Evaluating the Impact of the Number of Access Points in Mobile Robots Localization Using Artificial Neural Networks

Gustavo Pessin, Fernando S. Osório, Jó Ueyama, Jefferson R. Souza, Denis F. Wolf
Institute of Mathematics and Computer Science (ICMC)
University of São Paulo (USP) – São Carlos, SP, Brazil
{pessin, fosorio, joueyama, jrsouza, denis}@icmc.usp.br

Torsten Braun
Institute of Computer Science and Applied Mathematics
University of Bern – Bern, Switzerland
braun@iam.unibe.ch

Patrícia A. Vargas
School of Maths and Computer Science (MACS)
Heriot-Watt University – Edinburgh, Scotland
p.a.vargas@hw.ac.uk

ABSTRACT

Localization is information of fundamental importance to carry out various tasks in the mobile robotic area. The exact degree of precision required in the localization depends on the nature of the task. The GPS provides global position estimation but is restricted to outdoor environments and has an inherent imprecision of a few meters. In indoor spaces, other sensors like lasers and cameras are commonly used for position estimation, but these require landmarks (or maps) in the environment and a fair amount of computation to process complex algorithms. These sensors also have a limited field of vision. Currently, Wireless Networks (WN) are widely available in indoor environments and can allow efficient global localization that requires relatively low computing resources. However, the inherent instability in the wireless signal prevents it from being used for very accurate position estimation. The growth in the number of Access Points (AP) increases the overlap signals areas and this could be a useful means of improving the precision of the localization. In this paper we evaluate the impact of the number of Access Points in mobile nodes localization using Artificial Neural Networks (ANN). We use three to eight APs as a source signal and show how the ANNs learn and generalize the data. Added to this, we evaluate the robustness of the ANNs and evaluate a heuristic to try to decrease the error in the localization. In order to validate our approach several ANNs topologies have been evaluated in experimental tests that were conducted with a mobile node in an indoor space.

COMSWARE 2011, July 04-07, Verona, Italy

DOI 10.4108/comsware.2011.1

1. INTRODUCTION

The ability to estimate position correctly is a prerequisite to undertake a number of tasks in the autonomous mobile robotic area. Moreover, knowledge about localization can be used to track animals and people (e.g. to track the movement of people while practicing sports). Sensors like GPS can be used to provide global position estimation but this is restricted to outdoor environments and has an inherent imprecision of a few meters. While the use of GPS is quite common outdoors as a primary source for locating a position, a more accurate estimation can be obtained through a fusion of other sensors, like lasers and cameras [1].

In indoor spaces, sensors like lasers and cameras can be used for pose estimation [12, 16], but they require landmarks (or maps) in the environment and a fair amount of computation to process complex algorithms. These sensors also have a limited field of vision, which makes the task of localization harder. In the case of video cameras, the variation of light is also a serious issue. Another commonly used sensor is the encoder, which provides odometry. Odometry is a useful source of information in some cases [2, 14] but it has an incremental error that usually invalidates its use in real systems.

Wireless Networks are widely available in indoor environments and allow efficient global localization, while requiring relatively low computing resources. Other advantages of this approach are scalability, robustness, and independence of specific features of the environment. However, the inherent instability of the wireless signal does not allow it to be used directly for accurate position estimation. One machine learning technique that could reduce the instability of the signals of the WN is the Artificial Neural Networks; this is due to its capacity to learn from examples, as well as the generalization and adaptation of the outputs. It is a method that is widely used in applications that require approximation, prediction or classification [15].

The main objective of this paper is to make an evaluation of the ability of ANNs to obtain the position of mobile nodes by means of measurements from wireless devices (802.11b/g). The measurement from the wireless network is the Received Signal Strength Indication (RSSI). This value is used as the input of an ANN to learn the localization without any another information or any need to adopt a mathematical approach. Several topologies of ANNs are evaluated as well as the impact of the number of Access Points (APs) in mobile nodes localization by using Artificial Neural Networks. We use three to eight APs and show how the ANNs learn and generalize the data. In addition, we evaluate the robustness of the ANNs and broadening an idea outlined in [20], which aims to reduce the error in the localization by calculating the average of multiple measurements from the wireless network.

This paper has the following structure: Section 2 introduces a short theoretical description and discusses the applications of artificial neural networks and wireless networks. Section 3 outlines the methodology that is employed to set up and evaluate the experiments. Section 4 describes the evaluation of all the experiments that are carried out. The final section makes some concluding comments and examines the future perspectives of this area of research.

2. THEORETICAL BACKGROUND

2.1 Artificial Neural Networks

An Artificial Neural Network is a collection of units (neurons) connected by weighted links (synapses). Input and output units receive and transfer signals from and to the environment. The internal units are described as hidden because they do not have any contact with the external environment [17]. The basic attributes of an ANN can be divided into architecture and neurodynamics. The architecture determines the structure of the network, i.e. the number of neurons and their interconnectivity. Neurodynamics, in turn, defines the functional properties of the network, that is how it learns, recovers, combines and compares new information with knowledge that has already been stored [8]. In mathematical terms, ANNs are universal approximators, that carry out mappings in multivariable functions spaces [6]. The ability to learn and generalize[1] is one of the main advantages of ANNs, since it gives them a power that extends far beyond the simple direct mapping of inputs and outputs.

Neural networks are often used in applications that require approximation, prediction or classification. The study [19] examines an ANN that is able to perform the navigation of a robot in a simulated two-dimensional environment. The ANN controls the direction of the robot to areas with a lower density of occupation by vegetation; the inputs of ANN are the vegetation densities observed and the output is the angle at which the agent should move. In [18] there is an ANN which performs the navigation of a robot in a simulated three-dimensional environment; the ANN inputs consist of information collected from sensors (localization, orientation, distance to obstacles) while the outputs are the speed and angle which have to be applied to the linear and angular motors, respectively. In addition, the study [22] discusses an ANN that is used to classify navigable and non-navigable regions in images; the ANN inputs are attributes of color, formed by the average of color channels and entropy. Other studies using ANNs can be seen in [15, 25].

In this paper, the development of the ANN was undertaken with the aid of the Stuttgart Neural Network Simulator (SNNS) [24]. The SNNS is an environment created to develop topologies and to train ANNs which have a large number of learning algorithms, such as backpropagation, quick propagation, and resilient backpropagation, among others. The core system is developed in C and it can be operated entirely through command line, although it also has an interface developed in JAVA (JavaNNS). An application package from SNNS, (the SNNS2C), allows the ANN to be converted into a C code, which can be easily inserted into another application.

2.2 Wireless Networks

In most cases, a signal from a wireless network is propagated in a radial range. It can be directional depending on the type of antenna being used. The signal power decreases according to the distance from the server station. By means of trilateration, and by using at least 3 server stations, we can do a simple calculation to obtain the localization, in a similar way to the GPS system. However, unlike GPS, the signal from the wireless network shows a greater degree of instability and suffers more from interference [5, 7, 21].

In [3] it has been shown that obtaining an absolute performance in localization, by means of a wireless network, depends on environmental configuration. This means that different approaches are required for different environments, such as using different kinds of signals and filters. Evaluations in large indoor areas (like a building) presents more difficulties in localization due to the problem of attenuation and reflection of the signals on the walls and the different sources of interferences. The use of wireless localization to address the problem of localization inside a building can be seen in [4, 9, 10, 11].

Another approach for localization is the use of a Wireless Sensor Network (WSN); the main difference in these approaches is that in a WSN there is a large number of small sensors that pick up information from the environment. The information acquired by the sensors can be regarded as a fingerprint. This is an interesting solution, but it requires a lot of resources which could make the system expensive. Examples of work involving the use of WSNs to obtain localization can be found in [13, 23, 26].

3. METHODOLOGY

We have evaluated the use of an Artificial Neural Network to obtain the position of a mobile node in an indoor environment using data provided by a wireless network (802.11b/g). Our approach relies on the ANN learning and generalization capabilities in an attempt to reduce the effect of unstable data (due to signal strength oscillation), and increase the accuracy of the position estimation of the node.

The indoor environment used to obtain data can be seen in

[1] *Generalize* can be considered as the production of acceptable outputs for inputs not presented during learning.

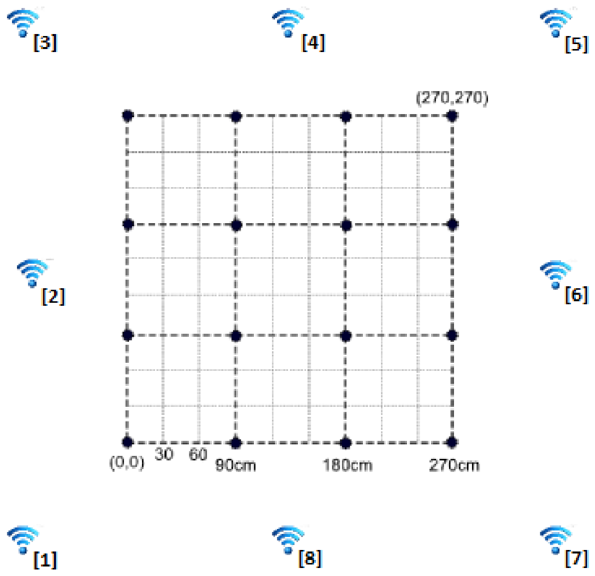


Figure 1: Graphical representation of the working area. It represents an area of 270cm x 270cm.

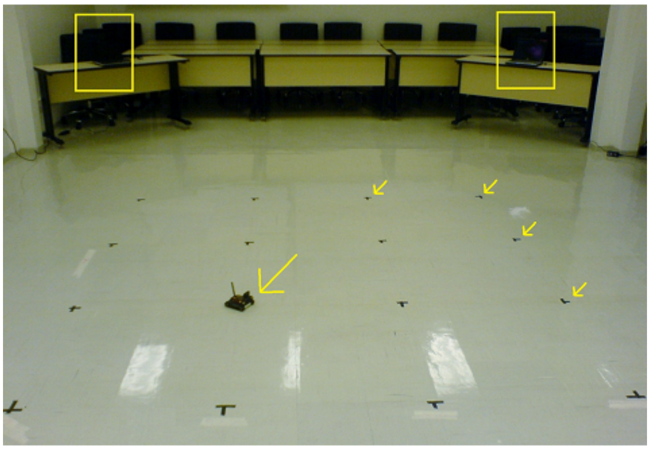


Figure 2: A section of the picture of the working area with the robot, similar to what is represented in Figure 1. The yellow rectangles show two sources of network signals (APs). The large arrow points to the robot. The small arrows show the plane marks (each 90cm long).

Fig. 1 and 2. The working area[2] of the mobile node is inside a room and is represented as a Cartesian plane. There are 8 access points (APs), as shown in Fig. 1. Fig. 2 only shows a partially mounted environment. The mobile node is located inside the plane with one wireless card which is used to scan the networks and signals provided by the APs. The data used to train the ANN was collected in 15 readings, each marked point – with a displacement of 90cm (Fig. 2) mapping out a plane of 270cm x 270cm; this means there were 16 points to read, resulting in 240 readings altogether.

Several ANN topologies have been evaluated to validate our approach by means of experimental tests conducted with a mobile node in this indoor environment. The inputs of the ANN are the signals received by the mobile node antenna from the 3, 4, 6 and 8 static positioned APs. The value obtained from the wireless networks is the Received Signal Strength Indication (RSSI). This value is obtained with the aid of the GNU/Linux command *iwlist*[3]. As we use the *iwlist* command, there is no need to establish a connection (or login) with the different specific networks. The scan of the networks, without a connection, provides enough information for this evaluation. Without a connection, the system becomes easier to use, more lightweight and flexible.

RSSI is a metric of the signal strength present in a received radio signal. The main advantage of using RSSI is its low cost. Since every wireless device implements the possibility to deliver this value in its circuitry, there is no need for the development or adaptation of any additional hardware [5, 21].

[2]The Fig. 2 shows a little robot inside the plane; however, a mobile computer was used to scan the WNs and to obtain the data used as the ANN input. The GNU/Linux command *iwlist* used to scan the networks has not yet been implemented in the robot.

[3]Used as *iwlist <interface> scanning*

The experiments are conducted in four steps: (i) We evaluate several ANN topologies seeking to obtain the topology which can achieve the lowest error; (ii) We use the best ANN (best hidden layer) obtained in the previous stage and analyze the behavior of the ANN when more than 3 inputs are used (we use 3, 4, 6 and 8 inputs – respectively 3, 4, 6 and 8 APs providing RSSI values); (iii) We evaluate how the ANN deals with errors in the inputs – we simulate failures in the APs, by injecting null values in some of the inputs; and (iv) We evaluate an idea to try to decrease the error in the localization by using averages and medians of multiple measurements from the wireless network, broadening the idea outlined in [20].

The number of neurons in the input layer is equivalent to the number of APs available in each evaluation. As we use 3 to 8 APs, the inputs of the ANN use one neuron for each network signal. The order is important, and hence, AP 1 was linked to neuron 1, AP 2 with neuron 2 and so on. The outputs of the network are two values, the coordinates (x, y) of the receiving antenna in the plane, a.k.a. the mobile node position. We trained the ANN with the power signal of each source antenna in the expectation of obtaining the position of the mobile node in the Cartesian plane. As a result, after training the ANN, we could use it to obtain the localization and to track the displacement of the mobile node along a path.

The error is measured in centimeters, using the distance formula (distance between two points), as shown Eq. 1. The value d is the error (distance, in centimeters), (x_1, y_1) are the expected value from ANN validation set and (x_2, y_2) are the obtained value when using the ANN.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{1}$$

Table 1: Results of the best ANN for each topology (in cms).

	ANN Topology									
	3x4x2	3x4x4x2	3x8x2	3x8x8x2	3x12x2	3x16x2	3x20x2	3x24x2	3x28x2	3x32x2
Average error	120.37	127.49	123.78	120.04	122.10	120.94	120.44	121.31	122.36	122.25
Std. dev.	59.63	51.89	54.59	58.38	54.07	58.12	59.08	53.28	52.84	53.80
Larger error	220.30	221.66	204.67	213.06	238.31	246.33	220.76	208.89	194.53	203.86
Smaller error	2.77	28.30	12.22	0.76	18.46	0.98	6.32	4.39	18.21	22.71

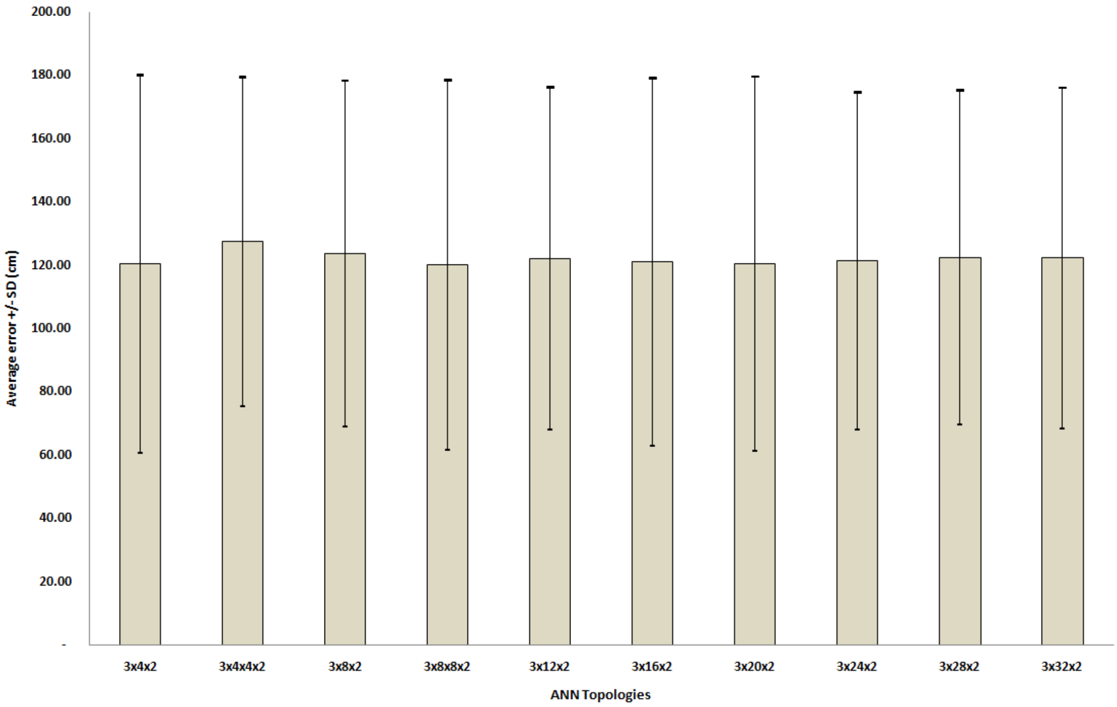


Figure 3: Graphical representation of values from Table 1 – results of the best ANN for each topology.

4. EXPERIMENTS AND RESULTS

In this section we describe the four experiments conducted for this article[4]. In the first, we evaluated several ANN topologies seeking to obtain the topology which could achieve the lowest error. In the second, we used the best ANN (best hidden layer) obtained in the previous stage and analyzed the ANN behavior when more than 3 inputs were used (we used 3, 4, 6 and 8 inputs – respectively 3, 4, 6 and 8 APs providing RSSI values). In the third stage, we evaluated how the ANN deals with error in the inputs – we simulated failures in the APs, by injecting null values in some of the inputs. In the fourth, we evaluated an idea for reducing the error in the localization by using averages and medians of multiple measurements from the wireless network.

With the aim of obtaining the topology which can achieve the lowest error, we have evaluated the impact of using 10 different hidden layers in the ANN. The input of the ANN takes account of three RSSI values from 3 different access points (we use values from APs {3,5,8} – Figure 1). The output are the expected (x, y). The hidden layers are with {4, 4x4, 8, 8x8, 12, 16, 20, 24, 28, 32}. Table 1 shows the best ANN for each topology with the results converted to centimeters. We ran the ANN training until it reached 200.000 cycles and used a script to analyze the results and obtain the training cycle with the Optimum Generalization Point (OGP), in which the ANN has the best generalization capabilities. As neural networks are susceptible to random values used in the initialization of the weights, we ran each ANN 5 times with different random seeds.

As can be seen, from Table 1 and Figure 3 the results for all the evaluated ANN topologies are fairly similar. In this way, there is no significant difference in using any of the hidden layers evaluated. Although quite similar, the ANN which had the lowest error (taking into account the sum of the average errors and standard deviation) was the ANN with 24 neurons in the hidden layers. Because of this, we will use this ANN topology in the next stage.

To analyze the ANN behavior when more than three access points are included, we use 4, 6 and 8 RSSI values from different access points. When using four inputs, we use values from APs {1,3,5,7} and when using six inputs, we use values from APs {1,2,3,5,6,7}. These choices were made in an attempt to maintain a balance on all sides of the working area. When using eight inputs, we used all the available APs. We trained these ANNs by means of the same methodology employed for the previous set, but with different input layers being considered. Table 2 shows the best ANN for each topology with the results converted to centimeters. Figure 4 shows the graphical representation of Table 2.

As can be seen from Table 2 and Figure 4, when the number of access points is increased, the average error decreases considerably. The average error falls from 121.31cm (using 3 inputs) to 90.46cm (using 8 inputs). But the standard deviation remains very similar in all cases.

Figure 5 shows the histogram of the error when account is taken of the different input layers. Using 3 (Figure 5(a)), 4 (Figure 5(b)), 6 (Figure 5(c)) and 8 (Figure 5(d)) inputs, it is clear that Figures 5(c) and 5(d) shows the errors are more concentrated at the beginning of the histogram than is the case with Figures 5(a) and 5(b). In addition, it can be seen in Figure 5(d) that there is a lower tail in relation to Figure 5(c), which means there are fewer results with a large error. Moreover, Figure 5(d) shows the errors are more concentrated at the beginning of the histogram than is the case with Figure 5(c), which means there is a lower error in Figure 5(d).

[4]Scripts, ANN topologies and data files are available in http://goo.gl/sjgnd

Table 2: Results of the best ANN for each different input layer (in cms).

	ANN Topologies			
	3x24x2	4x24x2	6x24x2	8x24x2
Average error	121.31	114.37	97.58	90.46
Std. dev.	53.28	49.55	59.92	52.16
Larger error	208.89	225.36	296.22	227.48
Smaller error	4.39	15.45	16.05	4.43

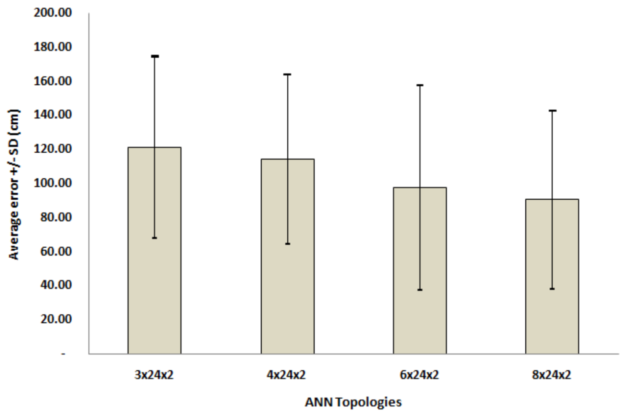
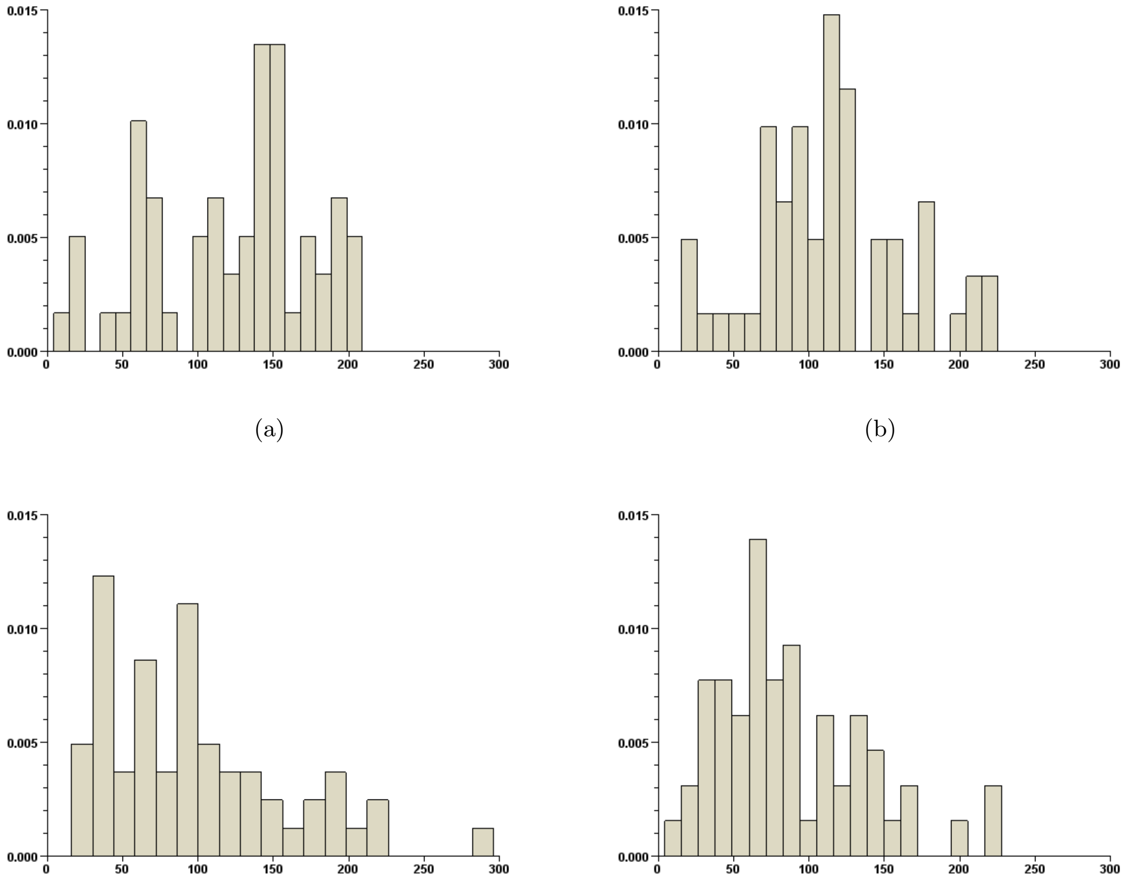


Figure 4: Graphical representation of values from Table 2 – results of the best ANN for each different input layer.

In the third stage, we sought to evaluate how the ANN deals with the problem of errors in the inputs – we simulated failures in the APs, by injecting null values in some of the inputs. In this way, we pick the best ANN from the previous stages (ANN with 8 inputs and 24 neurons in the hidden layer). We used the trained ANN to read the original validation set and obtain the output values for all of the sets. For each complete reading in the validation set, we put a null value in one input neuron (keeping the true values of the other seven). This meant that we ran the ANN eight times, one for each AP, and in this way obtained the difference between the expected and obtained values.

Table 3 shows the results of the best ANN using the original inputs and using the simulated failure inputs. In the light of this data set, it is evident that the ANN with simulated input error really produces worse results. Figure 6 shows the histogram of the error taking account of the ANN with original 8 inputs and with simulated input error. It can be seen in Figure 6(b) that it has a big tail, and obtains really worse errors than Figure 6(a).



(a) (b) (c) (d)

Figure 5: Histogram of the error considering different input layers. (a) Using 3 inputs. (b) Using 4 inputs. (c) Using 6 inputs. (d) Using 8 inputs. The x axis represents the error in centimeters.

Table 3: Comparison between ANN using the original 8 inputs and using simulated failure inputs (in cms).

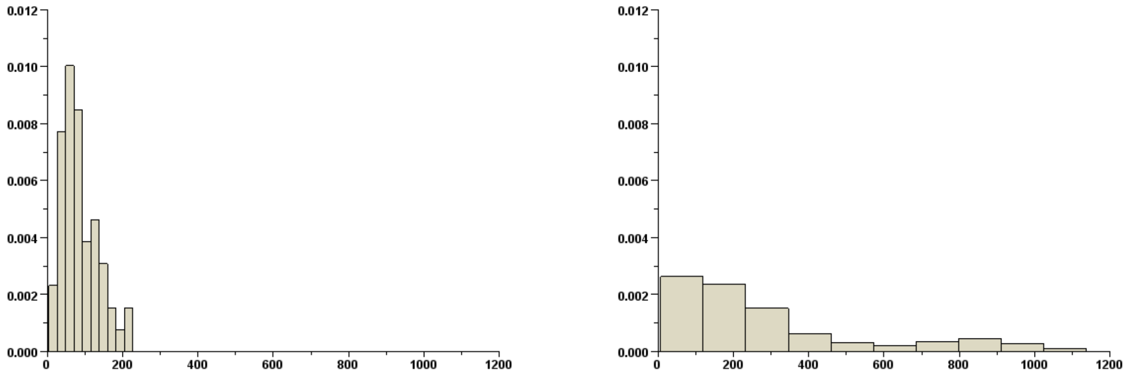
	Original 8 inputs	Injecting error
Average error	90.46	286.37
Std. dev.	52.16	269.37
Larger error	227.48	1137.05
Smaller error	4.43	6.73

Table 4: Results of the best ANNs trained with average and median of multiple scans (in cms).

	ANN Inputs			
	Avg of 4 scans	Mdn of 4 scans	Avg of 8 scans	Mdn of 8 scans
Average error	45.92	43.19	34.69	27.40
Std. dev.	26.35	33.08	24.72	17.98
Larger error	116.04	150.04	132.32	91.79
Smaller error	8.70	1.81	7.08	4.92

In the fourth stage, we sought to evaluate a simple idea to reduce the error; to achieve this, we made a comparison between the use of average and the use of medians of multiple readings of the wireless network signals as ANN input. Hence, the average and medians of multiples readings of wireless networks were used as input of the ANN. We evaluated the average and medians of 4 and 8 readings of the wireless network signals (scans of wireless signals). In this evaluation, we used the best ANN topology chosen from the second stage (ANN with 24 hidden neurons where all eight APs were used). We also ran the training and validation of the ANN 5 times using different random seeds. Table 4 and Fig. 7 show the results, in centimeters, of the best ANNs.

It can be seen in Table 4 and in Fig. 7 that by using the average and the median of the multiple readings from the wireless network as the input in the ANN, we could reduce the average error in the ANN learning from 90.46cm (without using the average of the scan) to 34.69cm (using an average of 8 scans) and to 27.40cm (using a median of 8 scans). We



(a)

(b)

Figure 6: Histogram of the error. (a) Using original 8 inputs. (b) Injecting error (null value) in one neuron in the ANN input. The x axis represents the error in centimeters.

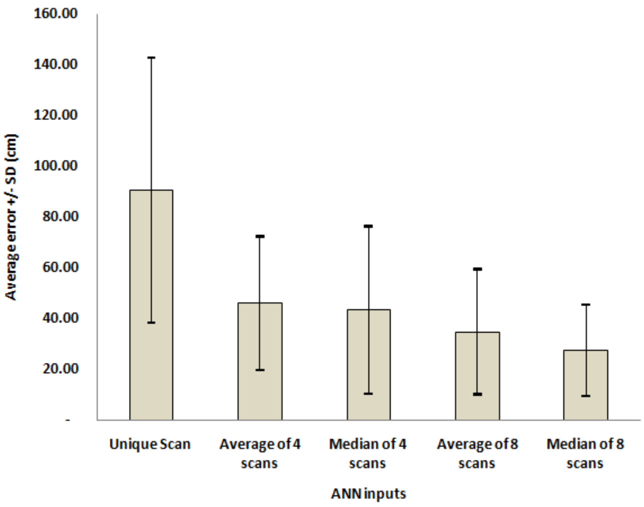


Figure 7: Results of the best ANNs trained with average and median of multiple scans.

can see, from these tables and figures, that the use of the median provides better values than the use of the average. When the median of 8 readings is considered, there is a reduction of 69.7% in the average error compared with the situation when multiple scans were not used. In addition, it was possible to reduce the standard deviation from 52.16cm (without using the average of the scan) to 17.98cm (using a median of 8 scans). This means there was a reduction of 67.9% in the standard deviation.

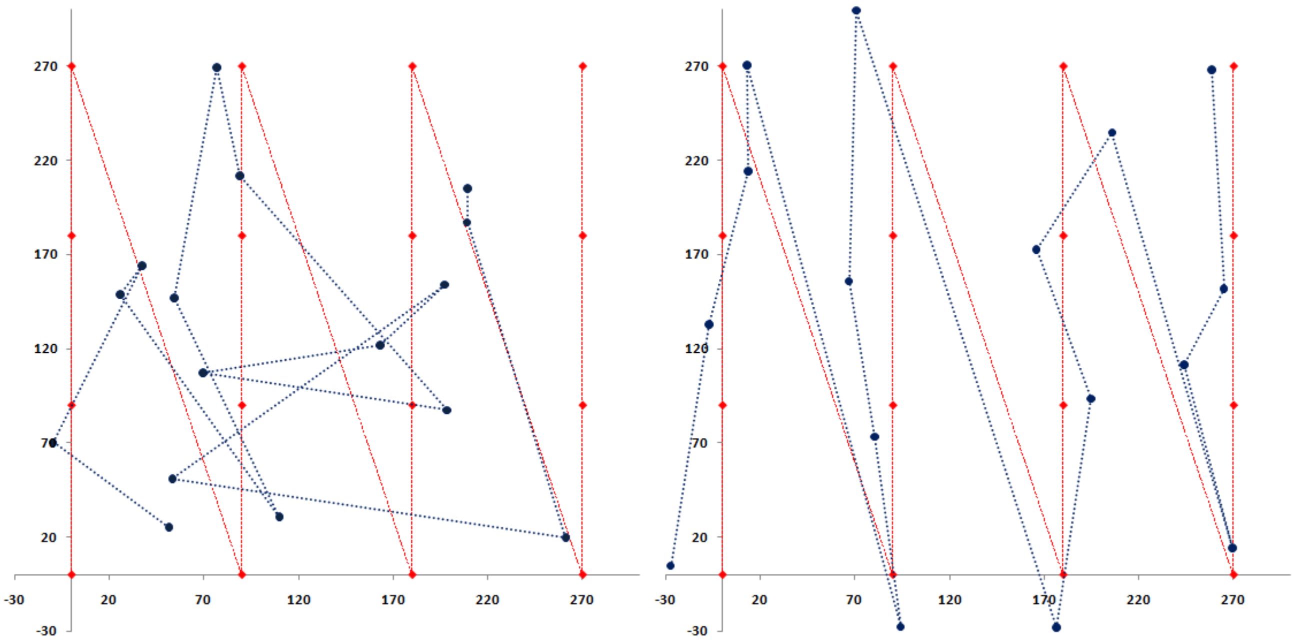
In the final stage, the mobile node was used to traverse a path (and find the track) in the environment, taking account of wireless scanning (each with a 90cm x 90cm displacement). The paths can be seen in Fig. 8. Fig. 8(b) shows results that are significantly better than those in Fig. 8(a). We can see from Fig. 8 that the tracked path improves with the use of the median.

The Fig. 9 shows the partial spatial distribution of the data used as ANN input. It shows four sets, that take account of the 3 APs. The axis x, y, and z are in dBm (RSSI unit of measure).

5. CONCLUSION

Accurate position information is a requirement to accomplish several tasks in the mobile robotic area. Some sensors like GPS provide global position estimation but it is restricted to outdoor environments and has an inherent imprecision of a few meters. In indoor spaces, other sensors like lasers and cameras can be used for pose estimation, but they require landmarks (or maps) in the environment and a fair amount of computation to process complex algorithms. These sensors also have a limited field of vision, which makes the localization task harder. In the case of video cameras, the variation of light is also a serious issue. Wireless Networks (WN) are widely available in indoor environments and allow an efficient global localization that requires relatively low computing resources. Other advantages of this approach are scalability, robustness, and the independence of specific features of the environment. However, the inherent instability of the wireless signal does not allow its direct use for accurate position estimation.

In this paper we have evaluated the use of an Artificial Neural Network to obtain the position of a mobile node in an indoor environment by making use of the data provided by the wireless network. We evaluated several topologies of ANNs. The results from ANN learning show that regardless of the number of neurons in the hidden layer, they remain very similar. But, if a larger number of APs was used, the error could be reduced from 121.31cm (using 3 APs) to 90.46cm (using 8 APs). In addition, when an error was injected in the ANN, the results became really worse. This fact might discourage the use of the ANN since a failure in some AP might be identified. To further reduce the error, it has been shown that the use of multiple readings and the use of median instead of average can reduce the error from 90.46cm to 27.40cm.



(a) (b)

Figure 8: Paths using the best trained ANNs. (a) Just one scan in the ANN input. (b) Median of 8 scan in the ANN input. Red line: original path. Blue line: ANN tracking. The x and y axis are in centimeters.

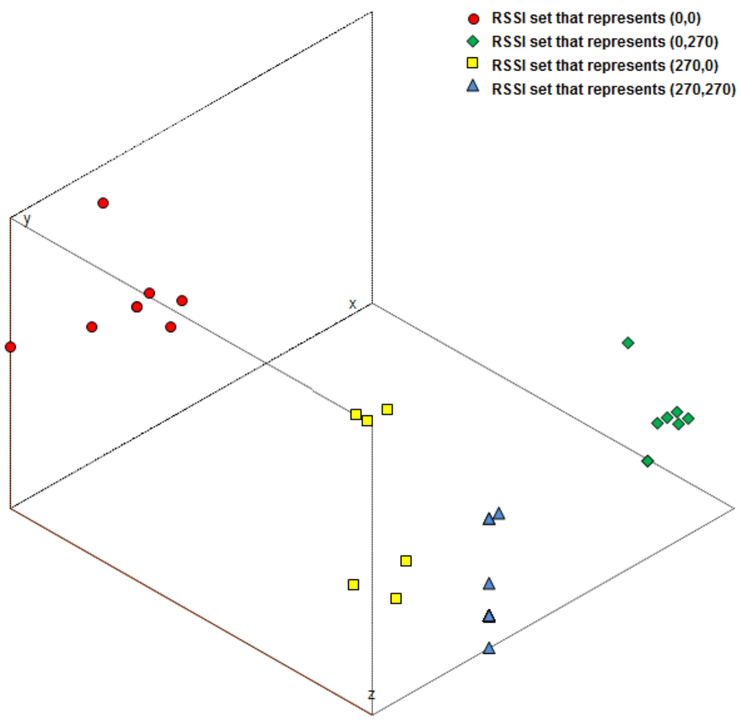


Figure 9: Partial spatial distribution of the data used as ANN input. It shows four sets, that take account of the 3 APs. The axis x, y, and z are in dBm (RSSI unit of measure).

6. FUTURE WORK

There are two major research projects being planned: (i) Seeking improvements in the system to obtain the orientation of the mobile node and (ii) Making evaluations in large buildings. The first step in making these evaluations is to study how to reduce the signal fluctuation. Another future work planned is to carry out an evaluation of this approach in multi-node localization.

7. ACKNOWLEDGMENTS

The authors would like to acknowledge the financial support granted by FAPESP, process ID 2008/05346-4 and CNPq process ID 483699/2009-8. We would also like to acknowledge the financial support from CNPq and FAPESP to the INCT-SEC (National Institute of Science and Technology – Critical Embedded Systems – Brazil), processes ID 573963/2008-8 and 08/57870-9. Finally we would also like to acknowledge CAPES and FAPESP for their financial support for this research (doctoral grant) and thank Diego F. Sciotti for his help in setting up the test environment.

8. REFERENCES

[1] M. Buehler, K. Iagnemma, and S. Singh. *The 2005 DARPA Grand Challenge: The Great Robot Race*. Springer, 2007.

[2] K. S. Chong and L. Kleeman. Accurate odometry and error modelling for a mobile robot. In *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, volume 4, pages 2783 –2788 vol.4, Apr. 1997.

[3] E. Elnahrawy, X. Li, and R. Martin. The limits of localization using signal strength: a comparative study. In *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on*, pages 406 – 414, 2004.

[4] P. Espinace, A. Soto, and M. Torres-Torriti. Real-time robot localization in indoor environments using structural information. *Latin American Robotics Symposium and Intelligent Robotics Meeting*, 2008.

[5] M. S. Gast. *802.11 Wireless Networks: The Definitive Guide*. O'Reilly, 2002.

[6] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989.

[7] M. Jacobsson, I. Niemegeers, and S. H. de Groot. *Personal Networks: Wireless Networking for Personal Devices*. Wiley, 2010.

[8] S. V. Kartalopoulos. *Understanding Neural Networks and Fuzzy Logic: Basic Concepts and Applications*. Wiley, 1995.

[9] A. Ladd, K. Bekris, G. Marceau, A. Rudys, D. Wallach, and L. Kavraki. Using wireless ethernet for localization. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, 2002.

[10] A. Ladd, K. Bekris, A. Rudys, D. Wallach, and L. Kavraki. On the feasibility of using wireless ethernet for indoor localization. *Robotics and Automation, IEEE Trans. on*, 20(3):555 – 559, 2004.

[11] A. M. Ladd, K. E. Bekris, A. Rudys, G. Marceau, L. E. Kavraki, and D. S. Wallach. Robotics-based location sensing using wireless ethernet. In *Proc. of the 8th annual int. conf. on Mobile computing and networking*, MobiCom '02, pages 227–238. ACM, 2002.

[12] H. Lang, Y. Wang, and C. de Silva. Mobile robot localization and object pose estimation using optical encoder, vision and laser sensors. In *Automation and Logistics, 2008. ICAL 2008. IEEE International Conference on*, pages 617 –622, 2008.

[13] D. M. Liu Zhen and W. Qi. Localization and tracking algorithm for mobile robot based on wireless sensor network. *Kybernetes*, 39:1040–1049, 2010.

[14] A. Martinelli. The odometry error of a mobile robot with a synchronous drive system. *Robotics and Automation, IEEE Trans. on*, 18(3):399–405, 2002.

[15] T. M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[16] A. Napier, G. Sibley, and P. Newman. Real-time bounded-error pose estimation for road vehicles using vision. In *Intelligent Transportation Systems (ITSC), 2010 13th Int. IEEE Conf. on*, pages 1141–1146, 2010.

[17] S. Nolfi and D. Floreano. *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. The MIT Press, 2000.

[18] G. Pessin, F. S. Osório, A. Y. Hata, and D. F. Wolf. Intelligent control and evolutionary strategies applied to multirobotic systems. In *Proceedings of IEEE-ICIT 2010 International Conference on Industrial Technology*, pages 1427–1432, Viña del Mar, Valparaíso, Chile, 2010. Los Alamitos, California, USA: IEEE Computer Society.

[19] G. Pessin, F. S. Osório, S. R. Musse, V. Nonnenmacher, and S. S. Ferreira. Utilizando agentes autônomos com aprendizado para a identificação e combate de incêndios em Áreas florestais. In *Anais do VII Simpósio de Informática do Planalto Médio (SIPM)*, 2007.

[20] G. Pessin, F. S. Osório, J. Ueyama, D. F. Wolf, and T. Braun. Mobile robot indoor localization using artificial neural networks and wireless networks. In *Proc. of First Brazilian Conference on Critical Embedded Systems (I CBSEC)*, pages 89–94, 2011.

[21] J. Ross. *The Book of Wireless: A Painless Guide to Wi-Fi and Broadband Wireless*. No Starch Press, 2008.

[22] D. Sales, P. Shinzato, G. Pessin, F. Osório, and D. Wolf. Vision-based autonomous navigation system using ann and fsm control. In *7th Latin American Robotics Symposium (LARS)*, pages 1–6, 2010.

[23] M. Vieira, R. Govindan, and G. Sukhatme. Scalable and practical pursuit-evasion with networked robots. *Intelligent Service Robotics*, 2:247–263, 2009.

[24] A. Zell, G. Mamier, M. Mache, R. Hübner, S. Herrmann, T. Soyez, M. Schmalzl, T. Sommer, A. Hatzigeorgiou, D. Posselt, T. Schreiner, B. Kett, G. Clemente, and J. Wieland. Stuttgart neural network simulator - user manual, version 4.1 (report no. 6/95). Technical report, University of Stuttgart, 1995.

[25] G. P. Zhang. Neural networks for classification: a survey. *IEEE Transactions on Systems, Man, and Cybernetics*, 30:451–462, 2000.

[26] Y. Zhong, X. Kuang, and P. Huang. Improved algorithm for distributed localization in wireless sensor networks. *Journal of Shanghai Jiaotong University (Science)*, 15:64–69, 2010.