IROS Workshop on Visual Control of Mobile Robots (ViCoMoR 2012)
October 11th, 2012, Vilamoura, Algarve, Portugal

From the general navigation problem to its image based solutions

Adrien Durand Petiteville[1] and Viviane Cadenat[1]

Abstract— **This article presents a brief study of mobile robots navigation. In a first part, we provide an overview of this problem, analyzing the different involved processes and showing several architectures allowing to organize them. In a second step, we consider the vision based navigation problem. From the previous analysis, we highlight the interest of using topological maps in this context and propose an overview of existing works in this area. Finally, we present our own solution to the problem, showing its relevance and its efficiency.**

I. INTRODUCTION

In this paper we consider the well known autonomous navigation problem. It consists for the robot in reaching a goal through a given environment while dealing with unexpected events [1]. Thus, the navigation generally involves six processes: perception, modelling, planning, localization, action and decision. A wide range of techniques are available in the literature for each of them. As these processes cooperate within an architecture to perform the navigation, they cannot be designed independently and it is necessary to have an overview of the problem to select suitably the different methods. This article aims at (i) providing such an overview, (ii) presenting the visual solutions and (iii) positioning our own work in this general context.

II. THE NAVIGATION FRAMEWORK

In this section, we present the different processes and the associated methods before highlighting several architectures.

A. The navigation processes

1) Perception: In order to acquire the data required by the navigation, a robot can be equipped with both proprioceptive and exteroceptive sensors. The first ones (odometers, gyroscopes, . . .) provide data relative to the robot internal state whereas the second ones (camera, laser telemeters, bumpers, . . .) give information about the environment. The sensory data may be used by four processes: environment modelling, localization, decision and robot control.

2) Modelling: A navigation process generally requires an environment model. This model is initially built using *a priori* data. It is not always complete, and can evolve with time. In this case, the model is updated thanks to the data acquired during the navigation. There exists two kinds of models, namely the metric and/or topologic maps [2].

The metric map is a continuous or discrete representation of the free and occupied spaces. A global frame is defined and the robot and obstacles poses are known with more or less precision in this frame. The data must then be expressed in this frame to update the model [1].

The topologic map is a discrete representation of the environment based on graphs [1]. Each node represents a continuous area of the scene defined by a characteristic property. The areas are naturally connected and can be limited to a unique scene point. The characteristic property, chosen by the user, may be the feature visibility or belonging to a same room. Moreover, if a couple of nodes verifies an adjacency condition, then they are connected. The adjacency condition is chosen by the user and may correspond for example to the existence of a path allowing to connect two sets, each of them represented by a node. A topologic map is less sensitive to the scene evolutions: it has to be updated only if there are modifications concerning the area represented by the nodes or the adjacency between two nodes.

Metric and topologic maps can be enhanced by adding to nodes sensory data, actions or control inputs. These informations may be required to localize or control the robot. There also exists hybrid representations of the environment based on both metric and topologic maps [3], [4], [5].

3) Localization: For navigation, two kinds of localizations are identified: the metric one and the topologic one [2]. The metric localization consists in calculating the robot pose with respect to a global or a local frame. To do so, a first solution is to use only proprioceptive data. However this solution can lead to significant errors [6] [7] for three reasons. The first one comes from the pose computation process which consists in successively integrating the acquired data, which induces a drift. The second one is due to the model which is used to determine the pose: an error which occurs during the modelling step is automatically transferred to the pose computation. Finally, the last one is related to phenomenons such as sliding which are not taken into account. It is then necessary to consider additional exteroceptive information to be able to localize the robot properly. Visual odometry [8] [9] [10] is an example of such a fusion.

The topological localization [3] [11] [12] [13] consists in relating the data provided by the sensors with the ones associated to the graph nodes which model the environment. The goal is to determine the situation in a graph and not with respect to a frame [2]. Topological localization is not very sensitive to measurement errors unlike its metric alter-ego. The precision depends on the accuracy with which the environment has been described.

4) Planning: The planning step consists in computing, using the environment model, an itinerary allowing the robot to reach its final pose. The itinerary may be a path, a trajectory, a set of poses to reach successively, . . . There

[1]CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France, Univ de Toulouse, UPS, LAAS ; F-31400 Toulouse, France[adurandp,cadenat]at laas.fr

exists a large variety of planning methods depending on the environment modeling. An overview is presented hereafter.

A first approach consists in computing the path or the trajectory using a metric map. To do so, the geometric space is transposed into the configuration space. The configuration corresponds to a parametrization of the static robot state. Thus a robot with a complex geometry in the workspace is represented by a point in the configuration space [2]. Then planning consists in finding a path or a trajectory in the configuration space allowing to reach the final configuration from the initial one [1]. With a continuous representation of the environment, a path or a trajectory can be obtained using visibility graphs or Voronoi diagrams [14]. With a discrete scene model, planning is performed thanks to methods from graph theory such as A^* or Dijkstra algorithms[15] [16]. For the two kinds of maps, planning may be time consuming. A solution consists in using probabilistic planning: *probabilistic roadmap* [17] [18] or *rapidly exploring random tree* [19].

When the environment model is incomplete at the beginning of the navigation, unexpected obstacles may lie on the robot itinerary. A first solution to overcome this problem consists in adding the obstacles to the model and then to plan a new path or a new trajectory [1]. In [20], [21], authors propose to consider the trajectory as an elastic band which can be deformed if necessary. A global re-planning step can then be required for a major environment modification.

When using a topological map without any metric data, the planned itinerary is generally composed of a set of poses to reach successively. These poses can be expressed in a frame associated to the scene or to a sensor. The itinerary is computed using general methods from the graph theory. Depending on the precision degree used to describe the environment, it is possible that the planned itinerary does not take into account all the obstacles. This issue has to be considered during the motion realization.

5) Action: To perform the tasks required by the navigation, two kinds of controllers can be designed: state feedback or output feedback [22]. In the first case, the task is generally defined by an error between the current robot state[1] and the desired one. To make this error vanish, a state feedback is designed. The control law implementation requires to know the current state value and therefore a metric localization is needed. In the second case, it consists in making the error between the current measure and the desired one vanish. This measure depends on the relative pose with respect to an element of the environment, called feature or landmark. When using vision, the measures correspond to image data (points, lines, moments; etc. [25]). For proximity sensors, they are defined by distances provided by ultrasound [26] and laser [5] [27] telemeters. The measures are directly used in the control law computation, which means that no metric localization is required. However the landmark must be perceptible during the entire navigation to compute the control inputs.

[1]The state may correspond to the robot pose with respect to the global frame or to a given landmark [23] [24].

6) Decision: To perform a navigation, it may be necessary to take decisions at different process states. These decisions may concern high level, e.g. a re-planning step [21], or low level, e.g. the applied control law [26]. They are usually taken by supervision algorithms based on exteroceptive data.

B. The navigation architectures

The different processes required by a navigation have now been identified. Here, we present examples of navigation architectures based on the previously presented processes. We propose to organize our presentation around the controllers.

1) "State feedback" based architecture: First, we consider a robot controlled using a state feedback controller in a free space. The initial and final configurations are defined in a world frame. To compute the control inputs, the state value has to be known at any time. The robot capacity to geometrically localize itself is a necessary condition to successfully perform the navigation. Let us note that, the distance that the robot can cover is only limited by the localization precision. Indeed, a too large error on the state value will result in inconsistent control inputs.

We now consider a cluttered environment. In this case there are two solutions, either reactive or planning based. The reactive one consists in controlling the robot using two controllers : a first one making the error between the current and the desired poses vanish, allowing to reach the goal, and a second one performing the obstacle avoidance using exteroceptive data. It is then necessary to develop a supervision module selecting the adequate controller. This solution, which guarantees the non-collision with obstacles, does not allow to ensure the navigation success. Indeed, the obstacle avoidance is locally performed and does not take into account the goal. The second solution consists in following a previously planned collision free path. To this aim, the environment has to be modeled using a map. If the model represents the whole scene, then the navigation simply consists in following the planned itinerary using a state feedback controller. A supervision module is no more required. If the environment is not completely modeled, it may be necessary to update it when an obstacle appears on the robot path. After the update, a re-planning step is performed. In this case, a supervision module which decides to update and re-plan is mandatory. Finally, it should be noticed that the metric localization is required and limits the navigation range for each solution.

As a conclusion, when the robot is controlled using state feedback controllers, the metric localization is a decisive element, as the navigation success depends on the localization quality. Moreover, in a cluttered environment, a model is quickly mandatory to converge towards the desired pose or to avoid obstacles. The metric localization and modeling processes are very sensitive to measurement errors. It is then necessary to pay attention to the methods performances when the navigation is based on state feedback controllers.

2) "Output feedback" based architecture: We now consider a robot controlled using an output feedback controller in a free space. The initial pose is unknown whereas the desired one with respect to a landmark is defined by measures. The robot can converge toward the desired pose if the landmark can be perceived at any instant. It is now the sensor range which limits the navigation range. In this case no metric localization is required.

When the environment is cluttered, a first solution consists in using a sole output feedback controller to reach the desired pose while avoiding obstacles. A second idea is to control the robot thanks to two output feedback controllers : the first one allows to reach the desired pose and the second one guarantees non collision. A supervisor selecting the adequate controller is then required. For both solutions, local minima problems may occur. Moreover, the navigation range is still limited by the sensors range. Global informations must then be used to perform a long range navigation. These global information can be added using a metric map or a topological map. In the first case, it is possible to plan a path taking into account the features availability at each pose. The planned itinerary is then composed by several landmarks successively used to compute the control inputs. Moreover, for a static environment, joint limits, visibility and obstacles can also be considered in the planning step. Nevertheless, this approach requires environment, robot and sensors reliable models. In the second case, a topological map is used to provide the necessary global information. Here, the additional data associated to the graph nodes correspond usually to the desired features or landmarks. As previously, the planned itinerary is made of measures or landmarks set to reach. This approach is based on a partial environment representation. The model is then less sensitive to the environment modifications, but does not allow to take into account several constraints such as obstacles or joints limits during the planning step. A topologic localization is needed.

III. THE VISUAL NAVIGATION

Now we focus on the vision based navigation problem. The camera is then used as the main sensor, which still allows to select any of the previous presented approaches as shown in [28] and [29]. In these works, the authors propose an overview of visual navigation splitting the methods into two main categories: the metric map based ones and the topological map based ones. Following our previous analysis, we have selected the topological approach. Indeed, in this case, the metric localization is no more required, limiting the inaccuracy due to the use of noisy data in the state computation process. Furthermore, a topological map provides sufficient data to perform a navigation task, without significantly increasing the problem complexity. Finally, this representation is less sensitive to scene modifications. We present hereafter methods based on such an approach.

A. Related works

In [30], the scene is modelled by a graph whose nodes correspond to the corridors. The robot navigates into the corridors using an image based visual servoing relying on the vanishing point as visual feature. This method is then limited to an environment composed of corridors. Other approaches propose to model the environment during a pre-navigation step. During this phase, images obtained for several close robot configurations are memorized. A topological map, also called visual memory, is built by organizing the images [4]. The planned itinerary is called visual road [31]. This approach is performed using omnidirectional [32] [33] [34] [35] [36] or pinhole camera [37] [38] [39] [40] [36]. However, none of these approaches take into account the two major problems of visual navigation : occlusions, *i.e.* the landmarks loss, and collisions with obstacles. A set of works [41] [42] [43] has produced a visual navigation allowing to avoid unexpected obstacles while tolerating partial occlusions. The topological map is also built during a pre-navigation step. Time variant visual features are used by the visual servoing while the obstacle avoidance is performed thanks to a potential fields based control law. Thus, the learnt path can be replayed using a topological map while avoiding collisions.

B. Our approach

We propose a similar approach to perform the navigation while dealing with collisions and total occlusions [44]. Following the above analysis, we have chosen to use a camera, a topological map and several output feedback controllers organized in a supervision algorithm. We present hereafter our approach detailing our choices for each process. More details can be found in [44].

1) Perception: Our robot is equipped with a camera and a laser able to detect respectively the landmarks of interest and the obstacles. Our approach will rely on these two data.

2) Modelling: We now focus on the topological map, which consists of a directed graph. Each node corresponds to a landmark present in the scene. If there are n_l landmarks, then the graph is composed of n_l nodes. A point, corresponding to the desired robot pose S_i with respect to the landmark T_i, is associated to each node N_i, with $i \in [1,...,n_l]$. An arc $A(N_j,N_k)$ is created if the landmark T_k, associated to the node N_k, can be seen from the pose S_j, associated to the node N_j, with $j \in [1,...,n_l]$, $k \in [1,...,n_l]$ and $j \neq k$. Moreover, sensory data D_i extracted from an image of the landmark T_i taken at the pose S_i is associated to the node N_i.

3) Localization: During the navigation, and especially at the beginning, the robot has to localize itself into the graph. The localization process identifies the landmarks that are in the field of view of the camera. Localization is performed using the sensory data associated to each node. It consists in making a test of similarity between two images. To this aim, the descriptors of the current image and those from the data base are matched. The image from the data base which has the best similarity with the current image is selected. Then we consider that the robot situation in the graph corresponds to the node containing the selected image.

4) Planning: The initial and final poses are obtained from the localization process and from the user. They are now considered as known. The path T_P made of a sequence of n_P landmarks $[T_{P1},...,T_{Pn_P}]$ to reach, is planned using the Dijkstra algorithm [16] which provides the shortest path.

5) Action: To perform a long range navigation, we use three output feedback controllers [44]. The first one allows to perform a short range navigation with respect to a landmark, which will be referred to as "sub-navigation". This controller is defined by a classical image based visual servoing [45], which makes the error between the current and desired images vanish. The second one performs the obstacle avoidance by stabilizing the robot on a path defined thanks to telemetric data [46]. The last one is intended to avoid unsuitable motions when switching from one landmark to the other [44]. The transition between each controller is performed by a dynamic sequencing allowing to guarantee the control law continuity [47]. Thus, using these two controllers, the robot can successively reach the landmarks composing the path while avoiding the obstacles.

To manage the occlusions problem, we have used the algorithm [48]. It allows to predict the visual features next position from the previous visual data and the visual features depth. The latter is estimated thanks to a predictor/corrector using a number n_{pc} of images allowing to provide an accurate estimation even in the presence of noisy data [49]. Using these tools we can deal with total occlusions.

6) Decision: The decision process has to activate or deactivate the available tools in order to guarantee the long range navigation success. We propose to use a supervision algorithm to perform the decision process. The algorithm, summarized in figure 1, is built using the following strategy.

First of all, the robot localizes itself to determine the initial node in the graph. Then, knowing the desired pose, a path T_P composed by a set of landmarks to reach is computed. Then, the initialization phase is executed. It consists in making small rotations to estimate the visual features depth of landmark T_{P1}. Thus occlusions can be managed during the sub-navigation with respect to T_{P1}. When the initialization phase is over, the sub-navigation to T_{P1} is launched. If the robot is too close to an obstacle, the obstacle avoidance controller is used. During the sub-navigation, the robot regularly looks for the next landmark T_{P2}. If this latter is not found, the robot continues the current sub-navigation and restarts the depth estimation process. When it converges, T_{P2} is one more time looked for. This loop is repeated until the next landmark is found or the current sub-navigation is over. In the latter case, the robot turns on itself to identify the next target. If it is not found, then the graph is updated and a new path is planned. If there is no path to reach the desired landmark, the navigation fails. We consider now that the landmark T_{P2} has been found. The sub-navigation, obstacles avoidance and looking for the next landmark processes are repeated using the same conditions as previously until the robot reaches the desired pose or the navigation fails.

IV. SIMULATIONS

We have simulated a long range navigation using Matlab™software. The considered cart-like robot is equipped with a camera mounted on a pan-platform and a laser telemeter. In the scene shown in figure 2(a), there are $n_l = 9$ artificial landmarks made of a different number

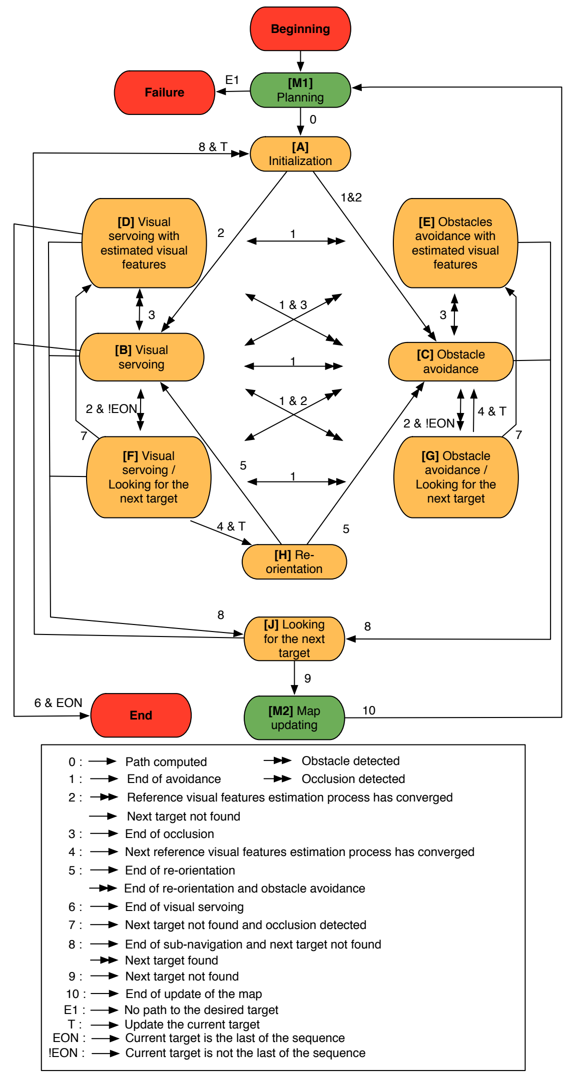
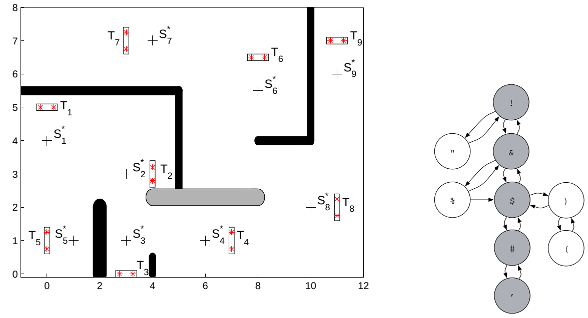


Fig. 1. Supervision algorithm for a long range navigation

of points. The occluding obstacles are represented in black whereas the non-occluding one is in gray. To model the environment, the robot is successively placed at the desired poses S_i^*, with $i \in [1, \ldots, n_l]$. Thus, we obtain the topological map presented in figure 2(b). It should be noticed that the topological map is not complete, as the chosen poses S_i^* do not allow to connect all the nodes. For example, the selected S_8^* does not allow to relate N_8 and N_6, although other choices would have permitted it. However, it is not a problem as there exists a path allowing to reach the desired pose.

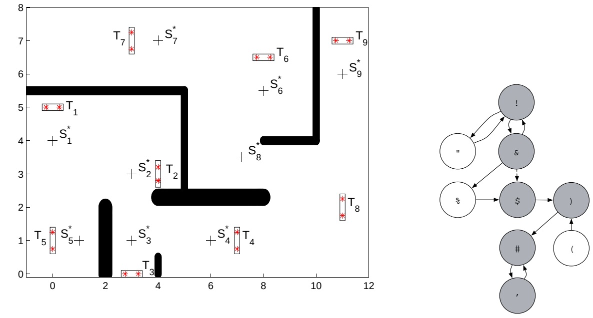
From its initial pose near S_5^*, the robot must reach S_7^* with respect to the landmark T_7. After a localization and using the topological map, the shortest path $T_P = [T_1, T_2, T_4, T_6, T_7]$ is computed (see Fig. 2(b)). Then, the mission starts and the supervision algorithm selects the current task to perform until T_7 is reached. Figure 3 shows the corresponding task sequencing and robot trajectory. As we can see, the mission is successfully realized despite the unexpected obstacles.

We propose a second simulation to illustrate the re-



(a) Environment of navigation #1 (b) Environment #1 topological map

Fig. 2. Mapping #1



(a) Environment of navigation #2 (b) Environment #2 topological map

Fig. 4. Mapping #2

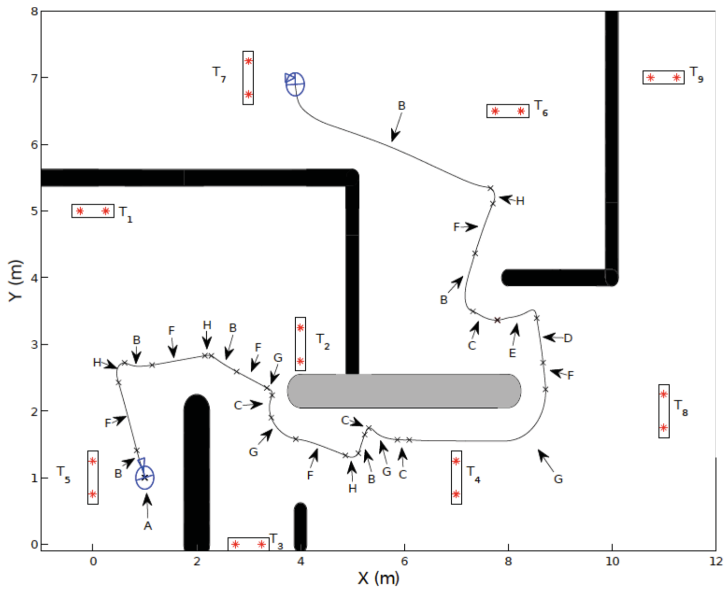


Fig. 3. Robot long range navigation #1 (letters correspond to the current executed task (see figure 1))

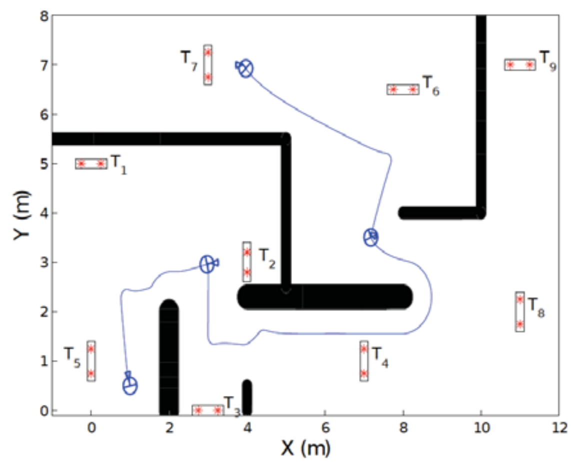


Fig. 5. Robot long range navigation #2

planning phase and the necessity to provide the most complete topological map. We consider the same environment as previously, except that all obstacles are now occluding. If we use the same poses S_i^*, reaching S_7^* from S_5^* is impossible because nodes N_4 and N_6 cannot be connected anymore. To overcome this problem, a new pose S_8^* allowing to relate N_8 and N_6 is defined (see Fig 4(a)). The proposed map is then more complete than the previous one, showing the importance of the choice of each S_i^*. The corresponding environment map is shown in figure 4(b). Note that we have willingly introduced an error in the map by connecting nodes N_2 and N_4 whereas this relation does not exist anymore.

To reach S_7^* the robot now plans the following path $T_P = [T_1, T_2, T_4, T_8, T_6, T_7]$. Then the navigation starts and the robot reaches S_2^* but cannot find T_4. At this time, a localization process is performed, showing that only T_2 and T_3 can be perceived from the robot current position. The map is then updated by suppressing the link between N_2 and N_4. A new path $T_P = [T_3, T_4, T_8, T_6, T_7]$ is computed, and the navigation is launched again. Using the adequate controllers the robot then performs the task and reaches S_7^*.

V. CONCLUSION

This paper was focused on the navigation problem. We have first highlighted the different processes involved in the navigation and shown their organization within several possible architectures. Then, we have considered the vision based solutions, showing the interest of using a topological map. We have finally positioned our own works in this general framework, demonstrating its efficiency to perform a visual navigation despite collisions and occlusions. One of the next challenges will be to take into account the presence of mobile obstacles (vehicles, human beings, ...) to improve the robot autonomy in a real environment.

REFERENCES

[1] H. Choset, K. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki, and S. Thrun, *Principles of Robot Motion*. MIT Press, Boston, 2005.

[2] R. Siegwart and I. Nourbakhsh, *Introduction to autonomous mobile robots*, ser. A bradford book, Intelligent robotics and autonomous agents series. The MIT Press, 2004.

[3] S. Segvic, A. Remazeilles, A. Diosi, and F. Chaumette, “A mapping and localization framework for scalable appearance-based navigation,” *Computer Vision and Image Understanding*, vol. 113, no. 2, pp. 172–187, February 2009.

[4] E. Royer, M. Lhuillier, M. Dhome, and J.-M. Lavest, "Monocular vision for mobile robot localization and autonomous navigation," *International Journal of Computer Vision*, vol. 74, no. 3, pp. 237–260, 2007.

[5] A. Victorino and P. Rives, "An hybrid representation well-adapted to the exploration of large scale indoors environments," in *IEEE International Conference on Robotics and Automation*, New Orleans, USA, 2004, pp. 2930–2935.

[6] D. Cobzas and H. Zhang, "Mobile robot localization using planar patches and a stereo panoramic model," in *Vision Interface*, Ottawa, Canada, June 2001, pp. 04–99.

[7] J. Wolf, W. Burgard, and H. Burkhardt, "Robust vision-based localization for mobile robots using an image retrieval system based on invariant features," in *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, vol. 1, 2002, pp. 359 – 365 vol.1.

[8] D. Nister, O. Naroditsky, and J. Bergen, "Visual odometry," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 1, june-2 july 2004, pp. I–652 – I–659 Vol.1.

[9] A. Comport, E. Malis, and P. Rives, "Real-time quadrifocal visual odometry," *International Journal of Robotics Research, Special issue on Robot Vision*, vol. 29, 2010.

[10] Y. Cheng, M. Maimone, and L. Matthies, "Visual odometry on the mars exploration rovers - a tool to ensure accurate driving and science imaging," *Robotics Automation Magazine, IEEE*, vol. 13, no. 2, pp. 54 –62, june 2006.

[11] R. Sim and G. Dudek, "Learning visual landmarks for pose estimation," in *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, vol. 3, 1999, pp. 1972 –1978 vol.3.

[12] L. Paletta, S. Frintrop, and J. Hertzberg, "Robust localization using context in omnidirectional imaging," in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 2, 2001, pp. 2072 – 2077 vol.2.

[13] B. Krösea, N. Vlassisa, R. Bunschotena, and Y. Motomura, "A probabilistic model for appearance-based robot localization," *Image and Vision Computing*, vol. 19, pp. 381–391, 2001.

[14] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu, *Spatial Tessellations - Concepts and Applications of Voronoi Diagrams*. John Wiley, 2000.

[15] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination od minimum cost paths," *Systems Science and Cybernetics, IEEE Transactions on*, vol. 4, no. 2, pp. 100–107, 1968.

[16] E. W. Dijkstra, "A short introduction to the art of programming," Aug 1971.

[17] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *Robotics and Automation, IEEE Transactions on*, vol. 12, no. 4, pp. 566 –580, aug 1996.

[18] R. Geraerts and M. H. Overmars, "A comparative study of probabilistic roadmap planners," in *Proc. Workshop on the Algorithmic Foundations of Robotics (WAFR'02)*, Nice, France, December 2002, pp. 43–57.

[19] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," *In*, vol. TR 98-11, no. 98-11, pp. 98–11, 1998.

[20] Khatib, Jaouni, Chatila, and Laumond, "Dynamic path modification for car-like nonholonomoic mobile robots," in *IEEE Int. Conf. on Robotics and Automation*, April 1997, pp. 490–496.

[21] F. Lamiraux, D. Bonnafous, and O. Lefebvre, "Reactive path deformation for nonholonomic mobile robots," *Robotics, IEEE Transactions on*, vol. 20, no. 6, pp. 967 – 977, dec. 2004.

[22] W. S. Levine, *The Control Handbook*. CRC Press Handbook, 1996.

[23] S. Hutchinson, G. Hager, and P. Corke, "A tutorial on visual servo control," *IEEE Trans. on Rob. and Automation*, vol. 12, no. 5, pp. 651–670, 1996.

[24] D. Bellot and P. Danes, "Towards an lmi approach to multiobjective visual servoing," in *European Control Conference 2001*, Porto (Portugal), September 2001.

[25] F. Chaumette, "Image moments: a general and useful set of features for visual servoing," *Robotics, IEEE Transactions on*, vol. 20, no. 4, pp. 713 – 723, aug. 2004.

[26] D. Folio and V. Cadenat, "A controller to avoid both occlusions and obstacles during a vision-based navigation task in a cluttered environment," in *European Control Conference (ECC05)*, Seville, Espagne, December 2005, pp. 3898–3903.

[27] S. Thrun, W. Burgard, and D. Fox, "A real time algorithm for mobile robot mapping with applications to multi-robot and 3d mapping," in *IEEE International Conference on Robotics and Automation*, San Francisco, CA, USA, April 2000.

[28] G. Desouza and A. Kak, "Vision for mobile robot navigation: a survey," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 2, pp. 237 –267, feb 2002.

[29] F. Bonin-Font, F. Ortiz, and G. Oliver, "Visual navigation for mobile robots : a survey," *Journal of intelligent and robotic systems*, vol. 53, no. 3, p. 263, 2008.

[30] R. Vassalo, H. Schneebeli, and J. Santos-Victor, "Visual servoing and appearance for navigation," *Robotics and autonomous systems*, 2000.

[31] Y. Matsumoto, M. Inaba, and H. Inoue, "Visual navigation using viewsequenced route representation," in *IEEE Int. Conf. on Robotics and Automation*, Minneapolis, USA, 1996, pp. 83–88 –2692.

[32] J. Gaspar, N. Winters, and J. Santos-Victor, "Vision-based navigation and environmental representations with an omni-directional camera," *IEEE transactions on robotics and automation*, vol. 6, no. 6, pp. 890–898, 2000.

[33] Y. Yagi, K. Imai, K. Tsuji, and M. Yachida, "Iconic memory-based omnidirectional route panorama navigation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, pp. 78–87, 2005.

[34] T. Goedemé, M. Nuttin, T. Tuytelaars, and L. V. Gool, "Omnidirectional vision based topological navigation," *International Journal of Computer Vision*, vol. 74, no. 3, pp. 219–236, 2007.

[35] O. Booij, B. Terwijn, Z. Zivkovic, and B. Krose, "Navigation using an appearance based topological map," in *IEEE Int. Conf. on Robotics and Automation*, Rome, Italy, 2007, pp. 3927– 3932.

[36] J. Courbon, Y. Mezouar, and P. Martinet, "Autonomous navigation of vehicles from a visual memory using a generic camera model," *Intelligent Transport System (ITS)*, vol. 10, pp. 392–402, 2009.

[37] S. Jones, C. Andresen, and J. Crowley, "Appearance based process for visual navigation," in *Intelligent Robots and Systems, 1997. IROS '97., Proceedings of the 1997 IEEE/RSJ International Conference on*, vol. 2, sep 1997, pp. 551 –557 vol.2.

[38] G. Blanc, Y. Mezouar, and P. Martinet, "Indoor navigation of a wheeled mobile robot along visual routes," in *International Conference on Robotics and Automation*, Barcelona, Spain, 2005.

[39] Z. Chen and S. Birchfield, "Qualitative vision-based mobile robot navigation," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, may 2006, pp. 2686 –2692.

[40] T. Krajnìk and L. Přeučil, *A simple visual navigation system with convergence property*. H. Bruyninckx et al. (Eds.), 2008.

[41] A. Cherubini and F. Chaumette, "Visual navigation with a time-independent varying reference," in *IEEE Int. Conf. on Intelligent Robots and Systems, IROS'09*, St Louis, USA, October 2009, pp. 5968–5973.

[42] ——, "A redundancy-based approach to obstacle avoidance applied to mobile robot navigation," in *Proc. of IEEE Int. Conf. on Intelligent Robots and Systems*, Taipei, Taiwan, 2010.

[43] A. Cherubini, F. Spindler, and F. Chaumette, "A redundancy-based approach for visual navigation with collision avoidance," in *ICVTS proceedings*, 2011.

[44] A. Durand Petiteville, S. Hutchinson, V. Cadeant, and M. Courdesses, "2d visual servoing for a long range navigation in a cluttered environment," in 50^{th} *IEEE Conference on Decision and Control and European Control Conference*, Orlando, USA, December 2011.

[45] F. Chaumette and S. Hutchinson, "Visual servo control, part 1 : Basic approaches," *IEEE Robotics and Automation Magazine*, vol. 13, no. 4, 2006.

[46] P. Souères, T. Hamel, and V. Cadenat, "A path following controller for wheeled robots wich allows to avoid obstacles during the transition phase," in *IEEE, Int. Conf. on Robotics and Automation*, Leuven, Belgium, May 1998.

[47] P. Souères and V. Cadenat, "Dynamical sequence of multi-sensor based tasks for mobile robots navigation," in *SYROCO*, Wroclaw, Poland, September 2003.

[48] D. Folio and V. Cadenat, *Computer Vision - Treating Image Loss by using the Vision/Motion Link: A Generic Framework*. IN-TECH, 2008, ch. 4.

[49] A. Durand Petiteville, M. Courdesses, V. Cadenat, and P. Baillion, "On-line estimation of the reference visual features. application to a vision based long range navigation task," in *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, October 2010.