Combining 3D SLAM and visual tracking to reach and retrieve objects in daily-life indoor environments

Pierre Gergondet, Damien Petit, Maxime Meilland, Abderrahmane Kheddar, Andrew I. Comport, Andrea Cherubini

The 11th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI 2014)
Nov. 12 – 15, 2014 at Double Tree Hotel by Hilton, Kuala Lumpur, Malaysia

Combining 3D SLAM and Visual Tracking to Reach and Retrieve Objects in Daily-Life Indoor Environments

Pierre Gergondet[1,2], Damien Petit[1,2], Maxime Meilland[3],
Abderrahmane Kheddar[1,2], Andrew I. Comport[3], and Andrea Cherubini[2]

[1]CNRS-AIST Joint Robotics Laboratory (JRL), UMI3218/CRT, Tsukuba, Japan
[2]CNRS-UM2 LIRMM, Interactive Digital Human group, UMR5506, Montpellier, France
[3]CNRS-I3S Laboratory, University of Nice Sophia Antipolis, France

Abstract - In this paper, we draw perspectives to endow a humanoid robot with capabilities to reach known object in an indoor environment by combining continuous monitoring and building using SLAM and visual tracking. We integrates and exploits two key features: object recognition using the toolbox BLORT, and a SLAM (Simultaneous Localization And Mapping) software, that unifies volumetric 3D modeling and image-based key-frame modeling to be used in tracking. Using these two modules, we show that it is possible to reach a given object in the environment providing its model is registered and known. Our integration software is exemplified using a humanoid robot HRP-2, we present experimental results that illustrates the performance of our approach.

Keywords - Humanoid Robots, SLAM, Autonomous Navigation, Object recognition.

1. Introduction

In the frame of the RoboHow.Cog EU project[1], we are developing with different partners, technologies by which service robots learn to achieve tasks from web-enabled instructions or from observing humans on a daily basis. The demonstrators of the project consists in having a humanoid robot that operate tasks, in indoor environments such as houses or offices. Examples such tasks in houses are cooking in kitchen, arranging rooms, assisting trailed persons, etc.; in offices the humanoid robot could serve café to workers, or bring printed papers, replace printer's paper, prepare the meeting room, etc.

In order to do so, a humanoid robot shall first be able to have a knowledge of the environment and be able to navigate within it to retrieve objects of interest for a given task. For example, in order for the robot to feed paper to the printers or bring printed documents to owes, it has to know where the printer is and reach it, despite its place can change because of a new arrangement or cleaning that may results in the position of the printer to change slightly from its original pose. As far as humanoids are concerned, it is well known that its localization, w.r.t the environment can hardly be achieved solely by embedded sensors such as accelerometers, encoders, etc. Indeed, drifts in position estimation is unavoidable during walking inside the environment even when the walking path is well predefined. Moreover, such a path cannot be static and has to be adapted w.r.t to persons eventually met during the walking.

SLAM (Simultaneous Localization and Mapping) [1] provides recently a very mature technology [2] and allows the robot to simultaneously and continually build the map of the environments. Recent experiments conducted on our humanoid HRP-2 and HRP-4 robots with our colleagues's software in [3], demonstrate an amazing precision in localization with real-time performances. SLAM however builds a "rigid model" of the surrounding without semantics nor the possibility to distinguish objects that composes the environment. SLAM can also be used in closed-loop navigation to reach a given target defined as desired 3D coordinate point in the model. If previously mapped objects composing the entire scene as moved w.r.t to their mapped position, they are considered as outliers, yet the rigid map can be corrected to integrate this change in the position, see recent work in [4].

In the other hand, recent advances in robotic visual tracking [5] [6] is demonstrated in complex tasks that are achieved in robust closed-loop fashion among which reaching, manipulation, navigation, when targets of interest are totally or partially in the field-of-view of the robot's embedded camera(s). However, closed-loop visual servoing is still difficult to solve with humanoid robots due to many aspects: weak odometry, important blur caused by rapid movements or the sway motion and the feet impacts generated during walking movements [7] [8] [9].

Our work in this paper is rather technical; we demonstrate that by combining 3D SLAM developed in [3] and the visual object recognition and tracking provided by BLORT [10] we allow a humanoid robot to autonomously retrieve and reach known objects in indoor environment which model is build from SLAM. The integration scheme we present is simple yet effective and shows a lot of promises for future extensions that will allow to increase the complexity of the scene and incorporate the designated task of the robot within the navigation scheme. This is a first and important step to enable robots to competently perform everyday manipulation activities [11].

We first introduce the tools that are used to perform the object localization and the navigation in an indoor environment using a single commercially available cheap Asus Kinect RGB-D. We then focus on the necessity to integrate such tools and how we did it. Finally, we show

[1] `www.robohow.eu`

978-1-4799-5333-2/14/$31.00 ©2014 IEEE

some results that we obtained on the HRP-2 robot and discuss future improvements to our method.

2. Technological bricks

In this section, we introduce the two key modules that we aim to integrate in order to make the recognition of the object and the navigation towards it possible: an object recognition and tracking toolbox, BLORT, and the SLAM software: D6DSLAM.

2.1 Object recognition

BLORT, the Blocks World Robotic Vision Toolbox, is propose in [10]. It is a open source software that aims at providing a set of tool for robotics to:

- recognize known objects in the environment, and
- track those objects over a sequence of images.

Therefore, it operates in two phases. First, using Scale-Invariant Feature Transforms (SIFT) [12] associated to the objects, and learned prior to operation, it tries to recognize the known objects. Then, the object is tracked over a sequence of images, using a bootstrap filter method [13]. A sample from the outcome of the first phase can be seen in Figure 1a, while the Figure 1b illustrates the result of the tracking phase.

2.2 Navigation in indoor environments

To navigate towards the object recognized by BLORT, we initiate the SLAM software D6DSLAM that is presented in [3] and outlined in Figure 2. D6DSLAM unifies volumetric 3D modeling and image-based key-frame modeling, to provide and update a rich 3D map of an unknown environment as well as provide accurate localization of the robot in its surrounding environment. D6DSLAM proved to be very robust and accurate in robotic scenarios, and in particular, it handles very well the typical sway motion of a humanoid robot during walking, that is usually problematic for visual servoing applications [14]. We indeed tried using BLORT for visual tracking instead of D6DSLAM, but the latter appear to be a more robust percept.

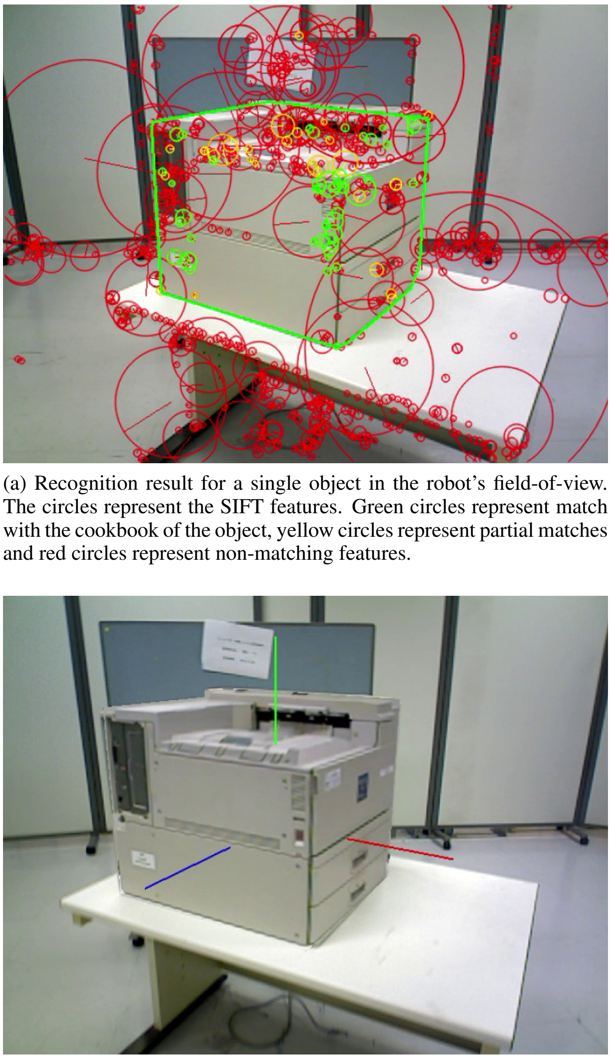
In order to provide information for navigation, the software was augmented with a pixel tracker. It is possible to select a pixel in the current image obtained from the camera. Once this pixel has been selected then its 3D position will be tracked and streamed on the network continuously as the robot progresses and the software builds a larger and more complex map.

3. Integration of object recognition and mapping software

In this section we first discuss the limits of object recognition from an autonomous navigation perspective, and thus the need for an integrative scheme between object recognition and autonomous mapping. We then describe our strategy to combine them.

3.1 Limitations of object recognition for navigation

A toolbox such as BLORT is not limited to object recognition, but can provides tracking capabilities, as we have mentioned previously. Thus, we might also rely on the tracking capability of BLORT to navigate and reach the object or spot of interest. This is however limited for mainly two reasons:



(a) Recognition result for a single object in the robot's field-of-view. The circles represent the SIFT features. Green circles represent match with the cookbook of the object, yellow circles represent partial matches and red circles represent non-matching features.

(b) Tracking result for a single object in the robot's field-of-view. The blue, red and green lines represent the object's frame. Thus reflecting the object's position and orientation in the current view.

Fig. 1: Example of BLORT usage on a printer.

1. On one hand, the tracker obviously requires the object or the image of interest to be in the field-of-view of the camera in order to operate. Unfortunately, in some cases, e.g. fulfill a robotic or task constraints such as obstacle avoidance, it may be necessary to have the object temporarily out of the camera field-of-view.

It is also possible to recognize the object of interest when it re-enters the image, by reinitializing the BLORT tracker. However, this may require additional object learning, since the recognition accuracy varies tremendously with the distance to the object.

Therefore, the robot should navigate towards the object by relaxing the field-of-view inclusion constraints whenever needed without jeopardizing the navigation robustness.

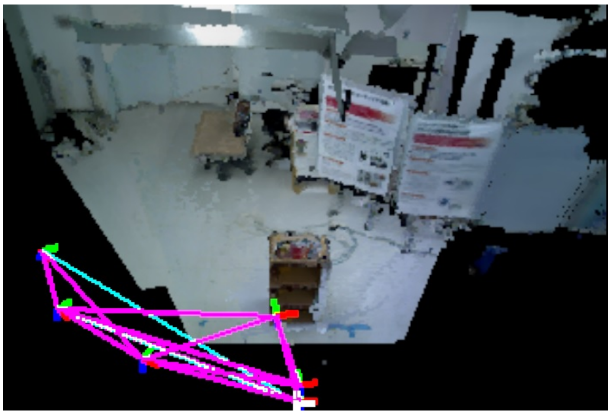


Fig. 2: A view of the environment map generated by the D6DSLAM software and the robot's localization within this map. The nodes of the purple graph represents key-frames while the blue line indicates the current dislocation of the robot compared to the key-frame that is used as a reference for localization.

2. On the other hand, an object tracker provides information only about the object of interest. This is not sufficient to provide a safe route in a changing environment comprised of obstacles, walls, doors, humans, etc.

Previously described causes, suggest that we use object recognition and tracking to search and first retrieve an object of interest in the environment, and eventually updating this position if the latter changes, but we will rely solely on SLAM for vision-based navigation.

3.2 Combining object tracking and SLAM

In order to retrieve known objects in the humanoid surrounding environment, we propose to follow a two-phases strategy:
1. The *research phase* consists in trying to first locate the object within the environment
2. The *object-oriented navigation phase* where we autonomously go towards the object of interest that was previously detected.

We now describe each of these phases: how the recognition information is transmitted to the localization software and finally, how the localization information is provided to the autonomous navigation scheme.

A. Search phase

The goal of this phase is to locate the object. The first step is to query BLORT about the presence of the object within the current field of view of the robot. If the object has been detected by BLORT then we can retrieve the position of the object and dispatch it to the SLAM software. If the object has not been detected, we have to search for the object in another location. This constitutes the core behavior of this 'Search phase'.

In this first implementation, two cases are to be considered:
1. either the robot has not 'seen' everything around it. Then it should rotate on itself and try to find the object in its field-of-view; or
2. the robot has 'seen' everything around it, in which case it will go towards other unexplored locations or request help from the human user. This location can be determined either randomly or provided by the SLAM –for example, an unknown part of the map behind a wall.

However, in our opinion the search phase shall be seriously considered under a semantic SLAM search. Indeed, if we are able to structure SLAM memory in a way where objects are labeled and associated with SLAM data structure, then the object search can boils into a search within the data structure which then provide (at least) a preliminary guess of the location that BLORT would confirm. This would then require the possibility for the robot to plan and reach the presupposed location containing the object of interest using only SLAM data structure. For example, if we are starting at the entry of a corridor with multiple rooms, the current implementation checks in the room on the left, then the room on the right then continue forwards into the corridor to check the following rooms. However, this is definitely not the optimal way. In that case, a semantical approach would allow the robot to first search the rooms that are more likely to contain the object thus saving time in the achievement of its mission [15].

B. From recognition to autonomous navigation

Once the object has been recognized, we determine which pixel corresponds to the centroid of the object. This pixel is then transmitted to the localization software which then starts to track this pixel within the egocentric frame of the robot, in real-time. This tracking information is then used by the robot for autonomous navigation.

Note that if the object position is not conform to the current SLAM memory –for example, in the case where the robot had already processed the object of interest in a previous mission as part of its current SLAM and meanwhile, the object has changed its position–, it is detected as an outliers and the SLAM updated comfortably to its current arrangement in the updated SLAM data. It is this update that will be used for tracking.

C. Object-oriented navigation phase

Given the localization information, the robot starts walking towards the object, using the navigation algorithm described in [16]. This navigation scheme allows us to set a navigation goal, but also to specify way-points that should be reached prior to the destination, and are used to avoid obstacles. Once the object is reached, the task is achieved, and other tasks can be performed to interact with the reached object.

4. Results

In this section, we present the experiments that were conducted with the HRP-2 humanoid robot. We introduce two kinds of experiments:
1. First, a simple experiment where the object is already present in the scene as we start. The goal of this experiment is to validate the use of SLAM in tracking and the autonomous navigation algorithm that we developed.

2. In the second experiment, the object is hidden from the robot at the beginning and the robot must retrieve it and then navigates toward it.

For each demonstration, we describe the actual situation of the experimental room (which is not provided to the robot), the strategy that is devised by the robot to search for the object, the initial configuration provided by the recognition software and the final map built by D6DSLAM. A video of the experiments can be downloaded at this url[2]. In the video, the upper-left corner shows a third-person view of the robot executing the task, the upper-right corner shows the output from the recognition phase – note that this computation does not occur in real-time since SIFT features computation and classification requires heavy computations, the bottom-left corner shows the map being built by SLAM and finally, the bottom-right corner shows the RGB and depth pictures acquired from the embedded camera.

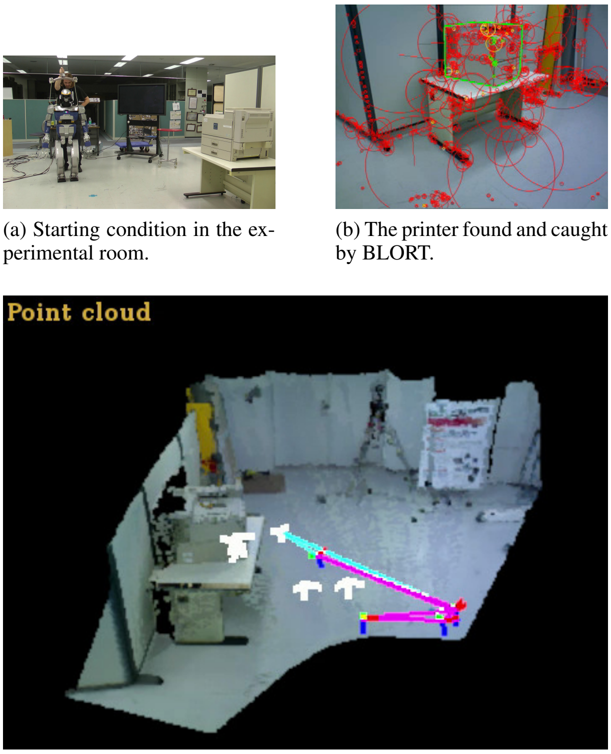
4.1 Experiment 1: object already in the scene

This experiment is the simplest of the two. The robot starts in $(x, y) = (0, 0)$ and the object to find is a printer located on a table at roughly $(x, y) = (1.7, 1.5)$, see Figure 3a. The object is not visible at the beginning. The robot then turns on the left to 'scan' the room and BLORT then ables to locate it, as seen in Figure 3b. Finally, once the printer is caught by BLORT, its centroid is passed to D6DSLAM which the humanoid robot uses to close the loop and navigate autonomously toward the object. We pre-programmed the robot to stop at a safe distance that would allow further manipulations with the printer. The final map produced in this demonstration is illustrated in the Figure 3c.

4.2 Experiment 2: hide and seek

In the second experiment, the humanoid robot's and the printer's initial positions are similar to the previous experiment. However, a wall, as seen in Figure 4a, occludes the vision of the printer by the robot. By this set-up, we aim to somehow mimic the corridor scenario that we introduced earlier in this section. After looking around for the object and not finding it in its immediate surroundings, the humanoid robot decides to go forward as hinted by the user in the beginning. Once this move is completed, it starts looking around again until BLORT is finally able to locate the object, as illustrated by the Figure 4b.

Once the object found, and its centroid given to D6DSLAM, the humanoid robot can then reach the object relaying only on SLAM. It also stops at the same safe distance defined in the previous experiment. The distance between the robot and the printer is computed with respect to its embedded camera. The final map resulted from this demonstration is visible on the Figure 4c.

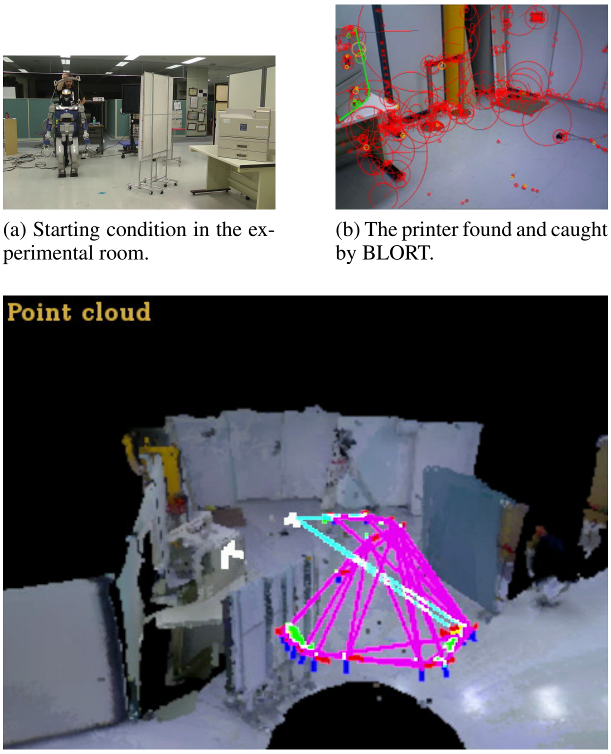


(a) Starting condition in the experimental room.

(b) The printer found and caught by BLORT.

(c) Final map of the scene once the object is reached.

Fig. 3: Results from the first experiment.



(a) Starting condition in the experimental room.

(b) The printer found and caught by BLORT.

(c) Final map of the scene once the object is reached.

Fig. 4: Results of the second experiment.

5. Conclusion

In this paper, we show the benefits of integrating object recognition and SLAM to find learned known objects of

[2] https://dl.dropboxusercontent.com/u/74372876/URAI-2014.mp4

interest in the humanoid surrounding environment. This integration was successfully demonstrated in an experiment with an humanoid robot. This would allow us to properly navigate towards an object before executing further tasks.

The results are promising, however, many problems are still to be tackled. In particular, the SLAM software's use can be further extended. For example, the information provided by the map can be used to provide a collision free path using various waypoints for navigation. This information can also be used to improve the research phase and 'guess' possible locations for the object we are searching. Furthermore, we observed that when using D6DSLAM, rotation of the humanoid robot on itself are not very well handled by the localization. This can be seen in the generated maps, on the Figure 3c and Figure 4c. This problem can be resolved by extending the integration between the D6DSLAM software and the robot control system to provide supplementary information about the robot's attitude and hence improve its estimation while possibly prevent noisy inputs due to fast movements of the camera.

Finally, labeling objects using D6DSLAM's data structure and planning paths based on this data can constitute a complete solution for a humanoid robot to evolve on a daily basis using a simple kinect motion solution.

Acknowledgement

This research is supported by the European Union FP7 IP RoboHow.Cog (FP7-ICT-288533) [3].

References

[1] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part i," *Robotics Automation Magazine, IEEE*, vol. 13, pp. 99–110, June 2006.

[2] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinect-Fusion: Real-time dense surface mapping and tracking," in *IEEE International Symposium on Mixed and Augmented Reality*, (Basel, Switzerland), pp. 127–136, IEEE Computer Society, 26-29 October 2011.

[3] M. Meilland and A. I. Comport, "On unifying key-frame and voxel-based dense visual SLAM at large scales," in *International Conference on Intelligent Robots and Systems (IROS)*, IEEE/RSJ, 2013.

[4] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. J. Kelly, and A. J. Davison, "SLAM++: Simultaneous localisation and mapping at the level of objects," in *IEEE Conference on Computer Vision and Pettern Recognition*, (Portland, Oregon), pp. 1352–1359, 23-28 June 2013.

[5] F. Chaumette and S. Hutchinson, "Visual servo control, Part I: basic approaches," *IEEE Robotics and Automation Magazine*, vol. 13, pp. 82–90, December 2006.

[6] F. Chaumette and S. Hutchinson, "Visual servo control, Part II: advanced approaches," *IEEE Robotics and Automation Magazine*, vol. 14, no. 1, pp. 109–118, 2007.

[7] O. Stasse, B. Verrelst, A. Davison, N. Mansard, F. Saidi, B. Vanderborght, C. Esteves, and K. Yokoi, "Integrating walking and vision to increase humanoid autonomy," *International Journal of Humanoid Robotics,special issue on Cognitive Humanoid Robots*, vol. 5, no. 2, pp. 287–310, 2008.

[8] P. Alcantarilla, O. Stasse, S. Druon, L. M. Bergasa, and F. Dellaert, "How to localize humanoids with a single camera ?," *Autonomous Robot*, vol. 34, no. 1-2, pp. 47–71, 2013.

[9] D. J. Agravante, A. Cherubini, A. Bussy, P. Gergondet, and A. Kheddar, "Collaborative human-humanoid carrying using vision and haptic sensing," in *IEEE International Conference on Robotics and Automation*, (Hong Kong, China), pp. 607–612, 31 May - 7 June 2014.

[10] T. Mörwald, J. Prankl, A. Richtsfeld, M. Zillich, and M. Vincze, "BLORT - The Blocks World Robotic Vision Toolbox," *Best Practice in 3D Perception and Modeling for Mobile Manipulation (in conjunction with ICRA 2010)*, 2010.

[11] M. Li, H. Yin, K. Tahara, and A. Billard, "Learning object-level impedance control for robust grasping and dexterous manipulation," in *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2014. Accepted for publication.

[12] D. G. Lowe, "Object recognition from local scale-invariant features," in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 2, pp. 1150–1157, IEEE, 1999.

[13] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*. Springer, 2001.

[14] C. Dune, A. Herdt, O. Stasse, P.-B. Wieber, K. Yokoi, and E. Yoshida, "Cancelling the sway motion of dynamic walking in visual servoing," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pp. 3175–3180, IEEE, 2010.

[15] M. Tenorth, L. Kunze, D. Jain, and M. Beetz, "Knowrob-map - knowledge-linked semantic object maps," in *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on*, pp. 430–435, Dec 2010.

[16] D. Petit, P. Gergondet, A. Cherubini, M. Meilland, A. I. Comport, and A. Kheddar, "Navigation assistance for a bci-controlled humanoid robot," in *IEEE International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (IEEE-CYBER 2014)*, 2014.

[3] www.robohow.eu