Applications of Localized Image Processing Techniques in Wireless Sensor Networks

Divya Devaguptapu and Bhaskar Krishnamachari
Autonomous Networks Research Group
Department of Electrical Engineering - Systems
University of Southern California
Los Angeles, CA 90089, USA

ABSTRACT

We describe the application of image processing techniques for data refinement in sensor networks, by mapping network nodes to pixels in an image. Due to their localized, distributed nature, these techniques are inherently scalable and therefore desirable for use in large sensor networks. We examine two specific problems: cleaning of uncorrelated sensor noise, and the decentralized detection of edges (such as the perimeter of a chemical leak). Our simulation results show that the performance of these processing techniques depends critically upon both sensor density and radio range.

Keywords: Boundary, Edge Detection, Sensor Networks, Image Processing

1. INTRODUCTION

The emergence of wireless sensor networks is considered to be a major paradigm shift in technology. The possibility that large numbers of low-cost, power-efficient sensing devices can be deployed without fixed infrastructure for a range of monitoring and data-gathering applications has opened up a wide array of research areas[1].[2]

We focus on two data-processing problems that arise in sensor networks - noise cleaning and edge detection, and discuss the performance of localized image processing-based techniques for these problems as a function of network density and radio range. Sensor measurements are likely to be noisy, but when the environmental phenomena of interest are spatially correlated and sensor noise is uncorrelated, it is possible to combine information from nearby sensors in order to mitigate the noise. The edge detection problem is of considerable interest in scenarios involving diffuse phenomena such as chemical leaks whose perimeter needs to be tracked.

Both noise cleaning and edge detection are standard problems in classical image processing[3].[4] There is a natural analogy between data processing in sensor networks and image processing techniques. If we consider an instantaneous snapshot, the environmental phenomena of interest in an operational area can be represented with arbitrary fidelity by a sufficiently high-resolution image (that we refer to as the *environment image*). Due to their localized, distributed nature, classical image processing techniques are inherently scalable and therefore desirable for use in large sensor networks. In order to apply these techniques, one can envision individual sensors as providing values of specific pixels in the image. One main challenge in applying the image processing techniques then lies in the fact that nodes may not be regularly placed and not placed with sufficient density. In particular, these techniques perform well at high densities when each pixel contains a sensor node. A second challenge has to do with the notion of neighborhood. In classical image processing the neighbors of each pixel are the eight adjoining pixels; in the case of sensors, the notion of neighborhood is determined by the radio transmission power: each node can communicate locally with other nodes that lie within its effective radio range. The optimal choice of radio range for the best performance of an image processing-based technique can be highly dependent on the node density. We examine these issues through experimental simulations.

We note that other recent efforts have also examined the parallels between image processing and sensor networks, particularly for edge/boundary detection and tracking. Liu, Cheung, Guibas and Zhao[5] present

Further author information: (Send correspondence to Bhaskar Krishnamachari)
Divya Devaguptapu: devagupt@usc.edu
Bhaskar Krishnamachari: bkrishna@usc.edu

a centralized scheme that uses Hough transforms to track a moving boundary as a point in the dual space. Ganesan, Estrin, and Heidemann[6] have proposed a generalized hierarchical architecture for multi-resolution querying of regularly-placed sensor networks that is based on wavelet-transforms; this architecture is shown to be useful for queries involving boundaries and edges. Nowak and Mitra[7] present a hierarchical boundary estimation algorithm that is shown to be asymptotically optimal in trading off energy for mean square error. The non-hierarchical decentralized edge detection algorithm we investigate focuses on identifying nodes near the boundary. This algorithm is also discussed by Chintalapudi and Govindan.[8] Their results are corroborated by our complementary work which utilizes different performance metrics and provides additional insights into the relationship between radio range and sensor placement density.

The rest of the paper is organized as follows. We introduce classical image processing techniques for noise removal and edge detection in section 2. Section 3 describes our methodologies, simulations and results pertaining to the application of these techniques to the corresponding problems in sensor networks. We then provide concluding comments and a discussion of future work in section 4.

2. IMAGE PROCESSING TECHNIQUES

We first present a brief tutorial on classical image processing techniques pertinent to noise cleaning and edge detection.

2.1. Noise Cleaning

In images, noise usually appears as discrete isolated pixel variations that are spatially un-correlated. Pixels in error often appear visually markedly different from their neighbors. This visual perception is the basis for many noise reduction algorithms in image processing. Several linear and non-linear techniques have proven highly effective for noise cleaning.

Noise added to an image generally has a higher spatial frequency spectrum than the normal image components since it is spatially un-correlated. Hence, simple low pass filtering proves effective for noise cleaning. A spatially filtered output image $G(i,j)$ can be formed by the discrete convolution of an input image $F(i,j)$ with an MxM impulse response array $H(i,j)$, according to the relation $G(i,j) = \Sigma\Sigma F(m,n)H(m-i-C, n-j+C)$ where $C = (M+1)/2$. For noise cleaning H should be of low-pass form with all positive elements. There are several spatial domain linear noise cleaning filters, among which the Mean filter is popular. The impulse response array for the mean filter is given as

$$H = 1/9 \begin{vmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{vmatrix}$$

These arrays are called masks, and are normalized to unit weighting so that the noise-cleaning process does not introduce an amplitude bias in the processed image.

2.2. Edge Detection

Local discontinuities in image amplitude attributes can be defined as edges. An edge is characterized by the height, slope angle and the horizontal coordinate of the slope point. The two generic approaches to edge detection are differential detection and model fitting. In the differential detection approach, spatial processing is performed on an original image to produce a differential image with accentuated spatial amplitude changes. Then a differential detection operation is executed to determine the pixel locations of significant differentials. There are two major classes of differential detection: first-order derivative and second-order derivative. For the first order class, some form of spatial first order differentiation is performed and the resulting edge gradient is compared to a threshold value. An edge is judged present if the gradient exceeds the threshold value. Where as, in the second order derivative, an edge is deemed present if there is a significant spatial change in polarity of the second derivative.

In our work we will focus on the first order derivative edge detection technique for sensor nets. This technique involves generation of gradients in two orthogonal directions of an image. The edge gradient in the discrete domain is generated in terms of a row edge gradient $G_r(i,j)$ and column edge gradient $G_c(i,j)$, and the spatial amplitude gradient is given by $G(i,j) = \surd(G_r(i,j)^2 + G_c(i,j)^2)$ A good discrete approximation of the continuous differentials is to form the running difference of pixels along rows and columns of the image. The Prewitt filter is one commonly used approximation involving 3×3 pixel edge gradient operators given by the following masks:

Row Gradient

$$G_r = \begin{vmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{vmatrix}$$

Column Gradient

$$G_c = \begin{vmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{vmatrix}$$

Clearly the application of this technique to sensor networks is non-trivial and requires modification since the nodes are not placed regularly in pixel-like grids. We will discuss the pertinent modification in the next section.

3. IMAGE-BASED PROCESSING IN SENSOR NETWORKS

3.1. Assumptions and Simulation Framework

The following are some assumptions we make about the sensor network. We assume that every node knows its location in terms of an (x, y) coordinate in space. The neighbors of a particular node are determined based on its radio range R. All nodes that fall within the communication radius R of a particular node, are taken as its neighbors, all of whose locations are then communicated to this node. In our simulations we take this communication radius R in terms of pixels. So, if R is 25, then, every node located within 25 pixels of a particular node would be its neighbor. We also assume that there is an underlying protocol that takes care of all the necessary communication of information within the network.

We use gray-scale image files as an environment on which these image processing algorithms can be simulated. These images create an effective platform to carry out the simulations since sensor nodes can be randomly deployed as points on these images. Sensor measurements are taken as the pixel intensities on which the nodes lie. The operating range of measurement is taken as $0 - 255$ (0 is low intensity-black and 255 high intensity-white). For both noise cleaning and edge detection we test our algorithms using different sensor environments. These experimental simulations are done using MATLAB.

3.2. Noise Cleaning

3.2.1. Methodology

To create distinct simulation environments, we consider two images, one which is predominantly white, i.e having similar intensity pixel values, and another which has almost equal high and low intensity pixels with visually apparent differences in pixel intensities. Gaussian noise is added to both these images in order to simulate faulty sensor measurements. Nodes are then randomly deployed as points on these images.

In image processing, the mean filter is a 3×3 mask and the filtering is done at every pixel using the 8 neighbors that surround it. But sensor nodes are not placed in a regular grid fashion and hence do not have 8 distinct neighbors as do pixels. They could have more or less than 8 neighbors, since their neighbors are obtained based on communication range.

In order to appropriately apply the mean filter to sensor nets, we approximate the effect of this filter by computing the mean of all the neighbors that surround a particular node. Since we assume that every node not only knows the locations of its neighbors but also their measurements, the node computes the mean of all the readings so obtained and replaces its original reading with this computed mean.

3.2.2. Simulation Results

To analyze the results obtained using our approach, we first execute the conventional noise cleaning algorithm on a noisy image A, to obtain a filtered image A_1. We then compare A_1 with image A_2 which is obtained by using our noise filtering technique for sensor nets.

The metric we use to evaluate the performance of our algorithm is the *mean square error* (MSE), which we define as the mean of the square of the difference in pixel intensities obtained using the traditional image processing noise cleaning technique and our noise cleaning technique for sensor nets. We compute this MSE using images A_1 and A_2.

Figures 1 and 2 indicate the variation of MSE based on the communication range of the sensor network. From these figures it is clearly evident that error increases as the communication range of the nodes is increased. An explanation for this could be due to the fact that as the communication range is increased the number of neighbors with which a particular node node is averaging out its value also increases. Therefore as the communication range increases, the odds of a node averaging its value with uncorrelated sensor readings increases, thus increasing the mean square error. The performance of these algorithms also depends heavily on the environment in which these nodes are placed. In figure 1 the variation in the MSE is not as significant as it is in figure 2. The environment image used for figure 2 is more uncorrelated than figure 1, due to the presence of pixels with distinct pixel intensity variations. Therefore there is a more obvious increase in error with the increase in communication range for figure 2.

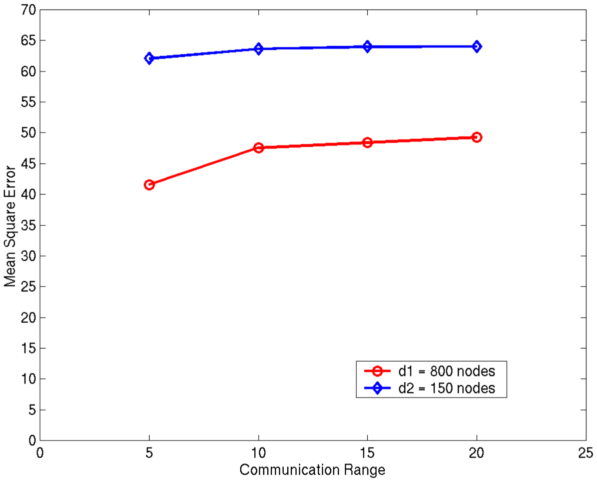

Figure 1. The increase in MSE with respect to increasing communication ranges for an environment image (image 1) whose pixels have similar intensity values.

Figures 3 and 4 indicate that density of deployment also has a substantial impact on the performance of this algorithm. As is apparent from these figures, for this algorithm to work favorably, not only must there be an optimal value of density and communication range, but there also must exist correlated sensor readings. Figure 4 shows that the increase in the density of deployment has an adverse effect on the MSE value. For small radio ranges, the performance of the mean filter improves with high density, but for large radio ranges, the performance drops drastically even with increase in density.

3.3. Edge Detection

3.3.1. Methodology

Similar to noise cleaning, the image environment is an image with the white segment of the image representing the phenomena. We again use two different images as shown in figure 5, one with a definite edge and the other

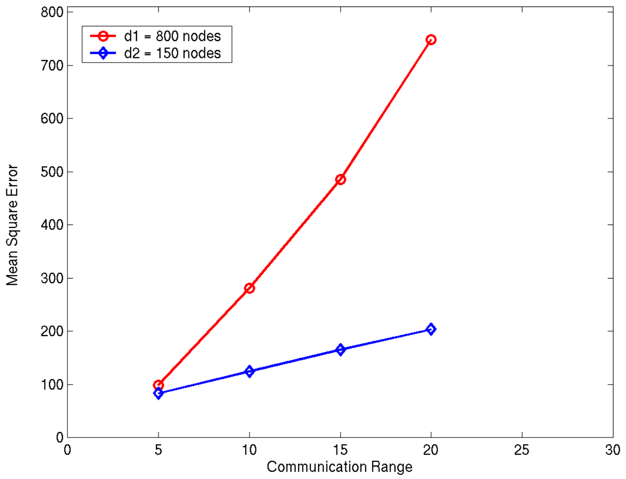

Figure 2. MSE increases significantly for increasing communication range, when an environment image (image 2) with distinct pixel intensity variations is used.

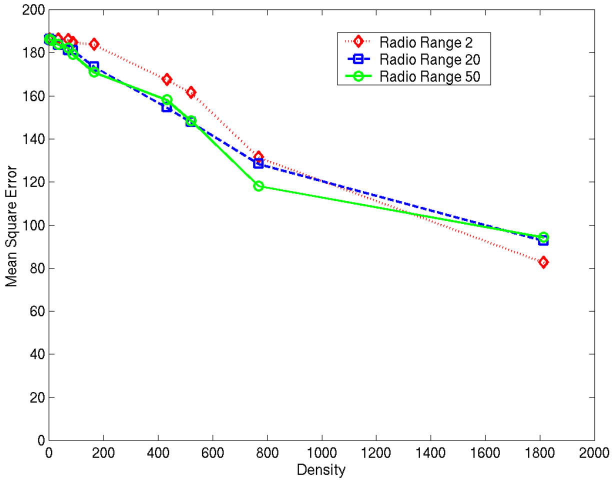

Figure 3. MSE is low for high densities and low communication ranges (image 1).

with a curved edge.

Every node determines locally whether it lies on an edge or not, by applying the Prewitt Filter. The fundamental difference between pixels in an image and sensor nodes is that the sensor nodes do not have spatial regularity of information like pixels in an image. Most image processing algorithms rely on the fact that information is regularly placed, but sensor nodes are usually deployed in a random fashion. Hence, the Prewitt mask cannot be directly applied. Due to random deployment there could be a case where nodes may not have neighbors that fall into a particular sector of the mask. Under these conditions, the values for nodes in those sectors are assigned the value of the node that is computing the edge, which is the central node. Conversely, there could be multiple nodes in a sector. In such a case, the sector value assigned is the mean of all node values lying in that sector. Every node determines the values of both x and y gradients by applying the mask. The magnitude of this gradient G is then computed. If the gradient is greater that a certain threshold T the node

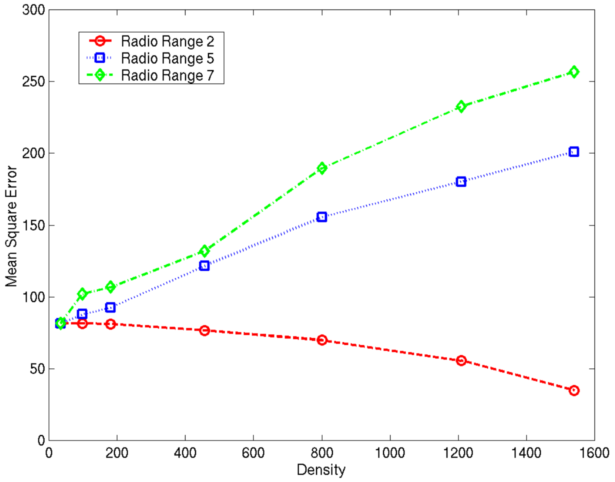

Figure 4. MSE decreases with density when communication range is low, and deteriorates with density when communication range is high (image 2).

deems itself present on the edge.

3.3.2. Simulation Results

To analyze the performance of this algorithm, we define a metric *mean offset* as the distance between the edge obtained using the sensor nodes and the edge as obtained using a traditional image processing technique on the image. We calculate this *mean offset* in two ways, one as the mean distance between every edge sensor from the closest edge pixel and the other as the mean distance between every edge pixel to its closest sensor.

Through simulations we learn that the algorithm works best only when the density of the nodes is extremely high. For smaller densities,the edge is not as definitive as it is for high densities. This again is due to the fact that the image processing algorithms work best only when there is dense spatial regularity of information.

As can be seen from figures 6 and 7, the *mean offset* of the edge sensor nodes from the closest edge pixel decreases as communication range increases. At the same time, the *mean offset* of the actual edge to the closest edge sensor node increases as the communication range is increased.

We also choose another metric *number of edge nodes/mean offset* (n/d) to determine the optimal performance of this algorithm. When plotted against communication range for various densities, n/d, shows an optimal communication range for a given density. As seen in figures 8 and 9, the peak of the curve determines the optimal communication range. Also, from figure 10 we can see that the value of this optimal communication range decreases as the density increases.

4. CONCLUSIONS

We have investigated the application of classical localized image-processing techniques for data processing in sensor networks. In particular, we focused on the noise cleaning and edge-detection problems. Our results indicate that for image processing techniques to work efficiently in sensor fields, the optimal choice of communication range depends critically upon the density of deployment and vice-versa. For the noise-cleaning problem, we showed that for small radio ranges, the performance of mean-filter improves with higher density, while for larger radio ranges, it deteriorates with higher density. For the edge-detection problem, we showed that for a fixed density, the performance initially improves with radio range, but after an optimal point, it deteriorates with radio range. The optimal radio range for edge-detection decreases with increasing sensor density.

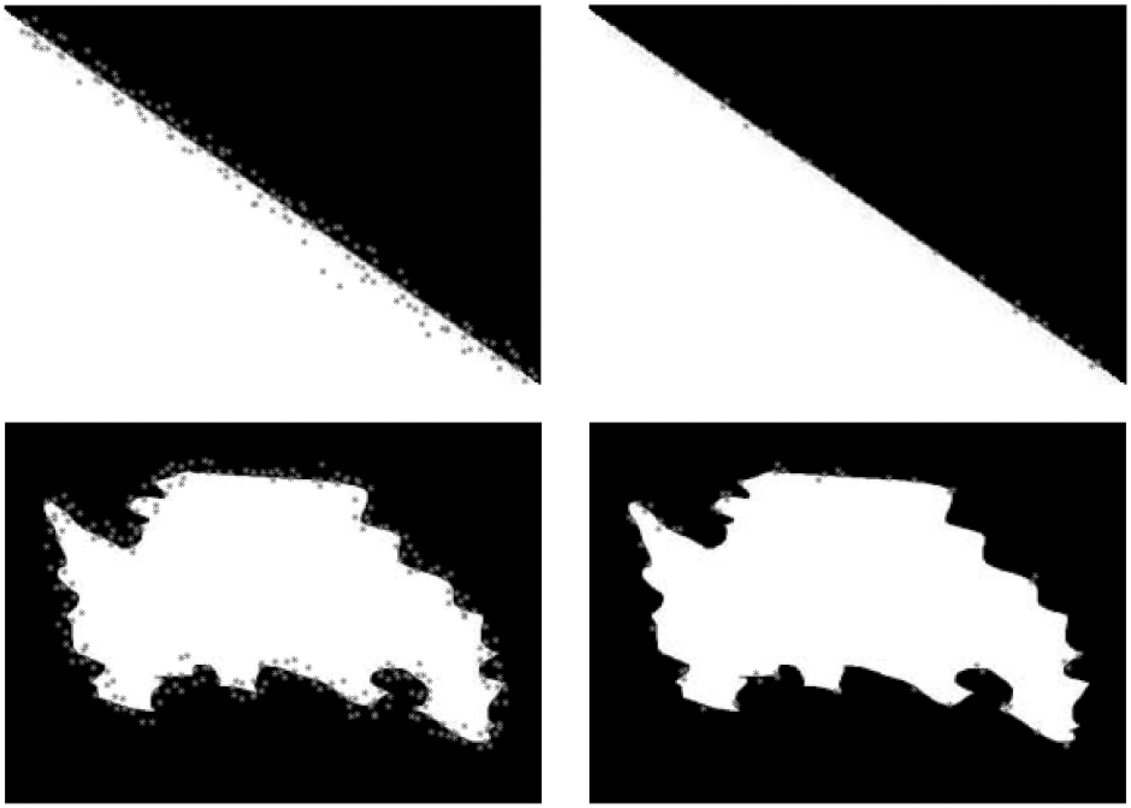

Figure 5. The scattered points on the environment images indicate the edge nodes obtained by the sensor network localized edge detection algorithm. For a higher communication range (top and bottom left) many nodes show up as edge sensor nodes but they may be farther from the actual edge on average; the reverse occurs at a lower communication range (top and bottom right).

In this paper we have described a boundary detection algorithm that determines the boundary of a static phenomenon at a given point in time. Future extensions of this work could borrow from video processing techniques to track the boundaries of dynamic phenomena such as an expanding chemical leak over time. Energy-quality tradeoffs would also be worth considering in this context. We would also like to enhance our simulation studies with pertinent mathematical analysis.

REFERENCES

1. D. Estrin *et al. Embedded, Everywhere: A Research Agenda for Networked Systems of Embedded Computers*, National Research Council Report, 2001.
2. I. Akyildiz, , W. Su, Y. Sankarasubramaniam, and E. Cayirci, “A Survey on Sensor Networks,” *IEEE Communications Magazine*, Vol. 40, No. 8, pp. 102-114, August 2002.
3. W.K. Pratt, *Digital Image Processing*, 2 Ed, John Wiley and Sons.
4. A.K. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall Inc.
5. J. Liu, P. Cheung, L. Guibas, and F. Zhao, “A Dual-Space Approach to Tracking and Sensor Management in Wireless Sensor Networks”, ACM International Workshop on Wireless Sensor Networks and Applications Workshop, Atlanta, September 2002
6. Deepak Ganesan, Deborah Estrin, and John Heidemann, “DIMENSIONS: Why do we need a new Data Handling architecture for Sensor Networks?” In *Proceedings of the ACM Workshop on Hot Topics in Networks*, October, 2002.

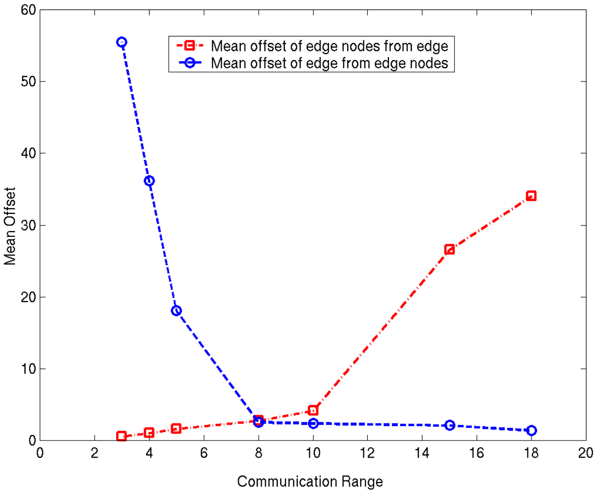

Figure 6. Variations in *mean offset* with communication range for a density of 1750 nodes on image 1.

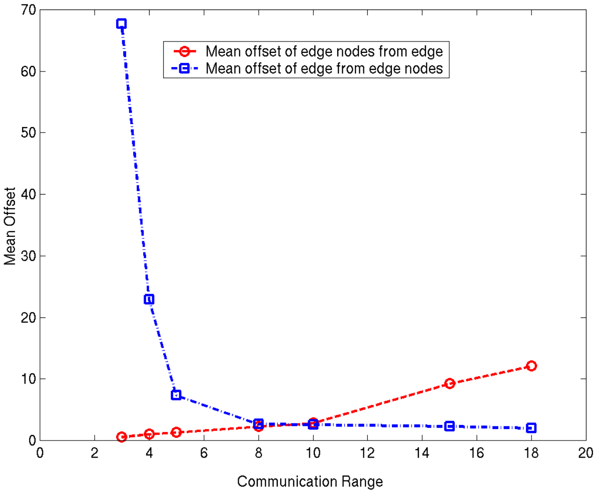

Figure 7. Variations in *mean offset* with communication range for a density of 2052 nodes on image 2.

7. R. Nowak and U. Mitra, “Boundary Estimation in Sensor Networks: Theory and Methods,” *2nd International Workshop on Information Processing in Sensor Networks*, Palo Alto, CA, April 22-23, 2003.
8. Krishna Kant Chintalapudi, Ramesh Govindan, “Localized Edge Detection in Sensor Fields,” *IEEE Workshop on Sensor Networks Protocols and Applications, SNPA '03*, 2003.
9. B. Krishnamachari and S.S. Iyengar, “Efficient and Fault-tolerant Feature Extraction in Sensor Networks,” *2nd International Workshop on Information Processing in Sensor Networks*, Palo Alto, CA, April 22-23, 2003.

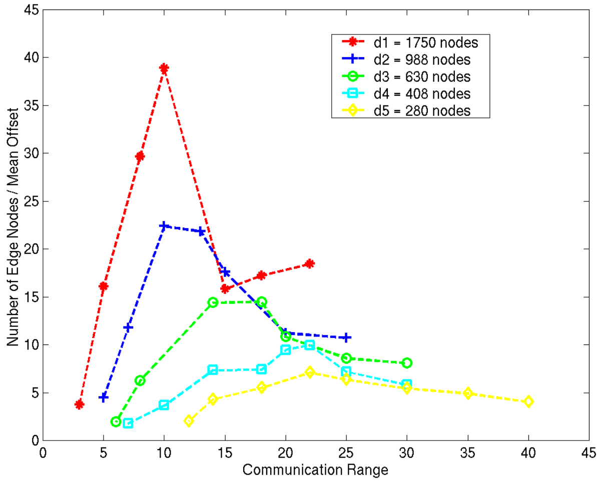

Figure 8. The peaks of the curves show the optimal communication ranges for a given density of nodes, for image 1.

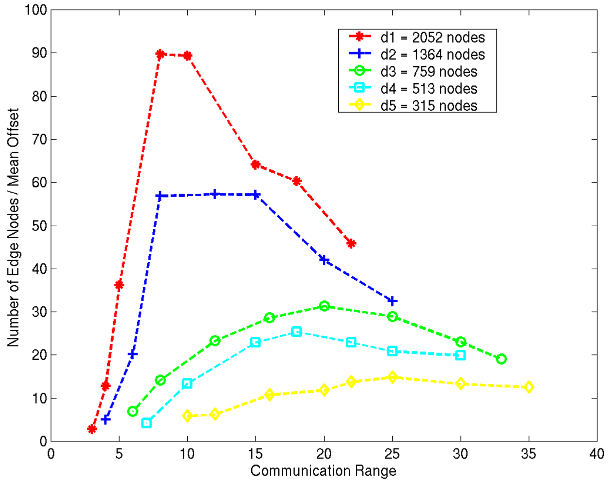

Figure 9. The peaks of the curves show the optimal communication ranges for a given density of nodes, for image 2.

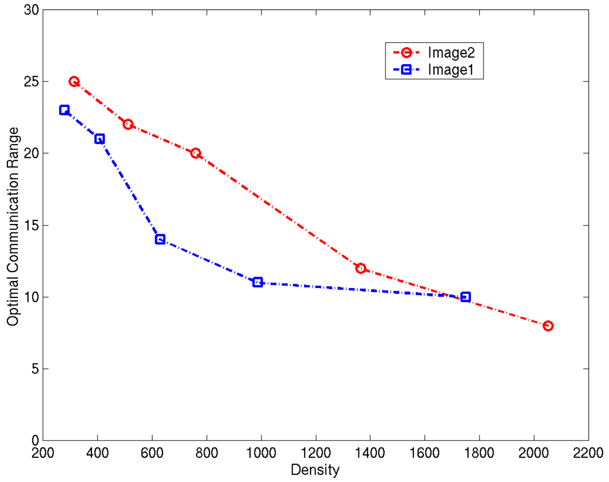

Figure 10. Optimal communication ranges for varying densities of nodes.