# SANDIA REPORT

SAND2004-4346
Unlimited Release
Printed September 2004

# On Computer Vision in Wireless Sensor Networks

Teresa H. Ko and Nina M. Berry

Prepared by Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

**NOTICE:** This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865)576-8401
Facsimile: (865)576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800)553-6847
Facsimile: (703)605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2004-4346
Unlimited Release
Printed September 2004

# On Computer Vision in Wireless Sensor Networks

Teresa H. Ko and Nina M. Berry
Embedded Reasoning Institute
Scalable Computing Research and Development
Sandia National Laboratories
P.O. Box 969, MS 9158
Livermore, CA 94551-0969
{thko,nmberry}@sandia.gov

### Abstract

Wireless sensor networks allow detailed sensing of otherwise unknown and inaccessible environments. While it would be beneficial to include cameras in a wireless sensor network because images are so rich in information, the power cost of transmitting an image across the wireless network can dramatically shorten the lifespan of the sensor nodes. This paper describe a new paradigm for the incorporation of imaging into wireless networks. Rather than focusing on transmitting images across the network, we show how an image can be processed locally for key features using simple detectors. Contrasted with traditional event detection systems that trigger an image capture, this enables a new class of sensors which uses a low power imaging sensor to detect a variety of visual cues. Sharing these features among relevant nodes cues specific actions to better provide information about the environment. We report on various existing techniques developed for traditional computer vision research which can aid in this work.

# Acknowledgment

We would like to thank Philip Heermann for supporting of fundamental research between wireless sensor networks and imaging. His ability to see the promise of the integration of computer vision and wireless sensor networks is greatly appreciated.

We would like to thank Regan Stinnett for his constant belief and innovative approaches to wireless sensor networks. His insight into the real customer needs was critical in shaping the scope and direction of this work.

We would like to thank Ron Kyker for the initial discussions of imaging's role in wireless sensor networks. His knowledge of the hardware constraints were invaluable to formulating a feasible plan of attack.

# Contents

1 Introduction . . . . . . . . . . 7
2 Motivation . . . . . . . . . . 7
2.1 Enabling Technologies . . . . . . . . . . 8
2.2 The A in SDAC . . . . . . . . . . 9
3 Body of Research . . . . . . . . . . 10
3.1 Constraints . . . . . . . . . . 10
3.2 Fundamental Questions Addressed . . . . . . . . . . 11
3.3 Framework . . . . . . . . . . 13
3.4 Enhanced Capabilities . . . . . . . . . . 15
4 Preliminary Suvery . . . . . . . . . . 16
4.1 Features . . . . . . . . . . 16
4.2 Similarity . . . . . . . . . . 17
4.3 Learning . . . . . . . . . . 17
4.4 Hardware . . . . . . . . . . 18
5 Conclusion . . . . . . . . . . 18
References . . . . . . . . . . 25

# Figures

1 framework . . . . . . . . . . 14

# Tables

1 Image Features . . . . . . . . . . 17
2 Blob Features . . . . . . . . . . 18
3 Blob Finding Features . . . . . . . . . . 19
4 Tracking Features . . . . . . . . . . 20
5 Similarity Measure . . . . . . . . . . 20
6 Supervised Learning . . . . . . . . . . 21
7 Unsupervised Learning . . . . . . . . . . 22
8 Hardware . . . . . . . . . . 22

# On Computer Vision in Wireless Sensor Networks

## 1 Introduction

Wireless sensor networks (WSNs) are increasingly called upon to assist in securing our nation. Because of the small form factor, wireless communication, and battery power of an individual node, it is easily deployable in difficult to monitor environments and at a scale not previously feasible. To fully exploit the capabilities of WSNs, this paper asserts the benefits of distributed computation and imaging. Imaging provides orthogonal information from the traditional sensors considered in small low-power WSNs and distributed computation allows for rapid information extraction at the source of the events. The motivation for this work is described in Section 2. Section 3 contains a proposed approach to address the challenges of embedding computer vision techniques in a distributed manner through a WSN. A preliminary survey of existing image-based information extraction techniques is summarized in Section 4.

## 2 Motivation

WSNs are typically thought of as a data collection tool. Data is funneled through an ad-hoc network of low-power microprocessors with embedded sensors to a centralized base station. This base station has infinite power and memory resources compared to the distributed sensor nodes enabling complex processing on the incoming data and interface with an end user, [14, 18, 1]. When dealing with small amounts of information at controlled intervals, as in Kyker's work [14] which sampled radiation levels or Polastre *et al.*'s work [18] which sampled various environmental conditions, a centralized data poll is feasible on the limited available bandwidth. Yet, inherently, this is not a scalable architecture. System bandwidth is proportional to the number of nodes connected to the base station, as well as dependent on the communication power of these nodes. These nodes are typically relatively few because the wireless sensor networks are placed in large and inaccessible areas. The location of the base station can often far from the center of activity. The full processing power of the base station cannot be utilized because not enough data can be received, and the full sensing power of the nodes cannot be utilized because of the bottleneck at the base station. In a recent field test [1], Arrora *et al.* illustrated how centralized data fusion is capable of flooding the network layers and provide little to no information to the end user.

## 2.1 Enabling Technologies

Another limitation of WSN is their inability to readily incorporate image sensors into their system. Yet as humans, we rely heavily on our sight because it provides rich orthogonal information compared to our other senses. There is much to leverage in research devoted to the exploitation of the information provided by images, from clustering of images to recognition/tracking objects within the images or video.

Recently, driven by the incorporation of cameras in cellular phones, there exists low power CMOS image sensors which consume approximately 2 millijoules to capture an image and have a form factor of less than 10 mm x 10 mm x 5 mm. Their form factor can be easily interfaced with current WSN platform.

Images captured by these sensors are around 100,000 bytes and require a larger memory and processing power than some sensor node platforms provide. Recent advances in XScale processing (e.g., Stargate [34] and PASTA [23]) provide adequate interfaces to be integrated into a wireless sensor node. The PASTA board [23] is particular attractive because of its modular design fitting well within the Embedded Reasoning Institute (ERI) framework for sensor systems [3]. A more powerful processor also means greater power consumption, so it is important that this module can be powered down and used only when needed. On a similar vein, Digital Signal Processing (DSP) boards are becoming increasingly low power and supported with key image processing functions to add in efficient analysis of raw data.

Researchers are beginning to see the possibilites that are opened up by these technological breakthroughs and investigating the possibility of embedded vision. Two examples of embedding vision within small devices is the work done by Rowe, *et al* [24], and Viola and Jones [32]. Rowe, *et al* have used an embedded 8-bit processor and image sensor to direct the motion of a small robot. Their research has proven the capability of rudimentary vision techniques in small embedded devices. We have extended this idea of embedded vision to extracting information in a distributed environment. Another example system is Viola and Jones's work in face detection on the XScale platform [32]. While their work demonstrates the size and complexity of an algorithm we can expect from an XScale platform, we recognize that even their work would be too computational expensive for WSNs. To be viable in wireless sensor networks, the XScale processor needs to remain in sleep mode as much as possible to conserve power. The architectural scheme used in [35] and [32] quickly reduces the required computation by first ruling out unlikely search spaces, a necessary concern when working within the power constraints presented by wireless sensor networks.

Another body of work led by Yang [37] has focused on crowd monitoring using WSNs. While not currently implemented on small embedded devices and assumes a known location and orientation of the cameras, the underlying research provides insight into distributed computer vision. Other work on the use of multiple cameras are Rahimi's work [20] in the simultaneously tracking of a single user and the discovery of node locations and orientation, and Khan's work [11] in tracking moving cars along a highway. While these approaches have taken a centralized approach in the acquisition and processing of images, their work illuminates the challenges of comparing images taken from different cameras.

Some system approaches has incorporated image sensors into the traditional wireless sensor network. Sensoria [17] relies on a simpler sensor, a Passive Infrared Sensor, to determine if an event of interest has occured, which then triggers an image capture and subsequent image transfer back to the base station. The SDAC system [12] proposed using an image as part of the decision making processing by performing simple local image processing techniques to extract key features which either refute or affirm the presence of an event and only transfer parts of an image which contains the event.

## 2.2 The A in SDAC

The network layers of WSNs have been optimized to handle transfers of small amounts of data between nodes. While investigations have begun on how to transfer large amounts of data for reprogramming of the sensor nodes [26, 5], these solutions can be inadequate for sensor system tasks due to time constraints. For example, [26, 5]'s methods take minutes to traverse a few hops. The cause for this delay steams from the limited bandwidth available and the unreliability of wireless communication. By the time the image is received at the base station, the event of interest (i.e., person) could have traveled far away and be out of detection range.

Clusters of computers working collaboratively are capable of providing greater computation than the fastest single computer. The strength of parallel or distributed computing has been shown throughout the high performance computing community. Yet, many of the current sensor network deployments do not attempt to utilize the processing power of the distributed nodes. In many ways, the nature of sensor networks lends itself easily to parallel and distributed computing because the sensed data is naturally distributed across the network.

The concept of Sense, Decide, Act, and Communicate (SDAC) developed by Berry *et al.* [3] describes the potential of WSNs. Rather than the centralized data pull that most research is focused on, SDAC-type systems are envisioned to have embedded intelligence at the sensor node level, placing decision making and acting at the souce of detection. In WSNs, it is essential to take advantage of the distributed processors because of the time and power constraints of these type of systems. There exists a tradeoff between the amount of instantaneous information the system can provide and how long the system can provide information in general. Assuming infinite bandwidth, constantly shipping the maximum amount of information back to a base station will provide greater resolution of the environment, but will also result in a shorter lifespan of the sensor network. Reducing the amount transmitted will increase the lifespan, but information will be missed. Introducing user feedback to this system can provide a varying amount of information, but the time constraint of monitoring real-life events most likely will be too tight to utilize this. *In situ* processing is necessary to improve the tradeoff between the resolution of information and the lifetime of the sensor networks [3]. *In situ* processing allows for sensor nodes to immediately comprehend the environment through their on board analysis of the sensor data. Through this analysis, they are able to perform specific actions to better inform the end user, by gathering more detailed sensed data, performing more complicated computation on the sensed data, or directing the activities of other nodes. These actions can be performed with little delay, capturing information that would be otherwise lost on time-sensitive events.

# 3 Body of Research

The future generations of wireless sensor networks and vision based systems are entwined. To realize the full potential of WSNs, there needs to be a method of capturing the information provided in images when they can not be transported back to the base station for centralized analysis. For acquiring cutting edge speed and robustness, computer vision research needs to take advantage of distributed dense information from different viewpoints. The following sections will describe the initial scope to begin this work, general research questions which arise when integrating these two fields, and the proposed architecture to address these issues.

## 3.1 Constraints

Our design of a distributed vision-based wireless sensor network has been governed by these underlying constraints:

- Unreliable communications exists between nodes.
- Processing and communications are power intensive operations.
- Memory is limited.

Wireless communication in real-life large-scale deployments is inherently unreliable and difficult to predict through simulations and testbeds. A number of issues arose in the Great Duck Island deployment [18] illustrating unexspected failures in the networking and application layers. Unreliable power-intensive communications suggests local processing, but local processing is constrained by the processors selected for WSNs and the memory limitations on sensor nodes [3].

These constraints suggest that a different approach is required than the traditional centralized data feed used in current WSNs and a centralized image processing used in computer vision. We propose work in developing a framework which exhibit these features:

- There is no need to share images between nodes or with the base station.

- Each nodes does not need to solve the entire problem. (e.g., execute all feature extractions.)
- Features are cascaded through the network.
- Features are selected and weighted according to their relevance.
- Computation is scaled and conservative.

The work presented in [12] illustrates information extraction at the sensor nodes so that images would not need to be transported back to a centralized base station. By extracting simple features from the images of an object passing through the view of the camera and comparing these features against feature templates associated with humans, we were able to differentiate times when a person was or was not in a scene. Specifically, the sensor node looked for skin colored objects and moving objects within its view point. We used the heuristic that an image of humans would have skin colored pixels and that the human would be moving in the scene as feature templates. While these features demonstrated an extremely compact algorithm in terms of memory and processing, more sophisticated features would provide finer granularity, reducing false postivies and misses, penhancing the ability to distinguish between different people. More sophisticated features are necessary to distinguish between different events as well as providing a thorough characterization of events across sensor nodes. This follows the same concept of AdaBoost [32], which uses cascaded detectors which become increasingly complex and detailed to quickly rule out false positives by using simple statistically discovered orthogonal features minimizing computation.

## 3.2 Fundamental Questions Addressed

Fundamental questions arise when integrating cameras within a WSNs, issues that computer vision research typically do not need to address. These three questions outline the key issues.

**Feature Detection**: *How can events of interest (e.g., person, group of people, tank movement) be described as a collection of features that are computational feasible within a wireless sensor network?*

The human vision system can distinguish between different objects (such as a tank and a person) because there are distinguishing features that each exhibit. The features which we perceive are in abundance and have unknown processes. Even if it were known, to attempt to replicate this system on a sensor node would be an intractable problem considering the tight power constraints and limited processing power. The state of the art research in computer vision has shown it is possible to characterize an object uniquely by detecting people within an image [32], or even characterizing different people by certain biometrics [15]. Even so, the high resolution of object descriptions are rarely coupled with real-time processing. The state of the art real-time systems typically incorporate prior information and sacrifice accuracy for robustness. While having proven success with relatively unlimited power and memory constraints running on a centralized Pentium 4 processor, it is unclear how these systems would perform in a constrained environment. Because of the high cost of processing images, efficient feature extraction that is kept to a minimum in terms of processor cycles and memory space is needed for successful computation.

**Object Resolution**: *How are these features manifested across differing images in time, differing position and orientation of cameras, and differing position and orientation of the event?*

While the first question addresses the constraints imposed by the physical hardware of WSNs, the second question addresses the constraints imposed by time and space. Events are moving in the field of sensors constantly changing their pose, position, and orientation. Distributed cameras are placed in random positions and orientations capturing different viewpoints of an event. This wide set of variables are rarely dealt with all together because many of the systems currently being developed are for environments where more assumptions can be made. A centralized calibration of a few cameras is often employed to help resolve these unknowns where the centralized computing unit has access to all the raw images between cameras. Without a centralized computing unit, the features chosen to interpret events need to be salient across these variables so that sensor nodes begin to understand their relationships to one another. In this context, a feature or a set of features are the representation of the event. This representation needs to be easily shared between nodes and interpretable independently of the raw image.

The attempt to characterize an object by features in a computationally constrained environment encourages the distribution of feature extraction to occur at different nodes to gather incrementally knowledge of an event, while attempting to correlate features with events pushes parallel extraction of features to resolve multiple detections of an event from the same object. Although the same features can be used to describe an event as to resolve multiple detections, there is an additional required step in resolving multiple detections. In this case, we need to define an appropriate similarity measure which takes into account the different views of the event. Typical similarity measures like mean squares might be too weak of a paradigm. Or rather, it could be that the features are not an appropriate representation. Finding the right representation for features is as critical as finding the right similarity measure.

**Aggregation of Distributed Information**: *How does a network of sensors fuse features across nodes with minimal communication to enable the classification and tracking events of interest?*

The last question addresses the problems that occur when dealing with information that is distributed across different units. To exploit the advantages of having multiple cameras dispersed throughout the environment, a method of effectively gathering aggregated information about the event and reporting this back to the user is essential. Each node should independently be capable of determining whether or not they are the current leader. Most leader election implementations for wireless sensor networks are not dynamic with the sensed data. For example, some methods use randomness for enhanced security [3]. Organizing the network based on sensed information allows for dynamic changes in leadership which are implicit, reducing the algorithm footprint in memory and processing. From [16], it is clear as long as sensor nodes are not identical, nodes can determine a leader without an external intervening force. The data received from various sensors differentiates the sensor nodes. Because of the power constraints of WSNs, it is critical that the features shared among the sensor nodes maintains this difference. While the challenges discussed above push for salient features which are detectable across time and space, the features which will be of use for efficient aggregation of the event description need to capture the differences from sensor node's viewpoint.

## 3.3 Framework

To maintain the adaptability and flexibility of WSNs, we propose a probabilistic framework for the representation of features. Recognizing that the environment in which WSNs are deployed is not predictable or easily accessible, using statistics allows features to be easily shared among different sensors and nodes. While each feature provides unique information in its own format, the detection of events lies in the combination of multiple values of multiple features. Statistics provide a common language in which the features can be represented. Information flows from the image captured from the CMOS image sensor through feature extractors, which represent the information in the image in a compact explicit form, and then is compared against other features for orrelating across sensor units and past knowledge as shown in Figure 1.

Features can be used to describe an image, describe an object, and differentiate an object from an image. Image description features provide information about the entire image, but do not actually locate the object in either the world or the camera coordinates. Typically, these features can be computed quickly, and therefore can be used as a first pass filter for reducing unnecessary computation. Similarly, features that describe an object can help differentiate events from one another. Though these features can be the same as the image features, their instantiation may be different due to a higher level representation of the event. To determine which pixels provide information about an event, it is essential to have features which can differentiate the event from the rest of the image. For example, these features could simple be the features that provide information about the event searched across the entire image. While effective, this method would be computationally expensive.

We rely on these features to represent the image throughout the sensor system. The information contained within the image about an event is transformed by feature extractors into an event description. Then, a quantitative measurement of similarity against other detected events by various similarity measures can be computed. The feature extractors and similarity measure are designed to be independent modules which can be easily selected and deselected for the appropriate application. In addition, these modules can be adjusted in real-time for adaptive computation for drilling deeper into the feature space and extracting more details about an event and adjusting for appropriate features given the dynamic

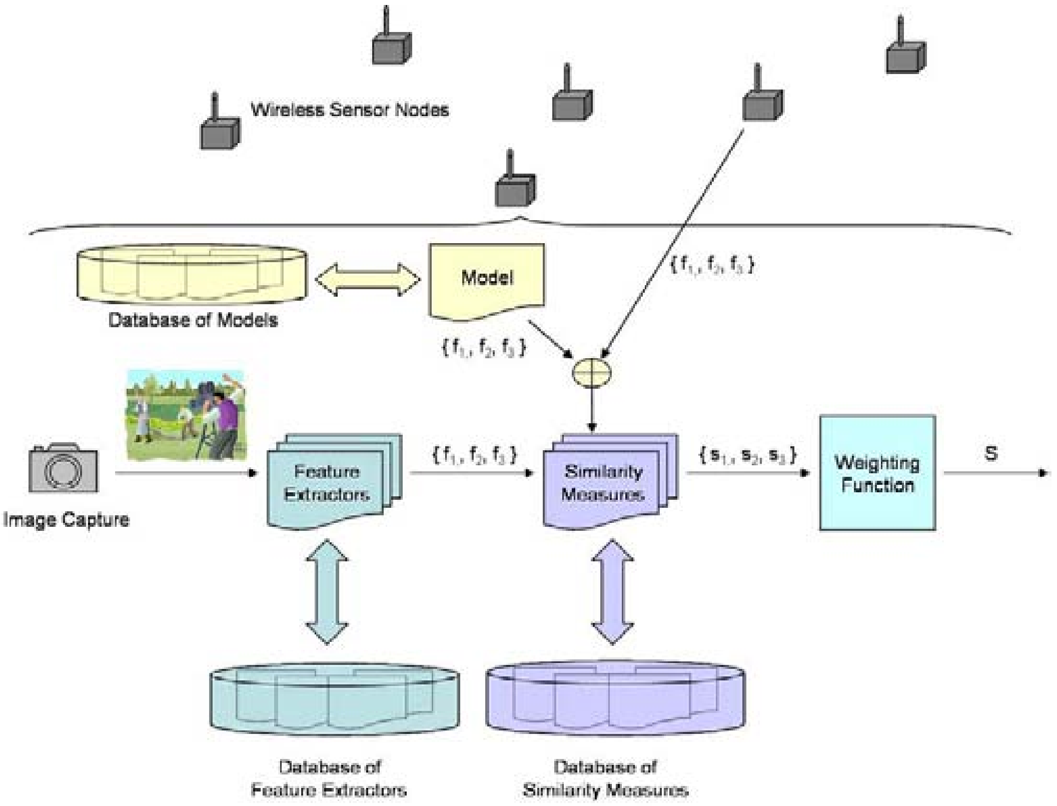

**Figure 1.** Decomposition of proposed framework. Processing begins with the capture of an image. Selected features are extracted from the image and compared against models built offline or incoming features from other sensor node sto determine correlations between events. The selection of features, similarity measures, models, and weighting function is dependent on previously extracted features.

environment.

The similarity measures computed from a set of features used to associate data can be viewed as a measure of an individual node's ability to detect events. In the simplest case, the sensor node which has the strongest response to an event would be the natural leader of the information. For concreteness, this could be the sensor node which is closest to the event. Future leaders can be hypthesized by detected trajectories. As the event moves, the leader naturally shifts to the sensor node which is closest. Leaders have the dual responsibility of getting information back to the user about the event and directing the feature extraction in a way that allows the accumulation of new information rather than only the reinforcement of known information. So even if the detection of an event by a single embedded processor can only tell that something of interest has occurred, by arming other sensor nodes with this information, over time collectively they are able to discover more detailed information, cascading the information through the network. Identifying the event, then resolving the event to a person, then a soldier, and then a solider who is running is possible through exploitation of the distributed processors.

To conserve power, the extraction of all possible features does not need to occur at each node. Because of the density of the image sensors throughout the environment, it is also not necessary to rely on only the information from a single image. By providing an adaptable similarity measure which adjust according to which features are extracted, we can attempt to maximize the relevant information extracted while minimizing computation. Features can be selected according to their relevance to a specific task and the environment. For example, if a small color range is detected, we can reduce the influence of color-based features on our classification of the event.

While adding imaging to WSNs has pushed the need for embedded processing, this approach is designed in a modular fashion to readily incorporate other sensors which can provide better or orthogonal footprints to be used for classification and tracking of events in WSNs.

## 3.4 Enhanced Capabilities

The application of computer vision research can provide several new or enhanced capabilities to current WSNs. General capabilities are discussed briefly below.

*Tracking* Automated target tracking has been a widely researched field, but results are still not reliable enough for integration into commercial systems. Most existing systems use a very limited number of cameras to monitor a small area in detail [6] or very large area with cameras placed at a distance [11] which is prone to error due to the low resolution of information and possible occlusion. Through the high sensing fidelity of WSNs, we believe that many of the difficulties of tracking can be mitigated, providing details across large areas. The areas occluded in one camera's viewpoint can be viewed through other cameras. More images provide greater information to reduce the errors in tracking estimations. Details of tracked objects provide a means of differentiating overlapping paths.

*Identification*. While some sensors systems [3] demonstrate the ability to use very simple sensors (e.g., acoustic, seismic, passive infrared) to detect activity within an area, the addition of an image sensor can further classify this activity. The work presented in [32] used a cascaded Adaboost with weighted simple Haar-like features to detect faces. The cascade allows the system to quickly remove unlikely candidates without first computing all possible features required in the model. This task can easily be distributed across nodes, where each sensor only computes one level of the cascade or even a subset of the features within each level. It seems likely that although we are generating an Adaboost model of specific events of interest in the traditional method and using the model on different segmented images, the performance should not be hurt due to the indepedence of the features from one another. Coordinating and reducing the feature extraction redundancy across different nodes is the central challenge of making Adaboost feasible to WSNs. Experiments may show that a partial decision from a random subset of features in each level at a single node may be all that needs to be shared across the network for a consensus as to whether or not to proceed to the next level.

*Scene Reconstruction* To achieve real-time scene reconstruction, sacrifices to accuracy is often made for speed. One of the state of the art offline systems, [19], illustrated a complete automated scene reconstruction algorithm, which has very few assumptions about intrinsic camera properties. It does have the assumption of small changes of viewpoints between frames, which would be difficult for distributed sensor network to fulfill. At the heart of this work is the ability to correlate extracted feature points across frames, but the probability of occluded points increases as the viewpoint difference increases. Intelligent handling of missing data is increasingly important not only due to the physical reality of occlusion but also due to the unreliability of the wireless link.

*Information-based Organization* In the attempt to resolve multiple detections of an event, the sensor node concurrently acquire an understanding of their environment. Sensor nodes, that were previously unaware of their location respective to one another, can hypothesize which nodes they share the same viewpoint by using the similarity measures associated with the features. Over time as more events are detected, these hypotheses will be reinforced or negated. In this way, the WSN becomes organized in overlapping clusters based on shared information. An important feature of this process is that over time, the WSN provides more information about the event as it traverses the environment and more accurate tracking as the system improves its informational clusters.

# 4 Preliminary Suvery

## 4.1 Features

The simplest feature would be the raw pixel values, or its grayscale equivalent. Though valid and useful, other features can make explicit certain characterisics that are embedded in the pixel values. Characterizing an image with simple features can provide valuable information to end users of a WSN. In some applications, such as scene reconstruction, we can use these features to provide a rough reconstruction, indicating if the camera is viewing buildings, open fields, natural scenes. Image features can also be used to measure the amount of information a particular image viewer is capable of providing, due to improper orientation or times of day. When trying to characterize a specific event within the image, many of the image features can also be used on the subpixels pertaining to an event. Table 1 and 2 outline some of the possible features that can potentially be extracted on a wireless sensor node. The challenge of partitioning the foreground blob pixels from the background pixels can be mitigated by using features outlined in Table 3. Correlating images across time to track events through space, different features listed in Table 4 can be used.

**Table 1.** Image Features

| Image Feature | Description | Citation |
|---|---|---|
| color histogram | Bins similar colors of an image the same bin. Differentiate between indoor and outdoor images. Determine if an image has colors associated with a particular event. | [12] |
| changes in color histogram | Bins only changes between two images. Changes indicate rapid changes do to events or gradually change do to the environment. | [12] |
| wavelet histogram | Same as above with a different measure. | |
| changes in wavelet histogram | Same as above with a different measure. | |
| frequency histogram | Same as above with a different measure. | |
| changes in frequency histogram | Same as above with a different measure. | |
| number of blobs | A high level feature which can be estimated by through the analysis of lower level image segmentation features and blob features. | |
| binary pixels | Set a threshold pixel value, and filter image into a binary representation. Quickly reduces memory footprint. | [37] [29] |
| zoning | Partition image into zones and describe zone with one pixel values. Quickly reduces memory footprint. | [29] |
| brightness | Computes the level of saturation in the image. The amount of useful information can be estimated of this value. | |

## 4.2 Similarity

Researchers have used a variety of similarity measures to compare features collected from different frames, sources, or models. Table 5 list a few of the possible mathematical formulations we coud use to measure how similar two events are to one another.

## 4.3 Learning

Computer vision relies heavily on machine learning techniques to provide higher level understanding. While there are many more than outlined in Table 6 and 7, this outlines some of the more prevalent algorithms currently used.

**Table 2.** Blob Features

| Blob Feature | Description | Citation |
|---|---|---|
| above image features | | |
| haar-like wavelets | After having computed an integral image,through one pass of the image, computing a haar-like wavelets cost only a constant number of operations. | [32] |
| haar wavelets in time | Same as above. | [33] |
| size | Size would be a good pruning mechanism for reducing false positives given that we have a calibrated camera and a horizon line. | |
| aspect ratio | A relative measure which can help characterize classes of objects. This would require capture of entire object. | |
| Coutour Energy | Need to further investigate computational complexity. This could be user for | [22] |
| Curvilinear Continuity | Higher level feature. Describes how continuous the segmentation is. The smaller the angle changes the more continuous the segmentation is. | [22] |
| texture | A possible feature to further classify or distinguish between objects. | [29] |
| Parallel Contrasting Rectangles | A subclass of the above. Useful and simple to calculate texture feature for quickly discriminate different objects. | [21] |
| Harris corners | A corner detector. Useful for determining corners of man-made objects, like buildings and cars. | [19] |

### 4.4 Hardware

The low-power solution for embedding image sensors into WSNs still requires further advances in sesnosrs and board design. Listed in Table 8, we list some of the state-of-the-art components which we have considered which can aid us in this vision.

## 5 Conclusion

We have presented a distributed approach to extracting features which is capable of reducing the computation and communicates at the sensor nodes, two critical factors in increasing the lifetime of the sensor system for a variety of potential applications. This work has been motivated by the need for WSN which is reactive, long-lived, and informative. The collaboration between sensor nodes is exploited within the network to facilitate intelligent distributed feature extraction, to discover information-based clusters, and to present the

**Table 3.** Blob Finding Features

| Blob Finding Feature | Description | Citation |
|---|---|---|
| Above Blob Features | Use features to search through image. Through effective, this option can be too computationally expensive, because computation scales by the sample rate rather than the event rate. | [32] |
| Edges | Only a simple edge detection algorithm which finds sharp contrasts is likely to be computationally feasible, but can provide helpful clues as to where to start looking for objects of interest. | |
| Colors | If known aprior, the color of the object can be used to locate objects of interest. | [35] [13] [24] |
| Color Changes | Changes in the image tells us that something has changed. It can be used an initial first pass filter. | |

user with higher level information about the system using less bandwidth and power.

**Table 4.** Tracking Features

| Tracking Feature | Description | Citation |
|---|---|---|
| motion field | The motion field is the perspective projection onto the image plane of the true 3-D velocity field of moving surfaces in space. Optical flows are calculated from teh first-order derivatives of the motion field, and sometimes does not provide a good estimate of the motion field. Need to further investigate computational complexity. | [31] [4] |
| direction of center of mass | A higher level feature, requiring the detection of the blob pixels. Direction of motion relative to the camera can help hypothesis absolute motion with several cameras. | |
| velocity | A higher level feature, requiring knowledge from several cameras or several features. Velocity may be estimated by the size change of an image as well as the direction of the center of mass. Velocity is critical when trying to estimate the future position when sampling at a low rate. | |
| visual hull | | [37] |

**Table 5.** Similarity Measure

| Similarity Measures | Citation |
|---|---|
| mean square distance | [29] |
| histogram difference | [29] |
| cumulative histogram difference | [29] |
| chi-squared measure | [22] |
| texture similarity | [22] |

**Table 6.** Supervised Learning

| Techniques | Description | Citation |
|---|---|---|
| Nearest Neighbors | Assigns class to data point according to its nearest neighboring data points. | [28] |
| Regresssion | Finds closest approximation to a function. Linear, Polynomial, Logistic variations. | |
| Linear Discriminant Analysis | Uses Bayes to determine a boundary between classes. Sometimes approximated with Gaussians. | |
| Expectation-Maximization Algorithm (EM) | Finds a good local maximum of likelihood as formulated by Bayes. | |
| Markov Chain Monte Carlo (MCMC) | A method of simplifing computationally difficult problems through the use of an random walks or sampling of the problem space. | [30] |
| Kalman Filter | A probabilistic framework which efficiently estimates the past, present, and future state of a process with the assumption of Gaussian densities. Variants include Discrete Kalman Filter and Extended Kalman Filter. | [27] |
| Particle Filtering | A probabilistic framework which efficiently estimates the past, present, and future state of a process with learned dynamic models. A generalization of the Kalman filter. Also known as Sequential Monte Carlo methods and Condensation. | [8] |
| Boosting | Combines multiple weak classifiers (slightly better than guessing) to form a single strong classifier). | [30] [2] |
| Bagging | Combines classifiers built from a collection of training sets to form a stronger classifier. | |
| Neural Networks | A collection of neurons which send simple scalar messages to one another based on weighted inputs from sensors or other neurons, to arrive at a global decision. | [25] [36] |
| Support Vector Machines (SVM) | Separates two classes of data by transforming the values into a higher-dimensional kernel space where they are separatible through a kernel function. | [13] |
| Graphical Models | Captures the depedencies of varables through probability. | [21] |
| Hidden Markov Models (HMM) | Uses observations to estimate the underlying state which produce these observations over time. Can be representedas a graphical model. | [21] [13] |

**Table 7.** Unsupervised Learning

| Techniques | Description | Citation |
|---|---|---|
| Density Estimation | Fit distributions to data. Gaussian, Kernel variations. | [7] |
| Mixture Models | Use multiple probability distributions to provide a better fit to the data. | [7] [9] |
| K-Means | Iteratively reestimates k means and reassociates the nearest neighbors to each mean | [21] [2] |
| Self-Organizing Maps | A constrained version of k-means clustering which creates a constrained topological map from higher-dimension data. | |
| Principal Component Analysis (PCA) | Transforms data points from correlated variables to (sometimes smaller) uncorrelated variables. | [10] [2] [36] |
| Indepedent Component Analysis (ICA) | Decouples data that arises from underlying independent sources. | [2] [36] |

**Table 8.** Hardware

| Hardware | Description | Citation |
|---|---|---|
| CMUCam | Low-power processor with embedded CMOS Sensor. An easy testbed system for featherweight signal processing. | [24] |
| PASTA | A USC-ISI developed XScale processor with a modular interface which matches with the ERI framework. Still in development phases. | [23] |
| Stargate | An Intel developed XScale processor which interfaces with the mote platform. An easy testbed for developing distributed algorithms. | [34] |

# References

[1] A. Arora, P. Dutta, S. Bapat, V. Kulathumani, H. Zhang, V. Naik, V. Mittal, and H. Cao. Line in the sand: A wireless sensor network for target detection, classification, and tracking. Technical Reprort OSU-CISRC-12/03-TR71, Ohio State University, 2003.

[2] Serge Belongie, Jitendra Malik, and Jan Puzicha. Matching shapes. In *International Conference of Computer Vision*, 2001.

[3] N. Berry, J. Davis, T. Ko, R. Kyker, R. Pate, R. Stinnet, J. Baker, A. Cusner, C. Van Dyke, B. Kyckelhahn, and D. Stark. Wireless sensor systems for sense/decide/act/communicate. Technical Report SAND2003-8803, Sandia National Laboratories, December 2003.

[4] Jean-Yves Bouguet and Pietro Perona. Visual navigation using a single camera. In *International Conference of Computer Vision*, 1995.

[5] Adam Chlipala, Jonathon Hui, and Gilamn Tolle. Deluge: Data dissemination in multi-hop sensor networks. Cs294-1 project report, U.C. Berkeley, December 2003.

[6] Trevor Darrell, David Demirdjian, Neal Checka, and Pedro Felzenswalb. Plan-view trajectory estimation with dense stereo background models. In *IEEE International Conference on Computer Vision*, 2001.

[7] Shiri Gordon, Hayit Greenspan, and Jacob Goldberger. Applying the information bottleneck principle to unsupervised clustering of discrete and continuous image representation. In *International Conference on Computer Vision*, October 2003.

[8] Michael Isard and Andrew Blake. Condensation – conditional density propoagation for visual tracking. *International Journal of Computer Vision*, 1998.

[9] Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixture of local experts. *Neural Computation*, pages 79–87, 1991.

[10] Tony Jebara. Images as bags of pixels. In *International Conference on Computer Vision*, October 2003.

[11] Sohaib Khan, Omar Javed, and Mubarak Shah. Tracking in uncalibrated cameras with overlapping field of view. In *Performance Evaluation of Tracking and Surveillance (PETS) with CVPR*, December 2001.

[12] Teresa Ko, Nina Berry, and Ronald Kyker. A minimalistic approach to image processing for low-power sensor networks. In *SPIE Defense and Security Symposium: Visual Information Processing*, April 2004.

[13] Teresa Ko, David Demirdjian, and Trevor Darrell. Untethered gesture acquisition and recognition for a multimodal conversational system. In *International Conference of Multimodal Interfaces*, November 2003.

[14] Ron Kyker. Hybrid emergency radiation detection: a wireless sensor network application for consequence management of a radiological release. In *SPIE's Defense and Security Symposium: Digital Wireless Communications VI*, April 2004.

[15] Lily Lee and William E.L. Grimson. Gait appearance for recognition and classification. In *Fifth IEEE International Conference on Automatatic Face and Gesture Recognition*, May 2002.

[16] Nancy A. Lynch. *Distributed Algorithms*. Morgan Kaufmann, 1996.

[17] William M. Merrill, Fredric Newberg, Kathy Sohrabi, William Kaiser, and Greg Potti. Collaborative networking requirements for unattended ground sensor systems. Technical Report 0-7803-7651-X, Sensoria Corporation, 2002.

[18] Joe Polastre, Robert Szewcyk, Alan Mainwaringa, David Culler, and John Anderson. *Analysis of Wireless Sensor Networks for Habitat Monitoring*. Kluwer Academic Pub, 2004.

[19] M. Pollefeys, R. Koch, M. Vergauwen, and L.Van Gool. Automated reconstruction of 3d scenes from sequences of images. In *ISPRS Journal of Photogrammetry and Remote Sensing*, 2000.

[20] Ali Rahimi and Brian Dunagan. Sparse sensor networks. In *MIT Project Oxygen: Student Oxygen Workshop*, 2003.

[21] Deva Ramanan and D. A. Forsyth. Using temporal coherence to build models of animals. In *International Conference of Computer Vision*, October 2003.

[22] Xiaofeng Ren and Jitendra Malik. Learning a classification model for segmentation. In *International Conference of Computer Vision*, October 2003.

[23] R. Riley, B. Schott, J. Czarnaski, and S. Thakkar. Power-aware acoustic processing. In *2nd International Workshop on Information Processing in Sensor Networks*, April 2003.

[24] Anthony Rowe, Charles Rosenberg, and Illah Nourbakhsh. A low cost embedded color vision system. In *Proceedings of IROS 2002*, 2002.

[25] Henry A. Rowley, Shumeet Baluja, and Takeo Kanade. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20, January 1998.

[26] Thanos Stathopoulos, John Heidemann, and Deborah Estrin. A remote code update mechanism for wireless sensor networks. Technical Report 30, CENS.

[27] J. Str(oe)m, T. Jebara, S. Basu, and A.P. Pentland. Real time tracking and modeling of faces: An ekf-based analysis by synthesis approach. In *Modeling People Workshop at the International Conference on Computer Vision*, 1999.

[28] Kah-Kay Sung and Tomaso Poggio. Example-based learning for view-based human face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20, January 1998.

[29] O. Trier, A. Jain, and T. Taxt. Feature extraction methods for character recognition - survey. *Pattern Recognition*, 29, 1996.

[30] Zhuowen Tu, Xiangrong Chen, Alan L. Yuille, and Song-Chun Zhu. Image parsing: Unifying segmentation, detection, and recognition. In *International Conference of Computer Vision*, October 2003.

[31] A. Verra and T. Poggio. Motion field and optical flow: qualitative properties. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(5):490–498, May 1989.

[32] Paul Viola and Michael Jones. Robust real-tiem object detection. In *Second International Workshop on Statistical and Computational Theories of Vision - Modeling, Learning, Computing, and Sampling*, July 2003.

[33] Paul Viola, Michael Jones, and Daneil Snow. Detecting pedestrains using patterns of motion and appearance. In *International Conference on Computer Vision*, October 2003.

[34] Roy Want, Trevor Pering, Gunner Danneels, Muthu Kumar, Murali Sundar, and John Light. The personal server: Changing the way we think about ubiquitous computing. In *Ubiquicomp 2002: 4th International Conference on Ubiquitous Computing*, pages 194–209, September,October 2002.

[35] T. Wilhelm, H.J. B(oe)hme, and H.M. Gro(sz). Looking closer. In *1st European Conference on Mobile Robots*, 2003.

[36] T. Wilhelm, H.J. B(oe)hme, and H.M. Gro(sz). Statistical and neural methods for vision-based analysis of facial expressions and gender. In *International Conference on Systems, Man and Cybernetics*, 2004.

[37] Danny B. Yang, Hector H. Gonzalez-Banos, and Leonidas J. Guibas. Counting people in crowds with a real-time network of simple image sensors. In *International Conference of Computer Vision*, October 2003.

# DISTRIBUTION:

1 MS 9158
Teresa Ko, 8961

1 MS 9158
Nina Berry, 8961

1 MS 9003
Ken Washingtn, 8900

1 MS 9003
Jim Handrock, 8960

1 MS 9158
Mike Koszykowski, 8961

1 MS 9158
Mitch Sukalski, 8961

1 MS 1002
Philip Heermann, 15230

1 MS 0865
Regan Stinnett, 01903

3 MS 9018
Central Technical Files, 8945-1

2 MS 0899
Technical Library, 9616

1 MS 0619
Review & Approval Desk, 9612

1 MS 0161
Patent and Licensing Office, 11500