Robust industrial robot real-time positioning system using VW-camera-space manipulation method

Article *in* Industrial Robot · January 2014

DOI: 10.1108/IR-04-2013-340

3 authors, including:

Yong Liu
Nanjing University of Science and Technology
48 PUBLICATIONS **272** CITATIONS

SEE PROFILE

Steven B. Skaar
University of Notre Dame
59 PUBLICATIONS **1,361** CITATIONS

SEE PROFILE

Robust industrial robot real-time positioning system using VW-camera-space manipulation method

Yong Liu, Dingbing Shi and Steven Baard Skaar
School of Computer Science and Technology, Nanjing University of Science and Technology, Nanjing, China

Abstract

Purpose – Vision-based positioning without camera calibration, using uncalibrated industrial robots, is a challenging research problem. To address the issue, an uncalibrated industrial robot real-time positioning system has been developed in this paper. The paper aims to discuss these issues.

Design/methodology/approach – The software and hardware of this system as well as the methodology are described. Direct and inverse kinematics equations that map joint space into "camera space" are developed. The camera-space manipulation (CSM) algorithm has been employed and improved with varying weights on camera samples of the robot end effector, and the improved CSM is named VW-CSM. The experiments of robot positioning have been conducted using the traditional CSM algorithm and the varying-weight CSM (VW-CSM) algorithm, respectively, both without separate camera calibration. The impact on the accuracy and real-time performance of the system caused by different weights has been examined and discussed.

Findings – The experimental results show that the accuracy and real-time performance of the system with the VW-CSM algorithm is better than the one with using the original CSM algorithm, and the impact on the accuracy and real-time performance of the system caused by different weights has been revealed.

Originality/value – The accuracy and real-time performance of the system with the VW-CSM algorithm is verified. These results prove that the developed system using the VW-CSM algorithm can satisfy the requirements of most industrial applications and can be widely used in the field of industrial robots.

Keywords Positioning, Camera-space manipulation, Industrial robot, Robust

Paper type Research paper

I. Introduction

There has been a strong interest in industrial robot positioning in manufacturing automation. Especially the task of robot positioning without camera calibration with high-precision demand is a challenging topic. Because of the often daunting demands of calibration-based methods (Liu and Xi, 2011), vision-based robot control is seen as a way to achieve both robustness and precision of visually guided manipulation without the need to acquire and sustain very precise calibration of cameras and manipulator kinematics (Skaar, 1996).

Vision-based robot control is traditionally classified into two types of systems: position-based and image-based systems (Hutchinson *et al.*, 1996). In position-based visual servoing (PBVS), the calibration of camera and robot is essential; any errors in calibration of the vision system will lead to errors in the 3D reconstruction and subsequently to errors during task execution (Corke and Hutchinson, 2001).

In image-based visual servoing (IBVS), there are problems with time delay due to image processing and differences between the image processing period and control period. Therefore, realization of precise and high-speed visual servo is difficult because of the time delay of image processing. Kinbara *et al.* (2006) proposes a new visual servo system for manipulators and an estimation method of image features has been applied to a fixed camera to overcome these problems, but it still cannot avoid the drawback of image Jacobian calibration. IBVS performs very poorly for cases that involve large rotations about the optical axis. Corke *et al.* (2009) proposes using polar, instead of Cartesian coordinates, of feature points overcomes this limitation. Moments are generic (and usually intuitive) descriptors that can be computed from several kinds of objects, defined either from closed contours or from a set of points. Tahri and Chaumette (2005) proposes a new IBVS scheme based on image moments, valid for dense and discrete objects. A dynamic quasi-Newton method for uncalibrated, vision guided robotic tracking control with fixed imaging is developed and demonstrated. This method does not require calibrated kinematic and camera models (Piepmeier *et al.*, 2004). Liu *et al.* (2006) proposes a depth-independent interaction matrix, which differs from the traditional interaction matrix in that it does not depend on the depths of the feature points.

The current issue and full text archive of this journal is available at **www.emeraldinsight.com/0143-991X.htm**

Industrial Robot: An International Journal
41/1 (2014) 70–81

[DOI 10.1108/IR-04-2013-340]

This work was supported in part by National Natural Science Foundation of China under grants 61175082, Jiangsu Science & Technology Pillar Program under grants BE2011192, National Human Resources and Social Security Ministry, and Technology Foundation for Selected Overseas Chinese Scholar and Ministry of Personnel of China.

Most of the methods for camera calibration make use of a calibration pattern to find the parameters of the camera model. However, Skaar first proposed a different approach, named camera-space manipulation (CSM), to the use of vision. It is based upon estimating the relationship between the position of certain cues on the robot manipulator in at least two cameras and the corresponding joint rotations of a robot (Skaar *et al.*, 1987). The task of vision-based robot control can be completed without camera calibration. Juan Manuel Rendon-Mancha presents a new version of the camera-space-manipulation method. The set of nonlinear view parameters of the classic CSM is replaced with a linear model. Recently, the CSM method has been successfully implemented in a wide variety of fields including space exploration, warehouse environments, industrial tasks, and mobile robots (Rendon-Mancha *et al.*, 2010).

Robust accuracy and real-time performance of localization are the two most important measurement indicators of vision-based robot positioning. In this paper, the CSM algorithm has been applied to a six-DOF robot, and the real-time performance and positioning accuracy of the developed uncalibrated system has been improved by proposing the improved CSM algorithm which takes varying weights on vision samples of the robot end effector. Relative to more conventional alternatives – teach/repeat for blind operation, and visual guidance from various forms either of camera/kinematics calibration or visual servoing – we believe CSM offers great prospect for real-world use due to its precision and robustness. Rather than using globally calibrated system parameters, we rely on estimates that produce locally precise "camera-space kinematics" which can be used to calculate and then drive joint rotations to the place where all camera-space maneuver objectives are realized in a minimum of two separated, 2D image planes, or "camera spaces". The estimates are precise within a usefully large asymptotic region of the maneuver terminus, and easily updated to extend to neighboring regions with the addition of a few new image-plane samples, simultaneous with joint-rotation samples. The local nature of the estimation scheme means that, although the models used may be globally unable to characterize the system well (e.g. neglect of some lens distortion and small DH-parameter errors), there is a large, local asymptotic region within which estimates used to plan the movement are very accurate.

The system developed with the VW-CSM algorithm can satisfy the real-time, precision, and robustness requirements of most industrial applications and can be widely used in the field of industrial robots. Among various enabled industrial applications are those, such as outdoor operations, where robots and image sensors are not permanently fixed in place, but where precision of delivery of the end effector relative to an arbitrarily located work piece is still essential. Provided cameras, robot base, and work piece can be relied upon to remain stationary between the typically short time interval between establishment of camera-space objectives (usually using structured light and possibly some high-level human supervision) and maneuver execution, the method is reliably precise without prior calibration. CSM is, in this respect, slightly more demanding that visual servoing – which would continue to close errors even in the presence of ongoing shifts in component positions – however, we believe that CSM's attendant ability to apply whatever end-member visual samples are conveniently available, given conveniently placed stationary-camera positions, make its implementation more practicable than that of visual servoing. Moreover, there is no need for immediate image processing with CSM; the maneuver trajectory is planned based upon the batch of current-maneuver samples that have been processed, and, when an additional samples becomes available, these are added at whatever point they come in order to refine the "trajectory to go" as planned in joint space. As will be seen in the paper, there is little improvement beyond some point of adding more images. Continuous feedback is not a requirement; there is no instability due to feedback delay; and the averaging effect of applying multiple, pixel-quantized, image samples permits sub pixel accuracy of estimates of the camera-space kinematics on the basis of which the maneuver-to-go is planned and executed in joint space.

This paper is structured as follows. In Section II, the structure of the VW-CSM-based robot positioning system is described. In Section III, the relevant theory is described, which maps joint space into "camera space". In Section IV the control strategy of the developed system is depicted. In Section V, the real-time performance and accuracy of the developed uncalibrated industrial robot real-time positioning system are verified using several different kinds of experiments; also the impact on system performance caused by the VW-CSM algorithm is analyzed.

II. The developed robot positioning system

A. System setup

The structure of this industrial-robot, real-time, servo-positioning system, is shown in Figure 1 and the actual, physical system is shown in Figure 2. The figures show the main components of the robot positioning system, which consists of two cameras, an industrial robot, a designed end-effector fixture and an industrial computer. The computer communicates with the ABB robot via sockets based on TCP/IP protocol and can send orders to control the robot movement according to joint-level instructions or Cartesian instructions. The image processing program in the computer captures images by using Ethernet network from each camera to the industrial computer.

In Figure 2, two cameras are mounted about 120 cm away from the active workspace of the robot, on the top of the work platform, with a separation of 60° between focal axes so that the end-effector of robot and the target points are in a good field of view ($0.6\,\text{m} \times 0.6\,\text{m}$ with the $z = 0.2\,\text{m}$ w.r.t robot base frame) in the two cameras.

The designed fixture is mounted on the end-effector of the robot and its relative position with respect to the end-effector is precisely known. There are artificial features (one white circle and three black circles), as shown in Figure 3, on the surface of the designed fixture, the centroid of all these artificial features can be easily located in 2D camera-space using a special-purpose, automatic, image-analysis program. The coordinate systems of robot and camera are shown in Figure 4, separately and consequently the 3D coordinates and 2D camera space coordinates of the artificial features are described in these defined frames.

B. Software

The main modules of the software in the developed system are shown in Figure 1, which include the image processing program, CSM-algorithm program, and robot control algorithm program. In Figure 3, the camera space coordinates

Figure 1 The designed structure of industrial robot real-time servoing positioning system

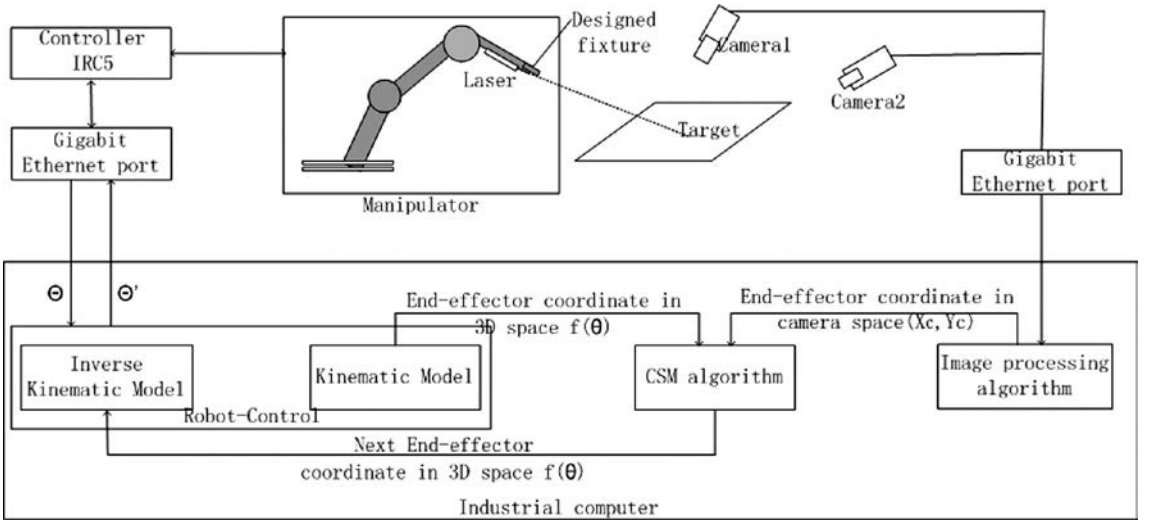
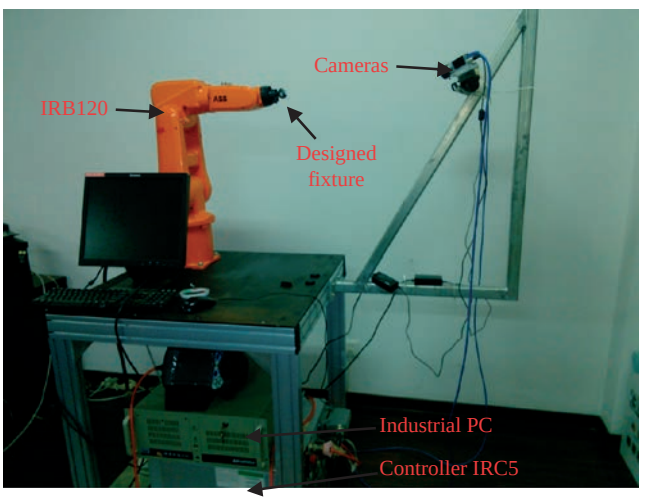


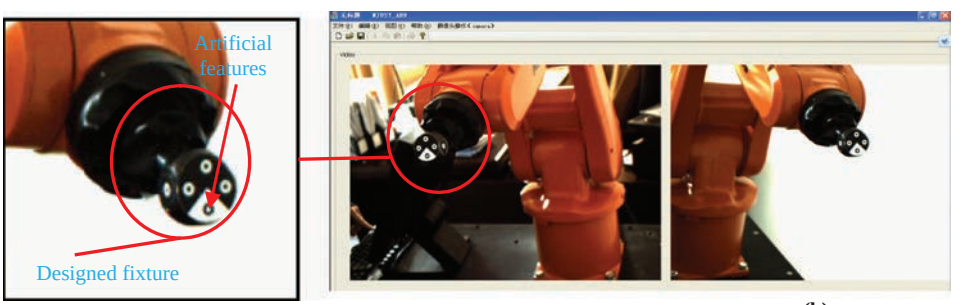
Figure 2 The developed industrial robot real-time positioning system



of the artificial features (circles) on the designed fixture can be calculated by the image processing program. The principle upon which detection of the circles, and location of their centers, is based is invariance, under orthographic transformation, of certain geometric properties of the feature. In particular, provided the mark is placed upon a flat surface, lines or strings of pixels passing through the feature's center appear in the image with ratios of approximately d:l:d, where d is the number of "dark" pixels and "l" is the number of light pixels of the center. Because the feature is printed with a white-center diameter equal to 1/3 of the overall diameter, depending upon the intensity threshold used, "l" will approach "d" when the string of pixels passes through the centroid. The algorithm sweeps horizontally through the image until this condition is met to within a given tolerance or threshold. Subsequent, vertical and diagonal sweeps through the candidate center are then used to verify the feature and calculate a centroid that represents an average of the four passes, in accordance with Figure 5. Orthographic mapping of the physical feature into the image is predicated on a sufficiently large ratio of the distance component along the camera focal axis of the feature to that feature's physical diameter. This ratio is greater than 100 for the present tests.

The coordinates of the artificial features in camera space are calculated by the image processing program. Simultaneously, the coordinates of the artificial features in robot space are achieved by the forward kinematic model with the joints of robot. Then the coordinates of artificial features in camera space and in robot space are sent to CSM-algorithm program, the relationship between the coordinates of artificial features in nominal 3D space and their corresponding coordinates in

Figure 3 The visual scenes of two cameras



Notes: (a) The visual scene of camera 1; (b) the visual scene of camera 2

Figure 4 The 3D space and camera space

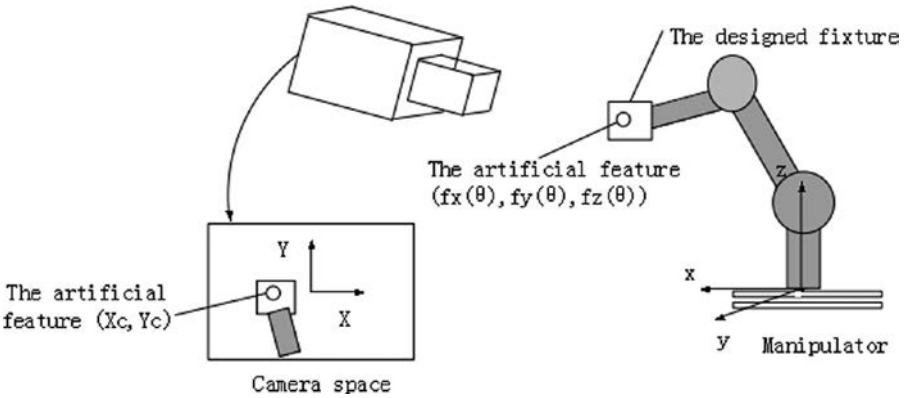
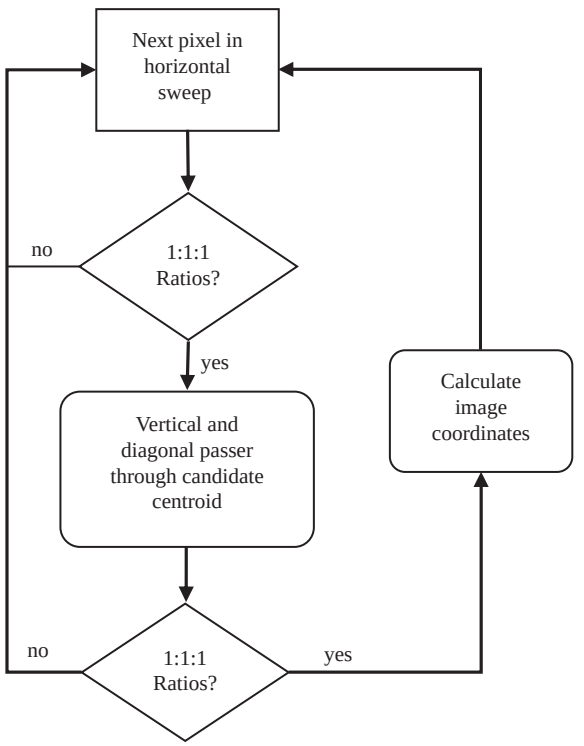


Figure 5 Flow chart of the image processing program



camera space can be estimated. Next, 3D coordinates of target can be calculated using the CSM-algorithm with the achieved relationship and the coordinates of target in two camera spaces. Finally, the joints are calculated by the inverse kinematics with the achieved 3D coordinates of target point to control the robot.

III. Mapping model

A. CSM model

As mentioned in the introduction, CSM is a visual control method which is based on estimating the relationship between the position of artificial features on the robot manipulator and their corresponding position in images taken by at least two cameras. The relationship (Skaar *et al.*, 1987) is described by the following expressions:

$$X_c = \left(C_1^2 + C_2^2 - C_3^2 - C_4^2\right) f_x(\theta) + 2(C_2C_3 + C_1C_4) f_y(\theta) + 2(C_2C_4 - C_1C_3) f_z(\theta) + C_5 \quad (1)$$

$$Y_c = 2(C_2C_3 - C_1C_4) f_x(\theta) + \left(C_1^2 - C_2^2 + C_3^2 - C_4^2\right) f_y(\theta) + 2(C_3C_4 + C_1C_2) f_z(\theta) + C_6 \quad (2)$$

where $C = [C_1, C_2, C_3, C_4, C_5, C_6]$ are referred to as "view parameters", and where (X_c, Y_c) represents the location of the center of artificial features, a positioned tool point, at the robot end-effector in 2D camera space. $f(\theta) = [f_x(\theta), f_y(\theta), f_z(\theta)]$ represents the nominal location of the center of artificial features of the robot end-effector in the 3D robot space which can be achieved by forward kinematics as follows.

The Denavit-Hartenberg (Denavit and Hartenberg, 1955) convention is widely used for defining frames of reference for describing the forward kinematics. A model of the IRB120 robot according to D-H conventions is given in Table I; the seventh axis refers to the D-H parameters between the end-effector and the designed fixture:

$$^{i-1}T_i = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i)\cos\alpha_i & \sin(\theta_i)\sin\alpha_i & a_i\cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i)\cos\alpha_i & -\cos(\theta_i)\sin\alpha_i & a_i\sin(\theta_i) \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

where $^{i-1}T_i$ represents the homogeneous transformation from $i-1$th frame to ith frame, the a_i, α_i, d_i, θ_i are parameters associated with link i and named as link length, link twist, link offset, and joint angle, respectively, (Denavit and Hartenberg, 1955).

Table I D-H parameters for ABB IRB 120 robot

Axis	θ	d	a	α
1	θ_1	d_1	0	α_1
2	θ_2	0	a_2	0
3	θ_3	0	a_3	α_3
4	θ_4	d_4	0	α_4
5	θ_5	0	0	α_5
6	θ_6	d_6	0	0
7	0	d_7	a_7	0

Combining the six coordinate frames, nominal forward kinematics of the IRB120 become:

$$T = {}^{0}T_1\,{}^{1}T_2\,{}^{2}T_3\,{}^{3}T_4\,{}^{4}T_5\,{}^{5}T_6\,{}^{6}T_7 \tag{4}$$

The coordinates of end-effector in robot space are obtained using the expression which follows:

$$P = [\,px, py, pz\,] = [f_x(\theta), f_y(\theta), f_z(\theta)] \tag{5}$$

where $\theta = [\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6]$ are the joints, $P = [\,px, py, pz\,]$ are the corresponding end-effector coordinates, $f_x(\theta)$ is the top element of matrix $T\left[p_x^*, p_y^*, p_z^*\right]$, where p_x^*, p_y^*, p_z^* are the (precisely known) coordinates of the feature- and tool-point with respect to the seventh frame and given by $T(1, 4)$, $f_y(\theta)$ and $f_z(\theta)$ are the second and third elements. The matrix T can be obtained using expression (4).

For convenience, expressions (1) and (2) are rewritten as follows:

$$X_c = g_x(f(\theta), C) \tag{6}$$

$$Y_c = g_y(f(\theta), C) \tag{7}$$

The view parameters C are estimated for each camera using the method of least squares with acquisition of a number of joints-space and camera-space coordinates, the view parameters $C = [C_1, C_2, C_3, C_4, C_5, C_6]$ are estimated independently for each camera by minimizing the following function over all C:

$$\mathcal{J}_1(C) = \sum_i \left(\omega_i\left(X_c^i - g_x(f_i(\theta), C)\right)^2 + \omega_i\left(Y_c^i - g_y(f_i(\theta), C)\right)^2\right) \tag{8}$$

where $\left(X_c^i, Y_c^i\right)$ represents the ith camera-space sample of the white circle center on the robot end-effector in 2D camera space. The function $f_i(\theta)$ represents the ith modeled function of a feature center on the robot end-effector in robot space. The coefficient ω_i refers to the relative weight given to each of the samples in order to take into account the approximate distance of the samples to the target (Skaar *et al.*, 1990; C'ardenas *et al.*, 2003).

An important step, which is known as flattening, is performed in order to increase the applicability of the orthographic model of expressions (1) and (2) (Gonzalez-Galvan *et al.*, 1997). The main contribution of this step is to reduce, in the calculations, the effect of perspective distortion, thereby fitting the camera-space data to the orthographic camera model. Z_r is the roughly anticipated value of Z-distance along each camera axis of the target points, average position in physical space. The modified samples (X_{c0}, Y_{c0}) are obtained from the original detected camera-space location (X_c, Y_c) as follows:

$$\begin{aligned} X_{c0} &= X_c \frac{Z_i}{Z_r} \\ Y_{c0} &= Y_c \frac{Z_i}{Z_r} \end{aligned} \tag{9}$$

where Z_i represents the Z-coordinate of the ith detected visual feature, whereas Z_r represents an approximate anticipated Z-coordinate of one manipulated feature at maneuver termination: $Z_i = Z_r + \Delta Z_i$

$$\Delta Z_i = c_{31}\Delta x^i + c_{32}\Delta y^i + c_{33}\Delta z^i$$

$$\begin{aligned} c_{31} &= 2(e_1e_3 + e_0e_2) \\ c_{32} &= 2(e_2e_3 - e_0e_1)c \\ {}_{33} &= e_0^2 - e_1^2 - e_2^2 + e_3^2 \end{aligned}$$

$$e_{i-1} = \frac{C_i}{\left[C_1^2 + C_2^2 + C_3^2 + C_4^2\right]^{1/2}}, \quad i = 1, 2, 3, 4$$

where the $(\Delta x^i, \Delta y^i, \Delta z^i)$ can be calculated by expression (5) representing the ith-pose coordinates of sampled points at the robot end-effector in 3D physical space.

Sample points can be flattened by expression (9), then new view parameters C can be obtained, followed by more flattening. Continuous iteration occurs until the view parameters C converge.

When the view parameters C and coordinates of target point in two camera space are given, the joint rotations associated with 3D coordinates of the target point $f(\theta) = (f_x(\theta), f_y(\theta), f_z(\theta))$ can be calculated by minimizing the following function:

$$\begin{aligned} \mathcal{J}_2(f(\theta)) = &\left(\left(x_{c1}^t - g_x(f(\theta), C)\right) + \left(y_{c1}^t - g_y(f(\theta), C)\right)\right. \\ &\left. + \left(x_{c2}^t - g_x(f(\theta), C)\right) + \left(y_{c2}^t - g_y(f(\theta), C)\right)\right) \end{aligned} \tag{10}$$

where $\left(x_{c1}^t, y_{c1}^t\right)$ are the coordinates of a target point in the first camera space, and $\left(x_{c2}^t, y_{c2}^t\right)$ are the coordinates of the same target point in the second camera space.

Then the joints $\theta = [\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6]$ are calculated by the inverse kinematics with the achieved 3D coordinates of target point $f(\theta) = (f_x(\theta), f_y(\theta), f_z(\theta))$ to control the robot. In this paper, we adopt a numerical solution for the inverse kinematics. The result within some convergence criterion for dP below, can be achieved after a number of iterations. This solution minimizes the change in joint coordinates, from a given, starting pose, in a least-squares sense.

Provided that it is small the total positional error $dP = T - P$ of the robot due to the joint can be approximated as:

$$\begin{aligned} d\theta &= (\mathcal{J}'\mathcal{J})^{-1}\mathcal{J}'dP \\ \theta &= \theta + d\theta \end{aligned} \tag{11}$$

where $d\theta = [d\theta_1, d\theta_2, d\theta_3, d\theta_4, d\theta_5, d\theta_6]'$ is small errors in the joints, $T = [tx, ty, tz\,]$ is the desired position and P can be got by expression (5), $\mathcal{J}$ is the Jacobian matrix [3, 6]. Use of the pseudo inverse permits application of varying numbers of the robot's full complement of six degrees of freedom (DOF). For the point-positioning tests below, the minimum of DOF is three.

B. The VW-CSM algorithm

In order to improve the convergence speed, a varying weighting scheme for end-effector samples has been proposed. In the conventional method, the reciprocal of the Euclidean distance to the target is defined as the weight. The use of this weighting scheme has improved the positioning accuracy and real-time performance effectively, but can be further improved. In addition, the relationship between positioning using the CSM algorithm and weight is not clear. In order to investigate improvements in our weighting scheme, four different varying weighting formulas with variable coefficients have been defined as follows:

$$\omega_1 = k/dis \tag{12}$$

$$\omega_2 = k + 1/dis \tag{13}$$

$$\omega_3 = 1/(dis)^k \tag{14}$$

$$\omega_4 = k/(dis)^k \tag{15}$$

where *dis* is the currently estimated Euclidean distance from any given sample point to the target point, and k is the varying parameter which affects the weight of points which are closer to the target point. The weights can be brought into equation (8) as follows:

$$\mathfrak{J}_1(C) = \sum_i \left(\omega_k \left(x_c^i - g_x(f_i(\theta), C) \right)^2 + \omega_k \left(y_c^i - g_y(f_i(\theta), C) \right)^2 \right) \tag{16}$$

where four different values of k(1, 2, 3, 4) have been tested, where (x_c^i, y_c^i) represents the ith coordinates of the white circle center on the robot end-effector in 2D camera space, where $f_i(\theta)$ represents the nominal ith coordinates of the white circle center at the robot end-effector in world space, and where C is the vector of view parameters.

IV. Robust positioning strategy

Combined with Figures 1 and 6, the steps of robot positioning under the developed system are summarized as follows:

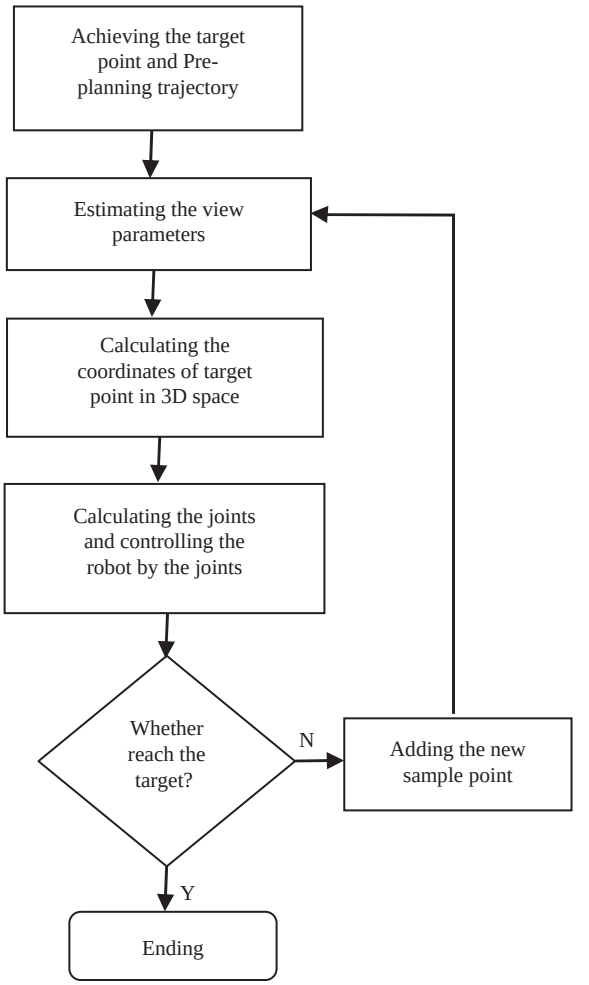
1 The target point should be found first. In general, the target point coordinates in two camera spaces are achieved by the image processing program. In real application, the target points are recognized by shooting a laser beam from a laser pointer. In this paper, the experiments are completed by setting the robot in an arbitrary position and orientation, then the artificial feature on the surface of the designed fixture is as the target, only in this way, the positioning error can be accurately calculated.

2 Next, we create a pre-planned trajectory. If just using one circle as the feature point, no less than three positions should be sampled by two cameras, if using the three black circles as the feature points, only one position should be sampled by two cameras at beginning. Then the 3D coordinates of feature points on the end-effector of the robot can be achieved using the forward kinematics model of expression (4). The 2D coordinates of feature points in camera space can be detected using the image processing algorithm, the 2D coordinates of feature points are flattened by expression (9) and the view parameters C are estimated by minimizing equation (8).

3 The 3D coordinates of any given target point are calculated with the view parameters C and target point coordinates in the two camera space by minimizing equation (10), After this, robot movement is controlled on the joint level to the position with θ is achieved according to results from the inverse kinematics in expression (11).

4 Judge whether the robot reached the target point with adequate accuracy (which can be assessed in these tests, because the true or correct target is known); if not, go to step (2) and then estimate the next view parameters C with addition of the newly sampled positions and with variable weight ω.

Figure 6 Flow chart of the control strategy



Individual cameras "camera-space objectives" can be taken in such a way as to control six-DOF, rigid-body position/orientation, but in the present study it is convenient to instead use a three-DOF, 3D point-positioning task. The reason lies in the approach used for this study in order to evaluate the weighting schemes. Rather than applying structured light, we instead place the robot into an arbitrary pose. One particular point, designated by a circular mark, is detected with high precision in each of two uncalibrated cameras. This point then becomes the target for the maneuver. The advantage of this target is that, when we hold all but the first three joint rotations fixed, we know in advance, independently of robot calibration, the values of the first three joint rotations that produce a perfect pose. Calculation of the error in producing this pose based upon the incoming sequence of image samples leading up to this terminus is conveniently and precisely calculated using the nominal robot kinematics.

V. Experiment and discussion

Real-time performance and robust accuracy of positioning are the two most important measurement indicators of vision-based robot positioning. The real-time performance and positioning accuracy of the industrial robot positioning system are tested and verified.

A. The positioning error in 2D camera space and in 3D space with one feature

The accuracy of the CSM-based industrial robot system has been verified by combining the accuracy analysis in 2D camera space and 3D physical space. A group of sample points which are randomly distributed in the camera space has been collected to estimate the view parameters C; we sample the coordinates of these points in camera space and obtained view parameters C by minimizing equation (8). The distribution of original points and best-fit points is shown in Figures 7 and 8, which displays that the fits have a good coincidence with the original points, with the average error for each best fit given in rows 2 and 3 of Table III.

Figure 7 The distribution of original points and calculated points in camera 1

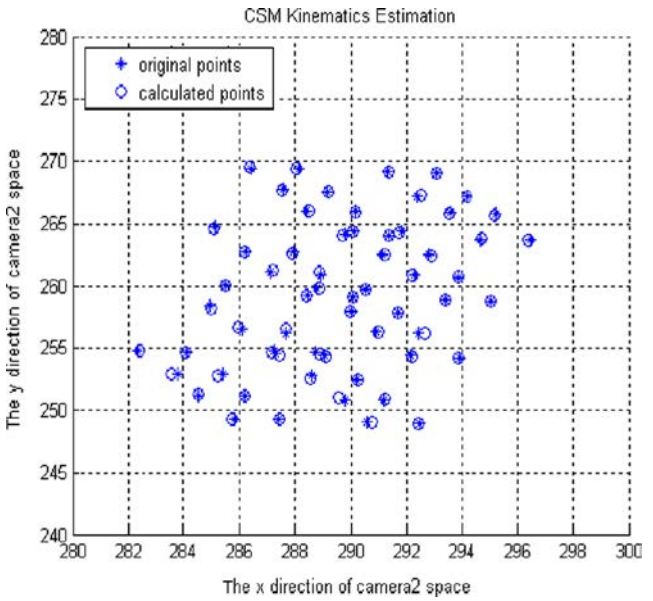
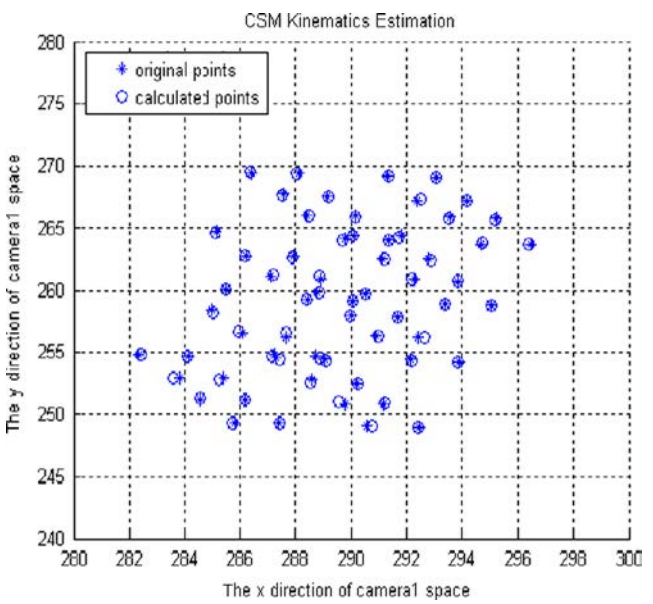


Figure 8 The distribution of original points and calculated points in camera 2



This experimental result verifies the high accuracy in camera space. The closeness of the best fits in camera space also indicates the predictive camera-space accuracy for new joint poses in the vicinity of the joint poses used to create the fits.

Another experiment has been taken to verify the accuracy in 3D space. Again 21 sample points randomly distributed in 3D space have been collected. That is to say, 21 groups of coordinates of sample points in 3D space and in two different camera spaces have been collected. To decrease the effect of random error of sample points, the center of the three black circles has been used to replace a single cue which is named "virtual points". The procedure takes advantage of the linear mapping from 3D physical space into 2D camera space for a highly localized cluster of three points. In total, 21 "virtual points" in 3D space and two camera spaces have been collected. Then, making one of the 21 virtual points as our target, each in turn, we use the other 20 points to estimate the view parameters C. By minimizing equation (10), the 3D coordinates of virtual target point can be calculated. The error between the calculated coordinates of target points and the measured coordinates of target points is depicted in Table II, and the distribution of calculated points and measured points is shown in Figure 9. This experimental result verifies the high accuracy of positioning in 3D physical space.

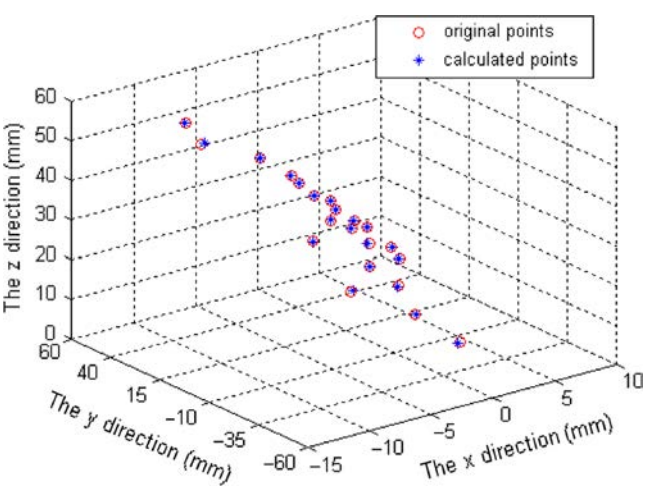
B. The positioning with different groups of feature points

The task of robot positioning has been accomplished with one white circle being used as the feature point to estimate the

Table II The positioning error in 3D physical space with one virtual feature

Set name	The error (Euclidean distance: mm)				
Point 1-5	0.3367	0.2178	0.2952	0.4847	0.4591
Point 6-10	0.5522	0.4852	0.4888	0.0962	0.0670
Point 11-15	0.0911	0.5136	0.3972	0.2021	0.2910
Point 15-20	0.3631	0.2297	0.1191	0.1464	0.1164
Point 21	0.3234				
Mean error			0.2989		

Figure 9 The distribution of original points and calculated points in 3D space



view parameters C under the CSM-based robot servoing positioning system with the proposed control strategy. The positioning accuracy in 3D physical space has been calculated. The average errors in x-, y-, and z-directions are displayed in row 4 of Table III. The trajectory of error changing with number of iterations is shown with Figure 10. The experimental result shows that the CSM-based robot positioning system has a high accuracy, when a large number of sample poses has been used.

Next, the positioning has been completed by using the three black circles as a single virtual cue in order to estimate the view parameters C and controlling the white circle to the target position. The experimental results are shown in Figure 11 and row 5 of Table III; the error in the x-direction finally tends to 0.12 mm and to 0.34 and 0.13 mm of error, respectively, in each of the y- and z-directions; the Euclidean error between robot final position and target position tends to 0.36 mm. Another experiment has been done also with the three black circles, individually, to create total of 60 samples per camera, in order to estimate the view parameters C, and predict the white circle as the target position. Then making one white circle of the 21 groups of points as targets, each in turn, the other 20 groups of black circles are used to estimate the view parameters C. By minimizing equation (10), the 3D coordinates of each target point can be calculated. The error between the calculated coordinates of target points and the actual coordinates of target points is depicted in Table IV.

C. The discussion of positioning accuracy and real-time performance

According to the experiments conducted above the accuracy of the CSM-based robot positioning system has been verified. The high accuracy of the system is confirmed with high accuracy in 2D camera space and in 3D physical space. In order to verify the accuracy in 3D space, the point-positioning task under this system has been applied using information from one feature point and also using information from three feature points on the end effector. The number of iterations in the task of positioning with only one feature point is much more than the number using three feature points; that is to say, the case with more feature points requires fewer image samples and therefore costs less time to complete the positioning. In addition,

Table III The error magnitudes

Set name	*x*-direction error	*y*-direction error	*z*-direction error	Euclidean distance
Error in camera 1 (pixel)	0.065	0.069		0.106
Error in camera 2 (pixel)	0.081	0.093		0.137
First positioning error in 3D space (mm)	0.07	0.29	0.15	0.33
Second positioning error in 3D space (mm)	0.12	0.34	0.13	0.36

Figure 10 Robot positioning with one feature points

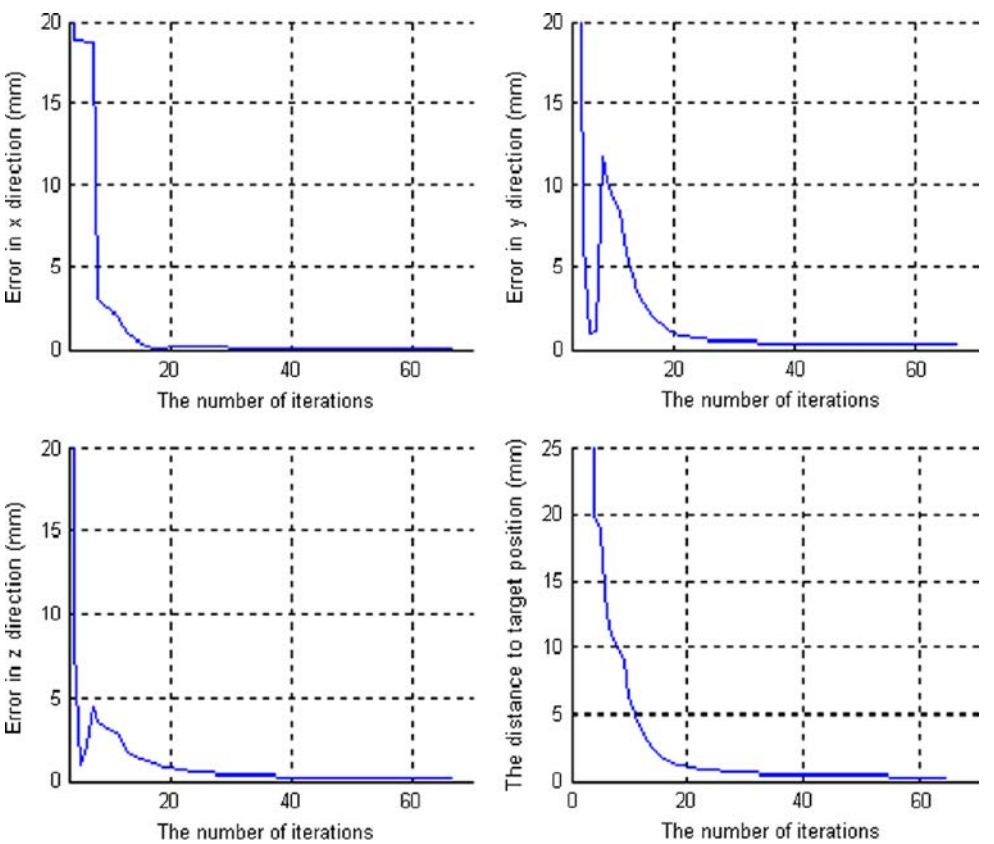


Figure 11 Robot positioning with three feature points

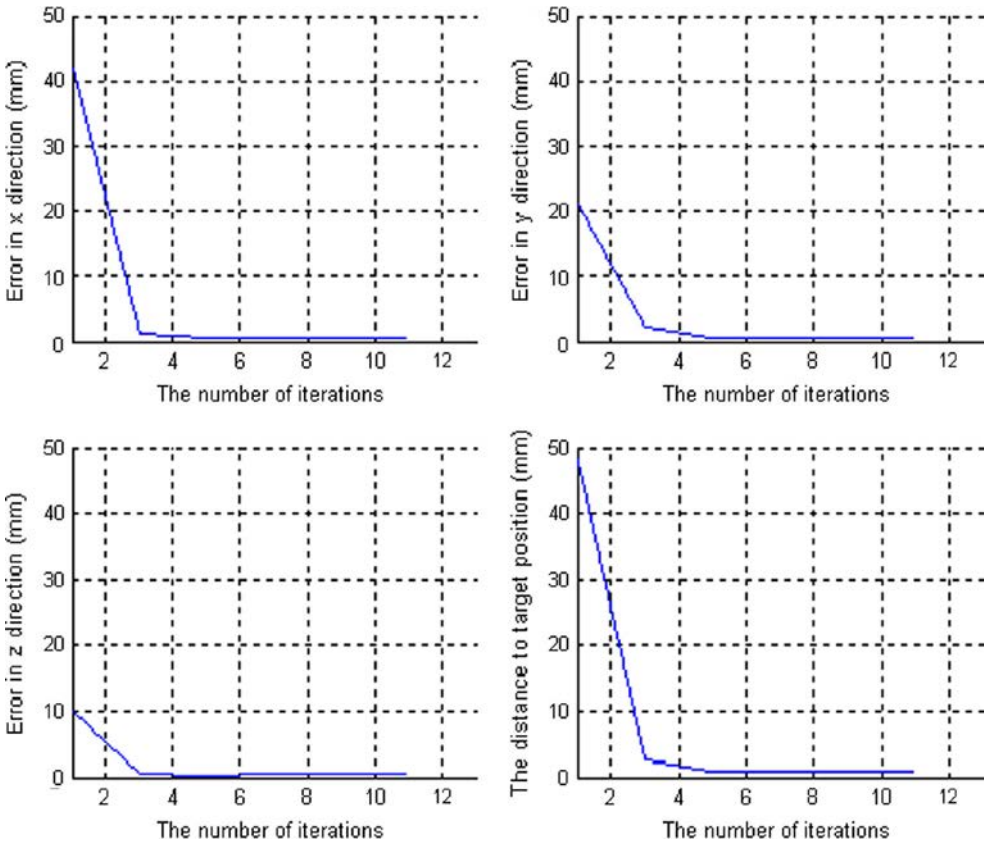


Table IV The error in 3D space with three features

Set name	The error (mm)				
Point 1-5	0.3912	0.4135	0.3786	0.2980	0.3182
Point 6-10	0.4722	0.4517	0.4042	0.4866	0.3119
Point 11-15	0.4484	0.3452	0.2531	0.3500	0.2076
Point 15-20	0.4729	0.3260	0.3209	0.3092	0.2725
Point 21	0.2042				
Mean error			0.3541		

the positioning error has been compared between positioning with only one "virtual" feature point – three closely spaced feature points averaged in camera space – and the same three black feature points treated separately, and the result shows that the former case demonstrates higher accuracy, as indicated in the results in Tables II and IV. The number of iterations required for a given terminal accuracy can be decreased by adding end-effector feature points for purposes of estimation.

Before analyzing the real-time performance of the developed system, the run time of the main parts of our system has been determined. The average time cost of executing the image processing program and CSM program is 84 and 285 ms, respectively. A single iteration requires about 400 milliseconds, including the image processing program, CSM program, robot control program and some other tasks. The convergence speed, to a given level of accuracy, with three feature points is much faster than the speed in the experiment with only one feature point, which can be seen clearly by comparing Figures 10 and 11. The processing time of this system can reach approximately 5 s for a requirement of high accuracy of 0.3 mm when using the method with three feature points, however, the real-time performance of this system can reach 2 s if such high accuracy is not needed – for example if just1 mm is sufficient resolution, which suffices for many applications. So there is a tradeoff between real-time performance and accuracy.

D. The experiment and discussion of positioning error using the VW-CSM algorithm

This section focuses on the relationship between positioning accuracy and weights that are introduced in equation (8). Four different weighting schemes have been defined as expressions (12) through (15). The four different weights all have a variable parameter and the impact on positioning error caused by varying these parameters has been investigated experimentally. A group of sample points that is randomly distributed in camera space has been collected and the white circle was used as the feature point in order to estimate the view parameters C using equation (8) with the four different functional forms for varying weights; the average positioning error has been calculated and shown in Figures 12-15 and Table V.

According to Table V, the accuracy of positioning with the four defined weights is better than the case with equal weight. However, the accuracy of positioning changes only slightly with these four defined weights relative to the equal-weight case. According to Figures 12 and 13, the changing coefficient k has no effect on the average error of point positioning. But the positioning accuracy with ω_1 is higher than that

Figure 12 The average error of positioning with varying ω_1

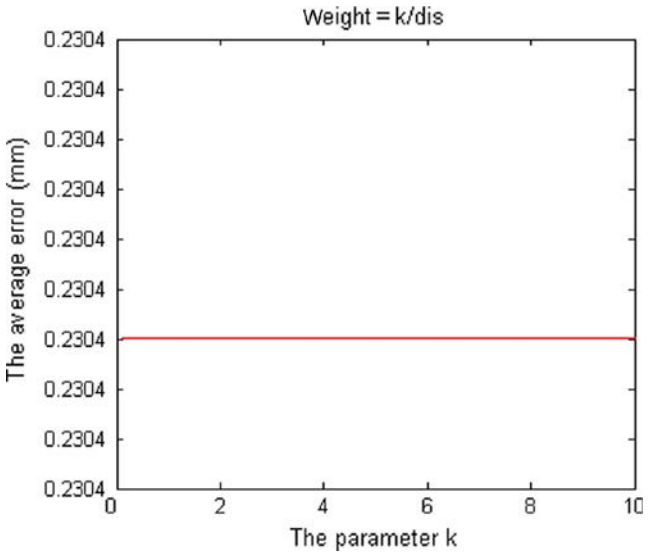


Figure 13 The average error of positioning with varying ω_2

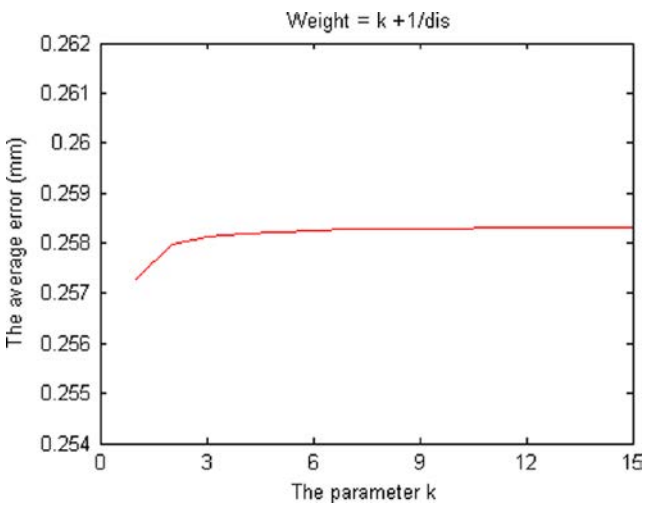


Figure 14 The average error of positioning with varying ω_3

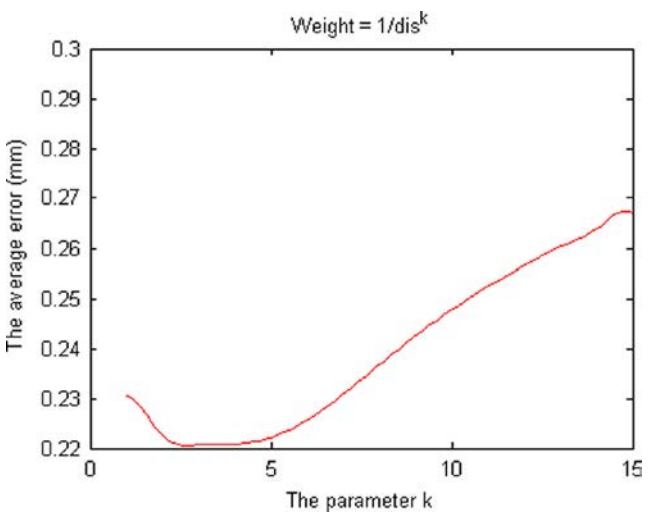


Figure 15 The average error of positioning with varying ω_4

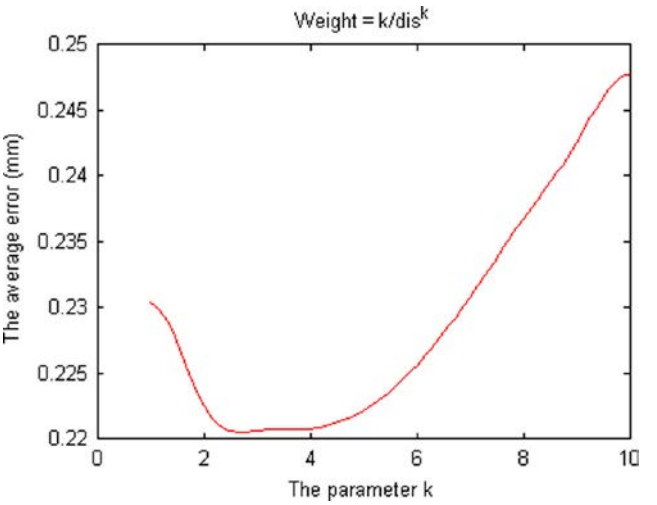


Table V The positioning error with different weights

Set name	The smallest average error of positioning (mm)
Equal weight	0.4691
Weight 1	0.2303
Weight 2	0.2572
Weight 3	0.2206
Weight 4	0.2203

with ω_2, as can be seen by comparing Figures 12 and 13, and Table V. In Figures 14 and 15, the smallest error is achieved when the coefficient k is set to three, and the error grows when the coefficient k exceeds three.

E. The experiment and discussion of real-time performance using VW-CSM algorithm

Through all the above experiments and results, the system demonstrates high positioning accuracy, but the real-time performance is not satisfactory. Some analysis concerning the CSM algorithm has been undertaken to find a way to decrease the time cost of robot positioning. During the process of robot positioning, new end-effector sample points are incorporated by the algorithm. The time cost adding these has been calculated to establish the degree to which this affects real-time performance, and the result is shown in Figure 16.

According to Figure 16, the time cost increases slowly with more and more end-effector sample junctures added. Therefore, the number of sample points has relatively little impact on the real-time performance. The most effective way to improve real-time performance is to reduce the number of images and iterations. In order to improve the convergence speed, the weight has been introduced in equation (8) and four different kinds varying weights have been defined in expression (12)-(15).

The positioning has been completed by using the three black circles to estimate the view parameters C by minimizing equation (8) with the four different varying weights, and controlling the white circle to the target position under the CSM-based industrial robot positioning system with the proposed control strategy. The iterations of positioning with different weights has been listed in Table VI.

Figure 16 The time cost of CSM program with changing sample points

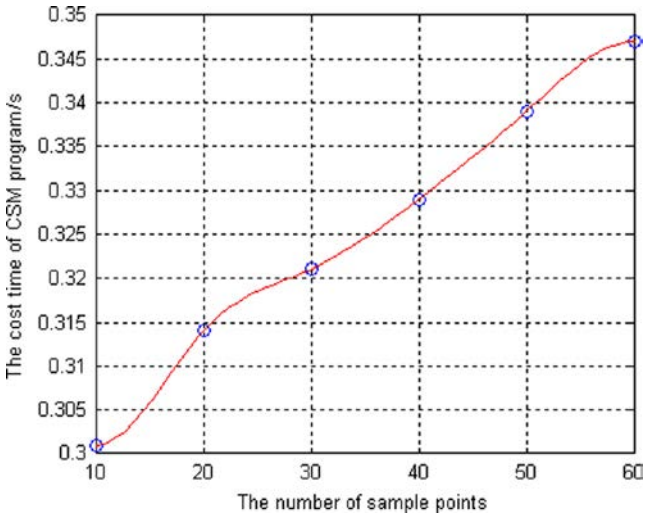


Table VI The effectiveness of different weights

Set name	The numbers of iteration
Equal weight	12
ω_1	7
ω_2	8
ω_3	5
ω_4	4

The task of positioning with ω_4 only costs four iterations which is much better than other weights. What is more, the relationship between real-time performance and varying weights has been explored. According to Figures 17 and 18, the time of positioning continues to decrease with the increasing of coefficient k. Combining Figures 15 and 18, the real-time performance can be improved with little sacrifice of positioning accuracy.

VI. Conclusions

In this paper an efficient uncalibrated industrial robot real-time, visually guided positioning system has been developed.

Figure 17 The time cost of positioning with varying ω_1 and ω_2

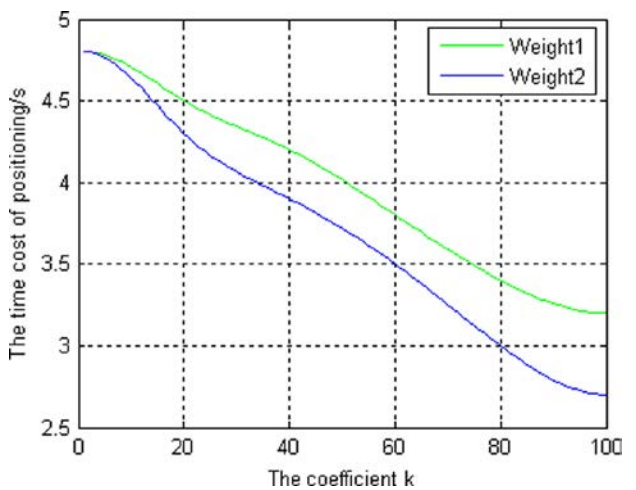
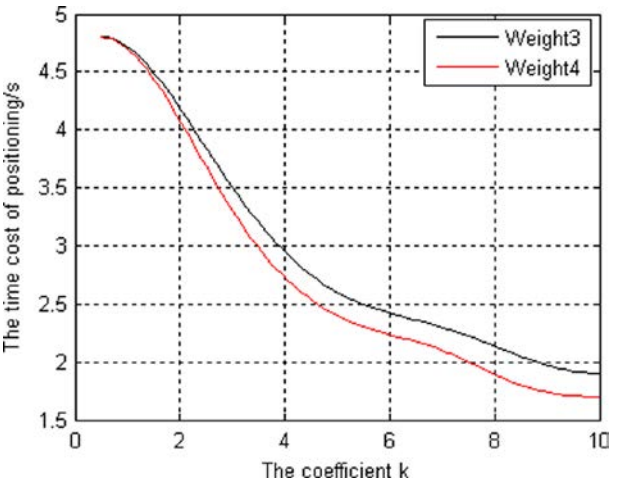


Figure 18 The time cost of positioning with varying ω_3 and ω_4



The positioning error of the developed system can reach 0.2 mm in 3D robot space without calibration of the two stationary cameras. The real-time performance and the accuracy of the system have been improved with adoption of the VW-CSM algorithm. These results prove that the developed system can satisfy the requirements of industrial applications and can be widely used in the field of industrial robots.

The high accuracy of the developed industrial robot positioning system has been verified by examining the accuracy analysis in 2D camera space and 3D physical space. And also the accuracy and real-time performance of the developed system have been compared by positioning with different combinations of feature points used for estimation. The result shows that using more features results in improved real-time performance, but the accuracy becomes lower. The VW-CSM algorithm has been proposed, which takes varying weights on vision samples of the robot end effector. The positioning error will decreases at first and then increases with further skewing of the weight. And the real-time performance also improves with more highly skewed weighting.

References

C'ardenas, A., Seelinger, M., Goodwine, B. and Skaar, S. (2003), "Vision-based control of a mobile base and on-board arm", *Int. J. Robotics Res.*, Vol. 9 No. 22, pp. 677-698.

Corke, P.I. and Hutchinson, S. (2001), "A new partitioned approach to image based visual servo control", *IEEE Trans. Robot. Automation*, Vol. 17 No. 4, pp. 507-515.

Corke, P.I., Spindler, F. and Chaumette, F. (2009), "Combining Cartesian and polar coordinates in IBVS", *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5962-5967.

Denavit, J. and Hartenberg, R.S. (1955), "A kinematic notation for lower pair mechanisms based on matrices", *J. Appl. Mech. ASME*, Vol. 22, pp. 215-221.

Gonzalez-Galvan, E.J., Skaar, S.B., Korde, U.A. and Chen, W.Z. (1997), "Application of a precision enhancing measure in 3-D rigid-body positioning using camera-space manipulation", *Int. J. Robotics Res.*, Vol. 16 No. 2, pp. 240-257.

Hutchinson, S., Hager, G.D. and Corke, P.I. (1996), "A tutorial on visual servo control", *IEEE Trans. Robot. Automation*, Vol. 12 No. 5, pp. 651-670.

Kinbara, I., Komadda, S. and Hirai, J. (2006), "Visual servo of active cameras and manipulators by time delay compensation of image features with simple on-line calibration", *SICE-ICASE International Joint Conference*, pp. 5317-5322.

Liu, Y.H. and Xi, N. (2011), "Low-cost and automated calibration method for joint offset of industrial robot using single-point constraint", *Industrial Robot: An International Journal*, Vol. 38 No. 6, pp. 577-584.

Liu, Y.H., Wang, H., Wang, C. and Lam, K.K. (2006), "Uncalibrated visual servoing of robots using a depth independent interaction matrix", *IEEE Trans. Robot.*, Vol. 22 No. 4, pp. 804-817.

Piepmeier, J.A., McMurray, G.V. and Lipkin, H. (2004), "Uncalibrated dynamic visual servoing", *IEEE Trans. Robot. Automation*, Vol. 20 No. 1, pp. 143-147.

Rendon-Mancha, J.M., Cardenas, A. and Garcia, M.A. (2010), "Robot positioning using camera-space manipulation with a linear camera model", *IEEE Trans. Robot.*, Vol. 26 No. 4, pp. 726-733.

Skaar, S.B. (1996), "Efficient camera-space manipulation", *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, pp. 3407-3412.

Skaar, S.B., Brockman, W.H. and Hanson, R. (1987), "Camera space manipulation", *Int. J. Robotics Res.*, Vol. 6 No. 4, pp. 20-32.

Skaar, S.B., Brockman, W.H. and Jang, W.S. (1990), "Three dimensional camera space manipulation", *Int. J. Robotics Res.*, Vol. 9 No. 4, pp. 22-39.

Tahri, O. and Chaumette, F. (2005), "Point-based and region-based image moments for visual servoing of planar objects", *IEEE Trans. Robot.*, Vol. 21 No. 6, pp. 1116-1127.

Corresponding author

Yong Liu can be contacted at: liuy1602@msu.edu