Proceedings of the 2005 IEEE
International Conference on Robotics and Automation
Barcelona, Spain, April 2005

Autonomous Terrain Mapping and Classification Using Hidden Markov Models

Denis F. Wolf, Gaurav S. Sukhatme
Robotic Embedded Systems Laboratory
Department of Computer Science
University of Southern California
Los Angeles, CA, USA
denis, gaurav@robotics.usc.edu

Dieter Fox
Robotics and State Estimation Laboratory
Department of Computer Science & Engineering
University of Washington
Seattle, WA, USA
fox@cs.washington.edu

Wolfram Burgard
Department of Computer Science
University of Freiburg
Freiburg, Germany
burgard@informatik.uni-freiburg.de

***Abstract*— This paper presents a new approach for terrain mapping and classification using mobile robots with 2D laser range finders. Our algorithm generates 3D terrain maps and classifies navigable and non-navigable regions on those maps using Hidden Markov models. The maps generated by our approach can be used for path planning, navigation, local obstacle avoidance, detection of changes in the terrain, and object recognition. We propose a map segmentation algorithm based on Markov Random Fields, which removes small errors in the classification. In order to validate our algorithms, we present experimental results using two robotic platforms.**

I. Introduction

Autonomous navigation is a fundamental capability for a mobile robot. When traversing rough terrain, the robot must have the ability to avoid not only obstacles but also parts of the terrain that are considered not safe for navigation [1]. In this paper, we present an online algorithm that builds a 3D map of the terrain and classifies the mapped regions as navigable or non-navigable areas. The maps created by our approach have numerous applications such as: local avoidance of non-navigable areas, path planning, and object matching and recognition. We are particularly interested in the first two applications.

Outdoor 3D maps have been addressed by the computer vision community for many years [2] [3] and also more recently by the robotics community [4] [5]. There are diverse applications for 3D site modeling. Those applications range from path planning to urban modeling. Mostly ground robots are used for the mapping task [6] [7], but flying robot platforms have also been successfully used [8] [9].

Different methods have been used to create the 3D representations of the environment such as: point clouds [7], triangular meshes, and planar structures [10]. Our approach uses point clouds to represent the terrain. This method can represent fine details of the environment and it is also easy and fast to compute. On the other hand, it requires considerable memory space to represent large areas.

Part of the 3D outdoor mapping effort by the robotics community is focused on the 3D terrain mapping problem. This is an important problem when one is exploring unknown terrain. Applications for terrain mapping range from path planning and local obstacle avoidance to detection of changes in the terrain and object recognition [11]. Planetary exploration is an interesting example of practical application for this research topic [12].

Different platforms and methodologies have been used for the terrain mapping task. In [13] an incremental mapping algorithm using a 3D laser range finder is proposed focusing on the representation of the uncertainties in the pose of landmarks. The approach presented by [14] builds terrain maps from sequences of unregistered low altitude stereo vision image pairs. An autonomous blimp has been used [15] to build maps of the terrain, also using stereo images. In [16], results of terrain mapping using a walking planetary rover are presented. The approach presented by [17] performs terrain mapping based on multiple sensors. A multi robot approach for terrain mapping is proposed in [18].

Our approach for terrain mapping uses ground robots equipped with 2D laser range finders. The range sensors are mounted pitched down on the robots. As the robot moves, the range information generates a 3D point cloud, which models the terrain. Our algorithm classifies the terrain into two categories: navigable and non-navigable areas. We consider flat parts of the terrain such as walkways navigable areas. These areas are considered safe for navigation. Grass and gravel are considered non-navigable (or less desirable) areas. Although in many cases grass is very flat our algorithm efficiently differentiates those areas from concrete walkways. Depending on the application, different types of terrain may be considered navigable and non-navigable. For example, in a planetary exploration context, areas with large rocks that may damage the robot may be considered non-navigable. Classified maps can be very useful for path planning and safe navigation. Since our algorithm can be executed in real time, a robot can online identify and avoid the unsafe areas performing safe autonomous exploration and mapping.

After the mapping and classification steps, we propose a map segmentation algorithm based on Markov Random Fields [24]. This algorithm has been extensively used in image processing and it eliminates small errors cased by sensor noise or error in the classification making the map more uniform.

0-7803-8914-X/05/$20.00 ©2005 IEEE.

(a) Pioneer AT (b) Segway RMP

Fig. 1
ROBOTIC PLATFORMS USED DURING THE EXPERIMENTS WITH FRONT MOUNTED LASER PITCHED DOWN.

II. MAPPING

Point cloud maps can be generated fairly easily when pose estimation and range information are available [7]. Accurate pose estimation for mobile robots has been an issue for many years [19]. The pose information provided by the robot's internal odometer usually drifts. Over long runs, the error grows without bound making the information provided by the odometer completely wrong. In indoor structured environments, scan-matching techniques have been successfully used to reduce the error in the robot's pose estimation [26]. In outdoor environments, scan matching is not efficient due to the lack of structure. In this case, GPS information can be used as a global reference, but usually the horizontal error present in the information provided by GPS sensor makes it undesirable to precisely estimate the pose for the robot.

In order to have a reasonable pose estimate, we combined the information provided by the odometer and IMU. This does not allow us to correctly close arbitrarly large loops (which is beyond the scope of this paper), but it corrects a considerable part of the drift in the odometric information, providing a consistent local estimate for the pose of the robot.

Two mobile platforms have been used as experimental test beds: an ActiveMedia Pioneer AT and a Segway RMP (Figures 1a and 1b). The information provided by the laser is projected into 3D Cartesian space using standard geometry. As the robots move forward, the 3D range information about the environment is obtained. It is also interesting to notice that due to the dynamics of the RMP, this robot pitches dramatically when accelerating and decelerating. In those cases, the pitch is taken into account when the map is being built.

III. TERRAIN CLASSIFICATION

Our terrain classification algorithm is based on Hidden Markov Models and is divided into two parts. The learning step and the classification step. We classify each point in the 3D map as navigable or non-navigable. Flat surfaces such as concrete walkways are considered navigable areas while grass, gravel, and obstacles are considered non-navigable. At the end of the classification step, the maps generated by our algorithm clearly show the areas where the robot is allowed to navigate (safe areas) and the areas considered unsafe for navigation. Even though some times the difference in the roughness of concrete walkways and grass is very small, our approach is capable of detecting them successfully.

Hidden Markov Modeling is a powerful statistical tool, with many applications (see [20] for a tutorial). An HMM can be defined as follows:

1) N, the number of possible states in the model. Individual states are denoted as $S = s_1, s_2, ..., s_N$, and a specific state at time t as q_t.

2) M, the number of observation symbols per state. The observation corresponds to the output of the system being modeled. Individual symbols are denoted as $V = v_1, v_2, ..., v_M$.

3)The state transition probability distribution $A = a_{ij}$ where:

$$a_{ij} = P(q_{t+1} = s_j | q_t = s_i), 1 < i, j < N$$

4) The observation symbol probability distribution in state j, $B = b_j(k)$, where:

$$b_j(k) = P(v_k^t | q_t = s_j), 1 < j < N, 1 < k < M$$

where v_k^t means the simbol v_k at time t.

5) The initial state distribution $\pi = \pi_j$, where:

$$\pi_i = P(q_1 = S_i), 1 < i < N$$

For convenience, the compact notation $\lambda = (A, B, \pi)$ will be used to characterize a HMM.

One of the general problems that can be solved using HMM is: given the observation sequence $O = O_1, O_2, ..., O_T$, and the model λ, how to compute the optimal correspondent state sequence $Q = q_1, q_2, ..., q_T$ (i.e., the state sequence that best explains the observations)? In a nutshell we are trying to maximize the expression $P(Q|O, \lambda)$. This general problem fits in our context as follows.

The points in our 3D map will be the state sequence. Each scan provided by the range sensor will generate a single state sequence Q. Each point can assume one of two possible states: navigable (A) and non-navigable (U). The real state of each point is not directly given by the range sensor. The sensor's output is range measurement, which has to be converted into observations O.

Navigable areas in our context will be characterized by flat terrains. On those terrains the points generated by the range sensor are expected to be well aligned, with a minimal variance in altitude. Conversely, the non-navigable areas are characterized by rougher terrains. The 3D points that represent those areas are expected to be not well aligned, with some variance in altitude.

The information provided by the range sensor cannot be used directly as observations in our algorithm, since it is only a measurement of the distance between the sensor and the nearest object in some specific direction. Therefore the data provided by the range sensor are represented as a sequence of points in 3D Cartesian space. Given that sequence of points, the observation O_n for a specific point s_n will be the difference in the altitude of s_n compared to the altitude of its neighbor points. In this manner, flat terrains that are not at the elevation as the robot are also classified as navigable. It is important to notice that instead of having a discrete set of observations for individual states, our approach uses continuous values.

The observation symbol probability distribution B is calculated based on the observations O and the covariance of the points classified as A and U . The pdf B is calculated as follows:

$$b_j(k) = N(O_t, 0, \Sigma_k)$$

where $b_j(k)$ is the value of b for the jth point in the scan and k is one of the possible states the point can assume (A or U). The term $N(O_t; 0, \Sigma_k)$ corresponds to the value of O_t applied to a gaussian function with zero mean and Σ_k as covariance. The term η is a normalizing factor.

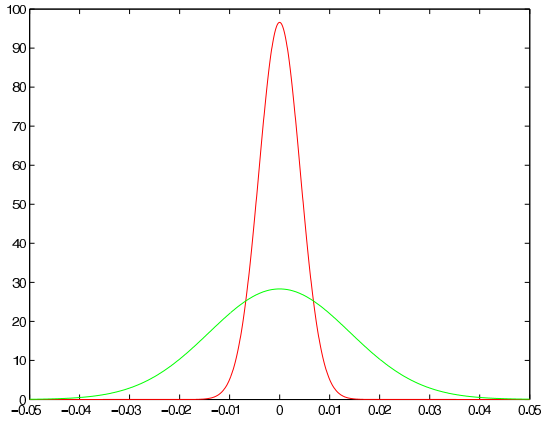


Fig. 2
GAUSSIANS FOR A AND U POINTS.

One can use HMM also to learn the parameter of the model λ, but in our case, the learning can be done fairly easily through the use of examples. Given a set of range scans, it is possible to manually label the true state for each point on each scan. Given these labeled data, the calculation of the state transition probability distribution A is straightforward. As we have only two possible states (A and U), it consists of counting the number of times that a point labeled A is followed by a points labeled A or U. The same rule applies for points labeled U. The numbers must be normalized so that the probability distribution sums to 1.

Calculating the initial state distribution π is also easy when labeled data are available. The number of states labeled as A and U must be counted and the distribution also needs to be normalized. Based on the labeled data, it is also possible to calculate the variance for the points classified as A and U. In this case, it is necessary to calculate a mean for the altitude in the points that belong to a specific state. After that it is necessary to calculate the amount of variation in the altitude of those points in comparison to the mean. Figure 2 shows the gaussian pdfs for the points A and U.

Now, as all the variables are already defined and the HMM is defined for our context, it is necessary to find the solution for the stated problem. The solution will be the sequence Q, which maximizes the expression $P(Q|O, \lambda)$. Maximizing $P(Q|O, \lambda)$ is equivalent to maximizing $P(Q, O|\lambda)$. In our context, the solution will be the best classification for the points of each scan that maximizes the observations provided by the range sensor.

The solution for the problem above can be acquired using the Viterbi Algorithm [21], which is based on dynamic programming techniques.

In order to find the single best state sequence, $Q =$

$q_1, q_2, ..., q_T$, for a given observation sequence $O = o_1, o_2, ..., o_T$, we define the quantity

$$\delta_t(i) = maxP(q_1, q_2, ..., q_t = i, O_1, O_2, ..., O_t|\lambda)$$

where $\delta_t(i)$ is the best score (highest probability along a single path, at time t, which accounts for the first t observations and ends in state S_i). By induction we have

$$\delta_{t+1}(i) = [max\delta_t(i).a_{ij}].b_j(O_{t+1})$$

In order to retrieve the state sequence, we need to keep track of the argument which maximized the previous equation, for each t and j. We do this via the array $\psi_t(i)$. The complete procedure for finding the best state sequence can now be stated as follows:

Algorithm 1 Viterbi Algorithm

1) Initialization:
$\delta_1(i) = \pi_i b_i(O_i)$
$\psi_1(i) = 0$

2) Recursion:
$\delta_t(i) = max[\delta_{t-1}(i).a_{ij}].b_j(O_t)$
$\psi_t(i) = argmax[\psi_{t-1}(i).a_{ij}]$

3) Termination:
$p^* = max[\delta_T(i)]$
$q_T^* = argmax[\delta_T(i)]$

4) Path backtracking:
$q_t^* = \psi_{t+1}(q_{T+1}^*), t = T-1, T-2, ..., 1$

After applying the Viterbi algorithm to each range scan, all the points in our map will be classified as A or U.

IV. Map Segmentation

It may happen that parts of the map are not correctly classified due to sensor noise, presence of spurious objects (like leaves) in the flat parts of the terrain, or just errors in the HMM classification. When those errors are small parts of the map (they may be considered noise), segmentation techniques can be used to fix them.

Segmentation techniques have been used for many years by the computer vision community [23]. Among several segmentation methods, Markov Random Fields (MRF) have been extensively used in image processing. For a complete overview of MRF theory see [25].

It is unlikely that in an area with a large majority of cells A there are few cells U. It is also unlikely that among several scans that contain both points labeled A and U there is a scan that only contains points labeled A. The MRF technique make those points agree with their neighbors As a result, A and U regions are well defined and clustered.

In order to use MRF as a segmentation tool, we approximate our 3D map to a 2D grid. Each point is projected on the grid based on its x and y coordinates. Each cell in the grid is labeled as A or U according to the classification of the points projected on that cell. It is important to notice that not all the cells in the grid have a label, because the distribution of points in the xy plane is not uniform.

The basic idea of MRF is that the probability distribution for each cell in the grid is specified conditionally on the probability distribution of its neighbor cells. After the application of the filter, all points projected on each cell are labeled with the same label of that grid cell.

Let C_i be a random variable taking the values A or U, and denote by $n_i^{(k)}$ $(k = 1, 2, ..., n)$ the number of k neighbors of C_i that are labeled as A. A simplified MRF model may be specified as:

$$\frac{P(C_i = A|grid)}{P(C_i = U|grid)} = exp\Big(\alpha + \sum_{k=1}^{n} (\beta_k n_i^{(k)})\Big)$$

where α and β are respectively the prior about the number of A cells and the importance ratio based on the distance to the cell C_i. As β is increased the chance of each grid cell value agree with the value of its neighbors increases.

V. Experimental Results

In order to validate the approach presented in this paper, extensive experimental tests have been performed. Our experiments have been done using both an ActiveMedia Pioneer AT and a Segway RMP robots. Both robots were equipped with SICK laser range finders and a Microstrain IMU. Player [22] has been used to perform the low level control of the robots.

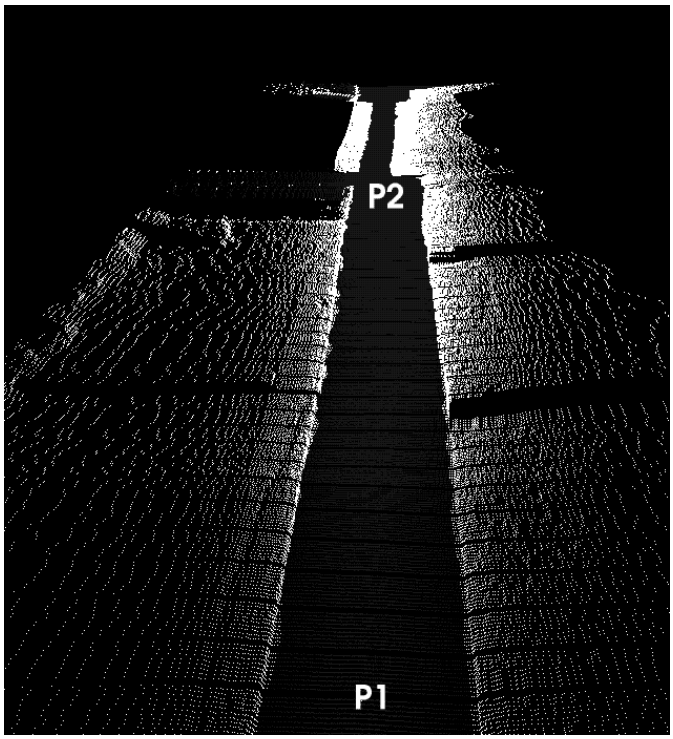
On the Pioneer platform, the laser sensor was mounted at 42cm height and a pitch angle of 40. On the RMP the laser was mounted at 93 cm height and a pitch angle of 35, which allowed the robot to map the terrain approximately 1.3m ahead of the robot.

Our experiments have been performed in different parts of the USC campus. Although the grass was very flat in some terrains, the robot could successfully differentiate grass from concrete. Figure 3a shows the actual environment (approximately 50m long) and Figure 3b shows the 3D model where A and U areas are correctly classified. The Walkway (A points) are colored grey and grass and bushes (U points) are colored white. The points P1 and P2 are used to represent corresponding places in Figures 3a and 3b.

During the experiments, most of the maps have been built when the robots were manually driven with a joystick. But some autonomous navigation experiments using the RMP have also been performed. As our mapping and classification algorithm can be executed in real time, the robot could online use the information about the areas it should avoid and it kept itself in the navigable areas while autonomously mapping the environment. A small movie of the autonomous navigation can be found at:
http://robotics.usc.edu/~denis/research/hmm_nav.avi.

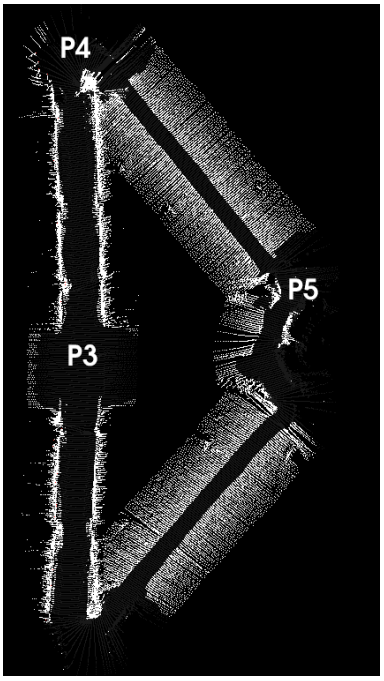
The maps created using HMM classification present some errors. In those cases, despite the roughness of the terrain,

(a) Walkway



(b) 3D Map of the Walkway

(c) Doheny Library garden



(d) 3D segmented map of the garden

Fig. 3
EXPERIMENTAL RESULTS.

many points on the grass align with their neighbor points, resulting in mistakes in the classification. This type of error is more likely to happen with points that are distant from the robot. This is because due to the nature of the SICK laser sensor, the representation of those areas is more sparse. Figure 3c shows a part of Doheny Library garden, this environment includes walkways, grass areas, and bushes. The 3D terrain map after the segmentation step can be seen in Figure 3d. Most of the classification errors are removed. The points P3, P4, and P5 are used to represent corresponding places in Figures 3c and 3d.

VI. Conclusion and Future Work

We have proposed an algorithm to build 3D terrain maps using mobile robots and range sensors and to classify areas in the map as navigable or non-navigable. Our approach for classification is based on Hidden Markov Models and it is efficient enough to differentiate concrete flat terrains from grass, gravel and other types of obstacles. The maps generated by our algorithm can be used for path planning and local obstacle avoidance, and as our algorithm can be executed in real time, it allows the robot to autonomously navigate avoiding unsafe (non-navigable) areas and obstacles. We also proposed a map segmentation algorithm based on Markov Random Fields, which removes small classification errors from the maps.

As future work, we propose more complex models to represent the terrain. Instead of only two possible states (navigable or non-navigable), our maps could have different models for grass, gravel, depressions, and other obstacles. Combining different sensors such as range information and images is also part of our future work. As these different sensors provide different types of data, they could be combined in order to obtain a better representation of the environment.

VII. Acknowledgements

The authors thank Jonathan Ko for his support during the initial experiments at the University of Washington. This work is supported in part by DARPA MARS contract DABT63-99-1-0015, Gaurav Sukhatme's NSF career grant IIS-0133947, and by the grant 1072/01-3 from CAPES-BRAZIL.

References

[1] Borenstein, J. and Koren, Y., "The Vector Field Histogram - Fast Obstacle-Avoidance for Mobile Robots." *in IEEE Journal of Robotics and Automation, Vol. 7, No. 3., pp. 278-288, 1991.*

[2] Hu, J., You, S., Neumann, U. "Approaches to Large-Scale Urban Modeling," *In IEEE Computer Graphics and Applications,V ol.23, No.6, pp. 62-69, 2003.*

[3] Fruh, C., Zakhor, A. "Constructing 3D City Models by Merging Aerial and Ground Views," *In IEEE Computer Graphics and Applications ,Vol.23, No.6, pp. 52-61, 2003.*

[4] Thrun, S., Burgard, W., Fox, D. "A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping.' *In Proceedings of the IEEE International Conference on Robotics and Automation, pp. 321-328, 2000.*

[5] Brenneke, C., Wulf, Oliver, Wagner, B. "Using 3D Laser Range Data for SLAM in Outdoor Environments.' *In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, Las Vegas, pp. 188-193, 2003*

[6] Montemerlo, M., Thrun, S. "Large-scale robotic 3-d mapping of urban structures." *In Proceedings of the International Symposium on Experimental Robotics (ISER), Singapore, 2004..*

[7] Howard, A., Wolf D. F., Sukhatme, G. S., "Towards 3D Mapping in Large Urban Environments" *To appear in IEEE/RSJ International Conference on Intelligent Robots and Systems, Sendai, Japan, pp. 419-424, 2004..*

[8] Thrun S., Diel, M., Haehnel, D. "Scan Alignment and 3D Surface Modeling with a Helicopter Platform," *In Proceedings of the International Conference on Field and Service Robotics, Lake Yamanaka, Japan, 2003.*

[9] Miller, R., Amidi, O. "3-D Site Mapping with the CMU Autonomous Helicopter," *In Proceedings of the 5th International Conference on Intelligent Autonomous Systems (IAS-5), pp. 765-774, 1998.*

[10] Thrun, S., Martin, C., Liu, Y., Haehnel, D., Montemerlo, R. E., Chakrabarti, D., Burgard, W. 'A Real-Time Expectation Maximization Algorithm for Acquiring Multi-Planar Maps of Indoor Environments with Mobile Robots," *In Transactions on Robotics and Automation, Vol.20, No.3, pp. 433-442, 2004.*

[11] Ye, C., Borenstein, J. "A new terrain mapping method for mobile robots obstacle negotiation," *Proceedings of the UGV Technology Conference at the 2003 SPIE AeroSense Symposium, Orlando, pp. 52-62, 2003..*

[12] NASA "NASA Space Telerobotics Program," *in http://ranier.hq.nasa.gov/telerobotics_page/Technologies/0240.html, visited in September, 2004.*

[13] Betge-Brezetz, S., Hebert, P., Chatila, R., Devy, M. "Uncertain map making in natural environments." *In Proceedings of IEEE International Conference on Robotics and Automation (ICRA'96), Minneapolis, pp.1048-1053, 1996.*

[14] Jung, I. K., Lacroix, S. "High resolution terrain mapping using low altitude aerial stereo imagery." *In Proceedings of IEEE International Conference on Computer Vision (ICCV'2003), Nice (France), pp. 946-951, 2003.*

[15] Lacroix, S., Jung, I. K. "High resolution terrain mapping with an autonomous blimp." *In Proceedings of International Conference on Intelligent Robotics and Systems. Lausanne (Switzerland), pp. 781-786, 2002.*

[16] Krotkov, E., Hoffman, R. "Terrain Mapping for a Walking Planetary Rover," *In IEEE Trans. Robotics and Automation, Vol. 6, No. 10, pp. 728-739, 1994.*

[17] Kweon, I., Kanade, T. "High-Resolution Terrain Map from Multiple Sensor Data." *In IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 14, No. 2, pp. 278-292, 1992.*

[18] Fregene, K., Madhavan, R., Parker, L. E. "Incremental Multiagent Robotic Mapping of Outdoor Terrains." *In Proceedings of IEEE International Conference on Robotics and Automation, pp. 1339-1346, 2002.*

[19] Fox, D., Burgard, W., Dellaert, F., Thrun, S. "Monte Carlo localization: Efficient position estimation for mobile robots." *In Proceedings of the National Conference on Artificial Intelligence (AAAI), Orlando, pp. 343-349, 1999.*

[20] Rabiner, L. R. "A tutorial on hidden Markov models and selected applications in speech recognition." *In Proceedings of IEEE, vol. 77, no. 2, pp. 257–286, 1989.*

[21] Forney Jr., G. D. "The Viterbi Algorithm." *In Proceedings of the IEEE Vol. 61, No. 3, pp. 268-278, 1973.*

[22] Gerkey, B. P., Vaughan, R. T., Stoy, K., Howard, A., Sukhatme, G. S, Mataric', M. J. "Most Valuable Player: A robot device server for Distributed Control." *In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1226-1231, 2001.*

[23] Forsyth, D. A., Ponde, J. "Computer Vision: A Modern Approach." *Prentice Hall, 1st. edition, 2002.*

[24] Nasab, N. M., Analoui, M., Delp, E. J. "Robust and efficient image segmentation approaches using Markov random field models." *In Journal of Electronic Imaging Vol.12, No.1, pp. 50-58, 2003.*

[25] Kindermann, R., Snell, J. L. "Markov Random Fields and Their Applications." *Contemporary Mathematics, American Mathematical Society, 1980.*

[26] Gutmann, J.-S. and Schlegel, C. "Amos: Comparison of scan matching approaches for self-localization in indoor environments. " *In Proceedings of the 1st Euromicro Workshop on Advanced Mobile Robots (EUROBOT?96), pp. 61-67, 1996.*