Image Based Visual Servoing for Tumbling Objects

by

Abhijeet Kumar, Gunshi Gupta, Avinash Sharma, Madhava Krishna

in

International Conference on Intelligent Robots and Systems
(*IROS 2018*)

Madrid, Spain

Report No: IIIT/TR/2018/-1

Centre for Robotics
International Institute of Information Technology
Hyderabad - 500 032, INDIA
October 2018

Image Based Visual Servoing for Tumbling Objects

Mithun P*[1], Harit Pandya*[1], Ayush Gaud[1], Suril V. Shah[2] and K. Madhava Krishna[1]

Abstract— **Objects in space often exhibit a tumbling motion around the major inertial axis. In this paper, we address the image based visual servoing of a robotic system towards an uncooperative tumbling object. In contrast to previous approaches that require explicit reconstruction of the object and an estimation of its velocity, we propose a novel controller that is able to minimize the feature error directly in image space. This is achieved by observing that the feature points on the tumbling object follow a circular path around the axis of rotation and their projection creates an elliptical track in the image plane. Our controller minimizes the error between this elliptical track and the desired features, such that at the desired pose the features lie on the circumference of the ellipse. The effectiveness of our framework is exhibited by implementing the algorithm in simulation as well on a mobile robot.**

I. INTRODUCTION

One of the major areas of space science that seek immediate attention and commercial drive is providing On-Orbit Services (OSS) such as refueling of satellites, repair and refurbishment of disabled satellites, active debris removal etc. Moreover, these tasks are now becoming essential as the number of aging satellites has continued to increase [1]. These autonomous on-orbit services include intricate maneuvers, and hence, use of robots is inevitable for successful completion of these tasks [2], [3], [4]. Image-Based Visual Servoing (IBVS) is one the methods [5],[6], which provides autonomy in reaching the desired pose to provide robust grasp in comparison to conventional controls. Although approaches for the capture of fixed object [7] are well studied, servoing to a dynamic object by a robotic arm system are seldom reported in the literature.

Conventionally, the strategy for robotic capturing of a moving object is two-fold. It initially requires tracking and prediction of object motion over a period of time, and later a planning algorithm to execute robot motion to reach the desired pose. Earlier estimation approaches [8], [9] employed a sequence of stereo or monocular camera images and a kinematic model of object to estimate the kinematics and structure of a rigid body. To improve the accuracy of estimation, several researchers have proposed model based estimation methods for motion prediction. A capture strategy based on Kalman filter for tumbling satellite was initially studied in [10]. Here, the authors assumed that the target has a symmetric geometric shape and the target's size, shape and mass were all known. A marker-less 3D model-based tracking algorithm to the problem of satellite tracking using a monocular camera was initially studied in [11]. An absolute pose estimation that first uses LiDAR to generate a point cloud and then the Iterative Closest Point (ICP) algorithm to align the point cloud to 3D CAD model is discussed in [12], [13]. In a very recent approach [14], pose estimations of a tumbling object are obtained using incremental smoothing and mapping (iSAM) with the help of a stereo camera. The above-mentioned approaches aimed to estimate the structure and motion of the tumbling object either explicitly use a CAD model or employ reconstruction techniques. Which was further used to plan a trajectory in the Cartesian space. Visual servoing on the other hand directly controls the robot in the image space. This motivated the choice of image based visual servoing for capture of tumbling orbiting objects in this paper.

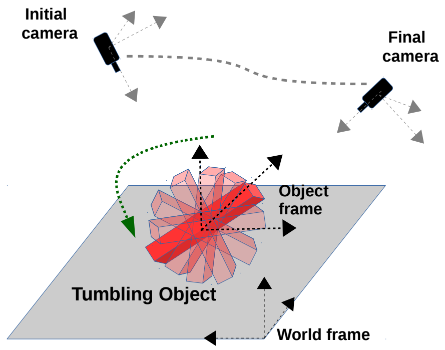
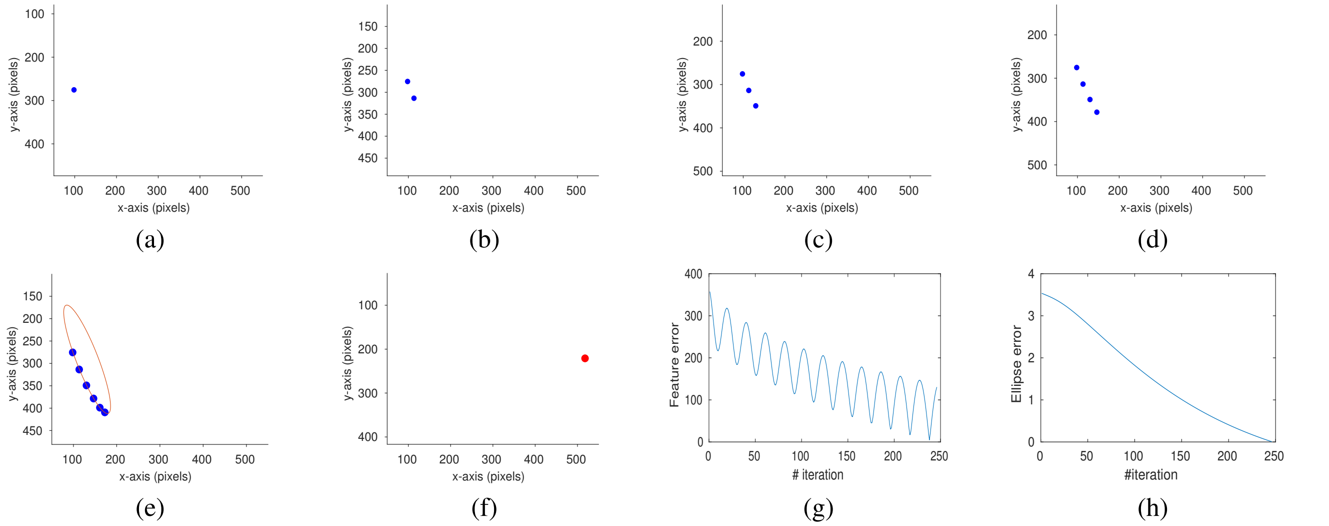


Fig. 1. **Aim:** Perform Image Based Visual Servoing using a 6-DOF camera towards an uncooperative tumbling object. The servoing task requires the camera to move to the desired view of target such that the error between proposed features and the desired stationary features become zero.

Visual servoing employs image information to control a robot, which plays an important role in autonomous robot navigation. One of the advantages of IBVS is that it only uses image features to control the robot without estimating relative pose. The features extracted from visual information are used as a feedback for the control loop [15], such that the robot can attain the desired pose with respect to a given object. As such, several variants of IBVS methods are available based on the image features used [15], [16], [17] and tracking methods [18], [19], [20], [21]. The tracking and visual servoing for moving objects [22], [23] has been studied earlier. A few approaches [24],[25], [26] report visual servoing methodology for capturing a tumbling target, however they also rely on CAD models or markers to estimate the pose of the object. So far, a CAD model free image based visual servoing towards a tumbling uncooperative object has

[1] International Institute of Information Technology Hyderabad,India
[2] Indian Institute of Technology Jodhpur,India
[*] Contributed equally.
{mithun.p,harit.pandya}@research.iiit.ac.in,ayush.gaud@gmail.com, surilshah@iitj.ac.in, mkrishna@iiit.ac.in.



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 2. **Ellipse fitting in image space and error comparison between conventional IBVS and proposed method:** (a-d) Observing a tumbling feature point for few frames. (e) An ellipse is fitted to the observed feature path. (f) Desired feature point to be reached by visual servoing. (g) Oscillating feature error observed while performing conventional IBVS. (h) Feature error converges smoothly by using the proposed visual servoing controller.

not been reported.

In this regard, we propose a novel estimation free IBVS method for reaching the desired pose near a tumbling object as shown in Fig. 1. Unlike common approaches our method does not require estimation of inertia, centroid, angular velocities or orientation of the unknown tumbling object. For a rigid object, rotating around an axis, it is evident that the features follow a circular path in 3D space. Now the features extracted from the projected image of rotating object will follow an elliptical motion in the image plane of the camera. In our algorithm, we take advantage of this elliptical motion of the moving feature points. With this knowledge, a proposed error function between elliptically moving points and stationary desired points is minimized under IBVS framework.

In [27] and [28], ellipse is used as feature to perform visual servoing. However, these approaches consider stationary circles or spheres in the world whose projections are ellipses in the image plane. On the other hand, in case of our approach, features from the tumbling object are used to obtain an ellipse. Moreover, the control law in the aforementioned approaches deals with static objects and is not applicable for tumbling objects. Thus, our method is the first to deal with visual servoing towards a tumbling object without explicit estimation of it's structural or kinematic attributes. Our approach is generic enough to be extended for any robot servoing to a tumbling object and space robot is one of the potential areas of application.

Contributions: The main contributions of the proposed work are as follows: (a) We present a novel visual servoing strategy that is able to attain a desired pose with respect to a rotating target without explicit reconstruction or estimation of target's motion. This is achieved via converting dynamic features into spatial domain by modeling them as ellipse. (b) We further propose an error function that minimizes the error between the elliptical track of the dynamic features and static desired point. (c) We finally present the analytical formulation for interaction matrix, relating camera velocity to change in the elliptic features. The code for this approach is available at: `http://robotics.iiit.ac.in/urls/H8Tm.htm`

II. PRELIMINARIES

This section presents IBVS preliminaries that surface behind the framework presented in the paper. The basic idea behind vision-based control is introduced here and it forms the basis of the framework proposed in the paper.

A. Classical IBVS

In classical image based visual servoing methods, visual information is represented by a set of features $(\mathbf{s})$ extracted from the image measurements [7]. The objective of visual servoing is to command the robot to reach the desired position. This is achieved by defining a task function that minimizes the error between the current $\mathbf{s}$ and the desired configuration of features $\mathbf{s}^*$ [15], i.e.

$$\mathbf{e} = (\mathbf{s} - \mathbf{s}^*) \tag{1}$$

Here $\mathbf{s} \in R^k$ is the feature vector of k feature points. Once the features and the error are defined, the conventional approach is to design a velocity controller by using the relationship between the time variation of features and the camera velocity given by

$$\dot{\mathbf{s}} = \mathbf{L}\mathbf{t}_c \tag{2}$$

Where $\mathbf{t}_c \in R^6$ is camera velocity consisting of linear and angular velocities $(\mathbf{v}, \mathbf{w})$, and $\mathbf{L} \in R^{k\times 6}$ is the interaction matrix. If the desired feature $\mathbf{s}^*$ in (1) is not changing with time, the rate of change of error is nothing but rate of change of current features expressed as $\dot{\mathbf{e}} = \dot{\mathbf{s}}$. Thus, the main

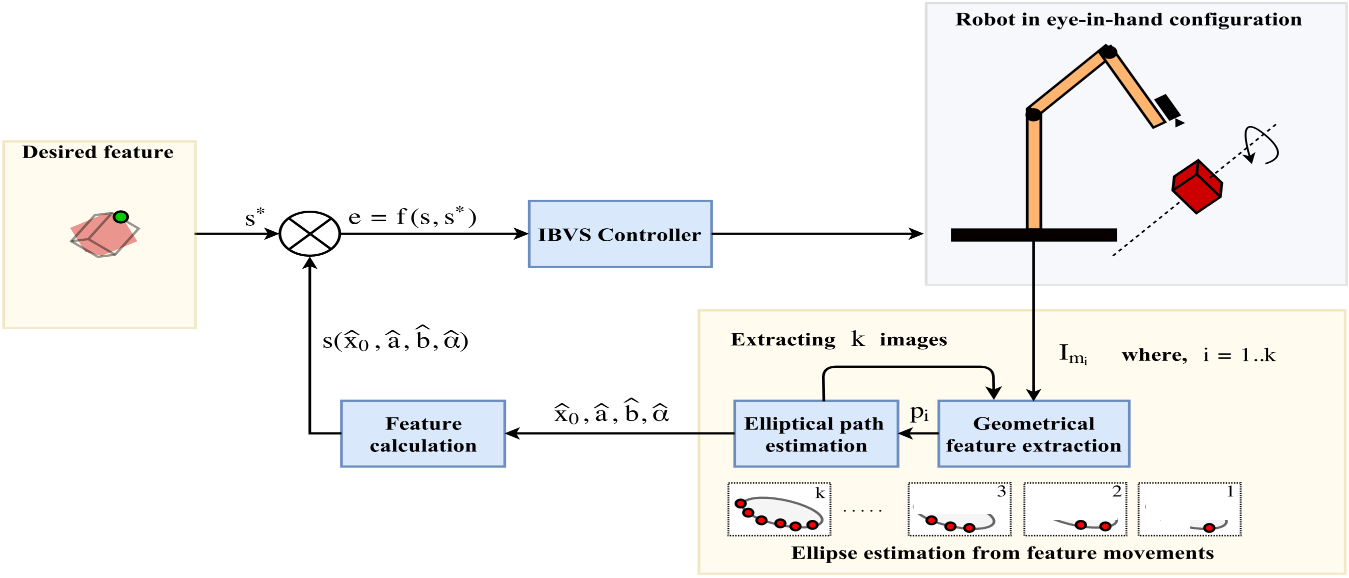

Robot in eye-in-hand configuration
Desired feature
s^*
$e = f(s, s^*)$
IBVS Controller
$s(\widehat{x}_0, \widehat{a}, \widehat{b}, \widehat{\alpha})$
Extracting k images
I_{m_i} where, $i = 1..k$
$\widehat{x}_0, \widehat{a}, \widehat{b}, \widehat{\alpha}$
p_i
Feature calculation
Elliptical path estimation
Geometrical feature extraction
k
.....
3
2
1
Ellipse estimation from feature movements

Fig. 3. **Block diagram for visual servoing towards a tumbling object:** After initially observing the extracted features of an object for a few frames, the motion of the object's features is represented by ellipses in the image plane. Considering these ellipses and desired stationary features, an error function is calculated and given to Image Based Visual Servoing (IBVS) controller.

objective of visual servoing is to nullify the error function which is defined in (1). For exponential decay of error, $\dot{\mathbf{e}}$ is selected as

$$\dot{\mathbf{e}} = -\lambda \mathbf{e}, \tag{3}$$

Where λ is a scalar gain which determines the speed of convergence. Upon substituting (3) in (2), the visual servoing controller can be expressed as

$$\mathbf{t}_c = -\lambda \mathbf{L}^+ \mathbf{e}. \tag{4}$$

Where $\mathbf{L}^+$ is the pseudo-inverse of the image Jacobian, and λ is a scalar gain which determines the rate of convergence of the visual servoing.

III. Outline of Proposed Tumbling Object Visual Servoing Algorithm

The overall idea of the proposed visual servoing control law for a tumbling object is to minimize an error between the ellipse created by feature movements and the desired stationary feature point in the image space. The conventional feature-based visual servoing methods will fail as the feature error $\mathbf{s}(t) - \mathbf{s}^*$ continuously changes due to the object's rotation resulting in an oscillatory behavior as shown in Fig. 2(g). Another challenge faced by the conventional control law is that for a tumbling object features might disappear in the image plane at different intervals of time. To overcome these issues, moving features are estimated into an ellipse and its parameters act as features which encapsulates the tumbling behavior of the object. Later a control law is formulated to reduce the error between the proposed ellipse features and fixed desired feature point.

As shown in Fig. 2, each iteration of the algorithm starts by observing the tumbling object for a while. After observing feature motion for a few frames, a feature path can be easily estimated using an ellipse traced in the image space. This ellipse includes the estimated path of the future feature positions. Since the robot motion results in the change in feature ellipse, the aforementioned estimation step is necessary after every iteration. The accuracy of this feature ellipse estimation depends upon the number of locations of same features considered in consecutive frames. The elliptical path estimated is then defined in terms of ellipse parameters given by ellipse center, major axis length, minor axis length and orientation of ellipse. Once the ellipse is obtained for all the features, the next step is to formalize a visual servoing control law.

Since visual servoing is a feedback control method, an error function is required to model the control law. In the proposed method, points extracted from the desired object are considered as goal features as shown in Fig. 3. It may be noted that the desired features are stationary while the object under consideration is tumbling. This makes the formulation significantly different from the existing visual servoing methods. Furthermore, the error function is modeled by considering the fact that, at the desired pose the instantaneous features will belong to circumference of the ellipse. Considering this relation, the control law iteratively minimizes the error between the ellipse and the desired location of the feature point corresponding to it (which remains stationary). A comparison between the error plots from conventional visual servoing [15] and proposed visual servoing is shown in Fig. 2(g-h). It can be observed that in conventional IBVS, servoing towards a tumbling object results in oscillations in the error with multiple local minima. While the proposed error function follows a smooth continuously decreasing profile which results in a stable camera motion.

Algorithm 1

Input: Starting camera configuration ($\mathbf{q}_1$) and desired feature ($\mathbf{s}^*$) of object

Output: Camera velocity ($\mathbf{t}_c$) in Cartesian space to minimize error $\mathbf{e} = f(\mathbf{s}, \mathbf{s}^*)$

```
procedure TUMBLING OBJECT VISUAL SERVOING
    Initialize_Camera($\mathbf{q}_1$)
    $\mathbf{s}^* \leftarrow$ Desired_Points
    while || e || > $r$ do
        for (Number_Of_Frames < $k$) do
            $\mathbf{p}_i \leftarrow$ Feature_Extraction(Image)
        end for
        $\mathbf{s} \leftarrow$ Generate_Ellipse_Feature($\mathbf{p}$)
        $\mathbf{L}_s \leftarrow$ Get_Interaction_Matrix($\mathbf{s}$)
        $\mathbf{e} \leftarrow$ Calculate_Error($\mathbf{s}$,$\mathbf{s}^*$)
        $\mathbf{t}_c \leftarrow$ Control_Law($\mathbf{L}_s$,$\mathbf{e}$)
        Move_Camera($\mathbf{t}_c$)
    end while
end procedure
```

The interaction matrix relating this change in error to the camera velocity is obtained from the parametric formulation of the ellipse and is explained in the next section IV. This formulation of interaction matrix for visual servoing framework is first of its kind and it ensures the system to reach the desired pose by minimizing the error. An algorithmic implementation of the proposed methodology is shown in Algorithm 1.

With the objective to develop a visual servoing control law for tumbling object, following assumptions are considered. (a) The object is under pure rotation and tumbling with respect to a single axis only. (b) The rate at which the object tumbles is moderate. The object should not move at an integer multiple of sampling rate of the camera and it should not be static or else the robot will wait indefinitely for the motion in features to fit an ellipse. The following sections give a detailed formulation of the proposed algorithm.

IV. Feature Modeling and Interaction Matrix

The initial part of this section focuses on feature modeling approach for visual servoing towards a tumbling object. Later, a formulation of the interaction matrix using these features is presented.

A. *Ellipse as a dynamic visual feature*

In the case of single axis rotation of a rigid object, the trajectory incurred by any point $\mathbf{X} = (X, Y, Z)$ is circular around the axis of rotation with center as the object's center of gravity. While the projection of this point in the image space, say $\mathbf{x}$, will follow an elliptical trajectory. The motion of this point $\mathbf{X}(t)$ can be modeled as an ellipse in the image plane, shown in Fig. 3. Since, the coordinates of $\mathbf{X}(t)$ in the image plane $\mathbf{x}(t)$ vary with respect to time under constant velocity, the time parameterized equation of the ellipse is given by

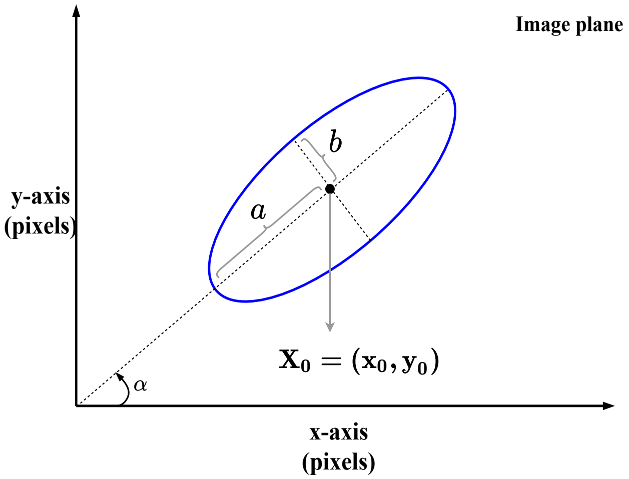


Fig. 4. **Ellipse as a visual feature:** The parameters of the ellipse obtained from the tumbling feature points are centroid (x_0, y_0), length of major axis $2a$, length of minor axis $2b$ and orientation of ellipse α with respect to the x-axis.

$$\mathbf{x}(\mathbf{x}_0, a, b, \alpha, t) = \mathbf{x}_0 + \mathbf{R}[acos(t), bsin(t)]^T. \tag{5}$$

Where, $\mathbf{x}_0, a, b, \alpha$ and t are ellipse center, major axis length, minor axis length, the orientation of ellipse and time respectively, as shown in Fig. 4. The 2D rotation matrix $\mathbf{R}$ having angle α with respect to x-axis of image plane is given by

$$\mathbf{R} = \begin{bmatrix} cos(\alpha) & -sin(\alpha) \\ sin(\alpha) & cos(\alpha) \end{bmatrix}. \tag{6}$$

Which specifies the orientation of the ellipse in the image plane. Equation (5) is further used to compute the coordinates of the point $\mathbf{x}(t)$ at any time instance t, .

Thus, the elliptical path obtained by the motion of feature points over-time could be predicted simply by estimating the parameters of the ellipse $\mathbf{x}_0, a, b, \alpha$. This is achieved by initially observing the camera coordinates for a few frames and solving the following least-squares optimization problem between the observed values of data points $\mathbf{x}$ and the value estimated using (5):

$$\widehat{\mathbf{x}_0}, \widehat{a}, \widehat{b}, \widehat{\alpha} = argmin_{\mathbf{x}_0, a, b, \alpha} \|\mathbf{x} - \widehat{\mathbf{x}}\| \tag{7}$$

In (7), solving for the five parameters requires at least five observations. However, considering noise in the feature extraction and matching, it is desirable to include more observations. Note that more observation may delay servoing as the step of estimating ellipse needs to be performed at every servoing iteration. Therefore, the number of observations is a tuning parameter which is a trade-off between speed and accuracy of the system. After the parameters of the ellipse are computed, the time parametrized equation (5) can be rewritten into the spatial domain as:

$$\mathbf{R}\begin{bmatrix}\frac{(x-x_0)^2}{a^2}\\ \frac{(y-y_0)^2}{b^2}\end{bmatrix}=1. \tag{8}$$

After simplification, the above equation is reduced to the proposed visual feature representation is given by:

$$\mathbf{s}=\begin{bmatrix}s_x\\ s_y\end{bmatrix}=\begin{bmatrix}cos(\alpha)\frac{(x_0-x)}{a}+sin(\alpha)\frac{(y_0-y)}{b}\\ -sin(\alpha)\frac{(x_0-x)}{a}+cos(\alpha)\frac{(y_0-y)}{b}\end{bmatrix}. \tag{9}$$

B. Interaction matrix formulation

Once the features are obtained, next task is to compute the interaction matrix, which is a mapping between the change of visual features to camera velocity so that a velocity controller can be designed [15]. In general, an interaction matrix for visual features $\mathbf{s}$ with respect to the 6-DoF camera pose $\mathbf{r}$ is given by:

$$\mathbf{L_s}=\begin{bmatrix}\mathbf{L}_{s_x}\\ \mathbf{L}_{s_y}\end{bmatrix}=\frac{\partial\mathbf{s}}{\partial\mathbf{r}}. \tag{10}$$

By taking partial derivatives of (9), $\mathbf{L}_{s_x}$ and $\mathbf{L}_{s_y}$ are obtained as follows

$$\begin{aligned}\mathbf{L}_{s_x}&=s_y\mathbf{L}_\alpha-\frac{cos(\alpha)(x_0-x)}{2a^2}\mathbf{L}_a-\frac{sin(\alpha)(y_0-y)}{2b^2}\mathbf{L}_b\\ &\quad+\frac{cos(\alpha)}{a}\mathbf{L}_{x_0}+\frac{sin(\alpha)}{b}\mathbf{L}_{y_0}\\ \mathbf{L}_{s_y}&=-s_x\mathbf{L}_\alpha+\frac{sin(\alpha)(x_0-x)}{2a^2}\mathbf{L}_a-\frac{cos(\alpha)(y_0-y)}{2b^2}\mathbf{L}_b\\ &\quad-\frac{sin(\alpha)}{a}\mathbf{L}_{x_0}+\frac{cos(\alpha)}{b}\mathbf{L}_{y_0}\end{aligned} \tag{11}$$

Where, $\mathbf{L}_\alpha,\mathbf{L}_a,\mathbf{L}_b,\mathbf{L}_{x_0},\mathbf{L}_{y_0}$ are the interaction matrices relating the parameters of the ellipse with the camera pose. One of the challenging task is to obtain the values of these interaction matrices. For calculating the same following method is adopted.

One possible way to estimate these interaction matrices is by considering three points, one at the center of the ellipse $\mathbf{x}_0$, second one at one end of major axis $\mathbf{x}_1$, and the last one at the end of minor axis $\mathbf{x}_2$. We already have the knowledge of $\mathbf{x}_0$ by fitting ellipse on the observed values of $\mathbf{x}$ using (7). The other two points could also be simply computed using ellipse parameters a, b and α.

$$\begin{aligned}\alpha&=atan2(y_1-y_0,x_1-x_0)\\ a&=\sqrt{(x_1-x_0)^2+(y_1-y_0)^2}\\ b&=\sqrt{(x_2-x_0)^2+(y_2-y_0)^2}\end{aligned} \tag{12}$$

Once these three points are calculated, an analytical formulation of the interaction matrices pertaining to the ellipse parameters can be obtained by following equations.

$$\begin{aligned}\mathbf{L}_\alpha&=\frac{-(y1-y0)}{(x_1-x_0)^2+(y_1-y_0)^2}(\mathbf{L}_{x_1}-\mathbf{L}_{x_0})\\ &\quad+\frac{-(x1-x0)}{(x_1-x_0)^2+(y_1-y_0)^2}(\mathbf{L}_{y_1}-\mathbf{L}_{y_0})\\ \mathbf{L}_a&=\frac{(x1-x0)}{a}(\mathbf{L}_{x_1}-\mathbf{L}_{x_0})+\frac{(y1-y0)}{a}(\mathbf{L}_{y_1}-\mathbf{L}_{y_0})\\ \mathbf{L}_b&=\frac{(x2-x0)}{b}(\mathbf{L}_{x_2}-\mathbf{L}_{x_0})+\frac{(y2-y0)}{b}(\mathbf{L}_{y_2}-\mathbf{L}_{y_0})\end{aligned} \tag{13}$$

The interaction matrices for a point located at (x, y) are the same as the conventional formulation [15] and are given by:

$$\begin{aligned}\mathbf{L}_{x_i}&=[-1/Z\ \ 0\ \ x/Z\ \ xy\ \ -(1+x^2)\ \ y]\\ \mathbf{L}_{y_i}&=[0\ \ -1/Z\ \ y/Z\ \ 1+y^2\ \ -xy\ \ -x]\end{aligned} \tag{14}$$

where Z is the depth of point (x, y). The next step is to formulate a control law for the ellipse based features s.

V. IBVS FOR TUMBLING OBJECT

Commonly image based visual servoing approaches consider the minimization of feature error as the task function $\mathbf{e}=\mathbf{s}-\mathbf{s}^*$. The central idea behind this approach is that when the feature error regulates to zero, the desired pose is achieved. Such error law is not applicable when the object is itself tumbling. Two cases of control law failure can be observed there.

- Case 1: Feature points in image plane will be visible at one point of time and will not be visible in another.
- Case 2: If the feature movement is perpendicular to camera $z-$axis, the robot end-effector will try to move in an elliptical path to follow the rotating features.

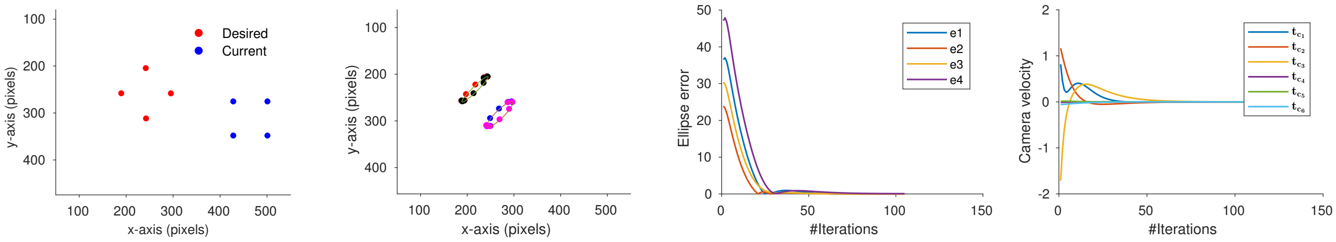
The invalid case may result from combination of the above. To overcome such failure situations, few approaches consider the motion compensation in the control law [29], for dynamic object. However, in such cases, an estimate of object's velocity is required.

Instead, in the proposed method of visual servoing towards tumbling object, a dynamic visual feature or feature ellipse is considered, which has to be matched with a desired static feature point. Thus, the error $\mathbf{e}=f(\mathbf{s},\mathbf{s}^*)$ to be minimized is in between an ellipse and a point. With the features mentioned in (9), the error could be written as:

$$f(\mathbf{s},\mathbf{s}^*)=\begin{bmatrix}e_x\\ e_y\end{bmatrix}=\begin{bmatrix}cos(\alpha)\frac{(x_0-x^*)}{a}+sin(\alpha)\frac{(y_0-y^*)}{b}\\ -sin(\alpha)\frac{(x_0-x^*)}{a}+cos(\alpha)\frac{(y_0-y^*)}{b}\end{bmatrix}. \tag{15}$$

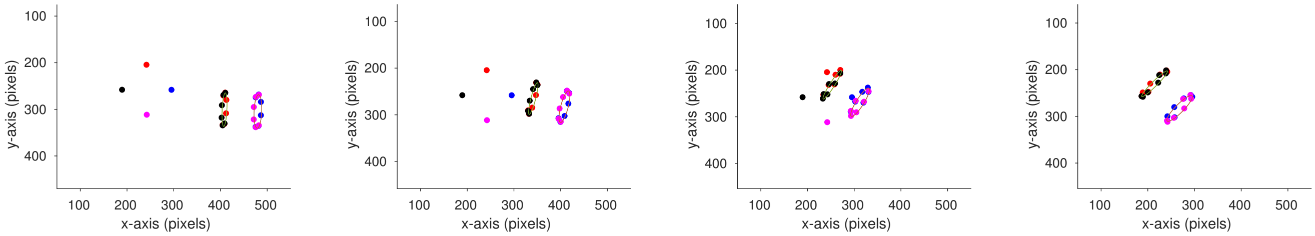
Where, $\mathbf{s}^*=(x^*,y^*)$ is the desired static point. Note that the error will be zero when the desired point lies on the boundary of the ellipse. Thus, there are infinite solutions to this problem. Since the object is moving there are also infinite ways of matching the desired pose. Once the feature error is zero one can wait till the object comes to a pose where indeed $\mathbf{s}-\mathbf{s}^*=0$. It may be noted that this case is not considered in the proposed framework.

With the proposed novel interaction matrix in (10) and error function in (15), the required camera velocity can be



(a) Initial point features (b) Obtained final pose (c) Error (d) Velocity

Fig. 5. Visual servoing towards a tumbling object: Ellipse features are modeled from corner points of a rotating square.



(a) t=0 (b) t=6 (c) t=16 (d) t=50

Fig. 6. Progression of ellipses changes at various time stamps during proposed visual servoing.

calculated using (4) and is given by

$$\mathbf{t}_c = -\lambda \mathbf{L}_s f(\mathbf{s}, \mathbf{s}^*) \tag{16}$$

It will be shown in the following section that the above controller results in (a) exponential decrease of error function and (b) convergence of camera and joint velocities are regulated to zero at the desired pose.

VI. Results

This section discusses the simulation as well as real results of successful visual servoing towards a tumbling object using the proposed framework. In order to validate the proposed control law an eye-in-hand system is considered, and reaching the desired pose is considered as the convergence of the experiment. The object under consideration is a square exhibiting a tumbling motion with its four corners used as features. In order to evaluate the robustness of this approach, a number of experiments for different camera positions are performed. Initial part of this section discusses results for a general case, where the results shows progression of ellipses and successful operation of the proposed algorithm. Later, results of visual servoing for different axes rotations of an object are compared to show the efficacy of the framework. It may be noted that at the converged state the feature will be rotating in the image plane with camera at the desired pose.

A. 3D positioning task

The goal of this experiment is to exhibit 6-DOF camera motions for reaching the desired pose. The object considered here is a square object placed at a known world coordinate. In order to verify the proposed approach in simulation, we use perspective projections of objects corners as the features, and any state-of-art feature extraction methods can be used for real-world operations. The object position in world frame is considered at $(x, y, z) = (0, 0, 0)$ and orientation at $(w_x, w_y, w_z) = (0^0, 0^0, 0^0)$. Fig. 4 shows plots of successful visual servoing operations performed by the proposed method. The starting and desired camera pose in word coordinate is at $(20, 10, 10, 0^0, 0^0, 0^0)$ and $(-20, 5, 6, 5^0, 3^0, 45^0)$ respectively and corresponding feature points in the image plane is shown in Fig. 5(a). The convergence of the task is considered as bringing the ellipses close to the desired points (shown in Fig. 5(b)) such that the points lie on the circumference of their respective ellipses. Fig. 5(c-d) shows the corresponding feature error and velocity plots for the servoing operations performed. From the plots, it can be observed that the velocity curves follow exponential decay and the error plots shows the successful convergence of cost function. The feature movement for this experimentation at different stages are shown in Fig. 6(a-d). From these figures, it is clear that the feature movement from starting to the desired pose is as expected.

B. Robustness to axis of rotation

The goal of this experiment is to show that the proposed approach is robust to the change in axis of rotation of the object. The different axes of rotations of the object were considered from a same initial camera position as shown in Fig. 7(a,e,i). Here three axes of rotations z, y and x are considered and corresponding experimental results are shown in each row of Fig. 7. It may be seen, the control law converges in the case of different axis of rotation for the object under study. This shows efficacy of the proposed framework. A few more simulation were also carried with

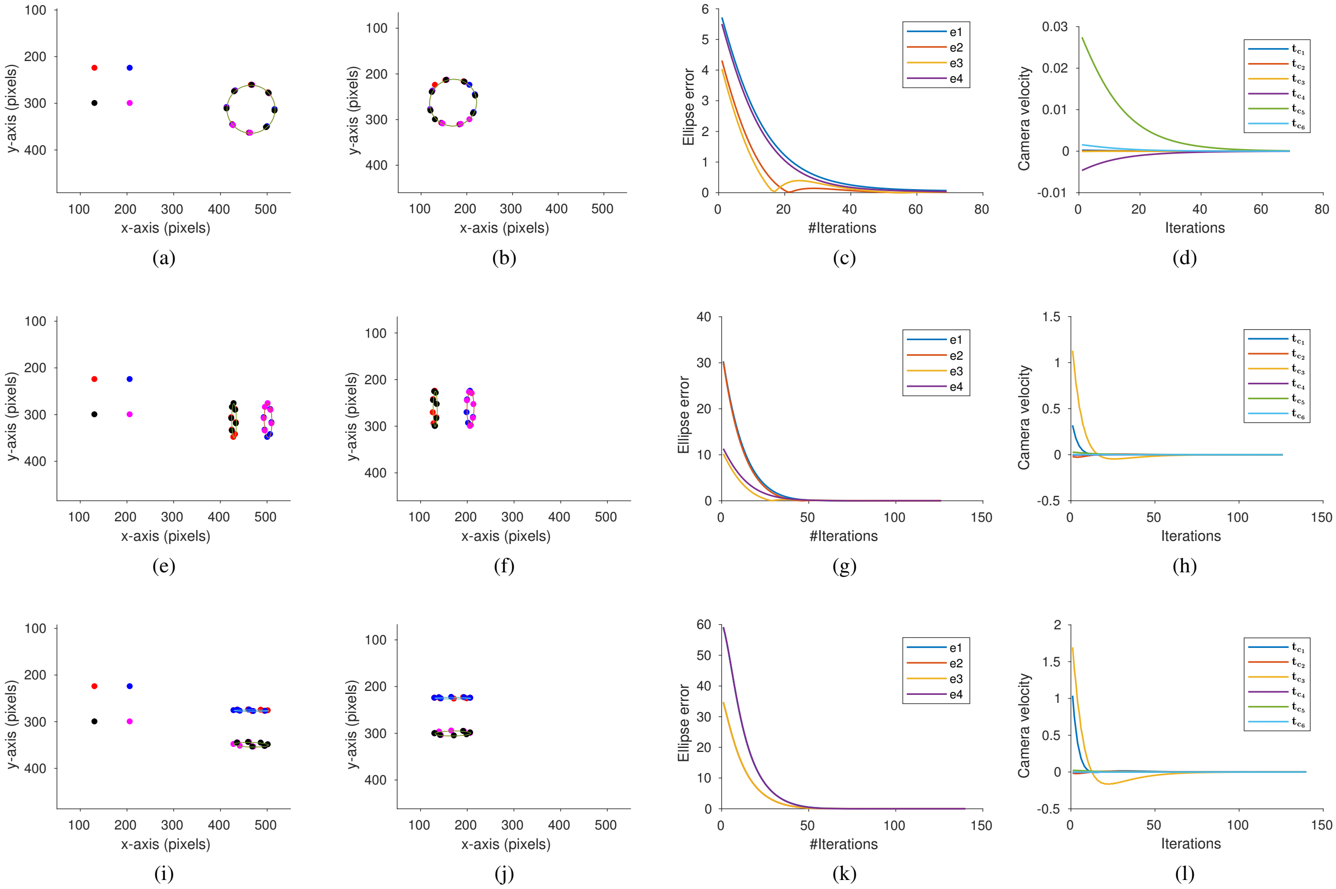


Fig. 7. Ellipse fitting and visual servoing for z, y and x axis object rotations. The Error and velocity plots validates the convergence of control law in each case.

different initial and final poses which are not reported for brevity.

C. Experiment on real robot

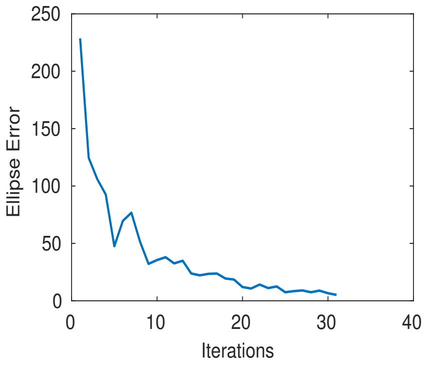
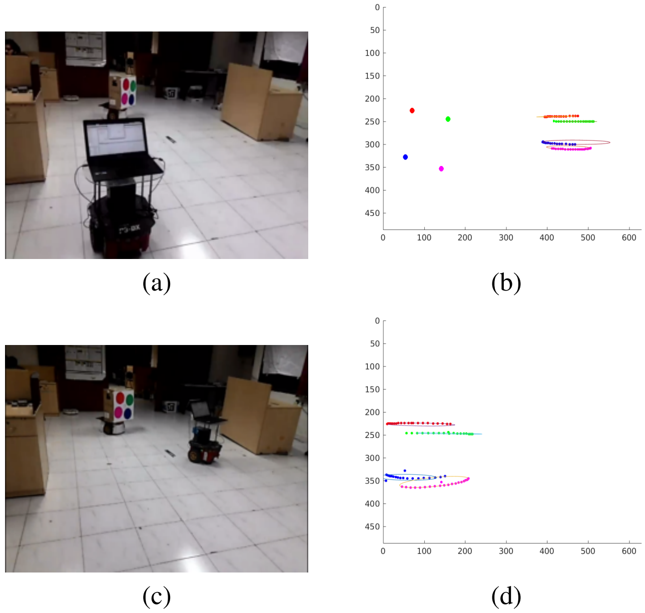


Fig. 8. **Error plot of real experiment:** Mean of ellipse error converges to zero while performing real experiments using mobile robots. This validated our proposed methodology

We have considered two Pioneer P3-DX for the real experimentation. The first robot is considered as a tumbling object with markers attached to it. The second robot has to servo towards a desired pose of the object. Upon executing proposed algorithm in this scenario, the second robot is able to reach the desired pose successfully. Fig. 9(a,c) shows the starting and final position of the robot. The corresponding image plane features are shown in Fig. 9(b,d). It can be observed that the dynamic ellipse features have converged to the desired points by moving towards the left in the image plane. This shows the efficacy of our framework in real world also. From the plot in Fig. 8 it can be observed that the feature mean error converges to zero in an exponentially decreasing manner. This validates the convergence of the proposed methodology.

VII. Conclusion

We presented a novel visual servoing controller that uses dynamic image features to collectively represent the motion of a tumbling object. The servoing is achieved by minimizing a distance function between the current elliptically moving points and the desired stationary points. The proposed framework has removed the need for a direct estimation of object's motion from the servoing process. We also introduced the analytical formulation of the ellipse features and interaction matrix, which is first of its kind, and using the same in visual servoing control law resulted in successful visual servoing towards a tumbling object. We are presently working on the stability analysis of proposed methodology and a method to



(a) (b)

(c) (d)

Fig. 9. **Real robot experiments** The proposed algorithm is tested on a mobile robot where the task is to reach a desired pose in image plane. Here the tumbling object motion is exhibited by the first robot having markers on it. (a) Second robot at initial position observes the rotating feature on the first robot. (b) The rotating features are modeled as ellipses, and these ellipses has to move towards left to reach the desired four points. (c) Robot has reached the desired pose. (d) The ellipse features converged to desired stationary points.

grasp the rotating object once it reaches the desired pose. An extension of proposed approach for multi-axis object rotations will be taken up in the future.

References

[1] H. Klinkrad, "Orbital debris and sustainability of space operations," in *Handbook of Satellite Applications*. Springer, 2013, pp. 1145–1174.

[2] S. Kawamoto, S. Nishida, and S. Kibe, "Research on a space debris removal system," *NAL Res Prog (National Aerospace Lab. Japan)*, vol. 2002, pp. 84–87, 2003.

[3] D. Reintsema, J. Thaeter, A. Rathke, W. Naumann, P. Rank, and J. Sommer, "Deos–the german robotics approach to secure and de-orbit malfunctioned satellites from low earth orbits," in *Proceedings of the i-SAIRAS*, 2010, pp. 244–251.

[4] F. Sellmaier, T. Boge, J. Spurmann, S. Gully, T. Rupp, and F. Huber, "On-orbit servicing missions: Challenges and solutions for spacecraft operations," *Space Operations: Exploration, Scientific Utilization, and Technology Development*, p. 213, 2010.

[5] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *ieee Transactions on Robotics and Automation*, vol. 8, no. 3, pp. 313–326, 1992.

[6] P. I. Corke, "Visual control of robot manipulators-a review," *Visual servoing*, vol. 7, pp. 1–31, 1993.

[7] 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.

[8] G.-S. Young and R. Chellappa, "3-d motion estimation using a sequence of noisy stereo images: Models, estimation, and uniqueness results," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 8, pp. 735–759, 1990.

[9] T. J. Broida and R. Chellappa, "Estimating the kinematics and structure of a rigid object from a sequence of monocular images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 6, pp. 497–513, 1991.

[10] H. Nagamatsu, T. Kubota, and I. Nakatani, "Capture strategy for retrieval of a tumbling satellite by a space robotic manipulator," in *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, vol. 1. IEEE, 1996, pp. 70–75.

[11] A. Petit, E. Marchand, and K. Kanani, "Vision-based space autonomous rendezvous: A case study," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 619–624.

[12] F. Aghili, M. Kuryllo, G. Okouneva, and C. English, "Robust vision-based pose estimation of moving objects for automated rendezvous & docking," in *Mechatronics and Automation (ICMA), 2010 International Conference on*. IEEE, 2010, pp. 305–311.

[13] S. Ruel, T. Luu, and A. Berube, "Space shuttle testing of the tridar 3d rendezvous and docking sensor," *Journal of Field robotics*, vol. 29, no. 4, pp. 535–553, 2012.

[14] T. P. Setterfield, D. Miller, J. J. Leonard, and A. Saenz-Otero, "Smoothing-based estimation of an inspector satellite trajectory relative to a passive object," in *Aerospace Conference, 2017 IEEE*. IEEE, 2017, pp. 1–11.

[15] F. Chaumette and S. Hutchinson, "Visual servo control. i. basic approaches," in *MRA*, 2006.

[16] F. Chaumette, "Image moments: a general and useful set of features for visual servoing," *IEEE Transactions on Robotics*, vol. 20, no. 4, pp. 713–723, 2004.

[17] J. Xin, B.-J. Ran, and X.-M. Ma, "Robot visual sliding mode servoing using sift features," in *Control Conference (CCC), 2016 35th Chinese*. IEEE, 2016, pp. 4723–4729.

[18] E. Marchand, P. Bouthemy, F. Chaumette, and V. Moreau, "Robust real-time visual tracking using a 2d-3d model-based approach," in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 1. IEEE, 1999, pp. 262–268.

[19] V. Servoing, "Visual servoing and visual tracking," *Urbana*, vol. 51, p. 61801.

[20] A. A. Hafez, V. Chari, and C. Jawahar, "Combining texture and edge planar trackers based on a local quality metric," in *Robotics and Automation, 2007 IEEE International Conference on*. IEEE, 2007, pp. 4620–4625.

[21] R. Richa, R. Sznitman, R. Taylor, and G. Hager, "Visual tracking using the sum of conditional variance," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference On*. IEEE, 2011, pp. 2953–2958.

[22] F. Chaumette and A. Santos, "Tracking a moving object by visual servoing," *IFAC Proceedings Volumes*, vol. 26, no. 2, pp. 643–648, 1993.

[23] J. Pomares and F. Torres, "Movement flow-based visual servoing to track moving objects," in *Automation Congress, 2004. Proceedings. World*, vol. 15. IEEE, 2004, pp. 241–246.

[24] A. Petit, "Robust visual detection and tracking of complex objects: applications to space autonomous rendez-vous and proximity operations," Ph.D. dissertation, Université Rennes 1, 2013.

[25] I. Rekleitis, E. Martin, G. Rouleau, R. L'Archevêque, K. Parsa, and E. Dupuis, "Autonomous capture of a tumbling satellite," *Journal of Field Robotics*, vol. 24, no. 4, pp. 275–296, 2007.

[26] B. P. Larouche and Z. H. Zhu, "Autonomous robotic capture of non-cooperative target using visual servoing and motion predictive control," *Autonomous Robots*, vol. 37, no. 2, pp. 157–167, 2014.

[27] P. Rives and H. Michel, "Visual servoing based on ellipse features," in *Intelligent Robots and Computer Vision XII: Active Vision and 3D Methods*, vol. 2056. International Society for Optics and Photonics, 1993, pp. 356–368.

[28] T. Shen and G. Chesi, "Visual servoing path-planning with elliptical projections," in *Informatics in Control, Automation and Robotics*. Springer, 2018, pp. 30–54.

[29] F. Chaumette and S. Hutchinson, "Visual servo control. ii. advanced approaches [tutorial]," *IEEE Robotics & Automation Magazine*, vol. 14, no. 1, pp. 109–118, 2007.