Sensorium

Research Infrastructure for Managing Spatio-Temporal Objects in Video Sensor Networks

Proposal Description

C.1 Introduction

In recent years, there has been an increase in the deployment of digital video sensors in public structures. In addition, there are a number of large-scale research projects that propose innovative approaches for the widespread use of video sensors in intelligent rooms [315], and homes [173], instrumented classrooms [5], roadways [76, 120, 121, 177, 129, 147, 149, 152, 292, 305], and vehicles [11, 13, 115, 128, 136, 232, 293] that observe the driver and/or watch for moving objects in the vehicle path. Furthermore, there has been an expanding research effort in the area of visual surveillance of human activity by computer [79, 80, 109, 127, 137, 154, 160, 168, 196, 219, 224, 225, 236, 246, 255, 256, 259, 319]. What unifies these applications is their use of distributed video sensors, which can either be mobile or fixed in a space.

Based on these trends, it is reasonable to expect that inexpensive smart video cameras will be embedded in most machines and buildings within the next decade, and it is likely that these sensors will be bundled with computers and connected to the network. In many ways, this trend could prove beneficial to society, in that information collected by sensors could be shared for the better good. Such benefit is especially clear in the case of assistive spaces; e.g., classrooms, offices, or homes for the elderly or handicapped. However, an ubiquitous network of smart sensors would present critical challenges in computer science, particularly in the areas of computer vision, databases, networking, resource management, and cryptographic security and privacy. The proposed research effort aims to make fundamental contributions towards addressing these issues.

Harnessing the power of these environments will hinge on our ability to build applications capable of gathering, interpreting and storing data from distributed sensors and to provide scalable mechanisms for managing the networks and systems resources that these applications consume. We propose to address these challenges within a research infrastructure composed of a sensor network of video cameras spanning several rooms, networked processing units and a terabyte database which are managed together to satisfy user queries including those generated by mobile users within this environment. Up until now, methods for addressing scheduling, resource management, and network protocols to support sensor networks have been mostly implementation-specific. Furthermore, issues of cryptographic security and ensuring personal privacy have been essentially ignored, since the issues of tracking and recognizing human identity and activity are such challenging problems unto themselves.[1] But as the number of embedded sensors increases, and the state of the art in tracking algorithms improves, the lack of a generic framework to address the needs of embedded sensor networks will become critical. Our research effort, which aims to develop such a framework, is an integrated project involving researchers with primary research interests spanning core disciplines of computer science including computer vision, databases, networking, real-time systems and security. We envision the Sensorium research infrastructure as fertile ground for fostering synergistic collaborations between research groups that have not previously collaborated within our department and for facilitating new research directions within groups that would not be possible otherwise.

System Configuration and Overall Goals: Sensors in the Sensorium infrastructure are fixed video cameras with pan/tilt/zoom capability. At any instant in time, we refer to the finite spatial extent visible to a sensor as that sensor's *view volume*; objects within that volume are recorded at a resolution which can vary due to pixel and frame rate sampling. For a group of video cameras, the union of view volumes define a *domain volume* for a monitored room or public space. Each video camera is attached to PCs for processing and indexing sensory information. A user with her mobile unit can ask the Sensorium to perform high-level, automated sensing tasks, such as locating and tracking a human moving through the Sensorium. Queries can be sent either through a wireless access point or through another user within communication range (as in *ad-hoc* networks). In response to user queries, or in cooperation with other sensors, cameras perform basic operations such as adjusting their resolution or altering their view volumes. Responding to a single such query is already a challenging research problem from a networking and computer vision standpoint, but we are interested in an even broader challenge: Designing a scalable Sensorium system architecture capable of handling concurrent queries of this form which will in general place competing demands on systems resources.

[1] This is elaborated upon in the letter of support from members of the Privacy Foundation.

In contrast to current sensory systems which are conceived (and built) as *special purpose* systems with custom-developed architectures, we intend to build our architecture from commodity hardware and with freely available software, opening up much wider access to sensory systems to the computer science research community. However, our desire to develop sensory commodity system for a broad class of environments also highlights some of the current impediments to building such a system. Among these, we feel that multi-resolution encoding of sensory data and the awareness of the system of spatio-temporal constraints at every level of its architecture will be the critical core technologies. In addition, we feel that there are key supporting technologies that will need to be developed to efficiently use available resources.

C.2 Catalyst Sensorium Applications

In this section we present two "vertical" applications that we plan to target. The seeds of these applications are already in place as evident in existing projects pursued by the PIs. It is important to emphasize that the development of these applications is not a goal, but a catalyst for the development of generic technologies that could be used (and reused) in *multiple* vertical applications.

C.2.1 RealityWeb: Weaving Physical Spaces into Cyber Space

Objects accessed through today's "Cyber World," i.e., "Web," are virtual. They can be controlled and replicated. They are served through well-defined procedures. Each object has a name, the Uniform Resource Locator (URL) that allows its retrieval. On the other hand, objects in the "Real World" are physical. They can only be accessed through sensory means (e.g., web-cams that monitor physical spaces and transmit live video over the internet). Currently, objects in these physical spaces can only be *passively observed* through the Web. The proposed SENSORIUMwill enable an *active understanding* of the physical world into the cyber world. It will give physical objects, which include things and places, as well as people and their activities, unique identities. The object identity is not simply linked to a physical space and accessed with the web-cam's URL, but it instead has its own "Uniform Resource Identity" (URI), which can be used to search for the object in all physical spaces that are accessible through the Web. The object can then be uniquely retrieved, and actively monitored and tracked.

The existing infrastructure of web-cams is the result of an explosive and ad-hoc growth of camera installations all over the world. In addition to web-cams, a vast number of digital video cameras have been deployed for surveillance purposes, for example, of stores, ATM machines, airports, and other public and commercial facilities. These cameras are untapped resources that provide the opportunity to merge the physical and cyber worlds in an integrated, well-defined, and privacy-protecting manner. There are also large-scale research projects that propose the use of video sensors in intelligent rooms, homes, instrumented classrooms, roadways, and vehicles. In addition, there has been extensive work in the area of visual surveillance of human activity. Most of these projects focus on the computer vision aspects of the problem and disregard the complex networking, systems, and data management issues. None have tried to embed the monitored physical space into the Web. Similarly, much of the research in sensor networking has abstracted away specific application-level issues that arise in this context.

At any instant in time, we refer to the finite spatial extent visible to a particular video camera as that sensor's *view volume*; objects within that volume are recorded at a resolution which can vary due to pixel and frame rate sampling. For a group of video cameras, the union of view volumes can be used to define a *domain volume* for a monitored room or public space. Each video camera is attached to PCs for processing and indexing sensory information. A user with her mobile unit can ask the SENSORIUM to perform high-level, automated sensing tasks, such as locating and tracking a human moving through the SENSORIUM. Queries can be sent over a backbone network either through a wireless access point or through another user within communication range (as in *ad-hoc* networks). In response to user queries, or in cooperation with other sensors, cameras perform basic operations such as adjusting their resolution or altering their view volume. Responding to a single such query is already a challenging research problem from a networking and computer vision standpoint, but we are interested in an even broader challenge: Designing a scalable SENSORIUM system architecture that could be easily reproduced to create a *Web of domain volumes* (e.g. various rooms in various buildings on campus) that comprise a "REALITYWEB" accessible to users through an appropriate REALITYWEB Browser.

To build the REALITYWEB, we will develop vision, database, and network services that are capable of gathering, interpreting, routing, and storing data from distributed video sensors. These services will be used to answer queries about the physical world on the Web—in other words, enable us to "surf the physical world".

In contrast to current sensory systems which are conceived (and built) as *special purpose* systems with custom-developed architectures, we intend to build our architecture from commodity hardware and with freely

available software, opening up much wider access to sensory systems to the computer science research community. However, our desire to develop sensory commodity system for a broad class of environments also highlights some of the current impediments to building such a system. Among these, we believe that (1) multi-resolution encoding of sensory data, and (2) cognizance of spatio-temporal and resource constraints at every level of a system's architecture will be critical core technologies. These two threads are common to all Sensorium research projects supporing the RealityWeb.

C.2.2 The Sensorium as an Environment Supporting People with Disabilities

This vertical application aims to help people with severe disabilities to communicate with sensor-rich environments. People with severe disabilities (e.g., cerebral palsy or traumatic brain injuries) cannot speak, have only limited voluntary motions, and so have difficulty controlling their environment and communicating with family, friends, and care providers. To help facilitate communication in this setting, we will explore new methods for real-time automatic and robust detection, tracking, and interpretation of human body components and their motion in video under normal lighting conditions. The goal is to recognize the movements of, for example, a finger, foot, facial feature, or wheelchair in real time and interpret them as a message. Messages may be directed to the care provider to request water or simply to express joy; or they may control the person's environment, for example, select web pages, switch off the TV, or open a door. The advantage of video-based assistive technology is that it is preferred by subjects with disabilities, because it is more comfortable to use than user-borne accessories, such as helmets or tongue-activated switches.

The choice of body feature used for communication will depend on the specific user's physical skills. We will develop a mechanism for automatic customization of both video-interpretation algorithms and augmentative communication software in order to meet the user's specific needs. Different body features will result in different detection, tracking, and interpretation performance. The proposed system has the potential to support a high communication rate and result in less perceived exertion, since it will focus on the motions that a person can make most comfortably.

Various techniques have been developed by the computer-vision research community with the intention to detect and track people without disabilities and analyze their gestures, actions, and behaviors. Although these interpretation tasks are very difficult, a number of techniques, including our own work, have been successful or show great promise and will be used in the proposed project as starting points for new algorithms. This research requires a set of synchronized cameras so that the human can be imaged from different viewpoints, represented by a 3D model, and his or her motion tracked and interpreted in real time. To that end, existing computer vision research (e.g. at CMU, MIT, UMD, and Georgia Tech) on monitoring human activities using video sensor networks has had little interaction with the systems and networks communities, although there are many quality-of-service issues that need to be resolved.

Consider the example of a girl whose voluntary head motion is limited to slowly shaking her head and with whom one of the PIs has been working for over a year now. The girl can stir her wheelchair by moving her head and touching a mechanical control. If our research is successful, we will be able to automatically detect and track her with video while she is moving through a room and heading for a door. The vision algorithms will control the door, which will open automatically for her. The girl's father may be in a different room and will be able to monitor his daughter on a computer screen. After passing through the door, the girl may stop in front of a computer screen. Cameras will analyze her facial movements and interpret them as a message to start a spelling program for sending email to the girl's mother.

There are many computer vision tasks in this example and each has a set of quality-of-service constraints that need to be achieved. The processing of the video from each camera can be modelled as a periodic task consisting of several subtasks. We will determine the relative priorities of these tasks. For example, we need to make sure that the hand-off between cameras in the two rooms must be performed in a timely manner. Once the girl is detected in front of the computer, we must assure that the surrounding cameras zoom in on the face so that the spelling program is activated and the facial features can be modeled in 3D at a high level of resolution. We may want to assign a significant fraction of computational resources to support the spelling task and less to the transmitting of video to the monitoring parent. However, if the vision system detects that the child may be choking, the resources should be reallocated automatically to alert the monitoring parent, enabling him to view the child at a high resolution so that he can react quickly and appropriately.

The seeds of this vertical application are already in place [130, 131]; however, our current research is restricted to single-camera computer user interfaces and its real-time features are guaranteed through laborious and ad-hoc code tweaking. The availability of the Sensorium will take our effort to a different level by enabling us to monitor a wider range of human activities with multiple networked cameras and at multiple levels of detail, supported by networking and real-time resource management techniques [15, 19, 309, 310] for seamless real-time performance.

C.3 Core Projects and Technologies Enabled Through the Infrastructure

We present specific projects in a number of core CS research areas that will directly benefit from the acquisition of the Sensorium. In the next section, the symbiotic, nature of these projects will be discussed.

C.3.1 Computer Vision Projects – Betke and Sclaroff

The technical challenges associated with developing tools to track humans in the Sensorium are modeling, interpreting, and predicting human motion in video streams obtained by a collection of interacting cameras, where each camera can provide different spatial and temporal information. To be efficient, accurate, and robust, algorithms must employ multiple scales in space and time as well as multiple layers of detail. Projects to be pursued will focus on two core research problems associated with the Sensorium: (1) Estimating and encoding humans and their motion at multiple levels of detail, and (2) Determining correspondence of the same person seen in different camera views based on the color distribution of clothing, body motion and pose, and facial appearance. The developed methods will then be used for model-based data compression, resource prediction in the Sensorium, and indexing of spatio-temporal databases of observed human activities.

Estimating, Predicting, Recognizing, and Encoding Humans in Motion: The motion of an individual human can be represented at many different levels of detail, for example, by a moving blob of pixels extracted from each image, the human's overall 3D-motion trajectory, gross 3D motion of his or her body parts, or detailed parameterizations of facial expressions. We therefore propose to develop *layered encodings* of human motion that enable multiple levels of space/time compression and accuracy; for example, the first layer provides greatest compression and lowest accuracy, the next layer, when added to the first, provides increased accuracy, etc. One possible layered encoding of human motion would be:

- Layer 1: Position and orientation of the moving object over time.
- Layer 2: Actions performed, e.g., running, sitting, jumping.
- Layer 3: Motion descriptions in terms of parameters of a generic *avatar*.[2]
- Layer 4: Detailed 3D geometric models including texture maps.

Encodings of human activity are not only needed for information transmission, fusion, or storage. Reliable trajectory estimation and prediction are also needed for scheduling of network and computational resources within the Sensorium. Trajectory estimation will support prediction of hand-off among sensors of moving people as well as prediction of visibility from a particular sensor's viewpoint. It will allow us to choose a camera viewpoint for zooming in on people's faces, should we want to identify them. Finally, trajectory information will aid in classifying motions of individuals to support database queries and data mining, as will be described in Section C.3.2.

There has been a good deal of research effort in tracking and predicting human motion as observed from a video camera, e.g., [30, 61, 66, 70, 79, 80, 108, 109, 112, 127, 137, 150, 154, 160, 168, 169, 189, 196, 198, 219, 224, 225, 231, 235, 236, 246, 255, 256, 258, 259, 270, 280, 315, 316, 318, 317, 319]. Low-level image processing methods have been shown to work surprisingly well when a simplified version of the general problem is assumed; e.g., there is only one moving object, objects do not occlude each other, or objects appear at a limited range of scales and orientations. While higher-level, 3D model-based techniques can address some of these limitations [169, 231, 258], such methods require guidance by a human operator, and are therefore not fully automatic.

Another problem is occlusion, which can occur when another object partially (or completely) obstructs a camera's view of the human; e.g., two people's paths cross in the image yielding a temporary occlusion of one person from that camera's viewpoint. Therefore, any general system must include algorithms that can reliably detect and maintain the tracking of moving objects before, during, and after an occlusion.

Keeping these issues in mind, we propose to take the following approach to layered encoding of human activity in the Sensorium. To recover a representation for layer 1, we will develop new algorithms for tracking and predicting human motion as viewed from multiple, pan/tilt/zoom video cameras. Promising results have been obtained for 3D trajectory estimation in monocular video using recursive estimation theory, in particular

[2]The term *avatar* is commonly used in virtual reality, when a computer graphics model is used to represent an actual human in cyberspace. In our context, an avatar model is just standard (generic) polyhedral human model that can be used to iconically and anonymously encode the human motion.

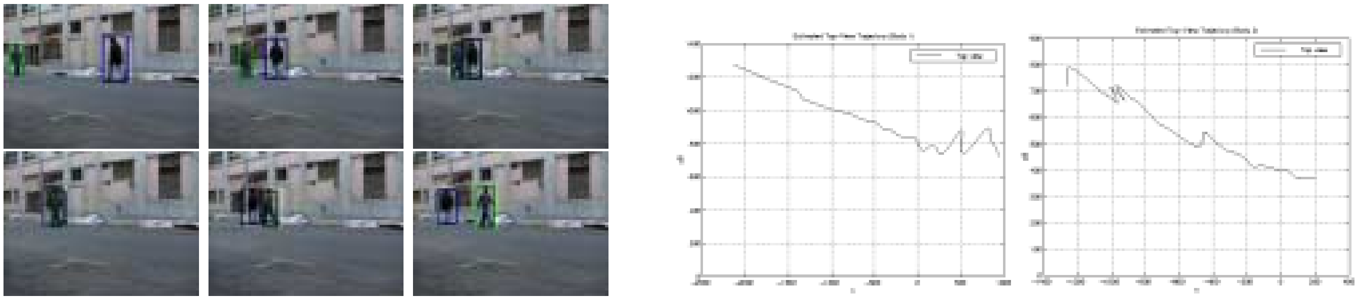
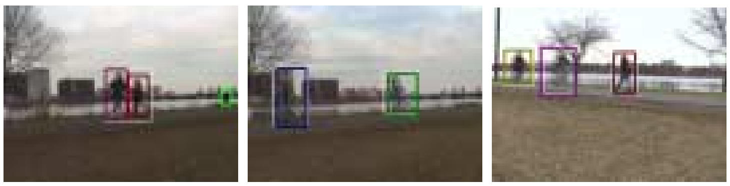


Figure C-1: Tracking example: 2 bodies walking in different trajectories, occluding each other. The estimated minimum bounding boxes (MBBs) for each moving object are shown overlaid on the input video images. The graphs show the motion trajectory recovered (top view) for the two moving bodies.



Actions	Walk	Run	R. blade	Bike	Totals
Walk	-	0.21	0.12	0.02	0.35
Run	0.19	-	0.08	0.01	0.28
R. blade	0.11	0.12	-	0.00	0.23
Bike	0.02	0.02	0.01	-	0.03
Totals	0.32	0.35	0.21	0.03	**0.27**

Figure C-2: Feasibility study: we collected sequences (3 hours total) of random people performing different actions on a pedestrian road in a outdoor environment. We trained the system to recognize four different actions (walking, running, roller blading, biking). Example frames from the sequences along with the confusion matrix for classification are shown.

the extended Kalman filter (EKF) [259]. Using this approach, we can reliably segment and maintain the tracking of moving objects before, during, and after occlusion (e.g., see Fig. C-1). In another promising approach, we have shown that normalized correlations of MBBs or blobs, and low-level edge and color processing, can be employed to detect and track multiple moving objects under challenging conditions [64, 67, 63, 62]. We utilized a hard real-time operating system called "Maruti," whose scheduling guarantees – prior to any execution – that the required deadlines are met and the vision system will react in a timely manner [266].

Three-dimensional trajectory, occlusion, and segmentation information are utilized in extracting stabilized views of the moving object that are then used as input to action coding modules for layer 2. To address coding actions, we have conducted feasibility studies with an approach that looks at motion patterns of each blob in the image plane [73, 111, 259] (e.g., see Fig. C-2). As part of the proposed research effort, this general approach will be employed to include encoding new actions and exploit information available in multiple views. Another important use for layer 2 encodings is in *motion mining*. As will be described in Section C.3.2, we would like to develop data mining methods that find common motion patterns through clustering of the Sensorium database. As part of this effort, we plan to develop ways to use the resulting clustered motion data in "retraining" the statistical pattern recognition algorithms to gain improved accuracy in estimating human motion and classification of human activity in video.

The avatar encoding of layer 3 requires estimating body posture and motion given video from one or more camera views. Most of the known techniques [30, 80, 127, 150, 168, 169, 189, 231, 255, 258, 315] require the use of controlled viewing conditions, and/or user initialization. The difficulty stems from the number of degrees of freedom in the human body, the complex underlying probability distribution, ambiguities in the projection of human motion onto the image plane, self-occlusion, scene clutter, insufficient temporal or spatial resolution, etc. Recently, we have developed a new method for estimating body posture and motion that has shown promise [263, 262]. The basic approach maps image features (e.g., moments) directly to likely 3D-body pose parameters via machine learning techniques. The system is trained once (off-line) with 3D-motion capture data. An example reconstruction of body posture using this method is shown in Fig. C-3. As part of the proposed research project, we will extend this approach to use multiple cameras (as needed) to gain more accurate estimates of avatar parameters.

Determining Correspondence of People Seen in Different Camera Views and/or at Different Times: Another key research problem is that of determining correspondence of the same person as seen in different camera views or at different times. Methods for determining correspondence of the same person seen in different camera views based on clothing color distribution, motion/pose, and trajectory patterns will be

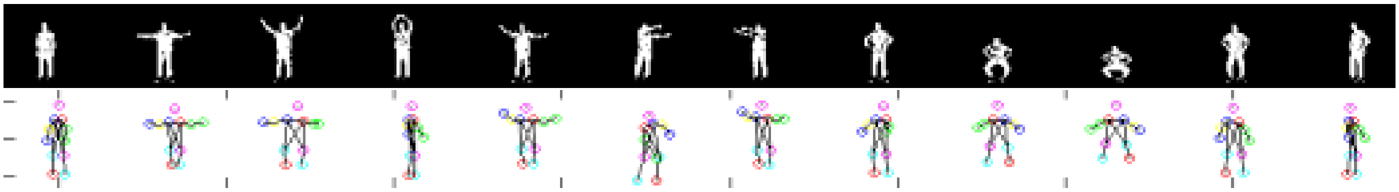


Figure C-3: Reconstruction of avatar parameters from a real video sequence (every 30th frame shown). The top row shows the blob segmentation for each frame used as input to the estimation algorithm. The second row shows the estimated body pose obtained, where the body is represented in terms of a simple skeletal model for the avatar.

developed. Matching different views of the same person is a critical first step in fusion of video streams to obtain detailed avatar models with texture maps for encoding layer 4.

To identify the same person in different but contemporaneous camera views, we can compute and match color histograms for each moving blob as seen contemporaneously from different camera viewpoints. This approach could be combined with matching views of the same activity or body pose (given encoded layers 2 and 3 as described above). Furthermore, we can combine information available in multiple contemporaneous views of the same person to gain improved estimates of body motion parameters via iterative refinement.

Identifying the same person seen at different times is challenging in general. People could change clothing, hair styles, illumination can change, etc. New methods are required for person identification via face, gait, or anthropometrics. Information about gait and/or anthropometrics can be obtained via data mining and/or a spatio-temporal database query. Face recognition and tracking methods can also be employed [32, 29, 106, 112, 162, 95, 214, 223, 265, 298]; however, an important additional component is needed: methods will be needed for tracking and zooming in on people with actively controlled cameras to gain close-ups of faces, selected from the best available, unobstructed camera viewpoint, along lines of [66, 216].

C.3.2 Database Systems Projects – Kollios

An important research challenge associated with Sensorium-like environments is the design of new access methods (indices) for efficient search and retrieval of the very large spatio-temporal datasets. Mining interesting patterns in these datasets can help better model the behavior of objects, classify new objects and predict future behavior. In the context of this project we plan to develop new index and data mining algorithms for efficient analysis and retrieval in large spatio-temporal datasets.

Spatio-temporal Indexing: Consider a spatio-temporal database system that stores sensory data. A user can connect to such a database through the network and submit queries that specify spatial and temporal predicates to retrieve all records that satisfy such predicates. An example is a selection query of the form "find all objects contained in a given area S of a monitored space at a give time t," or an aggregate query of the form "find how many objects passed through area S during the time interval T." If the time predicate refers to the past, then we call these queries *historical* queries. On the other hand, if the queries refer to current and future positions of objects, we call them *continuous* or *future* queries.

A simple approach to indexing spatio-temporal datasets for historical queries, is to consider time as another spatial dimension. Then, a dataset of 3-dimensional moving objects can be represented as a 4-dimensional dataset and a spatial access method (SAM) [268, 126] can be used for indexing. However, treating time as a spatial dimension is problematic due to different characteristics of time and space. An index based on this approach is usually very inefficient [291, 234, 178]. A better approach is to combine an efficient spatial index and the partially persistent methodology [117, 31, 300, 195, 297]. A data structure is called *persistent* if an update applied to it creates a new version of the data structure while the previous version is still retained and can be accessed. *Partial* persistence implies that all versions can be accessed but only the last version of the structure can be modified.

To use the partially persistent methodology, we have first to address the important issue of how to represent the moving objects into the index. Consider an object that moved from an initial to a final position during a time interval T. The simplest way to represent this object, is to compute the smallest bounding box that contains it at all time instants in T (see Sec. C.3.1). Note that the representation created by this approach can be very large due to object movement. We propose to use artificial updates and represent the object using more than one box. The idea is to partition the time interval T into smaller intervals and for each subinterval use a smaller bounding box to represent the object [178, 181]. The advantage of this approach is that we approximate the moving object with higher accuracy. The disadvantage is that we increase the size of the dataset and sometimes we need to access more disk blocks to answer a query. Therefore, the challenge is to find how many times and where to split the lifespan intervals of moving objects, in order to keep the space linear in the number of objects and decrease the average query time. To achieve that, we plan to develop an analytical model that describes the query performance of the partially persistent index. Then, a cost function based on this model will be used to find the best splitting points for each moving object.

For continuous queries, users submit queries about the current and future locations of objects (e.g. persons, GPS, wireless handheld). Therefore, the system needs to index not only the current state but also the future states of the database. Continuous movements pose formidable challenges [284, 98, 143]. In traditional database systems, data is assumed to remain unchanged unless it is explicitly updated. This model is appropriate when data changes in discrete steps, but it is inefficient for applications with continuously changing data, since it would require updating the database for each object at each time instant; clearly an inefficient and infeasible solution due to the prohibitively large update overheads. A better approach is to abstract each object's location as a function of time $f(t)$ and update the database only when one of the parameters of f change [284, 311, 86]. While this approach decreases the update overhead, it introduces a variety of problems since the database is not directly storing data values but functions to compute these values. In this project we will examine how to index these functions.

One approach is to map such functions to d-dimensional points and then use a multidimensional access method to index these points [180, 179, 7]. Range and nearest neighbor queries must also be transformed to the new data space. Since the shape of the queries changes under the transformation, we need to design new access methods to answer efficiently the new form of queries. Another approach is to parameterize a spatial index structure using velocity vectors and enable the index to be "viewed" at any future time [267]. Note that if many objects change direction/speed at the same time instant the update overhead can be significant. In that case, we need to develop index structures that can accommodate the high update rates and at the same time being available for answering queries. To achieve that we propose to use parallel and main memory data structures. We plan to examine how existing structures can become parallel and design new structures if necessary.

Mining Spatio-temporal Datasets: The goal of a data mining task is to efficiently find and describe structure and patterns in large datasets—patterns that were previously unknown and not stored explicitly in the database. Examples of data mining tasks include clustering [218, 78, 326, 138, 33, 148] and outlier detection [175, 176, 243, 81, 161].

In spatio-temporal datasets it is important to cluster objects based on motion information. That is, given a set of moving objects divide this set into groups of similar objects using their trajectories. The trajectory of an object is given as a sequence of 2 or 3-dimensional points, where each point represents the location of the object at a specific time instant. The first step is to define a similarity (distance) function between two trajectories. An approach is to use the sum of the Euclidean distance (or other p-norm distance) of the location of the objects at each time instant. However this approach has some drawbacks since it is not robust against outliers and small deviations. A similarity model for trajectories has to be simple to allow efficient computation of the similarity but at the same time it has to be expressive enough to capture intuitive similarity between two trajectories. In recent years, there has been extensive work in time series similarity models and retrieval [8, 9, 35, 77, 97, 144, 99, 119, 184, 134, 321, 185, 172, 170, 171, 229, 233, 240, 239, 110, 74] but most of these approaches concentrate on one dimensional time series (with the exceptions of [187, 320]). In this project we will investigate methods to extend models for one dimensional time series to trajectories of moving objects. Our recent results [182] suggest that using the longest common subsequence [9, 110] model we get more intuitive methods to define the similarity between trajectories.

Given the similarity function, we will design efficient and effective algorithms to cluster trajectory data. One approach is to select first a small set of representative trajectories and cluster them using a main memory algorithm. Another approach is to modify main memory algorithms to work efficiently for very large datasets (e.g., by using index methods to access disk resident data). Objects that do not belong to any cluster can sometimes provide useful knowledge about the dataset. We intend to design algorithms to efficiently find such outliers in a trajectory database.

C.3.3 Networking Projects – Byers, Crovella, and Matta

The Sensorium will provide ubiquitous connectivity and services to its users as they roam through the environment with their mobile access devices. A mobile user can send its queries and receive responses either through an access station (often referred to as *base station*), or through another user within communication range (as in *ad-hoc* networks). In such an environment, a number of challenging issues arise in the context of effectively delivering performance-sensitive streams (e.g. bandwidth-intensive video) to power-limited (battery-operated) devices over bandwidth-limited, error-prone wireless links. To this end, we define a range of activities that the network infrastructure would be required to perform. The key requirements that the network must enable are: distributed *coordination* across sensors, *efficient* transport of sensor data and *flexible* routing of information to mobile users. As examples of satisfying these requirements in practice, we briefly describe research challenges associated with three representative problems: sensor hand-off and object tracking, wireless transport and proxy services, and routing of sensor data.

Sensor Hand-off and Object Tracking: As an object moves through space-time, or as environmental variables such as lighting or occlusion change, the sensor with the best view of the object will change dynamically. Therefore, the responsibility of tracking the object will frequently have to be *handed off* from one sensor to another, or from one sensor domain to another, subject to resource management constraints. Leveraging from previously described work on estimating the velocity and direction of an object, hand-off to a new sensor or sensor domain can often be anticipated, ensuring high quality tracking [124]. Anticipating hand-offs will be especially important for stateful hand-off across sensor domain boundaries, an activity which can be optimized by prefetching object state and authenticating objects for hand-off prior to their arrival at domain boundaries. Such seamless hand-off is especially challenging in the presence of timing constraints (e.g. when an object is moving rapidly). Of course, protocols for efficient hand-off will also have to be carefully optimized to respond quickly to unanticipated or incorrectly predicted actions.

As tracked objects move around the Sensorium, one challenge is to quickly locate the target object [252]. Section C.4.2 outlines integrated activities to investigate techniques for efficient mobility management and object query/tracking.

Wireless Transport and Proxy Services: The transport of sensory data over wireless links subject to certain delivery requirements (e.g. in terms of loss rate, delay jitter) poses challenging problems due to the unreliability of the wireless link. Furthermore, this unreliability—in terms of high bit error rate—varies in time depending on the fading process. To improve its reliability, two classes of link-level error control are often used: automatic repeat request (ARQ) and forward error correction (FEC). Recent studies have shown that using a combination of ARQ and FEC gives better performance that ARQ or FEC alone [87, 191, 22, 167, 269, 257]. Hence, our investigations will focus on hybrid ARQ/FEC error control (also known as hybrid ARQ) deployed by access and mobile devices under different types of coding schemes.

The use of link-level ARQ in the presence of an end-to-end transport protocol (e.g. a TCP-friendly protocol) has great advantages in terms of "hiding" from the end-to-end protocol errors that are caused by channel fading, ultimately improving the overall end-to-end throughput [250, 248]. However, this comes at the expense of added complexity in the control, since now we have to deal with the interactions between two "nested" control loops. If not tuned properly, link-level retransmissions can have an adverse effect on the end-to-end throughput. One promising approach for tuning the link-level ARQ protocol is to manipulate the parameters of the associated FEC. FEC reduces the mean ARQ retransmissions per unique packet, possibly increasing the effective throughput (i.e., "goodput") and decreasing the bandwidth requirement. But this comes at the expense of some overhead that could reduce the goodput, subsequently increasing the bandwidth requirement. Determining the optimal level of FEC control is one important objective of our research. The FEC code rate can be varied depending on timing requirements and channel conditions. FEC adaptiveness will be achieved by manipulating the encoder's parameters such as the puncturing rate.

A wireless access station can also support other *proxy* services, besides recovery from wireless losses. For example, an access station can hide startup delay and jitter experienced by video streams by caching video objects. The proposed infrastructure will allow us to study the trade-offs between cache requirements at proxies, network bandwidth, and quality of reception at the clients. We will investigate the effect of the length and characteristics of the control loops that get formed between the proxy and the end-systems (or other proxies). This is crucial because as the proxy is pushed further away from clients, it can serve more streams destined to more clients, but longer feedback loops and possibly higher jitter and losses on the path segments from the proxy to the clients, may render the proxy control ineffective for some (or all) clients. Finally, our proxy services will be designed and evaluated for *energy efficiency* in terms of power consumed by transmissions between the proxy and the mobile client. For example, a mobile client may access the proxy via another nearby client to conserve energy.

Efficiency also demands that sensors and network protocols preprocess responses to user queries in a representation suited to the application, to make the best use of available network bandwidth. Preprocessing at sensors or proxies will take the form of data compression, sensor fusion, multi-resolution encoding, and semantic encoding. More significantly, protocols for disseminating this information efficiently must have the capability to synchronize and merge real-time sensory data about a given object from multiple sensors on the fly and must be capable of transmitting multi-layered encoding streams to many users efficiently. This particular application can be addressed with a customized *concast* [307, 89, 88] in which merge semantics and timing semantics are specified so as to meet the targeted resolution subject to delivery and resource constraints.

Routing of Sensor Data: Streams carrying sensory data can be bandwidth-intensive and encoded for reliability/QoS or layered quality. Routing these streams poses new challenges in the design of path selection protocols. In such environment, *dispersity routing* provides an attractive technique [238, 207, 206] towards a more efficient resource allocation. For example, multiple layers of a stream are routed to the tracker using multiple paths rather than a single path. Since the packets of different layers are transmitted along multiple paths, the original stream needs to be reconstructed at the destination. On the positive side, we note that by distributing the bandwidth requirements across several paths, one can utilize the network resources more efficiently. In fact, in some scenarios, there may not be any single path that can be used to route a given stream, while there may be multiple paths that can be used to route the "fractional parts" or layers of the stream. Also, a connection established along multiple paths is more fault-tolerant to link failures. Note that a response carried by multiple streams over multiple paths can come from a single source, from a select set of replicated sources (as in a SomeCast paradigm [325]), or from different sources with different views of the tracked object (as in a *concast* paradigm [88]).

In order for dispersity routing to be effective, we need to ensure that the cost of synchronizing the different layers is kept low. This could be achieved by selecting paths with appropriate delay and jitter characteristics [237] and controlling the sending schedule of the different substreams.

To route tracking data to multiple concurrent users, multicast will be employed to realize substantial bandwidth savings over the alternative in which separate point-to-point connections are established to each user/application. Since applications may have different tracking requirements or security privileges that are not easily addressed by a single multicast session, we intend to employ new ideas from SomeCast [325] and *layered multicast* [208, 220, 301, 85, 83, 84], in which separate multicast sessions carry encoded portions of the target or track the target with increasing levels of detail. In the SomeCast scenario, a client chooses a subset of the multicast sessions to listen to and decode their data, so as to recover the original data while satisfying its own QoS requirement. In the *layered multicast* and layered coding scenario, clients with sufficient security privileges and interest may subscribe to all layers, and get a detailed tracking perspective, while others may subscribe to a base layer, in which perhaps only the approximate position of the target is revealed. We will also investigate novel ideas in dispersity routing of multicast streams, where a *logical* multicast tree is formed with multiple paths between logical nodes to achieve high availability (fault-tolerance).

C.3.4 Operating Systems Projects – Bestavros and West

Processes to be executed in the Sensorium are subject to spatial and temporal constraints. Spatial constraints arise from the fact that some processes can only execute on a small subset of processing elements that are tightly coupled with the sensory device that is closest to the domain object under consideration. Temporal constraints arise from the fact that some processes (e.g. compression or encryption) must be able to process data at a pre-specified rate and multiple data sources must be temporally consistent [242, 241]. In an environment such as the Sensorium, resource arbitration and management decisions are further complicated by the dynamic nature of the aforementioned spatio-temporal constraints due to (1) mobility (e.g. spatial constraints may change as a result of an object moving away from a camera or as a result of occlusion), (2) variable resource utilization (e.g. variable frame sizes), and (3) dynamic task priorities based on overarching mission goals (e.g. tracking a fast-moving object may be of higher priority than a stationary object).

Scheduling and Resource Management Services: Consider a single sensing device, consisting of a sensor (e.g. camera), a compute engine (e.g. PC), and a communication link with fixed upstream/downstream bandwidth (e.g. 100Mbps Ethernet). Such a device is likely to be hosting the execution of a set of processes, each of which is subject to real-time QoS constraints. Such processes may be modeled as periodic tasks with periodic resource consumption needs, whereby each process requires the exclusive use of the device for a specific length of time in each period. Obviously, it is desired that all such processes be able to multiplex their use of the device while preserving their respective QoS constraints. Traditional scheduling algorithms devised for such systems have focused on a strict "hard" deadline semantics. The classical Rate Monotonic Scheduling (RMS) algorithm of Liu and Layland[190] ensures the satisfaction of hard deadlines by requiring that for each task either the periodic resource usage be constant, or the periodic worst-case resource usage be known *a priori.*

Given such knowledge, RMS guarantees the satisfaction of all deadlines, provided that a simple schedulability condition is satisfied. Periodic Sensorium processes (e.g. tracking, compression, encryption) are examples of periodic real-time applications in which (1) tasks have highly variable utilization requirements, and (2) deadlines are firm. For such applications, RMS is too restrictive in assuming a constant (or maximal) resource requirement, and it provides a more stringent guarantee on deadlines than is necessary.

In [15, 19], we developed a Statistical Rate Monotonic Scheduling (SRMS) paradigm—a generalization of RMS that allows scheduling of periodic tasks with highly variable execution times and statistical QoS requirements.[3] The main tenet of SRMS is that the variability in task resource requirements could be smoothed through aggregation to yield guaranteed QoS. This aggregation is done over time for a given task and across multiple tasks for a given period of time. The SRMS scheduler is a simple, preemptive, fixed-priority scheduler. The SRMS job admission controller manages the QoS delivered to the various tasks through admit/reject and priority assignment decisions. In particular, it ensures the important property of task isolation, whereby tasks do not infringe on each other. To enable efficient local admission control in sensory devices embedded in the Sensorium, we intend to implement SRMS-based scheduling services.

The SRMS paradigm considers only one type of uncertainty—namely the uncertainty of the periodic resource requirements for a task. In a Sensorium setting (and in embedded systems in general), other uncertainties exists. In particular, embedded systems (by virtue of the hostile environments in which they are deployed) may be prone to frequent transient failures (e.g. due to interference). One way of mitigating this uncertainty is to enable a process that experiences a transient failure to restart. By properly *characterizing transient failures* (e.g. rates, recovery times), it is possible to translate the impact of such failures into periodic execution time variability. Such translation (or mapping) enables the SRMS paradigm to be able to manage *both* reliability and timeliness QoS constraints in an integrated fashion. The Sensorium will provide us with the necessary infrastructure that will enable an experimental evaluation of this integrated paradigm.

Our discussion so far has considered "local" admission tests and schedulability analysis for task sets submitted to a single sensory element. In the Sensorium (as well as in any distributed embedded system), it is also necessary to address "global" admission and placement issues. Specifically, in such a distributed environment, services for admitting incoming tasks and for dispatching such tasks to sensory elements. Such services must be tuned to enable a maximization of some system objective function (i.e. taking into consideration priority, QoS constraints, load balancing, system mission, etc.) For the Sensorium, we plan to apply mechanisms that use *spatially-constrained* load profiling, whereby the system may try to perform on-line capacity planning to maximize the chances that an incoming spatially-constrained task can be accepted. This work will extend our prior work on load profiling to address the integration of spatial and temporal constraints.

Operating System Instrumentation for Coordinated Resource Management: In the Sensorium, the choice of *when* and *where* to send data depends on many factors, including: 1) the quality of service (QoS) and real-time requirements associated with processing the data, 2) the resource availability at the destination host where processing is to be performed, 3) the bandwidth availabilty along the network link between the source and destination hosts, and 4) the locations of objects in the environment. This choice is made more challenging by the fact that all of these factors change dynamically over time and depend on the content captured at run-time. As a concrete example, consider some examples of how resource requirements may change at run-time for the problem of tracking objects in the Sensorium. In order to track objects correctly, the update frequency may need to increase when tracking multiple objects which are in close proximity, the required resolution of objects may change as they more closer or further away from some point of interest and the CPU processing time required to locate a target object in an image may depend on the background content. Qualitatively, this means we must vary the sampling rate and resolution of sensory data based on the state of the enviroment being sensed. In summary, system support is required to manage both communication-bound and computation-bound resources, so that data is captured, processed and transferred in keeping with its potentially dynamically-varying QoS requirements.

The approach that we propose is to build system-level components, that reside within the end-hosts and the network (i.e., the network interfaces, and the switches/routers), to manage both communication- and computation-bound resources, so that sensory data, from potentially thousands of sensors, can be managed. Highly scalable solutions to the management of such data, require operating systems to be instrumented with application-specific functions for monitoring the *actual* service provided to tasks that manage the data.

[3]SRMS relaxes the pivotal assumption of RMS—namely that the resource requirement of a periodic task is fixed. Several other relaxations of this assumption have been explored in the literature, including the incremental and design-to-time task models in [314, 100, 69, 295], the multiframe and skip-over task models in [215, 288, 183, 34], and the slack stealing and reservation models in [282, 287, 285, 286, 188, 244, 245, 294, 18].

This is similar to the notion of 'extending' the functionality of a system to meet the needs of individual applications [36], or allowing applicaton-specific management of resources via an interface that supports the construction of library-based operating systems [118].

Should the monitored service be insufficient to meet the QoS requirements to manage the data, then mechanisms and policies are needed to adapt the service offered by the system. Likewise, if application-specific monitors observe that insufficient resources are available to perform a particular operation on a data set, then mechanisms and policies must be deployed to alter the allocation of the same or other resources, to take compensatory action. For example, if there is insufficient bandwidth, to transfer sensory data from one host to another, then it may be possible to compress the data by using a CPU-bound task, thereby reducing the demand for communication-level resources at the cost of using more computation-level resources (i.e., CPU cycles).

The seeds of the framework we intend to deploy in the Sensorium are exemplified in the *Dionisys* architecture developed by Prof. West [308]. Dionisys supports applications with dynamically varying resource requirements, by providing appropriate mechanisms and policies that can be dynamically linked into kernel services. Many research groups have proposed architectures and middleware [90, 135, 303, 217, 20, 299], to provide runtime QoS guarantees. For example, the SWiFT toolkit [132] has been used to construct feedback control components in a modular fashion, to support adaptive resource management [133]. Rather than focusing on independent resource managers (as is done in SWiFT), Dionisys focuses on mechanisms to support *coordinated* service (and, hence, resource) management. Specifically, explicit coordination between pairs of service managers is achieved by having the behavior of one service manager affected by events from another service manager. By implementing and embedding Dionisys in the Sensorium, we would be able to investigate the issues and tradeoffs in the performance of different adaptation configurations (i.e., different combinations of feedback control loops), involving the coordination of multiple service managers.

C.4 Symbiotic Projects Enabled Through the Infrastructure

In this section, we discuss a number of collaborative projects that demonstrate the symbiotic relationship between the research agendas of various subsets of the PIs.

C.4.1 Motion Mining – Betke, Kollios, and Sclaroff

The Sensorium provides the infrastructure for collecting large datasets of human motion. As these datasets grow, there will be an opportunity to analyze this massive data archive to gain new insights that can be used to improve our understanding and models of human motion. Insights gained through *motion mining* could lead to improved methods for computer-assisted physical rehabilitation, occupational safety, and ergonomics, as well as improved methods for sports training, medicine, and diagnosis. Furthermore, motion mining could lead to improved computer vision and pattern recognition algorithms that are specially tuned to basic patterns or clusters found in human motion databases.

Many current methods require searching of descriptive text fields that are entered by hand. Given the spatio-temporal nature of human motion, temporal alignment of the text annotations by a human can be quite laborious and error prone. In addition, text annotations can severely limit what kinds of motion patterns can be found/retrieved in the collected data; this is because such indexing is limited to only those annotations entered. Patterns not annotated or unnoticed are not searchable. To achieve a motion mining system, what is needed are methods to directly index, retrieve, and find patterns in the time series and trajectory stored in databases of human motion.

This motion mining effort requires tight collaboration between researchers in two disciplines: Stan Sclaroff and Margrit Betke in computer vision and George Kollios in data mining. The proposed effort will focus on:

1. Labeling time-varying objects and estimating blob motion trajectories in image sequences. This includes building statistical models of the background and detection/segmentation of time-evolving image blobs.
2. Encoding time-varying objects and their motion at multiple levels of detail. This includes developing methods for automated extraction of visual and motion features, and recursive estimation of each blob's motion trajectory.
3. Classifying moving or time-varying image regions into semantic classes of interest. This involves developing classifiers that can map a blob's visual features and/or motion patterns into relevant class labels.
4. Indexing the motion of a large set of moving objects and efficiently answering similarity queries. For example given the trajectory of an object we would like to find other objects that moved in a similar way. These methods will help to perform on-line exploratory data analysis on large spatiotemporal datasets.

5. Efficient clustering and outlier detection to be used for classification, prediction and data modeling of the moving objects according to their motion characteristics.

Spatio-temporal indexing relies on robust computer vision algorithms like those described in Section C.3.1: (1) can detect, estimate, and encode relevant information about human motion in image sequences, and (2) determine correspondence of the same person seen in different camera views and at different times, based on the color distribution of clothing, body motion and pose, and facial appearance. Conversely, computer vision algorithm accuracy could be improved by exploiting patterns identified in mining of motion databases. In motion mining, the problem is how to efficiently index and retrieve the motion data from the database, as well as how to analyze and exploit such data; solutions to these problems will be developed through the database research outlined in Section C.3.2.

C.4.2 Location Management for Mobile Objects – Byers, Kollios, and Matta

The SENSORIUMwill provide synergy between PIs working in databases and networks in the area of mobility management. The main challenge for this effort is *"how can we build a highly available distributed architecture that manages the location of objects as they continuously move around the* SENSORIUM*, so as to reliably and efficiently answer queries, including advance future queries?"*

In the SENSORIUM, the sensor with the best view of a moving object (e.g., person) may change frequently. Therefore, the responsibility of tracking the object will frequently have to be *handed off* from one sensor to another, or from one sensor domain to another. Consequently, the location of the object has to be continually maintained in order to answer queries about that object. If we can accurately model and predict the movement of objects, we can enhance the operation and performance of the system in at least three respects: (1) sensor handoff can be anticipated and tracking resources can be allocated in advance to ensure smooth operation; (2) advance queries about the future (e.g. objects that will be in the range of a sensor in the next 10 minutes) can be answered; and (3) location information can be updated less frequently as long as the exact (current) location of objects can be accurately predicted.

The SENSORIUM offers unique opportunities for building and exploiting mobility models. Unlike traditional systems which assume GPS-enabled objects, sensors themselves will collaborate to recognize objects and build their mobility (trajectory) models. Furthermore, instead of examining the individual trajectories of mobile objects [124], we will exploit database techniques (such as those discussed in Section C.3.2) in answering future range queries to speed up operations such as predictive sensor handoff.

Mobile network location services have been proposed only for the case in which a centralized database system is used to store the location of all moving nodes. However there is not much work on providing advance query capabilities. In our project we plan to consider exactly this problem. Our idea is to selectively cache or replicate the location of mobile objects using a detailed cost-based model that integrates database and communication costs. In our previous work [252], object locations are locally cached based on the query-to-mobility ratio, and subject to constraints such as cache size and requirements on network resources for keeping caches up-to-date. The query-to-mobility ratio (QMR) estimates for a tracked object the relative frequency of local queries to movement across sensors or sensor domains. Intuitively, we would like to cache the location of objects which (1) are tracked most frequently so the cache entry can serve as many queries as possible, and (2) move least frequently so the cache entry is more likely to be valid (i.e. points to the correct location). Ideally, to minimize cache misses and invalid cache entries, the system should have infinite caches and instantaneously know *every* movement a mobile object makes; clearly an impossible and prohibitively expensive proposition. In this project, we plan to reduce caching errors at lower update cost by augmenting our QMR estimation with tracking and mobility prediction models.

A cache miss can be overcome by accessing a *local* location database replica, rather than a remote home database. Replicated databases increase the availability and responsiveness of the system. In our previous work [253], replicas were hierarchically organized and location of objects were selectively replicated so as to reduce the search time and localize the dessimination of location updates. We plan on extending the cost-based model of distributed caching and replication to consider the access time needed for the database to answer a query using the index. For example, if we use an R-tree access method or one of its variants, we could use existing work on modeling R-tree performance under different query workloads [290].

Finally, each replica can be responsible for a service area covering a number of sensor domains. Once the service area where the tracked object resides is located, the different sensor domains in the area can be paged to locate the exact location of the object. The trade-off between the cost of updating replicas and the cost of paging will be investigated.

C.4.3 Real-Time Multiple Object Detection & Tracking – Bestavros, Betke, Sclaroff, and West

An important capability of the Sensorium is the ability of an individual camera to track multiple moving objects robustly and in real-time. The problem of tracking objects in real-time is difficult even when performed in well-structured environments. Prior work by the PIs on tracking multiple vehicles from a car moving along a highway [64] demonstrated that difficulty due to such issues as traffic volume, driver behavior, lighting, and occlusion. In a Sensorium environment, similar difficulties are complemented with more complex QoS requirements. Specifically, tracking processes for different objects in the Sensorium may possess QoS constraints that vary significantly. While it may be adequate to support a "soft" deadline semantics for some processes, it may be necessary to support much "harder" deadlines for others.

In prior computer vision research work of ours [64], deadlines were guaranteed by using the Maruti real-time operating system [266], which allows for off-line schedulability analysis using worst-case execution times of periodic task sets—thus *a priori* ensuring satisfaction of hard real-time constraints. Three aspects of this approach to satisfying real-time constraints are not acceptable for Sensorium-like environments: (1) assuming that all deadlines are hard is not warranted, (2) assuming that worst-case periodic resource utilization is representative of actual loads is not warranted given the extremely high variability of such tasks, and (3) the off-line rather than on-line nature of the admission control process. To accomodate these requirements, it is far better for tracking processes to utilize scheduling services that allow the support of a continuum of deadline semantics (as opposed to the traditional soft/hard semantics).

In prior operating system and real-time scheduling research of ours, novel techniques, architectures, and services for supporting a smoother specification of real-time QoS constraints were developed. These techniques enable the specification of real-time constraints in a combinatorial fashion or in a statistical fashion. Combinatorically, the DWCS schedulers [309] allow QoS constraints to be spelled out as x out of every consecutive y deadlines must be met, whereas statistically, the SRMS schedulers [15] allow QoS constraints to be spelled out as $p\%$ of all deadlines must be met over a long enough periodic execution.

To facilitate the development of real-time visual tracking (and similar applications) in Sensorium-like environments, an API is clearly needed to enable a seemless implementation of such functionalities. A natural framework on top of which this API could be built our Linux-based Dionysis architecture [310] and SRMS API [19]. The Dionysis architecture allows monitoring and control of system resources to be accomplished in an adaptive fashion and is particularly suited for the dynamic nature of Sensorium-like environments.

The specific research projects to be undertaken as part of this collaborative effort between the Vision and Operating Systems groups will aim to: (1) develop an appropriate set of parametrizable "QoS templates" that can be used as off-the-shelf components in the development of real-time applications involving visual tracking; (2) provide appropriate utilities that enable the profiling/characterization of the periodic resource consumption of tracking processes. Such profiling is necessary for resource reservation (or schedulability analysis); (3) develop new visual tracking algorithms that are tolerant to the "predictable" uncertainties that result from using schedulers that guarantee real-time constraints in a combinatorial (DWCS) or statistical (SRMS) manner; (4) extend the SRMS/DWCS scheduling services to enable the coordination of multiple resources (e.g., camera, CPU, network) and the accommodation of failure semantics of embedded devices.

C.4.4 Programming with Flows – Bestavros, Byers, Kfoury, Matta, and West

Programming new services for Sensorium-like environments suffers from the same lack of organizing principles as did programming of stand-alone computers some thirty years ago. Programming language technology improved through better understanding of useful abstraction mechanisms for controlling computational processes. Finding analogous abstraction mechanisms for sensor network environments is complicated by its *open* nature, versus the simpler, *closed* nature of stand-alone systems. In a closed system, software is written in high-level languages having well-developed formal calculi for analyzing semantics, logics for validating program properties, type disciplines for statically enforcing desirable dynamic properties, with complementary compiler implementation technology. While successful for closed systems at their functional core, the collective treatment of input/output, sequentiality, state, order, concurrency and continuations is harder for open systems, where interaction with an external environment is crucial.

Essential to every programming language, in a closed or open world, are the identification of primitive expressions, the means of building complex processes, and the abstraction of those processes via naming. Understanding how to program a Sensorium will depend on crafting the right such features. We believe, based on our experience in network service design as well as programming language design, that a critical component for advancing this agenda is the treatment of network flows as first-class values—just as procedures, continuations, and environments have been implemented as first-class values in general-purpose programming languages. Specifically, we believe that in order to rapidly experiment with and deploy a wide range of new services in a Sensorium-like setting, it is necessary to adopt a more powerful model for the naming, creation, sharing, and processing of network flows. In effect, we need to adopt new programming paradigms especially adapted to such an infrastructure, in which network flows are first-class values. First-class objects are ones that can be explicitly *named* and used as parameters of (and returned as the values of) procedures.

Informally, a flow is an abstraction for the communications that must take place in the network to effect data transfers or control signals between two entities. What constitutes a "flow" depends on the layer at which such transfer (or signaling) occurs. For example, at the transport layer, a flow is the sequence of packets from the same source(s) to the same destination(s). This is commonly captured by the abstraction of a "socket", which provides an appropriate encapsulation of flow features for network programming. We are working on the development of new support for network flows in two inter-related ways:

1. *Extending the notion of flow to include various new parameters.* Flow parametrization enables processing and differentiation of flows based on safety constraints that govern proper use.
2. *Elevating the notion of flow to a first-class datatype.* This presumes the design of appropriate programming languages and/or adaptation of existing ones to manipulate, combine and modify flows.

For conventional programming languages types and their generalizations have been found to offer many benefits. Strong static types have been used with increasing success in enforcing safety of programs, e.g., in Java, Haskell and ML. Since the late 1980's, types have been used in organizing compilation, code optimization and generation, and execution. Also, types have been used to analyze and enforce security features of programs. If the evolution of conventional programming languages is any indication, from the moment we deal with flows as first-class values we expect that types will play the same role in enforcing safety/reliability/security of programs manipulating network flows.

The collaborative effort described above will focus on the following specific areas: (1) Transport Services to support various degrees of reliability, congestion management, and timeliness, (2) Routing Services to support various degrees of mobility and multicasting, (3) End-to-End Services to support various degrees of QoS and security, (4) Naming mechanisms to support flexible composition of network services, and (5) Type inference to support efficient and safe flow-oriented programming. Towards the goals outlined above, the proposed research will be a collaborative effort carried primarily by members of established research groups in Networking Systems and in Programming Languages.

A key component of this collaborative effort is prototyping. To that end, the utility of a paradigm in which network flows are first-class values will be demonstrated by implementing *NetBench*—a programming environment in which a core set of network flow types and operations will be supported, along with a type-checking and type-inference system to handle types of network flows. Our ability to develop *NetBench* and demonstrate its utility hinges on the availability of a programmable networking infrastructure supporting applications with significant needs for operations on flows (e.g., merger, handoff, multicast, compression, synchronization). The Sensorium infrastructure (and associated applications) provide us with such an opportunity.

C.4.5 Security and Privacy – Itkis and Reyzin

As underscored in a support letter from members of the Privacy Foundation, systems that observe and track humans raise legitimate privacy concerns. While a precise privacy policy is a necessary first step, it will not, by itself, protect subjects from misuse of data, as it addresses only *authorized* use of the system. Thus, it is imperative to study how to protect data collected by Sensorium from *unauthorized* use. Such unauthorized use can come both from inside the system (perpetrated by those who have some legitimate access to the data) and from outside the system (perpetrated by those who break in or eavesdrop to gain access to the data).

To prevent unauthorized access from within the system and ensure that each individual has access only to the appropriate data, a privilege management mechanism is required. Such mechanisms range from simple Access Control Lists (ACLs), to more powerful tools such as PolicyMaker [72] and KeyNote [71]. It is necessary to investigate, however, whether KeyNote or similar tools are appropriate for the Sensorium. Because of the vast amounts of various kinds spatio-temporal data about each subject, it will be necessary to develop security policies that are both flexible and simple to specify, so as not to impair the system's usability.

Regardless of the granularity of the access policy, however, some entities (for example, the system administrator) will necessarily have vast access privileges. In order to prevent potential misuse of such privileges by any single person, a security policy could require that particularly sensitive operations be authorized and controlled by multiple persons. Such authentication techniques as Accountable Subgroup Multi-signatures ([213]) may be of value.

For efficiency reasons, it would be desirable to group users with similar privileges—for example, all users allowed to access data about a certain subject. However, such privileges will often need to be revoked on an individual basis, particularly when the users are accessing the system via mobile devices, which are quite likely to get lost or stolen. Group key management techniques, such as those in [304, 91, 157], will be quite useful in the Sensorium.

Authentication mechanisms will also be necessary to prevent unauthorized access from outside the system. Because of the limited power of mobile devices used to access the system, the traditional authentication methods based on public key cryptography, certificates and public key infrastructures may be too slow and unwieldy for the task. The system would benefit from the use of gradual authentication mechanisms, such as Asymmetric MACs [156, 91]. Moreover, particular devices may not always be rigidly associated with the users of Sensorium, which introduces additional key management challenges. New methods of managing keys of mobile users are are currently investigated in other projects [158, 159].

Finally, it will be necessary to prevent information from being stolen while in transit, especially because the Sensorium network contains mobile nodes and wireless links. As a starting point for addressing issues of Sensorium network security, we intend to utilize protocols based on existing standards [116, 123, 151]. In particular, the Wireless Application Protocol (WAP) forum has proposed the Wireless Transport Layered Security (WTLS) protocol [306] that is intended for the wireless setting. However, the specifics of the Sensorium network will create new requirements, not addressed in the current standards, which we plan to investigate. These requirements include, in particular, the need to securely guarantee certain network resources to particular tasks, preventing denial of service attacks.

C.4.6 Measurement and Characterization – Bestavros and Crovella

Our department was home to the first study that characterized Web client workloads and established many important properties of Web traffic (e.g. self similarity, heavy-tailed sizes of transfers, locality of reference, the Zipf profile of object popularity). Synthetic workload generation tools built in our research labs (which leveraged on the findings of these studies) are used by *hundreds* of research labs worldwide. More importantly, observations and invariants gleaned from our characterization efforts have led to the development of significantly more effective protocols (e.g. caching algorithms, traffic pacing algorithms, routing protocols, Web request scheduling protocols, etc.) The catalyst for this success and impact was the ability of our Web and InterNetworking Group (WING) to collect *representative* Web client traces and HTTP server logs [107].

The availability of traces from applications developed for the Sensorium will provide us (and the measurement and modeling community in general) with invaluable datasets, which we intend to use for purposes of modeling and characterization of realistic workloads in distributed sensory networks and mobile ad-hoc networks. Examples of important characterization and modeling questions include: (1) correlation properties of traffic from spatially co-located (video) sensors, (2) characteristics of aggregated sensory traffic; (3) spatio-temporal distribution of mobile cells in an ad-hoc networking environment, and (4) periodic computation and communication footprints of sensory tasks (e.g. tracking). Understanding these (as well as other) characteristics of sensory and ad-hoc networks is crucial not only for the evaluation of proposed protocols and standards, but also for the development of effective solutions that capitalize on any properties or invariants that may be gleaned through these characterization efforts.

The availability of the Sensorium will provide the opportunity to pursue the aforementioned measurement and characterization efforts. More importantly, it is likely to enable us to effect the solutions advanced by other groups working on Sensorium projects (e.g. vision, databases, security, resource management, etc.)

C.5 Conclusion

The proliferation of networked, embedded and mobile digital video sensors in our society is likely to result in a paradigm shift in many areas of basic computer science research to address (1) the unique spatio-temporal aspects of sensory (visual) data acquisition, processing, representation, communication, storage, real-time indexing and retrieval, and security and privacy management, and (2) the challenges of Quality-of-Service management and coordinated resource arbitration of sensory networks, which are both embedded and mobile. This proposal requests funds to acquire a Sensorium—a research infrastructure that enables the pursuit of a number of related projects addressing these challenges within the confines of two important vertical applications. The availability of the Sensorium will not only foster collaboration and promote technology transfer, but also it will provide a fertile ground for a truly intradisciplinary training of undergraduate and graduate students.