ITR: Scalable Location-Aware Monitoring (SLAM) Systems

Hari Balakrishnan, Erik Demaine, Mike Stonebraker, Seth Teller
Laboratory for Computer Science & Department of EECS
Massachusetts Institute of Technology
Cambridge, MA 02139.

November 9, 2001 – 4:05 P.M.

A. Summary

This proposal describes *SLAM*, a scalable network architecture integrating millions of real-world sensors with actuators and distributed software applications. SLAM will enable a broad variety of novel monitoring and control applications including rapid disaster response, scalable crime detection and prevention, facilities maintenance, asset monitoring, and navigation. SLAM solves three problems:

1. Full exploitation of a sensor's data stream requires knowledge of contextual information, particularly location and time.
2. Fine-grained monitoring of millions of assets and facilities requires the physical deployment of sensors in the environment—an intensive and cumbersome manual task.
3. Use of deployed sensors/actuators by distributed software applications requires network infrastructure.

The SLAM architecture has three main components that address these issues:

1. Cricket, a ubiquitous and precise location infrastructure. No current location-sensing technology works everywhere in all places and at all times. Cricket is a novel *multi-sensor* location architecture to solve this problem, using a combination of RF and ultrasound indoors and at building perimeters, and GPS outdoors. Cricket incorporates self-configuration algorithms and energy-efficient protocols for scalability and longevity.

2. An activated environment and efficient activation method. SLAM requires that the subject environment be activated with sensors and actuators. Without special attention, the activation process could become unmanageable due to the complexity of the environment. Therefore SLAM provides *virtual* location-based tagging, typically for immobile objects. The human installer "affixes" virtual tags to physical regions or objects by pointing at them with a Cricket-equipped handheld device, triggering an association of a unique identifier and the tagged entity's location and other attributes in a persistent store. This eases environment activation.

3. A scalable network infrastructure connects sensor information and events to software handlers. The network consists of fixed and mobile *sensor proxies*, physically co-located with the objects and events they monitor, to integrate location, identity, and temporal information to form an event stream. Sensors and their proxies communicate using sensor-specific low-energy communication protocols. Applications are written as *event handlers* distributed across the network. SLAM provides support for dynamically distributing handlers across proxies and compute servers, routing events to handlers, and performing query processing operations.

The proposed SLAM architecture introduces three innovative ideas: ubiquitous, energy-efficient location infrastructure (drawing on ideas from beacon-based location systems, computational geometry, and wireless networking); virtual region and object tagging for environment activation and asset management (drawing on ideas from geometric modeling and database management systems); and distributed proxy-based event and response processing (drawing on ideas from networking and database systems).

Starting with an existing environment (a building, campus, or town), the operational model to put SLAM in place is as follows. First, the location infrastructure is activated. Location beacons are placed in the environment, and a digital representation of the environment is constructed, enabling location inference anywhere within the environment. Second, the environment is activated. Sensors and virtual and physical tags are affixed to objects of interest within the environment (and environment representation). Third, the SLAM network is activated, connecting raw sensor data streams to sensor proxies. The proxies annotate sensor data streams with location and temporal information, and forward them to appropriate handlers via the event-processing network. Handlers produce further events, as well as actions and notifications to be forwarded to actuators or humans.

As a challenging test case, we plan to deploy SLAM on a large university campus with millions of interesting entities. These include many sensors in offices, machine rooms, physical plant, and laboratories to monitor power, temperature, humidity, and pressure; smoke and fire detectors; burglar alarms and physical intrusion detection systems; motion detectors; monitors of leaks, floods, chemicals, and hazardous materials; large-scale theft- and crime-prevention apparatus, and navigation aids. The goal is to monitor the university's physical assets and improve the personal safety of over ten thousand individuals moving in and around thousands of offices, labs, and common spaces in hundreds of buildings.

Our target SLAM system will focus initially on three capabilities at MIT with a variety of interested partners: efficient facilities monitoring and maintenance (with MIT Physical Plant); scalable asset monitoring for inventory, crime prevention and detection (with the MIT Property Office, MIT Campus Police, and MIT Libraries); and navigation assistance, including both personal way-finding and pervasive active signage (with the MIT Schedules Office and the MIT Safety Office).

C. Project Description

This is the PIs' first proposal to NSF on any topic related to this proposal.

1 Motivation and Challenges

SLAM is a scalable location-aware computational network architecture integrating millions of real-world sensors with actuators and software applications. To provide a more specific context, consider a mixed indoor-outdoor architectural space extending over hundreds or thousands of meters, with hundreds of distinct buildings, such as a university campus, factory complex, or town. This proposal envisions instrumenting such an environment with an infrastructure for accurate localization, and a wide variety of sensors to monitor the environment and the things, resources, and events within it at appropriate spatial and temporal granularity (Figure 1).

This section begins with motivating scenarios that assume that a ubiquitous location system is in place, and that any sensor (temperature, motion, light, moisture, hazmat, handheld PDA) can report its sensor reading and approximate location over a passive or active wireless link. This section describes several motivating application scenarios. The remainder of the proposal describes the central challenges in designing, deploying, and using SLAM.

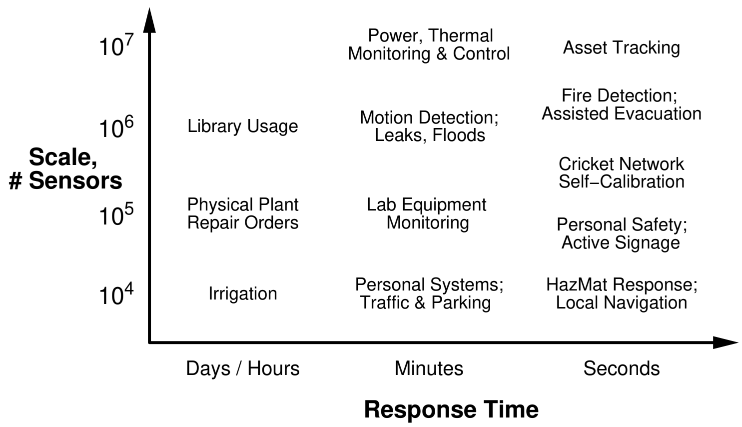

Figure 1: Sensor densities and response rates.

1.1 Safety Scenarios

Personal safety. A student walks through campus, and feels threatened by another individual. The nearest wired "blue phone" to campus police is fifty yards away, but there are wireless panic buttons every five meters on all interior and exterior walls. The student runs, slapping the buttons as he passes. Campus police are alerted, their wall-size campus map showing the time-trail of the student's path. Police respond, carrying handheld devices that display the reports of any nearby motion sensors.

Hazardous material response. An instrumented air filter notices significant deviation from the baseline distribution of particle flow and size distribution, and infers that hazardous materials may have been introduced into a building's air supply. SLAM's event handler reverses the air flow through this filter, increases the flow through all clean intakes, and alerts the Safety Officer for this space.

The Safety Officer orders evacuation. SLAM's event handler uses the building space model (air ducts, offices) to estimate the bloom of the hazmat cloud, and generates a customized evacuation route on the fly, minimizing the contact of evacuees with the hazmat cloud. Active signage advertises this route to all passers-by. The SLAM network, in order to free bandwidth for high-priority traffic from evacuation sensors, motion detectors, etc., quenches all non-critical sensors in the area.

1.2 Asset Tracking Scenarios

Any university or corporate campus has assets: fixed assets, such as electrical and building fixtures; movable (but usually stationary) assets such as furniture, server and desktop computers, and printers; and mobile assets such as vehicles, laptops, and books. Some fixed assets, like landscaping or paint or carpet, are continuous, but can be divided into discrete patches by the imposition of a suitable coordinate grid. Even a small organization may have tens of thousands of such assets; a large campus or company may have tens of millions, depending on the granularity of asset tracking. Our example scenarios envision tracking the location and state of every asset, throughout its life-cycle from perceived need, to acquisition or creation, through installation, use and maintenance, to obsolescence, decommission and disposal.

In the library. Logging book checkouts is a poor gauge of usage, because books and journals are often pulled from the stacks, perused for a few hours in the library, and often mis-shelved. Exploiting each book's unique-ID tag, the librarian decides to shelve journals in more distant stacks according to their frequency of usage. Users seeking a journal type its title as usual, and are led to the issue by their handheld. Mis-shelved items are easily located.

In the garage. The service manager checks usage of all vehicles, and dispatches them in order to equalize wear and tear. Each part's location is tracked and stored, so the parts replacement and repair history for every vehicle is automatically archived. A failure-prediction mechanism, based on usage, nearly eliminates accidents that would otherwise have been caused by bad brakes, steering rods, etc.

In the maintenance garage. The service manager checks usage of all vehicles, and dispatches them in order to equalize wear and tear. Each part's location is tracked and archived, so the parts replacement and repair history for every vehicle is automatically archived. A failure-prediction mechanism, based on usage, nearly eliminates accidents that would otherwise have been caused by worn brakes, tires, and steering rods.

In the parking garage. Every employee has a physical tag on his or her car, and every parking space has an occupancy sensor. The environment is instrumented with location-sensing hardware. Upon arrival, active signage leads the employee to an open space. The employee's handheld infers the parking spot chosen. Upon departure, the handheld leads the employee back to the car. (Note that this scenario does not require that any central database know the location or identity of the car or driver. Security and privacy considerations are discussed in Section 6.)

In the office. People at a meeting collaborate by sharing files temporarily with everyone else in the room. One person brings a laptop, and another brings a projector; they infer their mutual proximity and the presentation is routed from laptop to projector for display without the need for a physical connection. When a participant prints his notes, they are spooled to the physically nearest printer, and the participant's handheld directs him to the printer.

Theft prevention. An LCD projector is reserved from the audio-visual department for use within the same building. A theft-tracking process monitors the location of the projector and discovers it moving outside the building toward the edge of campus. The reserver is notified, and if the motion is not approved, the police are called.

1.3 Physical Plant Maintenance Scenarios

The fire marshal, during a routine inspection, notices some hazardous electrical wiring. She points her handheld device, running the safety inspection application, at the problem. The device infers its own location and orientation, retrieves a geometric model of the building from the network, and displays a few candidate assets (sprinkler system, wiring, fire alarm, blocked exit) for maintenance tagging. The marshal selects the wiring. A virtual position-based "tag" is attached to these wires in the asset database, amending the asset log and calling for maintenance.

A physical plant supervisor spots a broken banister in a stairwell. She points her handheld device, running the maintenance inspection application, at the banister. The device localizes itself in a CAD model of the building, displays a few candidate assets for maintenance tagging (floor tiles, banisters, wall paint). She indicates the banister. A virtual (position-based) tag is attached to the banister in the asset database, amending the asset log and calling for maintenance.

A pump monitor detects unusual behavior, and adds a task to the database with an estimated priority. In extreme circumstances, the device generates active signage events to warn all people currently in the building that it is unsafe, and alerts physical plant. Flow to the pump is reduced if possible.

1.4 Local Navigation Scenarios

A mobility-impaired student in a wheelchair wishes to get from his dorm room to an appointment with student services. He indicates his destination to his handheld, which then leads him to his appointment, avoiding stairs and current construction areas (updated frequently by Physical Plant).

A colloquium is about to start in a building conference room. For a half-hour before the talk starts, active signage displays the talk title, speaker and abstract, and directions to the room. Users with handheld devices get talk notifications and directions as well. Scaling up, a tour group of fifty people arrives on campus, and is given handheld devices with which to take a self-guided tour. Tour-planning software routes them through the campus while keeping them away from other groups and congested hallways. Scaling up again, commencement is scheduled and 5,000 guests are expected. Vehicle traffic, car parking, pedestrian traffic, and seating are monitored, and smart signage directs arriving drivers to available parking and walkers to free seats.

1.5 Fundamental Challenges

The SLAM architecture addresses three principle challenges in realizing these scenarios.

First, full exploitation of a sensor data stream requires knowledge of contextual information, particularly location and time. While the Global Positioning System (GPS) supports reasonably accurate position estimation in many outdoor settings, it fails near and inside most buildings and in urban areas. We propose a novel *multi-sensor* location architecture, *Cricket*, that solves this problem in a scalable manner, by deploying beacons in the environment and constructing a digital representation of the environment. Cricket uses a combination of RF, ultrasound, and GPS-based sensing technologies; it incorporates self-configuration algorithms and energy-efficient protocols for scalability and longevity. Section 3 describes the location infrastructure and associated research challenges, including scalable location inference algorithms, self-configuration algorithms, and energy-efficient beacon scheduling.

Second, fine-grained monitoring of millions of assets and facilities requires the physical deployment of sensors in the environment—an intensive and cumbersome manual task. SLAM addresses this with a scalable *environment activation* method. In addition to lightweight physical sensors and tags affixed to mobile objects of interest, SLAM uses an efficient activation method based on *virtual tags*. Here, a human installer points a Cricket-equipped handheld at a region or object, after which the system associates the object's location with other attributes in persistent storage. Section 4 describes environment activation and associated research challenges.

Third, for remote monitoring and control of sensors and actuators, they must be networked to software entities that can process the data they provide and take appropriate action. This requires progress on three fronts: (1) getting the raw data produced by sensors onto a network, (2) annotating raw sensor data with contextual information such as data origin, location, and time, and (3) passing sensor event streams to software handlers that process this information, generating events for other software agents, actuators, or humans. SLAM's network architecture has two levels. The first level consists of sensor proxies that gather and annotate raw sensor data into location-stamped and time-stamped streams. The second level consists of a scalable event delivery and processing network, interconnecting software handlers with proxies, actuators, and (through displays) interested human parties. Section 5 discusses the SLAM network and associated research challenges.

Our proposal builds on ideas and techniques from other fields including sensor networking, location sensing, computational geometry, geometric modeling, scalable wireless networking, and database management systems. Section 8 summarizes related work.

2 System Architecture

Entities in the environment—sensors (e.g., temperature, pressure, humidity, smoke, chemical) and actuators (e.g., control elements in physical plant and machine rooms), equipment (e.g., monitors, projectors, laptops, etc.), and other items (e.g., books in libraries, personal assets, etc.)—must be monitored in a scalable way, and useful status information about them integrated into application software. Application software can then use the stream of entity status information to take appropriate action for any event.

Figure 2 shows the system context and proposed architecture of SLAM systems. The architecture has three main components: (1) a ubiquitous location system that enables fixed and mobile computing devices to obtain information about their location, (2) an environment activation system that uses physical and virtual object tagging to scalably instrument entities, and (3) a scalable network that co-locates "sensor proxies" with sensors and actuators, and, given events and actions from sensor proxies and other event handlers, generates appropriate responses (state changes, alarms, etc.).

A SLAM system deployed on a large campus or small town will have to manage on the order of 10^7 monitored entities, and when scaled up to a city, on the order of 10^9 entities. The ability to scale well to these domains is the primary design goal for the architecture proposed here.

Most SLAM applications require knowledge of the physical location of sensors and other objects. This requires a location system effective in both indoor and outdoor environments. SLAM's location system, called *Cricket*, combines standard GPS receivers with custom RF and ultrasonic beacons and listeners to infer information about physical location (in GPS coordinates), the logical space (e.g., which part of a room), and orientation relative to a coordinate system. For scaling, Cricket combines active self-configuring beacons, which are easy to deploy on room ceilings, with passive listeners, which receive signals from beacons and process them to provide information to applications via a powerful API. Section 3 describes the research issues in designing a ubiquitous location system.

Cricket enables applications to determine the location of any listener. However, it is not practical to attach physical listeners to tens of millions of objects. We propose instead a scalable *environment activation* system, in which the environment's entities of interest are instrumented to provide useful information using the idea of *tagging*. Tagging allows unique identifiers to be associated with real-world sensors and objects in a lightweight manner.

Tags may be *physical* or *virtual*. Physical tags are small, inexpensive, RF-ID stamps [25] or barcodes attached to mobile or movable entities. Virtual tags are a new idea: they are like physical tags, but have no physical manifestation.

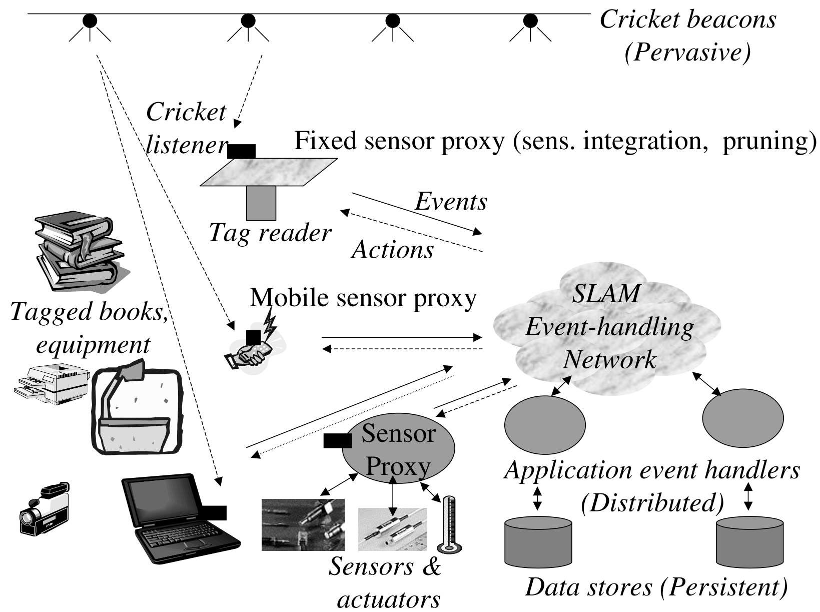

Figure 2: System context and architecture for SLAM systems. Entities in the environment such as sensors/actuators, equipment, books, fixtures, etc. are *tagged* with lightweight physical or virtual tags. *Sensor proxies* are implemented on static or mobile computers with location sensors, tag readers and other sensor-specific hardware/software. Proxies integrate raw sensor information with contextual information such as location and time, prune the sensor stream and quench sensors as necessary, address the resulting event stream to a named destination, and pass the data stream on to the SLAM event processing network. This network schedules registered events and dispatches them to various named destinations in priority order. Applications are implemented as asynchronous event handlers that process events and actions dispatched to them, and may store summaries and log information in persistent storage.

Instead they exist as attributes of the object's representation in a persistent data store. Section 4 describes how physical and virtual tags can be applied efficiently by a human installer.

The third component of SLAM is the scalable network for integration of data and event streams between sensors and software applications. The two components of the SLAM network are sensor proxies and a scalable event-handling infrastructure. *Sensor proxies* are pervasively-deployed computational elements that interact with real-world entities, integrate raw sensor data with location and time information, and provide the software interface to monitor and actuate real-world entities. The *event-handling infrastructure* allows applications to be viewed as *event handlers* that process input data streams (e.g., from sensors) into output data streams (e.g., requests to actuators, notifications of critical events to humans, or streamlined data to other event handlers).

In more detail, *sensor proxies* are small distributed processors with attached Cricket listeners. They integrate raw information from multiple sensors, and control and route the event data streams to the event handling infrastructure. One example of a sensor proxy is a tag reader with an RF-ID reader and Cricket listener, programmed to periodically determine which (if any) of a set of objects with specified IDs is in the proxy's vicinity. Upon discovering an absent or non-responsive object, the proxy sends a timestamped event message addressed to some named destination reporting the absence. When deployed within spaces, or at boundaries between spaces (e.g., at the door leading out of a building), such sensor proxies can effectively detect and track theft. For instance, if a laptop were to "walk out" of an office, then the building, the sensor proxy would detect that and send an event to a laptop-tracking event handler. The handler could correlate this information with the current location of the laptop's owner, or check to see if the laptop was registered as removable from the space or building. An inconsistent response would cause the proxy to notify the laptop's owner, and perhaps campus security.

Sensor proxies may be mobile, in particular implemented on handheld devices carried by people. For example, a campus maintenance supervisor on a trip across campus may point her handheld at various sensors and physical things (e.g., objects requiring attention) and cause information about them to be sent to some event handler or database.

A maintenance worker can later query this database to determine where repairs are needed. Section 4 describes the challenges in sensor activation in detail.

We advocate an *event-driven* data processing model for SLAM, in which applications are designed as *event handlers* that react to streams by generating output streams. The result of this design philosophy is a "process flow" structure of computations, interconnected via arcs that signify the types of event streams that flow between them. The SLAM network provides several process flows for popular primitive functions (system failure alarms, theft detection, etc.). It also provides a design-time GUI to facilitate the construction of large SLAM process flows, and a GUI that facilitates debugging and performance monitoring. Lastly, SLAM provides an efficient, distributed run-time system to efficiently dispatch and execute handlers. The primary challenge in the design of the SLAM network is dealing with a large number of asynchronous and heterogeneous streams.

Some process flows may require storage of historical values. For example, a process flow responsible for logging and reporting the monthly duty cycle of a projector will require 30 days of past location history. Such historical information must be kept in back-end databases. We anticipate that a process-flow diagram compiler will generate appropriate storage directives to a database system and the run time calls necessary to access data.

3 Location Infrastructure

Most applications in SLAM require location information. The first requirement of SLAM's location system is that it work everywhere and at all times. No current location system is capable of this. To provide ubiquitous coverage indoors, in urban areas around around tall buildings, and outdoors, we propose a *multi-sensor* approach to location determination.

The second requirement of a location system is that it scale well. There are several aspects to scalability, including good wireless bandwidth scaling of the location infrastructure and the ability to scalably locate millions of entities. We propose an architecture based on *active beacons* deployed in spaces of interest, combined with *passive receivers* to infer location.

The third requirement is that the location infrastructure be easy to deploy and maintain. We propose to develop a set of geometric design, placement, and *self-configuration algorithms* to ease deployment and maintenance burden.

SLAM's location system, called *Cricket*, provides three types of location information.

1. Position, expressed as (x, y, z) coordinates in a standard (GPS) coordinate system. Cricket transforms the local coordinates provided by any specific location sensing mechanism to standard GPS coordinates.
2. Spatial location, expressed as a semantically-meaningful space identifier. Many applications don't care about specific coordinates, but only about which space some device is in, such as a room number or type (office, conference, lounge, kitchen).
3. Orientation (i.e., compass heading). Without orientation, one's location is just a dot on a map. Providing orientation enables users to "point" at objects in the world, and way-finding applications to "lead" users through the world. Orientation also plays an important role in virtual object tagging.

3.1 Multi-Sensor Location

Cricket incorporates information from a variety of sources, extending our efforts on small-scale indoor location systems [43]. Cricket will combine a variety of location sensors, including GPS for outdoor operation [35], coupled RF and ultrasound for fine-grained indoor operation [43], and proximity information from cellular base stations and 802.11 wireless access points [7]. Finally, Cricket will combine multiple ultrasonic receivers indoors [44], and a magnetic compass outdoors, to infer orientation.

The Cricket location infrastructure consists of *active beacons* and *passive listeners*. Beacons self-configure, propagating GPS coordinates from outdoors to indoors to establish a single, pervasive coordinate system, and broadcasting location information. Listeners infer their own positions by triangulating distance estimates from several beacons. Listeners attach to host devices (sensors, handhelds, actuators) via an RS-232 interface.

3.1.1 Location Inference Algorithms

Each Cricket beacon autonomously broadcasts information on a short-range RF signal. To reduce collisions, beacons transmit according to a randomized schedule. Concurrent with the RF transmission each beacon sends an ultrasonic (US) pulse. Any listener receiving the RF signal and (a short time later) the US pulse can infer its distance from the beacon by multiplying the difference in arrival times by the speed of sound.

Three problems complicate this ideal situation in practice. First, a listener receiving a reflected US pulse overestimates its distance to the originating beacon. Second, pulses from different beacons can overlap in time, causing the listener to wrongly associate the RF from one beacon with the ultrasonic signal from another. Third, the speed of sound is not constant, but varies with air pressure (which depends on temperature, weather, altitude, etc.).

These factors require *inference algorithms* at the listener to distinguish true samples from spurious ones. Our current solutions to this problem use simple statistics, but do not scale well to more than a small number of beacons.

> **Scalable inference.** *Given a set of beacons transmitting on an unknown randomized schedule, obtain a time series of distance samples. Develop an online algorithm to obtain best distance estimates to each beacon within a configurable time period T.*

Two observations apply in the design of an efficient algorithm for this problem. First, reflections cause the statistics of observed samples to be multi-modal. Second, there is a trade-off between the value of T and the precision of the estimate; the bigger T, the better the estimate in general, as long as the device does not move.

Location estimates arriving from different sensors (RF+US and GPS) must be combined. However, the error in each estimate is unknown. We plan to experiment with sensor fusion algorithms to achieve reliable end-to-end location capability from multiple uncertain position estimates.

Different location technologies have different coordinate systems. For instance, a coupled RF + ultrasound technology deployed in a building will in general have a different native origin and orientation than GPS. However, for applications to work seamlessly across multiple technologies, having a unified coordinate space is important. We plan to propagate GPS information from outdoors to indoors, to establish GPS coordinates pervasively through building interiors and other areas where GPS fails.

> **Coordinate space fusion.** *Consider a network of beacons, each of which knows its approximate locations in one or more coordinate spaces, and only some of which can communicate. Design a distributed matching algorithm to find rigid transforms that unify the location systems to a single coordinate space, in such a way that is robust to errors and transient outliers.*

3.1.2 Energy-efficiency

Beacons will typically not have wired power, so will have to run for months or years on power from batteries or small solar cells. Thus beacons must be energy-efficient. This leads to an intriguing distributed scheduling problem, since beacons do not need to be on and broadcasting at all times. Indeed, as the number of beacons in a vicinity increases, the expected frequency at which each beacon must broadcast decreases.

> **Energy Scheduling.** *Design a distributed constraint scheduling algorithm that determines, for each beacon, when it should be awake and broadcasting. The constraint is that any mobile device in any position should be able to hear the necessary number of devices within a given time bound T. Thus several beacons can time-share the burden of coverage. One key issue in this problem is whether beacons can share a central clock.*

A central theme in our proposed solutions to these problems is to use randomization and proximity information to avoid synchrony in time gaps which would lead to poor time coverage. For example, for the simple situation in which beacons are uniformly distributed throughout the space, a reasonable approach to scheduling is that each beacon broadcasts every $\Theta(T/d)$ time units, where d is the average density of beacons heard by a mobile device. The problem becomes significantly more complex with varying densities of beacons and ranges through variable media (air vs. walls).

3.2 Location Infrastructure Activation

The previous section addressed run-time issues in Cricket. This section discusses *activating* the location infrastructure with minimal human effort. Beacons must be placed in a way that demarcates the meaningful spaces in an environment, and that ensures that any listener will hear enough beacons to determine its own position. Once placed, each beacon must be configured with the position and space it inhabits.

3.2.1 Beacon Placement

A strength of the Cricket architecture is that beacons can be placed on ceilings and walls without precise calibration or wiring, but the placement is not completely arbitrary. If two parts of a room must be distinguished, then beacons should be placed at roughly equal distances from the desired boundary, such that a listener anywhere can use the closest beacon to infer which space it is in.

To deduce its *position*, a listener requires distance estimates from three or more beacons [44]. This at least three beacons must be visible to, and in range of, every possible listener location. Deducing *orientation* is harder: at least one of the beacons must be at an angle of 45° or less from the listener (which uses several closely-spaced receivers in an asymmetrical configuration [44]).

Scalable Beacon Placement. *Design an efficient beacon placement algorithm that positions the fewest possible beacons in a building with specified floorplan or 3D model, so that an arbitrarily placed mobile device is within appropriate range R of enough beacons subject to the above constraints.*

The placement problem is closely related to *guarding and visibility* in computational geometry [9, 51]. In particular, much is known about the simple version of scalable beacon placement in which $R = \infty$ and only one beacon must be seen from any location [31, 40]. We propose to build on these results to solve the more general problem. In particular, we will analyze the computational complexity of the problem, design simple algorithms with good worst-case bounds, and design approximation algorithms that guarantee good performance in all situations (not just the worst case).

For practical use, the following harder problem needs to be solved.

Incremental Placement. *Design an incremental placement algorithm for positioning additional beacons to improve coverage beyond an existing partial placement.*

We will develop a greedy algorithm that, given an existing beacon constellation, computes the positioning of $k \geq 1$ additional beacons that will yield the greatest coverage benefit. We expect that a useful tool to solve this problem is the farthest-point Voronoi diagram [42]. We envision the algorithm running on-line on the handhelds of a human installer, giving him/her guidance as to where to place the next beacon.

3.2.2 Automating Beacon Configuration

Two pieces of information must be configured into each beacon: its (x, y, z) coordinates in some coordinate system, and the space the beacon is in (e.g., building and room number). Manual configuration is infeasible even for a few hundred beacons. Therefore we plan to develop *self-calibrating* beacon networks, robust to incremental insertion, deletion, and motion of beacons.

Our self-configuration strategy is to bootstrap a rigid, global coordinate system solely from inter-beacon distances. For example, given four beacons that are pairwise within range to communicate, the measured distances encode a rigid tetrahedron, from which we can compute a local coordinate frame common to all four beacons. This frame is determined up to an unknown absolute translation and rotation; GPS coordinates of any three beacons suffice to pin these remaining degrees of freedom. In the general setting, there are thousands of beacons, most within range of only a constant number of other beacons, and only a small fraction of which know their GPS locations. We can represent the beacon network as an undirected graph, and propagate local and global coordinate information to arrive at a single, globally consistent position estimate for each beacon [3, 4].

This problem is fundamentally connected to *rigidity theory*, an area of combinatorial geometry [18, 19]. Specifically, a beacon connectivity graph and measured distances along each edge determine a unique coordinate frame precisely if that metric graph has a *unique geometric realization* in $\mathbb{R}^3$. Normally it suffices for the graph to be *rigid* or even *infinitesimally rigid*, both of which are well-studied concepts [17]). We believe that many algorithmic tools from rigidity theory can be adapted to our setting.

Beacon-centric self-calibration. *Design a distributed constraint-satisfaction algorithm that runs across all beacons, detecting and exploiting redundant coverage to improve end-to-end accuracy.*

The system should also be robust to movement of the beacons. To achieve this, each beacon will periodically infer its own location from that of its neighbors. Gross inconsistency will trigger a local or global re-execution of the self-configuration phase.

It is impossible to have a fully self-configuring mechanism for spaces, since they require semantic knowledge of how people demarcate and name spatial regions. However, we envision a method in which a human moves through a space, informing each beacon of which space it is in. A simple graphical or voice interface on the handheld enables the installer to indicate the identity, boundaries, and adjacencies of each space by interaction with or physical motion of the handheld.

3.3 Scaling and Privacy Considerations

Cricket's use of active beacons and passive listeners scales well: each beacon need interact with a constant number of other beacons within range, and an unlimited number of listeners can operate in the same space at the same time. This is in contrast to affixing to each object an active transponder which periodically transmits its ID to a centralized location [32]. While this approach enables tracking, it does not scale well, nor does it protect user privacy.

Section 4 describes SLAM's virtual tagging to scale to large numbers of entities without requiring that every entity have a Cricket listener attached.

Our proposed architecture enables applications to preserve *user privacy*. This issue is addressed in detail in Section 6.

4 Activated Environments

Section 3 described "location activation": the process of placing position beacons into a physical environment in order to enable space and location inference by a handheld mobile listener, and constructing a functional data representation of the spaces and adjacencies within the environment. Analogously, this section describes "environment activation": first, the placement of physical and virtual identity tags into the environment and environment representation respectively; and second the deployment of fixed and mobile proxy sensors, physically co-located with the objects and events they are to monitor, which integrate identity, location, and temporal information to form an event stream for processing by the SLAM network (Section 5).

Just as a human activator delineated physical spaces to construct the environment model, we envision the activator moving through the environment, activating sensors and objects of interest within each space. Each activation involves affixing a virtual "tag" to fixed objects, and a physical tag to moveable objects.

4.1 Virtual Tags

Virtual tags are unique IDs attached to immobile objects. The tag is virtual because it does not require the physical application of tags to objects. The human installer moves through the space, affixing virtual tags to each object of interest, and indicating each object's type to the handheld device. Each tag includes a unique identifier, which SLAM thereafter persistently associates with the object and its location. (Note that virtual tags can never be lost forever, as they can be reacquired simply by making a spatial query in the vicinity of the fixed object.) Indoors, for example, the activator could tag power outlets, data jacks, lighting fixtures and switches, air vents, water sources and drains, etc. Outdoors, the activator could tag sewer drains, lighting poles, emergency phones, signage, etc.

Virtual tagging requires a fine-grained location infrastructure that provides reasonably accurate orientation information. We propose to use the Cricket compass facility for this.

4.2 Physical Tags

A physical tag is small and cheap, and encodes a unique bar-coded digital ID (e.g., a 64-bit number). It can reply with its ID [25] to an RF query. The activator would apply physical ID tags to printers, laptops, phones, lab equipment, etc., informing the database of the object's existence, type, and location. Note that environment activation can be done "lazily," that is, in order of importance and necessity, and can be done in parallel by any number of installers, so long as they assign globally unique identifiers.

Activation populates the environment model with objects that are already in place: workstations, printers and the like. Physical tags could also be attributed by type and applied to objects as they are put in place; as they are received at the building's loading dock; or even at each object's point of manufacture.

4.3 Readers

One goal of SLAM is scaling to millions of entities, each with an identity and location. It is not feasible at present to attach location-sensing hardware to *every* entity of interest in a large campus, but we are still interested in where they are located with high precision. For this purpose we define *sensor proxies*, run on fixed or mobile location-enabled devices, which annotate incoming data streams with location information, and forward the annotated streams. Section 5.1 discusses this in detail.

One important class of sensor proxy is a *reader*, which combines location sensing with the ability to read RF-IDs or optical bar-codes. This achieves scalable deployment in a two-level hierarchy. At the lowest level, every entity has a cheap (less than US $0.10) and tiny RF-ID tag or bar code. At the next level, a reader is responsible for keeping track of entities in any given space. Because each reader has Cricket's location-sensing hardware attached, it is possible to determine the location of any tagged object, at any time, in a scalable manner.

Readers can be deployed in both fixed and mobile configurations. Fixed readers can perform continuous monitoring functions, by repeatedly querying objects, and raising an alarm if an object fails to respond or is detected in an illegal location. Mobile readers can be used to "promote" dumb (RF-ID only) objects to first-class location-tagged objects: the object provides its ID, and the reader annotates it with location (using Cricket) and the object's characteristics (from the database). This simple method of attaching lightweight tags combined with tag readers instrumented with sophisticated location-sensing hardware allows a scalable system to be achieved in practice.

4.4 Mapping

The combination of virtual tagging and a location system can be used to develop a good mapping service for any environment. This service will provide representations of architectural spaces together with the objects situated in each space. This information is useful for developing user interfaces and for enabling system administrators to determine where to deploy additional sensors, actuators, or other entities.

We envision this mapping service constantly running in the background on the network, updating spatial representations as objects are added or removed from any environment. We choose a representation based on architectural spaces [51, 11], grouped by function and delineated in a common coordinate frame. For example, the indoor environment is partitioned into regions roughly corresponding to offices, corridors, conference rooms, lounges, elevator shafts, and the like. Analogously, outdoor spaces are represented by conceptual and functional grouping as courtyards, alleys, sidewalks, streets and the like. Indoor-outdoor and transitional spaces, such as entry stairs and ramps, foyers, and loading docks, are also represented. At a building system level, mains and ducts for air, water, power, data, and the like are represented. Depending on function, other elements such as openable windows may be represented as well.

Spaces and physical assets within them are represented as nodes in a multi-valued adjacency graph whose edges arise from the functional relationship of the space or asset to others in the representation. So, for example when representing pedestrian traffic, a corridor space shares an adjacency graph edge with every office reachable from the corridor, but two offices sharing a wall do not share such an edge, since to move from one office to the other a person would have to move through the common corridor. A graph over the same set of nodes (spaces), but representing air supply, does introduce an edge between the two offices, if they share ventilation ducting (enabling air to move directly from one office to the other). An openable window would induce an adjacency edge between an office and outdoor space for evacuation planning purposes, to model air flow, or to evaluate security against theft, but not for ordinary entrance to the office.

The deployment of a ubiquitous location system such as Cricket makes the acquisition problem significantly easier. A human installer can simply move through the environment, waving his hand-held around in each space while indicating the type of space he is in. Objects in the vicinity will be automatically discovered and sent to the mapping database. A geometric post-processing step would then organize all this information into a map of the space.

5 The SLAM Network

This section describes the research challenges and proposed solutions in the design of the SLAM network. This network has a two-level structure. The first level interconnects sensors with co-located sensor proxies, so that the raw data from the sensors can be annotated with contextual information (location, time, etc.) and converted into event streams that can be handled by other software components. The second level of the network provides a large-scale event-processing infrastructure to match events with appropriate handlers, and deliver event responses and actions to other handlers, actuators, or humans. The proposed design of the SLAM network involves a novel integration of ideas from two research communities—networking and database systems—with the intersection coming at the combination of event routing and sensor flow control with query processing and optimization.

5.1 Sensor Proxies

An activated environment is full of tagged objects and sensors reporting object and environmental status. We introduce *sensor proxies* to organize this mass of raw, distributed, transient sensor data into a scheduled, spatio-temporally indexed event stream. An *event stream* in SLAM is a raw sensor data integrated with identity, location, and timestamp information.

Proxies can be fixed, i.e., attached to the physical environment, or mobile, for example, attached to a handheld device or to a maintenance vehicle. Figure 3 illustrates some example proxy sensor configurations. In addition to annotating raw sensor data with contextual information to produce a data stream, sensor proxies implement sensor-specific communication protocols to communicate with sensors. The main issue in these protocols is energy-efficiency.

With current technology trends, it usually takes a lot more energy to transmit/receive data than perform some amount of computation, e.g., to fuse redundant sensor streams [33, 38]. In SLAM, we propose to build on this previous work to develop sensor-specific fusion algorithms that will reduce the amount of information being transmitted at the expense of some computation on the proxy. This is a well-known idea; our contribution will be to develop this further in the context of specific sensors—tags and location sensors in particular.

The other important function performed by sensor proxies relates to flow control, where they receive messages from handlers in the network that request that certain sensors be quenched because the information they provide is redundant or resource-intensive.

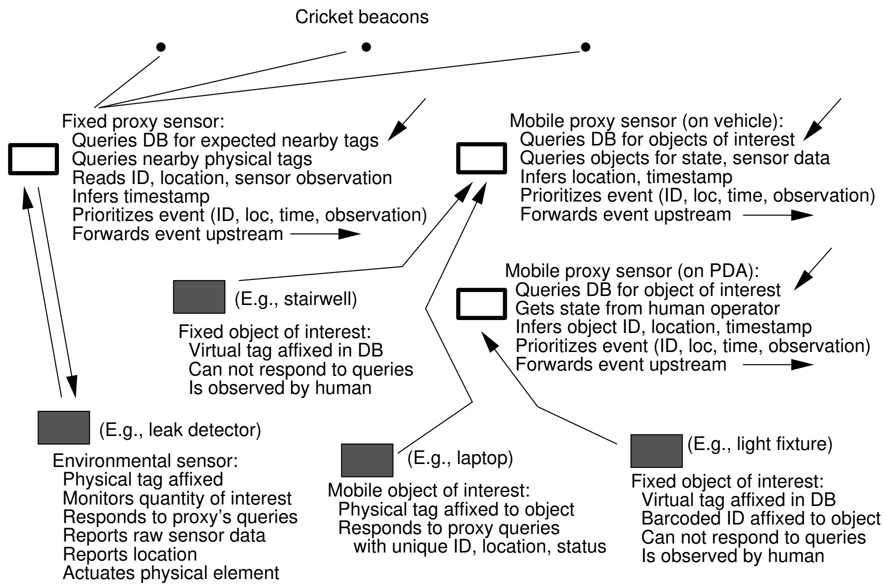

Figure 3: Proxy sensors interacting with virtual and physical tags.

5.2 SLAM Handler Network

Sensor proxies generate event streams that are processed by event handlers. SLAM applications are written as event handlers, each responding to one or more named event types. SLAM provides an event-handling network that performs three tasks: (1) distributing handlers on a distributed set of compute servers, (2) scalably routing events to the appropriate handler(s) in the network, and (3) supporting event processing at the handler nodes.

5.2.1 Distributed Processing

We envision SLAM application developers using a two-step process to write applications. First, they set up a process-flow diagram of basic event handlers (including sensor proxies) in a GUI. Then, they write code for each handler. The process-flow is passed to a compiler that undertakes the task of partitioning the handler code on to compute servers that will execute this code. One of the problems that needs to be solved is deciding how event handlers that may perform computationally intensive tasks are partitioned across servers and proxies.

Developing a GUI compiler is relatively straightforward, but finding an optimal solution to this graph-partitioning distributed-scheduling problem is decidedly not. We propose to investigate reasonable heuristics as well as theoretical algorithms with provable bounds for this problem in the context of stream processing. For example, in a simple heuristic, SLAM can record the processing capability of each server and proxy, and for each sensor, we can iteratively examine the first processing handler in the SLAM process flow and try to push it onto the node. This heuristic will push processing as close to message origination as possible, and lower the volume of messages that must be processed in the rest of the network. Likewise handlers can be pushed close to the machines to which output messages are delivered. We also propose to analyze the theoretical complexity of versions of this problem by employing approximation algorithms and fixed-parameter tractability [22]. Our expectation is that positive theoretical results will strongly impact the design of an efficient partitioning mechanism.

SLAM must deal with dynamic spikes in system load caused by external events. We propose to generalize the static algorithms discussed above to deal with dynamic load adjustment. One result will be a dynamic partitioning of handlers to compute nodes. Although we envision this happening over longer time-scales than typical event generation and response times, we are concerned about "flash crowd"-like activity when a sudden impulse (e.g., gas leak) triggers a large number of sensors and sensor proxies to generate great amounts of traffic to the handlers. Handling this concern requires dynamism in the location of handlers in the network.

5.2.2 Event Routing

Because handlers are not statically assigned to compute nodes, a scalable mechanism is required to route events to handlers and proxies. If each application handler were to independently deal with the problem of discovering the network locations where the other handlers or actuators were running, the system would be highly complex. We therefore propose to provide a discovery mechanism in the SLAM network in which desired destination handlers are simply named, and the network takes care of routing the messages to the appropriate places in a scalable manner. In addition to unicast message routing, events may have to be replicated to route to multiple handlers.

We propose to investigate several different ways of solving the scalable routing problem. The simplest approach would be to maintain this information in the centralized process-flow GUI and, each time a handler moves in the network, update this information in the GUI. New handlers can now be easily added to the GUI and the appropriate network plumbing between handlers set up.

For long-running SLAM systems, and in systems where trusted third parties may introduce themselves in the stream flow to observe streams of interest, it may not always be possible to maintain and update a centralized GUI. We require a distributed and scalable discovery protocol that can return where named event handlers are currently running. We propose to investigate this by deriving inspiration from distributed service [1, 8, 20] and document discovery [24, 30] protocols. In particular, we will investigate the design of a scalable and dynamic event routing mechanism layered over a scalable peer-to-peer key-value lookup protocol such as Chord [49], CAN [45], Pastry [46], or Tapestry [55].

5.2.3 Event Handling

Choice of primitive handlers. Although it is reasonable for a SLAM application developer to construct his/her own handlers from scratch to implement any desired functionality, SLAM should provide a palette of useful parameterized handlers to perform common event stream manipulation functions. This is analogous to the primitive operations provided by relational database systems, such as join, project, and select.

One clearly desirable primitive operation is to match up the streams of messages produced by two sensors or two intermediate handlers, as well as to merge two streams. For example, one might want to know if a laptop and its power cord are in the same room. This requires matching two streams on common geographic location. However, messages in SLAM are asynchronous, and the two streams may report values at different times. As such, we need to generalize the traditional notion of join into a windowed operation, in which two values can be joined together if their values match within a predetermined time window. A second challenge is that some sensors are imprecise, especially for location. Relational database systems have traditionally assumed that data items have exact values. In SLAM, we have to generalize operations to work on imprecise data elements.

Other primitive operations include delaying an event response until some condition is true, such as the arrival of another event; value interpolation, so that a stream can provide a value between measurements; and applying a user-defined function over the historical values in a stream. Providing a primitive handler in each case that is parameterized by a function definition will allow handler support code to deal with necessary data storage, and let the user concentrate on the specific function.

Query processing and optimization. In a large SLAM network, we expect to have thousands, if not hundreds of thousands, of handlers. Handlers in the application's workflow will routinely generate events and query each other. Combining all the handlers into a massive query and then applying conventional query optimization is not a workable approach. Instead, SLAM must focus on a collection of alternate tactics.

First, it will often be useful to combine multiple handlers together and co-locate them. Whenever handlers are commutative, the cheaper one can be moved forward to be performed in advance of the more expensive one. Although this is generally a good tactic, it must be used with care, because there will be handlers that generate more messages than they consume. In this case, a more complex test for reordering must be applied.

Another optimization tactic is to use special data structures to speed up interpretation. Consider the situation in which a handler's output is split a number of times and sent to a large collection of subsequent handlers. Moreover, suppose each handler is some sort of filter operation. In the MIT power plant, for example, there might be many different actions to take, depending on the value of an input sensor. In this case, it is possible to assemble all the filters into a single "superhandler," supported by an R-tree or other spatial index. The superhandler inserts incoming values into the R-tree in order to discover which processing handlers must be activated. We propose to investigate data structures of this sort that can accelerate processing of a SLAM network.

We also propose to examine the tradeoff between storing values and recomputing them. As mentioned earlier, windowed operations require SLAM to maintain a database of historical values to be used by handler operations. The basic tactic to be explored is to move one or more data sets in this database from their current location between two operators, to a position earlier in the diagram. This will force SLAM to reprocess intervening handlers, thereby adding to the processing load. On the other hand, the total amount of storage required may be reduced.

Ad-hoc queries. Users and administrators may wish to make *ad-hoc queries* of a running SLAM system, not as part of the application workflow. One obvious way to do this is to allow a casual user to add handlers to a running SLAM system. However, the previous section indicated that handlers may be moved around and combined. Hence, the

SLAM stream to which the casual user connects may not exist in the operational system. Also, this approach modifies the code of a running program, a tactic that is usually considered bad software engineering. Lastly, if the casual user asks a historical query, the required information may not have been retained by SLAM.

A possible solution is to allow a SLAM application administrator to specify that additional history should be kept, even though it is not required for correct system operation. This could be specified by annotating arcs in a SLAM diagram. Furthermore, such an annotation would be a constraint to the optimizer that the arc cannot be removed. Then, the casual user could add handlers that performed queries of the saved data on this arc. However, there may be better ways to support ad-hoc queries, which we have to investigate.

Provisioning, flow control and quality-of-service. A steady state static analysis is needed to determine if the hardware on which SLAM is running is sized adequately for the expected load placed upon it. If not, SLAM can take one of several corrective actions. It can repartition the diagram over the server machines to place less load on the machine that has too little capacity. Alternatively, it can place "drop" handlers into the diagram to cut down on the message activity and lower load. Of course, intelligently performing this function requires a model of quality of service in SLAM for each output. We propose to investigate such a model and algorithms to deal with static load adjustment.

In response to sudden spurts in activity, particular handlers or proxies may be overwhelmed by data from certain sensors. If the raw information being provided is redundant or less important, these streams should be throttled. We propose a simple back-pressure strategy through the handler network via sensor proxies to throttle raw sensors and event-generating handlers. In addition, we propose to investigate priority-based event-generation in the handler network, so that more important events are scheduled before less important ones. Unlike in traditional data networking, the notion of "quality-of-service" (QoS) in event-driven stream systems is not well-developed. Developing a deeper understanding and model of what QoS means in the context of SLAM systems is an important research goal. As we gain experience with SLAM, we expect to develop insights that can help model and solve QoS-related problems.

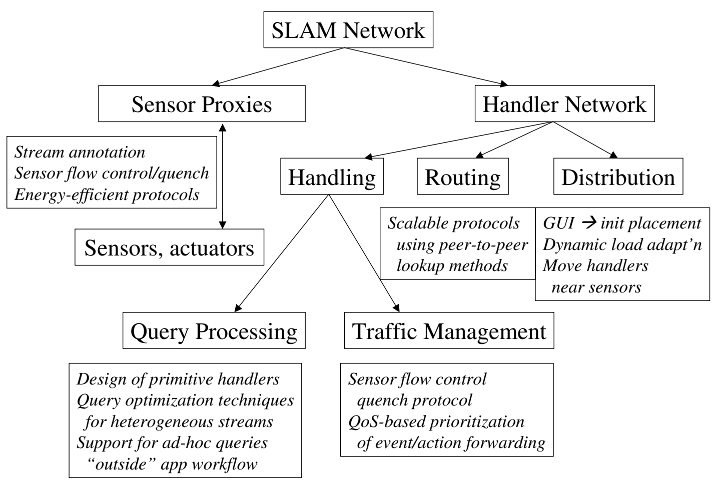

Figure 4: Research problems in the SLAM network architecture.

6 Security and Privacy Challenges

An obvious concern with the proposed network architecture is the sensitivity of the copious data communicated. Thus an important research issue in SLAM is to design appropriate privacy models and build corresponding security mechanisms. We describe below some of the main issues and planned solutions, but expect to become aware of all relevant issues only as we gain experience as SLAM is deployed.

Access to knowledge of the location of humans and objects on their person must be restricted. Otherwise, most humans would be unwilling to wear location devices because of the loss of privacy from being tracked. Fortunately, the Cricket location infrastructure has an inherent privacy capability. Because the mobile devices are passive, the beacons in the location infrastructure do not know which devices are nearby or even whether any device is nearby. Thus each device has control over how its location information is distributed.

More generally, any sensor data stream may be sensitive so its distribution must be controlled. For example, security-camera output may only be appropriate for certain personnel and for certain monitoring processes. One natural scheme for restricted distribution of sensor streams is *capabilities* [50]. Each sensor stream is encrypted according to a unique private access key whose possession is restricted to trusted devices. Each device or group of similar devices additionally has a public/private key pair so that the access keys may be securely communicated via public-key cryptography [2].

A second important issue is authentication: e.g., to prevent malicious alarms, we need a mechanism for verifying that a stream actually comes from the claimed sensor. One way to achieve this uses digital signatures [2]). When a new sensor device is installed, a human of appropriate authority signs the device's key pair as authentic for a particular set of attributes including sensor type. The device then signs its sensor stream data with its public/private key pair. A content-distribution mechanism [12] may be employed to distribute the signed public keys of sensor devices, thereby enabling verification of signed sensor streams.

The third important issue is misconfiguration. While SLAM is designed to maximize automation and minimize the possibility of human error (for example, by automating location information), each instrumented object inevitably requires some initial human tagging of attributes such as the type of object (laptop, printer, wall, table, etc.). In the future we might imagine that devices are sufficiently tagged upon manufacture with minimal error, but a SLAM system built on today's components needs to tolerate such misconfiguration errors.

We plan to investigate several approaches for dealing with misconfiguration. First, a background process may search for inconsistencies in data. This problem is of course inherently difficult, but may become feasible in the expected case that misconfiguration errors are rare. For example, a human may specify certain "normal conditions" (e.g., objects tagged as pumps are normally found in rooms tagged as maintenance rooms) whose violations are indicated to a system administrator. Second, when a misconfiguration is detected, either automatically or by a human, there must be an easy mechanism for humans to update the information. For example, the human can point her handheld to the misconfigured object, bring up configuration data, and make changes. Whether these changes are made directly or through a request to a system administrator depends on the authority granted to that user's public key.

We envision that the software handlers are written primarily by trusted users and therefore can be run in a fairly trusting environment. However, it is essential that a bug in a software handler has a limited effect outside that handler. One component of a solution to this problem is a limiting mechanism that prevents the communication network from being flooded by data from sensors or handlers that are not expected to produce so much data. Each data generator can have a signed capability representing the allowed bandwidth; critical and emergency devices can be allocated unlimited bandwidth. Another hard issue is protection against denial-of-service attacks on sensor and sensor proxy streams. We plan to look at these in later stages of the project.

In general, the flexibility of SLAM enables a powerful set of privacy policies. Indeed, we believe that social acceptance of simple unobtrusive sensors such as heat and motion sensors will be much broader than traditional security cameras, thereby allowing more widespread use and monitoring while still supporting most desired applications. Where needed, of course, SLAM can also incorporate security-camera feeds, but we believe that even non-intrusive sensors can provide great benefit especially in campus environments.

7 Preliminary Results

We have designed and prototyped a small-scale indoor location system [43, 44] (see Figure 5) and expect to build on it over the next few years in SLAM. We will expand this basic prototype to solve the problems raised in Section 3. We expect the end-result to be a scalable, ubiquitous, easy-to-deploy, and robust location infrastructure.

We have developed energy-efficient protocols (LEACH and Span) for disseminating information in ad hoc network of sensors. We will use a variant of LEACH [33] for the communication between sensors and a co-located sensor proxy. Span, an energy-efficient topology formation algorithm for ad hoc networks [13], will be a starting point for designing an efficient topology over which beacons can communicate with each other during their self-calibration phase.

We have continued a thread of work originating from efficient architectural walkthroughs [51, 26] to capture and simulate complex spaces effectively [11]. For example, we have recently parsed MIT

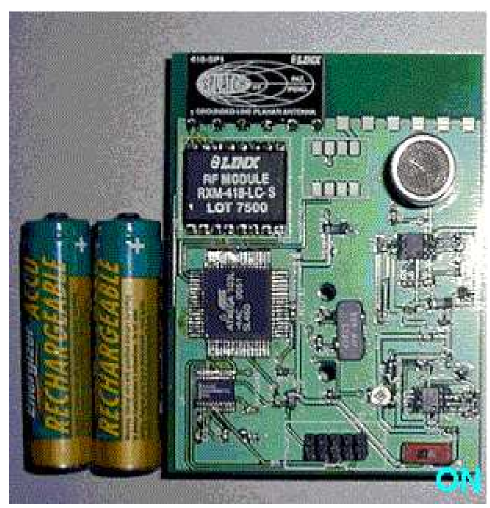

Figure 5: Our listener prototype with RF and ultrasonic sensors for small-scale indoor location estimation. We expect to develop SLAM's location infrastructure on this basic hardware.

Physical Plant's floorplans representation to produce reference office and adjacency maps, useful to validate and visualize the Cricket location infrastructure.

We have developed an efficient loop-free routing algorithm that exploit location information in the network in order to reduce packet headers to minimal size [10]. Specifically, given just the current coordinates of the packet, the coordinates of the neighboring nodes, and the coordinates of the destination, our algorithm determines where the packet should next be delivered. This routing strategy should prove useful in the design of the beacon self-configuration protocol.

8 Related Research

Sensor Networks & Systems. Sensor networks and systems are a current area of fertile research, and numerous projects have recently begun. We describe only the ones most closely connected with SLAM here.

CITRIS [15] is a large-scale project in California that is addressing several issues related to sensor networks. A particular connection to our project is the application of emergency situations like earthquakes. CITRIS's goal is to build pervasive, secure, energy-efficient, and disaster-proof information systems. SLAM can be seen as a key component in any such system for enabling omnipresent location information and sensor-streaming infrastructure.

Auto-ID [47] proposes attaching persistent physical identifiers to all objects, for example to improve supply chain, inventory, and maintenance efficiency. In contrast, SLAM focuses primarily on monitoring and control applications. AUTO-ID has no notion of virtual tags.

The "Smart Dust" project at Berkeley is pursuing research in miniature sensors and self-configuring communication protocols for inter-sensor communication. SLAM's approaches to beacon configuration may prove useful in Smart Dust; likewise, SLAM will benefit from the novel sensors being developed by Smart-Dust. Estrin's LECS lab at UCLA is pursuing a number of interesting sensor networking projects. Synergies with SLAM include self-configuring network protocols and algorithms and location systems. However, whereas most projects in LECS focus on remote sensors collaborating with each other, SLAM's focus is on integrating sensor information with monitoring and control software running on traditional machines in a scalable manner.

SLAM incorporates a combination of novel ideas not found in many other current systems. These include a ubiquitous and precise location system that works in all places, a virtual tagging mechanism for environment activation, novel low-energy communication protocols between sensors and sensor proxies, and a scalable event processing architecture that integrates ideas from networking (event routing) with ideas from database systems (query and data processing). These ideas are all required for SLAM to scale to tens of millions of entities.

Location. In outdoor areas where line-of-sight connectivity to satellite is available, GPS [28, 35] works well in providing location information. However, it does not work well in indoor or urban areas, where buildings and other obstacles block the reception of signals from satellites. In addition, GPS does not provide the degree of precision required for mobile applications indoors because of the low RF signal strength, high RF noise, and the reflections of RF signals due to the presence of metallic objects. Therefore researchers have explored technologies for indoor environments based on infrared (Active Badge [53]), radio frequency (RF) signal strengths (RADAR [7]), ultrasound [29], and a combination of RF and ultrasound (Bat [32] and the current indoor version of Cricket [43]). These systems work in indoor environments but not elsewhere.

The Bat system requires a carefully calibrated infrastructure of sensors on ceilings, all wired together and connected to a centralized controller that schedules transmission events [32]. It takes many months to deploy such a system in even a modestly-sized building, and is hard to maintain.

We expect variants of these approaches to be a good starting point for our self-calibration system. Doherty *et al.* model the position estimation problem in ad-hoc sensor networks as a convex optimization problem, showing that under some conditions it is possible for the nodes to discover their positions relative to one another [21]. Savvides *et al.* [48] describe another approach to this problem.

Data processing systems. The SLAM event processing network is related to database research in adaptive query processing, continuous queries, triggers, and stream processing.

Adaptive query processing techniques (e.g., [5, 34, 36, 52]) address efficient query execution in unpredictable and dynamic environments (e.g., Internet data sources) by revising the query execution plan as the characteristics of incoming data changes. It is possible that adaptive query processing is also useful for efficient processing of stream data and we plan to investigate the utility of this technique.

A special case of SLAM processing is as a continuous query system. Many continuous query systems have focused on the topic of indexing the queries. A system like Niagara [14] is concerned with combining multiple data sources in a wide-area setting while we are initially focusing on the construction of a stream server that can process very large numbers of streams.

Recent closely related work on stream data query processing architectures shares many of the goals and target application domains with SLAM. The Streams project [6] attempts to provide complete DBMS functionality along with support for continuous queries over streaming data. The Fjords architecture [39] addresses continuous query execution strategies that combine push-based sensor data with data from traditional sources that produce data using a pull-based, blocking interface.

Our work is likely to benefit from and contribute to the considerable research on temporal databases [41], since SLAM has a fundamental temporal aspect to it. Likewise, we can benefit from the literature on real-time databases [41, 37] and main-memory databases [27]. Finally, many recent papers [6, 54, 16, 23] have addressed the data management needs of specific monitoring or tracking applications. We take inspiration from these works as we try to design a general-purpose system that can be used for all these tasks.

9 Broader Impact of Proposed Research

The proposed ubiquitous location-sensing infrastructure has applications and utility beyond SLAM systems. Just as outdoor GPS has enabled numerous applications, Cricket is likely to enable numerous location-aware applications that can work reliably in all settings. We have already started sharing our small-scale Cricket prototype with other universities and companies such as Dartmouth, Duke, Univ. of South Carolina, Nokia, Philips, and Intel. We expect the significantly harder-to-develop system that provides ubiquitous coverage to be of wide interest to other researchers and developers.

The self-configuration algorithms and energy-efficient protocols proposed herein have widespread applicability in other wireless ad hoc and sensor systems. The idea of virtual tagging can be used in other pervasive computing systems to rapidly instrument an environment's assets. The event-handling architecture of the SLAM network is likely to provide general lessons for designing distributed systems with asynchronous event flows.

The proposed SLAM applications on MIT campus has already elicited excitement among interested campus partners, particularly the libraries. In addition to providing a real testbed for our research ideas, these applications have the potential to improve the quality of life of students and workers on our campus. If successful, we plan to deploy these systems on other interested campuses as well.

As important as the software and hardware reference schematics that we will release and protocols we will develop are the papers and students that will result from the proposed project. The papers will provide precise documentation on the proposed technologies and their importance. In addition to the software and hardware, the papers are the prime source of information that companies and research institutes will use to evaluate and adopt the proposed ideas.

In the end, the best technology transfer is through students. With the proposed research project, the students involved will gain extensive experience with wireless networks, communication protocols, hardware design, database systems, large-scale software engineering, algorithms (particularly in scheduling and computational geometry), and graphics. Expertise in these areas will make them of great value to universities, research labs, and industry. Academic research projects such as this one are of particular importance, because broad experience with these core areas of computer science and engineering is hard to obtain in industry. In the long run, these students are the next generation of educators, researchers, and entrepreneurs who will drive future innovation in computer science and engineering.

10 Education

There are a variety of ways in which SLAM will have a positive impact on student education and quality of life. Many of the research and engineering issues inherent in deploying SLAM can be used as material in systems, network, graphics, and geometry courses. Architecting and implementing SLAM will generate dozens of undergraduate research positions over five years. It will also be a fertile ground for class projects, particularly in graduate courses in networking, systems, graphics, and algorithms.

One tangible product of SLAM deployment is a rich set of geometric representations, models, and sensor data for extended environments. These data will be useful in a variety of undergraduate courses, for example in architecture, urban planning, and computer graphics.

SLAM's library application should make it easier for students to retrieve both heavily-used and infrequently-used books and journals from library table-tops, circulation counters, and dead storage. Its active signage component will make it more likely that students will be aware of, and attend, lectures, colloquia, and special events. Its safety and theft detection component will improve the safety of students and the response time of campus police in times of trouble.

11 Management Plan

Project management will be handled by the PI, Hari Balakrishnan, who will be responsible for the overall management of the ITR effort including all policy and budget decisions. Purchases, salaried personnel, space allocation, etc., will all be Balakrishnan's final responsibility.

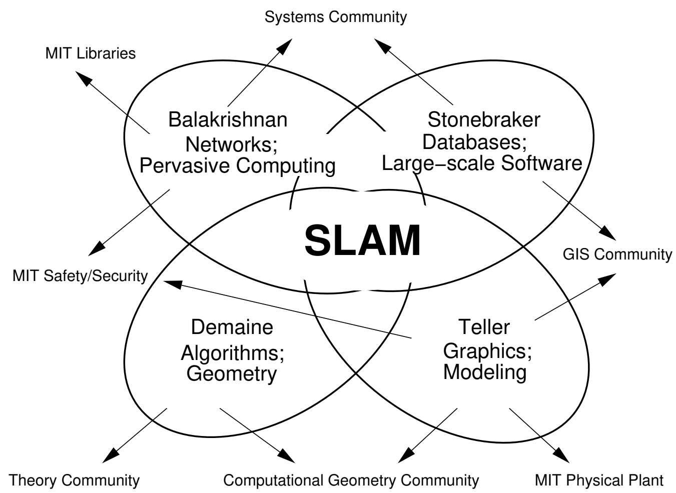

Figure 6: PI expertise and connections to other communities outside and within MIT.

The four co-PIs will coordinate and collaborate on SLAM research and development. Prof. Victor Zue, currently the interim Director of the Laboratory for Computer Science, will act as liaison between the co-PIs and the administrative units of the Laboratory for Computer Science, EECS Department, and Institute.

There will be a Principal Research Scientist (1 FTE) whose primary role will be software engineering and supervising sensor fabrication. Post-doctoral researchers, PhD students, Master's students, and UROPs (undergraduate researchers) will be supervised by one or more PIs as appropriate.

11.1 Breakdown by SLAM Components

Responsibility for advancing the three main SLAM components will be divided as follows:

Cricket location infrastructure. Balakrishnan, Demaine, and Teller will work on the design and implementation of Cricket, including sensor design, algorithms, and software engineering. Balakrishnan and Teller have already collaborated on the design of the Cricket compass for deducing orientation in indoor settings [44]. Demaine's expertise in algorithms and computational geometry provides good synergies for this effort.

Environment activation. Teller will lead the development and population of the functional geometric environment representation. This effort requires a persistent object and attribute management store, an area in which Stonebraker is an expert. Virtual tagging requires a precise orientation and position capability, which Balakrishnan and Teller have been working together on. Already, one of our students is jointly working with us on this problem, and we expect this to grow as the project blossoms.

The SLAM network. This component of the SLAM architecture requires innovation at the intersection of the networking and database systems communities. Balakrishnan and Stonebraker are experts in these respective areas, and have been working together on this design. The distribution of handlers leads to interesting algorithmic problems in distributed scheduling, which Demaine will collaborate on. Our plan is to implement and evaluate theoretical ideas, rather than keep the theoretical developments divorced from the real system.

11.2 Breakdown by Application Focus

Our target SLAM system will focus initially on three capabilities, with a variety of interested partners: efficient facilities monitoring and maintenance (with MIT Physical Plant); scalable asset monitoring for inventory, crime prevention and detection (with the MIT Property Office, MIT Campus Police, and MIT Libraries); and navigation assistance, including both personal way-finding and pervasive active signage (with the MIT Schedules Office and the MIT Safety Office).

Interactions with MIT Libraries: Balakrishnan has been working with staff of the LCS Reading Room and MIT Libraries, who are keen to incorporate asset tracking and navigation methods into their operating procedures. We have already scoped out what the initial pilot will contain.

Interactions with MIT Physical Plant: Teller has collaborated extensively with MIT's Physical Plant over the past few years, in an ongoing effort to achieve rapid procedural modeling of all campus buildings and floorplans. Teller will serve as the main liaison between the SLAM effort and the Physical Plant staff.

Interactions with GIS community: Stonebraker and Teller have had extensive interaction with this community as part of other research efforts (Stonebraker's work with NASA Earth Observing System, Teller's City Scanning project for rapid urban model capture).

Interactions with MIT Safety Office and MIT Police: Balakrishnan and Teller will liaison with these administrative units.

Interactions with MIT Schedules Office: Demaine will lead in working with this office, and other event planners and schedulers around LCS and MIT, toward pervasive active signage.

11.3 Five-Year Plan

Our overall plan has three phases. Much of the basic research and early deployment will be done in the first several years, in order to tackle scaling issues immediately. Later we will focus more on evaluation and technology transfer to our partners both within and outside of MIT.

Broadly speaking, our strategy will be to deploy a prototype location infrastructure and sensor network rapidly, over an extended environment (several floors of our building), in order to generate a challenging operational scale. Algorithmic development will proceed in parallel. The current five-year plan can be summarized as:

Year 1. Develop a multi-sensor location system. At the same time, deploy our current indoor prototype with physical RF-ID tags on a building-wide scale, evaluating performance and documenting the scaling problems. Work on solving these problems. Design algorithms for distributed event handling and query processing in the SLAM network. Evaluate energy-efficiency of prototype.

Year 2. Design a virtual tagging system by leveraging the orientation capabilities of indoor sensing. Deploy active beacons more widely across MIT. Integrated design incorporating GPS and 802.11 access point information for location. Design and implement a generic technology-independent location API. Complete a preliminary prototype of the SLAM network, including query optimization and a simple static handler distribution. Engage with physical plant in deploying sensors and SLAM handlers.

Year 3. Conduct experiments with initial prototype and library tracking pilot application. Evaluate performance and scaling problems. Evaluate energy-efficiency and traffic management problems. Develop sensor flow control protocols. Design energy-efficient sensor-specific protocols. Engage with campus security for asset tracking and start deploying more widely in library system and physical plant. Start investigating security and privacy issues.

Year 4. Develop and implement scalable handler distribution and dynamic event routing methods. Develop a variety of handlers performing interesting data manipulation operations for physical plant and security applications. Evaluate performance of real system and develop modeling techniques for understand how large-scale sensor systems behave. Understand load behavior and response. Develop simple security solutions and deploy them.

Year 5. Demonstrate complete ubiquity of location system and show the benefits of a single system that works everywhere. Develop a smaller version of earlier prototypes. Show scaling up to 10^7 entities using both emulation (and rigorous stress tests) and in real deployment at MIT. Gain practical experience with social acceptance and hurdles.

We will release software and hardware schematics on an on-going basis through the project for the research community. We expect to contribute actively to the top conferences in the different areas covered in this proposal, including those run by ACM SIGMOBILE, ACM SIGCOMM, ACM SIGMOD, ACM SIGACT, and IEEE networking and data management conferences. The students working on this project will gain expertise in multiple areas of computer science: networking, algorithms, database systems, and graphics/vision.

D. References Cited

[1] ADJIE-WINOTO, W., SCHWARTZ, E. AND BALAKRISHNAN, H. AND LILLEY, J. The design and implementation of an intentional naming system. In *Proc. ACM Symposium on Operating Systems Principles* (Kiawah Island, SC, Dec. 1999), pp. 186–201.

[2] ALFRED J. MENEZES, P. C. V. O., AND VANSTONE, S. A. *Handbook of Applied Cryptography*. CRC Press, October 1996.

[3] ANTONE, M., AND TELLER, S. Automatic recovery of relative camera rotations for urban scenes. In *Proc. CVPR* (June 2000), pp. II–282–289.

[4] ANTONE, M., AND TELLER, S. Scalable, absolute position recovery for omni-directional image networks. In *Proc. CVPR (to appear)* (2001).

[5] AVNUR, R., AND HELLERSTEIN, J. M. Eddies: Continuously adaptive query processing. In *Proceedings of the ACM SIGMOD International Conference on Management of Data* (2000).

[6] BABU, S., AND WIDOM, J. Continuous queries over data streams. *SIGMOD Record 30*, 3 (2001).

[7] BAHL, P., AND PADMANABHAN, V. RADAR: An In-Building RF-based User Location and Tracking System. In *Proc. IEEE INFOCOM* (Tel-Aviv, Israel, Mar. 2000).

[8] BALAZINSKA, M., AND BALAKRISHNAN, H. Twine: Scalable intentional resource discovery for pervasive computing environments. `http://nms.lcs.mit.edu/projects/twine/`.

[9] BIEDL, T. C., BOSE, P., DEMAINE, E. D., AND LUBIW, A. Efficient algorithms for Petersen's matching theorem. In *Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms* (Baltimore, Maryland, January 1999), pp. 130–139.

[10] BOSE, P., BRODNIK, A., CARLSSON, S., DEMAINE, E. D., FLEISCHER, R., LÓPEZ-ORTIZ, A., MORIN, P., AND MUNRO, J. I. Online routing in convex subdivisions. In *Proceedings of the 11th Annual International Symposium on Algorithms and Computation* (Taipei, Taiwan, December 2000), vol. 1969 of *Lecture Notes in Computer Science*, pp. 47–59.

[11] BUKOWSKI, R., DOWNS, L., SIMMONS, M., SÉQUIN, C., AND TELLER, S. Citywalk: A second generation walkthrough system. In *In Proc. Virtual Systems and Multimedia (VSMM)* (October 2001).

[12] *Proceedings of the 1st–6th International Workshops on Web Caching and Content Distribution*. Elsevier, 1996–2001.

[13] CHEN, B., JAMIESON, K., BALAKRISHNAN, H., AND MORRIS, R. Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. In *Proc. ACM MOBICOM* (Rome, Italy, July 2001).

[14] CHEN, J., DEWITT, D. J., TIAN, F., AND WANG, Y. Niagaracq: A scalable continuous query system for internet databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data* (2000).

[15] CITRIS – center for information technology research in the interest of society. `http://www.citris.berkeley.edu/`.

[16] COIFMAN, B. Identifying the onset of congestion rapidly with existing traffic detectors. Tech. Rep. UCB-ITS-PWP-99-17, University of California, Berkeley, November 1999. Submitted to *Transportation Research: Part A/B*.

[17] CONNELLY, R. Rigidity. In *Handbook of Convex Geometry*, vol. A. North-Holland, 1993, pp. 223–271.

[18] CONNELLY, R., AND DEMAINE, E. D. Geometry and topology of polygonal linkages. In *Handbook of Discrete and Computational Geometry*, 2nd ed. CRC Press, 2002. To appear.

[19] CONNELLY, R., DEMAINE, E. D., AND ROTE, G. Straightening polygonal arcs and convexifying polygonal cycles. *Discrete & Computational Geometry* (2002). To appear. A preliminary version appeared in Proc. 41st Symp. Foundations of Computer Science, 2000, pp. 432–442.

[20] CZERWINSKI, S., ZHAO, B., HODES, T., JOSEPH, A., AND KATZ, R. An Architecture for a Secure Service Discovery Service. In *Proc. ACM/IEEE MOBICOM* (Aug. 1999), pp. 24–35.

[21] DOHERTY, L., PISTER, K., AND GHAOUI, L. Convex position estimation in wireless sensor networks. In *Proc. IEEE INFOCOM* (April 2001).

[22] DOWNEY, R. G., AND FELLOWS, M. R. *Parameterized Complexity*. Springer-Verlag, 1997.

[23] DUFFIELD, N. G., AND GROSSGLAUSER, M. Trajectory sampling for direct traffic observation. *IEEE/ACM Transactions on Networking 9* (2001), 280–292.

[24] FastTrack. `http://www.fasttrack.nu/`.

[25] FINKENZELLER, K. *RFID Handbook—radio-frequency identification fundamentals and applications*. John Wiley & Sons, 1999.

[26] FUNKHOUSER, T., SÉQUIN, C., AND TELLER, S. Management of large amounts of data in interactive building walkthroughs. In *Proc. 1992 Workshop on Interactive 3D Graphics* (1992), pp. 11–20.

[27] GARCIA-MOLINA, H., AND SALEM, K. Main memory database systems: An overview. *IEEE Transactions on Knowledge and Data Engineering 4* (1992).

[28] GETTING, I. The Global Positioning System. *IEEE Spectrum 30*, 12 (December 1993), 36–47.

[29] GIROD, L. Development and characterization of an acoustic rangefinder. Tech. Rep. 00-728, Computer Science Department, University of Southern California, April 2000.

[30] Gnutella. `http://gnutella.wego.com/`.

[31] GONZÁLEZ-BANOS, H., AND LATOMBE, J.-C. A randomized art-gallery algorithm for sensor placement. In *Proceedings of the 17th ACM Symposium on Computational Geometry* (2001).

[32] HARTER, A., HOPPER, A., STEGGLES, P., WARD, A., AND WEBSTER, P. The Anatomy of a Context-Aware Application. In *Proc. 5th ACM MOBICOM Conf.* (Seattle, WA, Aug. 1999).

[33] HEINZELMAN, W., CHANDRAKASAN, A., AND BALAKRISHNAN, H. Low-energy adaptive clustering hierarchy. *IEEE JSAC* (2002). To appear. Earlier version in Proc. HICSS 2000.

[34] HELLERSTEIN, J. M., FRANKLIN, M. J., CHANDRASEKARAN, S., DESHPANDE, A., HILDRUM, K., MADDEN, S., RAMAN, V., AND SHAH, M. A. Adaptive query processing: Technology in evolution. *IEEE Data Engineering Bulletin 23* (2000), 7–18.

[35] HOFFMANN-WELLENHOF, B., LICHTENEGGER, H., AND COLLINS, J. *Global Positioning System: Theory and Practice, Fourth Edition*. Springer-Verlag, 1997.

[36] IVES, Z. G., FLORESCU, D., FRIEDMAN, M., LEVY, A., AND WELD, D. S. An adaptive query execution system for data integration. In *Proceedings of the ACM SIGMOD International Conference on Management of Data* (1999).

[37] KAO, B., AND GARCIA-MOLINA, H. An overview of realtime database systems. In *Real Time Computing*, W. A. Halang and A. D. Stoyenko, Eds. Springer, 1994.

[38] KULIK, J., HEINZELMAN, W., AND BALAKRISHNAN, H. Negotiation-based Protocols for Information Dissemination in Wireless Sensor Networks . *ACM Wireless Networks* (2001). To appear. Earlier version in Proc. ACM MOBICOM 1999.

[39] MADDEN, S., AND FRANKLIN, M. J. Fjording the stream: An architecture for queries over streaming sensor data. In *Proceedings of the 18th International Conference on Data Engineering* (2002).

[40] O'ROURKE, J. *Art Gallery Theorems and Algorithms*. Oxford University Press, 1987.

[41] OZSOYOGLU, G., AND SNODGRASS, R. T. Temporal and real-time databases: A survey. *IEEE Transactions on Knowledge and Data Engineering* (1995).

[42] PREPARATA, F. P., AND SHAMOS, M. I. *Computational Geometry: an Introduction*. Springer-Verlag, 1985.

[43] PRIYANTHA, N., CHAKRABORTY, A., AND BALAKRISHNAN, H. The Cricket Location-Support System. In *Proc. 6th ACM MOBICOM Conf.* (Boston, MA, Aug. 2000), pp. 32–43.

[44] PRIYANTHA, N., MIU, A., BALAKRISHNAN, H., AND TELLER, S. The Cricket Compass for Context-Aware Mobile Applications. In *Proc. 7th ACM MOBICOM Conf.* (Rome, Italy, July 2001), pp. 1–14.

[45] RATNASAMY, S., FRANCIS, P., HANDLEY, M., KARP, R., AND SHENKER, S. A scalable content-addressable network. In *Proc. ACM SIGCOMM* (San Diego, CA, August 2001), pp. 161–172.

[46] ROWSTRON, A., AND DRUSCHEL, P. Pastry: Scalable, distributed object location and routing for large-s cale peer-to-peer systems. In *Proceedings of the 18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001)* (Nov. 2001).

[47] SARMA, S., BROCK, D., AND ASHTON, K. The networked physical world: Proposals for engineering the next generation of computing, commerce and automatic identification. Tech. Rep. MIT AUTO-ID-WH-001, MIT Auto-ID Center, December 2000.

[48] SAVVIDES, A., HAN, C.-C., AND STRIVASTAVA, M. B. Dynamic fine-grained localization in ad-hoc networks of sensors. In *Proc. 7th ACM MOBICOM Conf.* (Rome, Italy, July 2001), pp. 166–179.

[49] STOICA, I., MORRIS, R., KARGER, D., KAASHOEK, M. F., AND BALAKRISHNAN, H. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proc. ACM SIGCOMM* (San Diego, Aug. 2001).

[50] TANENBAUM, A. S., MULLENDER, S. J., AND VAN RENESSE, R. Using sparse capabilities in a distributed operating systems. In *Proceedings of the 6th International Conference on Distributed Computing Systems* (May 1986), pp. 558–563.

[51] TELLER, S., AND SÉQUIN, C. H. Visibility preprocessing for interactive walkthroughs. *Computer Graphics (Proc. Siggraph '91) 25*, 4 (1991), 61–69.

[52] URHAN, T., AND FRANKLIN, M. J. Dynamic pipeline scheduling for improving interactive query performance. In *Proceedings of 27th International Conference on Very Large Data Bases* (Rome, Italy, 2001).

[53] WANT, R., HOPPER, A., FALCAO, V., AND GIBBONS, J. The Active Badge Location System. *ACM Transactions on Information Systems 10*, 1 (January 1992), 91–102.

[54] WOLFSON, O., CHAMBERLAIN, S., SISTLA, P., XU, B., AND ZHOU, J. DOMINO: Databases fOr MovINg Objects tracking. In *Proceedings of the ACM SIGMOD International Conference on Management of Data* (Philadelphia, PA, 1999).

[55] ZHAO, B., KUBIATOWICZ, J., AND JOSEPH, A. Tapestry: An infrastructure for fault-tolerant wide-area locat ion and routing. Tech. Rep. UCB/CSD-01-1141, Computer Science Division, U. C. Berkeley, Apr. 2001.

E. Biographical Sketches

Hari Balakrishnan

Professional Preparation

School	Degree	Date
University of California, Berkeley	Doctor of Philosophy	August 1998
Indian Inst. of Technology, Madras	Bachelor of Technology	June 1993

Appointments

Organization	Position	Date
Massachusetts Inst. of Technology	Assistant Professor	July 1998–present
Univ. of California, Berkeley	Graduate Student Researcher	August 1994–August 1998
Univ. of California, Berkeley	Graduate Student Instructor	August 1993–May 1994

Five Publications Relevant to Proposal

1. Priyantha, N. B., A. Chakraborty, and H. Balakrishnan, "The Cricket Location-Support System," *Proceedings of the 6th ACM MOBICOM Conference,* Boston, MA, August 2000.

2. Priyantha, N. B., A. K. L. Miu, H. Balakrishnan, and S. Teller, "The Cricket Compass for Context-Aware Mobile Applications," *Proceedings of the ACM MOBICOM Conference,* Rome, Italy, July 2001.

3. Heinzelman, W. R., A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocols for Wireless Microsensor Networks," *IEEE JSAC*, 2002. (To appear. Earlier version in Proc. HICSS 2000.)

4. Kulik, J. L., W. R. Heinzelman, H. Balakrishnan, "Negotiation-Based Protocols for Disseminating Information in Wireless Sensor Networks," *ACM Wireless Networks,* 2001. (To appear. Earlier version in Proc. ACM MOBICOM 1999.)

5. Adjie-Winoto, W., E. Schwartz, E., H. Balakrishnan, and J. Lilley, "The design and implementation of an intentional naming system," *Proceedings of the 17th ACM Symposium on Operating Systems Principles (SOSP),* Kiawah Island, SC, USA, December 1999.

Five Other Selected Publications

1. Andersen, D. G., H. Balakrishnan, M. F. Kaashoek, and R. T. Morris, "Resilient Overlay Networks," *Proceedings of the 18th ACM Symposium on Operating Systems Principles (SOSP),* Banff, Canada, October 2001.

2. Chen, B., K. Jamieson, H. Balakrishnan, and R. T. Morris, "Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks," *Proceedings of the ACM MOBICOM Conference,* Rome, Italy, July 2001.

3. Snoeren, A. C. and H. Balakrishnan, "An End-to-End Approach to Host Mobility," *Proceedings of the 6th ACM MOBICOM Conference,* Boston, MA, August 2000.

4. H. Balakrishnan, H. S. Rahul, and S. Seshan, "An Integrated Congestion Management Architecture for Internet Hosts," *Proceedings of the ACM SIGCOMM '99 Conference,* Cambridge, MA, USA, September 1999.

5. Balakrishnan, H., V. N. Padmanabhan, S. Seshan, and R. H. Katz, "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links," *IEEE/ACM Transactions on Networking,* December 1997. (Earlier version in ACM SIGCOMM '96.)

Synergistic Activities

1. New MIT graduate course on computer networks, with heavy emphasis on teaching students to do research. Over ten published papers have resulted from the work of students in this course in the past three years.

2. Systems developed from past research being used by researchers in the networking community: Snoop protocol for wireless TCP, Congestion Manager (NSF-funded), Cricket (Version 1) indoor prototype.

3. Program committee member for ACM MOBICOM, ACM SIGCOMM, ACM SOSP in past years.

4. Program Co-chair for ACM SIGCOMM 2002.

Awards and Honors

Award	Date
Ruth and Joel Spira Award for Distinguished Teaching, MIT	2001
Award paper, 8th IEEE Hot Topics in Operating Systems Workshop	2001
Award paper, 6th ACM MOBICOM Conf.	2000
ACM Doctoral Dissertation Award	1998
C.V. Ramamoorthy Distinguished Research Award, U.C. Berkeley	1998
Okawa Foundation Fellow	1996
Award paper, 1st ACM MOBICOM Conf.	1995
Award paper, Winter Usenix Technical Conf.	1995

Recent Collaborators

Anantha Chandrakasan (MIT), Erik Demaine (MIT), Sally Floyd (ACIRI/ICSI), David Gifford (MIT), John Guttag (MIT), Frans Kaashoek (MIT), David Karger (MIT), Randy Katz (UC Berkeley), Robert Morris (MIT), Venkata Padmanabhan (Microsoft), Srinivasan Seshan (CMU), Scott Shenker (ACIRI/ICSI), Ion Stoica (UC Berkeley), Mike Stonebraker (MIT), Seth Teller (MIT)

Graduate Advisor

Randy Katz (UC Berkeley)

Graduate Students (Completed)

Wendi Beth Heinzelman (Univ. of Rochester)

Erik D. Demaine

Professional Preparation

School	Degree	Date
University of Waterloo	Doctor of Philosophy	2001
University of Waterloo	Master of Mathematics	1996
Dalhousie University	Bachelor of Science	1995

Appointments

Organization	Position	Date
Massachusetts Institute of Technology	Assistant Professor	2001–present
University of Waterloo	Instructor	1999

Five Publications Relevant to Proposal

1. R. Connelly, E. D. Demaine, and G. Rote, "Straightening Polygonal Arcs and Convexifying Polygonal Cycles," *Discrete & Computational Geometry*, to appear. Preliminary version in FOCS 2000.

2. T. Biedl, M. Demaine, S. Lazard, A. Lubiw, J. O'Rourke, M. Overmars, S. Robbins, I. Streinu, G. Toussaint, and S. Whitesides, "Locked and Unlocked Polygonal Chains in Three Dimensions" *Discrete & Computational Geometry*, 26(3):283–287, 2001. Preliminary version in SODA'99.

3. M. A. Bender, E. D. Demaine, and M. Farach-Colton, "Cache-Oblivious B-Trees," in *Proc. 41st Annual Symposium on Foundations of Computer Science*, November 2000, pages 399–409.

4. T. C. Biedl, P. Bose, E. D. Demaine, and A. Lubiw, "Efficient Algorithms for Petersen's Matching Theorem," *Journal of Algorithms*, to appear. Special issue of selected papers from SODA'99.

5. E. D. Demaine, I. Foster, C. Kesselman, and M. Snir, "Generalized Communicators in the Message Passing Interface," *IEEE Transactions on Parallel and Distributed Systems*, to appear.

Five Other Selected Publications

1. E. D. Demaine, A. López-Ortiz, J. I. Munro, "Experiments on Adaptive Set Intersections for Text Retrieval Systems," in *Proceedings of the 3rd Workshop on Algorithm Engineering and Experiments*, Lecture Notes in Computer Science, January 2001.

2. E. M. Arkin, M. A. Bender, E. D. Demaine, S. P. Fekete, J. S. B. Mitchell, and S. Sethia, "Optimal Covering Tours with Turn Costs," in *Proc. 12th Annual ACM-SIAM Symposium on Discrete Algorithms*, January 2001, pages 138–147.

3. E. D. Demaine and A. López-Ortiz, "A Linear Lower Bound on Index Size for Text Retrieval," in *Proc. 12th Annual ACM-SIAM Symposium on Discrete Algorithms*, January 2001, pages 289–294. Invited to a special issue of *Journal of Algorithms*.

4. E. D. Demaine, A. López-Ortiz, and J. I. Munro, "Adaptive Set Intersections, Unions, and Differences," in *Proc. 11th Annual ACM-SIAM Symposium on Discrete Algorithms*, January 2000, pages 743–752.

5. P. Bose, A. Brodnik, S. Carlsson, E. D. Demaine, R. Fleischer, A. López-Ortiz, P. Morin, J. I. Munro, "Online Routing in Convex Subdivisions," in *Proc. 11th Annual International Symposium on Algorithms and Computation*, Lecture Notes in Computer Science, volume 1969, December 2000, pages 47–59. Invited to a special issue of *International Journal of Computational Geometry and Applications*.

Synergistic Activities

1. Co-organizer of biweekly algorithmic open-problem solving session, enabling a major form of collaboration between faculty and students and exposure of students to research, University of Waterloo (1999–2001)

2. Conference organization: DIMACS Workshop on Folding and Unfolding (Oct. 2002); Symposium on Robot Arm Manipulation, AAAS Meeting (Feb. 2002); Seminar on Algorithmic Combinatorial Game Theory, Schloss Dagstuhl (Germany, Feb. 2002); 13th Canadian Conference on Computational Geometry (Waterloo, Canada, Aug. 2001),

3. Conference program committee: 13th Annual ACM-SIAM Symposium on Discrete Algorithms (San Francisco, California, January 2002); 13th Canadian Conference on Computational Geometry (Waterloo, Canada, August 2001).

4. Exposing research to the community. In the past 2 years, I have given plenary talks at 5 conferences, invited talks at 11 institutions, invited talks at 12 conferences and workshops, as well as 19 contributed talks at conferences.

Recent Collaborators

Oswin Aichholzer (TU Graz), Esther Arkin (SUNY Stony Brook), Michael Bender (SUNY Stony Brook), Hari Balakrishnan (MIT), David Benoit (InfoInteractive Inc.), Marshall Bern (Xerox PARC), Therese Biedl (U. Waterloo), Prosenjit Bose (Carleton U.), Broňa Brejova (U. Waterloo), David Bremner (U. New Brunswick, Fredericton), Andrej Brodnik (Luleøa Technical U.), Svante Carlsson (Luleøa Technical U.), Eowyn Cenek (U. Waterloo), Timothy Chan (U. Waterloo), Robert Connelly (Cornell U.), Carmen Cortés (U. Sevilla), Martin Demaine (MIT), Christian Duncan (U. Miami), Vida Dujmović (McGill U.), David Eppstein (U. California, Irvine), Jeff Erickson (U. Illinois, Urbana-Champaign), Martin Farach-Colton (Google Inc.), Sándor Fekete (TU Berlin), Rudolf Fleischer (Hong Kong U. Science and Technology), Greg Frederickson (Purdue U.), Erich Friedman (Stetson U.), Ian Foster (U. Chicago), Mordecai Golin (Hong Kong U. Science and Technology), Angèle Hamel (Laurier U.), George Hart (georgehart.com), Barry Hayes (PlaceWare Inc.), Michael Hoffmann (ETH Zurich), Ferran Hurtado (Polytecnic U. Catalunya), Lars Jacobsen (U. Southern Denmark), Craig Kaplan (U. Washington), Carl Kesselman (U. Southern California), Stephen Kobourov (U. Arizona), Eric Kuo (U. California, Berkeley), Stefan Langerman (McGill U.), Sylvain Lazard (INRIA Lorraine), Alejandro López-Ortiz (U. Waterloo), Anna Lubiw (U. Waterloo), Andrea Mantler (U. North Carolina, Chapel Hill), Henk Meijer (Queens U.), Joseph Mitchell (SUNY Stony Brook), Pat Morin (McGill U.), Ian Munro (U. Waterloo), Joseph O'Rourke (Smith College), Mark Overmars (Utrecht U.), Belén Palop (U. Rey Juan Carlos), Irena Pashchenko (Stanford U.), Venkatesh Raman (Inst. Mathematical Sciences), Suneeta Ramaswami (Rutgers U.), Steven Robbins (McGill U.), Günter Rote (FU Berlin), Vera Sacristán (Polytecnic U. Catalunya), Robert Sedgewick (Princeton U.), Saurabh Sethia (Oregon State U.), Steven Skiena (SUNY Stony Brook), Marc Snir (IBM Research), Jack Snoeyink (U. North Carolina, Chapel Hill), Michael Soss (McGill U.), Michael Stonebraker (MIT), Ileana Streinu (Smith College), Seth Teller (MIT), Godfried Toussaint (McGill U.), Helena Verrill (Hannover U.), Tomáš Vinař (U. Waterloo), Ming-wei Wang (U. Waterloo), Sue Whitesides (McGill U.).

Graduate Advisors

Anna Lubiw and Ian Munro (Ph.D.); David Taylor (M.Math.).

Graduate Students

1. Nicole Immorlica, Ph.D. candidate at MIT (tentative).

Michael Stonebraker

Professional Preparation

School	Degree	Date
University of Michigan	Doctor of Philosophy	1971
Princeton University	Bachelor of Science and Electrical Engineering	1961

Appointments

Organization	Position	Date
Massachusetts Inst. of Technology	Senior Lecturer	2001–present
Required Technology, Inc.	Chief Technology Officer	2001–present
Cohera Corporation	Founder and Chief Technology Officer	1997–2001
Informix Corporation	Chief Technology Officer	1996-2000
Univ. of California, Berkeley	Professor of the Graduate School	1994–1999
Illustra Corporation	Founder and Chief Technology Officer	1992-1996
Univ. of California, Berkeley	Professor	1982–1994
Ingres Corporation	Founder and Chief Technology Officer	1980-1992
Univ. of California, Berkeley	Associate Professor	1976–1982
Univ. of California, Berkeley	Assistant Professor	1971–1976

Vivant! Corporation, Electric Knowledge Corporation, PowerMarket Corporation, FiveNine Solutions, Inc.
Member of the Technical Advisory Committee

Five Publications Relevant to Proposal

1. Michael Stonebraker, Paul M. Aoki, Witold Litwin, Avi Pfeffer, Adam Sah, Jeff Sidell, Carl Staelin, Andrew Yu, "Mariposa: A Wide-Area Distributed Database System," *VLDB Journal* 5(1):48–63, 1996.

2. Michael Stonebraker, Paul M. Aoki, Robert Devine, Witold Litwin, Michael A. Olson, "Mariposa: A New Architecture for Distributed Data," ICDE 1994, pp. 54–65.

3. Chris Olston, Allison Woodruff, Alexander Aiken, Michael Chu, Vuk Ercegovac, Mark Lin, Mybrid Spalding, Michael Stonebraker, "DataSplash," SIGMOD 1998, pp. 550–552.

4. Alexander Aiken, Jolly Chen, Michael Stonebraker, Allison Woodruff, "Tioga-2: A Direct Manipulation Database Visualization Environment," ICDE 1996, pp. 208–217.

5. Michael Stonebraker, Jolly Chen, Nobuko Nathan, Caroline Paxson, Jiang Wu, "Tioga: Providing Data Management Support for Scientific Visualization Applications," VLDB 1993, pp. 25–38.

Five Other Selected Publications

1. Allison Woodruff, Michael Stonebraker, "Supporting Fine-grained Data Lineage in a Database Visualization Environment," ICDE 1997, pp. 91-102.

2. Michael Stonebraker, "The Integration of Rule Systems and Database Systems," *IEEE Transactions on Knowledge and Data Engineering* 4(5):415–423, 1992.

3. Michael Stonebraker, Greg Kemnitz, "The Postgres Next Generation Database Management System," *Communications of the ACM* 34(10):78–92, 1991.

4. Michael Stonebraker, Anant Jhingran, Jeffrey Goh, Spyros Potamianos, "On Rules, Procedures, Caching and Views in Data Base Systems," SIGMOD 1990, pp. 281-290.

5. Margo I. Seltzer, Michael Stonebraker, "Transaction Support in Read Optimizied and Write Optimized File Systems," VLDB 1990, pp. 174–185.

Synergistic Activities

1. Wrote several major public domain DBMS prototypes, including Ingres and POSTGRESQL. The latter is currently widely used as a teaching instrument in universities and forms the basic for several commercial DBMS products.

2. Have founded the New England Database Symposium (NEDS) as a mechanism for Boston-area DBMS researchers to network and collaborate.

Awards and Honors

Award	Date
ACM Software System Award	1988
ACM SIGMOD Innovation Award	1992
National Academy of Engineering	1998
Forbes Magazine – named one of 8 innovators driving the Silicon Valley wealth explosion in 80th anniversary issue	1998
Computer Reseller News – Hall of Fame Member	1999

Recent Collaborators

Hari Balakrishnan (MIT), Mitch Cherniak (Brandeis), Erik Demaine (MIT), Joey Hellerstein (Berkeley), Betty Salzberg (Northeastern), Seth Teller (MIT), Stan Zdonik (Brown).

Graduate Advisor

Dr. Arch Naylor (Michigan, now retired)

Graduate Students

Sample of more than 20 completed PhD students: Gerald Held (until recently VP/Eng'g of Oracle Corp.), Robert Epstein (founder and until recently VP/Eng'g of Sybase Corp.), Paula Hawthorn (until recently VP/Eng'g of Informix Corp.), Dale Skein (founder and CTO of Vitria Corp.), Mike Carey (until recently MTS at IBM Almadin Research Laboratory), Margo Seltzer (Assoc. Professor of Computer Science at Harvard), Anant Jhingran (Manager, Database Department of IBM Almaden Research Laboratory).

Seth Teller

Professional Preparation

School	Degree	Date
University of California, Berkeley	Doctor of Philosophy, Computer Science	1992
University of California, Berkeley	Master of Science, Computer Science	1990
Wesleyan University, Middletown CT	Bachelor of Arts, Physics	1985

Appointments

Organization	Position	Date
Massachusetts Inst. of Technology	Associate Professor	1998 – present
Massachusetts Inst. of Technology	Assistant Professor	1994 – 1998
Princeton University	Post-Doctoral Researcher	1993 – 1994
Hebrew University of Jerusalem	Post-Doctoral Researcher	1992 – 1993
Univ. of California, Berkeley	Research Assistant	1988 – 1992

Five Publications Relevant to Proposal

1. Funkhouser, T., C. Séquin, and S. Teller, Management of large amounts of data in interactive building walkthroughs. In *Proc. 1992 Workshop on Interactive 3D Graphics*, Boston, MA, March 1992, pp. 11–20.

2. Funkhouser, T., D. Khorramabadi, C. Séquin, and S. Teller, The UCB System for Interactive Visualization of Large Architectural Models. In *Presence*, 5(1):13–44, Winter 1996.

3. Capps, M., and S. Teller, Communications Visibility in Shared Virtual Worlds. In *Proc.* 6^{th} *Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises*, June 1997.

4. Priyantha, N. B., A. K. L. Miu, H. Balakrishnan, and S. Teller, “The Cricket Compass for Context-Aware Mobile Applications,” In *Proc. ACM MOBICOM Conference*, Rome, Italy, July 2001.

5. Bukowski, R., L. Downs, M. Simmons, C. Séquin, and S. Teller, Citywalk: A Second Generation Walkthrough System, In *Proc.* 7^{th} *International Conference on Virtual Systems and Multimedia (VSMM)*, Berkeley, CA, October 2001.

Five Other Selected Publications

1. Teller, S., and C. Séquin, Visibility preprocessing for interactive walkthroughs. In *Computer Graphics (Proc. SIGGRAPH ’91)*, 25(4):61–69, 1991.

2. Teller, S., Fowler, C., T. Funkhouser, and P. Hanrahan, Partitioning and ordering large radiosity computations. In *Computer Graphics (Proc. SIGGRAPH ’94)*, 28:443–450, 1994.

3. Teller, S., N. Boyd, B. Porter, and N. Tornow, Distributed Development and Teaching of Algorithmic Concepts. In *Computer Graphics (Proc. SIGGRAPH ’98 Educational Track)*, July 1998.

4. Coorg, S., and S. Teller, Temporally Coherent Conservative Visibility. In *Computational Geometry: Theory and Applications*, 12(1-2):105–124, February 1999.

5. Antone, M., and S. Teller, Scalable, Absolute Registration of Omni-Directional Image Networks, In the *International Journal of Computer Vision* (to appear).

Synergistic Activities

1. Teaching efforts include extensive development of use of geometric visualization and algorithm inspection tools.

2. Research efforts include an ambitious project to achieve automated model capture capability for architectural environments. The project seeks to generate “reference models” by procedural generation of 3D CAD models from incompletely-specified 2D CAD. Such reference data can also be useful for the location infrastructure described in the accompanying proposal.

3. Program committee member for SoCG (2000) and SIGGRAPH (2002).

4. Aggressive recruiting and retention of women and under-represented minority students for UROP (Undergraduate Research Opportunity Program), AUP (Advanced Undergraduate Project), MEng (Master of Engineering), and PhD positions.

5. Founded the MIT Computer Graphics Colloquium as a mechanism for invited speakers in the field to give talks to students and researchers at MIT.

Awards and Honors

Award	Date
Samuel Silver Research Award, UC Berkeley	1992

Recent Collaborators

Hari Balakrishnan (MIT), Erik Demaine (MIT), David Dobkin (Princeton), Laura Downs (UC Berkeley), Celeste Fowler (SGI), Michael Hohmeyer (Berkeley, ICEM CFD), Tom Funkhouser (AT&T Labs, Princeton), Pat Hanrahan (Stanford), Tomás Lozano-Pérez (MIT), Carlo Séquin (UC Berkeley), Maryann Simmons (UC Berkeley), Mike Stonebraker (MIT)

Graduate Advisor

Carlo Séquin (UC Berkeley)

Graduate Students (Completed)

Satyan Coorg (Liberate), John Mellor (Rose-Hulman Inst. of Tech.), Kavita Bala (Cornell), George Chou, Matthew Antone (Consultant)

H. Facilities, Equipment and Other Resources

The MIT Laboratory for Computer Science (LCS) has several PCs donated by Intel and has, under other federally funded grants, purchased many PCs. A campus-wide 802.11 wireless network is in place at MIT. MIT LCS has ongoing relationships with several companies doing PCB layout and assembly for sensor production and fabrication and to undertake large-scale manufacture of Cricket beacons and listeners.

LCS provides administrative services for all principal investigators who submit proposals through LCS. These administrative services are run by the Headquarters Staff and include Fiscal, Personnel, Library, Facilities and other LCS operations.

These services are supported by an Allocated Project Level Cost. Allocated costs are assessed against all contracts and grants at LCS. The costs are determined as percentages of direct salaries and materials and services. These percentages are currently 7.1% for Personnel and 1.4% for Materials and Services. We constantly review these services to make reductions whenever possible.

The Personnel (or Lab Support) Allocation covers the costs of those individuals who provide LCS administrative services such as accounting, budget preparation, personnel, payroll, etc. This is calculated by multiplying the secondary MTDC Base by 7.1%.

The Materials & Services Allocation covers the cost of phones, office/computer supplies, etc., for the same administrative personnel whose salaries are covered under the personnel support. This is calculated by multiplying the secondary MTDC Base by 1.4%.