Adaptive Level-of-Detail Planning for Efficient Humanoid Navigation

Armin Hornung   Maren Bennewitz

Abstract— In this paper, we consider the problem of efficient path planning for humanoid robots by combining grid-based 2D planning with footstep planning. In this way, we exploit the advantages of both frameworks, namely fast planning on grids and the ability to find solutions in situations where grid-based planning fails. Our method computes a global solution by adaptively switching between fast grid-based planning in open spaces and footstep planning in the vicinity of obstacles. To decide which planning framework to use, our approach classifies the environment into regions of different complexity with respect to the traversability. Experiments carried out in a simulated office environment and with a Nao humanoid show that (i) our approach significantly reduces the planning time compared to pure footstep planning and (ii) the resulting plans are almost as good as globally computed optimal footstep paths.

I. INTRODUCTION

While path planning for wheeled robots can be considered to be largely solved, navigating with robots possessing a higher number of degrees of freedom, such as humanoid robots, is still a challenging problem. For wheeled robots operating in a planar, two-dimensional world, it is usually sufficient to compute 2D paths in a grid-based representation of the environment. These plans can be computed very efficiently and can be easily followed.

This method can be also applied to humanoid robot navigation with controllers to follow the collision-free 2D paths, e.g., by planning footsteps in a local area around the 2D path [1], [2], or by executing fixed gaits [3]. However, the 2D path itself may be non-optimal for humanoids since it does not consider the capability to avoid obstacles by stepping over them. For example, walking through obstacles lying on the floor or climbing stairs is only possible with discrete stepping motions. A conventional 2D planner would have to choose a large detour or return no solution at all. Several approaches, therefore, compute a sequence of footstep actions that the robot executes (e.g., [4], [5], [6]). In these approaches, planning is carried out in the space defined by a given set of footsteps, which is more computationally demanding than computing a 2D path due to the higher number of possible state transitions. However, the resulting paths are typically shorter than the ones resulting from 2D planning in cluttered environments. There are further techniques for planning whole body motions to pass difficult areas including steep or rough terrain. This is, however, seriously more complex due to the high-dimensional search
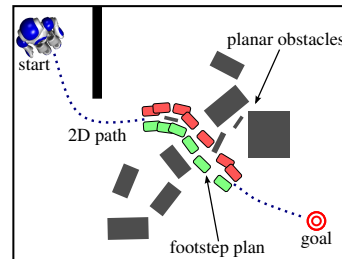

Fig. 1. We classify the environment into regions of different complexity where adequate planners are employed. For open spaces, planning a coarse 2D path is sufficient. A detailed plan requiring more complex computation (e.g., a footstep plan) is only generated where needed.

space and is not feasible yet for longer sequences during real-time navigation [7], [8].

In this paper, we propose to combine detailed motion planning in areas requiring complex navigation capabilities and coarse global path planning in an efficient manner. Our approach classifies the environment into areas of different complexity and samples transitions to connect the individual regions. Based on the classification result, the system decides at which level of detail to plan the motion.

The advantages of our approach are two-fold. First, efficient paths can be found since the difficult regions can be passed instead of choosing detours around them. Second, the computational burden of globally planning detailed motions is seriously reduced. Our framework is generally applicable for planning motions at different levels of detail depending on the type of environment and the robot's capabilities. It can be also applied to plan with different modes of locomotion for different types of terrain [7].

Here, we consider humanoid navigation in indoor environments. We apply fast 2D planning for open spaces and use footstep planning to step over planar obstacles such as clutter on the floor, uneven surfaces, or other areas that have to be avoided in order to prevent a fall (see Fig. 1). To transition between the individual areas, we densely sample the border regions and use footstep planning in a precomputation step to estimate the traversal costs for the difficult areas.

As the experiments obtained from simulation and with a real humanoid show, our adaptive planning strategy has significantly lower computational costs than pure footstep planning while the resulting paths are similarly as efficient as globally computed footstep paths. As a further contribution, we extended our footstep planner implementation [4]. As a result, we can now use Anytime Dynamic A* (AD*) for an efficient replanning of footstep sequences[1].

[1]Open source implementation in ROS available at
`http://www.ros.org/wiki/footstep_planner`

II. RELATED WORK

Research in the area of collision-free motion planning for humanoids differs in terms of level-of-detail and planning horizon. Several techniques concentrate on generating only the next movements of the robot [9], [10]. Other approaches plan discrete walking actions on a discretized grid [3], [11]. A drawback of these planning methods is that they are prone to end up in local minima, as their fixed motion primitives (e.g., walking straight, sideways, and turning in 45° steps) do not correspond to the full flexibility of a humanoid, e.g., to step over obstacles.

Chestnutt *et al.* [5] proposed to plan the whole path entirely with footsteps using A*. To make the search feasible, the authors defined a set of seven possible actions. Later, the authors extended their technique so that invalid footsteps can be adjusted in a local area around the reference action [6]. Recently, Garimort *et al.* [4] introduced efficient, dynamic re-planning using D* Lite for footstep planning. While footstep planning exploits the stepping capability of humanoids and is more efficient than planning for the whole body, it is still computationally demanding for long distance path planning.

Further approaches plan a global 2D path for the robot and then locally determine footsteps to follow the trajectory [1], [2]. While these methods enable fast planning, in contrast to our work they are prone to end up with non-optimal paths in local minima as the global planner is not aware of the local planner's capabilities.

Some authors developed techniques to generate whole body motions that also contain non-upright gaits. For example, Hauser *et al.* [12] presented a probabilistic planner that first samples contacts of predesignated parts of the robot's body and points of the terrain which are subsequently connected by a probabilistic roadmap method. In a later work, the authors make use of precomputed motion primitives to guide the search and generate high-quality motions [7]. Kanoun *et al.* [8] combine footstep planning with inverse kinematics to reach kinematic goals. The whole body is controlled so that complex tasks, such as picking up an object from the ground, can be carried out. The general problem of planning whole body motions is that these approaches require long planning times and are only applicable to short paths. A promising approach, however, would be to include whole-body motions as another mode of navigation in our framework. A whole body planner would be invoked only where needed, e.g., to climb ladders or traverse rough terrain.

Finally, there are motion planning approaches which also plan at a varying level of detail or use a combination of planners [13], [14], [15]. These methods adapt the planning level in order to obtain accurate short-term results, and only rough long-term results. This is particularly useful when planning in highly dynamic environments. In order to avoid local minima and decide on the feasibility of the plan in more static scenarios, however, it is necessary to compute a complete plan instead of using a set of local plans.

The authors in [16] and [17] plan motions in non-flat environments by first decomposing the map into different
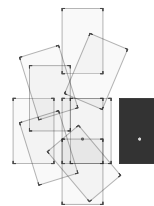

Fig. 2. Footstep parameterization for a humanoid with ten different steps, shown as displacements of the left foot relative to the right supporting foot.

regions that are locally 2D. Depending on the terrain classification (e.g., slope, stairs, or flat ground), they use a local planner based on motion primitives defined for that region. The regions are then connected by heuristically choosing sparse transitions and applying a global planner. While this enables real-time capable planning, the results strongly depend on the manually defined motion primitives and may again be globally not optimal due to the prior decomposition and the sparse transition between the regions. Therefore, our approach uses a dense sampling of transition points and allows an adaptive switching between the differently detailed planners.

Using a similar idea as in our approach, Morales *et al.* [18] proposed to classify the planning space with machine learning techniques in order to apply different roadmap-based planners. Since this technique was particularly designed for roadmap planners, it is unclear how it can be applied to using separate search-based planners operating in different search spaces.

In contrast to all these approaches, we plan a complete path from the start to the goal by using detailed footstep planning to cross complex obstacle regions and otherwise applying a fast, more coarse 2D planner. In this way, our approach generates highly efficient paths for humanoids.

III. FOOTSTEP PLANNING

We first recapitulate footstep planning, which constitutes a common approach for humanoid motion planning. By computing a sequence of collision-free footstep positions to the goal, this method inherently takes the capability of humanoid robots to step over obstacles into account. A discrete set of footstep transitions, illustrated in Fig. 2, is used in a heuristic search such as A^* [5]. States hereby consist of the position and orientation (x, y, θ) of the current supporting foot. The next state can be reached by applying one of the possible footstep actions and changing the supporting foot.

Starting from the initial state, the planner successively adds footstep transitions in order to find the most cost-efficient path to the goal. The transition costs are given by the costs of executing the corresponding step, i.e., they are based on the distance the step covers and a constant cost in order to favor paths with fewer steps. Each new state is checked for collisions and is discarded in case it collides with obstacles. Otherwise the planner expands it further by investigating the next transitions. The search is hereby guided by a heuristic that helps to focus on promising states leading to the goal.

Common heuristics for footstep planning are to use the straight-line distance to the goal, or the estimated costs along
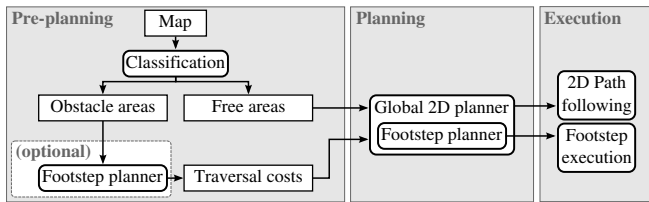
Fig. 3. Overview of our system. See text for explanation.

a 2D path planned with A*. The latter is potentially inadmissible because it does not reflect the humanoid's capability of stepping over obstacles. In practice, however, the resulting paths are usually no less efficient and significantly faster to compute as the heuristics guides the search more focused towards the goal [4], [5].

Note that, in contrast to standard 2D path planning, footstep planning takes also the orientation of the states into account. The higher number of possible state transitions and the more complex collision checks are the reasons why footstep planning is computationally more demanding than 2D path planning.

For this work, we extended our footstep planner [4] to build upon the Search-based Planning Library (SBPL, [19]). We apply the Anytime Dynamic A* (AD*) search [20] for efficient anytime replanning during navigation.

IV. ENVIRONMENT REPRESENTATION AND CLASSIFICATION

A. Environment Representation

We assume the environment to be given and represented as an annotated 2D grid map containing planar and non-planar obstacles. Each cell of the grid map contains information about whether or not the cell is covered by an obstacle and about the type of the obstacle. Planar obstacles correspond to areas which the robot can avoid by stepping over, such as uneven floor, clutter, or edges of stairs. Non-planar obstacles are walls or furniture, which the robot generally has to avoid.

B. Classification and Segmentation

Our proposed method, illustrated in Fig. 3, first classifies a given environment map M into regions of different complexity where individual planners are applied. In this work, we consider two different complexities: wide areas in which it is sufficient to plan 2D paths for safe navigation and areas containing narrow passages or planar obstacles requiring more detailed planning.

To classify the environment, we segment the map based on the humanoid's walking circumcircle, also considering its swaying walking motion. This corresponds to the clearance needed during walking when following a 2D path while neglecting more detailed motion planning. Using a distance map D containing the Euclidean distance to the closest obstacle for each cell (x, y), we determine connected areas larger than the clearance radius r. In the corresponding regions, 2D path planning can be applied to generate efficient paths avoiding obstacles. This *free* area is denoted as $A \subseteq M$
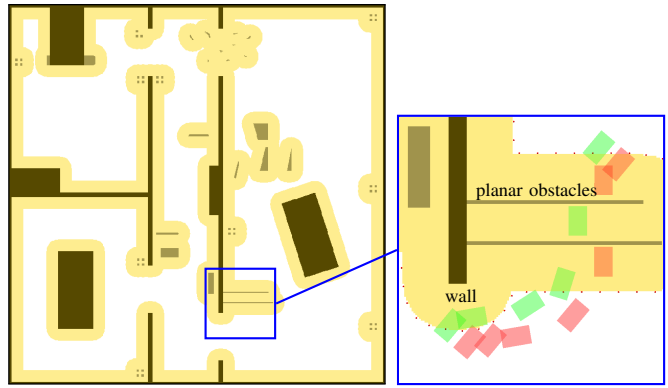

Fig. 4. Classification of an indoor environment: General obstacles are black, planar obstacles gray. The white area was classified to be suitable for 2D planning. In the yellow areas, a footstep planner can be used to plan efficient, collision-free paths. The close-up on the right shows sampled contour points (red dots) and an exemplary footstep path between two points.

and computed according to

$$A = \{(x, y) \in M \mid D(x, y) \geq r\} , \tag{1}$$

followed by a segmentation into individual disjunct regions

$$A = A_1 \cup A_2 \cup \cdots \cup A_n . \tag{2}$$

In standard 2D planning with an enlarged robot circumcircle, the remaining areas $F = M \setminus A$ would be avoided to not risk collisions. In contrast to that, our approach applies more detailed, complex planning in these regions, taking into account more degrees of freedom to find collision-free paths. This directly results in shorter paths since, e.g., obstacles can be closely passed or stepped over. F is also segmented into disjunct regions

$$F = F_1 \cup F_2 \cup \cdots \cup F_m , \tag{3}$$

which we call *obstacle* or *footstep* regions.

Fig. 4 displays the classification of a typical indoor environment with various planar and non-planar obstacles.

V. EFFICIENT PLANNING

We now present our planning approach which relies on the classified and segmented environment. We first describe how traversability costs for the obstacle areas F_i are estimated, which are subsequently used to aid global planning.

A. Estimation of Traversability Costs

The contour $C(F_i)$ of a footstep area F_i contains the 2D map cells of the free area A which lie on the border to F_i. To determine the actual traversability of all F_i and to estimate the corresponding path costs, we densely sample pairs of entry and exit points t_j:

$$T_i = \{t_1, \ldots, t_{n_i}\} \subset C(F_i) \times C(F_i) \tag{4}$$

Our system then applies a footstep planner for all pairs in $T = \bigcup_i T_i$. At this stage, the start and goal orientations for footstep planning are given by the straight-line connection between the corresponding entry and exit points and we apply footstep planning with the admissible straight-line heuristic (see Sec. III) to connect them.
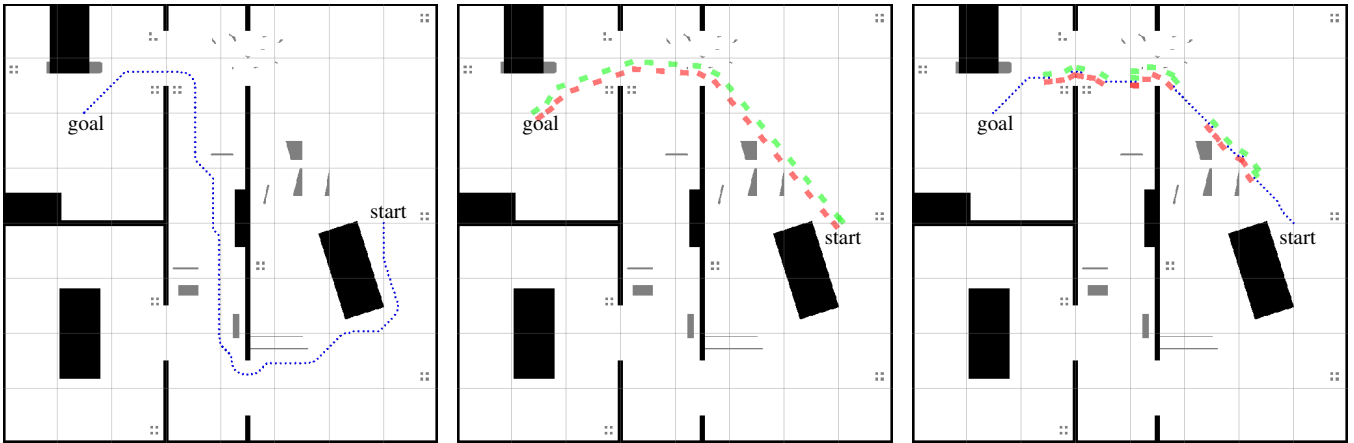
Fig. 5. Comparison of different path planning methods for a humanoid in an indoor office environment. *Left*: A conventional 2D path (blue dots) can be quickly computed but takes detours around cluttered areas and needs to respect larger safety margins to obstacles. *Middle*: A footstep path exploits the stepping capabilities of humanoid robots resulting in a more efficient path, but takes substantially more time to compute. *Right*: Our proposed approach combines the advantages of both methods by using a conventional 2D path planner in open spaces and footsteps planning in more complex regions.

If a footstep path is found, the resulting costs yield an estimate of traversing the area F_i from the corresponding entry point to the exit point. Note that this value is only an estimate for the global planning stage later, since then the pose orientation may be different from the one used for precomputation. In this way, we obtain the costs of traversing the obstacle region F_i between each pair of states in T_i.

The preplanning process is carried out once for a given map. The estimation is reasonably fast because the footstep plans are comparably short and they can be easily parallelized as all plans are independent. To avoid the precomputation e.g. in non-static environments, a heuristic can be derived from once learned traversal costs. It estimates the costs based on the the average traversal costs normalized by the straight-line distance.

Note that the above technique cannot handle situations in which start or goal are inside an obstacle area. To deal with these situations, we sample contour points inverse proportional to their Euclidean distance to the starting/goal location and perform footstep planning to connect them.

B. Global Planning

Input to the global planner, which computes the path between the current robot pose and the desired goal state, is an augmented map consisting of an annotated 2D map M, regions A_i and F_i, and the costs of traversing each F_i for the transitions T_i. For global planning, we allow transitions between all neighboring cells in the free areas A and additionally between all entry and exit points in the T_i with the precomputed traversal costs.

Note that it is important to have a unified cost metric for all different planners. In our case, the footstep planning costs correspond to the time it takes the robot to execute the footsteps. Hence, we scale the costs of 2D paths in the map so that they are normalized with respect to the footstep planner. Accordingly, the costs for a 2D path of a certain length is the same as walking on a footstep plan of the same distance between start and goal. By adjusting the scaling factor, it is possible to give preference to one plan over the

other, e.g., when single footsteps are executed slower than a fast path following behavior.

Global planning now corresponds to an AD* search that proceeds by expanding states in A and T. The heuristic that estimates the costs based on the straight-line distance to the goal hereby guides the search. Whenever the goal is reached, the computed path so far consists of a sequence of 2D path segments and obstacle region traversals. Then, a footstep plan for each segment crossing an obstacle region is generated. Opposed to the preplanned estimate, this footstep plan now uses the correct orientation of the entry and exit poses which is given by the connecting 2D path segments.

This global planning strategy is highly efficient and yields cost-minimal paths in terms of the given cost metric. Only due to the sampling of transitions, the paths might differ from the optimal footstep paths found by globally planning footstep actions, which we will evaluate in Sec. VI.

C. Plan Execution and Replanning

For execution, the robot can follow the 2D paths easily by walking with an omnidirectional controller to sub-goals on the path. As soon as a footstep plan segment is reached, the robot then switches to the footstep controller.

Note that it is necessary to compute a complete path at the beginning to decide on the feasibility of the plan and to avoid local minima. During plan execution, changes in the environment or an updated localization estimate may render the current plan invalid. With AD* planning, however, parts of the original plan can be reused for efficient replanning in these cases [4].

VI. EXPERIMENTS

We finally present experimental results obtained in a simulated environment and in a real robot experiment. The simulation environment consists of a hallway with three rooms in a $8 \times 8\,\mathrm{m}^2$ office (see Fig. 4). This environment contains various planar obstacles (e.g., outlets, clutter, edges) as well as non-planar ones (e.g., walls and furniture). We used a map resolution of $1\,\mathrm{cm}$ for accurate collision checks.

The footstep parameterization for the footstep planner is displayed in Fig. 2.

A. Precomputation

First, we evaluate the computational costs required to estimate the traversal costs in the precomputation phase. Table I displays the average precomputation times for different densities of sampled contour points T and the planning performance for ten different planning problems as average and standard deviation. We exploit parallelization in the classification phase by running four threads on a 3.4 GHz Intel Core i7 CPU. As can be seen, a costly dense estimation is not required to yield acceptable planning performance in terms of time and path costs. Only when there are too few transitions across obstacle areas (more than 0.6 m apart), the performance degrades. We hence use a distance of 0.2 m between the contour points of footstep regions.

B. Qualitative Evaluation

Fig. 5 shows the resulting plans of the different methods for an example scenario. Planar obstacles are displayed in gray, all other obstacles in black. The 2D plan (left image) was fast to compute in less than a second, but takes the robot on a longer path as it requires detours. The global footstep plan (middle image) takes substantially longer to compute (29 s), but results in a more efficient path as planar obstacles are stepped over instead of walking around them. For this global footstep planning, we used the shortest 2D path as heuristic during the search to enable the fastest planning times and required AD* to converge to the optimal solution. Compared to that, our approach combines the advantages of both in that it is as fast as 2D planning and results in an efficient path (right image). Footstep planning is only invoked where needed. The path costs in this scenario are only 2% higher compared to the footstep plan, while being 51% less compared to 2D planning.

C. Statistical Evaluation

For a thorough comparison, we plan a 2D path, a footstep path, and a path using our adaptive method between ten different start and goal configurations in the environment, each requiring a path length of approximately 8 m in the optimal case. As evaluation criteria we use the planning time (on a single core of a 3.4 GHz Intel Core i7 CPU) and path costs for each plan. The path costs are normalized with respect to footstep planning, as explained in Sec. V-B.

Table II displays the aggregated results as average and standard deviation. Footstep planning yields the best paths, but takes up to 94 s to return the solution in the most complicated scenario. Conventional 2D planning is faster, but the resulting paths are significantly longer as they cannot pass close to obstacles or step over objects. In contrast to that, our adaptive approach yields fast results and leads to paths that are as efficient as full footstep plans. The average planning time, which includes planning the short footstep sequences to pass obstacles, is even smaller than with 2D planning since fewer states are expanded.

Precomputation			Planning performance	
Density [m]	# plans	time [s]	time [s]	Path costs
0.05	32266	3640	0.44 ± 0.18	12.05 ± 1.11
0.1	7820	884	0.47 ± 0.20	11.97 ± 1.19
0.2	1892	202	0.41 ± 0.32	11.78 ± 1.08
0.4	516	51	0.49 ± 0.47	12.16 ± 1.16
0.6	236	27	0.35 ± 0.20	12.47 ± 0.68

Approach	Planning time [s]	Path costs
2D planning	0.51 ± 0.11	19.18 ± 5.28
Footstep planning	43.70 ± 25.66	11.78 ± 1.16
Adaptive with precomputation	0.41 ± 0.32	11.78 ± 1.08
Adaptive, no precomputation	0.99 ± 0.54	12.41 ± 1.43

Precomputing footsteps to estimate the traversal costs is not always practical in dynamically changing environments. Thus, we derive a heuristic from our classification as described in Sec. V-A. Planning performance with no precomputation available results in 5% longer plans with a small increase in planning time, as more non-relevant footstep segments are planned.

D. Real-world Evaluation

We now evaluate our approach in a real-world scenario with a Nao humanoid robot. As before, we compare a 2D plan, a footstep plan, and our adaptive planning approach. In this experiment, the robot has to pass a narrow passage of planar obstacles with start and goal approximately 0.85 m apart.

Our humanoid is equipped with a Hokuyo URG-04LX laser range finder mounted in a modified head. By interpreting the data of the laser range finder and the robot's proprioception, our localization system can account for motion drift and accurately determine the robot's pose [21].

Nao's walking engine can be controlled with both omnidirectional velocities for path-following and single footstep placements for more accurate control. When following 2D paths, we can let the walking engine generate the joint angle trajectories in real-time but there is no direct control over the exact foot placements [22]. With footstep control, Nao can walk on a planned sequence of footsteps. For footstep planning, we use a set of 12 footsteps in Nao's stepping range [4].

The environment and the resulting 2D path is shown in Fig. 6. Because there is no control over the exact foot placements when following a 2D path, a larger clearance (shaded gray) is needed to ensure collision-free motion. In this scenario, this results in a detour with twice the path costs of the other plans. While a smaller clearance allows the robot to avoid the detour, it results in a collision with the obstacles (Fig. 6, right). Both of these 2D paths can be planned within less than 0.01 s.
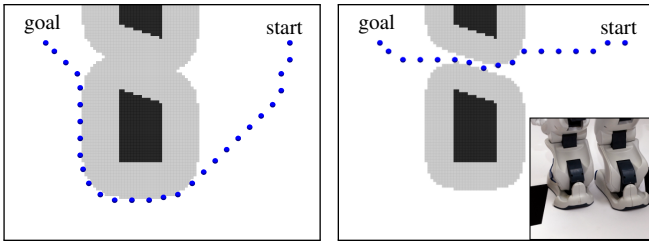
Fig. 6. *Left*: A conventional 2D path for the Nao humanoid has to respect a larger clearance to ensure collision-free motion, leading to detours around narrow passages. *Right*: Using a smaller distance to obstacles allows the 2D path to pass the obstacles, but risks collisions because there is no direct control over the exact foot placement.
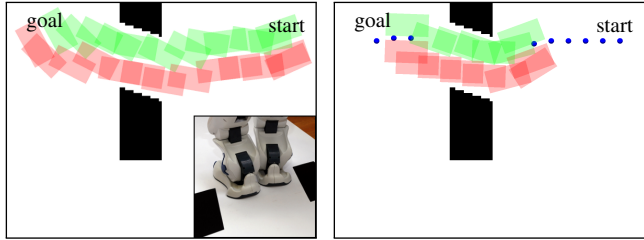

Fig. 7. *Left*: Footstep planning allows the robot to safely pass tight spaces with no collisions, but requires longer planning times than a 2D path. *Right*: Our approach uses a a 2D path and applies footstep planning wherever needed, resulting in efficient planning times.

Contrary to that, the greater flexibility of footstep planning allows the robot to pass the obstacles collision-free (Fig. 7, left). However, planning this sequence of footsteps takes 0.45 s. Using our adaptive approach, the robot efficiently plans within 0.05 s and is able to pass the obstacles. The advantage of our approach is that costly footstep planning is only invoked where needed.

VII. CONCLUSIONS AND OUTLOOK

We presented a novel path planning approach for humanoid robots leading to highly efficient paths. Our method first classifies the environment into regions of different complexity with respect to the traversability. To decide on the traversability and to obtain an initial estimate of the traversal costs, we employ footstep planners between points sampled on the contours of obstacle regions as an optional preprocessing step. Fast but coarse 2D planning is used in the open areas that are sufficiently distant to obstacles. These plans are augmented with detailed footstep plans through the obstacle regions. A heuristic enables efficient planning when the environment is not static and precomputation is not possible.

We evaluated our approach in a large office environment and with a real Nao humanoid. While our resulting plans are almost as efficient as paths resulting from planning the complete trajectory on footstep basis, they are significantly faster to compute. Our framework is general enough to be applied to other robotic systems in different scenarios, e.g., with different modes of locomotion to cross different types of terrain. In future work, we plan to augment our framework with efficient collision checks in 3D where required [23].

REFERENCES

[1] T.-Y. Li, P.-F. Chen, and P.-Z. Huang, "Motion planning for humanoid walking in a layered environment," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2003.

[2] J. Chestnutt and J. Kuffner, "A tiered planning strategy for biped navigation," in *Proc. of the IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, 2004.

[3] J.-S. Gutmann, M. Fukuchi, and M. Fujita, "A modular architecture for humanoid robot navigation," in *Proc. of the IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, 2005.

[4] J. Garimort, A. Hornung, and M. Bennewitz, "Humanoid navigation with dynamic footstep plans," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2011.

[5] J. Chestnutt, M. Lau, K. M. Cheung, J. Kuffner, J. K. Hodgins, and T. Kanade, "Footstep planning for the Honda ASIMO humanoid," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2005.

[6] J. Chestnutt, K. Nishiwaki, J. Kuffner, and S. Kagami, "An adaptive action model for legged navigation planning," in *Proc. of the IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, 2007.

[7] K. Hauser, T. Bretl, J.-C. Latombe, K. Harada, and B. Wilcox, "Motion planning for legged robots on varied terrain," *Int. Journal of Robotics Research (IJRR)*, 2007.

[8] O. Kanoun, J.-P. Laumond, and E. Yoshida, "Planning foot placements for a humanoid robot: A problem of inverse kinematics," *Int. Journal of Robotics Research (IJRR)*, 2010.

[9] R. Cupec, G. Schmidt, and O. Lorch, "Experiments in vision-guided robot walking in a structured scenario," in *Proc. of the IEEE Int. Symp. on Industrial Electronics*, 2005.

[10] M. Yagi and V. Lumelsky, "Local on-line planning in biped robot locomotion amongst unknown obstacles," *Robotica*, vol. 18, no. 4, pp. 389–402, 2000.

[11] K. Okada, T. Ogura, A. Haneda, and M. Inaba, "Autonomous 3D walking system for a humanoid robot based on visual step recognition and 3D foot step planner," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2005.

[12] K. Hauser, T. Bretl, and J.-C. Latombe, "Non-gaited humanoid locomotion planning," in *Proc. of the IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, 2005.

[13] N. Vahrenkamp, C. Scheurer, T. Asfour, J. J. Kuffner, and R. Dillmann, "Adaptive motion planning for humanoid robots," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2008.

[14] S. Zickler and M. Veloso, "Variable level-of-detail motion planning in environments with poorly predictable bodies," in *Proc. of the European Conf. on Artificial Intelligence (ECAI)*, 2010, pp. 189–194.

[15] R. Steffens, M. Nieuwenhuisen, and S. Behnke, "Multiresolution path planning in dynamic environments for the standard platform league," in *Workshop on Humanoid Soccer Robots at the IEEE-RAS Int. Conf. on Humanoid Robots*, 2010.

[16] R. B. Rusu, A. Sundaresan, B. Morisset, K. Hauser, M. Agrawal, J.-C. Latombe, and M. Beetz, "Leaving flatland: Efficient real-time three-dimensional perception and motion planning," *J. Field Robotics*, vol. 26, no. 10, pp. 841–862, 2009.

[17] S. Candido, Y.-T. Kim, and S. Hutchinson, "An improved hierarchical motion planner for humanoid robots," in *Proc. of the IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, 2008.

[18] M. Morales, L. Tapia, R. Pearce, S. Rodriguez, and N. M. Amato, "A machine learning approach for feature-sensitive motion planning," in *Int. Wkshp. on Alg. Found. of Rob. (WAFR)*, 2004.

[19] M. Likhachev, http://www.ros.org/wiki/sbpl, 2010.

[20] M. Likhachev, D. Ferguson, G. Gordon, A. Stentz, and S. Thrun, "Anytime dynamic A*: An anytime, replanning algorithm," in *Proc. of the Int. Conf. on Automated Planning and Scheduling (ICAPS)*, 2005.

[21] A. Hornung, K. M. Wurm, and M. Bennewitz, "Humanoid robot localization in complex indoor environments," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2010.

[22] D. Gouaillier, C. Collette, and C. Kilner, "Omni-directional closed-loop walk for NAO," in *Proc. of the IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, 2010.

[23] A. Hornung, M. Phillips, E. G. Jones, M. Bennewitz, M. Likhachev, and S. Chitta, "Navigation in three-dimensional cluttered environments for mobile manipulation," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2012.