31 Mar 2007

Wireless Sensor Network Wildfire Detection System

Justin Enderle

Recommended Citation

Enderle, Justin, "Wireless Sensor Network Wildfire Detection System" (2007). *Opportunities for Undergraduate Research Experience Program (OURE)*. 188.
https://scholarsmine.mst.edu/oure/188

Wireless Sensor Network Wildfire Detection System

Justin Enderle

03/31/2007

1 Abstract

One of the current difficulties in battling the destructive and costly wildfires is in obtaining up to date, accurate information of the fire's current location and intensity. Current methods relying primarily on satellite technology are too slow and inaccurate, therefore a better method is needed to help lower the destruction caused by wildfires and reduce the resources needed to battle them. This research proposes distributing a large number of cheap sensors across an area encompassed by wildfire capable of organizing themselves into an ad-hoc wireless sensor network to monitor the fire's current intensity and location. The majority of work performed pertaining to this research is in developing and analyzing the simulation tools needed to accurately test the wireless sensor network wildfire detection system. Commercially available software was used to generate realistic fires to test the system with, while custom software was developed to test how accurately randomly distributed sensors can predict a fire's outer perimeter intensity and location. The simulation results show this proposed method to be a promising solution to the current lack of information available for fighting wildfires. The predicted fireline is generated using a combination of algorithms to extract the most important information from the sensor nodes and generate a best guess for the fire's location. The best guess is compared against the ideal, previously known fireline, and found to be consistently accurate provided enough sensor nodes are distributed throughout the region. Further analysis still needs to be performed to determine the ideal number of sensor nodes required for any geographic area while maintaining accurate fire and and outer fireline location. Even with the needed future analysis, the current results provide a strong base by which an argument can be made for the effectiveness and feasibility of such a system.

2 Introduction

Wildfires are an extremeley destructive and unpredictable force of nature that our society battles every year. With nearly 39.3 million acres destroyed from 2000 to 2005 [1], it is a significant issue that impacts many lives. Methods are needed to aid in the suppression and extinguishing of wildfires. One of the difficult aspects of fighting a wildfire is in knowing is location in order to use firefighting resources most effectively. The dense smoke that often surrounds a wildfire can mask the actual location of the fire's outer perimeter. Satellite systems currently in place have the capability to analyze the location and basic characteristics of a fire on the group, but they are often limited by how often they provide updated imagery, with only a handful of updates daily, and their maximum available resolution, potentially as inaccuarate as 500 meters[3]. There is certainly room for improvement in the methods by which firefighters receive their information, and that improvement is an important step toward better controlling wildfires.

A proposed method for aiding the effort to provide faster, more reliable wildfire information is by using a wireless sensor network to monitor the environment in and around the fire's location, and accurately predict the intensity and location of the fire's perimeter. The wireless sensor network is composed of a large number of individual sensor nodes distributed over a large area encompassing the fire's location. Each node is relatively inexpensive and consists of an individual power source, wireless transceiver, and basic sensing electronics. The sensors wirelessly communicate with each other using ad-hoc networking to work together and transfer individual sensor measurements to a central location where the measurement values are analyzed and a coherent fireline location determined. Although it is conceivable that the sensor nodes could have the capability to detect multiple environment characteristics, such as infrared radiation, temperature, humidity, and smoke content, the extent of this work focuses only on the assumption that each node contains the ability to monitor the infrared radiation from all directions.

My undergraduate research primarily focused on assisting with understanding and developing the necessary tools to analyze the capabilities and performance of using a wireless sensor network for wildfire detection. These software tools are what eventually provide the results which leads to the decision of whether such a system has the potential to increase the effectiveness of current firefighting methods. The software tools, therefore, are an extremely integral and critical portion of understanding and evaluating this proposed system.

3 Body

The analysis of the proposed wildfire detection sensor network was done solely by simulation. The simulations consisted of the Fire Area Simulator, also known as FARSITE, used to produce realistic wildfire scenarios for analysis, as well as the Detection of Wildfire Simulator (DOWsim)[2], used to analyze the feasibility of using randomly distributed sensor nodes to estimate a wildfire's intensity and fireline location. While FARSITE is freely distributed software available online for download, DOWsim was primarily developed by Matt Gann and Michael Ellebrecht who are currently studying the proposed method for wildfire detection as their master's thesis topics. A detailed description of each simulation tool's purpose and usage is provided, as well as an analysis of the results and potential future developments for the system.

3.1 FARSITE Analysis

My work primarily began with the analysis and use of FARSITE. FARSITE is a fire modeling tool which uses landscape and environmental conditions to simulate the growth of a wildfire across a geographic region over a predetermined time frame. My primary responsibilities were to become acquainted with the software, and create a document outlining all of the setup and optional parameters required to generate consistent simulation results. Beyond that, I was also involved in expanding the total number of simulations available for analysis, using different landscape sizes to provide further variability in the fires tested by the fireline detection algorithm implemented within DOWsim.

After running through FARSITE tutorials which illustrated the entire simulation procedure from initial setup to generating results, I began to understand the structure of FARSITE and the setup conditions required to begin generating and running simulations. All of the landscape and environment conditions must be supplied to the simulator by way of setup files. Basic simulations require at minimum the following files: landscape characteristics, rate of spread adjustments, initial fuel moisture, as well as wind and weather conditions. The landscape file is actually generated from of a subgroup of files describing the landscape's elevation, slope, aspect, and canopy cover. For the purposes of our current research, it is not necessary to analyze and understand the purpose and influence of each setup file, but only that they are required and must adhere to certain structure requirements. All of the mentioned setup files are essentially just text files containing the required information entered in a specific format. All of the files can either be altered within FARSITE, altered using a text editor, or even altered and/or generated by a custom program specifically designed to edit and format FARSITE simulation files.

The landscape files are what generate the spatial characteristics of the environment. All of the information describing the geographic area for which the simulation is run is represented as a 2D grid in a text file. The number of grid entries and the area each grid location represents ultimately determines both the overall size of the landscape simulated as well as the maximum achievable environment resolution. For example, a 2D grid consisting of 100 x 100 entries and an entry size of 10m x 10m corresponds to a 1km x 1km landscape region.

A landscape's characteristics are changed by altering the value of the number entered for each grid entry in the individual landscape files. For this research, without having a solid foundation for how the majority of the characteristics impact simulation results, the simplest setup was chosen, consisting of uniform elevation, slope, aspect, and canopy cover values. The only landscape file significantly altered for the different simulation scenarios was the fuel model.

In the presence of different fuel types, the rate and direction of the fire line advancement is altered. Some fuels increase the speed and mobility of the fire, while others drastically impede its progress. Random mixtures of both types of fuel cause complicated and non-uniform firelines to occur which more closely resembles real world wildfire characteristics complicated by their reaction to the varying landscape. It was therefore found necessary to generate separate, random fuel models for each simulation to achieve unique and more complicated fireline results.

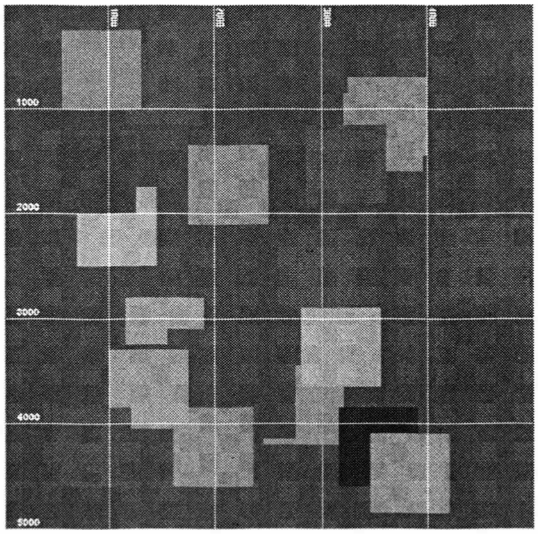

Figure 1: FARSITE random fuel model example

In order to expedite the process of creating a custom fuel model for the individual simulations, a Perl script was created capable of generating a random fuel model for any required grid size and resolution. Figure 1 shows an example fuel model generated by the script, with each color representing a different fuel type across the landscape. Each fuel model file is initialized as a single uniform value for every grid location, upon which random blocks of other fuel types are superimposed. Both the size and number of such blocks are configurable when executing the script. The random fuel distribution indeed serves its purpose well, shown by the simulation output shown in Figure 3, depicting its more complex shape. Without such random fuel type distributions, a fire line such as Figure 2 might be generated, which hardly test the limits of a fireline detection system.

The final important characteristic of the FARSITE simulator analyzed was the format and

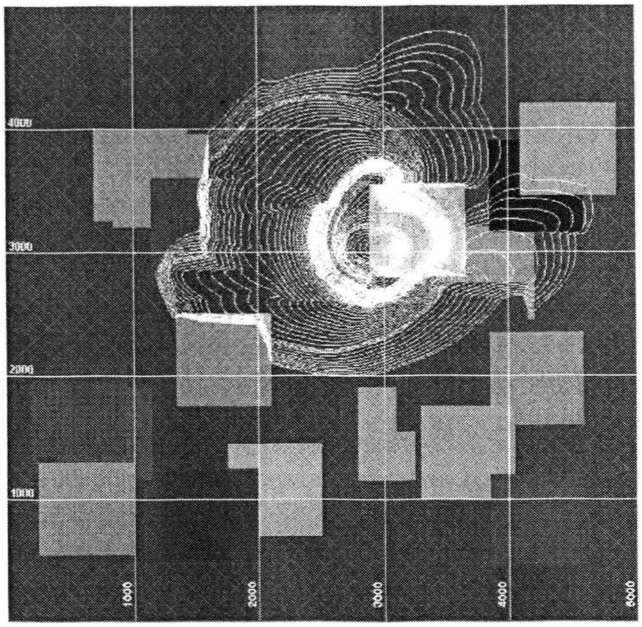

Figure 2: FARSITE simulation with random fuel model

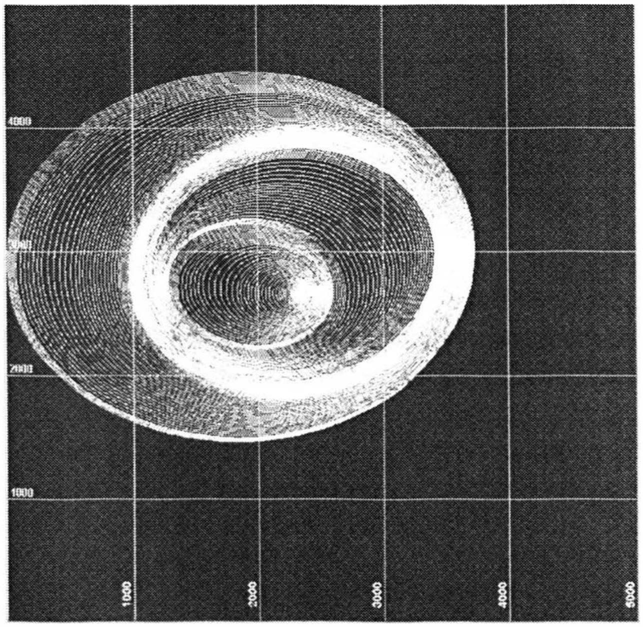

Figure 3: FARSITE simulation with uniform fuel model

information content of the output file. Looking at Figure 2, multiple rings are visible. Each ring corresponds to the fire's location at a particular point in time (time step). An output file is generated throughout simulation for each time step which records the x-y coordinates of the fireline within a specified resolution. In its current state, the fire line detection algorithm depends solely on the current fireline information, making no references the fireline's previous locations or characteristics. Because of this, DOWsim only requires simulation results for a single time step, which is selectively extracted from the complete output file which contains information pertaining to the fire across all time steps. FARSITE allows the user to choose what information is saved to the output file for each x-y coordinate of the fireline. The information currently stored for use by DOWsim is the x-y coordinates of the fire line, the current time step being analyzed, the fireline intensity, flame length, rate of spread, and spread direction.

Originally only five sample fires were generated to test the capabilities of the fireline detection system, all having a 1km x 1km grid size with a resolution of 10m. To further analyze the capabilities and requirements of the proposed system, 20 more fires were generated for further testing and analysis. Five unique fire simulations each for 5km x 5km, 10km x 10km, 15km x 15km, and 20km x 20km landscape sizes were created. The increase in grid size was also met with a corresponding increase in resolution as well in order to maintain a consistent 100 x 100 entry grid size, keeping the simulation computation time a constant. With a larger pool of fires generated, it next became time to work directly with DOWsim to generate and analyze detection results using the newly created wildfire simulations.

3.2 DOWsim Analysis

The Detection of Wildfire Simulator (DOWsim) is the core software tool used to analyze the wildfire detection system. The system is implemented with MATLAB, taking advantage of the large number of predefined functions readily available. DOWsim picks up where FARSITE left off, making use of the output file containing the fireline results. I began by first reading through and understanding the functionality and methods used by the simulation. Next I expanded the flexibility of DOWsim by adding the capacity to handle landscape files larger than 1km x 1km. Finally, an enhancement of the existing algorithm was implemented, increasing the likelihood of the detection system in predicting the correct fireline location. A detailed analysis of the structure of DOWsim, along with its algorithm for detecting the outer perimeter and intensity of a wildfire is provided below.

The first step required to use configure the DOWsim software is to define the grid size, resolution, and number of nodes being distributed across the landscape area. After defining those characteristics, a random distribution of the specified number of sensor nodes is generated across the landscape, with all of the locations stored in a data file for storage and later availability. In conjunction with the randomly distributed sensors, an ideal distribution of sensors is also generated and stored, placing a sensor node at every location of the grid. This ideal sensor node distribution is later used to calculate the ideal intensity map by which the random distribution can be compared against.

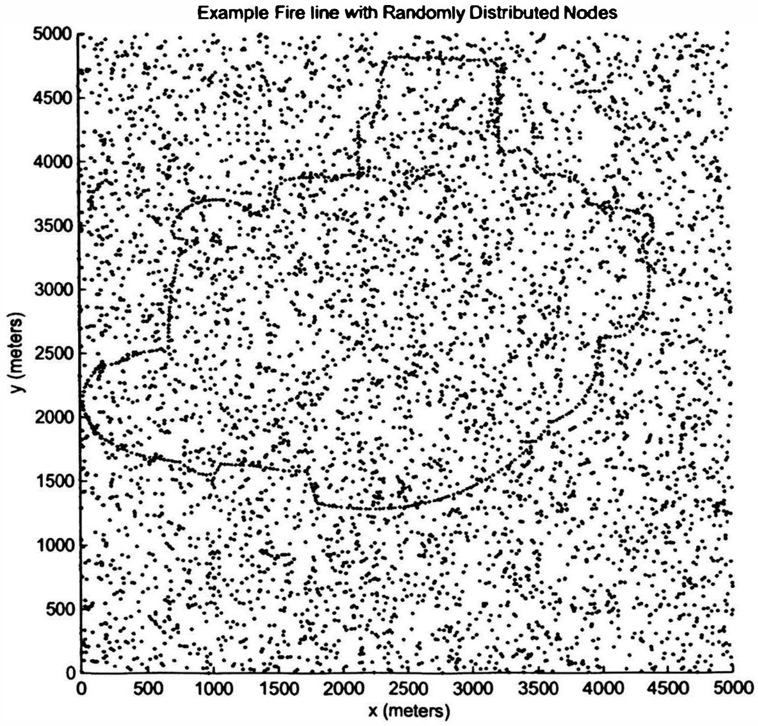

Figure 4: Fireline and random distribution of 5000 sensors

A plot of an original, ideal fireline showing the random node distribution is shown in Figure 4. The ideal fireline information is read in from the FARSITE simulator output file. For each point stored and plotted for the ideal fireline, all of the information gathered and stored in the FARSITE simulator output is available, previously defined as the x-y coordinates of the fire line, the current time step being analyzed, fireline intensity, flame length, rate of spread, and spread direction. The information stored for each fireline point is used to calculate the infrared radiation detected by the individual sensor nodes for both the random and ideal distributions. Because the infrared radiation detected by each sensor is assumed to be isotropic, the infrared radiation contribution for each fireline point stored by FARSITE must be calculated and summed to arrive at the final energy amount detected at a sensor location. In a real world environment, the amount of energy present at to each node is simply analyzed by the node's sensor, but because this is a simulation, the level of energy reaching each node must be calculated to achieve the most accurate results. This process of calculating detected infrared radiation energy is performed for all node locations for the random and ideal sensor distributions. The radiation intensity level for all grid locations has been calculated at this point, and the ideal intensity map is shown in Figure 5. The white regions correspond to high intensity levels, with dark red and black corresponding to low intensity levels.

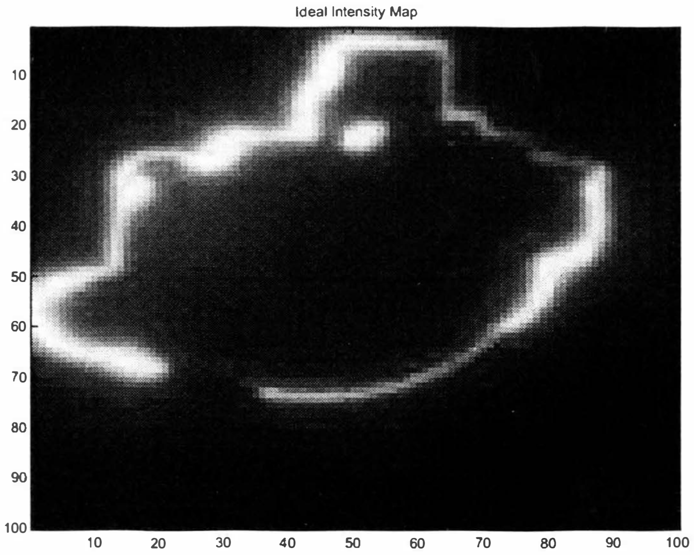

Figure 5: Ideal intensity map

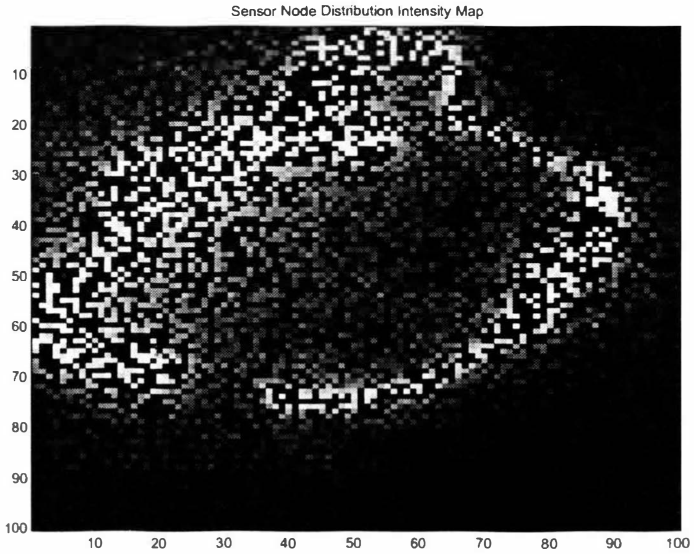

Figure 6: Intensity levels for grid locations with sensor information

After the radiated energy present at the random sensor node locations is calculated, a MATLAB function is used to perform 2D linear interpolation to estimate the intensity level across all points of the grid. Figure 6 shows the intensity values for the random node locations, while Figure 7 shows the intensity map resulting from a linear interpolation of the available sensor intensities. Upon visual inspection, the ideal and interpolated intensity maps look

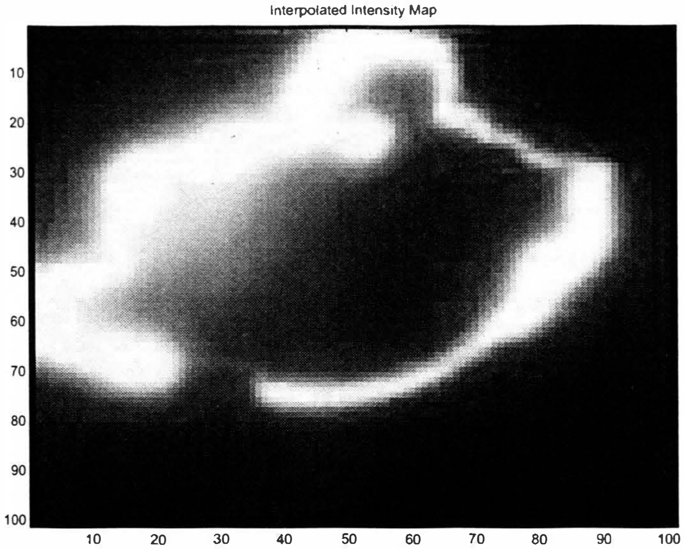

Figure 7: Interpolated intensity map

very much alike. A metric is later described and analyzed to determine how close the interpolated values are to the ideal values. As this method uses interpolation, the accuracy of such a method is heavily dependent on the amount of information present to interpolate from, where the available information content is directly related to the number of sensors distributed throughout the area. It would therefore be sensible to assume that increasing the number of sensors increases the overall accuracy. This assumption is later evaluated using a test statitistic that tests the ideal and interpolated intensity map correlation.

Now that an intensity value has either been directly sensed or estimated for each x-y coordinate of the grid, it is possible to visualize the grid as a 3D surface, with the third axis, or height, corresponding to the intensity value present at each x-y location. Figure 8 shows what the 3D surface plot looks like for the fireline and sensor node distribution shown in Figure 4. The plot resembles a mountainous area, with the highest points of the region corresponding to the highest intensity points. By following and plotting the location of the ridge line for the 3D surface, a good approximation to the location of the fireline can be made. A watershed function exists in MATLAB, which analyzes the 3D surface and identifies the different regions across the x-y 2D plane where water would naturally accumulate, following the assumption that water flows downhill and pools at a local minima. The grid locations that separate the local minima where water would accumulate corresponds directly to the peaks of the 3D surface, and therefore directly to the high intensity values located throughout the wildfire region. The grid locations which correspond to these ridges can be saved separated and saved to a separate matrix for further analysis.

Figure 9 shows the result of using the watershed function and ridge locations to estimate the location of the fireline. As the figure illustrates, no single enclosed path exists that obviously

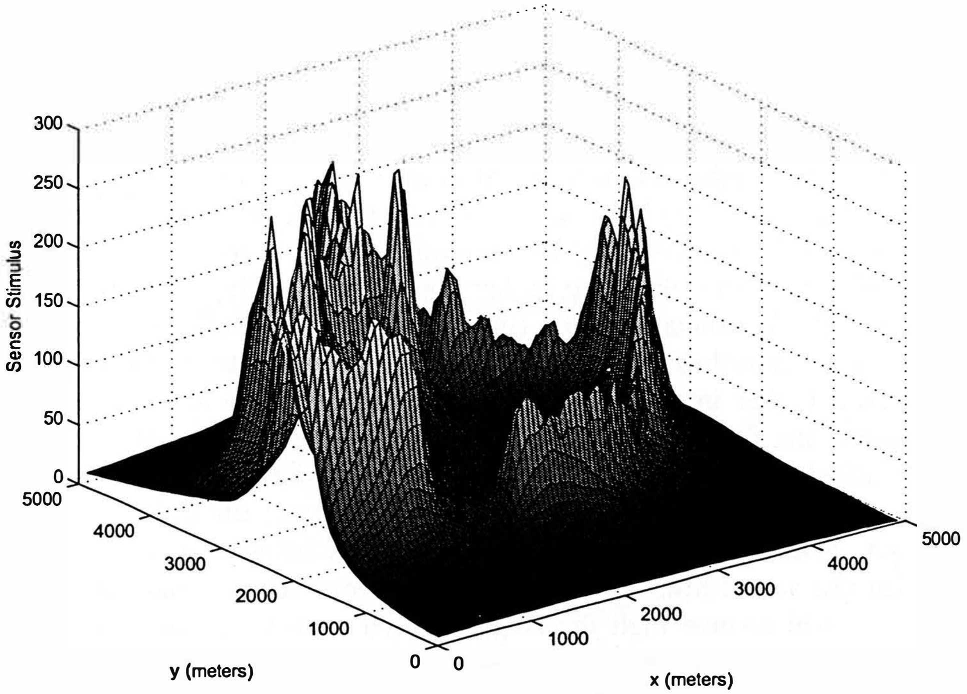

Figure 8: Intensity map surface plot.

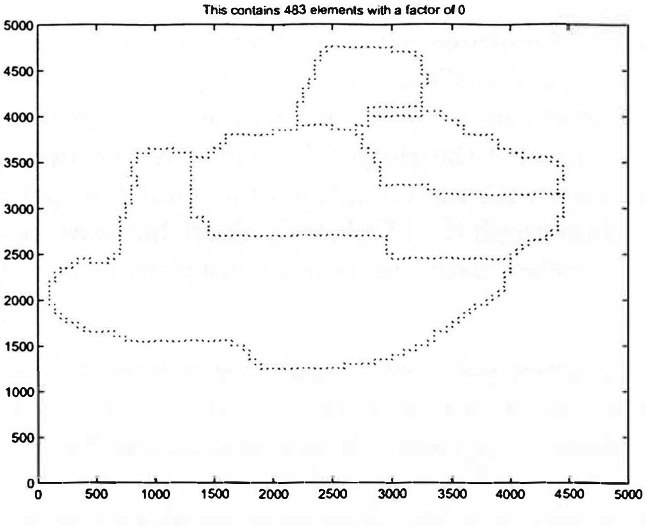

Figure 9: Original fireline estimation from surface plot

represents the genuine fireline. There are sections where multiple fireline paths are available, but only one is correct, and additional methods must be used to narrow down and determine which path is the correct one.

Originally, the fireline estimation algorithm relied solely on the implementation of a path finding method named Dijkstra's algorithm to determine which path was the correct one. The concept of Dijkstra's algorithm is to assign some cost value associated with each possible path in a graph, and ultimately choose the path with the lowest accumulated cost associated with it. The cost parameter is often used in the context of distances between locations having multiple paths between them, with the higher cost corresponding to a longer distance. For this system, the cost is associated with the sum of the intensities of each point along a particular path. It is hypothesized, that it will be most accurate to choose the path which leads to the overall higher intensity sum. This is based on the assumption that the path actually containing the fireline will naturally contain higher intensity values and will be chosen. Unfortunately, such is not always the case, and occasionally an incorrect path can be chosen. Depicted in Figure 10 is one example of an incorrect fireline estimation as the final algorithm output. What is essentially occurring is that although the incorrectly chosen path does not contain the actual fire, the incorrect path is either close enough to high intensity firelines that they still receive high levels of radiated energy, or there are more points in the incorrect path than the correct one, allowing more points to contribute to the intensity sum metric. This can be a difficult task to solve without somehow changing the metric by which a particular path is chosen. The difficulty is that, without assuming additional sensor capabilities per node, the intensity value per grid position is the only information available by which to make a decision. One way to improve the accuracy is to simply eliminate as many false paths as possible before running Dijkstra's algorithm. This method was implemented and is further described.

What causes the presence of multiple paths in the fireline estimation graph is the presence of additional local minima on the 3D surface plot. All that is required to eliminate such a region is to somehow connect one local minima region to the next, essentially creating one large region in its place. Even if the ridge that separates the two local minima regions is almost completely connected with the exception of a single grid point, it still will not show up on the fireline estimation graph that Dijkstra's algorithm uses, as even one missing point in a path is considered a broken path and is not considered as a viable option connect one graph point to another.

As it is likely that an incorrect path will contain lower intensity values as opposed to the correct path, an option to eliminate some of the incorrect paths is to set a threshold value by which all points in the intensity map less than the threshold will be set to the minimum value present across the entire graph. This occurs before the watershed function is run to define the local minima and ridge points. If a low intensity point along a false estimation path where a ridge previously existed is set to the graph's minimum value, that point can no longer possibly represent any kind of local maxima and thus will eliminate a connected ridge from

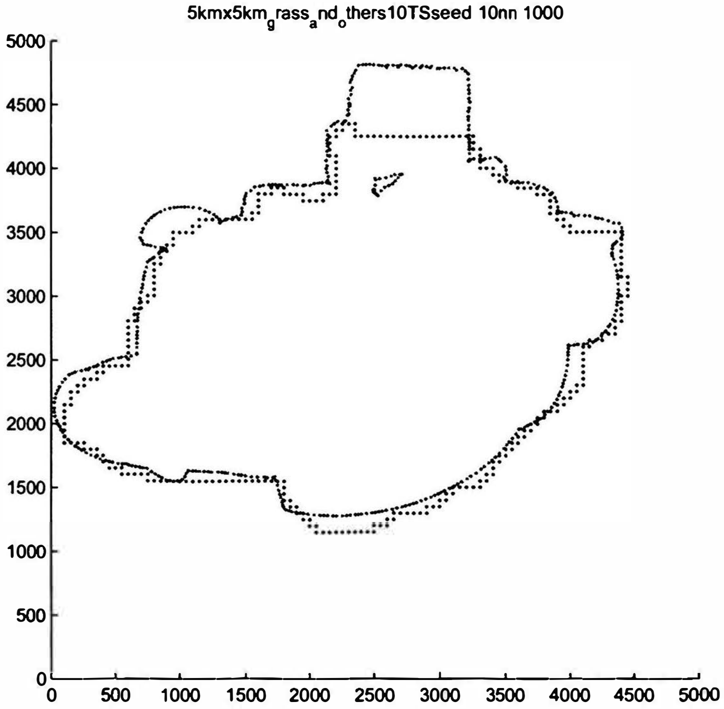

Figure 10: Incorrect final fireline estimation

the possible pathways. This has the effect of breaking false ridges apart and eliminating false paths. If the correct threshold value is used, the entire correct fireline remains while ideally all other paths are incorrect paths are eliminated. The trick to implementing this option is in choosing the correct threshold to decide which elements should be set to the minimum value. Choosing an incorrect threshold value can eliminate a position along the actual fireline, removing it as an option for Dijkstra's algorithm as well, resulting in essentially no fireline being detected.

When executing the MATLAB code, there exists a variable which keeps track of the number of points on the graph currently being considered as potential points that represent the fireline. When implementing the minimum intensity threshold value for which all valid fireline points must meet, it is possible to detect whether the threshold value is too high by analyzing the number of potential points eliminated by implementing the threshold after running the watershed function. If the number of elements post-threshold is drastically lower than the original graph, it is likely that not only have the alternate paths with only a few elements been eliminated, but the main, actual fireline path has also been eliminated as well, such as in Figure 11.

A method was implemented to estimate the most accurate threshold value to eliminate the largest amount of incorrect paths possible without eliminating the correct path. What occurs with the implemented method is that the number of elements present for the original

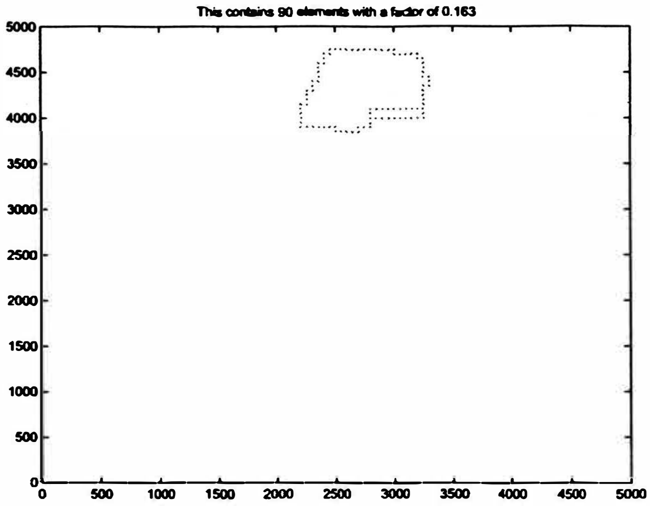

Figure 11: Intensity threshold value too high, main fireline elements eliminated

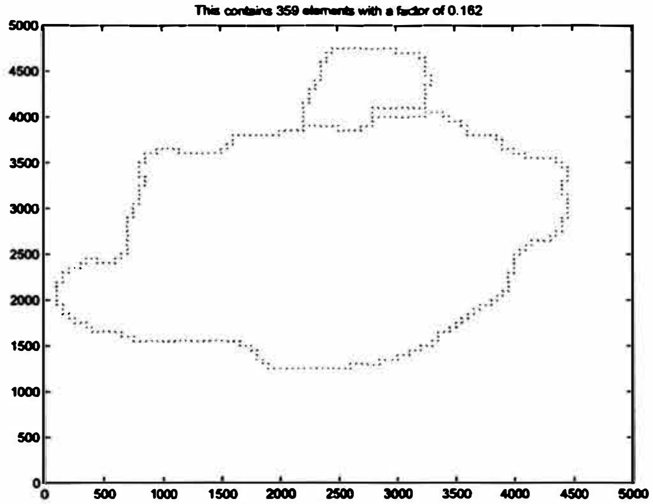

Figure 12: Final fireline estimation with dynamic intensity threshold value

intensity map (Figure 9) with no threshold set is analyzed, then the threshold value is increased in increments of 2.5% of the maximum intensity value. After each increase of the threshold value, the number of elements still present is compared to the number present for the previous threshold value tested. If the number of elements present drops below 50% of its previous value, it is assumed that an element from the main fireline has been eliminated and a broken graph has resulted (Figure 11). The threshold value is then backed down to its previous value. At this point, smaller increments of .1% of the maximum intensity value are used for the threshold and tested as before until a broken graph is once again detected. The immediately previous threshold value is now the value used for the final output, and the watershed function is run for a final time. The resulting graph (Figure 12) contains fewer false path options considered by Dijkstra's algorithm. It should be noted that this method eliminates an increasing number of false paths as the grid size increases. A higher grid size allows for the existence of more false ridgelines. A 500 x 500 grid produces the original watershed output shown in Figure 13, with Figure 14 showing the output graph after implementing the threshold. A drastic reduction in alternate paths is shown for this example.

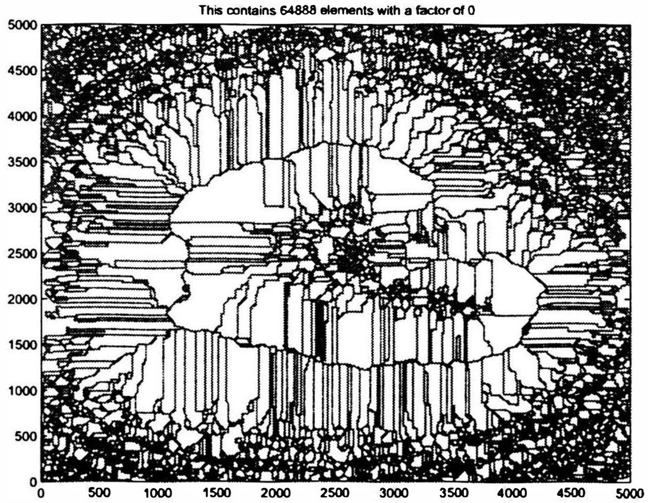

Figure 13: Higher resolution original fireline estimation

Figure 15, 16, and 17 show some example results for the fireline detection algorithm. The red outline depicts the ideal fireline supplied by the FARSITE output, with the blue line being the estimated fireline. The estimated fireline is shown to be a close approximation to the ideal one, with only small discrepencies occuring. The number of nodes required for an accurate estimate is not equal amongst all fires. The complicated fireline shown by Figure 15 required 4500 nodes for the algorithm to be a close approximation, while the fire from Figure 16 required only 2500 nodes for accurate results. Figure 17 used 3500 nodes and achieves a close approximation as well. The number of nodes can also vary depending on the node distributions, as one distribution may lead toward more favorable detection of the difficult

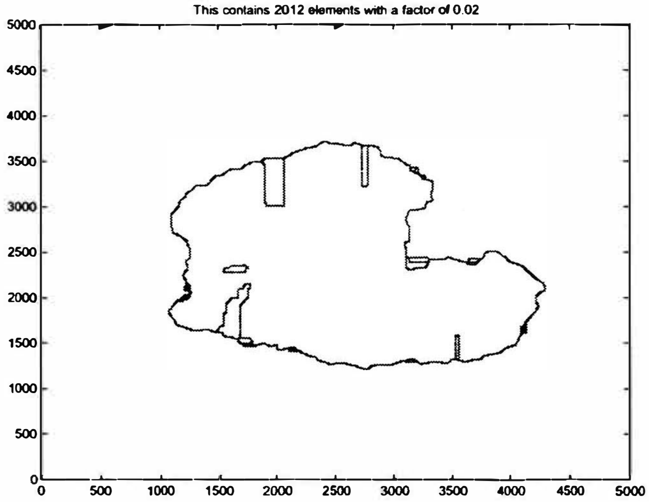

Figure 14: Higher resolution fireline estimation with dynamic intensity threshold

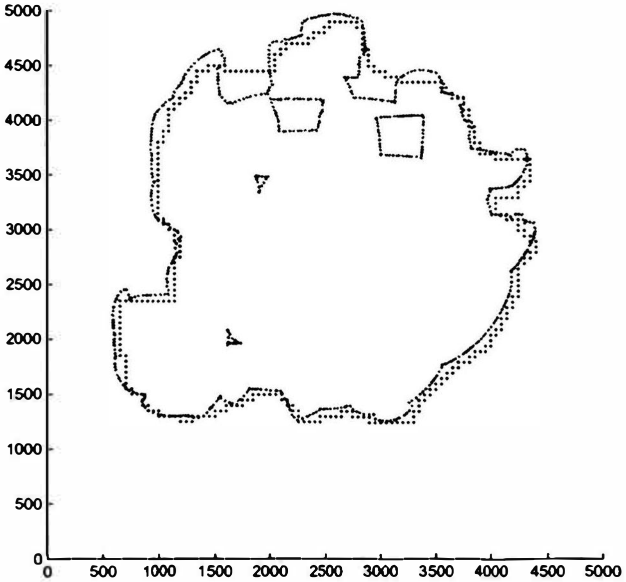

Figure 15: Estimated fireline, 4500 nodes

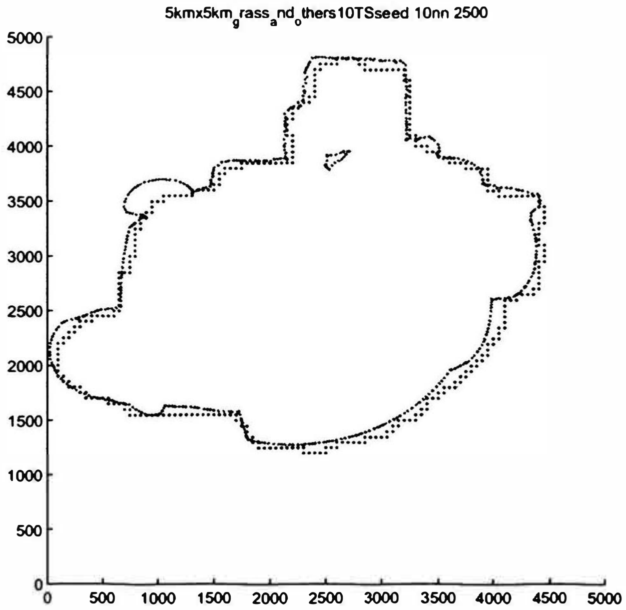

Figure 16: Estimated fireline, 2500 nodes

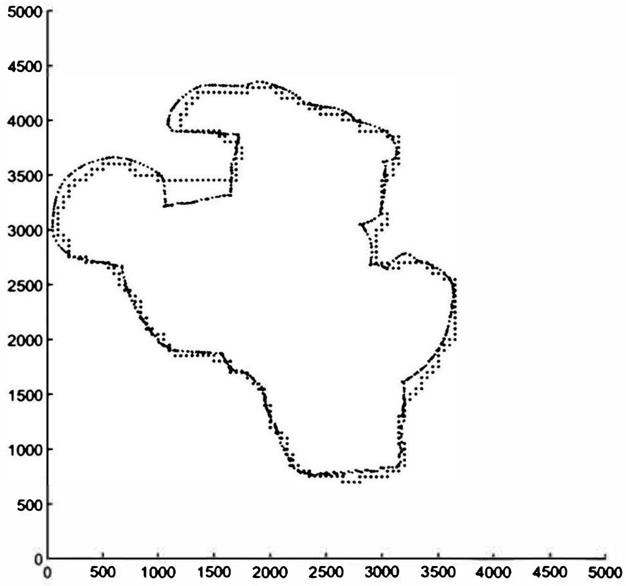

Figure 17: Estimated fireline, 3000 nodes

fireline elements than others. These example results illustrate that there isn't a lower bound of the number of sensors necessary for accurate estimation, that the lower bound depends on the contour of the fire itself. However, by using a minimum of 5000 nodes for all fires, the liklihood of accurate detection will be high.

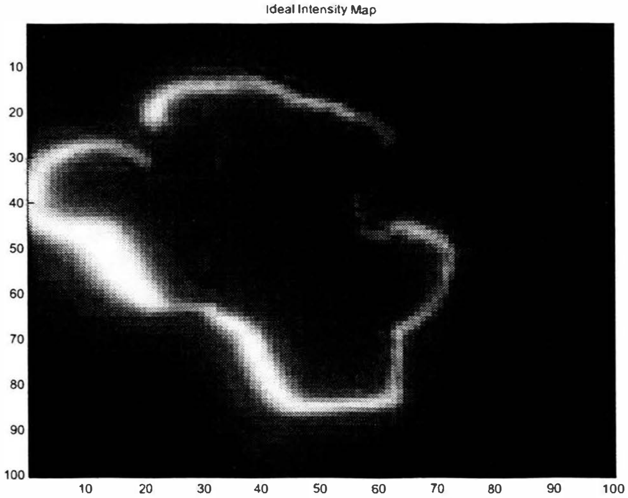

Figure 18: Ideal intenstiy map with low intensity inner curve

Even with the modified algorithm, a type of incorrect path can still exist which is difficult to eliminate and avoid choosing as the correct fireline. Figure 17 shows an example of such a region. The issue with such a region is that the actual fireline is of a lower intensity, shown in Figure 18, making the path intensity sum relatively small, whereas the incorrect path has very high intensity values contributed by the surrounding high intensity fire elements. Using the previously described method of establishing a threshold has no effect on such a path, as all of the elements along the path are still high enough where any threshold value which would eliminate the incorrect path would ultimately eliminate some valid points along the actual fireline as well. The ability to cope with such regions is being left up to future research. A higher node count has been found to be beneficial in avoiding such mistakes, but this is a costly and not always reliable solution to the problem, as there needs to be a guarantee that the increased node density actually occurs directly around the difficult region to have any positive effect. Perhaps future methods may be developed depending on a different metric for path decisions using additional sensor data provided by nodes with more sophisticated sensors. Even with the potential errors for these regions in fireline estimation, the intensity map still remains quite accurate, which used in conjunction with the estimated path still provides a coherent and useful characteristic of the wildfire region.

The original analysis of DOWsim's capabilities to accurately estimate intensity maps and fireline locations was solely performed using 1km x 1km size areas. Based on those results, it was shown that for that region size, a distribution of 1000 nodes provided high accuracy, with any further increase in the number of distributed nodes providing only small gains in the intensity map accuracy. With the availability of fire simulations for larger geographic regions, it then became possible to analyze the accuracy of the system for different sized landscapes. The DOWsim software was updated to accommodate these larger area FARSITE outputs, and the number of required nodes for accurate measurement results is currently being analyzed. Along with this updated flexibility, increased efficiency was also introduced by further analyzing certain calculations being performed for every simulation, such as the ideal sensor distribution results, and under what conditions they actually needed to be recalculated. Eliminating redundant calculations managed to cut down the time required to conduct multiple, sequential simulations for the same fire. Any decrease in simulation time simply increases the speed at which results can be analyzed and conclusions made.

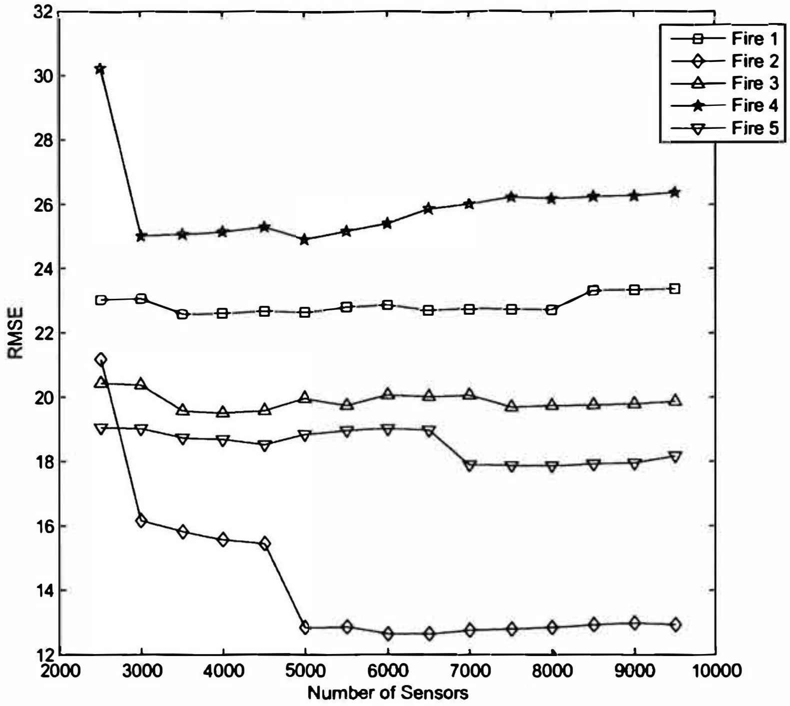

Figure 19: Root Mean Square Error plot

One of the primary metrics for analyzing the accuracy of the sensor network is by comparing the infrared energy detected at each grid position for the ideal distribution of sensor nodes to the energy calculated by the random sensor distribution along with interpolation. The error between the ideal and interpolated values is analyzed for each grid location, and quantized with the sum of the squares of the errors being taken the square root of. This value is calculated for each simulated fire multiple times with an increasing number of nodes distributed across the fire region. An example plot for 5 different 5km x 5km simulated fires is shown in Figure 19. These are the curves that are currently being used to analyze and determine the proper number of nodes necessary to achieve accurate results with limited potential gain for increased node distributions. Other potential metric values may be explored in the future to also take into account how accurate the fireline estimation algorithm is for that particular node distribution. With the current metric, the accuracy of the intensity map may be maximized while the final fireline estimation graph still has significant errors. This type of situation could occur if the difference between a correct and incorrect estimated fireline hinges upon subtle differences in intensity values. It may not seem evident solely by analyzing the RMSE graph that an increase in intensity accuracy would dramatically increase the fireline estimation. Some metric which takes that type of condition into account would be more value for analysis purposes.

Future work for this system should consist of analyzing the networking characteristics of such a system, ultimately making the decision as to which is the most capable ad-hoc routing protocol to transfer the information from the sensors to the central processing location. Future analysis also needs to be made to determine the ideal number of nodes required for larger geographic areas, ideally resulting in some standard, scalable calculation that maintains its effectiveness and accuracy across a wide range of landscape sizes

Although the current fireline estimation system still needs additional analysis, it should be evident by the current results that it has legitimate potential to become a robust and accurate method of wildfire detection. Assuming the sensor nodes can be manufacturing in large quantities at cheap prices, the value of such a system in increasing the overall efficiency of battling a wildfire has the potential to be of great benefit.

4 Aknowledgements

I would like to first acknowledge Dr. Shoukat Ali, my undergraduate research advisor for providing me the opportunity to expand my knowledge and experience of research at an academic level. I believe the students of the Electrical and Computer engineering department is losing a great teacher as he moves to work in industry. The experience has had a large influence in developing my skills as an engineer and researcher, but also in leading me to pursue graduate studies. I would also like to acknowledge and thank Matthew Gann and Michael Ellebrecht for welcoming me to work with them on their project. They were always willing to provide useful guidance to help me further understand the project and software tools used. Finally I would like to thank the University of Missouri - Rolla for providing and funding the Opportunities for Undergraduate Research Experience program. It is encouraging to know my university encourages such research and experience beyond the classroom setting.

References

[1] National Interagency Fire Center, http://www.nifc.gov/index.html.

[2] M. Gann, M. Ellebrecht and S. Ali, "A Simulation Environment Modeling the use of Wireless Sensor Network for the Detection and Mapping of Wildfires." Currently Unpublished.

[3] MODIS Fire and Thermal Anomalies, http://modis-fire.umd.edu/index.asp.