Online Motion Planning for HOAP-2 Humanoid Robot Navigation

Mohammed Elmogy, Christopher Habel, and Jianwei Zhang

Abstract— **Autonomous robot navigation is becoming an increasingly important research topic for mobile robots. In the last few years, significant progress has been made towards stable robotic bipedal walking. This is creating an increased research interest in developing autonomous navigation strategies which are tailored specifically to humanoid robots. Efficient approaches to perception and motion planning, which are suited to the unique characteristics of biped humanoid robots and their typical operating environments, are receiving special interest. In this paper, we present a time-efficient motion planning system for a Fujitsu HOAP-2 humanoid robot. The sampling based algorithm is used to provide the robot with minimal free configuration space which is sampled to extract the robot path. For collision detection, a cylinder model is used to approximate the trajectory for the body center of the humanoid robot during navigation. It calculates the actual distances required to execute different actions of the robot and compares them with the distances to the nearest obstacles. The A* search algorithm is then implemented to find smooth and low-cost footstep placements of the humanoid robot within the resulting configuration space. The proposed hybrid algorithm reduces searching time and produces a smoother path for the humanoid robot at a low cost.**

I. INTRODUCTION

Humanoid robotic hardware and control techniques have been developing rapidly during the last decade. Lately, several companies have announced the commercial availability of various humanoid robot prototypes such as Sony QRIO, Fujitsu HOAP, ZMP NUVO, and Honda ASIMO. The ability of these bipedal humanoid robots to step over and onto obstacles makes them ideally suited for environments designed for humans [1]. They have been attracting many researchers due to their performance of human-like behaviors [1-3]. As the robot body has many degrees of freedom, the robot shape and its available actions are approximated for finding solutions efficiently [2, 4]. In order to improve the autonomy and overall functionality of these robots, reliable sensors, safety mechanisms, and general integrated software tools and techniques are needed.

The navigation autonomy for humanoid robots comprises an increasingly important research area. The development of practical motion planning algorithms and obstacle avoidance

M Elmogy is with Technical Aspects of Multimodal Systems (TAMS), Informatics department, University of Hamburg, Vogt-Kölln-straße 30, D-22527 Hamburg, Germany. (email: elmogy@informatik.uni-hamburg.de).

C. Habel is with Knowledge and Language Processing (WSV), Informatics department, University of Hamburg, Vogt-Kölln-straße 30, D-22527 Hamburg, Germany. (email: habel@informatik.uni-hamburg.de).

J. Zhang is with Technical Aspects of Multimodal Systems (TAMS), Informatics department, University of Hamburg, Vogt-Kölln-straße 30, D-22527 Hamburg, Germany. (email: zhang@informatik.uni-hamburg.de).

software is considered as one of the most important fields of study in the task of building autonomous or semi-autonomous robot systems. This will lead to a rising demand for software and algorithms useful to improve the usability and autonomy of humanoids. The motion planning problem is characterized by the ability to compute a collision-free path of a mobile robot from its initial position to a final position through its workspace. It covers many different aspects of mobile robotics, including path planning among obstacles, optimal path finding among weighted regions, path planning to traverse narrow passages, etc [5]. There are many applications that rely on motion planning, such as: systems for robot guidance, assembly and disassembly, and computer animation [6].

Many researchers [3, 7, 8] have concentrated on various approaches to generate reliable and stable gaits with feedback, and also on developing global navigation autonomy for humanoid robots. Emphasis has primarily been laid on pre-generating walking trajectories, online trajectory generation, and dynamic balance, without accounting for obstacles. Most of them use one of the two main approaches of biped locomotion: static or dynamic. The main objective, in any case, is to produce a gait as natural and stable as possible. Static walkers rely on the static equilibrium condition: maintain the Center of Gravity (CoG) on the convex hull within the contact area with the ground. This approach denies inertial forces. Therefore, it can be applied only if robot movements are very slow. Dynamic walkers achieve fast and natural walking motion following the principle of dynamic equilibrium: they use Zero Moment Point (ZMP) [9] instead of CoG, so that inertia components and gravity are considered [8].

The rest of this paper is organized as follows. Some current related work and an overview of motion planning algorithms are discussed in section 2. In section 3, we will introduce an overview of our humanoid robot navigation system architecture and then discuss the proposed hybrid motion planning algorithm for HOAP-2 humanoid robot in more detail. Section 4 describes the system implementation. Conclusion and future work are presented in section 5.

II. RELATED WORK

Motion planning has been studied for several decades and many motion planning algorithms have been proposed in the literature [10-14]. Robot motion planning algorithms can be classified into being either global or local. While a global method is one that assumes complete knowledge about its environment, a local method assumes partial knowledge.

Given the existing extensive literature on this subject, researchers are starting to look at the problem in a more general and realistic form such as navigation in higher-dimensional spaces and time-varying environments [12, 15, 16].

In mobile robotics, the degrees of freedom are usually small. Therefore, many search techniques can be specially tailored to the path planning problem. If one can neglect orientation, the resulting 2D configuration space can be searched efficiently by grid-based approaches [2, 10, 15] which overlay configuration space (C-space) with a grid. They assume that each configuration is identified with a grid point. At each grid point, the robot is allowed to move to adjacent grid points as long as the line between them is completely contained within free configuration space (C_{free}). This discretizes the set of actions, and search algorithms (such as A*, D*, and dynamic programming) are used to find a path from the start to the destination. These approaches require the setting of a grid resolution. Search is faster with coarser grids, but the algorithm will fail to find paths through narrow portions of C_{free}. Furthermore, the number of points on the grid grows exponentially in the C-space dimension, which makes them inappropriate for high-dimensional problems.

Potential fields [17, 18] and sampling-based algorithms [6, 11] can be used to solve the high-dimensional motion planning problems. Potential field techniques treat the robot's configuration as a point in a potential field that combines attraction to the goal with repulsion from obstacles. These algorithms have the advantage that the trajectory is produced with little computation. However, they are usually incomplete and may fail to find a free path, even if one exists, because they can get trapped in a local minimum. Another problem with these methods is that they are not very suitable to generate an optimal path [19]. On the other hand, sampling-based motion planning techniques provide a faster practical solution by sacrificing completeness, in which a set of sampling points are used to represent the C-space that is used in constructing solutions. Traditionally, sampling-based algorithms are based on uniform sampling which considers the whole environment as uniformly complex and thus the overall sampling density will be equivalent to the density needed by the most complex region. The result is that every region in the C-space has the same computational complexity, which is worst hen narrow passage areas exist in the environment. Furthermore, paths produced by randomized planners usually contain non-smooth segments because of this randomness and the absence of optimization criteria [6].

On the other hand, Global navigation strategies for mobile robots can usually be obtained by searching for a collision-free path in the environment. Because of the low dimensionality of the search space, very efficient and complete (or resolution-complete) algorithms can be employed. For humanoid robots, conservative global navigation strategies can be obtained by choosing an appropriate bounding volume (e.g. a cylinder) [2], and by designing gaits for following navigation trajectories computed by a path planner. Motion planning algorithms that account for system dynamics typically approach the problem in one of two ways: 1) decoupling the problem by first computing a kinematic path, and subsequently transforming the path into a dynamic trajectory, or 2) searching the system state-space directly by reasoning on the possible controls that can be applied.

In our system, we used the cylinder model to approximate the trajectory for the body-center of the HOAP-2 humanoid robot. We approximate the robot's shape by using one cylinder surrounding its body. It allows us to perform the collision detection check in a constant time by comparing the cylinder radius to the distance of the nearest obstacle which is calculated by triangulation. The sampling-based algorithm is used to provide the robot with minimal C_{free} and sample C_{free} to generalize the robot path. We used the Probabilistic Roadmap Mechanism (PRM) with a non-uniform sampling density to avoid the computational complexity of generating a denser search area. The A* search algorithm is then implemented to find smooth and low-cost footstep placements of the humanoid robot within the resulting path in C_{free}.

III. System architecture

We developed a motion planning system for a Fujitsu HOAP-2 humanoid robot to execute the user's navigation tasks in a miniature city environment. Fig. 1 shows the main building blocks of our system and the output from each stage. The robot navigation is based on the route described by the user to generate an initial path estimation which is supplied to the motion planner. The humanoid robot begins from the start point and moves along that path to collect information and recognize the landmarks by using its stereo vision. Based on the new findings and the processed route, the path is then re-planned to adjust the robot's position during navigation.

At the beginning, the user verbally describes the route to the robot by using our route instruction language (RIL) [20]. Then, the route instructions are parsed and analyzed by the instruction interpreter to present the motion actions, spatial relationships, and landmarks in a symbolic representation. The resulting symbolic script is used to generate a topological map of the route to introduce a more abstract representation of the route environment [21].

After generating the topological map, the resulting symbolic representation is divided into segments and supplied to the motion planner as an initial path estimation of the route. The sampling-based technique processes each path segment to get minimal C_{free} and finds an initial plan that takes any known obstacles into account. To calculate suitable footstep placements which should be processed by the humanoid robot to execute the path, we used the A*

search algorithm to generate the shortest and lowest cost sequence of footstep locations. The motion planner returns an optimal sequence of footstep locations according to the cost functions and plausible sets of footstep locations defined. The humanoid robot's cameras are used to capture the current view and process it to recognize the landmarks which are present along the route. A color histogram and SIFT algorithms are used to recognize the route landmarks. The distance estimation between the robot and the processed landmark is calculated by triangulation [22].

In the following subsections, the system building blocks will be explained. The implementation of the online algorithm for robot motion planning and the generation of the humanoid robot footstep location sequence will be elucidated in detail.
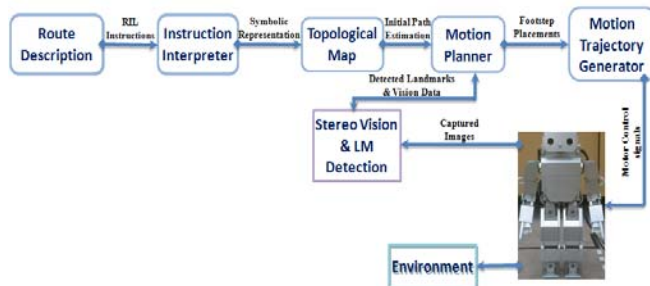

Fig. 1. The motion planning system architecture

A. Route Description and Topological Map Generation

The route is described for the humanoid robot verbally by using our Route Instruction Language (RIL). It is intended as a semi-formal language for instructing robots and it is designed to be an intuitive interface that will be easy and natural for non-expert users to instruct their robots. Fig. 2 shows an example of a route description from the railway station to the McDonald's restaurant in our miniature city.

```
$START( RailwayStation, right)
$GO( forward, into, Street)
$PASS( Saturn, left)
$PASS( BurgerKing, right)
$PASS( C&A, left)
$BE( at, CrossRoads)
$GO( forward, into, Street)
$PASS( Church, left)
$PASS( KarStadt, right)
$BE( at, CrossRoads)
$TURN( right)
$GO( forward, into, Street)
$PASS( TownHall, left)
$STOP( McDonalds, left)
```

Fig. 2. A route description from the railway station to the McDonald's restaurant in our miniature city by using RIL.

After describing the route, the instruction interpreter parses the RIL instructions and then analyzes them syntactically. It also combines definitions from the lexicon according to the parse structure of the instruction, creating a symbolic representation of the route description. This symbolic script is represented in an abstract graphical representation as a topological map to describe relationships among features of the route environment, without any absolute reference system. The topological map prevents the robot from getting trapped in local loops or dead-ends in unknown environments. It supplies the robot with global information about the route in graph-like descriptions, where nodes correspond to significant, easy-to-distinguish landmarks, and arrows correspond to actions or action sequences that connect landmarks. Fig. 3 shows the generated topological map after processing the "railway station –McDonald's" route.
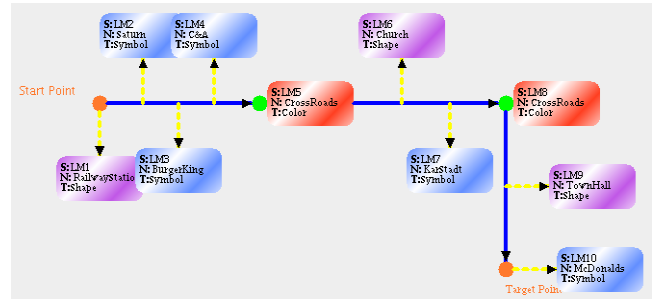

Fig. 3. The topological map from railway station to the McDonald's restaurant in our miniature city.

In the topological map representation, the blue arrows represent the estimated robot path, whereas the yellow dashed lines represent the positions of the landmarks. The rounded rectangles represent the processed landmarks and their colors indicate the type of landmark. Finally, orange circles indicate the start and destination points, whereas green circles represent intermediate nodes in the robot's path [21].

B. Motion Planner

The motion planner is used to accomplish autonomous biped locomotion. It allows the humanoid robot to take advantage of its bipedal capabilities and navigates in its surrounding environment. The planner operates at the level of footsteps and it ignores the lower-level details of leg movements and control. Its main task is to find a sequence of actions as close to optimal as possible that causes the robot to reach the goal location while avoiding the obstacles in the environment. The motion planner has two inputs as shown in Fig. 1. First, the symbolic representation of the processed route is supplied as an initial path estimation. The second is the output of the stereo vision and landmark detection stage which is processed during robot navigation. The motion planner consists of three main components: collision detection, path planning, and footstep planner. In the following paragraphs, the motion planning components will be presented in detail.

Collision detection is considered to be one of the crucial factors in path planning. For humanoid robots, there is an effective and simple way to detect collision by choosing an appropriate bounding volume approximating the shape of the robot. A trajectory for the body-center of a humanoid robot is computed by approximating its shape by using one cylinder surrounding its body as shown in Fig. 4. A cylinder model is useful during humanoid robot turns and lateral

walking to calculate the actual processing space required. As the positions of the nearest obstacles to the robot are calculated by using triangulation, a cylinder model of the humanoid robot can be checked for obstacle avoidance in a constant time. Simply put, if the distance between the robot and the obstacle is known, then it will be compared to the radius of the cylinder model. When the robot walks straightforward or sideways, no additional space is needed. Therefore, the obstacle distance is compared to the cylinder radius plus the expected step distance. Otherwise, as humanoid robots are non-holonomic, they cannot turn in place without requiring additional space. For turns, the humanoid robot wants extra turning space in a cylinder model, and then the cylinder model is enlarged by twice the turn radius to let the humanoid robot turn in a specific direction. Therefore, such an approximation enables a humanoid robot to find paths in real-time and includes actions such as walking sideways through a narrow space.
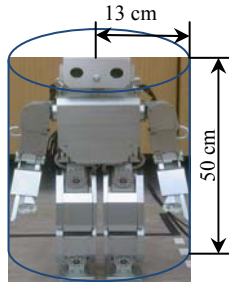

Fig. 4. Approximating the HOAP-2 humanoid robot shape by using one cylinder.

The second component of our motion planner is the path planning stage which processes the route description as segments. Each segment represents the distance between two adjacent landmarks in the robot path. It is processed as an independent path with its own start and end points. The path planning stage is implemented by using a simplified sampling based algorithm which represents a faster practical solution of discovering the search space. It returns only the path not the ability to execute that path. We used the probabilistic roadmap approach (PRM) to sample the C-space, retrieve C_{free}, and determine the robot path. A crucial ingredient of that approach is a sampling algorithm that samples the C_{free} by using a medial axis approach which returns with the minimal free search space to avoid the computational complexity of generating a denser search area. The PRM approach consists of a preprocessing phase and a query phase. In the preprocessing phase, collision-free configurations – called samples – are generated which can be considered as the vertices of the roadmap. Then, a number of attempts are made to connect each sample to its nearest neighbors to generate a roadmap graph (G). In the query phase, the start and goal configurations for each segment are connected to the graph. The path is obtained by performing a shortest path query on the graph. To avoid bad estimations for the path planner, we used limited time for processing and generating a roadmap graph. Finally, the C_{free} will be divided into small cells of 6.5 cm width x 5 cm

height to generate a search grid. The resulting grid of cells will be supplied to the footstep planner with the path graph to search for the best footstep placements of the humanoid robot.

The footstep planner is used to get the shortest footstep placements between the start and the destination in each processed path segment. The A* path planning algorithm [10, 23] is popular due to its accuracy and calculation speed in searching for an optimal solution. It is used to generate the shortest and lowest-cost sequence of footstep locations to reach the target point. A* works by exploring grid nodes (cells) and calculating the cost function F(n) which is the sum of the following three functions:

1) *Step Cost G(n)*: This is the cost of making the desired step from the start node to node n. It is calculated with respect to the values listed in Fig. 5.

2) *Estimated Heuristic Cost H(n)*: This is the estimated cost from node n to the goal. It uses a heuristic search to estimate the cost of the goal node and it minimizes the cost of the path so far. A* is optimal if the estimated cost to the goal is always underestimated. Since the shortest distance between two points is a straight line, Euclidean distance serves as a very accurately estimated cost to the goal, making A* well suited for fast computations.

3) *Clearance Cost C(n)*: It is used to insure that the generated paths are directed to the middle of the empty space (C_{free}) not adjacent to the obstacles. It indicates how far the robot is from an obstacle. It is a normalized cost that is inversely proportional to this distance d, so that the closer the point is from an obstacle the higher its cost, according to:

$$C = \frac{D-d}{d} \qquad (1)$$

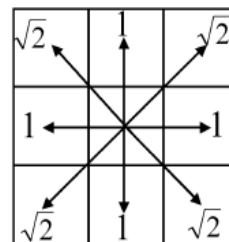where D indicates the clearance distance beyond which the node will be assigned a zero cost.

Fig. 5. The step cost values of the A* search algorithm.

After the cost functions have been calculated, the planner computes the optimal sequence of footstep locations to reach the desired goal. The robot actions are modeled by storing a symmetric collection of candidate footstep transitions for both feet. A sequence of footstep placements to reach a goal in an environment is computed from a discrete set of feasible footstep locations corresponding to stable candidate stepping motion trajectories. The planner returns the solution as an ordered list of the footsteps which should be processed to reach the goal. The motion planner algorithm is

shown in Fig. 6. The planner searches the space until it finds a path to the goal or exceeds a specified time.

```
Cf.Insert(P0, 0, 0, 0,NULL);
foreach Segmenti ∈ Path do
    c←a configuration in Cfree
    V←V ∪ {c}
    Nc ←a set of nodes chosen from V
    for each c0 ∈ Nc do
        if c0 and c are not connected in G then
            if a path between c0 and c is found then
                add the edge c0c to E
    end
    while runningTime < tmax do
            Pb← Cf.ExtractMinimum();
            if GoalReached(Pb, Pg) then
                return Pb;
            end
            e←E(Pb.time);
            A←F(Pb,e);
            foreach a ∈ A do
                Pn←T(Pb, a, e);
                Gn←LocationCost(e, Pn);
                Hn←ExpectedCost(e, Pn);
                Cn←ClearnessCost(e, D, d);
                Cf.Insert(Pn, Gn, Hn, Cn, Pb);
            end
        end
    end
end
```

Fig. 6. Motion planner Algorithm

where C_f is the candidate footstep displacement and has five arguments: start step, step cost, estimated heuristic cost, clearance cost, and the desired step. The T is footstep transition which calculates action effects, A is the set of possible actions that can be commanded to the humanoid robot, and E is the set of environments. G_n, H_n, C_n are step, heuristic, and clearness costs, respectively. P_0 and P_b represent the start and best points. The maximum allowed time of the footstep search is t_{max} to prevent end loops.

C. Motion Trajectory Generator

After the footstep placements are estimated, the walking parameters are calculated and sent to the humanoid robot's actuators to generate its motion. We considered the walking process as a symmetric, periodic and smooth motion. The ZMP walking pattern is used to produce humanoid robot gaits as dynamic and stable as possible to execute the robot's motion. The foot placement actions indicate the motion, turns, and change of orientation actions. These actions are divided into six humanoid footstep placement actions: straightforward, straight backward, turn right, turn left, sideways right, and sideways left. Table 1 shows some foot placement actions and their parameters for the HOAP-2 humanoid robot.

On the other hand, the robot's head is moved depending on the direction of landmarks and the movement range of the neck's motors. The robot will tilt its head to the right and left to detect the landmarks which are located at the road sides. It also looks down to the floor to detect landmarks such as crossroads and street boundaries in the miniature

city. The head orientations of the humanoid robot are divided into four actions: turn right, turn left, move up, and move down. The humanoid robot can turn its head by 45º in both the right and left directions. It can also raise its head by 45º and lower it by 15º.

TABLE I
HOAP-2 FOOT PLACEMENT ACTIONS

Action	Straightforward	Turn Right	Sideways Right
Footstep Shape			
Distance	0-10 cm	0-10 cm	0-4 cm
Angle	0º	0º-17º	0º-17º

IV. IMPLEMENTATION

We have implemented our approach on the second generation of Fujitsu's Humanoid for Open Architecture Platform (HOAP-2) [24]. HOAP-2 is equipped with 25 servo actuators: six for each leg, four for each arm, one for each hand, two for the head, and one for its waist. It has four force sensing registers (FSRs) in each foot to detect reaction forces from the floor. It is also equipped with an accelerometer and gyroscope inside the torso. The vision system consists of two 0.25`` CMOS cameras, capable of capturing frames of 320X240 pixels at 25fps.

The navigation task is described by the user as route instructions via a graphical user interface (GUI). The navigation process is executed in a miniature city which is built on a 5m x 3.2m area. It contains buildings of recognizable shapes and colors (such as the railway station and the town hall) and other buildings of unique characteristics and symbols as shown in Fig. 7.

Fig. 7. The miniature city.

We tested the proposed algorithm on simple routes (such as in Fig. 2) and it takes approximately 1.7 sec to process each segment in the processed route. Fig. 8 shows some snapshots of the "railway station – McDonald's restaurant" route. We are still working on handling long and more complex routes which contain narrow passages.

V. Conclusion And Future Work

In this paper, we presented our navigation system for a HOAP-2 humanoid robot. It is based on the user's route description to generate an initial path estimation which is supplied to the motion planner. The route is processed to generate motion actions and spatial relationships which are represented in an abstract representation as a topological map. The resulting map describes the relationships between route landmarks, without any absolute reference system.

On the other hand, we presented a hybrid motion planning approach for the HOAP-2 humanoid robot to generate time-efficient and low-cost paths. The sampling based technique is used to construct an approximate representation of the free configuration space and returns with a grid of nodes to the footstep planner. The cylinder model is used to detect collisions in a constant and small amount of time. The footstep planner, which uses the A* path planning algorithm, is used to generate the shortest and lowest cost sequence of footstep locations for the humanoid robot to reach the target.

We tested our system by using simple route descriptions. In the future, we will use more complex routes such as routes with narrow passages and try to compensate for our algorithm to deal with these problems.
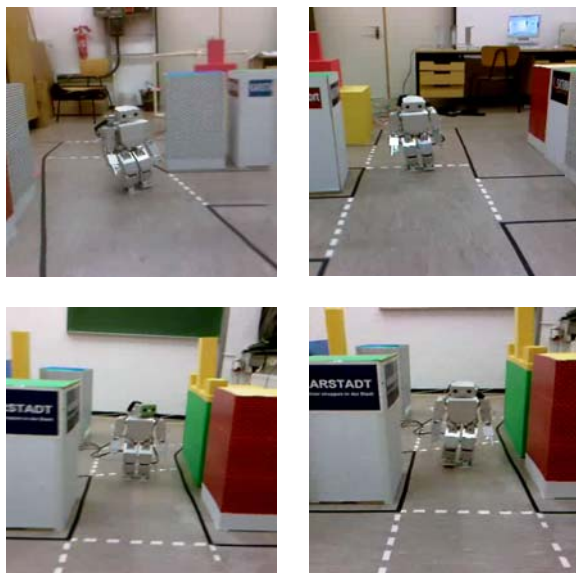

Fig. 8. HOAP-2 navigation in the miniature city

References

[1] M. Ogino, M. Kikuchi, J. Ooga, M. Aono and M. Asada, "Optic flow based skill learning for a humanoid to trap, approach to, and pass a ball," Robocup04, 2004 .

[2] J. Gutmann, M. Fukuchi and M. Fujita, "Real-time path planning for humanoid robot navigation," International joint conference on artificial intelligence (IJCAI-05), 2005, pp. 1232-1237.

[3] P. Michel, J. Chestnutt, J. J. Kuffner and T. Kanade, "Vision-guided humanoid footstep planning for dynamic environments," Proceedings of 2005 5th IEEE-RAS international conference on humanoid robots, 2005, pp. 13-18.

[4] R. Chalodhorn, D. B. Grimes, G. Maganis and R. P. N. Rao, "Learning dynamic humanoid motion using predictive control in low dimensional subspaces," Humanoid robots, 2005 5th IEEE-RAS international conference, 2005, pp. 214-219.

[5] G. E. Jan, K. Y. Chang and I. Parberry, "Optimal path planning for mobile robot navigation," IEEE/ASME Transactions on Mechatronics, vol. 13, no. 4, Aug., 2008, pp. 451-460.

[6] T. Taha, J. Valls Miro and G. Dissanayake, "Sampling based time efficient path planning algorithm for mobile platforms," Proceeding of the IEEE international conference on man-machine systems (ICOMMS 2006), Sep., 2006.

[7] K. Sabe, M. Fukuchi, J. Gutmann, T. Ohashi, K. Kawamoto and T. Yoshigahara, "Obstacle avoidance and path planning for humanoid robots using stereo vision," IEEE international conference on robotics and automation (ICRA'04), 2004.

[8] J. P. B. Rubio, C. Zhou and F. S. Hernández, "Vision-based walking parameter estimation for biped locomotion imitation," In Perception and Robotics, Vol. 3512/2005, 2005.

[9] M. Vukobratovic and B. Borovac, "Zero-moment point -thirty five years of its life," International Journal of Humanoid Robotics, vol. 1, no. 1 , 2004, pp. 154-173.

[10] J. Latombe, "Robot Motion Planning," Kluwer Academic Publishers, 1991.

[11] S. M. LaValle, "Planning algorithms," Cambridge University Press, 2006.

[12] A. M. Hussein and A. Elnagar, "A fast path planning algorithm for robot navigation with limited visibility," Systems, man and cybernetics. IEEE international conference, Oct., 2003, pp. 373-377.

[13] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, Principles of robot motion-theory, algorithms, and implementations. The MIT Press, 2005.

[14] M. Lau, and J. Kuffner, "Behavior planning for character animation," in 2005 ACM siggraph / eurographics symposium on computer animation, Los Angeles, CA, 2005, pp. 271-280.

[15] J. Kuffner, S. Kagami, K. Nishiwaki, M. Inaba and H. Inoue, "Online footstep planning for humanoid robots," IEEE int'l conf. on robotics and automation (ICRA'2003), Sep., 2003.

[16] O. Stasse, B. Verrelst, A. Davison, N. Mansard, B. Vanderborght, C. Esteves, F. Saidi, and K. Yokoi, " Integrating walking and vision to increase humanoid robot autonomy," in 2007 IEEE international conference on robotics and automation, 2007, pp. 2772-2773.

[17] P. Khosla and R. Volpe, "Superquadratic artificial potentials for obstacle avoidance and approach," Proceedings of the IEEE conference on robotics and automation (ICRA '88), April, 1988.

[18] E. Rimon and D. E. Koditschek, "Exact robot navigation using artificial potential function," IEEE Transactions on Robotics and Automation, vol. 8, no. 5, Oct, 1992, pp. 501-518.

[19] J. Yu, Q. Zhang,V. Kroumov,S. Cheng and Z. Zhang, "Path planning algorithm for robot in 3d environment based on neural network," Intelligent robotics and applications, 2008, pp. 1081-1088.

[20] M. Elmogy, C. Habel, and J. Zhang, " Spatial language for route-based humanoid robot navigation," In proceedings of the 4th international conference on spatial cognition (ICSC09), Roma, Italy, 2009.

[21] M. Elmogy, C. Habel and J. Zhang, "Robot topological map generation from formal route instructions," In proceedings of the 6th international cognitive robotics workshop at 18th european conference on artificial intelligence (ECAI), Patras, Greece, July, 2008, pp. 60-67.

[22] M. Elmogy and J. Zhang, "Robust real-time landmark recognition for humanoid robot navigation," In proceedings of the 2008 IEEE international conference on robotics and biomimetics (ROBIO), Bangkok, Thailand, Dec., 2008 .

[23] J. Barraquand, L. E. Kavraki, J. C. Latombe, T. Y. Li, R. Motwani and P. A. Raghavan, "Random sampling scheme for path planning," International Journal or Robotics Research, vol. 16, no. 6, 1997, pp. 759-774.

[24] Fujitsu Automation Co., Ltd., "HOAP-2 Instruction Manual," Third Edition, 2004.

[25] J. J. Kuffner, K. Nishiwaki, S. Kagami, M. Inaba and H. Inoue, "Motion planning for humanoid robots under obstacle and dynamic balance constraints.," In proc. IEEE int'l conf. on robotics and automation (ICRA'2001), Seoul, Korea, May, 2001, pp. 692-698.