

Edge Detection in Sensor Networks using Image Processing

**Thair Abdulkareem Al-Aish
University of Baghdad / College of Education Ibn Al-Haitham
Baghdad - Adhamia**

Abstract

Edge detection is a widely studied problem in image processing and there are several well-developed edge detection algorithms in that domain. These algorithms are inherently scalable, localized and distributed in nature, with same computational approach applied at each pixel. Every pixel utilizes the values of pixels in its local neighborhood. Therefore application of these image processing algorithms in large sensor networks for solving the problem of edge detection is tempting. But applying those techniques directly in sensor network domain is not possible as unlike regular pattern of pixels in an image the sensors locations are random within the domain. This is due to the random deployment of sensor nodes over the geographical area.

In this work we propose an approach to modify these image processing techniques, taking into consideration the randomness of sensor nodes in the sensor field, to solve the boundary detection problem in sensor networks. A sensor network could be deployed in a region where there exist distinct sensing environments. To detect the demarcation of this boundary is the primary focus of our work. we propose an approach to modify the existing gradient based edge detection techniques in image processing domain to solve the boundary detection problem in sensor networks. Since the sensor readings are spatially correlated, in order to moderate noisy measurements, we combine information from neighboring sensors in our scheme. We give anon-weighted neighborhood approach and a weighted neighborhood approach to interpolate the missing sensor values. Taking the radio range (R) and the density of deployment of the sensor nodes(D) as the two parameters we have analyzed the performance of our approach using different simulation environments and different performance metrics. Our analysis shows that the performance highly depends upon the two parameters R and D . With high density of deployment performance achieved increases. Finally we have defined a metric to achieve optimization in the choice of radio range (R) for a particular density(D).

1. Introduction

A Sensor Network is defined as a wireless network of low cost, densely deployed sensor nodes (sensors) distributed in an ad hoc fashion where a collection of such sensor nodes coordinate to perform distributed micro sensing of environmental phenomena.

A sensor is a device that has three basic capabilities: data acquisition, data processing and wireless data transmission. Sensor nodes are very small and cheap, low powered devices. As shown in Fig.(1) these nodes have four main components: 1.a power unit 2.a sensing unit 3.a processing unit 4.a transceiver. Some other application specific components may also be added like a location finding system, mobilizer, power generator etc(1).

The sensor node may have one or more sensors (as part of the sensing unit) attached that are connected to the physical world. Thus each sensor node is a separate data source that generates records with several fields such as the id and/or location of the sensor that generated the reading, a time stamp, the sensor type(if there are different types of sensors deployed), and the value of the reading. Sensor data might contain noise, and it is often possible to obtain more accurate results by fusing data from several sensors. Summaries or aggregates of raw sensor data are thus more useful to sensor applications than individual sensor readings (2).

Numerous techniques for edge detection have been developed and analyzed in the image processing literature. In sensor networks it is preferable to design the applications requiring coordination among the individual sensor nodes using localized algorithms (3). In the context of sensor field, a distributed computation in which sensor nodes constrain their communication to sensors within some neighborhood and achieve a desired global objective is said to be localized. Such algorithms are preferable as here since each node communicates only with other nodes in some neighborhood, the communication overhead scales well with increase in network size. Secondly, for the similar reason these algorithms are robust to network partitions and node failures.

Thus to approach the problem of edge detection in sensor networks, we can think of a localized scheme. As the image processing techniques are localized and distributed in nature, an attempt to apply such techniques to localized edge detection in sensor networks is tempting. These algorithms are inherently scalable and therefore desirable for use in large sensor networks by mapping network nodes to pixels in an image.

2. System Description

The basic problem we are approaching in our work is illustrated in Fig.(2).Our objective is to consider measurements from a collection of sensors and determine the boundary between two fields of relatively homogeneous measurements.

One fundamental difference between images and a sensor field is the spatial regularity of information. The nodes may not be regularly placed and every sensor node need not be having all the neighboring 8 nodes. This is due to the obvious infeasibility of deploying and maintaining thousands of sensors in a grid like regular fashion over large geographical extents. In a digital image however, pixels are regularly placed in a grid with each pixel having the eight adjoining pixels as the neighbors (see Fig.(3)). The other difference is that the nodes may not be deployed with sufficient density and the also this density may not be uniform at different locations in the geographical extent(4). Our approach relies on the following key assumptions regarding the sensor field and sensor nodes:

1. Identical nodes

2. Localization: The position of each and every node is known in some arbitrary global coordinate system perhaps by using a localization system (5, 6).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.

3. Stationary Nodes.

4. Underlying Communication Protocol: We assume that there is an underlying protocol that takes care of all the necessary communication of information within the network.

2.1 Terminology

Let SA_0 be the set of all sensors within the communication range R_0 of a sensor S_0 in the sensor field. Let (x_{s_0}, y_{s_0}) be its location in some known coordinate space and $V(x_{s_0}, y_{s_0})$ denotes the value sensed by it. All the nodes that fall within the communication radius R_0 can be considered as the neighbors of the sensor node S_0 , all of whose locations and sensed values will then be communicated to this node. The gradient is a vector, whose components measure how rapidly pixel values are changing with distance in the x and y directions. Thus, the components of the gradient the image $I(x,y)$ may be found using the following approximation:

$$\frac{\delta I(x,y)}{\delta x} = \Delta_x = \frac{I(x+d_x, y) - f(x,y)}{d_x} \quad \dots\dots[1]$$

$$\frac{\delta I(x,y)}{\delta y} = \Delta_y = \frac{I(x+d_y, y) - f(x,y)}{d_y} \quad \dots\dots[2]$$

Where d_x and d_y measure distance along x and y directions respectively and $f(x,y)$ represent the frequencies in the image in the x and y axes. In (discrete) images we can consider d_x and d_y in terms of numbers of pixels between two points. Thus, when $d_x=d_y=1$ (pixel spacing) and we are at the point whose pixel coordinates are (i,j) we have

$$\Delta_x = I(i+1, j) - I(i, j) \quad \dots\dots\dots[3]$$

$$\Delta_y = I(i, j+1) - I(i, j) \quad \dots\dots\dots[4]$$

In order to detect the presence of a gradient discontinuity we must calculate the change in gradient at (i,j) . We can do this by finding the following gradient magnitude measure,

$$\Delta = \sqrt{\Delta_x^2 + \Delta_y^2} \dots\dots\dots [5]$$

2.2 Prewitt Edge Detector

The 2D continuous image $I(m,n)$ is divided into N rows and M columns. A high-pass filter retains only the higher frequencies (abrupt changes such as edges) present in the image and removes all the uniformities. We can design a filter with a desired frequency response. In general, if a filter with a frequency response $F(fx, fy)$ is desired then a filter $H(x,y)$ can be designed to approximately match the desired frequency response. Here, fx and fy represent the frequencies in the image in the x and y axes. To detect edges, the image $I(x,y)$ can be filtered by convolving with the filter $H(x,y)$.

$$I'[m,n] = I[m,n] \otimes H[m,n] = \sum_{j=0}^{M-1} \sum_{k=0}^{N-1} H[j,k] I[m-j,n-k] \dots\dots\dots [6]$$

Here values of the integer variables j and k vary as $j = 0, 1, 2, \dots, M - 1$ and $k = 0, 1, 2, \dots, N - 1$. Within the context of digital image processing, the x and y are discretized into pixels. If the filter and the image are represented by matrices $H(i,j)$ and $I(i,j)$ respectively, the filtered image I_f is computed as a convolution of I and H . One important edge operator of the gradient type, and which we are using in our work is the Prewitt Edge Detector. The prewitt filter in digital image processing is a set of two matrices ,

$$H_x = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} \dots\dots\dots [7]$$

$$H_y = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix} \dots\dots\dots [8]$$

Detecting an edge using this technique involves generation of gradients in two orthogonal directions of the image. The edge gradient in the discrete domain is generated in terms of row edge gradient Δx and column edge gradient Δy , and the spatial amplitude gradient is given by equation 5 as discussed above. A value of Δ greater than the threshold value would indicate an edge. As can be seen from Fig.(4), it may happen that the sensor node S_0 does not have sufficient number of sensor nodes in its (3×3) neighboring grid in order to apply the edge filter (H) equations and compute row edge gradient Δx and column edge gradient Δy by using equations 3 and 4. In the example sensor field figure S_0 is having just one neighboring sensor node value. Over the whole sensor field we define $V(x_s, y_s)$ as follows,

$$H_y(x_{s_0}, y_{s_0}) = \begin{cases} V_s & \text{if value sensed by the sensor node at location } (x_s, y_s); \\ 0 & \text{if sensor node is absent;} \end{cases} \dots\dots\dots [9]$$

2.3 Non-weighted linear average

For the grid locations, say X at which there is no sensor node present to sense the phenomenon and provide the sensed value, node S_0 will approximate its value by considering the neighboring rectangular grid of location X and taking the average of the sensor values provided by the "available" sensor nodes present in that grid (see Fig 5).The size of this rectangular grid can be 3 x 3, 5 x 5 etc. and will be decided by the node S_0 . This decision will be based on number of hardware characteristics of the sensor node S_0 like the value of its radio range as this will limit the number of nodes from which S_0 can hear and can later use for applying the average operation. For example as in Fig (5)the neighboring grid of location X , of size say 3 x 3 can be convoluted with the simple 3 x 3 linear spatial filter divided by the number of available sensor nodes in that grid. This determines a linear mean of available neighboring sensor values that will be the approximated value of pixel location X.

2.4 Weighted Average

Other than taking a simple average of the available sensor values in the neighboring grid of position X, we can instead think of associating weights to each of the neighboring sensor values. This is done with the idea to compensate for the uneven weighing caused due to arbitrary positioning and variations in density of sensor nodes in the neighborhood. Basically we aim to devise a scheme that outputs the weighted average of each position's neighborhood, with the average weighted more towards the value of the central pixels. This is in contrast to the linear filter's uniformly weighted average.

2.5 Derivation of Weights for neighboring sensor values

We assume an isotropic exponential attenuation of the weights to be associated to the sensor values with respect to the grid location X, at which average is to be done. Considering the following equation,

$$W(x, y) = \frac{1}{\alpha|r - r_x|^\beta} \dots\dots\dots[10]$$

Here r and r_x denote the position coordinates of the available sensor node in the neighboring grid and the position X respectively. α and β are the two constants that can be chosen. In our approach we take $\alpha= 1.0$.

Now, if we take $\beta=2$ we make the denominator in above eq.10

$$|(r - rx)|^2 = (x_s - x_{s0})^2 + (y_s - y_{s0})^2 \dots\dots\dots [11]$$

which is effectively making $W(x,y)$ decreasing on an exponential scale (inversely proportional to square of radial distance). To determine the value at pixel location X

$$V_X = \sum_{\forall s \in (SA_{S_0} \cap \text{available nodes in } X\text{'s neighborhood})} W(x_s, y_s) V_s \quad \dots\dots[12]$$

Here $W(x_s, y_s)$ are weights associated with each sensor value. In the figure consider the location P with co-ordinates (x_s, y_s-1) , where there is no sensor node to provide the sensed value. Here in this example we are taking the neighboring grid to be of size 5×5 . In order to determine the value at this location the node S_0 will approximate its value using the "available" values from the neighboring grid nodes.

$$V_P = \frac{1}{N_P} \sum_{i=-2}^2 \sum_{j=-2}^2 V(x_{s+i}, y_{(s-1)+j}) W(x_{s+i}, y_{(s-1)+j}) \quad \dots\dots(13)$$

Here N_P is the number of available sensor node values in the neighboring grid that will be used for the averaging operation. Similarly for location Q with coordinates (x_s+1, y_s+1) we can determine by using a similar equation. In order to save on the computation expenses, which is one of the primary resource constraints (7) for sensor nodes, we slightly modify our above discussed approach to assign weights and present a simpler scheme. This modification aims to reduce the number of complex mathematical computations required. To achieve this, instead of associating the decreasing weights to each of the individual pixel locations in the sensor field, we associate these weights to the grid as a whole i.e all the sensor nodes (the pixel locations) located exclusively in a particular grid are associated with the same weight value. With this exclusivity condition (see Fig 6), we mean that the sensor values within 3×3 grid are associated with a weight value. Then, all the sensor values present in 5×5 grid but not present in the 3×3 grid are associated with another weight value and similarly sensor values present in 7×7 grid but not present in the 5×5 grid (and 3×3 grid) are associated with yet another weight value and so forth.

For the neighborhood grid of the pixel location X we determine the average of the exclusively available pixels in each of the grid (i.e 3×3 , 5×5 and so on) multiplied by the weight associated to the exclusively available pixel locations in that grid separately and then finally take the summation of all to determine the weighted average. Considering Fig.(7), showing an example situation, in 3×3 neighboring grid average of available 3 sensor values is taken. In 5×5 grid the average of available 6 sensor nodes (excluding the inner 3×3 grid) is taken multiplied with weight given to this grid. In 7×7 grid average of available 6 sensor nodes (excluding inner 5×5 grid) is taken multiplying by the weight given to this grid and so on. Finally we determine the sum of all these values and determine the weighted average and the final result predicts the value at pixel location X .

2.6 Algorithms

To summarize the steps of our schemes as discussed above, we give here the following algorithms.

Non-Weighted Approach Algorithm

1. For each sensor node S (with radio range R) in the sensor field that is detecting a phenomenon,

- For each of the neighboring pixel location X with coordinates (x,y) for which $V(x,y)=0$ i.e at which no sensor node is available for sensing the phenomenon.

/* Interpolation of the missing sensor value */

- i. Consider the neighborhood grid of location X of size $m \times m$. The value of m can be 3, 5, 7, 9.....etc. depending upon the value R.
- ii. Determine the linear average of available sensor values in the rectangular grid ($m \times m$) to determine the approximated value at pixel location X.

2. Apply the prewitt filter and determine Δ_x and Δ_y using equations 3 and 4

3. Compute the value $\Delta = \sqrt{\Delta_x^2 + \Delta_y^2}$

If the value of Δ exceeds a threshold value say Δ_0 then this sensor node S will be detected as an edge pixel.

Weighted Approach Algorithm

1. For each sensor node S (with radio range R) in the sensor field that is detecting a phenomenon,

- For each of the neighboring pixel location X with coordinates (x,y) for which $V(x,y)=0$ i.e at which no sensor node is available for sensing the phenomenon.

/* Interpolation of the missing sensor value */

- i. Consider the neighborhood grid of location X of size $m \times m$. The value of m can be 3, 5, 7, 9.....etc. depending upon the value R.
- ii. Determine the average of exclusively available sensor values in each of the rectangular grid (i.e 3×3 , 5×5 , 7×7 ..upto $m \times m$) separately.
- iii. Take the weighted average of these calculated values to approximate the value at X.
 - A. Multiply these average values obtained in (b) above with the weight associated with the exclusively available values in the grid. The weights associated with the rectangular grids 3×3 , 5×5 , 7×7 ...up to $m \times m$ are $w_1, w_2, w_3, \dots, w_{m-3/2}$ (for e.g exponentially decreasing weights $1, 1/2, 1/4, 1/8, \dots, 1/2^{(m-3/2)}$ respectively.

B. Divide this sum by the total summation of weights. i.e. $\sum_{i=1}^{m-3} w_i$ As for the

weights in step above, this summation is $2(1 - (\frac{1}{2})^{\frac{m-1}{2}})$

2. Apply the prewitt filter and determine Δ_x and Δ_y using equations 3 and 4.

3. Compute the value $\Delta = \sqrt{\Delta_x^2 + \Delta_y^2}$

If the value of Δ exceeds a threshold value say Δ_0 then this sensor node S will be detected as an edge pixel.

3. Simulation

For carrying out the simulation experiments, in order to simulate the sensor field we have used grayscale images (with 256 gray levels) and have applied our processing algorithms over these images. These give an effective simulating environment as random deployment of sensor nodes in a sensor field can be simulated by randomly deploying these nodes as points (pixels) over the images. We are taking the sensor readings as the pixel intensities on which these nodes lie. Thus the sensor reading scan vary from 0 (lowest intensity) to 255 (highest intensity).

Fig.(8) show the example images, with sizes 176 X 229 and 170 X 170 respectively that we have taken for carrying out the simulations. In Fig.8 (a) the environment to be detected is represented by a curved edge whereas in Fig 8 (b) the edges between the distinct phenomena are relatively straight and definite. We have studied and analyzed the performance of our algorithm over these two sensor field environments.

3.1 Choice of Parameters

The density of deployment of sensor nodes in the geographical area pose a challenge for the approach of applying image processing algorithms over the sensor field. Also, the radio range of the sensor node, the hardware characteristic of the device will determine the number and extent neighboring nodes it can hear from. Hence, to carry out our experiments and study the performance of our approach we have taken the following two parameters:

- Density of Deployment (D)
- Communication Range of sensor nodes (R)

3.2 Metrics to study the performance

There can be a number of metrics to study and analyze the performance of edge detection algorithms in image processing domain (8). Here in the domain of a sensor network field we are using the following performance metrics to analyze the performance of our approach of edge detection.

- P : Percentage of accurately detected edge sensors(i.e that coincide with the ideal edge pixels) to the total ideal edge pixels on the test image. This metric provides the direct measure of accuracy of our approach and should be preferably high.
- NSR :Noise-to-Signal Ratio, defined as the ratio of the number of detected edge sensors, which do not coincide with the ideal edge (falsely detected edge sensor

nodes), to number of detected edge sensors which coincide with the ideal edge(accurately detected edge sensor nodes).

- **A** :Average distance of every detected edge sensor node from its closest ideal edge pixel. This metric in a way gives the measure of the thickness(width) of detected edge and also of the falsely detected edge sensor node sand should be preferably low.
- **M** :Mean width of the detected edge, defined as the ratio of total number of detected edge sensors (accurately detected + falsely detected)to the number of ideal edge pixels. For achieving quality in the result this metric should be preferably low.

3.3 Implementation details

For simulation we designed a simple simulator by coding our algorithms in Java using Java 2.0 API. MATLAB provides a function (m ,n, density) which generates a random, m-by-n, sparse matrix with approximately density * m * n uniformly distributed nonzero entries. Using the above MATLAB function we generated a test image by randomly deploying pixels with gray value 0 over the original example image. We assume that all the pixels having non-zero gray values are having sensors nodes for sensing the phenomenon and vice versa. To observe the performance of the algorithm in different simulating environments we generated our dataset by creating 10 sets of test grayscale images from the original example images with different values of densities of deployment - 20% , 30 % and so on up to 90 % by uniformly distributing such sensor nodes as pixels over the grayscale images.

3.4 Results and Discussion

For analyzing the performance, we execute non-weighted and weighted algorithms over the test grayscale images having different densities of deployment of sensor nodes(non-zero pixels). We average the performance metrics obtained after 10 different runs of the algorithm over the 10 different datasets for a given density of deployment and communication range.

Fig.(9) and Fig. (10) show the result of the application of non-weighted algorithm over image 1 and image 2 respectively. Table 1 shows the variation of mean width and Noise-to-Signal (NSR) ratio for deployment density of 10000 nodes over the test image 1. The table shows the results from both non-weighted and weighted schemes. In both the approaches the mean width metric increases with increasing communication range(R) which simply shows that sum of the falsely detected edge sensors and the accurately detected edge sensors increase with the parameter R and adversely affecting the quality of result. NSR (negative property) value decreases with R up to a value of R and then starts increasing again which simply shows that after attaining the maximum

percentage accuracy number of falsely detected edge sensors goes on increasing and the number of accurately detected edge sensors remain relatively constant.

In Table.1, comparing the values of both the metrics(at each value of communication radius R) as obtained from non-weighted algorithm and weighted algorithm, it can be inferred that the quality of results obtained with the weighted scheme are better (although it is computationally more expensive). In case of non-weighted approach, number of falsely detected edge sensors is more than that we get by the application of weighted scheme.

Table .2 shows the maximum percentage of accuracy(P) comparison for non-weighted and weighted approaches for Test Image 1. It can be seen that the maximum accuracy results are quite comparable in both the approaches. The other results we have discussed here are by using non-weighted approach.

The graphs Figures 11, 12 show the variation of performance metrics plotted against the Communication radius R for various densities of deployment(D) for image 1 using non-weighted approach. Similar results for image 2 are shown in Fig's 13, 14. From Fig 11 and 13 it can be observed that the performance of the approach is better when the density of deployed nodes is high. This is because, with high density the sensor field characteristics start matching with that of a normal image having spatial regularity of information. As communication range R is increased the percentage of accurate prediction goes on increasing before becoming relatively constant and attaining the maximum possible percentage.

3.5 Observed Tradeoff

By observing the graphs for both the images (Fig 11, 12, 13, and 14) it can be seen that at any node density, there exists a tradeoff between the two following metrics against Communication range R i.e With increasing value of parameter R, we observe that:

- P, the percentage of detected sensor edge nodes that coincide with that of actual edge pixels goes on increasing and thereafter become relatively constant. i.e positive property increasing with R before becoming constant.
- A, the average distance of detected edge sensor node from the closest ideal edge pixel also goes on increasing before it gets relatively constant. Or in other words the percentage of falsely detected edge sensors goes on increasing i.e a negative property also increasing with R before achieving relatively stable value.

3.6 Optimization Metrics

Considering the tradeoff as discussed above we infer that at some value of communication range R, the efficiency of the algorithm achieved will be maximum. To determine this optimum value of communication range R, for a particular node density, we define a new metric as the ratio of the above two properties. i.e

$$\frac{\text{Percentage of accurate prediction (P)}}{\text{Average distance of detected sensor nodes from the ideal edge(A)}}$$

This metric i.e P/A plotted against R for various densities, gives us a near to optimum value of communication radius R value for a given density. Fig 15 gives this plot for image 2. The peak value shows the optimum choice of R for a particular node deployment density.

In an alternate way, we also define another metric P/M

$$\frac{\text{Percentage of accurate prediction (P)}}{\text{Mean width of the detected edge (M)}}$$

This metric i.e P/M plotted against R for various densities, also gives a near to optimum value of communication radius R value for a given density. Fig. (16) gives this plot for image 1. The peak value shows the optimum choice of R for a particular node deployment density.

4. Conclusion

In this work we have proposed an image processing approach to perform edge detection in wireless sensor network. There were several challenges involved in adapting image processing techniques to sensor networks. Unlike pixels in an image that forma uniform pattern, sensor nodes are randomly deployed and so their neighborhood is based on their communication ranges.

Since the sensor readings are spatially correlated, in order to moderate noisy measurements, in our scheme, we combine information from neighboring sensors. We have given two approaches namely a simple non-weighted average approach and other, a little computationally expensive weighted Average approach. We observed that the weighted average approach gives nearly comparable results but with a better quality than the simple non-weighted average approach although the former one is computationally intense than the latter.

By simulating the non-weighted average and weighted average algorithm over grayscale images, our results indicated that the optimal choice of communication range depends critically upon the density of deployment and vice-versa. Our analysis showed that for a given density of deployment, the performance initially improves with communication range of the sensor node, but after an optimal point, increasing the communication range has no effect on the performance but the quality of the result deteriorates. With decreasing sensor deployment density, the optimal radio range for edge-detection increases.

In this work we have given an analysis of the performance of image processing approach for boundary detection in sensor nets and have used several performance metrics to do the same. However, to increase the efficiency, there are some improvements that can be made in the proposed algorithms by taking into

consideration the energy issues and communication overheads among the sensor nodes. To reduce the communication cost involved, the algorithms given here can be modified by reducing various redundancies involved in the computation of interpolated values.

5.References

1. Hall, D. L., and Linas, J., 2001, Handbook of Multisensor Data Fusion
2. Madden,S.R, Franklin,M.J. and Hong, W., 2002 , A Tiny Aggregation Service For ad-hoc Sensor Networks
3. Estrin,D., Govindan, R., Heidemann, J., and Kumar, S., 1999 , Next Centuray Chllenges: Scalable Coordination in Sensor Network , ACM/IEEE, USA.
4. Loren, S.k and Weinmann, J., 2001, Research challenges in Wireless Networks of Biomedical sensor, In Mobil Computing and Networking,p.151
5. Moses, D.k., and Patterson, R., 2002 , A Self –Localization Method For Wireless Sensor Network , Eurasip Journal on Applied Signal Processing, Special Issue on Sensor Network
6. Savvides A., C, Srivastava, M., 2001, Dynamic fine-grained localization in ad-hoc Networks of Sensors., Proceedings of ACM/IEEE International Conference on Mobile Computing and Networking.
7. Akyildiz, W., Su, Y.S., and Cayirci, E. , 2002 , A Survey on Sensor Networks, IEEE Communication Magazine , Vol .40, NO. 8, p 102-114
8. T., P., and D., M. , 2004, A Study Of Edge Detection Algorithm, In Computer Graphics Image Processing , Vol.20, p 1-21.

كشف الحافة في شبكات التحسس باستعمال المعالجة الصورية

ثائر عبد الكريم العايش
جامعة بغداد – كلية التربية ابن الهيثم

الخلاصة

الكشف عن حافة هـ ي مشكلة مدروسة على نطاق واسع في معالجة الصور وهناك عدة خوارزميات متطورة لاكتشاف الحافة في هذا المجال. هذه الخوارزميات هي بطبيعتها الموقعية والانتشارية تعتمد نفس النهج الحسابية المطبقة على كل عناصر (بكسل) الصورة. كل بكسل يستعمل قيم الهكسلات في المواقع المجاورة له . ومن ثم تطبيق خوارزميات معالجة الصور في شبكات الاستشعار الكبيرة لحل مشكلة كشف الحافة هو المغري. ولكن تطبيق هذه التقنيات بشكل

مباشر في مجال شبكات الاستشعار غير ممكن فهو يعاكس النمط النظامي للبيكسلات في الصورة حيث أن أجهزة الاستشعار لها مواقع عشوائية داخل المجال. ويرجع ذلك الى انتشار عشوائي ل عقد الاستشعار على مدى الرقعة الجغرافية.

في هذا العمل نقترح نهجا لتعديل تقنيات معالجة الصور ، مع الأخذ في الاعتبار العشوائية لعقد الاستشعار في مجال التحسس ، من أجل حل مشكلة الحدود في الكشف عن شبكات الاستشعار. شبكة الاستشعار يمكن نشرها في المنطقة على شكل وحدات استشعار منفصلة. يمثل الكشف عن ترسيم الحدود محور التركيز الرئيسي لعملنا. نفترض تعديل الانحدار القائم على أساس تقنيات استكشاف الحافة في مجال معالجة الصور لحل مشكلة الاستكشاف في شبكات الاستشعار. ونظراً لكون قراءات الأجهزة مرتبطة مكانيًا ، ومن أجل تعديل قياسات الضوضاء ، جمعنا معلومات من أجهزة الاستشعار المجاورة في البرنامج. حيث افترضنا وجود جوار متوازن وجوار غير متوازن لاقحام قيم أجهزة الاستشعار المفقودة. وبلخذ المجال الراديوي (R) وكثافته انتشار لعقد الاستشعار (D) ; كبرامترات لتحليل الأداء باستخدام محاكاة بيئية مختلفة وقياسات أداء مختلفة. التحليل توضح على أن الأداء العالي يعتمد على البارامترين (R) و (D). ومع الكثافة عالية الانتشار تزداد كفاءة الأداء واخيرا حددنا نظام متري للوصول أفضل اختيار للطفيف الراديوي (R) لكل كثافة معينة (D).

| Comm. Radius(R) | Mean-Width | | NSR | |
|---------------------|--------------|----------|--------------|----------|
| | Non-weighted | weighted | Non-weighted | weighted |
| 3 | 0.77 | 0.77 | 23.14 | 23.14 |
| 5 | 1.03 | 0.85 | 12.22 | 11.71 |
| 8 | 2.12 | 1.51 | 15.23 | 10.77 |
| 10 | 2.87 | 1.56 | 13.42 | 8.64 |
| 15 | 3.47 | 1.96 | 15.02 | 9.82 |
| 20 | 3.97 | 2.14 | 16.21 | 10.02 |
| 25 | 4.37 | 2.45 | 17.42 | 10.24 |

Table.1: Variation of Mean width(M) and NSR with R for Non-weighted and Weighted Approaches, Deployment density = 10000 nodes ($\cong 25\%$)for Test Image1

| Number of Nodes | Max. percent. of Accuracy | |
|-------------------------|---------------------------|----------|
| | Non-weighted | weighted |
| 36000($\approx 90\%$) | 62.52 | 57.43 |
| 24000($\approx 60\%$) | 45.93 | 41.21 |
| 16000($\approx 40\%$) | 26.94 | 22.35 |
| 8060($\approx 20\%$) | 9.27 | 6.56 |

Table.2: Comparison of Maximum Percentage of Accuracy(P), Non-weighted and Weighted Approaches, for Test Image 1

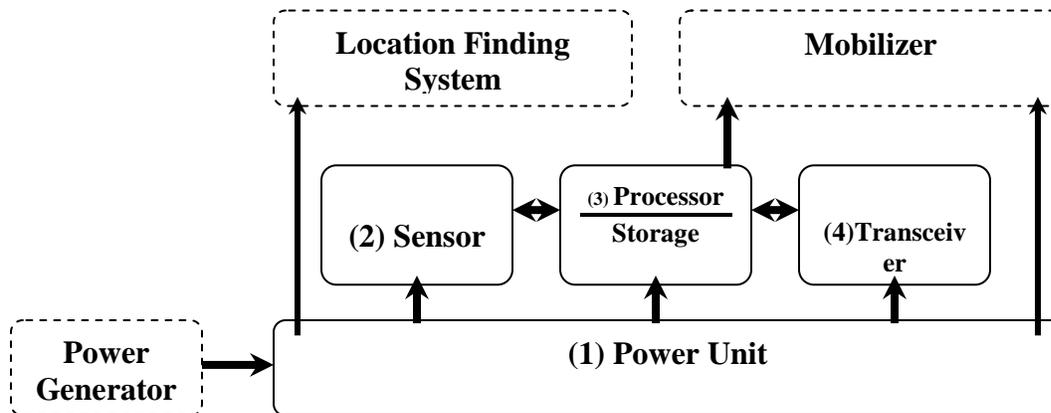


Fig.(1): Sensor Node – Architecture

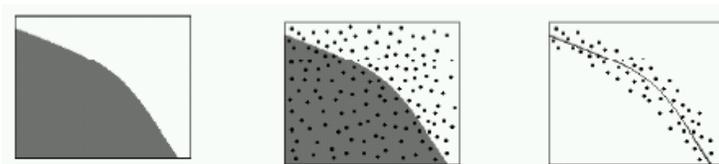
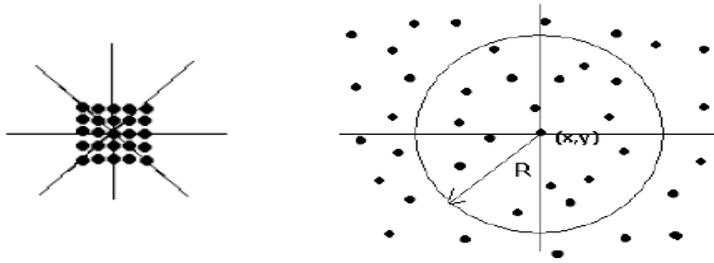
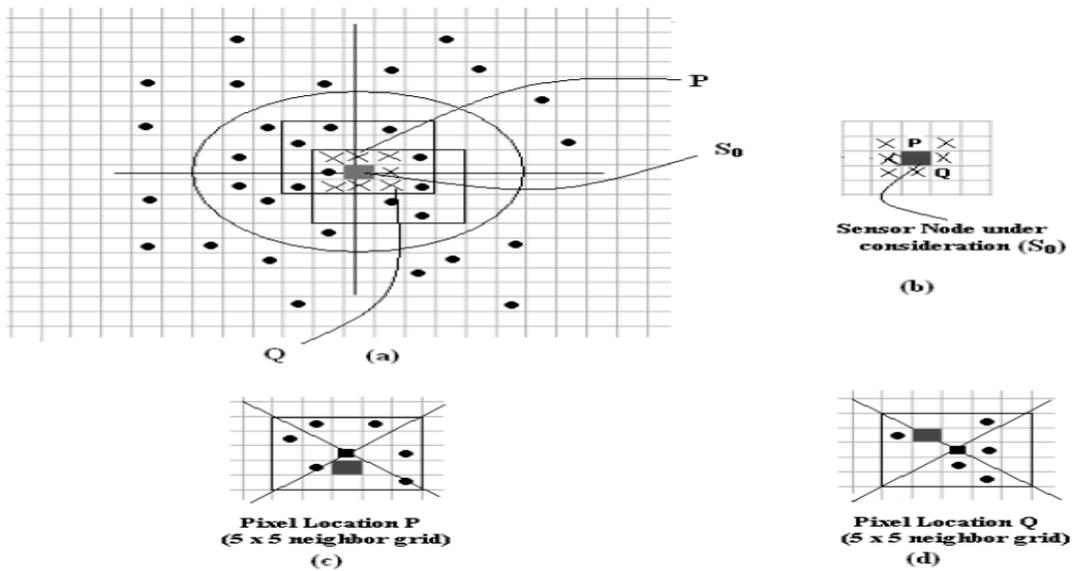


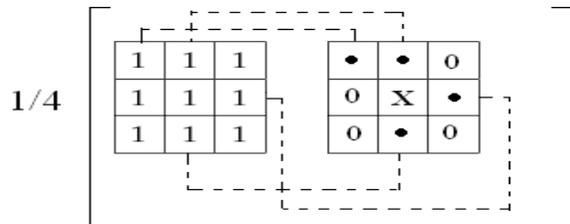
Fig .(2): Basic problem: Determining boundary between two distinct Environments



Fig(3): Comparison : Neighboring nodes in a Digital image and a Sensor field



Fig(4): (a)Sensor field : An Example, All nodes within the circular radius R_0 are in set SA_0 (b) 3×3 neighborhood grid of sensor node S_0 showing pixel positions P and Q (c) 5×5 neighborhood grid of pixel position P (d) 5×5 neighborhood grid of pixel position Q



Fig(5): Simple average of available sensor values in the example 3 x 3 neighboring grid. Here values from 4 available sensor nodes are added and this sum is divided by 4 to interpolate the value at central pixel position X

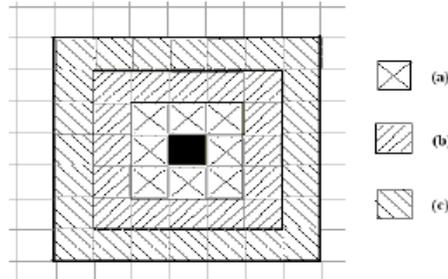


Fig.(6): Sensor nodes belonging exclusively to a particular grid. In the figure (a),(b) and (c) represent pixel locations exclusively belonging to 3 x 3, 5 x 5 and 7 x 7 grids respectively

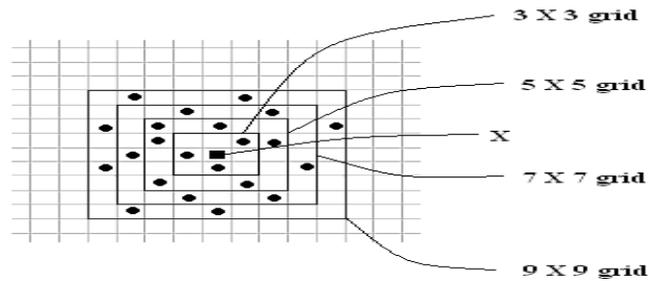


Fig.(7): Average of "available" sensor values in the neighboring grid. Here we take the average of "available" sensor values in each grid (i.e 3 x 3, 5 x 5 and so on) multiplied by the associated weight and then take the summation

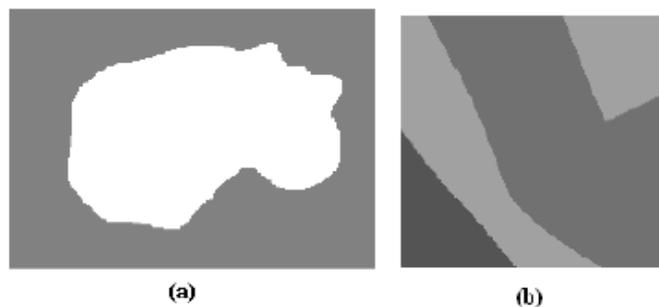


Fig.(8): (a)Test Image 1 - Size 176 x 229 (b)Test Image 2 - Size 170 x 170of

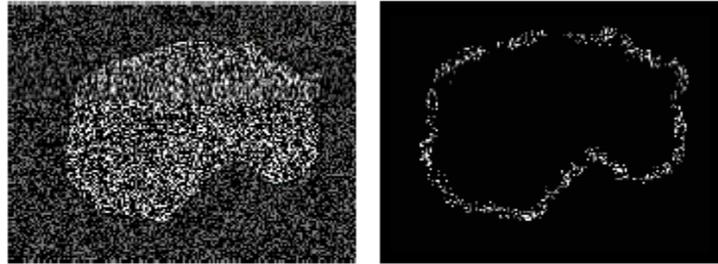


Fig 9: Detection Result on Image 1 : Density = 16000 nodes (\cong 40%), Comm. Range(R)=10

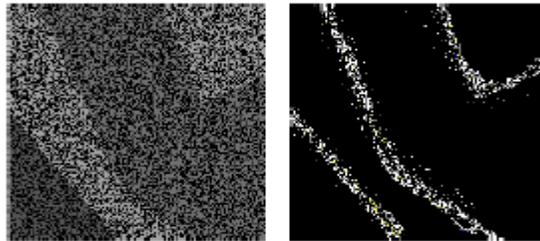


Fig.(10): Detection Result on Image 2 : Density = 17300 nodes (\cong 60%), Comm. Range(R)=12

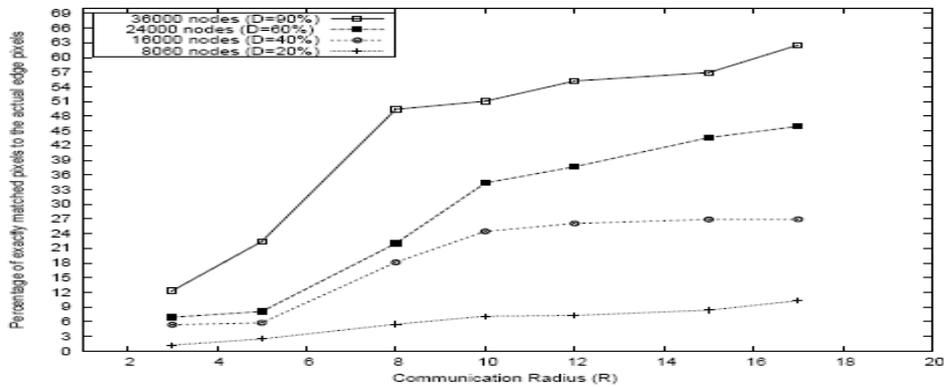


Fig.(11): Percentage of accuracy against Communication range R for varying deployment densities: Image 1

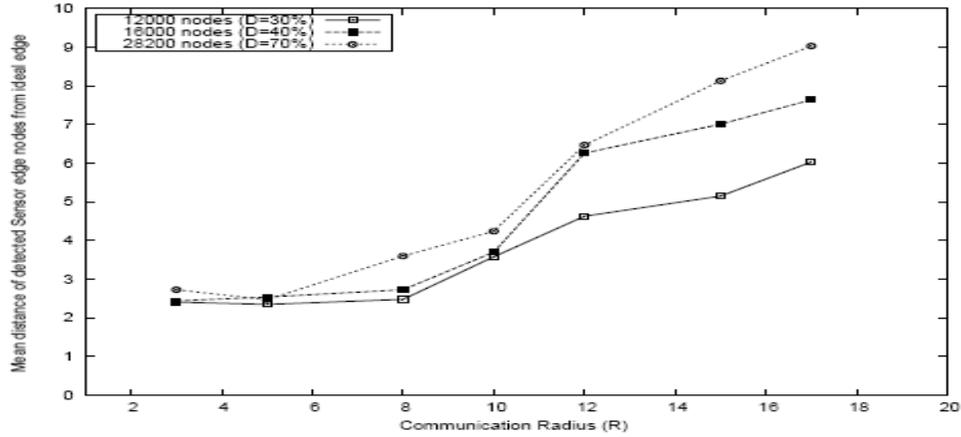


Fig.(12): Average distance of detected sensor edge nodes from the closest ideal edge point :Image 1

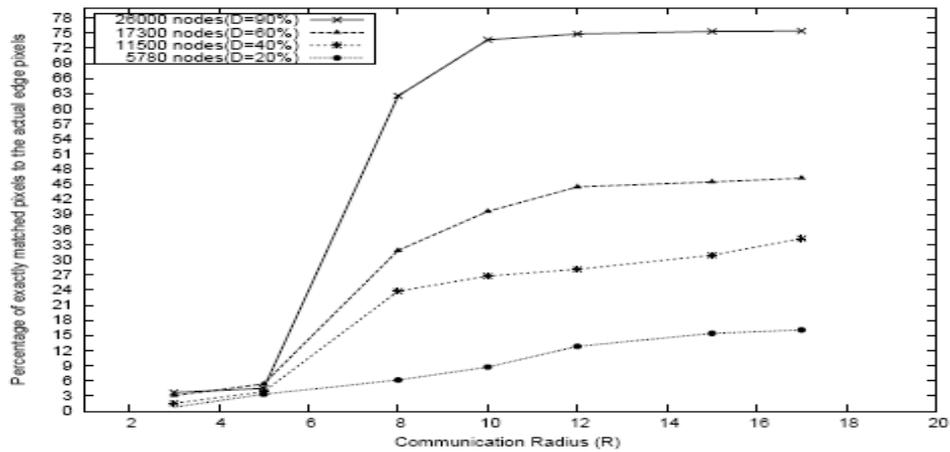


Fig.(13): Percentage of accuracy against Communication range R for varying deployment densities: Image 2

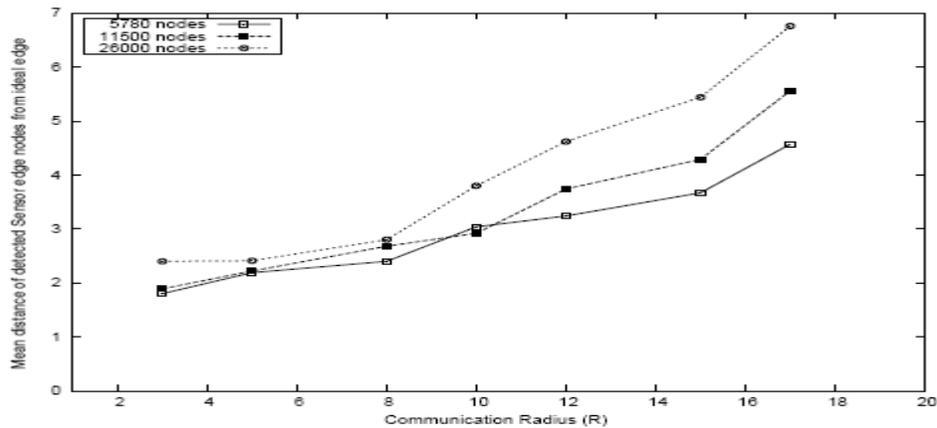


Fig.(14): Average distance of detected sensor edge nodes from the closest idea ledge point: Image 2 .

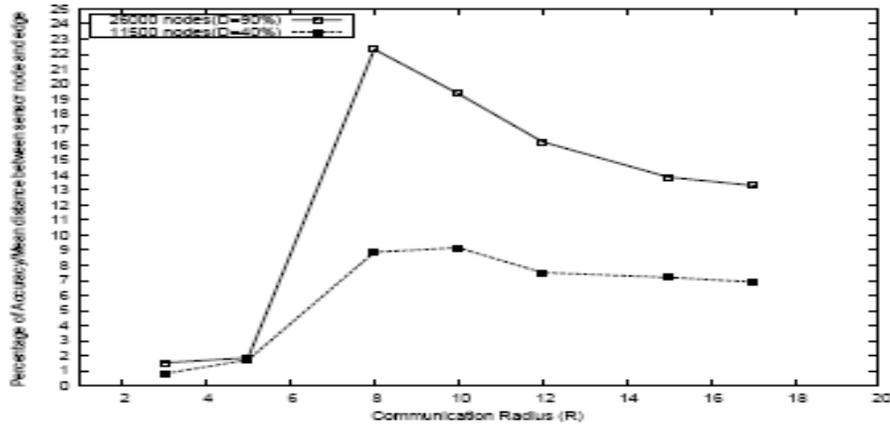


Fig.(15): Plot of P/A against R to determine optimum value of Communication range :Image 2.The peak values in the graph give optimum value of R

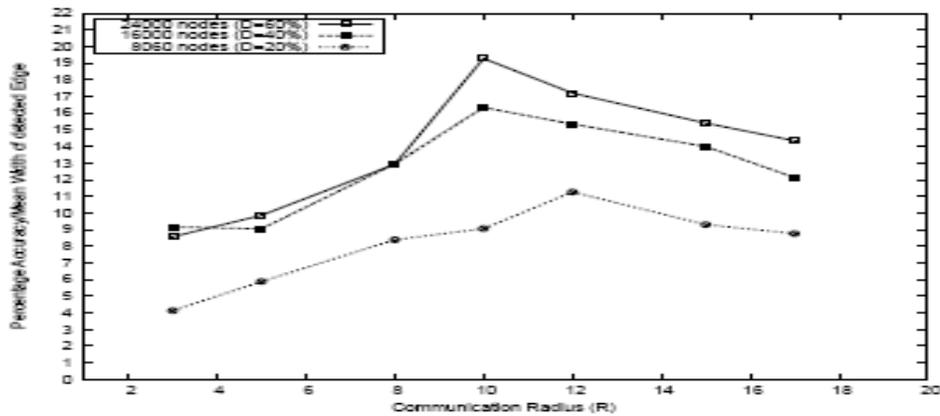


Figure 16: Plot of P/M against R to determine optimum value of Communication range: Image 1 The peak values in the graph give optimum value of R