

Noise Maps for Acoustically Sensitive Navigation¹

Eric Martinson, Ronald Arkin
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332

ABSTRACT

More and more robotic applications are equipping robots with microphones to improve the sensory information available to them. However, in most applications the auditory task is very low-level, only processing data and providing auditory event information to higher-level navigation routines. If the robot, and therefore the microphone, ends up in a bad acoustic location, then the results from that sensor will remain noisy and potentially useless for accomplishing the required task. To solve this problem, there are at least two possible solutions. The first is to provide bigger and more complex filters, which is the traditional signal processing approach. An alternative solution is to move the robot in concert with providing better audition. In this work, the second approach is followed by introducing noise maps as a tool for acoustically sensitive navigation. A noise map is a guide to noise in the environment, pinpointing locations which would most likely interfere with auditory sensing. A traditional noise map, in an acoustic sense, is a graphical display of the average sound pressure level at any given location. An area with high sound pressure level corresponds to high ambient noise that could interfere with an auditory application. Such maps can be either created by hand, or by allowing the robot to first explore the environment. Converted into a potential field, a noise map then becomes a useful tool for reducing the interference from ambient noise. Preliminary results with a real robot on the creation and use of noise maps are presented.

Keywords: Mobile Robots, Acoustics, Noise Maps.

1. INTRODUCTION

In many robotic applications, including Human-Robot Interaction [1], Intruder Detection [2], and Sniper Localization [3], robots are being equipped with microphones. In some cases, entire teams of robots [4], are released into the world to accomplish some task requiring audition. The problem with most applications is that the auditory task is at a very low level, only processing data and providing auditory event information to high-level navigation routines. If the robot, and therefore the microphone, ends up in a bad acoustic location, then the results from that sensor will remain noisy and useless for accomplishing the task required of the robot.

To solve this problem of microphones placed in a bad location, there are two possible methodologies. The first is to add bigger and/or more complex filters to the incoming audio signal. The primary advantage of this approach is that there is a large body of DSP work to build upon. If it is known a priori what sounds the microphone needs to filter out of the incoming signal, then a filter can usually be designed to solve it. The disadvantages of this approach largely stem from its implementation on a mobile robot. Filters are usually created with a specific location or type of noise in mind, but the robot's mobility may move the robot into a location that it was not designed to handle. Although we can generate filters to handle more than just one type of noise, greater complexity requires more processing and perhaps the use of dedicated processing equipment, which may be an excessive power drain. Finally, the purpose of filters is to remove information from the signal. A new application that uses different parts of the signal will require additional new filters to be created.

In this paper an alternative solution is pursued: moving the robot. A robot is better than a stationary microphone in that if it detects that it is in a bad location, it can move to a better physical position. This research presents a method for

¹ Copyright 2004 Society of Photo-Optical Instrumentation Engineers. This paper will be published in *Proceedings of SPIE*, vol. 5609, and is made available as an electronic preprint with permission of SPIE. One print or electronic copy may be made for personal use only. Systematic or multiple reproduction, distribution to multiple locations via electronic or other means, duplication or any material in this paper for a fee or for commercial purposes, or modification of the content of the paper are prohibited.

spatially mapping out the ambient noise in the environment, and using it to navigate out of, or around, acoustically poor locations.

The remainder of this paper is organized as follows. Section 2 presents related work in the area of auditory applications on mobile robots. Section 3 describes the experimental setup used for this research. Section 4 addresses the creation and use of environmental noise maps and the last section provides a summary and future work.

2. RELATED WORK

Audition is a tough sensory modality for robots. Setting filters for an incoming audio signal is a hard task to begin with, and roboticists want to add to the environment a machine that is loud, nearby, and constantly moving the microphone around. To build auditory applications on a robot, researchers have been forced to develop novel methods for overcoming the problems inherent to the domain.

Sound source localization is one application area in robotics that has received considerable attention. It is a difficult problem due to echoes and interference from noise sources in the surrounding environment. Work by Webb in the study of cricket phonotaxis [5] has suggested a uniquely robotic solution to the sound source localization problem. Her work uses the direct audio signal from a binaural microphone to control the direction of a Khepera robot. If no sound is heard, then the robot moves straight, otherwise the intensity differences between right and left “ears” cause the robot to move towards the loudest intensity. The path resulting from this behavior-based solution is not necessarily straight, but the simple control mechanism does produce movement towards the sound source. More recent work in biology [6] suggests that this simple localization method can be extended to steer the organism towards a particular specific signal, using a biological pattern matching technique to filter out uninteresting noise sources.

Another sound source localization method posed by Nakadai et al.[7] also uses the robot’s own movement to increase accuracy, but combines it with more traditional localization methods. Based on a stereovision algorithm for visual epipolar geometry and implemented on a robotic humanoid, the principle is to extract the inter-aural time difference from the incoming signal and then rotate the robot’s head to extract enough geometric information to localize. By using the robots own mobility for localization, the internal noises of the robot can be effectively ignored when they do not change direction with the rotation.

Beyond mobility, robots are also equipped with other types of sensors that can be used in combination with audition to enhance robustness. The goal of the robot CHASER [8], by Yamasaki and Anzai, was to maintain a distance close enough to a human speaker to perform word recognition while the speaker moved through a variety of environments with different levels of ambient noise. To perform this sound localization task while moving, CHASER combined information from a heat sensor ring with angular sound source measurements when someone spoke. To determine the correct following distance, CHASER also monitored the ambient noise level of the environment, moving closer if there was too much noise and vice versa. The results however, were still not outstanding as the word recognition programs available at the time were not very robust.

Augmenting an auditory application with other sensory input proved very useful for natural language interfaces on mobile robots. Work by Perzanowski et al. [9] at the Naval Research Laboratory uses a gesture recognition interface for autonomous vehicles to augment verbal commands. If a user wants a robot to move to some location, instead of specifying coordinates for the robot, the user verbally speaks a command, and points toward the desired destination. When the command is spoken, the robot focuses its attention on the user, and invokes a goal tracker to recognize what the gestures made by the user correspond to. The combination of visual cues and auditory commands helps eliminate problems from incorrectly classified words.

But not every technique needs to be quite so complicated to introduce robustness into a robotic auditory application. Work by Bischoff [10] uses a simple strategy to verify spoken commands. The robot HERMES is trained in some service robotics tasks, and a human user guides the robot by issuing verbal commands that the speech recognition system can interpret. In the presence of noise or poor speech recognition results, the robot is programmed to ask for confirmation to make the speech system more robust. This simple technique, which relies on a cooperative user, significantly improves performance with only the addition of an onboard speaker.

3. Experimental Setup

For this research, we choose to implement the robotic task in a domain outside the traditional robotics laboratory. The Aware Home Research Institute[11] on the Georgia Tech campus, is a residential laboratory designed for testing new technologies for humans aging in place. The lab is a fully operational home environment, capable of supporting a resident for extended periods of time. By working in this environment, we intend to explore the advantages of a mobile robot in a home environment, and bootstrap the development of real world applications. For this initial work in noise mapping, the robot was deployed only in the unoccupied kitchen and dining room area of the second floor seen in Figure (1). Since then, it has been extended to cover a larger section of the house.

Engineered into the Aware Home is a vision based system for position tracking. The real time implementation uses 10 overhead cameras, mounted in the ceiling, attached to 6 dedicated computers in the basement to estimate the position of multiple people in the common areas of the house. To track a robot, we only used the blob tracking component of the system to return the $\{x,y\}$ coordinates of two colored blobs attached to the robot. Using the relative positions of those two blobs, we could also estimate the angular orientation of the robot. On average, the vision system returns position data once per second.

The robot used is a Pioneer2-dx robot developed by ActivMedia robotics, with a full sonar ring and wireless control. The robot is further equipped with a laptop, mounted on the rear, dedicated to processing the audio data from an omnidirectional desktop microphone. The processing load on the laptop is not large, and the robot was originally intended to process the audio signal, but adding a sound card to the pioneer platform proved infeasible due to cost. Both systems then communicate with a central server via a wireless access point. Because the robot needs to update faster than once per second (the vision system sample rate), it uses odometric measurements to track position from the last measured location.

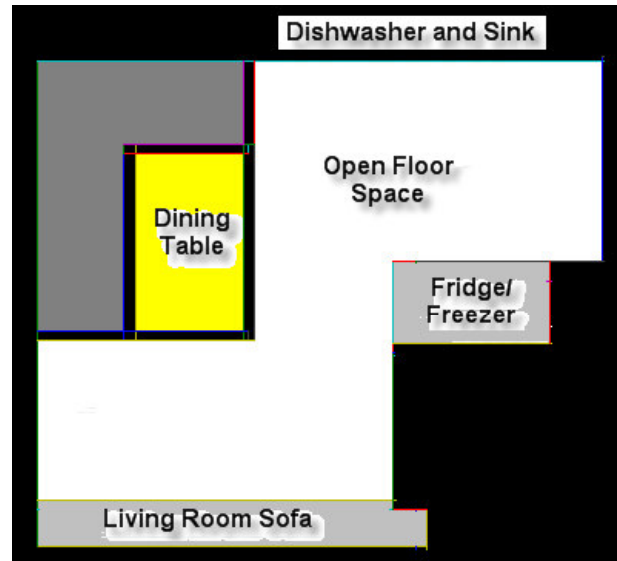


Figure 1. Floorplan of kitchen area in the Aware Home.

4. NOISE MAPPING

The solution proposed for the detection and avoidance of ambient noise on a mobile robot is a map-based approach. If the position of sources in the environment can somehow be estimated, either through prior knowledge or by actively sampling the environment, then we can create a map identifying areas of high and low ambient noise. Such a map can then be converted into an internalized plan that a robot can use in conjunction with other behaviors to avoid placing its microphone sensor in a noisy location. As it turns out, acoustical engineering has already provided a knowledge base upon that can be built upon in the form of the noise map.

The European Environment Agency defines noise maps as:

“The presentation of data on an existing or predicted noise situation in terms of a noise indicator, breaches of a limit value, the number of people affected in a certain area, the number of dwellings exposed to certain values of a noise indicator in a certain area, or on cost-benefit ratios or other economic data on mitigation methods or scenarios.”²

² http://glossary.eea.eu.int/EEAGlossary/N/noise_map. European Environment Agency. Accessed 9/8/04.

In summary, Noise Maps, also known as Noise Contour Maps, are a graphical tool commonly employed by acoustical engineers to plot the noise distribution or effects of noise distribution in an area. Most commonly, Noise Maps are developed for factory environments, where occupational safety is threatened by loud machinery or for noise impact predictions around proposed government facilities such as airports and roadways. However, they remain largely a presentation tool and are usually created by an engineer who uses his own experience to collect a minimum number of environmental samples.

In this work, the noise map representation is used to track the un-weighted sound pressure level (SPL) in the environment. The SPL is directly related to the magnitudes of the displacement amplitudes measured by the microphone[12], or strength of the measured acoustic signal:

$$SPL = \frac{1}{N} \sum_{i=1}^N |a_i - a_0|$$

where N is the number of samples per second, a_i is the sampled value, and a_0 = sampled mean value. If the SPL is a measure of the noise levels, then using a noise contour map to avoid high SPL areas in the environment is the same as avoiding noisy locations, or areas of high ambient noise. For the remainder of this paper, $N=11025$ samples/second, and the average SPL was calculated over one second of sampled data.

4.1. Creating a Noise Map

There are 3 different strategies for creating a noise map: (1) Use a model of the environment and positions of known sound sources to make a prediction; (2) Collect samples of the real environment by hand and construct the noise map; and (3) collect samples autonomously with the robot and construct the noise map. In Figure (2) each of these three different strategies to build noise maps of the kitchen area in the Aware Home is presented. The solid straight lines represent walls, and the star represents the location of an active radio sitting on top of the dining room table.

The first map (Figure (1)[top]) shows a very loud sound source, >90dB, modeled in MATLAB and assumes spherical spreading in an anechoic environment. It is an overly simple model because no echoes are involved and no other sound sources are considered. The result is a smooth graph where the noise levels drop off with the inverse square of

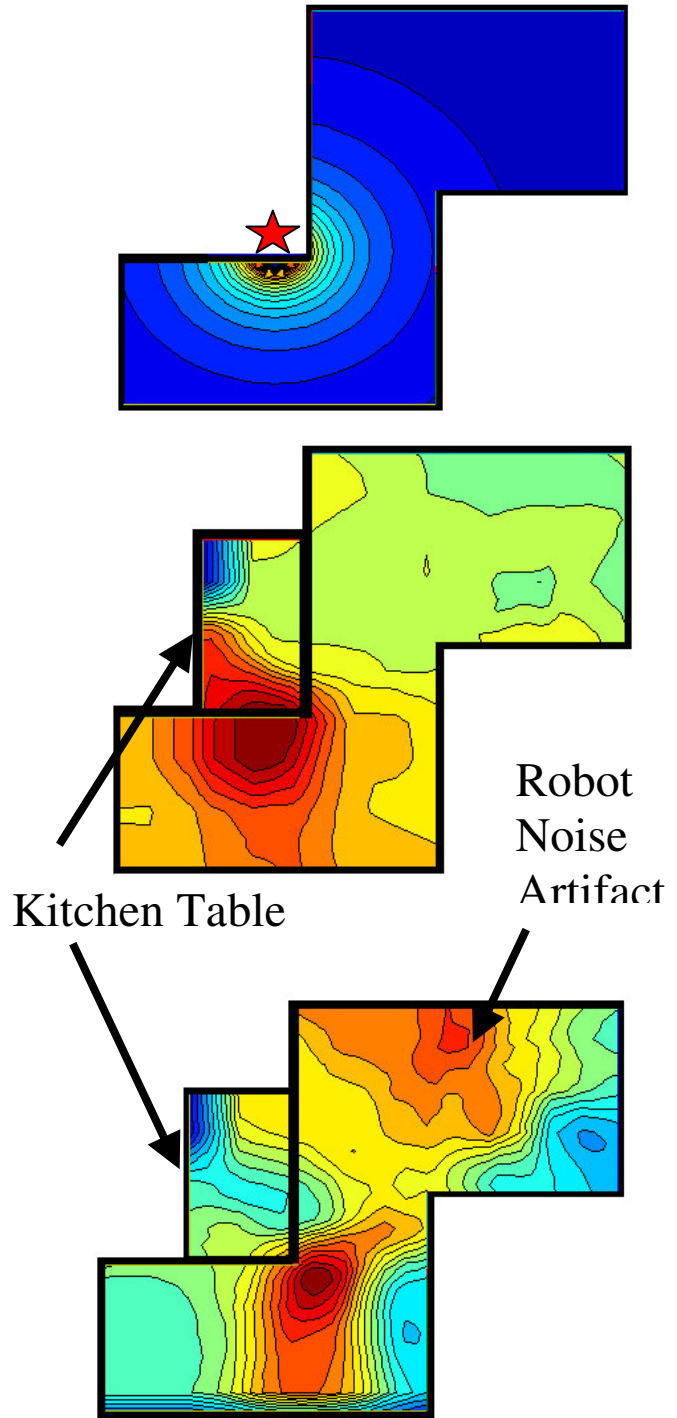


Figure 2. Noise Maps of the Aware Home kitchen. [Top] Ideal source modeled in MATLAB. [Middle] Map from hand collected samples. [Bottom] Map from robot collected samples.

the distance from the single source. Although the real environment is much more complex, this provides a working model for known sources until it can be mapped either by hand or by the robot. Furthermore, the smoothness of the resulting graph allows for smooth gradient fields that the robot can follow to avoid the sound source.

The other two maps were both generated using sampled data, either collected by hand or by a robot exploring the environment, to represent the current noise levels in the environment. Treating the map construction problem as a 2D function approximation, the K-means algorithm is used where $K=5$, to create the map where we do not have enough sample data. To allow real time map building while the robot is exploring the environment, a simple threshold was added to limit which points may be used in approximating the function; i.e., the K-means algorithm considered only points within a two-meter radius. Finally, a 2D hanning filter[13] of radius 0.3m was used to smooth the resulting map.

The middle map on Figure (2) was created from 250 hand sampled data points, using a microphone moved to 36 different locations around the kitchen. The overhead vision system was used to track the location of the microphone while sampling. The sound source measured was a single portable stereo generating static noise from the channel an inactive FM channel. The resulting contour map has a single large peak in front of the radio just as was predicted by the spherical spreading model. It is not quite as smooth as the idealized version, but represents the actual environment instead of a modeled version. Although no sample points were collected under the table since the vision system cannot track objects beneath the table, function approximation can still be used to generate an estimate for that area should the robot need to move in that region. Movement under the table, however, was not allowed in any of the tests reported below.

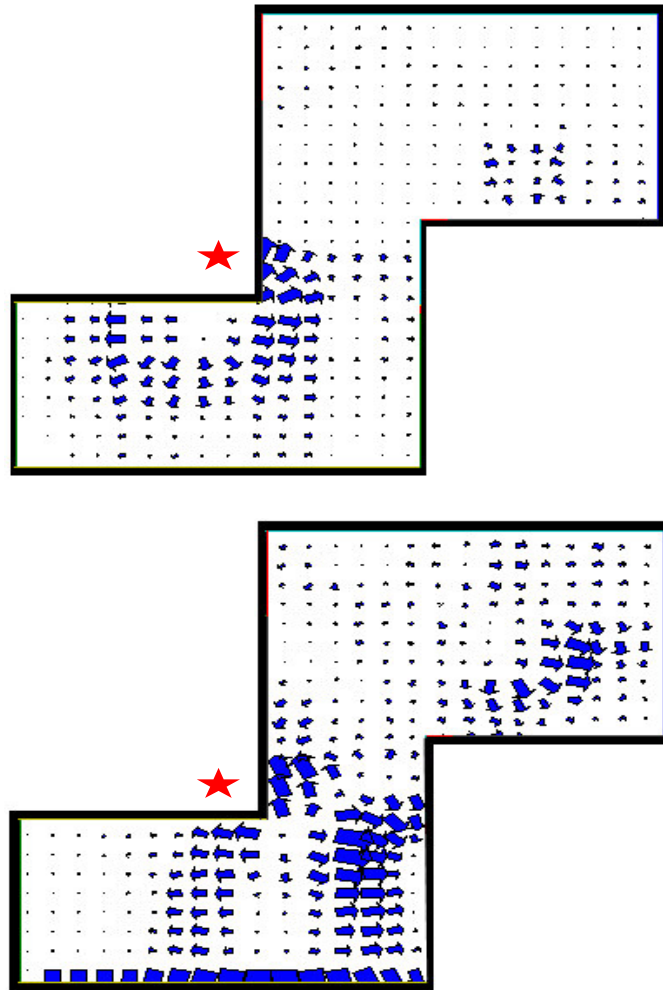


Figure 3. Vector field representation of gradient for (left) hand collected samples and (right) robot collected samples.

The third noise map, Figure (2)[bottom], was generated from sample points collected by the mobile robot covering the environment. The radio generating static noise was also used for this example in the same location as before. This final map however, is even rougher than the hand sampled one and the source appears shifted to the right. One reason for this distortion is that the robot does not sample the space as evenly as the person does. More samples are gathered from the middle, where obstacles do not influence the robot's motor behavior. Another reason for the noise peak displacement, as well as general roughness, is due to some inaccuracies in the visual localization system itself. When the vision system does not update fast enough, and odometry is used to estimate position, the error on localization estimates increases. The reason for this is that the angular position estimate returned by the vision system has a higher error than the $\{x,y\}$ estimate made by the vision system, so odometry measurements made from these coordinates are only as reliable as the angular position estimate.

A third reason for the roughness is that a moving robot generates its own noise that can interfere with the creation of a noise map. This generates echoes that can change the noise map, and may also introduce a "phantom" noise source, created due to wheel noise when the robot stayed in one location for too long during navigation. These false peaks in the noise contour map do not always occur, but when they do, it is difficult to remove them given the K-means approximation. Acquiring enough samples would remove the problem, but many of these samples will probably also be filled with wheel noise.

Despite these problems with the autonomous collection of samples, the resulting noise map is still useful as will be demonstrated in Section 4.4.

4.2. Building a Gradient Field

To enable a robot to use a noise map, a potential field is created that represents how the sound-scape should be navigated. The potential field is a 2D-gradient produced from the noise maps described previously, where it is easily converted into a vector field representation. Following this field results in a similar behavior to avoid-past[14], except instead of avoiding where the robot has been the gradient reflects the expected measurement strength in an area.

The creation of the gradient field is a simple formula applied iteratively for every point (i,j) in the map. Given a map (M), the gradient in the X-direction for each point is created as:

$$G_x(i, j) = \frac{M(i+1, j) - M(i-1, j)}{2}$$

and the gradient in the Y-direction is created as:

$$G_y(i, j) = \frac{M(i, j+1) - M(i, j-1)}{2}$$

If there were N points in the X-direction in map M, then the resulting gradient field would have (N-2) points in the X-direction, and is similarly smaller in the Y-direction. In practice, this difference in size is unimportant since the original map is not required to be exactly the same size as the space explored by the robot. The implementation used had 10-points/meter, so a missing point on the edge resulted in only a difference of a centimeter.

Figure (3) demonstrates the resulting vector field representations for both the hand sampled and robot sampled noise maps created earlier (Figure 2). Both maps show a strong avoidance from perceived peak. As discussed earlier, this peak is shifted slightly from the true peak in the robot-sampled map, but is still within the correct vicinity. The robot-sampled map also shows the effects of a “phantom” noise source in the kitchen area where the hand-sampled map reports a minimum.

4.3. Behavioral Controller

The controller used to guide the robot, employed a weighted summation of vectors generated by a set of behavioral functions. There were a total of 3 different tasks requiring the noise map, and one that did not: create noise map, avoid sound source, follow waypoints using a noise map, and follow waypoints without a noise map. The behaviors used in each task were as follows:

Task	Behaviors
Create Noise Map	Area Coverage, Avoid Obstacles, Wander
Avoid Sound Source	Avoid Obstacles, Follow Noise Map, Wander
Follow Waypoints w/ Noise Map	Avoid Obstacles, Follow Noise Map, Follow Waypoints, Wander
Follow Waypoints w/o Noise Map	Avoid Obstacles, Follow Waypoints, Wander

The Avoid Obstacles, and Wander behaviors are described in [15]. Follow Noise Map is the behavior which follows the potential field created in section 4.2. Follow Waypoints is implemented as an FSA using Move-To-Goal[15] behaviors to follow a trail of pre-specified waypoints.

4.3.1. Area Coverage

In order to create a noise map autonomously, the robot needed a strategy for traversing the environment, performing an area coverage task. Our strategy is a grid-based algorithm with the goal being to minimize the using knowledge of existing obstacles in the environment to construct shortest paths. The minimize the age of the audio samples in the noise map, keeping the map up to date even after one set of samples has been collected for an area. The heuristic algorithm is defined as follows:

1. Divide the environment into 1-m^2 grid cells $\{i,j\}$.
2. Compute the all-pairs shortest path $\{p_{ab,ij}\}$ and path distance between grid cells $\{d_{ab,ij}\}$.
3. Track, for every grid cell, the last time ($t_{i,j}$) an audio sample (SPL reading) was collected in that grid cell. Every grid cell is initialized to the current system time at start up.
4. At each time-step while the robot is moving,
 - a. Compute, for each grid cell $\{i,j\}$, a score $V_{i,j} = \left(1 - \frac{t_{i,j} - t_{\min}}{t_{\max}}\right) * \frac{1}{d_{ij,ab}}$; where $\{a,b\}$ is the cell the robot is currently located in. This score thus represents the tradeoff between the age of the knowledge about a grid cell vs. the distance to reach that cell.
 - b. Select the grid cell with the highest score as the goal.
 - c. Because the localization system has some amount of error, and we are using the robots current position to calculate score values, the order of these scores can change from turn to turn. To prevent the robot from rapidly switching between closely scored “goal” cells, we introduce some momentum to the previously selected path. If the grid cell with the highest score is not significantly greater (less than some threshold) than the current score of the goal cell selected at the last time-step, then use the previous “goal”.
 - d. Follow the shortest path $\{p_{ab,ij}\}$ to the selected “goal” cell $\{i,j\}$.

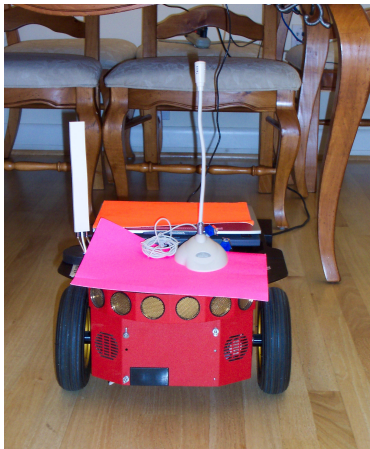


Figure 4. Picture of robot in kitchen area of aware home.

This heuristic was chosen for a number of reasons. For one, it uses a weighted system similar to the behavioral controller. This means simple integration with the system, and also allows a dynamic priority selection of distance versus time, and potentially other sub-goals that might influence the area to be moved too. The heuristic also had a small computational load because in this environment with a lot of obstacles, there were many inaccessible areas that a dynamic path creation scheme could spend a lot of computational time trying to reach. With this method, the computed score for each grid cell is inversely proportional to the path length and grid cells for which there is no path because of obstacles ($d_{ab,ij} = \infty$) have a zero value, so the robot never tries to move to them.

While the heuristic produces maps (see Figure (2)[bottom]) that do represent the environment, and which a robot can follow, there is still a problem with the approach when there are grid cells that are partially obstructed. Those that are more than half obstructed by obstacles are not a problem because they have an unreachable cell center and are never chosen as target destinations. But those cells that are less than half obstructed can generate local minima when combined with obstacle avoidance. The robot tries repeatedly to reach the cell, only to be driven back by obstacle avoidance. While the robot can reach all of the cells on this map eventually, each time the robot turns abruptly it makes noise which shows up on the noise map. If the robot stays in one place long enough, then a “phantom” noise source might show up on the created noise map.

	Average SPL Reading
No Noise Map	138
With Robot Created Noise Map	125 (109)

In this preliminary work, the robot completed 10 total runs along the waypoint path, five times each with and without using a noise map created from robot collected samples. One of the runs while the robot was using the noise map actually measured a higher average SPL than any run without using the noise map. The reason is that obstacle avoidance failed on this test and the robot crashed into the wall, producing loud noises which skewed the results. Even including that poor result, on average the robot using the noise map still performs better than without the noise map. Ignoring the one poor result, the robot does 19% better with the map than without.

These results suggest that the noise map can be useful for avoiding loud areas of the environment while actively performing some task besides noise avoidance. To generalize this work, a more thorough study over a larger area is being planned to determine the general usefulness of noise maps, and how they can be employed.

4.5. Variations in Sound Sources

While this work was primarily tested with a single source, an fm radio generating static noise, it also works with a number of other sound sources. The results, however, vary with the different attributes common to audio sources. An unsteady audio stream, for instance, will affect the roughness of the gradient on noise maps created from sample-based methods. Figure (6) illustrates two of these additional sources.

The left map of Figure (6) was created by hand sampling the environment surrounding a radio tuned to a local pop music station. This is an example of an unsteady sound stream, as the music varies in strength and pitch, and commercials periodically interrupt the music. Even though the data points were hand collected providing a more evenly distributed set of samples, the resulting contours are not smooth. Adjacent to the radio remains the loudest part of the room, but the resulting contours are very noisy and would be harder for a robot to follow. One possible solution to smooth out the map for these types of sources is to collect a lot more data points, especially in the vicinity of the radio. If multiple robots are introduced into the environment for autonomous collection of samples, then it might be feasible to collect enough data points over time. Another possible solution is to use larger 2D filters to smooth the map. This, however, might work better if we first recognize the source and so can use the most appropriate filter.

Another example source, common to the kitchen environment, is a refrigerator. Periodically, the compressor in the refrigerator turns on to produce more cool air. While it is on, there is a steady sound stream coming from the refrigerator that can be detected and mapped. The map on the right in Figure (6) is an example. The samples used to create this map were autonomously collected, and reflect the loud noises from the compressor and the resulting echoes off the surrounding hardwood environment. However, when the compressor is not on, sampled based maps do not reflect a sound source in that vicinity (see middle map of Figure (2)). The refrigerator is a statically located source, producing a medium duration steady strength stream. Because the source is steady strength while the compressor is on, a good solution would be to recognize that the compressor has turned on/off and switch between maps accordingly. Each map would be a good but separate representation of the environment with or without the compressor, and samples from far enough away could probably be used to construct both maps.

A third noise source during experimentation that was picked up regularly was traffic noise from outside the house. The Aware Home sits on a four-lane road over which both truck and car traffic flow regularly. During peak hours, the traffic could result in a loud area closest to the road, as well as a general increase in the overall noise level recorded on the noise map. The noise from cars is actually a moving source, but the overall noise from the road can be reflected as a statically located source, just outside the house, which produces an unsteady sound stream like the pop music from the radio. The difference is that the variation is more intense, ranging from nothing to loud tractor-trailers, and this noise problem is also dependant on the time of day. As a result, this is probably the hardest of the three types of sources to accommodate with a noise map. Like the refrigerator noise, the traffic noise can be partially represented by switching between maps for rush hour, noon, and other relevant times of day. However, extra sampling may not take care of the effects of an unsteady sound stream. It would produce an average road noise level, but this may be useless information for the application if loud trucks are equally loud throughout the kitchen area. In some cases, it may be more useful to recognize that a very loud truck has passed and then ignore those samples.

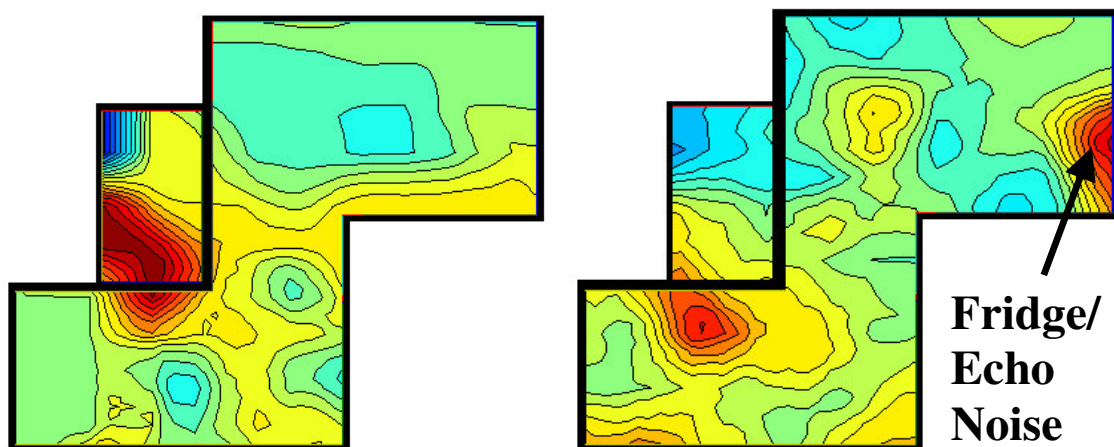


Figure 6. Noise Maps with other sources. (Left) Hand sampled map of radio playing local pop music. (Right) Robot sampled map of radio playing static plus refrigerator noise on the right.

A common feature of each of these solutions is that they seem to depend on first recognizing the type of interference; not necessarily the specific source, like a car engine, but at least some properties of the interference, such as duration and noise level regularity. This suggests a level of application dependence in determining which noise map should be used, and how the map should be collected. Machine learning could be used to try and discover time dependencies between types of noises, or if the information is available (like the time of rush hour), then the application designer can adjust the maps accordingly.

5. CONCLUSION

In summary, this paper has presented Noise Maps as a tool for robots to use in performing acoustically sensitive navigation. Three different techniques have been demonstrated for building Noise Maps: modeling known sources, hand sampled points, and autonomous sampling by the robot. The model-based approach is the simplest because there is no environmental sampling involved, and the resulting contours are clean, smooth and easy to follow. The hand sampled approach however, represents the actual state of the environment at some point in time and still provides relatively smooth contours, but requires the human to manually collect samples over the entire area. Finally, the autonomous approach to constructing a contour has the advantage of representing the latest state of the environment, but includes the noise that the robot itself generates, and the autonomous map creation heuristic used had drawbacks that can affect the overall correctness of the noise map.

The validity of the resulting noise maps were tested by converting the noise map into a potential field, and using it to avoid noisy areas in the environment. Two tests were performed. First, the robot used the noise map to move from a very poor auditory location to a quieter location in the environment. All three of the strategies for creating noise maps produced similar performance, although the model-based approach produced a greater variation in results from either of the sample-based methods. Second, the robot followed a waypoint mission while using the map to avoid noisy areas. In this scenario, only the autonomously generated map was tested, but it still demonstrated a reduction in general noise over the traditional waypoint implementation.

These results have indicated that noise maps can be useful tools for representing and avoiding noise in the environment. However, they have also indicated that different types of noise may require different strategies in collecting, and/or representing the noise appropriately for the application, so the use of noise maps should be application and environment driven. Thus the next step is to more thoroughly explore the noise map approach in support of a more realistic auditory application on a mobile robot.

REFERENCES

1. Fong, T.W., I. Nourbakhsh, and K. Dautenhahn. *A survey of socially interactive robots*. Robotics and Autonomous Systems, Special issue on Socially Interactive Robots. **42**: p. 143-166. 2003.
2. Huang, J., N. Ohnishi, and N. Sugie. *Mobile Robot and Sound Localization*. in the Proc. of *Intelligent Robotics and Systems (IROS)*: p. 683-689. 1997.
3. Youn, S.H. and M.V. Scanlo. *Robotic vehicle uses acoustic array for detection and localization in urban environments*. in the Proc. of *SPIE*: p. 264-273. 2001.
4. Parker, L., B. Kannan, X. Fu, and Y. Tang. *Heterogeneous Mobile Sensor Net Deployment Using Robot Herding and Line-of-Sight Formations*. in the Proc. of *Intelligent Robotics and Systems (IROS)*: p. 2488 - 2493. Las Vegas, NV. 2003.
5. Webb, B. *Robots crickets and ants: models of neural control of chemotaxis and phonotaxis*. Neural Networks. **11**: p. 1479-1496. 1998.
6. Hedwig, B., Poulet, J. *Complex auditory behaviour emerges from simple reactive steering*. Nature. **430**: p. 781-785. 2004.
7. Nakadai, K., K. Hidai, H.G. Okuno, and H. Kitano. *Epipolar Geometry Based Sound Localization and Extraction for Humanoid Audition*. in the Proc. of *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*: p. 1395 - 1401. Maui, Hawaii. 2001.
8. Yamasaki, N., Y. Anzai. *Active Interface for Human-Robot Interaction*. in the Proc. of *Int. Conf. on Robotics and Automation (ICRA)*: p. 3103-3109. 1995.
9. Perzanowski, D., A. Schultz, W. Adams, and E. Marsh. *Using a Natural Language and Gesture Interface for Unmanned Vehicles*. in the Proc. of *Unmanned Ground Vehicles II, Aerosense 2000*: p. 341-347. 2000.
10. Bischoff, R. *Towards the Development of 'Plug-and-Play' Personal Robots*. in the Proc. of *IEEE-RAS International Conference on Humanoid Robots*. Cambridge, MA. 2000.
11. Abowd, G., A. Bobick, I. Essa, E. Mynatt, and W. Rogers. *The Aware Home: Developing Technologies for Successful Aging*. in the Proc. of *Workshop held in conjunction with American Association of Artificial Intelligence (AAAI)*. Alberta, Canada. 2002.
12. Raichel, D.R. *The Science and Applications of Acoustics*. New York, NY: Springer-Verlag. 2000.
13. McClellan, J., W. Schafer, M. Yoder. *DSP First: A Multimedia Approach*. Upper Saddle River, NJ: Prentice-Hall, Inc. 1998.
14. Balch, T., R. C. Arkin. *Avoiding the Past: A Simple but Effective Strategy for Reactive Navigation*. in the Proc. of *Int. Conf. on Robotics and Automation (ICRA)*: p. 678--685. 1993.
15. Arkin, R.C. *Behavior-based Robotics*. Cambridge, MA: MIT Press. 1998.