Large-scale Situational Awareness with Camera Networks and Multimodal Sensing

Vision: Sensors of various modalities and capabilities, especially cameras, have become ubiquitous in our environment. Their intended use is wide ranging and encompasses surveillance, transportation, entertainment, education, healthcare, emergency response, disaster recovery, and the like. Technological advances and the low cost of such sensors enable deployment of large-scale camera networks in large metropolis such as London and New York. Multimedia algorithms for analyzing and drawing inferences from video and audio have also matured tremendously in recent times. Despite all these advances, large-scale reliable systems for media-rich sensor-based applications such as surveillance are yet to become commonplace. Why is that? There are several forces at work here. First of all, the system abstractions are just not at the right level for quickly prototyping such applications in the large. Second, while Moore's law has held true for predicting the growth of processing power, the volume of data that applications are called upon to handle is growing similarly, if not faster. Enormous amount of sensing data is continually generated for real-time analysis in such applications. Further, due to the very nature of the application domain, there are dynamic and demanding resource requirements for such analyses. The data intensive nature coupled with the lack of right set of abstractions for programming such applications have hitherto made realizing reliable large-scale surveillance systems difficult. The fundamental challenges include dealing with heterogeneity, scalability, virtualization, and mobility. This white paper raises the challenges and solution approaches for dealing with large-scale media-rich infrastructures for addressing the needs of large-scale sensor-based applications, often classified as *situation awareness* applications, using *smart surveillance* as a canonical example.

Evidence: We expect the proposed vision will lead to the following major advances: (a) developing the right set of programming abstractions and execution models for large-scale situation awareness applications; (b) addressing scalability both in terms of the physical infrastructure and the data deluge caused by modern sensors; (c) large-scale context-aware computing to prioritize the computation needs of the applications; and (d) flexible resource management and placement of computations to meet quality of service guarantees. We propose a programming model and an execution framework that virtualizes the sensor sources allowing the domain expert to concentrate on the analytics, while the system takes care of managing the physical sensor streams, networking, and computational resources commensurate with the application dynamics and resource availability. Such a separation of concerns will allow the rapid development and deployment of large-scale situation awareness applications.

Credentials: **Kishore Ramachandran** is the Director of Samsung Tech Advanced Research (STAR) center and Professor in the College of Computing at the Georgia Institute of Technology. He is the recipient of an NSF Presidential Young Investigator Award and has been PI on several NSF grants. **Liviu Iftode** is a Professor of Computer Science at Rutgers University. He has been the PI of several NSF grants including a CAREER, a medium ITR on cooperative computing for distributed embedded systems and a NeTS grant on vehicular computing systems. **Dr. Rajnish Kumar** is a research scientist at College of Computing at Georgia Tech. His research interests include camera sensor networks and distributed systems. **Santosh Pande** is an Associate Professor in the College of Computing and has been the PI of several NSF grants.

Kurt Rothermel is a Professor for Computer Science at University of Stuttgart. He is Director of Institute of Parallel and Distributed Systems and head of Collaborative Research Center Nexus, which is funded by German Science Foundation (DFG) and is conducting research in the area of mobile context-aware systems. **Dr. Boris Koldehofe** is a research scientist in the Institute of Parallel and Distributed Systems at University of Stuttgart.

Details: How can a vision expert write an application for video-based surveillance that spans 1000s of cameras and other sensors? How can we design a scalable infrastructure that spans a huge geographical area such as an airport or a city to support such applications? Which operator models are needed to support the efficient adaptation for the placement of computations and corresponding data streams? How can we transparently migrate computations between the edges of the network (i.e., at or close to the sensors) and the computational workhorses (e.g., Cloud)? How do we adaptively adjust the fidelity of the computation commensurate with the application dynamics (e.g., increased number of targets to be observed than can be sustained by the infrastructure)? These are some of the questions that our vision for large-scale situation awareness raises. Our current research on programming abstractions and execution frameworks is addressing many of these questions in order to provide the building blocks for accomplishing this vision.

Our research on the *target container* programming model provides an abstraction of a “target” to the application. Depending on the application a “target” may be a person, moving vehicle, or traffic congestion. The programming model expects the application programmer to provide “handlers” (e.g., detector and tracker in computer vision parlance) for the analytics on the streams to track a target. These handlers are invoked by the runtime as new targets appear in the environment. Transparent to the application, the execution engine delivers the streams needed for the tracking (i.e., the sensor sources are virtualized) as the target moves in the environment. The execution engine also manages all the plumbing and distribution of data and computation to ease the programming burden, providing a scalable per-target resource allocation according to target priority.

Our research on computational operator placement allows applications to define situations by providing a specification of relevant targets, computational operators and their relations. Furthermore, the application programmer can specify performance constraints as well as optimization goals that need to be met at runtime. The computational operator placement will determine autonomously the assignment of operators to computational resources by accounting for the characteristics of the underlying network as well as considering the current locations of targets and consumers. The placement has to adapt to several sources of dynamism including the mobility of targets, the number of available resources, the load of event streams, and the number of consumers. In order to support fast and scalable adaptation, operator placement must be based on independent and local decisions without complete knowledge of the operator network. Since every operator is a possible source of failure and operators are associated with state information, providing reliability at a reasonable cost is another critical research question towards accomplishing the vision.