Image-Based Visual Servoing with Light Field Cameras

Dorian Tsai[1], Donald G. Dansereau[2], Thierry Peynot[1] and Peter Corke[1]

***Abstract*—This paper proposes the first derivation, implementation, and experimental validation of light field image-based visual servoing. Light field image Jacobians are derived based on a compact light field feature representation that is close to the form measured directly by light field cameras. We also enhance feature detection and correspondence by enforcing light field geometry constraints, and directly estimate the image Jacobian without knowledge of point depth. The proposed approach is implemented over a standard visual servoing control loop, and applied to a custom mirror-based light field camera mounted on a robotic arm. Light field image-based visual servoing is then validated in both simulation and experiment. We show that the proposed method outperforms conventional monocular and stereo image-based visual servoing under field-of-view constraints and occlusions.**

***Index Terms*—Visual Servoing; Computer Vision for Automation**

I. Introduction

VISUAL servoing (VS) is a robot control technique that makes direct use of visual information by placing the camera in the control loop. is widely applicable and generally robust to errors in camera calibration, robot calibration and image measurement [1]–[3]. Most VS techniques fall into one of two categories. Position-based visual servoing (PBVS) uses observed features and a geometric object model to estimate the camera-object relative pose and adjust the camera pose accordingly. Image-based visual servoing (IBVS) uses the observed features directly to estimate the required rate of change of camera pose. However, most IBVS algorithms are focused on conventional monocular cameras that inherently suffer from lack of depth information, narrow field of view constraints, and struggle with occlusions and specular highlights. Light field (LF) cameras, also known as plenoptic cameras, offer a potential solution to these problems. As a first step in exploring LF for IBVS, this paper considers the multiple views and depth information implicit in the LF structure. To the best of our knowledge, light field image-based visual servoing (LF-IBVS) has not yet been proposed.

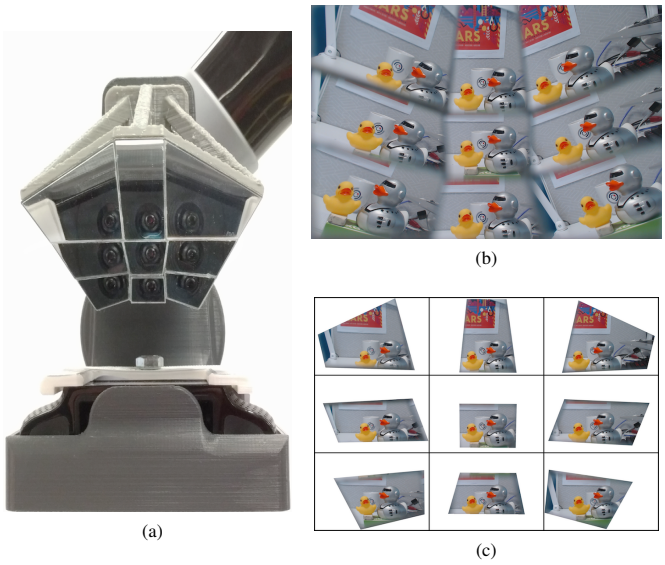


Fig. 1. (a) MirrorCam mounted on the Kinova MICO robot manipulator. Nine mirrors of different shape and orientation reflect the scene into the upwards-facing camera to create 9 virtual cameras, which provides video frame-rate light fields. (b) A whole image captured by the MirrorCam and (c) the same decoded into a light field parameterization of 9 sub-images, visualized as a 2D tiling of 2D images. The non-rectangular sub-images allow for greater FOV overlap.

The main contribution of this paper is the derivation, implementation and experimental validation of LF-IBVS. We derive image Jacobians for the LF. We define an appropriate compact representation for LF features that is close to the form measured directly by LF cameras. In addition, we take a step towards truly 4D plenoptic feature extraction by enforcing LF geometry in feature detection and correspondence. We validate our proposed method for LF-IBVS using both a simulated camera array and our custom LF camera adapter, shown in Fig. 1a, which we refer to as MirrorCam, mounted on a robot manipulator. Finally, we show that LF-IBVS outperforms conventional monocular and stereo IBVS, which can be considered a degenerate form of LF-IBVS, for objects occupying the same field-of-view and in the presence of occlusions.

The remainder of this paper is organized as follows. Section II provides some background, formulates the VS problem and explains the LF parameterization. Section III explains the derivations for LF image Jacobians, features, correspondence and the control system. Section IV describes our experimental setup with the MirrorCam. Section V shows our results, and

Manuscript received: September, 10, 2016; Revised December, 9, 2016; Accepted December, 30, 2016.

This paper was recommended for publication by Editor Francois Chaumette upon evaluation of the Associate Editor and Reviewers' comments. This research was partly supported by the Australian Research Council (ARC) Centre of Excellence for Robotic Vision (CE140100016). We also thank Steve Martin for helping to build the MirrorCam, and the other members of the ACRV for their insight and guidance.

[1]D. Tsai, T. Peynot and P. Corke are with the Australian Centre for Robotic Vision (ACRV), Queensland University of Technology (QUT), Brisbane, Australia {dy.tsai, t.peynot, peter.corke}@qut.edu.au

[2]D. Dansereau is with the Stanford Computational Imaging Lab, Stanford University, CA, USA. donald.dansereau@gmail.com

Digital Object Identifier (DOI): see top of this page.

provides a comparison to conventional monocular and stereo IBVS. Lastly, in Section VI, we conclude the paper and explore future work.

II. Background

LF cameras measure the amount of light travelling along each ray that intersects the sensor by capturing multiple views of a single scene [4]. In doing so, these cameras implicitly encode both geometry and texture, which allows for depth extraction. Conventional 2D images are thus replaced with 4D representations of rich visual information. There are several different LF camera architectures, with the most prevalent being the camera array [5], and the micro-lens array (MLA) [4]. Although LF cameras typically involve more complex calibration procedures than their conventional counterparts, LF cameras also offer extra capabilities. Table I compares conventional and LF camera systems for different capabilities and tolerances related to VS, given similar configurations, such as sensor size and number of pixels. Notably, stereo provides depth for a single baseline along a single direction (typically horizontally), but multi-camera and LF systems provide more detailed depth information. They can have both small and long baselines, and have baselines in multiple directions (typically vertically and horizontally). LF cameras have an advantage over conventional multi-camera systems for tolerating occlusions and specular reflections (or more generally non-Lambertian surfaces). This is largely due to the regular sampling, and because only LF cameras capture the refraction, transparency and specular reflections natively. As such, LF cameras can benefit from methods that exploit these capabilities [6].

Johannsen et al. recently applied light fields in structure from motion [7]. They derived a linear relationship using the LF to solve the correspondence problem and compute a 3D point cloud. They achieved an increase in accuracy and robustness, although their 3D-3D approach did not take full advantage of the 4D LF. Dong et al. focused on Simultaneous Localization and Mapping (SLAM), and demonstrated that an optimally-designed low-resolution LF camera allowed them to develop a SLAM implementation that is more computationally efficient, and more accurate than SLAM for a single high-resolution camera [8]. Dansereau et al. derived "plenoptic flow" for closed-form, computationally efficient visual odometry with a fixed operation time regardless of scene complexity [9]. Recently, Walter et al. used LF cameras to analyze specular reflection and detect features specific to specular reflections, which enabled robots to interact with glossy objects, and outperform their stereo counterparts [10]. These motivate the application of LF for robotics and LF-IBVS.

A. Image-Based Visual Servoing

IBVS uses the observed features directly to estimate the required change in camera pose rate (spatial velocity). IBVS makes use of an interaction matrix – more commonly, an image Jacobian, $\boldsymbol{J}$ – to map camera spatial velocity to the optical flow of points in the scene

$$\dot{\boldsymbol{p}} = \boldsymbol{J}(\boldsymbol{p}, {}^c\boldsymbol{P}; \boldsymbol{K})\boldsymbol{\nu}, \tag{1}$$

where ${}^c\boldsymbol{P} \in \mathbb{R}^3$ is the coordinate of a world point in the camera reference frame, $\boldsymbol{p} \in \mathbb{R}^2$ is its image plane projection, $\boldsymbol{K} \in \mathbb{R}^{3\times3}$ is the camera intrinsic matrix, $\boldsymbol{\nu} = [\boldsymbol{v};\ \boldsymbol{\omega}] \in \mathbb{R}^6$ is the camera's spatial velocity in the camera reference frame, which is the concatenation of the camera's translational velocity $\boldsymbol{v} = [v_x, v_y, v_z]^\intercal$ and rotational velocity $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]^\intercal$ in the camera reference frame.

The control problem is defined by the initial (observed) and desired image coordinates, $\boldsymbol{p}^\#$ and $\boldsymbol{p}^*$ respectively, from which the required optical flow

$$\dot{\boldsymbol{p}}^* = \lambda(\boldsymbol{p}^* - \boldsymbol{p}^\#)$$

can be determined, where $\lambda > 0$ is a constant. Combining both equations we can write

$$\boldsymbol{J}(\boldsymbol{p}, {}^c\boldsymbol{P}; \boldsymbol{K})\boldsymbol{\nu} = \lambda(\boldsymbol{p}^* - \boldsymbol{p}^\#), \tag{2}$$

which relates camera velocity to observed and desired image plane coordinates. However, it is not possible to uniquely determine the elements of $\boldsymbol{\nu}$ for a single observation $\boldsymbol{p}$. Typically we stack (2) for each of N image features,

$$\begin{bmatrix} \boldsymbol{J}(\boldsymbol{p}_1, {}^c\boldsymbol{P}_1; \boldsymbol{K}) \\ \vdots \\ \boldsymbol{J}(\boldsymbol{p}_N, {}^c\boldsymbol{P}_N; \boldsymbol{K}) \end{bmatrix} \boldsymbol{\nu} = \lambda \begin{bmatrix} \boldsymbol{p}_1^* - \boldsymbol{p}_1^\# \\ \vdots \\ \boldsymbol{p}_N^* - \boldsymbol{p}_N^\# \end{bmatrix} \tag{3}$$

and if $N \geq 3$ we can solve uniquely for $\boldsymbol{\nu}$

$$\boldsymbol{\nu} = -\lambda \begin{bmatrix} \boldsymbol{J_1} \\ \vdots \\ \boldsymbol{J_N} \end{bmatrix}^+ \begin{bmatrix} \boldsymbol{p}_1 - \boldsymbol{p}_1^* \\ \vdots \\ \boldsymbol{p}_N - \boldsymbol{p}_N^* \end{bmatrix}, \tag{4}$$

where $\boldsymbol{J}^+$ represents the pseudo-inverse of $\boldsymbol{J}$. Equation (4) is similar to the classical proportional control law for VS [1], except that we use the pseudo-inverse because we may have noisy observations forming a non-square matrix; the pseudo-inverse finds a solution that minimizes the norm of the camera velocity. It is important to note that VS is a local method based on a linearization of the perspective projection equation, but in practice it is found to have a wide basin of attraction. In later sections, we will generalize this approach for LF cameras by examining one possible representation of a LF feature and deriving a light field image Jacobian matrix for LF-IBVS.

B. Light Field Parameterization

We employ the relative two-plane parameterization in which a ray in homogeneous coordinates $\boldsymbol{\phi} = [s, t, u, v, 1]^\intercal$ is described by its points of intersection with two parallel reference planes; an s, t plane conventionally closest to the camera, and a u, v plane conventionally closer to the scene, with separation D [6], which is shown in Fig. 2. In this relative parameterization, u and v are expressed relative to s and t, respectively.

The rays emanating from a point in space, ${}^c\boldsymbol{P} = [P_x, P_y, P_z]^\intercal$ follow a pair of linear relationships [11], [12], as shown in Fig. 3

$$\begin{bmatrix} u \\ v \end{bmatrix} = \left(\tfrac{D}{P_z}\right) \begin{bmatrix} P_x - s \\ P_y - t \end{bmatrix}, \tag{5}$$

where each equation describes a hyperplane in 4D, $\mathcal{F}(s, t, u, v) \in \mathbb{R}^3$, and their intersection describes a plane $\mathcal{L}(s, t, u, v) \in \mathbb{R}^2$.

TABLE I
COMPARISON OF CAMERA SYSTEMS' CAPABILITIES AND TOLERANCES FOR VISUAL SERVOING

Systems	Perspectives	Field of View	Baseline	Baseline Direction	Aperture Problem	Occlusion Tolerance	Specular Tolerance
Conventional Cameras							
Mono	1	wide	zero	none	significant	no	no
Stereo	2	wide	wide	single	moderate	weak	no
Trinocular	3	wide	wide	three	moderate	moderate	no
Multiple cameras	n	wide	wide	multiple	minor	moderate	no
Light Field Cameras							
Array	n^2	wide	wide	multiple	minor	strong	yes
MLA [a]	n^2	wide	narrow	multiple	minor	strong	yes
MirrorCam [b]	n^2	narrow	wide	multiple	minor	strong	yes

[a] Based on n^2 pixels per lenslet
[b] Based on n^2 mirrors

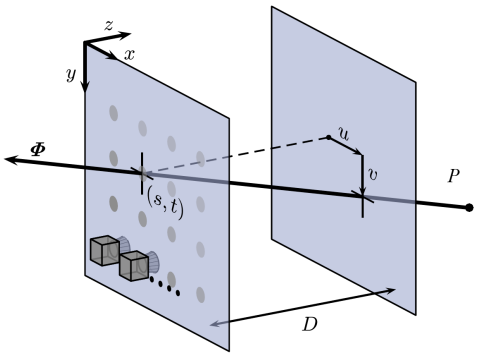


Fig. 2. The two-plane parameterization of light rays. Point $\boldsymbol{P}$ forms a ray Φ that intersects the two parallel planes. The intersecting points completely describe position and direction of the ray. By convention, the (s,t) plane is closer to the camera, and the (u,v) plane is closer to the scene, and taken relative to the (s,t) coordinates [6].

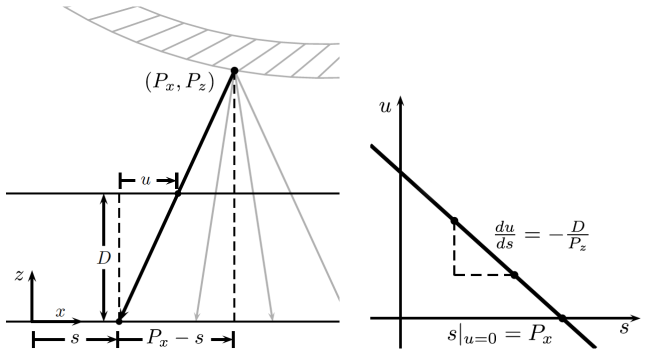


Fig. 3. (a) Light field geometry for a point in space for a single view (black), and other views (grey), whereby u varies linearly with s for all rays originating from ${}^c\boldsymbol{P}$. (b) The corresponding line in the s,u plane, having the slope w [6].

We define our LF feature with respect to the central view of the LF as $\boldsymbol{W} = [u_0, v_0, w]^\intercal$, where u, v is the direction of the ray entering the central view of the LF, i.e.

$$\begin{bmatrix} u_0 \\ v_0 \end{bmatrix} = \begin{bmatrix} u \\ v \end{bmatrix}_{s,t=0} = \left(\tfrac{D}{P_z}\right) \begin{bmatrix} P_x \\ P_y \end{bmatrix}. \tag{6}$$

The slope w relates the image plane coordinates for all rays emanating from a point in the scene. Fig. 3a shows the geometry of the LF for a single view of ${}^c\boldsymbol{P}$. As the viewpoint changes, that is, s and t change, the image plane coordinates vary linearly according to (5). In Fig. 3b, we show how u varies as a function of s, noting that v varies as a similar function of t. The slope of this line w, comes directly from (5), and is given by

$$w = -D/P_z, \tag{7}$$

noting that this slope is identical in the s,u and t,v planes. In the literature, this is refered to as the point-plane correspondence [6]. We exploit this aspect of the LF in the feature matching and correspondence process, described in Section IV-A. This representation is similar to the Augmented Image Space of [13] for perspective images where the image-plane coordinates are augmented with Cartesian depth. By working with slope, akin to disparity from stereo algorithms, we deal more closely with the structure of the light field.

III. LIGHT FIELD IMAGE-BASED VISUAL SERVOING

In this section, we derive the Jacobians, and describe how we exploit the LF for IBVS.

A. Continuous-domain Image Jacobian

Following the derivation for conventional IBVS, we wish to relate the camera's velocity to the resulting change in an observed feature $\boldsymbol{W}$ through a continuous-domain image Jacobian

$$\dot{\boldsymbol{W}} = \boldsymbol{J}_{\mathcal{C}}\boldsymbol{\nu}. \tag{8}$$

Differentiation of (6) and (7) yields

$$\dot{u_0} = D(\dot{P}_x P_z - P_x \dot{P}_z)/P_z^2, \tag{9}$$

$$\dot{v_0} = D(\dot{P}_y P_z - P_y \dot{P}_z)/P_z^2, \tag{10}$$

$$\dot{w} = D\dot{P}_z/P_z^2, \tag{11}$$

where u_0, v_0 and w are the feature positions and velocities with respect to the central camera frame.

We can write the apparent motion of a 3D point as

$$ {}^c\dot{\boldsymbol{P}} = -(\boldsymbol{\omega} \times {}^c\boldsymbol{P}) - \boldsymbol{v}, \tag{12} $$

yielding three components ${}^c\dot{\boldsymbol{P}}$ expressed in terms of ${}^c\boldsymbol{P}$ and $\boldsymbol{\nu}$. Substituting these expressions into (9)–(11) allows us to factor out the continuous-domain Jacobian

$$ \boldsymbol{J}_C = \begin{bmatrix} w & 0 & \frac{-wu_0}{D} & \frac{u_0 v_0}{D} & -D - \frac{u_0^2}{D} & v_0 \\ 0 & w & \frac{-wv_0}{D} & D + \frac{v_0^2}{D} & \frac{-u_0 v_0}{D} & -u_0 \\ 0 & 0 & \frac{-w^2}{D} & \frac{wv_0}{D} & \frac{-wu_0}{D} & 0 \end{bmatrix}. \tag{13} $$

While conventional image Jacobians require an estimate of depth, we note that $\boldsymbol{J}_C$ instead has slope w – an inverse measure of depth, which we can observe directly in the LF. The slope w is explicit in all columns of (13) except the last one, because the LF camera array spans both the $x-$ and $y-$ axes, and can therefore observe motion parallax about those axes. The optical flow for the final column is due to rotation about the optical axis, and is therefore invariant to depth. In contrast, depth is not explicit in the monocular image Jacobian for rotations about the x- and y-axes. Trinocular and multi-camera system image Jacobians would have similar depth dependencies to $\boldsymbol{J}_C$. Multiple views make parallax, and thus depth, observable in rotations about the x- and y-axes for the LF camera array. Additionally, $\boldsymbol{J}_C$ has a rank of 3, which implies that the stacked image Jacobian will be full rank with a minimum of 2 points for LF-IBVS, in contrast to a minimum of 3 image points for monocular image-based visual servoing (M-IBVS).

B. Discrete-domain Image Jacobian

In the discrete domain, we refer to i, j and k, l as the discrete versions of s, t and u, v, respectively. We observe our discrete-domain feature $\boldsymbol{M}$ as the discrete position and slope $\boldsymbol{M} = [k_0, l_0, m_x, m_y]^\intercal$, where $[k_0, l_0]$ are observations taken from the central view in i, j, and separate slopes m_x in the i, k dimensions and m_y in j, l. The general plenoptic camera is described by an intrinsic matrix $\boldsymbol{H}$ relating a ray $\boldsymbol{\phi}$ to the corresponding sample in the LF $\boldsymbol{n} = [i, j, k, l, 1]^\intercal$ as in

$$ \boldsymbol{\phi} = \boldsymbol{H}\boldsymbol{n}, \tag{14} $$

where in general $\boldsymbol{H}$ is of the form

$$ \boldsymbol{H} = \begin{bmatrix} h_{11} & 0 & h_{13} & 0 & h_{15} \\ 0 & h_{22} & 0 & h_{24} & h_{25} \\ h_{31} & 0 & h_{33} & 0 & h_{35} \\ 0 & h_{42} & 0 & h_{44} & h_{45} \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{15} $$

and the matrix $\boldsymbol{H}$ is found through plenoptic camera calibration [14]. However, we limit our development to the case of a rectified camera array, for which only diagonal entries and the final column are nonzero [6]. In this case h_{11} and h_{22} are the horizontal and vertical camera array spacing, in meters, and h_{33} and h_{44} are given by D/f_x and D/f_y, i.e. the inverse of the horizontal and vertical focal lengths of the cameras, expressed in pixels, scaled by the reference plane separation. The final column encodes the centre of the LF, e.g. for N_k samples in k, $h_{15} = \text{-}h_{11}(N_k/2 + 1/2)$ and $k = N_k/2 + 1/2$ is the centre sample in k. We also note that m_x and m_y encode the same information following the relationship

$$ m_x = \frac{h_{11} h_{44}}{h_{22} h_{33}} m_y. \tag{16} $$

We wish to express the image Jacobian of (8) in the discrete domain,

$$ \dot{\boldsymbol{M}} = [\dot{\bar{k}}_0, \dot{\bar{l}}_0, \dot{m}_x]^\intercal = \boldsymbol{J}_D \boldsymbol{\nu}, \tag{17} $$

where the observation is expressed relative to the LF centre, $\bar{k}_0 = k_0 + h_{35}/h_{33}$, $\bar{l}_0 = l_0 + h_{45}/h_{44}$.

From (14), we can relate the discrete and continuous-domain observations as

$$ u_0 = h_{33}\bar{k}_0, \;\; v_0 = h_{44}\bar{l}_0, \;\; w = \frac{h_{33}}{h_{11}} m_x = \frac{h_{44}}{h_{22}} m_y, \tag{18} $$

from which it is trivial to express the derivatives of the discrete observation in terms of the continuous variables:

$$ \dot{\bar{k}}_0 = h_{33}^{-1}\dot{u}_0, \dot{\bar{l}}_0 = h_{44}^{-1}\dot{v}_0, \dot{m}_x = \frac{h_{11}}{h_{33}}\dot{w}, \dot{m}_y = \frac{h_{22}}{h_{44}}\dot{w}. \tag{19} $$

Substituting the continuous-domain derivatives in (8), and (13) and discrete/continuous relationships in (18) into (19) allows us to factor out the discrete-domain Jacobian

$$ \boldsymbol{J}_D = \begin{bmatrix} \frac{m_x}{h_{11}} & 0 & \text{-}\frac{h_{33}}{h_{11}}\frac{\bar{k}_0 m_x}{D} & h_{44}\frac{\bar{k}_0 \bar{l}_0}{D} & \text{-}h_{33}\frac{\bar{k}_0^2}{D} - \frac{D}{h_{33}} & \frac{h_{44}}{h_{33}}\bar{l}_0 \\ 0 & \frac{m_y}{h_{22}} & \text{-}\frac{h_{44}}{h_{22}}\frac{\bar{l}_0 m_y}{D} & h_{44}\frac{\bar{l}_0^2}{D} + \frac{D}{h_{44}} & \text{-}h_{33}\frac{\bar{k}_0 \bar{l}_0}{D} & \text{-}\frac{h_{33}}{h_{44}}\bar{k}_0 \\ 0 & 0 & \text{-}\frac{h_{33}}{h_{11}}\frac{m_x^2}{D} & h_{44}\frac{\bar{l}_0 m_x}{D} & \text{-}h_{33}\frac{\bar{k}_0 m_x}{D} & 0 \end{bmatrix}. \tag{20} $$

IV. Implementation & Experimental Setup

In this section, we discuss the implementation details of our LF-IBVS approach, including how we exploit the LF structure for feature matching and correspondence. We then validate our proposed derivation of LF-IBVS using a closed loop control and the experimental setup described below.

A. Light Field Features

To our knowledge all prior work on LF features operates by applying 2D feature detectors to 2D slices in the u, v dimensions [7]. In this paper, we do the same. Our implementation employs Speeded-Up Robust Features (SURF) [15], though the proposed method is agnostic to feature type. However, as a first step towards truly 4D features, we augment the 2D feature location with the local light field slope, implicitly encoding depth.

Operating on 2D slices of the LF, feature matches are found between the central view and all other sub-images. Each pair of matched 2D features is treated as a potential 4D feature. A single feature pair yields a slope estimate, which defines an expected feature location in all other sub-images. We introduce a tuneable constant that determines the maximum distance between observed and expected feature locations, in pixels, and reject all matches exceeding this limit. We also reject features that break the point-plane correspondence discussed in Section II-B. By selecting only features that adhere to the planar relationship (5), we can remove spurious and inconsistent detections.

A second constant N_{MIN} imposes the minimum number of sub-images in which feature matches must be found. In the absence of occlusions, this can be set to require feature matches in all sub-images. Any feature passing the maximum distance criterion in at least N_{MIN} images is accepted as a 4D feature, and a mean slope estimate is formed based on all passing sub-images. N_{MIN} was set to 4 out of 8 sub-image matches for our experiments.

Feature matching between two light fields again starts with conventional 2D methods. A conventional 2D feature match finds putative correspondences between the central sub-images of the two light fields. Outlier rejection is performed using the M-estimator SAmple Consensus algorithm [16].

B. Mirror-Based Light Field Camera Adapter

There is a scarcity of commercially available LF cameras appropriate for robotics applications. Notably, no commercial camera delivers 4D light fields at video frame rates[1]. Therefore, we constructed our own LF video camera by employing a mirror-based adapter, based on previous work [17], [18]. We refer to this LF camera as the MirrorCam, which is depicted in Fig. 1a. The MirrorCam design, optimisation, construction, calibration, and image decoding processes are described in [19]. This approach splits the camera's field of view into sub-images using an array of planar mirrors, as shown in Fig. 1c. By appropriately positioning the mirrors, a grid of virtual views with overlapping fields of view can be constructed, effectively capturing a light field. We 3D-printed the mount based on our optimization, and populated it with laser-cut acrylic mirrors. Note that the LF-IBVS method described in this paper does not rely on this particular LF camera design, and applies to 4D light fields in general.

C. Control Loop

The proposed LF-IBVS control loop is depicted in Fig. 4. Notably, this control loop is similar to that of standard VS. Goal image features $f^* \in \mathbb{R}^3$ are compared to observed image features $f \in \mathbb{R}^3$. The pseudo-inverse of the Jacobian is computed, resulting in a camera spatial velocity $\boldsymbol{\nu}$, which is subsequently multiplied by a gain λ, as in (4).

Although velocity control is formulated in (4), the unoptimized algorithms to process the light fields from the MirrorCam currently operate at less than 0.1 Hz, which is impractical for velocity control. We therefore take a step-by-step approach and assume infinitesimal motion to convert $\boldsymbol{\nu}$ into a homogeneous transform ${}^c\boldsymbol{T}$ that we use to update the camera's pose. A motion controller moves the robot arm. After finishing the motion, a new image is taken and the feedback loop repeats until the image feature error converges to zero.

An important consideration in LF-IBVS is the feature representation, because the choice of feature representation in IBVS influences the Cartesian motion of the camera [20]. We have the option of computing the 3D positions of the points obtained from the LF; however, this would be no different from PBVS. Instead, we chose to work more closely to the native LF representation, working with projected feature position, augmented by slope. Doing so avoids unnecessary computation, and is more numerically stable as depth computation involves inverting slope.

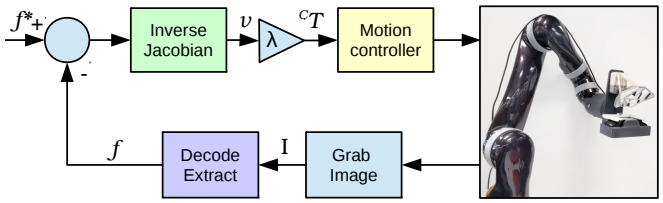


Fig. 4. The control loop for the VS system. Goal features f^* are given. Then f^* and f are compared, the $\boldsymbol{J}^+$ is computed, and camera velocity $\boldsymbol{\nu}$ is determined with gain λ and converted into a motion ${}^cT$. A motion controller moves the robot arm. After finishing the motion, a new image is taken and the feedback loop repeats until image features match.

We define the terminal condition for LF-IBVS as a threshold on the root mean square (RMS) error between all of the observed LF features and the goal LF features. We combine all of $\boldsymbol{M}$, and note that (u_0, v_0) are in meters, and $(\bar{k}_0, \bar{l}_0)$ are in pixels, but the slope w is unit-less. This issue can be addressed by weighting the components; however, for the discrete case, in practice we found that m_x and m_y had similar relative magnitudes. Additionally, we typically use a small λ of 0.1 in order to generate a smooth trajectory towards the goal view.

We found that the manufacturer's built-in inverse kinematics software became unresponsive for small pose adjustments[2]; therefore, we implemented a resolved-rate motion control method using a manipulator Jacobian to command camera spatial velocities to desired joint velocities [21]. We also changed the proportional, integral and derivative controller gains for all joints to $K_P = 2.0, K_I = 4.8$, and $K_D = 0.0$, respectively. With these implementations, we achieved sufficient positional accuracy and resolution to demonstrate LF-IBVS.

V. Results

A. Camera Array Simulation

In order to verify our LF-IBVS algorithm, we first simulated a 3×3 array of cameras. Four planar world points in 3D were projected into the image planes of the 9 cameras. A typical example of LF-IBVS is shown in Fig. 5. For this example, a small gain $\lambda = 0.1$ was used to enforce small steps and produce smooth plots as shown in Fig. 5a. The Cartesian positions and orientations relative to the goal pose converge smoothly to zero, as shown in Fig. 5b. Similarly, the camera velocity profiles in Fig. 5c converge to zero. Fig. 5d shows the image Jacobian condition number first increases, and then decreases to a constant lower value, indicating that the Jacobian becomes worse and then better conditioned, as the features move closer and then further apart, respectively. Together, these figures show the system converges, indicating that LF-IBVS was successful in simulation. Similar to conventional IBVS, a large λ results in a faster convergence, but a less smooth trajectory.

[1]Though one manufacturer provides video, it does not provide a 4D LF, only 2D, RGBD or raw lenslet images with no method for decoding to 4D.

[2]Limits were determined experimentally and confirmed by the manufacturer.

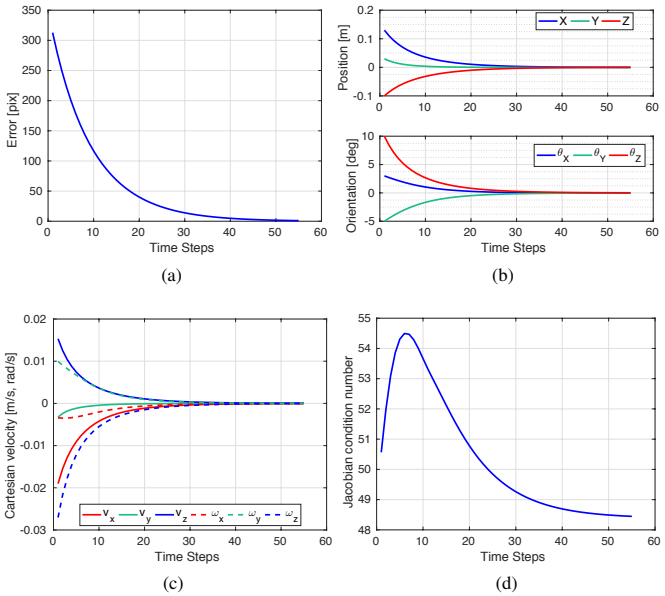


Fig. 5. Simulation of LF-IBVS, with (a) error (RMS of $f - f^*$) decreasing over time, (b) camera motion profiles relative to the goal pose, (c) Cartesian velocities, and (d) image Jacobian number for $\lambda = 0.1$. Error, relative pose and velocities all converge to zero.

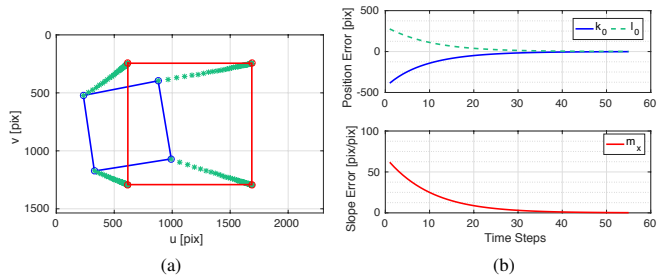


Fig. 6. Simluation of view (a) of the initial target points (blue), servoing along the image plane feature paths (green) to the target goal (red), and (b) the feature trajectory profile of $\boldsymbol{M} - \boldsymbol{M}^*$, corresponding to the top left corner of the target, which converges to zero.

Fig. 6a shows the view of the central camera, and the image feature paths as the camera array servos to the goal view. We see that the image feature paths are almost straight due to the linearization of the Jacobian. Fig. 6b shows the trajectories of the top-left corner of the target relative to the goal features, which also converge to zero. We note the slope profile matches the inverse of the z-position profile in the top red line of Fig. 5b, as it encodes depth.

For large initial angular displacements, we note that like regular IBVS, this formulation of LF-IBVS exhibited camera retreat issues. Instead of taking the straight-forward screw motion towards the goal, the camera retreats backwards, before moving forwards to reach the goal view.

B. Arm-Mounted MirrorCam Experiments

We also validated LF-IBVS using the MirrorCam mounted to the end of a Kinova MICO arm robot, shown in Fig. 1a. The robot arm and camera were controlled using the architecture outlined in Fig. 4. For the experiments, we first moved the

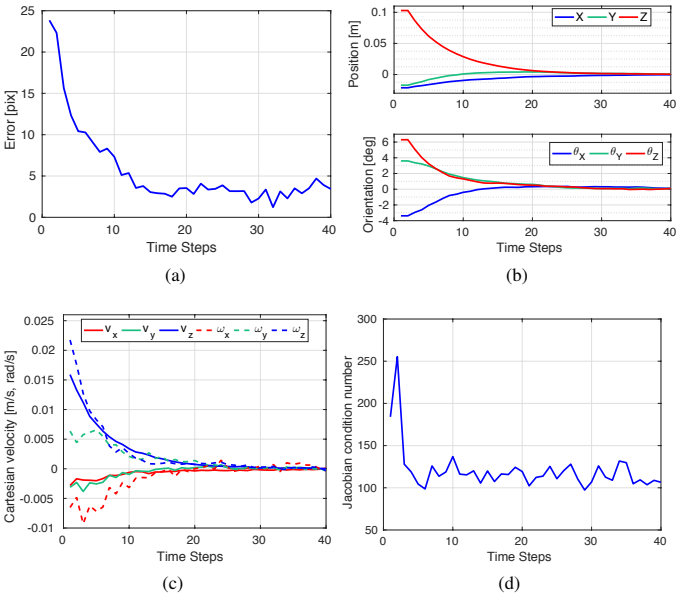


Fig. 7. Experimental results of LF-IBVS with MirrorCam on the robot arm, illustrating (a) the error (RMS of $\boldsymbol{M} - \boldsymbol{M}^*$) that converges after 20 time steps, (b) the camera motion profiles relative to the goal, which converge to zero, (c) the camera velocity profiles, which converge to zero, and (d) the image Jacobian condition number. Note the LF-IBVS outpeforms S-IBVS; the motion profiles are much smoother, and the velocities and condition numbers are an order of magnitude smaller than those from S-IBVS in Fig 8.

MirrorCam to the goal pose and recorded the goal view and its corresponding features. Then the camera was moved to an initial pose and made to servo back to the goal view using LF-IBVS. We tested the MirrorCam on a scene similar to Fig. 1b, with complex motion involving all 6 DOF from the initial pose.

Fig. 7 shows the performance of our LF-IBVS algorithm for the scene with $\lambda = 0.15$. Fig. 7a shows the error decreasing over time as the camera approaches the goal view, and converges after 20 time steps. We attribute the non-zero error to the arm's limited performance, which we address at the end of this section. Fig. 7b shows the relative pose of the camera to the goal in the camera frame converging smoothly to zero. Note that the goal pose is never the objective of LF-IBVS; rather, the image features captured at the goal pose drive LF-IBVS. Fig. 7c shows the commanded camera velocities also converge to zero. Fig. 7d shows the condition number for the image Jacobian, which decreases slightly as the system converges. We also note that despite only an approximate camera-to-end-effector calibration, the system converged, which suggests the robustness of the system against modelling errors.

LF-IBVS was compared against conventional M-IBVS and stereo image-based visual servoing (S-IBVS). Using the sub-images from the MirrorCam in Fig. 1c, we used the view through the central mirror for M-IBVS, and the two horizontally-adjacent views to the centre from the MirrorCam for S-IBVS. This was done to maintain the same FOV and pixel resolution. Implementations were based on [21], [22]. The average scene depth was provided for M-IBVS and S-IBVS to compute the Jacobian, although we note depth, or disparity

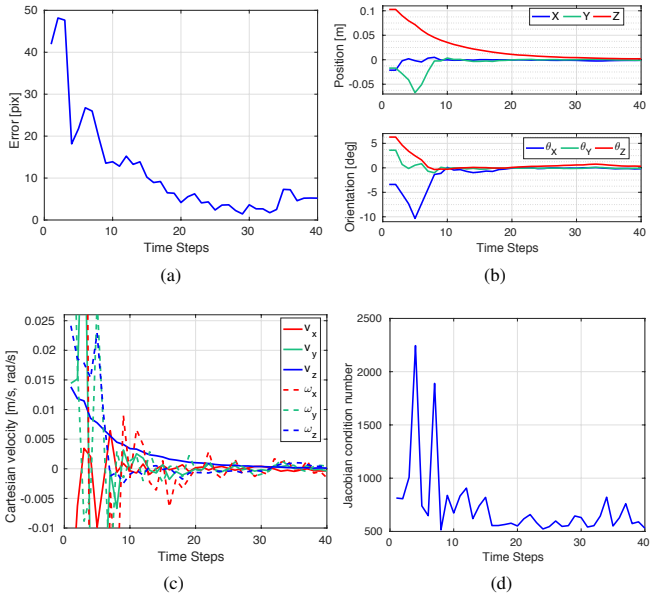


Fig. 8. Experimental results of S-IBVS with narrow FOV sub-images from the MirrorCam, on the robot arm, illustrating the performance in (a) the error (RMS of $\boldsymbol{p}-\boldsymbol{p}^*$) that eventually converges after 25 time steps, (b) the camera motion profiles relative to the goal that show an erratic trajectory at the start, (c) the camera velocity profiles that also vary greatly, and (d) the extremely large image Jacobian condition number.

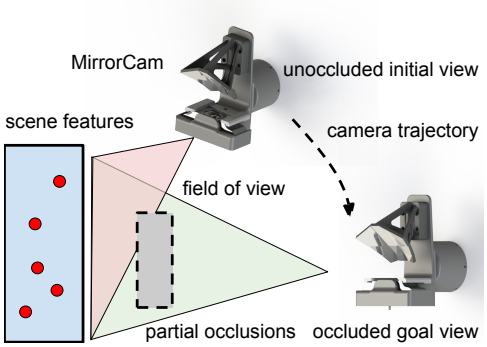
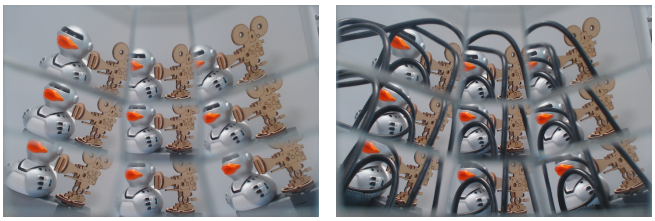


Fig. 9. Occlusion experimental setup, showing the initial view of the scene (red) with no occlusions, the camera trajectory that gradually becomes more occluded, and converging to the goal view with partial occlusions (green).



(a) (b)

Fig. 10. Occlusion experiments showing (a) the goal view with no occlusions from the MirrorCam, and (b) the goal view, partially occluded by a box of black wires. The arm was able to reach the partially-occluded goal view using LF-IBVS, but not M-IBVS or S-IBVS. Images shown are flipped vertically.

can be measured directly from stereo. All three IBVS methods were tested 10 times on the same goal scene and initial pose.

A typical case for S-IBVS is shown in Fig. 8. The image feature error is not uniformly decreasing at the start, but eventually converges after 25 time steps. The camera moves in an erratic motion at the start in the x- and y-axes, but still manages to converge to the goal pose, as seen in the relative pose trajectories and camera velocities in Fig. 8b and 8c. M-IBVS exhibited worse performance than stereo, to the extent that such erratic motion caused the robot to completely lose view of the goal scene. This is probably not because λ was too high for S-IBVS; smaller gains were tested for S-IBVS, but yielded the same poor performance.

Instead, we observe that the S-IBVS Jacobian condition number in Fig. 8d was an order of magnitude higher than LF-IBVS, producing an almost rank-deficient Jacobian; such a Jacobian becomes an inaccurate approximation of the spatial velocities, and yields erratic motion. We attribute this poor performance to the narrow FOV, and thus the lack of perspective change, which is required to differentiate rotation from translation, particularly about the x- and y-axes. In addition, the projected scale of the object being servoed against affects the performance of IBVS; smaller or more distant objects yield poorly-conditioned image Jacobians. These observations are not new or surprising [8]. However, they do suggest that LF-IBVS outperformed both of our constrained implementations of M-IBVS and S-IBVS, as LF-IBVS converged with a smooth trajectory regardless of the narrow FOV constraints of the MirrorCam.

Experiments with occlusions were also conducted using a series of black wires to partially occlude the scene. The setup is illustrated in Fig. 9 and 10. The goal, or reference image, was captured without the occlusions at a specified goal pose. An example image is shown in Fig. 10a. Next, the robot was moved to an initial pose, where the occlusions did not obscure the scene. Then the robot was allowed to servo towards the goal, along a path where the occlusions gradually obscured the goal view. The final goal image was partially occluded, as shown in Fig. 10b. M-IBVS, S-IBVS and LF-IBVS were run using the same setup. With the partially occluded views, M-IBVS and S-IBVS failed; whereas the LF-IBVS method servoed to the original goal pose.

Fig. 11 compares the number of features matched by LF-IBVS, M-IBVS, and S-IBVS in the occlusion experiment. Without any occlusions, we note that all three methods have a similar number of matched features at the goal view, although stereo and mono have slightly more matches than LF-IBVS throughout the experiment. This is likely because all 3 methods used similar 2D feature detection methods; however, our LF-IBVS approach also rejected those features that were inconsistent with LF geometry. With occlusions, M-IBVS fails at time step 5, when it is unable to match sufficient features. Similarly, the performance of S-IBVS quickly degrades at time step 10, as the occlusion covers most of the left view and significant portions of the right view. On the other hand, in the presence of occlusions, LF-IBVS has fewer matches than the unoccluded case, but still matches a consistent and sufficient number of features throughout its trajectory to converge. It is therefore apparent that LF-IBVS can utilize the LF camera's multiple views and baseline directions to handle partial occlusions. Trinocular and multi-camera systems may also benefit

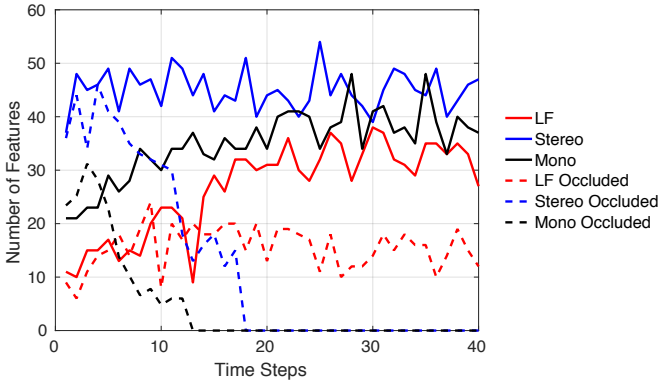


Fig. 11. Experimental results for number of features matched over time with occlusions (dashed), and without (solid), for LF-IBVS (red), S-IBVS (blue), and M-IBVS (black). Both stereo and monocular methods fail at time step 5 and 10, respectively, but LF-IBVS maintains enough feature matches to converge to the goal pose, which demonstrates that LF-IBVS is more robust to occlusions.

from the occlusion tolerance that we demonstrated, but would lack tolerance to specular highlights and other non-Lambertian surfaces as discussed in Table I.

VI. Conclusions and Future Work

In this paper, we have proposed the first derivation, implementation, and validation of light field image-based visual servoing. We have derived the image Jacobian for LF-IBVS based on a LF feature representation that is augmented by the local light field slope. We have exploited the LF in our feature detection, correspondence, and matching processes. Using a basic VS control loop, we have shown in simulation and on a robotic platform that LF-IBVS is viable for controlling robot motion. Further research into alternative feature types may address camera retreat and improve the performance of LF-IBVS.

Our implementation takes 5 seconds per frame to operate as unoptimized MATLAB code. The decoding and correspondence processes are the current bottlenecks. Through optimization, real-time LF-IBVS should be possible.

Our experimental results demonstrate that LF-IBVS is more tolerant than monocular and stereo methods to narrow FOV constraints and partially-occluded scenes. Robotic applications operating in narrow, constrained and occluded environments, or those aimed at small or distant targets would benefit from LF-IBVS, such as household grasping, medical robotics, and in-orbit satellite servicing. In future work, we will investigate other LF camera systems, how to further exploit the 4D nature of the light field features, and evaluate the performance of LF-IBVS in the presence of specular highlights and other non-Lambertian surfaces, where the method should strongly benefit from the light field.

References

[1] S. Hutchinson, G. Hager, and P. Corke, "A tutorial on visual servo control," *Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, 1996.

[2] F. Chaumette, "Potential problems of stability and convergence in image-based and position-based visual servoing," *Lecture Notes in Control and Information Sciences*, vol. 237, pp. 66–78, 1998.

[3] C. Cai, E. Dean-Leon, D. Mendoza, N. Somani, and A. Knoll, "Uncalibrated 3d stereo image-based dynamic visual servoing for robot manipulators," in *Intl. Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2013, pp. 63–70.

[4] R. Ng, M. Levoy, M. Bredif, G. Duval, M. Horowitz, and P. Hanrahan, "Light field photography with a hand-held plenoptic camera," Stanford University Computer Science, Tech. Rep., 2005.

[5] B. Wilburn, N. Joshi, V. Vaish, E. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz, and M. Levoy, "High performance imaging using large camera arrays," *ACM Transactions on Graphics (TOG)*, vol. 24, no. 3, pp. 765–776, 2005.

[6] D. G. Dansereau, "Plenoptic signal processing for robust vision in field robotics," Ph.D. dissertation, University of Sydney, Jan. 2014.

[7] O. Johannsen, A. Sulc, and B. Goldluecke, "On linear structure from motion for light field cameras," in *Intl. Conference on Computer Vision (ICCV)*, 2015, pp. 720–728.

[8] F. Dong, S.-H. Ieng, X. Savatier, R. Etienne-Cummings, and R. Benosman, "Plenoptic cameras in real-time robotics," *The Intl. Journal of Robotics Research*, vol. 32, no. 2, pp. 206–217, 2013.

[9] D. G. Dansereau, I. Mahon, O. Pizarro, and S. B. Williams, "Plenoptic flow: Closed-form visual odometry for light field cameras," in *Intl. Conference on Intelligent Robots and Systems (IROS)*. IEEE, Sept 2011, pp. 4455–4462.

[10] C. Walter, F. Penzlin, E. Schulenburg, and N. Elkmann, "Enabling multi-purpose mobile manipulators: Localization of glossy objects using a light-field camera," in *Conference on Emerging Technologies & Factory Automation (ETFA)*. IEEE, 2015, pp. 1–8.

[11] R. Bolles, H. Baker, and D. Marimont, "Epipolar-plane image analysis: An approach to determining structure from motion," *Intl. Journal of Computer Vision (IJCV)*, vol. 1, no. 1, pp. 7–55, 1987.

[12] D. G. Dansereau and L. T. Bruton, "A 4-D dual-fan filter bank for depth filtering in light fields," *IEEE Transactions on Signal Processing (TSP)*, vol. 55, no. 2, pp. 542–549, 2007.

[13] W. Jang, K. Kim, M. Chung, and Z. Bien, "Concepts of augmented image space and transformed feature space for efficient visual servoing of an eye-in-hand robot," *Robotica*, vol. 9, pp. 203–212, 1991.

[14] D. G. Dansereau, O. Pizarro, and S. B. Williams, "Decoding, calibration and rectification for lenselet-based plenoptic cameras," in *Intl. Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Jun. 2013, pp. 1027–1034.

[15] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-up robust features (SURF)," *Computer Vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[16] P. H. Torr and A. Zisserman, "Mlesac: A new robust estimator with application to estimating image geometry," in *Computer Vision and Image Understanding (CVIU)*. Elsevier, 2000.

[17] M. Fuchs, M. Kächele, and S. Rusinkiewicz, "Design and fabrication of faceted mirror arrays for light field capture," in *Computer Graphics Forum*, vol. 32, no. 8. Wiley Online Library, 2013, pp. 246–257.

[18] W. Song, Y. Liu, W. Li, and Y. Wang, "Light field acquisition using a planar catadioptric system," *Optics Express*, vol. 23, no. 24, pp. 31 126–31 135, 2015.

[19] D. Tsai, D. Dansereau, S. Martin, and P. Corke, "Mirrored Light Field Video Camera Adapter," Queensland University of Technology, Tech. Rep., December 2016.

[20] R. Mahony, P. Corke, and F. Chaumette, "Choice of image features for depth-axis control in image based visual servo control," in *Intl. Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2002, pp. 390–395.

[21] P. Corke, *Robotics, Vision and Control*. Springer, 2013.

[22] F. Chaumette and S. Hutchinson, "Visual servo control part 1: Basic approaches," *Robotics and Automation Magazine*, vol. 6, pp. 82–90, 2006.