Perception-Based Gait Planning for a Hexapod Robot Walking on Typical Structured Terrain

Xun Chai, Feng Gao and Yilin Xu

Abstract Legged robots have excellent terrain adaptability and can be used to accomplish rescuing and detecting tasks instead of human beings in harsh environment. This paper presents a framework developed to increase the autonomy and versatility of a hexapod robot. It combines terrain perception with four locomotion strategies, a flat-floor gait, a step-on gait, a step-down gait and a ditch-over gait. This way the robot can perceive the environment and distinguish four typical structured terrain, flat-floor step-on, step-down and ditch. Based on different terrain, the appropriate locomotion strategy is selected to be carried out. The terrain perception and the gait selection are performed autonomously. We present experiment trials of the Hexapod-III robot walking in structured environment including the flat-floor, step-on, step-down and ditch. The experiment results show that the robot has the ability to distinguish four typical structured terrain and pass through them autonomously.

Keywords Gait planning · Hexapod robot · Terrain perception

1 Introduction

In recent years, Fukushima nuclear power plant accident and other natural disasters like earthquakes, mudslides have prompted relevant research domains' awareness of importance and necessity of rescue robots. Compared with wheeled/tracked robots, legged robots only need some discrete footholds for locomotion on the

X. Chai · F. Gao (✉) · Y. Xu
State Key Laboratory of Mechanical System and Vibration, Shanghai
Jiao Tong University, Shanghai, China
e-mail: fengg@sjtu.edu.cn

X. Chai
e-mail: chaixun@sjtu.edu.cn

Y. Xu
e-mail: xuyilin@sjtu.edu.cn

X. Zhang et al. (eds.), *Mechanism and Machine Science*,
Lecture Notes in Electrical Engineering 408,
DOI 10.1007/978-981-10-2875-5_15

ground, which makes legged robots more suitably operate in complex and rugged environments. In last few years, many legged robots are designed. Generally legged robots can be classified into three categories, biped robots, quadruped robots and hexapod robots. For the famous biped robots, see Petman [1], Atlas [2], ASIMO [3]. For the famous quadruped robots, see BigDog [4], HyQ [5], SILO4 [6]. For the famous hexapod robots, see Ambler [7], Athelete [8], DLR Crawler [9], RHex [10]. The number, the mechanism and the layout of legs need well-designed in order to satisfy the requirements of stability, load capacity, walking velocity, DOFS and control.

In structured environments, legged robots need to build terrain map with the help of necessary vision sensors mounted on them. Then legged robots can plan the gait to adapt various terrain based on the terrain map. Some studies have concerned about terrain adaption for legged robots. Matt zucker and Nathan Ratliff [11] presented a novel optimization method for a quadruped robot, which can walk on challenge terrains. Their method plans a set of footholds and dynamic body motions by thoroughly rooting in optimization. Estremera and Cobano [12] proposed a continuous free-crab gait planning method for a six-legged robot. Their method derived three control modules relying on two heuristic empirical rules. They [13] also presented a method to generate continuous free crab gaits for the quadruped robots on rough terrains. Their experimental results showed the SILO4 quadruped robot could perform stable and omnidirectional locomotion on rough terrains. Hoepflinger and Hutter [14] introduced a method to evaluate and estimate the mechanical robustness of footholds for legged robots in irregular terrain without using the human expert knowledge or human defined criteria. Haynes and Rizzi [15] developed a robust method allowing the specification, control and transition of the stepping pattern for a six-legged robot. Their method generated gaits by merging through controllers that imposed appropriately placed repellors and the torus of relative leg phases.

In this paper, a perception-based walking strategy for the Hexapod-III robot on typical structured terrain is presented. The structure of this paper is organized as follows. In Sect. 2, the system overview and the problem formulation are introduced briefly. Processes of terrain map modeling and structured terrain classifying are described in Sect. 3. In Sect. 4, walking strategies for four typical structured terrain are presented in detail. Motion planning for the legged robot is discussed in Sect. 5. The experimental results are presented in Sect. 6. Section 7 summarizes and concludes the paper.

2 System Overview and Problem Formulation

As Fig. 1 shows, the six-legged robot is called Hexapod-III, which has six legs symmetrically arranged around the body. Its leg is a parallel mechanism having three chains, two of which are constructed by the universal joint, prismatic joint and spherical joint, the other one is constructed by the universal joint and prismatic

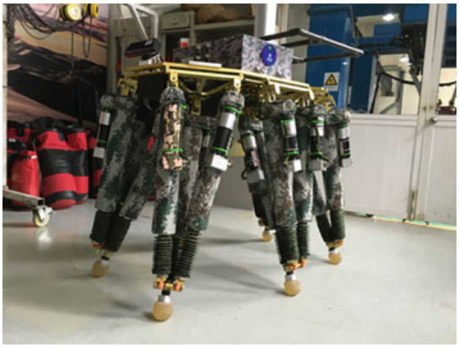

Fig. 1 The Hexapod-III robot

joint. The linear movement of the prismatic joint is achieved by ball-bearing screw which is actuated by the motor. The control unit is an industrial computer running real-time Linux, which sends planned data to drivers via EtherCAT. The robot is equipped with a stereo camera, a six-axis force/torque sensor and an IMU.

As Fig. 2a shows, the robot has 18 inputs $(q_i, i = 1{-}18)$, which are the prismatic positions. It has 24 outputs, 18 of which are the feet positions $(x_{F_i}, y_{F_i}, z_{F_i}, i = 1{-}6)$, the other six outputs are the positions and orientations of the body $(x_B, y_B, z_B, \alpha_B, \beta_B, \gamma_B)$.

During walking, four coordinate systems, the ground O_G, the stereo camera O_C, the body O_B and the foot O_F, exist all the time. Three transformation relationships T_C^R, T_R^G and T_R^F must be obtained. T_C^R is the transformation matrix from the stereo camera to the robot. The method of solving T_C^R has been presented in our previous work. T_R^G is the transformation matrix from the robot to the ground, which can be obtained from the body trajectory of the robot. T_R^F is the transformation matrix from the foot to the robot body which can be solved by the robot kinematics.

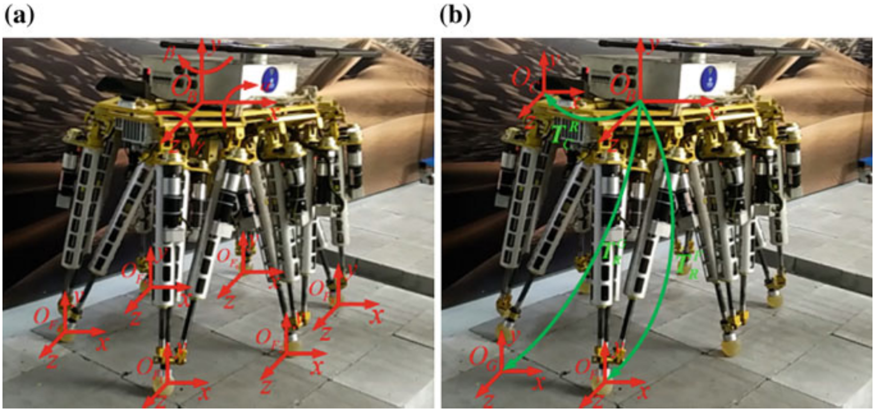

Fig. 2 **a** Inputs and outputs of the robot, **b** definition of the coordinate systems

3 Terrain Map Building and Terrain Classification

Accurate 3D coordinates of the terrain in front of the robot can be obtained from the depth image captured by the stereo camera. The raw terrain data consists of large amounts of point cloud, which spends too much storage space and takes a long time for the computer to process. Taking into account this issue, we decide to use a grid-type map to decrease the data amount and increase the updating and processing rate of the map. The grid-type map describes the terrain by using multiple square-shaped grids, and each grid stores the real height of the terrain.

Figure 3 shows the process of grid terrain map building. Firstly, the point cloud ${}^{C}P$ in O_C should be transformed in O_G by the following equation:

$$ {}^{G}P = T_R^G \cdot T_C^R \cdot {}^{C}P \tag{1} $$

Secondly, the point cloud ${}^{G}P$ is mapped into the corresponding grid by the following formulas:

$$ \begin{aligned} x_{\mathrm{G}}(i,j) &= (i-60) \times 0.025 \\ z_{\mathrm{G}}(i,j) &= j \times 0.025 \\ y_{\mathrm{G}}(i,j) &= \frac{\sum_{n=1}^{n(i,j)} y_w(n)}{n(i,j)} \end{aligned} \tag{2} $$

Considering the capacity of Hexapod-III, the grid size is regulated as 0.025 m × 0.025 m, and the grid number is 120 × 120. So the grid map represents the real terrain whose dimension is 3 m × 3 m. The grid height is obtained by

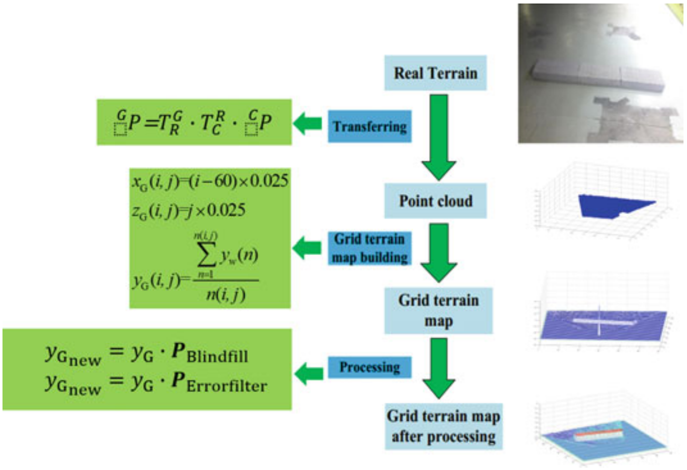
Real Terrain
${}^{G}P = T_R^G \cdot T_C^R \cdot {}^{C}P$
Transferring
$x_G(i,j) = (i-60)\times 0.025$
$z_G(i,j) = j \times 0.025$
$y_G(i,j) = \frac{\sum_{n=1}^{n(i,j)} y_w(n)}{n(i,j)}$
Point cloud
Grid terrain map building
Grid terrain map
$y_{G_{new}} = y_G \cdot P_{Blindfill}$
$y_{G_{new}} = y_G \cdot P_{Errorfilter}$
Processing
Grid terrain map after processing

Fig. 3 Grid terrain map building

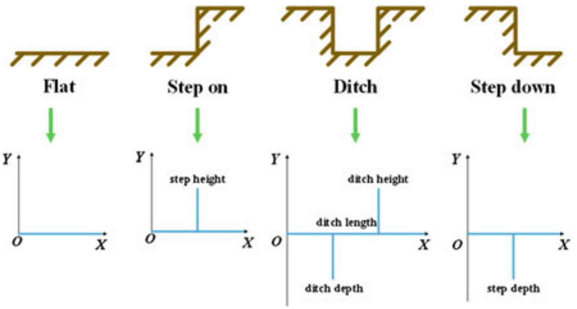

Fig. 4 Terrain classification

calculating the average height of the point cloud which are located in the same grid. Thirdly, two different kinds of wrong grid exists, which maybe influence the correctness of the map. One is the wrong data due to the outer noise and the sensor itself. The other one is the missing portion of the terrain caused by inevitable effect of occlusions in the sensors line of sight. The gird terrain map has to be processed further in order to exclude wrong grids as Fig. 3 shows. The processing algorithms are out of the scope of this article and won't be discussed here.

In this paper, we mainly focus on the classification of four common terrain in reality, the flat-floor, step-on, ditch, and step down. Based on the grid terrain map, we can extract the geometric features of the four terrain as Fig. 4 shows. For the flat-floor terrain, there is no rising and falling edges in the height direction. For the step-on terrain, there is only a rising edge in the height direction. The position and the height of the step-on terrain are decided by those of the rising edge. For the ditch terrain, there is a falling edge first, subsequently followed by a rising edge. The position of the ditch is decided by that of the falling edge. The length of the ditch is calculated by the differential position of the rising and falling edges. For the step-down terrain, there is only a falling edge in the height direction. The position and the depth of the step-down terrain are decided by those of the falling edge.

4 Walking Strategies

In this section, walking strategies are discussed in detail. The robot chooses the tripod waling gait, which is the fastest stable gait for the hexapod robot. The tripod walking gait enables the robot to have three nonadjacent feet touching with the ground all the time, the center of mass (COM) can be located in the support triangle easily as Fig. 5 shows.

For different terrain, the robot applies two different gait trajectories, the rectangular trajectory and the ellipse trajectory. The rectangular trajectory is typical but quite useful as shown in Fig. 6a. The determination of the rectangular trajectory only needs three parameters, the step length *a*, the height of the lifting foot *b* and the height of the falling foot *c*, which facilitates the planning process a lot. The

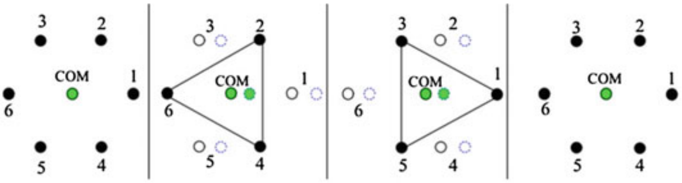

Fig. 5 The tripod walking gait

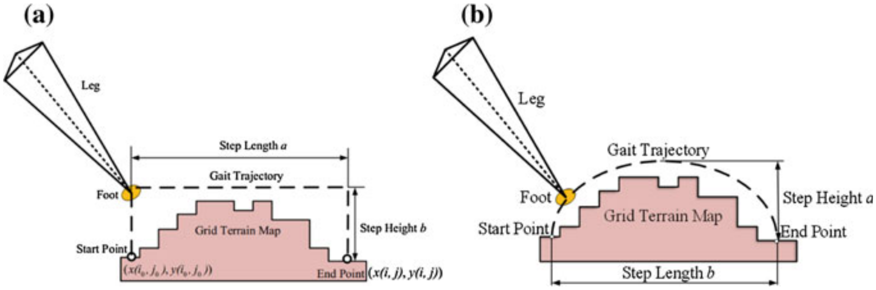

Fig. 6 **a** The rectangular trajectory, **b** the ellipse trajectory

rectangular gait is used on step-on and step down terrain, for it is easy to be planned to avoid the collision with the rising and falling edges. Ellipse trajectory is shown in Fig. 6b, which is smooth and has a shorter length compared to the rectangular trajectory. The ellipse trajectory is decided by two parameters the step length b and the step height a. It is used on flat-floor and ditch terrain, for it can satisfy the requirement of high walking speed and large workspace of the foot.

When the robot passes through the step-on, ditch and step down terrain, the detection error and the motion error may cause the robot to falling down and collision with the terrain. In order to ensure the robot's safety, it is important for the robot to move to a safe pose. As Fig. 7 shows, the safe pose is defined as a pose where the robot trunk is far from the sides' edge of the terrain and it is vertical to the front edge of the terrain. When the terrain is not a flat-floor after distinguishing, the edge of the terrain is detected and located. Then the robot will turn an appropriate angle to face the terrain vertically. If any foot of the robot is outside the terrain, it will move left or right to make its foot far away from the sides' edge of the terrain.

The walking strategy on step-on and step-down terrain are substantially similar, the detailed walking process is shown in Fig. 8. The robot will detect the position and the height of the step-on terrain (the position and the depth of the step-down terrain), which has been described in detail above. Then the robot decides to use the rectangular trajectory. The robot will first move close to the step to ensure the position of the next foothold within the leg's workspace. The robot will lift three nonadjacent feet, move them forward to a specified distance, then fall them down.

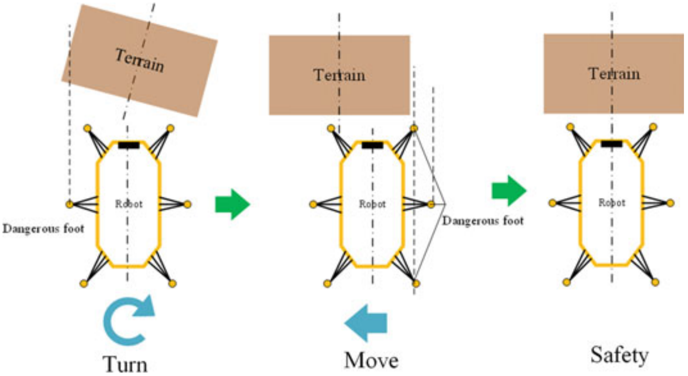

Fig. 7 Edge detection and pre-adjustment for safety

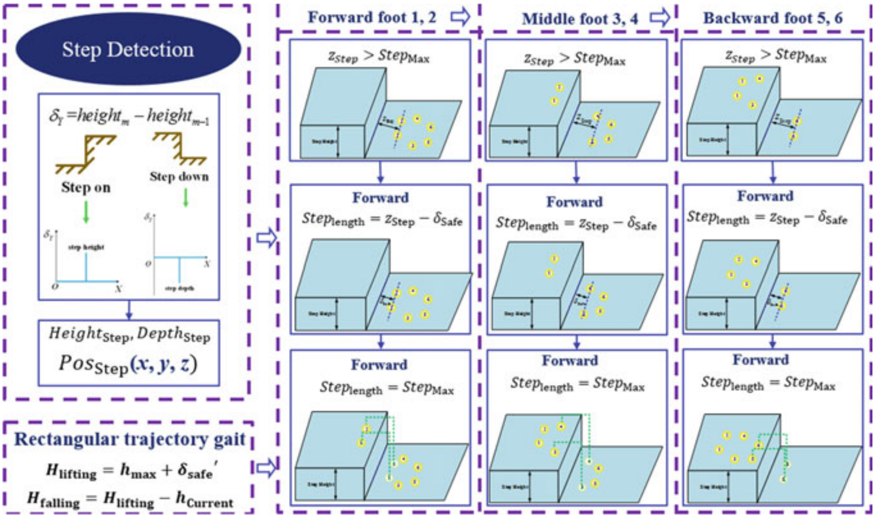
Step Detection
$\delta_Y = height_m - height_{m-1}$
Step on
Step down
δ_Y
step height
O
X
step depth
$Height_{Step}, Depth_{Step}$
$Pos_{Step}(x, y, z)$
Rectangular trajectory gait
$H_{lifting} = h_{max} + \delta_{safe}'$
$H_{falling} = H_{lifting} - h_{Current}$
Forward foot 1, 2
Middle foot 3, 4
Backward foot 5, 6
$z_{Step} > Step_{Max}$
Forward
$Step_{length} = z_{Step} - \delta_{Safe}$
Forward
$Step_{length} = Step_{Max}$

Fig. 8 Walking strategy on step-on and step-down terrain

Then the other three feet are motivated to move in the similar way. The robot will move its front two feet on or down the step, then the middle two feet, at last the back two feet. The step length, the foot lifting height and the foot falling height are calculated using following formulas.

$$\begin{aligned} Step_{\text{length}} &= Step_{\text{max}} \\ H_{\text{lifting}} &= h_{\text{max}} + \delta'_{\text{safe}} \\ H_{\text{falling}} &= H_{\text{lifting}} - h_{current} \end{aligned} \tag{3}$$

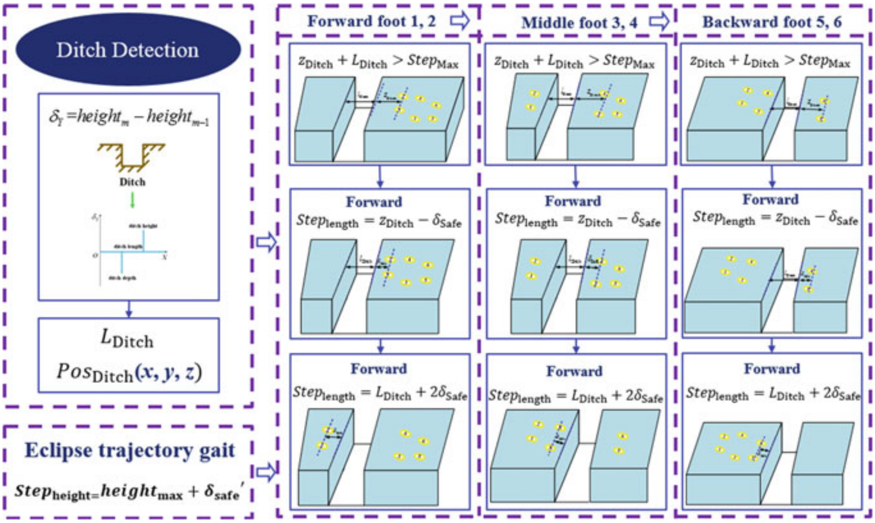
Ditch Detection
$\delta_T = height_m - height_{m-1}$
Ditch
L_{Ditch}
$Pos_{\text{Ditch}}(x, y, z)$
Eclipse trajectory gait
$Step_{\text{height}} = height_{\max} + \delta_{\text{safe}}'$
Forward foot 1, 2
Middle foot 3, 4
Backward foot 5, 6
$z_{\text{Ditch}} + L_{\text{Ditch}} > Step_{\text{Max}}$
Forward
$Step_{\text{length}} = z_{\text{Ditch}} - \delta_{\text{Safe}}$
Forward
$Step_{\text{length}} = L_{\text{Ditch}} + 2\delta_{\text{Safe}}$

Fig. 9 Walking strategy on ditch terrain

where $Step_{\text{length}}$ is the step length, $Step_{\max}$ is the maximum step length within the foot's workspace. H_{lifting} is the foot lifting height, $h_{\max}$ is the maximum height of the terrain. For the step-down terrain, $h_{\max}$ is the maximum depth of the terrain. δ'_{safe} is the safety margin avoiding the collision of the terrain. H_{falling} is the foot falling height, $h_{current}$ is the height of the next foothold.

The walking strategy on the ditch terrain is shown in Fig. 9. The robot will detect the position, and the length of the ditch first. Then it decides to use the ellipse trajectory. Before swinging its foot to the other side of the ditch, the robot will come close to the ditch in order to make sure the step length is longer than the ditch length. The robot will lift three nonadjacent feet, swing them to the planned footholds. Then it moves the other three feet in the same way. As Fig. 9 shows, the robot will first swing the front foot 1, 2 to the other side of the ditch. Then it comes close to the ditch and swings the medial foot 3 and 4 to the other side. At last it moves forward again and swings the back foot 5, 6 to the other side. The step length and the step height are calculated from following formulas:

$$\begin{aligned} step_{\text{length}} &= L_{\text{Ditch}} + 2\delta_{\text{Safe}} \\ step_{\text{height}} &= height_{\max} + \delta'_{\text{Safe}} \end{aligned} \tag{4}$$

where L_{Ditch} is the length of the ditch, $height_{\max}$ is the maximum height value of terrain within the ellipse trajectory. δ_{Safe} is the safe margin along the longitudinal direction, δ'_{safe} is the safe margin along the height direction.

Walking strategy on the flat-floor is basically the same to that on the ditch. The robot uses ellipse trajectory gait. The robot can walk with a high speed on the flat-floor. So the step length can reach the maximum length within its workspace. And the step height can be obtained by adding a small safe margin to avoid little terrain bumps.

5 Motion Planning

After obtaining the gait trajectory, prismatic positions should be computed to control the robot to walk along the planned trajectory. Inverse kinematics is needed, and our previous work have finished the kinematics model of the robot [16].

Figure 10 shows the kinematic model of single leg. The position of spherical joint S_f connecting the leg with the foot is obtained from the above section. U_1, U_2, U_3 are three universal joints connecting the leg with the body. S_1, S_2 are two spherical joints, which connects links with ankle joints. P_1, P_2, P_3 denote three prismatic joints, whose lengths are l_1, l_2, l_3. The geometric constraint equation of inverse kinematics is represented as follows,

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = {}^{L}A \begin{bmatrix} S_{f_x} \\ S_{f_y} \\ S_{f_z} \\ 1 \end{bmatrix} \tag{5}$$

where ${}^{L}A$ is the transformation matrix from the ankle coordinate system (ACS) to the leg coordinate system (LCS). $S_{f_x}, S_{f_y}, S_{f_z}$ are coordinates of S_f with respect to ACS. (x, y, z) denotes the coordinates of S_f with respect to LCS. Formula (6) is the detailed expression of ${}^{L}A$,

$${}^{L}A = \begin{bmatrix} \cos\alpha_1 \cos\beta_1 & -\cos\alpha_1 \sin\beta_1 & \sin\alpha_1 & l_1 \cos\alpha_1 \cos\beta_1 \\ \sin\beta_1 & \cos\beta_1 & 0 & l_1 \sin\beta_1 \\ -\sin\alpha_1 \cos\beta_1 & \sin\alpha_1 \sin\beta_1 & \cos\alpha_1 & -l_1 \sin\alpha_1 \cos\beta_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6}$$

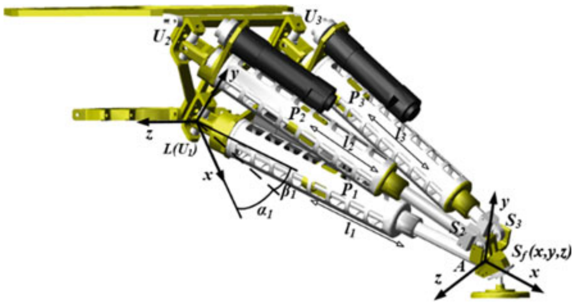

Fig. 10 Kinematic model of single leg

where α_1, β_1 are rotation angles of U_1 along two vertical directions, we can get the solution of the inverse kinematics by solving Eq. (5),

$$\begin{aligned} l_1 &= \sqrt{x^2 + y^2 + z^2 - S_{f_y}^2 - S_{f_z}^2} - S_{f_x} \\ \beta_1 &= \arcsin\left(\frac{y}{\sqrt{\left(l_1 + S_{f_x}\right)^2 + S_{f_y}^2}}\right) - \arcsin\left(\frac{S_{f_y}}{\sqrt{\left(l_1 + S_{f_x}\right)^2 + S_{f_y}^2}}\right) \\ \alpha_1 &= \arctan\left(\frac{S_{f_x} x - \left[\left(l_1 + S_{f_x}\right)\cos\beta_1 - S_{f_y}\cos\beta_1\right] z}{\left[\left(1 + S_{f_x}\right)\cos\beta_1 - S_{f_y}\sin\beta\right] x + S_{f_z}}\right) \end{aligned} \tag{7}$$

${}^{L}A$ can be calculated by substituting α_1, β_1, l_1 into Eq. (6), and lengths l_2, l_3 of prismatic joints P_2, P_3 can be obtained from following equations,

$$\begin{aligned} l_2 &= \left|\overrightarrow{{}^{L}U_2 {}^{L}S_2}\right| = \left|{}^{L}A \cdot {}^{A}S_2 - {}^{L}U_2\right| \\ l_3 &= \left|\overrightarrow{{}^{L}U_3 {}^{L}S_3}\right| = \left|{}^{L}A \cdot {}^{A}S_3 - {}^{L}U_3\right| \end{aligned} \tag{8}$$

where ${}^{L}U_2, {}^{L}U_3, {}^{L}S_2, {}^{L}S_3$ are positions of joints U_2, U_3, S_2, S_3 with respect to LCS respectively, and ${}^{A}S_2, {}^{A}S_3$ are positions of joints S_2, S_3 with respect to ACS respectively.

6 Experiments

In this section, experimental results are presented. Figure 11 shows the process of walking on the stair. The robot correctly detects the position and the height of the stair. Then it moves to the safe pose to guarantee safety. Before placing the feet on

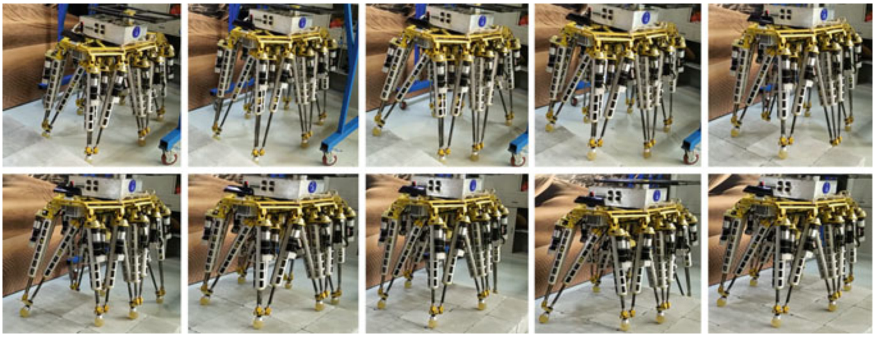

Fig. 11 Snapshots of walking on step-on terrain

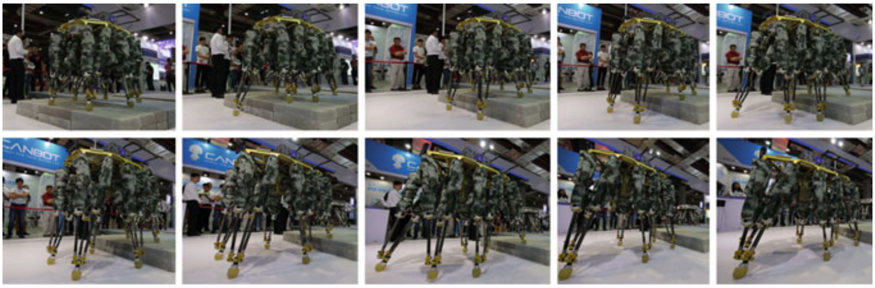

Fig. 12 Snapshots of walking on step-down terrain

the stair, it moves close to the stair to ensure that next footholds are within the workspace. And the robot trunk moves forward horizontally because of the accurate foot falling height.

Figure 12 shows the process of walking down the stair. The position and the depth of the stair are detected correctly. Then the robot moves to the safe pose. Before placing the feet on the ground, it moves close to the stair to ensure next footholds are within the workspace. From Fig. 12, we can see that the robot trunk maintains horizontally, which validates the walking strategy on step-down terrain.

Figure 13 shows the process of passing through the ditch. The position and the length of the ditch are calculated correctly. The robot uses the ellipse trajectory gait. Before swinging the feet to the other side of the ditch, it moves close to the nearer edge of the ditch to ensure the next step length is longer enough to pass the ditch safely. From Fig. 13, we can see that the robot passes through the ditch successfully.

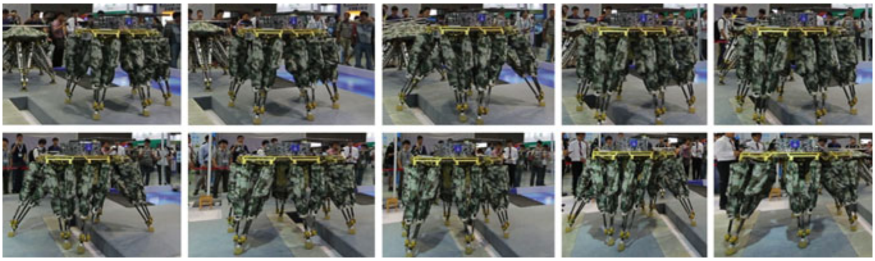

Fig. 13 Snapshots of passing through ditch

7 Conclusions

In this paper, we have presented a perception-based walking strategy for a hexapod robot on typical structured terrain. The walking strategy is based on the grid-type terrain map, which is modeled from point cloud captured by a stereo camera. The geometric feature of four typical structured terrain, the flat-floor, step-on, step-down and ditch are abstracted and detected from the grid-type map. In order to guarantee the robot safety, the pre-adjustment strategy is proposed. Then four walking strategies for different terrain are presented in detail. The motion planning method based on the robot kinematics is discussed too. At last, a serial of experiments are carried out and the results validate the practicability and theoretically of the proposed strategy.

Acknowledgments This study was supported by the National Basic Research Program of China (973 Program, No. 2013CB035501).

References

1. Nelson G et al (2012) Petman: a humanoid robot for testing chemical protective clothing. J Robot Soc Japan 30(4):372–377
2. Boston D http://www.bostondynamics.com/robot_Atlas.html
3. Hirose M, Ogawa K (2007) Honda humanoid robots development. Philos Trans R Soc A: Math Phys Eng Sci 365:9
4. Raibert M et al (2008) Bigdog, the rough-terrain quadruped robot. In Proceedings of the 17th world congress
5. Semini C, Tsagarakis NG, Guglielmino E, Focchi M, Cannella F, Caldwell DG (2011) Design of HyQ—a hydraulically and electrically actuated quadruped robot. J Syst Control Eng 225 (6):19
6. Santos PGD, Gálvez JA, Estremera J, García E (2003) SILO4—a true walking robot for the comparative study of walking machine techniques. IEEE Robot Autom Mag 10(4):10
7. Krotkov E, Simmons R (1996) Perception, planning, and control for autonomous walking with the ambler planetary rover. Int J Robot Res 15(2):26
8. Wilcox BH et al (2007) ATHLETE: a cargo handling and manipulation robot for the moon. J Field Robot 24(5):421–434
9. Görner M, Wimböck T, Hirzinger G (2009) The DLR Crawler: evaluation of gaits and control of an actively compliant six-legged walking robot. Ind Robot: Int J 36(4):8
10. Haynes GC, Rizzi AA (2006) Gaits and gait transitions for legged robots. In Proceedings of the IEEE international conference on robotics and automation
11. Zucker M, Ratliff N et al (2011) Optimization and learning for rough terrain legged locomotion. Int J Robot Res 30(2):175–191
12. Estremera J, Cobano JA (2010) Continuous free-crab gaits for hexapod robots on a natural terrain with forbidden zones: an application to humanitarian demining. Robot Auton Syst 58 (5):700–711
13. Estremera, J, de Santos PG (2005) Generating continuous free crab gaits for quadruped robots on irregular terrain. IEEE Trans Robot 21(6):1067–1076
14. Hoepflinger, MA, Hutter M (2013) Unsupervised identification and prediction of foothold robustness. In IEEE international conference on robotics and automation

15. Haynes G, Rizzi A, Koditschek D (2012) Multistable phase regulation for robust steady and transitional legged gaits. Int J Robot Res 21(14):1712–1738
16. Pan Y, Gao F (2013) A new 6-parallel-legged walking robot for drilling holes on the fuselage. Proc Inst Mech Eng, Part C: J Mech Eng Sci 228:753–764