

Purdue University

Purdue e-Pubs

Department of Computer Science Technical
Reports

Department of Computer Science

2000

Information Management in a Ubiquitous Global Positioning Environment

Walid G. Aref

Purdue University, aref@cs.purdue.edu

Susanne E. Hambrusch

Purdue University, seh@cs.purdue.edu

Sunil Prabhakar

Purdue University, sunil@cs.purdue.edu

Report Number:

00-006

Aref, Walid G.; Hambrusch, Susanne E.; and Prabhakar, Sunil, "Information Management in a Ubiquitous Global Positioning Environment" (2000). *Department of Computer Science Technical Reports*. Paper 1484. <https://docs.lib.purdue.edu/cstech/1484>

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries. Please contact epubs@purdue.edu for additional information.

**INFORMATION MANAGEMENT IN A UBIQUITOUS
GLOBAL POSITIONING ENVIRONMENT**

**Walid G. Aref
Susanne E. Hambrush
Sunil Prabhakar**

**Department of Computer Sciences
Purdue University
West Lafayette, IN 47907**

**CSD TR #00-006
March 2000**

Information Management in a Ubiquitous Global Positioning Environment

Category: Vision

Walid G. Aref Susanne Hambrusch Sunil Prabhakar

Department of Computer Sciences
Purdue University
West Lafayette, Indiana 47907
E-mail: {aref,seh,sunil}@cs.purdue.edu

March 3, 2000

Abstract

Recent developments in Global Positioning Systems (GPS) and personal locator technologies will enable environments in which (i) objects know their exact location as well as the locations of surrounding objects and (ii) objects are able to report their location and a profile back to regional servers. We envision a new environment where active GPSs are ubiquitously attached to every object in the environment. We term this environment the Ubiquitous Global Positioning Environment (UGPE, for short). Users in an UGPE environment will be able to subscribe to a variety of services. Services range from assisting the navigation of objects by continuously informing users of the surrounding obstacles/objects, to users forming groups of objects that are queried on a continuous basis. This vision paper highlights challenges and opportunities for database and information management research arising from this new environment. UGPE is characterized by a unique combination of ubiquity, large scale real-time spatio-temporal data, spatial and temporal fuzziness, and new types of queries. Understanding query processing and data management in UGPE is fundamental to the success of such environments and it provides a basis for using UGPE in novel applications.

1 Introduction

Global Positioning Systems (GPSs) allow receivers to determine their locations in the globe, up to the meter resolution. Currently, GPS receivers are relatively expensive and their use has been limited to navigation-oriented applications. Recent developments in GPS-based applications [10, 20, 22, 37, 33] will make GPS devices as popular as cellular telephones.

In this vision paper, we address the implications and challenges for future database research arising from an environment in which GPSs are ubiquitously attached to almost every object around us. We assume that objects know their exact locations as well as the locations of surrounding objects, and objects are able to report their locations back to regional servers. We term this environment the Ubiquitous Global Positioning Environment (UGPE). Objects of interest in this new environment may include humans, cars, laptops, eyeglasses, canes, desktops, pets, wild animals, bicycles, and buildings. In UGPE, data sent from the objects to regional servers enables an environment in which objects are not only aware of the locations of surrounding objects, but also of information related to these objects. We refer to the information reported about an object as the object's profile.

UGPE has enormous potential to improve the quality of life. Applications can range from locating lost or stolen objects, to tracing our little kids, completely automating traffic and vehicle navigation systems [11], and automatically annotating objects online in a video or a camera shot. UGPE can help the visually challenged navigate as well as locate, identify, and deal with objects around them. For example, UGPE can enable the visually challenged to identify an approaching taxi cab signal for it to stop.

In our vision, the realization of these applications will be achieved by UGPE providers offering a variety of services that users may subscribe to. The services range from assisting the navigation of objects by continuously informing them of the surrounding obstacles/objects, to users forming groups of objects which are queried on a continuous basis. For many queries in UGPE, the timely delivery of responses is critical. Due to the inherent value of spatio-temporal information, we believe that the development of UGPE is inevitable. Such environments pose several challenges for query processing and data management – problems that the database community is well equipped to address. Approaches for handling spatio-temporal data that have been developed thus far [3, 7, 9, 12, 14, 17, 25, 26, 28, 29, 30, 34] are initial steps in the direction of efficient management of large scale, real-time spatio-temporal data. It is imperative that the database community leverage this head-start and steer the development of these technologies rather than let the technologies develop and then attempt to alter the choices made by others.

In this paper, we focus on the data management and query processing challenges arising from UGPE. These challenges include the management of massive amounts of large-scale real-time spatio-temporal data, support for active queries, caching and prefetching, data aging, dynamic query processing, on-the-fly indexing techniques, adaptive spatial and temporal multi-resolution of the objects to control the volume and flow of data, the use of data replication, and approximate query processing. We also highlight some related challenges in data security and privacy, modeling, and

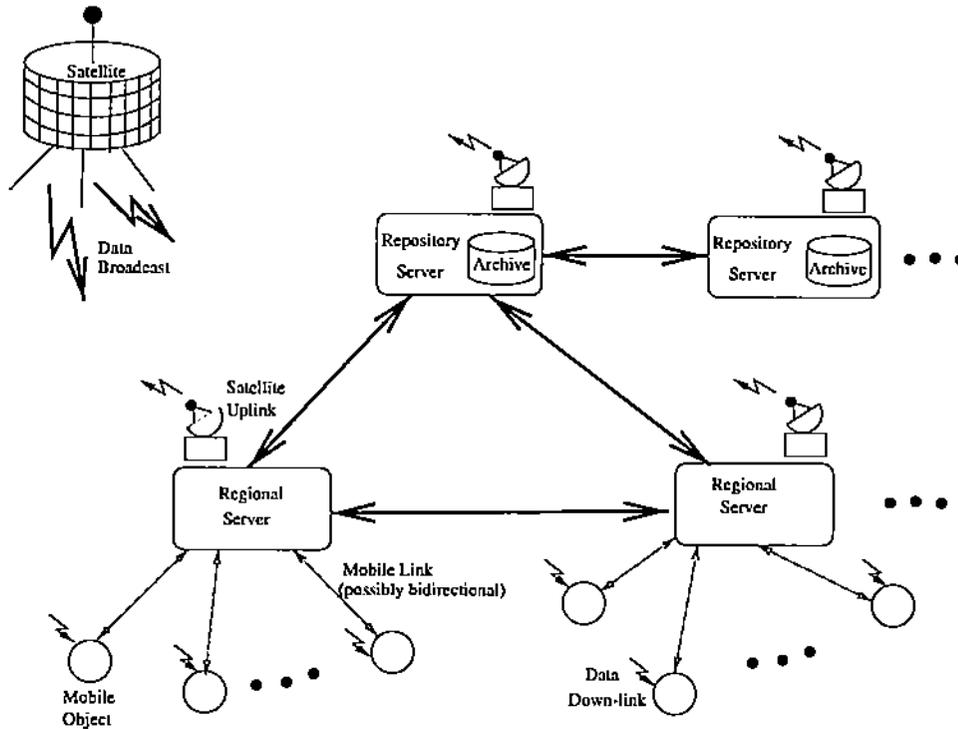


Figure 1: The UGPE Architecture

distributed systems.

The paper proceeds as follows. Section 2 gives an overview the services offered by UGPE. Section 3 discusses the assumed UGPE architecture and addresses the realization and impact of UGPE. Section 4 discusses query types for UGPE, and Section 5 presents research challenges in UGPE. Section 6 addresses research challenges beyond query processing and data management. Section 7 concludes the paper.

2 Services Provided by UGPE

We classify services provided by UGPE into five categories. The five categories are in part characterized by different data management and query processing needs. They also reflect what we consider to represent typical needs of user groups in UGPE. Simple and somewhat restricted forms of queries in these services are already commercially available for applications such as personal locator services [20] and emergency vehicle management [21]. Our proposed service categories are:

- Subscription Service
- Report-only Mode
- Navigation Service
- Ad hoc Query Service

- Group Management Services

In the *subscription service*, an object subscribes to get information about a specified group of objects. For example, one may subscribe to continuously get information about the locations of one's young children. The subscribing object specifies a spatial and temporal resolution. Another form of subscription is to a spatial region, for a certain period of time. In this case, all activities that take place in this region during the specified time interval are reported to the subscribing object.

We assume that each object has control over the possibly multiple profiles it exports. Subscribing objects receive only data they are authorized to. Multi-level profiling is useful in addressing privacy and security issues. Throughout we assume that an object "knows" only about the objects it is authorized to find out about.

Some objects, whether stationary or mobile, may be set to be in a *report-only mode*. In this mode, which can also be called an advertisement service, objects only report their profiles and locations. They do not receive information about other objects in the environment. For example, a person's pet can be set to be in the report-only mode, so that it reports its location. Clearly, such an object does not need to know the locations and identities of other objects in the environment. Other objects may express interest by *subscribing* to get information about some of the objects in report-only mode.

In the *navigation service*, the object is mobile and, as it navigates, it reports its location and receives relevant information about the surrounding objects, whether they are mobile or stationary. The navigation service is considered a basic service provided by UGPE. Given the real-time nature of the navigation service, many data management issues in UGPE need to be addressed in order to efficiently support the navigation service. These include spatio-temporally driven data caching and prefetching techniques, multi-resolution representations of the surrounding environment, and dynamic switching among the various resolutions based on the navigation pattern and the relative importance of the level of detail needed.

In the *ad hoc query service*, objects have the ability to issue ad hoc queries. We distinguish between snapshot and active queries. *Snapshot queries* represent queries which can be answered using already collected data (data is either at a regional server or a repository server). *Active queries* are queries whose responses depend on data progressively accumulated by the servers. An active query may report results at specified intervals of time or it may be a trigger query (i.e., a result is reported if a certain event happens). Active queries capture many of the core applications in UGPE. The effective handling of active queries in UGPE requires new algorithms and data organization techniques.

Group management services provide a collection of services in which a user identifies groups or sub-groups of objects and issues queries involving all objects in this group. For example, a group can represent children on a school trip or a flock of birds. For both examples, the query may be related to monitoring the movement of the objects as a group as well as with respect to each other. Group management queries can be active as well as snapshot queries.

3 Realization of UGPE

GPS devices have been available for some time now, although their use has been largely limited to navigation applications. The devices simply calculate their locations and combine this information with maps stored in the device. Similar devices that record the locations over time have been used to identify migration patterns of wild animals [31]. However in these traditional applications, the GPS devices are passive – they do not exchange any information with other devices or systems. More recently, GPS devices are becoming active entities that transmit and receive information that is used to affect processing. Examples of these new applications include vehicle tracking used [21], identification of closest emergency vehicles in Chicago [21], and Personal Locator Services [20]. Each of these examples represents commercial developments that handle small scale applications. Another example of the importance of location information is the emerging Enhanced 911 (E911) [41] standard. The standard seeks to provide wireless users the same level of emergency 911 support as wireline callers. It relies on wireless service providers calculating the approximate location of the cellular phone user. The availability of UGPE would further enhance the ability of emergency services to respond to a call e.g., using medical history information about the caller. GPS devices also provide extremely accurate timing based upon perfectly synchronized atomic clocks (providing upto 100 nanosecond accuracy). An example of the use of such timing information is in accurately synchronizing log entries for a distributed banking application [21]. Applications such as these, improvements in GPS technology, and reducing cost, augur the advent of UGPE-like environments.

In our vision, UGPE consists of a large number of mobile and stationary objects and a comparatively small number of fixed servers. Figure 1 gives a possible hierarchical architecture of UGPE. GPS satellites provide the objects with their geographical locations. Objects connect directly to regional servers that form the lowest level in this hierarchy and upload their location as well as the profiles they choose to report about themselves. Regional servers handle the incoming data and process time-critical queries. Regional servers communicate with each other, as well as with the high level servers – the repository servers. The repository servers may also be hierarchically organized. Key factors in the design of the system are scalability with respect to large numbers of objects and the efficient execution of queries.

The servers are interconnected by high bandwidth links. However, the mobile links between the regional servers and objects have low bandwidth and a high cost per connection. In order to avoid the bottleneck due to these links, we propose two solutions. For data that is being sent from the objects (i.e. the upload direction), we regulate the amount of data collected (*resolution*), and the rate at which it is sent (*upload frequency*). For query results and other data being sent down to the objects (the download direction), an alternative to the point-to-point mobile links is allowed. Servers can transmit data to a satellite which broadcasts the information over the air to all objects. This approach [1, 42], allows a server to send data to a very large number of objects. The objects receive data by “listening” to the broadcast.

4 Queries in UGPE

Regional servers in UGPE receive profile and location data as well as queries from objects in their geographic region. With the exception of the report-only mode, every subscription category allows a user to issue queries of a spatio-temporal nature. Queries supported in UGPE span various queries on spatio-temporal data. We expect that existing query processing techniques and approaches can be applied for traditional queries like membership, nearest neighbor, range queries, and aggregate queries [4, 6, 18, 24, 36, 40]. However, the nature and characteristics of UGPE will require new solutions and algorithms for active queries and group management services. Since many of the challenges are motivated by the processing needs and different types of active queries, we discuss active queries in this section.

4.1 Active Queries in UGPE

Active queries encompass a broad set of queries for which responses depend on data progressively accumulated. For the UGPE services we propose, active queries are crucial to subscription, ad hoc query, and group management services. We define and motivate four types of active queries which have different algorithms and data processing needs. In our discussion we distinguish between active queries which periodically report results at specified intervals of time and active queries for which a response is triggered by a specified event.

As already stated, objects in UGPE report their location and profile. This information does not always need to be reported continuously. To capture the different needs of objects with respect to when data is collected, we use the term resolution to represent the amount of change allowed for an object before a new data point must be collected. Resolution determines what data points are collected by an object. We use the term upload frequency to refer to the rate at which an object sends messages to the regional server. A more detailed discussion of resolution and upload frequency can be found in Section 5.1. Different active queries have different resolution and upload frequency needs and during the execution of one query these parameters may change. The following discussion on queries includes examples of such situations.

4.1.1 Density Queries

In a density trigger query, the user specifies a region R , a group of objects (by profile features), and a trigger condition C based on an aggregate function. A periodic query is similar (the condition specifies when reporting takes place). A response is triggered when condition C is satisfied. One example of a density query is monitoring the number of people in a building and triggering a response when this number approaches a given upper bound. Another example is monitoring the number of police cars in an area and triggering a response if a subregion contains too few police cars. In order to process density queries, an aggregate function on the region and the objects is progressively evaluated as new data arrives.

For some density queries, not all data in the region needs to be available in order to determine

the response of the query at any given time. However, all data is collected by the regional servers for future processing. As an aggregate function reports results closer to condition C , the regional servers may adjust the spatial and temporal resolutions as well as the upload frequency of all objects in and within the vicinity of region R . This is done to ensure the real-time performance and the accuracy needed.

4.1.2 Boundary Queries

The user specifies a region R , a group of objects, and a condition based on the relationship of individual objects to the region. A response is triggered when an object in the group satisfies this condition. For example, if the region is a playground and the group of objects corresponds to a set of children on the playground, the condition can be such that a response is triggered when a child gets too close to the boundary of the playground.

Boundary queries are quite different from density queries. In a boundary query, individual objects in the set have to be monitored by the regional servers. Two objects relevant to the response of the boundary query may have very different resolutions and upload frequencies. For example, when the position and a bound on the velocity of an object are known (this can be available in the object's profile), resolution and upload frequency may be low. On the other hand, when the object is closer to the boundary, resolution and upload frequency need to be high. Clearly, there will be limitations and restrictions on the resolution and the upload frequency. Hence, we need algorithms which are tailored towards the uncertainty and the fuzziness induced by these limitations. Getting close to the boundary is only one of the conditions a boundary query should be able to handle.

4.1.3 Flock Queries

The group management services provided in UGPE enable queries involving the relationship between objects in a group. Flock queries are one such type of query. The user identifies a collection of objects, referred to as a *flock*, through their profile. Each flock has an attached condition C . This condition can be a function on a relation between the objects in the flock or it can be a function on the interaction of objects in the flock to specified objects outside the flock. The following two examples capture typical scenarios of flock queries.

The flock corresponds to a "herd" and a herd needs to "stay together". Objects are able to make decisions on their own (they are generally viewed as being in a report-only mode). A herd can represent, for example, a herd of sheep, migrating birds, or a group of children. Condition C captures a spatial function on the objects. Examples are recognizing when an object is too far away from the remaining objects in the flock or when the overall configuration of the flock is not within a given specification (e.g., the smallest enclosing rectangle is too large).

A flock can also represent stationary or mobile objects needed for crucial tasks. For example, each object is a cane needed for walking or a pair of reading glasses. The flock query specifies one or more objects outside the flock. This can be a person who needs at least one object in the flock in close vicinity. The condition is set so that a response is generated when there is no object of the

flock within a minimum distance of the person.

4.1.4 Profile-dependent Conic Queries

A conic query is a basic navigational query. The object issuing the query, say object V , is moving and objects in V 's view are of interest. The query differs from what one could consider a generalized range query in that object V can dynamically change its profile as well as the profile of the objects of interest. For example, if V is an individual in a car, then at some point, V may be interested in open gas stations in its view. At some later time, V may be interested in restaurants. The restaurants of interests are determined by V 's profile: V might be interested in fast food places within the next 4 miles, but in gourmet restaurants within 25 miles. Prefetching and caching considerations based on the object's profile and velocity are crucial for the efficient execution of such a query in UGPE.

5 Research Challenges in UGPE

This section addresses research challenges in UGPE. It includes the processing and data handling for active queries, requirements and constraints for new algorithms, the development of on-the-fly indexing techniques, approximate query processing, the impact of data resolution and upload frequency on query responses, and data aging techniques. Many of these challenges are motivated by the processing needs and different types of active queries. The efficient processing of active queries is crucial to the success of UGPE. We discuss issues related to a particular topic before stating challenges.

Responding to active queries in a timely fashion is a must for UGPE. Hence, query processing research in UGPE needs to consider alternative approaches guaranteeing fast responses. Some active queries can generate a response before all relevant data has been received or processed. For example, in a density query with a trigger going off when an aggregate function does not satisfy a lower bound, processing of the query for a particular point in time can stop as soon as the regional server has determined that the lower bound has been exceeded. Another possible performance enhancement is through the development of incremental algorithms; i.e., algorithms which update current solutions instead of re-evaluating them from the underlying data.

Challenge 1: *Development of algorithms supporting the fast processing of active queries.*

5.1 Data Resolution and Upload Frequency

Objects do not need to continuously capture their location and profile data and report it to the server. Depending upon an object's current location, its profile, and currently executing active queries, it may be necessary to capture data only when a "significant change" has taken place. Significant change can be measured spatially (e.g., when the object has moved by at least a certain distance, Δd , from its previously captured location), temporally (e.g., when a certain period of time,

Δt , has elapsed since the last data was captured), when the object determines that its profile has changed sufficiently, or a combination of these. The term resolution describes the amount of change allowed for an object before a new data point must be collected. The resolution determines what data points are collected by an object. Different objects can collect data at different resolutions. By increasing the amount of change allowed, one can reduce the number of data points generated.

Objects need not upload each collected data point immediately to the regional server. Instead, one may allow objects to buffer several data points locally and send them in a single message to the server. Note that this does not affect the amount of data collected and eventually recorded at the servers. We use the term upload frequency to refer to the rate at which an object sends messages to the regional server. The upload frequency can be set as a time interval, a number of data points, or a "significant cumulative change" (over the buffered data points). By reducing the upload frequency, the number of messages sent is reduced, but the information about the object at the regional server becomes less synchronized with the actual data collected at the object.

The choice of resolution and upload frequency affects the accuracy of the information available at the regional server at a given point in time. Consequently, this influences the precision of the queries being processed. To ensure a desired level of query precision, it may be necessary to dynamically adjust the resolution and upload frequency of objects.

The level of resolution and the upload frequency needed depend on the current status of the query as well as the data collected in the past. For some queries, the resolution and upload frequency change for all objects under consideration. For others, the change is only for individual objects. Query processing algorithms need the ability to determine when a change in the resolution or the upload frequency should occur. In addition, they need to be able to adapt to imposed changes in these parameters.

Challenge 2: *The dynamic management of spatial and temporal resolutions and upload frequency for optimal query processing within system limitations*

Regional servers receive large amounts of data of different urgency and processing needs. Most of this data (e.g., that corresponding to cars reporting to a regional server) may not be relevant for answering the currently executing active queries. Such data needs only to be sent to a repository server, where it is stored and possibly used by a later query. The traditional approach of building an index [19] on all the incoming data is impractical in this setting. The time required to analyze the entire data and build the index structure would reduce the gains from using the index. Moreover, the data that arrived a few minutes ago (and which has been incorporated into the index) will soon be irrelevant for the active queries, and should be removed from the index. Regional servers need to process data relevant to active queries in a different way. The distributed nature of the queries and the data requires that arriving data be sent on (possibly in a different form) to all relevant queries. We envision the development of query-driven techniques such as data filters, on-the-fly indexing, and smart agents. These techniques are tailored to the needs of currently executing queries, as opposed to techniques such as indexing which are data-driven.

Challenge 3: *Developing query-driven techniques to reduce the amount of data that active queries have to process based upon properties of the continuously arriving data, and the semantics of ongoing queries. Such query-driven techniques may include a combination of data filters, on-the-fly indexes, and smart agents.*

Some changes in active queries, for example receiving a new active boundary query, have to be incorporated into the management of arriving data as soon as possible. Other changes do not have the same urgency. For example, if an active density query terminates, data related to the region in the active query no longer needs to be handled with the same urgency. However, doing so does not result in incorrect responses. It would be useful to have a metric that predicts the effort involved in changing the handling of the data for a given a set of updates. If the cost generated by the metric is high, updating is delayed. It is done later, possibly together with other updates (critical or non-critical).

Challenge 4: *The dynamic reconfiguration of data management techniques that address Challenge 3, in response to changes in active queries.*

When a regional server contains a number of processors which receive incoming data, the data arriving at one processor may be processed locally and a processor may not be aware of all active queries or all relevant data. Such situations lead to communication requirements. Data that is required by an active query may be shipped to the site where the query is being processed or the query may be migrated to where the data is being received.

Challenge 5: *Distributed query processing techniques for active queries.*

5.2 Data Replication, Caching, and Prefetching

Data replication is one approach used to achieve better performance through increased availability of data by maintaining multiple copies of the data on different sites. The availability of a number of copies results in better performance as well as fault-tolerance. Replicated data requires maintaining consistency among multiple copies if data items can change [8, 13, 15, 32, 38, 39].

In UGPE, collected data does not change in the traditional way. New data is appended and old data ages according to the selected aging process. Some of these queries may be executing at different regional servers. The regional server receiving this data, may choose to cache the replicated data for use by subsequent queries. Similarly, objects may cache some of the data that they receive in response to queries.

For some query types, it is possible to predict with reasonable accuracy, the data that will be accessed in the near future. Such prediction can be used to prefetch data to improve query performance. For example, given the current velocity of an object in the navigation mode, the profile of objects that lie ahead in the object's path can be prefetched and cached at the regional server where the query is executing.

A query issued by a user is received by one of the regional servers. The communication needs between regional servers when processing active queries are a crucial component of the overall performance. For density and boundary queries one can generally make use of locality; i.e., all objects report to the same regional server (or a set of regional servers known in advance). Profile-dependent queries may not satisfy this locality. At the same time, these queries allow us to employ prefetching and caching techniques based on profiles. Flock queries can involve objects reporting to geographically distributed servers not known in advance. In this situation one needs to decide whether data is shipped to servers or processing migrates.

Challenge 6: *The use of caching and prefetching to enable faster processing of time-critical queries. In particular, prefetching based upon navigation queries, simultaneously replication of data over regional servers, and replication and caching based upon the spatio-temporal and profile attributes of data.*

5.3 Exact versus Approximate Query Processing

Exact query processing refers to the traditional approach of generating responses that are exact for the available data. Exact responses are not always needed. In many applications involving large amounts of data (e.g., data warehousing) the primary goal is the fast computation of responses. Approximate answers are acceptable as long as the answer provided is a “good” approximation to the exact answer. Good approximation means that there exists some measure of quality for the generated response, for example, in the form of accuracy guarantees. Approximate query processing has recently received considerable attention [2, 5, 16, 27, 35]. However, many crucial aspects of approximate query processing are still not well understood. Approximate queries are relevant to query processing in UGPE. The distributed nature of the environment, the large data sizes, the need to respond quickly to active queries, and the different resolutions reported may render exact answers to queries infeasible.

In trigger queries, it is possible to allow a certain amount of “false reporting” (i.e., the trigger goes off when it shouldn’t). This captures the spirit of many active queries. For example, in a flock query a trigger goes off with probability close to 1 when the shape of the flock is outside the allowed configuration. The trigger may go off with probability ϵ when the flock is still in an acceptable configuration. Trading accuracy of false reporting (possibly measured by ϵ) against fast responses is an interesting issue to address.

Another meaningful example of an approximate query arises in a velocity query through a region R . Assume the query identifies objects going through region R with velocity higher than s . An approximate query identifies objects which have velocity s and the false reporting is now of the form of missing some objects with velocity s . A tradeoff exists between not reporting objects of smaller velocity for the sake of missing some objects of velocity higher than s .

A common and natural approach to approximate processing is to store and maintain a reduced representation of the data. If data is distributed geographically, this information should be stored at an easily accessible location. A number of reduced representations exists and have been stud-

ied [5]. These include representations obtained through sampling, in the form of summaries (e.g, histograms, aggregations), and hierarchical clusterings. The interaction between reduced representations and data filters is an interesting area to address.

Challenge 7: Developing approximate query processing methods which provide accuracy guarantees given the uncertainty and fuzziness in the underlying data due to the multi-resolution representations in both the spatial and temporal dimensions.

5.4 Data aging

Over time, the total volume of data stored at the repository servers will become very large. A number of applications and services have no need for old data. For example, it may be unimportant where exactly a pet was a year ago. Other applications may need previously collected data, but without critical performance requirements. A third scenario is that data is needed only in summarized form. Data aging addresses how data gets migrated, deleted, and/or summarized, possibly at lower resolutions, after a certain period of time. Depending upon the nature of the data and the application, the following operations can be applied to aging data.

1. Delete. Data is simply removed from the archives associated with the repository servers.
2. Checked out. The system exports data to a user-specified site. Checked-out data may or may not be available for queries issued in UGPE.
3. Summarize. Data is summarized using a data reduction technique.
4. Fading. The spatial and/or temporal resolution of the data increases.

Summarization and fading can be based upon spatio-temporal aspects of the data or the profile. The purging of data has an impact on the accuracy of the query answers computed. For example, the location data for an object over a year can be replaced by a single minimum bounding box that covers all positions occupied by the object during that year. In this case, the effective resolution of the data is increased. Summarization of profile data is more complicated and depends on the semantics of the data. Another approach is to use a combination of summarization and checking out. With this approach, queries can be answered with reduced accuracy using the summary data. If greater accuracy is required, the detailed data can be brought back and queried.

Challenge 8: Design of data aging techniques to effectively deal with large data volume. Develop metrics for the data aging techniques based on the importance, the expected future usage, and the impact on the precision of subsequent queries to help decide on the action to be performed.

5.5 Large Scale Real-Time Spatio-Temporal Data Management in UGPE

In UGPE, the efficient management of massive amounts of real-time spatio-temporal data poses a number of novel and challenging research problems. At the back-end of the UGPE hierarchy, we envision to have a distributed, possibly network-attached, storage management system, with a real-time spatio-temporal data warehouse, to manage the archival and repository of the data. Judicious placement and migration of data between the various levels of the hierarchy can significantly improve query performance. These operations can exploit spatio-temporal and profile properties of the data, as well as the history of queries that have been processed.

Challenge 9: Design of a distributed hierarchical storage manager for managing massive amounts of real-time spatio-temporal data.

6 Beyond Query Processing

New applications using UGPE. Smaller scale UGPE environments; e.g., at the home, office, hospital, a factory form an interesting research issue. A small scale UGPE environment is characterized by having only one server and the services are tailored for specific needs; e.g., for the visually challenged in a house or a rehabilitation hospital setup.

For example, an interesting new application is that of MPEG7-based Automatic Annotation of Video and UGPE-aware Cameras. In the film and video production industry, and with the emergence of the MPEG-7 standard [23], a new standard for content description in video data, film and video producers will have to support content description information along with the captured video. We envision developing tailored Hollywood-style UGPE environments, with UGPE-enabled cameras, where all the objects during the taping and recording of the scenes have active GPSs attached to them. The camera would continuously query UGPE for all the objects within its line of sight. The objects report their evolving profiles as well as their location information, and hence provide a direct way of annotating all the taped video. The profile data associated with UGPE objects will play a more central role in this case as it may reflect, e.g., the narration/dialogues that take place during the shot, or the actors names.

Challenge 10: Data management issues arising from a down-scaling of UGPE as it relates to applying the UGPE technology in local and highly specialized environments characterized by geographical locality and a single server.

Exploiting synchronized global clocks. The signals used by GPS devices to calculate precise locations can also be used to precisely determine time based upon the atomic clocks on board the GPS satellites. Thus using GPS devices, it is possible to obtain highly synchronized clocks for each object in UGPE – in effect providing a single global clock. The availability of such synchronized timing significantly simplifies the design of distributed protocols. Moreover, the relative ordering of arbitrarily distributed events is easily achieved. This is especially useful for reconciling the

information uploaded by each object in UGPE. Without such a clock, it would be very difficult to determine when exactly an object was at the location it reports.

Challenge 11: *How does the availability of a synchronized global clock affect distributed systems? What new applications are enabled?*

Multi-level Profiling and Privacy. An important concern for users that send profile information to servers to allow other users to query is the security and privacy of this information. Users should be able to control access to their profiles. As described earlier, an object can export multiple profiles at different times. It may also be possible for an object to export multiple profiles at the same time – by providing different “views” for users with different authorizations. For example, a person may only expose his/her name to another person who works for the same company.

Challenge 12: *What security mechanisms need to be enforced for UGPE. How can we ensure that unauthorized users are not able to infer information based upon spatio-temporal or profile information?*

UGPE and the Internet. The Internet masks physical location by allowing universal access to the data and computing. Although the physical location of computing devices and data are not inherently significant, physical locations of users and objects, and data related to physical locations are highly relevant to human activities. Environments like UGPE complement the Internet by augmenting the universal accessibility to provide the important spatio-temporal link to information. How best can such information be employed and integrated with the existing Internet infrastructure is an interesting question. For example, devices on the Internet are identified by IP addresses which have at best a very loose notion of physical location. How will the objects in UGPE be identified – can an IP-based approach be applied? For example, by using the IP address of an object’s current regional server plus an identifier to disambiguate between all objects reporting to that server at that time. Should objects in UGPE be identifiable uniquely across all such objects in the world, or should their context be relative to only regional or repository servers?

Challenge 13: *How does UGPE integrate with the Internet?*

As UGPE-like environments are incorporated into the world-wide-web, it will be necessary to support searching and retrieval of spatio-temporal profile data from various servers. Such support could be provided through the development of XML models for describing the data collections, and query languages incorporating the query types applicable to UGPE.

Challenge 14: *Development of XML models and query languages for UGPE data.*

7 Conclusion

In this vision paper, we argued that Ubiquitous Global Positioning Environments (UGPEs) will emerge in near the future. We addressed the implications and research challenges of such environments. Due to the spatio-temporal and real-time nature of UGPE the database community is

in a strong position to leverage existing work on spatio-temporal data to contribute to the success of UGPE. We proposed different service categories for UGPE and defined new queries arising naturally from these service categories. The challenges in UGPE for the database community include active query processing, adaptive spatial and temporal multi-resolution of the objects, the use of data replication, caching and prefetching, data aging, approximate query processing, and multi-level profiling and privacy.

References

- [1] Swarup Acharya, Rafael Alonso, Michael J. Franklin, and Stanley B. Zdonik. Broadcast disks: Data management for asymmetric communications environments. In *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data*, pages 199–210, 22–25 May 1995.
- [2] Swarup Acharya, Phillip B. Gibbons, Viswanath Poosala, and Sridhar Ramaswamy. The aqua approximate query answering system. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMod-99)*, volume 28,2 of *SIGMOD Record*, pages 574–576, June 1–3 1999.
- [3] K. K. Al-Taha, R. T. Snodgrass, and M. D. Soo. Bibliography on spatiotemporal databases. *SIGMOD Record (ACM Special Interest Group on Management of Data)*, 22(1):59–67, March 1993.
- [4] S. Arya. *Nearest Neighbor Searching and Applications*. PhD thesis, University of Maryland, College Park, MD, 1995.
- [5] Daniel Barbará, William DuMouchel, Christos Faloutsos, Peter J. Haas, Joseph M. Hellerstein, Yannis E. Ioannidis, H. V. Jagadish, Theodore Johnson, Raymond T. Ng, Viswanath Poosala, Kenneth A. Ross, and Kenneth C. Sevcik. The New Jersey data reduction report. *IEEE Data Engineering Bulletin: Special Issue on Data Reduction Techniques*, 20(4):3–45, December 1997.
- [6] S. Berchtold, B. Ertl, , D. Keim, H-P. Kriegel, and T. Seidl. Fast nearest neighbor search in high-dimensional spaces. In *Proceedings of the International Conference on Data Engineering (ICDE)*, Orlando, Florida, 1998.
- [7] Michael H. Bohlen, Christian S. Jensen, and Michel O. Scholl, editors. *Spatio-temporal database management: International Workshop STDBM'99, Edinburgh, Scotland, September 10–11, 1999: proceedings*, volume 1678, New York, NY, USA, 1999. Springer-Verlag Inc.
- [8] Yuri Breitbart and Henry F. Korth. Replication and consistency: being lazy helps sometimes. In *PODS '97. Proceedings of the Sixteenth ACM SIG-SIGMOD-SIGART Symposium on Principles of Database Systems, May 12–14, 1997, Tucson, Arizona*, pages 173–184, 1997.

- [9] Sushil Jajodia Claudio Bettini, Xiaoyang Sean Wang. Temporal semantic assumptions and their use in databases. *IEEE Transactions on Knowledge and Data Engineering*, 10(2):277–296, March/April 1998.
- [10] US Wireless Corp. The market potential of the wireless location industry. <http://www.uswcorp.com/USWCMainPages/laby.htm>.
- [11] E. D. Dickmanns, B. Mysliwetz, and T. Christians. An integrated spatio-temporal approach to automatic visual guidance of autonomous vehicles. *IEEE Transactions on Systems, Man, and Cybernetics*, 20(6):1273–1284, November-December 1990.
- [12] M. Erwig, R. H. Gutting, M. Schneider, and M. Vazirgiannis. Abstract and discrete modeling of spatio-temporal data types. In *Proceedings of the 6th International Symposium on Advances in Geographic Information Systems (GIS-98)*, pages 131–136, New York, November 6–7 1998. ACM Press.
- [13] A. W.-C. Fu and D. W.-L. Cheung. A transaction replication scheme for a replicated database with node autonomy. In Jorgeesh Bocca, Matthias Jarke, and Carlo Zaniolo, editors, *20th International Conference on Very Large Data Bases, September 12–15, 1994, Santiago, Chile proceedings*, pages 214–225, Los Altos, CA 94022, USA, 1995. Morgan Kaufmann Publishers.
- [14] Eric N. Hanson, Chris Carnes, Lan Huang, Mohan Konyala, Lloyd Noronha, Sashi Parthasarathy, J. B. Park, and Albert Vernon. Scalable trigger processing. In *Proceedings of the 15th International Conference on Data Engineering, 23-26 March 1999, Sydney, Australia*, pages 266–275. IEEE Computer Society, 1999.
- [15] M. Herlihy. Concurrency and availability as dual properties of replicated atomic data. *Journal of the ACM, JACM*, 37(2):257–278, April 1990.
- [16] P. Indyk and R. Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing (STOC)*, 1998.
- [17] Tsotras V. J., Jensen C. S., and Snodgrass R. T. An extensible notation for spatiotemporal index queries. *SIGMOD Record*, 27(1):47–53, March 1998.
- [18] George Kollios, Dimitrios Gunopulos, and Vassilis J. Tsotras. Nearest neighbor queries in a mobile environment. In *Spatio-Temporal Database Management, International Workshop STDBM'99*, volume 1678 of *Lecture Notes in Computer Science*, pages 119–134, Edinburgh, Scotland, September 1999. Springer.
- [19] George Kollios, Dimitrios Gunopulos, and Vassilis J. Tsotras. On indexing mobile objects. In *Proceedings of the Eighteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, May 31 - June 2, 1999, Philadelphia, Pennsylvania*, pages 261–272. ACM Press, 1999.

- [20] H. Koshima and J. Hoshen. Personal locator services emerge. *IEEE Spectrum*, 37(2):41–48, February 2000.
- [21] Trimble Navigation Ltd. Trimble customer solutions. <http://www.trimble.com/solution/index.htm>, 1999.
- [22] Rand McNally. Streetfinder gps for palm iic connected organizer. <http://www.randmcnally.com/palmIIC/index.html#receiver>.
- [23] International Standards Organization. MPEG-7 context and objectives. <http://drogo.cselt.stet.it/mpeg/standards/mpeg-7/mpeg-7.htm>.
- [24] N. Roussopoulos, S. Kelley, and F. Vincent. Nearest neighbor queries. In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, pages 71–79, San Jose, CA, 1995.
- [25] Betty Salzberg and Vassilis J. Tsotras. Comparison of access methods for time-evolving data. *ACM Computing Surveys*, 31(2):158–221, 1999.
- [26] H. Samet. *Applications of Spatial Data Structures: Computer Graphics, Image Processing, and GIS*. Addison-Wesley, Reading, MA, 1990.
- [27] Y. Saygin and O. Ulusoy. Involving fuzzy concepts in active mobile databases. *Lecture Notes in Computer Science*, 1460:28, 1998.
- [28] M. Schneider, M. Erwig, R. H. Gutting, and M. Vazirgiannis. Spatio-temporal data types : An approach to modeling and querying moving objects in databases. In *Proceedings of the ACM GIS Symposium '98*, 1998.
- [29] Sellis. Research issues in spatio-temporal database systems. In *SSD: Advances in Spatial Databases*. LNCS, Springer-Verlag, 1999.
- [30] A. U. Tansel, J. Clifford, S. Gadia, S. Jajodia, A. Segev, and D. R. Snodgrass. *Temporal Databases*. Series on Database Systems and Applications. Benjamin/Cummings Publishing, Redwood City (CA), USA, 1993.
- [31] Telonics. GPS applications for wildlife – a review. *Telonics Quarterly Review*, 9(1), 1996.
- [32] P. Triantafillou and F. Xiao. Supporting partial data accesses to replicated data. In Ahmed K. Elmagarmid and Erich Neuhold, editors, *Proceedings of the 10th International Conference on Data Engineering*, pages 32–42, Houston, TX, February 1994. IEEE Computer Society Press.
- [33] TruePosition. What is trueposition cellular location system? <http://www.trueposition.com/intro.htm>.
- [34] M. Vazirgiannis, Y. Theodoridis, and T. Sellis. Spatio-temporal composition and indexing for large multimedia applications. *ACM Multimedia Systems Journal*, 1997.

- [35] Susan V. Vrbsky and Jane W. S. Liu. Producing approximate answers to set- and single-valued queries. *The Journal of Systems and Software*, 27(3):243–251, December 1994.
- [36] Roger Weber, Hans-Jörg Schek, and Stephen Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In Ashish Gupta, Oded Shmueli, and Jennifer Widom, editors, *VLDB'98, Proceedings of 24rd International Conference on Very Large Data Bases, August 24-27, 1998, New York City, New York, USA*, pages 194–205. Morgan Kaufmann, 1998.
- [37] Jay Werb and Colin Lanzl. Designing a positioning system for finding things and people indoors. *IEEE Spectrum*, 35(9):71–78, September 1998.
- [38] Ouri Wolfson, Sushil Jajodia, and Yixiu Huang. An adaptive data replication algorithm. *ACM Transactions on Database Systems*, 22(2):255–314, June 1997.
- [39] Ouri Wolfson and Amir Milo. The multicast policy and its relationship to replicated data placement. *ACM Transactions on Database Systems*, 16(1):181–205, March 1991.
- [40] B.-K. Yi, H. V. Jagadish, and C. Faloutsos. Efficient retrieval of similar time sequences under time warping. In *Proceedings of the International Conference on Data Engineering (ICDE)*, pages 201–208, Orlando, Florida, February 1998.
- [41] J. M. Zagami, S. A. Parl, J. J. Bussgang, and K. D. Melillo. Providing universal location services using a wireless e911 location network. *IEEE Communications Magazine*, April 1998.
- [42] Stanley Zdonick, Michael Franklin, Rafael Alonso, and Swarup Acharya. Are “disks in the air” just pie in the sky? In *IEEE Workshop on Mobile Computing Systems and Applications*, December 1994.