The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Thinning and Smoothing of Randomly-sampled Support Transitions Toward Practical Motion Planning for Humanoid Robots

Toshiya Nishi and Tomomichi Sugihara

Abstract— **A motion planning method for humanoid robots in life environments is proposed based on a two-stage approach, where a sequence of double support postures to represent discrete milestones of the biped locomotion is planned first, and then, a time series of the whole body configuration to interpolate them. This paper discusses the first stage. RRT (Rapidly-explored Random Tree) is utilized for the planning. The main problem is how to modify the sequence of randomly sampled double support postures into practically acceptable one. Some post-processing techniques including thinning and smoothing are presented. A necessary condition of the series of milestones is that a pair of adjacent postures has to share one fixed supporting foot as the pivot. In order to thin out unnecessary milestones, bypass nodes are inserted and Dijkstra's method is applied. A computer simulation in which a humanoid robot travels in an environment with some pieces of furniture is demonstrated.**

I. Introduction

Motion synthesis of humanoid robots is a mathematically challenging problem. The first difficulty one may encounter is the high dimensionality of the configuration space for tens of motorized joints and 6-DOF mobile base. Since the base link is not mechanically connected to the ground, the configuration includes unactuated components so that a dynamical constraint is posed due to the unilaterality and friction limit of reaction forces. By far the hardest issue is that the dynamical constraint changes its form along with the contact states between the robot and the environment. In other words, to find the trajectory involves to find the transition of the dynamical constraint. In this sense, the geometric and dynamical constraints are complicatedly coupled, and thus, it is almost impossible to reach the goal by simply concatenating locally optimum trajectories.

In order to reduce the dimensionality of the search space effectively, some combinatorial planning methods, which use predefined typical humanoid motion primitives, have been proposed[1], [2], [3]. While they work in well-ordered environments on many simulations and real experiments, there still remains difficulties of application in cluttered fields such as our life environments, which contains life items, steps, slopes, walls, etc.

Another approach to this problem is to separate it into two-stages, where a sequence of double support postures to represent discrete milestones of the biped locomotion is planned first, and then, they are interpolated to be continuous with respect to both time and space. It simplifies the problem to some extent since one can know the change of the distribution of contact points, and accordingly, the dynamical constraint in advance. Several methods[4], [5], [6], [7] have been proposed based on Random-sampling techniques[8], [9] to exploit the richness of the whole body postures to travel on irregular terrains. They are known to be effective for the path planning in a large dimension. An inevitable issue of such an approach is that the produced path is jaggy and far from the optimal one. so that post-processing of the path to delete inessential milestones and smoothen it is required for practical applications. The difficulty in case of humanoid robots is that the robot has to move in the environment by discontinuously switching the contacting body parts – usually, feet – rather than by tracing a completely collision-free path. Thus, the post-processing strategy is not trivial.

This paper proposes a motion planning method for humanoid robots which is also based on the above two-stage approach and utilizes RRT-connect[9]. We focus on the former issue around the planning of a sequence of double support postures as milestones, while the latter will be discussed in another paper in the future. A necessary condition of the path to be planned in the first problem is that any pair of adjacent milestones has to share one supporting foot. An idea to thin out unnecessary milestones under the above condition is to insert bypass nodes to the randomly-sampled path and to apply Dijkstra's method[10] for the optimization. It is demonstrated in computer simulations that a humanoid robot travels in a life environment with some pieces of furniture.

II. Mathematical overview of problem

Let us consider a humanoid robot with N_j joints, which are fully-actuated, and represent the whole body configuration of it by $\boldsymbol{q} \in \mathcal{Q}$, where $\boldsymbol{q}$ includes six parameters for position and orientation of the base link, and $\mathcal{Q} \subset \mathbb{R}^{N_j+6}$ is the configuration space bounded by the limit of joint displacements. The angle-axis vector, for example, is available to represent the orientation of the base link without concern for the problem of singularity. Suppose the environment consists of undeformable objects and its profile is given by polygon soup. The aim of motion planning is to find a time series $\boldsymbol{q}(t)$ from the initial configuration $\boldsymbol{q}_0$ to the goal configuration $\boldsymbol{q}_G$ under the following conditions:

1) At least one facet on the body is in contact with the environment at any moment,

This work was supported in part by Grant-in-Aid for Young Scientists (B) #20760170, Japan Society for the Promotion of Science and by "The Kyushu University Research Superstar Program (SSP)", based on the budget of Kyushu University allocated under President's initiative.

T. Nishi graduated Kyushu University and now works at Denso, Japan.
T. Sugihara (zhidao@ieee.org) is with Department of Adaptive Machine Systems, Graduate School of Engineering, Osaka University, 2-1 Yamadaoka, Suita, Osaka, Japan.

Algorithm 1 MOTIONPLAN($\boldsymbol{q}_0$, $\boldsymbol{q}_G$)

$\{\boldsymbol{q}_1, \cdots, \boldsymbol{q}_S\}$ ←MILESTONEPLAN($\boldsymbol{q}_0$, $\boldsymbol{q}_G$)
for $i = 0$ to S **do**
  $(\boldsymbol{q}_{i,i+1}(t), T_{i+1})$←SEGMENTPLAN($\boldsymbol{q}_i$, $\boldsymbol{q}_{i+1}$)
end for
return $\boldsymbol{q}(t) = \sum_{i=0}^{S} \boldsymbol{q}_{i,i+1}(t)$

Algorithm 2 MILESTONEPLAN($\boldsymbol{q}_0$, $\boldsymbol{q}_G$)

P ←MILESTONERRT($\boldsymbol{q}_0$, $\boldsymbol{q}_G$)
P ←MILESTONESHORTCUTRANDOM(P)
P ←MILESTONESHORTCUTDIJKSTRA(P)
return P ←MILESTONEAVE(P)
Ensure: $P = \{\boldsymbol{q}_1, \cdots, \boldsymbol{q}_S\}$

2) Any body part doesn't penetrate into the environment, and
3) Any body part doesn't penetrate into another body part.

The condition 1) is posed for simplicity to avoid irregular situations where the robot hops or stands on toe. Let us define $\mathcal{C} \subset \mathcal{Q}$ as the set of $\boldsymbol{q}$ which satisfies the above three conditions. Namely, the following condition must be satisfied:

$$\boldsymbol{q} \in \mathcal{C}. \tag{1}$$

Since $\boldsymbol{q}$ includes unactuated components, it doesn't only involve the geometric but also dynamical constraint. An important fact is that the dynamical constraint essentially depends on the contact state since it is due to the limitation of external forces exerted at each contact point.

The problem can be simplified to some degree if it is separated into the planning of a sequence of double support postures to represent discrete milestones of the biped locomotion and a time series of the whole body configuration to interpolate them, since one can know the change of the distribution of contact points in advance. This idea is implemented as the pseudocode MOTIONPLAN in **Algorithm 1**. In the first stage, a discrete series of double support postures $P = \{\boldsymbol{q}_1, \boldsymbol{q}_2, \cdots, \boldsymbol{q}_S\}$ including the number of its elements S is planned as milestones by MILESTONEPLAN. Then, the time-series $\boldsymbol{q}(t)$ which interpolates the milestones is planned by SEGMENTPLAN in the second stage. This paper focuses on the former problem.

The sequence of milestones P is inserted between $\boldsymbol{q}_0$ and $\boldsymbol{q}_G$, where $\boldsymbol{q}_{S+1} \equiv \boldsymbol{q}_G$. Hereafter, we only consider the biped locomotion, in which the left and right feet alternately lands onto the ground, for simplicity. Each milestone $\boldsymbol{q}_i$ ($i = 0 \sim S$) represents the robot posture immediately before a step. Consequently, a combination of $\boldsymbol{q}_i$ and $\boldsymbol{q}_{i+1}$ segments the motion from the lifting-up of one foot to immediately before the next lifting-up of the other foot. During the transition from $\boldsymbol{q}_i$ to $\boldsymbol{q}_{i+1}$, at least one foot has to be thoroughly fixed with respect to the ground. Let us call such a fixed foot *the pivot foot* and the other *the stepping foot*. Suppose $\boldsymbol{r}_{Pi}$ and $\boldsymbol{r}_{Si}$ represent combinations of the position and orientation of the pivot foot and the stepping foot, respectively. The

Algorithm 3 MILESTONERRT($\boldsymbol{q}_0$, $\boldsymbol{q}_G$)

$N = \{\boldsymbol{q}_0, \boldsymbol{q}_G\}$, $E = \emptyset$
for $k = 1$ to $k_{\max}$ **do**
  $\boldsymbol{q}_{\mathrm{rand}}$ ←RAND($\mathcal{Q}$)
  $\boldsymbol{q}_{\mathrm{near}} \leftarrow \arg\min_{\boldsymbol{q}_j} \{d(\boldsymbol{q}_{\mathrm{rand}}, \boldsymbol{q}_j),\ \boldsymbol{q}_j \in N\}$
  $\boldsymbol{q}'_{\mathrm{new}}$ ←INTERDIV($\boldsymbol{q}_{\mathrm{near}}$, $\boldsymbol{q}_{\mathrm{rand}}$, $\varepsilon_{\mathrm{ms}}$)
  $\boldsymbol{r}_{S\mathrm{new}} \leftarrow \boldsymbol{r}_{P\mathrm{near}}$
  $\boldsymbol{r}_{P\mathrm{new}}$ ←SETTLE($\boldsymbol{r}'_{P\mathrm{new}}$)
  $\boldsymbol{q}_{\mathrm{new}}$ ←IK($\{\boldsymbol{r}_{P\mathrm{new}}, \boldsymbol{r}_{S\mathrm{new}}\}$, $\boldsymbol{q}'_{\mathrm{new}}$)
  if $\boldsymbol{q}_{\mathrm{new}} \in \mathcal{C}$ **then**
    $N \leftarrow N \cup \{\boldsymbol{q}_{\mathrm{new}}\}$, $E \leftarrow E \cup \{(\boldsymbol{q}_{\mathrm{near}}, \boldsymbol{q}_{\mathrm{new}})\}$
    if $\boldsymbol{r}_{P\mathrm{new}} = \boldsymbol{r}_{PG}$ **then**
      return P ←SHORTESTPATH(E)
    end if
  end if
end for
return nil

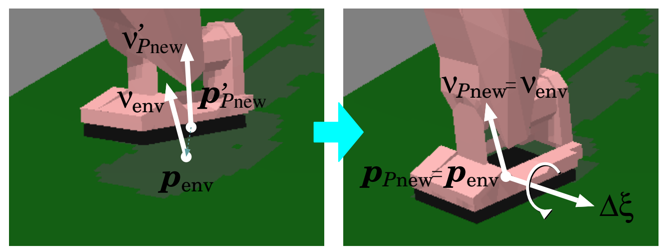


Fig. 1. SETTLE manipulation: the position and orientation of the stepping foot is modified to settle on the closest facet of the environment.

above assumption of biped locomotion poses the following condition:

$$\boldsymbol{r}_{Pi} = \boldsymbol{r}_{Si+1}. \tag{2}$$

This alternation sequence of the pivot foot defines the global locomotion. MILESTONEPLAN($\boldsymbol{q}_0$, $\boldsymbol{q}_G$) aims to find a geometrically and kinematically feasible pattern of locomotion and to significantly reduce the search space.

The pseudocode of MILESTONEPLAN($\boldsymbol{q}_0$, $\boldsymbol{q}_G$) is shown in **Algorithm 2**. First, MILESTONERRT($\boldsymbol{q}_0$, $\boldsymbol{q}_G$) plans a 'raw' path by utilizing RRT-connect[9]. Then, it is thinned out roughly by MILESTONESHORTCUTRANDOM(P) and finely by MILESTONESHORTCUTDIJKSTRA(P). Finally, MILESTONEAVE(P) smoothen the path by the avaraging technique. Each procedure will be detailed in the following sections.

III. PLANNING OF DISCRETE MILESTONES BY RRT-CONNECT AND CONTACT ENFORCEMENT

A necessary condition of P to be planned here is that a pair of adjacent milestones has to share one grounded foot as expressed by Eq.(2). This is guaranteed by a version of RRT-connect with *contact enforcement*[5], where the original RRT is used instead of RRT-connect in the following description for ease of explanation. MILESTONERRT($\boldsymbol{q}_0$, $\boldsymbol{q}_G$) in **Algorithm 3** formalizes the entire procedure.

N is the set of nodes initialized by $\{\boldsymbol{q}_0, \boldsymbol{q}_G\}$, while E is the set of directed arcs between the nodes in N initialized as an empty set. The algorithm iteratively expands a tree from $\boldsymbol{q}_0$ and adds valid arcs to the tree as branches until it reaches the goal or the number of iteration exceeds its maximum value $k_{\max}$ (failure case). The output P will be composed as the shortest path in E from $\boldsymbol{q}_0$ to $\boldsymbol{q}_G$ in the successful case (or nil in the failure case).

At each step of iteration, $\boldsymbol{q}_{\text{rand}} \in \mathcal{Q}$ is randomly sampled by RAND($\mathcal{Q}$), and the nearest-neighbor node $\boldsymbol{q}_{\text{near}}$ in the current tree is found in accordance with the metric defined for two configurations $\boldsymbol{q}_i$ and $\boldsymbol{q}_j$ as

$$d(\boldsymbol{q}_i, \boldsymbol{q}_j) \equiv \frac{1}{2} \sum_{k=1}^{N_l} \left\| \boldsymbol{e}(\boldsymbol{r}_k(\boldsymbol{q}_i), \boldsymbol{r}_k(\boldsymbol{q}_j)) \right\|^2, \qquad (3)$$

where N_l is the number of robot links and $\boldsymbol{r}_k(\boldsymbol{q})$ is the position and orientation of kth link of the robot for a configuration $\boldsymbol{q}$. $\boldsymbol{e}(\boldsymbol{r}_i, \boldsymbol{r}_j) \in \mathbb{R}^6$ means the residual vector between $\boldsymbol{r}_i$ and $\boldsymbol{r}_j$, which is defined as a combination of the difference vector in Cartesian space for the position and the equivalent angle-axis vector of diffrence matrix in $SO(3)$ for the orientation. Refer the paper[11] for detail.

Then, a candidate of the new node $\boldsymbol{q}'_{\text{new}}$ is generated by moving from $\boldsymbol{q}_{\text{near}}$ toward $\boldsymbol{q}_{\text{rand}}$ at a small distance ε_{ms} by INTERDIV($\boldsymbol{q}_{\text{near}}$, $\boldsymbol{q}_{\text{rand}}$, ε_{ms}) in **Algorithm 3**. The internal divisions in Euclidean sense are available for position of the base link and joint displacements, while SLERP[12] is available for orientation of the base link.

For contact enforcement, the set of $\boldsymbol{r}_{P\text{new}}$ and $\boldsymbol{r}_{S\text{new}}$ should be prepared. From Eq.(2), $\boldsymbol{r}_{S\text{new}} = \boldsymbol{r}_{P\text{near}}$. On the other hand, $\boldsymbol{r}_{P\text{new}}$ is computed as follows. If either foot of $\boldsymbol{q}_{\text{G}}$ is within the one-step range from $\boldsymbol{r}'_{P\text{new}}$, we define it as $\boldsymbol{r}_{PG}$ and $\boldsymbol{r}_{P\text{new}} = \boldsymbol{r}_{PG}$. Otherwise, $\boldsymbol{r}_{P\text{new}}$ is computed in such a way that the position and the normal vector of the new pivot sole coincide with the closest facet and its normal vector, respectively, in the environment with the minimum rotation. The detail is described in Appendix. This procedure is defined by SETTLE($\boldsymbol{r}'_{P\text{new}}$) in **Algorithm 3** and illustrated by Fig.1. By solving the inverse kinematics for redundant robots[11], $\boldsymbol{q}'_{\text{new}}$ is modified to $\boldsymbol{q}_{\text{new}}$ which satisfies both $\boldsymbol{r}_{P\text{new}}$ and $\boldsymbol{r}_{S\text{new}}$. If the condition (1) is satisfied, the node $\boldsymbol{q}_{\text{new}}$ and the arc $(\boldsymbol{q}_{\text{near}}, \boldsymbol{q}_{\text{new}})$ are added to N and E, respectively. Otherwise, it is discarded. The condition (1) is checked roughly by using OBB (oriented bounding box) and finely by GJK algorithm[13]. If $\boldsymbol{r}_{P\text{new}} = \boldsymbol{r}_{PG}$, $\boldsymbol{q}_{\text{new}}$ is regarded as $\boldsymbol{q}_S$ and the algorithm terminates.

IV. THINNING TECHNIQUE OF MILESTONES

Basically, the raw path acquired in the previous section is detouring and practically unapplicable. However, it is not permitted to simply shortcut milestones due to the constraint (2). Here, the idea of bypass insertion is presented.

Suppose $\boldsymbol{r}_{Li}$ and $\boldsymbol{r}_{Ri}$ are combinations of the position and orientation of the left and right feet, respectively, of a milestone $\boldsymbol{q}_i$. For two milestones $\boldsymbol{q}_i$ and $\boldsymbol{q}_j$ $(i < j)$, $\boldsymbol{q}_{\text{mid}}$ is generated as the middle configuration of them.

Algorithm 4 BYPASS($\boldsymbol{q}_i$, $\boldsymbol{q}_j$)

$\boldsymbol{q}_{\text{mid}} \leftarrow$INTERDIV($\boldsymbol{q}_i$, $\boldsymbol{q}_j$, 1/2)
if $\|\boldsymbol{e}(\boldsymbol{r}_{Li}, \boldsymbol{r}_{Rj})\| > d_{\text{th}}$ **then**
  $\boldsymbol{q}_{B1} \leftarrow$nil
else
  $\boldsymbol{q}_{B1} \leftarrow$IK($\{\boldsymbol{r}_{Li}, \boldsymbol{r}_{Rj}\}$, $\boldsymbol{q}_{\text{mid}}$)
  if $\boldsymbol{q}_{B1} \notin \mathcal{C}$ **then**
    $\boldsymbol{q}_{B1} \leftarrow$nil
  end if
end if
if $\|\boldsymbol{e}(\boldsymbol{r}_{Lj}, \boldsymbol{r}_{Ri})\| > d_{\text{th}}$ **then**
  $\boldsymbol{q}_{B2} \leftarrow$nil
else
  $\boldsymbol{q}_{B2} \leftarrow$IK($\{\boldsymbol{r}_{Lj}, \boldsymbol{r}_{Ri}\}$, $\boldsymbol{q}_{\text{mid}}$)
  if $\boldsymbol{q}_{B2} \notin \mathcal{C}$ **then**
    $\boldsymbol{q}_{B2} \leftarrow$nil
  end if
end if
return $\{\boldsymbol{q}_{B1}, \boldsymbol{q}_{B2}\}$

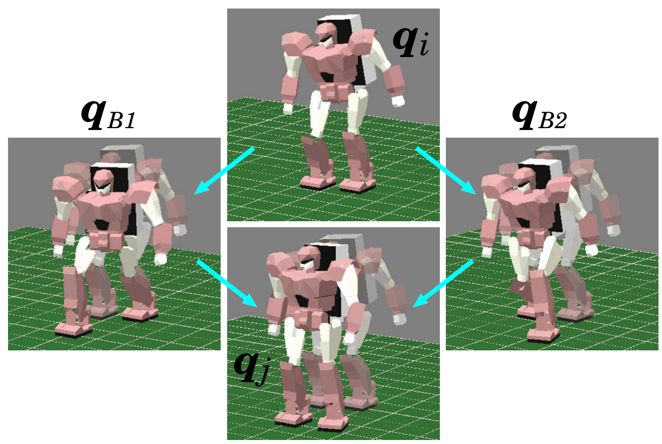


Fig. 2. BYPASS manipulation: two postures are inserted as bypass nodes through which the robot moves from a node to another only in two steps.

Then, it tries to modify $\boldsymbol{q}_{\text{mid}}$ to $\boldsymbol{q}_{B1}$ which achieves $\boldsymbol{r}_{Li}$ and $\boldsymbol{r}_{Rj}$ simultaneously through the inverse kinematics. If $\|\boldsymbol{e}(\boldsymbol{r}_{Li}, \boldsymbol{r}_{Ri})\| > d_{\text{th}}$ for a threshold d_{th} or $\boldsymbol{q}_{B1} \notin \mathcal{C}$, $\boldsymbol{q}_{B1}$ is set for nil. Otherwise, it can be a bypass node through which the robot moves from $\boldsymbol{q}_i$ to $\boldsymbol{q}_j$ in two steps. Likewise, $\boldsymbol{q}_{B2}$ can also be a bypass node if it achieves both $\boldsymbol{r}_{Lj}$ and $\boldsymbol{r}_{Ri}$ and $\boldsymbol{q}_{B2} \in \mathcal{C}$. This process is defined by BYPASS in **Algorithm 4** and illustrated in Fig.2.

The proposed thinning technique utilizes the above bypass insertion in two phases as follows. In the first 'rough-cutting' phase, two nodes $\boldsymbol{q}_i$ and $\boldsymbol{q}_j$ in P are picked up at random and tested if a bypass node is available for them. Namely, if one of $\boldsymbol{q}_{B1}$ and $\boldsymbol{q}_{B2}$ is not nil, the partial path from $\boldsymbol{q}_i$ to $\boldsymbol{q}_j$ is deleted and the bypass node is inserted between them. It is repeated until either it exceeds the maximum number of trials n_{sr} or the size of P becomes less than the boundary s. The procedure is explained by MILESTONESHORTCUTRANDOM(P) in **Algorithm 5**. After

Algorithm 5 MILESTONESHORTCUTRANDOM(P)

Require: $P = \{\boldsymbol{q}_1, \cdots, \boldsymbol{q}_S\}$
for $k = 1$ to $n_{\rm sr}$ **do**
  $\boldsymbol{q}_i \leftarrow$RAND($P$), $\boldsymbol{q}_j \leftarrow$RAND($P$) $(i < j)$
  if $j - i > 2$ **then**
    if $\{\boldsymbol{q}_{B1}, \boldsymbol{q}_{B2}\} \leftarrow$BYPASS($\boldsymbol{q}_i$, $\boldsymbol{q}_j$) **then**
      $P \leftarrow P - \{\boldsymbol{q}_i, \cdots, \boldsymbol{q}_j\}$
      if $\boldsymbol{q}_{B1} \neq$nil **then**
        $P \leftarrow P \cup \{\boldsymbol{q}_i, \boldsymbol{q}_{B1}, \boldsymbol{q}_j\}$
      else
        $P \leftarrow P \cup \{\boldsymbol{q}_i, \boldsymbol{q}_{B2}, \boldsymbol{q}_j\}$
      end if
      if SIZE(P)$< s$ **then**
        break
      end if
    end if
  end if
end for

Algorithm 6 MILESTONESHORTCUTDIJKSTRA(P)

Require: $P = \{\boldsymbol{q}_1, \cdots, \boldsymbol{q}_S\}$
$(N, E) \leftarrow$PATHTOGRAPH(P)
for $i = 0$ to G **do**
  for $j = G$ downto $i + 3$ **do**
    if $\{\boldsymbol{q}_{B1}, \boldsymbol{q}_{B2}\} \leftarrow$BYPASS($\boldsymbol{q}_i$, $\boldsymbol{q}_j$) **then**
      if $\boldsymbol{q}_{B1} \neq$nil **then**
        $N \leftarrow N \cup \{\boldsymbol{q}_{B1}\}$
        $E \leftarrow E \cup \{(\boldsymbol{q}_i, \boldsymbol{q}_{B1}), (\boldsymbol{q}_{B1}, \boldsymbol{q}_j)\}$
      end if
      if $\boldsymbol{q}_{B2} \neq$nil **then**
        $N \leftarrow N \cup \{\boldsymbol{q}_{B2}\}$
        $E \leftarrow E \cup \{(\boldsymbol{q}_i, \boldsymbol{q}_{B2}), (\boldsymbol{q}_{B2}, \boldsymbol{q}_j)\}$
      end if
      break
    end if
  end for
end for
return $P \leftarrow$DIJKSTRA(N, E, $\boldsymbol{q}_0$, $\boldsymbol{q}_G$)

the size of P is reduced, P is converted to a directed graph represented by (N, E), where N is a set of nodes and E is a set of directed arcs. The bypass insertion tests are conducted for all combinations of non-adjacent nodes in N, and the new bypassing arcs are added to E. Then, Dijkstra's method[10] is applied in order to update P with the optimum path for (N, E). MILESTONESHORTCUTDIJKSTRA(P) in **Algorithm 6** shows this procedure. An important fact is that it is guaranteed that the optimum solution path exists, since the original P can be a solution path even in the worst case.

V. SMOOTHING TECHNIQUE OF DISCRETE MILESTONES

Jaggedness of P acquired in the previous section can be reduced by an averaging technique described in **Algorithm 7**. Let us consider $\boldsymbol{q}_i$ for $i = 1 \sim S - 1$. Apart from the original position and orientation of its pivot foot $\boldsymbol{r}_{Pi}$,

Algorithm 7 MILESTONEAVE(P)

for $k = 1$ to $n_{\rm ave}$ **do**
  for $i = 1$ to $S - 1$ **do**
    $\boldsymbol{r}'_{Pi} \leftarrow$LINKINTERDIV($\boldsymbol{r}_{Pi-2}$, $\boldsymbol{r}_{Pi+2}$, $1/2$)
    $\boldsymbol{r}'_{Pi} \leftarrow$SETTLE($\boldsymbol{r}'_{Pi}$)
    $\boldsymbol{r}'_{Si+1} \leftarrow \boldsymbol{r}'_{Pi}$
    $\boldsymbol{q}'_i \leftarrow$IK($\{\boldsymbol{r}'_{Pi}, \boldsymbol{r}_{Si}\}$, $\boldsymbol{q}_i$)
    $\boldsymbol{q}'_{i+1} \leftarrow$IK($\{\boldsymbol{r}_{Pi+1}, \boldsymbol{r}'_{Si+1}\}$, $\boldsymbol{q}_{i+1}$)
    if $\boldsymbol{q}'_i \in \mathcal{C}$ and $\boldsymbol{q}'_{i+1} \in \mathcal{C}$ **then**
      $\boldsymbol{q}_i \leftarrow \boldsymbol{q}'_i$, $\boldsymbol{q}_{i+1} \leftarrow \boldsymbol{q}'_{i+1}$
    end if
  end for
end for

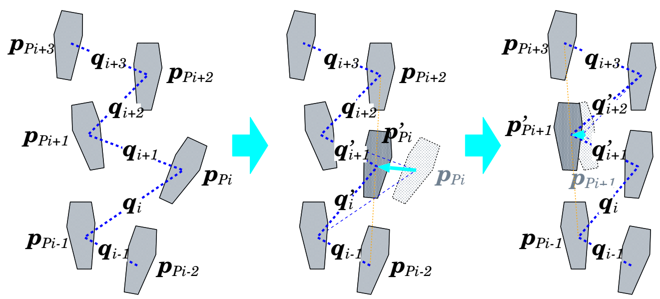


Fig. 3. MILESTONEAVE manipulation: the position and orientation of each pivot foot is averaged by its before-after locations.

$\boldsymbol{r}'_{Pi}$ is first computed as the middle division of $\boldsymbol{r}_{Pi-2}$ and $\boldsymbol{r}_{Pi+2}$ by LINKINTERDIV, and then modified so as to contact with the closest facet in the environment by SETTLE, where $\boldsymbol{r}_{P-1} \equiv \boldsymbol{r}_{S0}$. $\boldsymbol{r}'_{Pi}$ also serves as $\boldsymbol{r}'_{Si+1}$. Next, $\boldsymbol{q}_i$ and $\boldsymbol{q}_{i+1}$ are modified to $\boldsymbol{q}'_i$ and $\boldsymbol{q}'_{i+1}$, respectively, by the inverse kinematics for contact enforcement to the combination of $\{\boldsymbol{r}'_{Pi}, \boldsymbol{r}_{Si}\}$ and $\{\boldsymbol{r}_{Pi+1}, \boldsymbol{r}'_{Si+1}\}$, respectively. If and only if both $\boldsymbol{q}_i \in \mathcal{C}$ and $\boldsymbol{q}_{i+1} \in \mathcal{C}$ are satisfied, $\boldsymbol{q}'_i$ and $\boldsymbol{q}'_{i+1}$ are accepted and replaced with $\boldsymbol{q}_i$ and $\boldsymbol{q}_{i+1}$, respectively. As the above process is repeated up to $n_{\rm ave}$ times, the change of double support postures in the sequence is expected to be moderate as depicted by Fig.3.

VI. EXAMPLES OF BIPED LOCOMOTION PLANNING

Several case studies were conducted in simulations, supposing a humanoid robot[14] in which $N_j = 20$. The computer featured CPU Pentium D 3GHz and RAM 1GB. Fig.4 shows the tested environment which models a life room with a gate, a slope, steps and some pieces of furniture. The parameters were $\varepsilon_{\rm ms} = 0.2$, $d_{\rm th} = 0.2$, $n_{\rm sr} = 500$ and $n_{\rm ave} = 50$.

In the first scenario, the robot climbed up the slope and passed by the TV set and one of the chairs. The number of milestones created by MILESTONERRT was 62. In this case, it was under s, and thus MILESTONESHORTCUTRANDOM was not applied. MILESTONESHORTCUTDIJKSTRA reduces milestones to 20 in 6.7[s]. Fig.5 shows the raw and processed path in which only the loci of the center of mass (COM) are displayed. Then, the sequence of milestones was smoothened

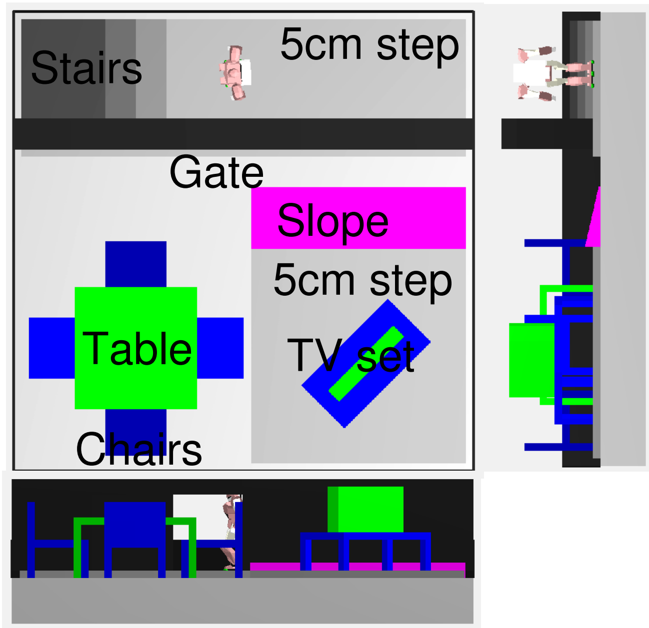
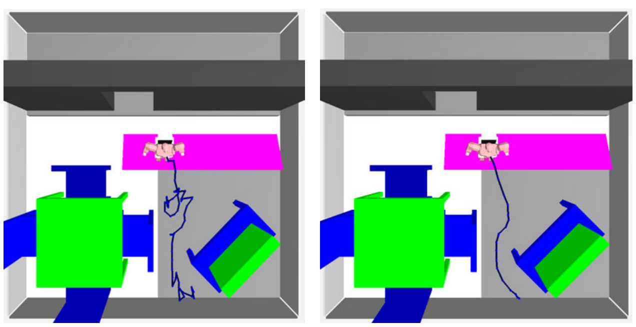


Fig. 4. Task environment modeling a life environment with a gate, a slope, steps and some pieces of furniture.



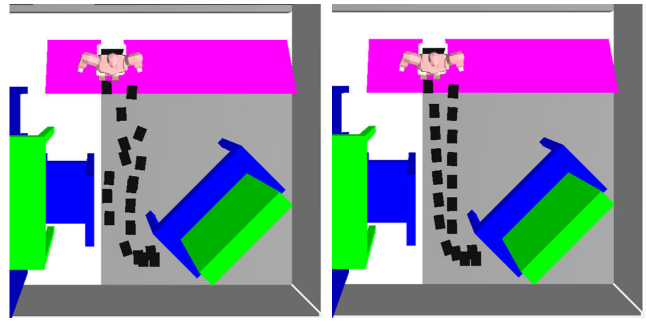
(a) output of MILESTONERRT (b) output of MILESTONESHORTCUTDIJKSTRA

Fig. 5. A sequence of discrete postures for a walking motion (only the center of mass (COM) path is displayed). (a) The number of milestones output by MILESTONERRT was 62. (b) MILESTONESHORTCUTDIJKSTRA reduces milestones to 20 in 6.7[s]. In this case, MILESTONESHORTCUTRANDOM was not applied since number of milestones of the raw path was under s.

by MILESTONEAVE, the output of which is shown in Fig.6. One can see that the footstamps were processed to be ordered in the result.

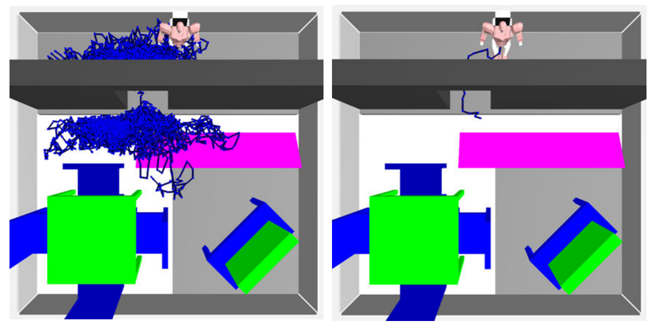
Next, the robot went through the gate in the second scenario. The number of milestones created by MILESTONERRT was 6790, and MILESTONESHORTCUTRANDOM reduced them to 12 in 8.9[s]. MILESTONESHORTCUTDIJKSTRA was not needed to be applied. For comparison, the original 6790 milestones were directly input to MILESTONESHORTCUTDIJKSTRA, but failed to find the optimum path even over 1 hour computation. Fig.7 shows the raw and processed path in which only the loci of COM are displayed.

The last scenario is that the robot travels in the room



(a) output of MILESTONESHORTCUT (b) output of MILESTONEAVE

Fig. 6. A sequence of footstamps before and after MILESTONEAVE. The smoothing was repeated 50times and consumed 2.5[s] in total.



(a) output of MILESTONERRT (b) output of MILESTONESHORTCUTRANDOM

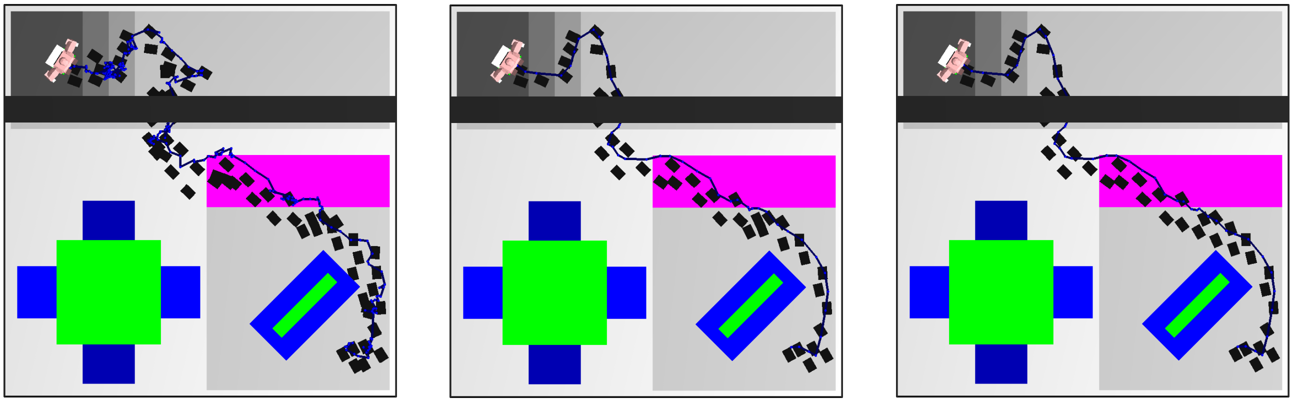
Fig. 7. A sequence of discrete postures for a walking-through-gate motion (only COM path is displayed). (a) The number of milestones output by MILESTONERRT was 6790. (b) MILESTONESHORTCUTRANDOM reduced milestones to 12 in 8.9[s].

going down stairs, passing through a gate, climbing up a slope and getting around furniture. Fig.8 shows footstamps and the locus of COM of the planned trajectory. Fig.8(a) is by only MILESTONERRT. Fig.8(b) is the result after MILESTONESHORTCUTRANDOM and MILESTONESHORTCUTDIJKSTRA. Fig.8(c) shows the output of MILESTONEAVE. The total computation time for the planning of discrete milestones was about 3132[s], which is broken down to 2967[s] for RRT-connect, 155[s] for the shortcut and 10[s] for the averaging. The total number of milestones is reduced from 376 to 47 by the shortcut. 44[s] for the planning and smoothing of continuous trajectories.

The number of nodes and the time for shortcut are listed in Table I to evaluate the performance of the proposed method.

VII. CONCLUSION

An effective planning of the transition of double support postures for humanoid robots was proposed. It utilizes RRT-connect for its high applicability to various, even irregular but frequently situated environments in our life scenes. It overcomes the defect of random-sampling-based approach that the resulted path becomes jaggy and detouring through some thinning and smoothing techniques. A particular constraint posed on the biped locomotion where the left and



(a) milestones planned by RRT-connect (b) milestones after shortcut (c) milestones after smoothing

Fig. 8. Footstamps and the locus of COM of the planned trajectory.

TABLE I

RESULT: THE NUMBER OF NODES AND TIME FOR SHORTCUT

scenario	number of milestones		time for shortcut
	original	shortcut	
Slope	62	20	6.7[s]
Gate	6790	12	8.9[s]
Stair+Gate+Slope	376	47	155[s]

right feet alternately lands onto the ground is handled by the bypass node insertion. Dijkstra's method optimizes the path where the existence of a solution path is guaranteed. It works when combined with the interpolation method for geometrically and dynamically consistent continuous motion, which is described in another paper in the future.

APPENDIX

SETTLE($\boldsymbol{r}'_{P\mathrm{new}}$) is an important procedure for contact enforcement. As noted in section III, it returns the position and orientation of one of the foot of $\boldsymbol{q}_{\mathrm{G}}$ if it is within the one-step range from $\boldsymbol{r}'_{P\mathrm{new}}$. This section explains the other cases which happen more generally.

As the left side of Fig.1 shows, suppose $\boldsymbol{p}'_{P\mathrm{new}}$ and $\boldsymbol{\nu}'_{P\mathrm{new}}$ are the position vector and the unit normal vector of the new pivot sole, respectively. Also, suppose $\boldsymbol{p}_{\mathrm{env}}$ and $\boldsymbol{\nu}_{\mathrm{env}}$ are the position vector and the unit normal vector, respectively, of the closest facet to the new pivot foot in the environment. The position of $\boldsymbol{r}_{P\mathrm{new}}$ is simply defined by $\boldsymbol{p}_{\mathrm{env}}$. Concerning with the orientation, the angle-axis vector $\Delta\boldsymbol{\xi}$ which converts $\boldsymbol{\nu}'_{P\mathrm{new}}$ to $\boldsymbol{\nu}_{\mathrm{env}}$ with the minimum rotation shown in the right side of Fig.1 is defined as

$$\Delta\boldsymbol{\xi} \equiv \theta\boldsymbol{n} \tag{4}$$

$$\theta \equiv \mathrm{atan2}(\|\boldsymbol{\nu}'_{P\mathrm{new}} \times \boldsymbol{\nu}_{\mathrm{env}}\|, {\boldsymbol{\nu}'_{P\mathrm{new}}}^{\mathrm{T}}\boldsymbol{\nu}_{\mathrm{env}}), \tag{5}$$

$$\boldsymbol{n} \equiv \frac{\boldsymbol{\nu}'_{P\mathrm{new}} \times \boldsymbol{\nu}_{\mathrm{env}}}{\|\boldsymbol{\nu}'_{P\mathrm{new}} \times \boldsymbol{\nu}_{\mathrm{env}}\|}. \tag{6}$$

The equivalent rotation matrix $\Delta\boldsymbol{R}$ with the above angle-axis vector is computed as

$$\Delta\boldsymbol{R} = \boldsymbol{1} + \sin\theta[\boldsymbol{n}\times] + (1-\cos\theta)[\boldsymbol{n}\times]^2, \tag{7}$$

where $[\boldsymbol{n}\times]$ is the outer-product matrix for $\boldsymbol{n}$. The orientation of $\boldsymbol{r}_{P\mathrm{new}}$ is obtained by multiplying $\Delta\boldsymbol{R}$ from the left side to the current orientation matrix of the new pivot foot.

REFERENCES

[1] Z. Shiller, K. Yamane, and Y. Nakamura, "Planning Motion Patterns of Human Figures Using a Multi-layered Grid and the Dynamics Filter," in *Proceedings of the 2001 IEEE International Conference on Robotics & Automation*, 2001, pp. 1–8.

[2] J. Kuffner, K. Nishiwaki, S. Kagami, M. Inaba, and H. Inoue, "Footstep Planning Among Obstacles for Biped Robots," in *Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2001, pp. 500–505.

[3] E. Yoshida, I. Belousov, C. Esteves, and J.-P. Laumond, "Humanoid Motion Planning for Dynamic Tasks," in *Proceedings of the 2005 IEEE-RAS International Conference on Humanoid Robots*, 2005, pp. 1–6.

[4] E. Yoshida, "Humanoid Motion Planning using Multi-Level DOF Exploitation based on Randomized Method," in *Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005, pp. 25–30.

[5] K. Hauser, T. Bretl, and J.-C. Latombe, "Non-gaited Humanoid Locomotion Planning," in *Proceedings of the 2005 IEEE-RAS International Conference on Humanoid Robots*, 2005, pp. 7–12.

[6] K. Harada, S. Hattori, H. Hirukawa, M. Morisawa, S. Kajita, and E. Yoshida, "Motion Planning for Walking Pattern Generation of Humanoid Robots," in *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007, pp. 4227–4233.

[7] K. Bouyarmane, A. Escande, F. Lamiraux, and A. Kheddar, "Potential Field Guide for Humanoid Multicontacts Acyclic Motion Planning," in *Proceedings of the 2009 IEEE International Conference on Robotics & Automation*, 2009, pp. 1165–1170.

[8] L. E. Kavraki, P. Svestka, J. C. Latombe, and M. Overmars, "Probabilistic Roadmaps for Path Planning in High Dimensional Configuration Spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, pp. 566–580, 1996.

[9] J. J. Kuffner and S. M. LaValle, "RRT-Connect: An efficient approach to single-query path planning," in *Proceedings of IEEE International Conference on Robotics & Automation*, 2000, pp. 995–1001.

[10] E. W. Dijkstra, "A Note on Two Problems in Connexion with Graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.

[11] T. Sugihara, "Solvability-unconcerned Inverse Kinematics based on Levenberg-Marquardt Method," in *Proceedings of the 2009 IEEE-RAS International Conference on Humanoid Robots*, 2009, pp. 555–560.

[12] K. Shoemake, "Animating rotation with quaternion curves," in *Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques*, 1985, pp. 245–254.

[13] E. G. Gilbert, D. W. Johnson, and S. S. Keerthi, "A Fast Procedure for Computing the Distance Between Complex Objects in Three-Dimensional Space," *IEEE Journal of Robotics and Automation*, vol. 4, no. 2, pp. 193–203, 1988.

[14] T. Sugihara, K. Yamamoto, and Y. Nakamura, "Hardware design of high performance miniature anthropomorphic robots," *Robotics and Autonomous System*, vol. 56, pp. 82–94, 2007.