

# Visual Servoing Using Trifocal Tensor

Marwan Osman

University of Burgundy  
71200 Le Creusot, France

marwan.aosman@gmail.com

François Chaumette

INRIA Rennes-Bretagne Atlantique and IRISA  
35042 Rennes, France

francois.Chaumette@irisa.fr

**Abstract**— Visual servoing is an approach for controlling the motion of a robotic system from visual measurements. Many works have been realized in the past in this area, and is mainly divided into two main categories: Image-based, and Pose-based Visual Servoing. The trifocal tensor is well known in computer vision for tracing geometric information from three images of the same scene. The purpose of this thesis is to design a visual servoing method of a 6-DOF robot based on the trifocal tensor coefficients. Few studies where conducted on this area but they didn't provide a generic analytical solution for 6-DOF robots. This method differs than the two main visual servoing approaches as the control loop is closed over projective measures, which are the trifocal tensor elements. These projective measures are found directly from images across three views, without explicitly recovering the camera pose or directly closing the loop in the image space. The trifocal tensor geometric model is more robust than the two view geometry models as it involves the information given by a third view, and the set of correspondences obtained is more robust to outliers.

## I. INTRODUCTION

Visual Servoing is a growing active field of research. Recent advances in robotics and computer vision domains has led to the emergence of this vision-based closed loop control methods to control a robot manipulator or a mobile robot through vision acquired by a camera. Many works have studied closing the control loop over visual features computed either from the image space (IBVS) or the estimated 3D pose of the scene (PBVS). There exist some approaches which cannot be considered part of the main two classes of Visual Servoing approaches IBVS and PBVS. These approaches differ than the two main visual servoing classes by closing the control loop over projective measures, which are not computed directly from the image space or the 3D pose. Epipolar geometry can be used to estimate depth, which appears in the interaction matrix of point features [9]. Chesi *et al.* controlled a holonomic mobile robot from a partially calibrated camera using the symmetry of epipolar geometry without point correspondences [10]. Benhimane and Malis developed a homography-based approach without reconstructing any 3D parameters [11]. Lopez-Nicolas *et al.* designed a homography-based controller which considers the non-holonomic constraints [12]. These methods use the two-view geometry between the observed and desired views and ignore their relation with the initial view. Epipolar geometry is not well-conditioned if the features are coplanar or the baseline is short, while homography-based approaches require dominant planes.

Recent works introduced using the three-views geometry in the visual servoing loop. The trifocal tensor encapsulates the intrinsic geometry between the three views, and It is analogous for the fundamental matrix in stereo vision systems. The trifocal tensor geometric model is more robust than the two view geometry models as it involves the information given by a third view, and the set of correspondences obtained is more robust to outliers. The application of trifocal tensor in visual servoing has been neglected until recently. Becerra and Sagues used a simplified trifocal tensor as measurement and estimate and track the pose of a non-holonomic mobile robot with Extended Kalman Filter (EKF) [8]. Lopez-Nicolas *et al.* used the constrained camera motion on a mobile robot and linearized the input-output space for control [6]. This approach provided an analytical interaction matrix, which relates the variations of 9 elements of the trifocal tensor to the motion velocities. Shademan used the trifocal tensor for 6-DOF visual servoing [7]. Unlike Lopez-Nicolas, Shademan didn't provide an analytical derivation for the interaction matrix, it was estimated numerically and used in the control law. Shademan argued that using the trifocal tensor to control 6-DOF visual servoing loop was not introduced prior to his work due to the difficulty in linearizing the input-output space in the case of the generalized 6-DOF camera motions.

The work of this paper is based on the previous approaches introduced by Lopez-Nicolas [6] and Shademan [7]. An approach to incorporate the trifocal tensor estimated from the three-views geometry into the visual servoing control loop task has been presented in this work. This approach presents a generalized 6-DOF visual servoing task, with the control loop being closed over projective measures, namely the trifocal tensor coefficients. To the best of our knowledge, this approach is the first to propose a fully analytical design for a visual servoing task based on the trifocal tensor. First, we present brief background on the Trifocal Tensor and Visual Servoing. Next, we present the analytical development of the new approach. We discuss the obtained results. Finally, we present an overall conclusion for the work in hand, as well as possible improvements and future work.

## II. BACKGROUND

### A. Trifocal Tensor

The Trifocal Tensor is a  $3 \times 3 \times 3$  array of numbers that incorporates all projective geometric relationships among three views [1]. It relates the coordinates of corresponding points or lines in three views, being independent of the scene

structure and depending only on the relative pose among the three views and their intrinsic calibration parameters. In this method, we assume the camera intrinsic parameters are known. It means before computing the tensor from image correspondences, matching points have to be transformed to calibrated coordinates. The geometric basis for the trifocal tensor can be deduced from the incidence relationship of three corresponding lines projecting a line in 3D-space. The computation is described in details in [1]. Let the Camera positions be  $C_{c^*}, C_c, C_i$  for the desired, current, and initial camera positions respectively. And their projection matrices be  $[I|0]$ ,  $[{}^c\mathbf{R}_{c^*}|{}^c\mathbf{t}_{c^*}]$ ,  $[{}^i\mathbf{R}_{c^*}|{}^i\mathbf{t}_{c^*}]$ . The Tensor relation can then be expressed as follows

$$\mathcal{T}_{(jkl)} = {}^cR_{c^*(kj)} {}^i t_{c^*(l)} - {}^c t_{c^*(k)} {}^i R_{c^*(lj)} \quad (1)$$

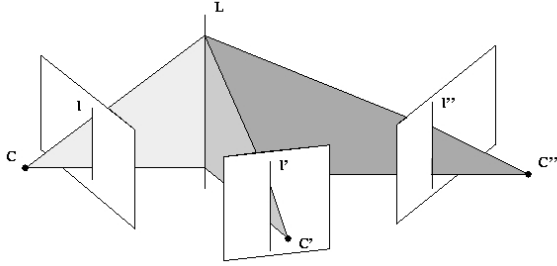


Fig. 1: Trifocal geometry of three views

$\mathcal{T}$  is then the trifocal tensor relating the 3 views together. It has 27 elements. There are 26 independent ratios apart from the common overall scaling factor of the tensor. However, the tensor has only 18 independent parameters. Each of the three camera projection matrices has 11 parameters which makes 33 in total. However, 15 parameters must be subtracted to account for the projective world frame, thus leaving 18 parameters.

Since the trifocal tensor embeds the geometry of the three cameras in the scene, this implies that the camera matrices may be computed from the trifocal tensor up to a projective ambiguity [1]. First, the epipoles  $e', e''$  are retrieved. Let  $u_i$  and  $v_i$  be the left and right null-vectors respectively of  $\mathcal{T}_i$ , i.e.:  $u_i^T \mathcal{T}_i = 0^T, \mathcal{T}_i v_i = 0$ . Then the epipoles are obtained as the null vectors of the following  $3 \times 3$  matrices:

$$e'^T [u_1, u_2, u_3] = 0 \text{ and } e''^T [v_1, v_2, v_3] = 0 \quad (2)$$

Next, to retrieve the camera matrices  $P', P''$ , the epipoles are first normalized to unit norm, and we have:

$$P' = [[\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3] e' | e'] \text{ and } P'' = [(e'' e''^T - I) [\mathcal{T}_1^T, \mathcal{T}_2^T, \mathcal{T}_3^T] e' | e''] \quad (3)$$

The trifocal tensor can be computed from feature correspondences across the three views. Computation of tensor using correspondences of points or lines has been studied extensively in literature. The main linear algorithm presented by Hartley [1] is the straightforward method to compute the tensor. Other methods make use of RANSAC (Random

Sample Consensus) to enable the tensor estimation to be robust to outliers that usually cause the linear algebraic method to fail [13]. Each triplet of corresponding image points gives 4 equations linearly independent. Therefore, a minimum set of 7 correspondences of non-planar points are needed for the trifocal tensor computation to uniquely determine the 27 entries of the tensor matrix. This method suffers from two issues however: the tensor is parameterized by all its entries and this does not take into account the tensor geometrical constraints, therefore, there is always a risk of obtaining an invalid tensor; and, using all available point correspondences causes the tensor to be significantly affected by the presence of strong outliers. Hence, an algebraic minimization is needed there after, to use the linear solution as an initial estimate and re-parameterize it by the 24 entries of the projection matrices  $P'$  and  $P''$ . The desired tensor is found by minimizing the residual error. The latter algorithm leads to a geometrically valid tensor. However, the main weakness of the algebraic minimization method is that all point correspondences are involved in the tensor computation and still suffers from outliers in practice.

## B. Visual Servoing

Visual Servoing refers to the family of closed loop control techniques to control the degrees of freedom of an actuated system with visual feedback [2]. The vision data may be acquired from a camera that is mounted directly on the robot, or the camera can be fixed in the workspace so that it can observe the robot motion from a stationary configuration. The goal of many robotic applications is to place the robot at a desired configuration to manipulate an object in the environment. The objective of closed loop control techniques is to minimize an error function  $e(t)$  defined as

$$e(t) = s(t) - s^* \quad (4)$$

where  $s$  is a vector of visual features, and the vector  $s^*$  has the desired values of these features at the desired configuration.

Then, to design the control scheme, the relationship between the time variation of the features  $s$  and the camera velocity has to be found. Noting the spatial velocity of the camera  $u_c = (v_c, \omega_c)$ , where  $v_c$  is the instantaneous linear velocity of the origin of the camera frame, and  $\omega_c$  is the instantaneous angular velocity of the camera frame. The relationship between  $\dot{s}$  and  $u_c$  is given by

$$\dot{s} = L_s u_c \quad (5)$$

where  $L_s$  is called the **interaction matrix** related to  $s$ . This matrix relates the rate of changes of the image feature velocities  $\dot{s}$  to the rate of change of the pose parameters which is the camera velocity. From (4) and (5), the relationship between the camera velocity and the derivative of the error is

$$\dot{e} = L_s u_c \quad (6)$$

Considering  $u_c$  as the input to the robot controller, to ensure an exponential decoupled decrease of the error, we would like to have  $\dot{e} = -\lambda e$ , using (6) we obtain the following control law equation

$$u_c = -\lambda L_s^+ e \quad (7)$$

where  $L_s^+$  is the pseudo-inverse of  $L_s$ . This is the basic visual servoing closed loop system framework. Different visual servoing approaches differ only in the selection of the visual features  $s$  and deriving their corresponding interaction matrices  $L_s$ .

There are three main classes of visual servoing: Image-based Visual Servoing (IBVS), Pose-based Visual Servoing (PBVS), and Hybrid Visual Servoing (HVS) [2] [3]. In IBVS, the features are computed from the 2D image data, while in PBVS, the features are computed from the estimated pose from image measurements. HVS use a combination of 2D and 3D visual features. The choice of the visual features is important and generally affects the performance of the visual servoing system. The IBVS approach is more robust to modeling and measurement noise than PBVS. However, the IBVS approach only guarantees local asymptotic stability. It also suffers from degenerate cases and may fail to reach the goal state. PBVS approach ensure global asymptotic stability if the pose estimation is perfect. The pose estimation algorithm must be realized in real-time to calculate the error at every control iteration, and hence more computationally expensive than IBVS approach. However, PBVS approach is more sensitive to image measurement noise and outliers. It usually also requires full knowledge of the geometric object model, and the camera intrinsic calibration.

### III. METHODOLOGY

To formulate the control law of the trifocal tensor based visual servoing (8), we need to establish the model of the system and find the relation between the input of the control and the trifocal tensor coefficients. That is to find the interaction matrix relating the control input and the tensor coefficients derivatives. The error will be the difference between the current tensor and the desired tensor values.

$$\begin{aligned} \dot{\mathcal{T}}_{(jkl)} &= L_{\mathcal{T}_{(jkl)}} u_c \\ u_c &= -\lambda L_{\mathcal{T}_{(jkl)}}^+ e \end{aligned} \quad (8)$$

#### A. Interaction Matrix

First, The derivatives of all the trifocal tensor elements with respect to time are generally as following:

$$\begin{aligned} \dot{\mathcal{T}}_{(jkl)} &= {}^c \dot{R}_{c^*(kj)} {}^i t_{c^*(l)} + {}^c R_{c^*(kj)} {}^i \dot{t}_{c^*(l)} \\ &\quad - {}^c \dot{t}_{c^*(k)} {}^i R_{c^*(lj)} - {}^c t_{c^*(k)} {}^i \dot{R}_{c^*(lj)} \end{aligned} \quad (9)$$

Since our initial camera  $C_i$  is fixed, the elements of the derivatives of its rotation matrix and its transpose vector are equal to zero, i.e.:  ${}^i \dot{t}_{c^*(l)} = {}^i \dot{R}_{c^*(jl)} = 0$ . Our trifocal tensor

elements derivative is then simplified to:

$$\dot{\mathcal{T}}_{(jkl)} = {}^c \dot{R}_{c^*(kj)} {}^i t_{c^*(l)} - {}^c \dot{t}_{c^*(k)} {}^i R_{c^*(lj)} \quad (10)$$

The spatial velocity of the camera is  $u_c = (v_c, \omega_c)^T$ , where  $v_c$  and  $\omega_c$  are the translational and rotational velocities of the camera. From the geometry of the scene, we can deduce the following relationships:

$$\begin{aligned} [\omega_c]_{\times} &= {}^c R_c^T {}^c \dot{R}_c = -{}^c \dot{R}_c^T {}^c R_c \\ {}^c \dot{R}_c^T &= -[\omega_c]_{\times} {}^c R_c^T \\ {}^c \dot{R}_c^T &= -[\omega_c]_{\times} {}^c R_{c^*} \\ {}^c \dot{R}_{c^*} &= -[\omega_c]_{\times} {}^c R_{c^*} \end{aligned}$$

Similarly, we have [2]:

$$\begin{aligned} {}^c \dot{t}_c &= {}^c R_c v_c, {}^c t_{c^*} = -{}^c R_{c^*} {}^c t_c \\ {}^c \dot{t}_{c^*} &= -{}^c \dot{R}_{c^*} {}^c t_c - {}^c R_{c^*} {}^c \dot{t}_c \\ {}^c \dot{t}_{c^*} &= [\omega_c]_{\times} {}^c R_{c^*} {}^c t_c - {}^c R_{c^*} {}^c R_c v_c \\ {}^c \dot{t}_{c^*} &= [\omega_c]_{\times} {}^c t_{c^*} - v_c \\ {}^c \dot{t}_{c^*} &= [{}^c t_{c^*}]_{\times} \omega_c - v_c \\ {}^c \dot{t}_{c^*(k)} &= [-I][{}^c t_{c^*}]_{\times} u \end{aligned}$$

Substituting back into the tensor derivation (10), we get:

$$\begin{aligned} \dot{\mathcal{T}}_{(jkl)} &= -([\omega_c]_{\times} {}^c R_{c^*})_{(kj)} {}^i t_{c^*(l)} \\ &\quad - ([{}^c t_{c^*}]_{\times} \omega_c - v_c)_{(k)} {}^i R_{c^*(lj)} \\ \dot{\mathcal{T}}_{(jkl)} &= -\left(\sum_m [\omega_c]_{\times (km)} {}^c R_{c^*(mj)}\right) {}^i t_{c^*(l)} \\ &\quad - (([{}^c t_{c^*}]_{\times} \omega_c)_{(k)} - v_{c(k)}) {}^i R_{c^*(lj)} \end{aligned} \quad (11)$$

From (1) and (11), we can compute the derivatives for all the coefficients, and deduce this compact form:

$$\dot{\mathcal{T}}_{(jkl)} = R_{i(lj)} v_{c(k)} - \sum_m [\omega_c]_{\times (km)} \mathcal{T}_{(jml)} \quad (12)$$

To control the six degrees of freedom, at least six tensor coefficients are necessary. However, to avoid singularities, more than six tensor coefficients are considered. The interaction matrix  $L_{\mathcal{T}_{(jkl)}}$  taking into account all the tensor coefficients can be deduced from (12). It is important to notice that pose of the initial camera location is necessary for the computation of the interaction matrix as well as the normalization factor. It is also to be noticed there exists a decoupling between the translational velocities in the resulting interaction matrix. This decoupling ensures to obtain smooth camera trajectories for the motion in the 3D space.

#### B. Tensor Normalization

Since the trifocal tensor is computed up to a scale factor, we propose a normalization step to get a fixed scale. The normalization factors  $\mathcal{T}_{kN}$  are used to obtain the normalized tensor  $\mathcal{T}_{jkl}$ . When the camera reaches the desired goal position, we can observe that the tensor coefficients are only

related to the translation of the camera at the initial position which is a constant value vector. Also from (12), the first column elements  $\mathcal{T}_{(111)}, \mathcal{T}_{(112)}, \mathcal{T}_{(113)}, \mathcal{T}_{(211)}, \mathcal{T}_{(212)}, \mathcal{T}_{(213)}, \mathcal{T}_{(311)}, \mathcal{T}_{(312)}, \mathcal{T}_{(313)}$  are respectively equal to  ${}^i t_{c^*(1)}, {}^i t_{c^*(2)}, {}^i t_{c^*(3)}, 0, 0, 0, 0, 0, 0$  at the desired goal position.

Choosing the elements  $\mathcal{T}_{(111)}, \mathcal{T}_{(112)}, \mathcal{T}_{(113)}$  as normalization factors would seem a good choice for our method, as we would be interested in coefficients converging to non-zero values. However, it was found that for some scene configurations, these three coefficients could have values equal to zero at the initial pose. Which means at normalization step, this would cause dividing by a zero value. To avoid this problem, we will use all the nine coefficients of the first column to be our first normalization factor. We can then normalize the elements  $T_{jkl}$  for  $k = 1$  as follows:

$$\begin{aligned} T_{jkl} &= \frac{\mathcal{T}_{jkl}}{\mathcal{T}_{1N}} \\ \mathcal{T}_{1N} &= (\mathcal{T}_{111}^2 + \mathcal{T}_{112}^2 + \mathcal{T}_{113}^2 + \mathcal{T}_{211}^2 + \mathcal{T}_{212}^2 + \mathcal{T}_{213}^2 \\ &\quad + \mathcal{T}_{311}^2 + \mathcal{T}_{312}^2 + \mathcal{T}_{313}^2)^{\frac{1}{2}} \end{aligned} \quad (13)$$

Unfortunately, this normalization factor is dependent on the tensor coefficients which varies with time. So we need to find also the derivative of the normalization factor. A general formula for  $\dot{T}_{(jkl)}$  can then be found as following

$$\begin{aligned} \dot{T}_{(jkl)} &= \frac{{}^i R_{c^*(lj)}}{\mathcal{T}_{kN}} v_{c(k)} - T_{(jkl)} \left( \sum_n \sum_m T_{nkm} \frac{{}^i R_{c^*(mn)}}{\mathcal{T}_{kN}} \right) v_{c(k)} \\ &\quad + T_{(jkl)} \left( \sum_n \sum_m T_{nkm} T_{nhm} \right) \omega_{(g)} \\ &\quad - T_{(jkl)} \left( \sum_n \sum_m T_{nkm} T_{ngm} \right) \omega_{(h)} \\ &\quad + \sum_m [\omega_c]_{\times(km)} T_{(jml)} \end{aligned}$$

where  $g = k\%3 + 1, h = (k + 1)\%3 + 1$

(14)

From (14), the final normalized interaction matrix can be computed. As we notice, this particular choice for the normalization factors has led to keep the decoupling property between the translational velocities, as each degree of freedom related tensor coefficients are normalized separately from other degrees of freedom. Other choices of different combinations of tensor coefficients as normalization factors, would introduce extra non-zero elements to the normalized interaction matrix and prevent keeping the decoupling property.

### C. Proposed Method

We have defined all the elements of the control law (8). The interaction matrix  $L_{T_{(jkl)}}$  is computed as indicated in (14). The error  $e$  is computed using the difference between the current tensor coefficients and the desired tensor coefficients,  $e = T_{(jkl)} - T_{(jkl)}^*$ . In Figure 2, we present the block diagram explaining the proposed trifocal tensor visual servoing

control loop. At initialization, the desired tensor  $T_{(jkl)}^*$  is computed from feature correspondences across three images obtained from the three camera poses: initial pose, desired pose, and current pose being equal to the desired pose.

The current tensor  $T_{(jkl)}$  is computed inside the visual servoing loop at each iteration. Then the interaction matrix is computed using the current tensor. For simplicity, we make the assumption that the initial pose is known and the interaction matrix can be computed directly. The new error value is computed along with the pseudo-inverse of the interaction matrix and fed back to the control law to compute the required velocities to drive the camera to the desired pose. The system converges and the loop is terminated when the camera reaches the desired pose. This is evaluated when the sum squared of the error reaches a value less than a defined threshold,  $1 \times 10^{-6}$  for example.

## IV. EXPERIMENTAL RESULTS

The proposed method was implemented in C++, using the ViSP library. ViSP is an open-source visual servoing framework library developed by INRIA and written in C++ [14]. It is a modular cross platform library that allows prototyping and developing applications using visual tracking and visual servoing techniques. The scene setup is defined with two poses for the camera: initial and desired, along with an object to be tracked in the scene. A trifocal tensor requires at least 7 non coplanar points for estimating a tensor. Throughout our experiments, we used two sets of 8 and 12 points. There are no much differences between the results of the two sets because we are not considering points mismatch and outliers throughout the experiments. The following results are for the set of 12 points.

### A. Experiment I: Pure Translation along x-axis, or y-axis

For first experiment, we consider a pure translational motion along one axis. It's the basic test that can be done to ensure the convergence of the control loop. We consider a small translation of  $0.1m$  along x-axis, then y-axis. Figure 3c shows the evolution of the camera velocities, and Figure 3d shows the evolution of the trifocal tensor coefficients error. Figure 3 shows the results of estimating the tensor through points correspondences. A translation along y-axis produces similar results.

We observe the camera trajectory is linear, which is satisfactory. However, the error is not exactly exponentially decreasing. Several tests were conducted to analyse the source of this behaviour, and it was found that using a subset of tensor coefficients for computing the error and the interaction matrix leads to a perfect exponentially decreasing error. A subset was chosen experimentally to handle this specific configuration, but different subsets are required for different configurations. Figure 4 shows a comparison between the error plots using all the tensor coefficients and using a subset of them.

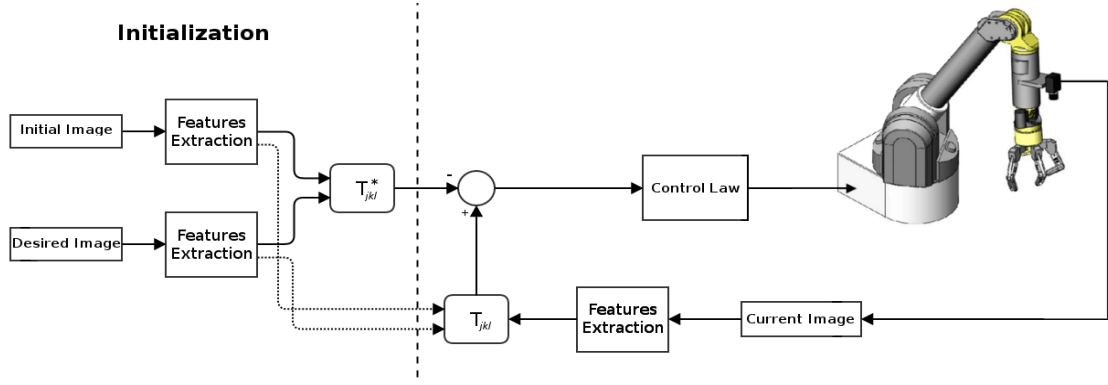


Fig. 2: Block diagram of the proposed Trifocal Tensor Visual Servoing loop control

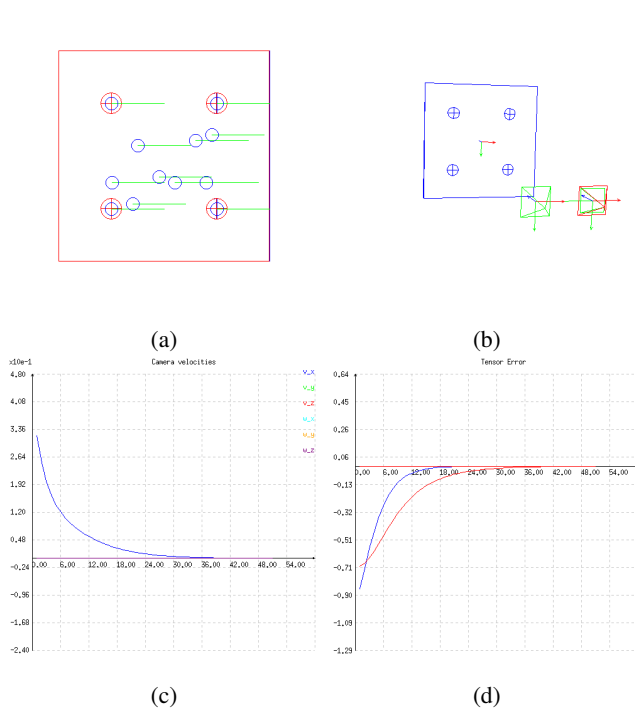


Fig. 3: Translation along x-axis using estimated tensor. (a) Image view. (b) Scene view. (c) Camera velocities. (d) Tensor coefficients error.

### B. Experiment II: Pure Translation along z-axis

Motion along the z-axis is usually more challenging than xyz motion in the IBVS approach. This is due to poor motion resolvability when the camera moves towards the feature points [18]. On the contrary, using the trifocal tensor features this motion is not different than other types of motions. Figure 5 show the results for the estimated tensor. Like a simple translation along x-axis or y-axis, a translation along z-axis also suffers the problem of selecting a subset of tensor coefficients.

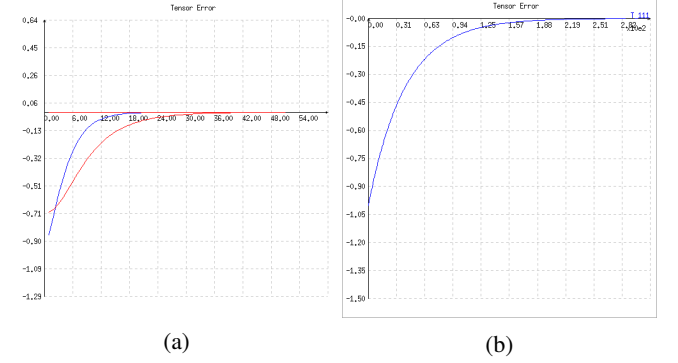


Fig. 4: Comparison between tensor coefficients error when (a) using all tensor coefficients (b) using a subset of tensor coefficients.

### C. Experiment III: Large rotation around z-axis

One of the most challenging IBVS configurations is the 180° rotation around the z-axis [2] [19]. This is due to the nature of the image-based control law which makes the camera retreat from the object instead of rotating around the z-axis. It is important to evaluate the visual servoing method for large z-axis rotations, close to 180°.

Here we consider a translation of 1m along z-axis and a 170° rotation along z-axis. As we can observe in Figure 6, the retreat problem did not occur. The image trajectories follow a spiral motion, which is exactly as expected due to the linear motion for the translation, and the rotational motion for the rotation.

### D. Experiment IV: Generic Motion

In this experiment, we choose a generic camera motion: translations of 0.25m, -0.3m, 0.2m, and rotations of 10°, -20°, 5° along x, y, and z axes respectively. The results in Figures 7 show smooth camera and image trajectories. However, we notice small oscillations in the camera velocities initially. Lopez-Nicolas [6] mentioned similar oscillations due to noise during image points extraction and matching. Although in our case we omit points mismatch and

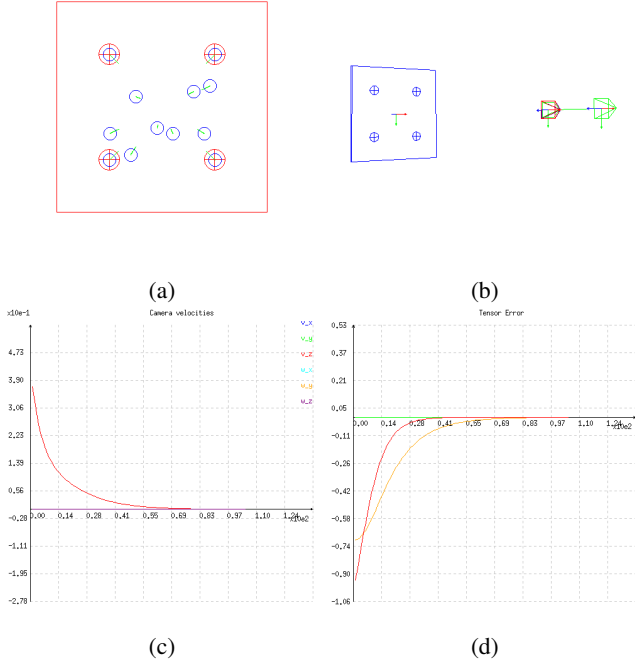


Fig. 5: Translation along z-axis using estimated tensor. (a) Image view. (b) Scene view. (c) Camera velocities. (d) Tensor coefficients error.

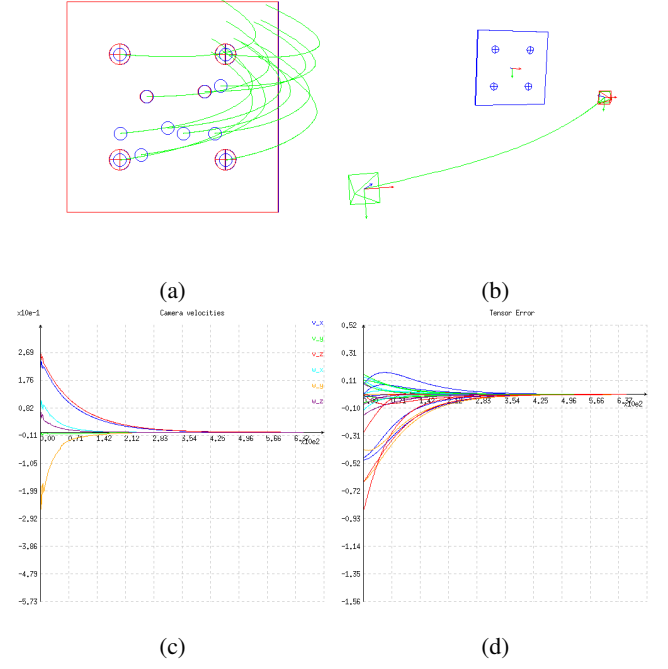


Fig. 7: Generic motion using estimated tensor. (a) Image view. (b) Scene view. (c) Camera velocities. (d) Tensor coefficients error.

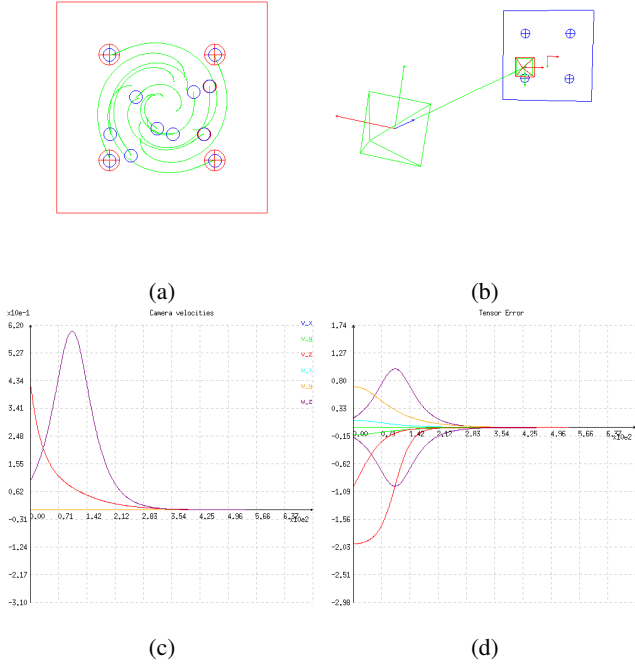


Fig. 6: Translation and large rotation along z-axis using estimated tensor. (a) Image view. (b) Scene view. (c) Camera velocities. (d) Tensor coefficients error.

outliers, this behaviour is most likely due to small numerical error in the trifocal tensor estimation.

## V. CONCLUSION AND FUTURE WORKS

### A. Conclusion

An approach to incorporate the trifocal tensor estimated from the three-views geometry into the visual servoing control loop task has been presented in this work. This approach presents a generalized 6-DOF visual servoing task, with the control loop being closed over projective measures, namely the trifocal tensor coefficients. To the best of our knowledge, this approach is the first to propose a fully analytical design for a visual servoing task based on the trifocal tensor. Previous works either provided an analytical approach for a 3-DOF non-holonomic mobile robot, or provided a full 6-DOF approach but with an interaction matrix estimated numerically rather than analytically.

The experiments showed that this approach is working practically with very satisfactory results. In addition, this approach does not suffer from some of the problems existing in IBVS methods as like the retreat problem.

### B. Future Works

As presented in the section IV, it was found that using all the trifocal tensor coefficients inside the visual servoing loop does not produce the best results in terms of perfectly exponentially decreasing of the tensor error values. Thus, an automatic optimality selection criteria is needed to dynamically choose the most relevant tensor coefficients. For

example, the singular value decomposition of the interaction matrix can reveal which degrees of freedom are most apparent. By selecting features and designing controllers that maximize these measures, the performance of the visual servoing system can be improved. These concepts have been discussed in details in [16] and [17].

Simulation and practical experimentation are not enough for judging the proposed method. We need to prove the existence of only one equilibrium state at the desired pose location. The analysis of the control law can be studied using Lyapunov stability analysis method [15].

We observe in (12) that the interaction matrix depends on the knowledge of the initial camera rotation and the normalization factors. In our experimentations, we made an assumption that the initial pose is known from the setup of the simulation. Practically, this information might not be available, and the initial pose needs to be estimated. The initial pose can be retrieved by computing the equivalent projection matrix for the estimated trifocal tensor as shown in (3). However, the estimation will still be up to a scale value and not the real value. This scale problem needs to be addressed properly during the tensor normalization step to avoid the scale effects on computing the correct values for the interaction matrix.

The current method assumes the cameras have already been calibrated before-hand. Practically, this may not be the case, so we need to extend the method to cover partially or totally uncalibrated cameras. The introduction of the camera intrinsic parameters matrix  $K$  into the formulas, will affect the obtained tensor computation, and hence, the corresponding interaction matrix.

## REFERENCES

- [1] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.
- [2] F. Chaumette and S. Hutchinson, "Visual servo control. i. basic approaches," *Robotics & Automation Magazine, IEEE*, vol. 13, no. 4, pp. 82–90, 2006.
- [3] —, "Visual servo control, part ii: Advanced approaches," *IEEE Robotics and Automation Magazine*, vol. 14, no. 1, pp. 109–118, 2007.
- [4] T. Rowland and E. W. Weisstein, "Tensor," <http://mathworld.wolfram.com/Tensor.html>, last visited on 03/02/2015.
- [5] C. Stover and E. W. Weisstein, "Einstein summation," <http://mathworld.wolfram.com/EinsteinSummation.html>, last visited on 03/02/2015.
- [6] G. López-Nicolás, J. J. Guerrero, and C. Sagüés, "Visual control through the trifocal tensor for nonholonomic robots," *Robotics and Autonomous Systems*, vol. 58, no. 2, pp. 216–226, 2010.
- [7] A. Shademan and M. Jagersand, "Three-view uncalibrated visual servoing," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 6234–6239.
- [8] H. Becerra and C. Sagues, "Pose-estimation-based visual servoing for differential-drive robots using the 1d trifocal tensor," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*. IEEE, 2009, pp. 5942–5947.
- [9] E. Malis and F. Chaumette, "2 1/2 d visual servoing with respect to unknown objects through a new estimation scheme of camera displacement," *International Journal of Computer Vision*, vol. 37, no. 1, pp. 79–97, 2000.
- [10] G. Chesi, D. Prattichizzo, and A. Vicino, "A visual servoing algorithm based on epipolar geometry," in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 1, 2001, pp. 737–742 vol.1.
- [11] S. Benhimane and E. Malis, "Homography-based 2d visual servoing," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, May 2006, pp. 2397–2402.
- [12] G. Lopez-Nicolas, C. Sagues, J. Guerrero, D. Kragic, and P. Jensfelt, "Nonholonomic epipolar visual servoing," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, May 2006, pp. 2378–2384.
- [13] P. H. Torr and A. Zisserman, "Robust parameterization and computation of the trifocal tensor," *Image and Vision Computing*, vol. 15, no. 8, pp. 591–605, 1997.
- [14] E. Marchand, F. Spindler, and F. Chaumette, "Visp for visual servoing: a generic software platform with a wide class of robot control skills," *Robotics Automation Magazine, IEEE*, vol. 12, no. 4, pp. 40–52, Dec 2005.
- [15] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot modeling and control*. Wiley New York, 2006, vol. 3.
- [16] B. Nelson and P. Khosla, "Force and vision resolvability for assimilating disparate sensory feedback," *Robotics and Automation, IEEE Transactions on*, vol. 12, no. 5, pp. 714–731, Oct 1996.
- [17] R. Sharma and S. Hutchinson, "Motion perceptibility and its application to active vision-based servo control," *Robotics and Automation, IEEE Transactions on*, vol. 13, no. 4, pp. 607–617, Aug 1997.
- [18] B. J. Nelson and P. K. Khosla, "Vision resolvability for visually servoed manipulation," *Journal of Robotic Systems*, vol. 13, no. 2, pp. 75–93, 1996.
- [19] 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.