Current challenges in humanoid navigation in dynamic environments

Ricardo Tellez and Luca Marchionni

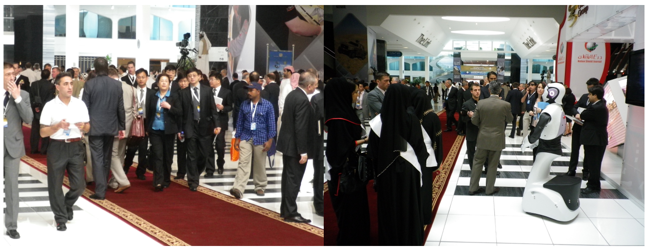

Figure 1. ADNEC exhibition center in Abu-Dhabi during IDEX exhibition in 2011. Size of the place is about 80.000 squared meters. It includes different types of floors, carpets and different flows of people at the same time.

***Abstract*—This paper describes current limitations of humanoid navigation systems in real human environments for all the parts of a navigation system: mapping, localization, path planning and obstacle avoidance. The identification of those problems comes from our experience in the deployment of the Reem series of humanoid robots in exhibition centers and shopping malls. A list of the latests solutions to those problems is also provided, even if they are partial and no holistic solution solves all the issues yet.**

I. Introduction

The ultimate goal of humanoid robot navigation is to make the robot move by itself (navigate) on a human environment, without any human intervention, and for a long period of time. By human environment we understand a place where humans move around, each one engaged in their own activities. The robot may or may not interact with people, but they will be moving around the environment, altering the environment, and performing actions that will prevent the robot from accomplishing its mission (like for example standing on the robot way).

In order to navigate, a robot has to create a map of the environment, be able to locate itself on the map, and plan safe trajectories on it. It also has to be able to avoid obstacles on its way to the goal as they appear. There exists a lot of different navigation techniques to accomplish these goals, but when those techniques are deployed on a real robot on a real human environment they have multiple limitations. The reason is that human environments are not static and simple environments. Instead, they are highly dynamic, they contain multiple exceptions to the main setup, are large and include people in their environment (see figure 1).

Some attempts of robot navigation in human environments have already been made with non humanoid robots in museums [1], exhibitions [2], shopping malls [3] or hospitals [4] but the degree of autonomy is far from the level desired. In all those cases, human supervision was required on a regular basis, either to remove special obstacles from the robot way, or to re-localize it. Humanoid navigation (biped and wheeled) in static environments has been shown in some works [5], [6], [7], [8], [9], [10], but none of them attempted a deployment on a real human environment.

Using human size humanoid robots makes it even more complicated because the robots are big, have special shapes that can collide with objects in a lot of different ways, can crash against people and harm them, and the human expectations over them are greater.

Based on our experience deploying Reem-B biped humanoid robot [11] in office environments, and Reem-H and REEM wheeled humanoid robots into different human setups like shopping malls and exhibition centers, we have identified the current major problems that prevent the use of humanoid robots in human environments, from the point of view of navigation systems. This paper describes those problems in each of navigation areas, including current approaches to partially solve them. The paper ends with the conclusions and future work section.

II. Problems in map creation

Mapping is the ability the robot has to create a representation or model of the environment that will later be useful for it in order to know where it is located and where does it have to go. The typical use of SLAM algorithms for mapping works as follows: there is an initial mapping phase where usually a human operator moves the robot around the environment and creates the map with it. Then this map is frozen and used for both localization and path planning, usually assuming that the world will remain unchanged.

This approach to map creation (and later use) introduces the following problems:

A. The autonomous creation of the map

The typical way to create a map is to provide the robot with a range sensor and an odometry system. Then, while the human moves the robot around the environment a SLAM algorithm processes the sensor readings to generate a map. Hence, map creation is the first moment in the robot deployment process where human assistance is needed.

Even if this human intervention can be considered as part of the setup process, when the environment is large (more than 10.000 m2), this process is very annoying for humans, specially when extra information has to be incorporated in the map as it is been built (like capturing images at specific places or recovering RFID information; all those tasks may require to

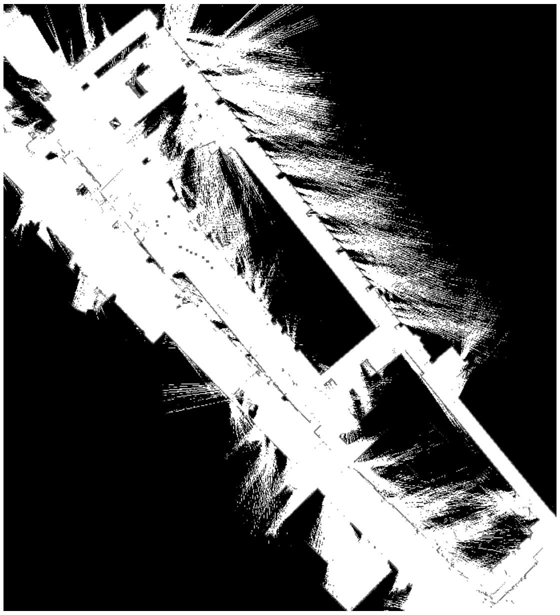

Figure 2. Automatic loop closure failed in the left-upper part of the map (size of the loop 50x10m). At the same time, debris due to ephemeral obstacles can be observed in the main corridor.

stop the robot making the whole mapping process a lot more exhausting).

In order to solve this problem *autonomous exploration* techniques can be used to allow the robot do the map by itself [12], [13]. Those techniques are mainly concerned about distinguishing which areas the robot has to visit next, and which ones are not necessary.

From an automatic mapping process, one would expect not to have to manually modify the map once it has been created by the robot. However, manual modification of the map may be required in some situations where the final map includes errors.

Errors in map creation can be due to several things. One example is failing to close a loop. In this situation, the automatic loop closure algorithm can fail closing it, and the result is a map with overlapping zones (see figure 2).

Another error in map creation can be due to ephemeral objects [14]. Those are objects that are in the environment during map creation (hence they will be included in the map) but they will not be in the environment during normal operation of the robot. Those objects include people moving around the robot while mapping, garbage, or doors that are closed when the robot moves (see debris in figure 2).

The effect of moving people can be solved in most of the cases, solutions range from using probabilistic filters to identify dynamic measurements [15], [16], or by tracking the moving objects [17], [18], [19]. The problem of people standing without moving has the same effect as stuff put in the place for a moment (like for example a suit case). This problem is due to the existence of *static temporal obstacles*, those are, obstacles that stay on a position of the environment long enough to be considered as part of the environment, when actually they are not.

The problem of introducing static temporal obstacles on a map is not solved yet. A practical solution to avoid them is to create the map when nobody is there paying special attention to the environment been clean, but this is not always possible to do. Hence, a better automatic solution may be required for this problem. In the next section, solutions for mapping over long periods are described. Those solutions can in some sense solve the problem of static temporal objects.

Summarizing, at present, the existence of the described problems requires a person to check the map automatically created by the robot and modify it accordingly.

B. The maintenance of a map over long periods of time

Maps change with time. People add new stuff to the environment, move stuff from one place to another, or just remove it. Sometimes, they change partially or completely the setup, like for example in exhibition centers, where every exhibition has its own layout. If the robot has to run by itself for long periods of time without requiring new setups from humans after the initial deployment, it must be able to detect those changes and re-arrange the map, either by modifying the current one or by creating a new one. This process is called long term mapping or *life long mapping*.

Just a few works have dealt with this problem:

The work of [20] introduces a method to maintain a map over time by representing the environment over multiple timescales simultaneously. By using this method, the authors are able to track both stationary and non-stationary elements of the environment. Their method allowed them to maintain the maps over several weeks.

In [21] the authors deal with the problem of changes in the long time, like for example places with doors or sofas been moved, or panels that change position. Proposed a solution to based on constructing sub-maps for each area where those and other dynamic aspects were observed. For example, the robot is able to detect the door as a dynamic obstacle by observing it in different configurations at different times. This technique can be used to remove static temporal obstacles of a map.

In [14] dealt with the problem of maintaining a limited amount of resources along the life of the robot in dynamic environments, by keeping reduced the size of the pose-graph. Since the number of observations increases with the time that the robot is running, it is a common problem to decide which observations to store and which ones don't, in order to maintain a limited amount of resources. This work only maintains those measures that present a gain in the expected information. Additionally, their approach improves from the previous ones in the sense that it can update the map after a localization failure, or when new large sections of a map have to be incorporated or changed.

All those approaches have demonstrated to work, but it is difficult to assert how they would work when used in the presence of a crowd (like in a shopping mall or an exhibition center) or in large environments (for instance, [14] work was demonstrated for only 2500 squared meters).

C. The mapping of large environments

Another problem of current mapping systems is the mapping of *large indoor environments* over which the robot will have to localize and generate path plans. We describe as large environments, environments of 10.000 square meters or more. Even if the standard SLAM approach of mapping such environments may work [22], [23], they become extremely inefficient when the map grows in size, specially when combined with other navigation requirements, like path planning, initial localization or visual localization. Additionally, the problem of closing loops becomes more complex due to the necessity of maintaining a larger number of hypothesis for loop closure.

The problem of large cyclic environments was discussed in [24]. In [25] the problem of closing loops is minimized by introducing the use of a Pose-SLAM method which only keeps non redundant poses and highly informative links. In [26] their system is able to close different nested loops along more than 2 km, by using a graph that encodes local frames (local submaps) with a transformation between adjacent frames (that is, between adjacent sub-maps).

Several works are exploring the same idea of constructing sub-maps that are connected in some way between each other (with different types of constrains), in order to generate a large map. For instance, [27] present a hierarchical method based on two levels, local and global, with an estimation of the relative locations between local maps. The authors also provide a method for closing large loops, but the whole system has only been tested in environments up to 10.000 square meters. Similar works are those of [28], [29].

Recently [30] have developed hybrid topological-metric approach maps based also in sub-maps that are continuously been updated. This continuous update helps to introduce changes in the map, like stuff that is removed, removal of static temporal objects, etc. The approach seems very promising for achieving life long mapping, even if the authors do not provide the size of maps they have constructed.

III. Localization problems

Localization is the ability to determine in which point of the map the robot is. Usually the robot captures data from the environment using a scan sensor (camera, laser, sonar, etc) and matches that data with the map it has constructed in the previous stage. Localization in static small environments has already been demonstrated for biped humanoids with several degrees of success [6], [8], [9], [10].

We observe the following problems when the environment is large and is populated with people:

A. The problem of initial localization

When the environment is crowded, an autonomous initial localization may be very difficult if not impossible. The initial localization problem is a simpler version of the kidnapping problem. There are several solutions to provide an initial localization [31], [32], [33] but they fail when the environment is crowded because the pose is not able to converge to the correct location. Due to wrong measurements related to the presence of people, the current observations of the robot do not match any place in the map (the robot doesn't localize), or even worst, match a place that is not the real one (the robot localizes in the wrong place). In those cases, only a manual set by the operators is the solution.

Figure 3. Four different images obtained by REEM robot at IDEX exhibition (its head in normal operation position). It can be observed that a large amount of visual localization information can be obtained by paying attention to the signs at the upper part of the images, which is almost never occluded.

Other approaches like [34], [35] exploit natural features like corners or walls to initialize the robot position, but they are not taking into account occlusions.

In [36] they solve the problem of occlusion by using indistinguishable artificial landmarks at a certain height that are difficult to be occluded by people. This implies a modification of the environment where the robot has to work accordingly. Additionally, the authors require other strong assumptions that may not be valid in real environments, like for example the number of visible landmarks that need to be provided to the algorithm.

None of the previous solutions would work on a place like the one depicted in figure 1 due to occlusions. From our experience we think that to solve this problem we have to base the initial localization in vision system, since they are able to capture more and better information than any other sensor. For example, figure 3 shows different images captured by humanoid robot REEM robot on a big exhibition. One can observe how a laser would be occluded in all those situations. Instead, vision can perceive the environment, specially all the upper part of the image that includes the signs of all the stands. This approach has at least two advantages:

- The visual information provided by the upper part of the images suffers less from dynamic obstacles occlusion.
- This approach doesn't require a special robot shape to accommodate special sensors in special configurations (like cameras or lasers pointing to the ceiling). Just two cameras mounted on its face in a human like configuration.

In this line of research, is the work of [33], [37] who used vision to identify natural landmarks. However, their work is not prepared for crowded environments, since it focuses in the whole image are which would be cluttered with people on a crowded environment. Another idea is the use of place recognition [38] even if the same problem of crowdedness is there.

The conclusion we draw from this section is that the localization algorithms are mature enough for simple human environments (they would work at home, for example), but they aren't robust enough for large and crowded environments.

B. The problem of maintaining localization in crowded environments

In this section, we assume that the robot has already been localized in some way, either using one of the approaches of the previous section, either made in a manual way by an operator.

Current localization systems are quite robust when the environment is fixed and corresponds to the map created in the previous stage. However, in real life, this is almost never the case. In real cases, the robot is surrounded by people that prevent the sensors match the map. At the same time they provide wrong sensor measures that intoxicate the localization algorithm. The usual solution is to filter this information and provide only good sensor lectures to the localization algorithm [1]. The problem is that, when the *crowdedness ratio*[1] is large, the number of good sensor data provided to the algorithm is very small.

There exists almost no works that address this problem. The only works that address this problem are the early works of [39] with their museum robot. Their solution was to use a camera pointing to the ceiling. All proposals that have a good enough behavior in crowded environments use the ceiling approach [40]. But this ceiling approach includes some restrictions on the shape of the humanoid robot in order to accommodate those cameras. Additionally, it may not work on places where the ceiling is very high, like the ones in the exhibition centers, shopping malls, train stations or airports, which usually range from 15 meters or have no ceiling at all.

Given that in this situation the robot starts from an already localized status, some works plan specific trajectories in order to maximize localization accuracy (that is, minimize localization error). For instance, [41] use a belief map built during map creation to plan trajectories that avoid places where the robot had a greater uncertainty on its localization. [42] generate paths to goals in a similar way, moving the robot through locations that the robot predicts to have less localization uncertainty.

Those approaches are based on the idea that landmarks required for localization will easily be retrieved once the robot path has been properly estimated through a minimum localization error path [43]. Even if this idea is original and may work in complex static places, still has to be demonstrated to work in dynamic environments where crowds never behave the same way. This means that estimations based on measures at a different time may not work at current time.

Recently some improvement on the localization ratio can be observed with the use of feasibility grids [44]. In this case, the authors create a model of the sensor for both stationary and moving objects and used them to create a feasibility map to be used in situations of crowded localization. Comparison against other methods like filtering is provided and show that feasibility grids are a better option for localization in crowded environments, even if no information is provided about the amount of people that is in the environment (no measure of crowdedness).

[1]We call the crowdedness ratio, the ratio between the number laser rays that impact on non-expected obstacles and the total number of rays. This value is always a number between zero and one.

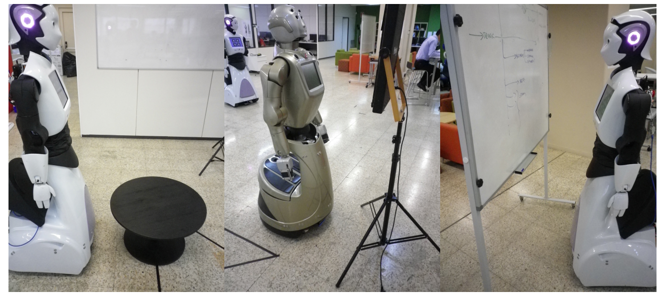

Figure 4. Different types of objects that we have encountered in the deployment of our robots Reem-H and REEM, that are difficult to detected (a small table, the foot of a light projector and a white board).

Additional solutions include the use of external devices that provide some kind of localization signal to the robot, like special marks on the environment (either visual or electromagnetic) [45], [46], [47], install additional beacons [48], or use indoor wifi localization systems [49]. All of them require a modification of the environment where the robot is deployed, as well as modify the robot itself to incorporate additional sensors. This solution makes the installation of the robot more complicated specially if the environment is large.

IV. Problems avoiding obstacles

Once the robot is localized on a map, it has to move safely in the environment. For a humanoid robot safe movement is difficult due to its size, special shape (arms can move outside the main body interfering with the sensors) and even the special way of moving (balancing, moving arms along the body, etc). Also, the obstacles one can find in human environments are of very different ways, shapes, and materials (see figure 4).

For humanoids, detecting obstacles is even more complex than with mobile bases for several reasons. First, the sensors need to be mounted on a reduced space while maintaining a certain shape (human shape). This fact limits the type of sensors that can be used and the number of them that can be incorporated in the robot. Second, depending on where in the robot the sensor is mounted, the noise associated to the sensor measure can become very large. Every physical link between parts of the robot in the path from the base of the robot to the sensor position, introduces a certain position error. As the number of links increase, so does the error associated to the sensor. In [6] the HRP robot uses a laser on the mouth to detect obstacles, but due to its imprecision, it is only used for object recognition and not for obstacle detection. In [50] a pivoting laser mounted on the humanoid hip is used to detect obstacles. Due to the errors accumulated over time, the memory of obstacles is deleted after every few steps.

So the main problems we devise in avoiding obstacles are described in the following sections.

A. *Obstacle detection*

Obstacle distance detection is the most important step of an obstacle avoidance system. Distances to obstacles are obtained using different methods and sensors. The fusion of all the available sensors provides the robot with a description of the obstacles in its surroundings. Then a planning can be applied to avoid those obstacles.

Current approaches to obstacle detection are far from perfect: for instance, in [8] two lasers on the robot feet were used to calculate the distances to obstacles. In [51] the PR2 robot used a tilting laser close to the mouth to generate a *cloud of points* to obstacles. Both approaches used lasers as their main obstacle detection device, being the second approach more complete because generates a 3D cloud of obstacles points. However, laser sensors have their limitations in detecting objects that one can found in a normal human environment like glass doors and tables, mirrors or metallic stuff. And even worst, when the laser is in front of mirrors or metallic parts, the distances provided by the sensor are longer than the actual ones (because of the reflexion effect).

Recently robots are being equipped with structured light scanner devices that improve the number of obstacles that can be detected by a single sensor, even if glass can not still be detected by those devices. Hence, the current solution to obstacle detection seems to have a combination of sensors that allow in the whole the detection of the greatest number of types of objects (laser, sonar, cameras, structured light scanners). Again, this solution has its own problems; for example, to include all those sensors on a humanoid robot may be difficult. Also, many sensors on a single robot may generate interferences between them (for example, sonars in the back may interfere with sonars in the front), and even worst, interfere with other robots.

B. *Handle strange situations*

If the robot has to move by its own for a long period of time on a crowded environment, it will have to handle several non common situations. From our experience we have identified the following:

1) Robot collisions: It is almost impossible to devise before hand all the types of objects and situations that a robot will encounter during its operation on a human environment. This means that, even if the robot is perfectly sensored to detect obstacles, eventually the robot will collide with something.

The robot must be prepared for this situation in two ways: first it has to be equipped with contact sensors like bumpers or artificial skin to detect the impact. Second, because it is impossible to include those sensors in the whole body of the robot, software detectors of collision situations must be incorporated in the robot. This software must be able to: first identify possible collision situations in the future (predict them), like in [52], and second, in case that everything failed and the robot finally collided, determine that the robot has collided based on its history and current behavior rather than on the value of a sensor, like in [53].

2) Interference between robots: It may happen that several robots have to work together in the same environment. In this situation, robots can interfere each other's sensors, producing detection of non existing obstacles, or even worst, not detecting obstacles that do exist in their paths.

Solutions to this problem may range from avoiding the use of sensors that can be interfered (like sonars or lasers) or minimize interference by partitioning the space using different strategies, like in [54], [55].

V. Path planning problems

In our experience we have found path planning in dynamic environments the most mature of the navigation systems. However, there are still some points where path planning can be improved.

A. *Planning in large environments*

Finding paths using large maps can be very time consuming. To avoid this problem, latest path planning techniques are integrated with life long mapping algorithms. Most of the mapping solutions presented in section II-C about mapping large environments already deal with path planning in this way. The typical solution is to have a master plan between large zones of the map, that is update only once every time the robot changes sub-map. Then local planning on each sub-map is made in order to follow the master plan.

Care has to be taken when planning in special situations. Those include the robot changing from one room to another room or the robot turning a corner. In those cases, the robot has no sensor data about what can expect after the door or the corner. Hence a special plan is required. Examples of plans for those situations can be found in [51].

B. *People aware navigation*

The classical approaches to robot planning assume again the world as a static environment. Hence at every time step the world is sensed and the position of obstacles updated. Then a path is calculated based on both the map and the obstacles. The result of this approach in dynamic environments is a behavior of the robot that looks chaotic, because the robot is continuously changing its path based on the flow of people. To avoid this behavior, the path planner has to plan being aware of the existence of different types of obstacles: dynamic (basically, people) and static ones. This type of path planning is called *people aware planning* [56]. Having such type of planning would provide the robots with a more fluent movements, and at the same time, more enjoyable and predictable for humans.

People motion behavior prediction is used in [57], [58] to adapt the path of the robot according to the activities of the people. Other recent works negotiate paths in a general way, independently if the moving obstacle is a person or something else that moves (a pet, a bicycle, a car, etc...) [59]. The work of [60] works on planning in dynamic large environments like train stations or airports been aware of people position.

Recent work for path planning includes robot movement on environments with deformable objects like curtains [61].

In this case, or in similar cases where the space is filled with deformable stuff, it may be interesting to notice that a collision free path may be not possible, and allow a robot path that collides under a specific contact threshold [62].

C. Taking profit of humanoid shape

Most of the path planning algorithms used in humanoid navigation assume the dimensions of the robot like a projection in 2D of the most external part of the robot. They do not take advantage of the fact that the humanoid is not a cylinder with the same shape at all heights. If the special shape and mobility of a humanoid are taken into account, the robot can provide a valid movement path on situations where a non humanoid robot would remain stacked.

For example, [63] takes advantage of the humanoid shape of an HRP2 robot to move along obstacles helping itself with the furniture in the room. Examples include the robot resting on a desk with an arm to move over an obstacle, or moving on a narrow space between a chair and a table. On a similar line of research [64] makes use of the special shape of a PR2 robot to navigate on cluttered environments.

VI. Conclusions

Even if navigation of humanoid robots in controlled environments can be considered solved in a large amount of cases, its application to real, dynamic, human environments is not. Partial solutions to those problems have been described, but no complete solution that solves them all is still available. The current tendency is to combine different solutions into a whole navigation system, like it is shown in the mapping+localization+path planning solutions of section II-C.

We identify as perception one of the biggest challenges in humanoid navigation, since the perception system is the one that allows the robot to detect the spaces where the robot can safely move. Additionally to the sensor system, the robot has to accommodate systems that infer collisions and react accordingly.

We haven't described mechanical problems that do also appear when deploying robots in real environment and that limit enormously the navigation solutions that can be provided theoretically. A comparison of what can be done theoretically with navigation software, and what can be physically done due to the mechanical limitations of a robot, is work for the future.

References

[1] W. Burgard, D. Cremers, Armin B. Fox, G. Hähnel, Dirk Lakemeyer, W. Schulz, Dirk Steiner, and S. Thrun, "Experiences with an interactive museum tour-guide robot," *Artificial Intelligence*, vol. 1-2, no. 114, 2000.

[2] G. Kim, W. Chung, K. rock Kim and M. Kim, "The Autonomous Tour-Guide Robot Jinny," in *International Conference on Intelligent Robots and Systems*, 2004.

[3] A. B. H.-M. Gross, H. Boehme, Ch. Schroeter, S. Mueller, A. Koenig, E. Einhorn, Ch. Martin, M. Merten, "TOOMAS: Interactive Shopping Guide Robots in Everyday Use - Final Implementation and Experiences from Long-term Field Trials," in *International Conference on Intelligent Robots and Systems*, 2009.

[4] Aethon, "http://www.aethon.com."

[5] J. Chestnutt, M. Lau, G. Cheung, J. Kuffner, J. Hodgins, and T. Kanade, "Footstep Planning for the Honda ASIMO Humanoid," *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 629–634.

[6] S. Thompson, S. Kagami, and K. Nishiwaki, "Localisation for Autonomous Humanoid Navigation," *2006 6th IEEERAS International Conference on Humanoid Robots*, pp. 13–19, 2006.

[7] S. Lengagne, N. Ramdani, and P. Fraisse, "A new method for generating safe motions for humanoid robots," *Sophia*, pp. 2–7, 2008.

[8] R. Tellez, F. Ferro, D. Mora, D. Pinyol, and D. Faconti, "Autonomous Humanoid Navigation Using Laser and Odometry Data," in *International Conference on Humanoid Robots*, 2008.

[9] J. Ido, Y. Shimizu, Y. Matsumoto, and T. Ogasawara, "Indoor Navigation for a Humanoid Robot Using a View Sequence," *The International Journal of Robotics Research*, vol. 28, no. 2, pp. 315–325, 2009.

[10] A. Hornung, K. M. Wurm, and M. Bennewitz, "Humanoid robot localization in complex indoor environments," *Robotics*, no. Mcl, pp. 1690–1695, 2010.

[11] R. Tellez, F. Ferro, S. Garcia, E. Gomez, E. Jorge, D. Mora, D. Pinyol, J. Oliver, O. Torres, J. Velazquez, and D. Faconti, "Reem-B: an autonomous lightweight human-size humanoid robot," in *International Conference on Humanoid Robots*, pp. 462–468, 2008.

[12] D. Joho, C. Stachniss, P. Pfaff, and W. Burgard, "Autonomous Exploration for 3D Map Learning," in *Autonome Mobile Systeme (AMS)* (K. Berns and T. Luksch, eds.), (Kaiserslautern, Germany), pp. 22–28, Springer, 2007.

[13] F. Amigoni, "Experimental evaluation of some exploration strategies for mobile robots," *2008 IEEE International Conference on Robotics and Automation*, vol. 67, pp. 2818–2823, 2008.

[14] K. Konolige and J. Bowman, "Towards lifelong visual maps," in *Proceedings of IROS*, 2009.

[15] D. Fox, W. Burgard, and S. Thrun, "Markov localization for mobile robots in dynamic environments," *rrr*, vol. 11, pp. 391–427, 1999.

[16] D. Hahnel, R. Triebel, W. Burgard, and S. Thrun, "Map building with mobile robots in dynamic environments," in *Proceedings of ICRA*, 2003.

[17] W. Montemerlo, M.; Thrun, S.; Whittaker, "Conditional particle filters for simultaneous mobile robot localization and people-tracking," in *International Conference on Robotics and Automation*, 2002.

[18] D. Hähnel, D. Schulz, and W. Burgard, "Mobile robot mapping in populated environments," *Advanced Robotics*, vol. 17, no. 7, pp. 579–597, 2003.

[19] G. Wolf, D.; Sukhatme, "Online simultaneous localization and mapping in dynamic environments," in *International Conference on Robotics and Automation*, 2004.

[20] P. Biber and T. Duckett, "Dynamic maps for long-term operation of mobile service robots," in *Proceedings of the RSS*, 2005.

[21] W. Burgard, C. Stachniss, and D. Haehnel, *Mobile Robot Map Learning from Range Data in Dynamic Environments*, vol. 35 of *Springer Tracts in Advanced Robotics*. Springer, 2007.

[22] A. Eliazar and R. Parr, "DP-SLAM: Fast, Robust Simultaneous Localization and Mapping Without Predetermined Landmarks," in *International Joint Conference on Artificial Intelligence*, 2003.

[23] W. Thrun, Sebastian Fox, Dieter Burgard, *Probabilistic robotics*, vol. 45 of *Intelligent robotics and autonomous agents*. MIT Press, 2005.

[24] K. Gutmann, J. Konolige, "Incremental Mapping of Large Cyclic Environments," in *International Symposium on Computational Intelligence in Robotics and Automation*, 1999.

[25] V. Ila, J. Porta, and J. Andrade-Cetto, "Information-Based Compact Pose SLAM," *IEEE Transactions on Robotics*, vol. 26, pp. 78–93, Feb. 2010.

[26] M. Bosse, P. Newman, J. Leonard, and S. Teller, "Simultaneous Localization and Map Building in Large-Scale Cyclic Environments Using the Atlas Framework," *The International Journal of Robotics Research*, vol. 23, no. 12, pp. 1113–1139, 2004.

[27] C. Estrada, J. Neira, and J. D. Tardos, "Hierarchical SLAM: real-time accurate mapping of large environments," *IEEE Transactions on Robotics*, vol. 21, no. 4, pp. 588–596, 2005.

[28] J. Modayil, P. Beeson, and B. Kuipers, "Using the topological skeleton for scalable global metrical map-building," *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 2, pp. 1530–1536.

[29] L. Paz, J. Tardos, and J. Neira, "Divide and Conquer: EKF SLAM in O(n)," *IEEE Transactions on Robotics*, vol. 24, pp. 1107–1120, Oct. 2008.

[30] *Navigation in Hybrid Metric-Topological Maps*, 2011.

[31] F. Thrun, Sebastian Fox, Dieter Burgard, Wolfram Dellaerta, "Robust Monte Carlo localization for mobile robots," *Artificial Intelligence*, vol. 128, no. 1-2, pp. 99–141, 2001.

[32] S. Jensfelt, Patric Kristensen, "Active Global Localization for a Mobile Robot Using Multiple Hypothesis Tracking," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 5, 2001.

[33] S. Se, D. Lowe, and J. Little, “Local and global localization for mobile robots using visual landmarks,” *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*, vol. 1, pp. 414–420, 2001.

[34] P. Nunez, R. Vazquezmartin, J. Deltoro, A. Bandera, and F. Sandoval, “Natural landmark extraction for mobile robot navigation based on an adaptive curvature estimation,” *Robotics and Autonomous Systems*, vol. 56, no. 3, pp. 247–264, 2008.

[35] S.-K. P. Soonyong Park, Soohwan Kim, Mignon Park, “Vision-based global localization for mobile robots with hybrid maps of objects and spatial layouts,” *Information Sciences*, vol. 179, no. 24, 2009.

[36] C. Ronzoni, D. Olmi, R. Secchi, C. Fantuzzi, “AGV global localization using indistinguishable artificial landmarks,” in *Automation, International Conference on Robotics and Automation*, 2011.

[37] S. Se, D. Lowe, and J. Little, “Vision-based global localization and mapping for mobile robots,” *IEEE Transactions on Robotics*, vol. 21, pp. 364–375, June 2005.

[38] J. Cadena, C. Neira, “A Learning Algorithm for Place Recognition,” in *ICRA Workshop on Long-term Autonomy*, vol. 1, 2011.

[39] Sebastian Thrun, M. Beetz, Maren Bennewitz, Wolfram Burgard, A.B. Creemers, Frank Dellaert, Dieter Fox, Dirk Hahnel, Chuck Rosenberg, Nicholas Roy, Jamieson Schulte and D. Schulz, “Probabilistic Algorithms and the Interactive Museum Tour-Guide Robot Minerva,” *The International Journal of Robotics Research*, vol. 19, pp. 972–999, Nov. 2000.

[40] C. Gamallo, C. V. Regueiro, P. Quint, and M. Mucientes, “Monte Carlo localization for a guide mobile robot in a crowded environment based on omnivision,” in *TAROS*, 2008.

[41] R. Valencia, J. Andrade-cetto, and J. M. Porta, “Path Planning in Belief Space with Pose SLAM,” in *International Conference on Robotics and Automation*, 2011.

[42] A. Censi, D. Calisi, A. De Luca, and G. Oriolo, “A Bayesian framework for optimal motion planning with uncertainty,” *2008 IEEE International Conference on Robotics and Automation*, pp. 1798–1805, May 2008.

[43] S. Montemerlo, M.; Thrun, “Simultaneous localization and mapping with unknown data association using FastSLAM,” in *International Conference on Robotics and Automation*, 2003.

[44] C.-C. Yang, Shao-Wen Wang, “Feasibility Grids for Localization and Mapping in Crowded Urban Scenes,” in *International Conference on Robotics and Automation*, 2011.

[45] P. S. Robert Baczyk , Andrzej Kasinski, “Vision-Based Mobile Robot Localization with Simple Artificial Landmarks,” in *Artificial Landmarks, Prepr. 7th IFAC Symp. on Robot Control*, 2003.

[46] S. Lee and J.-B. Song, “Mobile robot localization using infrared light reflecting landmarks,” *2007 International Conference on Control Automation and Systems*, vol. 2, pp. 674–677, 2007.

[47] Y. Zhou and W. Liu, “Preliminary Research on Indoor Mobile Robot Localization using Laser-activated RFID,” *2007 IEEE International Conference on RFID*, pp. 78–85, 2007.

[48] G. Kantor and S. Singh, “Preliminary results in range-only localization and mapping,” in *International Conference on Robotics and Automation (Cat. No.02CH37292)*, vol. 2, pp. 1818–1823, Ieee, 2002.

[49] M. Ladd, K. Bekris, A. Rudys, D. Wallach, and L. Kavraki, “On the Feasibility of Using Wireless Ethernet for Indoor Localization,” *IEEE Transactions on Robotics and Automation*, vol. 20, pp. 555–559, June 2004.

[50] J. Chestnutt, Y. Takaoka, K. Suga, K. Nishiwaki, J. Kuffner, and S. Kagami, “Biped navigation in rough environments using on-board sensing,” *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3543–3548, Oct. 2009.

[51] E. Marder-eppstein, E. Berger, T. Foote, B. Gerkey, and K. Konolige, “The Office Marathon : Robust Navigation in an Indoor Office Environment,” in *International Conference on Robotics and Automation*, 2010.

[52] F. Doshi, E. Brunskill, A. Shkolnik, T. Kollar, R. Tedrake, and N. Roy, “Collision detection in legged locomotion using supervised learning,” in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 317–322, Ieee, Oct. 2007.

[53] E. Trautmann and L. Ray, “Mobility characterization for autonomous mobile robots using machine learning,” *Autonomous Robots*, vol. 30, pp. 369–383, Mar. 2011.

[54] D. Shell and M. Mataric, “On foraging strategies for large-scale multi-robot systems,” *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2717–2723, Oct. 2006.

[55] G. Pini, A. Brutschy, M. Birattari, and M. Dorigo, “Interference reduction through task partitioning in a robotic swarm,” in *Sixth International Conference on Informatics in Control, Automation and Robotics*, no. April, pp. 52–59, Citeseer, 2009.

[56] E. A. Sisbot, L. F. Marin-Urias, R. Alami, and T. Simeon, “A Human Aware Mobile Robot Motion Planner,” *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 874–883, 2007.

[57] M. Bennewitz, W. Burgard, and S. Thrun, “Adapting navigation strategies using motions patterns of people,” in *IEEE International Conference on Robotics and Automation*, Ieee, 2003.

[58] C. Fulgenzi, A. Spalanzani, and C. Laugier, “Probabilistic motion planning among moving obstacles following typical motion patterns,” in *Environment*, pp. 4027–4033, Ieee, 2009.

[59] P. Raja and S. Pugazhenthi, “Path Planning for Mobile Robots in Dynamic Environments,” *International Journal of the Physical Sciences*, vol. 6, no. 20, pp. 401–405, 2011.

[60] M. Svenstrup, T. Bak, and H. J. Andersen, “Trajectory Planning for Robots in Dynamic Human Environments,” *Media*, pp. 4293–4298, 2010.

[61] B. Frank, M. Becker, C. Stachniss, W. Burgard, and M. Teschner, “Efficient path planning for mobile robots in environments with deformable objects,” *2008 IEEE International Conference on Robotics and Automation*, pp. 3737–3742, May 2008.

[62] B. Maris, D. Botturi, and P. Fiorini, “Trajectory planning with task constraints in densely filled environments,” *Computer*, pp. 2333–2338, 2010.

[63] E. Y. S. Lengagne, A. Kheddar, “Generation of Optimal Dynamic Multi-Contact Motions : Application to Humanoid Robots,” *IEEE Transactions on robotics (submitted)*, vol. X, no. X, pp. 1–10, 2011.

[64] A. Hornung, E. G. Jones, M. Bennewitz, and M. Likhachev, “Towards Navigation in Three-Dimensional Cluttered Environments,” in *International Conference on Intelligent Robots and Systems*, 2011.