We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS BOOK CITATION INDEX INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

12

Moving Target Tracking of Omnidirectional Robot with Stereo Cameras

Jun Ming Kuang, Ming Liu and Xiang Lin
Department of Electrical and Computer Systems Engineering
Monash University, VIC 3800
Australia

1. Introduction

The omnidirectional mobile robots have attracted great attention in last twenty years. Their major advantage superior to the traditional car-like mobile robots, whose motion is subject to the nonholonomic constraints, lies on the feature that their linear and rotational motions can be simultaneously and independently carried out. A typical mechanical structure of omnidirectional robot utilizes three universal driving wheels which can be either driven or slid along any direction [2]. This special motion pattern significantly simplifies the path planning and motion control tasks.

For the motion control of omnidirectional mobile robots, the majority of work so far, however, only consider the kinematic model for motion control. This is equivalent to assuming that the robots are massless bodies and therefore can ideally respond to the input motion control commands, which indeed does not reflect the real situation especially for heavy and fast moving robots. Due to this reason, efforts have been made to develop precise dynamic model to improve robots' performance [1]-[3]. In recent years, the visual servoing which combines control application with vision systems has become a hot topic. However, only a few studies [12], [13] combined the vision system with omnidirectional robot which has excellent maneuverability.

In this study we integrate the motion control with stereo vision for target tracking. Unlike previous work [4], [9], in which control laws were based on image space using the inverse image Jacobian matrix for motion estimation, our algorithm is to directly estimate the relative position and velocity errors in Euclidean space. By using visual feedback, a simple PD controller is proposed. The PD control ensures the asymptotic stability of the tracking error if the target is still which is equivalent to the case of docking, or of the bounded-input-bounded-output (BIBO) stability if the target is moving with varying but bounded rotational and linear velocities.

The paper is organized as follows: Section 2 presents the robot motion equation and tracking task. The dynamic model of the overall system is given in Section 3. The vision based relative motion and tracking error estimation is introduced in Section 4. The PD control law is proposed in Section 5. As a practical justification, the experimental tracking results obtained are given in Section 6. Finally, conclusions and future work are presented in Section 7.

Source: Stereo Vision, Book edited by: Dr. Asim Bhatti,
ISBN 978-953-7619-22-0, pp. 372, November 2008, I-Tech, Vienna, Austria

2. Robot dynamics and tracking task definition

2.1 Dynamics of the robot

Considering an omnidirectional mobile robot [2], its configuration space is a smooth 3-manifold and can then be locally embedded in Euclidean space $\mathbf{R}^3$. The robot has three degrees of freedom, i.e. two dimensional linear motion and one dimensional rotational motion. There are three universal wheels mounted along the edge of the robot chassis 120° apart from each other, and each wheel has a set of rollers aligned with its rim, as shown in Figure 1. Because of its special mechanism, the robot is able to simultaneously rotate and translate. Therefore, the path planning can be significantly simplified by directly defining the tracking task with the orientation and position errors obtained by the visual feedback.

On the top of the robot, as indicated in Figure 2, a pair of cameras, C_1 and C_2, are mounted. Let $\mathcal{E} = \{O_e, x_e, y_e\}$ be the earth coordinate frame and $\mathcal{R} = \{O_r, x_r, y_r\}$ be the coordinate frame fixed on the robot body whose origin is located at the mass center of the robot. The orientation of the coordinate frame of the left camera, $\mathcal{C} = \{O_c, x_c, y_c\}$, is assumed to be consistent with that of the robot, i.e. the rotation matrix between those two frames is an identity, but a translation $\mathbf{t}_r^c$ between them exists, as illustrated in Figure 2.

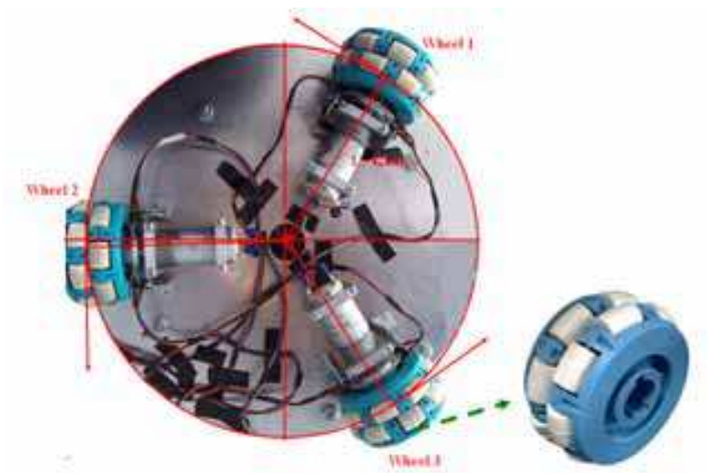

Fig. 1. The configuration of the wheel system

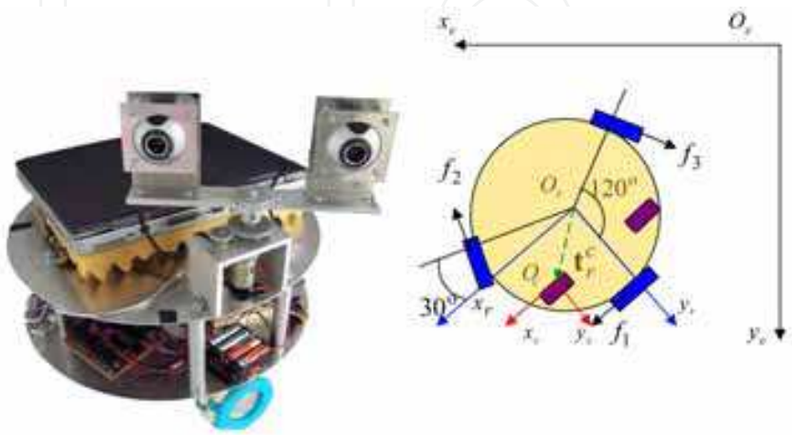

Fig. 2. Top view of omnidirectional mobile robot

Given this geometry layout, the configuration of the robot is given by $(\mathbf{R}_e^r(\theta_e^r), \mathbf{t}_e^r)$ in the earth frame, which can also be locally expressed as follows:

$$\mathbf{q}_r = \begin{bmatrix} \mathbf{t}_e^r \\ \theta_e^r \end{bmatrix} \in \mathcal{E} \tag{1}$$

where $\mathbf{t}_e^r = [x_e^r, y_e^r]^T$ are the position coordinates of the robot in the earth frame and θ_e^r is the relative rotation angle between the robot frame and the earth frame, $\mathcal{E}$.
As shown in [2], the dynamic model of the robot is given by:

$$\mathbf{M}\ddot{\mathbf{q}}_r = \mathbf{F} \tag{2}$$

where $\mathbf{M} = diag\{m, m, I\} > 0$ is the moment matrix in which $m > 0$ and $I > 0$ are the mass and inertia of the robot body, respectively; $\mathbf{F} = [F_x, F_y, \tau_\theta]_T$ is the force vector which includes the driving forces and the rotation torque applied along the directions of the axes, x_e and y_e, in the earth frame and about the mass center of the robot, respectively. Let $\bar{\mathbf{f}} = [f_x, f_y, f_\tau]^T$ be the forces expressed in the coordinate frame of the robot, $\mathbf{F}$ in (2) is revised as follows:

$$\mathbf{F} = \begin{bmatrix} \mathbf{R}_e^r(\theta_e^r) & 0 \\ 0 & 1 \end{bmatrix} \bar{\mathbf{f}} = \bar{\mathbf{R}}\bar{\mathbf{f}}$$

where

$$\mathbf{R}_e^r(\theta_e^r) = \begin{bmatrix} \cos(\theta_e^r) & -\sin(\theta_e^r) \\ \sin(\theta_e^r) & \cos(\theta_e^r) \end{bmatrix}$$

is the rotation matrix of the robot with respect to $\mathcal{E}$. Furthermore, let the forces generated from the three universal wheels be $\mathbf{f} = [f_1, f_2, f_3]^T$, and let the clockwise rotation of the robot viewed from the top of the robot be the positive direction as indicated in Figure 2, the following relation holds

$$\bar{\mathbf{f}} = \mathbf{T}_p\mathbf{f}$$

in which

$$\mathbf{T}_p = \begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & -\sqrt{3}/2 & \sqrt{3}/2 \\ r & r & r \end{bmatrix}$$

is the force projection matrix, r is the radius of the chassis, and $\sqrt{3}/2 = \cos 30°$ and $1/2 = \sin 30°$. Hence, robot model (2) becomes

$$\mathbf{M}\ddot{\mathbf{q}}_r = \bar{\mathbf{R}}\mathbf{T}_p\mathbf{f}. \tag{3}$$

In this model, the sliding friction of the wheels has been ignored. To handle it, a controller with varying control gains will be applied as those will be shown late.

2.2 Tracking task definition

To define the motion control task, Figure 2 is extended to Figure 3 that includes a moving target T to be tracked by the robot. $\mathcal{T} = \{O_t, x_t, y_t\}$ is denoted as the coordinate frame attached

to the target. Let $(\mathbf{R}_e^t(\theta_e^t), \mathbf{t}_e^t)$ be the configuration of the target in the earth frame, which can also be locally expressed by a 3D vector

$$\mathbf{q}_t = \begin{bmatrix} \mathbf{t}_e^t \\ \theta_e^t \end{bmatrix} \in \mathcal{E}. \tag{4}$$

In $\mathcal{T}$ there are n fixed vectors $\mathbf{p}_i = [p_{xi}, p_{yi}]^T$, for $i = 1, 2, \ldots, n$, indicating the locations of a group of feature points on the target as shown in Figure 3.

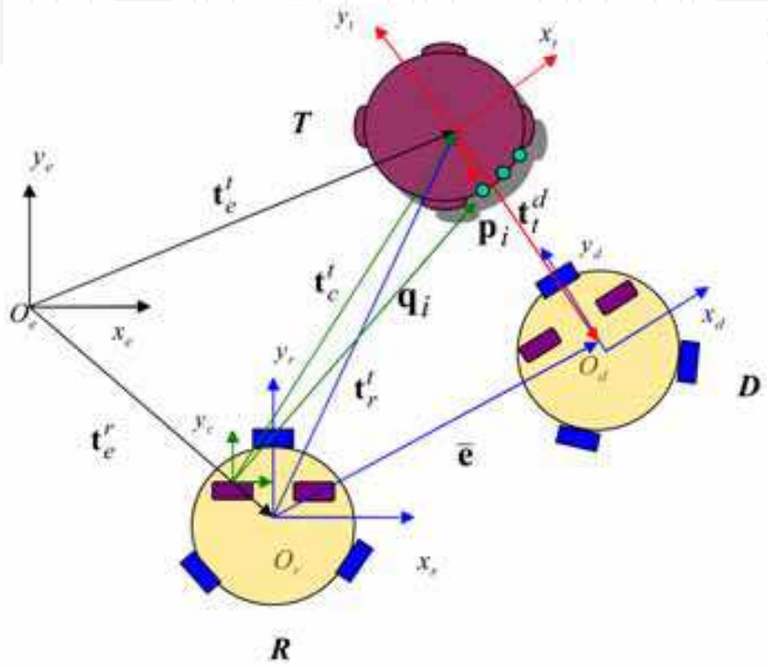

Fig. 3. Geometry layout of omnidirectional mobile robot, the target to be tracked and the destination configuration of the robot

A destination frame also called ideal configuration $\mathcal{D} = \{O_d, x_d, y_d\}$ is further specified and obtained by a constant translation $\mathbf{t}_t^d = [x_t^d, y_t^d]^T$ and a constant rotation angle θ_t^d from the target frame, as shown in Figure 3. The configuration of $\mathcal{D}$ expressed in $\mathcal{E}$ is then given by

$$\mathbf{q}_d = \begin{bmatrix} \mathbf{R}_e^t(\theta_e^t)\mathbf{t}_t^d + \mathbf{t}_e^t \\ \theta_t^d + \theta_e^t \end{bmatrix} \in \mathcal{E}. \tag{5}$$

The control objective is to control the omnidirectional robot, using the feedback from the vision system which grabs the images, recognizes the target and features, and supplies the estimated tracking errors, to track the target in such a way that the motion errors between the robot and configuration $\mathcal{D}$ are driven to be as small as possible.

3. Dynamic models of the overall system

Since the omnidirectional robot is driven by three DC motors, their dynamics are needed to be taken into account for the overall system modeling.

3.1 Motor dynamics and gear transmissions

Based on the mechanical and electrical characteristics of the motors, and denoting $\mathbf{J}$, $\mathbf{B}'$, $\mathbf{R}'_m$, $\mathbf{L}'$, $\mathbf{K}_m$ and $\mathbf{K}_e$ as the inertia moment matrix, the viscous damping matrix, the resistance matrix, the inductance matrix, the torque and the back EMF coefficient matrices, respectively, the model of three identical motors is given by:

$$\begin{cases} \mathbf{J}'\dot{\mathbf{W}}_m + \mathbf{B}'\mathbf{W}_m = \mathbf{K}_m\mathbf{I}_c - \mathbf{T}' \\ \mathbf{E} - \mathbf{K}_e\mathbf{W}_m = \mathbf{L}'\dot{\mathbf{I}}_c + \mathbf{R}'_m\mathbf{I}_c \end{cases} \tag{6}$$

where $\mathbf{J} = J\mathbf{I}$, $\mathbf{B}' = B\mathbf{I}$, $\mathbf{L}' = L\mathbf{I}$, $\mathbf{R}'_m = R_m\mathbf{I}$, $\mathbf{K}_m = k_m\mathbf{I}$ and $\mathbf{K}_e = k_e\mathbf{I}$ are positive definite matrices accordingly. $\mathbf{W}_m = [w_1, w_2, w_3]_T$ is the 3×1 vector of the angular velocities of the motors, $\mathbf{I}_c$ is the current vector of the motors, $\mathbf{T}'$ is the driven torque vector of the motor shafts and $\mathbf{E}$ is the input voltage vector.

Since the motors employed are quite small and their rotating speeds are rather slow (due to the slow motion of the target), their inductances $\mathbf{L}'$ and back EMF's, $\mathbf{E}_{emf}$ can be neglected. Thus, (6) can be simplified as:

$$\begin{cases} \mathbf{J}'\dot{\mathbf{W}}_m + \mathbf{B}'\mathbf{W}_m = \mathbf{K}_m\mathbf{I}_c - \mathbf{T}' \\ \mathbf{E} = \mathbf{R}'_m\mathbf{I}_c \end{cases} . \tag{7}$$

Motion mapping from the motors to robot is via gear boxes. The relation between the driven torque on the motor shaft side and that of a wheel shaft side, can be illustrated in Figure 4. The torque relation is given by $\tau'_i = f_i r_m = n f_i r_w = n\tau_i$ where τ'_i is the driven torque on a motor shaft, τ_i is that on wheel shaft and $n = r_m/r_w < 1$ is the gear ratio between the two shafts. As three motors have the same gear boxes, the relation can be expressed in the vector form:

$$\mathbf{T}' = n\mathbf{T} \tag{8}$$

where $\mathbf{T} = [\tau_1, \tau_2, \tau_3]^T > 0$ is the driving torque vector.

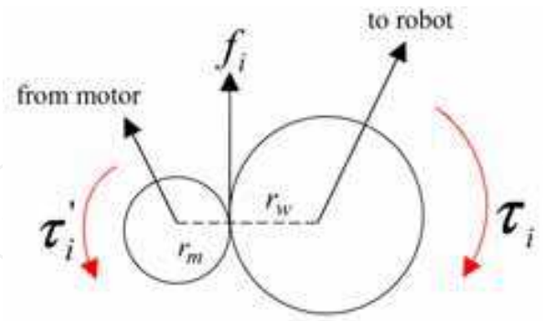

Fig. 4. Mechanism of gearing

Let v_{lx} and v_{ly} be robot's linear velocities along the x and y axes of its frame, w_r be its rotation angular velocity, and w_{m1}, w_{m2} and w_{m3} be the angular velocities of the three motors, the following relationships hold

$$\begin{cases} v_{lx} = nr'_w(w_{m1} - \frac{1}{2}w_{m2} - \frac{1}{2}w_{m3}) \\ v_{ly} = nr'_w(-\frac{\sqrt{3}}{2}w_{m2} + \frac{\sqrt{3}}{2}w_{m3}) \\ w_r = nr'_w(\frac{1}{3r}w_{m1} + \frac{1}{3r}w_{m2} + \frac{1}{3r}w_{m3}) \end{cases} \tag{9}$$

where r'_w is the radius of the wheel. (9) can be rewritten as

$$\mathbf{V} = nr'_w \mathbf{T}_v \mathbf{W}_m \tag{10}$$

where $\mathbf{T}_v$ is a constant mapping matrix given by:

$$\mathbf{T}_v = \begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & -\sqrt{3}/2 & \sqrt{3}/2 \\ 1/3r & 1/3r & 1/3r \end{bmatrix}.$$

Therefore the relationships between the linear and angular velocities of the robot and the angular velocities of the three motors are given by

$$\dot{\mathbf{q}}_r = \bar{\mathbf{R}}\mathbf{V} = nr'_w \bar{\mathbf{R}} \mathbf{T}_v \mathbf{W}_m, \tag{11}$$

or

$$\mathbf{W}_m = (nr'_w)^{-1} \mathbf{T}_v^{-1} \bar{\mathbf{R}}^{-1} \dot{\mathbf{q}}_r. \tag{12}$$

According to (12), motor motion equation (7) becomes:

$$\begin{cases} (nr'_w)^{-1} \mathbf{J}' \mathbf{T}_v^{-1} \bar{\mathbf{R}}^{-1} \ddot{\mathbf{q}}_r + (nr'_w)^{-1} \mathbf{T}_v^{-1} (\mathbf{J}' \dot{\bar{\mathbf{R}}}^{-1} + \mathbf{B}' \bar{\mathbf{R}}^{-1}) \dot{\mathbf{q}}_r = \mathbf{K}_m \mathbf{I}_c - \mathbf{T}' \\ \mathbf{E} = \mathbf{R}'_m \mathbf{I}_c \end{cases}. \tag{13}$$

3.2 The overall model

As $\mathbf{f} = n^{-1} \mathbf{R}_w^{-1} \mathbf{T}'$, (3) becomes:

$$\mathbf{M}\ddot{\mathbf{q}}_r = n^{-1} \bar{\mathbf{R}} \mathbf{T}_p \mathbf{R}_w^{-1} \mathbf{T}', \tag{14}$$

where $\mathbf{R}_w = diag\{r'_w, r'_w, r'_w\} > 0$ is the wheel radius matrix. By combining (13) and (14), the overall dynamic model including both motors and robot body then is given by

$$\begin{cases} (nr'_w)^{-1} \mathbf{J}' \mathbf{T}_v^{-1} \bar{\mathbf{R}}^{-1} \ddot{\mathbf{q}}_r + (nr'_w)^{-1} \mathbf{T}_v^{-1} (\mathbf{J}' \dot{\bar{\mathbf{R}}}^{-1} + \mathbf{B}' \bar{\mathbf{R}}^{-1}) \dot{\mathbf{q}}_r = \mathbf{K}_m \mathbf{I}_c - n\mathbf{M}\mathbf{R}_w \mathbf{T}_p^{-1} \bar{\mathbf{R}}^{-1} \ddot{\mathbf{q}}_r \\ \mathbf{I}_c = \mathbf{R}'^{-1}_m \mathbf{E} \end{cases}. \tag{15}$$

It can be rearranged as

$$\begin{cases} ((nr'_w)^{-1} \mathbf{J}' \mathbf{T}_v^{-1} + n\mathbf{M}\mathbf{R}_w \mathbf{T}_p^{-1}) \bar{\mathbf{R}}^{-1} \ddot{\mathbf{q}}_r + (nr'_w)^{-1} \mathbf{T}_v^{-1} (\mathbf{J}' \dot{\bar{\mathbf{R}}}^{-1} + \mathbf{B}' \bar{\mathbf{R}}^{-1}) \dot{\mathbf{q}}_r = \mathbf{K}_m \mathbf{I}_c \\ \mathbf{I}_c = \mathbf{R}'^{-1}_m \mathbf{E} \end{cases}. \tag{16}$$

Further let

$$\mathbf{A} = ((nr'_w)^{-1} \mathbf{J}' \mathbf{T}_v^{-1} + n\mathbf{M}\mathbf{R}_w \mathbf{T}_p^{-1}) \bar{\mathbf{R}}^{-1} \tag{17}$$

and

$$\mathbf{B} = (nr'_w)^{-1} \mathbf{T}_v^{-1} (\mathbf{J}' \dot{\bar{\mathbf{R}}}^{-1} + \mathbf{B}' \bar{\mathbf{R}}^{-1}), \tag{18}$$

the combined model equation, (16), can then be simplified as

$$\mathbf{A}\ddot{\mathbf{q}}_r + \mathbf{B}\dot{\mathbf{q}}_r = \mathbf{K}_m\mathbf{R}_m'^{-1}\mathbf{E}. \tag{19}$$

In real applications, motors $\mathbf{E}$ is generated by PWM (Pulse-Width Modulation) motor driver from a motion control microprocessor. As the relationships between the PWM signal and voltages $\mathbf{E}$ are almost linear, the $\mathbf{E}$ can be controlled by the PWM signals outputted from controller by a constant gain.

4. Relative motion estimation and tracking errors

In this section we present the estimation of relative position and velocity between the robot and target using machine vision. The outcome will be used for robot to carry out tracking task.

4.1 Rotation estimation

Since the moving target tracking is performed in 2-D space, The parameters needed to be estimated are rotation angle $\theta_e = \theta_e^t - \theta_e^r$, and 2-D translation vector t_c between the robot and target, as shown in Figure 5.

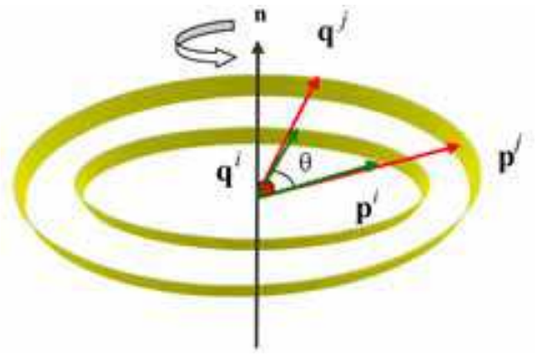

Fig. 5. The relationship between the feature vectors and the rotation angle

To solve θ_e, the feature points detected in the images are first recovered from the image coordinates back to two dimensional coordinates $\{\mathbf{q}_i\}$ expressed in the coordinate frame of the left camera by employing the reconstruction algorithm [14].

Expressed in the camera frame at the *k-th* step of the observation, the relative motion of the *i-th* feature point, $\mathbf{p}_i$, on the target is formulated by the following kinematic equation:

$$\mathbf{q}^i(k) = \mathbf{R}_c^t(\theta_e(k))\mathbf{p}^i + \mathbf{t}_c^t(i,k), \ (i = 1, 2, \cdots, n) \tag{20}$$

where $\mathbf{p}_i$ is the known coordinate of the *ith* feature point predefined in the coordinate frame of the target, $\mathbf{q}_i(k)$ is a two dimensional coordinate in the coordinate frame of the left camera, and $\mathbf{t}_c^t(k)$ is the translation between the origins of the target and the left camera viewed from $\mathcal{C}$. Considering another feature point, $\mathbf{p}_j$, and subtracting the projection equation of $\mathbf{p}_j$ from (20), the rotation is decoupled from the translation, as given by the following equation:

$$\mathbf{q}^i(k) - \mathbf{q}^j(k) = \mathbf{R}_c^t(\theta_e(k))(\mathbf{p}^i - \mathbf{p}^j), \quad i \neq j. \tag{21}$$

(21) indicates that the feature vector, ($\mathbf{p}_i$-$\mathbf{p}_j$), on the target becomes the one, ($\mathbf{q}_i(k)$-$\mathbf{q}_j(k)$), after the target rotates by angle $\theta_e(k)$ with respect to the left camera.

Given observation $\mathbf{q}_i(k)$ and $\mathbf{q}_j(k)$ and predefined feature point $\mathbf{p}_i$ and $\mathbf{p}_j$, the following relationship holds:

$$(\mathbf{p}^i - \mathbf{p}^j) \cdot (\mathbf{q}^i(k) - \mathbf{q}^j(k)) = |\mathbf{p}^i - \mathbf{p}^j||\mathbf{q}^i(k) - \mathbf{q}^j(k)| \cos\theta_e(k),$$

where $(\cdot)$ represents the inner product. Therefore rotation angle $\theta_e(k)$ can be resolved as:

$$\theta_e(k) = \arccos \frac{(\mathbf{p}^i - \mathbf{p}^j) \cdot (\mathbf{q}^i(k) - \mathbf{q}^j(k))}{|\mathbf{p}^i - \mathbf{p}^j||\mathbf{q}^i(k) - \mathbf{q}^j(k)|}.$$

As there are $s = n(n - 1) / 2$ such feature vectors. The optimal estimation of $\theta_e(k)$ can be obtained by using the following equation:

$$\theta_e(k) = \arccos \frac{1}{s} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{(\mathbf{p}^i - \mathbf{p}^j) \cdot (\mathbf{q}^i(k) - \mathbf{q}^j(k))}{|\mathbf{p}^i - \mathbf{p}^j||\mathbf{q}^i(k) - \mathbf{q}^j(k)|}. \tag{22}$$

In addition, the sign of the rotation angle is decided using the 2D polar frame.

4.2 Translation estimation

After rotation angle $\theta_e(k)$ is obtained, rotation matrix $\mathbf{R}_c^t(\theta_e(k))$ can be constructed as follows:

$$\mathbf{R}_c^t(\theta_e(k)) = \begin{bmatrix} \cos(\theta_e(k)) & -\sin(\theta_e(k)) \\ \sin(\theta_e(k)) & \cos(\theta_e(k)) \end{bmatrix}.$$

Consequently, the translation between the origins of the target and the left camera is calculated using the following equation:

$$\mathbf{t}_c^t(i,k) = \mathbf{q}^i(k) - \mathbf{R}_c^t(\theta_e(k))\mathbf{p}^i.$$

By defining the following estimation error:

$$E \stackrel{\text{def}}{=} \sum_{i=1}^{n} (\mathbf{t}_c^t(i,k) - \mathbf{t}_c^t(k))^2.$$

The optimal solution $\mathbf{t}_c^t(k)$ that minimizes E can be obtained as the solution of $dE/dt = 2\sum_{i=1}^{n}(\mathbf{t}_c^t(i,k) - \mathbf{t}_c^t(k)) = 0$ which is

$$\mathbf{t}_c^t(k) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{t}_c^t(i,k). \tag{23}$$

4.3 Relative velocity estimation

In addition to the relative orientation and position, the relative velocity between the robot and target is also required by the proposed controller. Consequently, the angle velocity is given by

$$w_e(k) = \frac{\theta_e(k) - \theta_e(k-1)}{T_i} \tag{24}$$

and the translation velocity is calculated by

$$\dot{\mathbf{t}}_k = \frac{\mathbf{t}(k) - \mathbf{t}(k-1)}{T_i} \tag{25}$$

where T_i is the sampling time.

4.4 Tracking error

The relative orientation, position and velocity between the left camera and the target have been calculated by (22)-(25). In addition, tracking error e_r and its derivative between the robot and ideal configuration $\mathcal{D}$ at which the robot is expected to arrive are also calculated in this subsection. For convenience, in the sequel, step index k in (22)-(25) is removed.

4.4.1 Relative tracking error in the frame of the robot

As the orientation difference between the target and configuration $\mathcal{D}$ is a given constant angle, θ_t^d, and the orientation of the left camera is the same as that of the robot, the relative angle between the robot and configuration $\mathcal{D}$ is

$$e_\theta = \theta_e + \theta_t^d$$

in which θ_e is given in (22). Therefore, given the predefined $\mathbf{t}_t^d = [x_t^d, y_t^d]^T$ which is the position coordinate of $\mathcal{D}$ in the coordinate frame of the target, the linear position error between the origins of the coordinate frames of the robot and $\mathcal{D}$ can be obtained as follows:

$$\bar{\mathbf{e}} = \begin{bmatrix} \bar{e}_x \\ \bar{e}_y \end{bmatrix} = \mathbf{R}_r^t(\theta_e)\mathbf{t}_t^d + \mathbf{t}_r^t. \tag{26}$$

As the robot frame has the same orientation as that of the left camera, $\mathbf{t}_r^t$ is obtained by the following equation:

$$\mathbf{t}_r^t = \mathbf{t}_c^t + \mathbf{t}_r^c$$

where $\mathbf{t}_c^t$ is given by (23) and $\mathbf{t}_r^c$ is a constant, as illustrated in Figure 2.
Therefore the 3D position error in the coordinate frame of the robot, i.e. $\mathbf{e}_r$, is given by

$$\mathbf{e}_r = \begin{bmatrix} \bar{e}_x \\ \bar{e}_y \\ e_\theta \end{bmatrix} = \begin{bmatrix} \mathbf{R}_r^t(\theta_e) \begin{bmatrix} x_t^d \\ y_t^d \end{bmatrix} + \begin{bmatrix} x_r^t \\ y_r^t \end{bmatrix} \\ \theta_e + \theta_t^d \end{bmatrix}. \tag{27}$$

Since configurations $\mathcal{T}$ and $\mathcal{D}$ have the same angular velocity, the velocity error between $\mathcal{D}$ and $\mathcal{R}$ is the same as that between the target and the robot. Taking the derivative of (26), it has that

$$\begin{aligned} \dot{\bar{\mathbf{e}}} &= \dot{\mathbf{R}}_r^t(\theta_e)\mathbf{t}_t^d + \dot{\mathbf{t}}_r^t \\ &= w_e \begin{bmatrix} -\sin\theta_e & -\cos\theta_e \\ \cos\theta_e & -\sin\theta_e \end{bmatrix} \begin{bmatrix} x_t^d \\ y_t^d \end{bmatrix} + \begin{bmatrix} \dot{x}_r^t \\ \dot{y}_r^t \end{bmatrix}. \end{aligned} \tag{28}$$

where w_e is the relative angular velocity between the target and the robot given in (24). Consequently, the corresponding 3D velocity tracking error, $\dot{\mathbf{e}}_r$, is given by

$$\dot{\mathbf{e}}_r = \begin{bmatrix} \dot{\bar{\mathbf{e}}} \\ \dot{e}_\theta \end{bmatrix} = \begin{bmatrix} \dot{\bar{\mathbf{e}}} \\ w_e \end{bmatrix} \tag{29}$$

where $\dot{\bar{\mathbf{e}}}$ is given in (28).

4.5 Relative tracking error in the earth frame

In last subsection, the relative tracking error between the robot and configuration $\mathcal{D}$ expressed in the frame of the robot has been presented. Indeed, those quantities can also be expressed in other frames such as the earth frame, $\mathcal{E}$. Recalling the notations of the robot and the target configuration given in (4), ideal configuration $\mathcal{D}$ can be written as a 3D vector in $\mathcal{E}$ as

$$\mathbf{q}_d = \begin{bmatrix} \mathbf{R}_e^t(\theta_e^t)\mathbf{t}_t^d + \mathbf{t}_e^t \\ \theta_e^t + \theta_t^d \end{bmatrix}, \tag{30}$$

and its velocity and acceleration are

$$\dot{\mathbf{q}}_d = \begin{bmatrix} \dot{\mathbf{R}}_e^t(\theta_e^t)\mathbf{t}_t^d + \dot{\mathbf{t}}_e^t \\ \dot{\theta}_e^t \end{bmatrix} \tag{31}$$

and

$$\ddot{\mathbf{q}}_d = \begin{bmatrix} \ddot{\mathbf{R}}_e^t(\theta_e^t)\mathbf{t}_t^d + \ddot{\mathbf{t}}_e^t \\ \ddot{\theta}_e^t \end{bmatrix}. \tag{32}$$

Given these the relative motion errors can be expressed in earth frame $\mathcal{E}$ as follows:

$$\mathbf{e}_e = \mathbf{q}_d - \mathbf{q}_r = \begin{bmatrix} \mathbf{R}_e^t(\theta_e^t)\mathbf{t}_t^d + \mathbf{t}_e^t - \mathbf{t}_e^r \\ \theta_e^t + \theta_t^d - \theta_e^r \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{e}}_e \\ e_\theta \end{bmatrix} \tag{33}$$

$$\dot{\mathbf{e}}_e = \dot{\mathbf{q}}_d - \dot{\mathbf{q}}_r = \begin{bmatrix} \dot{\mathbf{R}}_e^t(\theta_e^t)\mathbf{t}_t^d + \dot{\mathbf{t}}_e^t - \dot{\mathbf{t}}_e^r \\ \dot{\theta}_e^t - \dot{\theta}_e^r \end{bmatrix} = \begin{bmatrix} \dot{\bar{\mathbf{e}}}_e \\ \dot{e}_\theta \end{bmatrix}. \tag{34}$$

The relation between $\mathbf{e}_e$ in $\mathcal{E}$ and $\mathbf{e}_r$ in $\mathcal{R}$ is then given by

$$\mathbf{e}_e = \begin{bmatrix} \mathbf{R}_e^r\bar{\mathbf{e}} \\ e_\theta \end{bmatrix} = \bar{\mathbf{R}}\mathbf{e}_r \tag{35}$$

where

$$\bar{\mathbf{R}} = \begin{bmatrix} \mathbf{R}_e^r & 0 \\ 0 & 1 \end{bmatrix} \tag{36}$$

is a generalized 3×3 rotation matrix. The relative velocity error in $\mathcal{E}$ is:

$$\dot{\mathbf{e}}_e = \begin{bmatrix} \dot{\mathbf{R}}_e^r \bar{\mathbf{e}} + \mathbf{R}_e^r \dot{\bar{\mathbf{e}}} \\ \dot{e}_\theta \end{bmatrix} = \dot{\bar{\mathbf{R}}} \mathbf{e}_r + \bar{\mathbf{R}} \dot{\mathbf{e}}_r \tag{37}$$

which indicates the relation between the relative velocity errors in the two frames, i.e. the earth frame and the frame of the robot.

5. PD control

5.1 Control law design

To direct the robot to track the target, a motion controller is requested. For simplicity, a PD control law is designed for the tracking task. It will be shown that the proposed PD control can ensure the asymptotic stability of the tracking error if the target is still which is equivalent to the case of docking, or the bounded-input-bounded-output (BIBO) stability if the target is moving with varying but bounded rotational and linear velocities.

As **A** is obviously a nonsingular matrix given in (19), it can be rearranged to be:

$$\ddot{\mathbf{q}}_r + \mathbf{A}^{-1}\mathbf{B}\dot{\mathbf{q}}_r = \mathbf{A}^{-1}\mathbf{K}_m\mathbf{R}_m'^{-1}\mathbf{E}. \tag{38}$$

Changing the sign of (38) and then adding $\ddot{\mathbf{q}}_d + \mathbf{A}^{-1}\mathbf{B}\dot{\mathbf{q}}_d$ to both sides of it, it has that

$$\ddot{\mathbf{q}}_d - \ddot{\mathbf{q}}_r + \mathbf{A}^{-1}\mathbf{B}(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}_r) = \ddot{\mathbf{q}}_d + \mathbf{A}^{-1}\mathbf{B}\dot{\mathbf{q}}_d - \mathbf{A}^{-1}\mathbf{K}_m\mathbf{R}_m'^{-1}\mathbf{E}. \tag{39}$$

Recalling Equation (33), (39) can be revised as

$$\ddot{\mathbf{e}}_e + \mathbf{A}^{-1}\mathbf{B}\dot{\mathbf{e}}_e = \ddot{\mathbf{q}}_d + \mathbf{A}^{-1}\mathbf{B}\dot{\mathbf{q}}_d - \mathbf{A}^{-1}\mathbf{K}_m\mathbf{R}_m'^{-1}\mathbf{E} \tag{40}$$

which can be regarded as a second order linear system in terms of error states, $\dot{\mathbf{e}}_e$ and $\mathbf{e}_e$. We need to design a control law using **E** to make the tracking errors as small as possible as $t \to \infty$. Our target is to have (40) be the form of a typical second order linear system as:

$$\ddot{\mathbf{e}}_e + \mathbf{K}_v\dot{\mathbf{e}}_e + \mathbf{K}_p\mathbf{e}_e = \mathbf{u} \tag{41}$$

where $\mathbf{K}_v$ and $\mathbf{K}_p$ are ideal controller parameter matrices to be determined, and **u** is an equivalent input signal of the system. Ideally, if we can have $\mathbf{u} \equiv 0$, then $\dot{\mathbf{e}}_e, \mathbf{e}_e \to 0$ as $t \to \infty$, that is, the asymptotic stability can be achieved. However, due to the fact that $\ddot{\mathbf{q}}_d$ and $\dot{\mathbf{q}}_d$ are unmeasurable, **u** can not be zero unless $\ddot{\mathbf{q}}_d = \dot{\mathbf{q}}_d = 0$.

Let the constant parameter matrices, $\mathbf{K}_v$ and $\mathbf{K}_p$ be the diagonal matrices, i.e. $\mathbf{K}_v = diag\{k_{v1}, k_{v2}, k_{v3}\} > 0$ and $\mathbf{K}_p = diag\{k_{p1}, k_{p2}, k_{p3}\} > 0$ where k_{vi} and k_{pi} (i = 1, 2, 3) are given positive constants, i.e. the control gains. Thus, the model, (40), can be rewritten as:

$$\ddot{\mathbf{e}}_e + \mathbf{K}_v\dot{\mathbf{e}}_e + \mathbf{K}_p\mathbf{e}_e = \ddot{\mathbf{q}}_d + \mathbf{A}^{-1}\mathbf{B}\dot{\mathbf{q}}_d - \mathbf{A}^{-1}\mathbf{K}_m\mathbf{R}_m'^{-1}\mathbf{E} - \mathbf{A}^{-1}\mathbf{B}\dot{\mathbf{e}}_e + \mathbf{K}_v\dot{\mathbf{e}}_e + \mathbf{K}_p\mathbf{e}_e. \tag{42}$$

As $\ddot{\mathbf{q}}_d$ and $\dot{\mathbf{q}}_d$ on the right-hand side of (42) depend on the acceleration and the velocity of the target and are unmeasurable in practice, they are diffcult to be compensated, the control algorithm can only be applied to compensate other components, expressed as:

$$\mathbf{A}^{-1}\mathbf{K}_m\mathbf{R}_m'^{-1}\mathbf{E} = -\mathbf{A}^{-1}\mathbf{B}\dot{\mathbf{e}}_e + \mathbf{K}_v\dot{\mathbf{e}}_e + \mathbf{K}_p\mathbf{e}_e. \tag{43}$$

Therefore, the control law can be determined as:

$$\mathbf{E} = \mathbf{R}'_m \mathbf{K}_m^{-1}(\mathbf{A}(\mathbf{K}_v\dot{\mathbf{e}}_e + \mathbf{K}_p\mathbf{e}_e) - \mathbf{B}\dot{\mathbf{e}}_e). \tag{44}$$

Substituting matrix **A** in (17) and matrix **B** in (18) back to (44), and performing manipulation, control law (44) can be revised as:

$$\mathbf{E} = \mathbf{R}'_m \mathbf{K}_m^{-1}(\mathbf{C}\bar{\mathbf{R}}^{-1}(\mathbf{K}_v\dot{\mathbf{e}}_e + \mathbf{K}_p\mathbf{e}_e) - (nr'_w)^{-1}\mathbf{T}_v^{-1}(\mathbf{J}'\dot{\bar{\mathbf{R}}}^{-1} + \mathbf{B}'\bar{\mathbf{R}}^{-1})\dot{\mathbf{e}}_e) \tag{45}$$

where $\mathbf{C} = (nr'_w)^{-1}\mathbf{J}'\mathbf{T}_v^{-1} + n\mathbf{M}\mathbf{R}_w\mathbf{T}_p^{-1}$ is a constant matrix.

Recalling $\mathbf{e}_e = \bar{\mathbf{R}}\mathbf{e}_r$ in (35) and $\dot{\mathbf{e}}_e = \dot{\bar{\mathbf{R}}}\mathbf{e}_r + \bar{\mathbf{R}}\dot{\mathbf{e}}_r$ in (37) and applying $\dot{\bar{\mathbf{R}}}\mathbf{e}_r = \bar{\mathbf{R}}\bar{\mathbf{R}}^T\dot{\bar{\mathbf{R}}}\mathbf{e}_r = \bar{\mathbf{R}}\bar{\mathbf{S}}(w_r)\mathbf{e}_r$ to (45), finally it has that

$$\mathbf{E} = \mathbf{Q}(\mathbf{D}(\bar{\mathbf{S}}(w_r)\mathbf{e}_r + \dot{\mathbf{e}}_r) - \mathbf{U}(\mathbf{e}_r + \bar{\mathbf{S}}(w_r)^{-1}\dot{\mathbf{e}}_r) + \mathbf{C}\mathbf{K}_p\mathbf{e}_r) \tag{46}$$

where $\mathbf{Q} = \mathbf{R}'_m\mathbf{K}_m^{-1}$, $\mathbf{D} = \mathbf{C}\mathbf{K}_v - (nr'_w)^{-1}\mathbf{B}'\mathbf{T}_v^{-1}$ and $\mathbf{U} = (nr'_w)^{-1}\mathbf{J}'\mathbf{T}_v^{-1}$ are constant matrices and

$$\bar{\mathbf{S}}(w_r) = \begin{bmatrix} 0 & -w_r & 0 \\ w_r & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

is a skew matrix of the angular velocity of the robot.

Although the robot tracking is formulated in the earth frame as shown in (42), (46) is a controller based on the robot frame, in which $\mathbf{e}_r$ and $\dot{\mathbf{e}}_r$ are both expressed in the frame of the robot obtained by on-board vision system. w_r, the angular velocity of the robot, can be obtained by three on-board wheel encoders using (9).

5.2 Overall system stability analysis

Given designed control law (46) error model (42) can then be simplified as:

$$\ddot{\mathbf{e}}_e + \mathbf{K}_v\dot{\mathbf{e}}_e + \mathbf{K}_p\mathbf{e}_e = \ddot{\mathbf{q}}_d + \mathbf{A}^{-1}\mathbf{B}\dot{\mathbf{q}}_d. \tag{47}$$

To analyze the stability of the overall system given in (47), the similar procedure mentioned in [15] is followed. Denote $\mathbf{u} = \ddot{\mathbf{q}}_d + \mathbf{A}^{-1}\mathbf{B}\dot{\mathbf{q}}_d$, (47) can be rewritten as the typical second order linear system:

$$\ddot{\mathbf{e}}_e + \mathbf{K}_v\dot{\mathbf{e}}_e + \mathbf{K}_p\mathbf{e}_e = \mathbf{u} \tag{48}$$

where **u** is the input of the system. Certainly if $\ddot{\mathbf{q}}_d, \dot{\mathbf{q}}_d \to 0$, $\mathbf{e}_e \to 0$ and therefore $\mathbf{q}_r = \mathbf{q}_d$, i.e. the robot catches up with ideal configuration $\mathcal{D}$. This is clearly corresponding to the docking case of a still target.

However, in most cases, **u** usually is not zero as $\ddot{\mathbf{q}}_d$ and $\dot{\mathbf{q}}_d$ are not zero. Thus, it is necessary to analyze the stability of the system using BIBO stability. Since simplified error model (48) represents a linear second order decoupled differential equation with input **u**, the expressions will be developed for the L_∞ gain of the operations $H : \mathbf{u} \mapsto \mathbf{e}_e$ and $G : \mathbf{u} \mapsto \dot{\mathbf{e}}_e$.

Consider the L_∞ gain of the operator $H_i : u_i \rightarrow e_{ei}$, ($i$ = 1, 2, 3), where u_i denotes the i*th* component of the input vector $\mathbf{u}$ and the same definition is given to other vectors, and write the transfer function from input u_i to output e_{ei} as:

$$\frac{e_{ei}(s)}{u_i(s)} = h_i(s) = \frac{1}{s^2 + k_{vi}s + k_{pi}}. \tag{49}$$

As such, consider the L_∞ gain of the operator $G_i : u_i \rightarrow \dot{e}_{ei}$ and write the transfer function from input u_i to output $\dot{e}_{ei}$ as:

$$\frac{\dot{e}_{ei}(s)}{u_i(s)} = g_i(s) = \frac{s}{s^2 + k_{vi}s + k_{pi}}. \tag{50}$$

Then, the two transfer functions operators can be bounded as:

$$\begin{aligned} \|H_i\|_\infty &= \int_0^\infty |h_i(t)|dt, \\ \|G_i\|_\infty &= \int_0^\infty |g_i(t)|dt \end{aligned} \tag{51}$$

where $h_i(t)$ and $g_i(t)$ are the impulse response of the transfer functions, (49) and (50), respectively.
For simplicity the critical damping case where $k_{vi}^2 = 4k_{pi}$ is considered. Evaluating the responses (51) leads to the results:

$$\begin{aligned} \|H_i\|_\infty &= k_{pi}^{-1}, \\ \|G_i\|_\infty &= 4(ek_{vi})^{-1}. \end{aligned} \tag{52}$$

Then, bounds on the multi-input multi-output (MIMO) gains are given as:

$$\begin{aligned} \|H\|_\infty &= \mathbf{K}_p^{-1} = \alpha_1, \\ \|G\|_\infty &= 4(e\mathbf{K}_v)^{-1} = \alpha_2. \end{aligned} \tag{53}$$

Therefore, for zero initial condition, the results are:

$$\begin{aligned} \|\mathbf{e}_e\|_\infty &\leqq \alpha_1 \|\mathbf{u}\|_\infty, \\ \|\dot{\mathbf{e}}_e\|_\infty &\leqq \alpha_2 \|\mathbf{u}\|_\infty. \end{aligned} \tag{54}$$

From (54), if $\mathbf{u} \in L_\infty$, it is clear that $\mathbf{e}_e, \dot{\mathbf{e}}_e \in L_\infty$. Then for $\mathbf{u} = \ddot{\mathbf{q}}_d + \mathbf{A}^{-1}\mathbf{B}\dot{\mathbf{q}}_d$, if assume that a smooth, bounded desired trajectory is specified so that $\mathbf{q}_d$, $\dot{\mathbf{q}}_d$ and $\ddot{\mathbf{q}}_d$ are elements of L_∞, it is clear that:

$$\|\mathbf{u}\|_\infty \leqq \|\ddot{\mathbf{q}}_d\|_\infty + \beta_1 \|\dot{\mathbf{q}}_d\|_\infty \tag{55}$$

where $\beta_1 = \|\mathbf{A}^{-1}\mathbf{B}\|_\infty$. As the matrices, $\mathbf{A}$ and $\mathbf{B}$, only involve the constant matrices and the orthogonal matrix, $\bar{\mathbf{R}}$ whose norm is 1, the following results can be obtained:

$$\begin{aligned} \|\mathbf{e}_e\|_\infty &\leqq \alpha_1(\|\ddot{\mathbf{q}}_d\|_\infty + \beta_1 \|\dot{\mathbf{q}}_d\|_\infty), \\ \|\dot{\mathbf{e}}_e\|_\infty &\leqq \alpha_2(\|\ddot{\mathbf{q}}_d\|_\infty + \beta_1 \|\dot{\mathbf{q}}_d\|_\infty). \end{aligned} \tag{56}$$

The results given in (56) indicate that if $\dot{\mathbf{q}}_d$ and $\ddot{\mathbf{q}}_d \in L_\infty$, $\mathbf{e}_e$ and $\dot{\mathbf{e}}_e \in L_\infty$. That is, bounded input ($\dot{\mathbf{q}}_d$ and $\ddot{\mathbf{q}}_d$) results in bounded output ($\mathbf{e}_e$ and $\dot{\mathbf{e}}_e$). From these results, it is certain that the system satisfies the requirement of BIBO stability and the control law designed is theoretically effective.

6. Experimental verification

6.1 Experimental setup

The omnidirectional robot built up for tracking experiment mainly includes three software modules, namely, the image processing module, the pose estimation module and the control module. The first two are implemented in Microsoft Visual C++ 6.0 which operates on a laptop with a CPU of Intel 7250 at 1.60GHz and system memory of 512 MB, while the control module is implemented in MPLAB v7.01 and burned into microprocessor PIC18F452.

The communications between the laptop and the PIC is via a USB-based interface. The functions of the interface are to perform the signal conversion between the USB signals and three separate serial signals, to manage the interrupts and to successfully maintain the communications. Figure 6 shows the configuration of the real-time platform on which the image processing module and the pose estimation module are run.

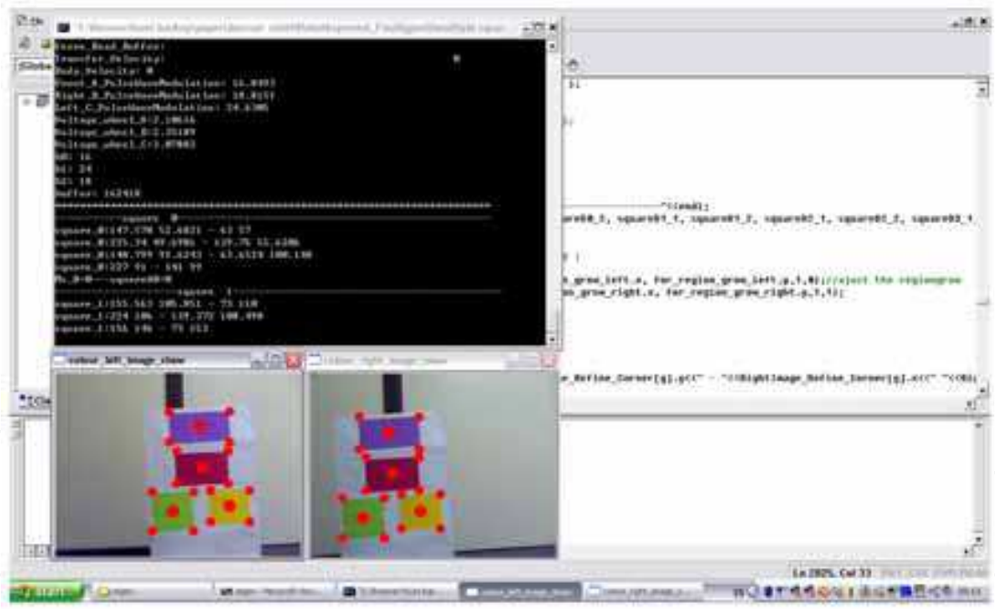

Fig. 6. The real-time platform of the experiment

The hardware of the robot can be roughly categorized into the mechanical module and the electronic module. The mechanical module includes the body of the robot which is constructed by two armor plates which have a radius of 0.21 meter, a pair of web-cameras, Logitech Pro5000, fixed on the top of the robot, and three universal wheels mounted along the edge of the robot chassis, 120° apart from each other and driven by three 12V DC geared motors. In addition, the speeds of the three wheels are measured by quadrature wheel encoders. Each encoder consists of a quadrature encoder pattern segment and a photo interrupter, ZD1901, as presented in Figure 7.

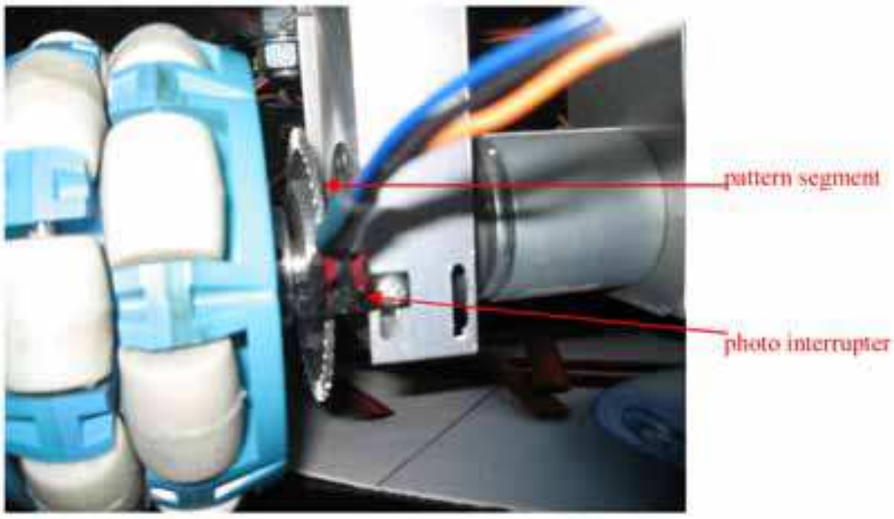

Fig. 7. The wheel encoder

The electronic module mainly involves three motor speed controllers NMIH-0050 Full H-Bridge which drive the motors to run in bidirection and vary their speeds by adjusting the PWM signals, and the PCB module which integrates all the electronic components utilized in the robot.

6.2 Image processing for target recognition

The image processing is employed for target recognition after stereo images are captured in real time. In order to simplify the image processing, a planar pattern which involves four rectangles in different colors is proposed in the tracking task, as shown in Figure 8. Prior to other parts of image processing, the captured images are first undistorted based on the lens distortion coefficients obtained by the camera calibration. Furthermore, as there are a number of existing irrelevant objects in the experimental environment, the target interested is extracted by segmentation approach. A segmentation scheme based on HSI color space (Hue, Saturation and Intensity) is applied to decouple the grey level component from the ones carrying the color information. Therefore, the effect of the light intensity is reduced, which results in a more robust segmentation.

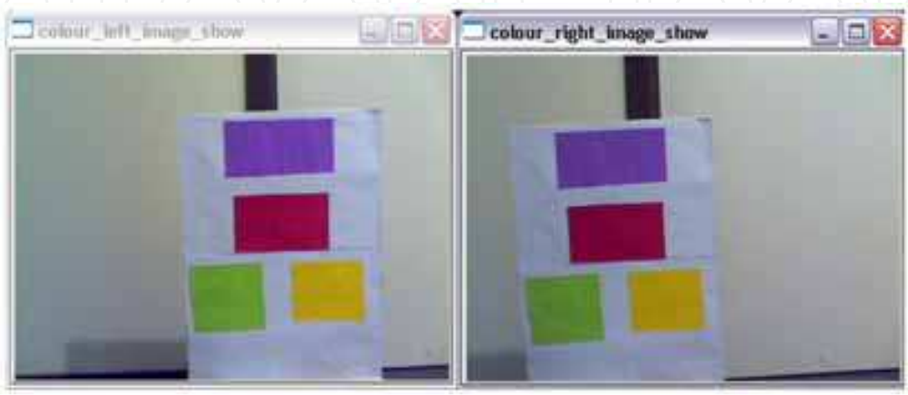

Fig. 8. The proposed target for tracking

To extract the regions of interest which include four rectangles respectively, Region Growing Segmentation, which groups pixels or subregions into larger regions with predefined criteria, is employed based on the color information of the rectangles.
Color information alone is not robust enough to recognize the target, as some objects which exclude the target may have similar colors. In order to reject the irrelevant objects which have the same colors as the target, shape information of the target is also introduced. As shown in Figure 8, the target includes four blocks which are rectangular. Based on this information, the objects which are not rectangular are rejected. By means of edge detector, the edges of the remaining objects can be directly obtained.
The edges of the four rectangles are straight lines. Therefore, the Hough transform is applied to the detected edges in order to find out the straight lines, and the contour-based shape representation approach is employed to describe the shapes of the objects constructed by the detected straight lines. Consequently, the objects whose shapes are roughly rectangular can be regarded as the targets and accepted, otherwise they are rejected.
After the four rectangles are recognized, a corner detector is applied to the interested regions for the detection of the feature points. The detected feature points in images will be recovered back to Euclidean space by using the optimal triangulation algorithm for the estimation of the relative orientation and position between the target and the robot. The Harris corner detector in OpenCV (Open Source Computer Vision Library)[16] is adopted as our corner detector to detect the corners of the four rectangles. To enhance the accuracy of the corner detection, the so called sub-pixel refinement for the corner detection is employed which iterates to find the sub-pixel accurate locations of the corners.
The complete procedure of the proposed image processing for the target detection and recognition is summarized and illustrated in Figure 9.

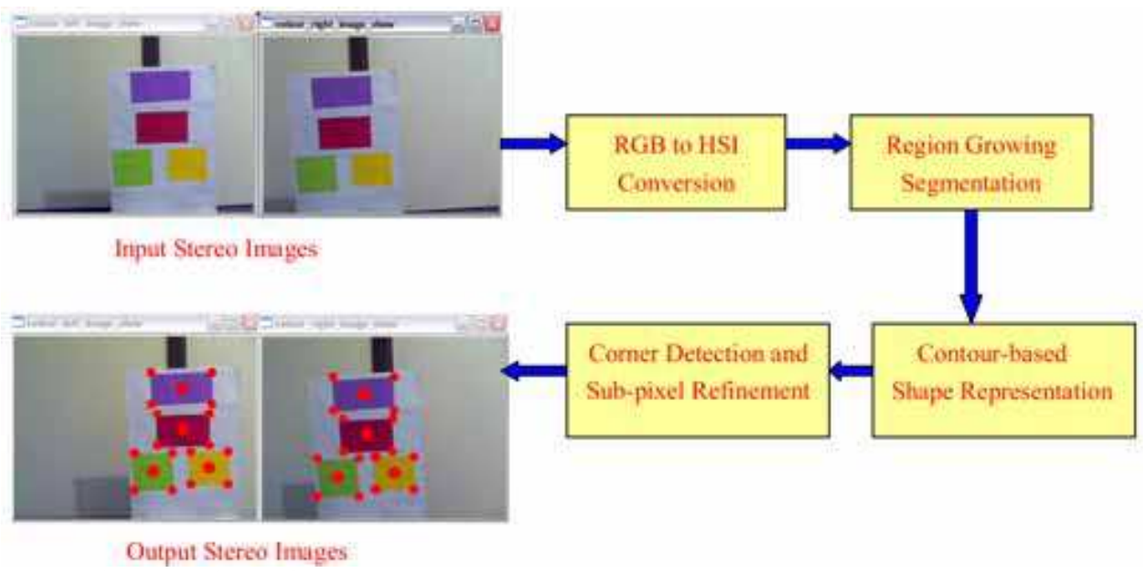

Fig. 9. The procedure of image processing

6.3 Experimental results and analysis

The experiment for the target docking and tracking is performed with the omnidirectional robot, and the experimental results are obtained by testing three docking scenarios and one tracking scenario.

The experiment is undertaken in a laboratory with a complex background. Many irrelevant objects are available in the laboratory. Natural light is used instead of using stable lighting. The planar object given in Figure 8 is employed as the target to be tracked. To record the trajectories of the robot for the visualization of the experiment, a colored chalk is fixed in the middle of the chassis during the experiment.

In addition, the relative rotation angle also called the relative orientation error, e_θ, and the position errors, $\bar{e}_x$ and $\bar{e}_y$, between the robot and ideal configuration $\mathcal{D}$ are calculated in real time and then saved into a separate file. Therefore, the change of the relative orientation error and position errors between the robot and configuration $\mathcal{D}$ during the experiment can be visualized by plotting the data from the file obtained.

The total time of one cycle through the complete control loop is about 0.05 second. The weight of the robot is around 10 *Kg* and its moment of inertia is about 3 Kgm^2. In addition, the relative rotation angle, θ_t^d, between the ideal configuration and the target is zero, and the linear position error between them is $\mathbf{t}_t^d = [0,-0.75]^T m$.

To avoid large overshoots or steady errors and reduce the effect of the un-modeled friction, the control gains, $\mathbf{K}_p$ and $\mathbf{K}_v$, are not fixed, that is, the control gains vary according to the relative orientation error and position errors between the robot and configuration $\mathcal{D}$. For instance, the large relative orientation error and position errors can produce large driven forces and therefore the small control gains are needed in order to avoid the large overshoot of the response or vice versa so as to reduce the steady errors of the tracking.

6.3.1 Docking scenarios

In docking scenarios, the target is still while the robot is controlled to move towards ideal configuration $\mathcal{D}$. As mentioned in the stability analysis, if the target is still, the tracking errors should satisfy the asymptotic stability, that is, the robot can finally dock itself to ideal configuration $\mathcal{D}$.

In the first docking scenario, as shown in Figure 10, the target is represented by the green star. The desired docking location, i.e. ideal configuration $\mathcal{D}$, is represented by a blue square and located at coordinate (225, 182) in the earth frame while the initial position of the robot is located at (169, 78) represented by a black circle, and the initial relative rotation angle between the robot and $\mathcal{D}$ is around zero. By observing the trajectory of the robot represented by the red line, it is clear that the position error, $\bar{\mathbf{e}}$, is driven to decrease while the movements of the robot are adjusted by the control signals. However, the position error, $\bar{e}_x$, increases while the robot goes through the range where $y = \{y_e \mid 100 \le y_e \le 120\}$. The reason is that the movements of the robot are affected by the friction between the wheels and the ground. Finally, the robot stops at location (221, 180) represented by the red circle and fairly close to ideal configuration $\mathcal{D}$ at (225, 182). The total experimental time is about 2.25 seconds.

For the second docking scenario, the result is illustrated in Figure 11. From the figure, it can be seen that ideal configuration $\mathcal{D}$ is at coordinate (184, 187) and the initial position of the robot is located at (275, 70). The initial relative rotation angle between the robot and ideal configuration $\mathcal{D}$ is -35°, that is, the robot needs to clockwise rotate 35° in order to coincide with the orientation of ideal configuration $\mathcal{D}$, as shown in Figure 12.

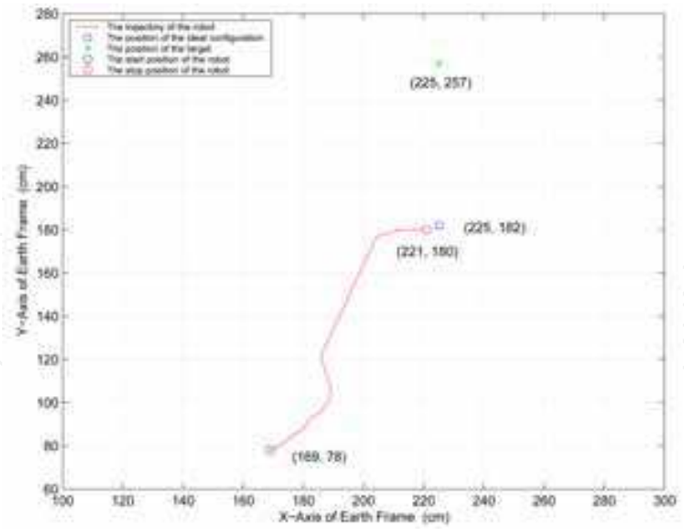

Fig. 10. The docking trajectory of the robot in Docking Scenario. 1

Furthermore, as the initial position error and relative rotation angle are big, large driven forces of the wheels are generated and therefore an overshoot comes up. The overshoot makes the robot pass ideal configuration $\mathcal{D}$ and then arrive at location (172, 186). After adjustments of control signals, the robot finally stops at location (190, 186). In terms of the orientation error between the robot and the ideal configuration, Figure 12 shows that the relative rotation angle dramatically decreases within the first second, followed by its fluctuation amongst 0° and -5°. Finally, the residual rotation angle is about -1°. The total time of the docking takes 2.5 seconds.

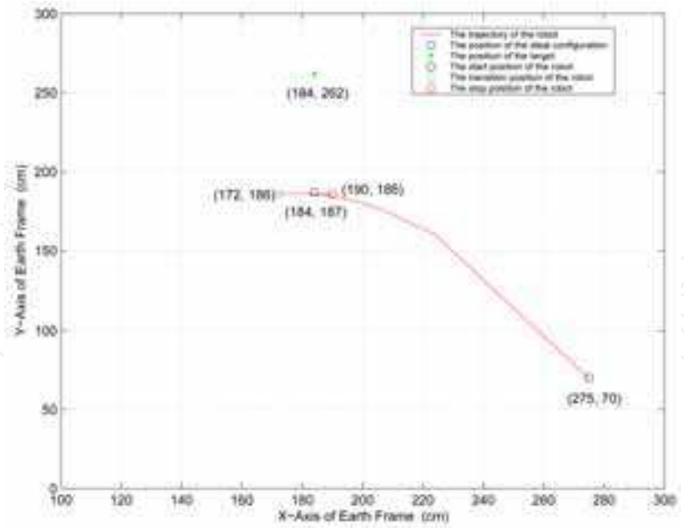

Fig. 11. The docking trajectory of the robot in Docking Scenario. 2

The third scenario of the docking is similar with the second one. The main differences are that the initial position of the robot in the third scenario is on the left-hand side of the ideal configuration and initial relative rotation angle e_θ = 42° while it is on the right-hand side and e_θ = -35° in the second one, as presented in Figure 13 and Figure 14. Figure 13 shows that the

robot starts from location (63, 176), passes through transition location (101, 286), and finally stops at (155, 284) which is close to $\mathcal{D}$ at location (151, 280) represented by the blue square. In addition, the relative rotation angle remarkably drops within the first 0.6 second, and the final one is close to 0°. The total docking time is 2.75 seconds.

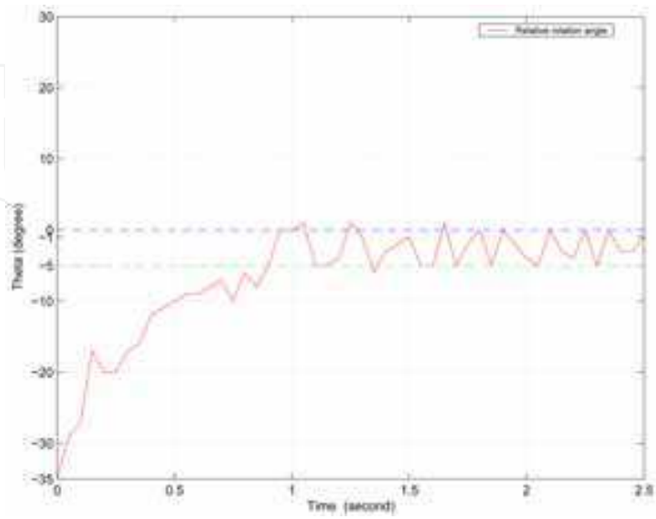

Fig. 12. The relative rotation angle between the robot and ideal configuration $\mathcal{D}$ in Docking Scenario. 2

Based on the results of the three docking scenarios reported, it is clear that the robot can not completely coincide with ideal configuration $\mathcal{D}$, that is, the residual errors exist. This is due to the influences of the un-modeled friction, the low resolution cameras employed and the unstable lighting. However, by using the proposed pose estimation algorithm and control scheme, the robot can successfully dock itself to the locations close to ideal configuration $\mathcal{D}$ at different positions and orientations. Therefore, the asymptotic stability is approximately satisfied in the docking scenarios.

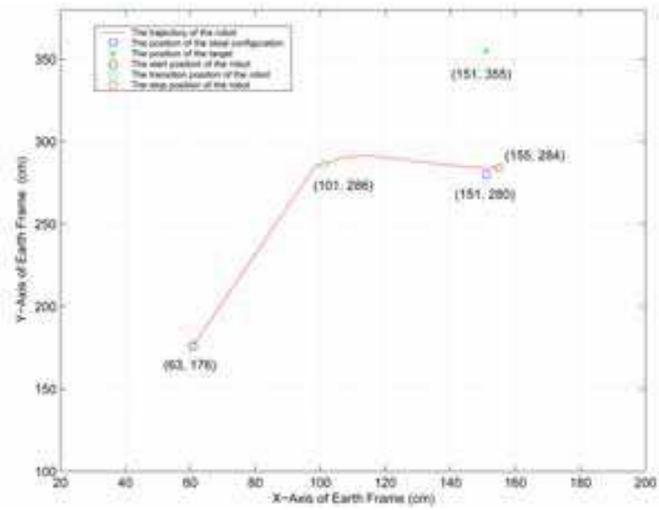

Fig. 13. The docking trajectory of the robot in Docking Scenario. 3

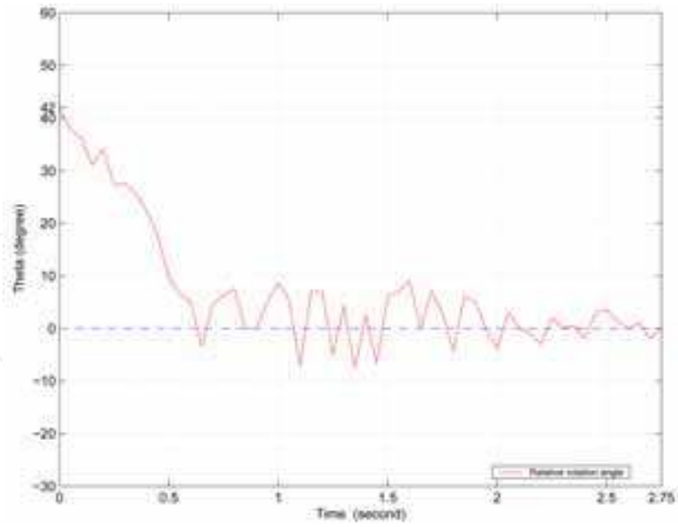

Fig. 14. The relative rotation angle between the robot and ideal configuration $\mathcal{D}$ in Docking Scenario. 3

6.3.2 Tracking

To validate the proposed robot's capability of the target tracking, a tracking experiment is performed. As shown in Figure 15, instead of being still, ideal configuration $\mathcal{D}$ clockwise moves along a predefined trajectory represented by the blue line at an average rate of $0.27m/s$. The predefined trajectory is a circle with a radius of 1 meter.

Initial location of configuration $\mathcal{D}$ is at coordinate (85, 201) represented by the blue circle while the initial position of the robot is (78, 118) represented by the red circle. Hence, the initial distance between them is about $0.833m$ and the initial relative rotation angle is about -58° as shown in Figure 15 and Figure 16. To make the tracking trajectory of the robot clear,

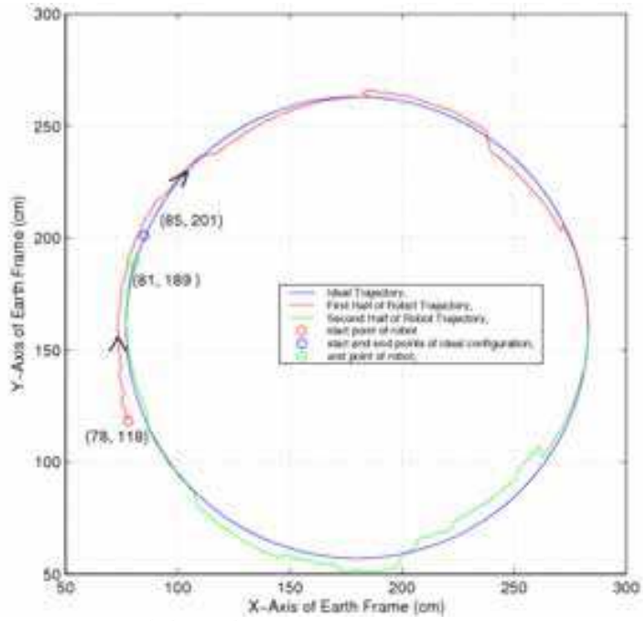

Fig. 15. The tracking trajectory of the robot

the first half of the robot trajectory is represented by the red line and the rest of it is represented by the green line. Finally, ideal configuration $\mathcal{D}$ arrives at coordinate (85, 201) again, but the robot stops at (81, 189) represented by the green circle. Thus, the residual position errors, $\bar{e}_x$ and $\bar{e}_y$, between the robot and the ideal configuration are $0.04m$ and $0.12m$, respectively, and the residual orientation error is around 0°, as shown in Figures 16-18.

Figure 15 shows that the tracking trajectory of the robot is basically consistent with that of the predefined trajectory of the ideal configuration, although some acceptable deviations between them exist. The factors which affect the tracking performance are summarized as follows:

1. Response delay: The response delay of the robot is mainly caused by the computational cost of the image and control modules and the friction. The response delay has the feedback signals, or more precisely the visual feedback and the wheel speed feedback, outdated so as to impair the tracking performance.
2. Friction: Due to the sliding motion of the wheels, there is strong friction existing between the wheels and ground. The friction plays an important role in affecting the performance of the tracking, especially the static friction which is considerably bigger than the dynamic one. In order to reduce the impacts of the friction on the tracking performance, an accurate model including friction effects should be considered and an advanced nonlinear control law should be applied.
3. Lighting condition: Lighting condition mainly exerts an impact on the image processing and therefore the pose estimation is affected. In this experiment, natural light is used rather than stable lighting. To improve the tracking performance, a laboratory with the relatively stable lighting condition is preferred.
4. Camera resolution: In this experiment, a pair of low resolution cameras are employed. Due to the limitation of the camera resolution, a small difference between the robot and ideal configuration $\mathcal{D}$ can not be detected on image planes. Therefore, the residual tracking errors exist.

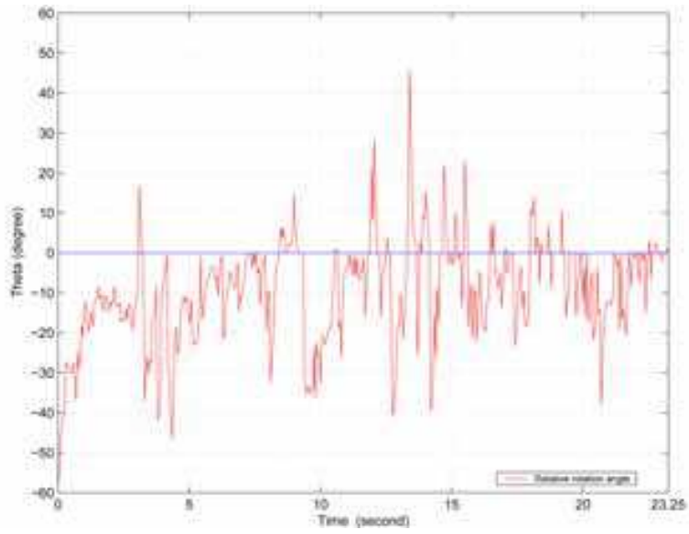

Fig. 16. The relative rotation angle between the robot and the ideal configuration expressed in the coordinate frame of the robot in the tracking scenario

Furthermore, Figure 16 shows the historic trajectory of the relative rotation angle obtained during the tracking and the total experimental time of 23.25 seconds. From the figure, it is clear that the relative rotation angle is large sometimes. This mainly results from either the sharp turn of the target, or of the robot, or of both. However, the large relative rotation angle is adjusted by the modulated control signals. Consequently, the relative rotation angle is bounded within 20° at most time, as reported in Figure 16. In terms of the position errors, Figure 17 and Figure 18 show that the position errors, $\bar{e}_x$ and $\bar{e}_y$, are bounded within 0.2m and 0.3m at most time, respectively, although the impacts of the sudden acceleration of the target and the response delay of the robot exist. Therefore, the tracking satisfies the BIBO stability.

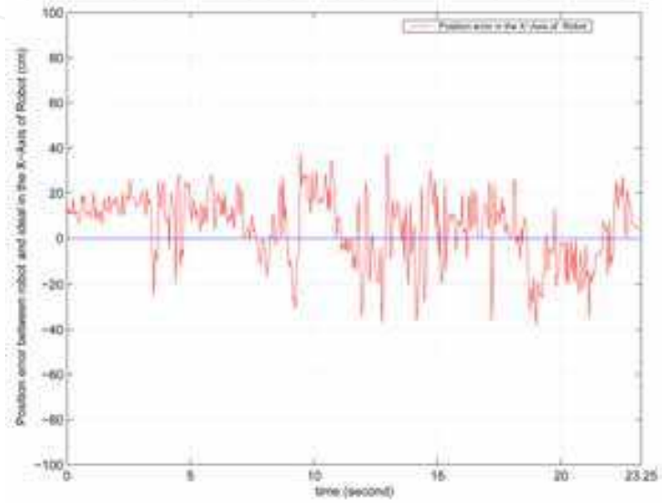

Fig. 17. The position error, $\bar{e}_x$ in the tracking scenario

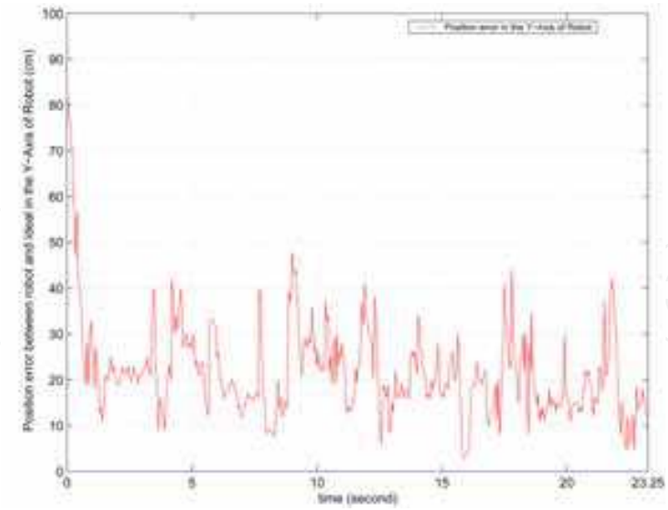

Fig. 18. The position error, $\bar{e}_y$ in the tracking scenario

The figures which include the docking and tracking trajectories of the robot are plotted with the sampled points on the physical trajectories chalked. The position errors shown in Figures 17 and 18 are expressed in the coordinate frame of the robot rather than the earth frame, and the relative rotation angle shown in Figures 12, 14 and 16 and the position errors

are not the ground truth values but the estimated values obtained by the calculation in real time.

7. Conclusions and future work

In this paper, the moving target tracking of an omnidirectional robot with stereo cameras has been presented. Unlike most car-like robots which are subject to the nonholonomic constraint, the omnidirectional robot proposed is holonomic with excellent maneuverability. Furthermore, this omnidirectional feature makes the path planning, which is a tough task for the car-like mobile robots, be a straightforward task.

The dynamic model of the robot is obtained first. Based on the dynamics, a simple PD controller with visual feedback is proposed. The PD control ensures the asymptotic stability of the tracking error if the target is still which is equivalent to the case of docking, or the bounded-input-bounded-output (BIBO) stability if the target is moving with varying but bounded rotational and linear velocities. The real-time experimental results demonstrate the feasibility and effectiveness of the proposed tracking scheme.

For future work, we plan to track target with a vision system which integrates one camera and some kind of sensor. In order to suppress the noise and enhance the tracking performance, a Kalman filter will be employed in our control loop.

8. References

R. L. Williams, B. E. Carter, P. Gallina and G. Rosati, "Dynamic Model with Slip for Wheeled Omnidirectional Robots", *IEEE Trans. Robot. Automat.*, vol. 18, pp. 285-293, June, 2002.

K. Watanabe, Y. Shiraishi, S. G. Tzafestas, J. Tang and T Fukuda, "Feedback Control of an Omnidirectional Autonomous Platform for Mobile Service Robots", *Journal of Intelligent and Robotic Systems*, vol. 22, pp. 315-330, 1998.

K L. Moore and N S. Flann, "A Six-Wheeled Omnidirectional Autonomous Mobile Robot", *IEEE Control Systems Magazine* , pp. 53-66, December, 2000.

G D. Hager, "A Modular System for Robust Positioning Using Feedback from Stereo Vision", *IEEE Transactions on Robotics and Automation*, VOL.13, NO.4, pp. 582-595, AUGUST 1997.

G. Artus, P. Morin and C. Samson, "Tracking of an Omnidirectional Target with a Nonholonomic Mobile Robot", *Proc. of the 11th Int. Conf. on Advanced Robotics*, pp. 1468-1473, Portugal, June, 2003.

H. Kase, N. Maru, A. Nishikawa and S. Yamada "Visual Servoing of the Manipulator using the Stereo Vision", *Proc. of the 1993 IEEE IECON*, pp. 1791-1796, VOL.3, Maui, HI, November, 1993.

N. Andreff, B. Espiau and R. Horaud "Visual Servoing from Lines", *Proc. of the 2000 IEEE International Conference on Robotics and Automation*, pp. 2070-2075, San Francisco, April, 2000.

A. K. Das, R. Fierro, V. Kumar, B. Southall, J. Spletzer and C. J. Taylor, "Real-Time Vision-Based Control of a Nonholonomic Mobile Robot", *Proc. of the 2001 IEEE International Conference on Robotics and Automation*, pp. 1714-1719, Seoul, Korea, May 21-26, 2001.

D. Burschka and G. Hager, "Vision-Based Control of Mobile Robots", *Proc. of the 2001 IEEE International Conference on Robotics and Automation*, pp. 1707-1713, Seoul, Korea, May 21-26, 2001.

D. Xu, M. Tan and Y. Shen, "A New Simple Visual Control Method Based on Cross Ratio Invariance", *Proc. of the 2005 IEEE International Conference on Mechatronics and Automation*, pp. 370-375, Niagara Falls, Canada, July, 2005.

H. Hashimoto, T. Kubota, M. Sato and F. Harashima "Visual Control of Robotic Manipulator Based on Neural Networks", *IEEE Transactions on Industrial Electronics*, pp. 490-496, VOL.39, No. 6, December, 1992.

S. Fujisawa, M. Ryuman, T. Yamamoto, H. Sogo, Y. Suita and T. Yoshida, "Development of Path Tracking Control for Omni-Directional Mobile Robot using Visual Servo System", *The 27th Annual Conference of the IEEE Industrial Electronics Society*, pp. 2166-2170, 2001.

M. Berkeme, M. Davidson, V. Bahl, Y Q. Chen and L. Ma, "Visual Servoing of an Omni-directional Mobile Robot for Alignment with Parking Lot Lines", *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, pp. 4204-4210, May, 2002.

R. Hartley and A. Zisserman, Multiple View Geometry *Cambridge University Press 2003*

S. He, Force Sensor Based Motion Control of Wheeled Mobile Robot *Master Thesis, Monash University, 2007*

Intel Open Source Computer Vision Library www.sourceforge.net/projects/opencvlibrary

Stereo Vision

Edited by Asim Bhatti

ISBN 978-953-7619-22-0

Hard cover, 372 pages

Publisher InTech

Published online 01, November, 2008

Published in print edition November, 2008

The book comprehensively covers almost all aspects of stereo vision. In addition reader can find topics from defining knowledge gaps to the state of the art algorithms as well as current application trends of stereo vision to the development of intelligent hardware modules and smart cameras. It would not be an exaggeration if this book is considered to be one of the most comprehensive books published in reference to the current research in the field of stereo vision. Research topics covered in this book makes it equally essential and important for students and early career researchers as well as senior academics linked with computer vision.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Jun Ming Kuang, Ming Liu and Xiang Lin (2008). Moving Target Tracking of Omnidirectional Robot with Stereo Cameras, Stereo Vision, Asim Bhatti (Ed.), ISBN: 978-953-7619-22-0, InTech, Available from: http://www.intechopen.com/books/stereo_vision/moving_target_tracking_of_omnidirectional_robot_with_stereo_cameras

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen