LA-UR- 09-00256

Approved for public release; distribution is unlimited.

Title: Sensor network based vehicle classification and license plate identification system

Author(s): Jan Frigo, ISR-3, Edward Rosten, Cambridge University, Vinod K. Kulathumani, West Virginia University, Sean Brennan, ISR-3, Eric Y. Raby, ISR-3

Intended for: IEEE INSS 2009, Pittsburgh,PA, November 17-19, 2009

Los Alamos
NATIONAL LABORATORY
EST.1943

Sensor network based vehicle classification and license plate identification system

Jan Frigo*, Vinod Kulathumani†, Sean Brennan*, Ed Rosten*, Eric Raby*
*Distributed Sensor Networks Group
Los Alamos National Labs
†Dept. of Computer Science and Electrical Engineering
West Virginia University

Abstract—**Typically, for energy efficiency and scalability purposes, sensors networks have been used in the context of environmental and traffic monitoring applications in which operations at the sensor level are not computationally intensive. But increasingly, sensor network applications are requiring data and compute intensive sensors such video cameras and microphones. In this paper, we describe the design and implementation of two such systems: a vehicle classifier based on acoustic signals and a license plate identification system using a camera. The systems are implemented in an energy-efficient manner to the extent possible using commercially available hardware, the Mica motes and the Stargate platform. Our experiences lead us to consider an alternate more flexible, modular, low-power mote architecture that uses a combination of FPGAs, specialized embedded processing units and sensor data acquisition systems.**

Keywords: wireless sensor networks, seismic, acoustic vehicle classification, license plate detection

I. Introduction

Sensor networks have been widely used in environmental monitoring and traffic monitoring applications. Typically, in order to be energy efficient and long lasting the operations at the sensor level are not computationally intensive. These devices operate at extremely low duty cycles and perform trivial operations. The data intensive operations are then left to one or more centralized units. However, there are scenarios where in order for sensor networks to be applicable, computationally intensive operations are unavoidable. For example, consider a network of cameras deployed in an airport in order to identify / recognize potentially harmful activities or people. As another example, consider a network of sensor deployed along a roadway to identify and verify traffic in a limited access facility. In order to be scalable, such systems need to work in a distributed mode where data is not processed at a central unit. At the same time, the data collected at the local sensors is quite intensive and often significant processing is required in-situ. The question then arises, how to design sensor network systems for energy-efficiency, ruggedized environments and autonomous operation?

In this paper, we describe the design, development and implementation of two deployed sensor network applications; (1) Real time classification of vehicles using commercial off-the-shelf (COTS) Mica motes and Stargates (2) Real time license plate classification using a low power camera and Stargates

The scenario for vehicle classification is that vehicles carrying heavy loads have to be identified with extremely low latency and high accuracy as they pass a sensor network based check point. The sensor network consists of 5 to 10 sensors. The check point is intended to be deployed along two-lane roadways, where vehicle arrivals are quite infrequent. We use acoustic signature based classification. This involves sampling at about 4 kHz and some intensive computations like the FFT. We use the Intel XScale processor based Stargate for this purpose. But a continuous sampling at this rate on the Stargate will drain the batteries in a number of hours. So we use a Mica2 mote to sample a seismic sensor (at less than 100 Hz) and employ a low complexity wavelet-based algorithm to detect an approaching vehicle. The seismic detection scheme triggers the acoustic sampling and processing on the Stargate. The Stargate uses a Fisher Linear Discriminate Vector (FLDV) based algorithm to classify the vehicles. We describe the details of this algorithm, the performance and energy utilization in Section 2.

For the license plate classification, the system detects the license plate from an image of the approaching vehicle at the access point and matches it with a list of target license plate numbers. In this scenario, we use a magnetometer with low sampling rate to act as a trigger for the camera. We then extract the license plate information from the extracted image using a Viola-jone object detection algorithm and the ID3 decision tree running on the Stargate. The details of this algorithm, the performance and the energy utilization are discussed in Section 3.

Our experiences developing these systems lead us to explore a more flexible, modular, low power DSN system. We would like to have a node architecture that separates the *real-time* sensor data acquisition, data processing and network communications processing, thus, resulting in a more flexible modular node architecture. Others have proposed such modular systems [?], however, our architecture *processing at-the-sensor* would use a custom reconfigurable or embedded processor based *sensor processing module* that is suitable for an energy-efficient, compute-intensive application. The network communication interface is standard, i.e. these sensor processing modules plug into a common interface to communicate with the rest of the network. A system with diverse types of sensors nodes such as video cameras and seismic sensors with very different bandwidth and computing requirements could be implemented using the same sensor processing hardware board connected to a common interface. Power is not wasted because an under-utilized reconfigurable processor can "turn off" gates that are unused or an embedded processor can go into sleep mode as required. By implementing high-performance, energy-efficient *processing at-the-sensor*, the system can achieve an improved network response-time and an extended life-time in a natural environment.

Outline: In Section 2, we describe the vehicle classification system. In Section 3, we describe the license plate detection system. In Section 4, we discuss our proposed modular node architecture and compare it with existing work. We conclude in Section 5.

II. Seismic-Acoustic Based Vehicle Classification System

In this section, we describe a real time vehicle classifier system for traffic monitoring, developed using seismic and acoustic sensors connected to a Mica2 sensor and a Stargate respectively. The goal of this system is to classify vehicles as they approach a specific region into 3 categories: (1) a light-weight vehicle such as a compact car, (2) a moderately heavy vehicle and (3) a very heavy vehicle. For our training and testing we chose a compact *car* [1], a *truck* [2], and a *HumV* [3] as representative vehicles of each category respectively. We assume that vehicles do not enter the monitoring area concurrently.

Challenges: The vehicles travel between 10 to 40 mph and stay within the influence region of the sensors for 8 to 10 seconds. However the spectral signature of the vehicle changes over time. It is only for a very short time of 1 to 2 seconds when the vehicle is at a sufficient distance from the sensors when spectral analysis yields accurate classification. But the classifier has no knowledge as to when the vehicle is closest to the sensor. Yet, the requirement for the classifier is an accuracy of over 99%.

Related work: The use of sensor networks in seismic and acoustic signal processing for vehicle and/or personnel detection and classification is a widely published area of research [15], [9], [11]. The approaches to vehicle detection vary from a very detailed frequency analysis to determine vehicle weight, number of cylinders and gears, and the type of fuel used by the engine [5] to target detection of different types of vehicles [8] and to classification and tracking using various sensors and statistical learning algorithms [15], [6]. Most DSN field experiments deploying seismic and acoustic sensors use hardware such as Crossbow's or Moteiv's mote technology [6], [14] along with a custom interface. In [3], the authors describe a vehicle classification system using a network of acoustic sensors using feature vectors formed by spectral characteristics at a local node and then combining the hypothesis at a central base station.

Most of these existing solutions using spectral characteristics assume a continuous sampling of the order of 1-4 KHz on the acoustic channel. Note that for our application requirements, a quick processing is required and therefore an extremely resource constrained device like a Mica mote is not suitable. At the same time, continuous acoustic sampling on a device like a Stargate is not energy efficient. So, here we use a low complexity wavelet algorithm for vehicle detection on the Mica mote and for the higher sampling rate and higher complexity classification algorithm we use the Stargate.

The existing solutions also assume that the closest point of approach is known for a vehicle, at which time the classification is performed. However, in the real deployment scenario, the spectral characteristics vary as the vehicle approaches and passes a sensor, thus, creating false classification results. In our algorithm, we handle this by choosing our feature vectors for different classes in such a way that an order is imposed. So if a vehicle is identified as a series of classes when it is within the region of detection, the class can be chosen as the *highest* in rank.

[1] 1994 Honda Accord LX, manual drive

[2] 2006 Diesel Chevy C4500 4x4

[3] 1994 H-1 with a 6.5 L Detroit Diesel Engine

A. System setup

The seismic-acoustic node configuration is shown in Figure 1. The seismic sensor is a 4.5 Hz geophone (GeoSpace Technology, GS-11). The geophone is placed 50 feet from the road to eliminate acoustic feedback in the sensor. The geophone is connected to a custom signal conditioning interface board and to the Mica2 mote via the Crossbow MDA320CA 16-bit A/D board. The acoustic sensor is a C01U - USB Studio Condenser Microphone. The microphone is placed 10-12 feet from the road and mounted 1 foot off the ground. The microphone is connected directly via a USB port to the Stargate (400 MHz, Intel PXA255 Processor, Linux based). The microphone has directional response and is mounted facing toward the roadway. Samson windshields on the microphones help filter wind noise.

The implementation uses the Crossbow Mica2 mote to trigger an "event" based on seismic information. The event is transmitted to the Stargate processor over the 900 MHz radio link on the Mica2 mote. The Stargate then samples the microphone and processes acoustic information and sends a classification to a base station visualizer (via an 802.11 network). We use this method of seismic detection triggered acoustic sampling and processing because it is an energy-efficient way of fusing the multi-sensor information to yield a single classification, i.e. the seismic detection runs continuously at approximately 60 mWatts. Moreover, the frequency characteristics for seismic detection show very similar peak frequencies (see Figure 2) so more frequency analysis would have to be done in order to develop an accurate classification and the Mica2 does not have sufficient computing capabilities to do this. For these reasons, we choose to combine both seismic and acoustic sensors to achieve a more reliable, energy-efficient classification.

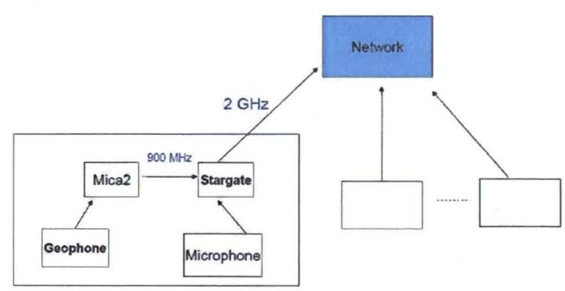

Fig. 1. Seismic-Acoustic Node Configuration

B. Seismic detection

The geophone is sampled at 100 Hz. The Mica2 mote computes the Haar Wavelet on a moving window of 128 samples every 10ms using 10 new samples each round and 118 samples from the previous round. The Haar wavelet is computed up to level 2 which computes the energy estimate of the 12-24 Hz band via the average of the coefficients of this band. The variance of the energy estimate is computed on a moving window of size 20. A variance threshold is used for vehicle *event* detection. A trigger is sent to the Stargate over the 900 MHz radio link when a vehicle detection has occurred.

The Haar Wavelet is chosen for its low-level of computational complexity which was required due to the 8-bit computing capability of the ATMEL processor on the Mica2 [6] and also because of the narrow peak frequency observed for all the vehicle categories. The frequency characteristics of the seismic data are given in Figure 2.

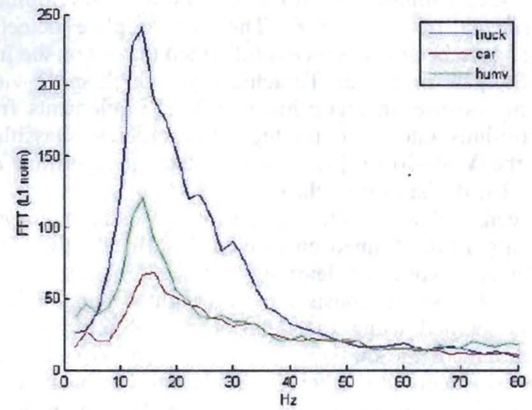

Fig. 2. Frequency characteristics of seismic detection

C. Acoustic classification

Upon receiving a trigger, the microphone is sampled at 4kHz. There are four sources of sound collected by the acoustic sensor, i.e. road/tire, engine, mechanical and air current noise. For the classifier, a 512-point integer FFT is implemented on the Stargate. The 512-point FFT is computed every 125 ms to obtain the spectral characteristics of the data, yielding an 8 Hz resolution. Frequencies lower than 64 Hz are not used due to variations in the microphone response and temporal variations (wind) at these lower frequencies.

We first obtain training data sets using multiple runs through each vehicle at different speeds. We use the samples collected during the 2 seconds when the vehicle is closest to the microphone for the training. We then identify the ideal feature vector set to do the classification between each pair of vehicles. For example to classify between a car and a truck, we use a 10 coefficient vector, formed by the average energy of 10 equally spaced bands in the 224 Hz to 368 Hz range because the spectral characteristics of the truck shows a distinctive spike in response at those frequencies. We use Fisher Linear Discriminant Vector analysis to identify the best projection vector given the training data. We obtain a similar projection vector to distinguish whether the vehicle is a Humv or a car/truck. These projection vectors are computed offline in matlab and then copied into the classifier program running on the Stargate.

In every round, the Stargate simply computes the dot product of this vector with the feature vector obtained in that round to perform the classification. The first classification is whether the event is a false positive. The Stargate then classifies whether the vehicle is a Humv or either of car or truck. Then it classifies whether the vehicles is a car or a truck. We find that this is the classification order that maximizes the distance between classes.

D. Integration of output

Once triggered the classifier operates for 3-4 seconds as the vehicle passes the sensor (until the seismic sensor sends a signal to stop classification). The acoustic classifier is operating in real-time and generates a classification output once every 125ms. The characteristics as an approaching vehicle differ from when the vehicle is at the closest point. We choose our feature vectors for different classes in such a way that an order is imposed. Thus if a vehicle is identified as a series of classes when it is within the region of detection, the class can be chosen as the *highest* in rank. For our set of classes, the order is *car, truck, HumV.* In other words, our feature vectors are such that there is an extremely low probability of a car being classified as a truck or HumV and a similar low probability of a truck being classified as a HumV during the entire region of detection.

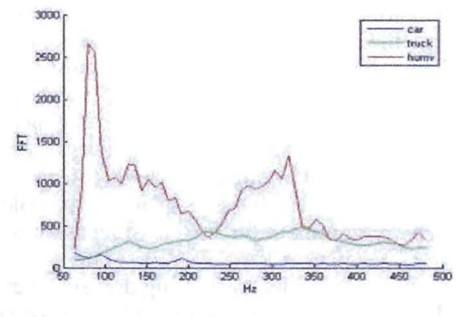

Fig. 3. Mean FFT coefficients for 3 vehicle classes

E. Performance

This initial work on the acoustic classifier also showed zero misclassification's in 10 test runs of each vehicle class. Note that, we maintain our testing environment to be similar to the training environment. We also ensure that vehicles enter the field one at a time. Training the classifier in a dynamic manner to different environments is a much more difficult problem and is a subject of future work.

We are using rechargeable 5v, 4800 amp-hour, NiMh batteries for the acoustic and seismic processing. The measured current for this acoustic node is 420 – 470 mamps. Thus, a total of approximately 2.35 Watts is utilized for the acoustic classifier over a period of 8-10 seconds. The energy utilization for the acoustic classification is then 7.05 to 9.4 Joules. At this rate, the Stargate based classifier can last for about 1000-1050 vehicle detections before the battery runs out. The seismic detection runs continuously at approximately 60 mWatts, i.e. 3v, 20 mamps. The energy for the seismic detection over 10 seconds of operation (which is the approximate time it takes a vehicle to approach and pass the node) is 0.6 Joules. "can you verify my estimate of how many vehicles may be classified before the acoustic battery runs out?"

III. Video Based License Plate Detection Node

The video sensor node performs license plate detection of a vehicle traveling on a roadway using a camera, a magnetometer and a Stargate. The system requirements are as follows: to capture a 640x480 pixel image of the aft end of a vehicle at anticipated vehicle speeds of 10 to 60 mph, to extract the license plate pixels only from the original image, thereby reducing the original image by approximately 60-90%, and to convert the license plate image to text with 99% accuracy using optical character recognition.

Challenges: The vehicles travel between 10 to 60 mph and stay within the influence region of the camera for 8 to 10 seconds. Thus, the image capture routine must be able to store a number of images quickly and select the best image among these to be processed. The camera is facing east, so the changing position of the sun creates various degrees of glare in the image. In addition, the changing seasons, yield a different background, i.e. all white in winter to colorful in summer. Yet, the requirement for the classifier is an accuracy of over 99%.

Related work: In the video processing domain, specialized hardware and processors for embedded video sensor systems exists on the market today such as Analog Device's Blackfin DSP, TI's OMAP5910 and OMAP5912, and Freescale's i.MX31 and i.MX27. DSN video processing has been implemented on specialized platforms [?] and on the Stargate for many applications such as low resolution image registration [2], fast image motion computation [7], and face detection [13]. The level of computational complexity for the application we have described is a challenge for the resource constrained Stargate.

A. System Setup

The license plate detection system consists of a webcam (with a 12 mm telephoto lens) connected directly to the Stargate via the USB port. A magnetometer based detection algorithm is used as a trigger for image capture. A learning algorithm running on the Stargate converts the selected image to license plate pixels only. The sensors and Stargate are mounted on a tripod located approximately 10 feet away from the road and 3 to 4 feet off the ground. To eliminate glare, the assembly is slanted at about a 45 degree angle to the road.

B. Image Capture

Figure 4 below is a typical vehicle image captured from the video node. We chose to use the webcam due to its ease of integration, low power, low cost and compact size. In addition, the image resolution with our telephoto lens was sufficient for the learning algorithm. The best range for image capture is within 8 to 15 feet from the camera. The magnetometer triggers on the front end of the vehicle and we capture images for 5 seconds at approximately 10 frames/sec. From this set of frames, an optimization routine selects the image with the largest number of license plate pixels. This frame is processed by the learning algorithm.

Fig. 4. License Plate Image Capture

C. License Plate Detection Algorithm

The software on the license plate detection node works by applying a classifier to every pixel in an image to create a rough segmentation of the license place, if it exists. From this, the bounding box of the license plate is found, and that section of the image is then resampled to a fixed size. The resampled image is used to perform the OCR to retrieve text.

The classifier is trained using data collected during typical field operation. This consists of various vehicles viewed at distances of between 8 and 75 feet from the camera, as well as some 'background' images containing no license plates. A 8 bpp greyscale image is used for the algorithm development.

1) Decision tree classifier: The license plate detection software has to be able to process full video images on the Intel XScale Stargate processor. To achieve very high speed video processing we use an algorithm which takes elements from two algorithms known to produce very efficient classifiers, namely the Viola-Jones [12] object detection algorithm and the ID3 [10] decision tree classifier.

The license plate detection is performed using a machine learning algorithm, trained on labeled data.Briefly, the Viola-Jones detector works by learning a cascade classifier. Each stage of the cascade consists of a weighted sum of weak classifiers learned using the AdaBoost[4] algorithm. If the weighted sum of the classifier results exceed zero, the stage classifies as foreground, otherwise it classifies as background. The weak classifiers are convolution kernels combined with a threshold: if the image convolved with the kernel exceeds the threshold, the classifier value is 1, otherwise it is -1. The kernels consist of a number of rectangular regions of uniform value. The use of the integral image allows the sum of pixel values within a rectangle of arbitrary size to be computed in constant time.

The goal of our algorithm is to not just detect, but also segment license plates, and in some cases, it can consist of a significant fraction of the image. As a result, we have found that a decision tree produces a more efficient classifier. Since decision trees are a richer classifier than cascades (cascaded are degenerate trees), the classifiers at each node need not be as strong, and so they can contain fewer features. This allows us to compute whether the weighted sum exceeds zero using a lookup table. Secondly, because there can be a large number of foreground pixels, the average depth of the tree is lower than the average depth required of a cascade, so pixels are classified in fewer operations.

2) Bounding box and resampling: The classifier is applied to every pixel of the image, and the number of pixels classified as a license plate in each row and column are recorded. Each row/column has a position and count associated with it, so the median (or any rank) row/column can be efficiently found with a linear search. The license plate is taken to initially lie between the $25th$ and $70th$ percentile horizontally and vertically. The width and height are grown a pixel at a time until the change in rank becomes small.

This algorithm is very robust to misdetections, both in terms of missing regions on the license plate, and false detections in the background. It is also very efficient, even when the license plate is a significant fraction of the image, since it avoids relatively expensive operations such as connected components analysis and hole filling. Finally, if the rectangle is unlikely to be a license plate, or contain a license plate which is too small it is rejected. The rectangle must reach a minimum size and pixel density to be considered.

Once the extent and therefore the bounding box of the license plate has been found, the rectangle is then resampled to a fixed size. Size reduction is performed by averaging rectangles of pixels to create a single pixel. This is performed efficiently by using the integral image already computed for the classifier, and the time is only dependent on the size that it is being resampled to. We do not upsample the image: if the bounding box is smaller than the desired size, we use the image within the bounding box without alteration. Once the license plate has been extracted, it is compressed using the standard 'libpng' library.

Figure 5 shows the output of this detection algorithm.

Fig. 5. License Plate Detection Result

D. Optical Character Recognition

The OCR application converts the license plate image to text. This information is then checked against a database of authorized vehicle license plates numbers. The Open Source OCR applications available for Linux [1] did not prove to be as reliable compared to the available COTS Windows products, therefore, the OCR is performed currently at the base station. We note that, running the OCR on the Stargate at 400 MHz would take up to 2-3 seconds as compared to less than 0.5 seconds on the base station 2 GHz Pentium. Currently, the OCR is being tested with classifier results from our training database.

E. Performance

On the Stargate XScale license plate detection node, the run-time performance is as shown in Table . The majority of the energy is consumed by the image capture routine. The idle power for the video node (with the webcam connected

Accuracy measurements for this license plate detection node were taken with two vehicles: a compact car, and a 2008 Diesel 4x4 pickup truck. (DonŠt remember what the accuracy was.) Future work for this node involves testing the output results with our OCR application.

	Time	Pwr	Energy
	s	W	J
Image capture	5	2.6	13
Detection	0.615	2.375	1.46
Resampling	0.0034	2.375	0.008
Compression	0.0173	2.375	0.041

TABLE I
ENERGY FOR LICENSE PLATE DETECTION PROCESSING

IV. MODULAR ENERGY-EFFICIENT SENSOR NODE ARCHITECTURE

In the previous two sections, we described the design and implementation of two applications of sensing systems using existing hardware to build such systems. Both these applications have some common characteristics: events occur rarely, event processing needs to be fast and reliable and requires significant computation and the system needs to be energy efficient. We note that the existing COTS hardware platforms, the motes and the the slightly more resourceful Stargate class of devices allow easy programming and by using a combination of the two we are able to decrease the *on* time for the Stargate and therefore be more energy efficient. That said, we also observe that there is room for improvement. Using custom reconfigurable, embedded processors can provide far more energy-efficiency for such typical sensor network based monitoring applications.

In this section we describe our proposed node architecture that is suitable for sensor networks systems where events are rare, but when they do occur there is significant computational complexity. Moreover, sensor types and processing are usually multi-modal with each type requiring different system resources. Our proposed architecture is designed to be ruggedized for a deployed natural environment, low power, modular, an experimental platform capable of interfacing to a wide variety of processing and sensor hardware. It is designed for ultra-low power data acquisition as well as in network processing.

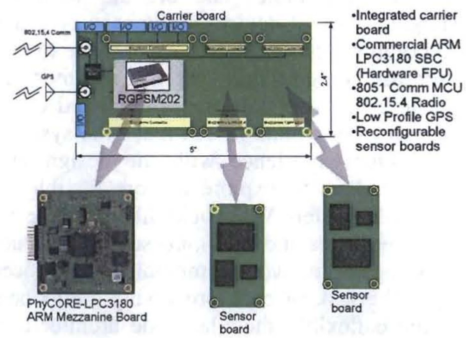

Fig. 6. Proposed modular node architecture

The architecture as shown in Fig. 6 combines a low power high performance ARM microcontroller mezzanine board 4, an embedded GPS module, and a Texas Instruments CC2431 wireless chip, with a variety of sensor interface options. The mezzanine carrier board's wireless subsystem consists of a single self contained COTS wireless system on chip (SoC), a CC2431, containing an embedded 2.4GHz 802.15.4 compliant radio, 32MHz 8051 microcontroller, 8KBytes RAM, and 128KBytes flash storage, as well as hardware accelerated encryption, location computation and MAC layer functionality. Additionally, a GPS module is mounted to the carrier board whose power and operation are controlled by the CC2431's embedded 8051 microcontroller. The CC2431 development tools consist of a C compiler and assembler for straight forward algorithm development. The carrier board can fully operate without the ARM mezzanine board if a high performance co-processor is not needed. This architecture will allow us to replace the ARM microcontroller, with a DSP, FPGA, or any other low power co-processor as the application dictates.

V. CONCLUSION

In this paper, we describe the design, deployment and demonstration of two sensor network applications (1) Real time classification of vehicles using commercial off the shelf Mica motes and Stargates (2) Real time license plate classification using low power cameras and Stargates. The vehicle classification system analyzes spectral features of real-time seismic and acoustic data as a vehicle approaches and passes the node. We use a low complexity wavelet algorithm for detection using a Mica mote which triggers a higher sampling rate and higher complexity classification algorithm on the Stargate. The classification uses the spectral features of the vehicle classes to compute ideal projection vectors for Fisher Linear Discriminant Vector analysis, which are fed into the Stargate to perform real time classification. Each vehicle stays within the detection region of a sensor for under 2 seconds, but this is sufficient for the system to classify with 100% accuracy the. The license plate detection system uses a camera to capture vehicle images and an efficient learning algorithm to reduce the original image to license plate pixels. This system has a letncy of around 2 seconds for processing the image.

Both these applications have some common characteristics: events occur rarely, even processing needs to be fast and reliable and requires significant computation and the system needs to be energy efficient. We note that the existing COTS hardware platforms, the motes and the the slightly more resourceful Stargate class of devices allow easy programming and by using a combination of the two we are able to decrease the *on* time for the Stargate and therefore be more energy efficient. That said, we also note that specialized embedded processors for heavy computations can make the system more energy efficient. Our experience with the design of these two applications lead us to explore a more flexible, modular, low-power DSN system.We would like to have a node architecture that separates the *real-time* sensor data acquisition, data processing and network communications processing, thus, simplifying the task of each processing component and resulting in a more flexible modular node architecture. The *processing at-the-sensor* would use a custom reconfigurable or embedded *sensor processing module* that is suitable for the specific sensor requirements. The network communication interface is standard, i.e. these sensor processing modules plug into a common interface to communicate with the rest of the network. A system with diverse types of sensors nodes such as video cameras and seismic sensors with very different bandwidth and computing requirements could be implemented using the same sensor processing board connected to a common communication interface. Power is not wasted because an under-utilized reconfigurable processor can "turn off" gates that are unused or an embedded processor can go into sleep mode as required. By implementing high-performance, energy-efficient *processing at-the-sensor*, power savings and improved network response-time can be realized. This will be a subject of our future work.

REFERENCES

[1] A. Acton. Linux ocr: A review of free optical character recognition software. *http://groundstate.ca/ocr*, May 2007.

[2] e. a. Balasubramanian, Harini. Image registration in low resolution visual sensor netowrks. 2008.

[3] M. Duarte and W. Hu. Vehicle classification in distributed sensor networks. *Journal of Parallel and Distributed Computing*, 64:826–838, 2004.

[4] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and anapplication to boosting. In *Computational Learning Theory*, pages 27–37, 1995.

[5] R. A. Gramann, M. B. Bennett, and T. D. OBrien. Vehicle and personnel detection using seismic sensors. *SPIE Conference on Sensors, C3I, Information and Training Technologies for Law Enforcement, vol. 3577*, Nov. 1998.

[6] e. a. Gu, Lin. Lightweight detection and classification for wireless sensor networks in realistic environments. *SenSys 2005*, Nov. 2005.

[7] X. Lu and R. Manduchi. Fast image motion computation on an embedded computer. 2006.

[8] A. Pakhomov and T. Goldburt. Seismic systems for unconventional target detection and identification. *Proceedings of SPIE, Sensors, and Command, Control, Communications and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense V, vol. 6201*, Nov. 2006.

[9] G. Prado. Acoustic-seismic sensors: Past experiences and future prospects. *Processings of SPIE, vol. 5611, p. 117-124*, 2004.

[10] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

[11] e. a. Succi, George P. Problems in seismic detection and tracking. *Processings of SPIE, vol. 4040, p. 165-173*, 2000.

[12] P. A. Viola and M. J. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 511–518, Lihue Kauai, Hawaii, USA, June 2001. Springer.

[13] Y. Weng and A. Doboli. Smart sensor architecture customized for image processing applications. Aug. 2004.

[14] G. Werner-Allen, O. Marcillo, J. Johnson, M. Ruiz, and J. Less. Deploying a wireless sensor network on an active volcano. *IEEE Internet Computing*, Mar. 2006.

[15] H. Xiao and et. al. Military vehicle classification via acoustic and seismic signals using statistical learning methods. *International Journal of Modern Physics C, vol. 17, no.2, pp. 197-212*, 2006.