Camera Localization methods for Intelligent Room Systems using RF and Stereo Vision Techniques

Pradeep Natarajan, Garima Nahar {pnataraj, nahar} @usc.edu
EE 599, Wireless Sensor Networks, Intstructor: Prof. Bhaskar Krishnamachari,
University of Southern California, LA, CA-90089.

1 Abstract:

One of the important components of a multi sensor "intelligent" room, which can observe, track and react to its occupants, is a multi camera system. This system involves the development of algorithms that enable a set of cameras to communicate and cooperate with each other effectively so that they can monitor the events happening in the room. To achieve this, the cameras typically must first build a map of their relative locations. In this paper, we discuss RF and vision based techniques for estimating distances between cameras. The algorithm proposed for RF can estimate distances with relatively good accuracy even in the presence of random noise. We have also described a vision-based algorithm for localization using stereovision techniques. This algorithm can compute the location of the camera given the location of a calibration object and vice versa.

2 Introduction:

Security in homes, offices, schools and buildings is a growing concern in the current world scenario. Camera surveillance systems can act as pre-emptive safety measures that can register any suspicious activity and take necessary action. The applications for such surveillance systems range from indoor deployments, to parking structures, military and defense applications. They can also be used for object tracking and monitoring applications.

In the current room surveillance systems, the video data from the cameras are archived in a main server. They operate continuously and send the information to a centralized server. This results in a very large amount of information. Much of the video data analysis is currently done by a human observer; this is tedious, time consuming and error prone. Archiving and storing these clips is a very big concern. In addition each camera performing this operation is oblivious of the kind of information the other camera is gathering resulting in redundancy. For a larger deployments, the load on the centralized server increases and might result in single point of failure if the server crashes.

Thus there is a need to build a system, which filters out the redundant data and sends only the interesting video clips to the server. In addition the cameras can share the computation work by talking to each other and the central server. The key to building such a system is the development of an automated camera localization system and efficient inter-camera communication protocols which enable the cameras to perform intelligent handoffs when a target moves out of a camera's range. The cameras can also be considered as sensors deployed indoors, in an ad-hoc manner sensing objects or movements. The main difference being, that unlike traditional sensors, the cameras are not power restricted and can have small processors on them. Thus, the camera localization problem is very similar to the localization problem in wireless sensor networks. In current systems, the camera localization process is done by manually measuring the locations of the individual cameras. The focus of this work is to explore ways to automate this process.

The rest of the paper is organized as follows. Section 3 lists the related work. In section 4 an algorithm for localization using RF signal strength measurements and the results obtained from our experiments are described. Section 4.1 gives a detailed description of the algorithm, Section 4.2, discusses the experimental setup used to validate the algorithm. In Section 4.3, the results and observations from these experiments are presented. Vision based localization techniques and the issues faced in such systems are discussed in section 5. Finally the conclusion and future work are presented in section 6.

3 Related Work:

A significant amount of research has been carried out for inter-node localization in wireless sensor networks using RF [1, 2, 3, 4, 5], ultrasonic [6, 8], IR [7, 11, 12] and optical [9] media.

RF based localization can be done over longer distances and also in non line-of-sight conditions. Besides, there are a lot of cheap off-the-shelf components that can be used. A very important characteristic of radio propagation is that increasing distance results in attenuation of the radio signal. The various radio propagation models are discussed in [10]. However, radio transceivers have relatively complex circuits and hence consume lot of power; in addition the received signal strength varies due to diffraction, refraction, reflection and multi path fading. Because of these problems, distance estimates based on the radio propagation models are not very accurate. The algorithm that we present in this paper for RSSI based localization, provides a method by which we can eliminate the effects of noise by taking multiple signal strength measurements instead of just one.

Our work on RF based localization is closest to the work done in [1]. In RADAR [1], the authors perform a similar task of estimating distance based on signal strength in indoor environment. Three base stations are used for this. The distance is computed from the received signal strength by applying a Wall Attenuation Factor (WAF) based on signal propagation model. The distance information is then used to locate a user by triangulation. The major draw back of this technique is that for applying WAF, complete information of the obstructions, number of walls should be known. As a result of which, it is not scalable.

The key difference between [1] and our technique is that RADAR attempts to track the location of people based on instantaneous signal strength measurements. On the other hand, in our algorithm the camera or node computes a weighted average based on signal strength measured over a period of time after eliminating noisy readings. The distance is then estimated from this weighted average.

The authors in [2] use a simple connectivity metric for localization in an outdoor environment. This localization algorithm makes strong assumptions about the radio model and symmetric distribution of reference nodes. In addition, it performs well in an uncluttered outdoor space. If the beacon interval is small it might result in network congestion and packet loss.

Other media such as Ultrasound can also be used for localization. In [6], the transmitters and receivers are synchronized by RF and ranging is done by ultrasound. [2] shows that ultrasonic localization does not perform well in outdoor conditions, since chances for interference from other ultrasound equipment increases. The *cricket* system [8] developed as a part of the *MIT Oxygen Project* [14] has developed a method for distance estimation using a combination of ultrasonic and RF devices. [15] gives a complete description of the RF/ultrasonic transceivers and the mathematics behind direction and distance estimation. This and other related works [4,5] indicate that distance estimates can be made with relatively high accuracy using a combination of RF and ultrasonic transceivers.

[7] discusses an IR based localization system in which, each person or object is tagged with an Active Badge which emits an IR pulse every 10 seconds. The sensors placed within the building receive this pulse and the location manager software calculates the object's location. Similar work is done in [11]. IR localization can be performed only in line of sight conditions. Hence, it is not ideal for camera localization because the cameras deployed will have distances in the order of meters and each camera might not be able to *see* the others.

Vision techniques are very common in mobile robot localizations. The authors in [11] discuss localization of mobile robots from visual information. They also use intensity as a metric for localization. [18] describes an image retrieval system based on features that are invariant with respect to image translations, rotations, and limited scale. [19] presents a fast self-localization approach in which the vision system extracts features for localization without processing the whole image and is a first step towards independence of lighting conditions.

4 RSSI Based Localization Algorithm:

In this section we propose an RF based localization algorithm, which can produce reasonably accurate distance estimates even in the presence of noise. This algorithm is based on the observations derived from our experimental data, which indicate that although there is significant instantaneous variation in received signal strength at a particular location due to random noise, over a

period of time the values observed tend to stabilize with respect to past measurements. Although the primary motivation of this work is to devise methods to automatically localize cameras deployed in a room, the ideas presented in this section are fairly generic. In our application we plan to implement them by placing radio transceivers on the cameras and then running the localization algorithm.

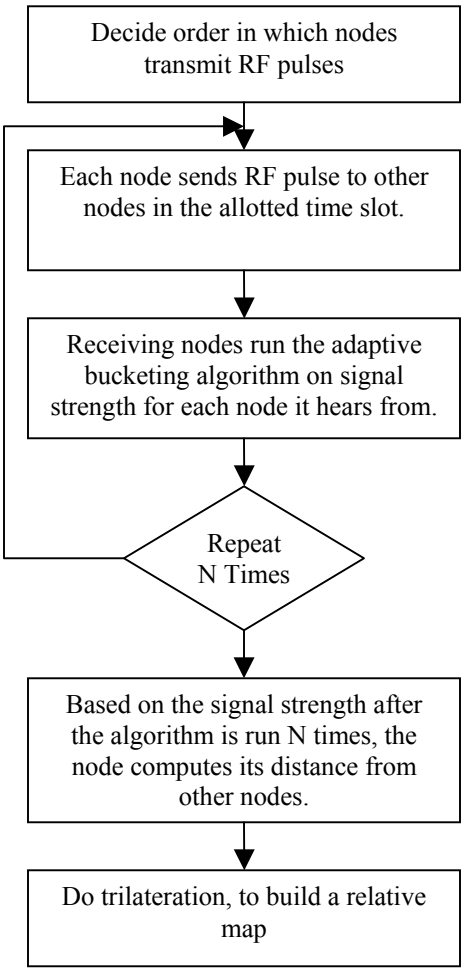


Figure 1: Localization Algorithm

4.1 Localization Algorithm:

The basic idea behind the proposed localization algorithm is to collect a set of signal strength measurements from each node, eliminate noisy outlier measurements in this set and then use the mean of the signal strengths from the remaining measurements to estimate the distances. Figure 1 describes the details of the proposed localization algorithm. Initially, the system decides the order in which the nodes transmit the signals. In our particular application, since the number of nodes is quite small (~5-10 cameras), we can manually configure the order. However, in large-scale deployments or in applications in which the nodes are physically inaccessible, deciding this order manually is infeasible. In these large-scale cases, we can use a technique similar to the CSMA/CD protocol that is widely used in networks. Here, each node tries to get the first position and transmits a pulse. If it senses that its signal is corrupted by signals from other nodes, it backs off for a random amount of time and tries to transmit again. The node, which first gets to transmit successfully, gets the first slot. This process is repeated until all the other nodes get their respective slots. We can reduce the energy costs involved in this process by adopting a clustering mechanism in which the nodes in a cluster decide the order, then the cluster heads decide the order amongst themselves and so on.

Once the order is decided, each node transmits an RF pulse in its allotted time slot while the other nodes listen. This process is repeated N times, where N is a protocol parameter. Thus, each node has N readings from every other node. An important thing to note here is that, choosing the value of N involves a tradeoff between accuracy and the energy and time costs involved since more measurements increase accuracy but require more time and energy. Statistically speaking, the effect of random noise is less pronounced if we have a large number of readings. However, a large value of N implies a greater time taken for localization and hence greater energy costs. Although in our specific application energy is not a constraint, we would still want to choose an optimal value of N so that we can perform localization quickly.

Every time a node receives the signal strength from another node, it runs the *Adaptive Bucketing Algorithm* to identify the cluster to which this reading belongs. The complete details of this algorithm are shown in Figure 2. Here, each node maintains a table containing the *Node ID* and *list of clusters* corresponding to that node. Each cluster is represented by the tuple *(Mean, NumValues)*, where *Mean* corresponds to the mean value of the cluster and *NumValues* indicates the number of readings that falls within the cluster. When a node receives an RF pulse from another

node, it first retrieves the entry in the table corresponding to that particular node. It then checks to see if the signal strength received belongs to any of the clusters of that node. This is done as follows – If S_i is the received signal strength, it belongs to a cluster $C[\mu_j, n_j]$ if

μ_j-Th <= S_i <= μ_j+Th

where μ_j is the Mean and n_j is the NumValues of cluster C. Here, *Th* is a protocol parameter that determines the size of the bucket. The value of Th can have a significant impact on the effectiveness of the algorithm in identifying and removing noisy readings. If the value of Th is too large, many noisy values will be incorrectly included in the cluster while a small value of Th will result in a large number of clusters with very few values in them.

```
N <- number if signal strength used to compute that average
L <- List of (μ, n)
S <- List of N signal strengths

Initialize L <- NULL

for (i=1 to N)
        boolean used false;
        for each l in L
                if (L.μ -Th<= S[i] <= L.μ +Th)
                then
                        L. μ = L. μ * l.n +S[i] ;
                                L.n+1
                        L.n = l.n+1;
                        used = true;
                end if
        end for
        if (! Used)
                add (L, (S[i], 1))
        end if
end for
for each l ∈ L
        for each m ∈ L- {l}
                if | l. μ –m. μ | < 2 * Th
                merge (l,m)
        end for
end for
choose l ∈ L with max (l.n)
```

Figure 2: Adaptive Bucketing Algorithm

If S_i belongs to a particular cluster, the parameters of the cluster are updated as follows –

$\mu_j = (\mu_{j,prev} * n_{j,prev} + S_i)/(n_{j,prev} +1)$

$n_j = n_{j,prev} + 1$

If S_i belongs to no cluster, we add a new cluster $(S_i, 1)$, to the list of clusters of that node.

Since the means of the clusters are continuously changed, it might be possible that the buckets of 2 clusters start overlapping. In such a case we add the values in the overlapping region to both the buckets and then merge the 2 buckets in the end. If $C[\mu_j, n_j]$ and $C[\mu_k, n_k]$ are 2 overlapping buckets then the merged bucket is $C[(n_k*\mu_k + n_j*\mu_j)/(n_j+n_k), n_j+n_k]$. Since we take the weighted average of the mean values while merging, the readings in the overlapping region tend to get more weight (twice the weight given to others) than the other values in the two buckets. The merging process also helps in overcoming the effects of choosing a wrong value for Th. Even if the value of Th is small, the bucket size can potentially grow larger if there is significant overlap between adjacent buckets. After the merging process, we choose the bucket, which has the maximum *NumValues*. We use the *Mean* value of this bucket to estimate the distance using an exponential decay model as shown below -

$$\left[\frac{\overline{P_r(d)}}{P_r(d_0)}\right]_{dB} = -10\beta \log\left(\frac{d}{d_0}\right)$$

Equation 1 [16]

The various parameters are explained below: -

$P_r(d)$: Power measured at distance d

$P_r(d_0)$: Power measured at reference distance d_0 = -47 db (Power of the Transmitting base station)

d_0: reference distance for this experiment was EEB 351 d_0 =5 meters

β: Path loss exponent. The value of β depends on the environment. Table 1 shows the value for various environments.

Environment		β
Outdoor	Free space	2
	Shadowed urban area	2.7 to 5
In building	Line-of-sight	1.6 to 1.8
	Obstructed	4 to 6

Table 1

Once we get the distance estimates between nodes, we can build a map of relative locations using *Trilateration*.

An, important point to note here is that the *Adaptive Bucket Algorithm* presented in this section is really an efficient way to estimate the cluster mean. The time complexity of this algorithm is O(kN) where, k is the number of clusters and N is the number of readings. The space complexity is O(k). In general we expect k to be much smaller than N because most of the readings except a few tend to be close to each other. The space-time complexity of this algorithm compares very favorably with other outlier removal techniques such as median calculation which has a O(N*logN) time complexity and O(N) space complexity. Also, the adaptive bucketing algorithm runs on the data as it arrives making it an online algorithm.

In the next subsections, we present the experiments we conducted with the RSSI based localization presented in this section and the results we obtained.

4.2 Experimental Procedure:

The experiments were conducted on the 3rd floor of EEB [13]. The base station was located in EEB 351, Networking Laboratory at USC. The base station was used as a transmitter and a laptop with a wireless card was used as a receiver.

We placed the laptop at different locations in the floor (marked by "X" in figure 3) and listened to the signal strength at each of these locations. We repeated this process at 22 different locations and obtained 14 different sets of readings at each location for a total of ~300 readings. In order to capture the variation in signal strength measurements over a short period of time and over a long period, we took sets of readings on the same day as well as over several days.

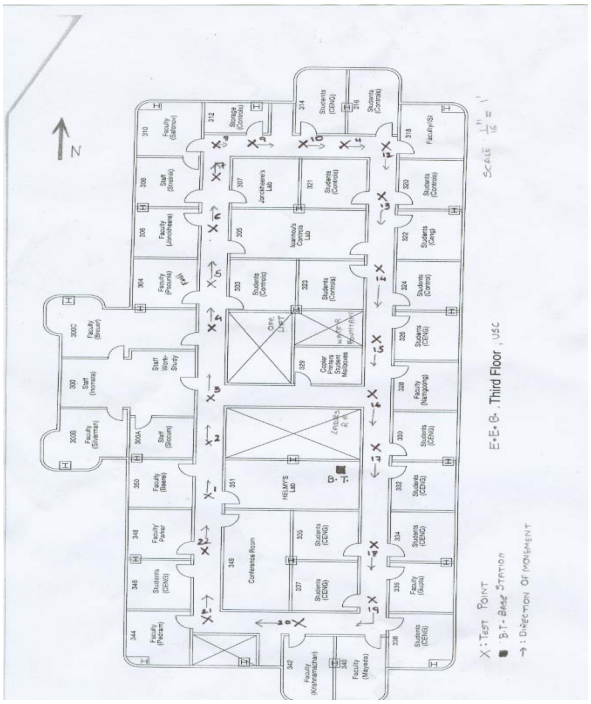


Figure 3: Floor Map of EEB 3rd Floor, USC.

Although, the topology of this setup is much different from the ones encountered in an actual implementation, it captures most of the variations in signal strength encountered in real life. Firstly, the locations were chosen so that they receive signal strengths at different qualities (good, fair and poor). Besides, most of the locations do not have a direct line of sight to the base station and hence experience significant multi path fading effects. Finally, the environment is subject to significant dynamics due to the movement of people and other activities. However, one factor that we did not include in our measurements is the effect of the direction of receiver on signal strength measurements. The main reason for this is that, in our application we can turn the receiver so that we can get the best signal strength possible. However, this may not be the case in many other applications. Hence a more careful analysis of the effect of direction of receiver on measured signal strength maybe needed.

4.3 Results:

In this section, we discuss the various results we obtained from the experimental setup. The distances of the test points from the base station were estimated from the signal strengths by

four techniques. Firstly the instantaneous signal strength measurements were used to compute distances. Then 5, 9 and 14 readings for each test point were chosen to calculate the mean, median and the weighted mean from the *Adaptive Bucketing Algorithm*. All the measured signal strengths at different test point locations are tabulated in the appendix B. A comparison of location estimation errors using all these four methods is described in the following subsections.

4.3.1 Distance Estimation:

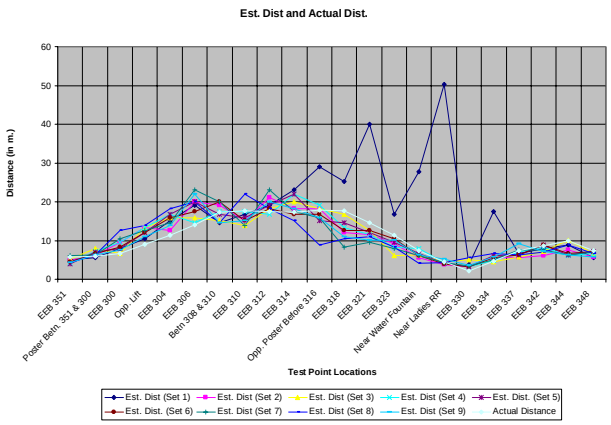


Figure 4: actual and estimated distances using instantaneous signal strength measurements

In figure 4, the distances estimated by using instantaneous signal strengths are depicted. As expected, the instantaneous distance estimates exhibit significant variation and hence the estimation errors also vary significantly. In particular, readings in set 1 are very weak and hence the distance estimates are wide off target. This may be explained by the fact that when this set of readings was taken, there was a high level of activity in the lab close to the base station. However, it can be observed that a majority of the readings provide estimates that closely follow the actual distance curve. This observation is the basic intuition behind our localization algorithm.

We used these signal strength measurements as inputs for the *Bucketing Algorithm* with N as 5, 9 and 14. Distances were then estimated using the weighted average with β=5 in the exponential decay model. As we can see from figure 5, the distance estimates with all three values of N, follow the 'actual distance' curve within 20-25% accuracy in most cases. Even the worst estimation error is about 70% when compared to a 600% error with instantaneous measurements.

The results of the estimated distances by using median of signal strengths measured are shown in figure 6. The median is computed by using 5, 9 and 14 readings of the signal strength.

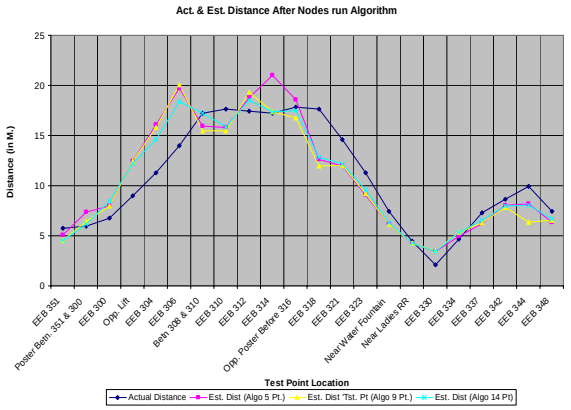


Figure 5: This graph indicates the actual distance and estimated distance at each test point after the nodes compute weighted average of the signal strength with varying N = 5, 9 and 14.

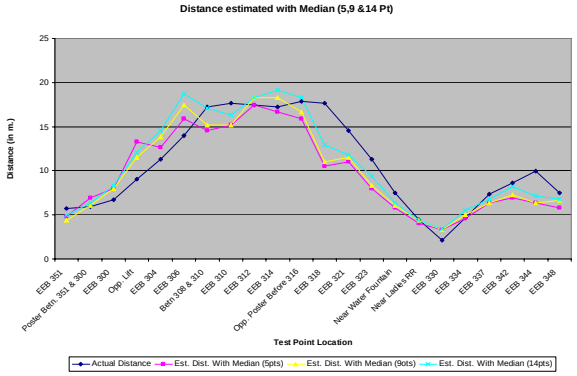


Figure 6: Distance Estimated with Median found by using 5, 9 and 14 Sig. Strength Measurements.

The distance estimation curve of the median also follows the actual distance curve fairly closely and the results are comparable to the bucketing algorithm results.

4.3.2 Relative Error in distance estimation:

In this section we plot the relative error in distance estimation at various locations. Relative error is defined by,

$$\text{Relative Error} = \frac{\text{Est. distance} - \text{Actual Distance}}{\text{Actual distance}}$$

Figure 7, denotes the relative errors computed by using instantaneous signal strength measurements.

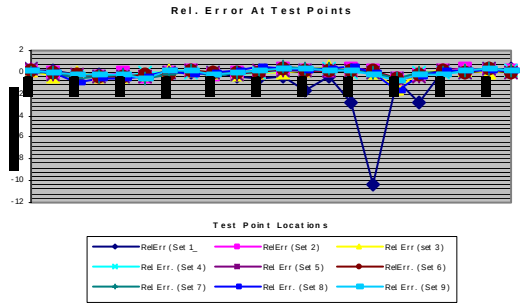


Figure 7: The relative error in estimated distances at test points due to individual signal strength measurements.

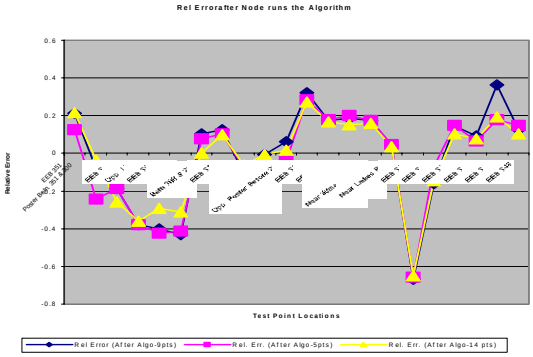


Figure 8: The error at test points after nodes run the adaptive bucketing algorithm with N=5, 9 and 14.

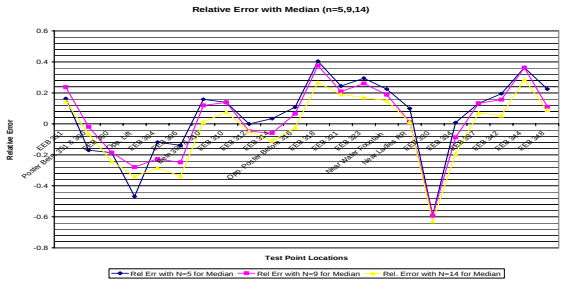


Figure 9: The error at test points after nodes use the median of readings with N=5, 9 and 14 for distance estimation.

As we can see from figure 7, the variation in error seems random. Figure 8 shows the relative errors obtained after using the Adaptive Bucketing Algorithm using N=5, 9, and 14. Figure 9 shows the errors obtained after using median signal strength measurements with N = 5, 9, and 14.

Again, figures 8 and 9 show that there is a close parallel between the errors obtained using median and adaptive bucketing algorithm on the set of readings we have. These results are quite understandable because both the median and adaptive bucketing algorithm are outlier removal techniques. Also, since the readings in most of the locations are well clustered with very few outliers as in figure 10, the median tends to be very close to the weighted mean estimated (figure 11). Thus, these results demonstrate the fact that the *Adaptive Bucketing Algorithm* is an efficient cluster mean estimation technique which involves much less space-time costs when compared to median calculation. Besides, in cases where there is a more significant variation in the measured signal strength, the bucketing algorithm performs a better job of finding the cluster mean. Figure 10, 11, 12, 13, 14 and 15 demonstrate this fact. However, a better cluster mean estimate need not necessarily translate to a better distance estimate. For example, while the distance estimates using ABA gives better results for locations 16, 21 and 10. On the other hand median estimates give better distance estimates for locations 11 and 13. This is because the distance estimate not only depends on the estimated mean but also on the β parameter of the radio attenuation model. However, even in these cases the difference between the estimates is quite small.

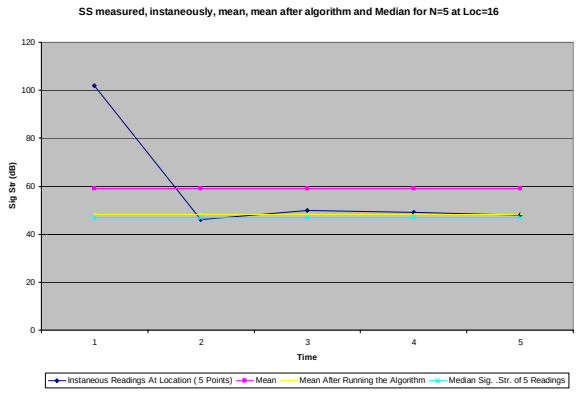


Figure 10: This graph depicts the variation of the 5 instantaneous signal strengths, its mean, median and weighted mean computed by the algorithm at location no. 16

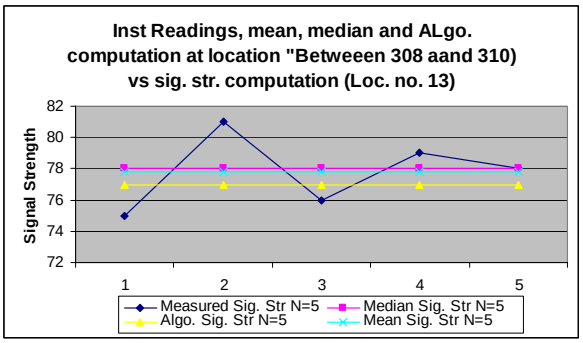


Figure 11: variation of the 9 instantaneous signal strengths, its mean, median and weighted mean computed by the algorithm at location no. 13

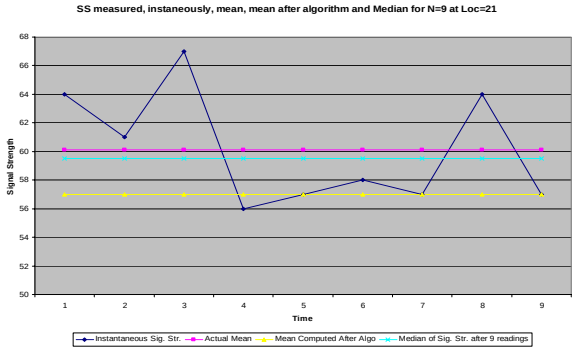


Figure 12: This graph depicts the variation of the 9 instantaneous signal strengths, its mean, median and weighted mean computed by the algorithm at location no. 21

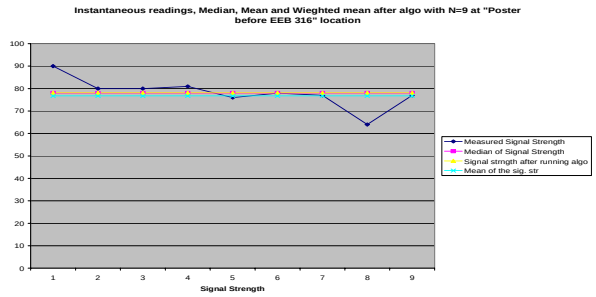


Figure 13: This graph depicts the variation of the 9 instantaneous signal strengths, its mean, median and weighted mean computed by the algorithm at location no. 11

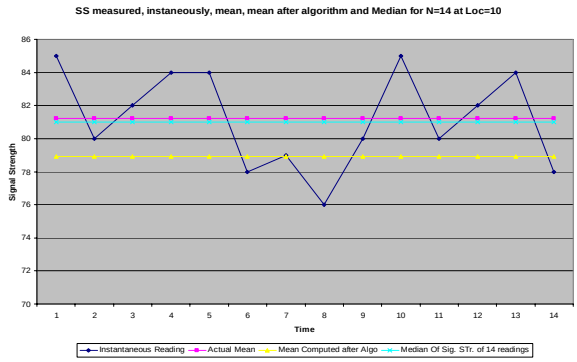


Figure 14: This graph depicts the variation of the 14 instantaneous signal strengths, its mean, median and weighted mean computed by the algorithm at location no. 10

4.3.3 Trilateration Results:

In this subsection we have presented the results of the actual location estimation using the distances obtained with the bucketing algorithm and median measurements. A short summary of the most salient results is shown in Table 2. The complete set of results can be found in the Appendix C. In Table 2, the first column indicates the locations used in the calculation, the second column shows the distance estimation errors for these locations when the adaptive bucketing algorithm was used, and the third column indicates the errors in the x and y co-ordinate estimates when the distances from the adaptive bucketing algorithm is used. For comparison, we have included the data for these locations from the median measurements in columns 4 and 5. In general we can say that greater distance estimation errors leads to greater location estimation errors. However, this may not be the case in many particular instances. Our observations on the data indicate that if one of the distance estimation errors exceeds 2 m there is a significant error in location estimates. But in most cases, the location estimation errors are well below 1.5 m in both the x and y co-ordinates.

Locations	Algo 14 Estimation Errors	Algo. Location (x,y)	Median 14 Estimation Errors	Median Location(x,y)
1, 9, 14	1.245361,-1.079342,1.714668	2.074924, -0.562951	0.828925,-0.833527,1.933085	2.063450, -0.225717
1,10,14	1.245361,-0.127611,1.714668	1.223799, -0.086321	0.828925,-1.885204,1.933085	6.525088, -2.724235
1,10,15	1.245361,-0.127611,1.199630	0.298272, 0.101128	0.828925,-1.885204,1.120427	3.307268 -2.072524
1,11,16	1.245361,2.425977,-0.675423	-1.233359, 0.337730	0.828925,-0.443527,0.054087	-0.375942, -0.516736
2,9,15	-0.204124,-1.079342,1.199630	1.903504, -0.057710	-0.399573,-0.833527,1.120427	1.663865, 0.287494
2,9,16	-0.204124,-1.079342,0.165749	0.425576, 0.025554	-0.399573,-0.833527,0.054087	0.378161, 0.359928
2,9,17	-0.204124,-1.079342,-1.353975	-0.007329, 0.049943	-0.399573,-0.833527,-1.314552	0.168074, 0.371764
2,10,13	-0.204124,-0.127611,2.425977	6.722935, -0.776274	-0.399573,-1.885204,2.775163	14.892076, -4.400577
8,13,20	1.782083,2.425977, 0.648951	-1.679884, 1.403871	1.354788,2.775163,0.463048	-0.024978 1.476493

Table 2: Location Estimation Errors

5 Vision based Localization:

An alternative localization approach that we considered is *Vision Based Localization*. This approach is particularly attractive in the context of the *Intelligent Room* because it uses the cameras themselves and requires no additional hardware. Also, the localization module, which gives the location of the camera relative to an object at a known location, can easily be reversed to an *Object Tracking* module, which gives the location of the object relative to the camera. Further, the accuracy of a vision based localization system and the factors that affect it are not well studied. The different steps in the Vision Based Localization system are shown in the figure 19.

In *step 1*, each camera first makes sweep of the room to locate the object. This is done by incrementing the tilt and pan values in small steps, capturing the image at each position, and checking if the *calibration object* is present in the image. After making a complete sweep of the room, the camera moves to the position, which had the best view of the object.

Then in *step 2*, the camera makes a small move to either side of the best position of *step 1* and grabs the images at these positions.

In *step3*, we estimate the distance between the object and the camera using *stereovision* technique. This technique makes use of the differences between images of the same scene taken from slightly different locations to estimate the distance. A more detailed mathematical explanation of the technique is given in the Appendix A. For estimating the distance using stereovision, we must first *segment* the object from the images grabbed. This is a complex problem and there are no known algorithms that work well under all conditions.

Then we repeat steps 1-3 at 3 different known locations of the calibration object. Using the distances estimated in these locations of the object, we estimate the location of the camera using *Trilateration*.

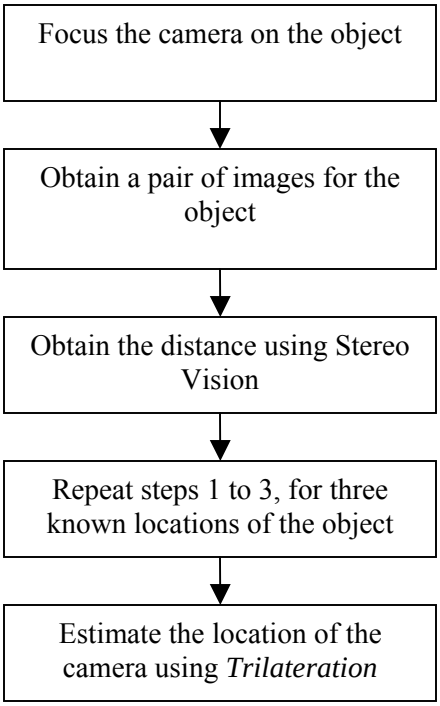


Figure 19: Stereovision based localization algorithm

5.1 Experimental Procedure:

In the experiments we conducted, we used a set of four cameras (*nerd-cams*, see figure 20) deployed at various locations in *ilab* at USC. Each of these cameras was connected to a unique node in the *Beowulf cluster* through a *firewire* connection. We used a board with bright yellow color as our calibration object.

Figure 20: Nerd Cams

Initially we train the system to identify the RGB values of the calibration object. We do this by showing the calibration object to the cameras in various locations and under different illumination conditions. We collect the RGB values from these positions and then fit a *Kalman filter* to the data. The Kalman Filter is a Gaussian model of the data and hence has two parameters – namely the mean (μ) and standard deviation (σ).

Then, in the first step of the localization, each camera tries to focus on the object. It does this by making a complete sweep of the room and then focuses on the location with the best view of the object. This is done by starting at *tilt=0* and *pan=0* and incrementing the pan in steps of 5 until it reaches 180 degrees. Then it increments the tilt by 5 and starts decrements the pan in steps of 5 until the pan becomes 0. The tilt is incremented again and the whole process is repeated until the tilt becomes 180. At each position of the camera, we pass the image grabbed through the *color segmenter* we have developed. In order to speed up the segmentation process, we decimated the image twice so that the size of the image that was actually passed to the segmenter was only $1/8^{th}$ the original size. This color segmenter basically checks the RGB value of each pixel in the image to see if it lies within a range specified by the parameters of the Kalman filter obtained in the training phase. In our case this range was chosen to be $[\mu-2\sigma, \mu+2\sigma]$. This is because, the shape of the Gaussian curve is such that ~96% of the area under it lies within this range. If the pixel lies within this range, it is classified as a "*white*" pixel. Otherwise, it is classified as a "*black*" pixel. We then proceed to connect the neighboring white pixels to obtain "blobs".

From these positions, we identify the position at which the grabbed image contained the maximum number of white pixels. This is the position, which has the best view of the calibration object. So, we then proceed to move the camera to this position and then adjust it so that the camera focuses exactly on the object. Now, the camera make two small moves on either side of this position and grab the images in these positions. These two images are then passed through our color segmenter [21]. The center of the object in these positions is approximated to be the center of mass of the white pixels. Then, using the tilt, pan and the center of mass of the object in these positions the *stereovision* algorithm, estimates the distance between the camera, and the calibration object. The pictures in Figure 21 show a stereo image pair obtained at a particular position of the calibration object and the corresponding segmented images.

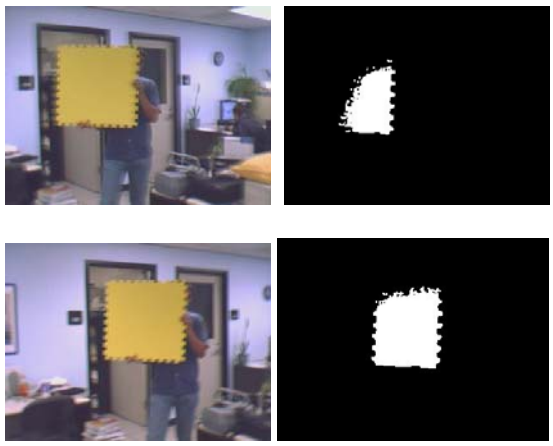


Figure 21: Original and Segmented images.

The figures show that the segmentation of the calibration object is not very accurate. Hence the estimated center of the calibration object, which is input to the stereo algorithm, is inaccurate and hence the distance estimates are way off the actual distances. This is mainly due to the fact that the segmentation algorithm that we are currently using is very simple. A more sophisticated algorithm based on more advanced features like shape is needed to achieve better segmentation. We are currently exploring such features based on the *Saliency Model* [20] developed by Itti et al. However, using such complicated features may make the algorithm slower than is feasible for our application. Also, another issue we faced is warping of images due to lens distortion. The images that we grabbed were not formed on an exactly flat plane, but instead were formed on a surface with some curvature. Hence the stereo calculations we did based on the flat plane assumption were not valid. Besides, variations in lighting and illumination conditions had an adverse impact on the segmenter. All these are fairly complicated problems. However, we hope that the work we have presented here explains the basic structure of a vision based localization system and the issues faced in their design.

6 Conclusions and Future Work:

In this paper, we presented a robust algorithm for localization that can give reasonably accurate distance estimates even in the presence of random noise. The results indicate that although there is significant variation in the instantaneous signal strength measurements in RF, over a period of time the values tend to stabilize. Besides, the clustering algorithm we presented can make fairly accurate estimates of the cluster mean and also has significantly less space-time complexity when compared to median calculation. Also, the results on our experimental data indicate that the localization errors due to the median and weighted mean measurements are nearly the same in most cases.

A useful extension to the *Adaptive Bucket Algorithm* will be to determine the *confidence value* of the estimated mean. This can be done by determining the ratio of the number of points in the final bucket to the total number of points measured and also by using the width of the final bucket. If the width of the bucket is large, then there is a large spread in the measurements and hence the confidence level must be low and vice versa. Also, if a large number of points are present in the final bucket, we must give a higher confidence value to the mean.

Another important direction for future work is to improve the vision based localization algorithm. This is important because such an algorithm not only performs localization, but also can be reversed easily to track objects. The key to achieving this is to improve the segmentation algorithm. Currently, the segmentation is done purely based on the RGB values of the pixels. Hence, the segmentation is highly dependent on lighting and illumination conditions. The features defined in the Saliency Model are a very good starting point for further improvements. This model proposes a hierarchy of features in which different *views* of an object are combined to form *instances,* which in turn are combined to form an *object model*. Also, since we can design the calibration object as we wish, we need to identify features that provide the best segmentation results and incorporate these features in the calibration object.

Apart from the RF based and Vision based localization discussed in this paper, there are several other media, which can be used. Some common examples are *infrared*, *laser* and *sonar-based* localization. Infrared localization has the disadvantage that it can be used only for short ranges. Laser requires very clear line-of-sight and also, the hardware for laser-based localization is very costly. Sonar is an attractive candidate to

consider. It has been demonstrated that sonar in combination with RF can provide very high accuracy distance estimates. A more careful study is needed to quantify the relative advantages and disadvantages of these methods.

7 Acknowledgements:

We would like to thank Prof.Bhaskar Krishnamachari for his help throughout the course of this work. We would also like to thank Shao Cheng and the Networking Lab at USC, for helping us with the RSSI based localization experiments. Finally, we would like to thank Prof.Laurent Itti, Prof.Michael Arbib and Nathan Mundhenk for their useful suggestions and also for providing us with the cameras and other hardware.

8 References:

[1] RADAR: An In-Building RF-based User Location and Tracking System; Paramvir Bahl and Venkata N. Padmanabhan, Microsoft Research,{bahl, padmanab}@microsoft.com

[2] N. Bulusu, J. Heidemann, and D. Estrin. GPS-less Low Cost Outdoor Localization for very Small Devices. IEEE Personal Communication, October 2000.

[3] A. Savvides, C.C. Han, and M.B. Srivastava. Dynamic Fine grained Localization in Ad-Hoc Sensor Networks. Proceedings of the Fifth International Conference on Mobile Computing and Networking, Mobicom 2001, pp. 166-179, Rome, Italy, July 2001

[4] T. He, C. Huang, B.M. Blum, J.A. stankovic, and T. Abdelzaher. Range-Free Localization Schemes for Large Scale Sensor Networks. MobiCom'03, San Diego, CA, USA. September 14-19, 2003.

[5] N. Patwari and A.O. Hero III. Using Proximity and Quantized RSS for Sensor Localization in Wireless Networks. WSNA'03. San Diego, CA, USA. September 19, 2003.

[6] A. Ward, A. Jones and A. Hooper "A new location technique for the active offices," IEEE Pesonal Communications, vol. 4, no. 5, Oct. 1007, pp. 42-47

[7] R. Want, A. Hopper, V. Falcao, and J. Gibbons. The active badge location system. *ACM Transactions on Information Systems,*10(1):91–102, January 1992.

[8] The Cricket Location-Support System Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan; MIT Laboratory for Computer Science,Cambridge, MA 02139; 6th ACM International Conference on Mobile Computing and Networking (ACM MOBICOM), Boston, MA, August 2000

[9] G.T. Sibley, M.H. Rahimi, and G.S. Sukhatme. Robomote: a tiny mobile robot platform for large-scale ad-hoc sensor networks. Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on , Volume: *2*, 11-15 May 2002.

[10] T.S. Rappaport, Wireless Communications – Principles and Practice, Prentice Hall, PTR, 1996.

[11] R. Azuma, "Tracking requirements for augmented reality", Communications of the ACM, vol. 36, no. 7, July 1993, pp 50-55.

[12] Vision-Based Localization for Mobile Platforms, Josep M. Porta and Ben J.A. Kr□ose
IAS Group, Informatics Institute, University of Amsterdam, Kruislaan 403, 1098SJ, Amsterdam, The Netherlands.

[13] USC MAP indicating location of EEB building, http://www.usc.edu/about/visit/upc//

[14] http://oxygen.lcs.mit.edu

[15] http://nms.lcs.mit.edu/papers/CricketCompass.pdf

[16] Pg. 189; The *ns* Manual (formerly *ns* Notes and Documentation); The VINT Project
A Collaboration between researchers at UC Berkeley, LBL, USC/ISI, and Xerox PARC.
Kevin Fall, Editor, Kannan Varadhan, Editor

[17] http://www.ruf.rice.edu/~lane/hyperstat/A27533.html

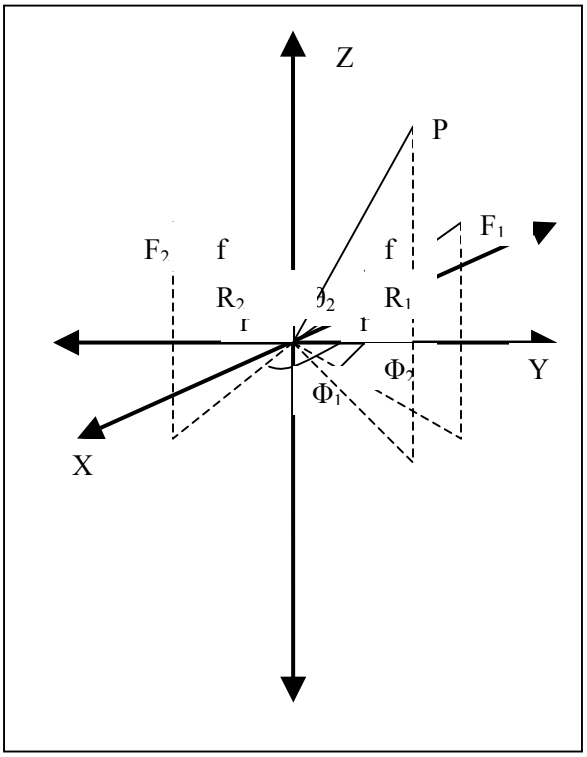
[18] Robust Vision-based Localization for Mobile Robots Using an Image Retrieval System Based on Invariant Features, J. Wolf, W. Burgard, H. Burkhardt , In Proc. of the IEEE International Conference on Robotics & Automation (ICRA), 2002

[19] Vision-Based Fast and Reactive Monte-Carlo Localization, Thomas Rofer, Matthias Jungel

[20] Computational Modeling of Visual Attention- Laurent Itti and Christof Koch

[21] Low cost high performance mobile robot design utilizing off the shelf parts and the Beowulf concept: The Beobot Project- T Nathanmundhenk et al.

Appendix A: Mathematics behind Stereo Vision



The above figure is the geometric representation of the camera in two different positions that is used for stereo calculations. The image plane in each of these positions is perpendicular to the lines OF_1 and OF_2 and is centered at R_1 and R_2 respectively. Let P(X, Y, Z) be the actual point whose location we are trying to estimate. Now, P lies on the line joining the image of P in the first position of the camera and the focus of the camera in this position (assuming pin hole camera model). Similarly, P also lies on the line joining the image and focus in the second position of the camera. Thus, P lies on the intersection of these two lines. This is the main principle used to determine P's location. The following discussion, presents a detailed mathematical treatment of this principle.

Firstly, the equation of the line OR_1 is given by,

$$\frac{x}{r*\sin\theta_1*\cos\Phi_1} = \frac{y}{r*\sin\theta_1*\sin\Phi_1} = \frac{z}{r*\cos\theta_1}$$

Since this line is perpendicular to the image plane at R_1, the equation of the image plane is given by,

$$(\sin\theta_1*\sin\Phi_1)x + (\sin\theta_1*\cos\Phi_1)y + (\cos\theta_1)z = r$$

Now, the y axis (this is different from the Y-axis in 3D shown in the figure) of the image plane is given by the line joining R_1 and Z axis along the image plane.
Hence, the equation of this line is given by,

$$\frac{x - r*\sin\theta_1*\sin\Phi_1}{-\sin\Phi_1} = \frac{x - r*\sin\theta_1*\sin\Phi_1}{-\cos\Phi_1} = \frac{z - r*\cos\theta_1}{\tan\theta_1}$$

Now, if (x_1,y_1) is the coordinate of the image in the image plane, the 3D location of the projection of the image point on the y-axis of the image plane is given by,

$$X_1' = r*\sin\theta_1*\sin\Phi_1 - y_1*\sin\Phi_1$$
$$Y_1' = r*\sin\theta_1*\cos\Phi_1 - y_1*\cos\Phi_1$$
$$Z_1' = r*\cos\theta_1 + y_1*\tan\theta_1$$

The actual image point lies at a distance x_1 from (X_1',Y_1',Z_1') along a line parallel to the x-axis of the image plane. The direction of vector of the x-axis of the image plane is the cross product of the image plane's y axis and OR_1 and is given by,
$(-i\ \cos\Phi_1/\cos\theta_1,\ j\ \sin\Phi_1/\cos\theta_1)$

Hence, the 3D location of the image point I_1 is given by,

$$X_1 = X_1' - x_1*\cos\Phi_1/\cos\theta_1$$
$$Y_1 = Y_1' - x_1*\sin\Phi_1/\cos\theta_1$$
$$Z_1 = Z_1'$$

Also, the 3D location of the focus is given by,

$$fx_1 = (r+f)*\sin\theta_1*\sin\Phi_1$$
$$fy_1 = (r+f)*\sin\theta_1*\cos\Phi_1$$
$$fz_1 = (r+f)*\cos\theta_1$$

Thus, the equation of the line I_1F_1 is given by,

$$x = X_1 + r_1*(fx_1 - X_1)$$
$$y = Y_1 + r_1*(fy_1 - Y_1)$$
$$z = Z_1 + r_1*(fz_1 - Z_1)$$

Similarly, the equation of the line I_2F_2 is given by,

$$x = X_2 + r_2*(fx_2 - X_2)$$
$$y = Y_2 + r_2*(fy_2 - Y_2)$$
$$z = Z_2 + r_2*(fz_2 - Z_2)$$

The point of intersection of these two lines can be obtained by solving for r_1 and r_2 as shown below-

$$(X_1-X_2) = r_2 (fx_2-X_2) - r_1(fx_1-X_1)$$
$$(Y_1-Y_2) = r_2(fy_2-Y_2) - r_1(fy_1-Y_1)$$
$$(Z_1-Z_2) = r_2 (fz_2-Z_2) - r_1(fz_1-Z_1)$$

Here, we have 3 equations to solve for 2 variables. Hence, we can solve any two of them, and check the results obtained by substituting them in the 3rd equation. If the results satisfy the 3rd equation, then the two lines intersect, otherwise they don't. Thus, upon solving equations 1 and 2 we get,

$$r_1 = \frac{(fy_2-Y_2)(X_1-X_2) - (fx_2-X_2)(Y_1-Y_2)}{(fx_2-X_2)(fy_1-Y_1) - (fy_2-Y_2)(fx_1-X_1)}$$

Substituting this value of r_1 in the equation of the line I_1F_1 gives us the co-ordinates of the point P.

Appendix B: Tabulated Signal Strength measurements for all the experiments

Actual Dist		1		2		3		4		5		6		7		8		9
	Sig Str.	Estimated	Sig Str.	Estimated	Sig Str.	Estimated	Sig Str.	Estimated	Sig Str.	Estimated	Sig Str.	Estimated	Sig Str.	Estimated	Sig Str.	Estimated	Sig Str.	Estimated
5.75	-52	5.03	51	4.809058	53	5.273027	48	4.188514	47	4	50	4.592615	56	6.054245	49	4.385913	49	4.385913
5.34	-54	5.53	56	6.054245	62	7.381049	60	7.278803	59	6.951203	58	6.638348	57	6.339573	57	6.339573	57	6.339573
6.73	-62	7.38	66	9.595332	58	6.638348	66	9.595332	62	7.381049	63	8.357184	68	10.52107	72	12.64311	61	7.621843
9.02	-68	10.52	73	13.245245	73	13.245245	73	13.245245	71	12.079807	71	12.07981	72	12.64311	74	13.86947	69	11.01631
11.31	-76	15.21	72	12.643111	77	15.924287	78	16.674776	78	16.674776	77	15.92429	75	14.52312	80	18.28353	74	13.86947
13.99	-81	19.14	82	20.047489	77	15.924287	75	14.523122	82	20.047489	79	17.46063	85	23.0176	82	20.04749	84	21.98163
17.26	-75	14.52	81	19.145205	76	15.207576	79	17.460634	78	16.674776	82	20.04749	82	20.04749	75	14.52312	75	14.52312
17.65	-78	16.67	76	15.207576	74	13.869474	79	17.460634	77	15.924287	76	15.20758	74	13.86947	84	21.98163	76	15.20758
17.45	-81	19.14	83	20.992298	80	18.283527	78	16.674776	81	19.145205	80	18.28353	85	23.0176	80	18.28353	82	20.04749
17.26	-85	23.01	80	18.283527	82	20.047489	84	21.981634	84	21.981634	78	16.67478	79	17.46063	76	15.20758	80	18.28353
17.84	-90	28.98	80	18.283527	80	18.283527	81	19.145205	76	15.207576	78	16.67478	77	15.92429	64	8.751046	77	15.92429
17.65	-87	25.23	71	12.079807	78	16.674776	73	13.245245	75	14.523122	72	12.64311	63	8.357184	68	10.52107	69	11.01631
14.58	-97	40	70	11.536126	72	12.643111	70	11.536126	71	12.079807	72	12.64311	66	9.595332	69	11.01631	67	10.04755
11.31	-78	16.67	65	9.16347	56	6.054245	63	8.357184	66	9.595332	68	10.52107	62	7.381049	63	8.357184	67	10.04755
7.46	-89	27.67	54	5.521537	57	6.339573	62	7.381049	58	6.638348	56	6.054245	56	6.054245	48	4.188514	57	6.339573
4.44	-102	50.36	46	3.81937	50	4.592615	49	4.385913	48	4.188514	47	4	49	4.385913	49	4.385913	53	5.273027
2.09	-46	3.81	43	3.327055	52	5.035702	43	3.327055	43	3.327055	43	3.327055	44	3.483854	54	5.521537	45	3.648043
4.63	-73	17.46	54	5.521537	50	4.592615	51	4.809058	56	6.054245	54	5.521537	52	5.035702	58	6.638348	55	5.781759
7.31	-58	6.63	54	5.521537	55	5.781759	59	6.951203	57	6.339573	57	6.339573	61	7.621843	57	6.339573	65	9.16347
8.63	-62	7.38	56	6.054245	60	7.278803	63	8.357184	64	8.751046	64	8.751046	61	7.621843	59	6.951203	60	7.278803
9.95	-64	8.75	61	7.621843	67	10.047545	56	6.054245	57	6.339573	58	6.638348	57	6.339573	64	8.751046	57	6.339573
7.46	-54	5.52	55	5.781759	58	6.638348	58	6.638348	60	7.278803	60	7.278803	59	6.951203	58	6.638348	55	5.781759

After Algo	N=5		N=9		N=14		Test Point
	Sig Str.	Estimated	Sig Str.	Estimated	Sig Str.	Estimated	
	52	5.035702	49.7	4.529602	49.58	4.504639	EEB 351
	60.332	7.390945	57.25	6.412982	56.32	6.144124	Poster Betn. 351 & 300
	62	7.381049	62	7.381049	*63.287*	8.468372	EEB 300
	71.6	12.41767	71.599	12.417665	71.307	12.251802	Opp. Lift
	77.25	16.10868	76.875	15.832883	*75.1538*	14.626351	EEB 304
	81.66	19.73604	82	20.047489	80.09	18.359461	EEB 306
	77	15.92429	76.34	15.447561	*78.725*	17.240902	Betn 308 & 310
	78.83	15.80229	76.34	15.447561	76.923	15.867317	EEB 310
	80.59	18.78711	81.2	19.32235	*80.29*	18.529343	EEB 312
	83	20.9923	78.857	17.346027	78.909	17.387611	EEB 314
	80.336	18.56863	78.115	16.763317	79.0625	17.510962	Opp. Poster Before 316
	72	12.64311	70.84	11.991125	72.336	12.846356	EEB 318
	70.8	11.96306	70.875	12.010469	71.133	12.154023	EEB 321
	64.667	9.024018	65.125	9.216372	66	9.595332	EEB 323
	56.332	6.147521	56.332	6.147521	56.727	6.26037	Near Water Fountain
	48.25	4.237015	48.286	4.243987	48.44	4.274251	Near Ladies RR
	43.857	3.460987	44	3.483854	43.75	3.443975	EEB 330
	51.667	4.958998	53.375	5.36488	53.133	5.305423	EEB 334
	56.599	6.223576	56.714	6.256623	57.75	6.562359	EEB 337
	62.25	8.073465	61.625	7.844405	62	7.981049	EEB 342
	62.5	8.166952	57	6.339573	62.14	8.032671	EEB 344
	57	6.339573	57.636	6.527998	58.15	6.684363	EEB 348

Appendix C: Location Estimation Errors

Note: In these tables the tables, the index for locations start from 0 instead of 1 as in Table 2 of main paper.

Table 1: Algo with N=14

Points	*Estimated Location Error*	
	X	Y
0 8 12	5.425885	-0.732620
0 8 13	2.074924	-0.562951
0 8 14	1.087828	-0.512971
0 8 15	-0.661773	-0.424384
0 8 16	-1.280893	-0.393036
0 8 17	-1.820787	-0.365700
0 8 18	-0.468810	-0.434154
0 8 19	-1.147285	-0.399801
0 9 12	6.323004	-1.119071
0 9 13	1.223799	-0.086321
0 9 14	0.298272	0.101128
0 9 15	-1.298446	0.424514
0 9 16	-1.594273	0.484428
0 9 17	-1.550389	0.475540
0 9 18	0.017583	0.157976
0 9 19	-0.208427	0.203750
0 10 12	10.574825	-2.950625
0 10 13	2.148144	-0.603954
0 10 14	0.678662	-0.194731
0 10 15	-1.233359	0.337730
0 10 16	-1.575955	0.433137
0 10 17	-1.564999	0.430086
0 10 18	-0.095133	0.020756
0 10 19	-0.372801	0.098081
0 11 19	3.395604	2.520628
0 12 19	1.180061	1.096350
0 13 19	0.217206	0.477371
0 14 19	-0.003254	0.335648
0 15 19	-0.832150	-0.197215
0 16 19	-1.253981	-0.468391
0 17 19	-2.010017	-0.954415
0 18 19	0.350142	0.562831
1 8 12	5.694981	-0.271314
1 8 13	2.635587	-0.098954
1 8 14	1.903504	-0.057710
1 8 15	0.425576	0.025554
1 8 16	-0.007329	0.049943
1 8 17	-0.247212	0.063457
1 8 18	1.280298	-0.022600
1 8 19	1.598569	-0.040530
1 9 12	6.722935	-0.776274
1 9 13	2.083210	0.269298
1 9 14	1.429090	0.416705
1 9 15	0.105485	0.714982
1 9 16	-0.057454	0.751701
1 9 17	0.143197	0.706484
1 9 18	1.699263	0.355821
1 9 19	2.176279	0.248324
1 10 12	10.881625	-2.819140
1 10 13	3.098233	-0.407384
1 10 14	1.873767	-0.027972
1 10 15	0.211253	0.487173
1 10 16	-0.044214	0.566332
1 10 17	0.041904	0.539647
1 10 18	1.427459	0.110320
1 10 19	1.719369	0.019869
1 11 19	6.197219	2.258795
1 12 19	3.395823	0.858096
1 13 19	2.141074	0.230722
1 14 19	1.790406	0.055388
1 15 19	0.671476	-0.504077
1 16 19	0.054286	-0.812672
1 17 19	-1.142439	-1.411035
1 18 19	0.841172	-0.419229
2 8 12	6.073396	0.377399
2 8 13	3.431351	0.559609
2 8 14	3.055923	0.585501
2 8 16	1.784494	0.673186
2 8 17	1.947291	0.661958
2 8 18	3.681396	0.542365
2 8 19	5.234776	0.435235
2 9 12	7.218634	-0.351389
2 9 13	3.237384	0.746887
2 9 14	2.923007	0.833612
2 9 16	1.952678	1.101289
2 9 17	2.311846	1.002208
2 9 18	3.771326	0.599593
2 9 19	4.862655	0.298537
2 10 12	11.010044	-2.764103
2 10 13	4.197354	-0.179980
2 10 14	3.279490	0.168175
2 10 16	1.805712	0.727195
2 10 17	1.956078	0.670159
2 10 18	3.162887	0.212404
2 10 19	3.883795	-0.061044
2 11 19	9.387308	1.960655
2 12 19	5.708849	0.609384
2 13 19	4.021702	-0.010384
2 14 19	3.481463	-0.208839
2 16 19	1.093343	-1.086108
2 17 19	-0.586993	-1.703375
2 18 19	1.162001	-1.060887
3 8 12	7.154700	2.231063
3 8 13	5.623173	2.373531
3 8 15	6.012362	2.337327
3 8 16	6.451466	2.296480
3 8 17	7.561815	2.193192
3 8 18	9.808992	1.984152
3 8 19	14.054515	1.589219
3 9 12	8.601624	0.834032
3 9 13	5.888075	1.843725
3 9 15	5.771224	1.887204
3 9 16	5.994335	1.804186
3 9 17	6.584002	1.584775
3 9 18	7.919975	1.087669
3 9 19	9.793095	0.390695
3 10 12	11.923017	-2.372828
3 10 13	6.646562	0.326753
3 10 15	5.303408	1.013948
3 10 16	5.274349	1.028815
3 10 17	5.502717	0.911976
3 10 18	6.487786	0.407987

```
3 10 19 7.682055 -0.203035
3 11 19 13.881710 1.540618
3 12 19 9.225990 0.231197
3 13 19 7.006461 -0.393046
3 15 19 4.113830 -1.206598
3 16 19 2.875100 -1.554991
3 17 19 0.435004 -2.241268
3 18 19 1.589798 -1.916482
4 8 12 8.176151 3.982122
4 8 13 7.646836 4.048286
4 8 14 8.964891 3.883529
4 8 15 9.567101 3.808253
4 8 16 10.472453 3.695084
4 8 17 12.252773 3.472544
4 8 18 14.787881 3.155655
4 8 19 20.535515 2.437201
4 9 12 9.632107 1.717302
4 9 13 7.798347 2.634182
4 9 14 8.363136 2.351788
4 9 15 8.261744 2.402484
4 9 16 8.564403 2.251154
4 9 17 9.187198 1.939757
4 9 18 10.325392 1.370660
4 9 19 12.195512 0.435600
4 10 12 12.188371 -2.259106
4 10 13 8.015231 0.609926
4 10 14 7.750369 0.792019
4 10 15 7.149746 1.204948
4 10 16 7.166602 1.193359
4 10 17 7.388834 1.040574
4 10 18 8.165421 0.506671
4 10 19 9.284875 -0.262953
4 11 19 15.981916 1.344337
4 12 19 10.688560 0.073931
4 13 19 8.138240 -0.538146
4 14 19 7.133029 -0.779396
4 15 19 4.783540 -1.343274
4 16 19 3.364624 -1.683813
4 17 19 0.625075 -2.341305
4 18 19 1.676088 -2.089062
5 10 12 14.102477 -1.438774
5 10 13 11.787672 1.390431
5 10 14 11.813424 1.358957
5 10 15 11.565670 1.661768
5 10 16 11.631269 1.581591
5 10 17 11.826370 1.343134
5 10 18 12.311217 0.750541
5 10 19 13.262090 -0.411634
5 11 19 20.059946 0.963213
5 12 19 13.936265 -0.275284
5 13 19 10.891354 -0.891109
5 14 19 9.564912 -1.159378
5 15 19 6.710855 -1.736603
5 16 19 4.932196 -2.096332
5 17 19 1.484479 -2.793623
5 18 19 1.978420 -2.693725
6 8 12 6.521171 1.145013
6 8 13 4.971139 1.833916
6 8 14 5.150135 1.754363
6 8 15 4.886594 1.871492
6 8 16 5.037193 1.804559
6 8 17 5.455364 1.618705
6 8 18 6.424760 1.187862
6 8 19 7.462381 0.726697
6 9 12 6.079618 -1.327689
6 9 13 4.596355 1.309220
6 9 14 4.600550 1.301764
6 9 15 4.427609 1.609215
6 9 16 4.467319 1.538618
6 9 17 4.593999 1.313411
6 9 18 4.919537 0.734677
6 9 19 5.161721 0.304127
6 10 12 5.393260 -5.171296
6 10 13 3.417374 -0.341355
6 10 14 3.212740 0.158861
6 10 15 2.962019 0.771735
6 10 16 2.939894 0.825819
6 10 17 2.975148 0.739641
6 10 18 3.190180 0.214010
6 10 19 3.293676 -0.038983
6 11 19 12.564140 1.663755
6 12 19 6.412021 0.533774
6 13 19 3.684647 0.032828
6 14 19 2.880305 -0.114909
6 15 19 0.714119 -0.512780
6 16 19 -0.345638 -0.707429
6 17 19 -1.927317 -0.997941
6 18 19 0.873326 -0.483537
7 12 19 -1.679884 1.403871
7 13 19 -3.984930 1.016107
7 14 19 -4.164674 0.985869
7 15 19 -5.515651 0.758602
7 16 19 -5.760216 0.717460
7 17 19 -5.325520 0.790587
7 18 19 -0.195253 1.653622
8 12 19 6.179217 0.558807
8 12 20 5.290299 -0.965053
8 12 21 5.522912 -0.566288
8 13 19 2.914180 0.131606
8 13 20 1.749015 -0.832669
8 13 21 2.241435 -0.425149
8 14 19 2.037082 0.016845
8 14 20 0.592498 -0.789434
8 14 21 1.316455 -0.385365
8 15 19 -0.350385 -0.295534
8 15 20 -1.366956 -0.716184
8 15 21 -0.390095 -0.311965
8 16 19 -1.392636 -0.431903
8 16 20 -2.128134 -0.687728
8 16 21 -0.975434 -0.286789
8 17 19 -2.682205 -0.600632
8 17 20 -2.896270 -0.659013
8 17 21 -1.455707 -0.266132
8 18 19 0.709956 -0.156798
8 18 20 -1.626240 -0.706491
8 18 21 -0.017597 -0.327987
8 19 21 -0.454723 -0.309185
9 12 19 8.046307 0.358044
9 12 20 6.187535 -1.235189
9 12 21 6.545200 -0.928618
9 13 19 2.025723 0.245510
9 13 20 0.501966 -0.385010
9 13 21 1.348770 -0.034608
9 14 19 0.728774 0.221268
9 14 20 -0.764902 -0.195572
9 14 21 0.398013 0.128963
```

9 15 19 -2.579696 0.159428
9 15 20 -2.814324 0.110884
9 15 21 -1.308177 0.422501
9 16 19 -3.569395 0.140928
9 16 20 -3.313206 0.185483
9 16 21 -1.631002 0.478040
9 17 19 -4.073275 0.131510
9 17 20 -3.486950 0.211464
9 17 21 -1.587251 0.470513
9 18 19 0.522848 0.217419
9 18 20 -1.743609 -0.049223
9 18 21 0.143361 0.172774
9 19 21 -0.053521 0.206646
10 13 19 4.636253 -0.089173
10 13 20 1.490766 -0.739964
10 13 21 2.488958 -0.533441
10 14 19 2.111559 0.005208
10 14 20 -0.417285 -0.347654
10 14 21 0.916720 -0.161514
10 15 19 -2.706599 0.185326
10 15 20 -2.908116 0.164479
10 15 21 -1.205780 0.340583
10 16 19 -3.908900 0.230272
10 16 20 -3.430454 0.271876
10 16 21 -1.591364 0.431797
10 17 19 -4.287837 0.244438
10 17 20 -3.543751 0.295171
10 17 21 -1.579851 0.429073
10 18 19 0.600713 0.061688
10 18 20 -1.738840 -0.075933
10 18 21 0.098153 0.032126
10 19 21 -0.144595 0.089550
11 19 21 1.858010 2.664328
12 19 21 0.680270 1.150091
13 19 21 0.162499 0.484385
14 19 21 0.041388 0.328672
15 19 21 -0.431267 -0.279027
16 19 21 -0.693307 -0.615937
17 19 21 -1.262549 -1.347819
18 19 21 0.300591 0.661933

Table 2: Median with N=14

0 8 12 5.559964 -0.402756
0 8 13 2.063450 -0.225717
0 8 14 0.854073 -0.164483
0 8 15 -0.699114 -0.085841
0 8 16 -1.090558 -0.066021
0 8 17 -1.600107 -0.040221
0 8 18 -0.143906 -0.113953
0 8 19 -0.564375 -0.092663
0 9 12 14.874865 -4.415329
0 9 13 6.525088 -2.724235
0 9 14 3.307268 -2.072524
0 9 15 0.340206 -1.471600
0 9 16 -0.660984 -1.268828
0 9 17 -1.913480 -1.015158
0 9 18 -1.030996 -1.193888
0 9 19 -1.978791 -1.001930
0 10 12 17.163078 -5.401021
0 10 13 5.510263 -2.155933
0 10 14 2.245608 -1.246788
0 10 15 -0.375942 -0.516736
0 10 16 -0.990734 -0.345529
0 10 17 -1.640115 -0.164688
0 10 18 -0.456367 -0.494340
0 10 19 -0.967700 -0.351943
0 11 19 3.500155 2.520249
0 12 19 1.604055 1.301328
0 13 19 0.548415 0.622702
0 14 19 0.161650 0.374068
0 15 19 -0.651831 -0.148885
0 16 19 -0.984570 -0.362788
0 17 19 -1.891111 -0.945564
0 18 19 0.363620 0.503906
1 8 12 5.825833 0.053017
1 8 13 2.618588 0.233707
1 8 14 1.663865 0.287494
1 8 15 0.378161 0.359928
1 8 16 0.168074 0.371764
1 8 17 -0.045398 0.383791
1 8 18 1.581875 0.292114
1 8 19 2.137272 0.260824
1 9 12 14.892076 -4.400577
1 9 13 6.852364 -2.588810
1 9 14 3.867532 -1.916172
1 9 15 1.150662 -1.303920
1 9 16 0.273648 -1.106283
1 9 17 -0.802804 -0.863702
1 9 18 0.081573 -1.062998
1 9 19 -0.326466 -0.971046
1 10 12 17.037905 -5.454668
1 10 13 6.038968 -2.046546
1 10 14 3.081511 -1.130151
1 10 15 0.731972 -0.402125
1 10 16 0.211840 -0.240957
1 10 17 -0.323918 -0.074948
1 10 18 0.792492 -0.420877
1 10 19 0.780995 -0.417315
1 11 19 6.159120 2.271747
1 12 19 3.755586 1.069981
1 13 19 2.389067 0.386721
1 14 19 1.839448 0.111912
1 15 19 0.746932 -0.434347
1 16 19 0.243628 -0.685998
1 17 19 -1.104020 -1.359823
1 18 19 0.815583 -0.400021
2 8 12 6.159580 0.625155
2 8 13 3.327588 0.820465
2 8 14 2.696723 0.863973
2 8 16 1.767431 0.928062
2 8 17 1.913362 0.917998
2 8 18 3.714086 0.793810
2 8 19 5.349039 0.681055
2 9 12 14.536327 -4.705503
2 9 13 6.856746 -2.586998
2 9 14 4.144657 -1.838835
2 9 16 1.019159 -0.976629
2 9 17 0.142512 -0.734795
2 9 18 0.958335 -0.959849
2 9 19 0.914815 -0.947844
2 10 12 16.246544 -5.793822
2 10 13 6.251458 -2.002583
2 10 14 3.717466 -1.041413
2 10 16 1.346950 -0.142252
2 10 17 0.941967 0.011363
2 10 18 1.908446 -0.355233

```
2 10 19 2.213276 -0.470858
2 11 19 8.965322 2.009485
2 12 19 5.807238 0.849373
2 13 19 3.989309 0.181562
2 14 19 3.214732 -0.102976
2 16 19 1.049973 -0.898194
2 17 19 -0.735163 -1.553958
2 18 19 1.062389 -0.893633
3 8 12 7.227665 2.456159
3 8 13 5.498424 2.617019
3 8 15 5.771425 2.591624
3 8 16 6.385846 2.534468
3 8 17 7.470469 2.433573
3 8 18 9.772838 2.219399
3 8 19 14.059045 1.820682
3 9 12 14.593947 -4.656115
3 9 13 7.887228 -2.160591
3 9 15 4.000382 -0.714322
3 9 16 3.531216 -0.539749
3 9 17 3.005999 -0.344320
3 9 18 3.770952 -0.628953
3 9 19 4.450368 -0.881759
3 10 12 15.918084 -5.934590
3 10 13 7.662277 -1.710689
3 10 15 4.369201 -0.025860
3 10 16 4.124604 0.099283
3 10 17 3.901984 0.213182
3 10 18 4.692845 -0.191445
3 10 19 5.477477 -0.592884
3 11 19 13.292461 1.605080
3 12 19 9.278410 0.476128
3 13 19 6.902922 -0.191978
3 15 19 3.814945 -1.060472
3 16 19 2.776483 -1.352539
3 17 19 0.238419 -2.066370
3 18 19 1.474880 -1.718615
4 8 12 8.332136 4.349538
4 8 13 7.689022 4.429927
4 8 14 8.830157 4.287285
4 8 15 9.628200 4.187531
4 8 16 10.739153 4.048661
4 8 17 12.558563 3.821234
4 8 18 15.185226 3.492902
4 8 19 21.145445 2.747874
4 9 12 14.289334 -4.917212
4 9 13 8.373588 -1.959339
4 9 14 6.704069 -1.124580
4 9 15 5.332381 -0.438736
4 9 16 5.017455 -0.281273
4 9 17 4.685502 -0.115296
4 9 18 5.342937 -0.444014
4 9 19 6.154713 -0.849902
4 10 12 15.153903 -6.262097
4 10 13 8.332097 -1.572106
4 10 14 6.911848 -0.595685
4 10 15 5.858862 0.128243
4 10 16 5.701532 0.236408
4 10 17 5.569901 0.326904
4 10 18 6.195310 -0.103065
4 10 19 6.990020 -0.649428
4 11 19 15.517395 1.397142
4 12 19 10.936822 0.297805
4 13 19 8.202072 -0.358536
4 14 19 6.880400 -0.675737
4 15 19 4.605283 -1.221765
4 16 19 3.388892 -1.513699
4 17 19 0.503901 -2.206097
4 18 19 1.588473 -1.945800
5 10 12 16.284454 -5.777575
5 10 13 12.190473 -0.773821
5 10 14 11.518774 0.047142
5 10 15 11.015927 0.661733
5 10 16 10.987820 0.696085
5 10 17 10.987649 0.696295
5 10 18 11.391583 0.202599
5 10 19 12.249565 -0.846047
5 11 19 20.868263 0.897061
5 12 19 15.486426 -0.191400
5 13 19 12.154550 -0.865263
5 14 19 10.384495 -1.223252
5 15 19 7.485503 -1.809565
5 16 19 5.804838 -2.149474
5 17 19 1.897709 -2.939680
5 18 19 2.068175 -2.905204
6 8 12 6.786362 1.699640
6 8 13 5.227234 2.392586
6 8 14 5.343170 2.341059
6 8 15 5.257698 2.379047
6 8 16 5.556811 2.246108
6 8 17 6.023145 2.038848
6 8 18 7.055498 1.580025
6 8 19 8.230036 1.058008
6 9 12 4.035285 -13.706398
6 9 13 -0.505705 -5.633529
6 9 14 -1.719659 -3.475388
6 9 15 -2.514555 -2.062240
6 9 16 -2.755911 -1.633162
6 9 17 -3.018758 -1.165877
6 9 18 -2.880602 -1.411489
6 9 19 -3.099199 -1.022872
6 10 12 4.553358 -10.805187
6 10 13 1.365811 -3.013406
6 10 14 0.729639 -1.458319
6 10 15 0.315197 -0.445239
6 10 16 0.230948 -0.239295
6 10 17 0.150482 -0.042602
6 10 18 0.316681 -0.448866
6 10 19 0.296362 -0.399198
6 11 19 11.911308 1.734160
6 12 19 6.618943 0.762093
6 13 19 3.676570 0.221657
6 14 19 2.509262 0.007253
6 15 19 0.442858 -0.372291
6 16 19 -0.376549 -0.522794
6 17 19 -2.094756 -0.838383
6 18 19 0.771534 -0.311921
7 12 19 -0.024978 1.476493
7 13 19 -2.657236 1.033683
7 14 19 -3.332681 0.920057
7 15 19 -4.734295 0.684271
7 16 19 -4.874054 0.660760
7 17 19 -4.926397 0.651955
7 18 19 -0.115093 1.461333
8 12 19 6.261858 0.800489
8 12 20 5.179852 -1.054378
8 12 21 5.597861 -0.337791
```

```
8 13 19 2.747893 0.340718
8 13 20 1.240115 -0.907098
8 13 21 2.109331 -0.187747
8 14 19 1.456488 0.171749
8 14 20 -0.368691 -0.846956
8 14 21 0.905184 -0.135956
8 15 19 -0.785530 -0.121599
8 15 20 -2.358693 -0.772563
8 15 21 -0.657768 -0.068732
8 16 19 -1.530628 -0.219089
8 16 20 -3.047804 -0.746802
8 16 21 -1.049867 -0.051867
8 17 19 -2.924821 -0.401506
8 17 20 -4.053106 -0.709221
8 17 21 -1.562039 -0.029838
8 18 19 0.586607 0.057933
8 18 20 -2.862895 -0.753715
8 18 21 -0.060835 -0.094406
8 19 21 -0.449828 -0.077676
9 12 19 19.321659 -0.603791
9 12 20 15.256504 -4.088209
9 12 21 15.526484 -3.856797
9 13 19 11.286537 -0.753980
9 13 20 6.421521 -2.767090
9 13 21 7.235295 -2.430356
9 14 19 7.798424 -0.819178
9 14 20 2.773247 -2.221554
9 14 21 4.012098 -1.875828
9 15 19 3.066041 -0.907634
9 15 20 -0.743135 -1.695740
9 15 21 0.940555 -1.347390
9 16 19 1.217167 -0.942193
9 16 20 -2.018851 -1.504978
9 16 21 -0.091631 -1.169810
9 17 19 -1.790696 -0.998414
9 17 20 -3.681873 -1.256302
9 17 21 -1.398626 -0.944950
9 18 19 1.087878 -0.944609
9 18 20 -2.748216 -1.395914
9 18 21 -0.390001 -1.118477
9 19 21 -1.156742 -0.986565
10 13 20 5.008315 -2.259785
10 13 21 6.255579 -2.001730
10 14 19 6.624592 -0.635767
10 14 20 0.956391 -1.426679
10 14 21 2.760515 -1.174941
10 15 19 0.637123 -0.411937
10 15 20 -2.433840 -0.729622
10 15 21 -0.129615 -0.491254
10 16 19 -1.035398 -0.349412
10 16 20 -3.317770 -0.547880
10 16 21 -0.811515 -0.329944
10 17 19 -3.175830 -0.269396
10 17 20 -4.299568 -0.346015
10 17 21 -1.539187 -0.157807
10 18 19 0.825054 -0.418962
10 18 20 -2.883714 -0.637125
10 18 21 -0.184240 -0.478332
10 19 21 -0.671752 -0.363007
11 19 21 1.701521 2.688346
12 19 21 0.698466 1.398703
13 19 21 0.136031 0.675572
14 19 21 -0.070286 0.410308
15 19 21 -0.525298 -0.174708
16 19 21 -0.724760 -0.431159
17 19 21 -1.347417 -1.231719
18 19 21 0.222317 0.786512
```