

Uncalibrated Stereo Visual Servoing for Manipulators using Virtual Impedance Control

Caixia Cai*, Nikhil Somani*, Suraj Nair*, Dario Mendoza and Alois Knoll*

* Robotics and Embedded Systems Lab, Department of Computer Science

{caica, somani,nair,knoll}@in.tum.de, Technische Universität München, Germany

Abstract—In this paper, we present an uncalibrated position-based fixed-camera Visual Servoing for robot manipulators, where the goal is to track the 3D position and orientation of the target. The stereo system with 2 USB cameras is uncalibrated with respect to the robot base frame and the transformation between them is estimated on-line while performing the task. Dynamic impedance control is designed to generate a dynamic trajectory for the robot manipulator considering the dynamic environment constraints, such as: robot singularities avoidance and (self-/obstacle-) collision avoidance. Experiments have been carried out to verify performance of the proposed system on a real industrial robot, where the calibration estimation process and handling of all uncertainties in the environment are demonstrated. Moreover the uncalibrated stereo camera system can be manually moved while performing the task in order to obtain a clearer view and the re-calibration is performed automatically and on-line.

I. INTRODUCTION

Visual Servoing (VS) is an approach to control motion of a robot manipulator using visual feedback signals from a vision system. Over the last decade, there has been significant growth in practical applications of Visual servoing. The use of a camera in closed-loop control schemes seems to be efficient, and many control laws and modeling have been proposed [1]. However, application of typical VS to large robot motions, i.e., industrial manipulators can result in unstable trajectories, due to: (i) the end-effector and the target object can move beyond the visual scope of the camera in the eye-to-hand robot system; (ii) the robot being commanded to move beyond its mechanical limits, or beyond its actuation limits; or (iii) the robot arm physically colliding into other objects within its workspace. This work aims to simultaneously address all three of the above challenges for visual servoing of large robot motions.

Visual servoing requires an object in the field of view of the camera, in order to use the feature error to control the robot [2]. Otherwise, the virtual link is broken and the control loop cannot continue to be closed. One solution is using two or more cameras to be fixed in the workspace in order to guarantee a panoramic sight of the workspace [1]. This approach requires calibration of the camera w.r.t the robot. However, in practice, due to the presence of uncertainties in the camera calibration parameters, consistent and accurate calibration is difficult and even small changes forces a re-calibration. Hence an uncalibrated visual servoing approach is beneficial and often necessary in many circumstances.

Uncalibrated Visual Servoing approaches have been proposed in [3], [4], [5] and [6]. But all these mentioned methods are applicable when there are no obstacles in the workspace.

Hosoda et al. [7] present an obstacle avoidance method by using a trajectory generator and an uncalibrated visual servoing controller. It does not need priori knowledge of the robot and the cameras, but this method is only suitable for static targets. Qing shan et al.[8] presented a uncalibrated visual servoing approach with obstacle avoidance, which ensures that the robot avoids the obstacle safely and tracks the target as accurately as possible. But this approach works only when the cameras are fixed and the obstacle is static.

This paper presents an implementation of uncalibrated, position-based stereo Visual Servoing with obstacle avoidance. The system is initially uncalibrated but the intrinsic camera parameters are known in the stereo system, which is a reasonable assumption since the camera intrinsics don't usually change. The transformation matrix between the stereo camera space and the base of a StaeubliTX90 robot is estimated on-line using real operating data. The advantage of this system is that it can avoid moving obstacles including the Humans safely during tracking moving targets. The controller has been designed to be able to control the dynamics of the robot. Virtual Impedance control is used to represent the dynamic environment constraints, such as: robot singularities avoidance and (self-/obstacle-) collision avoidance. Moreover the uncalibrated stereo camera system can be manually moved during performing the task in order to keep the target in the field of vision or avoid occlusion, and the re-calibration is performed automatically and on-line.

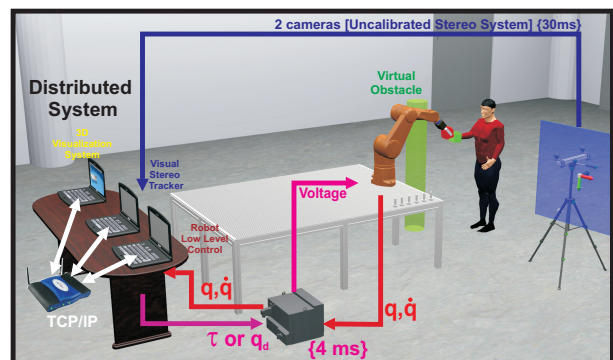


Fig. 1: Description of robotic experimental setup.

This paper is organized as follows: In Section II we introduce a stereo vision model and describe how the uncalibrated stereo system works. This model will be used in the Section III to design an virtual impedance control to generate the

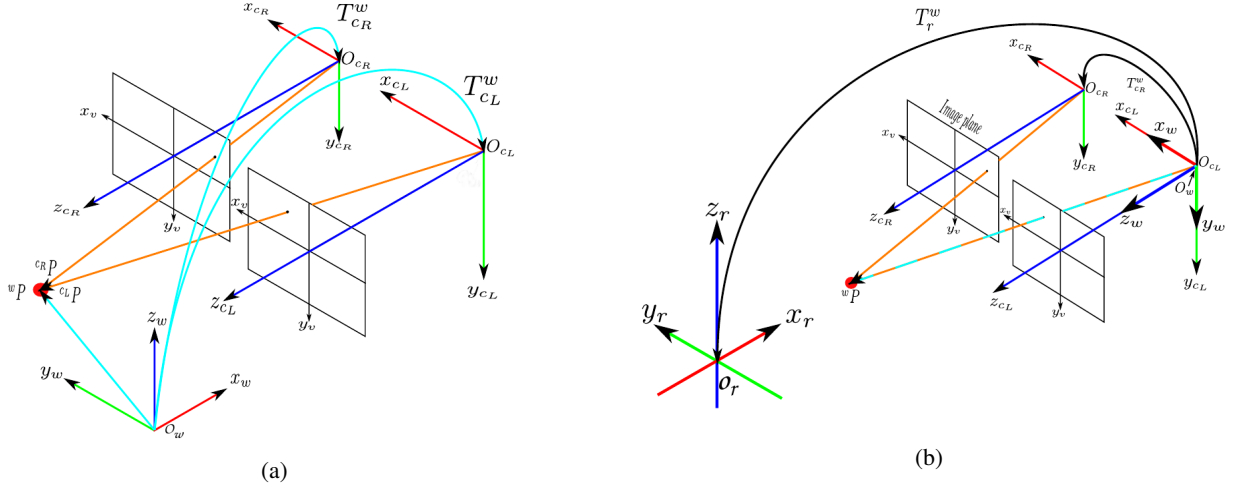


Fig. 2: The Stereo Camera Model: (a) shows a 3D point projects in two cameras, (b) puts the world coordinate frame on the left camera frame point.

robot trajectory dynamically. Section IV presents a real-world experiment (as illustrated in Fig. 1) and shows the results obtained in a dynamic environment. Finally, Section V draws the conclusions and presents directions for future work.

II. UNCALIBRATED STEREO TRACKING SYSTEM

A. 3D Recovery from Stereo System

In a two camera setup, each camera gives a different 3D back projection line. These two back projection lines usually coincide at exactly one point (Fig. 2a). So, given a stereo setup it is possible to find the 3D position of a point by observing its position in two different cameras[9].

For two cameras with two projection matrices P_1 and P_2 we have

$$\begin{bmatrix} a_1 \\ b_1 \\ w_1 \end{bmatrix} = P_1 \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad \begin{bmatrix} a_2 \\ b_2 \\ w_2 \end{bmatrix} = P_2 \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (1)$$

We know the observed image points in each camera x_1 , y_1 , x_2 and y_2 , given by a_1/w_1 , b_1/w_1 , a_2/w_2 and b_2/w_2 respectively, but we cannot assume that $w_{1,2} = 1$. We wish to solve for $X_w = [X, Y, Z]^T$, since this is the 3D position in the world frame.

Multiplying out in terms of the P_i , and substituting into $x_i w_i = a_i$ and $y_i w_i = b_i$ gives:

$$\begin{aligned} X x_i p_{31}^i + Y x_i p_{32}^i + Z x_i p_{33}^i + x_i p_{34}^i &= X p_{11}^i + Y p_{12}^i + Z p_{13}^i + p_{14}^i \\ X y_i p_{31}^i + Y y_i p_{32}^i + Z y_i p_{33}^i + y_i p_{34}^i &= X p_{21}^i + Y p_{22}^i + Z p_{23}^i + p_{24}^i \end{aligned} \quad (2)$$

where i is 1 or 2, and p_{11}^i denotes element 1,1 in P_i .

Since we want to solve this for $X_w = [X, Y, Z]^T$, these terms are collected on the left and other terms on the right, and the equation is then rewritten as:

$$A X_w = B \quad (3)$$

where

$$A = \begin{bmatrix} x_1 p_{31}^1 - p_{11}^1 & x_1 p_{32}^1 - p_{12}^1 & x_1 p_{33}^1 - p_{13}^1 \\ y_1 p_{31}^1 - p_{21}^1 & y_1 p_{32}^1 - p_{22}^1 & y_1 p_{33}^1 - p_{23}^1 \\ x_2 p_{31}^2 - p_{11}^2 & x_2 p_{32}^2 - p_{12}^2 & x_2 p_{33}^2 - p_{13}^2 \\ y_2 p_{31}^2 - p_{21}^2 & y_2 p_{32}^2 - p_{22}^2 & y_2 p_{33}^2 - p_{23}^2 \end{bmatrix} \quad (4)$$

and

$$B = \begin{bmatrix} p_{14}^1 - x_1 p_{34}^1 \\ p_{24}^1 - y_1 p_{34}^1 \\ p_{14}^2 - x_2 p_{34}^2 \\ p_{24}^2 - y_2 p_{34}^2 \end{bmatrix} \quad (5)$$

We need to solve this for X_w . Since A is not square, so the pseudo inverse is used:

$$X_w = (A^T A)^{-1} A^T B \quad (6)$$

B. Stereo Vision Model

If we put the world coordinate frame O_w at the left camera origin point O_{c_L} , see Fig.2b, then given the relation between right camera and left camera (defined by the orientation matrix $R_R^L \in SO(3)$ and the translation vector $t_R^L \in \mathbb{R}^{3 \times 1}$), we can define the projection matrices P_{c_L} and $P_{c_R} \in \mathbb{R}^{3 \times 4}$ for each camera as

$$P_{c_L} = K_L [I_{3 \times 3} \quad 0_{3 \times 1}] \quad P_{c_R} = K_R [R_R^L \quad t_R^L] \quad (7)$$

where K_L and K_R are the intrinsic camera matrices of the left and right cameras respectively¹.

Then, defining the observed image points in each camera as $p_L = [x_L, y_L]^T$, $p_R = [x_R, y_R]^T$, we can use triangulation [9] to compute the *relative* position $X_{c_L} = [x_c, y_c, z_c]^T$ with respect to the left camera O_{c_L} , which is shown in (6).

In order to control the robot, we need to compute the 3D position with respect to the robot base coordinate frame O_r . By knowing the transformation between O_w (the same as O_{c_L})

¹These parameters can be computed off-line.

and $O_r, T_r^w = T_r^{cL} = [R_r^{cL} \quad t_r^{cL}]$, we can get the robot end-effector position

$$X_{ef} = X_r = R_r^{cL} X_{cL} + t_r^{cL} \quad (8)$$

C. On-line Calibration of the Stereo System

We define this system as uncalibrated because we not only assume that the calibration of the stereo vision system (left camera O_{cL}) with respect to the robot base frame (O_r) is unknown, but we also consider the possibility of *on-line modification* of the parameters that define this relationship (e.g. R_r^{cL}). In our system, we estimate the transformation T_r^{cL} on-line without stopping robot during the task execution using the real-time information generated by the robot.

Finding the optimal/best rotation and translation between two sets of corresponding 3D point data, such that they are aligned, is a common problem. An illustration of the problem is shown below for the simplest case of 3 corresponding points (the minimum required points to solve).

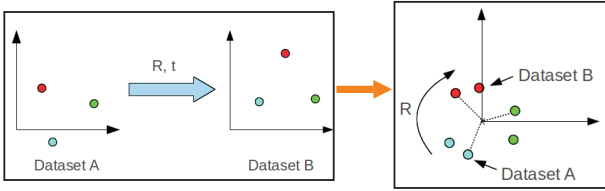


Fig. 3: Two sets of corresponding 3D point data A and B

Let the given 3D datasets A and B be represented in two coordinate frames. We want to find the best rotation R and translation t that will align the points in dataset A to dataset B.

$$P_B^i = R P_A^i + t \quad (9)$$

where $i = 1, 2, \dots, n$, ($n \geq 3$), $P = [x, y, z]^T$, P_A and P_B are points in dataset A and B respectively.

Finding the optimal rigid transformation matrix can be broken down into the following steps:

1) *Find the centroids of both dataset*: This bit is easy, since the centroids $C_{A,B}$ are just the average points and can be calculated as follows:

$$C_A = \frac{1}{n} \sum_{i=1}^n P_A^i \quad C_B = \frac{1}{n} \sum_{i=1}^n P_B^i$$

2) *Estimation of Base Orientation R*: There are several ways of finding optimal rotations between points. The easiest yet accurate way is using Singular Value Decomposition (SVD), as discussed in [10]. Define a 3×3 matrix M as

$$M = \sum_{i=1}^n (P_A^i - C_A) (P_B^i - C_B)^T \quad (10)$$

Then a least-squares fit of the rotation matrix can be written as

$$R = U \text{diag}(1, 1, \det(UV^T)) V^T \quad (11)$$

where $(U, S, V) = \text{svd}(M)$.

3) *Compute the translation t*: After R is obtained, t can be solved as:

$$t = C_B - R C_A \quad (12)$$

In our case, P_B is defined as a set of end-effector 3D positions X_r with respect to robot base frame O_r , while P_A denotes a set of the corresponding position vectors X_{cL} in stereo camera coordinate frame O_{cL} .

This on-line estimation of the transformation will be used in the next section to control the robot.

III. CONTROL LAW

A. Virtual Impedance Control

We put a virtual mass at the end-effector of the robot, which is a virtual Spring-Damper system, see Fig.4. The Stereo tracking system provides position of the target, which is considered as the desired position $X_{md} = [x_{md}, y_{md}, z_{md}] \in \mathbb{R}^{3 \times 1}$ of the mass. We control the mass movement dynamically while generating a safe path for the end-effector of the manipulator. The task can be divided in two steps: the first is to get the mass position $X_m = [x_m, y_m, z_m] \in \mathbb{R}^{3 \times 1}$ and the second is to set the desired robot movement q_d .

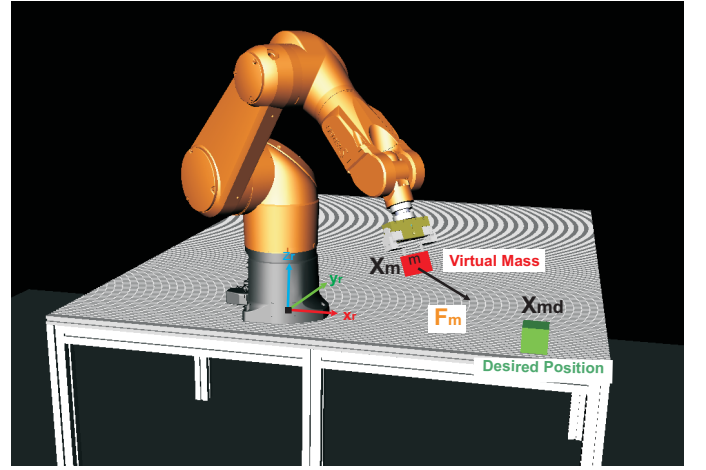


Fig. 4: A virtual mass is attached to the end-effector of the robot in order to control the robot, and F_m is the attractive force from the virtual mass to the target.

The dynamics of the virtual mass can be represented as:

$$m \ddot{X}_m = F - B \dot{X}_m - K X_m \quad (13)$$

$$\ddot{X}_m = m^{-1} (F - B \dot{X}_m - K X_m)$$

with

$$F = F_m + F_o \quad (14)$$

where F_m is the attractive force. We use the artificial potential field approach to compute virtual forces F_o for collision avoidance, where the obstacles are repulsive surfaces for the manipulator. Fig.5 shows all the models for these repulsive forces.

$$F_o = F_r + F_t + F_c + F_h \quad (15)$$

Then the robot control loop consists of the following steps:

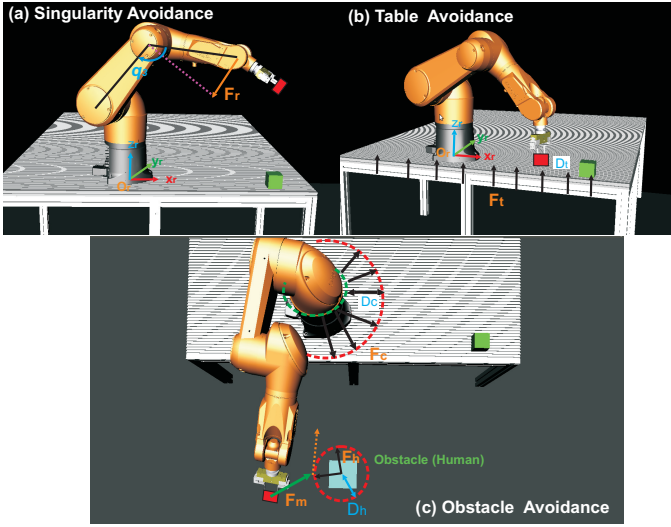


Fig. 5: The force models for environment constraints: (a) shows the force to avoidance singularity, (b) is the table avoidance force, (c) illustrates the self-collision and obstacle avoidance.

1) Compute total F :

a) Compute the attractive force:

$$F_m = -K_m(X_m - X_{md}) - B_m\dot{X}_m \quad (16)$$

where K_m and B_m are 3×3 matrix constants, and \dot{X}_m is the mass velocity.

In our stereo vision system, we only get the 3D position with respect to the left camera. So the force (16) can be rewritten as

$$F_m = -K_m R_r^{cL}(X_{m_s} - X_{md_s}) - B_m\dot{X}_m \quad (17)$$

where X_{m_s} and X_{md_s} are 3D positions in the stereo system, R_r^{cL} is the rotation matrix from robot base frame to the stereo system, which is estimated on-line in sec II-C .

b) Robot singularity avoidance force:

$$F_r = -K_r P_r D_r - B_r \dot{X}_m \quad (18)$$

where K_r and B_r are 3×3 matrix constants, and we define Δq as the absolute value of the difference between q_3 and $\pi/2$, D_r is the direction of the difference. Then

$$P_r = e^{\alpha_r \Delta q} - 1$$

where α_r is a constant.

c) Table collision avoidance force:

if $z_m > Dist_t$, where $Dist_t$ is the table height threshold, then

$$F_t = [0, 0, 0]^T \quad F_t \in \mathfrak{R}^{3 \times 1}$$

else: we define h_t is the distance between the X_m and the table in z dimension.

$$F_t = K_t P_t D_t - B_t \dot{X}_m \quad (19)$$

where

$$P_t = e^{\alpha_t (Dist_t - h_t)} - 1$$

$$D_t = [0, 0, 1]^T$$

d) Self-collision avoidance force:

if $h_c > Dist_c$, where $Dist_c$ is a constant representing the threshold on the distance (h_c) between the mass and robot base center in xoy plane, then

$$F_c = [0, 0, 0]^T \quad F_c \in \mathfrak{R}^{3 \times 1}$$

else:

$$F_c = K_c P_c D_c - B_c \dot{X}_m \quad (20)$$

where D_c is the direction of mass position in the xoy plane, and

$$P_c = e^{\alpha_c (Dist_c - h_c)} - 1$$

e) Obstacles avoidance force:

Using this framework, the robot can also be controlled in a way that it avoids collision with static entities (workbench) and dynamic entities (the human and the moving objects) in its environment. If the minimum distance of an obstacle to the robot body (h_h) is below a chosen security threshold ($Dist_h$), virtual forces can be computed and applied on the robot using potential fields.

Therefore when $h_h > Dist_h$ then

$$F_h = [0, 0, 0]^T \quad F_h \in \mathfrak{R}^{3 \times 1}$$

else:

$$F_h = K_h P_h D_h - B_h \dot{X}_m \quad (21)$$

where D_h is the normal directions of the obstacle surface, and

$$P_h = e^{\alpha_h (Dist_h - h_h)} - 1$$

2) After computing the total force F , we can use dynamics of the mass system to get the velocity and the position of the virtual mass X_m, \dot{X}_m , given by (13).

B. Dynamic Trajectory Generation

As we discussed in the previous subsection, virtual impedance control can generate a safe and singularity-free dynamic trajectory for the robot which also includes collision avoidance.

Once the mass position has been set, the new end effector desired position $X_{ef} = X_m$ is known. In our experiment, we detect 4 points on the robot end-effector. Therefore we can compute the end-effector pose which includes 3D position X_{ef} and orientation information R_{ef} . Then the desired q_{di} , $i = 1, 2, \dots, n$ can be calculated from the Inverse kinematics of the robot. This q_{di} is transmitted to the robot's control unit in real time.

IV. EXPERIMENTS

Experiments are performed to validate and evaluate this work on a standard industrial robot. In this task, we use real time tracking using AR markers² for identifying the target pose and the current pose of the robot end-effector. The target is carried by a human, and the robot end-effector tracks the target in a dynamic environment. Every marker provides 2D image features for 4 corner points, which can be used to compute the 3D position and orientation of the marker with respect to the camera frame. Several robot attributes such as singularity avoidance, self-collision and obstacles avoidance are implemented to ensure safety of the robot and human. This experiment also shows how the orientation matrix is estimated on-line enabling an uncalibrated visual servoing system, where occlusions due to camera placement can be handled in a natural and intuitive way by simply manually moving the camera to a better position.

A. System Overview

This system consists of 3 sub-systems, see Fig. 1: a) the Visual Stereo Tracker, b) the Robot Control System and c) the 3D visualization System.

1) *Visual Stereo Tracker*: The stereo system is composed of 2 USB cameras fixed on a tripod. The stereo rig is uncalibrated with respect to the robot base frame and can be manually moved. In order to avoid a multiple-sampling system, an extended Kalman filter (EKF) is used to estimate the visual position (sampling period $1ms$), whereas the reference is updated each $30ms$ with the real visual data of both cameras.

2) *Robot Control System*: The robot system comprises of a StaubliTX90 industrial robot arm (6 DOF), a CS8C control unit and a Workstation running on GNU/Linux OS with real-time patch, see Fig. 1. The data communication between the PC and the control unit is over a local network based on *TCP/IP*. In this article, joint positions $q_d(t) \in R^n$ were used to command the robot.

3) *3D Visualization System*: This module performs OpenGL based real-time rendering of the workspace in 3D. It uses Qt and Coin3D as its backbone, see [11]. This system visualizes the configuration of the robot arm and positions of the target in real-time. This is achieved by means of *TCP/IP* communication. Fig.4 shows a sample screenshot of this visualization.

B. On-line Orientation Matrix Estimation

As described in Section II, a coarse on-line estimation of the orientation matrix is computed using the real-time information generated by the robot.

Object occlusion occurs in Fig.6 (a), where the stereo system can then be moved to maintain the targets in the field of view, see Fig.6 (b). The camera motion is detected by the system and a process for coarse estimation of the orientation matrix between the stereo system and the robot base frame is started. The robot performs a small motion and a set of points

are collected, shown in Fig.7 (d). During this stage, the robot moves in free space, where the integration of environment constraints (obstacle-, self- and singularities avoidance) play an important role for safe/stable robot motions. We compare the actual end-effector position with the recovered end-effector position that is obtained by using the estimated rotation. Fig.7 (a) and (c) depicts these two positions trajectories. It can be clearly observed that the error between them is close to zero after on-line rotation matrix estimation, as illustrated in Fig. 7 (b).

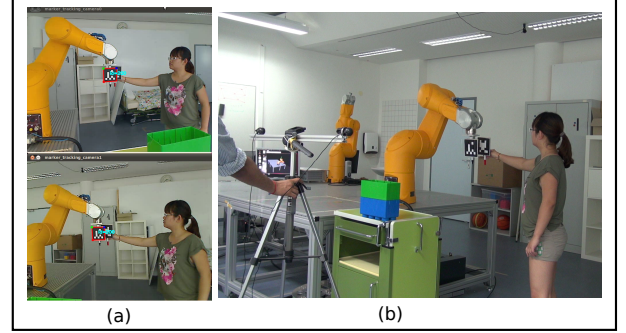


Fig. 6: Target occlusion

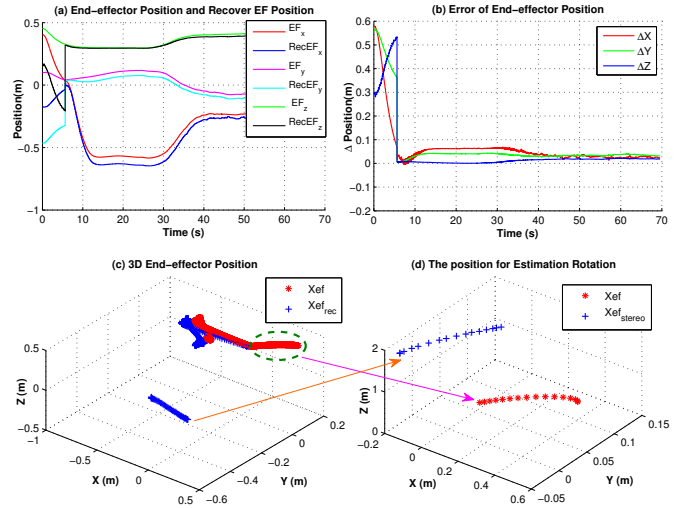


Fig. 7: Results of the on-line orientation matrix estimation: (a) and (c) show the 3D position of the end-effector in the base frame and the recovered position for the end-effector using the estimated rotation, (b) shows the errors between them, (d) illustrates an example of some selected points for the estimation of the orientation matrix.

C. Impedance Control for Environment Constraints

This experiment demonstrates real time tracking for the robot end-effector according to the moving target carried by the human. We address the tracking for both translation and rotation motions, which verifies the effectiveness of the estimation methods of R for the calibration-free robotic hand-eye coordination under stereo visual feedback.

²ArUco: <http://www.uco.es/investigacion/grupos/ava/node/26>

To demonstrate stability, we test our system under several environment constraints using virtual impedance control to combine visual servoing control with different environment forces. The experiment demonstrates the results of singularity avoidance, where the robot does not reach the singular condition ($q_3 = 0$), even when the user tries to force it. Using the normal directions of the object surfaces in the environment that are provided by the Kinect sensor, the robot can avoid collisions. This experiment also shows table avoidance where motion of the robot is constrained in the $z_r - axis$ by the height of the table (the end-effector is not allow to go under the table) but it can still move in the x_r and y_r axes. Self-collision handling is also demonstrated in this experiment. The control law explicitly can plan the robot path dynamically and avoids potential collisions with workspace obstacles and even humans safely.

1) *Human Avoidance*: In this experiment, the robot can be controlled so that it avoids collision with static entities (e.g. a workbench) and dynamic entities (e.g. the human and the moving objects) in its environment. In this task, the avoidance is done in a reactive way with dynamically updated collision scenes that can be interfaced with a variety of sensors. To compute the virtual forces that repel the robot from surrounding obstacles, we need to compute the minimum distance of all objects in the environments model to the robot.

For human avoidance, we use a Kinect sensor to detect and track the human skeleton, which we use to build our human model, as shown in Fig. 8. While performing the task, we check every joint in the human model and find the minimum distance to compute the repulsive force for the robot control.

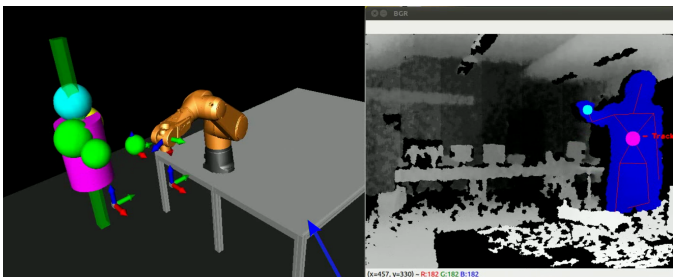


Fig. 8: Avoid Human: Use kinect to track human and build the human joints model.

Fig. 9 shows the process of how the robot avoids the obstacle (human) while continuing to track the target.

A video illustrating all these experimental results can be found at: <http://youtu.be/MG9KIHkCU9k>

V. CONCLUSION

In this paper, we proposed an uncalibrated stereo visual servoing algorithm which was found to be efficient for tracking a moving target in eye-to-hand configuration. The stereo vision system is composed of 2 USB cameras fixed on a tripod. We also proposed an algorithm for the estimating the extrinsic parameters of the stereo system using the real operation data of the end-effector. This calibration-free stereo tracking system

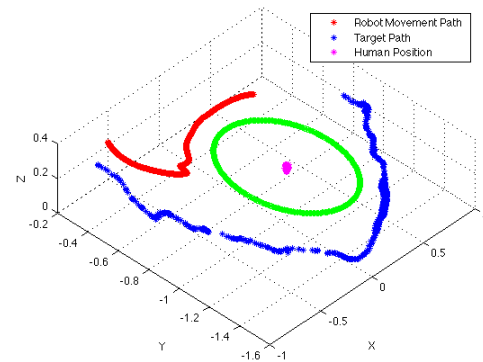


Fig. 9: Avoid Human (m): robot movement(red line) and target movement(blue line), Human position shows in (green circle)

was applied in an impedance control approach to generate a robot dynamic system with trajectory free of collisions and singularities. The proposed approach was evaluated on a real industrial robot (StaeubliTX90). The experiment illustrated the effectiveness of the uncalibrated visual system and the stability while handling dynamic environment constraints. The stereo system can be moved to maintain the targets in the field of view when the object occlusion occurs. A planned future extension is to integrate this uncalibrated stereo visual servoing approach with model based tracking approaches and also to validate the work in more practical use-cases.

REFERENCES

- [1] S. Hutchinson, G. Hager, and P. Corke, "A tutorial on visual servo control," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, Oct. 1996.
- [2] K. Tarabanis, R. Tsai, and A. Kaul, "Computing occlusion-free viewpoints," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 3, pp. 279–292, Mar. 1996.
- [3] J. Piepmeier, G. McMurray, and H. Lipkin, "A dynamic quasi-Newton method for uncalibrated visual servoing," in *IEEE International Conference on Robotics and Automation*, vol. 2, May 1999, pp. 1595–1600.
- [4] K. Hosoda and M. Asada, "Versatile visual servoing without knowledge of true Jacobian," in *IEEE/RSJ/GI International Conference on Intelligent Robots and Systems*, vol. 1, Sep. 1994, pp. 186–193.
- [5] J. Piepmeier, G. McMurray, and H. Lipkin, "Uncalibrated dynamic visual servoing," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 1, pp. 143–147, Feb. 2004.
- [6] P. Jiang, L. Bamforth, Z. Feng, J. Baruch, and Y. Chen, "Indirect Iterative Learning Control for a Discrete Visual Servo Without a Camera-Robot Model," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 37, no. 4, pp. 863–876, Aug. 2007.
- [7] K. Hosoda, K. Sakamoto, and M. Asada, "Trajectory generation for obstacle avoidance of uncalibrated stereo visual servoing without 3D reconstruction," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, Aug. 1995, pp. 29–34.
- [8] Q. Fu, Z. Zhang, J. Shi, L. Zhang, and R. Ma, "Uncalibrated visual servoing with obstacle avoidance using SQP method," in *IEEE International Conference on Mechatronics and Automation(ICMA)*, Aug. 2009, pp. 2031–2036.
- [9] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision (2. ed.)*. Cambridge University Press, 2006.
- [10] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-Squares Fitting of Two 3-D Point Sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 5, pp. 698–700, Sep. 1987.
- [11] M. Rickert and A. Knoll, "The Robotics Library RL: <http://sourceforge.net/projects/roblib/>," in *ICRA Workshop on Open Source Software in Robotics*, 2009.