

Research Article

Discrete-State-Based Vision Navigation Control Algorithm for One Bipedal Robot

Dunwen Wei

School of Mechanical Engineering, Northwestern Polytechnical University, Xi'an, Shaanxi 710072, China

Correspondence should be addressed to Dunwen Wei; dunwenwei@hotmail.com

Received 28 November 2014; Revised 28 February 2015; Accepted 3 March 2015

Academic Editor: Victor Santibáñez

Copyright © 2015 Dunwen Wei. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Navigation with the specific objective can be defined by specifying desired timed trajectory. The concept of desired direction field is proposed to deal with such navigation problem. To lay down a principled discussion of the accuracy and efficiency of navigation algorithms, strictly quantitative definitions of tracking error, actuator effect, and time efficiency are established. In this paper, one vision navigation control method based on desired direction field is proposed. This proposed method uses discrete image sequences to form discrete state space, which is especially suitable for bipedal walking robots with single camera walking on a free-barrier plane surface to track the specific objective without overshoot. The shortest path method (SPM) is proposed to design such direction field with the highest time efficiency. However, one improved control method called canonical piecewise-linear function (PLF) is proposed. In order to restrain the noise disturbance from the camera sensor, the band width control method is presented to significantly decrease the error influence. The robustness and efficiency of the proposed algorithm are illustrated through a number of computer simulations considering the error from camera sensor. Simulation results show that the robustness and efficiency can be balanced by choosing the proper controlling value of band width.

1. Introduction

Vision navigation of mobile robots has been known as an open and challenging problem over the last few decades [1, 2]. Currently, most of the vision navigation systems are investigated based on 3D space [3–5]. 3D vision navigation needs multicamera [3, 4, 6] or omnidirectional camera [7] to provide the robot with data on its orientation and direction. In some cases, the hardware that is needed to implement the algorithm can be more costly than the robot itself. This fact makes the practical realization of such method in most real world robotic systems questionable. However, for the robot moving on a plane ground, it is feasible to navigate using single camera system [8–12]. The beauty of single camera navigation system is low cost and it can make the image processing simple. For the real application of one bipedal robot called Cornell Ranger [13], the goal of task is to track specific objective (the specific objective is the tracked trajectory composed of the point, line, and curve corresponding to the precise position, straight path, and curving path). However, the Cornell Ranger has its inherently required

features. Firstly, the Cornell Ranger is a kneeless 4-legged bipedal robot which is energetically and computationally autonomous. The light weight and low energy cost are the significant aspects and should be considered seriously. From the view of experience, single camera navigation system is a better option to be considered compared with multicamera navigation system, because multicamera navigation system would consume much more energy and the implementation of it is much more complex. Secondly, the wheeled mobile robots can capture a sequence of views at the same level height, while for bipedal robots, the discrete walking gaits determine that the vertical position of cameras mounted in robot is varying, so a large sequence of images captured by cameras are under different level height, which gives rise to more complex vision processing.

Vision navigation with the specific objective usually named as path following [14, 15] can be defined by specifying desired timed trajectory and includes both aspects, path recognition and path planning. On the one hand, the path recognition, usually called path finding [16], is to reconstruct surrounding scene and to find the specific objective by vision

system. Although many environment features may be visible from the image captured by camera, only few of these features are necessary to estimate the robot's position and orientation [17]. How to extract the useful information from a single camera is a difficult problem. On the other hand, the path planning [18] is a navigation algorithm which leads the robot to the specific objective according to the information from the vision system. Currently, various algorithms have been developed to solve the path planning problem for autonomous robot navigation, such as genetic algorithm [19] and fuzzy logical controller [20]. The accuracy, efficiency, and robustness are three important indices to evaluate navigation algorithm. The accuracy and robustness of vision navigation algorithm proposed in [8] were evaluated qualitatively by experiments. In [21], one localization algorithm based on single landmark was proposed and its effectiveness was shown by simulation. However, there are no quantitative definitions established about accuracy, efficiency, and robustness to evaluate vision navigation quantitatively. Moreover, fewer algorithms have dealt with the problem of how to balance the robustness and efficiency.

There are three contributions in this paper. Firstly, the concept of desired direction field is proposed. In order to estimate the robustness and efficiency of navigation algorithm, the quantitative definitions of robustness and efficiency are given. This concept can be expanded to deal with all navigation problems of mobile robots with the specific objective. Secondly, in order to figure out the problem mentioned above for the discrete walking gaits of bipedal robots, the discrete-state-based vision method of using single camera is presented to obtain the position and direction information between bipedal robot and specific objective at the phase of path recognition. At the phase of path planning, one piecewise-linear method (PLM) is proposed to design the desired direction field and achieve navigation algorithm to control the bipedal robot walking on a plane surface. Lastly, combined with the improved method with band width, the robustness and efficiency of such proposed algorithm are studied and balanced by setting the proper control parameters.

This paper is organized as follows. Section 2 defines the navigation problem with the specific objective. Section 3 gives the background of Cornell Ranger and briefly introduces vision navigation algorithm. Sections 4 and 5 deal with the problems of path recognition and the design of direction field, respectively. Section 6 shows how the improved control method with band width can restrain the noise influence from the camera, and simulation results are offered and compared. Finally, conclusions are summarized in Section 7.

Notation. Objects in a manifold are denoted by unbold letters or symbols. Their local coordinate representations are denoted by bold letters and symbols. Thus, if $q \in \Omega$ and Ω is a n -dimensional configuration manifold, then $q = [q_1, q_2, \dots, q_n]^T \in \mathfrak{R}^n$ represents the local coordinates of q . In this paper, generally capital letters will be used to denote desired quantities and lowercase letters to denote actual quantities.

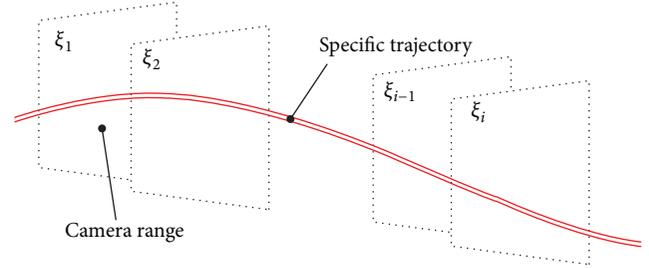


FIGURE 1: The sequence of discrete state spaces $\xi_1, \xi_2, \dots, \xi_{i-1}, \xi_i$ at different intervals along specific trajectory.

2. Problem Definition

2.1. Discrete State Space. Generally, for self-navigation robots, the vision system is one of the most important parts and the camera is used to capture the information about surrounding environment. Since the vision system is mounted on the robot, the robot intuitively is recognized as an observer always keeping stationary in relation to itself. The observer is always coupled with the concept of state space, and different observers are corresponding with different state spaces. The surrounding environment composes one new time state space $\xi(t)$ in relation to the observer (robot) at the instant of time t . All of the state spaces compose the state space set denoted by $S(n)$ in the whole time T . The state space set $S(n)$ can be defined by

$$S(n) := \{\xi(t) \mid \xi : \mathfrak{R} \rightarrow \mathfrak{R}^n, t \mapsto \xi(t), t \in [0, T]\}. \quad (1)$$

Here, n is the dimension of state space. ξ is a mapping from time space t to state space $\xi(t)$. The range of each image captured by the camera at different discrete intervals is only a subset of a state space. For simple consideration, although the picture information captured by the camera is only a subset of a state space, it is sufficient to define picture information as a discrete time state space after the range according to the picture image being thought of as extending infinitely, shown in Figure 1. The n -dimensional discrete state space set can be defined as the following:

$$\tilde{S}(n) := \{\tilde{\xi}_i \mid \tilde{\xi}_i \in \mathfrak{R}^n, i = 1, 2, \dots, N\}. \quad (2)$$

Here N is the number of elements $\tilde{S}(n)$ and also is equivalent to the number of images from camera. There exists a projective mapping between the state space affixed image and the state space with the observer of robot, which can be expressed by

$$\gamma_p : \tilde{S}(n) \rightarrow S(n); \quad \tilde{\xi}_i \mapsto \xi_i. \quad (3)$$

All the material treated here can be general and applicable to n dimensions. In this paper, we focus on one bipedal robot walking on the level ground with discrete walking gaits. All of environment information is embedded in the 2D level ground, so the dimension of state space is two-dimensional space and $n = 2$. The time state space set is very useful for analysis of path recognition in Section 4.

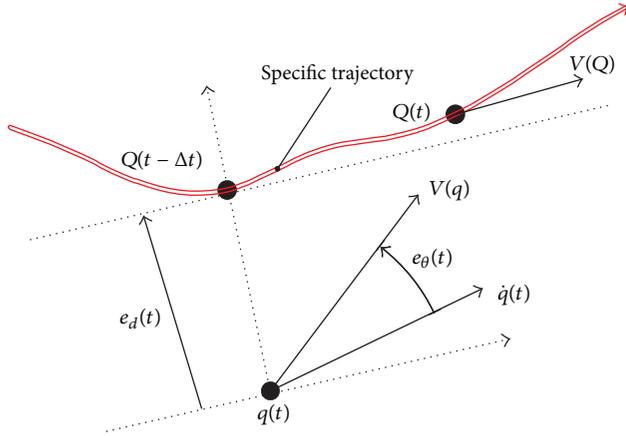


FIGURE 2: Illustration of navigation with the specific trajectory: the $Q(t)$ and $q(t)$ are the desired and actual positions at the time t ; the $V(q)$ and $\dot{q}(t)$ are the desired and actual velocity vectors; the e_d and e_θ are distance error and angle error.

2.2. Problem Description. The navigation problem with specific path that we are concerned about here can be considered as specifying desired timed trajectories (generally, desired timed trajectory denoted by $Q(t)$ is the best perfect path planning scheme with time t ; in this case, the robots spend the least time to accurately track/follow the specific objective, shown in Figure 2) in space; that is, the robots with navigational characteristics should possess the function of tracking to one specific point at the instant of time. Taking the state space as n -dimensional manifold denoted by \mathfrak{R}^n , the desired trajectory and actual trajectory are two curves in this manifold and called one-dimensional embedded submanifold of \mathfrak{R}^n , respectively. So both of them can be parameterized as a function of time t :

$$\begin{aligned} \gamma_Q: \mathfrak{R} &\rightarrow \Omega; & t &\mapsto Q(t), \\ \gamma_q: \mathfrak{R} &\rightarrow \Omega; & t &\mapsto q(t), \end{aligned} \quad (4)$$

where Ω is the n -dimensional configuration manifold in \mathfrak{R}^n . The perfect navigation is that the robot can come to the specific point at the specific time point, so the design goal of navigation algorithm is to make the tracking error $q(t) - Q(t)$ converge to 0 at all time points. Actually, the actual timing of tracking the specific objective is unimportant, and the most important aspect is that the robot should move along the specific objective with right direction. The actual position and the desired location could just be out of step by a time interval Δt ; that is, $q(t) - Q(t - \Delta t) = 0$. If the timing is not critical, there is no need to keep up with the desired timed trajectory. We define the distance error as

$$e_d(t) = \min_{\Delta t} \|q(t) - Q(t - \Delta t)\|_2. \quad (5)$$

On the one hand, a considerable navigation algorithm demands (1) the distance error $e_d(t)$ gets close to 0 as soon as possible, and (2) the navigation controller should try to minimize the time interval error Δt even though the time is

not critical quantity, while generally it requires the robot to accomplish the navigation task with the shortest time.

The forward velocity and direction are the two aspects to keep the robot moving along the specific trajectory. Let $T_q\Omega$ be the tangent space of Ω at the specific configuration q . A desired velocity vector field V is a map. Consider the following:

$$V: \Omega \rightarrow T\Omega; \quad q \mapsto V(q) \in T_q\Omega, \quad (6)$$

where $T\Omega = \cup_{q \in \Omega} T_q\Omega$ is the tangent bundle of Ω . Thus, the desired velocity field defines a tangent vector (the desired velocity) at every point of the configuration space. Roughly speaking, a velocity field would encode a specific trajectory if it points toward the trajectory and is tangent to it whenever it belongs to the specific trajectory. Given a specific desired velocity field, we define the α -related velocity field error as

$$e_\alpha(t) = \dot{q}(t) - \alpha V(q(t)), \quad (7)$$

where $\dot{q}(t) \in T_{q(t)}\Omega$ is the actual velocity of robot, and $\alpha > 0$ is a constant scalar. The goal of velocity field tracking control is to cause $e_\alpha(t) \rightarrow 0$ for some α . Notice that if $e_\alpha(t) \equiv 0$, the robot travels in the direction of $V(q(t))$, so that the ODE $\dot{q}(t) = \alpha V(q(t))$ is satisfied. The speed at which the robot follows the trajectory will be proportional to the constant α . That means once the direction of robot is the same as the direction of desired velocity field, the navigation algorithm would manipulate the robot following the specific trajectory. After defining the relative angle between desired velocity vector and the actual velocity vector as direction error denoted by e_θ , we can get that

$$\cos(e_\theta(t)) = \frac{\dot{q}(t) * V(q(t))}{(\|\dot{q}(t)\| * \|V(q(t))\|)}. \quad (8)$$

More conveniently, we take the directions of both the desired velocity vector and the actual velocity vector as $\Theta(t)$ and $\theta(t)$. So, the concept of direction field of the velocity vector is fabricated to simplify (7). The aim of the navigation algorithm should meet

$$\begin{aligned} e_d(t) &= \min_{\Delta t} \|q(t) - Q(t - \Delta t)\|_2 = 0, \\ e_\theta(t) &= \Theta(q) - \theta(q(t)) = 0. \end{aligned} \quad (9)$$

In this way, the robot can move along the specific objective without distance and direction offsets. The following are several remarks.

Remark 1. The distance error is also called minimum distance, shown in Figure 2, denoted by $e_d(t)$. We specify $e_d(t) > 0$ if the robot is on the left side of specific objective; inversely, $e_d(t) < 0$ if on the right side.

Remark 2. The direction error denoted by $e_\theta(t)$ is the difference between the actual direction and desired direction. The regulation that the counterclockwise of $e_\theta(t)$ is positive is specified in the Figure 2.

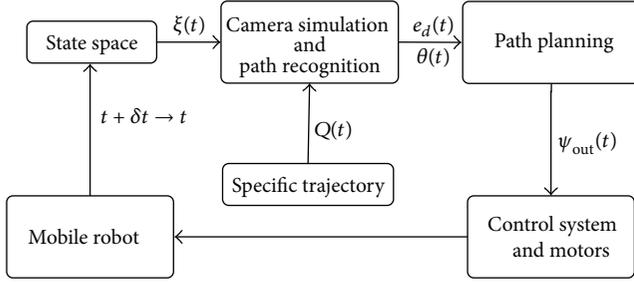


FIGURE 3: Process diagram of vision navigation.

2.3. Process Diagram. Figure 3 shows the process diagram of vision navigation. First of all, the robot as the observer constructs a state space $\xi(t)$ at the instant of time t after being given the initial position. Then, the camera takes the picture that contains the information about the surrounding environment and recognizes the specific objective according to the picture information. By the means of path recognition, two important parameters values of the distance error $e_d(t)$ and the relative angle $\theta(t)$ can be computed. After that, we design one path planning algorithm as a mapping function like

$$h : (e_d(t), \theta(t)) \mapsto \psi_{out}(t) \quad (10)$$

to control robot moving with the principles of (9). Here, $\psi_{out}(t)$ is the precise value of the steering angle. The actuator of navigation system manipulates the robot to head along the specific objective according to such steering commands. And then the robot moves to the new position and forms the new time state space $\xi(t + \delta t)$ at the instant of $(t + \delta t)$.

2.4. Efficiency and Robustness. The efficiency and robustness are the two aspects to evaluate the performance of the navigation algorithm. From the view of the efficiency, one better navigation algorithm should use less time to finish the navigation task. Here the total time of navigation task is called time efficiency. It is easy to see that the total steering angle is the main index of evaluating the efficiency. The less the total steering angle is, the less the energy is consumed and the higher the time efficiency is. Here the total angle of steering is denoted as the actuator effect. However, from the view of robustness, we should decrease the tracking error during the whole process. The tracking error, actuator effect, and time efficiency are three measurements to improve the efficiency and robustness, and the strict definitions are given as follows in order to evaluate the performance of navigation algorithm and study how to balance the robustness and efficiency by optimizing proper parameters.

Definition 3. Tracking error $\Gamma_{e_d(t)} = \int_0^T |e_d(t)| dt$ is the sum of the distance errors during the whole time T . That indicates the robot moves along with the specific objective with a smaller error at all time and the tracking error is small. In this situation, the robustness of such system is high.

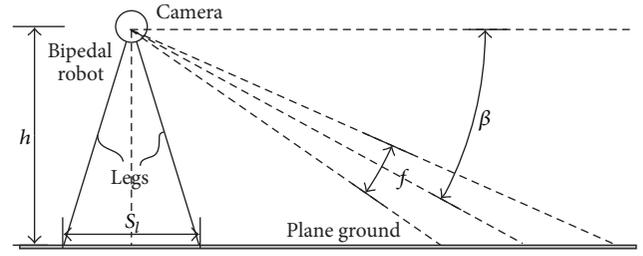


FIGURE 4: Illustration of principle of single camera vision navigation.

Definition 4. Actuator effect $\Gamma_{out} = \int_0^T |\psi_{out}(t)| dt$ is the sum of the steering angles during the whole time T . That indicates the smaller the steering angle is with which the robot tracks the same specific objective, the smaller the actuator effect is. In this situation, the consumed energy of actuator is smaller, so the efficiency of such system is higher.

Definition 5. Time efficiency $\Gamma_{\Delta t} = \int_0^T |\Delta t| dt$ is the sum of interval times. Usually, for the navigation system with the strict requirement of time, the navigation algorithm will be the best if the robot can finish tracking the specific trajectory with the shortest time.

In the following part, one discrete-state-based vision navigation algorithm using one camera is proposed, applied in one bipedal walking robot called Cornell Ranger. Since the biped robot has the characteristics of discrete gaits, the vibration resulting from discrete gaits results in difficulty to extract the useful information about specific objective according to the sequence of images. However, the navigation algorithm treated here can be expanded and applied in other wheeled mobile robots.

3. Application in One Bipedal Robot

3.1. Background of Bipedal Robot: Cornell Ranger [13]. Cornell Ranger is a kneeless 4-legged bipedal robot which is energetically and computationally autonomous. Ranger walked nonstop 65.2 km (40.5 miles) ultra-Marathon without recharging at Cornell University [13]. Unlike many bipedal robots, Cornell Ranger has four legs in two pairs. The outer pair moves together, acting as one leg, as does the inner pair. Each leg has an ankle joint and a foot but no knee joint. For each pair of legs the ankle joints are mechanically connected. There is a steering joint in the inner feet. The steering motor actuates this joint to adjust its heading direction during walking. At each step, it falls and catches itself under the control of trajectory generator and stabilizing controller. During walking, Cornell Ranger has its special characterizations as follows.

(1) *Step Length S_l .* The step length of Cornell Ranger is the distance between the adjoining foot-touching points shown in Figure 4. The value of step length S_l is constrained by the mechanism of the robot and measured by experiments. Although the experimental results show that the step length

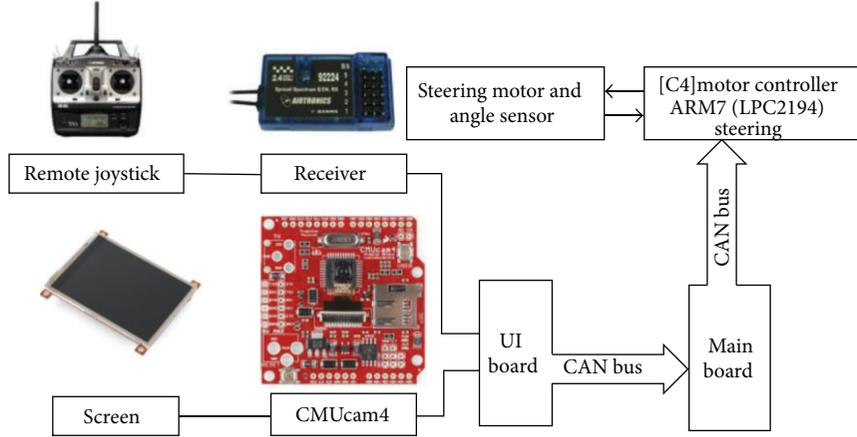


FIGURE 5: Navigation system of Cornell Ranger.

has a minor dependence on the condition of the substrate of ground, this influence can be ignored. Generally, this value can be thought of as constant. Thus we chose one relative large step length from several experimental results. The reason will be introduced in detail in Section 5.

(2) *Maximum Steering Angle ψ_{max}* . This value is the feasible steering angle moving forward in one direction. That means the robot can move forward with the direction range of $[-\psi_{max}, \psi_{max}]$. Negative value means the robot turns right while the positive value means it turns left relative to the robot. This parameter value also can be obtained by the experiment, while we choose the relative smaller value as the maximum steering angle. The reason also can be found in Section 5.

(3) *Step Number N* . Cornell Ranger is one highly efficiently passive walking robot. The mechanism could be thought of as one pendulum, and the natural walking frequency is almost determined by the mechanical structure. That means $S_l \cdot N = V \cdot T$ where the total step number $N = f_w \cdot T$, f_w is the walking frequency, and V is the mean velocity of walking. The vision navigation we treated in Section 2 is continuous for mobile robots like wheeled robots or aerial robots, while Cornell Ranger has one bipedal walking robot with discrete gaits. The number of total steps is one variable by which the walking distance can be calculated after considering the step length S_l as constant; thus the time T can be replaced by step number N and the instant time t by i .

We define the navigation algorithm in one function as

$$\psi_{out}(i+1) = F(e_d(i), \theta(i), S_l, \psi_{max}), \quad (11)$$

where $\psi_{out}(i+1)$ is the steering angle. $e_d(i)$ and $\theta(i)$ are the distance error and the actual heading direction. For this reason, the definitions in Section 2.4 should be reformulated as follows: tracking error $\Gamma_{e_d(i)} = \sum_{i=1}^N |e_d(i)|$; actuator effect $\Gamma_{out} = \sum_{i=1}^N |\psi_{out}(i)|$; time efficiency $\Gamma_{\Delta t}$ could be evaluated by the total number of steps N .

3.2. *Navigation Control of Cornell Ranger*. The navigation system of Cornell Ranger includes the radio remote control for human operation and the vision navigation based on single camera shown in Figure 5. Radio remote control is comprised of one remote joystick and one receiver. The receiver accepts the frequency signal carrying the control commands from transmitter and then transfers such signals into pulse width module (PWM) command. UI board receives the PWM signals from the receiver and sends processed signal to main board by CAN bus. Main board demodulates the signals into the corresponding commands. These commands include start of walking, stop of walking, human operation switch, and steering angle of inner feet. Vision navigation of Cornell Ranger is the main focus in this paper and is composed of one CMUcam4 module [22]. The CMUcam4 we used is a fully programmable embedded computer vision sensor. One attention we should be concerned about is that the time to obtain discrete state space should be at the same moment in every walking cycle. Here, we choose the instant of both two feet touching down the ground. That means each walking step has one discrete state space. Figure 6 is the flow chart of navigation control of Cornell Ranger. After obtaining the discrete state space, CMUcam4 can identify the specific trajectory with color by simple algorithm using a little power. Section 4 will introduce how CMUcam4 obtains the values of distance error $e_d(i)$ and the forwarding direction of robot $\theta(i)$. After that, the control navigation algorithm proposed in Section 5 calculates the steering angle $\psi_{out}(i+1)$. During the next step, Cornell Ranger adjusts the walking direction from discrete state space $\tilde{\xi}_i$ to $\tilde{\xi}_{(i+1)}$. By this way, the Cornell Ranger uses single camera to achieve autonomous vision navigation.

4. Path Recognition

In this section, some principles about the application and simulation of visual camera are introduced. Such principles are used to derive the relationship between the real coordinates and the pixel coordinates. The method based on discrete

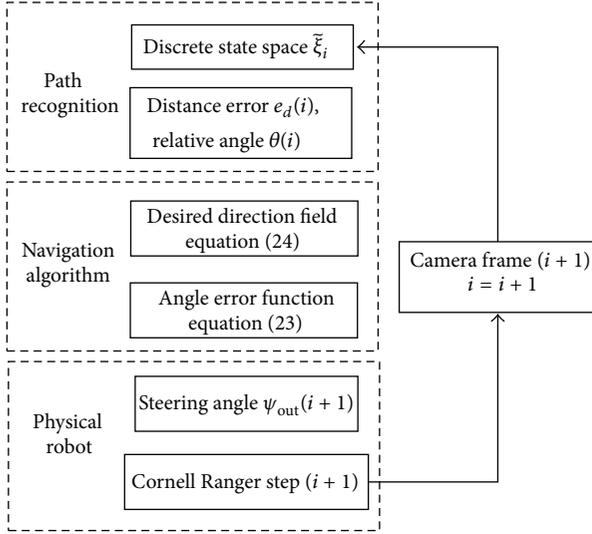


FIGURE 6: The flow chart of navigation control of Cornell Ranger.

state space is presented and applied to such bipedal robot to obtain two parameter values: distance error $e_d(i)$ and relative angle $\theta(i)$.

4.1. Camera Simulation. Usually the pin-hole model gives a way to compute the world coordinates from the pixel coordinates. Referring to [23], we can derive the following compact equation between the pixel coordinates and the world coordinates:

$$z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A \cdot K \cdot [R \ T] \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}, \quad (12)$$

where

$$A = \begin{bmatrix} f \cdot m_x & \gamma & u_0 \\ 0 & f \cdot m_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \quad K = \begin{bmatrix} 1 + k_1 r^2 + k_2 r^4 & 0 & 0 \\ 0 & 1 + k_1 r^2 + k_2 r^4 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (13)$$

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}, \quad T = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix},$$

and A is called intrinsic matrix which contains 6 intrinsic parameters. These parameters include focal length f , image format (m_x, m_y) , principal point (u_0, v_0) , and skew coefficients γ ; K is called the radial distortion matrix and contains

two parameters. R and T are the extrinsic parameters which denote the coordinate system transformations from 3D real world coordinates to 3D camera coordinates. Equivalently, the extrinsic parameters define the center position of camera and the heading direction of camera in world coordinates. T is not the position of the camera but the position of the origin of the world coordinate system expressed in coordinates of the camera-centered coordinate system. The position T of the camera expressed in world coordinates is $C = -R^{-1}T = -R^T T$ (since R is a rotation matrix). More details can be found in [23]. We need to know the intrinsic parameters of camera when we do vision computation. These intrinsic parameters are A and K in (12) that produced a given photograph or video. The process of finding those intrinsic parameters is called camera calibration. Camera calibration method can be found in [24]. After obtaining the intrinsic parameters, the extrinsic parameters R and T should be determined by the camera position and direction. Cornell Ranger uses a fully programmable embedded computer vision minicamera called CMUcam4 to recognize a specific trajectory. CMUcam4 is mounted on the middle of hip with one inclined angle β and height h relative to the level ground as shown in Figure 4. According to the definition of discrete state space, the projective position of the center of camera on level ground is the original position of each discrete state space $\xi(i)$. By this way, the world coordinate is constructed based on the discrete state space. And the extrinsic parameters R and T can be expressed by β and h as follows:

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} \cos \beta & \sin \beta & 0 \\ \sin \beta & -\cos \beta & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (14)$$

$$T = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ h \end{bmatrix}.$$

Assuming that the ground is level, any pixel $m(u, v)$ in pixel coordinates corresponds to one fixed point $P_w = (x_w, y_w, 0)$ in the world coordinates shown in Figure 7. In the camera simulation, we assume there are no radial distortion and image skew. According to the function (12), the relationship between pixel $m(u, v)$ and the world coordinate $P_w = (x_w, y_w, 0)$ can be expressed as

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \cdot (r_{31}x_w + r_{32}y_w + t_z) = \begin{bmatrix} f \cdot m_x & \gamma & u_0 \\ 0 & f \cdot m_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11}x_w + r_{12}y_w + t_x \\ r_{21}x_w + r_{22}y_w + t_y \\ r_{31}x_w + r_{32}y_w + t_z \end{bmatrix} \quad (15)$$

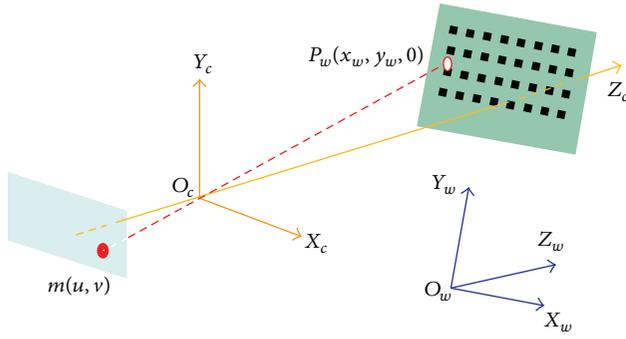


FIGURE 7: The relationship of pixel $m(u, v)$ in pixel coordinates corresponds to one fixed point $P_w = (x_w, y_w, 0)$ in the world coordinates.

by setting

$$M = \begin{bmatrix} r_{11} - \frac{u - u_0}{f} r_{31} & r_{12} - \frac{u - u_0}{f} r_{32} \\ r_{21} - \frac{v - v_0}{f} r_{31} & r_{22} - \frac{v - v_0}{f} r_{32} \end{bmatrix} \quad (16)$$

$$N = \begin{bmatrix} t_x - \frac{u - u_0}{f} t_z \\ t_y - \frac{v - v_0}{f} t_z \end{bmatrix}, \quad P'_w = \begin{bmatrix} x_w \\ y_w \end{bmatrix}.$$

The above equation can be simplified as

$$P'_w = M^{-1} N. \quad (17)$$

Using (15) and (17), the relationship between the real coordinates and the pixel coordinates is known. Traditionally, a series of images captured by the camera is used to determine the specific objective. However, a large amount of images might result in time consuming of picture processing, in order to improve the image processing efficiency and consider the influence from the noise of camera. The path recognition method based on discrete state space will be described in the following subsection.

4.2. Discrete-State-Based Path Recognition Method. Path recognition method based on discrete state space uses the data of current and previous images from the camera to obtain the parameters $e_d(i)$ and $\theta(i)$. The value of such two parameters is the input of the path planning algorithm proposed in Section 5. The image from the camera is the rectangular configuration shown in Figure 8(a), in which the center point is chosen as the original point. The mapping configuration is one ladder-shaped range corresponding to the rectangular image in Figure 8(b). By virtue of the technique of color following embedded in CMUcam4, the value of even distance between the specific objective and the middle line can be read directly using the color following programme. One attention we should be concerned about is that the time to obtain those images should be at the same moment in every walking cycle. In our scheme, we choose the instant of two feet touching down the ground. By this way, the

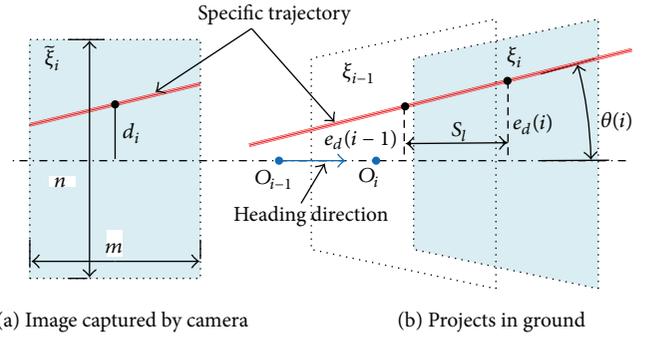


FIGURE 8: Illustration of image sequences path recognition method.

camera noise is caused by the horizontal and vertical motion of the center of mass. As shown in Figure 8(b), ξ_{i-1} and ξ_i are the two ordered sequences in $S^m(n)$, and the path recognition method based on discrete state space uses the data from two images: the previous ξ_{i-1} and the current ξ_i . The middle point position d_i in image is the only one known parameter in the image state space ξ_i . The real position in the state spaces ξ_{i-1} and ξ_i can be calculated by (17)

$$P_w(i-1) = \begin{bmatrix} x_w(i-1) \\ y_w(i-1) \end{bmatrix} = M^{-1}(u_{i-1}, v_{i-1}) \cdot N(u_{i-1}, v_{i-1}),$$

$$P_w(i) = \begin{bmatrix} x_w(i) \\ y_w(i) \end{bmatrix} = M^{-1}(u_i, v_i) \cdot N(u_i, v_i). \quad (18)$$

Here $u_{i-1} = u_i = 0$; $v_{i-1} = d_{i-1}$; $v_i = d_i$. We approximately take this middle distance as the minimal distance between robot and target path and simply take the relative angle $\theta(i)$ as one linear equation approximately as follows:

$$e_d(i) = y_w(i),$$

$$\theta(i) = \frac{y_w(i) - y_w(i-1)}{S_l}. \quad (19)$$

Here S_l is the step length.

5. Navigation Algorithm

The navigation algorithm is the control method of path planning. The values of variables $e_d(i)$ and $\theta(i)$ calculated in Section 4 are the inputs of navigation algorithm. The output value is the steering angle. As shown in Figure 9, the navigation algorithm is composed of two steps. The first step is to design a desired direction field. The desired direction $\Theta(i)$ at arbitrary position is known according to such designed direction field after giving the two walking parameters both the step length S_l and maximum steering angle ψ_{\max} . The second step is to calculate the angle error $e_\theta(i)$. The precise value of output steering angle can be calculated by comparing with the angle error $e_\theta(i)$.

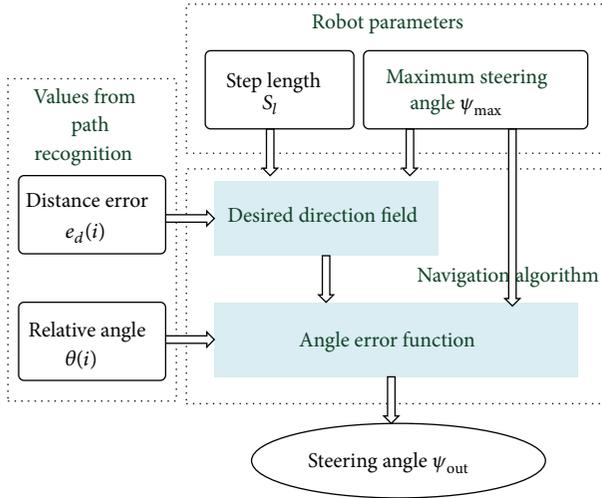


FIGURE 9: Navigation algorithm including two steps: (1) the design of the desired direction field and (2) the calculation of angle error.

5.1. Desired Direction Field Design. As discussed in Section 2, for any position $q(i)$ in the coordinates, the corresponding desired direction $\Theta(i)$ according to the defined desired direction field can be calculated. There are many feasible desired direction fields in the barrier-free space. In order to design one desired direction field with the best performance, some principles of designing the desired direction should meet the following assumptions or requirements: (1) tracking the specific objective without overshoot, which means the robot should move in one side of specific objective and neither cross that specific objective nor reach to the other side; (2) tracking the specific trajectory with shortest path, which means tracking of the specific objective will use the least number of steps with the highest time efficiency defined in Section 2.4. In order to meet such two requirements, the robot should move along one circle with the minimum turning radius $R_{\min} = (S_l/2)/\sin(\psi_{\max}/2)$. The shortest path method (SPM) with the minimum turning radius is called desired turning path related to S_l and ψ_{\max} . If this minimum turning radius is less than that the robots can handle, the overshoot will occur. That is the reason that we chose the maximum S_l and the minimum ψ_{\max} from the measured values to make sure of moving without overshoot in all cases. Once the step S_l and the maximum steering angle ψ_{\max} are constant, one suitable corresponding desired direction can be calculated by the minimum distance $e_d(i)$. This angle expressed as $\Theta(i)$ is called the desired direction and should meet the desired direction field function $\Theta(i) = \arccos((R_{\min} - e_d(i))/R_{\min})$ when $e_d(i) \leq R_{\min}$. Here $e_d(i)$ is the minimum distance. Considering the whole situation, the $e_d(i)$ can be divided into three regions A, B, and C shown in Figure 10. The region A can be named as the fine adjustment region, because the robot can get close to the target path precisely by two steps. In this region, the minimum distance should meet $e_d(i) < d_{\min}$. In the region C, the minimum distance meets

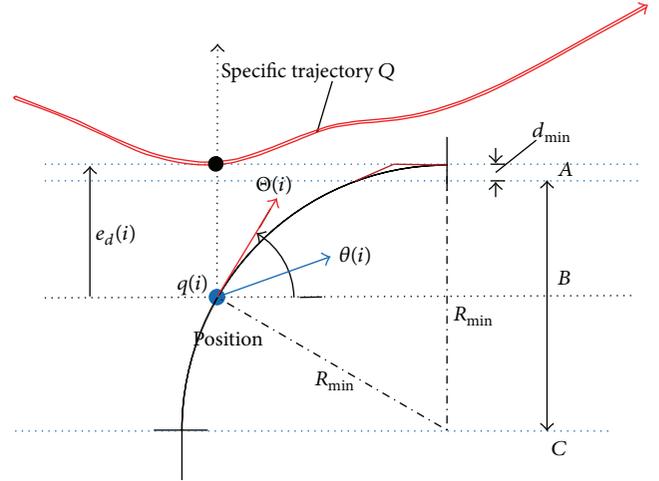


FIGURE 10: Illustration of the portions in full space: (1) region A if $e_d(i) < d_{\min}$, (2) region B if $d_{\min} < e_d(i) < R_{\min}$, and (3) region C if $e_d(i) > R_{\min}$.

the condition of $e_d(i) > R_{\min}$; the robot arrives at the specific trajectory with the vertical direction

$$\Theta(i) = \begin{cases} \frac{\pi}{2} & e_d(i) > R_{\min} \\ \arccos\left(\frac{(R_{\min} - e_d(i))}{R_{\min}}\right) & d_{\min} < e_d(i) < R_{\min} \\ \frac{d_w}{S_l} & 0 \leq e_d(i) < d_{\min} \end{cases} \quad (20)$$

But $\Theta(i)$ is the desired direction with the minimum distance of $e_d(i)$. The objective direction in the next position $q(i+1)$ can be predicted by the equation

$$\Theta(i+1) = \Theta(i) + \hat{\psi}_{\text{out}}(i+1). \quad (21)$$

$\Theta(i+1)$ is the objective direction in next position $q(i+1)$, and $\hat{\psi}_{\text{out}}(i+1)$ is the steering angle in the following step. Both of them are the unknown qualities, so the above function is an implicit function. The value of $\hat{\psi}_{\text{out}}(i+1)$ is 0 in the regions A and C, while it is ψ_{\max} in the region B.

5.2. Output Steering Angle. After the desired direction is obtained, the output of steering angle can be obtained by

$$\psi_{\text{out}}(i+1) = g(e_{\theta}(i)) = g(\Theta(i+1) - \theta(i)). \quad (22)$$

$e_{\theta}(i)$ is the direction error. $\psi_{\text{out}}(i+1)$ is the steering angle of $(i+1)$. Consider the following:

$$\psi_{\text{out}}(i+1) = \begin{cases} \psi_{\max} \cdot \min(k, 1) & \text{if } k > 0 \\ \psi_{\max} \cdot \max(k, -1) & \text{if } k \leq 0 \end{cases} \quad (23)$$

which is used as the motor commands to control robot direction. Here, $k = (\Theta(i+1) - \theta(i))/\psi_{\max}$. Function (23) indicates that the steering angle is the maximum steering

TABLE 1: The parameter value of simulation.

Symbols	Parameters	Value
h	Height of camera	1.1 m
β	Angle relative to camera	40°
S_l	Step length	0.7 m
ψ_{\max}	Maximum angle	4°

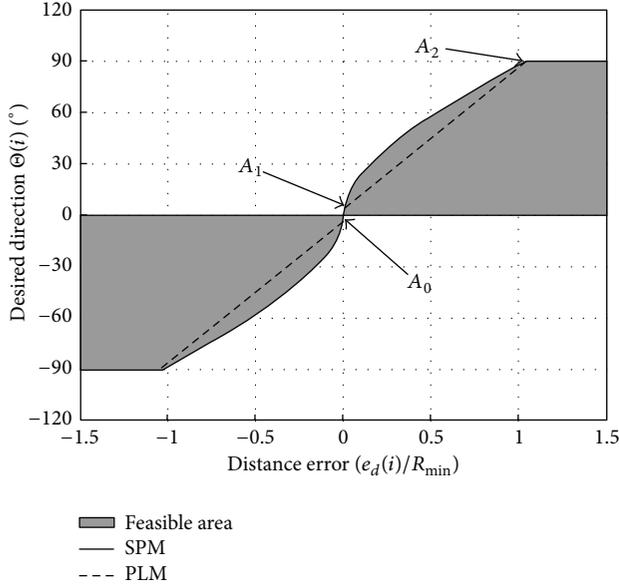


FIGURE 11: Illustration of desired direction field: (1) the solid line is the correspondent relationship between error distance ratio and desired direction of the shortest path method, (2) the dash line is the corresponding result using the PLF, and (3) the shadow area is the feasible area; that means arbitrary line in this area is the feasible direction field.

angle if the absolute value of $e_\theta(i)$ is much more greater than ψ_{\max} ; otherwise, it equals the value of $e_\theta(i)$.

Some parameter values are shown in Table 1. The solid curve in Figure 11 is the desired direction function of SPM using function (20). It can be seen that the relationship between $\Theta(i)$ and $e_d(i)$ is nonlinear as easily seen from Figure 11. Figure 12 shows the output steering angle function (23), which is one simple ramp function. Figure 13 gives the result of output steering angle under all different distance error and relative angle based on functions (20) and (23). Meanwhile, the vector quantization expression of the desired direction field in space is shown in Figure 14. Those verify that no matter where the initial position is, the robot always can get close to the specific trajectory along the desired direction field.

6. Improved Control Method

There are two problems of using the navigation control method proposed in Section 5: (1) the above SPM method solves the desired direction field according to function (20); $\Theta(i)$ and $e_d(i)$ have a nonlinear relationship; however,

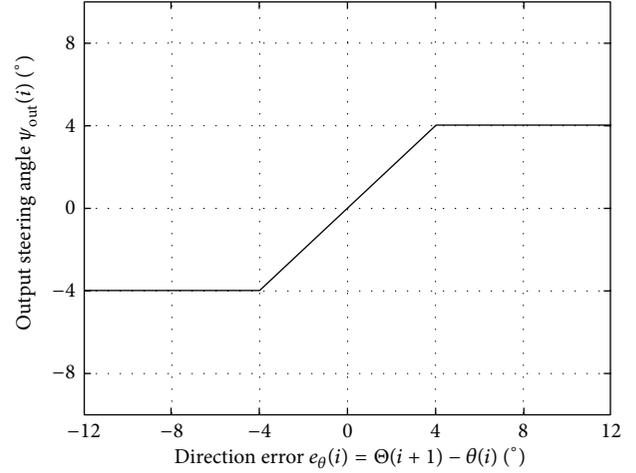


FIGURE 12: The relationship between the output steering angle $\psi_{\text{out}}(i)$ and direction error $e_\theta(i)$.

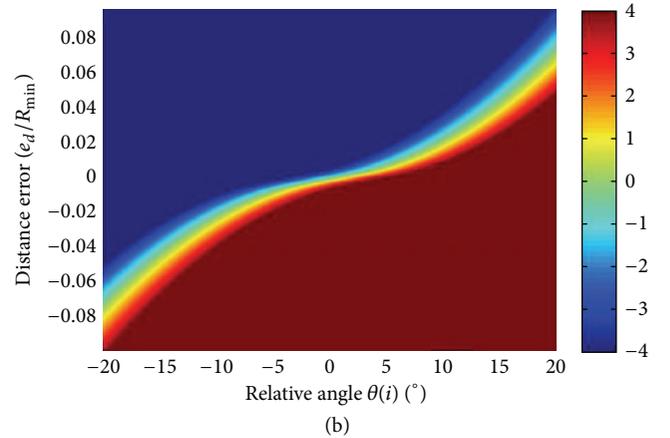
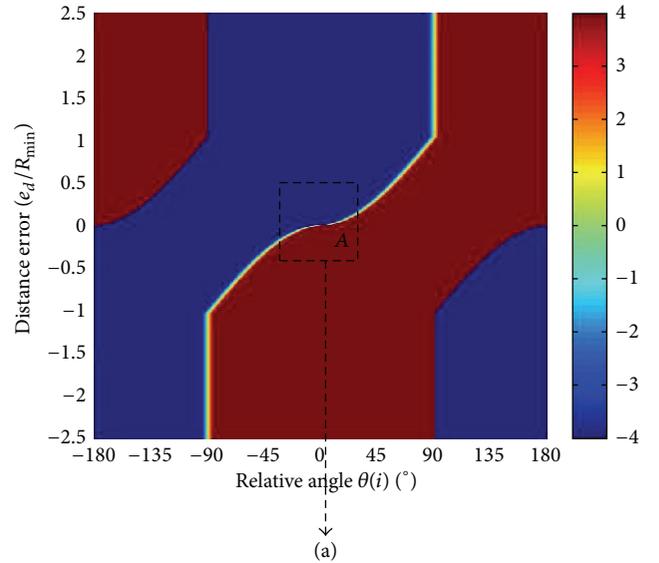


FIGURE 13: (a) The output of steering angle under different distance error and relative angle and (b) enlarged output steering angle in the small region A.

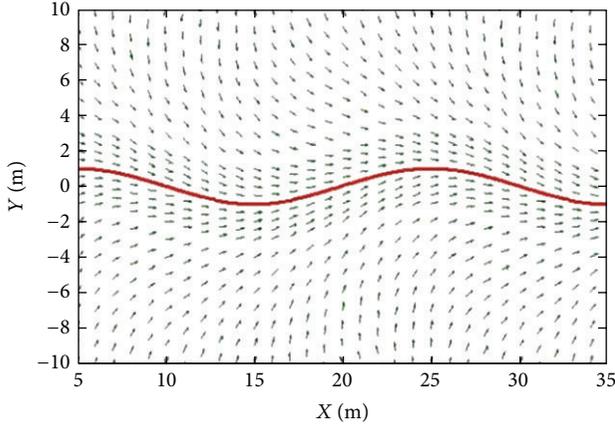


FIGURE 14: Illustration of the vector quantization expression of the desired direction field with the shortest path method in space. The red curve is the specific trajectory, and the arrowed line is the desired direction field.

the nonlinear computing demands much more time during the real control system and decreases the efficiency of control; (2) tracking no-width path precisely, especially for straight line path, would contribute to the severe fluctuation because of the error disturbance from the sensor noise and computation errors. These disturbances result in the uncertainties of $e_d(i)$ and $\theta(i)$.

6.1. Improved Control Method with Band Width. As can be seen from Figure 11, there is no overshoot if and only if the desired direction field function $\Theta(i)$ is in the shadow feasible area. Under this region, the arbitrary relationship has the corresponding desired direction field. A simplified linear method called piecewise-linear function (PLF) is constructed to substitute the nonlinear relationship when the three key points shown in Figure 11 are chosen. A_0 is the original point. $A_1(S_l \cdot \sin(\psi_{\max}), \psi_{\max})$ is the critical point between region A and region B, and $A_2(R_{\min}, \pi/2)$ is the transition position between regions B and C.

In order to consider the error from the camera sensor, we can assume that there exists a feasible width which provides the space of tracking path. When the robot does not run beyond this feasible region, there is no need to give steering control command to adjust direction. The width of such feasible path is called band width denoted by W herein. Combined with the PLF, function (20) can be modified as

$$\Theta(i) = \begin{cases} \frac{\pi}{2} & e_d(i) > R_{\min} \\ Ae_d(i) + B & d_{\min} < e_d(i) < R_{\min} \\ \frac{e_d(i)}{S_l} & e_d(i) < d_{\min} \\ 0 & 0 < e_d(i) < W \end{cases} \quad (24)$$

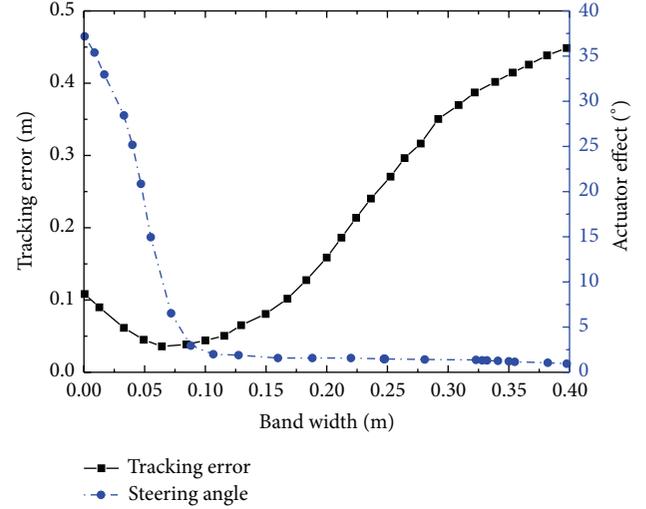


FIGURE 15: The influence of band width W on the tracking error and actuator effect.

Here,

$$A = \frac{(\pi/2 - \psi_{\max})}{(R_{\min} - S_l \cdot \psi_{\max})}, \quad (25)$$

$$B = \frac{(\pi/2 - \psi_{\max})}{(R_{\min} - S_l \cdot \psi_{\max})} \cdot S_l \cdot \psi_{\max} + \psi_{\max}.$$

$\Theta(i) = 0$ when $e_d(i) < W$ even though $e_d(i) < d_{\min}$. It can be seen that W is also one main parameter to determine the output of steering angle.

It is shown in Figure 15 that the actuator effect decreases with the increase of width band, while the tracking error reaches a minimal value at the neighbor of $W = 0.1$. The nice performance of both tracking error and actuator effect can be obtained at the band width $W = 0.1$ intuitively. Figure 16 compares the simulation results between the band width $W = 0.1$ and no band width. The output of steering angle of the method with band width is much smaller than that without band width if given the same sensor noise, which verifies that the control method with band width can restrain the disturbance and keep the navigation robust.

6.2. Optimization between Robustness and Efficiency. Using the definitions of tracking error $\Gamma_{e_d(t)} = \sum_{i=1}^N |e_d(i)|$ and actuator effect $\Gamma_{\text{out}} = \sum_{i=1}^N |\psi_{\text{out}}(i)|$ to evaluate the navigation performance, the optimal objective can be formulated by

$$\min f(W) = \alpha \Gamma_{e_d(t)} + (1 - \alpha) \Gamma_{\text{out}}, \quad (26)$$

where α is the weight of tracking error. Note that the unit of actuator effect is rad herein. Because of the nonsmooth characteristics of the optimal objective function, the simultaneous optimization method in MATLAB toolbox is used to deal with this optimal function.

Figure 17 shows the relationship of band width, tracking error, actuator effect, and objective function value with

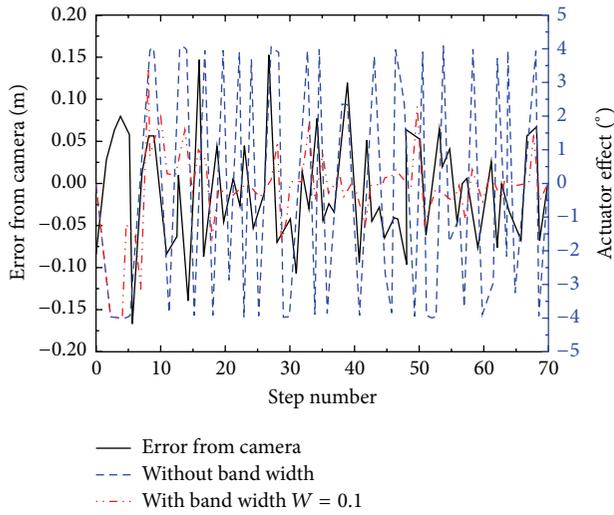


FIGURE 16: Comparison of the steering angle of tracking a straight line with 70 steps with sensor noises under two different control methods: (1) without band width method; (2) with band width control method $W = 0.1$.

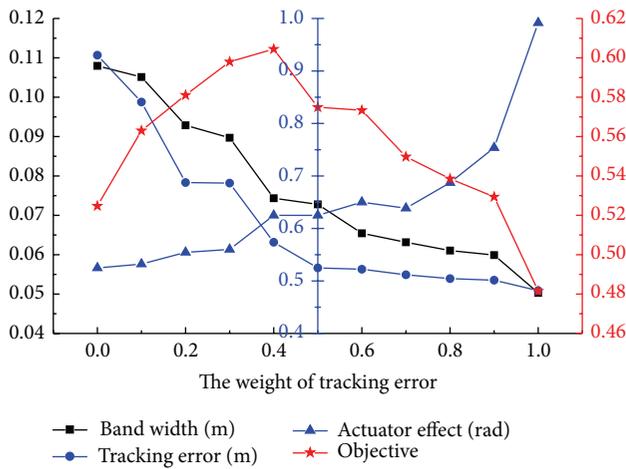


FIGURE 17: The relationship of band width, tracking error, actuator effect, and objective function value with the weight of the tracking error.

the weight of the tracking error, while the relationship between robustness and efficiency is described by Figure 18. From these results, it can be seen that the band width is one controllable parameter to balance robustness and efficiency, and robustness and efficiency are interrelated and interact with each other.

7. Conclusion

The vision navigation algorithm based on discrete state space is proposed and suitable for bipedal robots with single camera system. Considering the real situation, the improved control method with band width used to design desired direction field is presented to considerably restrain the noise disturbance from the camera sensor. The relationship between

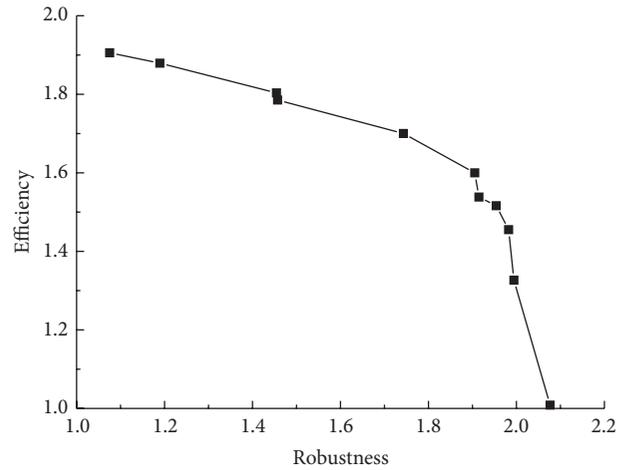


FIGURE 18: The relationship between robustness and efficiency.

robustness and efficiency is demonstrated by our simulation results. The results show the robustness and efficiency are interrelated and interacted with each other. Meanwhile, the band width value of the improved control algorithm depends on the error from the camera sensor. As for this, the band width is also one variable to balance the robustness and the efficiency of this system. This navigation algorithm can be easily expanded to other mobile robots with the heading velocity and the maximum steering velocity.

Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The author gratefully acknowledges the financial support from China Scholarship Council (CSC) for one-year study at the Cornell University. The author also thanks the Cornell Ranger team at the Cornell University.

References

- [1] S. Chen, Y. Li, and N. M. Kwok, "Active vision in robotic systems: a survey of recent developments," *The International Journal of Robotics Research*, vol. 30, no. 11, pp. 1343–1377, 2011.
- [2] G. N. DeSouza and A. C. Kak, "Vision for mobile robot navigation: a survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, pp. 237–267, 2002.
- [3] C.-L. Hwang and C.-Y. Shih, "A distributed active-vision network-space approach for the navigation of a car-like wheeled robot," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 3, pp. 846–855, 2009.
- [4] A. Das, O. Naroditsky, Z. Zhu, S. Samarasekera, and R. Kumar, "Robust visual path following for heterogeneous mobile platforms," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '10)*, pp. 2431–2437, 2010.
- [5] K. Konolige, M. Agrawal, R. C. Bolles, C. Cowan, M. Fischler, and B. Gerkey, "Outdoor mapping and navigation using stereo

- vision,” in *Experimental Robotics*, vol. 39 of *Springer Tracts in Advanced Robotics*, pp. 179–190, Springer, Berlin, Germany, 2008.
- [6] A. Chatterjee, N. N. Singh, O. Ray, A. Chatterjee, and A. Rakshit, “A two-camera-based vision system for image feature identification, feature tracking and distance measurement by a mobile robot,” *International Journal of Intelligent Defence Support Systems*, vol. 4, no. 4, pp. 351–367, 2011.
- [7] E. Menegatti, A. Pretto, A. Scarpa, and E. Pagello, “Omnidirectional vision scan matching for robot localization in dynamic environments,” *IEEE Transactions on Robotics*, vol. 22, no. 3, pp. 523–535, 2006.
- [8] E. Royer, M. Lhuillier, M. Dhome, and J.-M. Lavest, “Monocular vision for mobile robot localization and autonomous navigation,” *International Journal of Computer Vision*, vol. 74, no. 3, pp. 237–260, 2007.
- [9] Z. Chen and S. T. Birchfield, “Qualitative vision-based path following,” *IEEE Transactions on Robotics*, vol. 25, no. 3, pp. 749–754, 2009.
- [10] N. X. Dao, B.-J. You, and S.-R. Oh, “Visual navigation for indoor mobile robots using a single camera,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '05)*, pp. 1992–1997, IEEE, August 2005.
- [11] J. J. Guerrero and C. Sagüés, “Uncalibrated vision based on lines for robot navigation,” *Mechatronics*, vol. 11, no. 6, pp. 759–777, 2001.
- [12] C. Sagüés and J. J. Guerrero, “Visual correction for mobile robot homing,” *Robotics and Autonomous Systems*, vol. 50, no. 1, pp. 41–49, 2005.
- [13] A. Ruina et al., Cornell ranger 2011, 4-legged bipedal robot, 2014, http://ruina.tam.cornell.edu/research/topics/locomotion_and_robotics/ranger/Ranger2011/.
- [14] A. Mohammed and L. Wang, “Vision-based robotic path following,” *International Journal of Mechanisms and Robotic Systems*, vol. 1, no. 1, pp. 95–111, 2013.
- [15] P. D. Cristóforis, M. A. Nitsche, T. Krajník, and M. Mejail, “Real-time monocular image-based path detection,” *Journal of Real-Time Image Processing*, pp. 1–14, 2013.
- [16] A. S. Huang, D. Moore, M. Antone, E. Olson, and S. Teller, “Finding multiple lanes in urban road networks with vision and lidar,” *Autonomous Robots*, vol. 26, no. 2-3, pp. 103–122, 2009.
- [17] P. Sala, R. Sim, A. Shokoufandeh, and S. Dickinson, “Landmark selection for vision-based navigation,” *IEEE Transactions on Robotics*, vol. 22, no. 2, pp. 334–349, 2006.
- [18] N. Sudha and A. R. Mohan, “Hardware-efficient image-based robotic path planning in a dynamic environment and its FPGA implementation,” *IEEE Transactions on Industrial Electronics*, vol. 58, no. 5, pp. 1907–1920, 2011.
- [19] T. W. Manikas, K. Ashenayi, and R. L. Wainwright, “Genetic algorithms for autonomous robot navigation,” *IEEE Instrumentation & Measurement Magazine*, vol. 10, no. 6, pp. 26–31, 2007.
- [20] W. Gueaieb and M. S. Miah, “An intelligent mobile robot navigation technique using RFID technology,” *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 9, pp. 1908–1917, 2008.
- [21] H. Sert, A. Kökösy, and W. Perruquetti, “A single landmark based localization algorithm for non-holonomic mobile robots,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '11)*, pp. 293–298, IEEE, Shanghai, China, May 2011.
- [22] Cmucam4, 2014, <http://www.cmucam.org/projects/cmucam4>.
- [23] J. Heikkila and O. Silven, “A four-step camera calibration procedure with implicit image correction,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1106–1112, San Juan, Puerto Rico, June 1997.
- [24] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

