A Nonparametric Learning Approach to Vision Based Mobile Robot Localization

Gregory Z. Grudic

Dept. of Elec. and Comp. Eng.
University of British Columbia
Vancouver, British Columbia, Canada
gregg@ee.ubc.ca
http://www.ee.ubc.ca/~gregg

Peter D. Lawrence

Dept. of Elec. and Comp. Eng.
University of British Columbia
Vancouver, British Columbia, Canada
peterl@ee.ubc.ca
http://www.ee.ubc.ca/~peterl

Abstract

A nonparametric learning algorithm is used to build a robust mapping between an image obtained from a mobile robot's on-board camera, and the robot's current position. The mapping uses 19,200 unprocessed pixel values (160 by 120 pixel image). Because the learning algorithm is nonparametric, it uses the learning data obtained from these raw pixel values to automatically choose a structure for the mapping without human intervention, or any a priori assumptions about what type of image features should be used. The learning data consisting of a series example image inputs and corresponding position values, is collected in a calibration phase where the robot randomly traverses its intended workspace. This process of building visual localization maps for mobile robots is completely general and can be applied to any implementation which uses on board cameras. We demonstrate the feasibility of this approach on a mobile platform performing in a robotics laboratory workspace. This workspace is visually cluttered, with humans and other objects continually moving within the robot's environment. The mapping learned in this environment is robust to these dynamic visual features and consistently reports timely localization information (at greater than 7 Hz) to within acceptable limits.

1 Introduction

In order for a mobile robot to work and navigate successfully, it must be able to control its position within its environment. Towards this end, many researches have proposed using visual information obtained from cameras located on board the mobile robot for localization. Some recent examples of this include [8] [2] [1] [9]. Many such systems have demonstrated very good performance within the environments they were designed for. However, most vision based localization systems cannot easily be adapted to environments which they were not originally intended for. For example, a vision system which uses models of objects within the robot's indoor environment for localization, may not easily be adapted for outdoor environments which require other image features for navigation. This is because most current approaches require a human to pick specific landmarks by which to navigate, thus making the localization system application specific [10], often requiring a complete redesign for each new environment.

In this paper we propose an approach to vision based mobile robot localization which is similar to that proposed in [10]. We make use of an on board camera and a *temporary* calibration setup which continuously reports the robot's position, to learn the mapping between the on board camera's image and its position. During this calibration phase, the robot is directed to move through its environment and its position and corresponding camera images are recorded. Then a learning algorithm is used to construct the mapping between raw image pixels and the robot's position.

As with any vision based localization system, the implicit assumption being made is that the there are visual features in the environment which are essentially fixed, and that these can be differentiated from transient features (such as humans and objects which move around) that are not permanent and must be treated as noise. However, unlike in [10] where a small subset of derived image features is used to learn the localization mapping, or in implementations where a human designer attempts to choose the best set of visual features, we make no assumptions about what type of features are required for effective localization. We propose using raw pixel values (19,200 in the current paper) as inputs to our learning algorithm and therefore our approach is completely general in that it can be applied equally effectively in any visual environment.

The use of a general nonparametric learning algorithm for building our vision based localization mappings is key to our approach, because no *a priori* assumptions are required about which model structure is needed to build an effecting mapping. Other approaches to learning vision based localization have used fixed, *parametric* learning algorithms, thus limiting the type of localization mappings which can be learned [10]. The nonparametric learning paradigm is able to automatically choose an appropriate model structure based solely on the learning data [3]. However, it is only recently that nonparametric learning algorithms have been discovered which can effectively build the high dimensional mappings (i.e. 19,200 inputs used here) that are required for learning vision based localization [7]. These new algorithms are able to build mappings using tens of thousands of input variables, which allows them to effectively build robust visual localization mappings using raw (unprocessed) pixel inputs.

2 The Proposed Framework

There are two specific issues associated with learning a mapping between what a robot sees and its current location within the world. The first is how one goes about learning this mapping. The second is, given that we have learned this mapping for a specific area, how will the robot know if it has strayed outside this learned area? These are addressed in the following sections.

2.1 Learning the Visual Mapping

We use a 3 step procedure to build the image to position mapping. The first step involves collecting the learning data, the second step uses an off-line learning algorithm to build the mapping, and the final step involves verifying the validity of the mapping within the robot's workspace. A description of each of these steps is given below.

STEP 1: Collecting The Learning Data. The goal of this step is collect camera images and corresponding position data which span the entire workspace of the mobile robot. A mobile robot is expected to perform within a specific workspace defined its assigned task. For example, an office robot might be expected to deliver mail to various offices within a single building; a manufacturing robot might need to deliver parts to certain locations in a production line; and a fruit picking robot might be required to pick fruit in a specific part of an orchard. Hence, the first step of our procedure involves moving the robot throughout its workspace region and collecting images and corresponding robot positions. This requires the use of a temporary calibration system which can report the position and orientation of the mobile robot while images are being collected. In Section 3.2 one such calibration system is briefly described.

STEP 2: Building the Mapping. The goal of this step is to learn the mapping between the camera image and the mobile robot's position using the data collected in STEP 1. We symbolize this mapping as $\mathbf{M}(\cdot)$ and represent it as follows:

$$(x^{cam}, y^{cam}, \theta^{cam}) = \mathbf{M}(\mathbf{Cam_1}, ..., \mathbf{Cam_n}) \qquad (1)$$

where $\mathbf{Cam_1}, ..., \mathbf{Cam_n}$ are the n cameras located on the mobile robot, x^{cam} is the estimated x position of the mobile robot given the current camera images, y^{cam} is the estimated y position of the mobile robot given the current camera images, and θ^{cam} is the estimated θ, the mobile robot's orientation, given the current camera images. For building this mapping $\mathbf{M}(\cdot)$ we use the *SPORE-1* nonparametric learning algorithm described in [7] and [4]. This algorithm was chosen because it was shown to work well on very high dimensional learning problems, both in real world human-to-robot skill transfer experiments [5] [6], and on very high dimensional [4] synthetic data.

STEP 3: Test Validity of the Mapping. The goal of this step is to collect a set of test data to determine the validity of the mapping learned in STEP 2. This involves once more moving the mobile robot throughout its workspace region and collecting a new set of images and corresponding robot positions (obtained from the temporary calibration system). It is important that this test data is independently generated to ensure that an unbiased estimate of the accuracy of the learning mapping is obtained. It is also important that this new set of data span the complete workspace of the robot. This test data is then passed through the learned mapping $\mathbf{M}(\cdot)$ and an error estimate is obtained for how accurate the mapping is. We estimate the error of the mapping by the symbols ϵ_x^{max}, ϵ_y^{max} and ϵ_θ^{max} which are define as follows:

$$\max_{TestData} |x^{cam} - x^{cal}| + E_x^{cal} = \epsilon_x^{max} \qquad (2)$$

Figure 1: Example images from Camera 1.

$$\max_{TestData} |y^{cam} - y^{cal}| + E_y^{cal} = \epsilon_y^{max} \quad (3)$$

$$\max_{TestData} |\theta^{cam} - \theta^{cal}| + E_\theta^{cal} = \epsilon_\theta^{max} \quad (4)$$

where x^{cam}, y^{cam} and θ^{cam} are the estimates of position and orientation as defined in Equation (1), x^{cal}, y^{cal} and θ^{cal} are the position and orientation of the mobile robot obtained using the calibration system, and E_x^{cal}, E_y^{cal}, and E_θ^{cal} are the maximum errors in x, y, and θ respectively, relative to the calibration system (note that E_x^{cal}, E_y^{cal}, and E_θ^{cal} are determined by the specific calibration system being used as indicated in Section 3.2). If the calculated values for ϵ_x^{max}, ϵ_y^{max} and ϵ_θ^{max} are within error tolerances for the entire workspace of the robot, then the task of learning the mapping $\mathbf{M}(\cdot)$ is done. However, if they are not, then the test data is used to identify in which regions of the robot's workspace the accuracy of the mapping is unacceptable, and needs to be improved. Once these regions are determined, then more learning data needs to be collected from these regions, and added to the original set of learning data. This augmented learning data is then used to build a *new* mapping in STEP 2. Thus, we iterate Steps 2 and 3 until acceptable localization error is obtained for the robot's entire workspace.

2.2 Straying Outside the Learned Workspace

If we assume that the mapping $\mathbf{M}(\cdot)$ was learned in the desired workspace region, then it is important to be able to determine if the mobile robot strays out of this learned region. When the robot leaves the learned workspace region, then the mapping $\mathbf{M}(\cdot)$ is no longer valid and therefore using in to navigate is not a good option.

We use the following active sensing strategy to detect when the robot is no longer within the learned workspace. First a reading is obtained at the mobile robot's current position using Equation (1). This first position is termed the robot's initial position and is symbolized by x_i^{cam}, y_i^{cam} and θ_i^{cam} (values obtained using Equation (1)). The robot is then moved to a second (termed final) position by directing it to travel a predefined path using odometric readings as a guide. We symbolize this odometry based relative motion by the symbols Δx^{odom}, Δy^{odom} and $\Delta\theta^{odom}$ for changes in x, y and θ respectively. Then Equation (1) is used to obtain an estimate for the robot's final position, which is symbolized by x_f^{cam}, y_f^{cam} and θ_f^{cam}. Then, if at least one of the following 3 conditions is *NOT* true, the mobile robot has strayed outside the learned workspace:

$$||x_i^{cam} - x_f^{cam}| - \Delta x^{odom}| + E_x^{odom} < \epsilon_x^{max} \quad (5)$$

$$||y_i^{cam} - y_f^{cam}| - \Delta y^{odom}| + E_y^{odom} < \epsilon_y^{max} \quad (6)$$

$$||\theta_i^{cam} - \theta_f^{cam}| - \Delta\theta^{odom}| + E_\theta^{odom} < \epsilon_\theta^{max} \quad (7)$$

where E_x^{odom}, E_y^{odom}, and E_θ^{odom} are the maximum errors in x, y, and θ respectively, due to the odometric system. One should note that the above condition must be true by definition because the mapping $\mathbf{M}(\cdot)$ in Equation (1) can only be meaningful in areas where learning has taken place.

3 The Experimental Test-bed

Our nonparametric learning based robot localization scheme was tested in a 5 meter by 6 meter region of the UBC robotics laboratory. The process of acquiring the vision data is based on a robot that has three different on board cameras. The first, here termed *Camera 1*, has an image plane which is perpendicular to the floor and faces towards the front of the robot. The second, termed *Camera 2*, also has an image plane which is perpendicular to the floor but faces a direction which make an angle of 90 degrees (counter clockwise) with respect to *Camera 1*. The

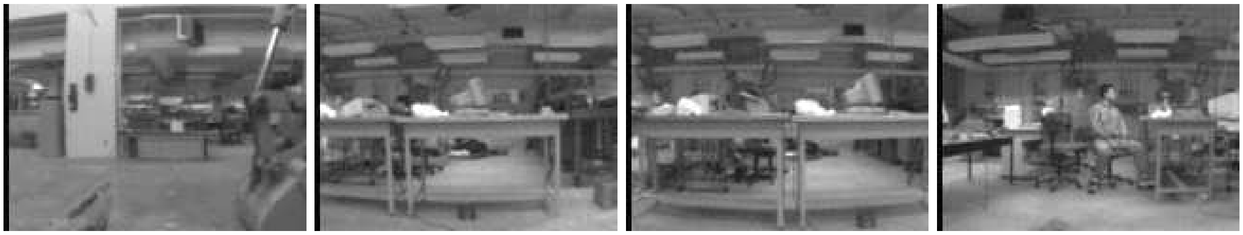

Figure 2: Example images from Camera 2.

final camera, *Camera 3*, has an image plane which is parallel to the ground and faces towards the ceiling of our workspace. Typical images seen through these cameras are shown in Figure 1, Figure 2 and Figure 3. Each of the three cameras generates an 8 bit grey scale image of 160 by 120 pixels.

The mobile robot platform we are prototyping in these experiments has 3 wheels, 2 independently actuated and one free to rotate. The independently actuated wheels each have odometer sensors which allow us to obtain rough estimates of the robot's motion. For our initial prototyping experiments we use the calibration camera to substitute for the odometer readings which are used in Section 2.2. The maximum errors in our odometer readings are therefore assumed to be the same as those of the calibration system. Thus $E_x^{odom} = 0.11$ meter, $E_y^{odom} = 0.10$ meter, and $E_\theta^{odom} = 3.8$ degrees, where these variables are defined in Section 2.2.

3.1 The Mobile Robot's Workspace

The mobile robot's workspace is a typical, cluttered robotics laboratory as depicted in Figure 1, Figure 2 and Figure 3. There are people continually moving throughout the lab, and chairs and desks are periodically being moved. Thus, the task of localizing the robot's position based on image data is difficult because potentially viable landmarks are periodically being moved, and the robot must learn not to depend on them for position readings. The total area of the workspace is 6 meters by 5 meters.

3.2 The Calibration System

We use a camera based calibration system which uses three light sources that are attached to the mobile robot during calibration. A calibration camera is located on the ceiling and is thus able to continually track the three light sources on the mobile robot as it moves within the workspace. The position and orientation of the mobile robot can directly be determined based on the position of the three light sources in the calibration camera's image. The calibration system is described in detail in [11].

The calibration system has the following errors: $E_x^{cal} = 0.11$, $E_y^{cal} = 0.10$, and $E_\theta^{cal} = 3.8$ degrees (these symbols are defined in Equation (2), Equation (3) and Equation (4) respectively).

4 Experimental Results

4.1 Using Two Cameras

In the first set of experiments *Camera 1* and *Camera 2* are used to build the mapping $\mathbf{M}(\cdot)$ as defined in Section 2.1. The initial learning data is generated by allowing the mobile robot to randomly move through its workspace. Throughout this random motion the pointing direction of *Camera 1* and *Camera 2* is maintained such that the image plane of *Camera 1* is parallel to the x axis (to within an error of 10 degrees) of the calibration camera, while the image plane of *Camera 2* is parallel to the y axis (to within an error of 10 degrees) of the calibration camera. Therefore the robot's cameras maintain the same orientation to within 10 degrees. Thus this first experiment doesn't require us to learn the robot's orientation, only its position. Figure 1 and Figure 2 give examples of the images obtained from *Camera 1* and *Camera 2* respectively.

During the learning phase 2000 images and corresponding robot positions are gathered from each of the two cameras. This learning data is then passed to the SPORE-1 nonparametric learning algorithm and two mappings are generated. Both of these have $19,200$ raw pixel inputs (i.e. 160 by 120 image) and 1 output. The input to the first mapping is the image from *Camera 1* and the output is the estimated position x^{cam}. The input to the second mapping is *Camera 2* and the output is y^{cam}. The learning algorithm took

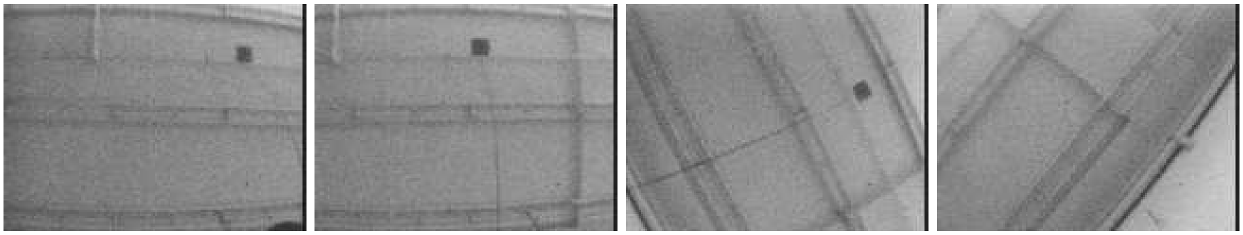

Figure 3: Example images from Camera 3.

approximately 1.5 hours (running on a SPARC 20) to generate these two mappings and each mapping takes up approximately 1.3 MB of memory.

In order to test the accuracy of the two mappings, the robot was once more randomly moved throughout its workspace. As with the learning data, throughout this random motion the world axis alignment of the cameras was maintained to within 10 degrees. During this test phase 1000 images and corresponding robot positions were gathered from each of the two cameras. When this test data was applied to the learned mappings, the following errors were observed (see Equation (2) and Equation (3) for definitions): $\epsilon_x^{max} = 0.15$ meters and $\epsilon_y^{max} = 0.14$ meters (note ϵ_θ^{max} is not calculated because orientation is not being learned). The mappings generate (x^{cam}, y^{cam}) position estimates at a rate 12 times a second on a SPARC 20.

Next, in order to test the localization system's ability to recognize when the robot is outside the learned workspace, we allowed the robot to leave this learned region and randomly performed 1000 odometric verification tests as described in Section 2.2. In each of the test examples, the orientation of the two cameras was maintained to within the same tolerances as for the learning phase, but the position of the robot was always outside the original 6 meter by 5 meter workspace. For 814 of these test points the robot recognized that it was outside of the learned region. However, the other 186 test examples, were close enough (within 1 meter) to the original workspace boundary to make the learned position mappings still valid to within error calibration limits, even though they where outside the learned workspace.

As indicated above, the orientation of the cameras was fixed throughout this set of experiments. This required us to directly control the robot's orientation as it moved around. Obviously it is desirable to use the learned mappings to control the robot's alignment when it is moving within the learned workspace region. We used a modified version of the odometric verification procedure define in Section 2.2 as follows. We started with the mobile robot within its learned workspace, but with the wrong orientation. The robot's orientation was then incremented and decremented in 5 degree steps until small odometer based motions in x and y were consistent as defined in Equation (5) and Equation (6). When consistency was achieved, then the robot had reached the same orientation (to with 10 degrees) which it had during the learning phase.

4.2 Using One Ceiling Pointing Camera

In the second set of experiments the ceiling pointing camera, *Camera 3*, is used to build the mapping $\mathbf{M}(\cdot)$ as defined in Section 2.1. The initial learning data is generated by allowing the mobile robot to randomly move through its workspace. Throughout this random motion, the orientation and the position where allowed to vary randomly (unlike in the previous experiment where the orientation was constrained). Figure 1 shows a set of example images obtained from *Camera 3*.

During this learning phase 2000 images and corresponding robot positions and orientations were gathered from *Camera 3*. This learning data was then passed to the SPORE-1 nonparametric learning algorithm which generated a mapping of $19,000$ raw pixel inputs (i.e. 160 by 120 image) to 3 outputs (i.e. $x^{cam}, y^{cam}, \theta^{cam}$ defined in Equation (1)). The learning algorithm took approximately 2 hours (running on a SPARC 20) to generate this mapping and the mapping takes up approximately 2.0 MB of memory.

In order to test the accuracy of this mapping, the robot was once more randomly moved throughout its workspace. During this test phase 1000 images and corresponding robot positions were gathered from *Camera 3*. When this test data was applied to the learned mapping, the following errors were observed (see Equation (2), Equation (3 and Equation (4 for definitions): $\epsilon_x^{max} = 0.18$ meters, $\epsilon_y^{max} = 0.17$ meters and $\epsilon_\theta^{max} = 8.3$ degrees. The rate at which these position estimates are updated is 7 times a second on a SPARC 20.

Next, in order to test the localization systems ability to recognize when the robot is outside the learned workspace, we allowed the robot to leave this learned region and randomly performed 1000 odometry tests as described in Section 2.2. For 748 of these test points the robot recognized that it was outside of the learned region. However, the other 252 test examples, as in the previous experiment were close enough (within 1 meter) to the original workspace boundary to make the learned position mappings still valid within calibration error limits, even though they where outside the learned workspace.

5 Discussion and Conclusion

The experimental results demonstrate that the proposed method generated vision based localization mappings which had errors that were comparable to those found in the calibration system. The learned mappings are robust in that they give consistent localization readings in a dynamically changing visual environment, at a rate of better than 7 position readings per second. The 3 different on board camera positions used in the experiment demonstrate that various visual queues can be used to learn effective localization mappings.

In addition, the experimental results indicate that the localization mappings generated can be used to determine when the mobile robot has strayed from the region where the mapping was learned. We are interested in using this to help automatically map dynamic environments. By knowing that the robot is no longer within a learned work-cell, we can extend the learned visual mappings using odometric readings, and thus begin to visually explore an unknown environment without using an external calibration system.

Finally, although these initial experiments indicate a promising approach to vision based localization, much more experimentation is needed to establish how it can best be used for various environments. Perhaps a better approach for some applications could be to use panoramic camera views, while other applications may best be served by using this method in a sensor fusion framework which incorporates other information such as sonar and odometry. Perhaps the most promising aspect of the proposed approach is that these new sensor inputs, even though they are not pixel values, can be directly incorporated into the nonparametric learning paradigm together with the image data. Thus, sensor fusion becomes an automatic byproduct of the way in which the learning algorithm constructs the localization mappings.

References

[1] D. Burschka, C. Eberst, and C. Robl. Vision based model generation for indoor environments. In *Proceedings ICRA97*, pages 1940–1945, Albuquerque, New Mexico, April 1997. IEEE.

[2] Fabio Cozman and Eric Krotkov. Automatic mountain detection and pose estimation for teleoperation of lunar rovers. In *Proceedings of ICRA97*, pages 2452–2457, Albuquerque, New Mexico, April 1997. IEEE.

[3] J. H. Friedman. An overview of predictive learning and function approximation. In V. Cherkassky, J. H. Friedman, and H. Wechsler, editors, *From Statistics to Neural Networks*, pages 1–61. Springer-Verlag, 1994.

[4] G. Z. Grudic. *Nonparametric Learning From Examples in Very High Dimensional Spaces.* PhD thesis, Electrical and Computer Engineering, University of British Columbia, August 1997.

[5] G. Z. Grudic and P. D. Lawrence. Human-to-robot skill transfer via teleoperation. In *IEEE Int. Conf. on Sys. Man and Cyb.*, 1995.

[6] G. Z. Grudic and P. D. Lawrence. Human-to-robot skill transfer using the spore approximation. In *IEEE Int. Conf. on Rob. and Aut.*, 1996.

[7] G. Z. Grudic and P. D. Lawrence. Is nonparametric learning practical in very high dimensional spaces? In *IJCAI-97*, NAGOYA, Japan, August 1997.

[8] T. Hague, J. A. Marchant, and N.D. Tillett. Autonomous robot navigation for precise horticulture. In *Proceedings of ICRA97*, pages 1880–1885, Albuquerque, New Mexico, April 1997. IEEE.

[9] Joan Martí, Joan Batlle, and Alícia Casals. Mobile-based objects recognition in industrial envirnments for autonomous vehicles. In *Proceedings of ICRA97*, pages 1632–1637, Albuquerque, New Mexico, April 1997. IEEE.

[10] Sabastian Thrun. A bayesian approach to landmark discovery and active perception in mobile robot navigation. *Technical Report, Carnegie Mellon University, School of Computer Science*, CMU-CS-96-122, May 1996.

[11] Henry Wong. Position and orientation sensing system. *Robotics Lab Technical Report*, July 1997.