CHAPTER 3

Problem Description

According to motion granularity, the motion-planning problem usually can be classified into global (gross) motion planning and local (fine) motion planning. Global motion planning problem concerns with working out the body logistics, such as planning a collision-free path between trees in a forest to reach some destination while local motion planning problem focuses more on limb logistics, such as planning hand motion to grab an awkwardly placed object. For the problem of walking on a layered environment for a virtual human, both types of planning needs to be considered in order to ensure that the desired task can be accomplished. The global path from the global motion planner is fed into the local motion planner to create corresponding footsteps and locomotion. However, the local motion planner may fail to generate locomotion for the given path. In this case, the planner should feedback the failure with reasons to the global planner to replan another global path. This replanning problem addresses the issues of how to utilize these feedback messages and previous search results to find an alternative path efficiently. In the following sections, we will describe the problems of global planning and local planning separately in more details. However, the global motion planning will be the main concern of this thesis.

3.1. Overall Problem Description

The input to a typical motion planning problem for a virtual human consists of the initial and
goal configurations of the virtual human, the locomotion abilities of the virtual human, and a geometric description of objects in the environment. In our approach, the planning problem can be decomposed into two sub-problems: the global motion planning for moving the body trunk of a virtual human and the local motion planning for realizing the global motion plan with appropriate locomotion. Although the global and local motion planners can be designed separately and solved sequentially, we think they should be linked together to solve the problem in sequence as well as to feedback failure situation for further replanning as shown in Fig. 3.1.

The inputs to our system include the locomotion abilities and kinematic description of the virtual human, and a geometric description of objects in the environment. The preprocessing phase then uses this information to construct reachability and collision model. User only needs to specify the initial and goal configuration of the virtual human, and the global planner will produce a path that satisfies all constraints and preferences. Then, the global path from the global motion planner is translated into a 1D ground profile and fed into
the local motion planner to create corresponding footsteps placement and locomotion. However, the local motion planner may fail to generate locomotion for the given path. In this case, the planner should feedback the failure with reasons to the global planner to compute another global path. The reasons should include the locomotion type, failure point, humanoid position and failure types. Taking this decoupled view can greatly reduce the complexity of such a planning problem. In the following sections, we will describe first the problems of humanoid modeling and then the global planning and local planning separately in more details.

3.2. Virtual Human Modeling

In our system, we use the virtual human model that is fully conformed to Humanoid Animation standard Level of Articulation one (LOA 1), which specifies a typical low-end real-time 3D hierarchy. In LOA 1, 18 joint objects of virtual human are defined, which indicates that more than 36 DOFs are involved. In global motion planning for a virtual human, we assume that the virtual human can be modeled as a bounding cylinder to simplifying collision detection. In our system, we use a large enclosing cylinder and a small enclosed cylinder to model a virtual human for various purposes. An example of this two-level modeling is shown in Fig. 3.2.(a) and (b) for frontal and lateral walking, respectively.

The radius $r$ of the inner cylinder is the size of the minimal region for a stable stance. The radius $R_L$ of the outer cylinder is determined according to the locomotion $L$. This bounding cylinder can greatly simplify collision detection. We use the largest lateral width orthogonal to the virtual human’s moving direction to determine $R_L$. For example, the
The cylinder used to plan the side-walk motion is smaller than the one for regular walking motion. The heights of both cylinders are also related to the virtual human’s height and locomotions. If we allow the virtual human to bend its upper body during the walking cycle, the actual height may be lower than the virtual human’s height. When using cylinders to model the geometry of a virtual human, we are actually ignoring its orientation at the planning time. We can recover the orientation of a virtual human in a post-processing step according to the locomotion used.

We assume that a virtual human can perform several locomotions and choose the most appropriate one according to the environment. Although we can consider several major locomotions, such as walking, crawling, jumping and climbing in the local motion planner, in current system, our local motion planner only supports frontal walking, lateral walking and
jumping. Moreover, we assume that the local planner can generate the motion transition from one type of locomotion to another at a given configuration without causing collisions.

3.3. **Global Motion Planning Problem**

The global planner assumes that we are given a geometric description of the objects in the workspace, locomotion abilities and the geometric and kinematic description of a virtual human. The workspace contains multiple layers, and each layer comprises of objects of various heights. Unlike the basic path-planning problem where the definition of obstacles is rather straightforward, the obstacles in our global planning problem are not explicitly given. Instead, an object is an obstacle to a virtual human only if there is no way for the humanoid to step onto or pass under the object due to the virtual human’s height and leg length. Besides, locomotion abilities are also key measurements to decide the obstacle region. For example, the narrow separated regions are not obstacles for jumping ability. In addition, a virtual human must stand on a large enough area in order to maintain a stable stance. If the ground of the workspace is described as a smooth surface, the slope of the surface cannot be too large to cause foot slippery. In sum, the planning problem is rather complex in real life, and we need to make assumptions to simplify the problem.

1. We assume a discrete workspace. The input to our planner could be a continuous function for the elevation of the ground and a polygonal description of the objects. However, we assume that we can convert these descriptions into several layers of elevation grids of some resolution. Each cell in a grid contains an elevation value for the whole cell in that layer. We denote the height of a cell $i$ at layer $l$ by $\alpha_{il}$, and the offset of
layer $l$ from some reference ground by $d^l$.

2. We assume that the resolution of the elevation grid is coarse enough for a virtual human’s foot to step onto a cell. We also assume that the maximal height that a virtual human can step onto is denoted by $h$, which is a property of the given virtual human.

3. The height of the virtual human is $H$, and we assume that the virtual human does not bend its body to pass an obstacle for now.

4. We assume that a virtual human will not stay in the object boarder region for more than some designated units of time, $M$. This situation happens when the geometry of a humanoid intersects the boarder. This assumption is to make sure that the virtual human does not stay in the boarder region except for the trespassing purpose. We also assume that a virtual human will not stay in the deep gap for more than some designated units of time, $N$. This assumption ensures that the virtual human does not stay in the gap region except for the striding purpose.

5. We assume that the geometry of the virtual human can be simplified to a bounding cylinder such that the orientation can be ignored and the collision detection can be simplified at planning time. We assume that a virtual human will always face forward and we can recover its orientation in a post-processing step.

To summarize, the objective of the global motion planner is to find a collision-free path for the body trunk of the virtual human to move from the initial configuration to the goal.
configuration in a two-and-half-dimensional space. The output of the planner is a global path that will be sent to the local planner for further processing.

3.4. Local Motion Planning Problem

The local planner described below is proposed and implemented by P.F. Chen. Details of the approach can be referred to [22]. The local planner aims to find a feasible locomotion for the lower body of a virtual human with a given global path. We assume that the output path from the global planner is a 3D stepwise curve. This curve is a polyline comprised of a set of vertical or horizontal connected line segments. In other words, we temporarily ignore the orientation change of the path and stretch the path into a one-dimensional stair-like profile, as shown in Fig. 3.3. According to the kinematic parameters of the virtual human, the local planner will generate a feasible and efficient plan for footstep placement and the corresponding locomotion for lower-body joints. A feasible motion plan must satisfy geometric and kinematic constraints. For example, the virtual human should be collision-free, and all joints are within their joint limits.
The local planning problem described above is challenging because the number of possible arrangements (each arrangement consists of a set of footsteps) grows exponentially with the length of the global path (or number of footsteps) even if we restrict the possible footstep sizes to a limited number. However, according to our daily walking experience, we typically plan foot placement only for the next two or three steps instead of for the whole path. Therefore, it is reasonable to take an incremental approach where we call the local planner in every step to plan only for a few steps (two or three, typically) ahead. Fig. 3.4 shows the examples of planning one and two steps ahead.

Another advantage of this approach is that we can allow the configuration of obstacles to change at run time without calling for global replanning immediately as long as the change does not prevent the local planner from generating feasible locomotion. Thus, we will redefine our local planning problem as finding a feasible locomotion for the next \( n \) steps with a given path profile. For the problem of finding foot placement, we assume that the local planner is called in every step of execution, and we will search for the most energy efficient
locomotion for the next $n$ steps, where $n$ is set to, say, 3. We take a simpler approach of computing this energy from statistic data obtained from real human walking experiments. [22] assume that the energy consumption is composed of three parts: horizontal movement, vertical movement, and joint movement. With linear combination of these measurements, we can compute the most efficient placement for each footstep, as shown in Fig. 3.5. The planner should return failure and indicate the failure location along the path if it cannot find a feasible locomotion plan for the next $n$ steps.

3.5. Replanning Problem

When the local planner returns the reason for failure, the global planner should use this information to generate another path efficiently. We assume that the local planner is able to feedback the failure reasons with the following information:

1. Failure point: the position that local planner can not generate the collision-free foot placement. This point should be marked to prevent the planner from generating the same
result. Since the local planner makes the footstep planning on the 1-D profile, the feedback point needs to be translated back to the 3D representation.

2. Current position of virtual human: our local planner plans \( n \) steps ahead at each step, which means that when local planner fails, the failure point can be discovered in \( n \) steps ahead. Then the global planner can replan immediately after the failure is reported. Therefore, we need to know the current position of the virtual human and use this point as the new initial configuration \( q_{ini} \) of the replanning problem.

3. Locomotion type: if the local planner fails, we only mark the failure point at the collision map for the given locomotion instead of the workspace. This approach can preserve the completeness of the planning since the possible replanning solutions may use another locomotion type.

How the global planner incrementally updates the environmental information by utilizing the feedback reasons will be main concern of the replanning problem. The ideal replanning routine should not construct or reconstruct additional preprocessing terrain knowledge and should have the performance of global path searching or even better.