

Application of Visual Servo Control in Autonomous Mobile Rescue Robots

H. Lang, M.T. Khan, K.-K. Tan, C.W. de Silva

Haoxiang Lang

Department of Mechanical Engineering,
University of Ontario Institute of Technology.
haoxiang.lang@uoit.ca

Muhammad Tahir Khan*

Institute of Mechatronics,
University of Engineering and Technology, Peshawar, Pakistan.
Corresponding author: tahir@uetpeshawar.edu.pk

Kok-Kiong Tan

Department of Electrical and Computer Engineering,
National University of Singapore.
Address: 4 Engineering Drive 3 Singapore 117576, Singapore.
eletankk@nus.edu.sg

Clarence W. de Silva

Department of Mechanical Engineering,
University of British Columbia.
desilva@mech.ubc.ca

Abstract: Mobile robots that integrate visual servo control for facilitating autonomous grasping and manipulation are the focus of this paper. In view of mobility, they have wider application than traditional fixed-based robots with visual servoing. Visual servoing is widely used in mobile robot navigation. However, there are not so many report for applying it to mobile manipulation. In this paper, challenges and limitations of applying visual servoing in mobile manipulation are discussed. Next, two classical approaches (image-based visual servoing (IBVS) and position-based visual servoing (PBVS)) are introduced along with their advantages and disadvantages. Simulations in Matlab are carried out using the two methods, there advantages and drawbacks are illustrated and discussed. On this basis, a suggested system in mobile manipulation is proposed including an IBVS with an eye-in-hand camera configuration system. Simulations and experimentations are carried with this robot configuration in a search and rescue scenario, which show good performance.

Keywords: Mobile Robots, Visual Servoing, Robot Control, Search and Rescue.

1 Introduction

Research and applications of mobile robots have drawn much attention in the robotics community because the emphasis of robotic research appears to have shifted from structured industrial environments to dynamic and partially-known natural environments (e.g., homes, cities, planet surfaces and deep sea). Mobile robots can be implemented in homecare, surveillance, exploration, search and rescue tasks where the working environments may be too hazardous for human workers. In such scenarios, robots are expected to have extensive autonomous ability and human capabilities, yet not suitable or impractical for human presence.

Examples for the necessity of applying autonomous search and rescue robots in practical applications were highlighted in the destruction of the World Trade Center in 2001 due to terrorist

attack in New York City and the damage caused by earthquake and tsunami in Sendai, Japan in 2012. In both incidents, not only did large numbers of victims lose their life, but significant number of rescues sacrificed their life in rescuing work. They were mainly caused by the search and rescue activities after those incidents because of dangerous environments and lacking of heavy-duty tools. Generally speaking, in any incidents, rescuers have about forty-eight hours to save humans in incident sciences. However, most of these time is wasted due to lacking of necessary resources or heavy-duty equipment for accessing damaged or contagious areas that are not suitable for humans. In this scenario, search and rescue robots have the advantages of being deployed instantaneously, and operating in any given working environments without human engagement.

An autonomous robot that is able to operate in unknown and dynamic workspace will have significant benefits in the above mentioned two cases in the areas of search and rescue. However, there is no such robot available in the market yet because of existing challenges in developing such systems in terms of both hardware and software. This paper focus on the control software of such system and emphasis on examining two traditional visual servoing methodologies along with their challenges in applying them in autonomous mobile manipulation applications. Two traditional methods of visual servo control are presented and examined through simulations; and comparative evaluation of these two methods are discussed, respectively. In the application of visual servoing, a suitable method for autonomous mobile rescue robots is suggested. Simulations and experimentations are carried out and demonstrated the effectiveness of the system and the methodologies.

2 Two Traditional Visual Servoing Methods

Vision-based mobile manipulation systems typically utilize camera with image processing and computer vision algorithms as its main sensory feedback in the control loop of mobile bases and their manipulators. This approach is called a visual servo control system or visual servoing, which involves many related research areas including image processing, computer vision, dynamic system modeling of robots and nonlinear control theory [1].

Vision sensors, usually cameras, are powerful sensors because they mimic human sense of vision and provide non-contact measurement of workspace. Furthermore, they can provide various types of sensed data such as pose, distance and objects in camera scenes. Because of these advantages of vision sensory, researchers have paid great attention to implement it in practical applications, especially in robotics areas. However, the image capturing hardware and processing software took seconds in producing image and doing analysis that makes real-time control impossible to apply in robotic applications [2]. Originally, Sanderson and Weiss invented a hierarchical Dynamic Look-an-Move structure to overcome the inefficiency of the vision system [3] where vision was utilized to provide control inputs to the joint-level controller. Then, the sub-control system utilized the joint feedback to internally stabilize the robot.

Nowadays, computer vision has benefited from the rapid development of computer technologies and image sensing hardware (CCD and CMOS) in terms of the processing time. Therefore, it makes possible for so called direct visual servoing where it heavily relies on the speed of vision processing for computing joint inputs [4] in robot control. Since 1990s, the term "visual servoing" (more appropriately, visual servo control) has accepted as a generic description of robotic system that has a visual feedback control loop. The subject has been studied in various forms for over twenty years now, with applications ranging from initially simple pick-and-place tasks to complex, real-time and autonomous mobile manipulation applications.

Visual servo control of mobile robots has drawn more attentions recently. When compared to the traditional fixed-base manipulators, one advantage of mobile robots is their ability to

navigate in a large workspace because of its mobility. A mobile manipulation system has better maneuverability and larger coverage of the workspace, which make it more widely applicable than a traditional fixed-based manipulator.

Autonomous navigation control of mobile robots employs the application of visual servo control. There is limited activity in this context possibly because mobile manipulation requires more accurate positioning performance and hence it is rather challenging. Some examples of visual servo application in mobile navigation are indicated in [5, 6]. The research and physical demonstration of visual servoing in mobile manipulation have been mostly limited two degrees of freedom DOF [7] and simplified robot models [8].

3 Developments in Visual Servoing Systems

In general, traditional visual servoing can be categorized into image-based and position-based approaches based on how the vision information is utilized [2]. The two traditional approaches can share similar control block diagrams with difference in reference of the control system and the processed information acquired from camera (Figure 1).

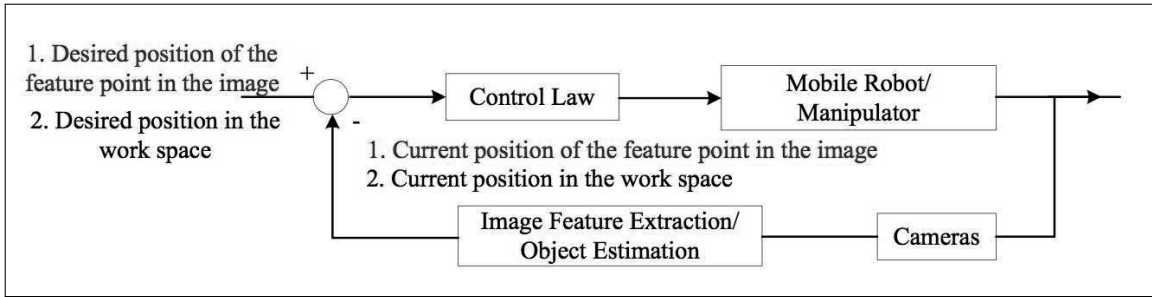


Figure 1: Block diagram of difference between position-based and Image-based visual servoing.

The objective of the controller is to minimize either the position error of the image or the position error of the object in the 3D space for two different methods respectively by controlling the motion of the robot joints. The error can be defined as follows:

$$e(t) = s(i(t), c) - s_r \quad (1)$$

where $e(t)$ is the error between desired and measured positions; s is the measurement from the image or pose estimation using computer vision; $i(t)$ is the image data from the camera; c is mathematic model of the camera; and s_r is the reference of the control system. $s(i(t), c)$ and s_r can be decided in detail depending on the type of visual servoing that is utilized. Once they are decided, a velocity controller of robot joints can be designed by

$$\dot{s} = L\xi_c^c \quad (2)$$

where \dot{s} is the velocity of the target objects (targets in the image from the camera in IBVS approach and 3D pose of the camera through 3D reconstruction in PBVS approach); L is the interaction matrix; and ξ_c^c is the velocity vector of the camera with respect to its own coordinate frame. Combining equations (1) and (2), the error dynamic equation can be written below:

$$\dot{e}(t) = L\xi_c^c \quad (3)$$

where $\dot{e}(t)$ is the time variation of the error. The square-of-error norm can be utilized to generate the control law as a candidate Lyapunov function:

$$\dot{s} = L^{-1}(-ke) \quad (4)$$

3.1 Image-based Visual Servoing Approach

The objective of image-based visual servoing (IBVS) is to decrease position errors of target objects in images acquired from the camera by moving each joint of the robot. The controller will continuously adjust the speeds of the robot joints based on controller outputs so that the trajectories of the target objects (u_i, v_i) navigate toward the desired positions (u_{di}, v_{di}) on camera sciences. The error vector of the target objects in the image plane is given by

$$e = \begin{bmatrix} u - u_d \\ v - v_d \end{bmatrix} = \begin{bmatrix} -s_x(r - r_d) \\ -s_y(c - c_d) \end{bmatrix} \quad (5)$$

where s_x and s_y represent the physical dimensions of image pixels of the camera sensor; and r , c are pixel coordinates. The velocity of the target objects can be expressed as

$$\dot{e} = \begin{bmatrix} \frac{d(u-u_d)}{dt} \\ \frac{d(v-v_d)}{dt} \end{bmatrix} = \begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} \quad (6)$$

Substituting equation (6) into (3), we can acquire the following equation:

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = L\xi_c^c \quad (7)$$

Assuming that the error dynamics obeys $\dot{e} = -ke$, a proportional controller based on the Lyapunov method can be designed accordingly:

$$\xi_c^c = L^{-1}(-ke) \quad (8)$$

where k is the proportional gain (a scalar), with $k > 0$. Therefore, the control law can be obtained by substituting equation (5) into (8), as

$$\xi_c^c = -kL^{-1} \begin{bmatrix} -s_x(r - r_d) \\ -s_y(c - c_d) \end{bmatrix} \quad (9)$$

In equation (9), r and c represent the pixel coordinate in the captured image by the camera. The desired velocities of the camera in the 3D space can be computed from the image measurements. Moreover, the developed controller guarantees asymptotic stability in the closed-loop system.

The image Jacobian (L) in the model of IBVS [9], which is also named as interaction matrix, describes the motion relationship between the target objects in the image and camera in 3D space in terms of their velocities, respectively. The detailed formulation of the interaction matrix is shown below:

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = L\xi_c^c = \begin{bmatrix} -\frac{\lambda}{z^c} & 0 & \frac{u}{z^c} & \frac{uv}{\lambda} & -\frac{\lambda^2+u^2}{\lambda} & v \\ 0 & -\frac{\lambda}{z^c} & \frac{v}{z^c} & \frac{\lambda^2+u^2}{\lambda} & -\frac{uv}{\lambda} & -u \end{bmatrix} \xi_c^c \quad (10)$$

where λ is focal length; and z^c is the distance between the object of interest and the camera coordinate frame in the 3D workspace. The distance must be either measured or approximated for completing the interaction matrix. In many applications, multiple objects of interest are usually measured for the purpose of estimating the distance information if there is no direct way of measuring it. Moreover, if it is a typical robotics arm with six degree-of-freedom (DOF), at least three objects of interest are required. In practical applications, four objects of interest are usually utilized so that the system can overcome both singularity of the interaction matrix and local minima [1–4].

3.2 Position Based Visual Servoing

In Position-based Visual Servoing (PBVS) approach [10] [11], features of the object of interest are extracted from images from one or more cameras. Both camera model and a geometric model of the target object are utilized to mathematically estimate the pose of the camera with respect to the target object in a 3D space. It involves three coordinate frames for designing a control system. They are current frame f_c , desired frame f_d , and reference frame f_r . In the workspace, the reference frame is typically attached to the target object. In the robot workspace, the function of the controller is to calculate and provide outputs for moving the robot with the ultimate goal of reducing the error between the current measured pose and the desired reference pose. The approach is also named 3D visual servoing because the control scheme is defined in the Cartesian space. However, the IBVS is usually called 2D visual servoing because it doesn't need to consider the pose of the robot in its workspace. Instead, it only considers target objects in provided image from cameras, which is a 2D workspace. The features in PBVS can be defined below:

$$s = (T, \theta U) \quad (11)$$

where T is a translation vector and θU is an orientation vector, respectively. The error dynamic equation for the controller can be written by

$$e = (T_r^c - T_d, \theta U) \quad (12)$$

The interaction matrix can be written as:

$$L = \begin{bmatrix} -I_3 & [T_r^c]_x \\ 0 & L_{\theta U} \end{bmatrix} \quad (13)$$

where $L_{\theta U}$ can be described by [18]:

$$L_{\theta U} = I_3 - \frac{\theta}{2}[U]_x + \left(1 - \frac{\text{sinc}\theta}{\text{sinc}^2\frac{\theta}{2}}\right)[U]_x^2 \quad (14)$$

where $\text{sinc}x$ is the sine cardinal, which is defined by $x\text{sinc}x = \sin x$ and $\text{sinc}0 = 1$. Therefore, the velocity control scheme can be generated as follows:

$$v_c = -kL^{-1}e \quad (15)$$

4 Visual Servoing Simulations

Figure 2 shows a typical image-based visual servoing system in Matlab simulation. In the simulation results, the control system detects the current position of the objects (they are four marks "o" in this case) which locate in the right bottom corner. By comparing the current locations of the objects with the desired locations which are marked by "*", a position error vector will be generated and fed into the image-based visual servoing controller for calculating the velocity commands for each joint of the robot. The robot applied the velocity commands for its joints; and move in the 3D workspace. Figure 2(a) shows the trajectory of the objects from the initial locations to the goal locations during the servoing. Figure 2(b) shows the convergence of the position error during servoing. Since the IBVS scheme decides the 3D motion of the robot based on 2D feedback information, it simplifies the control problem. However, it losses the control of the pose and trajectory of the robot in the 3D workspace. Therefore, obstacle

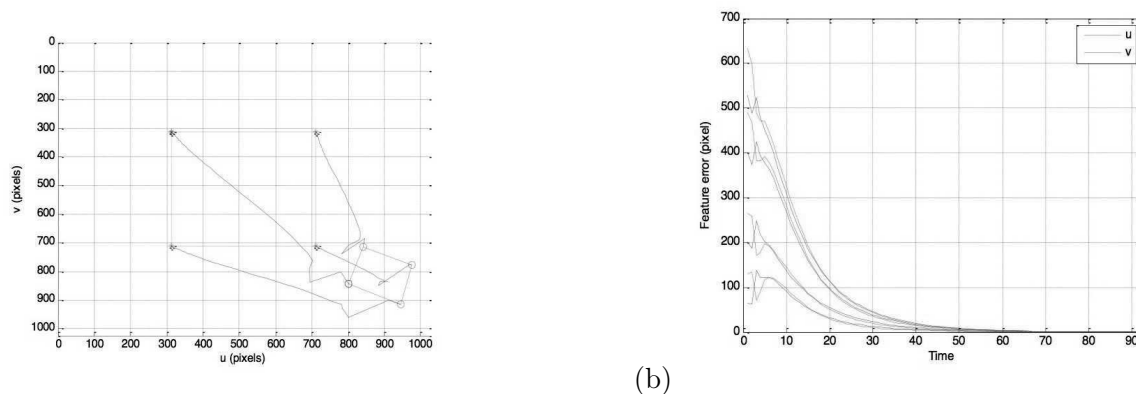


Figure 2: IBVS in Matlab simulation: (a) trajectories of objects of interest in camera scenes; (b) error convergence of the positions of the objects.

avoidance and other psychological constraints during the servoing cannot be considered. Moreover, if large displacement of the camera is required for moving from the current pose to the desired pose, the control system may easily fail because a large change of image scene can be caused by a small displacement of the camera in Cartesian space. Furthermore, the interaction matrix in the model of the IBVS introduces non-linear and time-varying terms, which are challenges of designing controllers.

Figure 3 demonstrates a typical position-based visual servoing in Matlab simulation. In this given example, the pose of the camera is controlled by eliminating the pose error between the current pose of the camera and the desired pose of the camera in 3D workspace. Figure 3(a) demonstrates the initial pose of the camera in the 3D workspace and the desired pose of the camera, respectively. Figure 3(b) records the velocity commands that are sent from the controller to each joint of the robot. Figure 3(c) shows the trajectory of the camera pose during servoing. Comparing the IBVS showed in above section, one of the most significant advantages of this approach is that it controls the pose and trajectory of the camera in the Cartesian space directly. Obstacles and other physical constraints can be easily considered and integrated into the system during servoing. However, there is a disadvantage in this approach. There is absolutely no control of the image, which will result in losing objects of interest in the camera scene. Eventually, it will cause the failure of the whole control system because there will be no data and information from the feedback loop in the control system. Moreover, 3D models of objects of interest have to be completely available. Moreover, 3D reconstruction is a challenge because multiple feature points are required from camera in order to calculate poses, which may not be the case during the servoing. Camera calibration result will be another issue for successful PBVS system.

5 Application of Visual Servoing in Autonomous Mobile Robots

Figure 4(a) shows a robotic search and rescue scenario, in a future city. It presents an emergency situation where a group of general-purpose robots identify injured humans and remove them from the dangerous region. In applications of autonomous search and rescue robots, the workspace is usually large, and sometimes unknown and dynamic; the base frame of the robot has ability to move around in a large area; and the objects of interest are not predefined well, which depend on different applications. Therefore, 3D models of target objects are unavailable in most of the time. In this situation, position-based visual servoing will fail and image-based visual servoing has the advantages over the position-based visual servoing of reducing the computational burden by avoiding unnecessary image interpretations. Also, it eliminates the calibration errors of camera parameters. Furthermore, the eye-in-hand camera configuration has more advantages

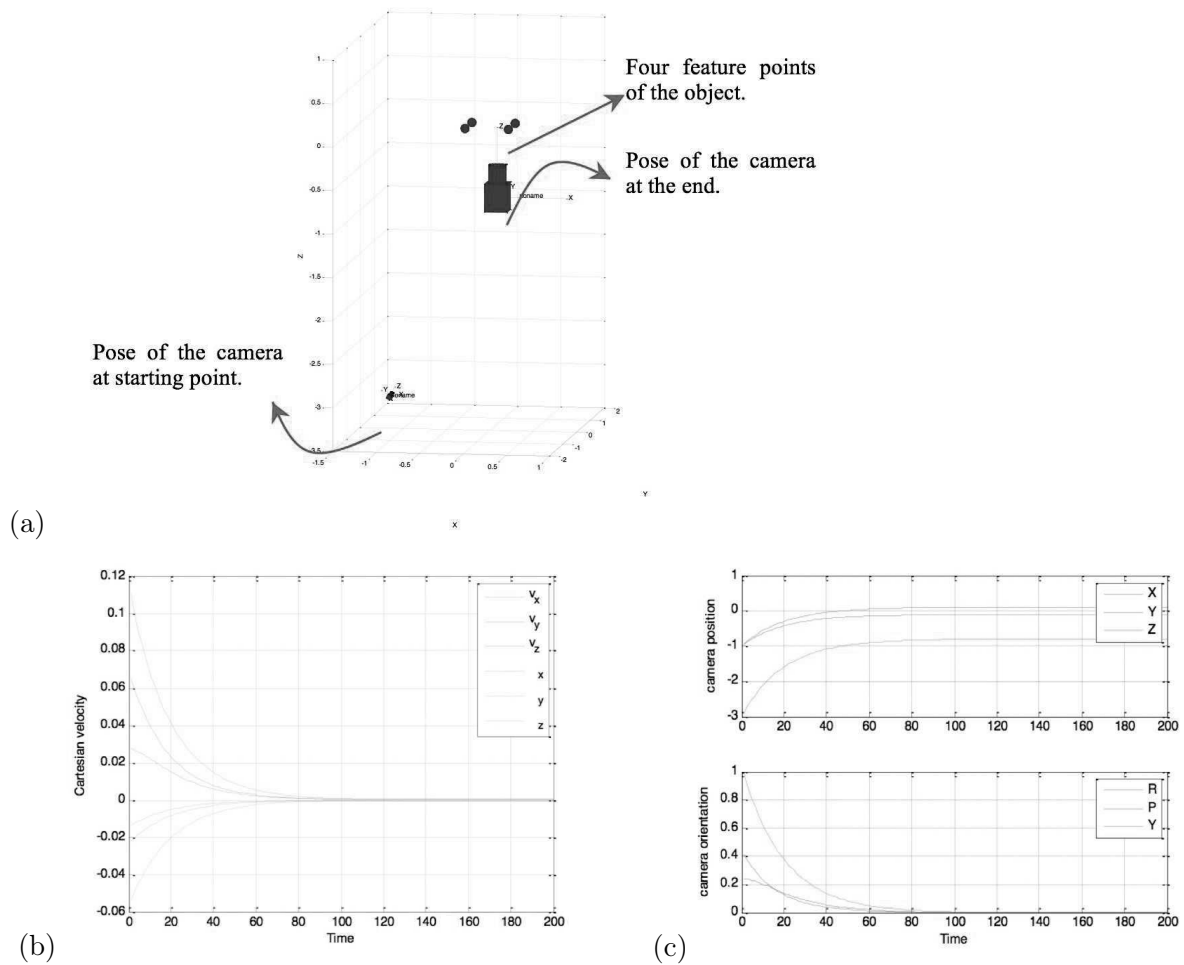


Figure 3: PBVS simulation in Matlab: (a) current and desired pose of the camera Cartesian space; (b) controller output; (c) trajectory of the camera pose in Cartesian space.

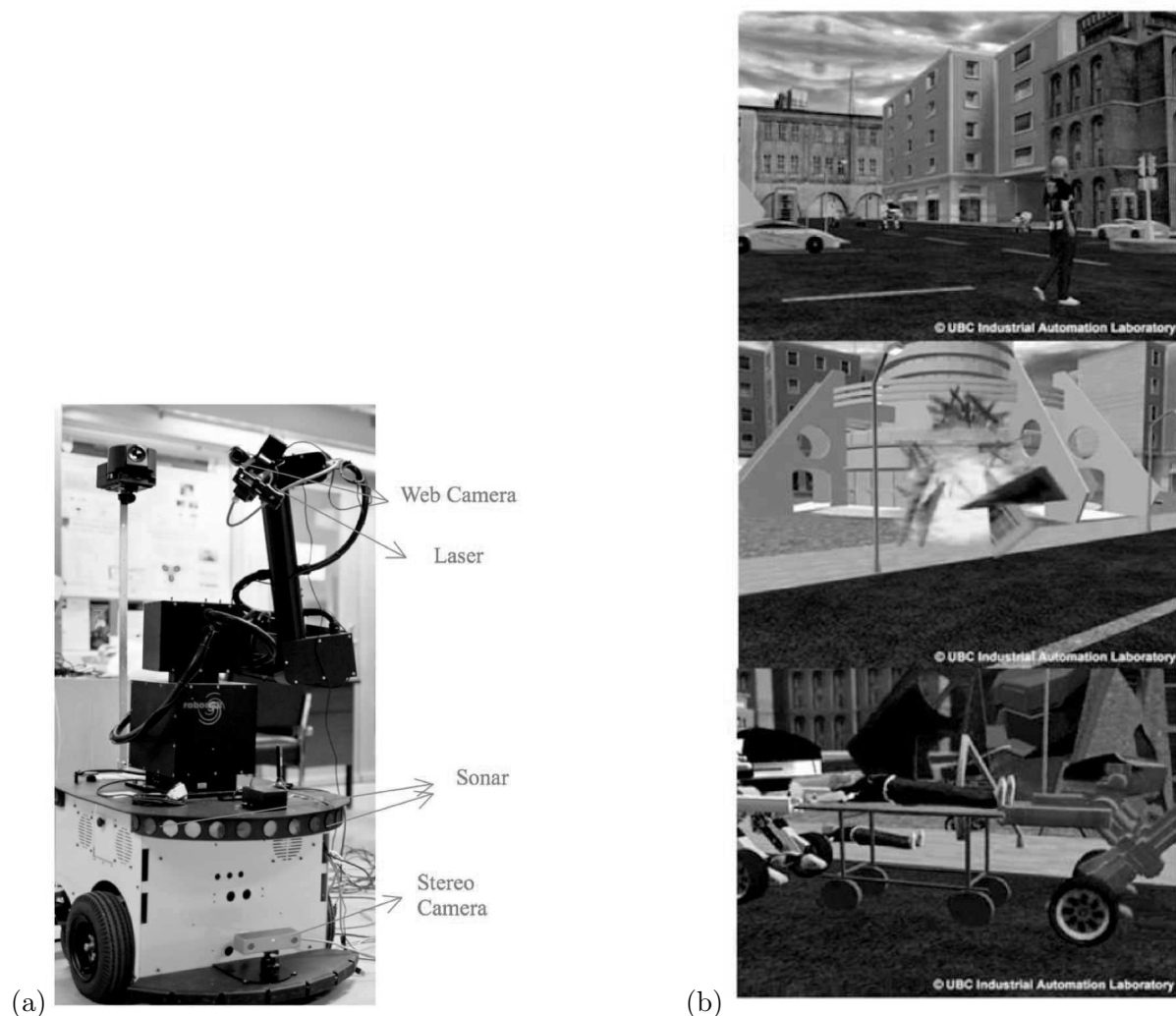


Figure 4: (a) Robot search and rescue scenario; (b) Physical configuration of the test bed.

than the eye-to-hand one in mobile robot applications. The suggested system for autonomous mobile rescue application is illustrated in Figure 4(b).

The given system contains a mobile base (pioneer powerbot), a manipulator (robuarm), and different types of sensors (a web camera, laser, sonar, and stereo camera). Both two cameras are configured as eye-in-hand configuration. The nonlinear terms in the IBVS model is linearized by updating the position of the feature points in each iteration. The time-varying term (the distance between object and camera) is also updated in each iteration by using the measurement results from the laser distance finder and sonar sensors. Therefore, there is no need for distance estimation in the proposed approach. Figure 4(b) describes an application of the proposed system where there are general-purpose robots in a future city (e.g., surveillance). If an incident or emergency situation occurs in the city, these robots are able to quickly response and change their roles to involve emergency cases such as search and rescue work.

The control law of traditional IBVS is discussed and shown in (9), which describes the relationship between the velocities of the camera and of the feature points in the image. Now, the relationship between velocity of the camera with respect to its own coordinate frame and speeds of each joints will be formulated, which is the mathematic model of IBVS of the robot. In order to generate the model, one should first consider the mathematic relationship between the velocity of the end effector with respect to the robot base frame and the velocity of each joint

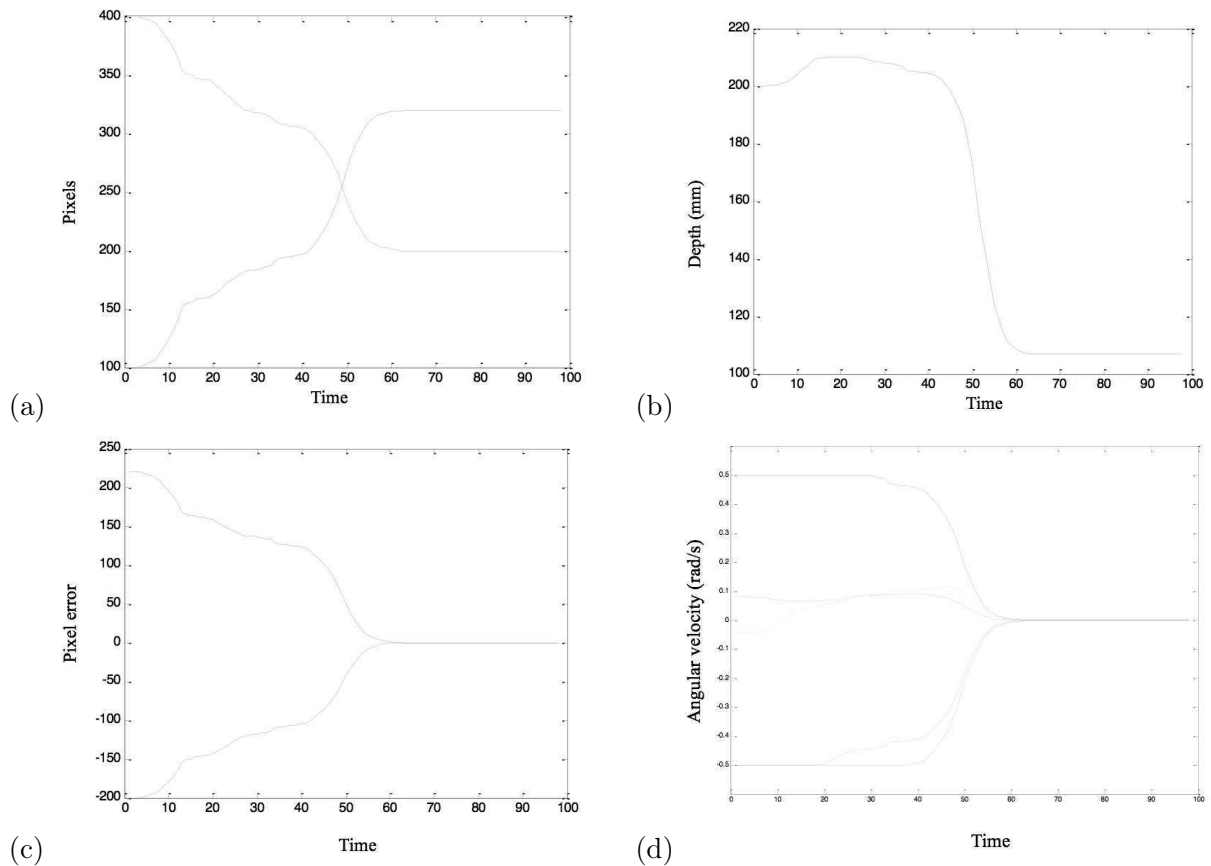


Figure 5: (a) the performance of the system in terms of the trajectory of the feature point; (b) distance information measured by laser distance finder; (c) error convergence of the system; (d) velocities commands from the controller.

as:

$$\xi_t^r = J_1 \dot{q} \quad (16)$$

where J_1 is the Jacobian matrix, and \dot{q} is the velocity vector of the robot joints. The velocity of the end effector with respect to its own coordinate frame is derived by

$$\xi_t^t = G^{-1} \xi_t^r \quad (17)$$

where $G = \begin{bmatrix} R_t^r & 0_{3 \times 3} \\ 0_{3 \times 3} & R_t^r \end{bmatrix}$. Since the camera is rigidly attached to the end effector, the camera frame and the end effector frame have a constant relationship of homogeneous transformation. Therefore, the relationship between camera velocity and the robot end-effector velocity may be written as:

$$\xi_t^t = J_2 \xi_c^c \quad (18)$$

where $J_2 = \begin{bmatrix} R_c^t & s(d_c^t) R_c^t \\ 0_{3 \times 3} & R_c^t \end{bmatrix}$.

Finally, by combining equations 9, 20, 21, 22, the mathematical representation of the relationship between object velocities on camera scenes and joint velocities of the robot can be written as:

$$\dot{s} = LJ^{-1} J_2 G^{-1} J_1 \dot{q} \quad (19)$$

Denoting the terms $LJ^{-1}G^{-1}J_1$ by M , equation (19) can be simplified as:

$$\dot{s} = M\dot{q} \quad (20)$$

The square of the error norm can be utilized to derive the control law as a candidate Lyapunov function:

$$\dot{q} = M^{-1}(-ke) \quad (21)$$

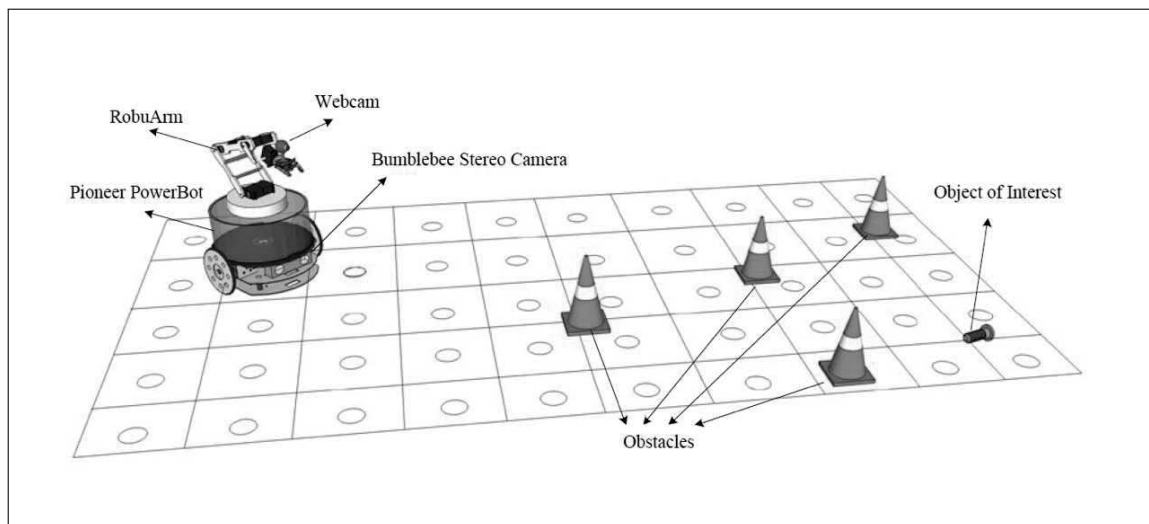


Figure 6: Experimental setup in the laboratory.

There are nonlinear and time varying terms in this model which can cause challenges in designing and implementing controller. In order to eliminate the nonlinear and time-varying terms, an adaptive parameter updating mechanism is introduced in the system to linearize and nonlinear terms and update the time varying terms so that the system can be considered a linear system. The time varying term in the model is the distance between camera and objects. A laser distance finder is utilized to update the modeling by using the depth measurement from the laser distance finder. The non-linear terms are about the position of the objects in the image plane in the interaction matrix. To eliminate those nonlinear terms, the position of the objects in the interaction matrix are updated in each iteration of the controller. Figure 5 shows the performance of the suggested IBVS system in Matlab simulation. The trajectory of the object on images is shown in Figure 5(a). The distance measurements between camera and object by the laser distance finder is illustrated in Figure 5(b). Figure 5(c) shows the performance of the controller, showing the error convergence. Figure 5(d) shows the velocity history of each joint of the robot during the servoing.

Figure 6 shows the experimental setup in our laboratory. The purpose of the mobile robot is to find the object of interest in the workspace, navigate to it and finally approach and grasp the object using image-based visual servoing. In this paper, only the visual servoing aspect is considered. During the final operation, the stereo camera mounted on the mobile base will detect the object of interest in the scene and guide the robot to move toward to the object by using visual servo control. The mobile robot approaches the object until the object is inside the workspace of the manipulator. Next, the visual servoing of the manipulator is activated to grasp the object.

Figure 7 presents an experimental result of the proposed system. Figure 7(a) shows the desired camera view, and Figure 7(b) shows the current camera view when the robot starts to servo. As discussed in the previous session, the purpose of the IBVS is to eliminate the image error between the current camera view and the desired camera view, by controlling the motion

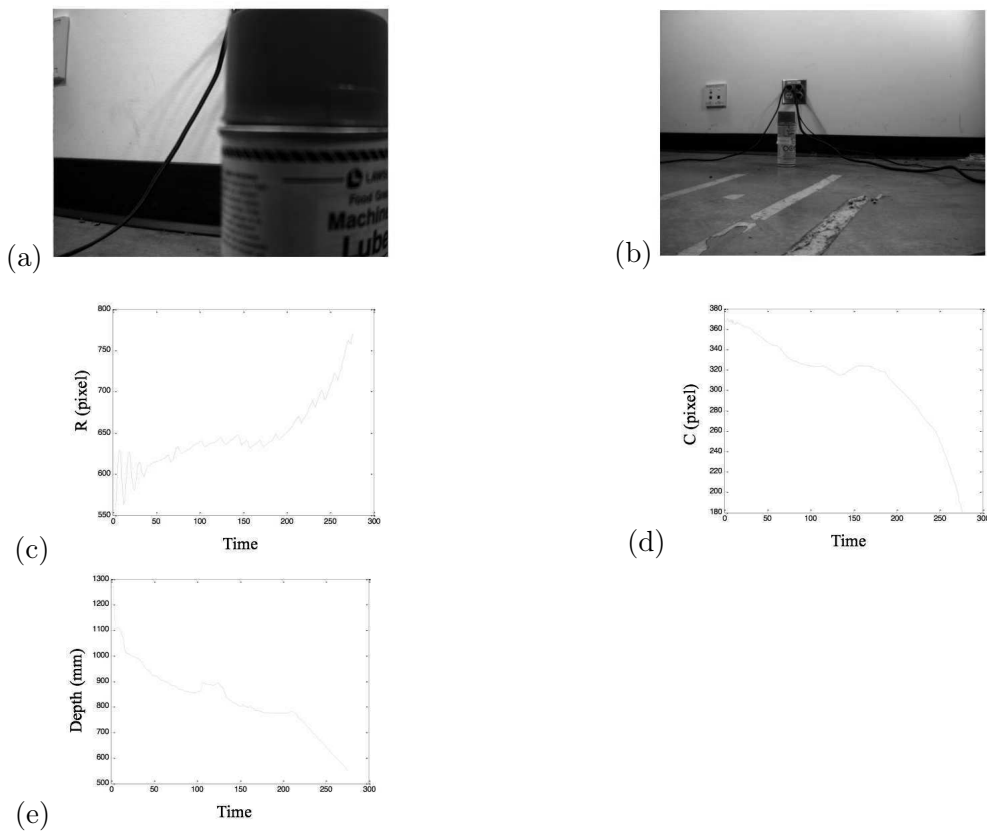


Figure 7: (a) Desired camera view; (b) current camera view; (c) trajectory of the object in in R; (d) trajectory of the object in C; (e) distance between the camera and the target object.

of the robot. Specifically, the system controls the robot motion by eliminating the position error of the object in the camera view. Figure 7(a) gives the trajectory of the object in the camera view in the row (R) axis; and Figure 7(b) gives the trajectory of the object in the camera view in the column (C) axis. Figure 7(d) gives the distance between the camera and object.

6 Conclusions

This paper introduced and described two basic methods of vision servoing for mobile robots. The detailed formulation of these two methods were discussed along with their advantages and disadvantages in applying in mobile robots. A suggested method including selection of visual servoing method and hardware configuration was proposed for applications of autonomous mobile manipulation. Both simulation and experimentation were carried out with the proposed system. The results showed the effectiveness and good performance of the methodology in a mobile robotic application.

Bibliography

- [1] Chaumette, F. ; Hutchinson, S. (2006); Visual servo control part 1: basic approaches, *IEEE Robotics and Automation Magazine*, 13(4): 82-90.
- [2] Agin, G. J. (1979); Real time control of a robot with a mobile camera, *SRI International, Menlo Park, Technical Note 179*.
- [3] Sanderson, A. C. ; Weiss, L. E. (1980), Image-based visual servo control using relational graph error signals, *in Proc. IEEE*,:1074-1077.
- [4] Hutchinson, S. ; Hager, G.D. ; Corke, P. I. (1996), A tutorial on visual servo control, *IEEE Transaction On Robotics and Automation*, , 12(5): 651-670.
- [5] Ma, Y. ; Kosecka, J. ; Sastry, S. S. (1999), Vision guided navigation for a nonholonomic mobile robot, *IEEE Transactions on Robotics and Automation*, 15(3): 521-536.
- [6] Fang, Y.; Dixon, W. E. ; Dawson, D. M. ; Chawda, P. (2005), Homography-based visual servo regulation of mobile robots, *IEEE Transactions on Systems, Man and Cybernetics - Part B*, 35(5): 1041-1050.
- [7] Wang, Y. ; Lang, H. ; de Silva, C. W. (2010), A hybrid visual servoing controller for robust manipulation using mobile robots, *IEEE/ASME Transactions on Mechatronics*, 15(5): 757-769.
- [8] Gangloff, J. ; de Mathelin, M. (2002), Visual servoing of a 6-DOF manipulator for unknown 3-D profile following, *IEEE Transactions on Robotics and Automation*, 18(4): 511-519.
- [9] Spong, M. W. ; Hutchinson S. ; Vidyasagar M. (2005); *Robot Modeling and Control*, Wiley.
- [10] Malis, E. ; Chaumette, F. ; Boudet, S. (1999), 1-1/2-D Visual Servoing, *IEEE Transactions on Robotics and Automation*, 15(2): 238-250.
- [11] Wilson, W. J. ; Hulls, C. C. W. ; Bell, G. S. (1996), Relative end-effector control using Cartesian position-based visual servoing, *IEEE Transactions on Robotics and Automation*, 12(5): 684-696.