

# Image Jacobian Estimation Using Structure From Motion on a Centralized Point

Victor Nevarez, and Ron Lumia, *Fellow IEEE*

**Abstract**—Image based visual servoing (IBVS) has become increasingly common in automated systems. IBVS control systems are dependent on the accuracy of an image Jacobian, which requires the knowledge of the internal and external parameters of the system. In most systems, some or all of these parameters are unknown. A centralized motion algorithm (CMA) is proposed to compute the image Jacobian with fast convergence and low iterations. Centralized motion is defined as a motion where a single feature point does not change its pixel coordinate value through the camera's motion. Therefore, the CMA is an algorithm that exploits the fact that the chosen feature point will have little to no changes in pixel coordinates. As a result, the CMA requires only a single stationary feature point, along with a minimum of two more feature points that will be dynamic, in the image to compute the image Jacobian. Using a Whole-Arm Manipulation (WAM) IBVS system, it was shown that an image Jacobian with an average condition number of 45, converges in an average of 4 iterations from 6 different starting positions. Experiments show that the CMA is a fast and effective online calibration algorithm.

## I. INTRODUCTION

Visual servoing has become increasingly important because of the demand for automated systems with machine vision. Image based visual servoing (IBVS) [1] is a powerful method of control since the control law is based on the features in the image plane which makes it more robust to disturbances and noise. Since some or all of the intrinsic and extrinsic camera parameters may be unknown, there are many different methods for on-line estimation of the image Jacobian [2], [3], [4], [5], [6], [7], [8].

Many of these methods require a calibration target or some knowledge of the system a priori. The first "knowledge-less" hand-eye calibration technique was proposed by Andreff *et al.* [9], using Structure-from-Motion. Structure-from-Motion (SfM) can be broken down into four steps: 1) image feature detection and description, 2) feature matching and analysis between image pairs, 3) epipolar geometry estimation, and 4)

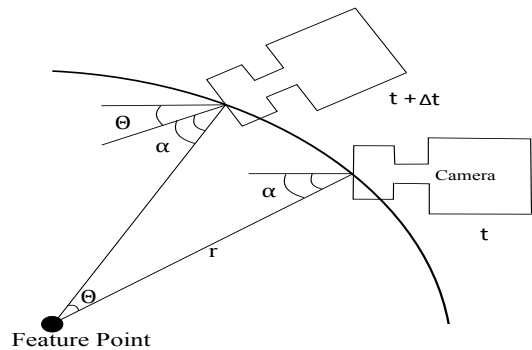


Fig. 1: Circular motion about a centralized point

3D point triangulation and transformation to a common frame. SfM can recover camera poses up to scale with the determination of a scaling factor being important in most works.

Cartesian models for IBVS are not required. Polar [10] and cylindrical [11] methods have been proposed where feature points are set at distance  $r$  and an angle  $\theta$  from camera axis center. From this, it was apparent that fixing the distance  $r$  and angle  $\theta$  and making motions about a feature point were only possible on a circular trajectory. This is one of the main points of the centralized motion algorithm (CMA); a feature point is chosen, and a circular trajectory is made such that the feature point keeps the same pixel location throughout the motion.

Only one centralized point is needed for the CMA to work because the motion about the point is what is being exploited. The CMA uses a SfM approach similar to Heller *et al.* [12] where we first do rational calibration separately. It then performs steps 1 and 2 proposed by Andreff *et al.*, but our feature matching is related to the parameters being determined unlike the analysis between the image pairs.

In this paper, we exploit the spherical relationship between the feature point and the camera through motion about a single feature point, which will be referred

to as the centralized point. Although a spherical motion is not the only motion to keep the feature point at the same pixel coordinate, a spherical trajectory is easier to work with for motion planning. We then use SfM to converge on a usable image Jacobian based on a circular error model.

The CMA is not meant to replace current image Jacobian determination methods or to determine the intrinsic camera properties. The CMA is a "knowledge-less" SfM technique that exploits the change in environment through its unique motion. The CMA is a faster more efficient online calibration algorithm which will be useful in robotics. The intention of this paper is to prove that this algorithm is a useful "knowledge-less" SfM method that could be considered for future work.

## II. CENTRALIZING PERSPECTIVE

The algorithm used for the estimation of an uncalibrated image Jacobian is based on extracting information from images about the aforementioned centralized point. To do this, the feature point must maintain the same pixel coordinates between motions. A motion that satisfies this property is a circular trajectory, or spherical trajectory for 3D motions, about the centralized point. In Figure 1 we see that this will hold true so long as the pose of the camera changes the same amount of  $\theta$  that it has moved along the circular arc.

To begin this process, we need to use a small angle approximation, explained in Section VIIB, to get an initial estimate of the chosen feature point's location as well as an initial estimate for the image Jacobian parameters. Since the circular motion satisfies this criterion, we can exploit any changes in pixel coordinates as the error. Using this, we can derive an error model to compute an image Jacobian iteratively.

## III. IMAGE JACOBIAN

Figure 2 depicts the type of projective geometry related to the camera and the motions the camera will make with respect to the feature point.

From this image, the pixel coordinates  $p = (u, v)$  can be modeled from the global coordinates  $P = (X, Y, Z)$  as

$$u = \frac{fX}{\rho_u Z} + u_0, v = \frac{fY}{\rho_v Z} + v_0 \quad (1)$$

Where  $f$  is the camera's focal length,  $\rho_u$  is the pixel width,  $\rho_v$  is the pixel height, and  $(u_0, v_0)$  is the principal point coordinate; these are the intrinsic parameters.

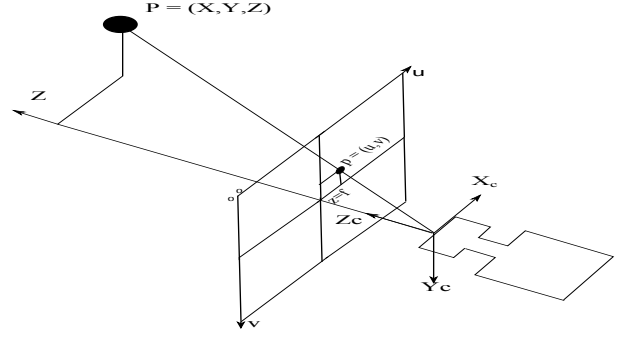


Fig. 2: Projective Geometry

Solving for the global coordinates

$$X = \frac{\rho_u \bar{u} Z}{f}, Y = \frac{\rho_v \bar{v} Z}{f} \quad (2)$$

where  $\bar{u} = u - u_0$  and  $\bar{v} = v - v_0$

Taking the temporal derivative of this projection system, and using the fact that  $\dot{P} = -\omega \times P - V$ , where  $\omega$  is the angular velocity of the camera and  $V$  is the linear velocity of the camera, the image Jacobian relationship becomes

$$\begin{pmatrix} \dot{\bar{u}} \\ \dot{\bar{v}} \end{pmatrix} = J_p \begin{pmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \quad (3)$$

$$J_p = \begin{pmatrix} -\frac{f}{\rho_u Z} & 0 & \frac{\bar{u}}{Z} & \frac{\rho_u \bar{u} \bar{v}}{f} & -\frac{f^2 + \rho_u^2 \bar{u}^2}{\rho_u f} & \bar{v} \\ 0 & -\frac{f}{\rho_v Z} & \frac{\bar{v}}{Z} & \frac{f^2 + \rho_v^2 \bar{v}^2}{\rho_v f} & -\frac{\rho_v \bar{u} \bar{v}}{f} & -\bar{u} \end{pmatrix}$$

To have an image Jacobian of full rank, the image Jacobian must be applied to three feature points.

$$J = (J_{p1}, J_{p2}, J_{p3})^T$$

It is still possible to have an ill-conditioned matrix which could lead to longer computation time and also make the Jacobian's output more sensitive to small inputs. As a result, our system is still susceptible to the same 3-point full rank matrix issues.

## IV. FEATURE LOCATION

To be able to estimate the necessary trajectory, we need to be able to estimate the feature point's location

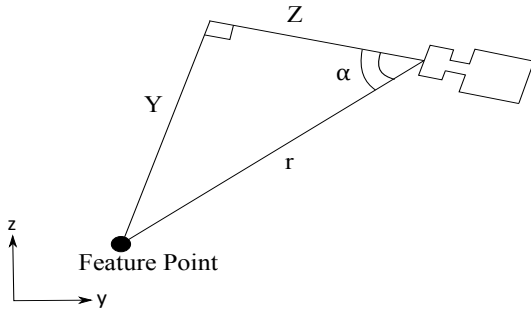


Fig. 3: Feature location with variables

in space,  $P$ . To do this, we will need to associate the distance from the centralized feature point to its depth. Figure 3 shows the depth relationship from the feature points to the camera's Y-axis as  $Z^2 + Y^2 = r^2$ .

Substituting (2) into the previous equation we get

$$Z^2[1 + (\frac{\rho_v \bar{v}}{f})^2] = r^2 \quad (4)$$

Simplifying

$$Z\sqrt{\beta} = r, \beta = 1 + (\frac{\rho_v \bar{v}}{f})^2 \quad (5)$$

For the horizontal case we simply replace  $\rho_v$  with  $\rho_u$  and  $\bar{v}$  with  $\bar{u}$ . The depth is given by

$$Z = \frac{r}{\sqrt{\beta}} \quad (6)$$

From this geometric relationship and using the trigonometric relationship, we can solve for the angle between the depth and the feature point

$$\alpha = \cos^{-1}(\frac{1}{\sqrt{\beta}}) \quad (7)$$

With the depth of the feature point and the estimated intrinsic parameters, we have a model for the feature point's location with respect to the camera. In Section VI, we will further discuss the fact that  $r$  is estimated and that the depth along with  $r$  will need to converge.

## V. PARAMETER CALCULATION

With three feature points, we now have six equations (using the image Jacobian (3) for each feature point) with five unknowns ( $u_0, v_0, f, \rho_u, \rho_v$ ). The only problem is the relationship between them is not linear. Further described in section VII, the principal point is calculated through a simple camera rotation, so we

can treat  $\bar{u}$  and  $\bar{v}$ , defined in section II, as known parameters.

Starting with the image Jacobian for the centralized feature point, we know its depth from (6). Again for simplicity, we will start with the vertical case and relate back to the horizontal case. For the vertical case, the only motions that follow the circular trajectory are in the camera's Y, Z, and  $\theta_x$  direction. Applying the following vector  $(0, \Delta Y, \Delta Z, \Delta \theta_x, 0, 0)^T$  to (3) we can solve for the focal length

$$f = \rho_u \frac{\bar{u}\bar{v}\Delta\theta_x}{\Delta u - \frac{\bar{u}\Delta Z}{Z}} \quad (8)$$

To simplify further, we will treat it as  $f = k\rho_u$  where  $k = \frac{\bar{u}\bar{v}\Delta\theta_x}{\Delta u - \frac{\bar{u}\Delta Z}{Z}}$ . We will use the information from the first feature point to determine the depths of the other two feature points.

$$Z_i = \frac{\bar{u}_i\Delta Z}{\frac{\bar{u}_i v_i \Delta \theta_x}{k} - \Delta u_i} \quad (9)$$

where  $i = 2, 3$

With the two depths known, we can now setup two equations to solve for  $\rho_v$  and  $\rho_u$ . Using the change in the other pixel coordinate, the  $u$  in this case, we find that the equations are two nonlinear equations with the form of:

$$ax_1^2 + bx_1x_2 + cx_2^2 = 0 \quad (10)$$

where  $a = \bar{v}_i^2\Delta\theta_x, b = \frac{\bar{v}_i\Delta Z}{Z} - \Delta v_i, c = \Delta\theta_x - \frac{\Delta Y}{Z_i}, x_1 = \rho_v, x_2 = \rho_u$  for  $i = 2, 3$

With these two equations, we can use a Newton-Raphson iteration method to converge on values for  $\rho_v$  and  $\rho_u$ .

With these two values, we are now able to calculate the focal length and now have all values for our image Jacobian. For the horizontal case, we simply switch out  $\Delta Y, \Delta \theta_x$  with  $\Delta X, \Delta \theta_y$  respectively and switch all of the  $u$  and  $v$  parameters to compute the image Jacobian for the horizontal case.

## VI. ERROR MODEL

From the image shown in Figure 4, we notice how the error varies depending on whether the estimated distance from the camera to the feature point is larger or smaller than the true value. Using the fact that the actual change in pixel value should be zero, we can make an error model for both horizontal and vertical

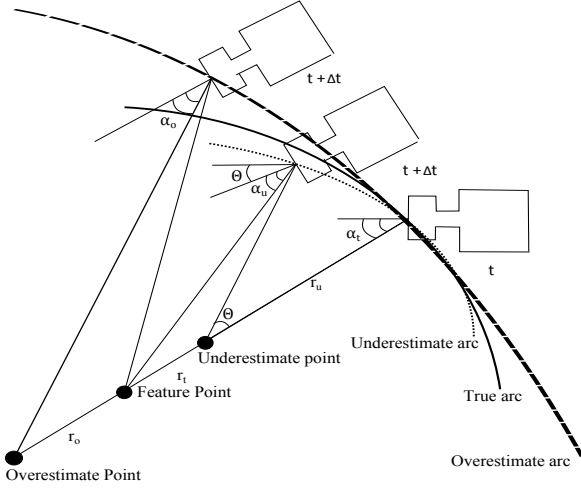


Fig. 4: Trajectory error with overestimate and underestimate

motions. For simplicity, we will use just the vertical case and apply the same approach to the horizontal case.

$$e = v_{act} - v_{est} \quad (11)$$

Using (1) to calculate  $v_{act}$ , the actual location of the centralized point, and using (6) to replace the depth, we can solve for an estimated distance to the feature point.

$$r_{act} = \frac{f_{act} \sqrt{\beta_{act}} Y_{act}}{\rho_{vact} (\bar{v} - e)} \quad (12)$$

Since solving for  $Y_{act}$ , the actual Y coordinate of the feature point, is not possible without knowing the extrinsic parameters, we will instead use an estimation for  $Y_{act}$  by using the estimated depth with (1).

$$Y_{act} = \frac{\rho_{vact} \bar{v} r_{est}}{f_{act} \sqrt{\beta_{act}}} \quad (13)$$

Combining (12) and (13), we can determine the radius for the vertical case. We can also determine the horizontal case by simply switching the pixel coordinates. Below is the vertical and horizontal cases respectively.

$$r_{act} = \frac{\bar{v} r_{est}}{\bar{v} - e}, r_{act} = \frac{\bar{u} r_{est}}{\bar{u} - e} \quad (14)$$

From (14), we can see the closer the error gets to zero the closer the estimated value reaches the true value. In the case of the estimated value being an underestimate, the error ends up being positive as shown in Figure 4, which leads to a larger new distance. Conversely, if

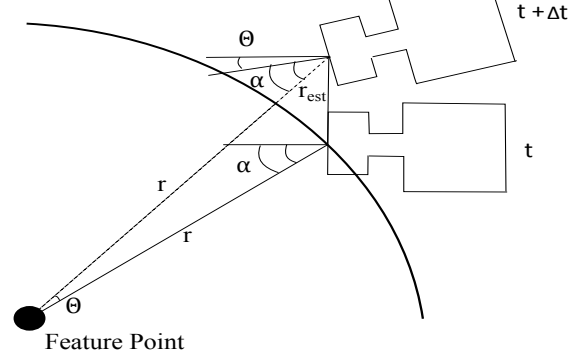


Fig. 5: Small angle approximation

the estimate is an overestimate, the new distance will decrease since the error will be negative. Consequently, the depth from (6) is similarly affected by under and over estimates.

```

1: procedure ROTATION( $\theta_z, u, v$ )
2:    $\Delta\theta_z \Rightarrow \theta_r$  (image rotation)
3:   return  $u_0, v_0$ 
4: end procedure

5: procedure SMALLANGLE( $Y, \theta_x, u_i, v_i, \epsilon_s$ )
Require:  $\Delta Y = Y_{small}, \theta_x \Rightarrow \theta_{max}, i = 1, 2, 3$ 
6:   if  $\Delta v < \epsilon_s \vee \theta_x = \theta_{max}$  then
7:     return  $r_{est} = \frac{\Delta Y}{\Delta \theta_x}$  (small angle approx.)
8:   end if
9:   return  $f, \rho_u, \rho_v$ 
10: end procedure

11: procedure JACEST( $f, \rho_u, \rho_v, x, \theta, \dot{x}, \dot{\theta}, u_i, v_i$ )
Require:  $E = 1, j = 1, i = 1, 2, 3$ 
12:   while  $E > \epsilon_J, \forall |E(j+1) - E(j)| > \epsilon_{conv}$  do
13:     if  $j = odd$  then
14:       Horizontal Motion
15:     else
16:       Vertical Motion
17:     end if
18:     return  $J, j = j + 1$ 
19:      $E = \Delta u_1^2 + \Delta v_1^2$ 
20:   end while
21: end procedure

```

Fig. 6: Centralized Motion Algorithm

## VII. CENTRALIZED MOTION ALGORITHM (CMA)

The pseudo-code for the CMA is shown in Figure 6. All variables are with respect to the camera's frame of reference as illustrated in Figure 2. We initialize assuming we do not know any intrinsic or extrinsic parameters of the system. The algorithm will be broken up into three main sections: Rotation, Small angle approximation, and Jacobian estimation.

### A. Rotation

For the first step, we rotate along the camera's Z-axis,  $\omega_z$  from (3), to isolate the principal point. This can be seen by isolating only the last column in (3). We solve for  $u_0$  and  $v_0$  by replacing  $\bar{u} = u - u_0$  and  $\bar{v} = v - v_0$  into the last column in (3).

### B. Small Angle Approximation

The next step is to estimate the distance to the centralized feature point. Since we have no extrinsic parameters, we can make an initial estimation by using a small angle approximation, as shown in Figure 5. From this, we are able to obtain our radius estimation along with estimating some intrinsic parameters.

The motion can be applied along either the X-axis or the Y-axis of the camera; in the example above, we started with a vertical motion. From Figure 5, we see that  $\Delta Y$  is the motion upwards and that  $\Delta\theta_x$  is the rotation of the camera. We use the ratio of these two values to estimate the distance of the feature point. We also return our first set of intrinsic parameters.

### C. Jacobian Estimation

The next step is to calculate a circular trajectory from the radius estimation along either the camera's X-axis or the Y-axis. For the best results, the motions need to alternate between the camera's X and Y direction to exercise both degrees of freedom. Otherwise, the intrinsic parameters will poorly estimate the unused degree of freedom, and the algorithm will not converge. Figure 4 shows how the estimation error will affect the pixel value of the image.

From the error along the respective pixel coordinate, we use the error model from (14) to compute the next distance estimate and the trajectory for the next motion. The algorithm's stopping criterion is based on the square of the pixel's error,  $E = \Delta u_1^2 + \Delta v_1^2$ . The algorithm stops when either the squared error falls below a threshold,  $E < \epsilon_J$ , or the change in the squared error falls below a different threshold,  $|E(j+1) - E(j)| < \epsilon_{conv}$ .



Fig. 7: Experimental setup during vertical and horizontal motion

## VIII. RESULTS

Experiments were conducted using the Barret Technology Whole-Arm Manipulation (WAM) robotic arm. Shown in Figure 7, the camera is attached to the end effector. The experiment was done with the small angle approximation motion being along the camera's Y-axis. Every motion afterwards alternates between camera's horizontal and vertical axes. Experiments were held in an environment where the intrinsic and extrinsic parameters were accurately calculated previously to test validity of the CMA.

The results in Figure 8 were taken from six different experiments all leading to convergence with well-conditioned Jacobian matrices along with the actual values of the parameters for the system. The first iteration is from the small angle approximation.

We can see how quickly the intrinsic parameters converge to their true values. From six different starting orientations, it took an average of 4 iterations to converge. The convergence criterion  $\epsilon$ 's were a bit conservative but with smaller tolerances, there should be less error with the intrinsic parameters. Interestingly, underestimating the radius causes an overestimation in the pixel width and height and an underestimation of the focal length.

It is important to notice that the first motion was a vertical motion and that every odd iteration  $\rho_v$  had less standard deviation. This was the same with even iterations for the  $\rho_u$  parameter. It is also important to realize that noise will affect the accuracy of the system but will not affect the stability for convergence. This can be seen in the standard deviation from the various experiments.

To determine whether the image Jacobian is well-

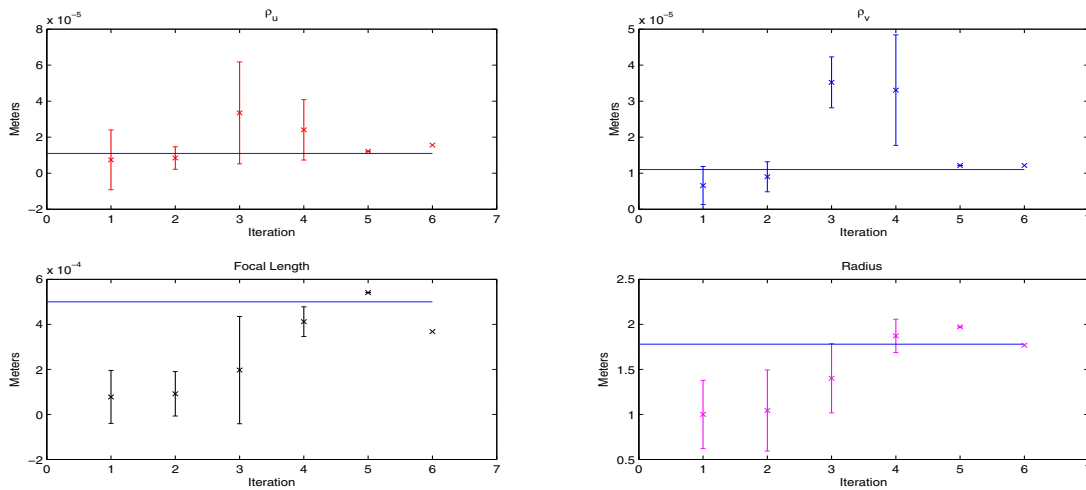


Fig. 8: Results from intrinsic parameters (solid blue is the ground truth)

conditioned, we used the condition number method. The average condition number was found to be 45. The acceptable ill-conditioned cutoff is around 100, while all Jacobians approaching 1 are better. [13] From this we can see that in an average of 4 iterations we obtain well-conditioned matrices.

## IX. CONCLUSION

Through the use of this centralized motion, we are able to fully extract all intrinsic and extrinsic parameters of an unknown camera system quickly and accurately. Using this method of information extraction, we plan on exploring object geometry identification and full implementation of SfM methods using centralization motion as the basis. These results have confirmed that a centralized motion around a feature point does indeed provide enough information to solve the unknowns of the system using only a single stationary point. Thus we have demonstrated a computationally cheap and useful "knowledge-less" SfM online calibration for consideration in future machine vision systems.

## X. ACKNOWLEDGMENTS

This work was supported in part by Sandia National Laboratories under Purchase Order 1098274.

## REFERENCES

- [1] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *Robotics and Automation, IEEE Transactions on*, vol. 8, no. 3, pp. 313–326, 1992.
- [2] B. Yoshimi and P. Allen, "Active, uncalibrated visual servoing," in *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, 1994, pp. 156–161 vol.1.
- [3] K. Hosoda and M. Asada, "Versatile visual servoing without knowledge of true jacobian," in *Intelligent Robots and Systems '94. 'Advanced Robotic Systems and the Real World', IROS '94. Proceedings of the IEEE/RSJ/GI International Conference on*, vol. 1, 1994, pp. 186–193 vol.1.
- [4] A. Ruf, M. Tonko, R. Horaud, and H. Nagel, "Visual tracking of an end-effector by adaptive kinematic prediction," in *Intelligent Robots and Systems, 1997. IROS '97., Proceedings of the 1997 IEEE/RSJ International Conference on*, vol. 2, 1997, pp. 893–899 vol.2.
- [5] N. Papanikolopoulos and P. Khosla, "Adaptive robotic visual tracking: theory and experiments," *Automatic Control, IEEE Transactions on*, vol. 38, no. 3, pp. 429–445, 1993.
- [6] N. Papanikolopoulos, B. Nelson, and P. Khosla, "Six degree-of-freedom hand/eye visual tracking with uncertain parameters," *Robotics and Automation, IEEE Transactions on*, vol. 11, no. 5, pp. 725–732, 1995.
- [7] E. Malis, "Visual servoing invariant to changes in camera-intrinsic parameters," *Robotics and Automation, IEEE Transactions on*, vol. 20, no. 1, pp. 72–81, 2004.
- [8] Y.-H. Liu, H. Wang, C. Wang, and K. K. Lam, "Uncalibrated visual servoing of robots using a depth-independent interaction matrix," *Robotics, IEEE Transactions on*, vol. 22, no. 4, pp. 804–817, 2006.
- [9] N. Andreff, R. Horaud, and B. Espiau, "On-line hand-eye calibration," in *Proceedings. Second International Conference on 3-D Digital Imaging and Modeling.*, 1999, pp. 430–436.
- [10] P. Corke, F. Spindler, and F. Chaumette, "Combining cartesian and polar coordinates in ibvs," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, 2009, pp. 5962–5967.
- [11] P. Corke and S. Hutchinson, "A new partitioned approach to image-based visual servo control," *Robotics and Automation, IEEE Transactions on*, vol. 17, no. 4, pp. 507–515, 2001.
- [12] J. Heller, M. Havlena, A. Sugimoto, and T. Pajdla, "Structure-from-motion based hand-eye calibration using l1 minimization," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, 2011, pp. 3497–3503.
- [13] P. Corke, *Robotics, Vision, and Control*. Manhattan, New York: Springer, 2011.