A Multi-Camera Framework for Visual Servoing of a Collaborative Robot in Industrial Environments

Erika Di Stefano*, Emanuele Ruffaldi*, Carlo Alberto Avizzano*

***Abstract*— A great part of today's industries tends to invest on automatic machines that can replace or collaborate with humans in typical repetitive tasks. Despite their high motion and positioning precision, most of these industrial robots operate blindly, causing the working system to be poorly robust to even slight changes of the working conditions. A solution to such an issue might be to make the robots capable of readjusting their actions according to a perceptual feedback, in particular made of visual data. In this work we propose a multi-camera framework for the visual servoing of a collaborative robot that has to manipulate untextured industrial pieces. The robot is supposed to recognize the object of interest and reach it with its end-effector. We adopt a multi-camera approach that overcomes typical issues related to single-camera schemes. The system contains an object recognition module that extends an already existing algorithm for 2D detection on images to approximate 3D localization in space. A final probabilistic recursive estimation process combines the measures provided by the different sensors in order to improve the target pose computation, considering all the possible uncertainty and disturbance sources that may interfer, thus making the system more robust and efficient.**

I. INTRODUCTION AND RELATED WORK

Human operators working in industries and manifacturing sectors daily perform repetitive tasks that sometimes can compromise their physical health or even cause a psychological state of alienation. However, automation is gaining ground in modern industries and robotic machines are able to replace or assist humans in typical tedious and tiring tasks, such as continuous picking, placing and assembling operations. Despite their high motion and positioning precision, these machines mostly act blindly, i.e they are supposed to move in a perfectly known workspace, with a perfectly known location and configuration of the objects they interact with. This means that a minimal disturbance in the predetermined working conditions may drastically worsen the system performance. Such situations can be avoided by providing perceptual capabilities to a robot that can actively interact with the surrounding environment and readjust its actions according to a visual feedback. The use of visual sensors is encouraged by the large amount of low-cost cameras, such as RGB and RGB-D cameras, that today exist in the market. On the other hand, visual servoing, i.e control of a robot's motion through visual information, is a widely addressed issue present in the literature.

The first attempts to control a robot's motion through visual feedback date back to the 80's - 90's [1], [2], [3]. However, the field gradually branched into two main approaches. In the image-based approach the robot control law explicitly depends on the image features, i.e particular keypoints that can be individuated on the image. In the position-based approach the features are used in conjunction with the geometric model of a target object to be reached in order to estimate its pose with respect to the camera. Although image-based schemes don't need image interpretation, reduce the computational delay and eliminate errors due to camera calibration [4], they require a non-linear and highly coupled control design and typical instability issues may occur [5]. A position-based approach, on the contrary, makes the target pose computation independent from the robot motion control. Nevertheless, the pose computation process may sometimes be non-trivial. Object pose can be computed if the vision system observes multiple point features on a known object. Numerous methods for its solution have been proposed and they can be mainly divided into analytic solutions, [6], [7], [8] and least-squares solutions [9], [10], [11]. Furthermore, two different types of camera configurations are generally adopted. In the eye-to-hand configuration the camera is fixed in the workspace, whereas in the eye-in-hand configuration the camera is attached to the robot's end-effector. An extensive explanation of the different schemes and approaches can be found in [12] and [13]. There are several works that employ a single fixed or moving camera [14], [15], [16]. However, in single camera based approaches the quality of the captured images is highly influenced by the location of the camera, as in the vicinity of the object the images are more accurate, but the camera field of view is limited, while at higher distances the field of view improves, but measurement errors may increase. Mixed eye-in-hand/eye-to-hand configurations have been studied in recent works. Kermorgant et al. [17] propose a low-level sensor fusion scheme for the positioning of a multi-sensor robot. Wang et al. [18] employ an eye-in-hand camera to track the position of a target object and a stereo camera to obtain the depth information of the object. Luo et al. [19] describe a hybrid eye-in-hand, eye-to-hand framework for object tracking and fetching. Marshall et al. [20] propose a Kalman Filter visual servoing control law. However, in each of these works the multi-sensor and control scheme is modelled by means of the classical Jacobian matrix and the typical before mentioned instability issues may occur. In other works, such as that of Lippiello et al. [21] the visual information of a hybrid camera configuration is assembled to compute the pose of a target object and control a robotic arm using image features in an Extended Kalman Filter. Nevertheless, the pose computation is constrained on the identification of discriminative features that must always be retrieved unambiguously on the image

*PERCeptual RObotics Laboratory, Scuola Superiore Sant'Anna

and therefore strongly depend on the image quality and the object texture.

We propose a hybrid multi-camera framework conceived for a position-based visual servoing of a collaborative robot that has to manipulate untextured industrial pieces. The multi-camera approach overcomes the issues related to single-camera schemes, such as object occlusions and view-dependent errors. The position-based approach expects the computation of a target pose, but the textureless nature of the industrial pieces prevents from employing the previously mentioned point-based methods. Hence we make use of an object recognition module that extends an already existing algorithm for 2D detection on images to approximate 3D localization in space through the use of strong graphical tools. We eventually make the system more robust and efficient by embedding a probabilistic model that considers the possible uncertainty and disturbance sources, implementing a recursive estimation process that combines the measures provided by the different sensors in order to provide a more accurate target pose and improve the chances of success of the robot's task.

The remaining of the paper is structured as follows: in section II we explain our approach, illustrating the system model and explaining the different components, section IV shows the experimental results and in section V we discuss the conclusions.

II. PROPOSED APPROACH

Our system consists of several stages described in the following. Firstly each image delivered by the different cameras is opportunely processed in order to detect the object of interest and provide a pose measure with respect to the relative camera frame. The pose measures are then exploited in a recursive filtering process that aims at computing different pose estimates. The pose estimates are eventually combined in order to obtain a single overall pose estimate and provide a reference end pose to the robot's end-effector. Figure (3) shows an overview of the different components and stages of our framework. The following sections explain the different parts in more detail.

A. WORKSPACE MODEL

Figure (1) shows the different components and reference frames that have to be taken into account in our system. If n is the total number of cameras, for simplicity we consider only one eye-in-hand camera attached to the robot's end-effector that approches the object, and $n-1$ fixed, eye-to-hand cameras.

Let's consider $\{S_0\}$ as the base reference frame and $\{S_j\}, j = 1, \cdots, n$ the frame attached to the j-th camera. Let's suppose that we know the transformation matrix ${}^0\boldsymbol{M}_j$ from each frame $\{S_j\}$ to frame $\{S_0\}$, where

$$ {}^0\boldsymbol{M}_j = \begin{bmatrix} {}^0\boldsymbol{R}_j & {}^0\boldsymbol{t}_j \\ \boldsymbol{0}_{1\times 3} & 1 \end{bmatrix} \tag{1} $$

${}^0\boldsymbol{R}_j$ is the rotation matrix and ${}^0\boldsymbol{t}_j$ the translation vector. In fact, ${}^0\boldsymbol{M}_1$ can be computed from the arm's forward kinematics, while each ${}^0\boldsymbol{M}_j, j = 1, \cdots, n$ can be computed after an extrinsic calibration of each camera with respect to the base frame. We collect the pose measures with respect to the different camera frames and then estimate the object's pose with respect to the base reference frame.

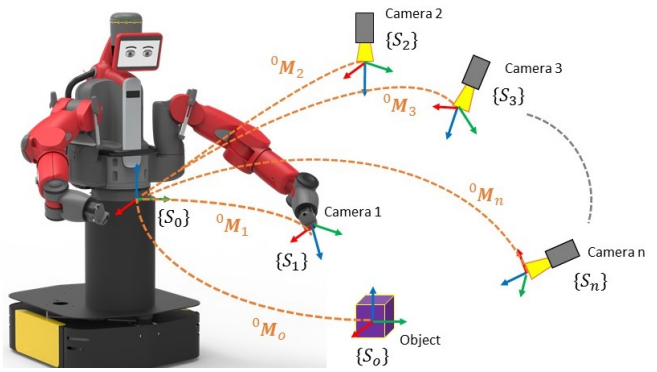

Fig. 1. General scenario with n cameras, one attached to the robot's end-effector and the rest fixed in the workspace. A reference frame $\{S_j\}, j = 1, \cdots, n$ is attached to each camera, frame $\{S_o\}$ is attached to the object and a base reference frame $\{S_0\}$ is defined.

B. OBJECT DETECTION

Our detection system relies on a template-matching algorithm conceived by Hinterstoisser et al. [22], based on colour gradients, that fits very well with industrial objects without texture. However, as most of the template-matching algorithms, it is based on the comparison of the image scene with previously stored models and hence it is not invariant to rigid transformations. This means that an object is detected in an image only if it corresponds to a specific template deriving from a particular 2D projection (depending on the object's 3D pose with respect to the camera). Since we are interested in not only detecting the object in the scene, but also compute its 3D pose in space, we adopt a graphics method that allows to extend the 2D detection process to 3D approximate localization. In particular, in a preliminary training stage where the system learns the objects that need to be recognized, we consider the CAD model of the object of interest and in a computationally strong graphics environment [23] we virtually project it in a series of reference poses with respect to a virtual camera. The virtual projection allows us to produce a series of synthetic images in which the colour gradient features are computed in order to create the templates. In this way each template is stored with the corresponding reference pose and during the real-time detection stage the algorithm is able to return a bounding box on the region of the image where the object has been detected and an approximate measure of the object's pose with respect to the relative camera in the 3D space. The pose measure is only approximated and coarse, since not all the possible poses can be recorded, but only a discrete number of points in the viewing sphere can be considered. Hence, each real-pose is always matched with the nearest template-pose present in the stored database. Figure

(2) illustrates how the detection stage works.

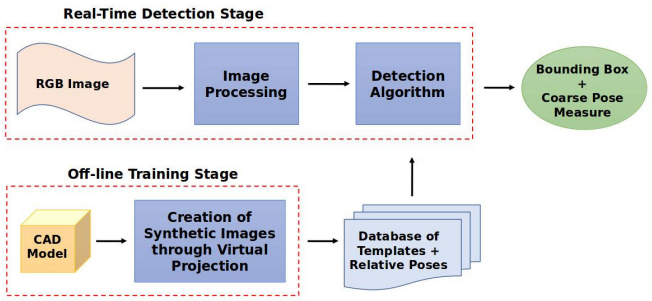

Fig. 2. The real-time object detection module accesses a database of stored templates and relative 3D poses, providing a 2D bounding box and a coarse pose measure. Templates are created during an off-line training stage through the object's CAD model and graphics software.

C. POSE ESTIMATION

Once the object has been detected on each of the images and a measure of the object's pose with respect to each camera has been provided, we need to combine the different measures and compute an overall result in order to allow the robot to reach the correct end-point with a specific end-effector configuration. However, as mentioned before, those provided in the detection stage are coarse measures of the real poses with respect to each camera, the measurement noise associated to the cameras can affect the detection process and the robot's end-effector position might not be known exactly. For this reason we decided to introduce a probabilistic model that takes into account the possible uncertainty and disturbance interferences, such as the sensors measurement noise, the detection confidence level, the imperfect knowledge of the robot's end-effector position, etc, conferring more robustness to the system. Furthermore, we assume that the object's image quality increases by approaching the object. Indeed, if we consider a *pin-hole* camera model, we realize that the pixel coordinates of a world point projected onto the image plane depend on the distance from the camera frame [24]. As a matter of fact, the object resolution increases in the vicinity and the pose computation improves consequently. We thus implement two parallel estimators: in the first one we employ the measures extracted from the eye-in-hand camera in conjunction with the arm motion dynamics in order to compute a pose estimate with respect to the local attached reference frame $\{S_1\}$. Indeed, the robot forward kinematics is reflected on the motion of the end-effector, that moves integrally with the camera; in the second estimator we collect the measures coming from the different fixed cameras, each mapped to the base frame $\{S_0\}$, and provide a pose estimate with respect to that frame. Equations (2) and (3) formalize our mathematical model.

$$\begin{cases} {}^1\boldsymbol{x}_k = \boldsymbol{f_1}\left({}^1\boldsymbol{x}_{k-1}, \boldsymbol{u}_k, {}^1\boldsymbol{\mu}_k\right) \\ {}^1\boldsymbol{z}_k = \boldsymbol{h_1}\left({}^1\boldsymbol{x}_k, {}^1\boldsymbol{\nu}_k\right) \end{cases} \tag{2}$$

$$\begin{cases} {}^0\boldsymbol{x}_k = \boldsymbol{f_2}\left({}^0\boldsymbol{x}_{k-1}, {}^0\boldsymbol{\mu}_k\right) \\ {}^0\boldsymbol{z}_k = \boldsymbol{h_2}\left({}^0\boldsymbol{x}_k, {}^0\boldsymbol{\nu}_k\right) \end{cases} \tag{3}$$

The state vector ${}^i\boldsymbol{x}_k$ and measurement vector ${}^i\boldsymbol{z}_k$ consist of the position and orientation components with respect to $\{S_i\}, i = 0, 1$:

$$\begin{aligned} {}^i\boldsymbol{x}_k &= \begin{bmatrix} {}^i\boldsymbol{p}_k^T & {}^i\boldsymbol{q}_k^T \end{bmatrix}^T \\ {}^i\boldsymbol{z}_k &= \begin{bmatrix} {}^i\boldsymbol{z}_{\boldsymbol{p}_k}^T & {}^i\boldsymbol{z}_{\boldsymbol{q}_k}^T \end{bmatrix}^T \end{aligned} \tag{4}$$

We adopt the unit quaternion representation for the orientation component in order to avoid singularity issues [25]. $\boldsymbol{u}_k$ in (2) denotes the eye-in-hand camera displacement due to the arm motion, referred to as an input. No input is present in (3) since the relative reference systems are all fixed. ${}^j\boldsymbol{\mu}_k$ and ${}^j\boldsymbol{\nu}_k$ respectively denote the process and measurement noise variables. In (2) the process noise is due to uncertainty associated to the arm's motion, whereas in (3) it is caused by initialization errors. On the other hand, the measures are corrupted by image digital noise and are affected by a discretization error following from the sampling of the viewing sphere performed in the training stage. We assume for simplicity and formal coherence that both process and measurement noise can be modelled as zero-mean, uncorrelated white Gaussian variables. Furthermore, ${}^0\boldsymbol{z}_k$ is an extended measurement vector where all the contributions are stacked up as shown in equation (5):

$$\begin{aligned} {}^0\boldsymbol{z}_k &= \begin{bmatrix} {}^0\boldsymbol{M}_2{}^2\boldsymbol{z}_{\boldsymbol{p},k} & \cdots & {}^0\boldsymbol{M}_n{}^n\boldsymbol{z}_{\boldsymbol{p},k} \end{bmatrix} \\ &= \begin{bmatrix} {}^0\boldsymbol{z}_{\boldsymbol{p},k}^{(2)T} & \cdots {}^0\boldsymbol{z}_{\boldsymbol{p},k}^{(n)T} \end{bmatrix}_k^T \end{aligned} \tag{5}$$

From now on, we will refer to the two parallel estimates at time step k as:

$$\begin{aligned} {}^1\hat{\boldsymbol{x}}_k &= \text{ Pose estimate with respect to } \{S_1\} \\ {}^0\hat{\boldsymbol{x}}_k &= \text{ Pose estimate with respect to } \{S_0\} \end{aligned} \tag{6}$$

Equations (2) and (3) can be, in fact, split into the position and orientation state and measurement models as a consequence of their decoupling (equations (7) and (8)).

$$\begin{cases} {}^1\boldsymbol{p}_k = {}^{1,k}\boldsymbol{R}_{\boldsymbol{c},k-1}{}^1\boldsymbol{p}_{k-1} + {}^{1,k}\boldsymbol{t}_{\boldsymbol{c},k-1} + {}^1\boldsymbol{\mu}_{\boldsymbol{p}_k} \\ {}^1\boldsymbol{q}_k = {}^1\boldsymbol{\mu}_{\boldsymbol{q}_k} \otimes {}^1\boldsymbol{q}_{k-1} \otimes {}^{1,k}\boldsymbol{q}_{\boldsymbol{c},k-1} \\ \\ {}^1\boldsymbol{z}_{\boldsymbol{p}_k} = {}^1\boldsymbol{p}_k + {}^1\boldsymbol{\nu}_{\boldsymbol{p}_k} \\ {}^1\boldsymbol{z}_{\boldsymbol{q}_k} = {}^1\boldsymbol{\nu}_{\boldsymbol{q}_k} \otimes {}^1\boldsymbol{y}_{\boldsymbol{q}_k} \end{cases} \tag{7}$$

$$\begin{cases} {}^0\boldsymbol{p}_k = {}^0\boldsymbol{p}_{k-1} + {}^0\boldsymbol{\mu}_{\boldsymbol{p}_k} \\ {}^0\boldsymbol{q}_k = {}^0\boldsymbol{\mu}_{\boldsymbol{q}_k} \otimes {}^0\boldsymbol{q}_{k-1} \\ \\ {}^0\boldsymbol{z}_{\boldsymbol{p}_k} = {}^0\boldsymbol{p}_k + {}^0\boldsymbol{\nu}_{\boldsymbol{p}_k} \\ {}^0\boldsymbol{z}_{\boldsymbol{q}_k} = {}^0\boldsymbol{\nu}_{\boldsymbol{q}_k} \otimes {}^0\boldsymbol{q}_k \end{cases} \tag{8}$$

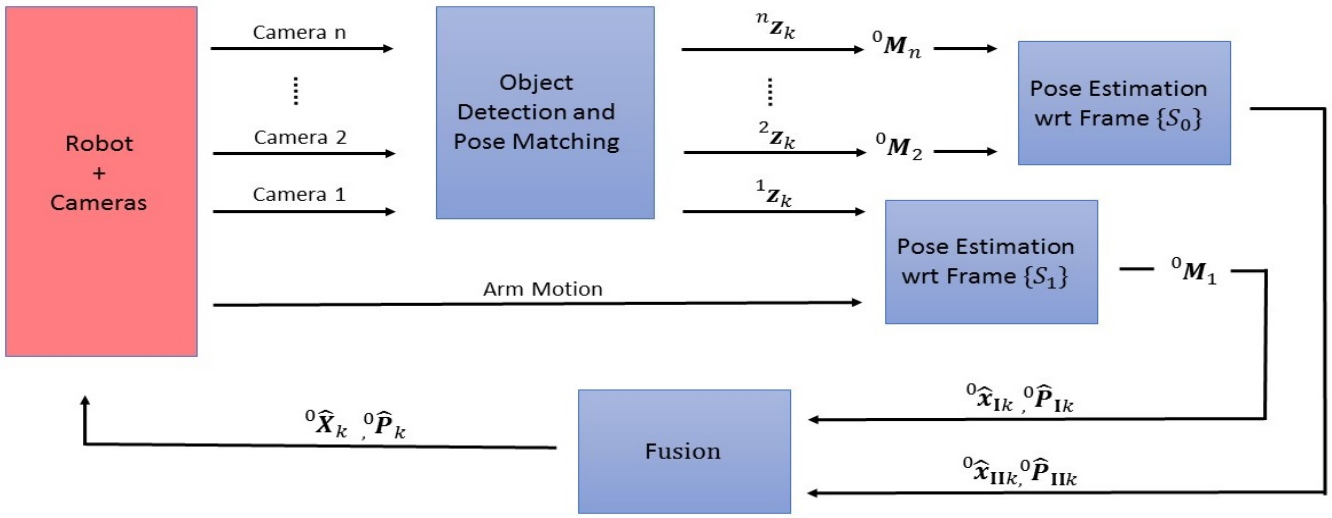

Fig. 3. Framework overview. The images provided by the cameras are given to the detection algorithm that matches a template and provides a rough object pose measure. The pose measures provided by the fixed cameras and those provided by the mobile camera together with the arm's motion dynamics are respectively used to compute an object pose estimate with respect to the fixed frame and an object pose estimate with respect to the mobile frame. The two estimates are eventually combined in order to provide an overall pose estimate to the robot.

Figure (3) shows an overview of the overall framework. The eye-in-hand camera displacement can be retrieved at each time step from the forward kinematics of the robot's arm. We indicate with $^{1,k}\boldsymbol{t}_{\boldsymbol{c},k-1}$ the translation occurred from the previous camera configuration $\{S_{1,k-1}\}$ to the current one $\{S_{1,k}\}$, and with $^{1,k}\boldsymbol{R}_{\boldsymbol{c},k-1}$ the rotation with corresponding unit quaternion $^{1,k}\boldsymbol{q}_{\boldsymbol{c},k-1}$. The symbol $\otimes$ denotes the quaternion product. We indicate with $^{i}\boldsymbol{Q}_{\boldsymbol{p}}, ^{i}\boldsymbol{Q}_{\boldsymbol{q}} \in \Re^{3\times3}$ the position and quaternion process covariance matrices respectively, with $^{i}\boldsymbol{R}_{\boldsymbol{p}}, ^{i}\boldsymbol{R}_{\boldsymbol{q}} \in \Re^{3\times3}$ the measurement noise covariance matrices and with $^{i}\boldsymbol{P}_{\boldsymbol{p}}, ^{i}\boldsymbol{P}_{\boldsymbol{q}} \in \Re^{3\times3}$ the estimate error covariance matrices relative to position and orientation. As discussed before, we assume that the measurement noise variance increases with the distance of the camera from the object of interest and decreases when the camera gets closer to it, i.e:

$$^{i}\boldsymbol{R}_{\boldsymbol{p},\boldsymbol{q}_k} = f(^{i}Z_k) \tag{9}$$

Furthermore, being $^{i}\boldsymbol{q}_k$ a unit quaternion, having the norm constraint $||^{i}\boldsymbol{q}_k|| = 1$, the uncertainty associated to the quaternion is characterized by three degrees of freedom. Indeed, $^{i}\boldsymbol{q}_k$ can alternatively be expressed as

$$^{i}\boldsymbol{q}_k = \left[\cos\left(\tfrac{\theta_k}{2}\right) \quad ^{i}\hat{\boldsymbol{w}}_k \sin\left(\tfrac{\theta_k}{2}\right)\right] \tag{10}$$

where

$$\theta_k {}^{i}\hat{\boldsymbol{w}}_k = {}^{i}\bar{\boldsymbol{v}}_k = v_0\hat{\boldsymbol{i}}_k + v_1\hat{\boldsymbol{j}}_k + v_2\hat{\boldsymbol{k}}_k \tag{11}$$

is the so-called *axis-angle* representation and the uncertainty can be associated to v_0, v_1 and v_2. $^{i}\boldsymbol{\mu}_{\boldsymbol{q},\boldsymbol{k}}$ and $^{i}\boldsymbol{\mu}_{\boldsymbol{q},\boldsymbol{k}}$ in (7) and (8) are thus the quaternions obtained from $^{i}\boldsymbol{\mu}_{\boldsymbol{v},\boldsymbol{k}} \in \Re^{3\times1}$ and $^{i}\boldsymbol{\nu}_{\boldsymbol{v},\boldsymbol{k}} \in \Re^{3\times1}$ according to (10) and (11). It is clear from equations (7) and (8) that the position dynamics is linear, whereas the quaternion dynamics is not. As a result, the position estimation can be carried out through a classical linear Kalman filter [26], while we employ an Unscented Kalman Filter [27] to capture the strong non-linearities involved during orientation estimation, adapting the equations of interest to the quaternion representation [28].

Position Estimation:

For the position estimation the classical Kalman equations are employed.

Prediction:

$$\begin{cases} {}^{1}\hat{\boldsymbol{p}}_k^- = {}^{1,k}\boldsymbol{R}_{\boldsymbol{c},k-1}\,{}^{1}\hat{\boldsymbol{p}}_{k-1} + {}^{1,k}\boldsymbol{t}_{\boldsymbol{c},k-1} \\ {}^{1}\hat{\boldsymbol{P}}_{\boldsymbol{p},k}^- = {}^{1,k}\boldsymbol{R}_{\boldsymbol{c},k-1}\,{}^{1}\hat{\boldsymbol{P}}_{\boldsymbol{p},k-1}\,{}^{1,k}\boldsymbol{R}_{\boldsymbol{c},k-1}^{T} + {}^{1}\boldsymbol{Q}_{\boldsymbol{p},k} \end{cases} \tag{12}$$

$$\begin{cases} {}^{0}\hat{\boldsymbol{p}}_k^- = {}^{0}\hat{\boldsymbol{p}}_{k-1} \\ {}^{0}\hat{\boldsymbol{P}}_{\boldsymbol{p},k}^- = {}^{0}\hat{\boldsymbol{P}}_{\boldsymbol{p},k-1} + {}^{0}\boldsymbol{Q}_{\boldsymbol{p},k} \end{cases} \tag{13}$$

Update:

$$^{i}\boldsymbol{K}_{\boldsymbol{p}_k} = {}^{i}\hat{\boldsymbol{P}}_{\boldsymbol{p},k}^{-}\,{}^{i}\boldsymbol{C}_k^{T}\left({}^{i}\boldsymbol{R}_{\boldsymbol{p},k} + {}^{i}\boldsymbol{C}_k\,{}^{i}\hat{\boldsymbol{P}}_{\boldsymbol{p},k}^{-}\,{}^{i}\boldsymbol{C}_k^{T}\right)^{-1} \tag{14}$$

where for $i = 0$

$$^{i}\boldsymbol{R}_{\boldsymbol{p},k} = diag\left({}^{0}\boldsymbol{R}_{\boldsymbol{p},k}^{(2)}, \cdots {}^{0}\boldsymbol{R}_{\boldsymbol{p},k}^{(n)}\right)$$
$$^{i}\boldsymbol{C}_k = \boldsymbol{I}_{(n-1)\times3}$$

$$\begin{cases} {}^i\hat{\boldsymbol{p}}_k = {}^i\hat{\boldsymbol{p}}_k^- + {}^i\boldsymbol{K}_{\boldsymbol{p}_k}\left({}^i\boldsymbol{z}_{\boldsymbol{p},k} - \boldsymbol{C}_k\,{}^i\hat{\boldsymbol{p}}_k^-\right) \\ {}^i\hat{\boldsymbol{P}}_{\boldsymbol{p},k} = \left(\boldsymbol{I}_{3\times 3} - {}^i\boldsymbol{K}_{\boldsymbol{p}_k}\,{}^i\boldsymbol{C}_k\right){}^i\hat{\boldsymbol{P}}_{\boldsymbol{p},k}^- \end{cases} \tag{15}$$

Orientation Estimation:

Prediction:

At first, we sample the quaternion state distribution function around the mean, i.e the state component ${}^i\hat{\boldsymbol{q}}_{k-1}$ estimated in the previous time step, and collect the samples in a vector of sigma points. For this aim, we build a vector of perturbation quaternions deriving from the Cholesky factorization of covariance matrix ${}^i\boldsymbol{P}_{\boldsymbol{q}}$. Since ${}^i\boldsymbol{P}_{\boldsymbol{q}}$ has 3×3 dimensions, we obtain firstly a sigma vector made of three-dimensional components ${}^i\boldsymbol{\mathcal{X}}_{\boldsymbol{P}_{\boldsymbol{v}}} \in \Re^{3(2N)\times 1}$:

$${}^i\boldsymbol{\mathcal{X}}_{\boldsymbol{P}_{\boldsymbol{v}}} = \sqrt{2N}\left[\cdots + \left(\sqrt{{}^i\boldsymbol{P}_q}\right)_l \cdots - \left(\sqrt{{}^i\boldsymbol{P}_q}\right)_m \cdots\right] \tag{16}$$

where $N = 3$, $l = 1, \cdots N$ and $m = N+1, \cdots, 2N$ and then transform the three-dimensional components into the corresponding quaternions according to (10) and (11) to obtain ${}^i\boldsymbol{\mathcal{X}}_{\boldsymbol{P}_{\boldsymbol{q}}} \in \Re^{4(2N)\times 1}$. By perturbing ${}^i\hat{\boldsymbol{q}}_{k-1}$ through the quaternion components of ${}^i\boldsymbol{\mathcal{X}}_{\boldsymbol{P}_{\boldsymbol{q}}}$ we obtain the following vector of sigma points:

$${}^i\boldsymbol{\mathcal{X}}_{\boldsymbol{q}} = \left[{}^i\hat{\boldsymbol{q}}_k \quad \cdots \quad {}^i\boldsymbol{\mathcal{X}}_{\boldsymbol{P}_{\boldsymbol{q}}\,j} \otimes {}^i\hat{\boldsymbol{q}}_k \quad \cdots\right] \in \Re^{4(2N+1)\times 1} \tag{17}$$

The sampling of the measurement and process noise probability distribution is carried out in the same way around the zero-mean to obtain ${}^i\boldsymbol{\mathcal{X}}_{\boldsymbol{\mu}_{\boldsymbol{q}}} \in \Re^{4(2N+1)\times 1}$ and ${}^i\boldsymbol{\mathcal{X}}_{\boldsymbol{\nu}_{\boldsymbol{q}}} \in \Re^{4(2N+1)\times 1}$. At this point, we project the sigma points ahead in time by application of the process model ${}^i\boldsymbol{\mathcal{F}}$:

$${}^i\hat{\boldsymbol{\mathcal{X}}}_{\boldsymbol{q}_k}^- = {}^i\boldsymbol{\mathcal{F}}\left({}^i\boldsymbol{\mathcal{X}}_{\boldsymbol{q}_{k-1}}, {}^i\boldsymbol{\mathcal{X}}_{\boldsymbol{\mu}_k}\right) \tag{18}$$

and we compute the mean quaternion ${}^i\hat{\boldsymbol{q}}_k^-$, as the state prediction, through the intrinsic gradient descent algorithm described in [29].
Next, we derive the estimate error covariance matrix prediction by considering the errors between the quaternion prediction and the quaternion components of prediction ${}^i\hat{\boldsymbol{\mathcal{X}}}_{\boldsymbol{q}_k}^-$ obtained from the propagation equation (18):

$${}^i\boldsymbol{e}_{\boldsymbol{q}_i} = {}^i\hat{\boldsymbol{\mathcal{X}}}_{\boldsymbol{q}\,j,k}^- \otimes ({}^i\hat{\boldsymbol{q}}_k^-)^{-1} \tag{19}$$

The error quaternions are transformed into the corresponding axis-angle error vectors ${}^i\boldsymbol{e}_{\boldsymbol{v}i}$ and the estimate error covariance matrix can be derived as:

$${}^i\hat{\boldsymbol{P}}_{\boldsymbol{q},k}^- = \frac{1}{2N}\sum_{i=1}^{2N} {}^i\boldsymbol{e}_{\boldsymbol{v}i}\,{}^i\boldsymbol{e}_{\boldsymbol{v}i}^T \tag{20}$$

Then we apply the measurement model ${}^i\boldsymbol{\mathcal{H}}$ to obtain the output prediction:

$${}^i\boldsymbol{\mathcal{Y}}_{\boldsymbol{q}_k^-} = {}^i\boldsymbol{\mathcal{H}}\left[{}^i\hat{\boldsymbol{\mathcal{X}}}_{\boldsymbol{q}_k^-}, {}^i\boldsymbol{\mathcal{X}}_{\boldsymbol{\nu}_k}\right] \tag{21}$$

Notice that

$${}^0\boldsymbol{\mathcal{Y}}_{\boldsymbol{q}}{}^- = \left[\left({}^0\boldsymbol{\mathcal{Y}}_{\boldsymbol{q}}{}^{(2)-}\right)^T \quad \cdots \quad \left({}^0\boldsymbol{\mathcal{Y}}_{\boldsymbol{q}}{}^{(n)-}\right)^T\right]^T \tag{22}$$

as in the second filter the measurement vector is augmented. The mean output quaternion prediction ${}^i\boldsymbol{y}_{\boldsymbol{q}\hat{k}}^-$ is computed analogously to ${}^i\hat{\boldsymbol{q}}_k^-$ by means of the iterative method, where

$${}^0\boldsymbol{y}_{\boldsymbol{q}}{}^- = \left[\left({}^0\boldsymbol{y}_{\boldsymbol{q}}{}^{(2)-}\right)^T \quad \cdots \quad \left({}^0\boldsymbol{y}_{\boldsymbol{q}}{}^{(n)-}\right)^T\right]^T \tag{23}$$

the output covariance matrix ${}^i\hat{\boldsymbol{P}}_{\boldsymbol{y}_{\boldsymbol{q}}}{}^-$ prediction is derived from the error quaternions:

$${}^i\boldsymbol{e}_{\boldsymbol{y}_{\boldsymbol{q}}i} = {}^i\boldsymbol{\mathcal{Y}}_{\boldsymbol{q}\,j,k}^- \otimes ({}^i\boldsymbol{y}_{\boldsymbol{q}\hat{k}}^-)^{-1}$$

These are transformed into the error vectors ${}^i\boldsymbol{e}_{\boldsymbol{v}_{\boldsymbol{y}}i}$ and used to compute the matrix prediction:

$${}^i\hat{\boldsymbol{P}}_{\boldsymbol{y}_{\boldsymbol{q}}} = \frac{1}{2N}\sum_{i=1}^{2N} {}^i\boldsymbol{e}_{\boldsymbol{v}_{\boldsymbol{y}}i}\,{}^i\boldsymbol{e}_{\boldsymbol{v}_{\boldsymbol{y}}i}{}^T \tag{24}$$

At last, we derive the cross-covariance matrix:

$${}^i\hat{\boldsymbol{P}}_{\boldsymbol{q},\boldsymbol{y}_{\boldsymbol{q}}} = \frac{1}{2N}\sum_{i=1}^{2N} {}^i\boldsymbol{e}_{\boldsymbol{v}i}\,{}^i\boldsymbol{e}_{\boldsymbol{v}_{\boldsymbol{y}}i}{}^T \tag{25}$$

Update:

First we compute the Kalman gain:

$${}^i\boldsymbol{K}_{\boldsymbol{q}_k} = {}^i\hat{\boldsymbol{P}}_{\boldsymbol{q},\boldsymbol{y}_{\boldsymbol{q}}}\,{}^i\hat{\boldsymbol{P}}_{\boldsymbol{y}_{\boldsymbol{q}}}{}^{-1} \tag{26}$$

where for $i = 0$:

$$\begin{aligned} {}^0\hat{\boldsymbol{P}}_{\boldsymbol{q},\boldsymbol{y}_{\boldsymbol{q}}} &= \left[{}^0\hat{\boldsymbol{P}}_{\boldsymbol{q},\boldsymbol{y}_{\boldsymbol{q}}}{}^{(2)} \quad \cdots \quad {}^0\hat{\boldsymbol{P}}_{\boldsymbol{q},\boldsymbol{y}_{\boldsymbol{q}}}{}^{(n)}\right] \\ {}^0\hat{\boldsymbol{P}}_{\boldsymbol{y}_{\boldsymbol{q}}} &= \left[diag\left({}^0\hat{\boldsymbol{P}}_{\boldsymbol{y}_{\boldsymbol{q}}}{}^{(2)}, \cdots, {}^0\hat{\boldsymbol{P}}_{\boldsymbol{y}_{\boldsymbol{q}}}{}^{(n)}\right)\right]^{-1} \end{aligned} \tag{27}$$

The residual is obtained from the quaternion difference. In the first filter it is computed as follows:

$${}^1\boldsymbol{q}_{\boldsymbol{r}\,k} = {}^1\hat{\boldsymbol{q}}_k^- \otimes ({}^1\boldsymbol{y}_{\boldsymbol{q}\hat{k}}^-)^{-1} \tag{28}$$

and then transformed to ${}^1\boldsymbol{v}_{\boldsymbol{r}\,k} \in \Re^{3\times 1}$. In the second one, the residual term is augmented:

$$
\begin{aligned}
&{}^0\boldsymbol{q}_{\boldsymbol{r}k} = \begin{bmatrix} {}^0\hat{\boldsymbol{q}}_k^- \otimes \left({}^0\boldsymbol{y}_{\boldsymbol{q}_k}^{(2)-}\right)^{-1} & \cdots & {}^0\hat{\boldsymbol{q}}_k^- \otimes \left({}^0\boldsymbol{y}_{\boldsymbol{q}_k}^{(n)-}\right)^{-1} \end{bmatrix} \\
&{}^0\boldsymbol{v}_{\boldsymbol{r}k} \in \Re^{3(n-1)\times 1}
\end{aligned} \tag{29}
$$

The correction term is obtained by multiplication with the Kalman gain:

$$
{}^i\boldsymbol{v}_{\boldsymbol{c}k} = {}^i\boldsymbol{K}_{\boldsymbol{q}_k}\, {}^i\boldsymbol{v}_{\boldsymbol{r}k} \tag{30}
$$

and the correction vector is afterwards transformed into the corresponding quaternion ${}^i\boldsymbol{q}_{\boldsymbol{c}k} \in \Re^{4\times 1}$.
Finally, the following state and covariance matrix updates can be performed:

$$
\begin{cases}
{}^i\hat{\boldsymbol{q}}_k = {}^i\hat{\boldsymbol{q}}_k^- \otimes {}^i\boldsymbol{q}_{\boldsymbol{c}_k}^{-1} \\
{}^i\hat{\boldsymbol{P}}_{\boldsymbol{q},k} = {}^i\hat{\boldsymbol{P}}_{\boldsymbol{q},k}^- - {}^i\boldsymbol{K}_{\boldsymbol{q}_k}\, {}^i\hat{\boldsymbol{P}}_{\boldsymbol{y}_q}\, {}^i\boldsymbol{K}_{\boldsymbol{q}_k}^T
\end{cases} \tag{31}
$$

Fusion and Reference End Pose Computation:

As a last step, the two parallel estimates are combined to obtain a unique estimate of the object's pose with respect to the base reference frame $\{S_0\}$. Consider the following expressions:

$$
\left({}^0\hat{\boldsymbol{x}}_{\boldsymbol{I}k}, {}^0\hat{\boldsymbol{P}}_{\boldsymbol{I}k}\right) = \left([{}^0\hat{\boldsymbol{p}}_k^T\ {}^0\hat{\boldsymbol{q}}_k^T]^T,\ d({}^0\hat{\boldsymbol{P}}_{\boldsymbol{p}_k}, {}^0\hat{\boldsymbol{P}}_{\boldsymbol{q}_k})\right) \tag{32}
$$

$$
\left({}^0\hat{\boldsymbol{x}}_{\boldsymbol{II}k}, {}^0\hat{\boldsymbol{P}}_{\boldsymbol{II}k}\right) = {}^0\boldsymbol{\Upsilon}_{1,k}\left([{}^1\hat{\boldsymbol{p}}_{\boldsymbol{k}}{}^T\ {}^1\hat{\boldsymbol{q}}_{\boldsymbol{k}}{}^T]^T,\ d({}^1\hat{\boldsymbol{P}}_{\boldsymbol{p}_k}, {}^1\hat{\boldsymbol{P}}_{\boldsymbol{q}_k})\right) \tag{33}
$$

where ${}^0\hat{\boldsymbol{x}}_{\boldsymbol{I}k}$ and ${}^0\hat{\boldsymbol{x}}_{\boldsymbol{II}k}$ respectively denote the state estimate with respect to $\{S_0\}$ and the one with respect to $\{S_1\}$ mapped into $\{S_0\}$, $d(\cdot)$ denotes $diag(\cdot)$ and ${}^0\boldsymbol{\Upsilon}_{1,k}$ denotes the mapping function of the couple $\left({}^1\hat{\boldsymbol{x}}_k, {}^1\hat{\boldsymbol{P}}_k\right)$ from $\{S_1\}$ to $\{S_0\}$.
We indicate with $\left({}^0\hat{\boldsymbol{\mathcal{X}}}_k, {}^0\hat{\boldsymbol{\mathcal{P}}}_k\right)$ the overall estimate at time step k with respect to $\{S_0\}$. First, we replace in ${}^0\hat{\boldsymbol{x}}_{\boldsymbol{I},\boldsymbol{II}_k}$ ${}^0\hat{\boldsymbol{q}}_{\boldsymbol{I},\boldsymbol{II}_k}$ with the vector ${}^0\hat{\boldsymbol{v}}_{\boldsymbol{I},\boldsymbol{II}_k}$. Then we compute the overall estimate error covariance matrix and obtain the overall state estimate by the following weighted sum:

$$
\begin{cases}
{}^0\hat{\boldsymbol{\mathcal{P}}}_k = \left({}^0\hat{\boldsymbol{P}}_{\boldsymbol{I}_k}{}^{-1} + {}^0\hat{\boldsymbol{P}}_{\boldsymbol{II}_k}{}^{-1}\right)^{-1} \\
{}^0\hat{\boldsymbol{\mathcal{X}}}_k = {}^0\hat{\boldsymbol{\mathcal{P}}}_k \left({}^0\hat{\boldsymbol{P}}_{\boldsymbol{I}_k}{}^{-1}\, {}^0\hat{\boldsymbol{x}}_{\boldsymbol{I}k} + {}^0\hat{\boldsymbol{P}}_{\boldsymbol{II}_k}{}^{-1}\, {}^0\hat{\boldsymbol{x}}_{\boldsymbol{II}k}\right)
\end{cases} \tag{34}
$$

The quaternion estimate is derived from the last three components of ${}^0\hat{\boldsymbol{\mathcal{X}}}_k$. At this point we have the object's pose estimate with respect to the base reference frame and can provide it to the robot's end-effector, as shown in figure (3).

III. EXPERIMENTAL RESULTS

The effectiveness of our framework was first tested on simulated data. We then practically applied our algorithm to the visual servoing of a Baxter Robot of Rethink Robotics [30] with the aim of picking an industrial part positioned on a table (Figure 4). For the purpose we used one of its two arms with the attached eye-in-hand camera, the second eye-in-hand camera fixed in the workspace, and an external camera positioned on the robot's torso. We preliminarily chose the reference covariance values of the measurement noise associated to each camera by collecting several pose measures of an object at a distance of 1.0 m, on the basis of a ground truth given by an optical marker. In the same way we determined the process noise covariance associated to the arm's motion by collecting several measures of the end effector's pose with the arm left still. The process noise considered in the filtering process with respect to the base frame was set to an arbitrary small value. Table I illustrates the obtained values, where $\boldsymbol{\sigma}_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}}$ is expressed in meters and $\boldsymbol{\sigma}_{\boldsymbol{v_0},\boldsymbol{v_1},\boldsymbol{v_2}}$ in radiants. The image streamings and the reference systems monitoring were managed together by means of the ROS operating system [31]. The estimation rate is dictated by the rate of the cameras, which is 30 Hz. Figure (5) shows the image scenes observed by the cameras. The industrial piece to be detected is positioned on the table next to other objects. The yellow bounding box puts the region where the object of interest has been detected in evidence. Figures (6) and (7) show the position and quaternion components of the object's pose estimate obtained with the simulated data. In each subplot we can see the two parallel estimates of the single component and the final overall estimate resulting from their combination. The combination of the two estimates produces a final estimate with a lower variance and a low error. Table II shows the results obtained on the real system. The table shows the standard deviation of the two parallel estimates, the standard deviation of the final estimate and the mean final estimate error. The percentage of picking successes goes around 90%.

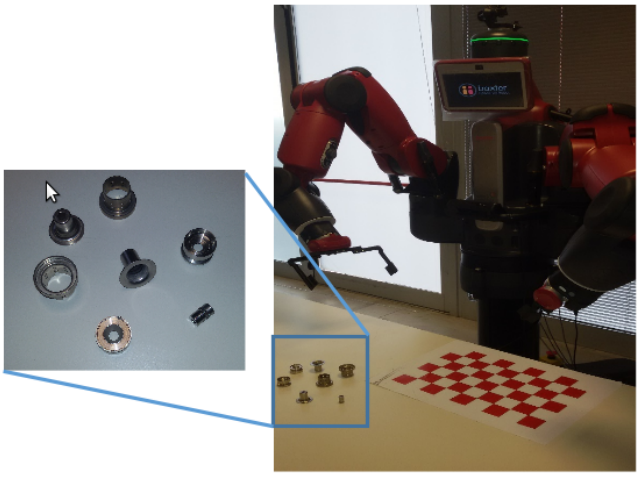

Fig. 4. Baxter in front of the table where the industrial pieces have to be picked.

IV. CONCLUSIONS

We presented a multi-camera framework conceived for a position-based visual servoing of a collaborative robot that has to manipulate untextured industrial pieces. The

Fig. 5. Camera images and detected object. From left to right: Image from the eye-in-hand moving camera attached to the robot's end-effector, image from the eye-in-hand fixed camera and image from the external camera on tje robot's torso.

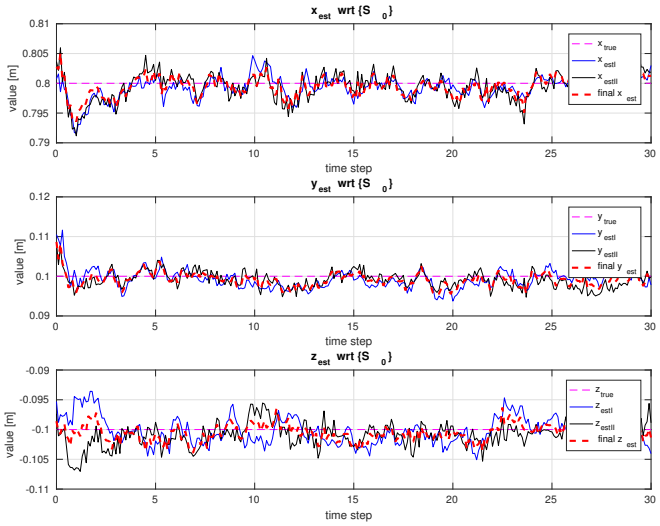

Fig. 6. Position estimation resulting x, y and z components. The magenta dashed line is the ground truth, the blue line is the specific component of ${}^0\hat{\boldsymbol{x}}_{\boldsymbol{I}}$, the black line is the component of ${}^0\hat{\boldsymbol{x}}_{\boldsymbol{II}}$ and the red dashed line is the component of final estimate ${}^0\hat{\boldsymbol{\mathcal{X}}}$.

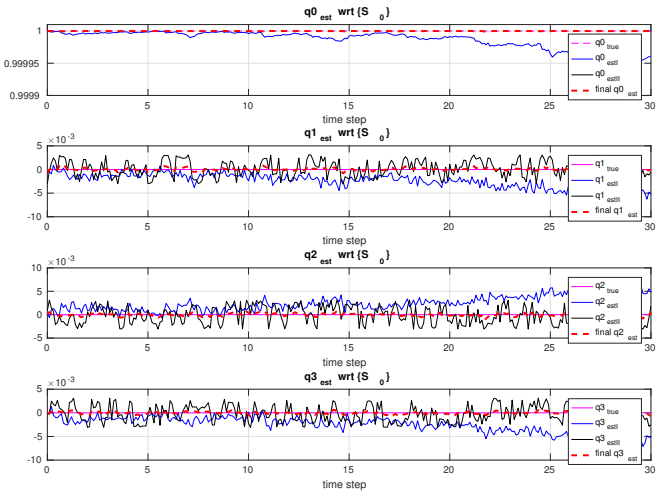

Fig. 7. Quaternion estimation resulting q_0, q_1, q_2 and q_3 components. The magenta dashed line is the ground truth, the blue line is the specific component of ${}^0\hat{\boldsymbol{x}}_{\boldsymbol{I}}$, the black line is the component of ${}^0\hat{\boldsymbol{x}}_{\boldsymbol{II}}$ and the red dashed line is the component of final estimate ${}^0\hat{\boldsymbol{\mathcal{X}}}$.

TABLE I

COVARIANCE VALUES OBTAINED EXPERIMENTALLY

	$\boldsymbol{\sigma_x}$	$\boldsymbol{\sigma_y}$	$\boldsymbol{\sigma_z}$	$\boldsymbol{\sigma_{v_0}}$	$\boldsymbol{\sigma_{v_1}}$	$\boldsymbol{\sigma_{v_2}}$
${}^0\boldsymbol{Q}$	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-3}
${}^2\boldsymbol{R}$	0.01	0.01	0.01	0.01	0.01	0.01
${}^3\boldsymbol{R}$	0.01	0.01	0.01	0.01	0.01	0.01
${}^1\boldsymbol{Q}$	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-3}
${}^1\boldsymbol{R}$	0.01	0.01	0.01	0.01	0.01	0.01

TABLE II

ESTIMATES STANDARD DEVIATION AND FINAL ESTIMATE ERROR

	x	y	z	v_0	v_1	v_2
std$_{\text{I}}$	0.008	0.008	0.01	0.004	0.006	0.004
std$_{\text{II}}$	0.01	0.009	0.01	0.004	0.005	0.004
std$_{\text{final}}$	0.007	0.007	0.008	0.002	0.004	0.003
mean error	0.006	0.006	0.007	0.006	0.005	0.006

framework includes an object detection and approximate localization stage, followed by a stochastic recursive pose estimation process that guides the robot motion control. The system was first tested through simulative experiments and then on a Baxter Robot. Experimental results show low pose estimate errors and high picking success, underlining the effectiveness of our approach. It should be noticed that the framework choices are not platform dependent and can be generalized to any kind of robot, regardless of its kinematics, and any number of eye-to-hand and eye-in-hand cameras.

ACKNOWLEDGMENT

The described material is based on the work carried out within the TAUM Project on transfer of human abilities to robots, co-financed by the European Structural Fund for Regional Development, programme POR CReO FESR 2007-2013 of Regione Toscana.

REFERENCES

[1] G. J. Agin, *Real time control of a robot with a mobile camera*. SRI International, 1979.

[2] D. Balek and R. Kelley, "Using gripper mounted infrared proximity sensors for robot feedback control," in *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, vol. 2. IEEE, 1985, pp. 282–287.

[3] F. Chaumette, P. Rives, and B. Espiau, "Positioning of a robot with respect to an object, tracking it and estimating its velocity by visual servoing," in *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*. IEEE, 1991, pp. 2248–2253.

[4] A. De la Escalera and J. M. Armingol, "Automatic chessboard detection for intrinsic and extrinsic camera parameter calibration," *Sensors*, vol. 10, no. 3, pp. 2027–2044, 2010.

[5] F. Chaumette, "Potential problems of stability and convergence in image-based and position-based visual servoing," in *The confluence of vision and control*. Springer, 1998, pp. 66–78.

[6] R. M. Haralick, D. Lee, K. Ottenburg, and M. Nolle, "Analysis and solutions of the three point perspective pose estimation problem," in *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on*. IEEE, 1991, pp. 592–598.

[7] D. DeMenthon and L. S. Davis, "Exact and approximate solutions of the perspective-three-point problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 11, pp. 1100–1105, 1992.

[8] M. Dhome, M. Richetin, J.-T. Lapreste, and G. Rives, "Determination of the attitude of 3d objects from a single perspective view," *IEEE transactions on pattern analysis and machine intelligence*, vol. 11, no. 12, pp. 1265–1278, 1989.

[9] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[10] D. G. Lowe *et al.*, "Fitting parameterized three-dimensional models to images," *IEEE transactions on pattern analysis and machine intelligence*, vol. 13, no. 5, pp. 441–450, 1991.

[11] R. R. Goldberg, "Constrained pose refinement of parametric objects," *International Journal of Computer Vision*, vol. 13, no. 2, pp. 181–211, 1994.

[12] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE transactions on robotics and automation*, vol. 12, no. 5, pp. 651–670, 1996.

[13] F. Chaumette and S. Hutchinson, "Visual servo control. i. basic approaches," *IEEE Robotics & Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.

[14] A. Sharma, I. Wadhwa, and R. Kala, "Monocular camera based object recognition and 3d-localization for robotic grasping," in *ISPCC*. IEEE, 2015, pp. 225–229.

[15] I. Siradjuddin, L. Behera, T. M. McGinnity, and S. Coleman, "A position based visual tracking system for a 7 dof robot manipulator using a kinect camera," in *Neural Networks (IJCNN), The 2012 International Joint Conference on*. IEEE, 2012, pp. 1–7.

[16] U. Khan, I. Jan, N. Iqbal, and J. Dai, "Uncalibrated eye-in-hand visual servoing: an lmi approach," *Industrial Robot: An International Journal*, vol. 38, no. 2, pp. 130–138, 2011.

[17] O. Kermorgant and F. Chaumette, "Multi-sensor data fusion in sensor-based control: application to multi-camera visual servoing," in *IEEE ICRA*, 2011, pp. 4518–4523.

[18] Y. Wang, B. Zuo, and H. Lang, "Vision based robotic grasping with a hybrid camera configuration," in *Systems, Man and Cybernetics (SMC), 2014 IEEE International Conference on*. IEEE, 2014, pp. 3178–3183.

[19] R. C. Luo, S.-C. Chou, X.-Y. Yang, and N. Peng, "Hybrid eye-to-hand and eye-in-hand visual servo system for parallel robot conveyor object tracking and fetching," in *Industrial Electronics Society, IECON 2014-40th Annual Conference of the IEEE*. IEEE, 2014, pp. 2558–2563.

[20] M. Marshall and H. Lipkin, "Kalman filter visual servoing control law," in *Mechatronics and Automation (ICMA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 527–5324.

[21] V. Lippiello, B. Siciliano, and L. Villani, "Position-based visual servoing in industrial multirobot cells using a hybrid camera configuration," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 73–86, 2007.

[22] S. Hinterstoisser, C. Cagniart, S. Ilic, P. Sturm, N. Navab, P. Fua, and V. Lepetit, "Gradient response maps for real-time detection of textureless objects," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 5, pp. 876–888, 2012.

[23] M. Woo, J. Neider, T. Davis, and D. Shreiner, *OpenGL programming guide: the official guide to learning OpenGL, version 1.2*. Addison-Wesley Longman Publishing Co., Inc., 1999.

[24] P. Sturm, "Pinhole camera model," in *Computer Vision*. Springer, 2014, pp. 610–613.

[25] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010.

[26] R. E. Kalman *et al.*, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[27] S. J. Julier and J. K. Uhlmann, "New extension of the kalman filter to nonlinear systems," in *AeroSense'97*. International Society for Optics and Photonics, 1997, pp. 182–193.

[28] E. Kraft, "A quaternion-based unscented kalman filter for orientation tracking," in *Proceedings of the Sixth International Conference of Information Fusion*, vol. 1, 2003, pp. 47–54.

[29] X. Pennec, "Computing the mean of geometric features application to the mean rotation," 1998.

[30] R. Robotics, "Baxter research robot," *http://cdn-staging. rethinkrobotics. com/wp-content/uploads/2014/08/BRR*, vol. 9, p. 13, 2013.

[31] M. Quigley, K. Conley *et al.*, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, 2009.