integer which is no less than $(d+k+1)/2$. While $d=50$, $k=36$, $p=100$, we have $pr \approx 2^{-160}$. It means that the computation the attacker should pay is $2^{160}$.

## 6 CONCLUSION

It can be seen that the first scheme proposed in this paper works well. The error rate of the original authentication system has not been affected much, and the computation of it is acceptable. The security, which we want to enhance, is largely strengthened. The modified scheme does not work as well as the first one. The error rate of it reaches a severe value. There is still work to do to solve the variation problem.

# References

[1] J. C. C. Barral and D. Naccache. Externalized fingerprint matching. *Proceedings of ICBA 2004*, July 2004.

[2] Y. F. G.I. Davida and B. Matt. On enabling secure applications through off-line biometric identification. *IEEE Symposium on Privacy and Security*, pages 148–157, 1998.

[3] J. Hall. *Notes on Coding Theory*. His Publisher, Erewhon, NC, 2003.

[4] A. Juels and M. Sudan. A fuzzy vault scheme. *Proceedings of IEEE Internation Symposium on Information Theory*, page 408, 2002.

[5] J. H. C. N. K. Ratha and R. M. Bolle. Enhancing security and privacy in biometrics-based authentication systems. *End-to-End Security*, 40, 2001.

[6] N. K. T. C. Clancy and D. J. Lin. Secure smartcard-based fingerprint authentication. *Proceedings of IEEE Internation Symposium on Information Theoryin Proc. ACMSIGMM 2003 Multimedia, Biometrics Methods and Applications Workshop*, pages 45–52, 2003.

[7] M. Turk and A. Pentald. Eigenfaces for recognition. *Journal of Cognitive Neuro-science*, Match 1991.

# Enhancements of Generalized Hough Transform for Improving Object Recognition

Ho Fai Wong
Department of Computer Science
Hong Kong Baptist University, Hong Kong
hfwong@comp.hkbu.edu.hk

## Abstract

*We propose a new method in object recognition based on the Generalized Hough Transform (GHT). Differ from using the tangent of the edge as the feature in the GHT, the new method uses the internal angular information at any 3 points along the edges of the object, with the help of the radii, to recognize the recognizing objects from the model objects. The new method is flexible, efficient and fast in recognition deformed objects. It can assist the traditional GHT for object recognition to enhance the recognition of deformed objects, as well as use it independently.*

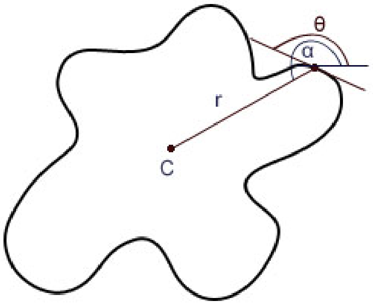


**Figure 1. Arbitrary Shaped-Model**

## 1 Introduction

Generalized Hough Transform (GHT) [2] is an effective object recognition method in recognizing 2-D arbitrary shaped objects. [5] Resembled from standard Hough Transform [3], GHT uses the edges of the arbitrary objects, as well as a reference point (or the centre) to measure the feature of the objects. Retrieves the angular information of any point lying on the edges with respect to the reference point such that these parameters can be used as the features of the object. GHT is robust in scaling and rotation. There are 2 stages in GHT, an offline model preprocessing stage, and an online recognition step. For each preprocessing stage, each point $p$ laid on the edge of the objects are extracted, and its parameters are used as a feature vector. These parameters such as tangent $\theta$, radius $r$ and the gradient of the radius $\alpha$ are used. A lookup table, namely, R-table, is used to store the angular information, indexed by the tangent $\theta$, i.e. for each tuple,

$$\theta\{(r_1, \alpha_1), (r_2, \alpha_2)...\}$$

In the recognition, finds the tangent $\theta$ of each point lying on the edge of the recognizing object, retrieves the entry from the R-table by using the $\theta$ as the indices. After that, prepares a 2-D accumulator array $A$. For each pair of parameter set $(r_e, \alpha_e)$ retrieved from the R-table using the tangent $\theta_e$ as the index, estimates the coordinate of reference point $(x_c, y_c)$ of the recognizing object by using the current coordinates of the edge point $(x_e, y_e)$, the retrieved information from the R-table, as well as the rotating factor $\phi$ and scaling factor $s$, i.e.,

$$\begin{aligned} x' &= r_e \cos(\alpha_e) \\ y' &= r_e \sin(\alpha_e) \end{aligned}$$

$$\begin{aligned} x_c &= x_e - (x' \cos(\phi) - y' \sin(\phi))s && (1) \\ y_c &= y_e - (x' \sin(\phi) + y' \cos(\phi))s && (2) \end{aligned}$$

At the accumulator array, gives a vote at the point corresponding to the estimated reference point of the recognizing object, i.e. $A[y_c][x_c] + +$. Till the end of the iterations, searches the accumulator array to locate the local maxima. If there is no significant local maxima point in the accumulator array, which means the recognizing object is not similar to the preprocessed model.

Although the GHT is robust to scaling and rotating object, it provides little robustness to slightly deformed objects

[6]. For a larger degree of deformation, GHT seems not detecting correctly. The probably reasons are:

- Edge point only provides local angular information with respect to the shape of the objects, it does not provide any global information about the structure of the object.
- For the deformed part of the recognizing object, the edge points are displaced with respect to the model objects, therefore the angular information ($\theta$ and $\alpha$) are altered. As the tangent is used as the index for searching the entry from the R-table, the deviated tangent $\theta$ will retrieve wrong angular information.

To tackle the weakness of the GHT, we proposed a modified algorithm, based on the idea of the original GHT, to provide the global information about the recognizing object [4] [1]. The newly proposed method makes uses of the global skeleton information about the object, rather than the local edge information, providing a macroscopic view of the object. Similar to the traditional GHT, A new lookup table is used to store the global angular information. In the recognition step, retrieves the records from the lookup table, and plot the vote on the accumulator array.

The newly proposed method has the following desirable features:

- Instead of using any point of the edges, it evaluates 3 points and extracts the feature of these points, the global perspective of the object are preserved.
- The enhancement procedures are similar to the traditional GHT, both algorithms train a set of models in preprocessing stages in offline manners. It is not difficult to implement with respect to GHT.
- The accuracy, efficiency can be adjusted according to the user target, therefore, the new method is flexible to recognition.

The new algorithm is desirable in recognizing deformable arbitrary object. Especially when the deformed areas are distributed into different small parts throughout the objects.

The Paper is divided into the following parts: Section 2 discusses the basic idea of the new method, explains the steps in preprocessing and that in recognition. Section 3 shows some experimental results of the newly purposed method, and compares the results with that in traditional GHT.

## 2 Generalized Hough Transform with Curvature Information

The improved Generalized Hough Transform makes use of the curvature nature of the shape and it is desirable for recognizing a deformable arbitrary objects. Instead of using any point laid on the edges of the object, the curvature information of 3 distinct points from the edges are used (see Figure 2). Together with the reference point $C$, a quadrilateral is formed. For any 3 edge points, $P_a, P_b, P_c$, a curvature angle $\phi$ is formed, which stores the curvature information of the arbitrary object. Curvature information shows the structure of the arbitrary object. Rather than using the tangent of any edge point of the object, it is more representative in describing the shape of the object. As the object is deformed, the tangent of the point in the deformed area has been deviated as well. In the recognition of traditional GHT, deviated tangent results in retrieving the inconsistent entries from the R-table, probing to a wrong area of interests.

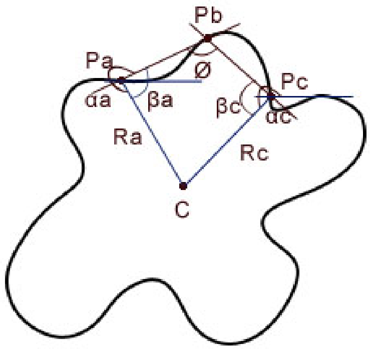


**Figure 2. Model of Using Curvature Information**

Curvature information of deformed areas of the object are altered as well, the degree of deviation depends on the deformability and the precision of the curvature sampling. For the preciseness of the sampling, it depends on the sampling distance of different edge points. The points are closer, the deviations are larger in deformed areas. Deviations are relatively small if these curvature points are distance. To tackle this problem, we can perform some treatments in both training (preprocessing) and testing (recognition) phases.

Similar to traditional GHT, there are 2 phases in proposed method, **Preprocessing** and **Recognition**. In preprocessing, we select a (set of) potential model(s) for training. In recognition, arbitrary objects are fed into the algorithms, and match the objects with the models.

### 2.1 Preprocessing

Similar to the traditional GHT, a lookup table is built for storing the models feature vectors (*I-table*). The steps of preprocessing are shown below (see Figure 2):

1. Selects any 3 points $P_a, P_b, P_c$ lying on the edge. Refers the reference point $C$, retrieves the curvature angle, $\phi$, and the side angles, $\beta_a$ and $\beta_c$, which are intercepted by the lines in the reference point to both point $P_a$ and $P_c$ respectively.
2. Retrieves the radius $R_a$ and $R_c$
3. Using the curvature angle $\phi$ as an index, inserts the feature vector representing these 3 points into the I-table, i.e.
$$\phi_i\{[(\beta_{ai}, r_{ai})(\beta_{ci}, r_{ci})]\}$$
4. Repeats step 1 - 3 with another set of points

Mentioned in above, the curvature information of the deformed areas are easily be altered. To improve the possibilities of the deformed recognizing objects can be successfully hashed the entries from the I-table, we can perform 2 improvement steps: 1) Random sampling and 2) repeatedly sampling with different distance between the points. Randomly sampling reduces the uncertainty of deformation, increases the probability of successfully find the match model. Repeatedly sampling of points with different distance can increase the accuracy in recognition.

### 2.2 Recognition

Recognition performs similar steps that are performed in preprocessing part. At the beginning, prepares a clear 2-D accumulator array for seeking the reference point. Samples with any 3 points, and finds the curvature angle $\phi$. Similar to GHT, after retrieving the $\phi$, uses it as an index to retrieve the records from the I-table, and plots the votes into the accumulator array.

For any 3 points, $P_{ai}, P_{bi}, P_{ci}$, and their curvature angle $\phi_i$, retrieves the records from I-table, i.e.

$$\phi_i\{[(\beta_{a1}, r_{a1})(\beta_{c1}, r_{c1})], ...[(\beta_{aj}, r_{aj})(\beta_{cj}, r_{cj})]\}$$

At the points $P_{ai}(x_{ai}, y_{ai})$ and $P_{ci}(x_{ci}, y_{ci})$, find the distance from the reference points $C_i$ to point $P_{ai}$ and $P_{ci}$ respectively.

For point A,

$$\begin{aligned} x'_{ai} &= r_{aj}\cos(\beta_{aj} + \alpha_a) \\ y'_{ai} &= r_{aj}\sin(\beta_{aj} + \alpha_a) \end{aligned}$$

For point C,

$$\begin{aligned} x'_{ci} &= r_{cj}\cos(\beta_{cj} - \alpha_c) \\ y'_{ci} &= r_{cj}\sin(\beta_{cj} - \alpha_c) \end{aligned}$$

After calculated the distance, we can estimate the reference point:

$$x_C = x_e - (x')s \quad (3)$$
$$y_C = y_e - (y')s \quad (4)$$

where $s$ is a scaling factor of the model and

$$x'_{ai}, x'_{ci} \in x', y'_{ai}, y'_{ci} \in y'$$

To find the reference point of the recognizing object, searches the local maxima at the accumulator array.

For recognizing the deformable arbitrary object, the curvature angle deviated from the original model. One of the method of increasing probabilities is relaxing the searching criteria from the I-table. Despite of exact matches of the curvature angle $\phi$, we introduce some threshold $t$ when searching from I-table, i.e., retrieves the records ranged $\phi \pm t$.

## 3 Experiments

Traditional Generalized Hough Transform and the curvature Hough Transform are compared. Both tests use a same set of training models, as well as recognizing objects.

### 3.1 Experiment 1 - Recognition of Star

A star-shaped object is selected as a training model (see Figure 3), and a set of recognizing objects are used (see Figure 4) by adding some noise to the original star and performed some rotations. In the experiment, we would like to test the robustness of the algorithms by adding some noise to the arbitrary object.

Figure 5 shows the recognizing result by using the traditional GHT and figure 6 shows the accumulator array. The brighter areas mean that the GHT was taking votes on there. As figure 5 shows that there are only 2 deformed stars are successfully recognized by the traditional Generalized Hough Transform. From the vote sheet (Figure 6), it found that only the bottom right star has the significant votes on the estimated reference point. Due to the greater degree of deformation and rotation occurred, the others have highly distributed votes, therefore the traditional GHT estimated their reference points wrongly.

Figure 7 shows the recognizing result by using the curvature information and figure 8 shows the accumulator array. The brighter areas mean that the GHT was taking votes on there. In the recognition stage, we set a threshold such that the curvature angle search the records ranged from $\phi \pm 0.005$ (in radian), the threshold of the curvature angle is small but the improvement of the result is significant. As figure 7 shows that 4 of them are successfully

| Algorithm (GHT) | Recognition Time (sec) | Threshold of indices (in radian) | No of Images Recognized |
|---|---|---|---|
| Traditional | 90.88 | 0 | 2 |
| Curvature | 49.46 | 0.005 | 4 |

**Table 1. Summary of Recognizing Deformed Stars**

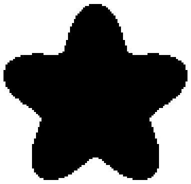


**Figure 3. A Star as an Training Model**

recognized. From the vote sheet (Figure 8),it shows that the votes are concentrated on the particular areas, giving precise estimated reference points. Due the the rotation invariance nature by using the curvature templates, so even the recognizing objects are rotated, but the recognizing results are still good.

### 3.2 Experiment 2 - Recognition of Leaves

In this experiment, we selected some leaves for testing the modified algorithms. Because of larger degree of deformation, we need to increase threshold by 0.005, i.e. the curvature angle retrieve the records ranged from $\phi \pm 0.01$.

As we can see, the shape of the leaves are similar, but if we only evaluate the tangent of the edges, we may not successfully recognizing all the leaves (see Figure 11), therefore, it is desirable to use the modified algorithm to find the leaves. Better recognizing results are found (see Figure 13).

## 4 Summary

Our proposed algorithm can be acted as an auxiliary tool in deformable object recognition. The basic idea is using the curvature nature of the edges, preserving the structural information of the object, to enhance the accuracy of the object recognition. For medical imaging, Hough Transform are widely adopted due to its fast and efficient nature. The new algorithm can enhance the efficiency of the Hough Transform to improve the medical imaging recognition.

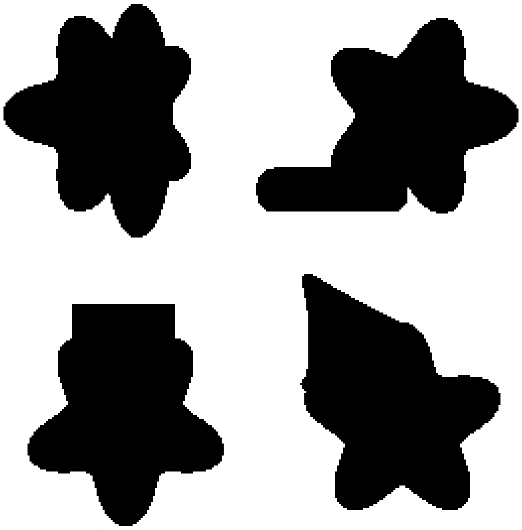


**Figure 4. A Set of Deformed Stars**

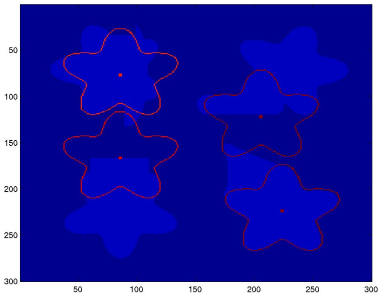


**Figure 5. Traditional GHT Results of Star**

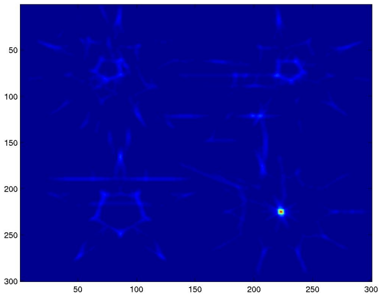


**Figure 6. The Accumulator Array of Using Traditional GHT**

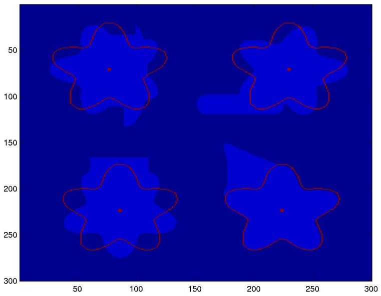


Figure 7. Curvature GHT's Results of Star

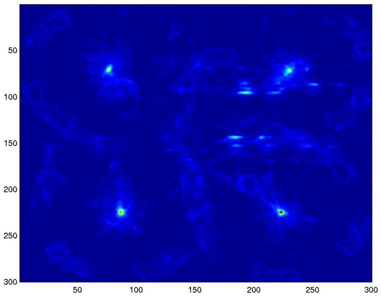


Figure 8. The Accumulator Array of Using Curvature GHT

| Algorithm (GHT) | Recognition Time (sec) | Threshold of indices (in radian) | No of Images Recognized |
|---|---|---|---|
| Traditional | 325.21 | 0 | 3 |
| Curvature | 338.31 | 0.01 | 4 |

Table 2. Summary of Recognizing Deformed Leaves

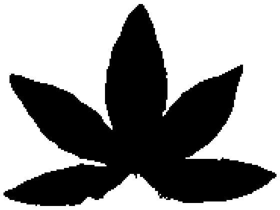


Figure 9. A Leaf as an Training Model

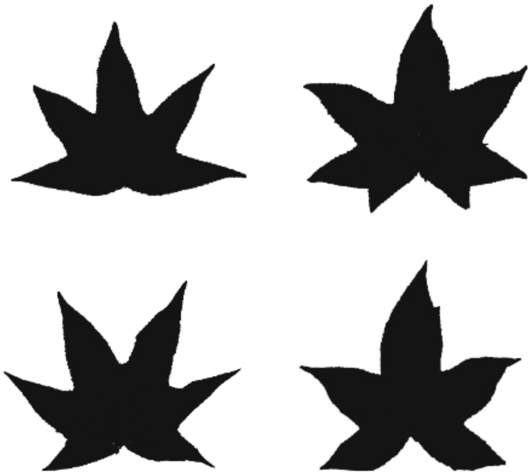


Figure 10. A Set of Leaves

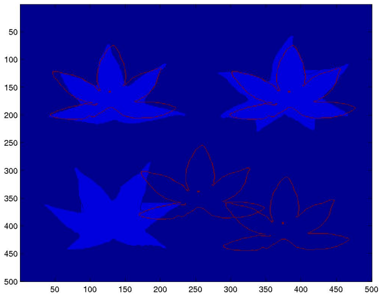


Figure 11. Traditional GHT Results of Leaves

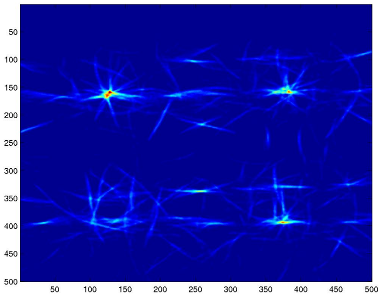


Figure 12. The Accumulator Array of Using Traditional GHT

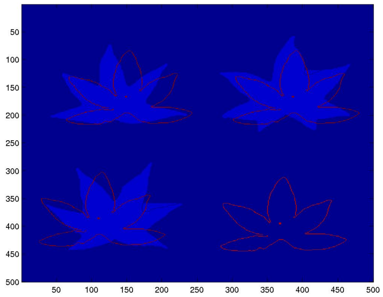


**Figure 13. Curvature GHT's Results of Leaves**

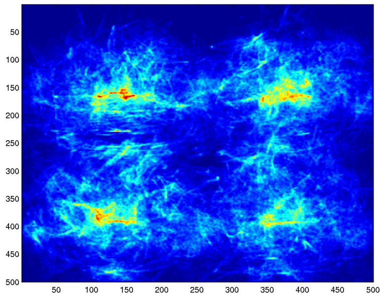


**Figure 14. The Accumulator Array of Using Curvature GHT**

## References

[1] M. M. E. Aguado, A. S. and M. S. Nixon. Invariant characterization of the hough transform for pose estimation of arbitrary shapes. In *In Proceedings of Proceedings of the British Machine Vision Conference BMVC2000*, pages 785–794, Bristol, UK, September 11-14 2000.

[2] D. H. Ballard. Generalizing the Hough transform to detect arbitrary shapes. 13:111–122, 1981.

[3] R. O. Duda and P. E. Hart. Use of the hough transform to detect lines and curves in pictures. *Communications of ACM*, 15(1):11 – 15, January 1972.

[4] N. Guil and E. Zapata. A new invariant scheme for the generalized hough transform. In *IASTED Int'l. Conf. on Signal and Image Processing*, pages 88 – 91, Orlando, FL, November 11 - 14 1996.

[5] Y. C. Hecker and R. M. Bolle. On geometric hashing and the generalized hough transform. *IEEE Transactions on Systems, Man, and Cybernetics*, 24(9):1328 – 1338, September 1994.

[6] K.-M. Lee and W. N. Street. Generalized hough transforms with flexible templates. In *Proceedings of the 2000 International Conference on Artificial Intelligence (IC-AI'2000)*, volume III, pages 1133 – 1139, Las Vegas, NV, USA, June 2000.