Vehicle Tracking and Autonomous Interception

Neeraj Kaushik
Department of Electronics and Communication Engineering
Faculty of Engineering, Teerthanker Mahaveer University, Moradabad, Uttar Pradesh, India

ABSTRACT: This paper define the architecture and implementation of PEG, a networked system of distributed sensor nodes that detects an uncooperative agent called the evader and helps apprehend the evader by an autonomous robot. PEG requires services such as election of leaders, routing, and aggregation of networks and control of the closed loops. Instead of using distributed network approaches of general intent for these functions, and use whole-system modeling and focus on spatial and physical properties to establish simple and efficient frameworks. This approach promotes the design of sensor networks and provides pragmatic solutions which leverage physical properties to simplify the design of embedded distributed systems. This paper used 110 sensor nodes to install PEG on an area of 500 square meters, and successfully intercepted the evader in both runs and faced functional problems such as node breakage, packing choices, in situ testing, network reprogramming, and device reconfiguration when implementing PEG. This paper address the approaches that have taken to tackle these problems and share our experience in deploying a large sensor network system.

KEYWORDS: Linearity feature, occlusions, shadow elimination, traffic surveillance, vehicle classification.

INTRODUCTION

In recent years, road congestion has been a major problem. Earlier approaches sought to install additional roads to reduce congestion but it is getting less and less possible to add additional lanes. Contemporary approaches emphasize greater knowledge and monitoring to allow more effective use of current resources.

The need for better traffic intelligence, and therefore growing reliance on traffic monitoring, has led to a need for improved vehicle detection such as wide-area detectors; whilst the high expense and safety threats associated with lane closures have driven the search for non-invasive detectors installed beyond the edge of the pavement. One innovative solution is vehicle monitoring via visual image analysis, which can provide conventional traffic parameters such as flow and distance, as well as novel parameters such as lane shifts and trajectories of vehicles. If the traffic paths, or trajectories, are plotted over a highway length rather than at a particular location, actual density may be calculated instead of merely measuring the occupancy of the detector. In addition, the conventional traffic parameters are more robust by averaging trajectories over space and time than equivalent measurements from point detectors, which can only average over time[1].

Additional vehicle trajectory information may lead to better accident identification, both by detecting stopped vehicles within the field of view of the camera and by finding lane shift maneuvers or acceleration deceleration patterns which are suggestive of accidents outside the field of view of the camera. Trajectory data may also be used to simplify labor intensive traffic experiments, such as analyzing vehicle movements in weaving areas or bottlenecks, historically used. The vehicle monitoring device will produce individual vehicle data (e.g. distance, headway, velocity, acceleration) that may lead to improved analysis of traffic patterns and a deeper understanding of driver behavior[2].

Furthermore, our group has demonstrated that the device can collect vehicle signatures and match same vehicle observations at different detector stations. Such matching of the signature can be used to calculate true link travel time and thereby estimate conditions between widely spaced detectors, rather than assuming local conditions are reflective of the entire network.

This paper outlines PEG design, implementation, and experience. Section II deals with our general design approach in relation to the related function. Section III defines the program, its hardware base, and the general architecture of the operating framework. Section IV identifies the elements for the in-network analysis of local identification and aggregation into a location calculation. Section V focuses on effective mobile to mobile routing, including efficient tree creation and routing of landmarks. Section VI explains navigation configuration and pursuer control. Section VII evaluates our system design, and Section VIII outlines several of the experiences that affect the overall design and implementation of the systems. Our main contributions are to:

• This paper explain the design and implementation of PEG, a networked system of distributed sensor nodes which detects an evader and helps a pursuer catch the evader.

• This use whole-system modeling and use spatial and physical properties to design distributed algorithms that are powerful and simple, assuming this solution to a variety of applications is applicable.

• They show one of the first working large-scale monitoring and pursing systems that uses small sensor nodes in computational and bandwidth.

• This paper provide realistic guidance on the implementation of large sensor network applications like kit configuration, testing strategies and network management tools at the highest level.

SYSTEM ARCHITECTURE

This paper developed tracking, routing, data collection, and chase tools to provide pursuers with reliable detection events rapidly and regularly and provide a sense of the overall in-formation flow and identify the resources in the constituent structure. Additional power management issues as well as authentication and encryption are beyond this paper's scope. The sensor network senses the evader and routes this information to the pursuer to intercept the evader, and the pursuer acts on this data. Essentially, the sensor layer provides two high-level services: self-location, and vehicle detection. The first core feature, position, is used to create a network-wide coordinate system from which the pursuer can map the detection events obtained to relevant physical locations. Using time-of - flight ultrasonic ranging systems with anchor-based localisation algorithms, ad-hoc self-location is accomplished.

The system architecture allows self-localization, but due to intermittent errors it was not used in live demonstration; this is rectified in later research.

When a vehicle is present, the nearby node sensing and detection portion can cause detection events, and activate the data aggregation leader election algorithm. The leader election process is carried out over a neighborhood service in tuple-space. The elected official would use landmark routing to spread the aggregated data to the pursuers, running over a basic tree building process.

Once sensor readings hit the pursuer, an object disambiguation device is used by the pursuer to assess the origin of the event: the evader, the pursuer or the disturbance. Sensor readings, which are calculated to correspond to the evader, are sent to the estimate provider for the evader location. The Pursuer location estimation software uses GPS[3] unit data to assess a pursuer location prediction. Estimates of the pursuer's and evader's position are sent to the interception service, which creates a destination for the pursuer to intercept. The path planning service manages this destination to create a viable pathway. Finally, the road is sent to the service path during which the pursuer is closely regulated along this road. Section VI further explores those processes.

In addition to the core functionally required for PEG, new system services are also implemented to ease the challenge of man- aging and network configuration at deployment time. The Configuration component allows system parameters which are useful for system tuning to be configured to run-time. The component Node Management is used for node identification, debugging, and power cycle management across the network.

SENSOR PLATFORM

Our system's sensor tier consists of Berkeley Mica2Dot motes, a quarter-sized processor with an 8-bit 4 MHz Atmel ATMEGA128L CPU [4]with an instruction buffer of 128 bytes, and 4 kB RAM. Its radio is a low-power Chipcon CC1000[5] radio that delivers around 2 kB / s device bandwidth for our specific antenna and environment, with a maximum contact range of about thirty meters. Each node uses a magnetometer to detect changes in a magnetic field likely caused by a moving vehicle nearby. An ultrasound transceiver at 25 kHz is used for time-of-flight ranging. A reflector cone is located above the transceiver to disperse the Omni-directional ultrasonic waves that greatly decreases the frequency radius to around 2 meters. There is a base at the bottom of the node, which secures the node to the ground and extends it a few inches above the ground. The plastic enclosure is all protected by the battery, voltage conversion board, magnetic sensor and the Mica2Dot. The side of the enclosure has a hole that lets the Mica2Dot attach to a quarter-wavelength piano wire antenna. The only sensor that is exposed is the ultrasonic transceiver at the top, the cone being securely mounted above it. The whole package is durable to vehicle damage, and the mechanism at the base holds the node upright even after collisions to lift the node a few inches off the ground plane for efficient radio contact. TinyOS, an event-driven operating system for networked applications in wire-less embedded systems, runs all nodes at the sensor tier. Implementing all of the key resources uses around 60 kB of system memory and about 3 kB of RAM[6].

HIGHER TIER PLATFORM

Our land robots are effectively GPS-equipped[3], handheld off-road laptops. Every robot runs Linux on a 266 MHz Pentium2 CPU with 128 MB of RAM, 802.11 wireless router, 20 GB of hard disk, off-road all-terrain boots, a motor controller module and high-precision GPS differentials. This architecture is enough to execute the basic, higher-tier services. The GPS usually offers forecasts of around every 0.1 seconds 0.02 mts. The robot's maximum speed is about 0.5 m/s, with autonomous speed modulation for each wheel. One pursuer and one evader in our launch, each of them the same type robot.

VEHICLE DETECTION

Detection of a vehicle in the network starts with collecting and analysis of data by a node leading to the creation of a location estimation report. In this segment, we'll see how an overall device bandwidth review guides the design of our vehicle detection phases.

Bandwidth-Driven Design

Sensor network is built for complete, reliable sensor coverage-a car excites at least four and up to nine sensors for sensors placed in a grid. Using this coverage criterion and develop the rest of the detection system with an appreciation of the effect on national bandwidth limitations of low-level decisions.

Assuming a local aggregate bandwidth of 39 packets per second, a single node may produce up to four reports a second until the communication channel is saturated by a area of nine nodes If each node sends these detection events, it will saturate the local channel leaving no bandwidth for other communications, such as routing these readings to the pursuer. Moreover, as more vehicles are added to the system, routing the data will increasingly tax the system's bandwidth.

This paper use local consolidation to simplify multiple detection events into one location log, and devote half of the overall bandwidth to swap local detection data, and the remaining bandwidth to pursuers for system-wide activities such as estimates of routing location. While a large portion of the network bandwidth is used for exchanging network detections, the pursuer nevertheless receives regular alerts of location. This decompose this entire cycle into three distinct phases: calibration and sensing, accounts of local identification, and estimate of the leader's election and position.

Calibration and Sensing

The magnetometer measures all the magnetic surrounding. This includes static structures such as the Earth's magnetic field (these magnetometers are often used for digital compassing) as well as underground metal pipes, metal in a desk chair, or rebar in a parking garage structure concrete, for example. To detect changes in the magnetic field induced by a traveling object, account must be taken of this static state in the measurements of each node. Every node subtracts every reading from the output of a moving average. It imposes a minimum observable speed on a vehicle because it would be indistinguishable from the static atmosphere from a relatively slowly moving body.

One contact didn't foresee is the relationship between the magnetometer and the radio communication. Radio transmissions excite significant readings from the magnetometer due to the proximity of the radio chip and the magnetometer chip, which is partly due to the small package design but also exacerbated by our hardware design. For a solution, when a radio packet is being transmitted or received at the source, it invalidate magnetometer readings for a limited period of time.

Routing

Through PEG, the primary routing prerequisite is to transmit the evader identification events to mobile pursuers, as sensed by the network. That is to be route packets to a couple of mobile destinations from several outlets. It varies from the standard multiple-to-one data processing traffic layout seen in many implementations of the sensor network. Nonetheless, it resembles some of the research contained in the literature on mobile computing and offers different approaches to support this mobile routing service. In this section, we first analyze these solutions and then address a easy and effective landmark routing method for arriving with a solution that could be applicable to structures other than PEG.

Design Approach

One architecture strategy is to view the entire network and mobile pursuers as one ad hoc mobile device, and to implement well-known mobile routing algorithms such as DSR[7], AODV[8] and TORA[9] to provide any-to-any routing operation. These protocols are designed to support any pair of separate traffic flows when PEG traffic is clustered and guided only to a few (pursuers) moving endpoints. Nonetheless, with this approach the resulting routing paths would be effective as these algorithms optimize routes based on the shortest path metric. Another solution is to decouple the network from the mobile pursuers and leverage the topology of the static network to popular the difficulty of routing communications. This is analogous to the home-agent function used in mobile computing, where a home-agent is assigned to any pursuer for data forwarding purposes. For example, recent research has been proposed to help group communication between a series of moving agents in a bounding box over a sensor network. It believes that any-to-any routing comes free using regional routing and retains a horizontal backbone around the boundary shell.

Communication between the moving agents and the network is achieved through those home agents on the backbone. Mobile agents need to register with the backbone to discover new home agents as they move; these migratory home agents help to create more effective routing routes. The level of connectivity thus relies on the overhead of the pace of backbone repair and home agent migration.

The approach also exploits the static network topology for efficiency and simplicity, but it do not assume any support for geographical routing. In addition, this reduce overhead protocol exchange by spreading soft state across the network and losing routing capacity marginally. To split the many-to-few routing problem into two subproblems, this paper use landmark routing: many-to-landmark and landmark-to-few. Landmark routing is a simple mechanism that makes use of a known rendezvous point to route packets to a few destinations from many sources. For a node in the spanning tree to route a detection event to a pursuer, it sends a message to the root node, the landmark, first up a spanning tree. The symbol then forward the message to the pursuer. The original landmark paper uses a hierarchy of landmarks to discuss the scalability of that approach. This strategy leads to longer routing paths because traffic needs to be routed via the symbol, which affects latency but requires less control bandwidth to maintain routes to the moving target than other protocols such as AODV.

Efficient Landmark Routing

Some nodes in the network will send messages to the landmark using the spanning tree built using the previous method, which must then be able to forward the messages to the running pursuers. In order to accomplish this, the pursuer informs the network periodically by picking a node in its vicinity to route a special message to the landmark, thereby laying a "crumb trail." Instead of keeping all the routing states at the landmark, this message deposits a "crumb" with each intermediate router on the spanning tree so that messages intended for the pursuers at the landmark can reverse the path. Because each node in the crumb trail knows the next hop, the overhead connectivity is smaller because there is no need to carry the whole reverse path in each packet.

This support multiple simultaneous crumb paths, which enable several mobile destinations. When depositing the crumb growing crumb trail is marked by the pursuer's ID.

The pursuer increases a series to assign these crumb trails a time dimension so these paths can follow the pursuer's location dynamically. All these routing states are poor in that they are stagnant over time, and thus rusty crumb trails naturally prune themselves.

CONCLUSION

The design and implementation of PEG enables us to establish relevant principles of system design which are useful to other networking systems. Our analysis of the whole system design provides a clean decomposition process for problems. To gain total consistency in program execution, it requires complexity to be put at the correct levels of the sys- tem. Simplicity is further achieved by taking advantage of the application's environmental and physical characteristics at deployment time. Protocols should take advantage of soft state, loose consistency, and quick failure to cope with the loss of wireless channel and somewhat unreliable hardware sensor network platform when appropriate.

The infrastructure for system management and debugging should be well designed to anticipate the need for system reconfigurations during deployment time. Our device decomposition allows each subsystem to be reusable across a wide range of applications for the sensor network. Abstraction of the community and processes for representative elections refer to any tracking program that involves collection of local data. For other data dissemination protocols, the density adaptive flooding mechanism prevents the diffusion storm problem. The landmark routing subsystem is useful for any moving-entity use. The network monitoring and testing tools are helpful as other sensor networks are deployed.

This paper demonstrate a working system that not only monitors sensory data, but also tracks and controls a higher-level system to real-time perform a cooperative task. On the sensor tier the system assumes very little requirements for processing and communication. In addition, the physical properties of PEG in our architecture to create a practical, simple design that is resistant to failures. We conclude the same principle of architecture will be practiced in designing potential sensing and actuating devices. We'll be deploying a larger order magnitude network in the near future to achieve many of the same goals as this work and would exploit the lessons from this work to build a model that is well designed for remote reprogramming, parameterization, and device management of long life and large scale. The main aim of this redeployment is to focus on data collection and the study of methodical processes. This new development user will be able to implement and quantify greater variation: robot size, node spacing, topology of the node, GPS resolution, fidelity sensing, sensing time, etc. This initial initiative outlined in this work has been beneficial to the field, and paper hope to extend it with a range of experiments it explain the behavior of the many aspects of this type of device and application in detail.

REFERENCES

[1] N. H. Amer, H. Zamzuri, K. Hudha, and Z. A. Kadir, "Modelling and Control Strategies in Path Tracking Control for Autonomous Ground Vehicles: A Review of State of the Art and Challenges," *J. Intell. Robot. Syst. Theory Appl.*, 2017, doi: 10.1007/s10846-016-0442-0.

[2] A. P. Aguiar and J. P. Hespanha, "Trajectory-tracking and path-following of underactuated autonomous vehicles with parametric modeling uncertainty," *IEEE Trans. Automat. Contr.*, 2007, doi: 10.1109/TAC.2007.902731.

[3] R. J. Aughey, "Applications of GPS technologies to field sports," *International Journal of Sports Physiology and Performance*. 2011, doi: 10.1123/ijspp.6.3.295.

[4] C. H. Yoo, W. J. Choi, I. Y. Chung, D. J. Won, S. S. Hong, and B. J. Jang, "Hardware-in-the-loop simulation of DC microgrid with Multi-Agent System for emergency demand response," 2012, doi: 10.1109/PESGM.2012.6345678.

[5] O. Khader and A. Willig, "An energy consumption analysis of the Wireless HART TDMA protocol," *Comput. Commun.*, 2013,

doi: 10.1016/j.comcom.2012.12.008.

[6] D. Apalkov, B. Dieny, and J. M. Slaughter, “Magnetoresistive Random Access Memory,” *Proc. IEEE*, vol. 104, no. 10, pp. 1796–1830, 2016, doi: 10.1109/jproc.2016.2590142.

[7] L. Pan, “An improved the DSR routing protocol in mobile ad hoc networks,” in *Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, ICSESS*, 2015, vol. 2015-November, pp. 591–594, doi: 10.1109/ICSESS.2015.7339127.

[8] S. Liu, Y. Yang, and W. Wang, “Research of AODV Routing Protocol for Ad Hoc Networks1,” *AASRI Procedia*, 2013, doi: 10.1016/j.aasri.2013.10.054.

[9] M. T. Jungwirth, D. Kumar, D. Y. Jeong, and R. E. Goodchild, “The nuclear envelope localization of DYT1 dystonia torsinA-ΔE requires the SUN1 LINC complex component,” *BMC Cell Biol.*, 2011, doi: 10.1186/1471-2121-12-24.