

# Embedded Vision-Based Autonomous Move-to-Grasp Approach for a Mobile Manipulator

Regular Paper

Jile Jiao<sup>1</sup>, Shuguang Ye<sup>2</sup>, Zhiqiang Cao<sup>1,\*</sup>, Nong Gu<sup>3</sup>, Xilong Liu<sup>1</sup> and Min Tan<sup>1</sup>

<sup>1</sup> State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China

<sup>2</sup> JiangSu King Source Electric Co., Ltd., Jiangsu, China

<sup>3</sup> Centre for Intelligent Systems Research, Deakin University, Waurn Ponds VIC, Australia

\* Corresponding author E-mail: zhiqiang.cao@ia.ac.cn

Received 29 Jun 2012; Accepted 11 Sep 2012

DOI: 10.5772/53276

© 2012 Jiao et al.; licensee InTech. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

**Abstract** This paper proposes a vision-based autonomous move-to-grasp approach for a compact mobile manipulator under some low and small environments. The visual information of specified object with a radial symbol and an overhead colour block is extracted from two CMOS cameras in an embedded way. Furthermore, the mobile platform and the postures of the manipulator are adjusted continuously by vision-based control, which drives the mobile manipulator approaching the object. When the mobile manipulator is sufficiently close to the object, only the manipulator moves to grasp the object based on the incremental movement with its head end centre of the end-effector conforming to a Bezier curve. The effectiveness of the proposed approach is verified by experiments.

**Keywords** Mobile manipulator, Embedded vision, Move-to-grasp, Bezier curve

## 1. Introduction

The development of autonomous mobile robots operating in unstructured and natural environments has been

studied extensively in robotics research. A mobile robot with a manipulator embraces mobile function and a high performance of operation, which is applicable in some challenging tasks such as cleaning of hazardous materials, transportation and rescue, etc [1]. Yamamoto and Yun present an algorithm to control the mobile platform so that the manipulator is maintained at a configuration which maximizes the manipulability measure, and simulation and experiment results verify its effectiveness [2]. In the follow-up work they study the effect of dynamic interaction between the manipulator and mobile platform [3]. In [4], Yamamoto and Fukuda propose an obstacle avoidance method for a coordinated task involving two mobile manipulators handling a common object, which is verified by simulation. [5] presents a task-oriented framework for dynamic coordination of mobile manipulator systems and studies the integration of mobility and manipulation, as well as cooperative manipulation. Platt *et al.* [6] introduce a refining sequence of controllers to solve the move-to-grasp problem, where controllers are defined in terms of controller goal regions and domains of attraction, and the

approach is demonstrated by experiments. Sheng *et al.* [7] develop a neural network-based methodology for motion control of mobile manipulators subject to kinematical constraints and experiments on a 4-DOF manipulator arm illustrate the improved performance of the method. [8] addresses the trajectory tracking control of mobile manipulator by using robust neural network-based sliding mode control, where the neural network may overcome the disadvantage of fixed large upper bound of sliding mode control. Simulation examples are given to demonstrate the performance of the method.

On the other hand, vision is broadly recognized as one of most versatile sensors with its abundant information, low cost and agility for use. For numerous autonomous mobile robots, vision measurement and vision-based control are commonly used [1,9]. Traditionally, vision measurement has enabled industrial robotic manipulators to grasp and move objects in a structured environment with respect to a given trajectory [1]. Nowadays, more rich information such as position, orientation, shape, colour and pixel coordinates of the specified object may be acquired, which is beneficial for robotic manipulators executing the grasp operation in a more flexible manner. Seelinger *et al.* [10] develop a high-precision visual control method for mobile camera-space manipulation for unmanned planetary exploration rovers. It achieves a high level of positioning precision, which is robust to model errors and uncertainties in measurement. The method's performance is verified by experiments. [11] proposes a graph-based representation for the problem of door opening, which is demonstrated on the PR2 mobile manipulation platform. A vision-based robot manipulator is designed and implemented in [12], and the robot manipulator can draw a picture and pick up balls with the assistance of vision systems. [13] presents an autonomous mobile manipulator that can overcome inherent system uncertainties and exceptions by utilizing three technologies: coordinated base and manipulator control, combined visual and force servoing, and error recovery through flexible task-level control, which is demonstrated by a "peg-in-a-hole" task. Jain *et al.* [14] present the assistive mobile manipulator EL-E with a focus on the subsystem that enables the robot to retrieve objects from and deliver objects to flat surfaces.

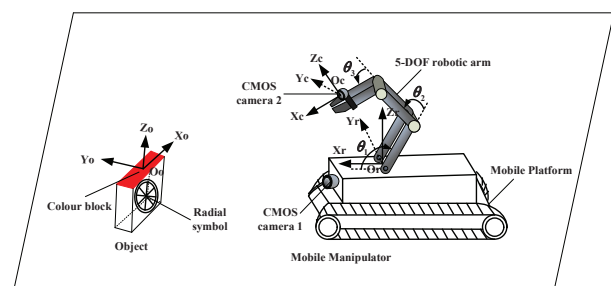
Compared to the framework in which the image is captured by CCD cameras, a method widely used in mobile robot systems, the system where the image is captured by CMOS cameras and processed by embedded processors is compact, power-efficient and low-cost [15,16]. With the increasing complexity of tasks and environments, the miniaturized mobile robot with manipulation capability under some low and small environments is required. In this case, image capturing through CMOS cameras and embedded vision-based control provide a better solution.

The main contributions of this paper are summarized as follows. The visual information from CMOS cameras is processed in an embedded way to meet the manipulation demands for a compact mobile manipulator under some low and small environments. By the coordination between mobile platform and manipulator postures, the object is kept within the view, which ensures the smooth execution of the task. Also, the Bezier curve is introduced to plan the grasping movement of the manipulator for better manipulation.

The remainder of the paper is organized as follows. Section 2 gives the description of the problem. In Section 3, the embedded vision-based autonomous move-to-grasp approach is described. The experiments are presented in Section 4 and Section 5 concludes the paper.

## 2. Problem Description

In many situations, a mobile manipulator is required to search autonomously for objects scattered in unstructured environments. Once an object of interest is encountered and identified, the mobile manipulator will endeavour to get close to it and pick it up with an onboard robotic arm based on sensing information. In this paper, autonomous move-to-grasp based on embedded vision is considered for driving a mobile manipulator approaching a specified object, which will be grasped by the manipulator. In order to complete the task, two CMOS cameras and a 5-DOF robotic arm are equipped on a mobile platform, where a CMOS camera 2 is fixed in the end-effector of manipulator, as shown in Figure 1. Each camera with an OV9650 optical chip captures the image, which will be processed by a corresponding embedded processing board running at 400MHz.



**Figure 1.** The move-to-grasp task for the mobile manipulator

Considering the size constraint of the compact mobile manipulator with embedded vision, the size of the object to be grasped is also small. From the viewpoints of environmental adaptability and better recognition from a farther distance, the object is endowed with a radial symbol with dual outer rings and a colour block on the top, which may satisfy the real-time computation requirement of an embedded processor. In Figure 1,  $O_oX_oY_oZ_o$  is the object coordinate frame,  $O_rX_rY_rZ_r$  is the mobile platform coordinate frame and  $O_cX_cY_cZ_c$  is the coordinate frame of camera 2.  $\theta_1$ ,  $\theta_2$  and  $\theta_3$  are pitch

angles of shoulder joint, elbow joint and wrist joint, respectively. The remaining two joints are the roll joint of the wrist and the gripper joint.

### 3. The Autonomous Move-to-Grasp Approach Based on Embedded Vision

The move-to-grasp process may be decomposed into three stages: wandering randomly when the mobile manipulator cannot detect the specified object, moving close to the object after the object is observed and grasping with the manipulator. Firstly, the mobile manipulator wanders to search for the object. Once the object is observed by recognizing the radial symbol feature based on the information provided by CMOS camera 1, it means that mobile manipulator is a little far from the object and the information is extracted for guiding the motion of the mobile platform. When it acquires the feature of the colour block through the CMOS camera 2, the related information is extracted to control the mobile platform to go on approximating the object. Also, the posture of the manipulator is adjusted to keep the object within view. When the mobile manipulator moves sufficiently close to the object, the mobile platform stops moving and then the manipulator executes the grasping operation.

#### 3.1 Object Recognition and Localization

The specified object is equipped with two features: a radial symbol with dual outer rings and a rectangle colour block. Next, the corresponding visual extraction processes are given.

##### 3.1.1 Information Extraction Based on a Radial Symbol

The radial symbol with dual outer rings adopted in this paper is an extension of that in [17], which is shown in Figure 1. There are some spokes with a centrosymmetry property, and the spokes near the centre taper off from the exterior. In this part, we consider the monochrome symbol and the features are extracted from a grey image.

In order to identify the pixel coordinates of the radial symbol centre, we may firstly eliminate some irrelevant points according to its low brightness feature and the centrosymmetry feature of the radial symbol centre. Then according to the radial feature of the symbol, a spoke-based sparse template (take 15×15 as an example) is designed to detect the centre point, as shown in Figure 2, and only points marked "x" will be checked. These points constitute a set  $\Omega$ . For a point  $P_c$  to be judged in the image, a 0-1 array is generated by binarizing the grey values of points in  $\Omega$  corresponding to  $P_c$ . If the number of changes from 0 to 1 is similar to the number of spokes, the point  $P_c$  is considered as a candidate centre point. After filtering, we obtain the image coordinate  $(u_c, v_c)$  of the centre point by averaging the coordinates of all

candidate points, and the average grey value  $g_a$  of all candidate points is also calculated.

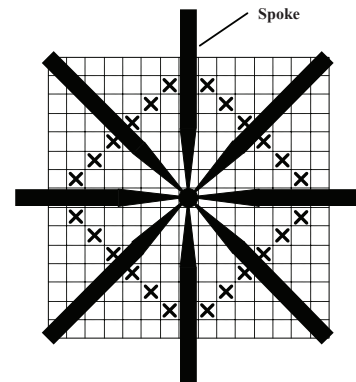


Figure 2. The spoke-based sparse template

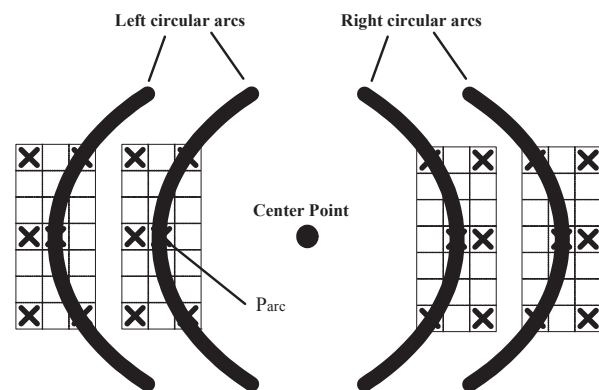


Figure 3. Curved edges recognition template

After the radial symbol is identified, it is necessary to measure the distance between the optical centre of camera 1 and the symbol by detecting the diameter of the symbol in the pixel. We consider two circular arcs at the left and right sides of the symbol, and their edges are detected by a sparse template (take 7×3 as an example), as shown in Figure 3. Only points marked "x" will be checked. For point  $P_{arc}$  in the curved edges recognition template, the grey values of three points marked "x" in the inner left circular arc are less than a given threshold  $T_e$ , whereas the other three points marked "x" in the template corresponding to this arc have grey values that are higher than  $T_e$ , where  $T_e$  is related to  $g_a$ . We illustrate the curved edge recognition process with the inner left circular arc. If there is a point on the left of the centre point with the same characteristics as those of  $P_{arc}$ , it is recognized as the edge of the inner left circular arc. Similarly, we obtain the edge of the inner left circular arc and the distance  $d_{lp}$  in pixel between it and the centre point of the symbol. Accordingly, the distance  $d_{rp}$  in the pixel between the edge of the outer right circular arc and the centre point is also obtained. Therefore, the diameter of the symbol in the pixel is expressed as  $d_{lp}+d_{rp}$ . Based on the pinhole model, the distance between optical centre of camera 1 and the symbol is then given as follows.

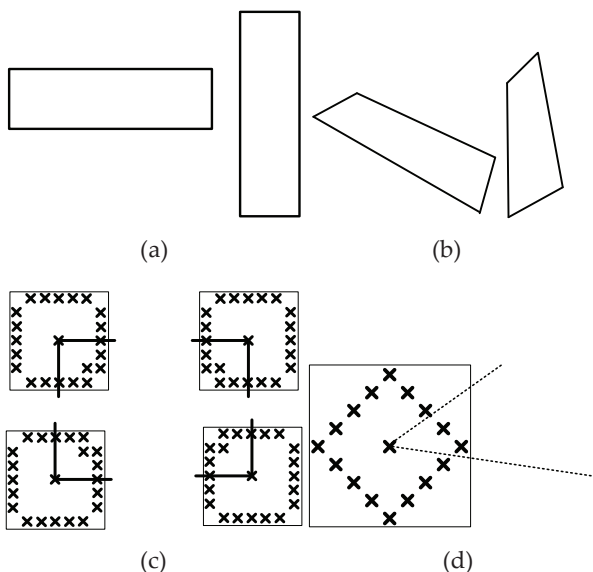
$$D = L * f / (d_{lp} + d_{rp}) \quad (1)$$

where  $L$  is the physical diameter of the symbol and  $f$  indicates the focal length in pixel.

### 3.1.2 Information Extraction Based on Rectangle Colour Block

On the top of the object of interest, there is a rectangle colour block. Different from common edge extraction and the corner matching method, in this paper, we make use of features of the colour block image from the CMOS camera 2 to extract corners directly, and then corners are used to measure the pose of the object. For the case where the object is not observed fully, the centre coordinate values ( $u_p, v_p$ ) of the seen part on the image plane are obtained.

When the object is seen in its entirety, we firstly consider the projection image of the rectangle colour block shown in Figure 4(a) and the recognition templates in Figure 4(c) are adopted to extract four corners of the block. There are four templates corresponding to four corners, respectively, and we take the top-left corner recognition template (7×7) as an example. Only points marked "x" are chosen for judgment. The centre point ( $i, j$ ) of the top-left template is considered as a corner only when the points ( $i, j$ ) and ( $i+2, j+2$ ) are colour points, and at most one of the two point sets ( $i-3, j-2$ ;  $j+2$ ) and ( $i+3, j-2$ ;  $j+2$ ) exists colour points, so does sets ( $i-2$ ;  $i+2, j-3$ ) and ( $i-2$ ;  $i+2, j+3$ ). The point with similar characteristics to that of the centre point in the top-left template is considered as a candidate top-left corner. After filtering, we obtain all candidate corners and the most top and most left point becomes the top-left corner. The bottom-left corner, top-right corner and bottom-right corner are also found in this manner.



**Figure 4.** Projection images of the rectangle colour block and the recognition templates; (a) Projection image of the rectangle block (b) Projection image of the rectangle block with deformation or rotation; (c) Corners recognition templates for (a); (d) Corner recognition template for (b)

Due to deformation or rotation, the projection image of the rectangle colour block changes (see Figure 4(b)) and the corner recognition templates in Figure 4(c) become invalid. In this case, we consider the template with adaptive size adjustment, as shown in Figure 4(d). Only points marked "x" are to be judged. The centre point ( $i, j$ ) of the template is considered as a candidate corner if ( $i, j$ ) is a colour point and the proportion of the number of colour points to that of all marked points is within a certain range. All points satisfying the characteristics of the centre point ( $i, j$ ) constitute corner candidate set  $S_c$ . After the four point clusters corresponding to four corners of the rectangle colour block are obtained by clustering according to the distribution of points in  $S_c$ , four corners are extracted based on the relative positions of these four clusters.

Based on the four corners extracted from the CMOS camera 2, the position of the centre point of the colour block  $P(p_x, p_y, p_z)$  relative to frame  $O_cX_cY_cZ_c$  is obtained according to the physical size of the rectangle colour block, the camera model [22] (see Eq.(2)) as well as the shape constraint-based pose measurement proposed by Xu *et al.* in [18].

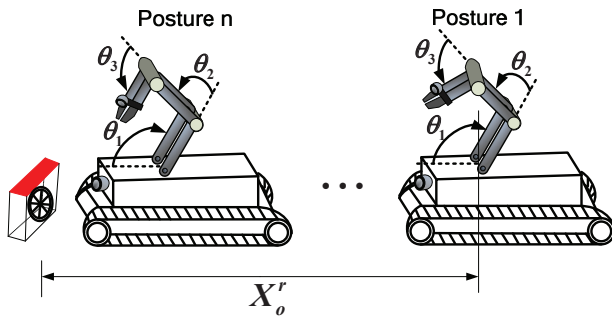
$$z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = M_{in} M_{out} \begin{bmatrix} x_t \\ y_t \\ z_t \\ 1 \end{bmatrix} = M \begin{bmatrix} x_t \\ y_t \\ z_t \\ 1 \end{bmatrix} = \begin{bmatrix} k_x & 0 & u_0 & 0 \\ 0 & k_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R_{3 \times 3} & P \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} x_t \\ y_t \\ z_t \\ 1 \end{bmatrix} \quad (2)$$

where ( $u_0, v_0$ ) is the image coordinate of the intersection point of the optical axis centre line and the image plane;  $k_x$  and  $k_y$  are the amplification coefficients of  $x$ -axis and  $y$ -axis direction, respectively; ( $x_c, y_c, z_c$ ) and ( $x_t, y_t, z_t$ ) are the coordinates of a point in  $O_cX_cY_cZ_c$  and  $O_oX_oY_oZ_o$ , respectively; ( $u, v$ ) is the image coordinate of the point; and  $R_{3 \times 3}$  is the rotation matrix. Intrinsic parameter array  $M_{in}$  is calibrated using the Matlab calibration toolbox [19].

### 3.2 Motion Control of Mobile Platform and Posture Adjustment of the Manipulator

The mobile platform is responsible for carrying the manipulator to a suitable grasp zone and its decision is autonomously made based on the extracted object information from CMOS cameras 1 and 2. In addition, during the approaching process based on camera 2, the manipulator has multiple varying postures, such as posture 1, ..., posture  $n$ , as shown in Figure 5. The postures of the manipulator are switched according to  $X_o^r$ , where ( $X_o^r, Y_o^r, Z_o^r$ ) is the position of point  $O_o$  in frame  $O_cX_cY_cZ_c$ . By the coordination between mobile platform and manipulator postures, the object is kept within the view of the CMOS camera 2.





**Figure 5.** The posture adjustment of the manipulator when the mobile platform approaches the object

Initially, the mobile platform moves randomly in the environment, and no camera sees the object. Once the object is observed by camera 1, the mobile platform adjusts the motion based on the image coordinate  $(u_c, v_c)$  of the centre point of the radial symbol and the distance  $D$  between optical centre of camera 1 and the symbol. The speed difference between left and right tracks is set to  $k_1 \cdot \arctan \frac{(v_c - v_0)}{D}$ , where  $k_1$  is a constant. When camera 2 sees the object in its entirety, the mobile platform adjusts its motion based on the position of point  $O_o$  in frame  $O_r X_r Y_r Z_r$ . The speed difference between left and right tracks is set to  $k_2 \cdot \arctan \frac{Y_o^r}{X_o^r}$ , where  $k_2$  is a constant.

$$\begin{bmatrix} X_o^r \\ Y_o^r \\ Z_o^r \\ 1 \end{bmatrix} = H_c^r \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} = \begin{bmatrix} R_c^r & d_c^r \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} \quad (3)$$

where  $H_c^r$  and  $R_c^r$  represent the homogeneous transformation and rotational matrix of frame  $O_c X_c Y_c Z_c$  to frame  $O_r X_r Y_r Z_r$ , respectively,  $d_c^r = [d_x, d_y, d_z]^T$  is the position vector of  $O_c$  in frame  $O_r X_r Y_r Z_r$ .

When part of the rectangle colour block is seen by camera 2, the mobile platform adjusts its motion according to  $X_o^r$  and the coordinate  $(u_p, v_p)$ . When  $v_p$  is approximately at the middle of the image in  $v$ -direction, the mobile platform moves ahead, otherwise, it only turns with the direction being related to  $v_p$ . Specifically, when  $X_o^r$  is small, the mobile platform only turns when  $v_p$  is approximately at the middle of the image in  $v$ -direction and it turns at a little angle with the direction being related to  $v_p$  in other cases. When the object is out of sight of camera 2, the mobile platform will move according to the previous decision.

### 3.3 Vision-Based Manipulator Grasp Control

The mobile platform stops when it reaches the suitable position, i.e., when the specified object is in the

workspace of the manipulator. Combined with object information provided by the CMOS camera 2, the grasping operation is executed with point  $E_p$  conforming to the trajectory planned by the Bezier curve according to the coordinates of  $E_p (X_{Ep}^r, Y_{Ep}^r, Z_{Ep}^r)$  and the object in frame  $O_r X_r Y_r Z_r$ , where  $E_p$  is labelled as the head end centre of the end-effector.

We consider the case where the manipulator grasps the object from its top with its posture returning to posture 1 for better manipulation. The cubic Bezier curve [21] is employed to plan the trajectory of  $E_p$ , which is given as follows.

$$P(t) = \sum_{i=0}^3 B_{i,n}(t) q_i = \sum_{i=0}^3 C_3^i (1-t)^{3-i} t^i q_i \quad (t \in [0,1]) \quad (4)$$

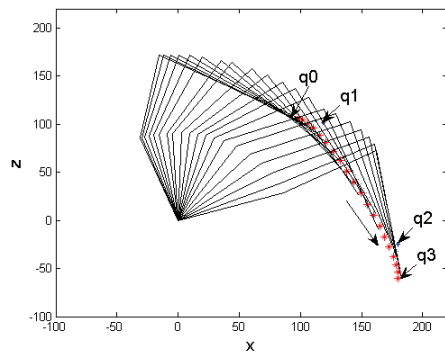
where  $q_i (i=0,1,2,3)$  are the coordinates of four control points of the curve and they are determined according to the coordinates of  $(X_o^r, Y_o^r, Z_o^r)$  and  $(X_{Ep}^r, Y_{Ep}^r, Z_{Ep}^r)$  in the initial grasping posture;  $B_{i,n}(t)$  is Bernstein basis polynomial. The Bezier curve is discretized into a series of sub-goal points for actual planning.

In order to ensure that the point  $E_p$  moves along the trajectory generated by the Bezier curve, the dynamic step angle-based incremental movement presented by Ning *et al.* in [20] is used. For our 5-DOF manipulator, only the pitch angles of the shoulder joint, elbow joint and wrist joint are responsible for the position of point  $E_p$ . The lengths of the linkages corresponding to these three joints are labelled as  $L_p (p=1,2,3)$ . The position of point  $E_p$  relative to frame  $O_r X_r Y_r Z_r$  can be obtained by

$$\begin{cases} X_{Ep}^r = \sum_{i=1}^3 L_i \cos(\sum_{k=1}^i \theta_k) \\ Z_{Ep}^r = \sum_{i=1}^3 L_i \sin(\sum_{k=1}^i \theta_k) \end{cases}$$

with  $Y_{Ep}^r$  as a mount-related constant.

The pitch angles may be adjusted according to  $\theta_p(t+1) = \theta_p(t) - \alpha_p$  ( $p=1,2,3$ ) when  $\theta_p(t) - \alpha_p$  is within  $[\theta_{pmin}, \theta_{pmax}]$ , and it remains unchangeable in other cases [20], where  $\theta_{pmax}$  and  $\theta_{pmin}$  are the upper and lower limits of joint angle  $\theta_p$ ;  $\alpha_p = \lambda_p [\delta_x G_{xp}(\theta_p) + \delta_z G_{zp}(\theta_p)]$  is the dynamic step angle;  $\lambda_p$  is a given positive constant related to the structure of the manipulator;  $G_{xp}(\theta_p)$  and  $G_{zp}(\theta_p)$  are the gradients of angle  $\theta_p$  along  $x$ -axis and  $z$ -axis directions of frame  $O_r X_r Y_r Z_r$ , respectively;  $\delta_x$  and  $\delta_z$  are the deviations between point  $E_p$  and next sub-goal point in  $x$ -axis and  $z$ -axis direction. Figure 6 gives a motion simulation of the manipulator with the head end centre of the end-effector along a trajectory generated by the Bezier curve.



**Figure 6.** The motion simulation of the manipulator based on the Bezier curve

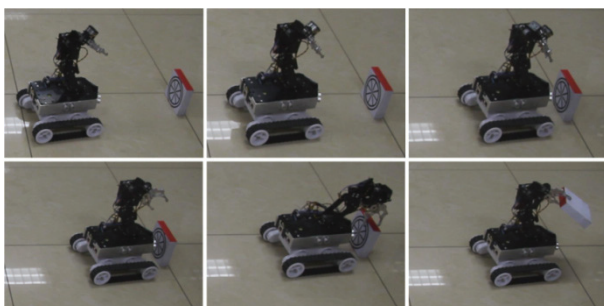
#### 4. Experiments

The proposed move-to-grasp approach for the mobile manipulator is verified by experiments. The parameters of the approach are as follows.  $k_1=44$ ,  $k_2=52$ ,  $\lambda_p=3.5e^{-5}$ . Intrinsic parameter array is

$$M_{in} = \begin{bmatrix} 334.02368 & 0 & 151.71122 \\ 0 & 334.08495 & 116.85215 \\ 0 & 0 & 1 \end{bmatrix}.$$

The lengths of linkages corresponding to the shoulder joint, elbow joint and wrist joint are 91mm, 88mm and 132mm, respectively.  $L=114.8\text{mm}$ ,  $f=672\text{pixel}$ . In the following experiments, we consider there are three postures for the manipulator and  $n=3$ .

In experiment 1, the mobile manipulator executes a short-range move-to-grasp task. The selected images are shown in Figure7, where the upper three images display the posture variation of the manipulator during the approaching process and the lower three images describe the grasping process while the mobile platform is stationary.



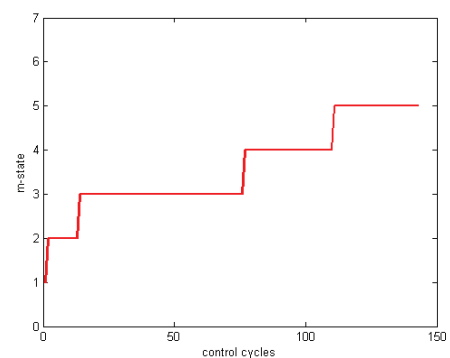
**Figure7.** A short-range move-to-grasp operation

Experiment 2 considers the case of a static object under a table and the move-to-grasp process is shown in Figure8. Figure9(a) gives the state transition diagram for the approaching process and the five states are considered for better description. The first two states are wandering randomly ( $m\text{-state}=1$ ) and approaching based on

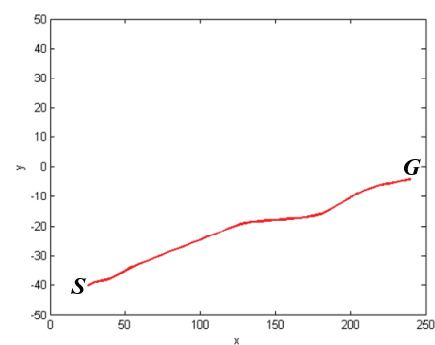
information from camera 1 ( $m\text{-state}=2$ ). The other three states are approaching based on information from camera 2 with posture 1, posture 2, posture 3 of the manipulator, which correspond to  $m\text{-state}=3, 4$  and  $5$ , respectively. The motion trajectory of the mobile platform is shown in Figure 9(b). From the experimental results, we can see that the mobile platform with the manipulator begins to execute the task from position  $S$  and it approaches the specified object after the object is observed. The posture of the manipulator is adjusted to meet the requirement of observation. When the mobile platform stops at position  $G$ , the manipulator begins to grasp and finally the task is completed.



**Figure 8.** Move-to-grasp operation for a static object under a table



(a)



(b)

**Figure 9.** The results of experiment 2; (a) State transition diagram for the approaching process; (b) Motion trajectory of the mobile platform

In experiment 3, the object is dragged manually and the selected images are shown in Figure 10. The state transition diagram for the approaching process is shown in Figure 11(a) and the motion trajectories of the mobile platform and the object are shown in Figure 11(b), where  $S_r$ ,  $G_r$  are the starting and ending positions of the mobile platform, and  $S_o$  and  $G_o$  are the starting and ending positions of the object. It is seen that the mobile manipulator may adjust itself with the motion of the moving object. After the object stops at position  $G_o$ , the mobile manipulator continues to approach the object and the object is picked up smoothly.

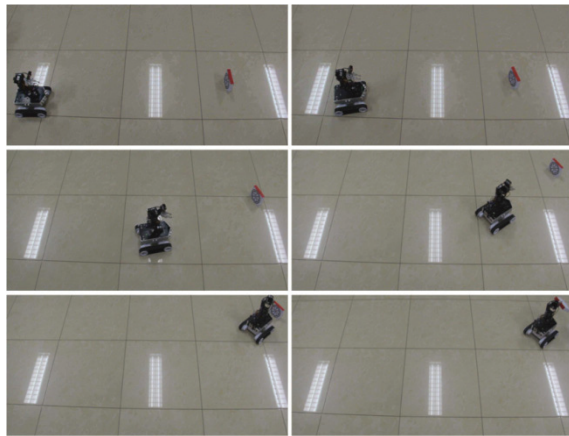


Figure 10. Move-to-grasp operation for a moving object

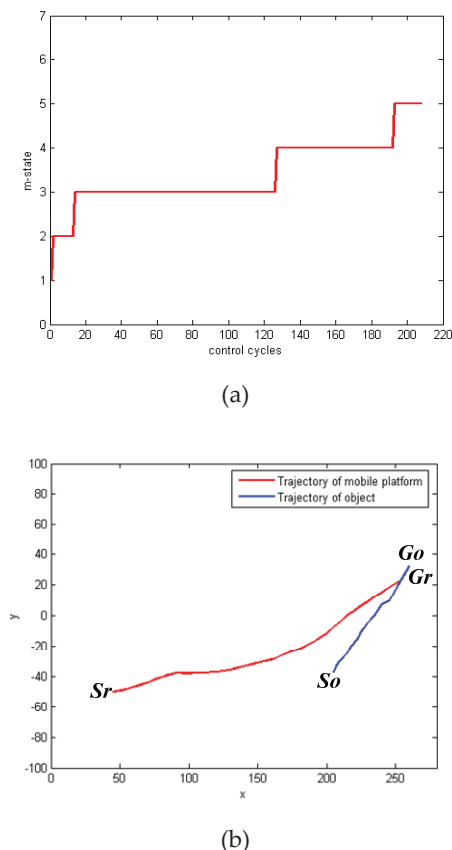


Figure 11. The results of experiment 3; (a) State transition diagram for the approaching process; (b) The motion trajectories of the mobile platform and object

## 5. Conclusions

This paper has studied an autonomous move-to-grasp operation based on embedded vision for a compact mobile manipulator with two CMOS cameras. The specified object is marked with a radial symbol and a rectangle colour block for recognition and localization. The mobile manipulator adjusts itself to approach the object by vision-based control. By combining the incremental movement method, the manipulator grasps the object with its head end centre of the end-effector moving along a Bezier curve. Experiments have been implemented to verify the effectiveness of the proposed approach. The ongoing and future work includes obstacle avoidance of the mobile platform as well as better coordination between the manipulator and the mobile platform.

## 6. Acknowledgments

This work is supported in part by the National Natural Science Foundation of China under grants 61273352, 61175111, and in part by the National High Technology Research and Development Program of China (863 Program) under grant 2011AA041001.

## 7. References

- [1] Wang Y, Lang H X, de Silva C W (2010) A Hybrid Visual Servo Controller for Robust Grasping by Wheeled Mobile Robots. *IEEE Transactions on Mechatronics*, vol.15, no.5, pp. 757-769
- [2] Yamamoto Y, Yun X P (1994) Coordinating Locomotion and Manipulation of a Mobile Manipulator. *IEEE Transactions on Automatic Control*, vol.39, no.6, pp. 1326-1332
- [3] Yamamoto Y, Yun X P (1996) Effect of the Dynamic Interaction on Coordinated Control of Mobile Manipulators. *IEEE Transactions on Robotics and Automation*, vol.12, no.5, pp. 816-824
- [4] Yamamoto Y, Fukuda S (2002) Trajectory Planning of Multiple Mobile Manipulators with Collision Avoidance Capability. *Proceedings of the International Conference on Robotics and Automation*, Washington DC, pp. 3565-3570
- [5] Khatib O (1999) Mobile manipulation: The robotic assistant. *Robotics and Autonomous System*, vol. 26, pp. 175-183
- [6] Platt R, Burridge R, Diftler M, et al. (2006) Humanoid Mobile Manipulation Using Controller Refinement. *IEEE-RAS International Conference on Humanoid Robots*, pp. 94-101
- [7] Sheng L, Goldenberg A A (2001) Neural-Network Control of Mobile Manipulators. *IEEE Transactions on Neural Networks*, vol.12, no.5, pp. 1121-1133
- [8] Xu D, Zhao D B, Yi J Q, Tan X M (2009) Trajectory Tracking Control of Omnidirectional Wheeled Mobile Manipulator: Robust Neural Network-Based Sliding Mode Approach. *IEEE Transactions on*

- Systems, Man and Cybernetics, Part B: Cybernetics, vol.39, no.3, pp. 788-799
- [9] Hutchinson S, Hager G D, Corke P I (1996) A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, vol.12, no.5, pp. 651-670
- [10] Seelinger M, Yoder J D, Baumgartner E T, Skaar S B (2002) High-precision Visual Control of Mobile Manipulators. *IEEE Transactions on Robotics and Automation*, vol.18, no.6, pp. 957-965
- [11] Chitta S, Cohen B, Likhachev M, et al. (2010) Planning for Autonomous Door Opening with A Mobile Manipulator. *IEEE International Conference on Robotics and Automation*, pp. 1799-1806
- [12] Wong C C, Shen Y J, Liu C C, et al. (2011) Vision-Based Robot Manipulator Design. Springer Berlin Heidelberg. Vol. 212, pp. 124-131
- [13] Hamner B, Koterba S, Shi J, et al. (2010) An Autonomous Mobile Manipulator for Assembly Tasks. *Autonomous Robots*, vol.28, no.1, pp. 131-149
- [14] Jain A, Kemp C C (2010) EL-E: An Assistive Mobile Manipulator that Autonomously Fetches Objects from Flat Surfaces. *Autonomous Robots*, vol.28, no.1, pp. 45-64
- [15] Spampinato G, Lidholm J, Ahlberg C, et al. (2011) An Embedded Stereo Vision Module for 6D Pose Estimation and Mapping, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Francisco, CA, USA, pp. 1626-1631
- [16] Ben-Tzvi P, Xu X (2010) An Embedded Feature-based Stereo Vision System for Autonomous Mobile Robots. *IEEE International Workshop on Robotic and Sensors Environments*
- [17] Liu X L, Qian H B, Cao Z Q, Tan M (2011) Visual Recognition Method Based on Symbol Features, *Huazhong Univ. of Sci. & Tech. (Natural Science Edition)*, vol.39, Sup. II, pp. 120-123 (in Chinese)
- [18] Xu D, Tan M, Jiang Z, et al. (2006) A Shape Constraint Based Visual Positioning Method for a Humanoid robot. *Robotica*, vol.24, no.4, pp. 429-431
- [19] Bouguet J Y (2010) Camera Calibration Toolbox for Matlab. [http://www.vision.caltech.edu/bouguetj/calib\\_doc/](http://www.vision.caltech.edu/bouguetj/calib_doc/)
- [20] Ning K J, Yang R Q, Zhang W J, Weng X H (2005) Development and Experiment of an Inverse Displacement Solution for Manipulator's Embedded Controller. *ROBOT*, vol.27, no.3, pp. 256-260 (in Chinese)
- [21] [http://en.wikipedia.org/wiki/B%C3%A9zier\\_curve](http://en.wikipedia.org/wiki/B%C3%A9zier_curve)
- [22] Ma S D, Zhang Z Y (2005) *Computer Vision*. Beijing: Tsinghua University Press (in Chinese)